Article

System of Systems Lifecycle Management—A New Concept Based on Process Engineering Methodologies

Dániel Kozma [1,*] **, Pál Varga** [1] **and Felix Larrinaga** [2]

1. Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 2 Magyar Tudósok krt., H-1117 Budapest, Hungary; pvarga@tmit.bme.hu
2. Faculty of Engineering, Mondragon University, Goiru kalea 2, Arrasate-Mondragon, E-20500 Gipuzkoa, Spain; flarrinaga@mondragon.edu
* Correspondence: kozma@tmit.bme.hu

Abstract: In order to tackle interoperability issues of large-scale automation systems, SOA (Service-Oriented Architecture) principles, where information exchange is manifested by systems providing and consuming services, have already been introduced. However, the deployment, operation, and maintenance of an extensive SoS (System of Systems) mean enormous challenges for system integrators as well as network and service operators. The existing lifecycle management approaches do not cover all aspects of SoS management; therefore, an integrated solution is required. The purpose of this paper is to introduce a new lifecycle approach, namely the SoSLM (System of Systems Lifecycle Management). This paper first provides an in-depth description and comparison of the most relevant process engineering methodologies and ITSM (Information Technology Service Management) frameworks, and how they affect various lifecycle management strategies. The paper's novelty strives to introduce an Industry 4.0-compatible PLM (Product Lifecycle Management) model and to extend it to cover SoS management-related issues on well-known process engineering methodologies. The presented methodologies are adapted to the PLM model, thus creating the recommended SoSLM model. This is supported by demonstrations of how the IIoT (Industrial Internet of Things) applications and services can be developed and handled. Accordingly, complete implementation and integration are presented based on the proposed SoSLM model, using the Arrowhead framework that is available for IIoT SoS.

1. Introduction

Nowadays, ICT (Information and Communications Technology) is continuously evolving and increasingly permeating companies, society, governments, and businesses. Organizations rely more and more on their ICT infrastructure for their business continuity. As a result, successful companies have recognized that management needs to deal with ICT as much as their other business parts. This attitude triggered the most prominent standardization organizations to provide comprehensive proposals for ITSM (Information Technology Service Management).

Even the best system is worth only as much as its weakest component; therefore, the management technologies and process engineering strategies must scale with systems' sizes. This statement is especially true for industrial systems where reliable system design is a critical success factor for manufacturing. Industry 4.0 has recently established its principles [1]. According to [2,3], the definition of Industry 4.0 can be summarized as a general terminology and concept for the fully digitized production considering the following abilities: interoperability, virtualization, decentralization, real-time capability, service orientation, and modularity. In the modular smart factories of Industry 4.0, CPS (Cyber-Physical System) processes can be monitored through virtual mapping and decentralized

real process decisions. Real-time communication and collaboration can occur between people, between machines, or between people and machines. With the IoT (Internet of Things) domain's support, internal and external organizational services can be offered and accessed from inside or outside the enterprise.

However, there is still a shortage of comprehensive lifecycle management standards and guidance due to the complexity that comes from various types of network architectures [4]. For large SoS (System of Systems), the SOA (Service-Oriented Architecture) approach has been investigated as the architecture that can handle large-scale automation [5]. To manage these architectures, SOA-based IIoT (Industrial Internet of Things) frameworks can serve as the infrastructure for satisfying the related conditions of SoS communication. The modular approach of Industry 4.0 requires applications to be integrated into industrial systems, e.g., CPSs, as efficiently and quickly as possible. This expectation requires, among others, up-to-date knowledge of the SoS states and the actions needed at state transitions. The research community has realized that, due to the complexity of SoS, its requirements and processes need to undergo adaptations to fit the development and management of this type of architecture according to new trends. There are proposals for this field [6–8], addressing the modeling and design requirements; however, after careful study of the literature, it can be concluded that both research answers and practical guidance for comprehensive SoS management are incomplete and not mature enough yet.

The purpose of the current article is precisely to address and close this gap; therefore, it introduces a generic lifecycle model, the SoSLM (System of Systems Lifecycle Management), enhanced by an industry-specific PLM (Product Lifecycle Management) model based on the widely accepted and used ITSM frameworks and process engineering methodologies [9–11]. In order to verify the relevance of the proposed model, real development is presented using the created SoSLM model. To put this into context, an IIoT framework, namely Arrowhead—used by hundreds of partner organizations in European research and innovation actions such as ECSEL MANTIS [12], Productive4.0 [13], or Arrowhead Tools [14]—is used for validation.

The rest of the paper is organized as follows. Section 2 presents the common lifecycle management models and the related gaps in the aspect of SoS management; Section 3 briefly summarizes the most used process engineering frameworks; Section 4 describes the research methodology; Section 5 defines the definition of product and, based on that definition, proposes a general PLM model taking into account the presented process engineering and ITSM frameworks; Section 6 introduces the Arrowhead framework for IIoT; Section 7 presents the proposed SoSLM model in detail, which is also validated through a real implementation; and Section 8 evaluates the results achieved. Finally, Section 9 concludes the paper.

2. An Overview on Common Lifecycle Management Models

Nowadays, process engineers can utilize a variety of software tools to perform projects and to achieve promising solutions quicker than ever before. Modeling and analysis tools have significantly impacted process engineers' abilities to analyze multiple alternatives rapidly, to enable modern design, and to make cost tradeoffs for decision-makers. However, many challenges remain, especially with regard to engineers collaborating on large projects requiring high-quality and timely solutions. This is crucial, as companies across the globe are meeting increasingly interconnected and highly competing environments.

On the one hand, the literature related to development strategies is concentrating on process optimization [15–17], where the need for safety, operability, and quality challenges are addressed. On the other hand, the research community continuously explores the possible combinations of process engineering approaches [18–21]. There are also studies about how the process engineering frameworks can be introduced and maintained [22] and about what the possible pitfalls could be [23].

To handle the SoS-related issues, the first step is to examine the currently available models in order to see whether a model is already available that meets expectations.

2.1. Software Development Lifecycle Management (SDLC)

The SDLC primarily focuses on planning, creating, testing, and deploying software products. There are several useful approaches to software development and maintenance, such as the spiral model, waterfall model, or the agile methodology, that are very common today [24–26]. In SDLC's case, one of the biggest challenges is choosing the optimal model, as software development is diverse [27]. Moreover, research in the field also examines software development lifecycles in general [28].

2.2. Application Lifecycle Management (ALM)

ALM is the product lifecycle management of computer programs that is a wider approach than the SDLC, which is limited to the phases of the typical software development stages. In contrast, ALM defines stages after the development lifecycle as well [29–31]. According to [32], the advanced version of ALM is the integrated ALM, where the developers and the used tools are harmonized with each other during the development and operation stages of the software products. Research concentrates on how ALM can be applied to the industry [33,34] and its impact [32,35].

2.3. Information Lifecycle Management (ILM)

ILM refers to strategies for administering storage systems on computing devices [36]. Furthermore, ILM is for applying obvious and comprehensive policies to ensure correct information management. Considering the ILM's scope, there are five phases identified: creation and receipt, distribution, use, maintenance, and disposition. ILM is receiving increasing attention due to the rise in information security awareness and in regulations of personal data collection and processing, such as the GDPR (General Data Protection Regulation). ILM is a particular management strategy specifically for information handling; hence, it is usually integrated or supplemented as part of a larger lifecycle management [37,38]. However, some studies deal with essential information management [39] and data protection questions [40].

2.4. Product Lifecycle Management (PLM)

As it was stated by [41], the PLM is a complete model of the product lifecycle from innovation through design, development, and operation to dismantling of manufactured products. PLM models help companies deal with the ever-growing complexity and challenges related to the development and enhancement of products. PLM can integrate many areas such as people, processes, industrial and business systems, and data. It provides feedback about product information for companies and their extended enterprise. The ITSM frameworks can provide good basics for PLM; furthermore, they are used very often for creating new, area-specific models. Given that this type of lifecycle management is organically tied to Industry 4.0, many studies are available on where and how the PLM can be used [42,43].

2.5. Service Lifecycle Management (SLM)

This approach is the practice of aligning service management, communication, and service supporting operations to maximize the uptime [44]. For the ITSM community, the ITIL (Information Technology Infrastructure Library) framework was of great help (see Section 3.4). Looking at the evolution, first, ITIL_V2 described the best practices, which turned into a more holistic, service lifecycle-oriented approach in ITIL_V3. where with the value-creation ability of IT services was increased and the role of ITSM processes moved to support the service lifecycle [45]. SLM is another area besides PLM that is of great concern to the research community. Although there are studies where only SLM has been analyzed [46,47], recently, more investigation has been conducted in how PLM and SLM can work together or can be integrated with each other [48–50].

2.6. Recent Lifecycle Management Researches

Current research can be divided into three main areas. The first area deals with improving or enhancing existing and widespread lifecycle management models [51–53]. Out of the presented models, an impressive number of research is available and is still ongoing concerning the SLM and PLM models, as these are the most evident models for supporting Industry 4.0 and SCM (Supply Chain Management) processes. The second area examines the integration and collaboration issues of the existing lifecycle management approaches, as it was exemplarily mentioned in Section 2.5. Furthermore, the third field is about the development of new lifecycle models in accordance with new needs. This often happens if an existing model needs to be changed so much to fulfill the context-specific function that, in the meantime, it loses its original character or suitable modeling techniques cannot be found at all. Such new approaches often address machine learning [54] and AI (Artificial Intelligence) lifecycle modeling issues [55,56], or other area-specific models [57–59].

2.7. Gap: A Need for System of Systems Lifecycle Management

In the SOA approach, the term service is used as an entity for information exchange between the service provider and consumer systems. If this spectrum is extended, we can introduce the SoS concept, where the systems work together to achieve a more complex target or a higher purpose. An SoS is defined as the result of a set or arrangement of independent systems that are integrated and combined. To distinguish SoS from more traditional systems, it is necessary to understand some of its characteristics. According to Maier [60], five main dimensions can characterize an SoS concerning other vast and complex but monolithic systems:

1. Operational independent: a system is an independent part of an SoS, and it can be used to compose other SoS.
2. Management independent: each system could be managed independently.
3. Evolutionary development: the SoS is flexible and modular enough to evolve with functions and purposes added, removed, and modified along its lifetime.
4. Emergent behavior: its functionalities cannot be mapped into any component of the system, i.e., the behaviors are emergent properties of the overall SoS and cannot be assigned to any of its components.
5. Geographical distribution: the system components are physically distributed.

Taking into account the previously described lifecycle management approaches, it can be seen that no single management model covers the characteristics of SoS. The ALM is mainly for computer programs; therefore, it cannot entirely satisfy all of the requirements of SoS. The ILM can serve as a sound basis on how to control the information flow between the lifecycle stages. However, it alone does not provide enough description for complete SoS management, although it must be integrated into the SoS. SDLC is mainly concerned with how to develop code. The scope and emphasis are preferably on the development techniques, which is, on the one hand, a significant part of SoS but, on the other hand, just one of its essential tools. The SLM is much closer to the solution, but as its name shows, the focus is merely on the services and their management. However, managing the SoS is a much more complicated task. The lifecycle stages of PLM are relevant for SoS, but the perspective does not entirely align with SoS. Consequently, PLM is built around product-specific processes that are not altogether suitable for systems. However, similarly to SLM, the PLM strategies should also be taken into account for SoS. Table 1 serves with a brief overview about the main benefits and gaps of the examined lifecycle management models. Based on the results, it can be stated that there is no lifecycle management model yet, which can satisfy in itself the SoS-related requirements.

Table 1. Benefits and limitations of the analyzed lifecycle models.

Lifecycle Management	Benefits	Limitations
ALM	Comprehensive for software products and applications	- Not designed for complex system of systems
ILM	Basis of information handling	- Does not answer how to develop applications or systems
PLM	Useful for production and product tracking	- The scope is on the product-specific requirements. - Characteristics may be different for products.
SDLC	Well-defined set of principles and methodologies for software developments	- Not applicable in every case, e.g., for small or evolutionary developments - Complicated to use in dynamically changing environments
SLM	Useful for service operation management	- The scope is on the service-specific requirements. - Characteristics may be different for services. - The focus is on enhancing functional performance.

To achieve the most fitting model, the tasks to be solved were divided into two parts. Since the core of SoSLM relies heavily on the terminology used by PLM models, the aim was first to develop a PLM model, especially for IIoT systems. Section 5 describes the reason behind it and the proposed model accordingly. However, still, the target remains a comprehensive model to SoS architectures, for which the requirements were not entirely covered by the developed PLM model. Thus, in the second step, the developed IIoT-compliant PLM model was extended to the service-oriented expectations, creating an SoSLM model that can be read in Section 7. The presented lifecycle management approaches are based mostly on process engineering and ITSM frameworks. Thus, the processes recommended by the widely used and accepted frameworks [9–11] were examined during the formation of the models. The literature review is included in the next section.

3. An Overview on Common Process Engineering and ITSM Frameworks

An essential aspect of the model design was to examine the current standards and accepted industrial solutions; therefore, a thorough review of the process engineering field is the initial step. Process engineering focuses on the operation design, optimization, control, and intensification of processes. The subsequent sections give a short overview of some widely used process engineering methodologies and ITSM frameworks—especially those that have IT (Information Technology) or industrial relevance [9–11].

3.1. COBIT 5

The next generation of ISACA (Information Systems Audit and Control Association) guidelines for IT management and governance is COBIT 5 (Control Objectives for Information and related Technologies) [61], which provides organizations with a comprehensive framework to be able to achieve their corporate IT management and governance goals. COBIT 5 helps companies gain the most benefits from IT by striking a balance between the results, by optimizing risks, and by leveraging resources based on the following five principles.

3.1.1. Meeting Stakeholder Needs

At the highest levels, the executives must decide who the stakeholders are and what their needs are, and then, they have to establish the hierarchy. At management levels, each manager must identify their stakeholders and the tasks to satisfy their needs.

3.1.2. Covering the Enterprise End-to-End

COBIT 5 integrates the governance of IT with enterprise governance. It covers all functions and processes required to govern and manage enterprise information and related technologies regardless of where that information is processed and addresses all relevant external and internal IT services as well as internal and external business processes.

3.1.3. Applying a Single, Integrated Framework

The approach aligns with the latest relevant frameworks and standards. It is supposed to have enterprise-wide coverage. It provides a basis to integrate effectively other standards, frameworks, and practices. It combines the previously distributed knowledge over different ISACA frameworks and provides a simple architecture for guidance materials and producing consistent product sets, as outlined in Figure 1.
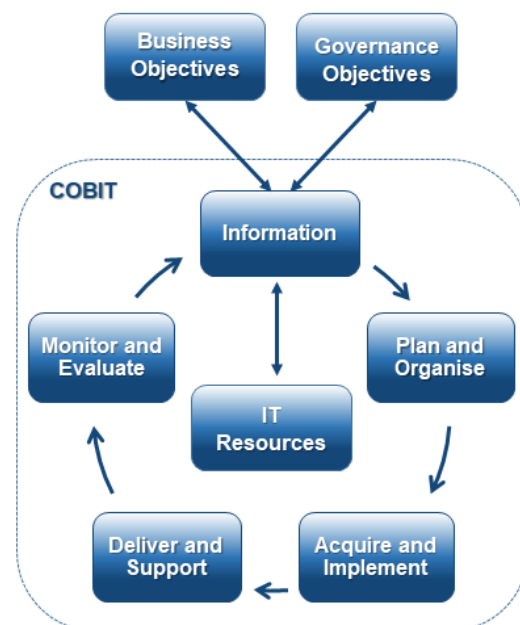

Figure 1. COBIT (Control Objectives for Information and related Technologies) lifecycle model.

3.1.4. Enabling a Holistic Approach

COBIT 5 describes a set of enablers to help implement a complete management and governance system for enterprise IT. The COBIT 5 enablers are principles; frameworks and policies; culture; processes; organizational structures; ethics and behavior; information; infrastructure and applications; services; and finally, people, skills, and competencies.

3.1.5. Separating Governance from Management

The COBIT 5 framework makes a bright contrast between management and governance. This encompasses different types of activities, requires different organizational structures, and serves different purposes.

3.2. Val IT

An organization needs a "strong" IT investment management, and in recent years, there has been a growing need for a framework that provides practical help in designing and managing the related processes.

To support these needs, the so-called Val IT framework was developed by the ITGI (IT Governance Institute). It is based on the best practices, corporate experiences, and literature [62]. This framework promotes business value-based IT investments. A set of techniques and methods that support IT investments' evaluation and management consistently provides a repeatable, comprehensive approach in which IT and business are equally important. Three main areas are addressed by the Val IT framework (Figure 2).
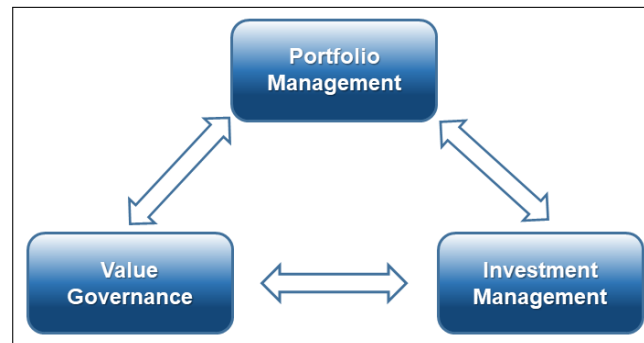

Figure 2. Val IT framework.

3.2.1. Portfolio Management

The goal of portfolio management—in the context of Val IT—is to ensure that an enterprise ensures optimal value across its portfolio of IT-enabled purchases. An executive engagement to portfolio management helps companies define investment thresholds; manage and establish resource profiles; prioritize, evaluate, defer, select, or reject new investments; optimize and manage the complete investment portfolio; and observe and report on portfolio display.

3.2.2. Investment Management

The IT-enabled investments of an enterprise must contribute to an optimal value. When organizational directors commit to investment management, they raise their ability to distinguish business requirements; to develop an accurate understanding of candidate investment programs; to examine alternative methods to implementing the programs; to define each program and document and to maintain a detailed business case for it, including the details of the benefits throughout the full economic lifecycle of the investment; to assign clear accountability and ownership, including those for benefits realization; to manage each program through its entire economic lifecycle including retirement; and to monitor and report on each program's performance.

3.2.3. Value Governance

Value management practices must be embedded in the company, enabling it to ensure optimal value from its IT-enabled purchases throughout its full economic lifecycle. An executive engagement to value governance helps settle the governance structure for value management in a manner that is wholly integrated with overall company governance; to provide strategic direction for the investment decisions; to define the characteristics of portfolios required to support new investments, IT services, assets, and other resources; and to improve value management on a continual basis, based on the lessons learned.

3.3. Risk IT

During the operation of companies, risk plays a critical role. Business decisions must also take into account the risks and potential benefits of the brought decisions. According to Risk IT [63], IT risk refers to the business risk associated with using IT. There are several approaches to risk assessment [64–66], but the result is common: the dimensions of risk assessment are the probability of occurrence and the caused business impact. As companies increasingly rely on their IT systems, IT risk assessment and continuous risk handling

have become critical issues. The Risk IT framework summarizes the key factors and form them into a general risk framework. As shown in Figure 3, the Risk IT framework presents subprocesses related to each area, i.e., input and output processes (relationships), and management procedures, and the framework defines the roles, responsibilities, process goals and metrics, and the maturity of the area.
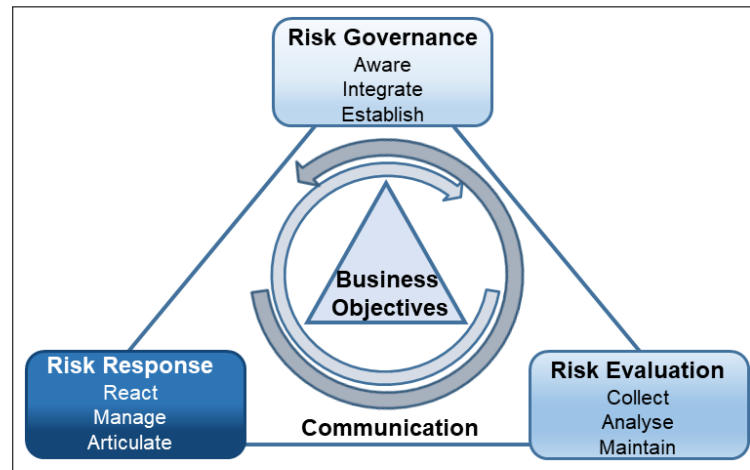

Figure 3. Risk IT framework.

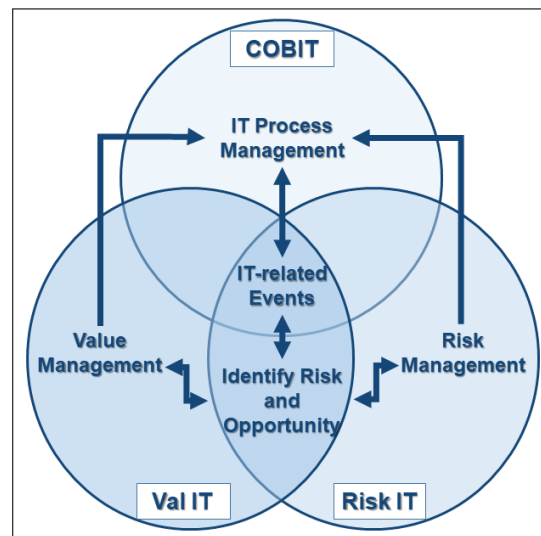The relationship between COBIT, Risk IT, and Val IT is shown in Figure 4.

Figure 4. Utilizing COBIT 5, Val IT, and Risk IT.

While COBIT supports IT risk management through the described control approach, Risk IT provides a framework that helps companies identify, manage, and perform risk-related management tasks. The Risk IT framework is entirely compatible with COBIT 5, so companies that use COBIT can also use Risk IT as their risk management framework. Moreover, the model is complemented by the Val IT suggested and measurable value management.

3.4. ITIL

ITIL provides a framework based on best practices for ITSM [67]. ITIL gives guidance to service providers for creating quality IT services, and for supporting this, it defines processes, (organizational) functions, and other capabilities. ITIL is one of the most widely known and recognized ITSM frameworks globally, and it has been used as a reference

model for ISO/IEC 20000, which is a formal international standard for organizations of service management. The ITIL framework is based on five phases of the service lifecycle, and accordingly, the core modules of ITIL also provide guidance on best practices for these phases. This guidance covers vital principles, required processes and activities, organizational frameworks and roles, technology, and challenges related to their implementation, critical success factors, and risks.

Figure 5 presents the ITIL service lifecycle structure.

Figure 5. ITIL_V3 (Information Technology Infrastructure Library) service lifecycle model.

As shown, the service lifecycle's core element is the service strategy around the design, transition, and operation phases. Each stage of the lifecycle influences and builds on forwarding information and feedback for the service providers. In this way, constant checks and balances can lead to continual service improvement throughout the whole service lifecycle. This ensures that the service provider can adapt and respond effectively to changes in customer requirements.

3.5. State-of-the-Art of the Presented Process Engineering Frameworks

As presented in the previous subsections, there are plenty of process engineering frameworks that could be applied in many areas. However, due to the heterogeneity of organizational systems and processes, it is rare to use only one framework; instead, it is more common to mix frameworks, selecting the most advantageous parts, creating a more complex but more field-specific solution.

3.5.1. Use Cases for COBIT 5

Due to its general nature, COBIT 5 can be used in most areas, which is also evident from the number of studies available. Perhaps the most significant benefits of its usage can be expressed in terms of improved service quality, reduced execution time of processes, and more efficient IT infrastructure management [68]. The application of COBIT 5 in other use cases considers its compliance with PCI DSS (Payment Card Industry Data Security Standard) [69] or the prevention of cyber attacks in supply chains [70]. This matter is very important, given the fact that, currently, the complexity and digital exposure of supply chains pose huge threats [71]. COBIT 5 research also targets CPS management in Industry 4.0, as presented in [72].

3.5.2. Use Cases for Val IT

Examples of the Val IT usage are also available in the literature. In this case, it can be observed that it is used in conjunction with another framework and rarely alone. A fascinating study is available on applying the Val IT framework to measure the maturity level for an ERP (Enterprise Resource Planning) system [73]. Moreover, there is a study that presents in detail how the usage of COBIT 5 and Val IT can benefit from a service-oriented approach to co-creating value in IT [74].

3.5.3. Use Cases for Risk IT

As mentioned earlier, Risk IT has been merged into COBIT 5, but publications are available where information is demonstrated: on the one hand, how Risk IT with COBIT 5 helps organizations prepare for IT-related audits [75] and, on the other hand, how they can help organizations perform risk assessments [76].

3.5.4. Use Cases for ITIL

ITIL deals specifically with service management, so this is a bit different from previous frameworks. However, the basic premise of Industry 4.0 is service-oriented behavior; therefore, the ITIL can provide a suitable solution for companies that want to be Industry 4.0-compatible. There are studies specifically from industrial segments [77,78] and generalized to service-based operation [79]. Furthermore, the combined use of COBIT 5 and ITIL is also widespread, especially in the context of information security [20,80].

4. Research Methodology

In order for the specific model to be developed, the DSRM (Design Science Research Methodology) [81] was used, which is specifically designed for research on information systems and related models. The DSRM defines six activities for the research work, which will be presented briefly in the next subsections.

4.1. Action 1: Problem Identification and Motivation

During the development of Arrowhead, many SoS-related lifecycle management questions arose, amongst others, the operation and integration management of the systems. As these issues became increasingly urgent to address, the use of an appropriate lifecycle model seemed to be a solution. To answer this question, first, a comprehensive literature review was performed to see whether a model that meets expectations is already available. Based on the SoS-related requirements, mapping them with the offers found in the literature, the development of a new model was designated as the solution. The literature review is presented in Sections 2 and 3.

4.2. Action 2: Define the Objectives for a Solution

Considering the changed scope of the product (introduced in Section 5.1), first, an Industry 4.0 PLM model was developed (presented by Section 5). During the use of the model, more specific requirements were defined for the SoS field (detailed in Section 2.7, which led to the development of an SoS-specific lifecycle management model.

4.3. Action 3: Design and Development

This research work has two artifacts: an Industry 4.0-compatible PLM model (Section 5) and one based on a more specific SoSLM model, presented in Section 7. The models are based on the widely accepted and used process engineering frameworks introduced in the literature review, which have evolved in response to the changing industrial environment triggered by Industry 4.0.

4.4. Action 4: Demonstration

The formed SoSLM model is demonstrated by a prototype implementation presented in Section 7.2, and based on the proof of concept use case, the model is used for management of the released system [82].

4.5. Action 5: Evaluation

A comprehensive evaluation can be read in Section 8. The section describes the main benefits and limitations of SoSLM within and outside of the Arrowhead-context.

4.6. Action 6: Communication

The components of the model were implemented in several experimental prototypes and European project use cases (Arrowhead, Mantis, and Productive 4.0). These projects involved industrial partners and research organizations where digitization, automation, SCM, and other challenges related to IoT and lifecycle management were addressed. Some of the experiments addressed in these projects and the model stage they confront are outlined next:

- Design, deployment, and system management supported by the Arrowhead framework in an industrial context (machine tooling) were implemented in Productive 4.0 [83,84].
- Research on business needs as well as on functional and quality requirements are considered in the design stage of [85] before developing an architectural solution for proactive maintenance in the Mantis project [86]. This use case was implemented in a company leader in the construction of power transmission components for metal forming machine tools such as clutches, brakes, or cams where CPSs are used to monitor performance.
- In Productive 4.0, the lifecycle model was also validated in the context of workflow management of the manufacturing process using the Arrowhead framework. The results are briefly demonstrated in Section 7.2 and detailed in [82,87].
- The lifecycle management issues are continually addressed in the current Arrowhead Tools project [88].

The next section introduces an Industry 4.0-compliant PLM model, which was developed based on the process engineering frameworks and lifecycle management models presented earlier and taking into account the changed scope of the product.

5. Product Lifecycle Management

The original goal of this work was primarily to model the interoperability and lifecycle phases of SoS environments; however, in the end, the solution was to use the PLM model as a starting point. The reason for this comes from the definition of the product itself.

5.1. The Definition of "Product"

In the traditional approach, the product term meant probably a physical item that was offered for sale. However, this terminology went through a lot of iterations over the decades, and now, it has expanded significantly. Now, products can be classified according to several categories, mainly in terms of whether they are tangible or intangible products. These categories can be further subdivided; for example, a tangible product can be raw materials, any goods, or assets, while a system, service, application, any virtual product, or even intellectual property can be considered an intangible product. In addition to these, of course, based on individual considerations, additional categories and classification can be defined, extending the product definition to more specific elements. However, the common characteristics of the products are mostly the same; a product must have a name, be made for a purpose, have a target audience, and be properly communicated for their use.

It can be stated that the reinterpreted definition of the product can even be inserted into any of the previously presented lifecycle management models. As the product has such an extensive definition, it is possible to define a core model that can serve as a basis for specific lifecycle models. As mentioned, products can be very diverse. Thus, due to the exponential growth of IoT, extended by industrial expectations, we focused primarily on providing a model that addresses and resolves the related gaps, detailed in Section 2.7; furthermore, the model is capable of adapting to the redefined product approach.

5.2. The Suggested PLM Model

The advent of IoT had a major impact on the world that triggered several industrial domains to improve their PLM [41]. Although companies across various industrial areas want to continuously meet the requirements of this fast-changing market; frequently,

they confront vital challenges in producing new and even more complicated products than before because of the industrial IoT specifications such as efficient deployment and engineering, real-time performance [89], high security [90], or proactive maintenance [91].

Currently, PLM experts are approaching and elaborating IoT strategies and solutions, especially for the industrial domain. One important change is that, now, PLM providers enable consumers to help develop the next generation of smart factories through continuous feedback [92]. Via feedback, useful information can be gathered, which can impact the architectural and development decisions of the CPSs, their automated workstations, and assembly lines, amongst others. However, the IIoT-related challenges are not only to solve interoperability issues but also to provide effective and reliable communication to the whole production ecosystem as well [93]; furthermore, collecting and processing as much information as possible.

During analysis of the PLM domain's state-of-the-art, considering the reinterpreted product definition and taking into account the Industry 4.0 expectations, a new PLM approach emerged [94], which is represented by Figure 6.
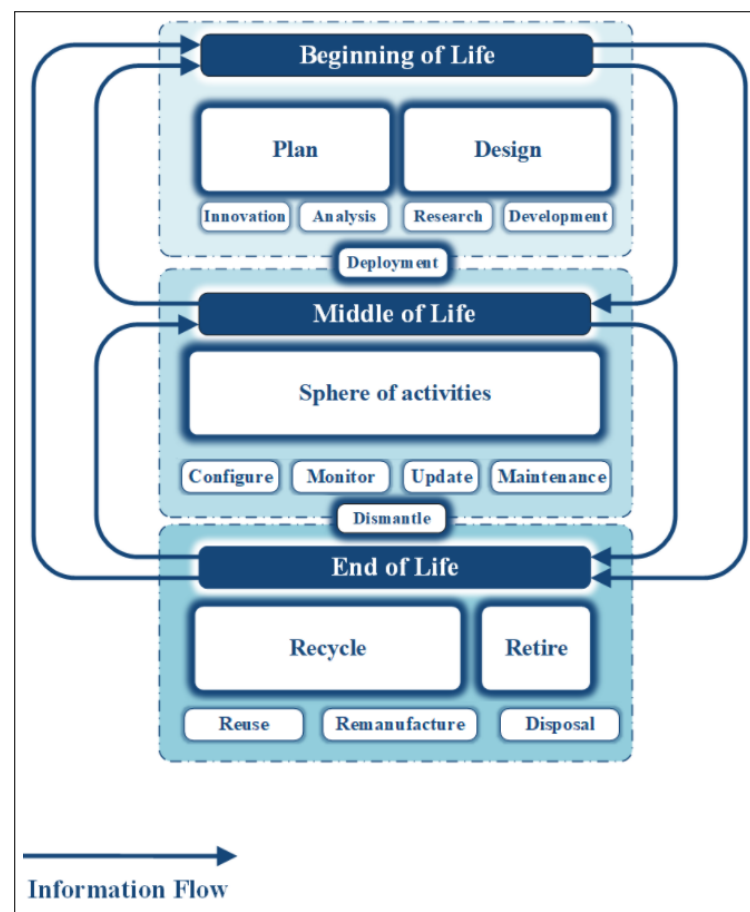

Figure 6. The suggested approach for the new PLM (Product Lifecycle Management) model [94].

The model is formed based on previously described process engineering frameworks and examined PLM models, extended by industrial experiences acknowledging the industrial expert's feedback as well. The model contains three main stages and their related phases and subphases.

5.3. Beginning of Life

The starting point of a product's lifecycle is named the Beginning of Life (BoL). This main stage usually includes the basic phases of the product to be developed, from the idea to its deployment [95]. In the COBIT 5 terminology, the "Information" and "Plan and

Organise" domains deal with the initialization phase, where the emphasis is on the best usage of information and technology reaching the company's goals and objectives. The Val IT portfolio management approach may still be relevant at this stage, as the subphases may reveal wrong considerations or other unexpected events. There are various risks along the lifecycle that must be taken into account as soon as possible, e.g., misleading financial forecasts and wrong research methodologies, insecure code development and all corresponding issues, unexpected events, or install and integration issues. Hence, risk assessment is critical here.

5.3.1. Plan

This phase can be considered the initial point that starts from the product's idea through the needs assessment and market research up to risk analysis.

- Innovation: the first subphase is the trigger event of the *Plan* phase. From the information gathered on the basis of existing products and in response to constantly changing trends, new product ideas emerge, from which new products can eventually be born.
- Analysis: there is a need for estimation and forecast about the product to be developed. The purpose of this subphase is to respond to questions such as whether the product meets customer and market needs, amongst others.

5.3.2. Design

After the *Plan* phase, the *Design* phase follows. If the *Analysis* brought about satisfying results, the product could be designed.

- Research: this subphase deals with defining the precise product functional specification based on the input requirements, taking into account the related architectural considerations and, furthermore, examining the related development techniques.
- Development: when the *Research* subphase is finished and the architectural concept is established, the development can start. In this subphase, the hardware and software components of the product are created.

5.3.3. Deployment

In the COBIT 5 terminology, the "Deliver and Support" domain focuses on the delivery phases of the IT-related artifacts. It covers areas such as accomplishment of the applications within the system and its outcomes as well as the support methods that enable effective and efficient execution of these systems. However, the COBIT 5 terminology handles these processes together, which, from a semantic point of view, is not the optimal one. From the PLM point of view, both concepts can be separated. The delivery is typically a process that occurs once in a product's life, while support is a continuous process that is built up of several parts and connects organically to the MoL (Middle of Life). Therefore, the PLM model proposes the *Deployment* phase as a transitional phase that connects BoL and MoL, where the focus is specifically on bringing the given product to life.

5.4. Middle of Life

After the BoL stage, the product can be deployed so that it comes to "life" and functions in accordance with its operation purpose. In a way, this appears in every process engineering framework and almost in all previous PLM models. In COBIT 5, this is the "Monitor and Evaluate" domain, while at Val IT, the "Value Governance" means something similar. Probably, this stage has the longest duration, where the product is monitored all the time, and according to the current statement, it can be configured, updated, or under maintenance.

5.4.1. Sphere of Activities

Basically, this is the phase where the product is alive. In most cases, this is the longest stage, where the product is constantly monitored and claimed to be configured, updated, or even maintained. The difference in Figure 6, compared to our previously presented

approach [94], is that the *Sphere of activities* is extended, and now, the *Monitor* appears as another subphase.

- Configure: During this subphase, the current system or application gets its configuration settings. In a smart ecosystem, the products can have multiple roles, and according to given tasks, they can be configured. Furthermore, deploying a new product in an existing ecosystem may require further integration steps, which are also handled by this subphase.
- Monitor: The products must be monitored during their lifecycle. On the one hand, necessary information can be gathered for the maintainers or even users about the operation. On the other hand, comprehensive and real-time monitoring can open the way for proactive maintenance by which the product life can be extended and even the operation cost can be reduced.
- Update: With updates, the product can be set to perform its tasks with other skills or enhanced capabilities. For most IoT applications, their internal settings will change sooner or later, e.g., bug fixes or even firmware updates are required. This can be handled with ad hoc or scheduled updates.
- Maintenance: It is also an important and highlightable subphase from the product point of view. The absence of *Maintenance* can cause more serious failures or unexpected downtime, which can also raise unnecessary expenditures in a corporate or industrial environment or can even lead to much earlier retirement of the product.

5.4.2. Dismantle

Similar to the *Deployment* phase, which connects the BoL and MoL stages, another phase is also required for connecting the MoL and EoL (End of Life) main stages. This phase is called *Dismantle* and takes effect when the product officially ceases to operate.

5.5. End of Life

After completion of the MoL, the product is discharged, where multiple use cases are possible. Given the multitude of products, it is very important that a single product goes through the EoL properly, where it must be completely disposed of or, if there is a chance, recycled or recovered. The number of IoT devices connected to the Internet is growing drastically. Hence forgotten, abandoned, or improperly disposed of products can unnecessarily overload the communications network or leave huge cybersecurity vulnerabilities [96]. Therefore, unlike previous lifecycle models, next-generation models need special attention for product EoL management.

5.5.1. Recycling

The market offers more and more product opportunities; therefore, the number of products is growing at an unprecedented rate and cause products to be exchanged more quickly. Although in many cases, a product is no longer able to meet the changing needs, it can still be used for other purposes with modifications.

- Reuse: It may be the case that, although a product is still usable, it will be replaced because a newer product is better suited to perform the expanding tasks given the changing needs. In this case, it is still possible to use the replaced product for a different purpose, for which software upgrades are sufficient.
- Remanufacture: This is similar to reuse, but in this case, the software update is not enough and some hardware improvements are also needed.

5.5.2. Retire

In practice, this is the subphase from which the product is certainly not returned to the lifecycle. However, this subphase usually represents the end of a long lifecycle, which can provide useful information for developing a new product or optimizing a particular product group.

- Disposal: There may be several reasons for withdrawing or replacing a product, which may result from the product no longer being able to meet the expectations or to perform its current function or from simply its support being expired. Nevertheless, the main causes of retirement are usually irreparable damages, inadequate maintenance, or simply technological aging. This subphase means the end of the product's life.

Recognizing the importance of feedback and looking at previous models, the presented PLM approach clearly defines *Information Sharing* as a key, which is an essential criterion for successful Industry 4.0-compliant lifecycle management. In pre-digitization times, there was no effective, real-time opportunity for communication monitoring, but now, it must clearly be a significant part of the lifecycle models. Figure 6 shows the relevant information flow between the main stages and their subphases. With *Information Sharing* and constant feedback to the previous stages, the ergonomics, industrial design, and safety requirements can be continuously enhanced. This is important because, on the one hand, it is often not possible to exactly identify and determine all the risks or specific operational requirements during the design phase. On the other hand, putting the focus on risks, it can be stated that risks' scopes and challenges are continuously changing and expanding quasi day by day. Consequently, it cannot be said that what was designed in the near past corresponds to today's cybersecurity or even functional requirements. The effective reaction is especially important for zero-day attacks where immediate feedback can help identify the problem quickly before it causes mass damage. For this information flow to take place, technological support is needed. Next-generation IoT frameworks and platforms can provide this assistance.

The next section will describe an industrial IoT framework, namely the Arrowhead, which can support the defined PLM-related processes.

6. The Arrowhead Framework

The Arrowhead framework [94,97] is designed to cover interoperability and integration issues for the IIoT world. It supports the collaboration of newly built and legacy CPS architectures based on the principles of SOA by applying the SoS approach. It realizes the local cloud concept empowered by inter-cloud communication capabilities. Each stakeholder has its local cloud(s), as shown in Figure 7, which can act as an SoS.

Their systems implement either intra- or inter-cloud information sharing as well as security and other policies. This type of communication carries several risks that need to be addressed. The Arrowhead framework is practically designed and continually developed to manage cybersecurity risks, amongst others. The risk management of Arrowhead aligns with the Risk IT framework combined with COBIT 5 and ISO 27001 recommendations. Considering [90], threats, vulnerabilities, and risks are continuously identified and managed in accordance with the Risk IT lifecycle, providing CIA (Confidentiality, Availability, and Integrity) to the clouds and their services [98].
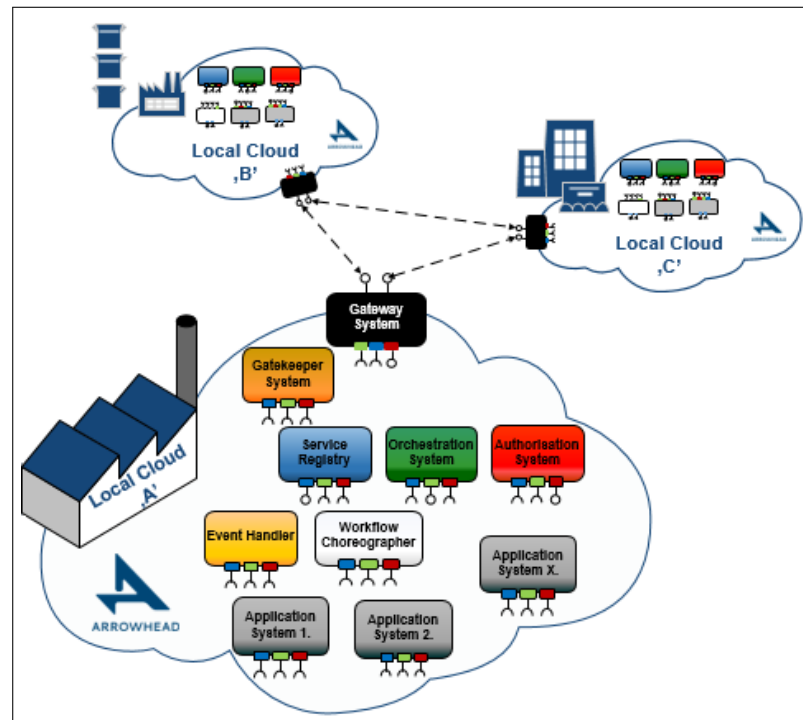
Figure 7. Inter-cloud communication with the Arrowhead framework [82].

6.1. Service Levels of Arrowhead

The Arrowhead framework defines three primary types of systems.

6.1.1. Mandatory Core Systems

The central, mandatory systems of an Arrowhead cloud [97] provide the necessary functionalities of SOA. This group includes the Orchestration System (mainly for service discovery and late binding), Service Registry (service providers can announce their active services), and Authorization System (to provide authorization, authentication, and accounting functions).

6.1.2. Supporting Systems

These systems maintain general services that are often needed in SoS, so integrators do not have to implement their solutions for such standard services. The Arrowhead framework defines various systems on this level, currently including the Gateway and Gatekeeper Systems for inter-cloud communication (data and control plane, respectively), the Workflow Choreographer and Executor (to trigger the next step in the process execution), the Event Handler (to circulate status and event information), and the Plant Description System (to keep track of SoS- or Plant-related meta-data), amongst others.

6.1.3. Application Systems

Typically, the application systems are local cloud-specific, distinct elements of the SoS. These provide (and, in fact, consume) the various application services—in a discoverable, late-bound, loosely coupled way that is defined by the SOA. These are mostly local systems, from the smallest sensors up to the biggest CPSs. Figure 8 describes the presented system levels.
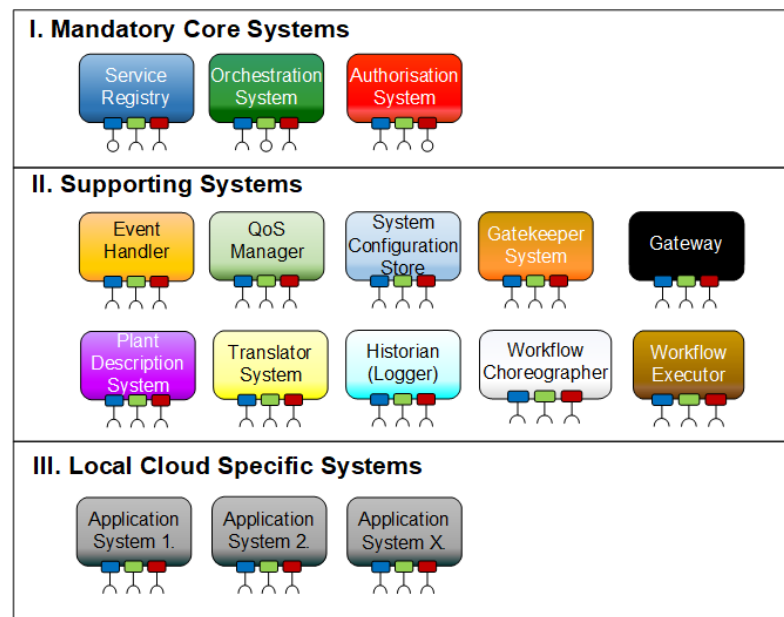
Figure 8. System types within Arrowhead [99].

6.2. Supporting PLM through the Arrowhead Framework

The Arrowhead framework offers solutions regarding the identified issues presented by [100] for the current PLM models, with special attention on how to connect the main lifecycle stages. The product lifecycle model—shown in Figure 6—has distinct elements, which can be interpreted as stages (and phases). Similar to the digital footprint in digital production, digital records of life-stage events can be created and accumulated. Besides their timestamp, these can include meta-data on physical location, environment, configuration, maintenance log, as well as usage statistics. Such meta-data can also be shared with the help of the Arrowhead framework as part of PLM—very similar to that done for the other management processes. All sorts of functions, participants, and stakeholders can obtain valuable, decision-changing information based on these data—including R&D or marketing departments and tertiary users (e.g., used-device providers).

6.3. Challenge: Supporting SoS Processes through the Arrowhead Framework

During development of the PLM model, the main processes were identified and the Arrowhead framework was fine-tuned with them, but new challenges are raised for Arrowhead. As shown, Arrowhead has a modular structure and its systems are constantly changing in line with trends. However, there was no agreed upon strategy or methodology for the management and integration of new systems within the Arrowhead, so the processes of system integration have begun to become opaque. Therefore, the need for an SoS lifecycle modeling methodology has increased, which can describe and understand SoS processes and related management actions. The aim is to develop a model that can typically describe IIoT scenarios. The analysis revealed that none of the lifecycle models fully cover our expectations. The previously created PLM model is the closest model we identify to a feasible solution, so at that point, it was analyzed which changes in the developed PLM model were needed to meet the SoS-specific expectations. The next section introduces the newly created SoSLM model, primarily designed for IIoT use cases.

7. Introducing the System of Systems Lifecycle Management Model

Industry 4.0 requires a modular and flexible factory structure, and its success depends on the consistent development and management of an agile methodology. After studying and testing the presented process engineering and ITSM frameworks and the created PLM model, the conclusion was that, while all are relevant and usable, SoS management and integration require a different approach.

Considering the requirements, the goal is to elaborate a methodology that is able

- to meet the Industry 4.0 requirements;
- to accommodate standard modeling techniques;
- to provide a single framework for the design, implementation, integration, and management of SoS;
- to maintain the hierarchical and modularity expectations;
- to provide tools for the agile and dynamic construction of systems;
- to enable the automatic deployment of systems;
- to enable interoperability and integrability for heterogeneous systems; and
- to enable an SOA guaranteeing adaptable, loosely coupled, and late-bound services

and that

- can be measured and evaluated and
- can adjust to new trends and changes.

In the following sections, we present a new methodology, i.e., System of Systems Lifecycle Management model (SoSLM), based on the developed PLM model described above. To demonstrate its performance, real service development and integration are also presented, using the defined SoSLM model.

7.1. Forming the SoSLM Methodology

The Arrowhead—presented in Section 6—is an ever-expanding framework that continuously introduces new systems or updates existing ones in line with the latest industry trends. As the expectations and requirements associated with Industry 4.0 frequently change [101], a methodology to support the formation and introduction of new systems needs to be developed. As previously emphasized, Arrowhead is SOA-based, built by different levels of systems (Section 6.1). These systems offer services that are used by other systems, internally (local cloud) or even externally (inter-cloud). These are the initial conditions on which the SoSLM approach are developed.

The primary function of the systems is thus servicing; therefore, the approach of the ITIL service cycle can be the basis and core part of the new model, and it can be considered the main SoSLM cycle. Based on the process engineering frameworks presented by Section 3, an IIoT-specific PLM model is introduced in Section 5. Arrowhead was built by taking into account the main elements of the mentioned PLM approach. In the design of SoSLM, the elements defined in the PLM model can be associated with the SoSLM main cycles and they can be integrated as the SoSLM model's subcycles.

However, the systems to be developed can be diverse and different in functionality, so it is essential that the main processes—along with the defined main and subcycles—are correctly identified. For a system to be integrated, it is a must that the developers, operators, users, or even other systems use the same tool sets for provision of the previously identified system-specific processes. To this end, a fourth part should also be designed, which brings participants to the same understanding of the tools to be used. The fourth ring can also be considered a kind of toolchain, which is a topic continuously occupying the research community [102,103]. In line with this trend, the SoSLM also supports the description of toolchains.

Figure 9 demonstrates the results of the utilized lifecycle management approach. The model is a concentric circle of four circles, where the rings are constant in terms of their functionality.
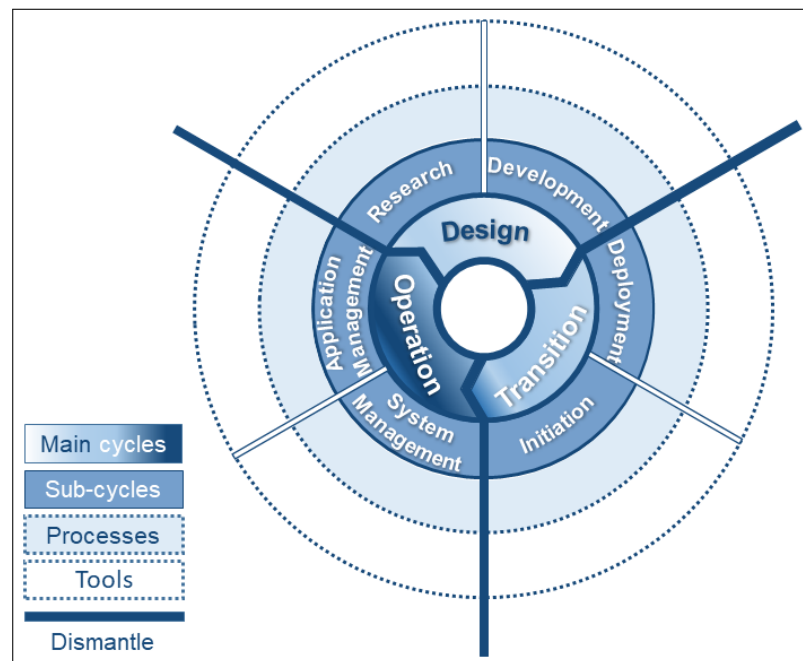
Figure 9. The basic SoSLM (System of Systems Lifecycle Management) model.

7.1.1. Main Cycles

The innermost circle refers to the main cycles, i.e., the design, transition, and operation, aligned with the ITIL service cycle.

7.1.2. Subcycles

The subcycles accommodate the second circle from inside. Here, the main elements come from the already presented PLM model, i.e., research, development, deployment, system management, and application management. However, a new subcycle can also be observed here, and that is the "Initiation" subcycle under the main transition cycle. Once deployment is complete, the initial configurations, settings, and necessary inputs must be available. Additionally, it is a mandatory requirement that correct operation of the delivered system is tested and validated. Thus, this subcycle is the last point where hidden faults and other unexpected behaviors can be revealed before the activity begins, so it is a critical stage in the aspect of risk management as well. The two inner circles are bound in terms of terminology, which means that these main and subcycle terminologies must appear for each SoSLM model.

7.1.3. Processes

Each subcycle can have several processes, depending on the functionality of the particular system. Processes thus cannot be adequately defined universally as in the case of main and subcycles. Based on the presented PLM model, processes can be—including but not limited to—for example, monitoring, updates, configuration, or maintenance.

7.1.4. Tools

For each process, it is crucial to determine the tool(s) to be used. The outside circle carries this information, which facilitates transparent operation between heterogeneous systems. Moreover, this circle serves as a development toolchain, which is a useful feature for the participants.

7.1.5. Dismantle

Proper care of the dismantle is essential. Typically, this is possible between the main cycles, where any of the processes described in the PLM model (reuse, remanufacture, or disposal) can be applied depending on the purpose.

7.2. An Implemented System Integration for Arrowhead Using the Newly Developed SoSLM Methodology

The Arrowhead framework was designed and developed to satisfy precisely the needs of Industry 4.0, supporting effective and secure service sharing through the SOA approach. Now, the development of the Arrowhead framework is in that stage where it can already satisfy the requirements of automated production. While addressing automated production issues through the Arrowhead framework, several alternatives came alive [104,105]. These approaches defined the critical elements of the automated production, i.e., implemented following the SOA principles, using distributed systems and resources, and based on predefined but dynamically changing state machines.

First, it needs to be validated that these kind of features and attributes can work together and that they can be applied within the Arrowhead framework. In order to verify this—and as a proof-of-concept—a prototype was built. The aim was to make sure, on the one hand, that Arrowhead can manage these features, and on the other hand, to see what architectural, design, and development criteria must be taken into account for the finalized, technology-independent workflow engine of the Arrowhead, namely the Workflow Choreographer described by [99].

During development of this prototype, the described SoSLM was used, and the formed model is shown in Figure 10 and can serve as a practical example and as a validation of the presented SoSLM. The details of this demonstration use case are presented by [87].
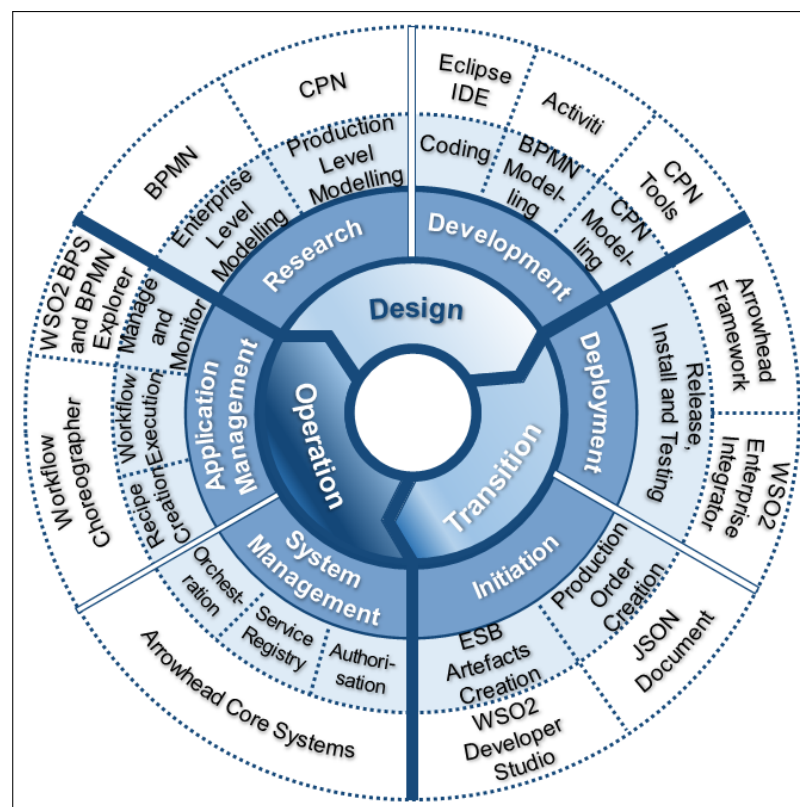

Figure 10. A practical example of system of systems lifecycle management—the main cycles, subcycles, processes, and tools used within the proposed dynamic, multi-level workflow management concept [87].

7.2.1. Design

To present a comprehensive management model, the design main cycle must be subdivided into other subcycles. Our approach suggests extending this using the described PLM processes in Section 5.3.2 to cover the precisely identified gaps.

- Research: for IoT application development, the first step is to identify the requirements and, based on them, find the optimum solution for the sphere of activities. The earlier presented case [99] was a bit complex because our workflow management approach addresses two kinds of workflow levels (enterprise and production); therefore, in our case, the research subcycle is about the selection of the appropriate workflow modeling languages. This separation subdivides the research subcycle into two further processes on the third ring, and accordingly, the selected languages can be seen at the fourth ring.
- Development: after the requirement specification, development can start. As it can be seen in Figure 10, the inputs of the third and fourth rings of the development subcycle are the outputs of the similar rings of the research subcycle. Additionally, appropriate development environments are chosen. In the current case, the development phase is split into three other processes: one environment for development and two further tools for the modeling languages.

7.2.2. Transition

The transition subcycle is basically about how the developed application can be brought to life. This partly aligns with the deployment stage of our PLM model.

- Deployment: now, the application is fit for deployment. Here, the target systems are the described IIoT framework, the Arrowhead, and the exemplary chosen WSO2 (Web Services Oxygenated 2) communication infrastructure, which can establish a connection between SOA-based environments [106].
- Initiation: mostly, this involves the operating environment's parameters and the necessary inputs of the developed application. In the current case, the developed workflow engine needs, on the one hand, a production document as input and, on the other hand, the address and other attributes and necessary artifacts of the chosen delivery platform.

7.2.3. Operation

This is the most significant lifecycle phase, which is the MoL according to the PLM model presented in Section 5, i.e., the *Sphere of activities*, where the application will exist and operate in the SoS architecture during its lifecycle. Most of the time, the newly developed applications are integrated into an existing system infrastructure. According to the Industry 4.0 expectations, the system must satisfy the flexibility and modularity requirements. As mentioned earlier, the *Sphere of activities* is a collective term, where the concept is that the subphases should always be formed according to the type of lifecycle model used. To manage SoS, the *Sphere of activities* is transformed into two further subcycles where the determinant lifecycle processes are the same as in the MoL stage of the introduced PLM model.

- System management: the presented model approaches the industrial-specific PLM strategy from a holistic view, assuming that the heterogeneous products can be organized and integrated into larger groups or systems and managed as SoS. In this case, it is crucial to distinguish the management tasks on the different levels of the SoS. System management refers to SoS-related activities. On this level, the emphasis is on the tasks of the main systems and creating the connection and other inter-cloud-related events, such as network management with remote clouds and managing their subsystems or CPSs. To support proper functioning, the system must be able to handle the integration and management of a newly installed application. In our case, the new application is integrated into the Arrowhead framework, where the regarding services—orchestration, service registry, and authorization—will be provided by the Arrowhead Core Systems.
- Application management: the smallest units, i.e., the smallest operating systems or applications, are the building blocks of the SoS, which also require careful management. This is the subcycle where the developed application operates. Its main processes are

identified on the third ring and assigned to the functions on the fourth ring. In the current implementation, the newly integrated application's workflow-specific features are monitored and managed by the used WSO2 infrastructure.

7.3. Continuous Enhancement, Improvement, and Risk Management

The circular model illustrates well that this is a continuous development strategy, where the information gathered during the cycles can serve as useful inputs for the management of further applications or even the enhancement and improvement of the current application. During the whole lifecycle, based on the gathered information and feedback, the risks are also taken into account; therefore, they can be continuously treated.

8. Discussion on Evaluation

The purpose of creating the SoSLM model was twofold: on the one hand, addressing the gaps of SoS-related issues raised during the development of the Arrowhead framework and, on the other hand, ensuring compliance with the presented Industry 4.0 requirements. In the end, a model that is modular and flexible was developed—and proposed here—helping to integrate new systems, to manage existing systems, and to retire or rebuild old systems. In the next subsections, the benefits and limitations of SoSLM are detailed, first in the context of Arrowhead and then extended in general compared to other lifecycle models. In addition, a universal, Industry 4.0-compatible PLM model was developed during the SoSLM design, which can serve as a later basis for other lifecycle models.

8.1. Benefits and Limitations of SoSLM within Arrowhead

Several challenges arose during the development of the Arrowhead framework, which initiated the generalized design of the presented SoSLM model. The conceptualization and development of the Arrowhead framework include the appearance of new systems, the transformation of existing systems, and the removal of obsolete systems. Although initially only a few systems were present, now, there are many more systems within the framework, and this number is dynamically growing.

Therefore, it became a significant challenge to manage the ever-changing Arrowhead ecosystem [83,87]. Stakeholders were taking into account new requirements in various subcycles related to various systems. Examples in the *Research* and *Development* subcycles include the refinement of secure inter-cloud servicing [98,107], the Event Handler [108], or improving the handling of the Quality of Service [109]. Examples in the *Initiation* and *Deployment* subcycles include integration and interoperability issues of new systems, such as the Workflow Choreographer [82,87,105], among others. There are integrative examples for the *System and Application Management* subcycles, such as the communication of the operational local cloud, the Arrowhead Management Tool, and system modeling tools [110]. The model also guides developers and operators of the given system's dependencies, either on the used development tools or systems within the Arrowhead framework (addressed in the fourth, i.e., "tools" ring of SoSLM).

However, there are also factors for which the model does not provide an answer. One such area is source code management [111], in which case the model includes information about the used tools that can help in the reverse and forward engineering processes. However, it does not address issues such as code structuring and documentation or methodology used during code development, which can definitely influence the system's operation. It is also recognized [112] that the model-driven development could support the safety run-time, of which the approach is also implemented by SoSLM. Furthermore, [113] have identified the critical role—and based on the advantages—of effective maintenance. Here, the SoSLM model helps in the aspect of necessary development that must be implemented if there is any change in the system's serving infrastructure. In addition to these, there are more general possibilities for using the model (outside Arrowhead), of which benefits and limitations are present by the next subsection.

8.2. Benefits and Limitations of SoSLM in General

One of the advantages of the model is that, although it is a new approach, its core is provided by the general SLM and PLM models, and process engineering frameworks, so it is easy to migrate from those models in many areas. Of course, such migrations or model replacements have to take context-specific issues into account. Still, with minor additions, SoSLM can serve as a complete alternative to these two models. Due to its universal nature, in addition to systems, several other product types can also be inserted into the SoSLM model.

Another advantage is that, following the model, every participant is aware of when, how, and what system, application, or different products should be used, leading to more efficient supply chain processes.

Although the model provides a continuous flow of information, it is essential to note that information management is not as detailed as in ILM models. SoSLM approaches the issue of information management from the perspective of technological infrastructure. In its currently proposed version, it does not deal specifically with, e.g., the handling of personal or sensitive personal data. However, this requirement is not so typical in this part of the SoS architecture and must usually appear in the particular system's functional specification. The focus of information management is more on the processes affecting the technology infrastructure and the operating systems, creating the opportunity for continuous optimization and proactive maintenance and addressing threats and vulnerabilities affecting the SoS.

It can also be stated that SoSLM itself is also a kind of SDLC, as it carries information about the main processes and subprocesses involved in the development of a system, although it is not detailed as an SDLC so far. However, the SDLC can be considered from the SoS perspective rather than a developer perspective.

Furthermore, SoSLM makes the tools used for development and operation available in a transparent way, which is a great advantage for describing the specific SoS toolchain.

9. Conclusions

The main novelty of the paper is that it introduces a newly developed SoSLM approach. To understand the motivation, the related literature was discussed, highlighting the main gaps in existing lifecycle management models. We performed a comprehensive analysis of the most widely used process engineering methodologies and ITSM frameworks, and using the results, in the first step, we developed a generic PLM model according to the reinterpreted product definition and, in the second step, based on the created PLM model, a SoSLM model was defined to meet the identified SoS-related requirements. The resulting model was enhanced with the service-oriented approach.

Hence, the paper presents qualitative and quantitative research artifacts as well. The PLM model yielded quantitative research, where the old PLM approaches were enhanced and product lifecycle phases were fine-tuned according to Industry 4.0 expectations. Still, the real novelty here is the appearance of a new core feature in the model, the information sharing capability that underlies Industry 4.0-compliant systems. Reflecting on this, a further important finding of the research work is the use of IoT-based digital frameworks and platforms for achieving new features and managing phases during the whole lifecycle. The SoSLM model can be considered a qualitative artifact, which is the paper's main topic, and solves the SoS-related lifecycle management problems. The SoSLM model, on the one hand, integrates SLM and PLM models, complemented by the ability of information sharing. On the other hand, it adds industry best practices to address the area-specific problems presented in gap analysis.

The SoSLM model is verified by a presented use case; furthermore, operation of the developed system was also validated in real circumstances using the Arrowhead framework that addresses interoperability and integrability issues primarily for the IIoT domain. The model covers the needs of SOA and SLM, thanks to the ITIL terminology, but it is also extended by the main PLM lifecycle phases and processes, i.e., from research

through development, deployment, continuous management, and monitoring up to the information feedback.

In general, the paper can help the research community as well as the system operators in several areas. On the one hand, it provides useful and comprehensive surveys on lifecycle management, process engineering, and ITSM frameworks; furthermore, it can serve as useful guidance for creating specific lifecycle management approaches. On the other hand, it includes an enhanced PLM model and a newly designed SoSLM model that can help operators in system management tasks of smart ecosystems, mainly in the SOA context.

In addition to the benefits, the paper also covered the limitations of the SoSLM model. The SoSLM models the lifecycle from an architectural point of view; therefore, it does not handle, for example, the detailed description of coding techniques. The same is true for the flow of information. Due to the SoSLM model's nature, continuous information sharing can be provided, but the SoSLM does not formulate a specific methodology for that particular function. Thus, compared with the presented lifecycle models, it can be stated that both SoSLM and the newly developed PLM models can be applied to several levels in different areas. Still, in some cases, they do not answer the context-specific questions in detail.

Regarding use, the model has been applied mostly on prototype or beta implementations. Thus, just as the current form of the model has evolved, one of the future works is still to apply and observe the model in more production environments, by which it can be further developed. Moreover, Industry 4.0 elements are still under standardization; therefore, the requirements for PLM and SoSLM could change in the future. The models must adapt to changing industrial trends and the actual relevance of the presented models needs further and continuous analysis and tests.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

ALM	Application Lifecycle Management
BoL	Beginning of Life
CIA	Confidentiality, Integrity and Availability
COBIT	Control Objectives for Information and related Technologies
CPPS	Cyber Physical Production System
CPS	Cyber Physical System
DSRM	Design Science Research Methodology
EoL	End of Life
ERP	Enterprise Resource Planning
GDPR	General Data Protection Regulation
ICT	Information and Communication Technology
IIoT	Industrial Internet of Things
ILM	Information Lifecycle Management
IoT	Internet of Things
ISACA	Information Systems Audit and Control Association
IT	Information Technology
ITGI	Information Technology Governance Institute
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
PCI DSS	Payment Card Industry Data Security Standard
PLM	Product Lifecycle Management
SCM	Supply Chain Management
SDLC	Software Development Life Cycle
SLM	Service Lifecycle Management
SOA	Service Oriented Architecture
SoS	System of Systems
SoSLM	System of Systems Lifecycle Management
Val IT	Value from IT investments
WSO2	Web Services Oxygenated 2

References

1. Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [CrossRef]
2. Hermann, M.; Pentek, T.; Otto, B. Design Principles for Industrie 4.0 Scenarios. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016. [CrossRef]
3. *Smart Manufacturing. Reference Architecture Model Industry 4.0 (RAMI4.0)*; British Standards Institution: London, UK, 2017. [CrossRef]
4. Kaye, D. *Loosely Coupled: The Missing Pieces of Web Services*; RDS Press: Kentfield, CA, USA, 2003.
5. Erl, T. *Service-Oriented Architecture*; Pearson Education (US): Upper Saddle River, NJ, USA, 2017.
6. Duarte, F.L.; de Castro, A.F.; Queiroz, P.G.G. *Reap-SoS: A Requirement Engineering Approach for System of Systems*; AIRCC Publishing Corporation: Copenhagen, Denmark, 2018. [CrossRef]
7. Heitmann, F.; Pahl-Wostl, C.; Engel, S. Requirements Based Design of Environmental System of Systems: Development and Application of a Nexus Design Framework. *Sustainability* **2019**, *11*, 3464. [CrossRef]
8. Vierhauser, M.; Rabiser, R.; Cleland-Huang, J. From Requirements Monitoring to Diagnosis Support in System of Systems. In *Requirements Engineering: Foundation for Software Quality*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 181–187. [CrossRef]
9. *Delivering Value to Today's Digital Enterprise*; Forbes Insight: Jersey City, NJ, USA, 2017.
10. Mora, M.; Raisinghani, M.; OConnor, R.V.; Gomez, J.M.; Gelman, O. An Extensive Review of IT Service Design in Seven International ITSM Processes Frameworks - Part I. *Int. J. Inf. Technol. Syst. Approach* **2014**, *7*, 83–107. [CrossRef]
11. Mora, M.; Gomez, J.M.; OConnor, R.V.; Raisinghani, M.; Gelman, O. An Extensive Review of IT Service Design in Seven International ITSM Processes Frameworks - Part II. *Int. J. Inf. Technol. Syst. Approach* **2015**, *8*, 69–90. [CrossRef]
12. MANTIS. Available online: http://www.mantis-project.eu/ (accessed on 8 April 2021).
13. Productive 4.0. Available online: https://productive40.eu/ (accessed on 8 April 2021).
14. Arrowhead Tools. Available online: https://productive40.eu/ (accessed on 8 April 2021).
15. Lakshmanan, A.; McBrien, A.; Zhang, J.; Dhole, V. Transformation of Process Engineering—A Software Perspective. In *Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design*; Elsevier: Amsterdam, The Netherlands, 2014; pp. 186–195. [CrossRef]
16. de P. Soares, R.; Secchi, A. EMSO: A new environment for modelling, simulation and optimisation. In *Computer Aided Chemical Engineering*; Elsevier: Amsterdam, The Netherlands, 2003; pp. 947–952. [CrossRef]

17. Scheer, A.W. *Business Process Engineering: Reference Models for Industrial Enterprises*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.

18. Nguyen, V.H.; Kolp, M.; Wautelet, Y.; Heng, S. Mapping IT Governance to Software Development Process: From COBIT 5 to GI-Tropos. In *International Conference on Enterprise Information Systems (ICEIS) (2)*; SciTePress: Funchal, Madeira, Portugal, 2018; pp. 665–672.

19. Amid, A.; Moradi, S. A Hybrid Evaluation Framework of CMM and COBIT for Improving the Software Development Quality. *J. Softw. Eng. Appl.* **2013**, *6*, 280–288. [CrossRef]

20. Sahibudin, S.; Sharifi, M.; Ayat, M. Combining ITIL, COBIT and ISO/IEC 27002 in Order to Design a Comprehensive IT Framework in Organizations. In Proceedings of the 2008 Second Asia International Conference on Modelling & Simulation (AMS), Kuala Lumpur, Malaysia, 13–15 May 2008. [CrossRef]

21. Scheer, A.W.; Nüttgens, M. *ARIS Architecture and Reference Models for Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2000. [CrossRef]

22. de Sousa Pereira, R.F.; da Silva, M.M. A Maturity Model for Implementing ITIL v3. In Proceedings of the 2010 6th World Congress on Services, Miami, FL, USA, 5–10 July 2010. [CrossRef]

23. Sharifi, M.; Ayat, M.; Rahman, A.A.; Sahibudin, S. Lessons learned in ITIL implementation failure. In Proceedings of the 2008 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 26–28 August 2008. [CrossRef]

24. Dennis, A.; Wixom, B.; Tegarden, D. The System Development Life Cycle. In *Systems Analysis & Design: An Object-Oriented Approach with UML*; John Wiley & Sons Inc.: New York, NY, USA, 2015; pp. 3–7.

25. Centers for Medicare & Medicaid Services. *Selecting a Development Approach*; Office of Information Technology: Baltimore, MD, USA, 2005.

26. Schwaber, K.; Beedle, M. *Agile Software Development with Scrum*; Pearson Education (US), Prentice Hall: Upper Saddle River, NJ, USA, 2002; Volume 1.

27. Guntamukkala, V.; Wen, H.J.; Tarn, J.M. An empirical study of selecting software development life cycle models. *Hum. Syst. Manag.* **2006**, *25*, 265–278.

28. Jain, R.; Suman, U. A systematic literature review on global software development life cycle. *ACM SIGSOFT Softw. Eng. Notes* **2015**, *40*, 1–14. [CrossRef]

29. Arya, A.; Böhm, M.; Bose, B.; Cerveau, L.; Endholz, P.; Geier, F.; Krause, M.R.; Krcmar, H.; Leimeister, S.; Madhukar, I.; et al. *Application Management: Challenges-Service Creation-Strategies*; Springer: Berlin/Heidelberg, Germany, 2011.

30. Linnartz, W. *Application Management Services und Support*; Publicis: Paris, France, 2004.

31. Hüttermann, M. *Agile Application Lifecycle Management*; Manning Publication: New York, NY, USA, 2011.

32. Tüzün, E.; Tekinerdogan, B.; Macit, Y.; İnce, K. Adopting integrated application lifecycle management within a large-scale software company: An action research approach. *J. Syst. Softw.* **2019**, *149*, 63–82. [CrossRef]

33. Kääriäinen, J.; Välimäki, A. Applying application lifecycle management for the development of complex systems: Experiences from the automation industry. In *European Conference on Software Process Improvement*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 149–160.

34. Lacheiner, H.; Ramler, R. Application lifecycle management as infrastructure for software process improvement and evolution: Experience and insights from industry. In Proceedings of the 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications, Oulu, Finland, 30 August–2 September 2011; pp. 286–293.

35. Kääriäinen, J.; Välimäki, A. Impact of Application Lifecycle Management—A Case Study. In *Enterprise Interoperability III*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 55–67.

36. Tanaka, T.; Ushijima, K.; Ueda, R.; Naitoh, I.; Aizono, T.; Komoda, N. Proposal and Evaluation of Policy Description for Information Lifecycle Management. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005. [CrossRef]

37. Hu, W. Information lifecycle modeling framework for construction project lifecycle management. In Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering, Leicestershire, UK, 20 November 2008; pp. 372–375.

38. Lee, J.H.; Song, J.H.; Oh, K.S.; Gu, N. Information lifecycle management with RFID for material control on construction sites. *Adv. Eng. Inform.* **2013**, *27*, 108–119. [CrossRef]

39. Chen, Y. Information valuation for information lifecycle management. In Proceedings of the Second International Conference on Autonomic Computing (ICAC'05), Seattle, WA, USA, 13–16 June 2005; pp. 135–146.

40. Petrocelli, T. *Data Protection and Information Lifecycle Management*; Prentice Hall PTR: Hoboken, NJ, USA, 2005.

41. Stark, J. Product Lifecycle Management. In *Decision Engineering*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 1–29. [CrossRef]

42. Cao, H.; Folan, P.; Mascolo, J.; Browne, J. RFID in product lifecycle management: A case in the automotive industry. *Int. J. Comput. Integr. Manuf.* **2009**, *22*, 616–637. [CrossRef]

43. Stark, J. *Product Lifecycle Management (Volume 4): The Case Studies*; Springer: Berlin/Heidelberg, Germany, 2019.

44. Kohlborn, T.; Korthaus, A.; Rosemann, M. Business and Software Service Lifecycle Management. In Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference, Auckland, New Zealand, 1–4 September 2009. [CrossRef]

45. Fischbach, M.; Puschmann, T.; Alt, R. Service Lifecycle Management. *Bus. Inf. Syst. Eng.* **2013**, *5*, 45–49. [CrossRef]

46. Yelmo, J.C.; Trapero, R.; del Álamo, J.M.; Sienel, J.; Drewniok, M.; Ordás, I.; McCallum, K. User-driven service lifecycle management–adopting internet paradigms in telecom services. In *International Conference on Service-Oriented Computing*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 342–352.

47. Kunz, S.; Brecht, F.; Fabian, B.; Aleksy, M.; Wauer, M. Aletheia–improving industrial service lifecycle management by semantic data federations. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Australia, 20–23 April 2010; pp. 1308–1314.

48. Peruzzini, M.; Germani, M.; Marilungo, E. Product-service lifecycle management in manufacturing: An industrial case study. In *IFIP International Conference on Product Lifecycle Management*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 445–454.

49. Wiesner, S.; Freitag, M.; Westphal, I.; Thoben, K.D. Interactions between service and product lifecycle management. *Procedia Cirp* **2015**, *30*, 36–41. [CrossRef]

50. Mahut, F.; Bricogne, M.; Daaboul, J.; Eynard, B. Servicization of product lifecycle management: Towards service lifecycle management. In *IFIP International Conference on Product Lifecycle Management*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 321–331.

51. Gunnarsson, Á.; Johnson, M. Application Lifecycle Management. In *Pro Microsoft Power BI Administration*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 63–78.

52. Leng, J.; Ruan, G.; Jiang, P.; Xu, K.; Liu, Q.; Zhou, X.; Liu, C. Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey. *Renew. Sustain. Energy Rev.* **2020**, *132*, 110112. [CrossRef]

53. Liu, X.; Wang, W.; Guo, H.; Barenji, A.V.; Li, Z.; Huang, G.Q. Industrial blockchain based framework for product lifecycle management in industry 4.0. *Robot. Comput. Integr. Manuf.* **2020**, *63*, 101897. [CrossRef]

54. Ashmore, R.; Calinescu, R.; Paterson, C. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *arXiv* **2019**, arXiv:1905.04223.

55. Miao, H.; Li, A.; Davis, L.S.; Deshpande, A. Towards Unified Data and Lifecycle Management for Deep Learning. In Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April 2017. [CrossRef]

56. Hummer, W.; Muthusamy, V.; Rausch, T.; Dube, P.; El Maghraoui, K.; Murthi, A.; Oum, P. Modelops: Cloud-based lifecycle management for reliable and trusted ai. In Proceedings of the 2019 IEEE International Conference on Cloud Engineering (IC2E), Prague, Czech Republic, 24–27 June 2019; pp. 113–120.

57. Voet, M.A. *The Generic Challenge: Understanding Patents, FDA and Pharmaceutical Life-Cycle Management*; BrownWalker Press: Irvine, CA, USA, 2020.

58. Bondan, L.; Franco, M.F.; Marcuzzo, L.; Venancio, G.; Santos, R.L.; Pfitscher, R.J.; Scheid, E.J.; Stiller, B.; De Turck, F.; Duarte, E.P.; et al. FENDE: Marketplace-based distribution, execution, and life cycle management of VNFs. *IEEE Commun. Mag.* **2019**, *57*, 13–19. [CrossRef]

59. Sarrigiannis, I.; Ramantas, K.; Kartsakli, E.; Mekikis, P.V.; Antonopoulos, A.; Verikoukis, C. Online VNF Lifecycle Management in an MEC-Enabled 5G IoT Architecture. *IEEE Internet Things J.* **2019**, *7*, 4183–4194. [CrossRef]

60. Maier, M.W. Architecting principles for systems-of-systems. *Syst. Eng.* **1998**, *1*, 267–284. [CrossRef]

61. ISACA. *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*; Information Systems Audit and Control Association: Schaumburg, IL, USA, 2012.

62. ITGI. *Enterprise Value: Governance of IT Investments, The Val IT Framework 2.0 Extract*; ISACA: Schaumburg, IL, USA, 2008.

63. ISACA. *The Risk IT Framework Excerpt*; Information Systems Audit and Control Association: Schaumburg, IL, USA, 2009.

64. Lainhart, J.W. COBIT™: A Methodology for Managing and Controlling Information and Information Technology Risks and Vulnerabilities. *J. Inf. Syst.* **2000**, *14*, 21–25. [CrossRef]

65. Vacca, J.R. *Computer and Information Security Handbook*; Morgan Kaufmann: Burlington, MA, USA, 2012

66. Disterer, G. ISO/IEC 27000, 27001 and 27002 for Information Security Management. *J. Inf. Secur.* **2013**, *4*, 92–100. [CrossRef]

67. Adams, S. *ITIL V3 Foundation Handbook*; The Stationery Office: London, UK, 2009.

68. Ribeiro, J.; Gomes, R. IT governance using COBIT implemented in a high public educational institution: A case study. In *Proceedings of the 3rd International Conference on European Computing Conference*; World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, WI, USA, 2009; pp. 41–52.

69. Ukidve, A.; SMantha, D.S.; Tadvalka, M. Analysis of Payment Card Industry Data Security Standard [PCI DSS] Compliance by Confluence of COBIT 5 Framework. *Int. J. Eng. Res. Appl.* **2017**, *7*, 42–48. [CrossRef]

70. Wolden, M.; Valverde, R.; Talla, M. The effectiveness of COBIT 5 Information Security Framework for reducing Cyber Attacks on Supply Chain Management System. *IFAC-Pap.* **2015**, *48*, 1846–1852. [CrossRef]

71. Warren, M.; Hutchinson, W. Cyber attacks against supply chain management systems: A short note. *Int. J. Phys. Distrib. Logist. Manag.* **2000**, *30*, 710–716. [CrossRef]

72. Savtschenko, M.; Schulte, F.; Voß, S. IT governance for cyber-physical systems: The case of Industry 4.0. In *International Conference of Design, User Experience, and Usability*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 667–676.

73. Dewi, R.S. Maturity Level Assessment for ERP Systems Investment Using Val IT Framework. *Procedia Comput. Sci.* **2019**, *161*, 250–257. [CrossRef]

74. Wilkin, C.; Campbell, J.; Moore, S.; Van Grembergen, W. Co-creating value from IT in a contracted public sector service environment: Perspectives on COBIT and Val IT. *J. Inf. Syst.* **2013**, *27*, 283–306. [CrossRef]

75. Andry, J.F. Audit of IT Governance Based on COBIT 5 assessments: A case study. *J. Nas. Teknol. Sist. Inf.* **2016**, *2*, 27–34. [CrossRef]

76. Astuti, H.M.; Muqtadiroh, F.A.; Darmaningrat, E.W.T.; Putri, C.U. Risks assessment of information technology processes based on COBIT 5 framework: A case study of ITS service desk. *Procedia Comput. Sci.* **2017**, *124*, 569–576. [CrossRef]

77. Spremic, M.; Zmirak, Z.; Kraljevic, K. IT and business process performance management: Case study of ITIL implementation in finance service industry. In Proceedings of the ITI 2008—30th International Conference on Information Technology Interfaces, Cavtat, Croatia, 23–26 June 2008; pp. 243–250.

78. Yao, Z.; Wang, X. An ITIL based ITSM practice: A case study of steel manufacturing enterprise. In Proceedings of the 2010 7th International Conference on Service Systems and Service Management, Tokyo, Japan, 28–30 June 2010; pp. 1–5.

79. Shang, S.S.; Lin, S.F. Barriers to implementing ITIL-a multi-case study on the service-based industry. *Contemp. Manag. Res.* **2010**, *6*. [CrossRef]

80. Kusumah, P.; Sutikno, S.; Rosmansyah, Y. Model design of information security governance assessment with collaborative integration of COBIT 5 and ITIL (case study: INTRAC). In Proceedings of the 2014 International Conference on ICT For Smart Society (ICISS), Bandung, Indonesia, 24–25 September 2014; pp. 1–6.

81. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]

82. Kozma, D.; Varga, P.; Szabó, K. Achieving Flexible Digital Production with the Arrowhead Workflow Choreographer. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 4503–4510. [CrossRef]

83. Aldalur, I.; Illarramendi, M.; Larrinaga, F.; Perez, T.; Sáenz, F.; Unamuno, G.; Lazkanoiturburu, I. Advantages of Arrowhead Framework for the Machine Tooling Industry. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 4511–4518.

84. Larrinaga, F.; Aldalur, I.; Illarramendi, M.; Iturbe, M.; Perez, T.; Unamuno, G.; Salvidea, J.; Lazkanoiturburu, I. Analysis of Technological Architectures for the New Paradigm of the Industry 4.0. *DYNA* **2019**, *94*, 267–271. [CrossRef]

85. Larrinaga, F.; Fernandez-Anakabe, J.; Zugasti, E.; Garitano, I.; Zurutuza, U.; Olaizola, J.; Anasagasti, M.; Mondragon, M. A Big Data implementation of the MANTIS reference architecture for predictive maintenance. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2019**, *233*, 1361–1375. [CrossRef]

86. Papa, G.; Poklukar, Š.; Frankó, A.; Sillitti, A.; Kancilija, A.; Šterk, M.; Hegedűs, C.; Moldován, I.; Varga, P.; Riccardi, M.; et al. Improving the Maintenance of Railway Switches through Proactive Approach. *Electronics* **2020**, *9*, 1260. [CrossRef]

87. Kozma, D.; Varga, P.; Larrinaga, F. Dynamic Multilevel Workflow Management Concept for Industrial IoT Systems. *IEEE Trans. Autom. Sci. Eng.* **2020**, 1–13. [CrossRef]

88. Venanzi, R.; Montori, F.; Bellavista, P.; Foschini, L. Industry 4.0 Solutions for Interoperability: A Use Case about Tools and Tool Chains in the Arrowhead Tools Project. In Proceedings of the 2020 IEEE International Conference on Smart Computing (SMARTCOMP), Bologna, Italy, 14–17 September 2020. [CrossRef]

89. Delsing, J. Local Cloud Internet of Things Automation: Technology and Business Model Features of Distributed Internet of Things Automation Solutions. *IEEE Ind. Electron. Mag.* **2017**, *11*, 8–21. [CrossRef]

90. Varga, P.; Plosz, S.; Soos, G.; Hegedus, C. Security threats and issues in automation IoT. In Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, 31 May–2 June 2017. [CrossRef]

91. Jantunen, E.; Zurutuza, U.; Ferreira, L.L.; Varga, P. Optimising maintenance: What are the expectations for Cyber Physical Systems. In Proceedings of the 2016 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC), Vienna, Austria, 11 April 2016. [CrossRef]

92. Yao, X.; Zhou, J.; Lin, Y.; Li, Y.; Yu, H.; Liu, Y. Smart manufacturing based on cyber-physical systems and beyond. *J. Intell. Manuf.* **2017**, *30*, 2805–2817. [CrossRef]

93. Kozma, D.; Soos, G.; Ficzere, D.; Varga, P. Communication Challenges and Solutions between Heterogeneous Industrial IoT Systems. In Proceedings of the 2019 15th International Conference on Network and Service Management (CNSM), Halifax, NS, Canada, 21–25 October 2019. [CrossRef]

94. Kozma, D.; Varga, P.; Soos, G. Supporting Digital Production, Product Lifecycle and Supply Chain Management in Industry 4.0 by the Arrowhead Framework—A Survey. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019. [CrossRef]

95. Soos, G.; Kozma, D.; Janky, F.N.; Varga, P. IoT Device Lifecycle - A Generic Model and a Use Case for Cellular Mobile Networks. In Proceedings of the 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 6–8 August 2018. [CrossRef]

96. Soos, G.; Varga, P. On the Security Threat of Abandoned and Zombie Cellular IoT Devices. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; Volume 1, pp. 996–1001. [CrossRef]

97. Varga, P.; Blomstedt, F.; Ferreira, L.L.; Eliasson, J.; Johansson, M.; Delsing, J.; de Soria, I.M. Making system of systems interoperable—The core components of the arrowhead framework. *J. Netw. Comput. Appl.* **2017**, *81*, 85–95. [CrossRef]

98. Hegedus, C.; Kozma, D.; Soos, G.; Varga, P. Enhancements of the Arrowhead Framework to refine inter-cloud service interactions. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016. [CrossRef]

99.   Kozma, D.; Varga, P.; Larrinaga, F. Data-driven Workflow Management by utilising BPMN and CPN in IIoT Systems with the Arrowhead Framework. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019. [CrossRef]

100. Menon, K.; Kärkkäinen, H.; Wuest, T.; Gupta, J.P. Industrial internet platforms: A conceptual evaluation from a product lifecycle management perspective. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2018**, *233*, 1390–1401. [CrossRef]

101. Oztemel, E.; Gursev, S. Literature review of Industry 4.0 and related technologies. *J. Intell. Manuf.* **2020**, *31*, 127–182. [CrossRef]

102. Lu, J.; Törngren, M.; Chen, D.J.; Wang, J. A Tool Integration Language to Formalize Co-simulation Tool-Chains for Cyber-Physical System (CPS). In *Software Engineering and Formal Methods*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 391–405. [CrossRef]

103. Appel, A.W. Verified Software Toolchain. In *Programming Languages and Systems*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–17. [CrossRef]

104. Derhamy, H.; Andersson, M.; Eliasson, J.; Delsing, J. Workflow management for edge driven manufacturing systems. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018. [CrossRef]

105. Varga, P.; Kozma, D.; Hegedus, C. Data-Driven Workflow Execution in Service Oriented IoT Architectures. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018. [CrossRef]

106. Indrasiri, K. *Beginning WSO2 ESB*; Apress: New York, NY, USA, 2016. [CrossRef]

107. Hegedus, C.; Varga, P.; Frankó, A. Secure and trusted inter-cloud communications in the arrowhead framework. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018; pp. 755–760. [CrossRef]

108. Rocha, R.; Maia, C.; Ferreira, L.L.; Varga, P. Improving and modeling the performance of a Publish-Subscribe message broker. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; Volume 1, pp. 5493–5498. [CrossRef]

109. Albano, M.; Barbosa, P.M.; Silva, J.; Duarte, R.; Ferreira, L.L.; Delsing, J. Quality of service on the arrowhead framework. In Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, 31 May–2 June 2017; pp. 1–8. [CrossRef]

110. Kulcsár, G.; Koltai, K.; Tanyi, S.; Péceli, B.; Horváth, A.; Micskei, Z.; Varga, P. From Models to Management and Back: Towards a System-of-Systems Engineering Toolchain. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–6. [CrossRef]

111. Cuenca, J.; Larrinaga, F.; Arenaza-Nuño, I. A software engineering process to develop services within the Arrowhead project. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 5232–5237.

112. Illarramendi, M.R.; Larrinaga, F.; Perez, J.M.; Etxeberria, L.; Elkorobarrutia, X.; Sagardui, G. MDE based IoT Service to enhance the safety of controllers at runtime. In *STAF (Co-Located Events)*; Aachen University: Aachen, Germany, 2019; pp. 33–38.

113. Campos, J.; Sharma, P.; Albano, M.; Ferreira, L.L.; Larrañaga, M. An Open Source Framework Approach to Support Condition Monitoring and Maintenance. *Appl. Sci.* **2020**, *10*, 6360. [CrossRef]