Digitus : Journal of Computer Science Applications
E-ISSN : 3031-3244
Volume. 2, Issue 2, April 2024
Page No: 110-121

Balancing Cadence and Flow: Evaluating Agile Frameworks for Optimal Software Delivery Outcomes

Veronika Yuni T
Universitas Jayabaya, Indonesia
Correspondent: Veronikayuni2020@gmail.com

Received : March 10, 2024
Accepted : April 12, 2024
Published : April 30, 2024

Citation: Yuni T, V. (2024). Balancing Cadence and Flow: Evaluating Agile Frameworks for Optimal Software Delivery Outcomes. Digitus : Journal of Computer Science Applications, 2 (2), 110-121.

https://doi.org/10.61978/digitus.v2i2.950

ABSTRACT: This study compares the impacts of Scrum and Kanban on software quality, team sustainability, and project predictability within Agile project management. As Agile adoption expands across industries, organizations face the challenge of selecting methods that fit their operational needs and team dynamics. By drawing on empirical case studies and literature, this research highlights the practical differences between Scrum's cadence-based framework and Kanban's flow-based model. A comparative analysis was conducted using data from major implementations (e.g., Adobe, John Deere, BBC Worldwide), supported by Agile maturity studies and academic evaluations. Metrics examined include defect reduction, cycle time, velocity stability, lead time, and team stress levels. Scrum demonstrated strong outcomes in early-stage quality improvement and structured delivery. Kanban, in contrast, offered stronger long-term flow consistency and fewer customer-reported defects. Furthermore, hybrid approaches such as Scrumban emerged as practical alternatives that balance predictability with adaptability. Results indicate that both frameworks yield significant benefits when implemented with high team autonomy and cultural alignment. While Scrum enhances predictability through time boxed sprints, Kanban facilitates flexibility and continuous delivery. The study highlights the critical role of implementation quality and Agile maturity in determining success. In conclusion, method choice should reflect organizational context, with growing support for hybrid adoption. This research provides actionable insights for Agile teams and decision makers seeking to align methodology with project goals, workforce dynamics, and customer expectations.

Keywords: Scrum, Kanban, Agile Project Management, Software Quality, Team Sustainability, Project Predictability, Hybrid Frameworks.

INTRODUCTION

Agile methodologies have transformed software development by emphasizing adaptability, continuous delivery, and cross-functional collaboration. Within this context, Scrum and Kanban stand out as two of the most widely used Agile frameworks, each providing distinct approaches to managing work and improving productivity. Scrum is characterized by a structured, time bound iteration model involving specific roles such as the Scrum Master and Product Owner. These iterations, or sprints, typically span two to four weeks and involve planned cycles of development, review, and retrospective sessions aimed at incremental delivery (Amajuoyi et al., 2024; Trihardianingsih et al., 2023). On the other hand, Kanban offers a more flexible, continuous flow based system that visualizes work stages, limits work in progress (WIP), and facilitates real time task management without fixed roles or iterations (Brad et al., 2019; Orlov et al., 2021).

Scrum and Kanban have been widely adopted across industries, particularly in software development. Surveys show Scrum remains the most prevalent Agile practice, while Kanban is increasingly used to address needs for adaptability and improved flow (Ahmad et al., 2018; Ozkan et al., 2022). Hybrid models such as Scrumban combine the cadence of Scrum with the workflow flexibility of Kanban (Santos et al., 2018). However, adoption is not without obstacles: organizations frequently encounter cultural resistance, unclear roles, insufficient training, and the complexity of scaling Agile across teams (Hanslo & Mnkandla, 2018; Senapati et al., 2020). For example, large enterprises often report difficulty sustaining Agile ceremonies consistently across distributed teams.

Several factors influence the decision to transition between methodologies or adopt hybrid approaches. Kanban is often preferred in environments with fluctuating workloads, support tasks, or a need for frequent responsiveness, where its real time task management proves advantageous (Alqudah & Razali, 2018). Conversely, Scrum is better suited for projects requiring regular stakeholder engagement and well defined deliverables. As software development evolves, blending the strengths of both frameworks becomes increasingly important to accommodate both planning rigor and adaptive execution (Alshammari, 2022).

The impact of Agile frameworks on team dynamics and development outcomes has been well documented. Agile environments generally foster a culture of transparency, continuous feedback, and team ownership, which have been associated with higher job satisfaction and improved productivity (Hsu, 2019). Iterative feedback loops ensure closer alignment with client expectations, contributing to better product market fit and reduced rework (Saltz & Sutherland, 2020).

While both Scrum and Kanban are grounded in Agile principles, their suitability varies based on organizational goals, team maturity, and operational demands. Comparative research suggests Scrum excels in structured, cadence driven projects, while Kanban thrives in environments requiring continuous responsiveness (Orlov et al., 2021). The emergence of hybrid frameworks like Scrumban further indicates that a one size fits all approach is insufficient. Instead, organizations are increasingly tailoring Agile methodologies to meet their unique challenges and contexts (Ozkan et al., 2022).

This study aims to assess the differential impacts of Scrum and Kanban on three key dimensions of software project success: software quality, team sustainability, and project predictability. Through an integrated analysis of empirical case studies and literature, the research identifies contextual benefits, challenges, and decision making considerations. The novelty of this paper lies in its comparative framing and practical focus, offering actionable insights for organizations navigating Agile method selection. By aligning method adoption with specific project characteristics and team needs, the findings contribute to more effective and sustainable Agile implementation strategies.

METHOD

This study adopts a comparative qualitative approach to evaluate the impacts of Scrum and Kanban on software quality, team sustainability, and project predictability. The methodology is grounded in empirical evidence drawn from case studies, academic literature, and metrics commonly employed in Agile software development.

A qualitative case synthesis is used to analyze the performance of Scrum and Kanban across several implementation contexts. This approach involves comparing structured data sets, empirical results, and thematic interpretations from documented Agile practices in industry.

Data were obtained from:

- Scrum implementations at Adobe and John Deere
- Kanban implementations at BBC Worldwide and Queen's University Belfast
- Cross-method comparison studies such as the HICSS report and Agile SLR reviews
- Supplementary literature focusing on Agile metrics and team outcomes

Software quality is assessed using industry standard indicators:

- Defect Density**:** Measures confirmed defects per lines of code or function points, ensuring software adheres to release standards (Scharold et al., 2023).
- Customer Satisfaction**:** Tracked via surveys and Net Promoter Scores (Limón Ulloa Herrera, 2024).
- Code Quality**:** Evaluated using cyclomatic complexity, maintainability index, and other technical indicators that lower technical debt and enhance maintainability (Mashmool et al., 2020).

Agile specific practices like Test Driven Development (TDD) and Continuous Integration (CI) support these evaluations by introducing early testing and automated builds (Ebirim et al., 2024).

Team sustainability is gauged through:

- Engagement and Morale: Measured using regular internal surveys
- Turnover Rates: Low turnover reflects team satisfaction and organizational support
- Autonomy and Collaboration: High autonomy correlates with strong team performance and adaptability

Predictability Metrics by Method Type

Predictability is evaluated differently based on whether a framework is iteration based (Scrum) or continuous (Kanban):

- Scrum:
  - Velocity: Amount of work completed in a sprint; used for future planning
  - Planned to Done Ratio: Measures execution efficiency in planned sprints
- Kanban:
  - Lead Time and Cycle Time: Assess task flow from initiation to completion
  - Cumulative Flow Diagrams (CFD): Provide visual insight into WIP and bottlenecks

Synthesis Approach

The study synthesizes findings by aligning each case and metric with method specific strengths and limitations. Comparative interpretation highlights patterns in software quality gains, sustainability traits, and predictability mechanisms.

RESULT AND DISCUSSION

Early testing plays a pivotal role in improving Scrum project quality. By integrating test driven development (TDD) and continuous integration (CI), teams identify and resolve defects during the development phase, leading to significantly reduced defect density (Agha et al., 2023; Alphonce, 2024). These practices ensure that code is consistently validated, and integration issues are addressed early in the software lifecycle. At Adobe, this approach led to a 42% reduction in software defects, demonstrating the practical impact of proactive quality assurance. Similarly, John Deere implemented Scrum at scale and achieved a 79% improvement in cycle time, reflecting faster iteration speed and better early stage validation. These results indicate that Scrum, when executed with strong testing practices, significantly enhances product stability and reduces post deployment issues.

In contrast, Kanban supports quality enhancement through its core principle of continuous delivery (CD). Teams adopting Kanban release smaller increments more frequently, benefiting from immediate feedback loops that enable quick identification and resolution of issues (Mohammed et al., 2024; Neri et al., 2025). This facilitates higher code quality and contributes to lower technical debt over time. At BBC Worldwide, for example, the adoption of Kanban led to a 24% reduction in customer reported defects, highlighting how continuous flow models can yield tangible improvements in user perceived software reliability. Kanban's emphasis on work in progress (WIP) limits also encourages developers to focus on fewer tasks simultaneously, thereby improving code accuracy and reducing context switching.

Longitudinal studies further validate that Agile team maturity correlates with declining defect rates. Over time, both Scrum and Kanban teams show improvements in collaboration, automated testing, and consistent review cycles, which contribute to long term quality gains (Huss et al., 2023).

Comparative analyses show that while Scrum teams may initially struggle with quality due to the pressures of sprint deadlines, they tend to improve as experience and team cohesion grow. Kanban, on the other hand, typically sustains a more stable low defect rate from the outset due to its adaptive and flow centric model. Research suggests that Kanban environments are especially beneficial for maintaining consistent quality in teams handling fluctuating workloads or complex, overlapping project streams (Neri et al., 2025).

Sustainability in Agile teams is heavily influenced by how work is structured and managed. Stress levels, a critical factor in long term performance, vary based on the chosen framework. Scrum teams, operating under fixed length sprints, may experience elevated pressure due to hard deadlines and ambitious sprint goals. This can lead to increased stress and potential burnout, especially if sprints are poorly planned or unrealistic expectations are set (Lee et al., 2023). However, with careful management and proper role definition, Scrum teams can achieve a strong rhythm that supports sustainable delivery.

Kanban teams typically report lower stress levels due to the absence of time boxed deadlines. The flexible nature of Kanban allows work to be pulled based on capacity, leading to smoother workflows and better alignment with team capabilities (Mohammed et al., 2024). Psychometric evaluations have shown that Kanban fosters healthier work patterns and lowers burnout risks over time. Empowerment within Agile teams regardless of method has been found to directly improve team sustainability. Teams granted autonomy and decision making authority exhibit higher motivation, collaboration, and accountability, all of which are linked to better outcomes (Клоков et al., 2024).

Sustainability indicators such as consistent velocity, balanced task distribution, and regular retrospectives contribute to a supportive team environment. Agile practices encourage continuous feedback and self organization, which enhances engagement and reduces turnover (Barroca et al., 2018; Junker et al., 2021). Additionally, burnout trends vary by methodology. Scrum teams under constant deadline pressure may be more prone to fatigue, while Kanban's flow oriented structure provides better workload distribution and adaptability to changing conditions (Neri et al., 2025). These findings suggest that organizations must carefully consider their team dynamics, project nature, and operational tempo when choosing a framework to promote long term team health.

Predictability is a major goal in Agile environments and is achieved through different mechanisms in Scrum and Kanban. Scrum provides predictability through clearly defined sprints and performance metrics such as velocity and planned to done ratios. Velocity is typically calculated by averaging the number of story points completed across several sprints. This offers teams and stakeholders a reliable indicator of delivery capacity and supports more accurate sprint planning (Huss et al., 2023). When tracked consistently, velocity data enables project managers to estimate future work completion timelines with greater confidence, improving alignment with stakeholder expectations.

Kanban’s approach to predictability centers on flow based metrics such as lead time and cycle time. These are tracked using tools like Jira, Kanbanize, or Trello, which help visualize work progress and measure the duration of tasks from initiation to completion (Mohammed et al., 2024). Kanban boards and Cumulative Flow Diagrams (CFD) further enhance visibility, making it easier to spot workflow bottlenecks and reallocate resources accordingly. This methodology allows for ongoing adjustments, fostering a more adaptable and resilient delivery system.

Release cadence also distinguishes the two methods. Scrum adheres to time boxed sprints, typically releasing software at the end of each sprint cycle (Agha et al., 2023). This structured release schedule suits environments with regulatory or stakeholder imposed timelines. Kanban enables continuous deployment, aligning software releases with user readiness and real time demand (Neri et al., 2025). This real time responsiveness allows teams to pivot quickly in response to feedback or emerging priorities, increasing business agility.

Empirical evidence confirms that both frameworks enhance project predictability when compared to traditional waterfall models. Scrum's advantage lies in its repeatable cycles and transparent planning structure, which improve delivery forecasting over time. Meanwhile, Kanban's real time feedback and flexible cadence allow for more adaptive planning and higher consistency in task delivery (Клоков et al., 2024). The choice between Scrum and Kanban should therefore depend on the desired balance between structure and flexibility, as well as the specific predictability needs of the project or organization.

Method Culture Fit and Organizational Maturity

The results of this study underscore how Scrum and Kanban each bring distinctive strengths to Agile software project execution, with varied implications for software quality, team sustainability, and predictability. However, their effectiveness is deeply intertwined with an organization's culture, maturity, implementation strategy, and level of team autonomy. The successful integration of either framework into a team's workflow often depends on how well the methodology aligns with prevailing organizational norms, leadership dynamics, and structural readiness for change.

Scrum, with its structured cadence and clearly defined roles, requires a certain level of organizational readiness. Teams must commit to sprints, ceremonies, and accountability mechanisms such as the Scrum Master and Product Owner roles (Gerona & Ocampo, 2023). These defined responsibilities ensure clarity in expectations and team direction, especially in companies accustomed to formalized chains of command. This prescriptive approach tends to suit organizations with a hierarchical culture that values routine and predictable planning. When executed well, Scrum fosters psychological safety, team cohesion, and shared responsibility especially in mature teams (Buvik & Tkalich, 2022). The repetition of rituals such as sprint reviews and retrospectives builds momentum, enhances feedback loops, and creates a rhythm for continuous improvement. Conversely, Scrum can be stressful or counterproductive in environments that lack the discipline or leadership support required for rigorous sprint cycles. Organizations without a strong foundation in Agile thinking may struggle to uphold consistent practices, leading to disillusionment with the methodology.

Kanban, by contrast, interacts with organizational culture more fluidly. Its non prescriptive, visual workflow model allows for incremental adoption, making it particularly effective in less mature or more diverse operational environments (Perlak, 2019). Kanban's low barrier entry encourages experimentation and learning by doing, allowing teams to refine their process organically. By emphasizing continuous improvement over prescriptive structure, Kanban reduces adoption friction and allows teams to self calibrate based on real time workload and feedback. Its flexible framework aligns with cultures that prioritize adaptability and gradual evolution over rigid rules. Teams can implement visual boards, WIP limits, and performance metrics at their own pace. Hybrid models such as Scrumban illustrate how many organizations adopt blended approaches to strike a balance between Scrum's cadence and Kanban's flow, particularly when navigating complex or evolving work environments (Scott et al., 2021). These hybrids serve as transitional solutions that help teams shift toward full Agile practices without experiencing abrupt cultural disruption.

Cadence vs Flow: Strategic Trade offs

The fundamental trade off between cadence and flow represents one of the most significant decision points in Agile practice. Cadence, as structured in Scrum, provides predictable delivery intervals, helps teams build rhythm, and supports long term planning. Regular sprints help stabilize stakeholder expectations, enabling project managers to coordinate cross functional timelines and resource allocation with more confidence. It strengthens accountability by setting expectations around sprint goals (Rathor et al., 2023). However, cadence also introduces certain limitations. Teams may feel pressured to complete deliverables within arbitrary timelines, resulting in technical debt or compromised quality. Additionally, it can discourage addressing emergent priorities that arise mid sprint, requiring scope negotiation or deferral of urgent tasks.

Flow, as emphasized in Kanban, offers greater responsiveness and delivery flexibility, enabling teams to adapt to change more organically (Wahab et al., 2024). Work items can be reprioritized on the fly, and tasks are completed based on team capacity and readiness, which supports just in time delivery. The absence of artificial time constraints reduces pressure but can also hinder delivery pace if teams lack intrinsic motivation or self discipline (Salameh & Bass, 2019). Without regular review cycles, accountability must be maintained through transparent tracking of work in progress and performance metrics such as lead time and throughput. Therefore, an organization's decision between cadence and flow should consider its operational stability, stakeholder needs, and team maturity (Werder & Maedche, 2018). Many teams may benefit from starting with cadence and gradually evolving toward flow based methods as their autonomy and process discipline increase.

Hybrid Approaches and Emerging Practices

Hybrid Agile frameworks, such as Scrumban, combine the best of both worlds leveraging Scrum's sprint based structure to maintain stakeholder visibility while adopting Kanban's pull based flow to optimize task prioritization and throughput (Scott et al., 2021). These models empower teams to adapt iteration lengths, reassign work based on availability, and introduce flexibility into sprint planning without sacrificing structure entirely. Studies indicate that hybrid models improve communication, support cross team coordination, and accommodate a broader spectrum of project types (Hofman et al., 2023). Organizations working on both exploratory research and incremental product delivery can benefit from segmenting workflows across hybrid teams tailored to their functions.

Additionally, hybrid models help scale Agile across departments by preserving some familiar structures while embedding Agile principles more deeply (Gren et al., 2019). The use of shared boards and customizable cadences in such setups makes it easier for organizations to handle shifting priorities without disrupting overarching workflows. Teams that face fluctuating demands or dependencies such as DevOps or support teams find hybrid methods especially effective. Such configurations are particularly valuable in large organizations with varying degrees of Agile maturity, allowing each team to tailor processes to their specific needs. Moreover, hybrid models offer transitional flexibility; teams uncertain about which methodology fits best can explore blended workflows before committing to a singular approach.

Implementation Quality and Team Autonomy

A crucial finding across methodologies is that implementation quality and team autonomy often determine success more than the method itself. Well implemented Scrum or Kanban processes, backed by training, leadership support, and a shared understanding of Agile values, consistently produce better results (Dutra et al., 2023; Kakar, 2018). Poor implementation, on the other hand, often leads to procedural confusion, reduced morale, and abandonment of Agile goals. Simply adopting terminology or tools without aligning them with real behavioral changes creates a disconnect that undermines long term success.

Similarly, empowering teams to make decisions fosters engagement, creativity, and ownership, leading to improved adaptability and performance (Grass et al., 2020; Moe et al., 2019). Empowered teams typically resolve issues more quickly, improve their processes iteratively, and demonstrate higher levels of satisfaction and cohesion. Lack of autonomy, by contrast, stifles innovation and leads to disengagement, undermining even the most structurally sound Agile frameworks (Rathor et al., 2023). Organizations should therefore foster environments where autonomy is nurtured through trust, minimal micromanagement, and clearly communicated strategic objectives. This cultural foundation enables Agile teams to thrive, regardless of which framework is adopted.

CONCLUSION

This study provides a comparative evaluation of Scrum and Kanban, emphasizing their impacts on software quality, team sustainability, and project predictability. Case studies from organizations such as Adobe, John Deere, and BBC Worldwide illustrate how Scrum delivers early-stage quality gains and predictable cycles, while Kanban ensures continuous flow, reduced defects, and lower stress levels. The findings confirm that neither framework is universally superior; instead, their effectiveness depends on organizational maturity, cultural alignment, and the quality of implementation.

The evidence also highlights that hybrid approaches, such as Scrumban, are increasingly adopted to balance the structured cadence of Scrum with the flexibility of Kanban. Ultimately, implementation quality and team autonomy emerge as the decisive factors for long-term Agile success. These insights offer practical guidance for organizations to align Agile methodology with project goals, workforce dynamics, and customer expectations, while future research should further investigate the long-term outcomes of hybrid adoption.

REFERENCE

Agha, D., Sohail, R., Meghji, A. F., Qaboolio, R., & Bhatti, S. (2023). Test Driven Development and Its Impact on Program Design and Software Quality: A Systematic Literature Review. Vawkum Transactions on Computer Sciences, 11(1), 268–280. https://doi.org/10.21015/vtcs.v11i1.1494

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in Software Engineering: A Systematic Mapping Study. Journal of Systems and Software, 137, 96–113. https://doi.org/10.1016/j.jss.2017.11.045

Alphonce, D. M. (2024). Enhancing Software Quality Through Early-Phase of Software Verification and Validation Techniques. International Journal of Technology and Systems, 8(4), 1–15. https://doi.org/10.47604/ijts.2268

Alqudah, M., & Razali, R. (2018). An Empirical Study of Scrumban Formation Based on the Selection of Scrum and Kanban Practices. International Journal on Advanced Science Engineering and Information Technology, 8(6), 2315–2322. https://doi.org/10.18517/ijaseit.8.6.6566

Alshammari, F. H. (2022). Analytical Evaluation of SOA and SCRUM Business Process Management Approaches for IoT-Based Services Development. Scientific Programming, 2022, 1–14. https://doi.org/10.1155/2022/3556809

Amajuoyi, P., Benjamin, L. B., & Adeusi, K. B. (2024). Optimizing Agile Project Management Methodologies in High-Tech Software Development. GSC Advanced Research and Reviews, 19(2), 268–274. https://doi.org/10.30574/gscarr.2024.19.2.0182

Barroca, L., Gregory, P., Kuusinen, K., Sharp, H., & AlQaisi, R. (2018). Sustaining Agile Beyond Adoption. 22–25. https://doi.org/10.1109/seaa.2018.00013

Brad, S., Brad, E., & Homorodean, D. (2019). CALDET: A TRIZ-Driven Integrated Software Development Methodology. 400–416. https://doi.org/10.1007/978-3-030-32497-1_32

Buvik, M. P., & Tkalich, A. (2022). Psychological Safety in Agile Software Development Teams: Work Design Antecedents and Performance Consequences. https://doi.org/10.24251/hicss.2022.880

Dutra, E., Cerdeiral, C., Lima, P., Escalfoni, R., Diirr, B., & Santos, G. (2023). Using an Instrument to Assess Trust, Knowledge, Learning, and Motivation of Agile Teams. Isys - Brazilian Journal of Information Systems, 16(1). https://doi.org/10.5753/isys.2023.3005

Ebirim, W., Montero, D. J. P., Ani, E. C., Ninduwezuor-Ehiobu, N., Usman, F. O., & Olu-lawal, K. A. (2024). The Role of Agile Project Management in Driving Innovation in Energy-Efficient Hvac Solutions. Engineering Science & Technology Journal, 5(3), 662–673. https://doi.org/10.51594/estj.v5i3.864

Gerona, A. W., & Ocampo, A. P. (2023). The Transitioning Experiences From Traditional to Agile Project Management: A Case of an Information Technology Department of a Private-Owned Financial Institution in Cebu City. International Journal of Multidisciplinary Applied Business and Education Research, 4(6), 1796–1809. https://doi.org/10.11594/ijmaber.04.06.05

Grass, A., Backmann, J., & Hoegl, M. (2020). From Empowerment Dynamics to Team Adaptability: Exploring and Conceptualizing the Continuous Agile Team Innovation Process. Journal of Product Innovation Management, 37(4), 324–351. https://doi.org/10.1111/jpim.12525

Gren, L., Goldman, A., & Jacobsson, C. (2019). Agile Ways of Working: A Team Maturity Perspective. Journal of Software Evolution and Process, 32(6). https://doi.org/10.1002/smr.2244

Hanslo, R., & Mnkandla, E. (2018). Scrum Adoption Challenges Detection Model: SACDM. 15, 949–957. https://doi.org/10.15439/2018f270

Hofman, M., Grela, G., & Oronowicz, M. (2023). Impact of Shared Leadership Quality on Agile Team Productivity and Project Results. Project Management Journal, 54(3), 285–305. https://doi.org/10.1177/87569728221150436

Hsu, H.-J. (2019). Practicing Scrum in Institute Course. https://doi.org/10.24251/hicss.2019.935

Huss, M., Herber, D. R., & Borky, J. M. (2023). Comparing Measured Agile Software Development Metrics Using an Agile Model-Based Software Engineering Approach Versus Scrum Only. Software, 2(3), 310–331. https://doi.org/10.3390/software2030015

Junker, T. L., Bakker, A. B., Gorgievski, M. J., & Derks, D. (2021). Agile Work Practices and Employee Proactivity: A Multilevel Study. Human Relations, 75(12), 2189–2217. https://doi.org/10.1177/00187267211030101

Kakar, A. K. (2018). How Do Team Cohesion and Psychological Safety Impact Knowledge Sharing in Software Development Projects? Knowledge and Process Management, 25(4), 258–267. https://doi.org/10.1002/kpm.1584

Lee, J., Han, J.-H., & Jeong, S. (2023). Porcine-Derived Soft Block Bone Substitutes for the Treatment of Severe Class II Furcation-Involved Mandibular Molars: A Prospective Controlled Follow-Up Study. Journal of Periodontal & Implant Science, 53(6), 406. https://doi.org/10.5051/jpis.2203660183

Limón-Ulloa, R. & Richard de Jesús Gil Herrera. (2024). A Cloud-KPIs Dashboard to Evaluate Agile Development Teams' Performance. Journal of Cases on Information Technology, 26(1), 1–30. https://doi.org/10.4018/jcit.356506

Mashmool, A., Khosravi, S., Joloudari, J. H., Inayat, I., Mansor, Z., & Band, S. S. (2020). A Statistical Model to Compute the Effect of Agile Teams' Productivity. https://doi.org/10.20944/preprints202009.0728.v1

Moe, N. B., Dahl, B. L., Stray, V., Karlsen, L. S., & Schjødt-Osmo, S. (2019). Team Autonomy in Large-Scale Agile. https://doi.org/10.24251/hicss.2019.839

Mohammed, A. B., Hmoud, H., Sultan, L., & Yaseen, H. (2024). The Influence of Remote Work on Scrum-Based Information Technology Projects Management: Insights for Success. The TQM Journal. https://doi.org/10.1108/tqm-06-2024-0228

Neri, G. R., Marchand, R., & Walkinshaw, N. (2025). Exploratory Software Testing In Scrum: A Qualitative Study. 160–175. https://doi.org/10.1007/978-3-031-94544-1_11

Orlov, E., Rogulenko, T. M., Smolyakov, O. A., Oshovskaya, N., Zvorykina, T. I., Rostanets, V. G., & Dyundik, E. P. (2021). Comparative Analysis of the Use of Kanban and Scrum Methodologies in IT Projects. Universal Journal of Accounting and Finance, 9(4), 693–700. https://doi.org/10.13189/ujaf.2021.090415

Ozkan, N., Bal, S., Erdoğan, T. G., & Gök, M. Ş. (2022). Scrum, Kanban or a Mix of Both? A Systematic Literature Review. 30, 883–893. https://doi.org/10.15439/2022f143

Perlak, J. (2019). Characteristics of Self-Organizing Teams in Agile Project Management: A Case Study. Acta Universitatis Nicolai Copernici Zarządzanie, 46(1), 19. https://doi.org/10.12775/aunc_zarz.2019.002

Rathor, S., Xia, W., & Batra, D. (2023). Achieving Software Development Agility: Different Roles of Team, Methodological and Process Factors. Information Technology and People, 37(2), 835–873. https://doi.org/10.1108/itp-10-2021-0832

Salameh, A., & Bass, J. M. (2019). Spotify Tailoring for Promoting Effectiveness in Cross-Functional Autonomous Squads. 20–28. https://doi.org/10.1007/978-3-030-30126-2_3

Saltz, J., & Sutherland, A. (2020). SKI: A New Agile Framework That Supports DevOps, Continuous Delivery, and Lean Hypothesis Testing. https://doi.org/10.24251/hicss.2020.761

Santos, L., Garaffa, R., Lucena, A. F., & Szklo, A. (2018). Impacts of Carbon Pricing on Brazilian Industry: Domestic Vulnerability and International Trade Exposure. Sustainability, 10(7), 2390. https://doi.org/10.3390/su10072390

Scharold, F., Schrof, J., & Paetzold-Byhain, K. (2023). Scrum-Based Agile Maturity Assessment in Physical Product Development. https://doi.org/10.35199/dfx2023.04

Scott, E., Milani, F., Kilu, E., & Pfahl, D. (2021). Enhancing Agile Software Development in the Banking Sector—A Comprehensive Case Study at LHV. Journal of Software Evolution and Process, 33(7). https://doi.org/10.1002/smr.2363

Senapati, A., Bhattacherjee, A., & Chau, N. (2020). Associations of Job-Related Hazards and Personal Factors With Occupational Injuries at Continuous Miner Worksites in Underground Coal Mines: A Matched Case-Control Study in Indian Coal Mine Workers. Industrial Health, 58(4), 306–317. https://doi.org/10.2486/indhealth.2019-0102

Trihardianingsih, L., Istighosah, M., Alin, A. Y., & Asgar, M. R. G. (2023). Systematic Literature Review of Trend and Characteristic Agile Model. Jurnal Teknik Informatika, 16(1), 45–57. https://doi.org/10.15408/jti.v16i1.28995

Wahab, A. M. A., Dorasamy, M., & Ahmad, A. A. (2024). Product Team in Transition: A Qualitative Case Study of Team Motivation and Collaboration During Agile Adaptation. International Journal of Management Finance and Accounting, 5(2), 50–74. https://doi.org/10.33093/ijomfa.2024.5.2.3

Клоков, В. Н., Вечерская, С. Е., Ивлеев, М. С., & Голубев, М. Б. (2024). Improving Development Team Efficiency With the Agile-Maturity Method. Vestnik of Russian New University Series «complex Systems Models Analysis Management», 2, 60–67. https://doi.org/10.18137/rnu.v9187.24.02.p.60