



# Criteria-based comparison of Systems Engineering and Agile Methods

A fundamental comparison of two development methods

Lynn, L., Humpert\*  
Fraunhofer Institute for mechatronic  
system design, Paderborn, Germany

Enrik, E., Mundt  
Fraunhofer Institute for mechatronic  
system design, Paderborn, Germany

Lukas, L., Bretz  
GTZ/VPD Systems Engineering Miele  
& Cie. KG

Harald, H., Anacker  
Fraunhofer Institute for mechatronic  
system design, Paderborn, Germany

Vanessa, V., Bernz  
Fraunhofer Institute for mechatronic  
system design, Paderborn, Germany

Roman, R., Dumitrescu  
Fraunhofer Institute for mechatronic  
system design, Paderborn, Germany

## ABSTRACT

This paper compares two popular design methods used in product development, namely Systems Engineering (SE) and agile methods. The comparison is based on various criteria, such as ease of implementation, object of innovation, software-dependency, customer relations, human centricity, organizational structure, values, and innovative capability, which were developed in a previous paper. The content of the comparison is supported by a literature review and a workshop with industry professionals. The findings show that both SE and agile methods have similarities and differences based on the criteria examined, and the choice of methodology depends on the project's specific context and objectives. While this paper provides only a general overview of each criterion and their compatibility, the results can assist product development teams in selecting the most appropriate approach for their needs. However, more research is necessary to enhance the effectiveness and efficiency of both approaches. This paper is a useful resource for practitioners and researchers interested in understanding the similarities and differences between SE and Agile and making informed decisions regarding their use.

## CCS CONCEPTS

• Software and its engineering; • Software creation and management; • Designing software; • Software design engineering;

## KEYWORDS

agile Systems Engineering, criteria-based comparison, design method, cooperative work

## ACM Reference Format:

Lynn, L., Humpert, Enrik, E., Mundt, Lukas, L., Bretz, Harald, H., Anacker, Vanessa, V., Bernz, and Roman, R., Dumitrescu. 2023. Criteria-based comparison of Systems Engineering and Agile Methods: A fundamental comparison of two development methods. In *2023 8th International Conference on Information Systems Engineering (ICISE 2023)*, December

\*Corresponding author



This work is licensed under a Creative Commons Attribution International 4.0 License.

ICISE 2023, December 16–18, 2023, Bangkok, Thailand

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0917-3/23/12

<https://doi.org/10.1145/3641032.3641038>

16–18, 2023, Bangkok, Thailand. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3641032.3641038>

## 1 INTRODUCTION

The demand for efficient and effective product development approaches that can handle increasing complexity and changing market requirements has led to a growing interest in cooperative work through both agile methods and Systems Engineering [1], [2], [3]. In a previous paper, we introduced the two approaches and derived a set of criteria upon which they can be compared on a fundamental level [4]. In this paper, we aim to use these criteria to compare and contrast agile methods (Agile) and Systems Engineering (SE) on a high level. To carry out this comparison, a literature review and a workshop with practicing experts is conducted. The literature review focuses on recent research articles and case studies that discuss the use of Systems Engineering and Agile product development in complex product development. The workshop's objective is to shed light on criteria that are not yet adequately covered by academic discourse. Participants consist of Agile and/or SE experts from various industries and academia, who share their experiences and insights into the practical application of these approaches. The comparison is based on the criteria identified in the previous paper, which include Complexity, Project size, Program Application, Risk Management, Team organization /interdependency, and Interdisciplinarity [4]. The results of the comparison will provide a high-level overview of the strengths and weaknesses of each approach and their compatibility for combined use. Before the comparison in Chapter 3, the methodological procedure is described in Chapter 2. Chapter 4 presents the results of the comparison.

## 2 PROCEDURE OF THE COMPARISON AND EXPLANATION OF THE CRITERIA

We follow the Design Research Methodology DRM [5]. The research methodology DRM consists of four stages: Research Clarification, Descriptive Study (DS) I, Prescriptive Study (PS), and DS II. In a first paper [4] the objectives and the problem (Research Clarification and DS 1) are worked out. The results are criteria for comparing agile methods with Systems Engineering. Subsequently, as shown in Figure 1, this paper builds the foundation for the solution (part of DS I) to extend the understanding about the two approaches. In the following, criteria of [4] are used to make a comparison between the two approaches. A structured literature review is conducted

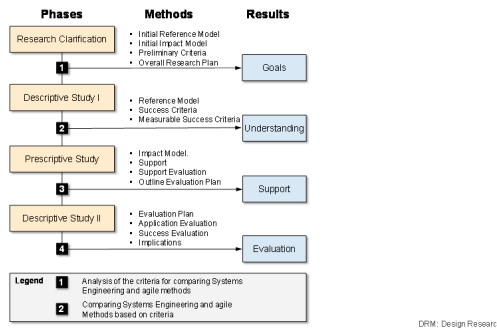


**Figure 1: classification of the paper in the phase-milestone Diagram of the DRM**

and the results are enriched with the help of an expert workshop. The qualitative workshop is conducted with 35 participants at a large conference called Tag des Systems Engineering (TdSE). The level of knowledge is stated in advance and showed that 90 percent of the participants had several years of experience in the field of Systems Engineering and with agile methods. The workshop prioritized criteria for which literature research findings did not yield a sufficiently comprehensive comparison. The following 18 criteria have been described based on the literature only (see section 3.1.). The remaining 13 criteria are explored in the workshop, because the literature did not yield enough information (see section 3.2.). A detailed explanation of the criteria and their derivation can also be found in the paper [4]. It should also be noted that the comparison is not meant to be comprehensive. The comparison is only intended to provide an overview.

### 3 COMPARISON: AGILE AND SYSTEMS ENGINEERING

First, comparisons based solely on available literature are listed. Then, comparisons that have been further developed through discussion and a survey in an expert workshop are presented. The comparison follows a structured format where systems engineering is described in relation to each criterion before agile methods are discussed. Finally, a brief summary is provided.

#### 3.1 Comparisons based on literature review

**Project size:** For a project to be successful, project size is also critical. According to the INCOSE Systems Engineering Handbook, there are hardly any statements about a fixed project size [6]. However, a study shows that 12 roles are relevant for systems engineering and project collaboration with the system engineer as the central role for a team [2] [7]. In contrast, in agile methods five roles have to be filled by a team of five to eight people to function well [8]. The task duration is fitted to suit the deadlines. While the project size is clearly specified in agile methods, it is not in SE.

**Program Application:** According to [9] Agile is generally used for internal or exploratory development, while SE is used for contracted development. Given the strong background of SE in the

defense and aerospace industries, contracted development is certainly a significant use case with detailed technical requirements available for systematic development. However, SE is also used for internal development with a strong focus on stakeholder engagement and understanding. This also provides a link to Agile, where requirements must be developed in partnership with stakeholders in the product development process. While there is significant overlap between Agile and SE in terms of program application, historic use of the approaches favors the use of Agile within exploratory development with unknown requirements or even goals, and SE within contractual and incremental product development where requirements, or at least the desired outcome, are known.

**Risk Management:** The key managed risks of Agile are traditionally user satisfaction and deliverable software, while for SE it's contract breach and safety of the system [9]. In order to manage these risks, there are defined risk management processes as defined by the ISO 15288 standard [10]. In Agile there are no defined risk management processes. However, risks are addressed through early and continuous prototyping and testing with relevant stakeholders. While there appear to be stark differences, the different approaches to risk management are not mutually exclusive and can benefit from a combination.

**Degree of Standardization:** Both Agile and SE processes are documented in standard bodies of work. The Agile methodology is documented in the Agile Manifest [11] and various methods such as Scrum and SaFE have procedures that are taught as standardized frameworks [12], [13]. A standard SE framework is laid out in the ISO15288 and further described in handbooks by INCOSE, NASA, DAU, and MITRE [9], [14], [15], [16]. Agile itself is only based on a set of values and guidelines that can be utilized in many forms, but methods like Scrum are highly standardized. SE on the other hand, has standardized framework and methods, which are used to varying degrees in industry. While Agile methods like Scrum have to be used in their standardized entirety, aspects of SE, such as requirements engineering practices, can be used separately.

**Team Organization / Interdependency:** The team organization depends on the task in SE. The systems engineer acts as a leader [6]. On the other hand, the team is clearly defined in agile methods. The team organization includes the development team, the role owners, the (Scrum) master, and, for leadership responsibility, management as roles [4]. In summary, the focus of SE is different, while agile methods clearly organize a team.

**Interdisciplinarity:** As the definition of SE explains [17], this is an interdisciplinary approach. The goal of SE is to integrate the interdisciplinary development processes and their results with a holistic system model. Interdisciplinarity is also addressed in agile methods. Scrum is one example. Scrum teams are interdisciplinary, meaning that members have all the skills needed to create value in each sprint. [12] Interdisciplinarity is an important aspect of both approaches and is continuously taken into account.

**Roles & Responsibilities:** In both SE and agile methods, roles and their responsibilities are immeasurable. However, they are fundamentally different from each other. In SE, diverse experts from all relevant disciplines are represented. These are coordinated by the systems engineer [6]. According to Sheard, the roles are technical manager, product manager, requirements manager, system developer, etc. [7]. In contrast, agile methods distinguish

between management and the development team [4]. For example, the following roles are also of fixed component: Product Owner(s), Coach(es) [4]. These roles have been standardized for most agile methods. In contrast to SE, there is, depending on the respective role, a corresponding responsibility to move the project forward [10].

**Origin:** SE originated as an approach to handle the development of increasingly complex mechatronic systems during World War II and the subsequent advances in communications and aerospace technology [18]. Agile, on the other hand, originated much later in 2001, with the aim of improving software development in an increasingly digital and rapidly changing world [10]. While SE has its origins in the development of complex physical systems and Agile in software development, both approaches are increasingly being applied to all domains.

**Abstraction Level:** SE as an approach is first defined at a high level of abstraction as an interdisciplinary approach. Individual approaches such as Model-Based Systems Engineering make this more concrete. [6] Agile methods are defined more specifically; auxiliary tools, such as the use of a Product-Backlog in SCRUM, are specified [12], [8]. SE itself is on a high level of abstraction and is concretized by certain methods, while agile methods such as Scrum provide clearer specifications.

**Type of Approach:** SE provides a framework for product development on a technical and project management level but can be described more as its own meta-discipline, as it supports the flow of information between other disciplines such as software, electronics, and mechanical engineering [19]. Agile at its core is a set of rules and guides than can be utilized any many different ways [11]. In practice, however, Agile is generally used to manage projects and the collaboration of the people involved [20]. While both approaches can guide the development of systems, Agile takes a more high-level project coordination role, while SE gives more comprehensive detail on the technical steps involved in developing successful systems [6].

**Sphere of Influence:** Both SE and Agile can and often are applied to individual development projects, meaning that each approach can be the leading development approach for specific projects or products, while the overhead organization is organized independently. However, within a project the horizontal communication and collaboration must be uniform for the chosen approach due the interdisciplinary nature of both SE and agile methods [6]. While SE or agile methods do not have to be the leading approach in other parts of the organization (especially overhead), it would be highly beneficial due to the interdisciplinary and transparent nature of both approaches.

**Communication:** Systems engineers oversee the communication between the team and the organization [6]. But the need to communicate what is being done is among project team members, across organizations, or over time to support future activities [6], [2]. A means of communication is the system architecture/model, which can be a single source of truth [2], [21]. In agile methods, the whole team communicates with each other [11], [8], [12]. There are regular meetings, such as the Daily Meeting, in which the team shares updates on the progress of the agreed tasks, challenges, and planned work [8], [12]. A means of communication is the Sprintboard, which represents the progress within a sprint

[8]. Communication is defined in both approaches with different methods.

**Variability / Environment:** SE is considered a structured approach to developing highly complex systems [6], while agile methods solve problems that are unpredictable [2]. Changing the system architecture in SE is expensive in a constantly changing environment and requires a greater degree of collaboration and up-front analysis. Agile methods, on the other hand, are specifically designed to perform well in highly variable environments [11].

**Traceability:** SE is a strongly traceability-driven approach [6], [22]. In contrast, not much is known about agile methods regarding traceability since they also prioritize experimentation before documentation and have a constantly changing backlog [11]. While SE places a high value on traceability, this criterion is rarely pronounced in agile.

**Lifecycle Coverage:** The focus of SE is on the entire V-model and entire lifecycle [6]. The V-model [23] is a development process that is often associated with SE [24], [22]. Agile methods also cover the entire product development phase. However, there is no structure in strict lifecycle phases, but only Sprints [12], [8].

**Tools:** While SE can in principle be done without any IT-tools, the usage of such tools is standard today. Either for singular a processes like requirements engineering (see Requirements Management) or for MBSE, which – depending on the definition – is only possible with digital models [25]. Thus, the tooling should be considered when planning the introduction of SE [26]. For Agile approaches, tools are widely used [27] but are not necessary, especially for small teams. SE-Tools typically focus on the authoring and management of content while Agile tools typically focus the project or process management.

**Methods:** For SE as well as agile approaches, a multitude of methods exist [28], [4]. In both cases many methods are typically bundled, e. g. within an approach like CONSENS [29], Harmony SE [30] or Scrum [12]. For SE a majority of experts agree that processes, methods, and tools need to be considered in synergy for successful SE implementation [31]. Like for tools, agile methods often focus on management of the project or process while SE often focuses on the creation of content (Models, etc.).

**Requirements Management:** Requirements management is a central aspect of SE, in which standardized methods and procedures are in place to manage the process from business or stakeholder needs to the lowest level system requirement [10]. Due to the agile value of “working software over comprehensive documentation”, the focus of requirements management is less on comprehensive requirements elicitation and documentation, but rather on user interaction and validation through testing [11]. The concept of developing systems based on user needs is the same in both approaches. However, agile methods tend to adopt user stories and backlogs as tools for requirements management, while SE translates user needs into comprehensive and detailed system requirements with standardized change management and system architecting processes [9].

**Reaction to Change:** In SE, changes are handled through a defined change management process, which is often accompanied by supporting tools [11]. While time-consuming, this process is thorough and analyses all associated changes through traceability. Agile methods use regular and frequent sprints, reviews, and

backlog refinements to notice and incorporate changes [13]. Both approaches have dedicated methods and tools to react to change, yet they differ significantly in their approach.

### 3.2 Comparisons based on literature and workshop insights

The criteria are listed according to the prioritization of the workshop participants. The criteria listed first were considered to be most important.

**Ease of Learning/Implementation:** The majority of workshop participants expressed the opinion that SE is difficult to implement, while only a few perceived it as an approach that is easy to comprehend. Implementation of SE was considered to be very time-consuming. In contrast, the perception of agile methods was more diverse, with some participants finding it to be only moderately time-consuming to implement, while others considered it to be extremely time-consuming. Regarding the learning process, Agile was perceived to be easier to understand. Overall, due to SE's technical depth and number of available standards, it is difficult to learn, while Agile is generally a much easier concept to learn. When it comes to implementation, however, both approaches face equally demanding challenges with a change in organizational design, procedures, and culture [31].

**Object of Innovation:** The criterion of innovation object was developed to determine the type of problems, products, or services to be solved in SE and Agile. There were various responses including system optimization vs. product optimization, stakeholder requirements, societal issues, length of the project execution process, uncertain and networked products. However, the most frequently mentioned response was complex problems. In contrast, the agile approach is mainly employed to address changing requirements or uncertain problems. Additionally, user interface, risky problems, short project execution process, uncertain or chaotic environments, and focus on the main functions were also mentioned as areas where agile approaches are applied.

**Verification:** Various approaches are also used for verification in both Systems Engineering and Agile. In Systems Engineering, for example, the test (most common response), demonstration, and review are used, as well as simulation, statistical analysis, a sample, a procedure, or HIL tests. In Agile, the demonstration, review, and tests are most frequently used, followed by the definition of Done, retrospective, simulation, statistical analysis, and prototyping.

**Software-Dependency:** Regarding the dependency of the approaches on software, tools, and the necessity of implementing the approaches in software applications, the participants agreed. Software is important for both approaches to support the process and methods of Agile and SE. Model-Based Systems Engineering thrives on continuous software support, while the origins of SE also managed with a document-centric approach. Agile can also be performed without software. Some approaches explicitly relied on a paper-based use of the methods [8]. The conclusion is that both approaches work independently of software to a certain degree. With a view to new forms of work, such as new work and end-to-end digitization, digital support can be useful.

**Customer relations and human centrality:** The degree of customer interaction and involvement, as well as the focus on stakeholder needs is a central element in both approaches [11], [6], [22]. In SE, various methods for stakeholder integration were mentioned by the experts, such as requirements management, lessons learned, workshops, stakeholder analysis, and specification exchange. For agile methods, product ownership, release days, stakeholder management, feedback, whiteboard, prototyping, and sprints were mentioned. Therefore, there are differences in the methods as named, but one should be careful whether these methods stand for themselves or are really part of the approaches under investigation. However, most of the participants answered that the user and other stakeholders are not included in the product development or are not included systematically. There is an increased need for research on how to involve the user, customer, and other stakeholders in system design, as also noted in the following paper [32].

**Organizational Structure:** Most of the respondents considered clear responsibilities and roles, followed by structuring the organizational structure after the system architecture to be the most important characteristics of an organizational structure ideal for SE. For agile the respondents favored a flat hierarchy for an ideal agile organizational structure. While there are some differences in the ideal organizational structure for both Agile and SE, there was an overwhelming agreement on the characteristics of interdisciplinary teams and flat hierarchies.

**Values:** SE is heavily characterized by transdisciplinarity, holism, stakeholder-centricity, and top-down and up-front development [33]. These and other values and principles have been derived both from theory and practice [34]. The values of Agile have been defined in the Agile Manifesto, including a focus on individuals and interactions, working software, customer collaboration, and response to change [11]. These values have been validated by the practical experience of workshop participants. While SE has a greater focus on structure, documentation, and traceability, both SE and Agile value interdisciplinarity, transparency, and collaboration. These commonalities enable a blending of the two approaches with respect to teamwork within a development project.

**Innovative Capability:** Agile is considered to be a very innovative development approach in which the solutions have the potential to be disruptive, with 84% of participants indicating this result. 68% of participants also considered this to be true for SE, however, there is a greater tail of opinions towards SE being used for more incremental development. Since both SE and agile approaches have been used to develop extremely innovative products, we cannot make a universally applicable statement of either approach being more innovative than the other. However, past and current industry practice skews rapid innovative development projects towards Agile, over SE being heavily used for large incremental development projects.

**Objectives:** Reasons for the usage of SE are manifold. Most mentioned are the goals to satisfy customer needs, manage complexity in large projects, and ensure transparency within the process. Means to these goals like interdisciplinary and holism were also often mentioned. However, we state that these are not goals in and of itself but means to reach other goals. A structured approach to SE goals can be found in [35]. For agile methods the most mentioned goals are fast reaction times, flexibility, time to market, focus,

**Table 1: Overview of each criterion’s compatibility between SE and Agile**

| Meta-criteria      | Criteria                                | Evaluation |
|--------------------|-----------------------------------------|------------|
| <b>Application</b> | Complexity                              |            |
|                    | Goals                                   |            |
|                    | Object of Innovation                    |            |
|                    | Innovative Capability                   |            |
|                    | Project Size                            |            |
|                    | Criticality                             |            |
| <b>Management</b>  | Program Application                     |            |
|                    | Ease of Learning/ Implementation        |            |
|                    | Risk Management                         |            |
|                    | Degree of Standardization               |            |
| <b>Personnel</b>   | Organizational Structure                |            |
|                    | Team Organisation/ independency         |            |
|                    | Interdisciplinarity                     |            |
|                    | Roles and Responsibilities              |            |
| <b>Principles</b>  | Origin                                  |            |
|                    | Values                                  |            |
|                    | Abstraction Level                       |            |
|                    | Type of Approach                        |            |
|                    | Extent of Documentation                 |            |
|                    | Communication                           |            |
|                    | Sphere of Influence                     |            |
|                    | Variability/ Environment                |            |
|                    | Customer relations and human cencricity |            |
|                    | Traceability                            |            |
| <b>Technical</b>   | Software-Dependency                     |            |
|                    | Lifecycle coverage                      |            |
|                    | Tools                                   |            |
|                    | Methods                                 |            |
|                    | Requirements management                 |            |
|                    | Reaction to change                      |            |
|                    | Verification (Testing)                  |            |
|                    | Legend                                  |            |
|                    | Compatible                              |            |
|                    | Partly compatible                       |            |
|                    | Not compatible                          |            |

and the ability to fail without big consequences (fail fast, feedback learning, etc.). Additionally, agile methods are a useful approach when technological and requirements uncertainty in a project are high [4]. The objectives suggest that SE leans towards large projects with high need for transparency and lower levels of uncertainty, whereas agile methods are suited to highly uncertain situations with high need for flexibility, fast changes, and expected fails – where negative impact should be minimized.

**Criticality:** Both approaches seem to not explicitly prefer or neglect criticality of the products under development. However, the product relevant norms and standards tend to processes as soon as a high criticality is given (e.g. medical products, autonomous vehicles, etc.). Our results show that within the borders of mandatory SE processes, Agile can strive. Often agile teams provide improved solutions on a short-term basis. However, even in these cases the

release and approval processes are not agile, so that the frequency of deployed updates is dictated by these processes.

**Complexity:** We asked the participants to choose one of the following four answers to characterize the complexity of agile methods: Simple (easy and standardized), Complicated (extensive, but standardized and not dynamically changing), Dynamic (Simple, but dynamical changes based on context), or Complex (extensive and dynamic). While some assessments overlap most participants see SE as a complex and Agility as a dynamic approach.

**Communication:** Communication in SE is often supported by System Models (MBSE or similar tools). Besides this, no SE-specific ways of communication were mentioned, so the flow of information is ensured by typical communication media. For Agile, tools like Jira and agile communication rituals like dailies, refinements, etc. were most mentioned. For both approaches a high emphasis was

put on the value of direct communication between humans and all mentioned means of communication complement each other well.

## 4 FINDINGS OF THE COMPARISON

The result of the comparison of agile methods and SE can be evaluated in a Table 1. A traffic light system (black, gray, white) is used to show an assessment of the compatibility. The comparison is used to check whether the two approaches are fundamentally compatible. The evaluation is based on an initial assessment. Overall, agile methods emphasize adaptability and quick course correction, while SE provides a structured approach to development. The criteria are classified according to the meta-criteria in [4].

## 5 OUTLOOK

The evaluation of the criteria will be validated in a workshop with experts in the field and to ask whether all synergies of the literature have been considered. The next step is to develop a method for the optimal combination of the two approaches (Prescriptive Study I) for defined use cases. Furthermore, the previous agile SE approaches can be analyzed and compared with the findings.

## 6 CONCLUSION

In summary, SE and Agile are two different approaches to product development, each with their own strengths and weaknesses. SE is a structured approach that emphasizes documentation, traceability, and top-down and up-front development. In contrast, Agile is an iterative and flexible approach that values collaboration, transparency, and quick adaptation to changing requirements. Both approaches are used to solve complex problems and require a certain level of software support. The criteria used for comparison were based on industry practice, literature review, and workshop insights, and covered topics such as ease of learning/implementation, verification, customer relations and human centricity, organizational structure, values, innovative capability, and objectives. While there are some differences in these criteria, such as the focus on complex problems in SE and changing requirements in Agile, both approaches value interdisciplinarity, transparency, and collaboration. Therefore, depending on the project objectives and specific requirements, either SE or Agile or a combination of both can be a suitable choice for product development and early validation purposes.

## ACKNOWLEDGMENTS

This research and development project is/was funded by the German Federal Ministry of Education and Research (BMBF) within the MoSyS project (02J19B090) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

## REFERENCES

- [1] S. Walker, "Erfolgsfaktoren für eine Produktentwicklung im 2021," *Aktuelle Technik*, 2021.
- [2] R. Dumitrescu, A. Albers, O. Riedel, R. Stark, J. Gausemeier, "Advanced Systems Engineering Wertschöpfung im Wandel", Paderborn, 2021.
- [3] D. Wilke, A. Schierbaum, L. Kaiser, and R. Dumitrescu, "Need for action for a company-wide introduction of systems engineering in machinery and plant engineering," *Proc. Des. Soc.*, vol. 1, 2021, pp. 2227–2236.
- [4] L. Humpert, E. Mundt, L. Bretz, H. Anacker, R. Dumitrescu, "Analysis of the criteria for comparing Systems Engineering and agile methods," *IEEE International Symposium on Systems Engineering*, October 2022.
- [5] L. T. M. Blessing and A. Chakrabarti, "DRM, a Design Research Methodology," 1st ed. Springer London, 2009.
- [6] INCOSE, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. New York: John Wiley & Sons, 2015.
- [7] S. A. Sheard, "Twelve Systems Engineering Roles," *INCOSE International Symposium*, vol. 3. San Diego, 1996, pp. 478-485.
- [8] A. Schröder, *Agile Produktentwicklung: schneller zur Innovation – erfolgreicher am Markt*, 2nd ed. Hanser Verlag, 2018.
- [9] R. S. Carson, "Can Systems Engineering be Agile? Development Lifecycles for Systems, Hardware, and Software," *INCOSE International Symposium*, vol. 23, 2013, pp. 16-28.
- [10] ISO, IEC, IEEE, ISO/IEC 15288 Systems and software engineering – System life cycle Processes, 2015.
- [11] K. Beck et al., "Twelve Principles of Agile Software," *Agile Manifesto*, 2001. URL: <https://agilemanifesto.org/principles.html> (accessed 13.04.2023).
- [12] K. Schwaber and J. Sutherland, *The 2020 Scrum Guide*, Scrum Guide, 2020. URL: <https://scrumguides.org/scrum-guide.html> (accessed 13.04.2023).
- [13] Scaled Agile Inc., "SAFe 6.0," *Scaled Agile Framework*, 2023.
- [14] NASA, *NASA Systems Engineering Handbook*. Washington, D.C., 2007.
- [15] DAU, *DAU Systems Engineering Handbook*. Fort Belvoir, 2023.
- [16] MITRE, *Systems Engineering Guide*. Bedford, 2014.
- [17] INCOSE, "What is Systems Engineering", *Systems Engineering Transforming Needs to Solutions*. URL: <https://www.incose.org/systems-engineering> (accessed 14.04.2023).
- [18] N. U. I. Hossain, R. M. Jaradat, M. A. Hamilton, C. B. Keating, S. R. Goerger, "A Historical Perspective on Development of Systems Engineering Discipline: A Review and Analysis," *Journal of Systems Science and Systems Engineering*, Springer, 2020, pp. 1-35.
- [19] T. Huth, T. Vietor, "Systems Engineering in der Produktentwicklung: Verständnis, Theorie und Praxis aus ingenieurwissenschaftlicher Sicht," *Gruppe. Interaktion. Organisation. Zeitschrift für Angewandte Organisationspsychologie (GIO)*, Vol 51, 2020, pp. 125-130.
- [20] M. Špundak, "Mixed Agile/Traditional Project Management Methodology – Reality or Illusion?," *Procedia – Social and Behavioral Sciences*, vol. 119, 2014, pp. 939-948.
- [21] A. Albers, R. Dumitrescu, J. Gausemeier, K. Lindow, O. Riedel, R. Stark, "Strategie Advanced Systems Engineering – Leitinitiative zur Zukunft des Engineering und Innovationsstandorts Deutschland," München, 2022.
- [22] L. Humpert, B. Röhm, A. Anacker, R. Dumitrescu, R. Anderl, "Method for direct end customer integration into the agile product Development," *CIRP Design*, 2022.
- [23] VDI/VDE 2206:2021-11 *Entwicklung mechatronischer und cyber-physischer Systeme*, 1995.
- [24] J. Gausemeier, R. Dumitrescu, D. Steffen, A. Czaja, O. Wiederkehr, C. Tschirner, "Systems Engineering in der industriellen Praxis," 2013.
- [25] A. M. Madini, M. Sievers, "Model-based Systems Engineering: Motivation, current status, and research opportunities," *INCOSE: Systems Engineering*, vol. 21, 2018.
- [26] J. Heihoff-Schwede, L. Bretz, L. Kaiser, R. Dumitrescu, "An Explanatory Model as Basis for the Introduction of Systems Engineering and capable IT-Infrastructure in Industry," *IEEE International Symposium on Systems Engineering*, Scotland, 2019.
- [27] D. Özkan, A. Mishra, "Agile Project Management Tools: A Brief Comparative View," *Cybernetics and Information Technologies*, vol. 19, 2019, pp. 17-25.
- [28] A. M. Schierbaum, "Systematik zur Ableitung bedarfsgerechter Systems Engineering Leitfäden im Maschinenbau," *Verlagsschriftenreihe des Heinz Nixdorf Instituts*, vol. 391, 2019.
- [29] J. Gausemeier, G. Lanza, U. Lindemann "Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung," *Hanser Verlag*, 2012.
- [30] H-P. Hoffmann, *Harmony Deskbook*, 4th ed, vol. 1. IBM, 2014.
- [31] D. Wilke, R. Grote, L. Bretz, H. Anacker, R. Dumitrescu, "Lessons learned from the Introduction of Systems Engineering," *Systems* 2023, vol. 11, 2023.
- [32] L. Humpert, M. Wäschle, S. Horstmeyer, H. Anacker, R. Dumitrescu, A. Albers, "Stakeholder-oriented Elaboration of Artificial Intelligence use cases using the example of Special-Purpose engineering," *33rd CIRP Design Conference*, 2023.
- [33] E. Mundt, D. Wilke, H. Anacker, R. Dumitrescu, "Principles for the effective application of Systems Engineering: A systematic literature review and application use case," (2023), unpublished.
- [34] M. D. Watson, B. Mesmer, G. Roedler, D. Rousseau, J. Calvo-Amodio, C. Keating, W. D. Miller, S. Lucero, R. Gold, C. Jones, D. Long, R. W. Russel, A. Sedmak, "Systems Engineering Principles," *INCOSE*, 2022.
- [35] L. Bretz, L. Kaiser, R. Dumitrescu, "An analysis of barriers for the introduction of systems engineering," *Procedia CIRP*, vol. 84, 2019, pp. 783-78.