Journal of Software Engineering Research and Development, 2026, 14:25, doi: 10.5753/jserd.2026.7887

Scrumban in Hybrid Environments: An Empirical Investigation of Organizational Use and Impact in Development Teams

Maria Alcimar Costa Meireles [**Federal University of Amazonas and Instituto de Desenvolvimento Tecnológico** | ***maria.meireles@icomp.ufam.edu.br***]
Karen Soares [**Instituto de Desenvolvimento Tecnológico** | ***karen.soares@indt.org.br***]
Ádria Afonso [**Instituto de Desenvolvimento Tecnológico** | ***adria.afonso@indt.org.br***]
Lívia Lobão de Araújo [**Instituto de Desenvolvimento Tecnológico** | ***livia.araujo@indt.org.br***]
Luana Lobão [**Instituto de Desenvolvimento Tecnológico** | ***luana.lobao@indt.org.br***]
Gabriel Pinheiro [**Instituto de Desenvolvimento Tecnológico** | ***gabriel.campos@indt.org.br***]

Abstract

Context: Agile software development methodologies, such as Scrum, Kanban, and Scrumban, have gained prominence in Software Engineering by promoting greater flexibility and workflow visibility. Scrumban, in particular, combines elements of Scrum and Kanban and represents a promising alternative in contexts that demand continuous adaptation, such as hybrid projects integrating software and hardware development. **Goal:** This study aims to understand how Scrumban is adopted, adapted, used, and perceived in hybrid software and hardware development contexts, considering both the operational perspective of development teams and the management perspective. **Method:** We conducted a field study at a Brazilian Research, Development, and Innovation (RD&I) institute using two complementary sources of evidence. We collected survey responses from 15 members of multidisciplinary development teams to capture their experiences and perceptions of using Scrumban. We complemented these data with semi-structured interviews with three management professionals directly involved in its adoption and implementation. We qualitatively analyzed both data sources using Open Coding and Axial Coding procedures informed by Grounded Theory and integrated the findings according to the research questions. **Results:** The findings show that the adoption of Scrumban was motivated by the need to improve coordination, workflow visibility, and integration across software and hardware activities. Participants perceived benefits related to communication, cross-area coordination, project monitoring, risk identification, decision-making, and resource planning. However, the findings also reveal that Scrumban requires adaptations to accommodate hardware-specific characteristics, including longer development cycles, physical and supplier dependencies, different task granularities, and activities that are less amenable to frequent decomposition. Differences in experience with agile practices and resistance to process changes also emerged as adoption challenges. The integration of team and management perspectives further demonstrates that Scrumban provides complementary value across organizational levels, connecting operational coordination with broader project and organizational needs. **Conclusion:** Our findings suggest that Scrumban can support hybrid software and hardware development when organizations treat it as an adaptable coordination structure rather than as a uniform process. Its use benefits from shared communication and workflow visibility mechanisms while requiring sufficient flexibility to accommodate domain-specific characteristics. The study provides evidence on the use of Scrumban in an underexplored hybrid development context and offers practical insights for organizations coordinating multidisciplinary software and hardware teams.

Keywords: *Scrumban, Hybrid teams, Agile methodologies.*

1 Introduction

In recent years, software development teams have widely adopted agile development as a standard approach, driven by the values and principles of the Agile Manifesto (Fuentes-Del-Burgo et al., 2022). Teams have employed methodologies such as Scrum, Kanban, and, more recently, Scrumban, a hybrid approach that combines elements of both to improve project management, enable continuous delivery, and maximize team efficiency through workflow visualization (Fuentes-Del-Burgo et al., 2022). However, when teams adopt these practices improperly, they may face increased waste, longer development cycles, and reduced product quality (Alqudah and Razali, 2018).

Although initially designed for the software industry, agile development has gradually found applications in hardware projects, primarily due to its advantages in handling volatile, uncertain, complex, and ambiguous scenarios (Weichbroth, 2022). However, directly transferring agile principles to the hardware domain poses specific challenges, as complications hinder the full implementation of agility in this context (Weichbroth, 2022).

Projects that involve simultaneous software and hardware development face specific challenges when exposed to traditional agile approaches (Atzberger and Paetzold, 2019). Previous studies have deepened the understanding of these obstacles (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019), highlighting four main categories of challenges: (i) product-related physical limitations, such as stakeholders' reluctance to provide feedback on non-functional prototypes, the shortage of professionals with interdisciplinary knowledge, and the difficulty of coordinating deliveries of different natures; (ii) mindset, which involves individual and cultural resistance to agility, insufficient training, and conflicts between agile teams and traditional organizational structures; (iii) scaling, which refers to the difficulty of extending agility across multiple projects, including issues with knowledge transfer, inflexible organizational structures, and leadership resistance; and (iv) team distribution, including communication barriers, cultural differences, and the lack of standardized tools (Atzberger and Paetzold, 2019). These factors highlight the added complexity of incorporating agile practices into hybrid contexts beyond the traditional scope of software development.

Despite the growing adoption of agile practices beyond software development, applying agile approaches to hardware and integrated software and hardware development remains challenging due to differences in development cycles, physical constraints, dependencies, and organizational practices (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019; Weichbroth, 2022). Although previous studies have investigated agile adoption in hardware development and identified the need to adapt agile practices to this context (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019; Weichbroth, 2022), evidence specifically addressing Scrumban in settings where software and hardware activities coexist remains limited (Meireles et al., 2025). In particular, further investigation is needed to understand how hardware-specific characteristics interact with Scrumban practices, how the approach is adapted to coordinate work across software and hardware domains, and how its use is perceived across different organizational levels. Addressing these aspects is relevant because the characteristics of hybrid development can affect coordination, workflow visibility, predictability, and the synchronization of activities across domains (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019).

Given this scenario, Scrumban has emerged as a promising alternative for hybrid projects by offering greater flexibility, visibility, and traceability throughout the development process (Patil and Neve, 2018; Bhavsar et al., 2020). This article presents an empirical investigation of the adoption and use of Scrumban at a Brazilian Research and Development and Innovation (RD&I) institute, considering two complementary organizational perspectives. We collected evidence from (i) a survey with 15 members of multidisciplinary development teams, capturing the operational perspective of professionals involved in Scrumban-based work, and (ii) semi-structured interviews with three managers directly involved in the definition, implementation, and monitoring of Scrumban, capturing the management perspective. Rather than treating these perspectives as independent studies, we analyze them as complementary sources of evidence for understanding how Scrumban operates in a hybrid software and hardware development context.

Accordingly, this study aims to understand how Scrumban is adopted, adapted, and perceived in hybrid software–hardware development, considering both operational and management perspectives. We address the following research questions:

- **RQ1:** What factors motivate the adoption of Scrumban in hybrid software–hardware development contexts?
- **RQ2:** How is Scrumban used and adapted to support coordination between software and hardware development activities?
- **RQ3:** What benefits do team members and managers perceive from using Scrumban in hybrid development?
- **RQ4:** What challenges and limitations emerge from using Scrumban in hybrid software–hardware development contexts?

We analyzed the collected data qualitatively and found that using Scrumban increased the visibility of task progress and promoted alignment among technical teams through visual boards and communication ceremonies. From a management perspective, the findings also indicate that Scrumban supports decision-making and improves project monitoring by providing structured visibility into workflows and dependencies across areas. However, participants also reported limitations, including difficulties estimating hardware tasks, external dependencies, and the rigidity of physical development cycles. Managers additionally highlighted challenges related to cultural resistance, differences in mindset between software and hardware teams, and the additional effort required for implementation and continuous process adaptation. The participants emphasized the need to adapt agile practices to better address the particularities of hybrid contexts. Furthermore, the results show that Scrumban contributes not only to operational coordination but also to strategic aspects, such as risk identification, prioritization, and resource planning.

By integrating these perspectives, our findings indicate that Scrumban does not operate as a uniform process across software and hardware development. Instead, it can provide a shared coordination structure while accommodating domain-specific workflows, development cycles, and dependencies. The findings also reveal that workflow visibility has different implications across organizational levels: while team members associate visibility with task progress and cross-area alignment, managers use this visibility to support project monitoring, risk identification, prioritization, and decision-making. Furthermore, we distinguish challenges inherent to hybrid software and hardware development from tensions that emerge when Scrumban practices interact with these characteristics, such as the mismatch between short iterative cadences and longer hardware development cycles.

Therefore, this study contributes empirical evidence on how Scrumban can be operationalized and continuously adapted in hybrid software–hardware development. It also provides an integrated view of Scrumban from operational and management perspectives, highlighting how domain-specific constraints shape its use and the organizational value derived from its visibility and coordination mechanisms. Based on these findings, we derive lessons learned and practical implications for organizations seeking to adopt or adapt Scrumban in similar multidisciplinary development contexts.

The remainder of this paper is organized as follows: Section 2 presents the background and related work on agile methods, Scrumban, and hybrid software and hardware development; Section 3 describes the research method, including the organizational context, research questions, data collection, and analysis procedures; Section 4 presents the findings by integrating evidence from the team and management perspectives; Section 5 discusses the findings in relation to previous research and presents the lessons learned and implications for practice; Section 6 discusses the study limitations; and Section 7 presents the final considerations and directions for future work.

2 Background and Related Work

2.1 Scrum

Scrum is an agile framework widely used in project management, especially in software development. It enables teams to work collaboratively by organizing work into iterative cycles called sprints, in which teams deliver functional product increments (Reich and Reich, 2025; Minella, 2025). Scrum emphasizes team self-organization and continuous collaboration among members, prioritizing the frequent delivery of value to the customer. It is structured around roles, events, and artifacts that organize the workflow and ensure greater transparency in project management (Minella, 2025).

Scrum defines three main responsibilities (Schwaber and Sutherland, 2020):

- **Scrum Master**: fosters an environment that supports the effective application of Scrum practices.
- **Product Owner (PO)**: organizes and prioritizes the work in the Product Backlog, ensuring that it reflects stakeholders' needs and maximizes value generation.
- **Scrum Team**: transforms selected backlog items into a potentially releasable increment during a Sprint.

At the end of each Sprint, the team and stakeholders inspect the results and adapt the next steps. The Sprint represents a time-boxed iteration in which a functional increment is developed (Schwaber and Sutherland, 2020).

The framework includes a set of events designed to enable continuous inspection and adaptation (Schwaber and Sutherland, 2020):

- **Sprint Planning**: defines the work to be performed in a given Sprint.
- **Daily Scrum**: supports continuous monitoring of progress toward the Sprint Goal and allows adjustments to the Sprint Backlog.
- **Sprint Review**: focuses on inspecting the increment and collecting feedback from stakeholders.
- **Sprint Retrospective**: provides an opportunity to reflect on the process and identify improvements for future Sprints.
- **Sprint**: acts as a container for all these events.

Scrum defines three main artifacts (Schwaber and Sutherland, 2020):

- **Product Backlog**: aligned with the Product Goal.
- **Sprint Backlog**: aligned with the Sprint Goal.
- **Increment**: aligned with the Definition of Done.

These artifacts represent the work and value generated during development and are designed to maximize transparency. These commitments reinforce empiricism and align the Scrum Team with stakeholders (Schwaber and Sutherland, 2020).

2.2 Kanban

Kanban is a visual management tool that originated in the Toyota Production System and is widely used in lean manufacturing practices (Martins et al., 2025). It controls and optimizes workflow through visual signals, promoting greater efficiency and reducing waste (Martins et al., 2025). This system directly relates to the concept of pull production, in which activities are performed according to demand (Martins et al., 2025).

Kanban structures work by dividing larger tasks into smaller, executable activities that move through well-defined stages of the process (Lei et al., 2020). Teams represent these tasks as cards on a visual board, where their position indicates the progress achieved (Lei et al., 2020). In addition, teams can mark tasks as blocked when impediments must be resolved before work can continue (Lei et al., 2020). Figure 1 illustrates an example of a Kanban board, where activities and their current states remain visible to the team.

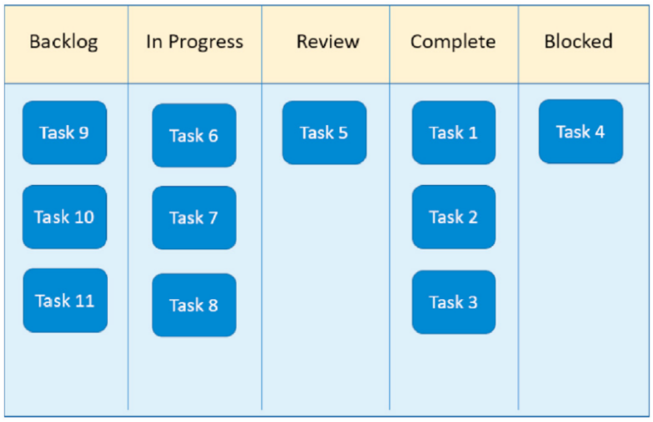

Figure 1. Example of a Kanban board adapted from Lei et al. (2020).

Kanban makes all required tasks and the progress of each task transparent to development team members and any other interested stakeholders (Lei et al., 2020). Overall, the Kanban structure brings clarity when dealing with complex problems. Domain experts can visualize how the team is addressing issues, including what has been completed, what is in progress, what still needs to be done, and what must be resolved before progress can continue (Lei et al., 2020).

Kanban was introduced into software development in 2004 (Gaborov et al., 2021). It is based on five principles: visualize the workflow, limit work in progress, measure and manage flow, make process policies explicit, and promote continuous improvement (Ahmad et al., 2016). In software development, Kanban enhances communication, collaboration, and integration among teams, resulting in faster delivery, increased productivity, and reduced costs. Additionally, its flexibility allows each organization to adapt its principles to its own needs, fostering continuous learning through practice (Gaborov et al., 2021).

2.3 Scrumban

Scrumban is characterized as a hybrid agile methodology that integrates practices from both Scrum and Kanban to promote greater flexibility and efficiency in project management (Alqudah and Razali, 2018). According to Alqudah and Razali (2018), this approach incorporates Scrum practices such as iterative planning, task prioritization, and continuous process improvement. At the same time, it adopts Kanban elements such as workflow visualization and Work in Progress (WIP) limits, which help enhance the flow and predictability of deliveries (Alqudah and Razali, 2018). Therefore, Scrumban prescribes multiple sprint cycles, defines executable tasks within each cycle, and monitors their progress using a Kanban board (Lei et al., 2020). Additionally, Albarqi and Qureshi (2018) emphasizes that Scrumban combines the benefits of both methodologies. It facilitates project management by adopting Scrum principles that promote team collaboration during fixed-length sprints and require cross-functional team members. It also incorporates Kanban mechanisms, such as WIP limits, workflow monitoring, and cycle time measurement, via visual boards.

Fuentes-Del-Burgo et al. (2022) argue that teams can understand Scrumban as a transitional method between Scrum and more advanced development models. By integrating Kanban elements into the Scrum context, Scrumban provides new perspectives and enhances teams' management capabilities. This integration allows teams to adapt more effectively to project-specific needs by combining the structured cadence of Scrum, such as sprints, daily meetings, planning, review, and retrospectives, with Kanban principles, including visual boards, WIP limits, and a focus on continuous improvement (Semerano and De Oliveira, 2024). By merging the strengths of both approaches, Scrumban aims to improve workflow management and support the constant delivery of value throughout the project lifecycle (Semerano and De Oliveira, 2024).

2.4 Related Work

Alqudah and Razali (2018) investigated how teams form the Scrumban method by selecting and combining practices from Scrum and Kanban. The authors identified the factors influencing this combination and provided guidelines to help agile teams adopt Scrumban effectively. They conducted the study in two phases. In the first phase, they reviewed the literature to identify practices, principles, and distinctions between Scrum and Kanban. In the second phase, they conducted semi-structured interviews with seven agile methodology experts from different countries. They analyzed the collected data qualitatively using content analysis in NVivo. The results revealed that the formation of Scrumban strongly depends on the context and the team's prior experience with Scrum and Kanban. The study identified several critical factors for selecting and combining practices, including method prescription, roles and responsibilities, adoption timing, team size, work batch size, requirements prioritization, feature size, lead time, cost, quality, and absence of technical practices. The findings indicated that Scrumban proved more effective than either Scrum or Kanban alone in reducing waste, improving quality, and saving time and costs by enabling flexible customization aligned with project characteristics.

Patil and Neve (2018) investigated the practical application of Scrumban to enhance productivity in software development. Their study aimed to demonstrate how combining practices from Scrum and Kanban could mitigate limitations observed in traditional methodologies while promoting greater agility, quality, and efficiency in development processes. To this end, the authors conducted an empirical approach in which development teams adopted Scrumban by removing the fixed sprint cycles of Scrum and incorporating Kanban practices such as WIP limits and just-in-time planning. The analysis relied on evidence collected during the implementation, including communication records, productivity metrics, and observations regarding process improvements. The main benefits identified included increased flexibility in handling changes, improved visualization of the workflow, reduced impediments, enhanced efficiency, and improved quality of the final product. The authors also noted that Scrumban supports kaizen (continuous improvement) and waste reduction, contributing to technical and business gains. The study reinforces Scrumban's suitability for teams that require greater adaptability without compromising visibility and control over the development process.

Patilla et al. (2023) explored how teams can adopt the Scrumban method to improve productivity in software development. The study followed a practical approach, implemented across multiple development teams in IT companies, and observed the impacts of Scrumban adoption in real-world environments. The methodology involved removing rigid aspects of Scrum, such as fixed sprints, and integrating Kanban practices, including WIP limitations, continuous prioritization, and workflow visualization. Field evidence was collected through direct observations of team performance and feedback from the professionals involved, including software engineers and module leaders. The results indicated that forming a Scrumban process depends on teams' experience with agile methods and the project's specific needs. The study identified several critical factors for combining practices: planning flexibility, adjustment of work batch sizes, delivery pace, team collaboration, and focus on continuous improvement. Scrumban led to improvements in lead time, process visibility, waste reduction, and delivery quality. Additionally, the methodology proved effective in adapting to change and delivering customer value, overcoming limitations observed in the isolated application of Scrum or Kanban.

studies specifically addressing Scrumban, previous research has investigated the adoption of agile approaches in hardware and combined software and hardware development. Atzberger and Paetzold (2019) examined the challenges of applying agile development to hardware and identified constraints related to physical products, organizational mindset, scaling, and team distribution. These findings indicate that characteristics of hardware development can complicate the direct transfer of agile practices originally established in software contexts.

Focusing specifically on projects involving both domains, Drutchas and Eppinger (2022) investigated organizations that worked simultaneously on hardware and software development through qualitative interviews. The authors identified challenges in applying agile approaches across hardware and software activities and documented adjustments to agile practices to support their use in hardware projects. Their findings reinforce the need to adapt agile practices to account for the different characteristics of hardware and software development rather than applying the same practices uniformly across both domains.

Rautakoura and Hämäläinen (2023) further investigated agile hardware development through a literature review and mapping study focused on System-on-Chip (SoC) development. The authors initially examined 730 papers and, after filtering, analyzed 25 studies in detail. Their results showed substantial heterogeneity in how agility is operationalized in hardware development: established agile software development methods were applied in only five of the analyzed studies, nine adopted ideas derived from agile hardware approaches, and eleven did not define a specific agile hardware development method. The authors also identified methodological gaps in the literature and highlighted the need for further research and more accurate reporting of agile development processes in hardware contexts.

Taken together, these studies reveal two complementary bodies of knowledge. Research on Scrumban has predominantly investigated its formation, adoption factors, and benefits in software development contexts (Alqudah and Razali, 2018; Patil and Neve, 2018; Patilla et al., 2023). In parallel, research on agile hardware and combined software–hardware development has demonstrated that physical constraints, differences in development characteristics, and cross-domain challenges require agile practices to be adapted rather than directly transferred from software development (Atzberger and Paetzold, 2019; Drutchas and Eppinger, 2022; Rautakoura and Hämäläinen, 2023). However, limited evidence exists on how Scrumban specifically is operationalized and adapted when software and hardware activities coexist, particularly regarding how its practices support coordination across domains and how its use is perceived at different organizational levels. Our study addresses this gap by investigating Scrumban in a real R&D environment from complementary team and management perspectives, examining its adoption and adaptation, its perceived benefits and challenges, and how characteristics of hybrid development shape its use.

3 Research Method

3.1 Research Design

We conducted a field study to investigate the adoption, adaptation, and use of Scrumban in a real Research, Development, and Innovation (R&D&I) environment. According to Stol and Fitzgerald (2020), field studies are conducted in natural settings that exist independently of the research initiative, making them suitable for investigating phenomena, events, people, teams, and systems in their actual contexts. This strategy enabled us to investigate how Scrumban was introduced and adapted in practice and how its use was perceived at different organizational levels.

The investigation combines two complementary sources of evidence: a survey with 15 members of multidisciplinary development teams and semi-structured interviews with three managers directly involved in the definition, implementation, and monitoring of Scrumban. The survey captures the operational perspective of professionals involved in the development activities, whereas the interviews provide a management perspective on the motivations for adoption, process implementation and adaptation, perceived benefits, and challenges. Rather than treating these data sources as independent studies, we use them as complementary evidence to investigate the same phenomenon within the organizational context.

The study is guided by the four research questions presented in Section 1, addressing the motivations for adopting Scrumban (RQ1), how the approach is used and adapted to coordinate software and hardware activities (RQ2), its perceived benefits (RQ3), and the challenges and limitations associated with its use in hybrid development (RQ4).

3.2 Organizational Context and Scrumban Adoption

The Instituto de Desenvolvimento Tecnológico is a nonprofit organization dedicated to technological innovation and the execution of Research, Development, and Innovation (R&D&I) projects for industry (Meireles et al., 2026, 2025). Its projects span various technological domains, including embedded systems, software applications, and integration with sensors, actuators, and physical devices. The multidisciplinary teams involved in these projects include software and hardware developers, automation engineers, mechanical designers, testing specialists, and Product Owners (POs).

Before introducing Scrumban, the teams followed different methodological approaches: software teams had already adopted agile methods, mainly Scrum, whereas hardware teams operated under more traditional planning cycles, characterized by sequential schedules, lower task granularity, and centralized documentation. This disparity hindered alignment across areas, reduced project visibility, and complicated joint delivery planning.

As the number of projects with hardware and software interdependence increased, the teams identified the need for a management approach that could support flexibility, predictability, and continuous collaboration. They adopted Scrumban as a viable alternative because it combines the structured rituals of Scrum with the continuous visualization and pull-based flow of Kanban.

Scrumban adoption at the institute began in 2022 and continued in subsequent projects. Between 2022 and 2026, Scrumban was used in 15 R&D&I projects. Of these, 11 (73.3%) involved both software and hardware development, while four (26.7%) involved software development only. Table 1 summarizes the distribution of these projects over the period. This continued use provides a broader organizational context for examining how Scrumban was incorporated into project practices beyond its initial adoption.

Table 1. Projects using Scrumban between 2022 and 2026

Year	Projects	SW only	SW + HW
2022	2	1	1
2023	2	1	1
2024	2	0	2
2025	4	1	3
2026	5	1	4
Total	**15**	**4**	**11**

The team survey captured the initial Scrumban adoption and adaptation experience from 2022 to mid-2025. After this period, the institute continued to use Scrumban in new projects, including projects initiated in the second half of 2025 and in 2026. Therefore, we distinguish the period covered by the initial investigation from the continued and expanded use of Scrumban within the organization.

3.3 Scrumban Implementation and Adaptation

Scrumban was not introduced as a fixed process. Its implementation involved an initial configuration followed by adaptations based on the characteristics of the multidisciplinary projects and the different development dynamics of software and hardware. The main steps of this process are described below.

1. **Selection of the Pilot Project**: The team implemented the initial phase of Scrumban in a project that already had a defined Work Plan (WP) with a clear scope and schedule. This foundation allowed them to begin the experience with a degree of predictability, which was essential for the first adoption cycle.
2. **Formation of a Multidisciplinary Team**: The team included software developers familiar with Scrum and hardware engineers without previous experience with agile methods. This configuration enabled the team to quickly identify friction points between the two domains and propose practical adjustments.
3. **Initial Assessment Using Planning Tools**: The team applied the CSD Matrix (Certainties, Suppositions, and Doubts) and the History Map to identify priority features, risks, and knowledge gaps. These tools helped align expectations between the team and management and supported the creation of an initial backlog.
4. **Definition of Rituals and Initial Cadence**: The team established a biweekly sprint cadence and adopted standard Scrum rituals: planning, daily meetings, reviews, and retrospectives. These events structured progress tracking and aligned incremental deliveries, particularly in software.
5. **Customization of the Jira Board**: The team customized the Jira board with columns common to all areas (e.g., Development Stage, Test Stage, Review Stage, see Figure 2). However, they defined distinct internal flows for each task card type to reflect the unique characteristics of software and hardware. For instance, hardware tasks did not necessarily go through all stages, whereas software tasks did.
6. **Establishment of Weekly Integration Checkpoints**: In addition to the standard ceremonies, the team created a weekly meeting between representatives from the software and hardware areas. These checkpoints focused on discussing task dependencies, bottlenecks, and planning adjustments. This practice facilitated early conflict detection and dynamic coordination of activities.
7. **Continuous Process Adaptation**: Over time, the team iteratively adjusted the process based on the characteristics of each domain. For example, the initial use of short sprints proved unsuitable for the pace of some hardware activities, leading the team to adopt more flexible cycles for this domain. The team also improved the use of labels and filters in Jira to identify blocked tasks, dependencies, and activities requiring external validation, such as supplier deliveries.
8. **Promotion of a Collaborative Culture**: Alongside tools and ceremonies, the team consistently promoted a culture of collaboration. Engineers were encouraged to participate actively in meetings, understand the context of other areas, and contribute to joint solution-building. This effort sought to address resistance to the use of agile practices in hardware activities and foster greater interaction between technical areas.

This description was reconstructed from the participants' accounts and the authors' knowledge of the organizational context, and it provides contextual information for interpreting the findings presented in Section 4.

3.4 Participants and Data Collection

We collected data from two complementary groups of participants to capture operational and management perspectives on the adoption and use of Scrumban. The first group comprised 15 professionals from multidisciplinary development teams with experience using Scrumban on their projects. The second group comprised three management professionals who were directly involved in the adoption and implementation of Scrumban at the institute. By combining these perspectives, we sought to examine both how professionals experienced Scrumban in their development activities and how management perceived its adoption, adaptation, and organizational use.

3.4.1 Team Survey

We conducted the survey with 15 professionals from multidisciplinary teams at the Instituto de Desenvolvimento Tecnológico (INDT). These professionals worked in different areas, including software development (SW Dev), software development specialization (SW Specialist), mechanical engineering (Mechanical Dev), automation (Automation Dev), computer vision (Computer Vision Engineer), testing (Tester), and product ownership (PO). This diversity allowed us to capture perceptions from professionals working in different technical and managerial roles involved in the development process.

We used a survey comprising 10 questions to collect participants' perceptions of their experience with Scrumban, including perceived benefits, challenges, and suggestions regarding its use. The complete survey instrument is available

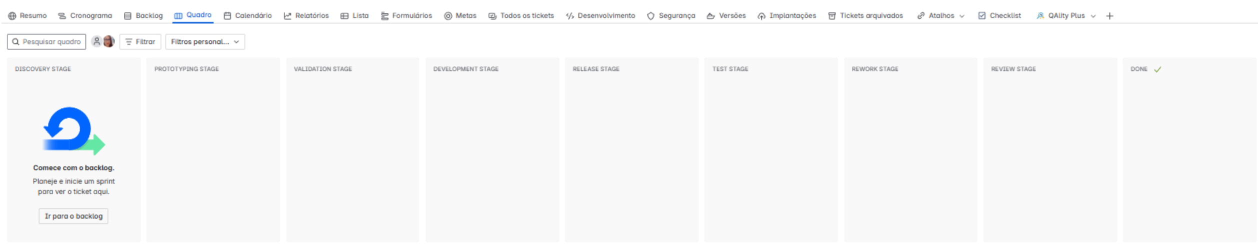

Figure 2. Customization of the Jira Board (in Portuguese)

in the supplementary material at Figshare. Table 2 presents the participants' roles, professional experience, experience using Scrumban, and the number of projects in which they had used the approach.

3.4.2 Management Interviews

To complement the operational perspective captured through the survey, we conducted semi-structured interviews with three professionals in management positions who were directly involved in the adoption and implementation of Scrumban at INDT. The participants comprised two managers and one director. Their involvement enabled us to examine, from a management perspective, the organizational motivations for adopting Scrumban, decisions regarding its implementation and adaptation, governance practices, and the challenges associated with its use. Table 3 presents the characterization of these participants.

We selected the interview participants for their direct involvement in the adoption and implementation of Scrumban and their knowledge of how the approach was incorporated into organizational practices. Their perspectives complemented those of the development professionals by providing information about organizational motivations, implementation decisions, process monitoring, and adaptations made throughout Scrumban adoption.

Two researchers participated in planning and reviewing the interview protocol, including the development of the interview script. Before the main data collection, we conducted a pilot interview to assess the clarity and comprehensibility of the questions and to adjust the protocol as needed.

The first author conducted all interviews via videoconference, scheduling them on different days to accommodate participants' availability. Before each interview, participants received and signed a consent form informing them about data confidentiality and the voluntary nature of their participation. We conducted and transcribed the interviews in Portuguese to preserve the participants' original meanings during the analysis.

The interview script comprised 30 questions addressing different aspects of Scrumban, including motivations for its adoption, implementation, and adaptation, governance practices, daily use and monitoring, perceived benefits, challenges, and adaptations made during its use. The complete interview script is available in the supplementary material at Figshare.

3.5 Data Analysis

We analyzed the survey and interview data according to the characteristics of each evidence source. We first analyzed each data source separately and subsequently examined their findings together according to the research questions. This strategy allowed us to preserve the perspectives provided by each participant group while identifying convergence and complementarity between operational and management perspectives.

3.5.1 Survey Analysis

We analyzed the closed-ended survey responses using descriptive statistics to summarize participants' perceptions of their experience with Scrumban. Regarding their overall experience with the approach, 20% of participants rated it very positive, 46.7% rated it positive, 20% rated it neutral, and 13.3% rated it negative. Figure 3 presents the distribution of these responses.

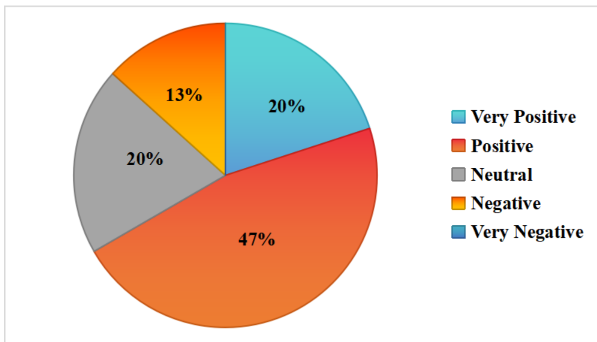

Figure 3. Overall experience with using Scrumban

We analyzed the responses to the open-ended survey questions using Grounded Theory (GT) coding procedures, following (Strauss and Corbin, 1998). The analysis involved Open Coding and Axial Coding. During Open Coding, we examined participants' responses and assigned codes to excerpts representing relevant concepts emerging from the data. We then compared the identified codes across responses and, during Axial Coding, examined their relationships, grouping related codes into broader categories. Throughout the analysis, we reviewed and refined the codes and categories to ensure that they consistently represented the participants' perceptions of the benefits, challenges, and adaptations associated with Scrumban.

Table 2. Characterization of Survey Participants

Part	Role	Experience in Role	Years using Scrumban	N. Projects using Scrumban
T1	Sw Dev	3 years	3 years	5
T2	PO	7 years	6 years	10
T3	Tester	15 years	10 years	15
T4	Mechanical Dev	5 years	1 year	3
T5	SW Dev	3 years	2 years	6
T6	SW Dev	6 years	6 years	6
T7	Automation Dev	5 years	2 years	5
T8	SW Specialist	10 years	7 years	7
T9	PO	10 years	6 years	6
T10	SW Specialist	17 years	7 years	4
T11	Computer Vision Engineer	6 years	1 year	2
T12	SW Specialist	10 years	7 years	10
T13	SW Dev	6 years	2 years	3
T14	Automation Dev	5 years	1 year 6 months	4
T15	Automation Dev	10 months	10 months	3

Table 3. Characterization of Interview Participants

Participant	Role	Responsibility
M1	Project Manager	Manages the technical and financial projects and oversees people through technical managers.
M2	General Project Manager	Manages the technical and financial aspects of projects and oversees people thought technical managers.
M3	Director of Innovation	Manages startup projects, oversees INDT Educacional, and handles scientific publications and intellectual property.

3.5.2 Interview Analysis

We analyzed the interview transcripts using Grounded Theory (GT) coding procedures, following Strauss and Corbin (1998). Consistent with the analysis of the open-ended survey responses, we applied Open Coding and Axial Coding to systematically identify and organize concepts emerging from the data. During Open Coding, we examined the transcripts and assigned codes to relevant excerpts from participants' accounts. We then used Axial Coding to examine relationships among the identified codes and organize them into broader categories that captured organizational aspects related to the adoption, adaptation, governance, benefits, and challenges of Scrumban.

Throughout the analysis, the researchers iteratively reviewed and discussed the emerging codes, categories, and their relationships. In addition to the first author, two researchers contributed to reviewing the analysis and discussing the interpretation of the findings. These discussions supported the refinement of the coding structure and helped ensure that the resulting categories adequately reflected the participants' accounts.

We used GT coding procedures as an analytical strategy rather than conducting a full Grounded Theory study aimed at theory generation. Therefore, we did not seek theoretical saturation. Instead, Open Coding and Axial Coding supported the systematic organization and interpretation of findings relevant to the research questions. We conducted the interviews, transcription, and analysis in Portuguese and translated excerpts into English only for reporting purposes.

For both data sources, the researchers continuously reviewed and discussed the emerging findings throughout the analysis. In addition to the first author, two researchers contributed to reviewing the analyses and discussing the interpretation of the results. These discussions supported the refinement of codes, categories, and relationships identified in the survey and interview data.

3.5.3 Integration of Evidence

After completing the analyses, we examined the findings from the team survey and management interviews together according to the research questions. We compared the perspectives to identify findings that converged across participant groups and findings that complemented one another. The survey primarily provided evidence of how professionals experienced Scrumban in their development activities, including perceived benefits and challenges, whereas the interviews offered additional explanations of organizational motivations, implementation decisions, process adaptations, governance practices, and management challenges.

We used this integration to develop a broader interpretation of Scrumban adoption and use within the organization rather than treating the survey and interviews as independent studies. When both sources addressed the same phenomenon, we considered their convergence or differences in their interpretations of the results. When an aspect emerged from a single source, we treated it as evidence specific to that source's perspective. This approach allowed us to distinguish shared perceptions from findings specific to either the operational or management perspective.

3.6 Research Quality

We adopted complementary procedures to strengthen the credibility of the qualitative findings. First, multiple researchers participated in reviewing and discussing the interview analysis, providing different perspectives on the data interpretation and helping refine the resulting codes and categories.

Second, following the interview analysis, we conducted a member-checking session with a participant directly involved in the adoption and use of Scrumban within the organization. During this session, we presented and discussed the findings with the participant to assess whether our interpretations adequately represented the investigated organizational context and to collect additional observations that could refine the analysis. We considered the participants' feedback when reviewing the findings.

Because the member-checking procedure involved one participant, we treated it as an additional credibility procedure rather than as confirmation of the findings by the complete participant group. This procedure provided an opportunity to examine the interpretations against the practical experience of a professional directly involved with the phenomenon under investigation.

4 Results

We report the results in line with the four research questions, integrating evidence from the two complementary data sources. The survey captures the operational perspective of multidisciplinary team members who use Scrumban in their daily activities, whereas the interviews provide the perspective of managers directly involved in its adoption, adaptation, and governance. Rather than analyzing these perspectives separately, we use them as complementary sources of evidence. When both sources address the same aspect, we examine their convergence or complementarity; when an aspect emerges from only one source, we report it as specific to that organizational perspective. To distinguish the two participant groups throughout the results, we use the identifiers T1-T15 for technical team members who participated in the survey and M1-M3 for management participants who participated in the interviews.

4.1 RQ1: Organizational Factors Motivating Scrumban Adoption

The first research question investigates the organizational factors that motivated the adoption of Scrumban in the hybrid software and hardware context. Evidence for this question comes primarily from the management interviews, as these participants were directly involved in decisions regarding the adoption and implementation of the approach. The analysis identified related motivations, including the coordination of multidisciplinary projects, flexibility for continuous adaptation, incremental value delivery, structuring the development process, integration between software and hardware activities, limitations associated with using Scrum or Kanban independently, and the need to accommodate different project and client contexts. Figure 4 summarizes these factors.

A central motivation was the **need to coordinate multidisciplinary work**. The projects spanned different technical areas, including software, artificial intelligence, mechanics, electrical engineering, and automation, necessitating an integrated view of activities, dependencies, and priorities. M1 explained: *"In the stages of project development and organization [...] these multidisciplinary projects require a holistic view of the project, an understanding of the development phases, and prioritization. Therefore, by recognizing these needs, we identified the need to implement methodologies that could provide this visibility for project coordination."* The need for coordination was therefore closely connected to the challenge of integrating work performed by areas with different development characteristics. M2 reinforced this point by emphasizing the importance of *"better understanding how one thing would interact with another, how the software would be ready to integrate with the hardware and vice versa,"* showing that the motivation involved both technical integration and coordination of the development process.

Flexibility was another important motivation. Because the projects involved technological uncertainty and changing demands, management sought an approach that could accommodate changes without losing control over project execution. M1 described this need as *"We identified this need [...] to have the ability to adjust and change direction quickly and in a controlled manner."* The diversity of projects reinforced this requirement. As M3 observed, *"we have very different types of projects here [...]"*, indicating that a rigid and uniform process would not adequately address all organizational contexts.

Participants also associated Scrumban adoption with the **need to organize work around incremental value delivery**. M1 stated that *"we needed to deliver value continuously to the customer in smaller increments,"* while M3 emphasized the importance of *"not spending time on things that the customer will not use."* This orientation was accompanied by a need for greater process structure. M1 referred to the *"need to create structures that allow us to establish development processes."* Together, these accounts indicate that management sought both flexibility and sufficient structure to organize, prioritize, and monitor multidisciplinary work.

Finally, participants reported that **Scrum and Kanban, when used independently, did not fully address these**

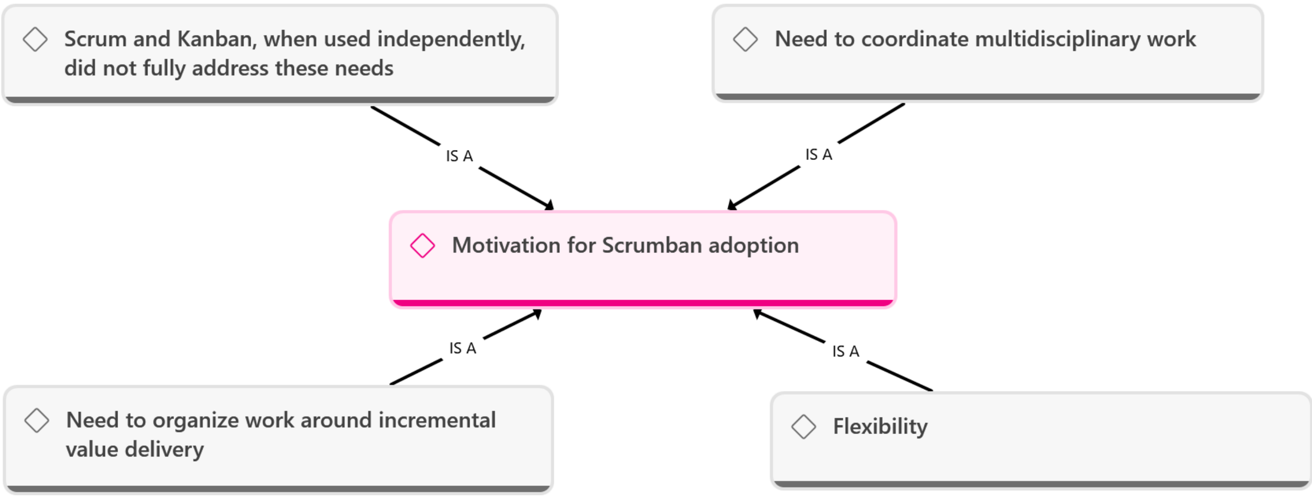

Figure 4. Organizational factors motivating Scrumban adoption

needs. M3 stated that *"Using only Kanban or only Scrum did not solve our problem,"* while M1 explained that *"we saw that the best path was to unify these two methodologies."* Thus, the motivation for adopting Scrumban emerged from the combination of coordination, integration, flexibility, incremental delivery, and process-structuring needs that neither approach was perceived to address adequately on its own.

Answer to RQ1. Scrumban adoption was motivated by a combination of organizational and technical needs rather than by a single factor. Management sought an approach capable of coordinating multidisciplinary work, integrating software and hardware activities, supporting incremental delivery, and providing sufficient structure while remaining adaptable to different projects and changing demands.

4.2 RQ2: Scrumban Use and Adaptation for Software and Hardware Coordination

The second research question examines how teams and management use, monitor, and adapt Scrumban in their daily work. Evidence from both data sources shows that Scrumban functions as a shared coordination structure, but its practices are not applied uniformly across software and hardware domains. Instead, teams and management continuously adapt ceremonies, task decomposition, boards, and monitoring mechanisms to accommodate differences between the two domains. Figure 5 summarizes the findings related to RQ2, integrating evidence from technical team members and management participants regarding how Scrumban is used, monitored, governed, and adapted in the organizational context.

The **adaptation of ceremonies** illustrates how this process occurs in practice. From the team's perspective, participants questioned the suitability of applying the same Scrum cadence to hardware. T15 observed that *"status changes in the hardware team happen much more slowly,"* while T4 recommended *"one or two dailies per week at most."* The management perspective complements these perceptions by showing how this difference resulted in an organizational adaptation. M1 reported: *"In the development of hybrid projects, involving hardware, the evolution of activities does not occur at the same pace as in software. Therefore, for example, we identified the need to avoid holding daily meetings. We moved to a lower frequency of meetings, but with a higher level of information quality regarding hardware progress."* Rather than eliminating coordination across areas, the organization differentiated the cadence while preserving shared planning: *"The adaptation was: separating the daily meetings according to the pace of each area, while keeping the planning unified to ensure alignment and integration at the appropriate moments of the project"* (M1).

Task decomposition represented another important adaptation. From the team's perspective, hardware activities were more difficult to represent at the granularity expected by the process. T9 stated: *"The biggest challenge is getting those responsible for hardware activities to break these tasks into smaller, more micro parts, making them easier to measure."* T13 similarly noted that *"hardware tasks are not always easy to break down."* The management perspective shows how the organization attempted to address this issue by introducing incremental decomposition into hardware work. M1 explained that *"At the same time, we managed to apply the slicing process in hybrid projects, which was not common in traditional hardware projects"* and that *"Instead of thinking about the entire machine, we started to view it as modules"*. According to M1, *"This allowed greater clarity, technical validation with the client, and better tracking of progress"*.

The **Kanban board** was also **adapted to represent the hybrid workflow**. Team members suggested making dependencies and different types of work more explicit. T3 proposed: *"I would rethink the board's customization rules for moving cards, the states assigned to each column, and improving the use of labels to identify tasks with extra demand, blockers, or dependencies on others."* T6 similarly suggested the need to *"adapt the Kanban board to reflect hardware dependencies,"* while T11 proposed, *"I would include item ordering/acquisition in the project pipeline."* These suggestions indicate that monitoring hardware requires representing not only development tasks but also physical and logistical dependencies that affect their progress.

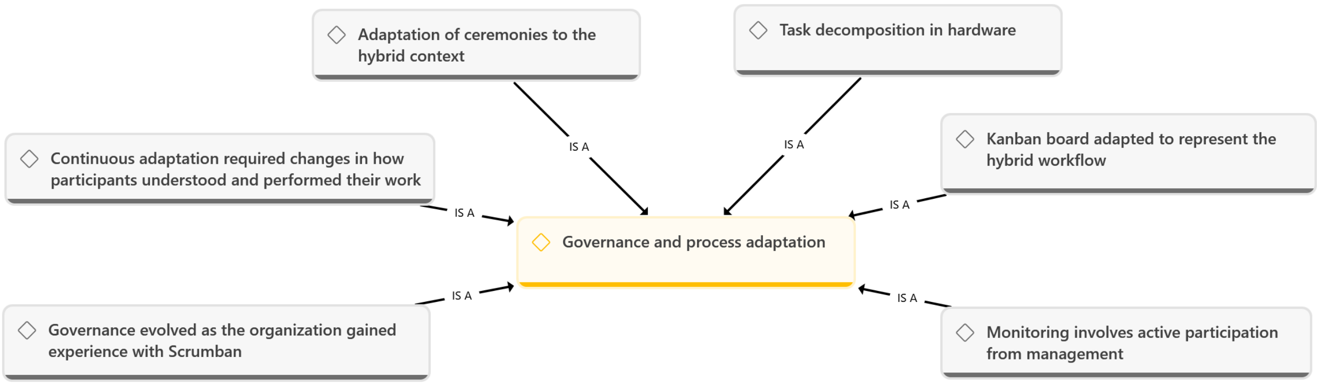

Figure 5. Governance and process adaptation mechanisms

Monitoring also **involves active participation from management**. M1 reported that *"Management participates in some ceremonies, such as planning"*, in addition to weekly status meetings in which project progress and the process itself are examined. According to M1, *"management can identify, for example, whether there are flaws in the definition of any Kanban column"*. Management also participated in the initial construction of the process: *"Management participated in these meetings where needs were identified, and the columns and ceremonies were defined"*, and *"it remained present throughout the entire process"* (M1).

However, **governance evolved as the organization gained experience with Scrumban**. M3 reported that *"at the beginning, we defined many operational aspects"*, whereas management later concentrated more on monitoring execution. M2 described part of this role as *"promoting conversation and alignment among people"*. Process adaptations also emerged from the teams themselves. M2 reported that *"some questions were raised [...] These situations emerged throughout the development and were brought to discussion with the entire team"*, resulting in adjustments to the board and workflow. These findings indicate that process governance became increasingly distributed between management and teams rather than remaining exclusively management-driven.

This **continuous adaptation also required changes in how participants understood and performed their work**. M2 reported that *"At the beginning, a change in the hardware team's mindset was necessary [...] there was initially a need for a shift in mentality"*. The team perspective similarly emphasizes that effective use depends on engagement with the process. T9 stated that *"there is no point in having a well-structured process if the team does not update the board or follow up properly*." Thus, monitoring depends not only on the existence of boards and ceremonies but also on participants consistently maintaining and interpreting these mechanisms.

Answer to RQ2. Teams and management use Scrumban through a shared but continuously adapted coordination process. Software and hardware retain common planning and visibility mechanisms, while ceremonies, task decomposition, board configurations, and monitoring practices are adjusted to their different development rhythms. Governance combines management involvement with team feedback, allowing the process to evolve as practical difficulties emerge.

4.3 RQ3: Perceived Organizational Benefits of Scrumban

The third research question examines the benefits perceived from using Scrumban. Evidence from both groups converges particularly on visibility, communication, and integration. The management interviews complement these operational benefits by showing how the information generated through Scrumban also supports project control, risk identification, decision-making, resource planning, and perceptions of team autonomy. Figure 6 summarizes the findings related to RQ3, integrating evidence from technical team members and management participants regarding the perceived benefits of using Scrumban in hybrid software and hardware development.

Visibility and project monitoring emerged as one of the clearest convergent findings. Team members emphasized the accessibility of information about ongoing work. T5 stated: *"visibility is one of the greatest benefits of Scrumban because its visual board displays the entire workflow to the team. This and daily meetings allow everyone to track progress in real time*." T1 similarly noted that *"it is very easy to see the status of activities, at which stage they are, and to estimate their completion*," while T7 explained that the ceremonies *"help us understand the status of project activities through the team's shared knowledge during ceremonies and through continuous team alignment*."

Managers associated this same visibility with greater control over project execution. M1 explained: *"Project control, because it became clear how the project was evolving in relation to the initial planning, especially with the definition of the backlog, prioritization, and so on."* The participant further noted that *"With this clarity and visibility of the execution stage [...] we can also forecast future activities [...] so we can achieve better control"*. M2 similarly stated that *"we can track, through the board, the parallelization of activities [...] so I believe it provides good visibility for management."* The two perspectives therefore emphasize different implications of the same mechanism: team members primarily associate visibility with understanding ongoing work, whereas managers associate it with monitoring, forecasting, and project control.

Communication and cross-area integration represent another point of convergence. From the team perspective, T5 stated that *"communication has been the main factor for this success. With its alignment rituals, the method ensures*

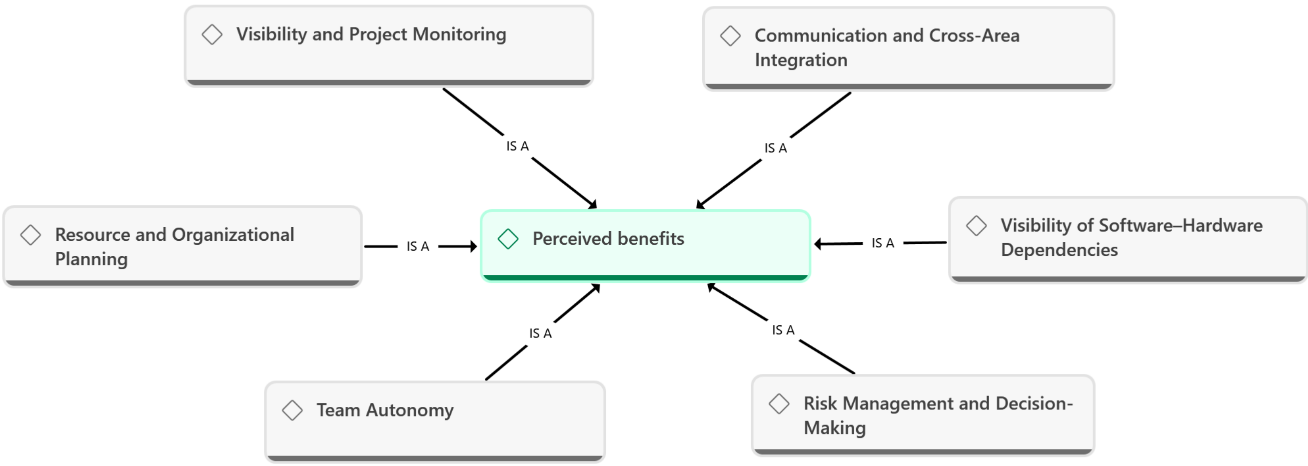

Figure 6. Perceived organizational benefits of Scrumban

constant communication between the software and hardware team." T11 noted that ceremonies *"create room for an interdisciplinary perspective*," and T14 stated that *"the meetings are essential for alignment, interaction, and understanding of the ongoing project, and they significantly support the team's progress*."

Managers perceived a similar effect. M1 reported that *"there was an improvement in engagement [...] the team started to communicate more and learned to express their needs better,"* whereas M2 described communication as the primary gain: *"for me, this was the biggest gain: communication."* M2 also observed that *"the method we used brought people much closer together."* Beyond increasing communication, managers associated the process with a shared understanding of the solution. M2 stated that *"one of the main gains was being able to see people with completely different focuses within the project aligning,"* that *"we got everyone focused on the solution,"* and that *"in addition to bringing people together, there was also a better understanding of the solution as a whole"*.

The **integrated view also helps make dependencies between software and hardware explicit**. T10 stated that *"with a well-configured board, the project's progress flow becomes visible to everyone, allowing for control over hardware and software dependencies*." T1 similarly reported that *"the parallel visualization of activities allows project management to gain more control and understanding of process progress, enabling more accurate deadline forecasting and quicker identification of blockers*." T5 further observed that *"with a visual map of end-to-end work, it would be easier to justify to management and other departments why deadlines are longer*." These accounts indicate that integrated visibility supports coordination across technical areas and helps communicate the implications of hardware dependencies to other organizational stakeholders.

The management interviews reveal additional uses of this visibility for **risk management and decision-making**. M1 reported that *"delivery risks and financial risks became much more visible,"* and that *"when you can better control the project [...] you can mitigate these risks."* M2 similarly associated Scrumban with decision support, stating that *"I believe it helps us be more assertive in these hybrid projects."* These findings complement the operational perspective by showing how managers use information about project flow and dependencies when evaluating risks and making project decisions.

Managers also associated the process with **team autonomy**. M3 stated that *"the team is self-managing [...] it will operate regardless of the manager,"* and identified *"the biggest gain was this culture of responsibility and self-sufficiency."* These accounts indicate that management perceived greater responsibility and autonomy among teams using the process.

Finally, M3 reported using the resulting visibility for **resource and organizational planning**, stating that *"it helped me even with hiring planning,"* and that *"I can plan the flow of new projects [...] decide whether to hire or not [...] and plan salary increases."* This finding extends the perceived benefits beyond day-to-day project coordination by showing how information about project flow and resource demand can inform broader management activities.

Answer to RQ3. Both groups associated Scrumban with greater visibility, communication, integration, and awareness of dependencies. Technical team members primarily emphasized operational visibility and cross-area alignment, whereas management participants connected these mechanisms to project control, risk identification, decision-making, autonomy, and resource planning. The findings therefore reveal complementary operational and managerial benefits of the same coordination mechanisms.

4.4 RQ4: Challenges and Limitations of Scrumban in Hybrid Contexts

The fourth research question investigates the challenges and limitations that emerge during Scrumban adoption and use. Integrating the two perspectives reveals two interconnected dimensions. Technical team members primarily described challenges arising from the characteristics of hybrid development, including different development cycles, physical and supplier dependencies, task granularity, estimation, and difficulties in representing hardware progress. Management participants complemented these findings by describing organizational challenges related to resistance to change, differences in agile experience and mindset, cross-area tensions, and the effort required to introduce and maintain the process. Figure 7 summarizes the findings related to RQ4, integrating evidence from technical team members and management participants regarding the challenges and limitations of adopting and using Scrumban in hybrid software and hardware development.

A central operational challenge is the **difference between software and hardware development cycles**. T6 explained that *"synchronizing distinct cycles is difficult—software allows continuous and iterative deliveries, while hardware depends on physical steps with less flexible timelines*." T15 similarly observed that *"change management becomes complex because software changes are agile, whereas hardware changes can take days or weeks for implementation, testing, and validation*." These differences also affect the compatibility between hardware work and sprint-based planning. T14 stated that *"this does not cause major project impacts, but it shows that hardware's longer execution cycles do not always align with sprint timelines*," while T12 argued that *"The sprint model is not feasible in the hardware context*."

Physical and external dependencies further constrain predictability. T11 explained that *"hardware tasks face setbacks inherent to the physical nature of the components involved, such as purchase delays and the need to ensure compatibility between devices from different architectures, which may require extending a sprint's duration*." T5 similarly stated that *"the reliance on external suppliers and the long lead times for physical prototype delivery make any hardware forecast complex*," while T12 observed that *"most of the time, the initial estimate does not match reality*." These dependencies can directly affect the software workflow when progress in one domain requires a deliverable from the other. T11 described this tension as *"there is a tough trade-off between handling the highest-priority hardware tasks or unblocking the software development flow that depends on an urgent hardware delivery*."

These characteristics also complicate **how progress is represented**. T15 warned that *"Scrumban can offer visibility, but without proper adaptations, it creates deceptive visibility that harms the team instead of helping*." The participant illustrated this limitation by explaining that *"the board shows the task 'stuck' for days, but in reality, the developer is solving multiple discovered technical issues*." T7 similarly noted that *"a challenge in hardware development is producing visible deliverables at the same frequency as the software team*," while T6 observed that *"visualizing the Kanban board becomes complicated when mixing agile development tasks with supplier dependencies or material delays*." Thus, although visibility emerged as a benefit in RQ3, the team perspective reveals an important boundary condition: the board may not accurately communicate progress when hardware work involves technical activities and dependencies that are not represented by card movement.

Maintaining that visibility can itself generate additional effort. T11 stated that *"the time spent on status reporting could be allocated to actual tasks."* T14 similarly reported that *"the time consumed updating status could be used for real activities, considering that the updates are small fragments that do not represent the whole picture."* This creates a tension between the granularity required for monitoring and the effort required to maintain that information. T2 summarized this issue by stating: *"We need to adjust monitoring, especially for the hardware team, since their deliverables are not like those in software."*

The management perspective complements these operational constraints by showing that Scrumban adoption also required an **organizational transition**. Teams accustomed to traditional planning initially had to adapt to incremental and dynamic forms of work. M1 explained: *"we needed to carry out a gradual transition. They were used to a traditional, structured schedule, so we initially brought that and started slicing and incrementing these schedules to model them into a Kanban format and introduce ceremonies, allowing us to listen and demonstrate results."* M1 also highlighted the *"difficulty in understanding how to implement increments and partial deliveries,"* and the challenge of *"moving from a static schedule to a dynamic one."*

Differences in previous experience with agile methods generated additional tensions between the areas. M2 reported that *"there were people in the software team who already mastered the methodology but did not always show empathy toward colleagues who did not understand it, leading to complaints from both sides."* These tensions were not limited to knowledge of the method. According to M2, *"we needed to understand that people think differently, as well as how they work and demonstrate value in these two types of development."* The structural differences between the domains also affected the interpretation of the process itself. M2 reported that *"there was no clear understanding, especially from the hardware team, of what each Kanban column meant and why it was important."*

Introducing Scrumban therefore **required participants to change established work routines**. M3 stated that *"implementing a new process takes the team out of their comfort zone,"* while M2 noted that *"there was a greater dedication of time to make this change happen."* These accounts indicate that continuous adaptation has an organizational cost: introducing the process requires guidance, monitoring, discussion, and time before the practices become incorporated into everyday work.

Answer to RQ4. The challenges arise from both the characteristics of hybrid development and the organizational change required to adopt Scrumban. Technical team members highlighted longer, less predictable hardware cycles; physical and supplier dependencies; difficulties in task decomposition and estimation; and limitations in representing progress using conventional agile mechanisms. Management participants highlighted the complementary organizational challenge of bringing together professionals with different methodological backgrounds and work practices, requiring learning, negotiation, process adaptation, and additional implementation effort.

4.5 Conceptual Model of Scrumban Adoption and Adaptation

Figure 8 presents a high-level conceptual model of Scrumban adoption and adaptation in hybrid software–hardware development contexts, derived from the relationships identified during the qualitative analysis. The model synthe-

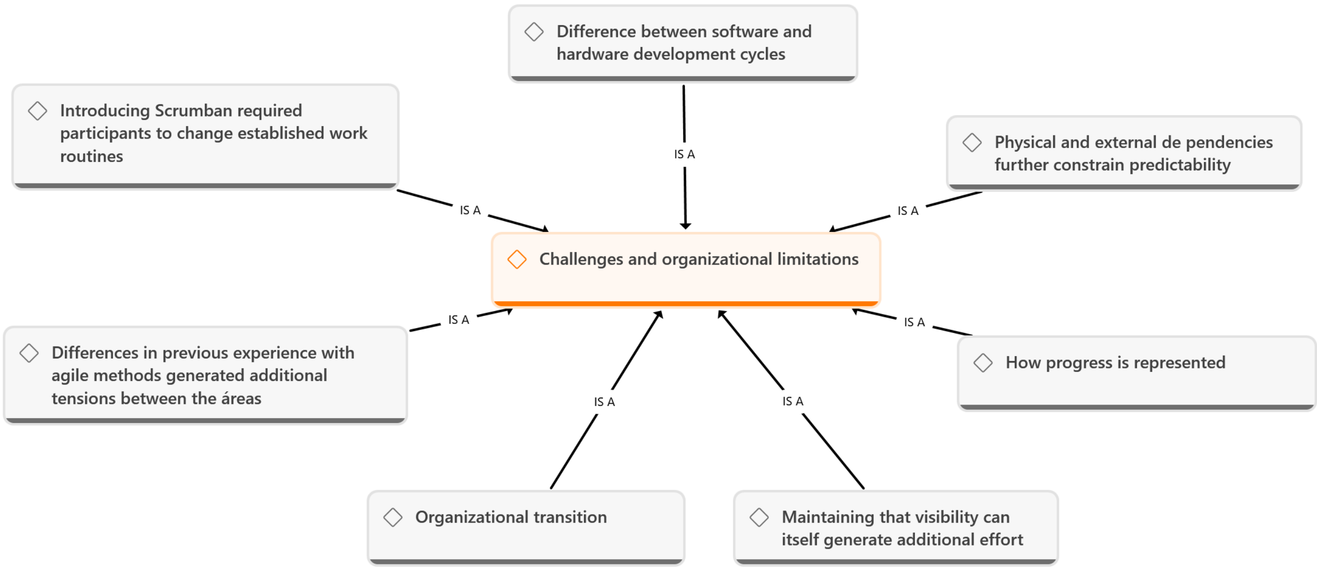

Figure 7. Challenges and limitations in hybrid Scrumban adoption and use

sizes the connections observed across the findings from both the technical team members and management participants. Rather than representing a prescriptive or strictly linear process, the model characterizes Scrumban adoption as context-dependent and iterative, in which the characteristics of hybrid development simultaneously create coordination needs and impose constraints that shape how Scrumban is adapted and used. These adaptations enable coordination mechanisms that contribute to perceived outcomes and, in turn, inform continuous adaptations of the approach.

The conceptual model begins with the **Hybrid Development Context**, which represents environments in which software and hardware activities coexist and depend on one another throughout project execution. The findings indicate that this context creates **Coordination Needs** associated with integrating multidisciplinary teams, increasing work visibility, managing dependencies across technical areas, supporting incremental deliveries, and providing sufficient flexibility to accommodate changes during development. These needs **motivate Scrumban Adoption**, as participants described the approach as a way to combine process structure with the flexibility and visibility needed to coordinate heterogeneous development activities.

At the same time, the **Hybrid Development Context** imposes **Hybrid Development Constraints** that distinguish these projects from conventional software development environments. The findings highlighted differences between software and hardware development cycles; physical and supplier dependencies; difficulties in decomposing certain hardware activities; limitations in representing hardware progress through conventional agile mechanisms; and differences in how technical areas engage with agile practices. These constraints do not necessarily result from Scrumban itself; rather, they characterize the environment in which the approach operates and therefore **shape Scrumban Adaptation**.

The model consequently distinguishes **Scrumban Adoption** from **Scrumban Adaptation**. While coordination needs motivate the organization to adopt Scrumban, the findings indicate that its application in a hybrid environment requires

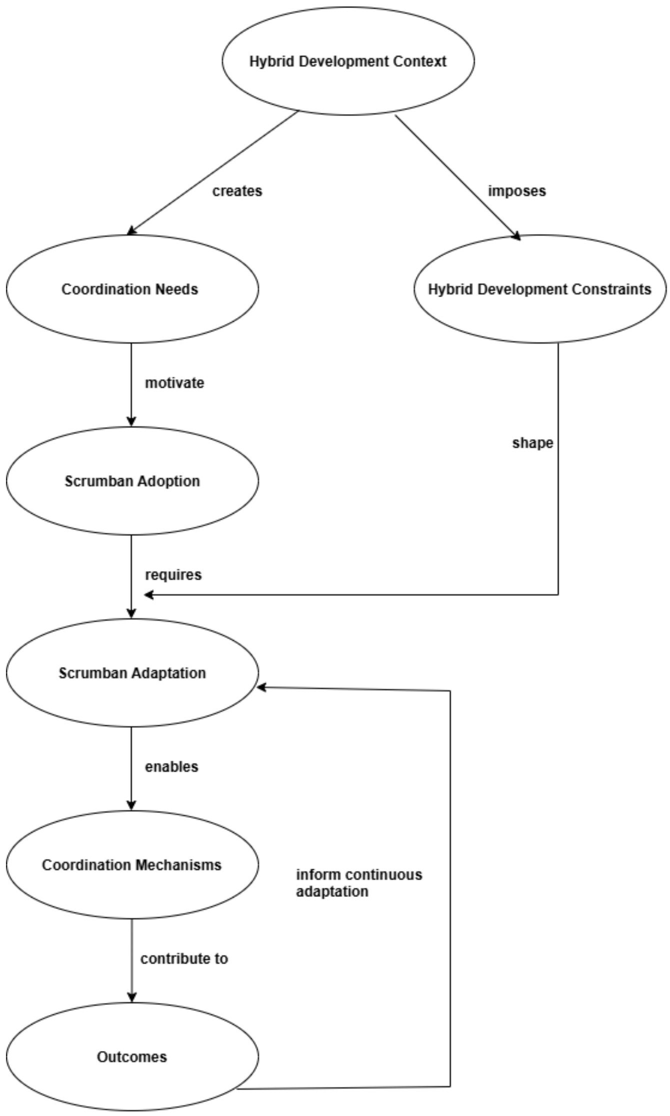

Figure 8. Conceptual Model of Scrumban Adoption and Adaptation in Hybrid Software and Hardware Development

adjustments to accommodate the characteristics and constraints of software and hardware development. Participants reported adaptations involving workflow configuration, task decomposition, meeting cadence, board structure, monitoring practices, and the coordination of dependencies across areas. Thus, the model represents Scrumban not as a uniform process applied identically across technical domains, but as an adaptable coordination structure whose practices are adjusted according to the characteristics of the work being coordinated.

These adaptations enable **Coordination Mechanisms** through which teams and management establish shared visibility and coordinate interdependent work. The findings show that visual workflows, shared boards, communication practices, integration checkpoints, and the explicit representation of dependencies help connect software and hardware activities while preserving flexibility for domain-specific workflows. These mechanisms provide a common coordination layer through which participants can monitor progress, identify dependencies, communicate across areas, and align activities without requiring software and hardware teams to follow identical development cycles.

The model further indicates that these **Coordination Mechanisms contribute to Outcomes** at both operational and management levels. At the operational level, participants associated Scrumban with greater workflow visibility, communication, cross-area integration, awareness of dependencies, and team autonomy. At the management level, participants described benefits related to project monitoring, risk identification, decision-making, prioritization, and resource and organizational planning. These outcomes should be interpreted as participants' perceptions of the effects associated with Scrumban use rather than as causal improvements in organizational performance.

Finally, the model represents Scrumban adaptation as an iterative process. The **Outcomes** experienced during its use **inform continuous adaptation**, as teams and managers identify practices that work effectively, persistent limitations, and new adjustments required by the characteristics of ongoing projects. In this sense, adaptation does not represent a single implementation stage following adoption. Instead, the findings suggest an ongoing cycle in which experience with the coordination mechanisms and their outcomes informs subsequent modifications to Scrumban practices. Overall, the conceptual model characterizes Scrumban in hybrid software–hardware development as an adaptive coordination approach in which contextual coordination needs motivate adoption, hybrid development constraints shape adaptation, and the resulting coordination mechanisms contribute to operational and management outcomes while supporting continuous process evolution.

5 Discussion and Lessons Learned

Our findings show that Scrumban provides a common coordination structure for hybrid software and hardware development while requiring adaptations to accommodate the characteristics of each domain. By integrating technical team and management perspectives, we found that Scrumban supports operational coordination through visibility, communication, and dependency management, while also providing information that supports project monitoring, risk identification, decision-making, and resource planning. At the same time, differences in development cycles, physical dependencies, task granularity, and prior experience with agile practices constrain the uniform application of Scrumban across software and hardware domains.

5.1 A Shared but Adaptable Coordination Structure

Our findings suggest that Scrumban should not be understood as a uniform process applied identically across software and hardware. Instead, it can provide a shared coordination structure in which teams maintain common mechanisms for planning, communication, and workflow visibility while adapting practices to the characteristics of each domain. This finding is consistent with previous research showing that Scrumban formation depends on project context and team characteristics (Alqudah and Razali, 2018), reinforcing the importance of selecting and combining Scrum and Kanban practices according to development needs. Previous studies also characterize Scrumban as a flexible alternative that combines structured Scrum practices with Kanban mechanisms for workflow management and continuous adaptation (Patil and Neve, 2018; Fuentes-Del-Burgo et al., 2022).

Our results extend these findings to hybrid software and hardware development. Previous research has shown that transferring agile practices from software to hardware involves challenges related to physical constraints, longer development cycles, dependencies, and different possibilities for incremental delivery (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019; Weichbroth, 2022; Drutchas and Eppinger, 2022). In our case, these differences did not prevent the use of a common Scrumban structure, but required adaptations in ceremonies, task decomposition, workflow configuration, and development cadence. Therefore, integration did not depend on software and hardware following identical processes, but on maintaining common coordination points while accommodating domain-specific characteristics.

5.2 Visibility as a Benefit and a Boundary Condition

Visibility emerged as one of the main benefits of Scrumban, supporting both operational coordination and management activities. This finding is consistent with previous Scrumban studies that associate workflow visualization with improved coordination, communication, and process monitoring (Alqudah and Razali, 2018; Patil and Neve, 2018). In hybrid development, visibility was particularly relevant because it helped participants understand dependencies between software and hardware activities and provided managers with information to monitor project progress, identify risks, and support decision-making.

However, our findings also reveal a boundary condition of this benefit. Hardware progress is not always adequately represented by frequent task transitions on a visual board. Physical development activities may remain in the same workflow state while engineers address technical issues, wait for components, integrate, or rely on external suppliers. These characteristics are consistent with challenges previously reported for agile hardware development, where physical constraints and dependencies can limit the direct application of practices originally developed for software environments (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019; Drutchas and Eppinger, 2022). Consequently, workflow visibility can be misleading when the process representation does not reflect the actual progress of hardware activities.

This finding suggests that organizations should seek meaningful rather than maximum visibility. Intermediate milestones, explicit representation of dependencies, and domain-specific workflow states may help represent hardware progress without requiring excessive task fragmentation or reporting effort. Thus, the value of visualization depends not only on making work visible but also on ensuring that the representation reflects how work actually progresses in each domain.

5.3 From Operational Coordination to Managerial Support

Our findings also broaden the understanding of the value generated by Scrumban. Previous studies have primarily associated Scrumban with benefits such as workflow management, flexibility, waste reduction, productivity, and coordination (Alqudah and Razali, 2018; Patil and Neve, 2018). In our study, technical team members similarly emphasized visibility, communication, and cross-area coordination. However, the management perspective revealed that these operational mechanisms also supported project control, risk identification, forecasting, decision-making, and resource planning.

This indicates that the same Scrumban mechanisms can generate complementary value at different organizational levels. Information used by technical teams to coordinate dependencies can also help managers understand project evolution and anticipate resource and delivery needs. At the same time, this managerial value depends on the quality of the underlying information. If boards do not adequately represent hardware progress or external dependencies, management interpretations may also be affected. Therefore, adapting Scrumban to hybrid development is relevant not only for technical coordination but also for ensuring that project information supports reliable organizational decision-making.

5.4 Lessons Learned and Implications for Practice

Based on the integrated analysis of technical team and management perspectives, we derive the following lessons for organizations adopting Scrumban in hybrid software and hardware development:

- **Use Scrumban as a shared coordination structure rather than a uniform process.** Software and hardware teams can share planning, visibility, and communication mechanisms while maintaining workflows and cadences appropriate to each domain. This flexibility is consistent with the contextual nature of Scrumban adoption reported in previous studies (Alqudah and Razali, 2018).
- **Adapt agile practices to the development rhythm of each domain.** Short iterations and frequent ceremonies may be suitable for software but difficult to reproduce in hardware activities characterized by physical constraints, longer cycles, and external dependencies (Ovesen and Dowlen, 2012; Atzberger and Paetzold, 2019; Drutchas and Eppinger, 2022).
- **Design for meaningful visibility.** Visual boards should represent not only task status but also relevant dependencies, intermediate milestones, and constraints. Otherwise, apparent workflow stagnation may not accurately represent the technical progress of hardware activities.
- **Treat software and hardware integration as a collaborative challenge.** Boards and ceremonies facilitate coordination, but effective integration also requires communication and mutual understanding among professionals with different technical backgrounds and previous experiences with agile practices.
- **Make physical and external dependencies explicit.** Supplier lead times, component acquisition, physical integration, and external validation can affect both hardware and software activities and should therefore be considered in planning and workflow monitoring.
- **Use workflow information across organizational levels.** Scrumban artifacts can support daily technical coordination while also providing information for project monitoring, risk assessment, prioritization, and resource planning.
- **Treat Scrumban adoption as a continuous learning process.** Organizations should expect to adjust ceremonies, workflows, task granularity, and monitoring mechanisms as teams gain experience with the approach. Such adaptation is particularly relevant because Scrumban practices can vary according to project and team characteristics (Alqudah and Razali, 2018).

6 Limitations

In this section, we discuss the limitations of our study following the recommendations for reporting empirical software engineering research proposed by Ralph et al. (2020).

Our study has limitations related to its qualitative design and the characteristics of the case under investigation. First, we conducted the study within a single RD&I institute, and the findings therefore reflect the organizational and technical characteristics of this specific context. The institute operates through multidisciplinary teams, highly specialized professionals, and organizational support to adapt development practices. These characteristics may differ from those of organizations with more rigid structures, lower team autonomy, different levels of agile maturity, or different forms of software–hardware integration. Rather than aiming for statistical generalization, we provide detailed information about the organizational context, participants, and research procedures to help readers assess the transferability of the findings to other settings. Future studies involving organizations with different structures, cultures, and levels of experience with agile and hybrid development could further examine the transferability of our findings.

The number and distribution of participants also represent a limitation. Although the technical team survey included 15 professionals from different roles involved in hybrid development, the management perspective was represented by three participants. This smaller number of management participants may have limited the diversity of strategic and organizational perspectives captured in the study. Moreover, these participants were directly involved in the adoption, implementation, and evolution of Scrumban, which may have influenced how they perceived and reported its benefits and limitations. To mitigate this limitation, we integrated their accounts with evidence obtained from the multidisciplinary technical team, allowing us to examine similarities, complementarities, and differences between operational and management perspectives.

The data collection methods also introduce limitations. Both the survey and interviews captured participants' perceptions and experiences rather than direct measurements of project performance. Therefore, findings on aspects such as visibility, predictability, communication, coordination, and decision-making reflect perceived effects of Scrumban and should not be interpreted as evidence of causal improvements in project performance. Future studies could complement these perceptions with quantitative and archival evidence, such as lead time, cycle time, throughput, delivery predictability, blocked-task duration, dependency records, or other project indicators before and after Scrumban adoption.

The retrospective nature of part of the data may also introduce memory and interpretation biases. Participants described experiences accumulated during the adoption and use of Scrumban, and their accounts may reflect later interpretations influenced by subsequent project experiences and outcomes. We mitigated this limitation by collecting evidence from professionals occupying different technical and managerial roles and by comparing their perspectives during the analysis. Nevertheless, future longitudinal studies could follow hybrid projects throughout their Scrumban adoption and collect evidence at different time points, reducing reliance on retrospective accounts.

Another limitation concerns the qualitative analysis. We used Open Coding and Axial Coding procedures, informed by Grounded Theory, to systematically identify and relate concepts emerging from survey responses and interviews. Because our objective was not to develop a theory, we did not pursue theoretical saturation. Therefore, the resulting categories should be understood as an analytical organization of the findings observed in this case rather than as a comprehensive theoretical representation of Scrumban adoption in hybrid development. To strengthen the credibility of the analysis, multiple researchers reviewed and discussed the codes, categories, and interpretations, and we conducted member checking with a participant involved in the investigated context.

Finally, the study focuses on Scrumban adoption within hybrid software and hardware projects in an RD&I environment. Although this context enabled us to examine interactions between agile practices and characteristics such as physical dependencies, different development cycles, and multidisciplinary coordination, the findings may not capture challenges that emerge in other hybrid settings. Future research could investigate Scrumban in different organizational and industrial contexts and examine whether the adaptations, benefits, challenges, and boundary conditions identified in this study also emerge under different project characteristics.

7 Final Considerations

This study investigated the adoption, adaptation, and use of Scrumban in a hybrid software and hardware development context by integrating the perspectives of development team members and management. Our findings show that Scrumban can provide a shared coordination structure for multidisciplinary development, supporting workflow visibility, communication, cross-area integration, and adaptation to changing project needs. At the same time, its use in hybrid contexts requires adaptations to accommodate characteristics of hardware development, including longer development cycles, physical and external dependencies, and activities that are less amenable to frequent decomposition and incremental delivery.

The findings indicate that the value of Scrumban extends beyond operational coordination. While development team participants emphasized benefits related to workflow visibility, communication, and coordination across technical areas, management participants highlighted how the information generated through Scrumban supports project monitoring, risk identification, decision-making, and resource planning. By integrating these perspectives, our study shows that the same coordination mechanisms can provide complementary value across different organizational levels, connecting the day-to-day management of technical activities to broader project and organizational needs.

However, our findings also show that Scrumban should not be applied uniformly across software and hardware. Differences in development rhythms, task granularity, physical constraints, supplier dependencies, and prior experience with agile practices require organizations to adapt workflows, ceremonies, planning mechanisms, and task representations to the characteristics of each domain. In particular, visualizing work does not necessarily mean accurately representing its progress. Hardware activities may remain in the same workflow state while technical work continues, creating the impression of stagnation. Therefore, organizations should seek meaningful visibility by representing relevant dependencies, intermediate milestones, and domain-specific workflow states rather than simply increasing task fragmentation or board updates.

Overall, our findings suggest that Scrumban can support hybrid software and hardware development when organizations treat it as an adaptable coordination structure rather than as a uniform process. Its use benefits from common mechanisms for communication and workflow visibility while preserving sufficient flexibility to accommodate domain-specific characteristics. The study contributes empirical evidence on the use of Scrumban in an underexplored hybrid development context and provides practical lessons for organizations seeking to coordinate multidisciplinary teams working across software and hardware boundaries.

Future research should investigate Scrumban in other organizational and industrial contexts to examine the transferability of the adaptations, benefits, challenges, and boundary conditions identified in this study. Longitudinal studies could also investigate how Scrumban practices and governance mechanisms evolve over time as teams gain experience with hybrid development. In addition, future studies could combine professionals' perceptions with quantitative and archival project data, such as lead time, cycle time, throughput, blocked-task duration, and delivery predictability, to examine how Scrumban relates to project performance. Further research may also investigate how Scrumban artifacts support organizational activities such as resource allocation, risk management, and portfolio-level decision-making.

Declaration of AI Use

We acknowledge the use of Grammarly and ChatGPT 5.1 Instant for the translation, improvement of spelling, grammar, vocabulary, and style of the text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Data Availability

The research instruments and materials used in this study are publicly available on https://figshare.com/s/dc003a37b0fa4a7c7030. To protect participants' privacy and prevent the disclosure of sensitive information, all personal information reported in this article has been anonymized.

Acknowledgments

We thank all participants of the empirical study and INDT for their support. This work results from the Research, Development, and Innovation (RD&I), using resources from the Informatics Law for the Western Amazon (Federal Law 11.196/2005). The present work is supported by: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (AUXPE-CAPES-PROEX) – Financing Code 001; Amazonas State Research Support Foundation – FAPEAM – through PDPG-CAPES, POSGRAD 2025-2026, POSGRAD 2026-2027; and Project No. 017/2024 – DIVULGA CT&I/FAPEAM.

References

Ahmad, M. O., Kuvaja, P., Oivo, M., and Markkula, J. (2016). Transition of software maintenance teams from scrum to kanban. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 5427–5436. IEEE.

Albarqi, A. A. and Qureshi, R. (2018). The proposed l-scrumban methodology to improve the efficiency of agile software development. *International Journal of Information Engineering and Electronic Business*, 11(3):23.

Alqudah, M. and Razali, R. (2018). An empirical study of scrumban formation based on the selection of scrum and kanban practices. *Int. J. Adv. Sci. Eng. Inf. Technol*, 8(6):2315–2322.

Atzberger, A. and Paetzold, K. (2019). Current challenges of agile hardware development: What are still the pain points nowadays? In *Proceedings of the design society: international conference on engineering design*, volume 1, pages 2209–2218. Cambridge University Press.

Bhavsar, K., Shah, V., and Gopalan, S. (2020). Scrumbanfall: an agile integration of scrum and kanban with waterfall in software engineering. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9(4):2075–2084.

Drutchas, J. and Eppinger, S. (2022). Guidance on application of agile in combined hardware and software development projects. *Proceedings of the Design Society*, 2:151–160.

Fuentes-Del-Burgo, J., Pérez, S., and Ángel, M. (2022). Comparative analysis of the board tool in the agile methodologies scrum, kanban and scrumban in software projects. In *26 th International Congress on Project Management and Engineering Terrassa*.

Gaborov, M., Karuović, D., Kavalić, M., Radosav, D., Milosavljev, D., Stanisaljev, S., and Bushati, J. (2021). Comparative analysis of agile and traditional methodologies in it project management. *Journal of Applied Technical and Educational Sciences*, 11(4):1–24.

Lei, H., O'Connell, R., Ehwerhemuepha, L., Taraman, S., Feaster, W., and Chang, A. (2020). Agile clinical research: A data science approach to scrumban in clinical medicine. *Intelligence-based medicine*, 3:100009.

Martins, L. T., Silva, M. E., Milnitz, D., and Marchi, J. J. (2025). Integração de tempos e métodos com kanban-um estudo de caso na indústria automotiva. *ERR01*, 10(5):e8692–e8692.

Meireles, M. A., Souza, E., Lobao, L., Oliveira, E., Steinmacher, I., Maldonado, J. C., and Conte, T. (2026). Applying design thinking in research and development environment: Insights from a case study in a brazilian r&d institute. *The Journal of Systems and Software 242 (2026) 113029*.

Meireles, M. A. C., Soares, K., Afonso, Á., de Araújo, L. L., Lobão, L., and Pinheiro, G. (2025). Applying scrumban in hybrid software and hardware teams: An experience report. In *Proceedings of the XXIV Brazilian Symposium on Software Quality*, SBQS 2025, pages 289–297, Porto Alegre, RS, Brasil. Sociedade Brasileira de Computação.

Minella, C. (2025). Managing agile across borders: A review of scrum implementation in globally distributed software development. *Journal of Policy Options*, 8(2):37–45.

Ovesen, N. and Dowlen, C. (2012). The challenges of becoming agile–experiences from new product development in industry and design education. In *International Conference on Engineering and Product Design Education*. The Design Society.

Patil, S. P. and Neve, J. R. (2018). Productivity improvement of software development process through scrumban: A practitioner's approach. In *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, pages 314–318. IEEE.

Patilla, H. J., Alviar, J. M. V., and Conislla, Y. M. (2023). Scrumban/xp: Propuesta para mejorar la eficiencia de la gestión de proyectos ágiles en el desarrollo de software. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E61):14–32.

Ralph, P., Hoda, R., and Treude, C. (2020). Acm sigsoft empirical standards.

Rautakoura, A. and Hämäläinen, T. (2023). Does soc hardware development become agile by saying so: A literature review and mapping study. *ACM transactions on embedded computing systems*, 22(3):1–27.

Reich, A. and Reich, N. (2025). Scrum in global software development: Challenges, risks, and mitigation strategies for effective project management. *Journal of Policy Options*, 8(1):43–50.

Schwaber, K. and Sutherland, J. (2020). O guia do scrum: O guia definitivo para o scrum: As regras do jogo.

Semerano, V. Z. and De Oliveira, L. F. (2024). O papel estratégico do analista de qualidade (qa) em equipes scrum, kanban e scrumban no desenvolvimento de software ágil. *Advances in Global Innovation & Technology*, 2(2):32–45.

Stol, K.-J. and Fitzgerald, B. (2020). Guidelines for conducting software engineering research. In *Contemporary Empirical Methods in Software Engineering*, pages 27–62. Springer.

Strauss, A. and Corbin, J. (1998). Basics of qualitative research techniques. Thousand oaks, CA: Sage publications.

Weichbroth, P. (2022). A case study on implementing agile techniques and practices: Rationale, benefits, barriers and business implications for hardware development. *Applied Sciences*, 12(17):8457.