Proceedings of the 59th Hawaii International Conference on System Sciences | 2026

Implementing Cooperative IoT Systems – A Product Development Method

Sven Kurrle
Ferdinand Steinbeis Institute
sven.kurrle@ferdinand-steinbeis-institut.de

Abstract

The Internet of Things (IoT) is increasingly capable of enabling cooperative products built by more than one company. Such cooperative IoT systems are developed by orchestrating existing products to provide novel value. This paper presents a novel development method for implementing such systems. The method was developed based on a 41-month longitudinal study in which teams from three large companies implemented a cooperative IoT system. The method identifies three phases that design, build, and evolve the IoT system. Three key factors observed in the case study are highlighted. This method provides a practical guide for research into IoT system development and practitioners seeking to implement such systems.

Keywords: Cooperative IoT Systems, Product Development Method, Internet of Things, Agile, Stage-Gate

1. Introduction

Internet of Things (IoT) systems are getting more mature every year. Initiatives like the digital twin consortia are driving industrial adoption forward while new standards like the *Internet of Things (IoT) — Reference architecture* standard (ISO/IEC 30141:2024; International Organization for Standardization [ISO], 2024) create clarity for industrial adoption (Object Management Group, 2024). This creates the foundation for the next wave of innovative IoT products that are built by the industry for the industry. Such IoT systems can be built by combining the existing capabilities of successful companies in cross-organizational collaboration (Weber et al., 2018).

This paper gives an overview of existing product development methods for products in general and IoT systems in the specific. Structuring development methods along a macro – micro axis reveals that one project can use a set of methods all at once. It might use a stage-gate process as a macro structure while using the V-model to structure software development and a scrum framework for workday structure. A literature review shows that existing product development methods are already being adapted to suit the development of IoT systems. But the existing methods fail to address the specific needs of cooperative IoT systems that are built across company borders. This research aims to fill this gap and develop a method for cooperative IoT system development that answers the question:

How can a product development method be designed to support cooperative IoT system development?

This paper summarizes the observations from a longitudinal case study that developed a cooperative IoT system. Researchers were able to actively participate in the development of a cooperative IoT system driven by three large companies. During a total of 55 workshops over a 41-month period the development progressed through several phases. Starting with a rough idea of the supposed value created to having the cooperative IoT system evolve with 15 vehicles connected to the live system. The research institute actively moderated the process and guided the ecosystem. Researchers were actively involved in all relevant activities and documented observations during the study.

The method presented in this paper originates from these observations and fills a gap in existing IoT system development. It can guide future implementations from a proof of concept to a realized pilot implementation.

2. Related Research

2.1 New Product Development Methods

Developing new products has long been a focus of engineering science. Methods in this field can be organized along a spectrum, ranging from macro-level approaches that examine the relationships between developmental phases, to micro-level techniques that structure day-to-day activities. The earlier methods like the waterfall model introduced by Royce (1987), the stage-gate model introduced by Cooper (1990), and the V-model introduced by Rook (1986) focus mainly on the macro structure of a development process.

The waterfall method proposed by Royce (1987) structures the implementation of large computer programs along a linear set of steps from eliciting

URI: https://hdl.handle.net/10125/112110
978-0-9981331-9-5

requirements through analysis, program design and coding, to testing and operations. Although this is often interpreted as being an inflexible step by step process, Royce himself stresses the need for iterative interactions between the various steps. He voices hope that such an iteration is confined to adjacent steps but acknowledges that in practice bigger jumps are the norm for design iterations. But in general steps follow each other. (Royce, 1987)

Cooper (1990) builds on this idea of successive steps by introducing gates in between each step. Cooper calls his steps the stages in which all development work is conducted. Gates act as a final check before work can progress to the next stage in the development process. Senior managers act as gatekeepers and apply predefined criteria to the development work of the previous stage to decide whether it is ready to proceed to the next stage. The work during the stage is structured as deliverables that are needed to pass the next gate. Although Cooper acknowledges that each business will require a tailored model, he provides a five stage “skeleton” model as a starting point. (Cooper, 1990)

The Stage-Gate model has recently seen new updates by combining it with agile methods to form an Agile-Stage-Gate Hybrid Model by Cooper et al. (2016). The model combines the macro stage-gate model with the more micro practices that form the agile framework. This combination promises a faster and more adaptive response to changes in customer needs. (Cooper et al., 2016)

The V-model introduced by Rook (1986) separates the development process into a set of phases that are like the ones found in the other two models. Each phase is completed by passing a review of a set stakeholder. The V-structure puts special emphasis on the code and unit test step that marks the lower turning point. From this point onwards units are successively further integrated into the overall system and changes here loop back to increasingly earlier stages of the design process. (Rook, 1986)

Agile methods are a loose collection of practices that originate from software development and are coalescing around the 12 principles published in the manifesto of agile software development (Agile Alliance, 2001). One of the core principles of agile development is to deliver valuable products early and regularly during the development process. The most used agile method is the Scrum framework (Guerrero-Ulloa et al., 2020).

These methods can provide structure to the product development process. While the waterfall, V-model and agile have an explicit software focus, the Stage-Gate model is more generally designed for product development. Methods like the Agile-Stage-Gate hybrid model try to bridge this gap and provide a unified method. What becomes clear is that the product to be developed has an influence on choosing the development method that best fits this process.

2.2 Cooperative IoT Systems

The development and operation of Internet of Things (IoT) systems increasingly rely on structured architectural approaches to ensure scalability and interoperability. Foundational efforts such as the Industrial Internet Reference Architecture (IIRA) (Lin et al., 2019) and the ISO/IEC 30141:2024 (ISO, 2024) standard provide comprehensive blueprints to guide the design and evaluation of IoT systems across domains. The IIRA defines essential viewpoints—business, usage, functional, and implementation—designed to align stakeholder concerns with technical solutions (Lin et al., 2019). Similarly, ISO/IEC 30141:2024 (ISO, 2024) establishes a reference architecture that formalizes essential IoT concepts, including device-level interaction with the physical world, trustworthiness, and viewpoint-based system descriptions.

Building on these principles, this study draws from a structured top-down modeling approach previously applied by Koch et al. (2010) in the German SME context and adapted to IoT ecosystems by Kurrle et al. (2022). In this approach, value creation serves as the starting point from which the information demand is defined. Which in turn is fulfilled through targeted information processing based on available data. This structured derivation from value to data ensures traceability and alignment with business goals. It was used and expanded for the developed method.

To operationalize this within IoT ecosystems, we integrate the above method with ecosystem initiation steps as proposed by Weber et al. (2018). This hybrid model enables us to define and realize IoT system phases aligning technical implementation with business-driven value creation. The approach ensures that data processing efforts are tightly coupled with specific information and stakeholder goals, echoing the business and functional viewpoints of ISO/IEC 30141: 2024 (ISO, 2024).

2.3 Why Cooperative IoT Systems are different

Cooperative IoT systems are built with existing components to orchestrate new value scenarios. They leverage the plethora of existing connected sensors, gateways and cloud components to create value faster than traditionally possible. Using existing supply chains by buying off-the-shelf products and adapting them to the specific use case means a lot of complexity can be avoided. Building a cooperative IoT system in this context moves the focus in product development away from mechanical design and coding software to orchestrating the component to provide value. It enables fast pivots and enables rapid experimentation, delivering features in as little as 15 minutes.

Combining this with a new mindset of rolling out new features fast but carefully as described by Kohavi et al. (2013) enables IoT systems to rapidly evolve. The increasing use of low-code environments in IoT system development (Ihirwe et al. 2020) makes it way more accessible to everyone in a development team. Now every member of an interdisciplinary team can understand the software code that provides value. This dramatically reduces misunderstandings and increases cohesion within the development team. This development favors smaller coherent teams with a clear focus over large, compartmentalized development units.

These properties of cooperative IoT systems mean that developing one as a product differs from developing a normal product in two important ways. First it focuses on creating new value for customers by orchestrating existing products. Secondly there is interdependence between cooperating companies. These mean that existing new product development methods might not be the perfect fit for developing cooperative IoT systems.

2.4 IoT System Product Development Methods

A structured literature review was conducted to identify existing research on product development methods that are applicable in IoT System development. Following the process laid out by Levy and Ellis (2006) literature was acquired from ProQuest and AISel databases. As a search term the review used: "IoT System" AND "Product Development" AND "Method". Of the 378 total peer-reviewed papers identified, 17 were selected as relevant by filtering for title and abstract. One paper was later added as it was referenced in a paper as an important review of IoT system development methods.

Table 1. Identified paper

	ProQuest	AISel
Total	116	262
Relevant	12(+1)	5

Most of the papers that were filtered out at this stage were focused on applying techniques to specific domains, for example healthcare. A set of papers that were filtered cover technical frameworks that are not related to the method of development. Some papers focused on platform-based business models. Literature that was classified as relevant will be covered below.

Guerrero-Ulloa et al. (2023) conducted an extensive literature review to identify proposed IoT system development methodologies. This review clusters identified methodologies into two categories. Methodologies in the first category were specifically designed for the development of IoT systems, while the ones in the second category were developed for traditional information systems and only later applied to IoT system development. Two notable examples in the first category are the INTER-METH methodology presented by Fortino et al. (2018) that is based on the waterfall methodology and the Test-Driven Development Methodology for IoT Systems presented by Guerrero-Ulloa et al. (2020) that is based on agile principles. The authors acknowledge that a plethora of other platforms, frameworks, and tools have been proposed in the literature they identified that cannot be classified as development methods but might still be relevant for someone aiming to develop an IoT system. The authors conclude that none of the IoT specific development methodologies have reached wider adoption, with none being used by others than their own authors. Most development efforts identified using Scrum as their only methodology while it is sometimes combined with other traditional methods like extreme programming, rapid prototyping, and Kanban.

Guerrero-Ulloa et al. (2023) mention Fortino et al. (2021) as another review of methodologies for IoT system development in their text. It was decided to include this as an additional paper in this review. Fortino et al. (2021) classifies existing development products into four classes: methodologies, frameworks, platforms, and tools. Those are than structured by the IoT system features they support. The paper identifies four key features: interoperability, smartness, scalability, and autonomy. It aims to provide an overview of currently employed tools and guide the engineering of further implementation. Nguyen-Duc et al. (2019) focuses on startups that develop IoT applications as Minimum Viable Products (MVP). The paper identifies three types of MVPs. A type 1 MVP has similar user experience to the final product, but all the actual functionality is carried out manually. In type 2 MVPs all the hardware functions are built out and operate as expected. Building a type 3 MVP entails a shift in focus from hard- to software. This classification is applicable to IoT startups that build their own hardware and software.

The other papers identified in the literature review have a diverging focus from the aim of this research. They are therefore only mentioned briefly. The following two papers focus on methods in product development but do not have an explicit focus on IoT. Vadan and Miclea (2023) developed a new method for software testing they call the Locate, Execute, Expect (LEE) design pattern. While this paper develops its method at the example of a smart home IoT system, it is not specific to IoT.

Yang et al. (2022) developed a new process of Design for Six Sigma (DFSS) that can assure design quality of a designed product.

The following five papers focus on IoT systems without going into product development methods. Varela et al. (2022) created a collaboration framework for industry 4.0 use cases. They stress the need for collaboration in such environments. Dou et al. (2021) introduces a concept that describes how social IoT can be mass customized and lists the key problems faced by such a system. Orłowski et al. (2021) build a continuous improvement model for IoT systems. They argue that such systems need the right environment to be able to mature in the field. Haße et al. (2022) identify eight design principles for digital twins and key requirements for sharing them. Digital twins are positioned as a hub through which data can be shared. Valderas et al. (2023) stresses the need for interdisciplinary teams in IoT development. They present an interdisciplinary development process that aims to support the creation of IoT-enhanced business processes.

The following two papers focus specifically on IoT security concerns. Orellana et al. (2024) built a guide for practitioners to navigate the trade-offs between security, performance energy consumption and processing capability. Balto et al. (2023) proposes testing IoT systems within a safe environment—they call Cyber Range—before deploying it to the real world. Specifically, they develop the concept of a Hybrid Cyber Range that combines emulated and real physical hardware to test the safety of IoT systems.

The following three papers are on a higher level of abstraction than the focus of this research. Tsaramirsis et al. (2022) describe how Industry 4.0 is a confluence of different technologies like IoT. They present a set of pillars that hold up Industry 4.0. Awouda et al. (2024) introduces an IoT-Digital Twin framework within the emerging context of sustainability, human-centricity, and resilience termed Industry 5.0. Unhelkar and Arntzen (2020) developed a framework that helps companies collaborate. They premise their research on the ability of modern communication technologies enabling new collaboration opportunities across company boundaries.

The following two papers aim to provide a general introduction into IoT. Dietz and Pernul (2019) introduce the reader to key characteristics of digital twins and argue that they can form the basis of a nested system of systems approach. Kumar et al. (2022) structure existing literature in IoT to provide a taxonomy that structures IoT architectures, applications, communication protocols, and key challenges.

The last paper covers an entirely different aspect of digital platform development. Karhu et al. (2020) present four tactics to implement a balanced digital platform strategy at the example of Apples IOS and Googles Android.

As shown by our literature review, the extant literature on methods for product development within the realm of IoT systems remains notably limited, particularly in the context of collaborative IoT system development. Only Guerrero-Ulloa et al. (2023), Fortino et al. (2021), and Nguyen-Duc et al. (2019) contribute to answering this paper’s research question directly. Others like Vadan and Miclea (2023) make contributions to IoT systems without mentioning product development methods. Others like Unhelkar and Arntzen (2020) and Varela et al. (2022) argue for an increase in cooperation across company borders. Valderas et al. (2022) stresses the need for interdisciplinary teams in developing IoT systems.

This research aims to fill this gap in literature by conducting an explorative case study to identify a possible method to collaboratively develop cooperative IoT systems.

3. Methods

This research employs a qualitative design science research approach to adapt existing product development methods to cooperative IoT systems following the framework proposed by Peffers et al. (2007). A holistic case study approach in accordance with Yin (2018) was used to ensure the artefact created is relevant in the fast-moving field of IoT. The use of qualitative case studies is a well-established approach in information systems literature to create artefacts that have a strong dependency on current practice (Benbasat et al., 1987).

3.1 Case description

The case was setup to develop a cooperative IoT system that provides transparency into intralogistics processes in a manufacturing plant. It was initiated in January 2020 and commenced in May 2023. A total of 55 workshops were conducted throughout this timeframe with further meetings added as needed. Researchers were guiding the project’s progress and created the space for participants to develop cooperative IoT systems.

The focus was put on handling large goods that are stored outside and therefore often not part of current warehouse management software. These goods are handled by 13 forklifts and 4 tow trucks in the given example for the pilot implementation. The main customer of the product was defined as the head of logistics for this part of the plant. The aim was to deliver relevant information to him, so he can improve his logistics processes.

Three companies joined forces to achieve this aim and built a scalable product based on this case. The manufacturing company hosted pilot implementation and provided logistics knowledge. An industrial service and cloud provider was part of the case bringing logistics knowledge as well as cloud infrastructure and development expertise to the project. A manufacturer of sensors and IoT gateways joined the case bringing knowledge on how to measure, process, and send real world signals.

A total of 14 participants can be considered the core of the project with an additional 13 participants that only supported the project. Participants were numbered roughly equal from all four participating organizations. The participants were separated into three groups with distinct focus areas which were structured in accordance with the viewpoints of the IIRA (Lin et al., 2019). The business group focused on the high-level guidance of the project and defining the value scenario. The domain group focused on the logistic background and designed tools to achieve this value scenario. The implementation group focused on building a pilot implementation and validating the theoretical design.

3.2 Data collection

The data analyzed in this paper was gathered throughout the case. Data was collected during workshops and augmented by the outcomes produced between them. The project's progress was documented in four ways: written protocols, images, slide decks, or work results. These in total 235 documents were clustered by workshop and group. Researchers actively participated in the projects development and were able to gain first-hand insights through participant observation that guided the creation of the artefact (Iacono et al., 2009). This form of cocreation of the artefact enhances the practical relevance of the developed artefact (Baskerville & Myers, 2004).

3.3 Data analysis

The collected protocols were first scanned for passages that are relevant to the question at hand. 503 segments of the protocols were deemed relevant in general. Those were clustered into processes and results by using the inductive category development method described by Mayring (2014). Additional key insights were flagged for consolidation into key factors to increase the methods usefulness.

4. Results

This research designed a method that guides the development of cooperative IoT systems. The method is structured along the logical progression of an IoT system interdependent elements. Those are boxed elements in Figure 1. Connecting them are the process steps that enable the progression up the stack towards the value being created at the very top. This structure was observed in the IoT systems built during the case study. Specific focus was placed on this in the workshops 6, 10, 12, and 46. The development of cooperative IoT systems during the case study can be separated into three main phases.

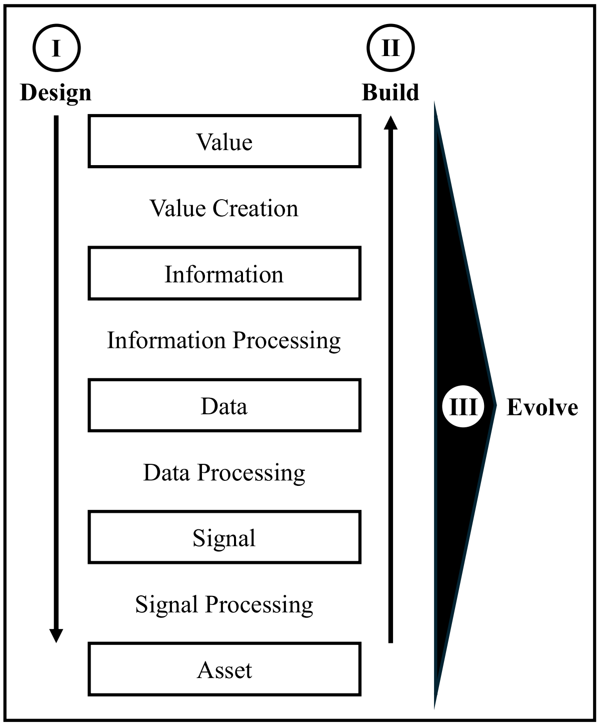

Figure 1. Method Phases

4.1 Designing a cooperative IoT system

The first phase is to design the entire system starting from the value it is supposed to provide down to the required assets. This is guided by a series of questions that the design team needs to answer consecutively from value to assets as shown in Figure 2.

While this design logic follows a clear progression from value through the intermediate steps of information, data, signal, to the asset, the design process does not necessarily flow linearly. The design questions are meant to convey a logical consistency that is required of the designed system. Every signal that is processed has a clear relationship through the hierarchy into the value it is creating.

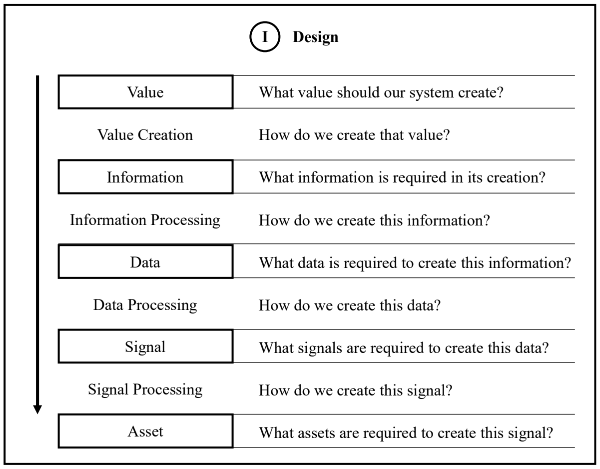

Figure 2. Design cooperative IoT systems

One example is the forklift tracking in the observed case. The logistics lead of one company wanted to gauge how many kilometers a forklift drives on any given day to benchmark the current stress level on the logistic process. The system designers would answer the questions as follows:

What value should our system create? It should help the logistics lead benchmark the current stress level on his process. **How do we create that value?** We create a dashboard with information on how many kilometers the aggregated forklift fleet covered in any given timeframe. **What information is required in its creation?** An aggregated kilometers count of all forklifts. **How do we create this information?** We need to sum up all individual forklift's kilometer counts. To get the individual kilometer counts we need to calculate the distance traveled between the current position and the previous position. **What data is required to create this information?** The current position of all forklifts in a reasonable interval to not generate to large an error by "cutting corners". **How do we create this data?** We send the current position of each forklift every two seconds to the cloud whenever the forklift is actively in use. **What signals are required to create this data?** We need a Global Navigation Satellite System (GNSS) signal with a quality indicator like the number of satellite fixes as well as a signal of when the forklift is in use. This "in use" signal will be derived from the GNSS based speed of the forklift. Is the speed faster than 1 km/h the forklift will be set to "in use" for the next 90 seconds. **How do we create this signal?** We process the longitude, latitude, and speed values from a GNSS receiver. **What assets are required to create this signal?** We require all forklifts to measure these signals.

After finalizing the functional design, the individual functions can be claimed by the participating companies. This process ensures that discussions on who does what are always based on a common understanding of the entire system architecture and can be resolved by the two interfacing parties.

4.2 Building a cooperative IoT system

A well designed, coherent system plan is the foundation for building an effective cooperative IoT system. The build phase starts on the asset layer and progresses up through the logic of the design towards the value. A practitioner must ask at each step of the way, whether the system that was built can deliver on the design requirements. The building process should focus on building all required components to deliver one valuable product first and then add additional ones. This makes fundamental problems occur early in the process and means changes will not result in a lot of lost effort.

Building a cooperative IoT system will inevitably lead to some deviations from the designed specification. They will require an adaptation in the design by people knowledgeable on the whole system. One core reason why making the interrelations within the system explicit and easily understood is an important part of a good design. Deviations might come in two ways:

First, the system built is incapable of delivering the designed performance and therefore not able to achieve a given functionality for the next layer up. An example of such a deviation could be observed for the GNSS speed trigger that was initially designed to trigger the forklifts "in use" state. It became clear after the implementation of this component, that the GNSS speed varied a lot more than previously planned. This resulted in false positive "in use" signals while the forklift was not in use. Designer and builder agreed to raise the threshold for "in use" to 2km/h which removed most of the false positive values. The issue got earmarked for further evolving down the line. The deviation from the plan lead to an informed ad-hoc decision to change the design reminiscent of an agile approach.

Second, the built system is showing inconsistent performance in real world conditions and can therefore only be relied upon in specific circumstances. One example of this could be observed when the GNSS position became less accurate, the closer the forklifts track comes to a production hall blocking its line of sight to GNSS satellites. This is especially pronounced when a forklift drives within a production hall. This led to a change of scope for the project, focusing on outdoor storage areas therefore changing the value the system can deliver.

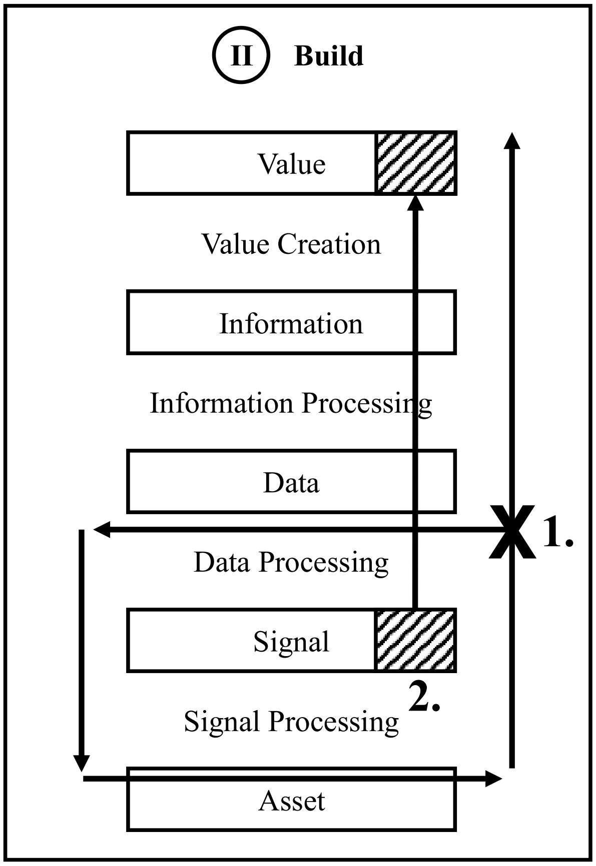

Figure 3. Building cooperative IoT systems

4.3 Evolving a cooperative IoT system

The aim of evolving the system is twofold. To improve the existing system, making it better at delivering the existing value, or to expand it to provide new value. Both require a different way of approaching the process of evolving the existing IoT system.

Incremental improvements should be made all the time if the system is lively. It is important that a single person takes responsibility for any part of the system. This person can change anything within the workings of his part without consulting others, if it does not change the interfaces with neighboring parts. One example of this was when the data processing on the forklift started to face performance issues. It was decided to switch an internal interface from REST API calls to a more lightweight MQTT interaction. This was implemented without having an impact on the other parts and was therefore decided upon and executed by the person responsible for this part.

If the output or input does change in a non-disruptive way, those impacted should be informed of the change. This makes sure they are aware of the change and can improve their own parts. One example is the forklifts position. During the first implementation it was just passed on as it was with a varying length dependent on the current measurement. That resulted in different length floating numbers being sent into information processing. The person responsible for data processing did some research and with knowledge of the value created cut the numbers at seven decimal places, corresponding to an accuracy of roughly 1cm. The person responsible for information processing was informed of this change and able to shrink the database while having no impact on the information displayed.

If the output or the input changes in a disruptive way, a cooperative redesign of this interface is necessary. All affected parties need to work on improving the system as a whole and should always keep the created value in focus when evolving the cooperative IoT system. One example of such a disruptive evolution is the change from getting the “in use” state of a forklift from the GNSS speed to getting it from the vehicle ignition voltage. This change in data processing requires a new signal acquisition on the asset as well as new signal processing. It does also change the upstream information and how it should be interpreted.

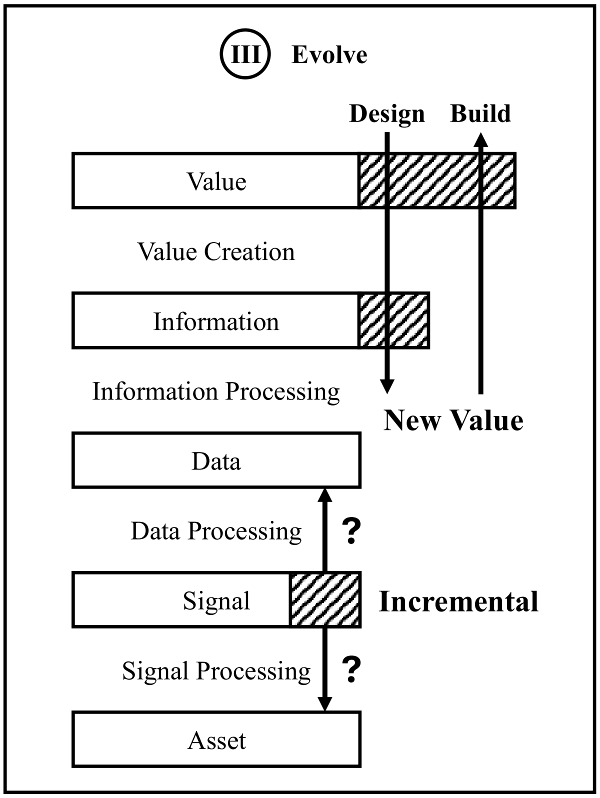

Figure 4. Evolving cooperative IoT systems

New value creation requires a different approach. Any new value creation idea needs to be verified with the person that will accrue this value. Only if the person agrees that this is a valuable change, should this initiative be further designed and built. The change will then start with its design phase, constructing a logical chain by questioning every part as described in 5.1 Designing a cooperative IoT system. The difference is that answers to the design questions are continuously compared with already existing parts of the system. It has often been observed that what seems like completely new value can be created by recombining existing information or enriching it with additional already present data. If so, the design process stops at this stage and moves to its building phase. One example of this could be observed when the logistics lead was presented with the first dashboard in workshop 18 that showed the kilometer traveled today of one test forklift. He realized, understanding the complete design architecture, that it would be trivial to also provide a map of where the forklift was driven today. The exact position of the forklift every 2 seconds was used to calculate the distance traveled and was therefore already in the set of available data.

4.4 Key Factors in implementing cooperative IoT systems

Simplicity is of the essence. All work should only be as complex as necessary to achieve the designed value. This makes sure that all participants in the project can easily understand the interrelated parts of the IoT system they are building. It creates opportunities for the different perspectives of business, domain, and implementation groups to constructively work together and build the system. Numerous observations in the case show the benefit of this breaking up of barriers between traditionally separate units. All participants in the project should try to break everything down to the most obvious pieces. The data required for the benchmarking of stress on the forklift can be put in one sentence understood by everyone: the current position of the forklift in latitude and longitude as well as the current state, either “in use” or “not in use”.

Process follows function as the functional IoT system is the main goal of this process. It is important to stress that only a system that provides value can be regarded as a functioning system. A system that creates nice dashboards but provides no value is not fulfilling its function and needs to be adapted. Whenever someone in the project feels like the value might be in jeopardy by a given change this needs to be discussed with the person that uses the system to create value.

Start small and evolve rapidly delivering valuable products early on and evolving them over time. This core principle of agile development is very applicable to implementing cooperative IoT systems. The logical structure built throughout the first two phases lends itself to being expanded and improved easily. This should encourage teams implementing cooperative IoT systems to start with the smallest value creating part and rapidly expand the systems capabilities. This also limits the business risks of cooperating, as the cooperation will only expand if it is mutually beneficial.

5. Discussion

The developed method for cooperative IoT system development extends the current field of product development methods by providing a clear focus on the logical architecture of IoT systems as described by the ISO/IEC 30141:2024 (ISO, 2024), Kurrle et al. (2022), and Koch et al. (2010). The method can be compared to the V-model in as far as it guides the development of a product along its structural design. But coding software products differs substantially from orchestrating cooperative IoT systems. The developed method therefore diverges from the V-model in its layers that are aligned with IoT system architectures as well as its phase structure that is enabled by simpler orchestration steps instead of coding. Future research should put the developed method into its broader methodical context with a macro structure like the three types of MVP identified by Nguyen-Duc et al. (2019). Or combined with the Agile-Stage-Gate Hybrid model by Cooper et al. (2016) to form a holistic methodical framework for cooperative IoT system development. The three identified key factors can act as a basis for more thorough investigation into the design principles that support cooperative IoT system creation. Those can help guide the implementation of future cooperative IoT systems. Beyond guiding IoT system implementation in general, this work opens opportunities to investigate how existing product development methods must evolve to ensure that a company’s products are inherently prepared for integration into cooperative IoT systems. Further research should also emphasize how to scale a system beyond its pilot implementation.

6. Conclusion

Existing product development methods are not meant for the fast-paced system evolutions that are possible when value is created by orchestrating existing products. This paper proposes a method to implement cooperative IoT systems in three phases based on real world observations of one longitudinal case study.

The first phase designs the cooperative IoT system from the created value down to the required assets. The second phase builds the cooperative IoT system from the assets upwards to create the designed value. All steps in the process of implementing the system are structured by the systems overall architecture providing guidance for inevitable iterative changes throughout the process. The third phase guides the evolution of the cooperative IoT System beyond its initial implementation to evolve and grow rapidly testing what really provides value for customers.

The paper creates a basis for further research into implementing cooperative IoT systems in the specific but can also help with IoT system implementation in general.

7. References

Agile Alliance. (2001). *Manifesto for Agile software development*. https://agilemanifesto.org/

Awouda, A., Traini, E., Bruno, G., & Chiabert, P. (2024). IoT-Based Framework for Digital Twins in the Industry 5.0 Era. *Sensors, 24*(2), 594. https://doi.org/10.3390/s24020594

Balto, K. E., Muhammad, M. Y., Shalaginov, A., & Katt, B. (2023). Hybrid IoT Cyber Range. *Sensors, 23*(6), 3071. https://doi.org/10.3390/s23063071

Baskerville, R., & Myers, M. D. (2004). Special issue on action research in information systems: Making IS research relevant to practice: Foreword. *MIS Quarterly, 28*(3), 329–335. https://doi.org/10.2307/25148645

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly, 11*(3), 369–386. https://doi.org/10.2307/248684

Cooper, R. G. (1990). Stage-gate systems: A new tool for managing new products. *Business Horizons, 33*(3), 44–54. https://doi.org/10.1016/0007-6813(90)90040-i

Cooper, R. G., & Sommer, A. F. (2016). The agile–stage-gate hybrid model: a promising new approach and a new research opportunity. *Journal of Product Innovation Management*, *33*(5), 513-526. https://doi.org/10.1111/jpim.12314

Dietz, M., & Pernul, G. (2020). Digital twin: Empowering enterprises towards a system-of-systems approach. *Business & Information Systems Engineering, 62*, 179–184. https://aisel.aisnet.org/bise/vol62/iss2/7

Dou, Z., Sun, Y., Wu, Z., Wang, T., Fan, S., & Zhang, Y. (2021). The Architecture of Mass Customization-Social Internet of Things System: Current Research Profile. *ISPRS International Journal of Geo-Information, 10*(10), 653. https://doi.org/10.3390/ijgi10100653

Fortino, G., Savaglio, C., Palau, C. E., de Puga, J. S., Ghanza, M., Paprzycki, M., Montesinos, M., Liotta, A., & Llop, M. (2018). Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. In G. Fortino, C. Savaglio, M. Palau, J. S. de Puga, M. Ghanza, M. Paprzycki, M. Montesinos, A. Liotta, & M. Llop (Eds.), *Integration, interconnection, and interoperability of IoT systems* (pp. 199–232). Springer. https://doi.org/10.1007/978-3-319-61300-0_8

Fortino, G., Savaglio, C., Spezzano, G., & Zhou, M. (2021). Internet of Things as system of systems: A review of methodologies, frameworks, platforms, and tools. IEEE *Transactions on Systems, Man, and Cybernetics: Systems, 51*(1), 223–236. https://doi.org/10.1109/TSMC.2020.3042898

Guerrero-Ulloa, G., Hornos, M. J., & Rodríguez-Domínguez, C. (2020). TDDM4IoTS: A test-driven development methodology for Internet of Things (IoT)-based systems. *Communications in Computer and Information Science, 1193*, 41–55. https://doi.org/10.1007/978-3-030-44703-8_4

Guerrero-Ulloa, G., Rodríguez-Domínguez, C., & Hornos, M. J. (2023). Agile Methodologies Applied to the Development of Internet of Things (IoT)-Based Systems: A Review. *Sensors, 23*(2), 790. https://doi.org/10.3390/s23020790

Haße, H., van der Valk, H., Möller, F., & Otto, B. (2022). Design principles for shared digital twins in distributed systems. *Business & Information Systems Engineering, 64*(6), 751–772. https://aisel.aisnet.org/bise/vol64/iss6/4 https://doi.org/10.3390/s23063071

Iacono, J., Brown, A., & Holtham, C. (2009). Research methods—A case example of participant observation. *The Electronic Journal of Business Research Methods, 7*(1), 39–46. https://academic-publishing.org/index.php/ejbrm/article/view/1207

Ihirwe, F., Di Ruscio, D., Mazzini, S., Pierini, P., & Pierantonio, A. (2020, October). Low-code engineering for Internet of Things: A state of research. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (pp. 1–8). ACM. https://doi.org/10.1145/3417990.3420208

International Organization for Standardization. (2024). *Information technology — Internet of Things (IoT) — Reference architecture* (ISO/IEC 30141:2024) (2nd ed.). https://www.iso.org/standard/88800.html

Karhu, K., Gustafsson, R., Eaton, B., Henfridsson, O., & Sørensen, C. (2020). Four tactics for implementing a balanced digital platform strategy. *MIS Quarterly Executive, 19*(2), 105–120. https://doi.org/10.17705/2msqe.00027

Koch, M., Lasi, H., & Kemper, H.-G. (2010). IT based production management in medium sized industrial businesses – an empirical study. In *Proceedings of the Mediterranean Conference on Information Systems (MCIS)*, Tel Aviv. https://aisel.aisnet.org/mcis2010/49

Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T., & Xu, Y. (2013). Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1168–1176). ACM. https://doi.org/10.1145/2487575.2488217

Kumar, A., Sharma, S., Singh, A., Alwadain, A., Bong-Jun Choi, Manual-Brenosa, J., Ortega-Mansilla, A., & Goyal, N. (2022). Revolutionary Strategies Analysis and Proposed System for Future Infrastructure in Internet of Things. *Sustainability, 14*(1), 71. https://doi.org/10.3390/su14010071

Kurrle, S., Hiller, S., Weber, P., & Lasi, H. (2022). Implementing a proof-of-concept in IoT-ecosystems: a case study in the hospitality industry. In *2022 Portland International Conference on Management of*

Engineering and Technology (PICMET) (pp. 1-8). IEEE. https://doi.org/10.23919/PICMET53225.2022.9882713

Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. *Informing Science, 9*. https://doi.org/10.28945/479

Lin, S.-W., Crawford, M., & Mellor, S. (Eds.). (2019). *The Industrial Internet of Things volume G1: Reference architecture* (Version 1.9). Industrial Internet Consortium. https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

Mayring, P. (2014). Qualitative content analysis: Theoretical foundation, basic procedures and software solution. Klagenfurt, Austria: Beltz.

Nguyen-Duc, A., Khan, K., Sohaib, S. B., & Lønnestad, T. (2019). Minimum Viable Products for Internet of Things Applications: Common Pitfalls and Practices. *Future Internet, 11*(2), 50. https://doi.org/10.3390/fi11020050

Object Management Group. (2024, January 18). *Industrial Internet Consortium integrates into Digital Twin Consortium to drive digital transformation through digital twins* [Press release]. https://www.omg.org/news/releases/pr2024/01-18-24.htm

Orellana, C., Cereceda-Balic, F., Solar, M., & Astudillo, H. (2024). Enabling Design of Secure IoT Systems with Trade-Off-Aware Architectural Tactics. *Sensors, 24*(22), 7314. https://doi.org/10.3390/s24227314

Orłowski, C., Cygert, D., & Nowak, P. (2021). Extended continuous improvement model for Internet of Things system design environments. *Journal of Information and Telecommunication, 5*(3), 279-295. https://doi.org/10.1080/24751839.2020.1847506

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rook, P. (1986). Controlling software projects. *Software Engineering Journal, 1*(1), 7–16. https://doi.org/10.1049/sej.1986.0003

Royce, W. W. (1987). Managing the development of large software systems: Concepts and techniques. In *Proceedings of the 9th International Conference on Software Engineering* (pp. 328–338). IEEE Computer Society. https://dl.acm.org/doi/10.5555/41765.41801

Tsaramirsis, G., Kantaros, A., Al-Darraji, I., Piromalis, D., Apostolopoulos, C., Pavlopoulou, A., Alrammal, M., Ismail, Z., Buhari, S. M., Stojmenovic, M., Tamimi, H., Randhawa, P., Patel, A., & Khan, F. Q. (2022). A Modern Approach towards an Industry 4.0 Model: From Driving Technologies to Management. *Journal of Sensors, 2022*(1). https://doi.org/10.1155/2022/5023011

Unhelkar, B., & Arntzen, A. A. (2020). A framework for intelligent collaborative enterprise systems: Concepts, opportunities and challenges. *Scandinavian Journal of Information Systems, 32*(2), Article 6. https://aisel.aisnet.org/sjis/vol32/iss2/6

Vadan, A.-M., & Miclea, L. (2023). Software Testing Techniques for Improving the Quality of Smart-Home IoT Systems. *Electronics, 12*(6), 1337. https://doi.org/10.3390/electronics12061337

Valderas, P., Torres, V., & Serral, E. (2023). Towards an interdisciplinary development of IoT-enhanced business processes. *Business & Information Systems Engineering, 65*(1), 25–48. https://aisel.aisnet.org/bise/vol65/iss1/3

Varela, L., Putnik, G., & Romero, F. (2022). The concept of collaborative engineering: a systematic literature review. *Production & Manufacturing Research, 10*(1), 784-839. https://doi.org/10.1080/21693277.2022.2133856

Weber, P., Morar, D., & Lasi, H. (2018). Transforming value chains into internet-based ecosystems: A testbed approach. In *Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET)*, Honolulu. https://doi.org/10.23919/PICMET.2018.8481829

Yang, C. C., Jou, Y. T., Lin, M. C., Silitonga, R. M., & Sukwadi, R. (2022). The development of the new process of design for six sigma (DFSS) and its application. *Sustainability, 14*(15), 9294. https://doi.org/10.3390/su14159294

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications. https://us.sagepub.com/en-us/nam/case-study-research-and-applications/book250150