



A Scrumban Integrated Approach to Improve Software Development Process and Product Delivery

 Sai Nikhil Donthi

Department of Software Engineering, University of Houston Clear Lake, LTIMindtree, Client – Oil and Gas Industry Houston, Texas

Email ID - donthisainikhil@gmail.com

OPEN ACCESS

SUBMITTED 19 August 2025
ACCEPTED 26 August 2025
PUBLISHED 26 September 2025
VOLUME Vol.07 Issue 09 2025

CITATION

Sai nikhil. (2025). A Scrumban Integrated Approach to Improve Software Development Process and Product Delivery. The American Journal of Interdisciplinary Innovations and Research, 7(09), 70–82. <https://doi.org/10.37547/tajir/Volume07Issue09-07>

COPYRIGHT

© 2025 Original content from this work may be used under the terms of the creative common's attributes 4.0 License.

Abstract- Scrum and Kanban have been the two traditional Agile practices which has been depicted as successful in enhancing software development processes, while each has its own drawbacks when implemented in software IT industry and put into practice individually. Scrum enforces sprint, roles, ceremonies, and load calculation which can occasionally result in overhead and rigidity. Although Kanban is flexible and having constant flow, it might not include task batching or structured prioritization. The research challenge of optimising the productivity of the software development lifecycle by integrating both of these strategies into a hybrid framework called the Scrumban. Methodologically, it synthesizes findings from industry case studies, scholarly literature, and real-world implementation addressing workflow visualization, work-in-progress (WIP) limits, sprint process and team role definitions. The study explores how Scrumban mitigates the drawbacks of its earlier models with increasing productivity, resource use, and deliver quality. Scrumban will serve as a hybrid framework to enhance the entire software development lifecycle, as evidenced by real-world use cases and KPIs such as execution time and defect reduction. Significant findings indicate that Scrum ban enhances adaptation in unpredictable circumstances, helps with continuous delivery, and supports cross-functional collaboration. Scrumban makes use of Kanban's flow productivity and

Scrum's structured planning to minimize redundant work, fix bottlenecks, while enhancing transparency. Use cases from enterprise applications, promotional platforms, and incident reporting applications prove Scrumban's scalability and savings potential. This research determines its findings that Scrum ban is suitable to organizations transitioning from traditional Agile approaches because it is a complete approach that supports both shifting business needs and developments in technology.

Keywords: Scrum, Kanban, Scrumban, productivity, sprint, LSD, SDLC, flexibility, WIP limits, continuous delivery, devops integration, Agile, KPI analysis, iterative development, workflow visualization, productivity metrics.

1. Introduction

The Agile approach is the combination of software development methodologies that resemble the Agile Manifesto's four objectives and twelve principles [2]. We can swiftly build a software application using the agile model in a short amount of time.

Scrum and Kanban methodologies were designed based on these objectives and concepts [2]. Individuals and interactions, working software, customer collaboration, and adapting to change are all highlighted in the values

[2]. The concepts encapsulated in Organisation's manifesto are refined into 12 principles [2], including the following instances of the principles [2]:

- a. During the project implementations, customers and developers/project team must collaborate on a daily basis.
- b. Agile processes are catered towards projects that involve collaboration, both in terms of work as well as the process being followed.
- c. The most well-organized and productive technique of passing on information between deployment and software development teams is face-to-face communication.
- d. The easiest method to track performance is to use working software that quantifies commercial value.
- e. Continuing a consistent cadence allows for long-term production and development.

Scrum is one of the first Agile methodologies. When companies initially move to an Agile strategy, it is one of the most commonly used Agile approaches. Figure 1 clearly demonstrates the agile process workflow in an organization.[3]



Fig. Agile Model

Figure 1: Agile methodology structure

Kanban, on the other hand, is a secondary rendition of the Agile method. This strategy is widely adopted by organizations all over the world. Hundreds of companies globally have adopted both methodologies. Agile teams, on the other hand, believe that there is neither Scrum or Kanban methodology does not for all projects. As a result, a comparison is required because they both focus on project management.

Scrum is a collection of procedures widely used in the software industry. The Scrum team consists of three major roles which are Product Owner (PO), a Scrum Master (SM), and a Development Team [1]. The Scrum Teams follow a well-defined process which are highlighted by four major elements such as: a Product Backlog, a Sprint Backlog, a Product Increment, and Acceptance criteria. During a Scrum sprint, the team is expected to be involved and accomplish Backlog Refinement, Sprint Planning, Daily Scrum meetings, Sprint Reviews, and Retrospective [3].

Kanban is considered to be flexible compared to the Scrum methodology because it entails a less detailed process and practices. Though Kanban is considered to be less detailed, it is considered to be the most powerful methodology since it addresses the issues that Scrum methodology is unable to solve. Also, it proves to be more effective for organizations to ensure that the deliverables meet the business values assigned. The main practices followed during the implementation of Kanban methodology are to utilise the physical Kanban board to track the team's workflow, actively reducing unused and inactive features, and implement continuous feedback to improve team collaboration.

This research will further elaborate on more details regarding each of the methodology (Scrum Vs Kanban) and identify the benefits and fallbacks of integrating these Agile methodologies into software development projects. Additionally, the research will also detail the similarities and differences between them and how they could be combined together as a hybrid process for a particular software project.

This paper also focuses on issues in existing Agile frameworks related to scalability, control, and a deficient structure. The challenges of role definition, performance evaluation, and adaptability are not effectively addressed by the hybrid models that are currently in practice. By putting forth the Scrumban

model as an integrated system that leverages the advantages of Scrum as well as Kanban with mitigating their drawbacks, this study is dedicated to address these gaps. its scalability in business applications, as well as how considerably it enhances workflow visualization, optimizes the best use of work-in-progress limitations, and enhances job flexibility, are all evaluated in this study. The paper also focuses at how Agile approaches might be improved with the help of AI-based project management tools.

2. Scrum – Agile Methodology

Scrum and the Scrum Manifesto were initially created by Ken Schwaber and Jeff Sutherland. According to the researchers, Scrum is "a framework that enables developers to address sophisticated compatibility challenges, effectively and efficiently deliver high quality products" to the customers [2]. All software methodologies have a best fit and suits its particular purpose. For example, a traditional software methodology like waterfall is geared towards smaller applications that have a precise and finalised requirement. Waterfall discourages constant changes to the customer requirements as the development life cycle follows a sequential process and makes it difficult to incorporate impulsive changes. Agile methodology takes into account such challenges to the change in requirement and has introduced an iterative process. This allows both small as well as large scale corporations to implement complex projects with freedom for periodic changes. The organizational structure of a Scrum project consists of predefined roles which includes a Scrum Master who ensures that the Scrum framework is followed, a Product Owner who is the business stakeholder of the project and a Scrum Team which consists of a combination of Software Developers, Quality Analyst and Business Analyst. The entire project requirement is divided into features and each feature is split into multiple stories. Each scrum team takes ownership of the stories based on the customer priority and delivers the work at the end of each sprint.

A Sprint is defined as a short period of time, usually 2 to 4 weeks as decided by the team, to complete the assigned stories. A group of sprints is considered to be a planned increment which often spans about 3 Bmonths. Below are scrum events which are regularly followed by the scrum teams.

A detailed workflow diagram of Scrum is shown in figure 2. It shows the complete process end to end followed in agile scrum from sprint planning to sprint review.

- a. **Sprint:** Sprint is a short period of time that has been agreed upon to complete the assigned work as part of the stories.
- b. **Sprint Planning:** An event prior to starting the sprint where the team determines the definition of Done for each particular task/story. The product owner provides the team with the objective of the sprint as well as reviews the backlog of the team.
- c. **Daily scrum meeting:** A daily event that allows

all the team members to discuss their progress, impediments and dependencies.

- d. **Sprint review:** The Scrum team provides the business stakeholders an update on the progress of the implementation completed during the most recent sprint and while also providing a platform for the customers to provide constructive feedback.
- e. **Sprint retrospective:** An event where the scrum team along with the product owner and scrum master highlight the positive and negative practices from the sprint and the processes to be followed in order to identify the areas for improvement.

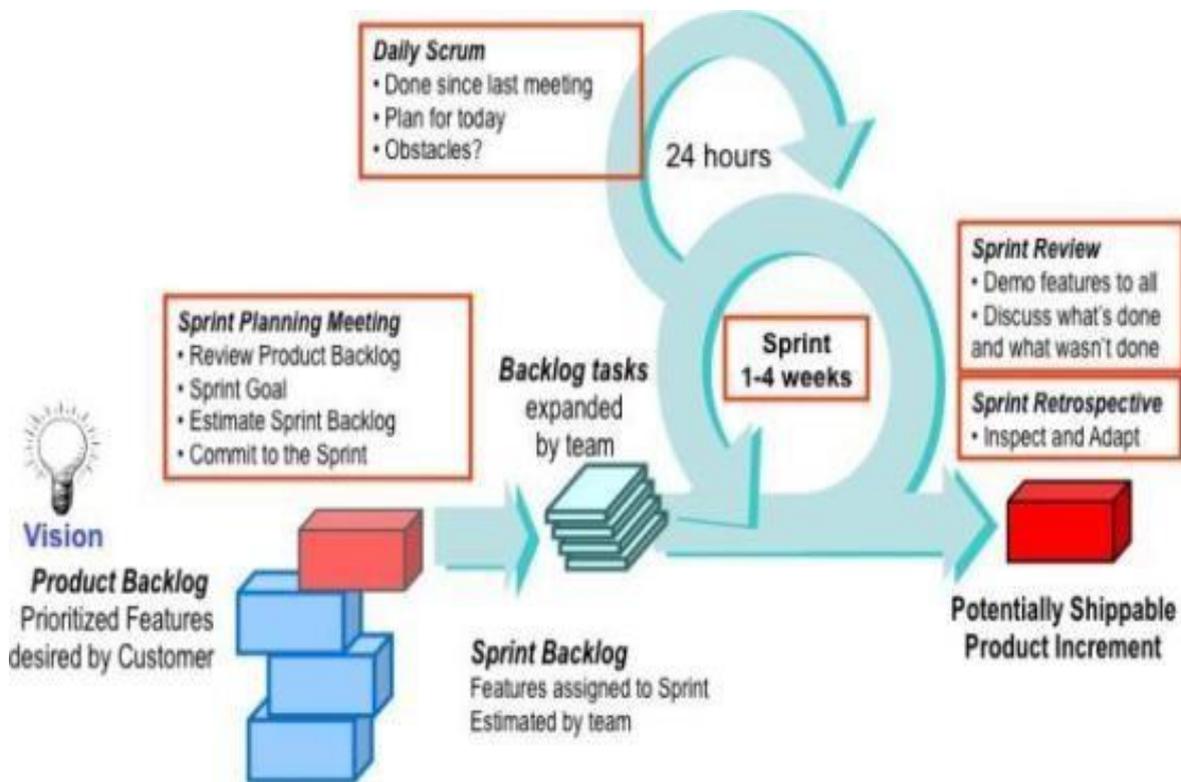


Figure 2: Workflow of a SCRUM team

2.1 Advantages of using Scrum:

1. Scrum helps the team to deliver the work quickly and consistently, and it ensures that time and resources are used adequately.
2. Large projects/client requirements are broken down into multiple manageable sprints.
3. Scrum meetings are an integral activity for the teams to get a clear perspective of the requirements.
4. During the sprint review meeting, the team can provide an update on the accomplished task/work and get the client feedback for product enhancement.

5. The entire development workflow is accomplished during a sprint by completing the coding as well as any testing required.

6. Scrum development shrewdly maintains a balance between delivering quantity while following the key principle of quality assurance. Every sprint in scrum involves testing the product which enables regular inspection of the development which aids the team to identify any quality issues, requirement gaps and allows early adjustments to the product.

2.2 Disadvantages of using Scrum:

1. The Scrum board does not provide an option to view the long-term goals of the team. Scrum provides a short-term snapshot of the work within the scrum board. Due to this, the team does not frequently review long-term goals as well as project status as they are focused on the particular Sprint boards that only provide status on the current team work.

2. The Agile methodology provides the Scrum team and its members the freedom and independence from the rigid structure of other methodologies. While this freedom can be refreshing, the short-term nature of Agile can lead to teams being side tracked. The lack of structured vision and planning along with final product documentation allows an opportunity for scope creep and change in requirements.

3. As Scrum is a fast iterative process that is broken down into short sprints, there isn't an opportunity to eliminate waste and refine the features or planned work to acquire maximum efficiency due to inflexible sprint structure.

4. As Scrum does not follow a traditional visualization platform, Bottlenecks are difficult to identify until a Scrum Review occurs.

2.3 Trade-off /Scalability challenges of using Scrum:

Scrum provides structured and predictability through defined roles and sprint cycles at the expense of flexibility. Mid-sprint changes are discouraged, and planning overhead can slow rapid experimenting work. This could lead to choosing quantity over quality and limit the adaptability in fast changing environments.

Adapting Scrum over large and hybrid teams becomes

“Lean software development has seven principles as the following: “Eliminate waste”, “Amplify learning”, “Decides as late as possible”, “Deliver fast as possible”, “Empowering the team”, “Build integrity in”, “See the whole” all of them can be applied in the field of software development” [10].

Drawbacks & Limitations of Lean Software Development:

1. Lack of formal structure in LSD unlike SDLC models results in ambiguity in decision making when

extremely difficult where the development teams, inhouse teams and operations team need to coordinate and work to achieve the application stability. It becomes difficult to clarify roles with multiple leads for different teams and also it becomes complex to conduct meetings and stand- ups.

2.4 Tools Used in Scrum:

Tool Implementation: Backlog management, sprint tracking, and burndown charts are all powered by Jira and Azure DevOps. However, the rationale of Scrum could be reduced by excessive customization and inadequate data quality.

3. Lean Software Development:

Lean software development (LSD) is an agile methodology which helps to increase the customer value by eliminating waste resources that are not adding any value to the project/product. This process helps to reduce the project life cycle and focuses on achieving the following:

- The iterative development process followed with the Lean approach provides an ability to detect and prevent errors early in the life cycle.
- Integrating automation through continuous integration and development improves the build and deploy process while eliminating dependencies on people.
- CI/CD allows detection of defects automatically while ensuring that project expectations are met on time.
- Identify vulnerabilities and security violations to ensure that the integrity of the application is intact while also delivering the project goals.

opted in large teams.

2. Lean focuses on less documentation which could be a challenge when considering long term maintenance projects and during onboarding of new team members.

3. LSD fits best for small enterprise applications as scalability could be a challenge when adopted for complex enterprise applications with large teams involved as it becomes difficult to identify waste.

4. There is a risk of over simplification as teams can skip steps like architectural review, stakeholder engagement and testing targeting speedy deliver and minimalism.

5. Shift in Cultural resistance could be challenge if adopted in organisations that have adopted SDLC from many years.

4. Kanban:

Kanban means "Visual Signal" in Japanese. Kanban is an agile practice that is aimed to help teams work more efficiently together. In Kanban when the current job is completed, the Kanban team moves on to the next task. Kanban works best for production support, ad hoc requests, and unanticipated tasks. Kanban assists in identifying bottlenecks and visualizing workflow. Figure 3 showcases the Kanban workflow in an organization when adopted.

KANBAN WORKFLOW



Figure 3: Workflow of a KANBAN

Principles of Kanban:

1. Visualize workflow through Kanban board.
2. Limit the amount of work in progress.
3. Enhanced flow of work – When one task or story is finished then only the next high priority task or story is picked to work upon.

4.1 Advantages of using Kanban

1. Helps in understanding the clear view of time that was spent on the project.
2. The Kanban system is very flexible as it has no particular duration of time for each phase and works on tasks which are of high priority and the changes are constantly reassessed based on most recent information.

3. Kanban focuses on continuous development as teams working on Kanban deliver small portions of the project continuously to the client.
4. Teams working on Kanban have multiple opportunities to sync with updated business requirements easily.
5. Kanban Teams work to increase productivity. A well designed Kanban system tries to eliminate waste in all of its forms.
6. Kanban helps in reducing duplicated effort, over processing.
7. It helps team members focus on their work when things are properly prioritised and arranged and let the team members decide the next high priority ticket.

8. Kanban also ensures increased efficiency by eliminating waste and helping team members focus on work.

9. Lead time for the project is kept minimum in Kanban project.

10. It reduces the total cost of the project by eliminating waste and overproduction.

4.2 Disadvantages of using Kanban:

1. Kanban is less effective if the project makes use of shared resources.

2. Kanban assumes stable recursive production plans and may cause demand surge problems.

3. As Kanban does not eliminate variability, the system may be disrupted due to unpredictable changes and lengthy downtimes in the project.

4. The Kanban system is not designed for projects with short production runs, highly variable product demand and poor quality products.

5. Breakdown at a point may result in the entire line being shut down in the Kanban project.

4.3 Trade-off /Scalability challenges of using Kanban:

Trade-offs: Kanban offers flexibility and visual workflow management, but it does not have formal roles or rituals. This could lead to unclear responsibilities and less control.

Scalability Issues: When implementing Kanban in teams of multiple people, precise responsibility management, demand modelling, and prioritizing capabilities are crucial. Lack of organized coordination could affect operational efficiency and prioritization.

5. Scrumban

Scrum and Kanban are the most commonly used methodologies for project management. In the current software development world, the best approach used by several companies is a combination of Scrum, Lean and

Kanban methodology known as Scrumban.

Scrumban provides a well-balanced approach that is flexible for both short- and long-term projects that are hamstrung by the limitations of both methodologies. This balance proves to be a great fit for fast paced and iterative projects as well as large- scale long-term projects. Figure 4 clearly depicts the workflow of Scrumban With WIP limitations and role interactions included across the board.

5.1 Usability of Scrumban:

Similar to Scrum methodology, Scrumban works in iteration of two – four weeks and inspired from the Kanban board, the entire team sprint work is visualized on the Scrumban board. Moreover, the Scrumban board provides a clear visualization of the work and allows flexibility to be integrated in the workflow. The process of pulling work, limiting Work in progress (WIP), and continuous workflows are all adopted from the Kanban methodology.

Multiple organizations have benefited by combining Scrum and Kanban methodologies. The Kanban team is meeting daily to provide the status and update the impediments of the project, which is usually a Scrum practice. Similarly, there have been examples within organizations where scrum teams are limiting the size of their task per column, which is usually considered as a Kanban practice.

The Scrumban methodology has seamlessly combined the major practices of both Scrum and Kanban in that the projects under Scrumban are pulling the needed work, utilising the Kanban board to measure the lead time, while tracking a product backlog similar to scrum, and limiting work in progress. Additionally, these projects have also integrated scrum practices such as daily stand-ups, retrospective while having well defined roles of the scrum team members consisting of Scrum Master, Product Owner and the development team.

5.2 Scrumban workflow:

SCRUMBAN DUAL-BOARD SYSTEM

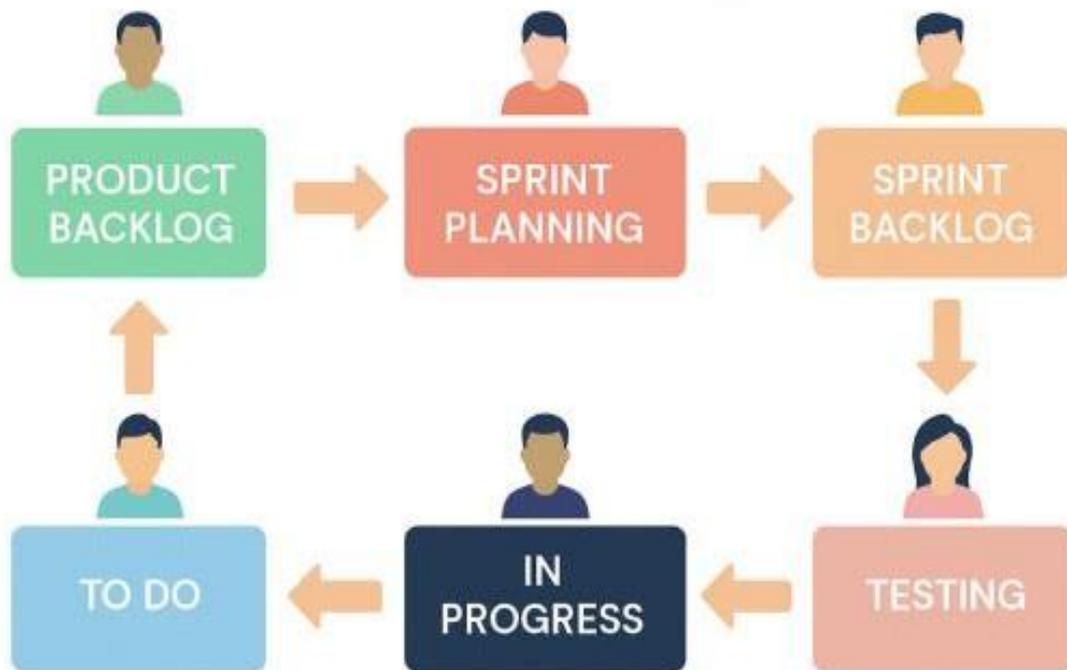


Figure 4: Workflow of a SCRUMBAN

This diagram figure 4 demonstrates how Scrum's structured planning (Product Backlog → Sprint Planning → Sprint Backlog) flows into the Kanban's continuous delivery stages (To Do → In Progress → Testing). It aims to demonstrate how Scrumban manages to balance between flexibility and scrum discipline.

The Scrumban approach follows the Scrum methodology of sprints/iterations implementation. The process is initiated by the Product Owner, who holds the responsibility of gathering the business requirements and prioritising the work based on business value to the customer. The next step in this process involves a discussion with the entire team to determine and eliminate waste from the product backlog.

In the Scrumban process, the team uses two types of boards which are General board and Scrumban board. Product owner is the responsible person to manage the general board to track the complete project status such as in-progress work, product backlog and completed work. Team lead and developers have the access to manage the Scrumban board which contains the information regarding the current sprint status.

1. Initially the product owner places the backlog stories under the TO DO column in the general board and the team begins by pulling the stories from the

product backlog into the current sprint. Then, the team starts placing the stories into the Ready column in the Scrumban board.

2. In the general board, the Product owner specifies the Sprint and task details of the team in the DOING column. The team begins working on the stories, which must be noted in the DOING column on the team Scrumban board.

3. Once the implementation of the task is completed then the team moves the respective story into the TESTING column on the Scrumban board. After successfully completing the testing of the story, the team moves the story into the DONE column on the Scrumban board.

4. Once the story is completed by the team then the product owner moves the task to the DONE column on the general board and gets the feedback from the customers for the completed task.

5. Once features are delivered, the product owner moves the task to the Delivered column on the general board.

The Scrumban team usually follows the same process for each sprint. Moreover, Scrumban methodology contains six meetings which are "Eliminate-waste meeting, Sprint

Planning, Daily meeting, Quality-Test meeting, Sprint Review, and Sprint Retrospective.” [10].

5.3 Scrumban Advantages:

1. It is time saving. In Scrumban, there is no need to estimate or plan the sprint as the team's plan where there is a demand.
2. Scrumban minimizes waste and eliminates whatever is not necessary to the customers. Scrumban does not limit the work in progress by time; instead it limits work at each stage.
4. In scrum ban, teams can bring onboard stories of various sizes and progress of each story is known at various stages of work.
5. In Scrumban, level of productivity is defined using lead time and cycle time.
6. As Scrumban uses a pull system to address technical tasks, team foundation grows stronger and the team building process is accelerated.
7. Facts and decisions are made just in time whenever required.
8. Scrumban supports continuous development similar to Kanban.
9. Scrumban ensures high quality products as the time that is saved helps us to focus on quality control.
10. Scrumban helps in process improvement by adding some scrum values as needed.
11. Team members will have perfect visualization of scrumban boards.
12. Scrumban combines the flexibility, adaptability, and monitoring ability of scrum and Kanban to ease the work.
13. Scrumban identifies and reduces bottlenecks.

5.4 Scrumban disadvantages:

1. There are no daily scrum meetings to give project managers information regarding the progress of the work and proper formal roles.
2. Has little documentation regarding the project changes.
3. Teams might struggle for planning and prioritization and it also lacks scaling across multiple teams.

4. It could lead to risk of over flexibility allowing end users request features that could disrupt application flow and may lead to rework.

5.5 Trade-off /Scalability challenges of using Scrumban:

Trade-offs: Scrumban integrates structure and flexibility, but it lags role definition and systematic objectivity. Teams could struggle with unclear ownership and unpredictable planning schedules as a consequence of lacking guidance.

Scalability Issues: Scrumban requires a defined framework for scalability, despite its flexibility. Incompatible processes, inconsistent WIP enforcement, and challenges aligning KPIs like cycle time and velocity could all be implications of multi-team coordination.

5.6 Mitigation Strategies of Scrumban:

1. As Scrumban does not have formal roles prescribed, team can decide and assign responsibility to ensure clarity and ownership of delegated tasks.
2. Establishing work in progress limits and cycle time to check bottlenecks and maintain the flow.
3. Set recurring planning meetings to check the progress and focus on the end objective.
4. Create small documentation to ensure new team members onboarded can refer to incase of bandwidth issues and new team members are hired to the team.

6. Use Case of Scrumban:

Scrumban is exclusively used in the development of applications that both structure and flexibility. Scrumban can be implemented in developing systems that involve change with changing marketing conditions and rapidly changing scope of work.

Scrumban benefits operational support projects and run and maintain projects which needs to address unforeseen issues and market conditions. For example , T-Mobile life application keeps updating its promotions on weekly basis offering users with wide variety of promotional discounts on every Tuesday and such kind of projects while Scrumban is implemented in managing the development team of this application can provide rapid results as required.

Similarly many companies have enterprise applications

that manage their employee business travels and incident reporting tools on their internal websites that report many major incidents to make sure they travel to safe locations reviewing the travel conditions of that country for which their business travel is planned. Such applications or tools implementing scrumban can benefit a lot and save on operational costs.

Also applications involving development related to marketing, design and content creators often utilize Scrumban to visualizing work dashboards and plan creative works.

Scrumban is also perfect for application development of

projects with huge backlog to complete and prioritize the work items based on business requirement to save business costs. IT run and maintain operations teams of such projects and devops team utilization of scrumban helps prioritise work and review which adds most benefit and helps in saving costs.

7. Comparison of Scrum, Kanban and Scrumban

Below table 1 clearly shows the advantages of Scrumban comparing each industry standard dimension that helps to make a decision in choosing the right agile practice to add value to the project based on the project size, team size and other factors.

Table 1: comparison of scrum, Kanban and Scrumban

Dimension	Scrum	Kanban	Scrumban
Development Approach	Iterative, time- boxed sprints	Continuou s flow	Hybrid of Scrum’s structure and Kanban’s flow
Planning Strategy	Formal sprint planning before each cycle	No formal planning; tasks pulled as needed	Minimal planning; pull-based prioritization
Team Roles	Defined roles: Scrum Master, Product Owner, Development Team	No prescribed roles	Roles adapted from Scrum; flexible assignment
Workflow Visualization	Sprint- based Scrum board	Stage- based Kanban board	Dual-board system with sprint and flow stages
Work-in- Progress (WIP)	Not enforced	Strict WIP limits per workflow stage	WIP limits applied to optimize throughput
Cadence & Delivery	Fixed- length sprints (e.g., 2–4 weeks)	Continuou s delivery	Continuous with optional sprint-like cycles
Performance Metrics	Velocity, burndown charts	Lead time, cycle time	Combines velocity, lead time, and cycle time
Change Management	Changes discourag	Changes allowed anytime	Flexible change managemen

Dimension	Scrum	Kanban	Scrumban
	ed mid- sprint		t with structural boundaries
Best Fit For	Teams with stable priorities and defined goals	Teams needing flexibility and rapid response	Teams transitionin g from Scrum to Kanban; projects with evolving requirement s
Ceremonies & Meetings	Daily standups, sprint planning, reviews, retrospect ives	Optional standups; no formal ceremonie s	Lightweight ceremonies: includes retrospectiv es and planning as needed
Scalability & Adaptability	Moderate scalability ; requires role discipline	High adaptabilit y; less scalable across large teams	Scalable across teams; adaptable to changing business and technical contexts

8. Analysis - Scrumban compared to previous hybrid agile models:

1. Scrumban offers integrated KPI framework by integrating its practices with quantitative performance metrics like cycle time, delivery time and defect reduction which has been bypassed in hybrid models and provides subjective benefits.
2. Scrumban offers Dual-Board Workflow Visualization unlike earlier hybrid models This analysis clearly defined dual-board approach that separates strategic planning (general board) from operational execution (Scrumban board) that enhances role transparency and clarity, especially in multi-team development life cycles.
3. Previous models like agile concurrent hybrid models and blended scrum provides role flexibility. Analysis proves that Scrumban offers practical balance following discipline and Kanban autonomous flexibility indicating support for team adaptability.
4. Analysis drawn from above research suggests Scrumban approach addresses scaling challenges dynamically and best suitable for devops, operations support and non-technical teams.
5. Analysis derives that Scrumban provides continuous improvement with regular retrospectives, optimizing on-going processes which earlier hybrid models lacked.

6. Scrumban unlike other hybrid models provides on-demand planning analysing the team’s capacity which reduces burden and boosts adaptability.

7. This analysis also proposes AI-based enhancement readiness of Scrumban for a future ready integration potential.

8. Use case analysis readiness with real time applications inclusion is rarely explored in earlier hybrid models which has been addressed in this paper.

9. This study indicates Scrumban aligns with Lean principles and helps in boosting delivery outcomes.

9. Conclusion

In Conclusion, the Scrumban methodology which is combination of both Agile scrum and Kanban provides a path forward overcoming the drawbacks and limitations of individual framework. Scrumban provides a balanced approach adhering to shifts in industry standards and business needs combining Agile scrum’s iterative sprint planning, structured roles and quality assurance with Kanban Visual workflow, flexibility and continuous delivery. Scrumban helps the development teams with the ability to quickly adjust with business needs while reducing unwanted work and increasing productivity.

Scrumban also helps to maintain high qualified industry standard developing quality application or product. Scrumban implementation puts its efficiency to the higher level where the products need active

development and where this a large scope of development in long term. In fast-paced application development, Scrum ban's pull based workflow and dual board structure offer complete

visibility and control of development process and improves team collaboration. Scrumban also ease the run and maintain operations of the application and performance that result in complete customer satisfaction in today's agile driven landscape.

Despite Scrumban's great potential for maintaining a balance between structure and flexibility, this study is primarily conceptual and is supported up by qualitative operational instances. Experimental evaluation in multiple project structures should be a part of future study to evaluate performance measures like team velocity, lead time, and the defect rate. Continued research is needed to tackle coordination and clarity of responsibility concerns that occur while Scrumban becomes widespread in large enterprise ecosystems. Integrating AI-based project management tools, like intelligent process automation or predictive backlog prioritization, might enhance the Scrumban's adaptability and decision-making skills.

10. References

1. Iqudah, M., & Razali, R. (2017). A comparison of scrum and Kanban for identifying their selection factors. 2017 IEEE International Conference on Environment and Electrical Engineering (EEEIC), 1–6. <https://doi.org/10.1109/ICEEI.2017.8312434>
2. Hema, Thota, S., Naresh Kumar, S., Padmaja, C., Rama Krishna, C. B., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. IOP Conference Series. Materials Science and Engineering, 981(2). <https://doi.org/10.1088/1757-899X/981/2/022060>
3. Agh, & Ramsin, R. (2021). Scrum metaprocess: a process line approach for customizing Scrum. Software Quality Journal, 29(2), 337–379. <https://doi.org/10.1007/s11219-021-09551-4>
4. Al-Zewairi, Biltawi, M., Etaiwi, W., & Shaout, A. (2018). Agile Software Development Methodologies: Survey of Surveys. Estonian Journal of Earth Sciences, 67(3), 74–. <https://doi.org/10.4236/jcc.2017.55007>
5. Sultania. (2015). Developing software product and test automation software using Agile methodology. Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), 1–4. <https://doi.org/10.1109/C3IT.2015.7060120>
6. Çetin, & Onay Durdu, P. (2019). Blended Scrum model for software development organizations. Journal of Software: Evolution and Process, 31(2), e2147–n/a. <https://doi.org/10.1002/smr.2147>
7. Žužek, Kušar, J., Rihar, L., & Berlec, T. (2020). Agile-Concurrent hybrid: A framework for concurrent product development using Scrum. Concurrent Engineering, Research and Applications, 28(4), 255–264. <https://doi.org/10.1177/1063293X20958541>
8. Adam Cetnerowski. (2015). The Scrumban [R] Evolution: Getting the Most Out of Agile, Serum, and Lean Kanban [Review of The Scrumban [R] Evolution: Getting the Most Out of Agile, Serum, and Lean Kanban]. Software Quality Professional, 18(1), 34–. American Society for Quality.
9. Schaefer, & Söllner, D. (2017). DevOps by Scrumban. HMD practice in business informatics, 54(2), 251–260. <https://doi.org/10.1365/s40702-017-0301-x>
10. Abdullah Albarq, & Qureshi, R. (2018). The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development. International Journal of Information Engineering and Electronic Business, 10(3), 23–35. <https://doi.org/10.5815/ijieeb.2018.03.04>
11. STOICA, GHILIC-MICU, B., MIRCEA, M., & USCATU, C. (2016). Analyzing Agile Development – from Waterfall Style to Scrumban. Informatica Economica, 20(4/2016), 5–14. <https://doi.org/10.12948/issn14531305/20.4.2016.01>
12. Law, & Lárusdóttir, M. K. (2015). Whose Experience Do We Care About? Analysis of the Fitness of Scrum and Kanban to User Experience. International Journal of Human-Computer Interaction, 31(9), 584–602. <https://doi.org/10.1080/10447318.2015.1065693>

13. Lynn, R. (2021, October 13). Kanban system: Maximize time and improve efficiency. Planview. Retrieved November 21, 2021, from <https://www.planview.com/resources/guide/introduction-to-kanban/kanban-system-maximize-time/>
14. Patil, & Neve, J. R. (2018). Productivity Improvement of Software Development Process Through Scrumban: A Practitioner's Approach. 2018 International Conference on Advances in Communication and Computing Technology (ICACCT), 314–318. <https://doi.org/10.1109/ICACCT.2018.8529405>