Requirements Engineering for Digital Twins: a Cross-Domain Systematic Literature Review

Aurora Macías[1[0000-0002-1398-8826]], Elena Navarro[1[0000-0001-9496-6890]], Carlos E. Cuesta[2[0000-0003-0286-4219]] and Uwe Zdun[3[0000-0002-6233-2591]]

[1] LoUISE Research Group, University of Castilla-La Mancha; Albacete, Spain
[2] VorTIC3 group, Rey Juan Carlos University; Móstoles -Madrid- (Spain)
[3] Research Group Software Architecture, University of Vienna; Vienna, Austria
mailto:Aurora.Macias@alu.uclm.es, Elena.Navarro@uclm.es, carlos.cuesta@urjc.es and uwe.zdun@univie.ac.at

Abstract. Digital Twins (DTs) are attracting growing interest for their potential to enhance the design and operation of complex systems such as Cyber-Physical Systems. A DT is an up-to-date virtual representation of real-world entities and processes, capable of reflecting current and past states and simulating future scenarios using Artificial Intelligence-related technologies. DTs are applied across domains like smart manufacturing and healthcare. Despite recent efforts to propose holistic approaches to DT design and development, a critical gap remains: the integration of Requirements Engineering (RE) practices into the DT development lifecycle is often overlooked or insufficiently addressed. This shortage can lead to DT architectures that fail to fully meet system or stakeholder goals. This paper conducts a Systematic Literature Review to assess the extent to which RE processes have been incorporated into DT development. From an initial set of 1230 publications, 20 were selected for detailed analysis. The study examines RE methods and techniques across different phases and evaluates their maturity. Findings show that few works explicitly address RE in DT development, with most contributions focusing on early RE stages and often overlooking essential DT system properties. It concludes that there is a significant lack of structured RE processes, methods, and tools tailored for DT engineering.

Keywords: digital twins, DT, requirements engineering, requirements specification, systematic literature review.

1 Introduction

Digital Twins [1] (DTs) have gained strong support from research, industry, and governments as a promising paradigm to enhance the design and operation of IoT-based and Cyber-Physical Systems (CPSs). These systems are composed of interconnected physical and virtual components, like DTs, and support novel and disruptive applications in smart environments [2][3]. DTs are already considered a cornerstone of the architecture of such systems, as they can provide them with monitoring and control capabilities [2][4][5]. Despite their growing adoption, there is still no universally accepted definition of DTs [6]. For example, the Digital Twin Consortium defines a DT as a virtual representation of real-world entities and processes, synchronized in real time and capable of simulating future states [4]. DTs are currently used for a wide variety of purposes, including monitoring, prediction, simulation, and augmentation.

Their technological benefits include their strong potential to reduce cost and development time for systems represented by or based on DTs, as well as to improve the understandability of such systems [6]. Furthermore, DT-based systems are transforming business by enabling holistic understanding, optimal decision-making, and effective action [4]. Like CPSs, DTs have been applied across a wide diversity of domains, such as smart manufacturing, autonomous driving, medicine and healthcare, etc [6].

Beyond the lack of a standard definition, the rapid development and adoption of DTs are hindered by the absence of standardized practices [7]. There is a general lack of well-defined processes, methods, or tools for engineering and operating DTs [6]. Current research highlights that there is a no widely accepted framework for their design [7][8][9]. Therefore, DT architectures often fail to establish traceability between architectural components and the system's intended functionalities, which should ideally be derived from well-specified requirements. Requirements Engineering (RE) and Software Architecture are closely linked disciplines [10][11][12][9], with RE artifacts serving as essential inputs for architectural design [11][5] by guiding the allocation of system functions, identified during requirements analysis, to architectural elements, ensuring alignment as the system is designed and evolves and its level of detail increases [10]. However, most DT proposals fail to clearly connect requirements and architectural components of DT [7], potentially leading to misaligned system architectures [7]. Given the complexity of DTs, RE plays a critical role in their successful development [5]. Rigorous requirements specification and modeling are vital to support subsequent design and implementation phases [5], ensuring that DT-based systems meet the demands of their application scenarios [13].

Despite the importance of RE in DT development, there is currently no comprehensive understanding of how RE processes are being applied, what challenges exist, and what gaps remain. This lack of clarity hinders the development of systematic, reusable, and scalable approaches to DT engineering. To address this gap, this paper presents a Systematic Literature Review (SLR) [14][15] with the aim of identifying and analysing the current state of RE practices in DT development. Following a search strategy involving four digital libraries, we initially found 1230 unique publications. From these, 20 were ultimately selected. The main contribution of this paper is a systematic and in-depth analysis on RE in the context of DTs across different domains addressed by current proposals. Additionally, we analysed the maturity level of the identified RE proposals. The analysis reveals that only a few proposals explicitly and clearly address RE activities in the development of DTs. Most of these activities focus on the early stages of the RE process, mainly requirements definition and analysis, and are often described in vague terms. Furthermore, the identified RE activities do not comprehensively address all the essential properties that a DT-based system should satisfy. We conclude that there is a lack of processes, methods, and tools for engineering DTs from an RE perspective. Moreover, this review lays the groundwork for the future development of a holistic framework that systematically integrates RE and architectural design in DT systems. This paper is structured as follows. Section 2 presents the related work. Section 3 then details our research method. Section 4 answers the Research Questions based on our findings. Section 5 discusses the threats to the validity of our review. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Related Work

Numerous studies have explored the definition, features, and applications of DTs. However, few have comprehensively addressed RE in the DT context, despite its critical role in ensuring system quality and stakeholder alignment. RE in DT development remains underexplored and fragmented across domains. This section analyses existing literature to identify open *issues* that motivate our systematic review.

Some systematic studies have addressed the entire development lifecycle of DTs but often (**Issue 3**) omit RE processes or activities that could be applied. For example, a cross-domain systematic mapping study [6] explores how DTs are defined, engineered, deployed, used or evaluated, considering architectural and requirements-related aspects, but without explicitly addressing RE processes. It concludes that research on DTs lacks established processes, methods, or tools for engineering or operating DTs. Similarly, a healthcare-focused SLR [16] highlights (**Issue 1**) the absence of a suitable holistic approach for DT development. The study argues that such approach should cover all stages of the systems engineering lifecycle, including requirements, architecture, and V&V. It also notes that most works (**Issue 2**) heavily focused on the problem formulation and conceptualization of DTs and neglect later stages needed for successful realization of DTs, which have not been thoroughly analysed or clearly identified.

Other studies have analysed the existing literature on RE for DT. Among them, an Industry 4.0 survey [17] combines literature analysis with practitioners' interviews, identifying key requirements like real-time data handling, integration, and fidelity, but also noting that (**Issue 4**) additional requirements remain to be identified. Furthermore, it lacks discussion of RE processes or activities. Another theoretical review in the context of CPS [5] covers key RE stages such as requirements elicitation, specification and modelling, and requirements verification and validation (V&V), and identifies common cross-domain requirements like data synchronization and some non-functional requirements (NFRs). However, it does not follow a systematic approach and calls for a dedicated research agenda.

Additional works addressing RE for DTs offer valuable (**Issue 6**) domain-specific insights but often limited to a single early-stage RE activity, if any, and lacking a broader perspective. For instance, [18] compares DT concepts in construction and production, addressing requirements elicitation and providing a formal description of functional (FRs) and quality requirements. A review in rail transportation [19] examines requirements management practices through literature analysis and surveys. The study emphasizes the importance of requirements planning and acquisition, identifying (**Issue 5**) deficiencies in current requirements management methods, tools, modelling practices, data models and toolchain interoperability. In the IoT context, Minerva et al. [13] review academic literature and identify key enabling technologies, scenarios, and architectural models for domain-independent DTs and defines the properties a DT must fulfil. However, the work does not discuss RE practices. ISO 23247 [20] defines a DT framework for manufacturing networks, including requirements for DT development and information exchange, attributes for manufacturing elements, and a reference architecture with functional views. However, the scope of our work is broader, as it analyses RE activities in DT development across domains.

Thus, six key *issues* emerge: (I1) lack of a holistic approach to DT development including RE phases, (I2) unclear definition of early DT development stages, (I3) absence of information on RE processes applicable to DT development, (I4) incomplete identification of requirements (FRs and NFRs) for DTs, (I5) deficiencies in requirements management methods, modeling practices, data models, tools, and toolchain interoperability, and (I6) lack of a broad, cross-domain perspective on RE for DTs. Issues 2 to 4 are foundational gaps that hinder the proposal of a truly holistic DT development approach (I1). These gaps highlight the need for systematic, cross-domain research of RE activities in DT development. This SLR addresses *Issues 3*, *4* and *6* across domains.

3 Research Methodology

In this work, a SLR has been conducted. SLRs constitute a trustworthy, rigorous, and auditable methodology for systematic reviews [14]. Based on the widely-used Kitchenham's guidelines [15], the study was structured into three main phases (see Fig. 1):

(1) definition of research questions (see Sect. 3.1) and identification of the search string (see Sect. 3.2), (2) selection of eligible articles (see Sect. 3.3), and (3) extraction and synthesis of relevant information (see Sect. 3.4). Fig. 1 illustrates this process, highlighting each phase and its main outcomes. The replication package, including the omitted details, is publicly available at https://zenodo.org/uploads/15758010.

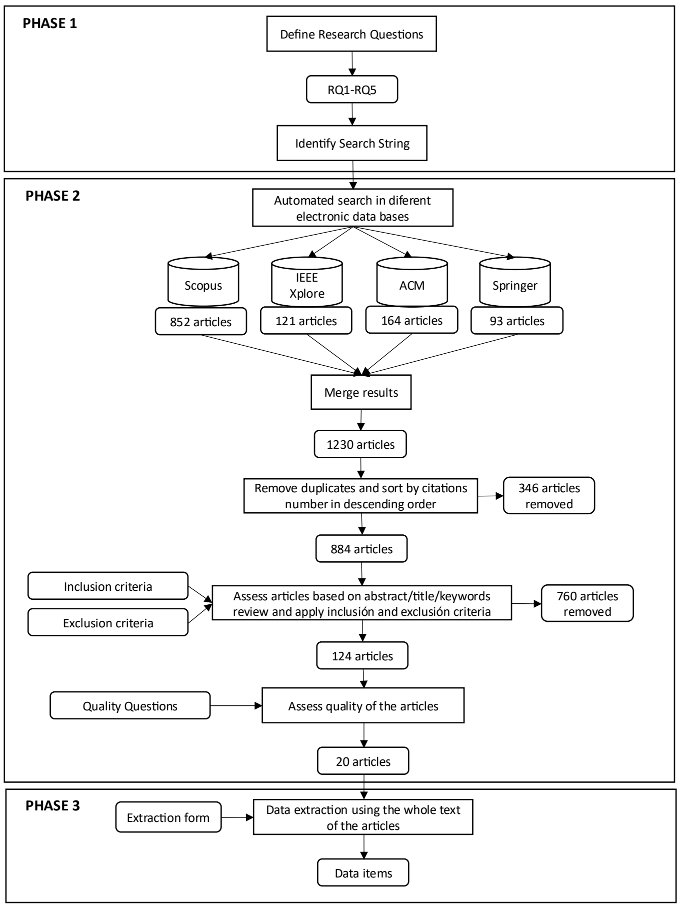

Fig. 1. Systematic Literature Review process conducted

3.1 Research Questions

As stated above, this work aims to identify how the RE process has been applied to the design and development of DT systems, and analyse the existing proposals related to the various RE activities for DTs. Thus, the main objectives (O) pursued by this SLR, in relation to the identified issues (I), and the derived Research Questions (RQ) are:

O1. Identify which RE activities and contributions are supported as part of the DT development process (related to **I3**, **I4, I6**).

- ***RQ1***: *Which phases of the RE process have been supported in the development of DTs?* This question aims to provide a starting point for understanding which RE activities are supported in DT development, focusing on commonly supported activities such as requirements elicitation, analysis, specification, V&V, and management, as recognized by international standards and prior relevant literature.

- ***RQ2**: What functional and non-functional requirements have been addressed in the RE processes for DTs?* This question seeks to identify the FRs and NFRs, including those defined by ISO/IEC 25010 [22] such as performance or usability. NFRs alongside FRs describe how software systems should perform its functions being essential for their success [21].
- ***RQ3**: What types of proposals address RE activities in the development of DTs?* This question seeks to classify the contributions that support RE in DT development by type (namely model, tool, process, method, etc. [23]). It also analyses whether these contributions are novel or adaptations.

O2. Identify the approaches adopted for software requirements modelling or specification in the context of DTs (related to **I3**, **I4, I6**).
- ***RQ4**: What styles of requirements specification have been used in the development of DTs?* This question aims to identify the requirements specification styles, such as scenario-based, goal-oriented, textual requirements and others, used in DT development. It also examines whether the authors provide feedback on the suitability of the chosen specification style.

O3. Determine the maturity level of RE proposals for DTs (related to **I3**, **I6**).
- ***RQ5**: What is the maturity level of the RE proposals for DTs?* This question aims to assess the maturity level of RE proposals for DTs by identifying the types of evaluations conducted, ranging from informal assessments, such as illustrative examples, to more formal empirical evaluations [23]. Both positive and negative results for empirical and non-empirical evaluations will be analysed.

3.2 Search String

Kitchenham's guidelines [15][24] were followed to construct and calibrate the search string. First, the major terms were derived from the research questions. Then, alternative spellings and synonyms for these terms were identified based on definitions found in the literature related to the RE activities listed in RQ2 and cross-checked against articles keywords in relevant RE articles. A pilot search using the string "digital twin*" and ("requirement* engineering" or RE) was conducted in Scopus, yielding 393 sources. Among these, only one article title included the term "requirements engineering", and it did not contain "digital twins". Based on this observation, the search string was designed to explicitly capture a wide range of RE activities, as these are more likely to appear in the title, abstract, or keywords of relevant studies. This approach enabled the retrieval of articles that address RE in practice. Nevertheless, including the general term "requirements engineering" could improve recall, and its inclusion is planned for future iterations of the review. V&V was included in the search string, since the proposals may involve some evaluation of DT characteristics based on requirements. However, keywords such as "examination", "evaluation", "experiment", or "assessment" were excluded, as they could retrieve articles involving some form of experimentation not necessarily related to V&V. As a multi-field search for "Digital Twin" was conducted, other related terms, such as "digital thread" or "digital shadow", were omitted. Therefore, publications contributing to DT research were expected to include this term in the title, keywords, or abstract. The search string was not further constrained, as the goal was to identify publications related to RE for DT regardless of the specific domain or application context. Finally, the Boolean operator AND was used to link the major terms and their synonyms, while the OR was used to include alternative spellings and synonyms. Additionally, both American and British spellings were considered. The search string was calibrated by performing a pilot test in the databases, reviewing the initial results for relevance, adjusting the terms using truncation, and validating the final version with the other authors. This process led us to define the following search string:

("digital twin") AND ("requirement* elicit*" OR "requirement* gather*" OR "requirement* collect*" OR "requirement* discover*" OR "requirement* identificat*" OR "requirement* defin*" OR "requirement* analys*" OR "requirement* classif*" OR "requirement* specif*" OR "requirement* model*" OR "requirement* communicat*" OR "requirement* document*" OR "requirement* represent*" OR "requirement* valid*" OR "requirement* verif*" OR "requirement* manage*" OR "requirement* evol*")*

The final version of the search string was reviewed and validated by all co-authors to maximize specificity, sensitivity, and coverage, thereby increasing the likelihood that the retrieved articles include terms related to the various RE activities for DTs and enabling the RQs to be answered.

3.3 Search Strategy

In this phase, the articles to be reviewed were retrieve from different electronic databases: Scopus, IEEE Xplore, ACM Digital Library, and SpringerLink. Google Scholar was excluded due to its documented limitations in structured literature retrieval [25], and to ensure the quality of the included sources. Additionally, the search was restricted to the first level, excluding reference-based snowballing to identify further studies on the topic. This decision was made because the initial search already identified the relevant articles, making further expansion unnecessary as stated in [26]. Moreover, since SLRs prioritize reproducibility and transparency, snowballing was avoided due to its lower traceability compared to database searches.

In the Scopus and IEEE Explorer databases, the search was conducted by applying the search string to the title, abstract and keywords fields only. The search was restricted to conference papers and journal articles in the engineering and computer science areas. Due to the limitations encountered in SpringerLink, a two-step process was applied to conduct the search using this database. First, the search string was applied to the full text of the articles. Then, the articles obtained were filtered by applying the search string to the title, abstract and the keywords using a third-party software. No publication year restrictions were applied. The search was executed multiple times, with the most recent run conducted on 8th of June 2025.

Table 1. Inclusion/exclusion criteria

Code	Inclusion criteria
IC1	Articles in English
IC2	Peer-reviewed publications
IC3	Articles whose full text is available
IC4	Articles that adopt RE activities in the development of DTs and provide sufficient detail: explicit descriptions of at least one RE activity and its application to DT development, including objectives, methods, or outcomes
Code	**Exclusion criteria**
EC1	Articles in any language other than English
EC2	Non-peer-reviewed publications such as a technical report, editorial note, preface, index, introduction, that present a special/ theme issue, or are unreferenced, illegible, or inaccessible (e.g., article not available through the search engines used)
EC3	Books, unpublished articles, editorial notes, master's/degree works and doctoral theses, grey literature: blogs, web pages
EC4	Articles that do not adopt RE activities in the development of DTs
EC5	Similar studies, i.e. studies by the same author at a conference or an expanded version in a journal, with similar results in both studies
EC6	Secondary or tertiary studies, that is based on other articles e.g., a literature review
EC7	Short articles with a length less than or equal to 3 pages

After performing the search, the results from the different databases were merged into a single list of 1230 articles. Then, 346 duplicate articles were removed. The titles, abstracts, keywords, and other metadata of the remaining articles were reviewed to determine their eligibility based on the predefined inclusion and exclusion criteria (see Table 1), resulting in the exclusion of 760 articles. The main reason for excluding such many articles was that, although they appeared in the search results, they did not address RE in the context of DT development, or they did not address it with sufficient detail. In many cases, even the titles did not include the terms “digital twins” or “requirements”, as illustrated by examples such as “In-process machine vision monitoring of tool wear for Cyber-Physical Production Systems”. Subsequently, the full text of the articles was read carefully to assess their relevance according to the exclusion criteria and the search expression. A quality assessment was then conducted using specific questions designed to evaluate, for instance, whether the studies clearly state their research objectives, leading to the exclusion of an additional 104 articles. As a result, 20 articles were selected for analysis (see References, Section 6.1). The selection and quality assessment were conducted by the first author and reviewed by the co-authors. Disagreements were resolved through discussion.

3.4 Data Extraction

This phase entails gathering relevant information to answer the RQs using a structured data extraction form that included fields for the demographic data of the articles and information related to the RQs. The most relevant surveys and literature reviews on RE for DTs were examined to identify fields related to the RQs and to gather further insights into how the RQs should be answered. The fields were carefully documented, and relevant data were recorded to support later analysis. Articles selected were read several times. The first author conducted the data extraction, while the remaining authors supported the decision-making process. Excel was used for data extraction, with a separate sheet for each RQ and an additional sheet for demographic data.

4 Results and Discussion

4.1 Analysed Articles: General Results

Twenty articles were selected for this study (Section 6.1), including 5 journal articles and 15 conference papers, all published between 2020 and 2025. The literature shows a growing interest in DT research, which extends to RE for DTs. However, the number of relevant articles found is low, which may indicate and confirm that research on RE for DTs is still emerging. Notably, several of the analysed articles come from disciplines other than informatics, including manufacturing, and even agriculture.

4.2 Research Questions

The findings (F) and challenges (CH) identified are coded as follows: the first number denotes the corresponding RQ, while the second number and optional letter indicate their order of appearance within that RQ. For example, (**F1-2**) refers to the second finding related to RQ1, and (**CH5-2B**) refers to the second sub-challenge of the second challenge related to RQ5, which concerns the limited justification provided by guidelines on how the goals of each step are achieved, offering only superficial validation.

RQ1. Which phases of the RE process have been supported in the development of DTs?

As noted in Section 2, (**F1-1**) tasks within RE activities often overlap. This has been also detected in the reviewed articles, as some works incorporate secondary RE activities within their adopted processes or frameworks. For instance, specification is sometimes included in elicitation/definition (e.g. P6), while in others (e.g. P7), the opposite occurs. This overlap is expected, as specifications facilitate requirements communication across RE activities. Table 2 presents the articles addressing common RE activities, categorized by their primary activity only, since details on secondary activities are often insufficient for thorough analysis. The findings support the statements from Section 2: (**F1-2**) current research primarily emphasizes the early stages of the RE process, especially "requirements definition", followed by "analysis" and "specification". P13 highlights these as the most critical activities due to the effort they involved, and the significant cost of changing requirements. Furthermore, only one article addresses verification, and two articles address validation. Regarding requirements management, only P20 explicitly addresses this activity. However, P16 highlights that (**F1-3**) effective requirements management can significantly reduce many of the negative effects that may arise during project development. Aspects such as the (**CH1-1**) need to improve requirements traceability and change management across activities are highlighted in the analysed articles as challenges that should be addressed in the near future.

Table 2. RQ1. RE process main activities supported by existing literature

RE activity	Number of articles	Articles
Elicitation/definition	5	P6, P9, P10, P13, P15
Analysis	6	P1, P2, P11, P14, P17, P19
Specification/modelling	5	P3, P4, P7, P12, P16
Validation	2	P5, P18
Verification	1	P8
Management	1	P20

Finally, some works, such as P1 or P4, (**F1-4**) combine RE activities with the design phase, treating requirements specification and design modelling as a continuous process, although the articles focus on different RE activities (and are categorized differently in the table). For instance, P1 emphasizes the mapping between requirements and design in DT-based applications, while P4 provides guidance on transforming documented requirements into validated system properties and architectural decisions as well as system's detailed design. This supports the idea that RE and design are often intertwined in practice. However, authors (**CH1-2**) do not offer explicit guidance on how to address this aspect. Moreover, a step involving the selection of the technology stack usually precedes architectural decisions, meaning that (**CH1-3**) architectural decisions are not technology-agnostic. This should be avoided, as a domain-driven approach is preferable to a technology-driven one. Such an approach enables a richer understanding of the processes and rules of the application domain [27], and consequently facilitates meeting the system requirements. This idea is also supported in P2, which states that, (**F1-5**) to design the components of a DT-based system that meet the requirements, the system boundaries and purpose must first be defined so that use cases, design goals, or data sources can be derived.

RQ2. What FRs and NFRs have been addressed in the RE processes for DTs?
DT-based systems, including their subsystems or components, models, and data storage, must meet various requirements (both FRs and NFRs), as noted by some articles (e.g. P3, P17). Requirements names and descriptions vary across works, and the use of non-standard terminology can hinder the understanding and comparison of systems in general. Table 3 lists only the non-domain-specific requirements identified, considering the cross-domain nature of the review, along with their types and sources. Some articles also identify dependencies among FRs and between FRs and NFRs, although these are generally described informally. Due to space limitations, such dependencies have been omitted.

Some FRs are specific to the application domain, while others are common across DT systems in different domains, for example, *exchange of information* (FR3). Among the FRs, FR3 is one of the most frequently addressed in the analysed articles, only surpassed by the *(uplink and downlink) data interaction requirement* (FR2), which usually helps to support FR3. Other common FRs are *decision-making* (FR10), which is defined as data-driven in some articles (e.g. P4), and *control* (FR11). Regarding data, several FRs are associated with *different stages of a common data lifecycle* [28], such as acquisition, preprocessing, storage, etc. (FR1, FR3, FR4, FR5, F6). As noted in P1, these requirements must be fulfilled to satisfy FR10 or FR8. The *DT model construction* requirement (FR4) is essential in DT systems to enable the necessary analysis. DT models representing the same physical entity may have different needs depending on the application. The initial establishment of a DT model of the physical entity depends on obtaining basic information of the physical entity. (**F2-1**) The prevalence of these requirements is explainable mostly by the definition of DT and its operation.

Table 3. RQ2. Non-domain-specific requirements addressed in the RE processes for DTs

Requirement category	Id (includes Type)	Articles
DT model construction	FR5	P1, P19
Data acquisition/collection	FR1	P1, P19
Uplink and downlink data interaction	FR2	P1, P4, P11, P15, P19
Data preprocessing	FR4	P1
Data storage	FR6	P1, P19
Failure prediction	FR8	P1
(Data-driven) decision-making	FR10	P1, P4, P19
Context- and self-awareness	FR12	P4, P10
Self-organization	FR13	P4
Exchange of information	FR3	P4, P1, P19
Simulation	FR7	P3
Tracking status or quality	FR9	P6, P19
Control	FR11	P10, P18, P19
Integration	NFR1	P15
Modularity	NFR2	P2, P3
Functional completeness	NFR3	P3
Accuracy	NFR4	P3, P6, P17
Robustness	NFR5	P3, P17
Performance	NFR6	P3, P17
Completeness	NFR7	P17
Timeliness	NFR8	P11, P13, P14, P17, P19
Interoperability	NFR9	P4, P5, P9, P11, P15
Consistency	NFR10	P5, P17
Reusability	NFR11	P5
Reconfigurability	NFR12	P6, P10
Availability	NFR13	P11
Scalability	NFR14	P14, P15
Expansibility	NFR15	P15
Fidelity	NFR16	P15
Safety	NFR17	P16, P18
Security	NFR18	P8, P16, P17
Privacy	NFR19	P17
Ethical and legal compliance	NFR20	P17

[a.] FR: Functional requirement; NFR: Non-functional requirement

Regarding NFRs, *interoperability* (NFR6) and *timeliness* (NFR7) are among the most frequently addressed, as many works emphasized the need to receive, and process information in real-time (NFR7), relaying on the seamless integration of the diverse components within the DT system (NFR6). Other addressed requirements relate to system and DT operation, such as *performance* (NFR6) or *robustness* (NFR5). Another group of NFRs, such as *modularity* (NFR2), *reusability* (NFR11), or *expansibility* (NFR15), concerns the extent to which a DT can be modified to improve, correct, or adapt it to changes, at least in requirements. An additional set of requirements includes *safety* (NFR17), *security* (NFR18) and *privacy* (NFR19), and *ethical and legal* *compliance* (NFR20). Some works identify the use of quality models, based on factors, criteria and metrics adapted from software development as a promising approach for DTs development [29]. For example, P4 points out that the use of DTs to enhance product innovation is shifting the required development effort from the mechanical field to the software domain, particularly in software intensive products driven by trends in digitalization, communication, and interconnection of systems. P16 is the only work that explicitly specifies requirements based on international standards, although these do not belong to the software engineering field. According to P3, in the manufacturing context, standard simulation tasks (FR7) must be linked to specifiable and testable quality requirements, or NFRs, of DTs as a precondition for enabling assessment and certification of DTs by independent instances. In fact, (**CH2-1**) relying on standardized requirements that can be systematically specified and tested is a general desirable aspect across different application domains and software development practices. The last two groups of NFRs are common to all software-intensive systems and share the same rationale. Furthermore, most of them are defined by quality international standards, such as (**F2-2**) modularity, reusability or performance, defined in ISO 25010 [22], suggesting that this standard could be used for guiding the specification of DTs.

The non-domain-specific requirements listed in Table 3 from a broad and comprehensive set, covering key FRs, most of which are aligned with the DT definition, along with some standard NFRs. (**CH2-2**) This set should be extended to include, at a minimum, domain-specific requirements and, ideally, be compared with the DT properties defined by Minerva et al. (see Section 2), with the aim of integrating both sets into a unified and comprehensive framework. In this regard, P1 notes that much works focus on the digital counterpart, while there has been (**CH2-3**) limited attention to requirements associated to the physical entity, which are essential for the successful integration of both components. Our findings support this observation, as only four requirements pertain to this aspect: *data acquisition/collection* (FR1), *uplink and downlink data interaction* (FR2), *integration* (NFR1), and *interoperability* (NFR9). Much of the current literature prioritizes the development of digital models, simulations, and analytics, often seen as the core value of DTs, while overlooks the physical component of digital twins, assuming it is already well-understood from traditional engineering fields. Similarly, (**CH2-4**) AI-related requirements lack attention due to the novelty of the field, and only four such requirements were identified in our results: *data preprocessing* (FR4), *failure prediction* (FR8), (data-driven) *decision-making* (FR10), and *simulation* (FR7).

RQ3. What types of proposals address RE activities in the development of DTs? Table 4 shows that most analysed articles offer guidelines addressing RE for DTs, mainly in the form of *methods* or *processes*. There is a clear (**CH3-1**) lack of suitable tools, techniques, and methodologies used to support RE activities in DT development, and further research is needed, as noted in Section 2 (issue *I5*) and further analysed in [P4].

Table 4. RQ3. RE proposal types

Proposal type	Number of articles	Articles
Method	8	P4, P7, P8, P10, P11, P16, P17, P19
Process	7	P1, P3, P5, P9, P12, P13, P14
Framework	3	P2, P6, P18
Model	1	P15
Technique	0	
Tool	0	
Methodology	0	

Notably, (**F3-1**) a considerable number of proposals integrate complementary approaches, motivated by the overlap among RE activities or as an attempt to address the deficiencies observed when these approaches are used in isolation, as well as to respond to CH3-1. The most common are processes based on established engineering approaches, such as Model-Based Software Engineering (MBSE) (P4, P5, P11, P18), specification techniques (P5, P9, P11, P18), and standards (P16). The modern DevOps lifecycle is also adopted in P10 to guide designers in systematically identifying functional and technical aspects of DT design. Sometimes, the goal is to achieve acceptable requirements traceability through the entire process, as in P13, which combines an MBSE-based approach and the Systems Modelling Language (SysML) to support traceability. Ideally, (**CH3-2**) requirements traceability should span requirements definition, architectural design, and the deployment of architectural components, and this process should be at least partially automated. In this regard, P1 presents an approach based on requirements traceability and emphasizes the (**CH3-3**) need for further research on deployment scheme design based on requirements across different domains with varying demands.

In addition, (**F3-2**) most proposals rely on existent approaches, either adapting them (P4), combining them (P13), or even borrowing them from other disciplines or domains (P10, P18). An exception is P7 (see RQ4), which presents a novel requirements specification language. Future work should (**CH3-4**) assess existing proposals against the identified challenges to determine whether any of them, or a combination, can effectively address these challenges, or whether novel approaches are needed.

RQ4. What styles of requirements specification have been used in the development of DTs?

The overlap also affects specification styles. Most specification proposals are part of broader proposals rather than focusing solely on the specification method. Moreover, Table 5 shows that some articles use multiple specification styles; for instance, P6 and P16 employ both textual and diagram-based requirements. Additionally, Table 5 highlights whether any formal or informal feedback on the proposal's suitability or feasibility is provided (e.g. P5, P6, P7). This feedback is generally positive, although often points out limitations or open issues and suggests further improvements.

Table 5. RQ4. Requirements specification styles used in DT development

Specification style	Number of articles	Articles
Textual requirements	10	P1, P2, P3, P5*, P6*, P7*, P14, P15, P16, P19
Requirements diagram	5	P6*, P10, P13, P16, P18
Ontology	3	P4, P5*, P9
Use cases	2	P18, P19
Goal-oriented	1	P7*

* Provided feedback on suitability/feasibility

The textual style is the most used for requirements specification. It involves listing requirements as plain descriptions, without using any structured template. In some articles (P6, P7, P19), it is used to (**F4-1**) present or explain the requirements before applying a more sophisticated approach. The second most used style is based on requirements diagrams, which illustrate relationships among requirements and often include textual descriptions of varying lengths. Ontologies are the third most used style and can be seen as a more advanced form of basic requirements diagrams. The most advanced style, goal-oriented (GO), is only used in P7, the sole DT-specific proposal. In the analysed work, the requirements are specified using a textual format, organized in a table structured based on the language's metamodel, without graphical elements. Using a suitable modelling language is highly recommended, as it enhances requirements communication, especially when (**CH4-1**) supported by appropriate tools.

Some articles specify requirements using diagrams or ontologies developed with certain standard approaches (P5, P13, P18), such as SysML, or graphical notations like

BPMN for modelling the sequence of activities in business processes. Other notations include DMN (Decision Model and Notation), used for the precise specification of business decisions and business rules, and SDMN (Shared Data Model and Notation), used for specifying shared data models (P13). It reflects the (**CH4-2**) need for standardization in the requirements specification style as well.

A considerable number of articles (P1, P3, P6, P10, P13, P18, P19) define (**F4-2**) different models or views for specifying the whole set of requirements. The resulting model may include, for example, a behavioural view to deal with the specification of functional aspects, and a structural view to address the specification of static data-related aspects. Most of these works specify different models as independent views of the system such as P5. A few others, for instance P13, integrate them to specify the relationships between requirements and process elements that satisfy them, as well as the data consumed or produced by such process elements. P10 defines business- and provider-focused viewpoints rather than the usual customer-oriented one, to specify FRs and NFRs of the different application services and their enabling DTs. However, the (**CH4-3**) proper integration of the different views remains as an open problem [P10].

An association between requirements and DT-based system architectural components is proposed using a manually developed matrix in P2. Notably, this association is based on DT components identified from a specific definition of a DT. Moreover, the requirements are specified in textual form. (**CH4-4**) A more systematic and automated process for constructing such an association would be desirable. Use cases and GO approaches are well-known methods for analysing and specifying requirements. Use cases are suitable for specifying concrete system behaviour, while GO approaches are more appropriate for specifying system constraints [30]. In this sense, GO approaches present a higher degree of expressiveness. In the analysed articles, goals (P7) are used to refine FRs and NFRs, explore alternatives, and map them to architectural constructs. (**F4-3**) GO approaches can be effectively used to bridge requirements and architecture, which is a desirable practice. Additionally, P19 proposes (**CH4-5**) a natural language processing method for analysing requirements specified as use cases, with the goal of automatically deriving system classes or components. Extending a similar processing method to more expressive approaches, like GO specifications, could be a promising step toward addressing *CH3-2* and *CH3-3***.**

RQ5. What is the maturity level of the RE proposals for DTs?

Table 6 shows that the case study with application scenarios is the most common evaluation approach in RE proposals for DT. No articles conduct experiments, the most formal and rigorous validation approach [23], and three lack any form of evaluation. This suggests that (**F5-1**) the maturity level of the field RE for DTs remains low.

Table 6. RQ5. Evaluation approaches used by DT proposals

Evaluation approach	Number of articles	Articles
Case study	12	P1, P6, P7, P10, P11, P12, P13, P15, P16, P17, P18, P19
No evaluation	3	P3, P8, P9
Example	3	P2, P4, P5
Survey	1	P14
Experiment	0	

Evaluating using examples (e.g. P2, P4) and case studies (e.g. P6, P10) follows an identical approach. The evaluation involves describing a (**CH5-1**) DT-based system, which is hypothetical in the case of examples, based on the proposed (**F5-2**) guidelines that define the process steps. The guidelines are usually divided into sections corresponding to the evaluated part of the system and its requirements and include the (**F5-**

2) summary of the validation results using a defined template. Typically, the guidelines include some (**CH5-2A**) subjective elements to justify the fulfilment of requirements. In other cases (e.g. P4), guidelines provide (**CH5-2B**) limited information and justification regarding how the goals of the different steps are achieved and the validation results, beyond simply affirming their completion. In most cases (e.g. P10), articles claim that the method can be feasibly applied in real-world use cases, although they do not provide any rationale to support this statement. Typically, they (**CH5-3**) fail to follow a suitable validation approach that could be standardized. Finally, although the assessment results are (partially) positive in all cases considering the scope of each work, several articles, namely P4, P5, P7, P13, P15, identify potential enhancements, open issues, and challenges that should be addressed in the near future. The above suggests that the approaches used to evaluate RE proposals for DTs should also be improved.

5 Threats to Validity

According to [31], the main threats to the validity of a SLR are publication selection bias, inaccuracies in data extraction, and misclassification. The use of four digital libraries in this SLR ensured broad coverage of journals and conferences in DTs and RE, supporting reasonable completeness. However, some relevant articles might have been missed due to indexing limitations. To ensure an *unbiased selection process*, rigorous rules were followed: the RQs were defined in advance, a multistage article selection process involving four researchers was implemented, and inclusion/exclusion decisions were documented following established guidelines [26]. The *data extraction* was performed by the first author and verified by the second, with disagreements resolved collaboratively. Due to the lack of standardized terminology in defining DT quality characteristics and the cross-domain nature of the review, data extraction was challenging and may have led to *misclassifications*. To mitigate this, a rigorous process was followed: the initial classification was performed by the first author, and then independently reviewed by the other three authors. Discrepancies were discussed collectively until consensus was reached, ensuring consistency and reducing potential errors.

6 Conclusions and Future Work

In this work, a SLR on RE for DTs across the different domains is presented. It investigates and identifies evidence related to activities, methods and techniques, types of requirements, and more, addressed by current proposals. The maturity level of the different RE proposals is also analysed. A total of twenty articles were ultimately included. The analysis revealed that (**RQ1**) (I3) few proposals explicitly and clearly address RE activities in the context of DT development. Most are in the early stages of research, and there is a frequent overlap among the tasks involved. Furthermore, (**RQ2**) (I4) these RE activities do not fully address the essential properties and requirements that a DT-based system should meet, partly due to the need to extend the basic set of domain-independent requirements including also AI-related requirements. We conclude that (**RQ3**, **RQ4** and **RQ5**) (I6) there is still a lack of processes, methods, and tools for engineering DTs from an RE perspective. This gap, as highlighted in I5, remains a significant challenge, although it could be partially addressed through the improved application of (a combination of) existing RE methods. An assessment of current methods is required to determine whether new ones are truly necessary.

Based on this analysis, several observations and recommendations (R) regarding open issues can be formulated, serving as a preliminary roadmap. First, there is a general need to enhance requirements traceability and change management across DT development activities, especially within RE. While elicitation is the most frequently addressed activity, followed by requirements analysis and specification, specification often overlaps with others and is sometimes merged with design. Some articles emphasize traceability between requirements and architecture. However, architectural decisions are frequently driven by technology choices rather than domain needs. (**R1**) A domain-driven approach is preferable to a technology-driven one, since the former fosters a deeper understanding of the processes and rules of the application domain and, consequently, and better alignment between architecture and requirements.

Regarding the requirements communication, (**R2**) GO approaches provide a more detailed and expressive requirements specifications than other methods, providing a more systematic alternative to, for example, basic textual styles. Furthermore, GO approaches, especially when incorporating textual descriptions, can support the traceability of requirements to architectural components, as well as their automatic analysis though certain types of processing, thereby facilitating component deployment.

As future work, we plan to investigate (RQ4) (I5, I6) the most suitable approach for specifying the (RQ2) (I4) requirements of a DT-based system, addressing R2. The aim is to provide a comprehensive and valuable input to a domain-driven approach, thereby also addressing R1. The goal is to integrate this specification approach with domain-driven design into (I1) a suitable holistic framework that offers clear guidance from requirements definition to the architectural design of DT systems, including the necessary automation.

Acknowledgments. This paper has been partially funded by the Erasmus+ grant for short-term doctoral stays 2025 (reference number 2023-1-ES01-KA131-HED-000113966-MOB-0733), awarded to Aurora Macías. It is also part of the R+D+i project PID2022-140907OB-I00 funded by MICIU/AEI /10.13039/501100011033 and ERDF, EU. It has also been partially supported by Junta de Comunidades de Castilla-La Mancha/ERDF (SBPLY/24/180501/000020) and by the University of Castilla-La Mancha (2023-GRIN-34436). Uwe Zdun's work was supported by the FFG (Austrian Research Promotion Agency) project MODIS (no.\ FO999895431).

References

6.1 Analysed Articles

P1. Y. Wei, T. Hu, Y. Wang, S. Wei, and W. Luo, "Implementation strategy of physical entity for manufacturing system digital twin," Robot Comput Integr Manuf, vol. 73, p. 102259, 2022, doi: https://doi.org/10.1016/j.rcim.2021.102259.

P2. E. Brucherseifer, H. Winter, A. Mentges, M. Mühlhäuser, and M. Hellmann, "Digital Twin conceptual framework for improving critical infrastructure resilience," *Automatisierungstechnik* vol. 69, no. 12, pp. 1062–1080, 2021, doi: doi:10.1515/auto-2021-0104.

P3. J. Mädler, I. Viedt, J. Lorenz, and L. Urbas, "Requirements to a digital twin-centered concept for smart manufacturing in modular plants considering distributed knowledge," in *14th International Symposium on Process Systems Engineering*, vol. 49, in Computer Aided Chemical Engineering, vol. 49. Elsevier, 2022, pp. 1507–1512. doi: https://doi.org/10.1016/B978-0-323-85159-6.50251-7.

P4. M. Sjarov, D. Kißkalt, T. Lechler, A. Selmaier, and J. Franke, "Towards 'Design for Interoperability' in the context of Systems Engineering," *Procedia CIRP*, vol. 96, pp. 145–150, 2021, doi: https://doi.org/10.1016/j.procir.2021.01.067.

P5. X. Hu *et al.*, "Ontology-centric industrial requirements validation for aircraft assembly system design," *IFAC-PapersOnLine*, vol. 55, pp. 3016–3021, 2022, doi: https://doi.org/10.1016/j.ifacol.2022.10.191.

P6. A. Ramesh, Z. Qin, and Y. Lu, "Digital thread enabled manufacturing automation towards mass personalization," *15th International Manufacturing Science and Engineering Conference, MSEC 2020*, 2020.

P7. B. S. Barn, T. Clark, S. Barat, and V. Kulkarni, "Towards the Essence of Specifying Sociotechnical Digital Twins," in *Proceedings of the 16th Innovations in Software Engineering Conference*, in ISEC '23. New York, NY, USA: ACM, 2023. doi: 10.1145/3578527.3578542.

P8. A. C. da Silva, S. Wagner, E. Lazebnik, and E. Traitel, "Using a Cyber Digital Twin for Continuous Automotive Security Requirements Verification," *IEEE Softw*, vol. 40, no. 1, pp. 69–76, 2023, doi: 10.1109/MS.2022.3171305.

P9. N. Moretti, X. Xie, J. M. Garcia, J. Chang, and A. K. Parlikad, "Federated Data Modeling for Built Environment Digital Twins," *Journal of Computing in Civil Engineering*, vol. 37, no. 4, p. 4023013, 2023, doi: 10.1061/JCCEE5.CPENG-4859.

P10. J. Dobaj, A. Riel, G. Macher, M. Egretzberger, "A Method for Deriving Technical Requirements of Digital Twins as Industrial Product-Service System Enablers," in *Systems, Software and Services Process Improvement*, Cham: Springer International Publishing, 2022, pp. 378–392.

P11. N. C. Schäfer, P. Burggräf, and T. Adlon, "Application of a Digital Twin for Proactive Production Planning," *SNAME Maritime Convention*, 2022.

P12. S. W. Chu *et al.*, "A Model-based Approach for Digital Testbed Development supporting Virtual Experimentation of an Unmanned Surface Vehicle," *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*, 2022.

P13. B. Intrigila, G. Della Penna, A. D'Ambrogio, D. Campagna, and M. Grigore, "Process-Oriented Requirements Definition and Analysis of Software Components in Critical Systems," *Computers*, vol. 12, no. 9, 2023, doi: 10.3390/computers12090184.

P14. M. Anan, D. Khamhangwong, S. Yamsa-Ard, P. Sureephong, and S. Wicha, "The Requirements Analysis of a Smart Monitoring System Designed for the Rice Milling Process: A Case Study of the Agricultural Cooperative Sector in Chiang Rai," in *2023 International Conference On Cyber Management And Engineering (CyMaEn)*, 2023, pp. 29–32. doi: 10.1109/CyMaEn57228.2023.10051025.

P15. A. Crespo Márquez, U. Leturiondo, J. A. Marcos, A. J. Guillén, and E. Candón, "On the Definition of Requirements for a Digital Twin. A Case Study of Rolling Stock Assets," in *16th WCEAM Proceedings*, Cham: Springer International Publishing, 2023, pp. 76–86.

P16. A. M. Hosseini, T. Sauter, and W. Kastner, "Safety and Security Requirements in AAS Integration: Use Case Demonstration," in *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*, 2023, pp. 1–8. doi: 10.1109/WFCS57264.2023.10144245.

P17. M. Brünninghaus and M. Redeker, "Integrating Worker Assistance Systems and Enterprise Resource Planning in Industry 4.0," in *2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2023, pp. 1–6. doi: 10.1109/ICPS58381.2023.10128096.

P18. J. H. Murphy, C. Dickerson, C. Goodier, S. Zahiroddiny, and T. Thorpe, "A Requirements Validation Framework for Major Infrastructure Projects," in *2022 IEEE International Symposium on Systems Engineering (ISSE)*, 2022, pp. 1–8. doi: 10.1109/ISSE54508.2022.10005510.

P19. J. Jura, P. Trnka, and M. Cejnek, "Using NLP to analyze requirements for Agriculture 4.0 applications," in *2022 23rd International Carpathian Control Conference (ICCC)*, 2022, pp. 239–243. doi: 10.1109/ICCC54292.2022.9805905.

P20. Rigby J, Jackson B. Requirements management for lean digital twins. Data-Centric Engineering. 2024;5:e22. doi:10.1017/dce.2024.17

6.2 Other References

[1] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, Springer, 2017, pp. 85–113. doi: 10.1007/978-3-319-38756-7_4.

[2] A. Macías and E. Navarro, "Paradigms for the conceptualization of Cyber-Physical-Social-Thinking hyperspace: A Thematic Synthesis," J Ambient Intell Smart Environ, vol. Preprint, pp. 1–32, 2022, doi: 10.3233/AIS-210492.

[3] A. Macías, D. Muñoz, E. Navarro, and P. González, "Digital Twins-Based Data Fabric Architecture to Enhance Data Management in Intelligent Healthcare Ecosystems," in International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022), Springer 2023, pp. 38–49.

[4] J. Turner, R, Ferris, S, Eckhardt, "Infrastructure Digital Twin Maturity: A Model for Measuring Progress," 2021. Accessed: Dec. 16, 2023. [Online]. Available: https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/Infrastructure-Digital-Twin-Maturity-Model.pdf

[5] T. Yue, S. Ali, P. Arcaini, F. Ishikawa "Towards Requirements Engineering for Digital Twins of Cyber-Physical Systems," in Leveraging Applications of Formal Methods, Verification and Validation. Practice, B. Margaria Tiziana and Steffen, Ed., Cham: Springer Nature Switzerland, 2022, pp. 9–21.

[6] M. Dalibor et al., "A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins," J. of Systems and Software, vol. 193, p. 111361, 2022, doi: https://doi.org/10.1016/j.jss.2022.111361.

[7] C. Nwogu, G. Lugaresi, A. Anagnostou, A. Matta, and S. J. E. Taylor, "Towards a Requirement-driven Digital Twin Architecture," Procedia CIRP, vol. 107, pp. 758–763, 2022, doi: https://doi.org/10.1016/j.procir.2022.05.058.

[8] A. Macías, E. Navarro, C. E. Cuesta, and U. Zdun, "Architecting Digital Twins Using a Domain-Driven Design-Based Approach," in IEEE 20th International Conference on Software Architecture (ICSA), 2023, pp. 153–163. doi: 10.1109/ICSA56044.2023.00022.

[9] E. Ferko, A. Bucaioni, and M. Behnam, “Architecting Digital Twins,” IEEE Access, vol. 10, pp. 50335–50350, 2022, doi: 10.1109/ACCESS.2022.3172964.

[10] T. Spijkman, S. Molenaar, F. Dalpiaz, and S. Brinkkemper, “Alignment and granularity of requirements and architecture in agile development: A functional perspective,” Inf Softw Technol, vol. 133, p. 106535, 2021, doi: https://doi.org/10.1016/j.infsof.2021.106535.

[11] “ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering,” ISO/IEC/IEEE 29148:2018(E), pp. 1–104, 2018, doi: 10.1109/IEEESTD.2018.8559686.

[12] R. C. de Boer and H. van Vliet, “On the similarity between requirements and architecture,” Journal of Systems and Software, vol. 82, no. 3, pp. 544–550, 2009, doi: https://doi.org/10.1016/j.jss.2008.11.185.

[13] R. Minerva, G. M. Lee, and N. Crespi, “Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models,” Proceedings of the IEEE, vol. 108, no. 10, pp. 1785–1824, 2020, doi: 10.1109/JPROC.2020.2998530.

[14] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering – A systematic literature review,” Inf Softw Technol, vol. 51, no. 1, pp. 7–15, 2009, doi: https://doi.org/10.1016/j.infsof.2008.09.009.

[15] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” 2007.

[16] Md. D. Xames and T. G. Topcu, “A Systematic Literature Review of Digital Twin Research for Healthcare Systems: Research Trends, Gaps, and Realization Challenges,” IEEE Access, vol. 12, pp. 4099–4126, 2024, doi: 10.1109/ACCESS.2023.3349379.

[17] L. F. C. S. Durão, S. Haag, l R. Ander, K. Schützer, and Zancul. Eduardo, “Digital Twin Requirements in the Context of Industry 4.0,” in Product Lifecycle Management to Support Industry 4.0, Springer International Publishing, 2018, pp. 204–214.

[18] M. and H. J. and V. O. Gerhard Detlef and Wolf, “Digital Twin Representations of Concrete Modules in an Interdisciplinary Context of Construction and Manufacturing Industry,” in Product Lifecycle Management Enabling Smart, Springer International Publishing, 2020, pp. 101–115.

[19] Y. Chen and J. R. Jupp, “Challenges to requirements management in complex rail transport projects,” Int J Prod Lifecycle Manag, vol. 15, no. 2, pp. 139–177, 2023, doi: 10.1504/IJPLM.2023.135336.

[20] ISO, “Digital twin framework for manufacturing, Standard ISO 23247-1:2021,” 2021. Accessed: Jun. 09, 2025. [Online]. Available: https://www.iso.org/standard/75066.html

[21] W. Behutiye, P. Karhapää, D. Costal, Oivo Markku, and X. Franch, “Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal,” in Product-Focused Software Process Improvement, Springer International Publishing, 2017, pp. 515–522.

[22] ISO/IEC 25010, “ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models,” 2011.

[23] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion,” Requir Eng, vol. 11, no. 1, pp. 102–107, 2006, doi: 10.1007/s00766-005-0021-6.

[24] B. Kitchenham, Z. Li, and A. Burn, “Validating Search Processes in Systematic Literature Reviews,” Jun. 2011, pp. 3–9. doi: 10.5220/0003557000030009.

[25] M. Boeker, W. Vach, and E. Motschall, “Google Scholar as replacement for systematic literature searches: good relative recall and precision are not enough,” BMC Med Res Methodol, vol. 13, no. 1, p. 131, 2013, doi: 10.1186/1471-2288-13-131.

[26] M. Piattini, G. Poels, M. Genero, A. M. Fernández-Saez, and H. J. Nelson, “Research Review: A Systematic Literature Review on the Quality of UML Models,” J. Database Manage., vol. 22, no. 3, pp. 46–70, Jul. 2011, doi: 10.4018/jdm.2011070103.

[27] E. J. Evans, Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional, 2004.

[28] A. Macías, D. Muñoz, E. Navarro, and P. González, “Data fabric and digital twins: An integrated approach for data fusion design and evaluation of pervasive systems,” Information Fusion, vol. 103, p. 102139, 2024, doi: https://doi.org/10.1016/j.inffus.2023.102139.

[29] J. Mädler, I. Viedt, and L. Urbas, “Applying quality assurance concepts from software development to simulation model assessment in smart equipment,” Computer Aided Chemical Engineering, vol. 50. , Elsevier, 2021, pp. 813–818. doi: https://doi.org/10.1016/B978-0-323-88506-5.50127-3.

[30] K. Watahiki and M. Saeki, “Combining goal-oriented analysis and use case analysis,” IEICE Trans Inf Syst, vol. 87, no. 4, pp. 822–830, 2004.

[31] F. Shull, J. Singer, and D. I. K. Sjøberg, Eds., Guide to Advanced Empirical Software Engineering. Berlin: Springer London, 2008.