The Business Impact of Inner Source and How to Quantify It

STEFAN BUCHNER, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
DIRK RIEHLE, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Inner source software development is the practice of using open source practices for firm-internal software development. Practitioner reports have shown that inner source can increase flexibility and reduce costs. Despite the potential benefits of inner source, there has been little research on its impact on businesses and their processes. To address this gap, we conducted a systematic literature review that identified which business processes are affected by inner source development, particularly within the accounting and management domain. Our review revealed the need for new dedicated community building processes within companies. In addition, we examined computational tools and techniques that can be used to measure inner source development. We found that existing tools and techniques are insufficiently suitable to manage inner source processes. Based on this, we propose research topics for future work on quantifying inner source.

CCS Concepts: • **General and reference** → **Surveys and overviews**; *Measurement*; *Metrics*; • **Software and its engineering** → Reusability; **Collaboration in software development**; • **Applied computing** → *Cross-organizational business processes*; **Business-IT alignment**; • **Social and professional topics** → *Project and people management*; *Transborder data flow*.

Additional Key Words and Phrases: Inner source, open source, internal open source, software engineering, software development, business processes, cost estimation, effort estimation, cost calculation, accounting, taxation, transfer pricing

ACM Reference Format:
Stefan Buchner and Dirk Riehle. 2023. The Business Impact of Inner Source and How to Quantify It. *ACM Comput. Surv.* 1, 1, Article 1 (January 2023), 27 pages. https://doi.org/10.1145/3611648

1 INTRODUCTION

Reacting flexibly to changes in rapidly evolving markets is crucial for companies that develop software as a core part of their products. Although agile software development makes it easier to adapt to customer needs, it often lacks general organizational flexibility, prompting companies to seek more agility on a larger organizational scale by adopting open-source principles for firm-internal software development [19].

Using open-source principles for internal development work is called inner source [60]. Rather than involving developers from outside the company, organizations apply the methods used in open-source development to enable people to develop and improve projects or modules that are internally (as far as possible) unrestrictedly available [30]. Inner source development is closely incorporated into internal review cycles with early and frequent feedback, enabling close collaboration across organizational boundaries [23].

Applying inner source to a company's development process and organizational structure can provide numerous advantages, even if not all of them are directly measurable. Inner source can lead to higher-quality software components due to input and reviews from previously unincorporated teams, better knowledge sharing, and increased employee satisfaction [11, 55, 60]. Moreover, inner source can facilitate component reuse, a characteristic of open-source software [52], which can make

Authors' addresses: Stefan Buchner, stefan.buchner@fau.de, Friedrich-Alexander-Universität Erlangen-Nürnberg, Open Source Research Group, Martensstr. 3, Erlangen, Germany, 91058; Dirk Riehle, dirk@riehle.org, Friedrich-Alexander-Universität Erlangen-Nürnberg, Open Source Research Group, Martensstr. 3, Erlangen, Germany, 91058.

platform development more efficient [55]. This, in turn, can lead to a more streamlined development process that reduces time-to-market and lowers costs, as several studies have demonstrated [13, 23].

The practical relevance of inner source is evident in the InnerSource Commons, an organization where both IT and non-IT companies come together to share and derive benefits from their experiences with inner source. According to a survey conducted by the InnerSource Commons, there is significant interest in inner source across a wide range of industry sectors. For instance, 37.8% of the respondents worked in the technology sector, while nearly 19.5% were from financial services, and 13.4% represented healthcare and pharmaceuticals sectors [1]. Due to its numerous developmental and organizational benefits, inner source is gaining popularity in both academia and industry [23]. However, it is not yet widespread, and there are various reasons for this within different business domains. For example, management's understanding of inner source or the developers attitude towards knowledge sharing, as identified by Edison et al. [23]. Recent research has primarily focused on the cultural and operational aspects of inner source. While examining the existing economic advantages, such as cost reduction [13], might be a useful motivator for the widespread adoption of inner source, only a few papers have examined its impact on business processes outside of engineering and attempted to measure and quantify it. Previous research suggests that the barriers to introducing inner source can be high, particularly because it is still unclear how exactly inner source creates strategic economic value for companies [23].

From a developer's perspective, inner source development may include contributing code across any organizational boundaries [12, 13]. This type of development occurs frequently and involves unpredictable flows of IP across internal organizational boundaries, making it difficult to measure, quantify and predict for economic purposes [10]. However, as this paper will demonstrate, existing tools and techniques are not easily applicable to the cross-boundary collaboration pattern of inner source.

Being able to economically assess and quantify inner source development can serve as the foundation for a wide range of business applications. For instance, Capraro [11] developed a basic model for measuring code contributions in inner source. Buchner and Riehle [10] demonstrated that economic assessment of inner source is feasible, although they focused on a taxation use case and a more generally applicable approach is necessary.

Failing to consider inner source quantification could result in significant harm, including illegal profit-shifting, as demonstrated in previous research [10]. This is due to cross-boundary code-flows between different taxable jurisdictions. The Organisation for Economic Cooperation and Development (OECD) has recognized software development as a significant challenge for tax officials [49, 50], and this is particularly relevant for inner source. Comprehensive measurement of inner source's IP-flows and impact on the business is crucial from a taxation perspective.

The purpose of this paper is to examine the impact of inner source collaboration on a company's operational and strategic processes and to suggest methods for measuring this impact to enhance effectiveness and innovation. Although our focus is on inner source development, we believe that our findings can be applied to improve a wide range of business processes that are related to or influenced by software development within organizations.

Moreover, this paper aims to address the background and limitations of current economic assessment tools and techniques for businesses, and why they are inadequate for the inner source use case. The paper examines both theoretical and practical aspects, discussing computational tools and techniques (such as algorithms, models, and methods) for measuring and estimating software-related efforts or costs, as well as exploring how inner source software development affects processes within the industry. The ultimate objective is to identify the connections between previously disconnected topics of theoretical effort estimation algorithms and business processes, particularly for their application within inner source. The end result is a comprehensive and integrated view of the topic of economic inner source assessment, encompassing multiple perspectives. Overall, this paper presents the following contributions:

- A survey of the effects of cross-boundary collaboration on businesses and their processes
- A survey of existing computational tools and techniques for measuring economic impact of inner source development
- An analysis showing how new computational tools and techniques can help solve existing challenges with inner source development and why no such algorithm exist yet
- A presentation of potential future research topics connected to economic inner source assessment.

The remainder of the article is structured as followed: Section 2 shows related work to economic assessment of inner source and explains the need for a review in more detail. Section 3 will give an overview of the research question of this paper as well as the methodology used to answer it. Section 4 will then present the results of the conducted systematic literature review, followed by an in-depth discussion in Section 5 on how themes identified during the literature review are connected as well as their implications to current research. Section 6 will lastly outline how this paper can contribute to future research followed by a conclusion.

2 RELATED WORK

Our goal is to understand the measurement of inner source development, how it affects business processes, and how to quantify it. We recognize that the economic assessment of inner source lies at the intersection of business processes and computational tools and techniques. Therefore, we will examine previous research in these two domains to gain a better understanding of the topic.

2.1 Previous inner source and business process research

Numerous papers have defined inner source (e.g. [14, 17, 45, 55, 58–60]), their benefits and challenges (e.g. [13, 55, 59, 60]) or took industry aspects directly into consideration (e.g. [27, 46, 55]). Edison et al. [23] published a literature review on inner source definitions, benefits, challenges, as well as research gaps. In addition to inner source specific research, many business processes and practices are also well defined within traditional economic research (outside of inner source).

For the accounting domain, handling cost calculations (e.g., full absorption costing [5]) or account models for platforms [38] have already been established. Basic algorithms and system designs for comprehensive accounting with computer systems were established as early as 1982/1996 [26, 44]. In the realm of taxation, the OECD has established fundamental approaches that, as previously mentioned, are not entirely appropriate for software development [49–51, 63]. While the OECD has outlined the fundamental principles of taxation, these principles are becoming increasingly problematic when applied to software development, including inner source.

Various methods exist to measure a company's success and aid in making strategic and operational decisions in management. For instance, risk management is commonly approached [22, 39, 57]. However, inner source development presents a challenge to traditional management approaches due to its cross-boundary collaboration pattern. Those were designed for the application within one business entity only (e.g. cost calculation). Consequently, measuring costs, staff development, and processes become more challenging in inner source. Furthermore, while software management practices are well-established, they face new challenges when applied to inner source development, as this paper will show. As a systematic literature review, this research will closely examine the existing literature, extending the work presented in this section.

2.2 Open topics

The concepts, advantages, and obstacles of inner source are well-established. However, research on measuring inner source collaboration is insufficient. Capraro identified inner source collaboration patterns [11], while Buchner and Riehle developed an algorithm to measure work time [10]. Initial research on inner source measurement for management accounting has also been conducted [31].

Edison et al. [23] noted a lack of metrics for measuring improvements resulting from inner source initiatives. While they identified several areas where further research is needed (such as management, inner source adoption, and methodologies), their proposal for measuring inner source impact was brief and did not address how companies can tackle measurement-related challenges.

Understanding the needs and basics for various inner source assessment metrics or tools helps answer the unsolved questions Edison et al. also proposed.

3 METHODOLOGY

This section gives an overview of the research question as well as the used methods.

3.1 Research question and goals

As previously presented, there has been limited research on the economic assessment of inner source development. Motivated by this gap, the present paper aims to address the following research question:

RQ: What is the economic impact of inner source on companies and how can it be quantified?

To address our research question, we investigate the impact of inner source on business processes. In doing so, we examine existing computational tools and techniques that measure software development. We classify these tools and techniques and analyze their suitability for application within the inner source domain.

3.2 Outline of the paper

The paper is structured around two main perspectives: business processes and computational tools and techniques. Each perspective is considered separately before bringing them together.

Figure 1 provides a detailed overview of the parts of the paper, including the methodology applied for each section, the results obtained, and the primary question addressed. These parts also serve as a roadmap for the paper and illustrate how they build upon one another. The first level in the figure (Part 1 and 2; Section 4) presents the findings of the systematic literature review, which includes an explanation of the codes and themes generated during thematic analysis. The second level (Parts 3a, 3b, 3c; Section 5) delves deeper into the artifacts resulting from the thematic analysis, examines their relationship, and discusses their implications for inner source. These insights can be utilized in future research to develop more robust economic inner source measurement models.

In detail, the goals of the parts are as follows:

Part 1: In this section, we are analyzing existing literature to identify business processes and explore how they are impacted by inner source software development.

Part 2: Here we focus on the computational tools and techniques that can be used to measure software development within businesses.

Part 3: We then demonstrate the interconnection between business processes, tools, and techniques by presenting:

- Part 3a) A high-level thematic map that illustrates the general relationship between the themes identified in our systematic literature review.

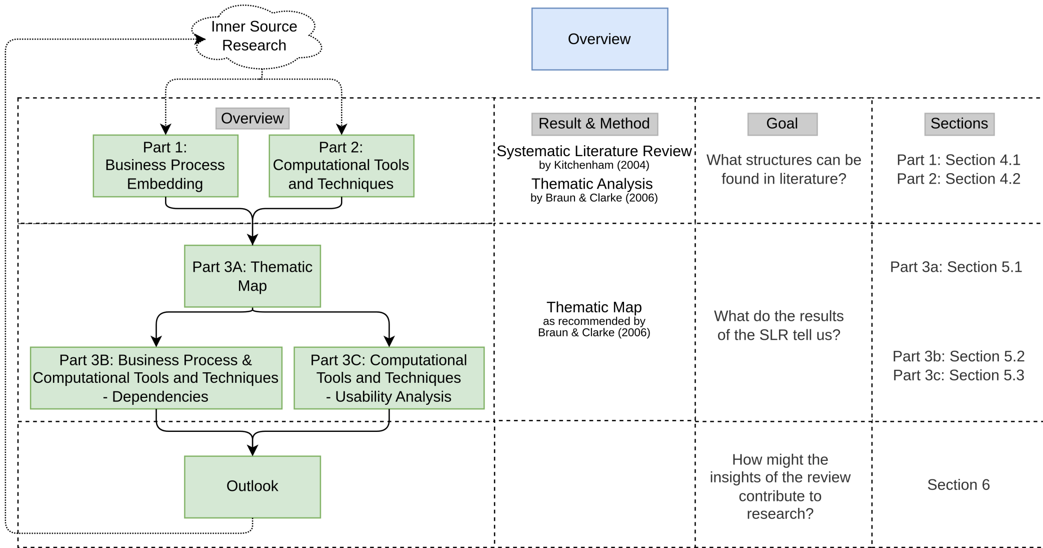

Fig. 1. Overview of the following sections, used methods, and results of the review

- Part 3b) An in-depth excerpt of the high-level thematic map that explains how existing tools and techniques can be used to measure or support business processes.
- Part 3c) Another in-depth excerpt of the high-level thematic map that highlights the suitability of certain tools and techniques for application within the inner source domain.

3.3 Methodological overview

3.3.1 Research process. To address the research question, two primary research methods were employed: The systematic literature review by Kitchenham [37] and thematic analysis by Braun & Clarke [9]. These methods were used in combination, as they complement each other. Kitchenham's systematic literature review does not provide a detailed explanation of the data extraction and synthesis process, which is the main focus of Braun & Clarke's thematic analysis. Both methods emphasized a non-linear/iterative approach. We followed this approach by repeatedly searching and filtering research (as suggested by Kitchenham) and then analyzing the data using Braun & Clarke's thematic analysis as part of Kitchenham's process. Therefore, we adopted the basic research process proposed by Kitchenham.

Figure 2 shows in detail the performed research steps and how the two approaches work together. The figure is divided into three columns, with the first two columns outlining the systematic literature review steps proposed by Kitchenham, while the third column illustrates the thematic analysis steps as per Braun & Clark.

The first column in our review outlines the planning steps we undertook prior to conducting our review. These steps included analyzing the need for a review, specifying the research question(s), writing a review protocol, and evaluating it. In the following section, we provide more details about the review planning process, while the research question was already presented in a previous section.

The second column shows the iterative steps of the conducted literature review: Identifying the research, selecting studies, assessing their quality followed by the data extraction and synthesis step. Kitchenham specified the data extraction and synthesis descriptively and not in all-detail. We

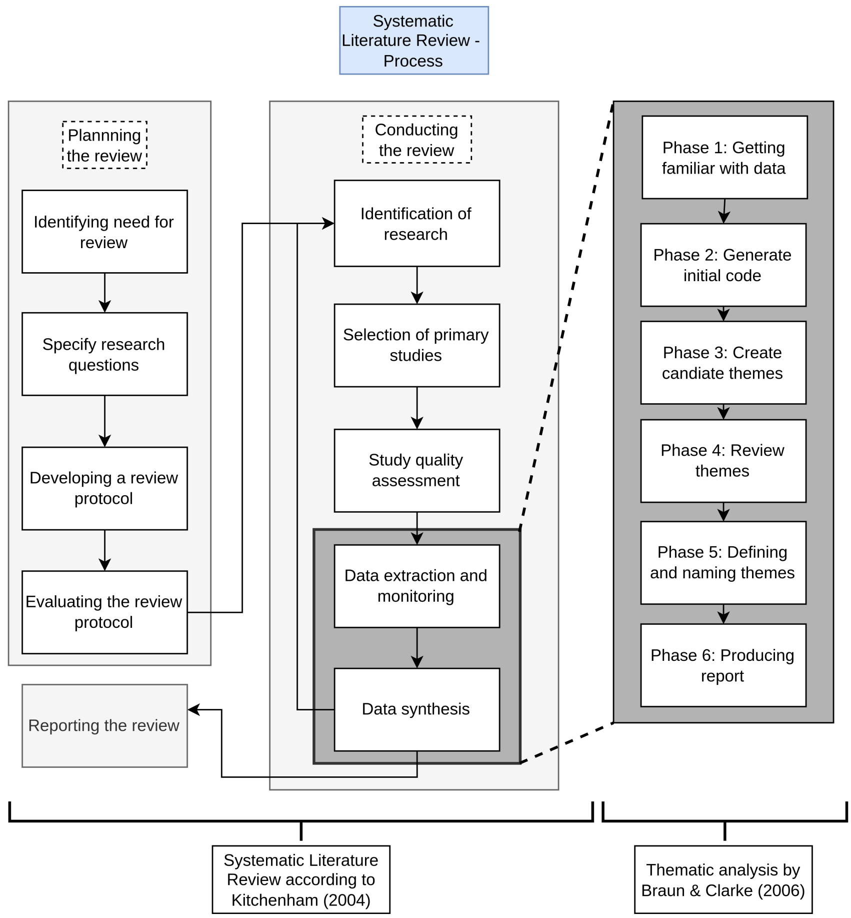

Fig. 2. Overview of the conducted research process based on Kitchenham [37] and Braun & Clarke [9]

used the thematic analysis framework from Braun & Clarke for these two steps of Kitchenham to be able to conduct our research.

The third column then explains in detail the six thematic analysis steps for analyzing the literature data: Getting familiar with the data, generating an (initial) code system (characteristics identified within the literature), followed by creating, reviewing, and naming themes (logically grouped codes) which are in the last step reported. Based on the created themes, their dependencies, and missing aspects the next iterations were conducted.

We conducted three major iterations to address our research question. In our first iteration, we identified that economic inner source assessment is influenced by two perspectives: business processes and computational tools and techniques. Although they are not completely independent from each other, they originate from different research backgrounds. The economic perspective is located within business-process research, while the tools and techniques are primarily located within the computer science domain. Therefore, we searched for relevant research papers in both domains using different search terms and analyzed them with different goals in mind. The following two sections will provide more detail about the steps we performed.

3.3.2 Research protocol. This section presents the details of the research protocol that we developed for the systematic literature review. Our protocol follows the order proposed by Kitchenham [37], and includes all the necessary information about the various steps we conducted. We created the protocol during the planning phase of our literature review.

Need for review: We already explained the need for a review (first step in the process proposed by Kitchenham) in detail within the previous sections. The main motivation is that it is still unclear how inner source creates business value and the lack of metrics therefore [11]. Previous research showed a low number of inner source tools and algorithmic procedures. Therefore, we conduct this review to identify how inner source impacts business processes and how existing tools and techniques can measure such impact.

Evaluation: As part of our systematic literature review, we followed the evaluation process described by Kitchenham. First, each author independently created a review protocol including research questions, keywords, qualitative criteria, and inclusion/exclusion criteria. We then evaluated each other's protocols before discussing and creating a mutually agreed upon version. We used the same approach for the research process itself, with each author conducting independent searches and proposing key findings that were then discussed together. Additionally, we solicited outside reviews from researchers in our research group who were not extensively versed in the topic. By following this evaluation process, we ensured the rigor and thoroughness of our literature review.

Databases: We used the databases Google Scholar, IEEE Xplorer, ACM Digital Library, Springer Link, Ebscohost, Wiley, and Scopus.

Identification process: The initial iteration of the review involved searching for fundamental concepts related to economic assessment and business processes. An essential component of the literature identification process was the use of forward and backward searches, also known as snowballing, which aided in the discovery of additional relevant information for answering the research question.

During the first iteration, we recognized that our research spans across different fields of science, including economics and computer science, and has been published in various journals with different contexts, using a range of keywords and phrases. Therefore, we decided to split our search and iterations into two main topics: the business process side of inner source and the tools and techniques side, which includes algorithms, applications/tools, methods, and models.

During our research, we found papers concerning different levels: Some were more theoretical / algorithmic-based and align well with the inner source principle, some look rather at the broader business process impacts. Classifying those was part of the thematic analysis by Braun & Clarke, which helped to identify literature gaps for the next iteration.

We realized that few inner source papers exist, particularly in the economic domain, where most methods are discussed without considering the software development method. Hence, during our selection and quality assessment, we looked through papers that do not directly relate to inner source to understand the current research state of measuring cross-boundary collaboration. This influenced our choice of inclusion and exclusion criteria.

Keywords: Following the two perspectives of business processes and computational tools and techniques, we used different keywords to identify relevant literature. Throughout our iterative process, we were able to expand our search terms by adding additional keywords. For both domains of review, we identified a common set of keywords related to software development approaches:

(Inner source OR open-source OR collaborative development OR cross-boundary collaboration OR cross border collaboration OR internal open-source OR software engineering OR software development OR DevOps OR agile OR platform)

That general search term was combined with the search term for each of the review domains. For the business process domain the search term was:

(Business processes OR management OR accounting OR controlling OR taxation OR transfer pricing OR organization OR businesses OR enterprises OR organizational principles OR organization forms OR absorption costing OR cost calculation OR project management OR risk management OR product management)

For the tools and techniques domain, we used the following search term:

(Software development OR programming OR ((cost OR effort) AND (calculation OR prediction OR estimation OR measuring OR quantifying OR computing OR calculating)) OR measurement OR KPI)

Quality criteria: The most crucial qualitative criteria for our study were the peer-review status of the papers and their publication in a recognized journal, conference, or as a well-defined (technical) report. For instance, algorithms must be comprehensible and reproducible, particularly in the case of reviewed tools and techniques. In addition to peer-review, we also considered papers originating from well-known organizations in the relevant domain, such as the OECD for transfer pricing [49, 50]. Moreover, we limited our scope to English-language papers. To assess papers, we used the rigor and relevance criteria proposed by Ivarsson and Gorschek [33] in their technology evaluation method: To assess the rigor of a paper, we evaluated (as proposed) whether the overall research context, study design, and validity (as well as threats to it) were discussed and to which extent. We evaluated practical relevance by examining the context and determining the degree of industry relevance (e.g. no student projects).

Inclusion and exclusion criteria: We additionally examined all papers which fulfilled the qualitative criteria towards their usefulness for our particular review. We have done this by defining inclusion and exclusion criteria. We included papers which

- are connected to inner source measurement in general
- present tools or techniques for measuring or predicting business (process) related aspects
- address problems within businesses and their processes connected to cross-boundary collaboration
- calculate work effort or costs on different levels (code-level, project-level, or business-level)
- give noteworthy insights useful for measuring inner source and affected processes

We explicitly excluded papers which

- have no thematic connection or usability within inner source, cross-boundary collaboration in general, or relate business processes
- presented tools and techniques that are non-repeatable. That especially affects machine leaning algorithms, mostly for cost calculation.
- presented tools and techniques which are not adaptable to inner source development (not able to assess cross-boundary IP-flow)

3.3.3 Thematic analysis. We completed the data extraction and synthesis step of Kitchenham's systematic literature review by using thematic analysis, as proposed by Braun and Clarke [9]. We followed the steps outlined in their methodology, as depicted in Figure 2. Our analysis encompassed two distinct perspectives: business processes and computational tools/techniques. To ensure comprehensive coverage, we identified important aspects, referred to as codes, and organized them into logical groupings, known as themes. This process was conducted independently for both domains.

Braun & Clarke [9] propose several approaches for thematic analysis. In this paper, we chose a deductive approach [9], starting from the previously presented research question (economic inner source assessment) and analyze/extract step by step the information gathered by the previous iteration. In our case, we started to look into inner source assessment options and their business process influence first. Next, we analyzed the applicability to the inner source domain.

Braun and Clarke distinguish between semantic and latent approaches to thematic analysis. While semantic analysis focuses on the explicit meaning of written aspects, latent analysis attempts to uncover the meaning behind the written words [9]. For our study, it is important to cover both aspects, as general patterns and algorithms can be analyzed by examining the initial goals of the papers. However, to fully understand the economic implications of inner source measurements, we also need to look beyond the surface-level meaning of certain papers. We have chosen to use a combination of both approaches, as only a few papers (especially those related to business processes) were originally written with inner source in mind. These papers need to be analyzed more deeply using a latent approach to extract their core aspects that are applicable to inner source.

4 SYSTEMATIC LITERATURE REVIEW

The following two subsections present the results of the systematic literature review, divided into two parts: Part 1 focuses on the business process view of inner source measurement, while Part 2 focuses on the computational tools and techniques view.

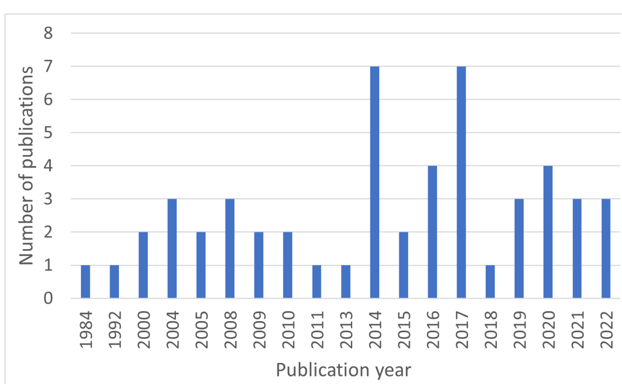

Fig. 3. Number of publications per year

Our systematic literature review (SLR) yielded 52 relevant papers across the business and computational tools and techniques domains. Our analysis identified 7 themes and 27 codes within these papers. While a majority of the papers were published within the last decade, the topic of inner source has garnered significant attention in recent years. Nevertheless, we also included older papers as they laid the fundamental economic and measurement foundations that still hold true today. Figure 3 displays the number of publications per year.

Table 1 provides a concise overview of the sources used to identify various themes. In the following sections, we will delve into the specific themes and codes, providing background information on their origins and significance.

Theme	# of Sources	Sources
A - Management processes	17	[3, 6, 13, 16, 21, 22, 31, 34, 39, 49, 55, 57, 58, 60, 62, 63, 65]
B - Accounting processes	21	[4, 5, 7, 10, 11, 15, 23, 30, 31, 36, 38, 43, 48–51, 53, 62–64, 67]
C - Development processes	17	[11, 13, 23–25, 28, 30–32, 40, 42, 49, 55, 58–60, 68]
D - Computation goals	12	[3, 4, 6–8, 16, 29, 36, 54, 56, 64, 67]
E - Algorithmic procedure	15	[3, 4, 6, 7, 10, 16, 20, 29, 35, 36, 47, 54, 56, 64, 67]
F - Data sources	19	[3, 4, 6, 10, 16, 20, 35, 36, 47]
G - Development context	3	[7, 8, 61]

Table 1. Overview of sources per theme

4.1 Part 1: Business process embedding

In our systematic literature review, we identified various business processes and related aspects that are impacted by inner source and its measurement. Figure 4 shows the resulting themes (A, B, C) and codes.

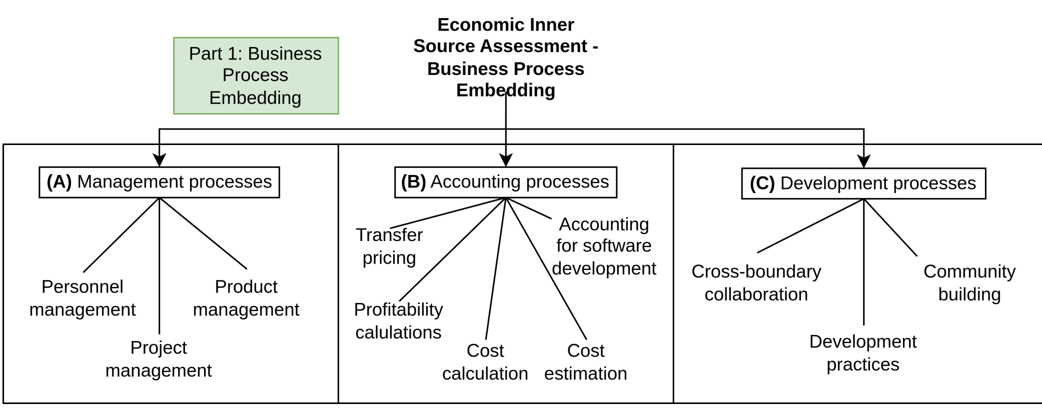

Fig. 4. Themes and codes on the business process perspective of measuring inner source

4.1.1 Theme A: Management processes. The first important domain influenced by inner source are management processes which need to be adopted to fit the cross-boundary collaboration pattern of inner source (see Section 5).

Personnel management. With the introduction of inner source, companies must carefully examine Human Resources (HR) processes. One change brought by inner source is the flexible way in which projects are organized and collaboration is facilitated. Rather than fixed projects, developers are now more flexibly assigned [13], which consequently affects HR processes (e.g. recruiting and workforce planning) and not only project management itself. Performance management is also impacted, as middle management often fears losing personal performance goals when making contributions to other organizations [55].

Product management. Managing product development with inner source can be challenging because contributions come from a large number of teams and departments, making it difficult to identify which department contributes to which extent [55]. This lack of clarity can make it harder for product management to track metrics accurately. Precise product management is crucial to managing a product throughout its life-cycle [21]. Inner source can also affect strategic product planning decisions, such as the introduction of software platforms [55]. Therefore, management needs to understand how inner source affects their product management processes and how to design their product within the new development environment to derive maximum benefits from it.

Project management. Cross-boundary collaboration patterns, such as inner source, have a significant impact on project management [31, 34, 65]. As contributions are made across organizational boundaries, companies must adapt their project planning and monitoring processes accordingly. Additionally, involving different legal units alters a project's risk management approach [22, 39, 57]. To account for the impact of inner source, key performance indicators (KPIs) [3, 16] and the Goal-Question-Metric (GQM) principle [6] are often used. It is crucial to ensure that these metrics and processes remain up-to-date when introducing inner source.

4.1.2 Theme B: Accounting processes. Besides management tasks various accounting processes are also impacted by inner source.

Transfer pricing. One important accounting process is the calculation of the value of IP contributions that flow across legal boundaries, also known as transfer pricing, which is used in taxation. The OECD has defined various well-established methods for calculating such a price [43, 49, 51, 53]. However, when using inner source, the choice of method is still unclear since inner source contributions frequently cross organizational boundaries [10, 48]. Organizations such as the OECD and United Nations (UN) have recognized new software development methods and their potential use as problem domains regarding profit shifting [49, 50, 63].

This problem was not only by companies and tax officials but also within the context of inner source businesses [30]. Researchers have already begun to develop initial approaches to solve this problem [10, 15, 31].

Profit calculation. Introducing inner source development can make accurately assigning value to individual contributing organizations challenging [10, 58], complicating profit calculation from an accounting perspective. Accurate profit and cost calculation are crucial for optimizing operational processes [62] and making informed strategic decisions. As companies adopt inner source, it's essential to adapt profit calculation processes to account for this new dynamic.

Cost calculation. Calculating costs for single departments, or cost centers, is a well-established task. However, in inner source, where code flows across organizational boundaries, it becomes more challenging to assign development costs to individual cost centers using traditional cost calculation approaches such as absorption costing [5].

Previous research [10, 11, 31] has proposed measurement and calculation methods to address this issue in inner source. However, a complete solution has yet to be presented.

Cost estimation. Predicting the future cost evolution is a crucial task in cost accounting, and it is influenced by inner source. Cost estimation plays a vital role in various steps throughout the product's life cycle [7], including maintenance [67]. Many papers have conducted cost estimation within agile environments, such as those by Bilgaiyan et al. [7], Karna et al. [36], and Usman et al. [64]. However, these models are not sufficient for use within inner source (as discussed in Section 5), and further research is necessary to adapt them accordingly.

Accounting for software development. In addition to identifying the general accounting processes that are influenced by inner source, we have also found that there are some initial approaches on how to account for software engineering in this context. The solutions we have come across include general approaches [4], dedicated inner source approaches [31], and those for platform organizations [38] where inner source plays a significant role. However, to date, no complete solution for accounting in inner source has been presented, and further work is needed in this area.

4.1.3 Theme C: Development processes. Next to dedicated management and accounting processes, we also identified aspects related to the software development process that are worth mentioning.

Cross-boundary collaboration. Inner source involves not only the artifacts produced by software projects, but also the way in which workflows and processes are organized to enable collaboration across organizational and legal boundaries [11, 23, 55, 59, 60]. As such, inner source becomes deeply integrated into the organizational structure of a company, not just its development teams.

It is important to note that not all traditional organizational forms (such as functional, divisional, or matrix structures) are ideal for developing cross-boundary projects. In fact, the functional organization can even be detrimental to software development [28], while a matrix organization may have limitations when it comes to large-scale organizations [25].

There are some existing organizational forms that are designed to support the development of complex products or systems, such as project-based or platform-based organizations [32, 40, 55]. However, there is still no ideal solution for organizing large-scale, high-frequency development work like inner source, even with the use of agile development methodologies [42] and existing best practices and guidelines (such as those outlined by Smite et al. in 2017 [68]).

Development practices. Inner source is deeply integrated into businesses' software development practices, largely due to the high-frequency, peer-review aspect that is adopted from open-source development [13, 24, 58]. Adopting inner source requires companies to rethink their code review processes, how they organize documentation, and how they handle contributors [59]. Inner source also impacts engineering organization and processes in general, such as DevOps [66].

Community building. Inner source requires new processes to be introduced, particularly for community building. To be successful, inner source relies on building communities within the company, which can be achieved through new exchange platforms [59]. It is also important for companies to exchange best practices with other companies that are executing inner source, such as through the InnerSource Commons [2]. Incentivization schemes [13, 18, 59] and related processes are essential for building communities and leveraging inner source within the company.

4.1.4 Concluding business process perspective. In Part 1, we have demonstrated that inner source has an impact on various business processes, including management, accounting, and community building. Inner source necessitates the introduction of new development processes, and as a result, established processes such as personnel, project, and product management need to be adjusted to maintain their effectiveness. Similarly, accounting processes, such as transfer pricing, profit calculations, and cost estimation, require modifications to adapt to the inner source model. Unfortunately, there is a lack of tools and principles for accurate accounting in inner source.

One of the key challenges we identified is the potential for inaccurate cost and profit calculations due to the cross-boundary collaboration pattern of inner source. These calculations are crucial for many strategic and operational decision-making processes within companies. The importance of cost calculation in business is emphasized by the International Federation of Accountants (IFAC):

> "Costing is inextricably linked to the organization's flow of resources to produce goods and services. The more accurately a costing model or system represents the operational flow of resources within an organization, the more clarity decision-makers will have in using cost data."
> [62]

The accurate measurement of cost and profit flow between organizational units is essential to fully realize the benefits that inner source offers.

4.2 Part 2: Computational tools and techniques embedding

Part 1 provided an overview of how the introduction of inner source can impact business processes. In Part 2, we will examine existing computational tools and techniques that businesses can use for economic assessment.

4.2.1 General aspects. The purpose of this paper is not to provide a comprehensive review and classification of all software effort estimation algorithms, tools, methods, and models. Instead, the aim is to identify and categorize computational tools and techniques that can be generally applied to inner source software development and can help solve business process-related challenges.

We classify a tool or technique as generally suitable for solving inner source challenges if it involves little to no manual work and is therefore computable and reproducible. We exclude manual work because certain business processes, such as taxation, require reproducible and extensively documented decision-making. Moreover, including non-computable tools and techniques is not a suitable option for inner source due to its high-frequency nature (code contributions are made by the minute). To be suitable for our analysis, the tools and techniques must be able to identify individual contributions or evaluate cross-boundary collaborations, as most of the problems with existing business processes stem from inner source characteristics.

It's important to note that this paper is not an accurate estimation of which topics are being researched more frequently or recently. Instead, we focus on the general algorithmic applicability to the inner source pattern. In addition, we take into consideration reviews, as they present the most important findings concisely.

Many of the reviews we identified focus on a specific methods, technologies, or approaches such as agile development or neural networks. We conducted a thorough review of relevant tools and techniques until further reviews did not yield any new insights related to measuring inner source.

Figure 5 outlines the themes (D, E, F, G) and corresponding codes identified during our thematic analysis. Detailed background information on the origin and motivation of this coding will be presented in the following sections. Table 2 maps the identified papers to their codes and themes, in addition to Figure 5. While some papers stated their approach, others required deeper analysis to identify their methods or classifications. For instance, some approaches were found through literature reviews, while others required examination of the calculation goal.

4.2.2 Theme D: Computation goals. Our research revealed that existing tools and techniques have been developed to serve various computational objectives. Some are specifically designed for management purposes, while others prioritize historical or predictive calculations. It is important to classify these tools and techniques in order to determine their suitability for the different types of business processes discussed earlier. A detailed discussion of this classification will be presented in Section 5.

Measuring for management. Many of the tools and techniques designed for software engineering are intended for various management tasks, such as calculating frequently used measurements like KPIs [3, 16], implementing the GQM model [6], or determining the contribution of developers to development [29]. In inner source, management metrics can be used to address the fear of middle managers not meeting their performance goals, as previously discussed [55].

Measuring cost/effort. A wide range of tools exists for measuring effort in software engineering, serving various purposes such as optimizing production processes [4], monitoring development, identifying bottlenecks, and future planning [29]. The focus of these tools is often on calculating

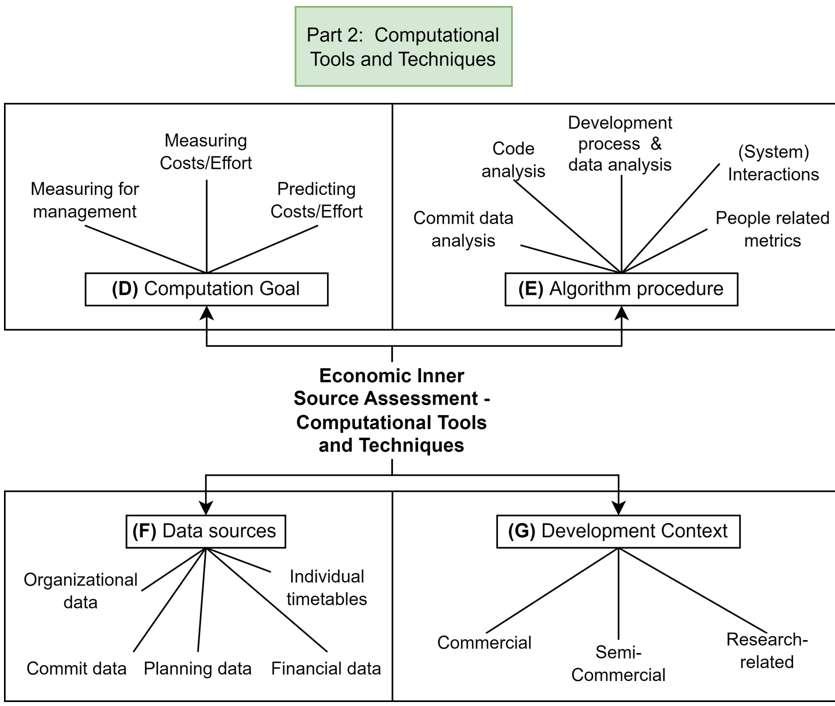

Fig. 5. Themes and codes on the computational tools and techniques perspective of measuring inner source

historic data, and different approaches have been used to achieve this goal, such as analyzing system interactions [29] or classifying work-time (part-time/full-time) [56].

Predicting cost/effort. We also identified tools and techniques designed to make predictions regarding software development cost and effort. While some of these tools have a more general focus [36], others were developed for specific purposes, such as maintenance [67], agile development [7, 64], or for use with open-source software [54]. Some of these tools and techniques have a long history, dating back to the 1980s with the development of one of the first cost estimation models, COCOMO [8].

4.2.3 Theme E: Algorithmic procedure. The tools and techniques used in the literature encompass a wide range of procedures to conduct their calculations. Many of the identified articles relied on intuitive calculations for aspects that are easy to quantify, while allocating less emphasis on quantifying social impact factors. In the following sections, we will delve into the suitability of specific approaches for their application within inner source.

Code analysis. One possible method for assessing inner source is analyzing the written code, which can involve either the code currently under development (code committed) [4] or more general metrics based on the final product's source code [67].

Commit data analysis. In addition, it is possible to analyze the commit data beyond the code that is generated during the development process. The approaches we identified are mostly based on mathematical calculations using lines of code and timestamps [10, 47]. This approach is especially suitable for inner source development and has already been applied to it [10], as commit data represents the lowest logical level of IP contributions made.

Table 2. Mapping between papers and the identified aspects (codes and themes)

		Computation Goal			**Algorithmic procedure**							**Data sources**				
Paper	Type	Management measuring	Measuring Effort	Predicting Effort	Code analysis	Commit analysis	People related metrics	System interactions	Development process & data analysis	Machine Learning	Mathematical approach used	Organisational data	Commit data	Planning data	Financial data	Time tables
[7]	R			x						x	x			x	x	
[35]	A		x								x			x		
[67]	R			x	x	x	x	x	x		x		x			
[36]	A			x					x		x			x		
[47]	A		x			x	x				x		x			
[4]	A		x		x			x					x		x	x
[56]	A		x			x	x				x		x			
[54]	A			x			x				x	x		x		
[29]	A	x	x					x					x			
[20]	T					x			x			x	x			x
[3]	M	x				x			x				x	x	x	
[16]	M	x				x	x		x			x	x	x	x	x
[6]	M	x					x		x			x		x	x	x
[64]	R			x		x	x			x	x	x	x	x		
[8]	A			x							x			x		
[10]	A		x			x					x	x	x		x	

A=Algorithm; M=Metric; R=Review, T=Toolset

People-related metrics. We also identified the use of people-related information as useful for estimating inner source. This includes information such as whether a person is part- or full-time [47] or the developer's experience [54]. This information may be usable for people-related management processes, such as within KPI/GQM calculations [3, 6, 16].

(System) interactions. Another method is to utilize interactions with different internal systems within businesses [4, 29] and base further calculations on the measured interactions, sometimes called activity-based [67].

Development process & data analysis. Moreover, it is possible to base measurements on insights into the development process itself, such as sprints [36]. This information can also be integrated into tools such as GrimoireLab [20], which is already used for analyzing inner source. Additionally, management calculations often rely on measuring these processes.

It is worth mentioning that a wide variety of mathematical approaches were used in the studies reviewed. Linear regressions were employed in most cases (e.g., [10, 47]) to varying degrees, and statistical analyses were also utilized (e.g., [10, 35, 56, 67]). These papers were additionally highlighted in Table 2 to indicate the use of mathematical methods.

We will not look closer into the vast array of machine learning algorithms and concepts available, as they are typically developed for specific purposes, and their transferability must be evaluated independently in each case.

4.2.4 Theme F: Data sources. In our review, we found suitable data sources for economic assessments of inner source. These sources can be particularly useful for newcomers to inner source measurement, providing an overview of the necessary data for accurate measurement.

Organizational data. Organizational data is crucial for many business processes that are influenced by the cross-boundary collaboration pattern of inner source (see Part 1). Having accurate and easily accessible organizational data helps companies adjust their processes to fit the inner source paradigm, thereby avoiding unnecessary risks. Such data can include the company's structures, such as teams and hierarchies [16], as well as employee numbers [20].

Commit data. Commit data is a vital source of information for measuring software development, as a large number of tools and techniques rely on it. Therefore, commit data is necessary for measuring inner source development. Commit data is one of the most critical data sources for inner source assessment, and it has already been used in dedicated inner source measurement research [10, 12, 31].

Planning data. Some identified papers have used planning data related to software projects, such as start and end dates [3], use-cases for analyzing functions and story-points [35], or sprint data [36]. However, these have not played a major role in the research dedicated to inner source that we identified.

Financial data. Although it is not often explicitly mentioned, financial data is an essential data source. As many of the tools and techniques directly or indirectly calculate costs or profits (e.g., [10]), the use of financial data is necessary.

Individual timetables. Our review also showed that data from various software systems used within companies can prove useful in inner source assessment. We identified techniques that rely on the time worked by individual people [6, 16, 47] or that require such data as additional input [4]. The Grimoirelabs tool [20] integrated meeting and communication data (Slack, Telegram, E-Mails) as well as data from ticket systems (like Jira) for their analysis.

To summarize, the various data sources we presented can be used individually or in combination to perform comprehensive inner-source analyses and adapt business processes accordingly. While some dedicated measurement approaches for inner source have utilized single sources, a comprehensive tool has not yet been presented.

4.2.5 Theme G: Development context. Although it plays a minor role in addressing inner source measurement challenges, the context in which the tools and techniques were developed is a noteworthy aspect that we were able to identify.

Commercial. Some tools and techniques (which we have reviewed to a lesser extent) were developed within a commercial environment. For example, Price System developed a parametric cost estimation system for hardware development in the 1960s and 70s [61]. However, since commercial systems are usually not freely available, they are not the focus of our research.

Semi-Commercial. Other tools and techniques were developed in a commercial context but are publicly available through research publications or books. Examples include the Cocomo model [8] and (Wideband) Delphi [7, 8].

Research-Related/Non-Commercial. The majority of the reviewed papers (not previously mentioned) originate from research or non-commercial contexts, although they might be used in commercial contexts.

5 PART 3: THEMATIC MAP

This section brings together Parts 1 and 2, identifying the relationship between the business processes affected by inner source and existing measurement approaches for software development. Part 3 is split into three sub-parts (3a, 3b, 3c), with 3a providing an overview and 3b and 3c analyzing the most important relationships and implications in-depth.

5.1 Part 3a: Thematic map

In the following, we will present the thematic map created within the research process as proposed by Braun & Clarke within Step 3 of the thematic analysis (See Figure 2). Braun & Clarke suggest creating a mind map (also called a thematic map) during the research. Consequently, all the connections between the themes and codes and the following in-depth discussions are based on the already presented results of the SLR and related literature (Section 5).

Figure 6 presents a high-level thematic map that depicts the themes in the two review domains (business processes, tools and techniques) and the general connections between them. In the following sections (Part 3b and 3c - Sections 5.2 and 5.3), we will examine these themes in more detail. Since not all themes are equally related to each other, we have categorized the connections into different types to make them easier to understand.

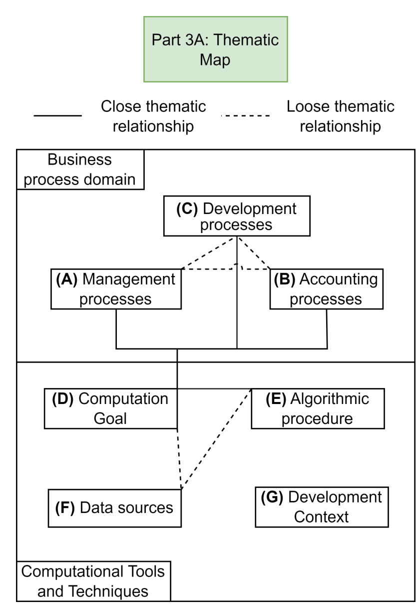

Fig. 6. Thematic map showing theme relationships

Close connections: Themes that are closely connected share codes that are somehow related to each other. This means that these themes have a similar way of working, solve similar problems, or have similar logic behind their classification. We have identified two main connections between the themes. Firstly, all business processes are related to the goals of the identified tools. Secondly, the way the tools and procedures work is closely related to their goals, as these tools and techniques are designed to solve problems that occur in similar business situations.

Loose connections: Themes with a loose connection are related to each other, but their processes, tools, and techniques do not impact each other in a centrally important way for the economic inner source assessment. For example, the computational goals (Theme D) and algorithmic procedures (Theme E) of tools and techniques are influenced by the available data sources (Theme F), but are not of central importance to them with regards to inner source measurement. The same is true for the business processes influenced by inner source (Themes A to C).

No connections: Theme G has no connection to any of the other themes. This is because the development context of a tool or technique (if it is a commercial product or a research paper) does not play an important role in solving the inner source business measurement problems.

In particular, themes that are closely related are of particular interest in this paper. They are discussed in more detail in the following sections.

5.2 Part 3b: Business process and tools & techniques dependencies

The following section details the link between the identified business processes and tools and techniques. Figure 7 displays four boxes, with the top three representing the business processes impacted by inner source (Themes A to C), and the bottom representing the objectives of the tools and techniques analyzed (Theme D). The figure illustrates the interconnection between the themes in both domains.

5.2.1 Dependency analysis. After reviewing the literature, it became apparent that the goals of computational tools and techniques (Theme D) cannot be entirely separated from the business processes affected by inner source (Themes A to C). This is due to the fact that the majority of the analyzed tools and techniques were originally designed to serve specific business purposes, such as calculating maintenance costs or estimating work time. As a result, these tools and techniques are intertwined with the business processes that they support.

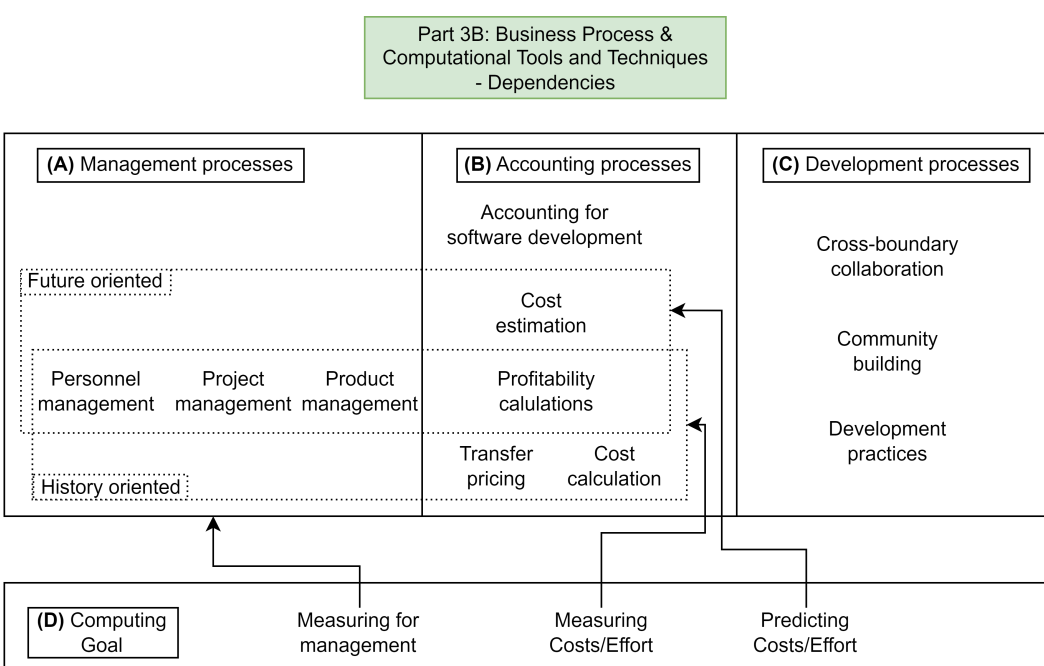

Fig. 7. Overview over the connections between the codes connecting the business and algorithmic view

Figure 7 illustrates that some processes in software development focus exclusively on either historic calculations (e.g., transfer pricing, cost calculation, Theme B) or estimations/predictions (e.g., cost estimations, Theme B), while others require both (e.g., profitability calculations in Theme B, or personnel management, product management, project management in Theme A).

Tools and techniques aimed at predicting cost or effort (Theme D, see Table 2) are best suited for prediction-oriented processes, such as management processes and profit/cost estimations in accounting. Conversely, tools and techniques designed to measure historic cost or effort are better suited for history-oriented business processes, like cost calculation and transfer pricing.

Additionally, tools and techniques classified under the "Measuring for management" code (Theme D) are well-aligned with the management-related processes of inner source (Theme A). For example, KPI calculations are more suitable for addressing the project management challenges of inner source.

These findings highlight the importance of measuring software development, particularly in the context of inner source. Inner source development impacts a wide range of business processes (Theme A to C), and being able to accurately measure it using the proposed tools and techniques (Themes D to G) can help address various challenges associated with inner source development.

5.2.2 Business and research implications. After conducting a thematic analysis, we found that inner source has a significant impact on various strategic and operational business processes beyond software development, such as accounting (see Theme B) and management (see Theme A). To effectively implement inner source, new community building and incentivization processes need to be introduced (see Theme C).

While several tools and techniques have been designed to support traditional development environments (see Theme D), not all of them were originally intended for inner source. Many were created for predictive or history-oriented business processes outside of inner source. Therefore, future research on inner source measurement should focus on making predictions and calculating historic events specifically with inner source in mind, in order to comprehensively handle it.

To provide guidance for future inner source measurement tools or models, we need to further examine the codes and themes in our analysis to identify which tools and techniques are better suited for the inner source paradigm, and which may require significant adjustments by future researchers. This will help create a comprehensive inner source measurement tool or model that can assist with as many business processes as possible.

5.3 Part 3c: Computational tools and techniques usability analysis

This section examines the applicability of certain tools and techniques for the inner source domain by analyzing their procedures (Theme E) and goals (Theme D).

Figure 8 illustrates the relationship between Themes D and E in detail. The figure consists of two main boxes, with Theme D (Gray) representing the computational goals and Theme E (White) representing the algorithmic procedures used. The figure depicts the identified procedures, classified by their suitability for specific calculation purposes, and how they relate to the computational goals of the algorithm. It provides an overview of the suitability of tools and techniques for applying certain procedures in inner source calculations or predictions.

Although we initially selected tools and techniques that are generally applicable to cross-boundary collaboration, not all of them are equally suitable for use in inner source or in all situations that come with it (e.g., predictive vs historic calculations discussed in Section 5.2). We have classified these tools and techniques into three basic types:

(1) Well-suited approaches: These tools and techniques are easily applicable to the cross-boundary pattern of inner source.

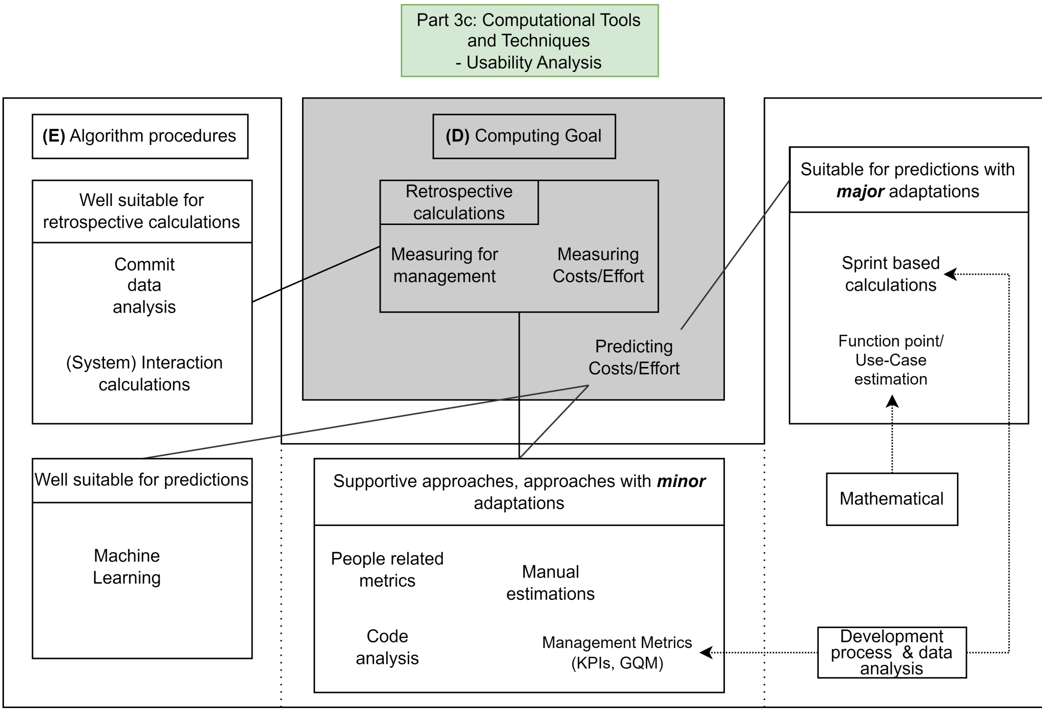

Fig. 8. Overview over the connections between the goals and procedures of computational tools and techniques

(2) Supportive approaches/approaches suitable with minor adaptations: These tools and techniques can be used in inner source with minor adjustments or provide additional support for other tools and techniques.
(3) Approaches suitable with major adaptations: These tools and techniques may be generally applicable to inner source but require major adjustments to benefit inner source.

Our main focus is on the ease of assessing individual cross-boundary contributions (transfers) with the identified tools and techniques, as these represent the most elementary part of inner source development.

5.3.1 Inner source usability analysis.

Well-suited approaches. Tools and techniques in this category are easy to apply to the cross-boundary pattern of inner source. Some are well-suited for retrospective calculations, while others are well-suited for predictive calculations.

For retrospective calculations using historic data, tools and techniques that support cross-boundary collaboration are ideal. Examples include individual system interactions [4, 29] and commit data [10]. These tools are directly assignable to contributions and are easy to use in management and accounting processes presented with Theme A and B.

To make predictions, a dedicated machine-learning model might be suitable, although specific machine learning solutions were not the focus of this review. The applicability of machine learning heavily depends on the goal and the business environment.

Supportive approaches. Tools and techniques in this group are usable for other tools and techniques as input or are suitable after minor adaptations.

Various tools and techniques fit into this category. Source code analysis, for example, may not be directly related to the transferred intellectual property, but it can enable code metrics (e.g., code complexity) on a commit level [4, 67]. The same is true for people-related metrics such as full- and part-time handling [56]. Moreover, management metrics and methods like KPI calculations can provide value support and validation information for potential future inner source tools and techniques developed in research and business [3, 6, 16].

Although manual methods were not actively reviewed in this paper, some of the methods mentioned in the literature, such as expert judgments, planning poker, and the Delphi method [7], are suitable for validation purposes of future tools. However, using manual inputs should be the exception, as it is time-intensive for the large amount of high-frequency inner source contributions and not easy to include in future software tools.

Approaches suitable with major adaptations. Tools and techniques in this category require significant modifications to be suitable for the inner source paradigm. Within this group, we have identified only predictive algorithms.

For example, function point/Use-case estimation, often used for effort estimation [35], does not always capture the fine-grained level of flexibility required for inner source contributions. Further research is needed to investigate how these methods can be adapted to meet the needs of inner source development. Similarly, sprint-based calculations (e.g. [36]) do not directly correlate with inner source IP transfer, but may provide valuable insights after appropriate adjustments have been made.

5.3.2 Business and research implications. We classified several tools and techniques and found that most of them require significant adjustment to be suitable for use in inner source, while only a few are well-suited for economic assessment in inner source. The supportive approaches that are well-suited for inner source deal with data structures that are directly assignable to transferable work within inner source, such as commits and system interactions.

One key takeaway from our classification is that most of the existing tools and techniques were not developed with inner source in mind, and therefore, require major or minor adaptations for use in inner source. Furthermore, the tools and techniques that are applicable to inner source were developed for specific use cases, such as transfer pricing [10].

To address these issues, we propose future research to integrate all suitable approaches into a tool that is specifically designed for the economic assessment of inner source. Such a tool would enable companies to adjust their processes and take advantage of all the benefits that inner source has to offer.

5.4 Key findings

The goal of this paper was to identify the economic impact of inner source on businesses and their processes, and to determine how such an impact can be quantified. Through thematic analysis and the resulting thematic map, we are able to answer our initially proposed research question.

We discovered that inner source affects a wide range of business processes within management and accounting (Part 1, Themes A to C). We also found that existing software development practices, such as code review and documentation handling, need to be adapted to facilitate inner source. Furthermore, new processes for community building, such as inner source incentivisation, need to be implemented to fully capitalize on the benefits of inner source.

Regarding quantification, we found that although many computational tools and practices exist to measure software development and support existing business processes (Part 2, Themes D to G), most are not yet suitable for handling the cross-organizational collaboration patterns of inner source (Part 3). We identified preliminary data sources and procedures capable of handling cross-organizational IP contributions that are well-suited for measuring inner source and related business processes.

As a result, future research should focus on developing tools and techniques that are capable of handling inner source flows and applying them within businesses. Additionally, we identified the need for both predictive and retrospective calculations to comprehensively cover inner source measurement.

6 OUTLOOK AND CONCLUSION

6.1 Limitations

In this section, we will discuss the limitations of our findings using the trustworthiness criteria proposed by Lincoln and Guba [41]. These criteria include credibility, confirmability, transferability, and dependability.

Credibility refers to whether the findings reflect the reality. In our work, we limited our review to literature sources only, as described in our review protocol in Section 3.3. While we did not directly include findings from industry through interviews or case studies, we mitigated this limitation by carefully selecting papers dealing with case studies or reviews handling industry perspectives and feedback. By doing so, we were able to integrate multiple industry perspectives in a thoroughly evaluated manner, as we checked the quality criteria of the papers and assessed their practical relevance.

Confirmability refers to avoiding researcher bias. We ensured confirmability in our review by having both authors conduct an independent review and thematic analysis and agreeing on the findings afterwards (inter-rater reliability). Additionally, we only included peer-reviewed papers and other literature reviews to reduce the risk of researcher bias.

Transferability refers to the applicability of the findings outside of the paper scope. We recognize that only a few papers we identified have directly taken inner source into consideration, as research in that domain is not yet widespread enough. However, we addressed this limitation by considering articles outside the inner source domain and reviewing the applicability of the identified tools and techniques to the inner source domain, as presented in Section 5. Moreover, we limited our literature review to reproducible tools and techniques (e.g. no machine learning) that involve (almost) no manual work. This enabled the transferability of the findings to a wide range of business processes and development measurement domains outside the review scope of inner source.

Finally, dependability refers to the replicability of the study design. While our chosen study design may limit dependability, we provided all the information proposed by Kitchenham [37] to ensure replicability. This includes a detailed review protocol, executed steps, key words, and quality criteria. We also provided the thematic maps created throughout our research process, following Braun and Clarke's [9] thematic analysis, to ensure transparency.

6.2 Broader research influence

The economic assessment of inner source provides a foundation for addressing the research questions posed by Edison et al. [23]. They identified a lack of clarity regarding how improvements in management through inner source can be measured, and our review has shown that tools and techniques capable of performing predictive calculations are essential for managing inner source-related processes.

Additionally, Edison et al. stated that the creation of business value through inner source is unclear. Our research lays the groundwork for answering this question by providing an overview of how inner source metrics should be created to comprehensively measure its impact.

Moreover, economic assessments of inner source can be crucial tools for a wide range of research agendas based on economic implications, extending beyond inner source development to accelerate general economic and software-related research.

6.3 Future research propositions

Based on the results of our systematic literature review and thematic analysis, we propose several aspects for future research. First, future research should focus on developing comprehensive tools and techniques to assess the impact of inner source on businesses and their processes. Such tools should integrate a wide range of data sources, including system and process data, in addition to commit data as used in current solutions (e.g., Buchner et al., 2022).

Another important challenge is to quantify the social aspects of inner source development, which can provide valuable insights into inner source adoption and team dynamics. To address this, further research should extend existing work on inner source community building and incentivization, and measure their impact.

In addition to retrospective tools and techniques, future research should also explore predictive algorithms dedicated to inner source development. This would help improve inner source planning and control, and provide valuable insights and improvements for both research and business through the use of machine learning algorithms.

To facilitate the introduction and management of inner source, we propose building one or more inner source measurement and accounting tools that can comprehensively deliver all necessary information. These tools should also be able to easily adapt existing processes to this new paradigm.

To measure the impact of inner source on businesses and their processes, we recommend conducting case studies that evaluate the usability of inner source measurement tools. We found that inner source influences a company's way of organizing their development teams and overall organizational structure (e.g., functional vs. platform organization, as discussed in Part 1). Therefore, we propose further research to measure the performance of different organizational structures and compare them with the application of inner source in terms of efficiency.

6.4 Conclusions

In this review, we analyze two perspectives on assessing economic benefits of inner source: the business process perspective (Part 1) and the tools and techniques perspective (Part 2). We then explore how these tools and techniques align with the business processes influenced by inner source, particularly for management and accounting. Additionally, we conduct an analysis of which types of tools and techniques are well-suited to handle business processes affected by inner source, and identify areas that require further research.

Our key finding is that current tools and techniques are insufficient to provide a comprehensive assessment of the economic benefits of inner source. Existing tools and techniques that are applicable to inner source rely on data sources and procedures that can identify cross-boundary IP flow. Future research should focus on developing predictive and retrospective directed processes that can handle these assessments more effectively.

Overall, this systematic literature review lays the foundation for potential future research that can improve inner source adoption, making it easier for companies to become more efficient and agile in responding to new market needs.

ACKNOWLEDGMENTS

This work was supported by the DFGs (German Research Foundation) Research Grants Program (Management Accounting for Inner Source, RI 2147/5-2).

We thank our colleagues for continuous feedback and proofreading of this article.

We acknowledge the use of ChatGPT (GPT-3.5, Version 23rd March 2023) for improving the readability of this research paper (and only that). As non-native speakers, we utilized it solely for correcting grammatical errors, correcting and suggesting appropriate words and phrases, and detecting spelling mistakes. We copied and pasted each paragraph of our research paper individually into the software. We then thoroughly reviewed each correction in every paragraph suggested by ChatGPT to ensure that our original intention remained unchanged, and that no contents were added or deleted. This process ensured that the improvements suggestions did not alter the meaning of our work in any way. ChatGPT was not used in any way for the literature identification, content creation, and writing of that paper.

REFERENCES

[1] InnerSource Commons. 2021. State of InnerSource 2021. https://innersourcecommons.org/learn/research/state-of-innersource-survey-2021/ Retrieved: 2023-07-12.
[2] InnerSource Commons. 2023. Community. https://innersourcecommons.org/community/ Retrieved: 2023-03-21.
[3] Ž Antolić. 2008. An example of using key performance indicators for software development process efficiency evaluation. *R&D Center Ericsson Nikola Tesla* 6 (2008), 1–6.
[4] Saulius Astromskis, Andrea Janes, Alberto Sillitti, and Giancarlo Succi. 2014. An Approach to Non-invasive Cost Accounting. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 30–37. https://doi.org/10.1109/SEAA.2014.53
[5] Bunea Bontas Cristina Aurora. 2013. THE COST OF PRODUCTION UNDER DIRECT COSTING AND ABSORPTION COSTING – A COMPARATIVE APPROACH. *Annals - Economy Series* 2 (2013), 123–129.
[6] Victor Basili, Mikael Lindvall, Myrna Regardie, Carolyn Seaman, Jens Heidrich, Jürgen Münch, Dieter Rombach, and Adam Trendowicz. 2010. Linking Software Development and Business Strategy Through Measurement. *Computer* 43 (05 2010), 57–65. https://doi.org/10.1109/MC.2010.108
[7] Saurabh Bilgaiyan, Santwana Sagnika, Samaresh Mishra, and M.N Das. 2017. A Systematic Review on Software Cost Estimation in Agile Software Development. *JOURNAL OF ENGINEERING SCIENCE AND TECHNOLOGY REVIEW* 10 (08 2017), 51–64. https://doi.org/10.25103/jestr.104.08
[8] Barry W. Boehm. 1984. Software Engineering Economics. *IEEE Transactions on Software Engineering* SE-10, 1 (1984), 4–21. https://doi.org/10.1109/TSE.1984.5010193
[9] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3 (01 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa
[10] Stefan Buchner and Dirk Riehle. 2022. Calculating the Costs of Inner Source Collaboration by Computing the Time Worked. In *Proceedings of the 55th Hawaii International Conference on System Sciences, HICSS '22*. virtual.
[11] Maximilian Capraro. 2020. *Measuring Inner Source Collaboration.* Ph. D. Dissertation. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU).
[12] Maximilian Capraro, Michael Dorner, and Dirk Riehle. 2018. The Patch-Flow Method for Measuring Inner Source Collaboration. In *Proceedings of the 15th International Conference on Mining Software Repositories* (Gothenburg, Sweden) *(MSR '18)*. Association for Computing Machinery, New York, NY, USA, 515–525. https://doi.org/10.1145/3196398.3196417
[13] Maximilian Capraro and Dirk Riehle. 2016. Inner Source Definition, Benefits, and Challenges. *ACM Comput. Surv.* 49, 4, Article 67 (Dec. 2016), 36 pages. https://doi.org/10.1145/2856821
[14] Noel Carroll, Lorraine Morgan, and Kieran Conboy. 2018. Examining the Impact of Adopting Inner Source Software Practices. In *Proceedings of the 14th International Symposium on Open Collaboration* (Paris, France) *(OpenSym '18)*. Association for Computing Machinery, New York, NY, USA, Article 6, 7 pages. https://doi.org/10.1145/3233391.3233530
[15] Yasin Ceran, Milind Dawande, Dengpan Liu, and Vijay Mookerjee. 2014. Optimal Software Reuse in Incremental Software Development: A Transfer Pricing Approach. *Management Science* 60 (03 2014), 541–559. https://doi.org/10.1287/mnsc.2013.1757
[16] Tjan-Hien Cheng, Slinger Jansen, and Marc Remmers. 2009. Controlling and monitoring agile software development in three dutch product software companies. In *2009 ICSE Workshop on Software Development Governance*. IEEE, 29–35. https://doi.org/10.1109/SDG.2009.5071334
[17] Danese Cooper and Klaas-Jan Stol. 2018. *Adopting InnerSource: Principles and Case Studies.* O'Reilly Media.
[18] Tapajit Dey, Willem Jiang, and Brian Fitzgerald. 2022. Knights and Gold Stars: A Tale of InnerSource Incentivization. *IEEE Software* 39 (11 2022), 88–98. https://doi.org/10.1109/MS.2022.3192647
[19] Isabel Drost-Fromm. 2018. Borrowing Open Source Practices at Europace. In *Adopting InnerSource: Principles and Case Studies*, Danese Cooper and Klaas-Jan Stol (Eds.). O'Reilly Media.

[20] Santiago Dueñas, Valerio Cosentino, Jesus M. Gonzalez-Barahona, Alvaro del Castillo San Felix, Daniel Izquierdo-Cortazar, Luis Cañas-Díaz, and Alberto Pérez García-Plaza. 2021. GrimoireLab: A toolset for software development analytics. *PeerJ Computer Science* 7 (July 2021), e601. https://doi.org/10.7717/peerj-cs.601
[21] Christof Ebert. 2014. Software Product Management. *IEEE Software* 31, 3 (2014), 21–24. https://doi.org/10.1109/MS.2014.72
[22] Christof Ebert, Bvs Krishna Murthy, and Namo Narayan Jha. 2008. Managing Risks in Global Software Engineering: Principles and Practices. In *2008 IEEE International Conference on Global Software Engineering*. IEEE, 131–140. https://doi.org/10.1109/ICGSE.2008.12
[23] Henry Edison, Noel Carroll, Lorraine Morgan, and Kieran Conboy. 2020. Inner Source Software Development: Current Thinking and an Agenda for Future Research. *Journal of Systems and Software* 163 (05 2020), 110520. https://doi.org/10.1016/j.jss.2020.110520
[24] Joseph Feller and Brian Fitzgerald. 2000. A Framework Analysis of the Open Source Software Development Paradigm. In *Proceedings of the Twenty First International Conference on Information Systems* (Brisbane, Queensland, Australia) *(ICIS '00)*. Association for Information Systems, USA, 58–69.
[25] Robert C. Ford and W. Alan Randolph. 1992. Cross-Functional Structures: A Review and Integration of Matrix Organization and Project Management. *Journal of Management* 18, 2 (1992), 267–294. https://doi.org/10.1177/014920639201800204 arXiv:https://doi.org/10.1177/014920639201800204
[26] Martin Fowler. 1996. *Analysis Patterns: Reusable Object Models (Addison-Wesley Series in Object-Oriented Software Engineerin)*. Addison-Wesley Longman, Amsterdam.
[27] Thomas Froment and Guillaume Angier de Lohéac. 2021. The Convergence of Struggles! Reusability Assessment of Inner-Source Components for Product Lines. *INSIGHT* 24, 1 (2021), 30–34. https://doi.org/10.1002/inst.12325 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/inst.12325
[28] R. Fuller. 2019. Functional Organization of Software Groups Considered Harmful. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*. IEEE, 120–124. https://doi.org/10.1109/ICSSP.2019.00024
[29] Georgios Gousios, Eirini Kalliamvakou, and Diomidis Spinellis. 2008. Measuring Developer Contribution from Software Repository Data. In *Proceedings of the 2008 International Working Conference on Mining Software Repositories* (Leipzig, Germany) *(MSR '08)*. Association for Computing Machinery, New York, NY, USA, 129–132. https://doi.org/10.1145/1370750.1370781
[30] Georg Gruetter, Diogo Fregonese, and Jason Zink. 2018. Living in a BIOSphere at Robert Bosch. In *Adopting InnerSource: Principles and Case Studies*, Danese Cooper and Klaas-Jan Stol (Eds.). O'Reilly Media.
[31] Julian Hirsch and Dirk Riehle. 2022. Management Accounting Concepts for Inner Source Software Engineering. In *Lecture Notes in Business Information Processing* (2022-11-08/2022-11-11), Noel Carroll, Anh Nguyen-Duc, Xiaofeng Wang, and Viktoria Stray (Eds.), Vol. 463 LNBIP. Springer Science and Business Media Deutschland GmbH, 101–116. https://doi.org/10.1007/978-3-031-20706-8{_}7
[32] Mike Hobday. 2000. The project-based organisation: an ideal form for managing complex products and systems? *Research Policy* 29, 7 (2000), 871–893. https://doi.org/10.1016/S0048-7333(00)00110-4
[33] Martin Ivarsson and Tony Gorschek. 2011. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering* 16, 3 (June 2011), 365–395. https://doi.org/10.1007/s10664-010-9146-4
[34] Capers Jones. 2004. Software project management practices: Failure versus success. *CrossTalk: The Journal of Defense Software Engineering* 17, 10 (2004), 5–9.
[35] Sungjoo Kang, Okjoo Choi, and Jongmoon Baik. 2010. Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development. In *2010 IEEE/ACIS 9th International Conference on Computer and Information Science*. IEEE, 743–748. https://doi.org/10.1109/ICIS.2010.126
[36] Hrvoje Karna. 2020. Data Mining Approach to Effort Modeling On Agile Software Projects. *Informatica* 44 (06 2020). https://doi.org/10.31449/inf.v44i2.2759
[37] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.
[38] Martin Kornberger, Dane Pflueger, and Jan Mouritsen. 2017. Evaluative infrastructures: Accounting for platform organization. *Accounting, Organizations and Society* 60 (2017), 79–95. https://doi.org/10.1016/j.aos.2017.05.002
[39] Y.H. Kwak and J. Stoddard. 2004. Project risk management: lessons learned from software development environment. *Technovation* 24, 11 (2004), 915–920. https://doi.org/10.1016/S0166-4972(03)00033-6
[40] Leonardo Leite, Fabio Kon, Gustavo Pinto, and Paulo Meirelles. 2020. Platform Teams: An Organizational Structure for Continuous Delivery. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (Seoul, Republic of Korea) *(ICSEW'20)*. Association for Computing Machinery, New York, NY, USA, 505–511. https://doi.org/10.1145/3387940.3391455
[41] Yvonna S Lincoln and Egon G Guba. 1985. *Naturalistic inquiry.* sage.

[42] Mikael Lindvall, Dirk Muthig, Aldo Dagnino, Christina Wallin, Michael Stupperich, David Kiefer, John May, and T. Kahkonen. 2005. Agile software development in large organizations. *Computer* 37 (01 2005), 26– 34. https://doi.org/10.1109/MC.2004.231
[43] Orly Mazur. 2016. Transfer Pricing Challenges in the Cloud. *Boston College Law Review* 57, 2 (2016), 643–693. https://lawdigitalcommons.bc.edu/bclr/vol57/iss2/6
[44] William E. McCarthy. 1982. The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review* 57, 3 (1982), 554–578. http://www.jstor.org/stable/246878
[45] Lorraine Morgan, Joseph Feller, and Patrick Finnegan. 2011. Exploring inner source as a form of intraorganisational open innovation. In *19th European Conference on Information Systems, ECIS 2011, Helsinki, Finland, June 9-11, 2011*, Virpi Kristiina Tuunainen, Matti Rossi, and Joe Nandhakumar (Eds.). 151. http://aisel.aisnet.org/ecis2011/151
[46] Lorraine Morgan, Rob Gleasure, Abayomi Baiyere, and Hong Phuc Dang. 2021. Share and Share Alike: How Inner Source Can Help Create New Digital Platforms. *California Management Review* 64, 1 (2021), 90–112. https://doi.org/10.1177/00081256211044830 arXiv:https://doi.org/10.1177/00081256211044830
[47] Donatien Koulla Moulla, Alain Abran, and Kolyang. 2021. Duration Estimation Models for Open Source Software Projects. *International Journal of Information Technology and Computer Science* 13 (02 2021), 1–17. https://doi.org/10.5815/ijitcs.2021.01.01
[48] A. Neumann. 2019. *Transfer Pricing in Inner Source Software Development*. Master's thesis. Hochschule des Bundes für öffentliche Verwaltung, Bruhl, Germany.
[49] OECD. 2015. *Aligning Transfer Pricing Outcomes with Value Creation, Actions 8-10 - 2015 Final Reports*. OECD. 192 pages. https://doi.org/10.1787/9789264241244-en
[50] OECD. 2017. *OECD Transfer Pricing Guidelines for Multinational Enterprises and Tax Administrations 2017*. OECD. 608 pages. https://doi.org/10.1787/tpg-2017-en
[51] Marcel Olbert and Christoph Spengel. 2017. International taxation in the digital economy : challenge accepted? *World Tax Journal : WTJ* 9, 1 (2017), 3–46. https://madoc.bib.uni-mannheim.de/41867/
[52] Open Source Initiative. 2007. *The Open Source Definition*. https://opensource.org/osd Retrieved: 2021-11-03.
[53] Christian Plesner Rossing, Martine Cools, and Carsten Rohde. 2017. International transfer pricing in multinational enterprises. *Journal of Accounting Education* 39, C (2017), 55–67. https://EconPapers.repec.org/RePEc:eee:joaced:v:39:y:2017:i:c:p:55-67
[54] Fumin Qi, Xiao-Yuan Jing, Xiaoke Zhu, Xiaoyuan Xie, Baowen Xu, and Shi Ying. 2017. Software effort estimation based on open source projects: Case study of Github. *Information and Software Technology* 92 (2017), 145–157. https://doi.org/10.1016/j.infsof.2017.07.015
[55] Dirk Riehle, Maximilian Capraro, Detlef Kips, and Lars Horn. 2016. Inner Source in Platform-Based Product Engineering. *IEEE Transactions on Software Engineering* 42 (12 2016), 1162–1177. https://doi.org/10.1109/TSE.2016.2554553
[56] Gregorio Robles, Jesús M. González-Barahona, Carlos Cervigón, Andrea Capiluppi, and Daniel Izquierdo-Cortázar. 2014. Estimating Development Effort in Free/Open Source Software Projects by Mining Software Repositories: A Case Study of OpenStack. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (Hyderabad, India) *(MSR 2014)*. Association for Computing Machinery, New York, NY, USA, 222–231. https://doi.org/10.1145/2597073.2597107
[57] Geoffrey G Roy. 2004. A risk management framework for software engineering practice. In *2004 Australian Software Engineering Conference. Proceedings.* IEEE, 60–67. https://doi.org/10.1109/ASWEC.2004.1290458
[58] Klaas-Jan Stol, Paris Avgeriou, Muhammad Ali Babar, Yan Lucas, and Brian Fitzgerald. 2014. Key Factors for Adopting Inner Source. *ACM Trans. Softw. Eng. Methodol.* 23, 2, Article 18 (April 2014), 35 pages. https://doi.org/10.1145/2533685
[59] Klaas-Jan Stol, Muhammad Ali Babar, Paris Avgeriou, and Brian Fitzgerald. 2011. A comparative study of challenges in integrating Open Source Software and Inner Source Software. *Information and Software Technology* 53, 12 (2011), 1319–1336. https://doi.org/10.1016/j.infsof.2011.06.007
[60] Klaas-Jan Stol and Brian Fitzgerald. 2015. Inner Source–Adopting Open Source Development Practices in Organizations: A Tutorial. *IEEE Software* 32, 4 (2015), 60–67. https://doi.org/10.1109/MS.2014.77
[61] PRICE Systems. 2021. Company overview: Price® Systems. https://www.pricesystems.com/about-us/ Retrieved: 2021-12-15.
[62] The International Federation of Accountants. 2009. *Evaluating and Improving Costing in Organizations*. International Federation of Accountants (IFAC). https://www.ifac.org/knowledge-gateway/preparing-future-ready-professionals/publications/evaluating-and-improving-costing-organizations
[63] United Nations. 2014. *United Nations Practical Manual on Transfer Pricing for Developing Countries*. United Nations. https://www.un-ilibrary.org/content/books/9789210561372
[64] Muhammad Usman, Emilia Mendes, Francila Weidt, and Ricardo Britto. 2014. Effort estimation in agile software development. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering* (Turin, Italy). ACM, 82–91. https://doi.org/10.1145/2639490.2639503

[65] J.M. Verner and N. Cerpa. 2005. Australian software development: what software project management practices lead to success?. In *2005 Australian Software Engineering Conference*. IEEE, 70–77. https://doi.org/10.1109/ASWEC.2005.14

[66] Anna Wiedemann, Manuel Wiesche, and Helmut Krcmar. 2019. Integrating Development and Operations in Cross-Functional Teams - Toward a DevOps Competency Model. In *Proceedings of the 2019 on Computers and People Research Conference* (Nashville, TN, USA) *(SIGMIS-CPR '19)*. Association for Computing Machinery, New York, NY, USA, 14–19. https://doi.org/10.1145/3322385.3322400

[67] Hong Wu, Lin Shi, Celia Chen, Qing Wang, and Barry Boehm. 2016. Maintenance Effort Estimation for Open Source Software: A Systematic Literature Review. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 32–43. https://doi.org/10.1109/ICSME.2016.87

[68] Darja Šmite, Nils Brede Moe, Aivars Šāblis, and Claes Wohlin. 2017. Software teams and their knowledge networks in large-scale software development. *Information and Software Technology* 86 (2017), 71–86. https://doi.org/10.1016/j.infsof.2017.01.003