Systematic Literature Review on the Implementation of the Scrum Framework in Software Development Using the PRISMA Method

Batya Nomi Panjaitan[a], Syarifah Atika[b]
[a,b]*Program Studi Teknik Informatika, Universitas Prima Indonesia*
Corresponding Author:
[b]*syarifahatika@unprimdn.ac.id*

ABSTRACT
The Scrum Framework is widely adopted in software development, but the evidence on how it is applied, what benefits it brings, and what challenges it faces is scattered across many study designs. This study is a Systematic Literature Review (SLR) reported in accordance with PRISMA 2020 to answer three questions: how Scrum is implemented (RQ1), what its benefits are (RQ2), and what its challenges are (RQ3). The search was conducted on Google Scholar for English-language publications from 2020 to 2025 using three search strings. Of the 17,400 results reported by Google Scholar, 250 records passed screening, 40 articles were judged eligible, and 13 studies were included in a narrative-thematic synthesis. A retrospective quality assessment, using instruments matched to each study design, could be carried out for six of the 13 studies whose full texts were accessible. Scores ranged from 35% to 90%, with two studies rated high, two moderate, and two low. The synthesis shows that Scrum implementation keeps its iterative core but is adapted through internal adjustments and the addition of other methods. The reported benefits are consistent in direction but conditional and mostly perception-based. The challenges are spread across the process, team, and organizational levels, with partial adoption, dual roles, and mid-sprint changes standing out in case studies of public organizations. The findings should be interpreted with caution because the search covered only one source, the documentation of the selection process is limited, and the study designs are heterogeneous.

Keywords : Scrum Framework, Agile, Software Development, Systematic Literature Review, PRISMA

INTRODUCTION

Software development requires methods that are adaptive, collaborative, and able to respond quickly to changing user needs. Sequential models are considered inadequate for products whose requirements change frequently, so Agile methodologies have emerged as an alternative (Hema et al., 2020) . Among Agile methods, Scrum is the most popular. (Verwijs & Russo, 2023) cite a market survey reporting that 81% of organizations used Scrum or its hybrid versions in 2021, while (Lee & Chen, 2023) cite another survey reporting that at least 72% of respondents applied Scrum or a hybrid approach that includes it.

Scrum has been studied in diverse contexts, from small-scale web projects (Pinto & Leme, 2024) to public organizations (Pinto & Leme, 2024), using varied research designs ranging

from surveys, case studies, and experiments to literature reviews and conceptual studies. Several systematic reviews have summarized specific aspects, for example the reasons for and ways of modifying Scrum across 925 studies (Hron & Obwegeser, 2022) and the tailoring of Scrum characteristics (Garcia et al., 2022) . Understanding of how Scrum is applied, what benefits are obtained, and what challenges are faced remains scattered across many sources and study designs in the recent literature.

This study conducts a systematic review of publications from 2020 to 2025, guided by the PRISMA 2020 statement (Page et al., 2021). It makes three main contributions: (1) it presents the literature search and selection process transparently, including the aspects that were not fully documented; (2) it performs a retrospective quality evaluation of the studies available in full text; and (3) it produces a comparative synthesis that identifies similarities, differences, and relationships among the findings, so that the discussion goes beyond summarizing each study. Based on these objectives, the study is formulated into three research questions (RQs):

- RQ1: How is Scrum applied in the software development process?
- RQ2: What benefits are obtained from applying Scrum?
- RQ3: What obstacles and challenges are faced in applying Scrum?

LITERATURE REVIEW

Scrum Framework

The Agile Manifesto values individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (Pinto & Leme, 2024) . Scrum is often compared with Kanban (Zayat & Senvar, 2020) and with sequential models such as waterfall (Hema et al., 2020) . Such comparisons are important because the choice of method depends on project characteristics rather than on the absolute superiority of any one method.

Agile and Its Comparison with Other Approaches

The Agile Manifesto values individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (Pinto & Leme, 2024)) . Scrum is often compared with Kanban (Zayat & Senvar, 2020) and with sequential models such as waterfall (Hema et al., 2020) . Such comparisons are important because the choice of method depends on project characteristics rather than on the absolute superiority of any one method.

PRISMA 2020

PRISMA is a reporting guideline for systematic reviews (Page et al., 2021) . The 2020 statement replaces the 2009 version and contains a 27-item checklist and a revised flow diagram. The 2020 flow diagram presents three phases, namely identification, screening, and included studies, with the assessment of full-text eligibility placed within the screening phase. The diagram requires reporting of records removed before screening (for example, duplicates), reports sought but not retrieved, and reasons for exclusion at the full-text stage. This study presents its selection flow in that format and explicitly marks the figures that were not documented.

Quality Assessment in SLRs

Quality assessment aims to evaluate the methodological rigor of the included studies so that the strength of evidence can be considered in the synthesis. Instruments are matched to the study design: (Dybå & Dingsøyr, 2008) developed 11 items to assess empirical studies of Agile software development, whereas systematic reviews are commonly assessed with the criteria of (Kitchenham et al., 2010), who also note that systematic reviews in software engineering still often fail to assess the quality of primary studies. Quality scores can be used either to filter studies or to weigh evidence; (Hron & Obwegeser, 2022) chose not to use them as a filtering criterion so that the sample would reflect the breadth of the literature.

METHODS

Research Design

This study is a Systematic Literature Review reported according to PRISMA 2020 ((Page et al., 2021) and guided by the SLR guidelines for software engineering (Kitchenham et al., 2010). The manuscript does not refer to a review protocol that was published or registered before the search. The search and selection were carried out in the initial study, while the quality assessment and the verification of data extraction were added retrospectively; both are described in Sections 3.6 and 3.7.

Research Questions

The three research questions are broken down into six synthesis themes (Table 1). Theme T6 cuts across the questions because it compares Scrum with other approaches.

Table 1. Research questions and synthesis themes

Code	Research question	Synthesis theme
RQ1	How is Scrum implemented in software development?	T1 Implementation context and patterns; T2 Roles in the Scrum Team; T3 Artifacts, sprints, and events
RQ2	What are the benefits of implementing Scrum?	T4 Reported benefits and their conditions
RQ3	What are the challenges of implementing Scrum?	T5 Challenges at the process, team, and organizational levels
Cross-RQ	How does Scrum compare with other approaches?	T6 Comparison of Scrum, Agile, and Kanban

Search Strategy

The search was conducted on Google Scholar with the publication year limited to 2020–2025 and the language limited to English. Three search strings were run (Table 2). Google Scholar was the only search source in this study.

Google Scholar displays the number of results as an estimate and generally allows browsing of only about the top one thousand results for each query. Therefore, the figure of 17,400 is the number of reported results, not the number of unique records that were examined one by

one. The search date, the number of results for each search string, the search field (full text or title), the order of results, and the reference management tools were not documented at the time of the initial search and therefore cannot be reported. The consequences for replication are discussed in Section 7.

Table 2. Search strings and the reported number of results

Code	Search string	Reported number of results
S1	(“Scrum Framework” AND “Software Development”)	Not itemized
S2	(“Scrum Framework” AND “Agile Software Development”)	Not itemized
S3	(“Scrum” AND “Software Engineering”)	Not itemized
Total	All three search strings	17,400 (aggregate estimate from Google Scholar)

Inclusion and Exclusion Criteria

The criteria in Table 3 are those that were used when the selection was carried out; they were not added to or changed during the revision stage. The initial search used the Google Scholar filters for publication year 2020–2025 and English language. Screening was then performed on the basis of the relevance of the title and topic to the Scrum Framework. Eligibility was assessed by examining abstracts and full texts for their relation to the Scrum Framework, their relevance to software development, and the availability of the full text. Articles that did not meet the inclusion criteria were not carried forward to the next stage. The relevance and quality assessments carried out after selection (Sections 3.7 and 4.3) are retrospective and were not used to exclude studies.

Table 3. Inclusion and exclusion criteria

Code	Criterion	Stage of application
I1	Published in 2020–2025	Identification (Google Scholar filter)
I2	Written in English	Identification (Google Scholar filter)
I3	Addresses the Scrum Framework	Screening (title and topic) and eligibility
I4	Related to software development	Eligibility
I5	Full text available	Eligibility
E1	Published before 2020	Screening
E2	Does not address the Scrum Framework	Screening and eligibility
E3	Not related to software development	Eligibility
E4	Duplicate	Before or during screening; number not recorded

Selection Process

The selection followed the flow shown in Figure 1. In the identification phase, Google Scholar reported 17,400 results for the three search strings with the publication year filter of 2020–2025 and the English language filter. In the screening phase, records were filtered by the relevance of their titles and topics to the Scrum Framework, and 250 records were carried forward. These 250 records were assessed for eligibility by examining abstracts and full texts against the criteria in Table 3, and 40 articles were judged eligible. Subsequently, 13 of the 40 articles were designated as the final studies to be synthesized. However, the number of exclusions and the reasons for them at each stage, as well as the criteria for designating the 13 final studies, were not fully documented.

Several items of selection information were not documented: the number of duplicates removed; the number of records actually examined in the early stage; the number of exclusions and the reasons for them at each stage; the number of reports that could not be retrieved; the criteria explaining the choice of 13 from the 40 articles; and the number of reviewers and whether the selection was carried out independently. The differences between stages (17,150; 210; and 27) are only arithmetic differences and are not reported as verified numbers of exclusions by reason.

Data Extraction and Synthesis

The extracted data included the authors, year, publication type, research design, context or object, main contribution, and findings related to the implementation, benefits, and challenges of Scrum. The initial extraction was performed on all 13 studies. At the revision stage, the extraction for six studies was verified against the full text { (Hema et al., 2020) ; (Hidayah et al., 2022) ; (Hron & Obwegeser, 2022) ; (Lee & Chen, 2023) ; (Pinto & Leme, 2024) (Verwijs & Russo, 2023) }. Three studies were verified against the abstract only { (Garcia et al., 2022) ; (Tavares et al., 2021) ; (Zayat & Senvar, 2020) }, and four studies relied on the initial extraction { (Gupta et al., 2022) ; (Hamdulay, 2023) ; (Irfansyah et al., 2025) ; (van der Poll et al., 2023) }. The basis of verification for each study is shown in Table 6.

The synthesis was narrative-thematic across six themes (Table 1), because the heterogeneous study designs made a meta-analysis impossible. For each theme, the synthesis was carried out in three steps: identifying convergent findings, identifying differences or tensions among findings, and interpreting their possible causes, such as differences in context, scale, and type of evidence, while taking the quality assessment results into account. Two secondary studies ((Garcia et al., 2022) ; *(Hron & Obwegeser, 2022)*) were each counted as one study, but their overlap with other primary studies could not be examined.

Retrospective Quality Assesment

The initial study did not report a quality assessment. At the revision stage, a quality assessment was added as a retrospective evaluation of the studies whose full texts could be opened and read. Six of the 13 studies met this requirement. The other seven studies were not assessed and were not scored because their full texts could not be verified at the revision stage. The instruments were chosen according to the study design (Table 4). Each item was

rated Yes (1), Partial (0.5), or No (0); items that were not relevant to the study design were removed from the denominator. Scores were expressed as percentages and categorized as High (≥ 80%), Moderate (60–79%), or Low (< 60%); these thresholds were set by the researchers and were not validated. The assessment was carried out by a single assessor without a second independent assessment, so inter-rater reliability cannot be reported. The scores were not used to exclude studies; their function was to weigh the strength of evidence in the synthesis. The type of publication source (journal or proceedings) was recorded descriptively and was not scored.

For conceptual studies or studies that propose an approach, Instrument C was developed and adapted by the researchers on the basis of evaluation characteristics relevant to this type of study. Therefore, this instrument is exploratory and has not undergone external validation.

Table 4. Quality assessment instruments by study design

Code	Study design	Instrument and items	Studies assessed
A	Primary empirical studies and implementation reports	11 items (Dybå & Dingsøyr, 2008): research-based; clear aims; adequate context; appropriate design; appropriate sampling strategy; control group; data collection; analysis; researcher–participant relationship; clear findings; value for research/practice	Hidayah et al. (2022); Pinto & Leme (2024); Verwijs & Russo (2023)
B	Secondary studies	4 DARE criteria items (Kitchenham et al., 2010): inclusion/exclusion criteria; adequacy of the search; quality assessment of primary studies; adequacy of information on primary studies	Hron & Obwegeser (2022)
C	Conceptual studies or approach proposals	7 items adapted by the researchers: problem/aim; literature foundation; details of the approach; evaluation/validation; discussion of limitations; support for conclusions; contribution	Hema et al. (2020); Lee & Chen (2023)

RESULTS

Study Selection Results

Figure 1 presents the selection flow in the PRISMA 2020 format, and Table 5 details the numbers and their documentation status.

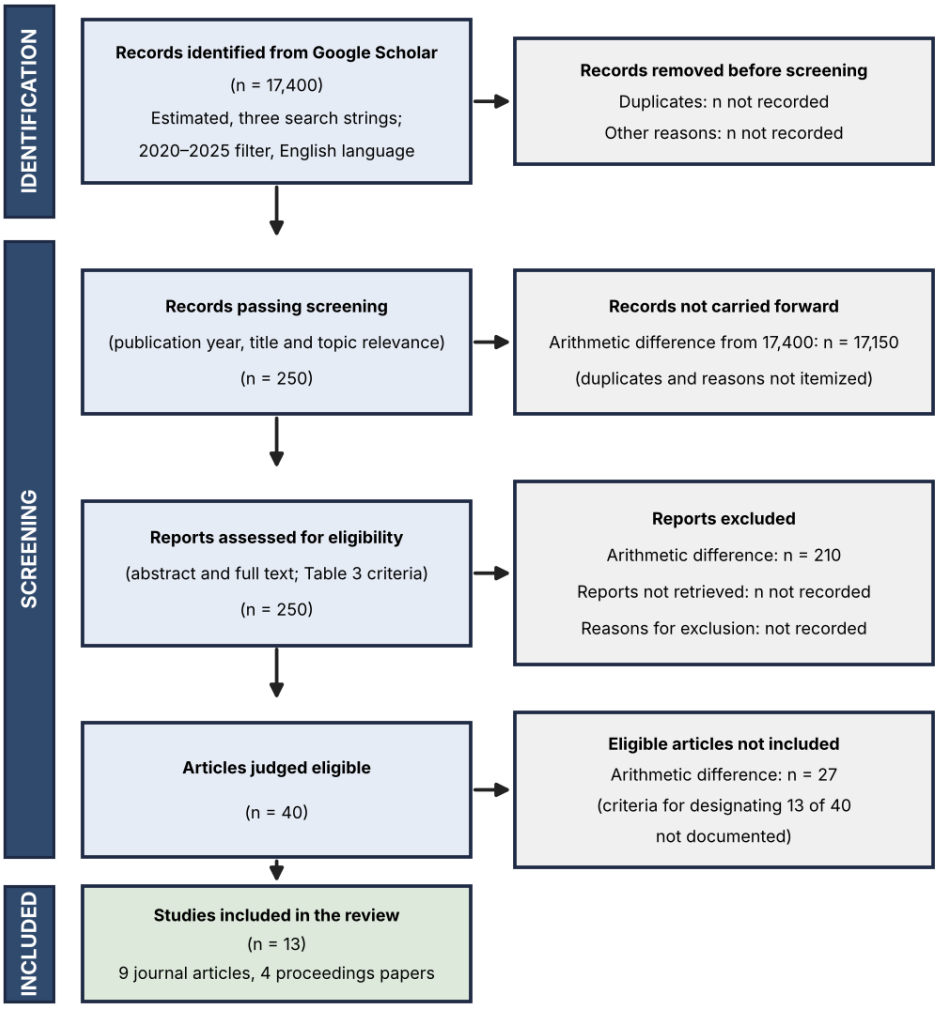


Figure 1. Study selection flow diagram (PRISMA 2020)

Table 5. Number of records at each selection stage and their documentation status

Stage	Number	Documentation status
Identification: results reported by Google Scholar	17,400	Aggregate estimate for the three search strings; number per string not itemized
Records removed before screening (duplicates; other reasons)	Not recorded	Not documented
Screening: records carried forward after filtering by title and topic relevance to the Scrum Framework	250	Number directly examined and number of exclusions not recorded; arithmetic difference from 17,400 is 17,150
Eligibility: reports assessed (abstract and full text)	250	Reports not retrieved: not recorded
Eligibility: articles judged eligible	40	Arithmetic difference of 210 from 250; reasons for exclusion not recorded
Included in the synthesis	13	Arithmetic difference of 27 from 40; designation criteria not documented

The figures in Figure 1, Table 5, and Section 3.5 are consistent. The decrease from 17,400 to 13 indicates progressive filtering, but because the reasons for exclusion were not documented, this study cannot assess the accuracy of that filtering.

Characteristics of the Included Studies

Table 6 summarizes the characteristics of the 13 studies. Nine studies are journal articles and four are conference proceedings papers (Hema et al., 2020; Gupta et al., 2022; van der Poll et al., 2023; Irfansyah et al., 2025). By year of publication, there are two studies from 2020, one from 2021, four from 2022, four from 2023, one from 2024, and one from 2025.

By design, there are two secondary studies (Garcia et al., 2022; Hron & Obwegeser, 2022); three primary empirical studies based on a survey, a case study, or theory testing (Hamdulay, 2023; Pinto & Leme, 2024; Verwijs & Russo, 2023); one implementation report (Hidayah et al., 2022); three studies that propose an approach or make a comparison with limited empirical evaluation (Lee & Chen, 2023; Tavares et al., 2021; Zayat & Senvar, 2020); and four studies that are conceptual or whose design could not be identified (Hema et al., 2020; Gupta et al., 2022; van der Poll et al., 2023; Irfansyah et al., 2025). Direct empirical evidence on how Scrum is used in practice is therefore relatively limited.

Table 6. Characteristics of the included studies

No	Study	Type	Design	Context/object	Main contribution	Verif.
1	(Hamdulay, 2023)	Journal	Survey (initial extraction)	IT organizations (practitioners)	Scrum, Agile, and Kanban are widely adopted and judged beneficial by most respondents without sacrificing functionality, time, budget, or quality	E

No	Study	Type	Design	Context/object	Main contribution	Verif.
2	(Zayat & Senvar, 2020)	Journal	Systematic Scrum–Kanban comparison with a focus group survey	Agile software development; project characteristics	Scrum suits new or complex projects with regular customer involvement; Kanban suits stable, continuous workflows	A
3	(Irfansyah et al., 2025)	Proceedings	Not identified in the initial extraction	Modern software development projects; large organizations	Emphasizes contextual adaptation; large organizations need resource investment and adaptation strategies	E
4	(Hidayah et al., 2022)	Journal	Implementation report on a single project	Web-based digital invitation platform (Invitin); team of 4	Describes the application of roles, backlogs, sprints, and the Definition of Done; the authors report clearer task planning and project monitoring (Trello) benefits without quantitative measurement	T
5	(Hema et al., 2020)	Proceedings	Conceptual/descriptive review with a proposed model	Software development in general	Describes Scrum roles, artifacts, and events; compares Agile and waterfall through a success-rate chart whose data source is not explained	T
6	(van der Poll et al., 2023)	Proceedings	Conceptual proposal (initial extraction)	Critical domain (banking)	Adds a formal method (Z notation) to the sprint cycle to improve artifact reliability; steep learning curve	E
7	(Garcia et al., 2022)	Journal	Systematic mapping (secondary study) with a practitioner evaluation of a feature model (TAM)	Tailoring of Scrum characteristics in Agile projects	Maps the tailoring of Scrum characteristics and represents them as a feature model evaluated by practitioners	A

No	Study	Type	Design	Context/object	Main contribution	Verif.
8	(Gupta et al., 2022)	Proceedings	Comparative study (initial extraction)	Software development in general	Scrum provides clearer structure through roles and ceremonies; the effectiveness of team communication determines success	E
9	(Tavares et al., 2021)	Journal	Tool development and experiment with Agile experts	Risk management in Agile software projects	Agile methods provide no dedicated risk management activities; the proposed tool improves the effectiveness of risk response planning	A
10	(Hron & Obwegeser, 2022)	Journal	Systematic review (925 studies; Scopus, Web of Science, IEEE Xplore)	Scrum modifications in various contexts, including outside software	Synthesizes nine goals and seven strategies for modifying Scrum; most studies are single case studies with limited theoretical grounding	T
11	(Verwijs & Russo, 2023)	Journal	Mixed methods: 13 exploratory case studies and a cross-sectional survey (CB-SEM)	Scrum teams; 4,940 respondents from 1,978 teams	Five high-level and 13 low-level factors explain team effectiveness (stakeholder satisfaction R^2 = 58%; team morale R^2 = 35%)	T
12	(Pinto & Leme, 2024)	Journal	Qualitative exploratory case study (questionnaire, document analysis, observation)	Third-party software factory serving public institutions; 10 respondents	Partial Scrum adoption; dual roles, reallocation of team members, and misalignment between the Product Owner and stakeholders were linked to low satisfaction with deliverables	T

No	Study	Type	Design	Context/object	Main contribution	Verif.
13	(Lee & Chen, 2023)	Journal	Hybrid approach proposal with a demonstrative case study and interview/questionnaire validation	Integration of Scrum and SPLE; blockchain-based online store; 5 experts and 44 practitioners	SPLE-Scrum supports component reusability; product management, project requirements, and product architecture factors are significantly related to perceived project success	T

Study Relevance and Quality

Relevance to the topic was reassessed retrospectively. Nine studies take Scrum as their main object (direct relevance). Four studies discuss Scrum together with other approaches or as part of Agile in general (partial relevance): Hamdulay (2023), Zayat and Senvar (2020), Gupta et al. (2022), and Tavares et al. (2021), whose abstracts discuss Agile methods in general. All studies were still included because this assessment was not used for exclusion.

Table 7. Relevance and results of the retrospective quality assessment

No	Study	Scrum focus	Instrument	Score	%	Category
1	Hamdulay (2023)	Partial	–	Not assessed	–	–
2	Zayat & Senvar (2020)	Partial	–	Not assessed	–	–
3	Irfansyah et al. (2025)	Direct	–	Not assessed	–	–
4	Hidayah et al. (2022)	Direct	A	3.5/10	35%	Low
5	Hema et al. (2020)	Direct	C	2.5/7	36%	Low
6	van der Poll et al. (2023)	Direct	–	Not assessed	–	–
7	Garcia et al. (2022)	Direct	–	Not assessed	–	–
8	Gupta et al. (2022)	Partial	–	Not assessed	–	–
9	Tavares et al. (2021)	Partial	–	Not assessed	–	–
10	Hron & Obwegeser (2022)	Direct	B	3/4	75%	Moderate

No	Study	Scrum focus	Instrument	Score	%	Category
11	Verwijs & Russo (2023)	Direct	A	9/10	90%	High
12	Pinto & Leme (2024)	Direct	A	7.5/10	75%	Moderate
13	Lee & Chen (2023)	Direct	C	6/7	86%	High

Of the six studies assessed, two were rated high (Verwijs & Russo, 2023; Lee & Chen, 2023), two moderate (Pinto & Leme, 2024; Hron & Obwegeser, 2022), and two low (Hidayah et al., 2022; Hema et al., 2020). The highest scores were achieved by studies with large-scale empirical data and explicit analysis. The two low-scoring studies describe the application or benefits of Scrum without traceable data collection or analysis. The scores were not used to exclude studies but were taken into account when weighing evidence in Section 5.

Mapping of Studies to Synthesis Themes

Table 8 maps the studies to the synthesis themes. The symbol ● indicates that the theme is a main finding or description of the study; ○ indicates indirect support, that is, an inference from the findings, a study's reference to the findings of others, or a study object that is not standard Scrum.

Table 8. Mapping of studies to synthesis themes

No	Study	T1	T2	T3	T4	T5	T6	Verif.
1	Hamdulay (2023)	●			●		●	E
2	Zayat & Senvar (2020)				●		●	A
3	Irfansyah et al. (2025)	●				●		E
4	Hidayah et al. (2022)	●	●	●	●			T
5	Hema et al. (2020)		●	●	●			T
6	van der Poll et al. (2023)			●		●		E
7	Garcia et al. (2022)	●		●				A
8	Gupta et al. (2022)					●	●	E
9	Tavares et al. (2021)					●		A
10	Hron & Obwegeser (2022)	●		●		○		T
11	Verwijs & Russo (2023)		●		●	○		T
12	Pinto & Leme (2024)	●	●	●		●		T
13	Lee & Chen (2023)	●		●	○	○		T
	Total (●)	7	4	7	5	5	3	
	Total (● and ○)	7	4	7	6	8	3	

DISCUSSION

The discussion is organized by research question: RQ1 covers themes T1–T3, RQ2 covers T4, and RQ3 covers T5, while T6 is discussed as a cross-question comparison. Each

attribution is read together with the quality of the study (Table 7) and its basis of verification (Table 6).

RQ1: Scrum Implementation in Software Development

Implementation context and patterns (T1)

The synthesized studies place Scrum in a wide range of contexts: IT organizations in general (Hamdulay, 2023), modern software projects in large organizations (Irfansyah et al., 2025), small-scale web projects (Hidayah et al., 2022), and software factories serving public institutions (Pinto & Leme, 2024). The common ground is the organization of work in iterative sprints. Hidayah et al. (2022) ran it with a team of four, whereas Pinto and Leme (2024) found teams of five to nine people with one Scrum Master and one Product Owner, in line with the guide.

Three patterns can be distinguished. The first is internal adjustment, that is, changing the elements of Scrum itself: Garcia et al. (2022) map the tailoring of Scrum characteristics and represent it as a feature model, while Hron and Obwegeser (2022) synthesize 925 studies and identify nine goals of modification (among them performance, specific contexts, combination with other frameworks, and distributed development) and seven modification strategies. The second pattern is the addition of other methods, such as Software Product Line Engineering (Lee & Chen, 2023) and formal methods (van der Poll et al., 2023). Adjustment changes how the framework is run, whereas addition extends its purpose, for example toward reusability or reliability.

The third pattern is unplanned deviation. In the case study by Pinto and Leme (2024), adoption was partial: roles were combined, team members were reassigned in the middle of a sprint, and priority changes entered without going through the Product Owner; the authors link this to outcomes that did not meet stakeholder expectations. The difference from the first pattern is important. Hron and Obwegeser (2022) included only studies that stated a reason for modification, so what they synthesized is deliberate modification, whereas unintentional deviation is better read as a failure to apply the framework. Hidayah et al. (2022) report roles and artifacts applied according to the guide in one small project, while the secondary studies show that modification is common. The two need not be read as contradictory because their units of analysis differ (a single project versus an aggregation of many practices). This explanation is hypothetical and was not tested in this study.

Roles in the Scrum Team (T2)

Hema et al. (2020) and Hidayah et al. (2022) agree that the role structure consists of the Product Owner, the Scrum Master, and the development team. Hema et al. describe the Product Owner as the owner or customer who manages the product backlog and the Scrum Master as the party who safeguards the Scrum process. Hidayah et al. divide a team of four into one Product Owner who analyzes user and business needs, one Scrum Master, and two developers (front-end and back-end), and stress that the three roles must not be omitted or combined.

Pinto and Leme (2024) examine exactly what was violated. Two Scrum Masters held other functions, one of them as Product Owner, and one Product Owner also served as Scrum Master; the authors interpret these dual roles as a burden that increases the likelihood of

neglected responsibilities. They also found a perception gap: the team's self-organization score (on a 0–10 scale) averaged 9.00 according to the Product Owner, 6.67 according to the Scrum Master, and 4.80 according to stakeholders, and stakeholders rated the team's understanding of the product and the business lower than the Product Owner did. The statements of Hidayah et al. and the findings of Pinto and Leme thus point in the same direction: clear and separate roles are considered important.

Verwijs and Russo (2023) complement this from another angle. The model, validated with 4,940 respondents from 1,978 teams, places stakeholder concern (collaboration with stakeholders, shared goals, the quality of the Sprint Review, and a focus on value), team autonomy, and management support as the high-level factors of team effectiveness. These constructs are not identical to the findings of Pinto and Leme, but both lead to the conclusion that a clear role structure is not enough without alignment with stakeholders and support from the organizational context. This comparison brings together a small case study and a large-scale validation study: the direction of the findings is consistent, but their magnitudes cannot be compared. None of the extracted studies explicitly compares alternative role configurations.

Artifacts, sprints, and events (T3)

Hema et al. (2020) and Hidayah et al. (2022) describe the artifacts and the sprint process as core elements that maintain transparency and regularity. Hidayah et al. show a concrete application: 16 user stories, three sprints with separate Product Backlog items, a Sprint Backlog board on Trello, a Definition of Done covering testing, documentation, and DevOps practices, and a Daily Scrum with three standard questions. However, the sprint duration is reported inconsistently within the same article (four weeks in the Product Backlog description and two weeks in Table 2), so the implementation details of this study should be read with caution.

Garcia et al. (2022) and Hron and Obwegeser (2022) show that in practice these elements are often adjusted, while Lee and Chen (2023) add a product management meeting and a Pre-Sprint phase for domain and application requirements engineering before Sprint Planning, and van der Poll et al. (2023) propose adding the formal Z notation to the sprint cycle (initial extraction). Pinto and Leme (2024) show the deviation side: stakeholders took part in Sprint Planning, the Daily Scrum, and the Sprint Retrospective, which according to the guide are intended for the team; sprints were extended beyond their time limits; and the status of backlog items in TFS did not reflect actual conditions, so inspection could not be carried out reliably.

These three groups of studies form a spectrum: standard description (Hema et al.; Hidayah et al.), adjustment and addition (Garcia et al.; Hron & Obwegeser; Lee & Chen; van der Poll et al.), and deviation (Pinto & Leme). A relationship between domain risk and the formality of artifacts appears to be a possible pattern: the addition of formal notation is proposed for critical domains, and Tavares et al. (2021) add risk management practices because Agile methods do not provide them. This added control carries a cost in the form of a steep learning curve (van der Poll et al., 2023), which runs counter to the goal of speed in Scrum. Because

this pattern rests on few studies from different domains and is partly verified only through the initial extraction, it is put forward as a hypothesis.

Answer to RQ1

Scrum implementation rests on a relatively stable iterative core (roles, artifacts, and sprints), with variation along three paths: internal adjustment, the addition of other methods, and unplanned deviation. The quality of application appears to be as important as its type: the study with the most detailed implementation description (Hidayah et al., 2022) is of low quality and does not measure outcomes, whereas the most concrete problems in practice are shown by a small case study (Pinto & Leme, 2024). Patterns by sector cannot yet be generalized because each specific context, such as the public sector and banking, is represented by only one study.

RQ2: Benefits of Scrum Implementation

The reported benefits include flexibility in responding to changing requirements, transparency and progress monitoring, collaboration with customers or stakeholders, productivity and efficiency, and stakeholder satisfaction and team morale. The degree of certainty differs among the studies.

Hamdulay (2023) reports that most respondents considered Scrum, Agile, and Kanban beneficial without sacrificing functionality, time, budget, or quality, but because the three approaches are combined, the specific contribution of Scrum cannot be separated (initial extraction). Zayat and Senvar (2020) place the advantages of Scrum under particular conditions, namely new or complex projects with regular customer involvement. Verwijs and Russo (2023) measured team effectiveness as stakeholder satisfaction and team morale and found that the factors in their model explain 58% of the variance in stakeholder satisfaction and 35% of the variance in team morale; among these factors, frequent releases (responsiveness) are positively related to effectiveness ($\beta = 0.402$).

Hidayah et al. (2022) and Hema et al. (2020) report broader benefits, but both have low quality scores (35% and 36%). Hidayah et al. state that Scrum increases team productivity but present no measurement and note the need for further testing to measure the success of Scrum. Hema et al. present a chart of the success rates of Agile projects compared with waterfall without explaining the source, sample, or method. Lee and Chen (2023) found that the product management, project requirements, and product architecture factors in the hybrid SPLE-Scrum approach are significantly related to project success according to 44 practitioners, but the proportion of variance explained is low to moderate (R^2 0.18–0.29) and the object is a hybrid approach, not standard Scrum.

There is a difference in tone: low-quality sources and general surveys tend to report broad benefits, whereas the more rigorous study (Verwijs & Russo, 2023) and the comparative study (Zayat & Senvar, 2020) stress the conditions, namely project conditions, team autonomy, and management support. The measures of benefit are also perception-based or narrative rather than objective measurements such as completion time or defect counts. Verwijs and Russo acknowledge the possibility of a halo effect because effectiveness was measured by self-report, and cross-sectional data do not permit causal conclusions. The

opposite side is shown by Pinto and Leme (2024): stakeholder satisfaction with deliverables was low (average 4.60 on a 0–10 scale), and the authors attribute this to partial adoption rather than to the framework itself, without testing a causal relationship. The source of the studies reporting benefits also deserves attention: Hron and Obwegeser (2022) note that many publications in the literature are written from a pro-Agile perspective, so the sample tends to lean toward positive results.

Answer to RQ2

The benefits of Scrum are reported in a consistent direction, but they are conditional and mostly based on perceptions or narrative claims. Only five studies directly support this theme (Table 8), two of which are of low quality, so the magnitude of the benefits cannot yet be concluded.

RQ3: Challenges of Scrum Implementation

The reported challenges can be grouped into three levels. **At the process level**, Tavares et al. (2021) state that Agile methods provide no specific activities for risk management. Lee and Chen (2023), citing earlier literature, mention difficulties in requirements identification and initial planning, insufficient attention to design and requirements engineering, and weak document traceability and configuration management. Hron and Obwegeser (2022) note that Scrum does not explicitly cover security, that the role of a technical architect is not defined, and that non-functional requirements are difficult to handle; this reads as a reason for modification rather than as a measurement of challenges.

At the team level, Gupta et al. (2022) highlight the importance of communication effectiveness (initial extraction), and van der Poll et al. (2023) note a steep learning curve when formal methods are added. Pinto and Leme (2024) provide the most concrete evidence: dual roles, reallocation of team members in the middle of a sprint, and difficulty understanding user stories (average difficulty score of 8.67 according to the Scrum Master and 7.80 according to stakeholders, compared with 2.00 according to the Product Owner, on a 0–10 scale). At the organizational level, Irfansyah et al. (2025) mention the need for investment and adaptation strategies in large organizations (initial extraction). Hron and Obwegeser (2022) show that Scrum assumes the availability of customers for frequent interaction, which is hard to meet in practice. Pinto and Leme report that priority changes were often communicated to managers or the Scrum Master rather than to the Product Owner, so the Product Owner's authority was bypassed, and the authors stress the need to change the sector's customary hierarchical paradigm in public organizations.

The links among the levels become visible when the findings of Pinto and Leme are read together with the model of Verwijs and Russo (2023). In that model, management support is positively related to team autonomy ($\beta = 0.427$), stakeholder concern ($\beta = 0.255$), and continuous improvement ($\beta = 0.168$), while team autonomy is strongly related to continuous improvement ($\beta = 0.655$). That weakness in these factors becomes an obstacle is our inference, not a direct finding of Verwijs and Russo. Priority changes that bypass the Product Owner in the study of Pinto and Leme are consistent with this inference, but the consistency is interpretive.

In the studies whose full texts were verified, two kinds of challenges can be distinguished: limitations of the framework in particular contexts (Hron & Obwegeser, 2022; Lee & Chen, 2023) and failures of application in practice (Pinto & Leme, 2024). The evidence is uneven: risk management is supported by only one study that is not specific to Scrum and was verified only at the abstract level (Tavares et al., 2021), whereas team-level and organizational-level challenges are supported by several studies with different designs, some of them from the initial extraction.

Answer to RQ3

The challenges of Scrum are spread across the process, team, and organizational levels and are interrelated. In the verified studies, the failure to apply practices with discipline (dual roles, mid-sprint changes, and outdated artifacts) stands out in the context of public organizations, whereas limitations of the framework appear in the need for risk management, requirements engineering, and security.

Comparison of Scrum, Agile, and Kanban (T6)

Three studies support this theme. Zayat and Senvar (2020) compare Scrum and Kanban according to project characteristics and place Scrum in new or complex projects with regular customer involvement, and Kanban in stable workflows. Gupta et al. (2022) and Hamdulay (2023) compare Scrum with other approaches, but both were verified only through the initial extraction. In practice, the two approaches are not always chosen exclusively: Hidayah et al. (2022) used an online kanban board (Trello) for the Sprint Backlog, and the team in the study of Pinto and Leme (2024) used Kanban alongside Scrum. These findings lead to the conclusion that the choice of method depends on project characteristics and may be a combination, but the comparative evidence remains thin.

Cross-Question Synthesis

Three factors recur across the research questions. First, project context determines the form of application and the suitability of the method (T1, T4, T6). Second, competence, separation of roles, and team autonomy are related to effectiveness and become a source of challenges when weak (T2, T4, T5). Third, organizational support and discipline in following the core practices are related to the quality of outcomes (T3, T4, T5). A summary of the synthesis for each theme is presented in Table 9.

Table 9. Summary of the comparative synthesis by theme

Theme	Convergent findings	Differences or tensions	Notes on strength of evidence
T1	The iterative core is retained across diverse contexts	Internal adjustment, addition of other methods, and unplanned deviation	Seven studies; specific contexts are each represented by one study

Theme	Convergent findings	Differences or tensions	Notes on strength of evidence
T2	The three roles are considered important and should not be combined	Dual roles found in practice; the Product Owner's perception differs from that of stakeholders	Four studies; one small case study and one large-scale study
T3	Artifacts and events maintain transparency	Standard descriptive studies versus additions and deviations	Seven studies; the detailed description is of low quality
T4	Benefits are consistent in direction, including flexibility and stakeholder satisfaction	Broad claims in low-quality sources versus conditional benefits in more rigorous studies	Five studies; perception-based measures
T5	Challenges at the process, team, and organizational levels	Limitations of the framework versus failures of application	Five direct studies and three indirect
T6	The choice of method depends on project characteristics	Exclusive versus combined use in practice	Three studies; two from the initial extraction

CONCLUSION

This study synthesized 13 studies (2020–2025) selected from Google Scholar using the PRISMA 2020 flow. For RQ1, Scrum implementation rests on a relatively stable iterative core and varies through internal adjustment, the addition of other methods, and unplanned deviation. For RQ2, the benefits are reported in a consistent direction, namely flexibility, transparency, collaboration, and stakeholder satisfaction, but they are conditional and mostly perception-based. For RQ3, the challenges are spread across the process, team, and organizational levels, with dual roles, mid-sprint changes, and outdated artifacts standing out in case studies of public organizations.

For practitioners, the findings of Pinto and Leme (2024) and Verwijs and Russo (2023) suggest paying attention to the separation of roles, the routing of priority changes through the Product Owner, the currency of artifacts, and management support for team autonomy. For researchers, studies are needed that measure outcomes objectively, examine the public sector and critical domains in more cases, and separate the contribution of Scrum from the other approaches used alongside it. Replicating this review with multiple databases, complete documentation of the selection, and independent assessors would strengthen the conclusions. The conclusions above are tentative given the limitations in Section 7.

LIMITATION

First, the search was conducted only on Google Scholar, which reports the number of results as an estimate, allows browsing of only part of the results, and changes over time; other databases were not covered, and exact replication is not guaranteed. Although the publication

year and language filters were applied through Google Scholar, technical documentation of the search date, the search settings, and the number of results for each search string is not available. This limits the exact repeatability of the search process. Second, the documentation of the selection is incomplete: the search date, the number of results per search string, duplicates, the number of exclusions and their reasons at each stage, the criteria for designating 13 of the 40 articles, and the number of reviewers were not recorded, so the PRISMA reporting is partial and the risk of selection bias cannot be assessed. Third, the review was limited to English-language publications from 2020 to 2025, and only 13 heterogeneous studies were synthesized, with only three primary empirical studies and one implementation report. Fourth, quality assessment and full-text verification could be carried out for only six studies; seven studies were not assessed, and some of their attributions rest on abstracts or the initial extraction.

The assessment was carried out by a single assessor, Instrument C is an adaptation by the researchers, the category thresholds were not validated, and scores from different instruments are not fully comparable. Fifth, four studies discuss Scrum together with other approaches or Agile in general, so the specific contribution of Scrum cannot be separated. Sixth, the two secondary studies may overlap with other primary studies. Seventh, the Agile literature tends to lean toward positive results (Hron & Obwegeser, 2022), so the reported benefits may be overly optimistic.

REFERENCES

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Garcia, L. A., OliveiraJr, E., & Morandini, M. (2022). Tailoring the Scrum framework for software development: Literature mapping and feature-based support. *Information and Software Technology*, *146*, 106814. https://doi.org/10.1016/j.infsof.2021.106814

Gupta, N., Sharma, H., Kumar, S., Kumar, A., & Kumar, R. (2022). A Comparative Study of Implementing Agile Methodology and Scrum Framework for Software Development. *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 1088–1092. https://doi.org/10.1109/SMART55829.2022.10047477

Hamdulay, N. A. (2023). Framework Study for Software Development via Scrum, Agile and Kanban. *The Online Journal of Distance Education and E-Learning*, *11*(2), 1388–1399.

Hema, V., Thota, S., Kumar, S. N., Padmaja, C., Krishna, C. B. R., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. *IOP Conference Series: Materials Science and Engineering*, *981*(2), 22060. https://doi.org/10.1088/1757-899X/981/2/022060

Hidayah, N. W., Sasmita, R. R., Mayangsari, M. K., Kusuma, O. G. W., Rante, H., & Fariza, A. (2022). Invitin Project: Scrum Framework Implementation in a Software Development Project Management. *INTEK: Jurnal Penelitian*, *9*(1), 58–65.

Hron, M., & Obwegeser, N. (2022). Why and How Is Scrum Being Adapted in Practice: A Systematic Review. *Journal of Systems and Software*, *183*, 111110. https://doi.org/10.1016/j.jss.2021.111110

Irfansyah, M. B., Waheed, B., Winarno, I., & Alimudin, A. (2025). Implementation of Scrum Framework in Modern Software Development Projects. *Proceedings of the 2025 International Electronics Symposium (IES)*, 875–880. https://doi.org/10.1109/IES67184.2025.11161121

Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, *52*(8), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

Lee, W.-T., & Chen, C.-H. (2023). Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering. *Electronics*, *12*(15), 3291. https://doi.org/10.3390/electronics12153291

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, n71. https://doi.org/10.1136/bmj.n71

Pinto, J. de S., & Leme, R. da S. (2024). Analysis of Project Management Principles with the Scrum Framework in Systems Development: A Case Study in a Public Organization. *Brazilian Journal of Operations \\& Production Management*, *21*(2), 1878. https://doi.org/10.14488/BJOPM.1878.2024

Tavares, B. G., Keil, M., da Silva, C. E., & de Souza, A. D. (2021). A Risk Management Tool for Agile Software Development. *Journal of Computer Information Systems*, *61*(6), 561–570. https://doi.org/10.1080/08874417.2020.1839813

van der Poll, J. A., Nyembe, F. H. N., & Lotriet, H. H. (2023). Formal Methods for an Agile Scrum Software Development Methodology. *Proceedings of the International Conference on Advanced Technologies*, *11*, 168–176. https://doi.org/10.58190/icat.2023.35

Verwijs, C., & Russo, D. (2023). A Theory of Scrum Team Effectiveness. *ACM Transactions on Software Engineering and Methodology*, *32*(3), 74. https://doi.org/10.1145/3571849

Zayat, W., & Senvar, O. (2020). Framework Study for Agile Software Development via Scrum and Kanban. *International Journal of Innovation and Technology Management*, *17*(4), 1–24. https://doi.org/10.1142/S0219877020300025