Scrumban: An Agile Integration of Scrum and Kanban in Software Engineering

Krunal Bhavsar, Vrutik Shah, Samir Gopalan

Abstract: ***Software Engineering (SE) technologies are emerging day by day and seeking ABPR (Agile Business Process Reengineering) for Software Engineering Management (SEM) frameworks in software development organizations. BPR can enable organizational capabilities to initiate and implement critical change in execution. Under the roof of agile and on the base of empiricism, the Scrum has been proven itself as an ABPR approach for software engineering management in the software development organizations, across the world, by improving productivity, self-organization and collaboration for standard software development processes. Scrum has been leading software project development practices using its own characteristics: Artifacts, Pillars, Values, Events and Roles. But still software project development organizations are facing some issues with their software project development and management processes, like no documentation policy which results into inaccurate estimation, internal states of each work item flowing through the Scrum Board, Sprint tracker and prediction towards goal achievement which can be considered as challenges for Scrum due its limitations as well as Scrum does not allow, any alteration in its rules. Such issues have raised a question against implementation of Scrum and have opened the door for the next level of research to answer the question, how to overcome the limitations of Scrum. Kanban can provide the solution to some of these issues but it can't provide complete SEM solutions to software Development Organizations. The aim of this research study is empirical analysis about how the formation of hybrid framework Scrumban as an integration of Scrum with Kanban, can resolve challenges of Scrum; using literature reviews, case study reviews, and research surveys; and this research has proposed a conceptual customized Scrumban framework, by keeping Kanban in the center of customization under the bound of Scrum rules. This research also concludes limitations of Scrumban, like project documentation, planning, large scaled project, distributed environment, team capabilities etc.; as each method has its own limitations.***

Keywords: ***Scrum, Kanban, Scrumban, AM - Agile Methodology, ABPR – Agile Business Process Reengineering, BPR - Business Process Reengineering, ESE – Empirical Software Engineering, SPI – Software Process Improvement, SEM – Software Engineering Management, SE – Software Engineering***

Revised Manuscript Received on February 28, 2020.
* Correspondence Author

Krunal Bhavsar*, Research Scholar, Computer Science & Engineering, Indus University, Ahmedabad, India. Email: krunalbhavsar@engineer.com; krunalbhavsar.rs@indusuni.ac.in. Contact: +91-9737007007.

Dr. Vrutik Shah, Research Guide, Computer Science & Engineering, Indus University, Ahmedabad, India.

Dr. Samir Gopalan, Research Co-Guide, Business Administration & Management, Indus University, Ahmedabad, India.

I. INTRODUCTION

Scrumban is an agile integration of Scrum and Kanban frameworks under the shelter of agile manifesto principles and on the base of empiricism. Software development methodologies are seeking Agile Business Process Reengineering (ABPR) by aligning customer values in the center of all the practices and the framework policies should be flexible, which can support, according to changing needs and expectations of customer [15]. With the intention of revolution in Traditional Software Development practices, due to their limitations [36], the incremental and iterative software development approaches were introduced in 1990, called Agile Methodology (AM) and later on integrated with the principles of Agile Manifesto [10] with the help of Agile methodologist, in 2001.

The concepts of Agile Manifesto are very simple and easy to understand but very complex to implement and execute them in real world. Various Agile methodologies are in existence nowadays and they are all having different characteristics which can be applied on various types of project development requirement. Scrum and Kanban are most adopted agile based frameworks, amongst all, in the software development domains.

But each method has their own limitations and the similar phenomenon is applicable on Scrum and Kanban also. Scrum is very popular amongst all Agile based frameworks, in between professional agile practitioners across the world [13], [14]. But Scrum does not specify explicit policies about workflow management and how the workflow item will pass through each state which can enable visualization of internal workflow.

Kanban [1] is built for the workflow management by visualizing each state of work item flowing passing through its life cycle, which can enable explicit transparency using Kanban board. Agile practitioners have been looking for a common methodology which can be applied on the development areas, to fulfill their purposes, to achieve desired goal. Since a decade so many experiments were performed to improve the capabilities of Software Engineering Management (SEM), by focusing product quality, customer values, team collaboration [37], [38] as a goal of productivity enhancement. With the belief that Scrumban can be more effective and beneficial for professional software practitioners of Scrum and Kanban, the community of Scrum and Kanban leaders have conceptualized and convinced for the formation of Scrumban by combining Scrum and Kanban with the aim of process transparency and productivity improvement for software product development. Scrumban derives some set of useful practices from Scrum and Kanban; and constitutes a robust framework that revolves agile transformations to address unpredictable problems also. Scrum has capabilities to address complex challenges; while Kanban is very simple and not rigid and can be fit with the Scrum rule and regulations [20]. Kanban has strength to address some of the rigid issues of Scrum. And these are main reasons behind selecting combination of Scrum and Kanban, amongst all agile frameworks. Now it's primary concern for this research that how to select features and services from Scrum and Kanban which builds robust Scrumban framework. Adoption and diversification from existing practices to the proposed structure leads towards ABPR (Agile Business Process Reengineering) for a software development organizations. Process Life Cycle Framework (PLCF) [29] is a structural way for imitating agile transformation in software development practices. Along with this, technologies are emerging in the software development industries with the evolution of AI (Artificial Intelligence) [17], [45] and ML (Machine Learning) [22] technologies. Hence their contribution to Software Engineering Management should be considerable with the ABPR by software development organizations. Scrum is very suitable for software product development while Kanban is suitable for information and work flow management for regular production industries but still flow based approach of Kanban is more convenient than time boxed approach of Scrum [27]; as procedural system flow reduces software failure by highlighting issues immediately [35]. The combination of Scrum and Lean based Kanban has been found very effective in Software Engineering Management [31].

II. RESEARCH OBJECTIVES

Formation of Scrumban as an Agile Business Process Reengineering (ABPR) using empirical analysis of combination of Scrum and Kanban to overcome the challenges of Scrum is the main goal of this research study. Following are primary objectives of the study:

- Formation of Scrumban by integrating Scrum and Kanban as an ABPR approach.
- An empirical analysis about why Scrumban is essential as an alternate of Scrum and Kanban.
- A proposed concept about how Scrumban can overcome challenges of Scrum.
- Conceptual vision for customized Scrumban approach for the Software Engineering Management (SEM), towards resolution of limitations of Scrumban.

Selection of optimal practices of Scrum and Kanban, which will help in achieving desired result in the form of hybrid framework and its limitations, is the aim of this research.

III. SCRUM

Scrum is designed to maximize capability of team towards productivity and quality [41] improvement. Takeuchi and Nonaka [21] announced the word Scrum for the first time in 1986. Ken and Jeff [16] integrated Scrum with Agile approaches in 1993, to form the standard Agile Software Engineering Management (ASEM) framework, using iterative and incremental approach. From the strategy of the game of Rugby, they defined the core ideology of Scrum. Scrum has been proven as fascinating [13] among all agile approaches, for Software Engineering Management. Scrum is built under the roof of Agile principles and on the empiricism control theory, which defines key characteristic of Scrum like Artifacts, Values, Pillars [39], Roles, and Events. Scrum is an iterative and incremental agile approach [25] based on process control theory [26] using flexible length (duration) of project from a week to a month with the goal of achievement of DoD as result of Sprint.

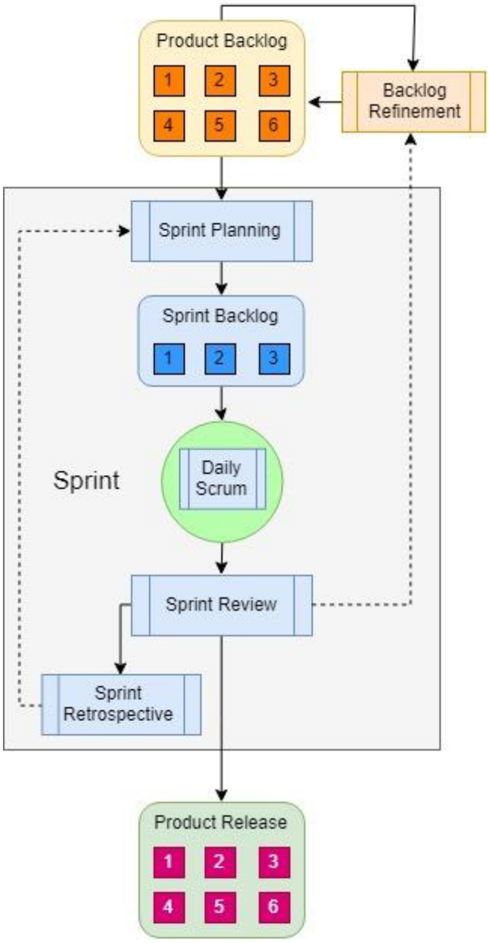


Fig. 1. Scrum Flow

Figure 1 represents the flow of Scrum along with some of its artifacts. Product owner manages product backlog items and refinement. Sprint consists of four core events: Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. Product Backlog Items (PBI) are pulled into Sprint Backlog (SB) during the Sprint Planning and then undertaken for development by the development team members and they discuss about updates during Daily Scrum meetings. Scrum Master takes care of impediments if any occurs during the Sprint. Developed backlog items are inspected during the Sprint Review by Scrum Team and external stakeholders.

Work items that are meeting with DoD (Definition of Done) is considered for release as a product increment. Sprint cycles continue until the end of product development.

IV. KANBAN

Kanban is designed for transparent workflow management by visualizing current state of each work item for the objective of scheduling system for lean and just-in-time manufacturing at Toyota by Taiichi Ohno [19] with the goal of limit WIP (Work-in-Progress). Kanban is derived from Lean manufacturing practices, by eliciting unnecessary processes to reduce the waste of time [32]. Work items are presented as a card on the Kanban board and visible to everyone in organization, to enable transparency about each work item and its state. Figure 2 represents the flow of Kanban, along with the visualization of work flow management for the state of each work item. Product development requirement are converted into Product Backlog Items (PBI) with prioritization of each one of them. Work items are pulled into 'To Do' list as per capacity of development team. According to WIP limit, each developer takes only single work item into process. Completed items are moved to 'Done' list. Kanban supports CICD of product release.

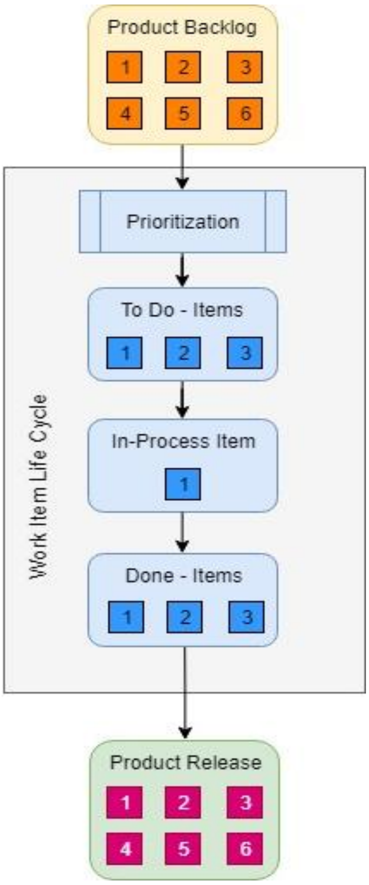
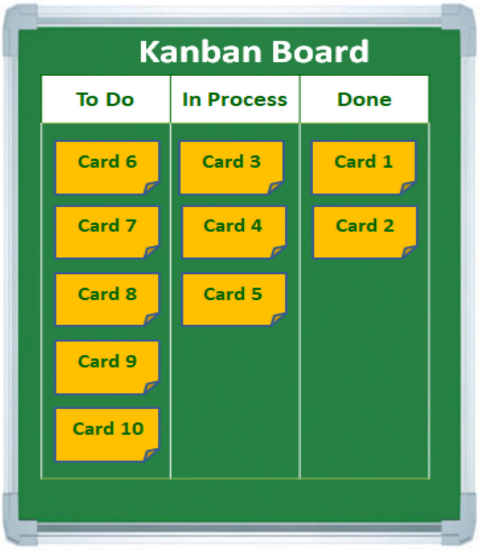


Fig. 2. Kanban Work Item Flow

Figure 3 shows Kanban board with categorization of Kanban cards. A single card item consists of a list of tasks where Card items are majorly categorized into 3 categories: 'To Do', 'In Process' and 'Done'.

- To Do: Cards in 'To Do' category, represent work in pipeline for a specific team or developer of the development team.
- In Process: Cards in 'In Process' category, represent work in process by the team members. A development team member pulls a card from 'To Do' pipeline. A single development team member can be assigned or occupied only one card at a time in this category under WIP limit.
- Done: Card is moved to 'Done' category, once a card holder (development team member) completes all the tasks listed in it.

Kanban board visualizes various states of each work item during its transition and helps in measuring progress of development activities. Transparency about the transition of work items that helps in Work Item Management (WIM); by reducing lead item and improves productivity of development team.



Kanban Board

To Do	In Process	Done
Card 6	Card 3	Card 1
Card 7	Card 4	Card 2
Card 8	Card 5	
Card 9		
Card 10		

Fig. 3. Combnation of Scrum and Kanban

Optimizing work flow is the main goal of Kanban, which can be achieved with the help of Kanban practices like, Limiting WIP (Work-in-Progress), Visualization, WFM (Workflow Management) and inspection of DoW (Definition of Workflow); and flow metrics WIP (Work In Progress), Cycle Time, WIA (Work Item Age) and Throughput.

V. SCRUMBAN

Scrumban is mixture of Scrum and Kanban by combining agility of empiricism with transparent workflow management system. Figure 4 indicates basic mixture of Scrum and Kanban in the form of Scrumban.

Core elements of Scrum that are used in the mixture:

- Sprint: Sprint planning, review and retrospective
- Pull System: Pull workload into Sprint backlog
- Push System: Prioritization and team decision to push work item into Sprint Backlog.

Core elements of Kanban that are used in the mixture:

- WIP Limit: Explicit limit indicating how many work items should be in process at a time.
- Shorten Lead Time: Management and planning of lead time using JIT concept.
- Kaizen: Maximize improvement and minimize waste.

- CICD: Continuous Integration and Continuous Delivery for the release of completed work items.
- WIM: Work item management and its visualization during transition through each stage.

Scrumban = Scrum + Kanban

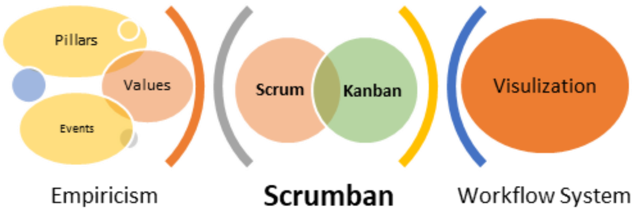
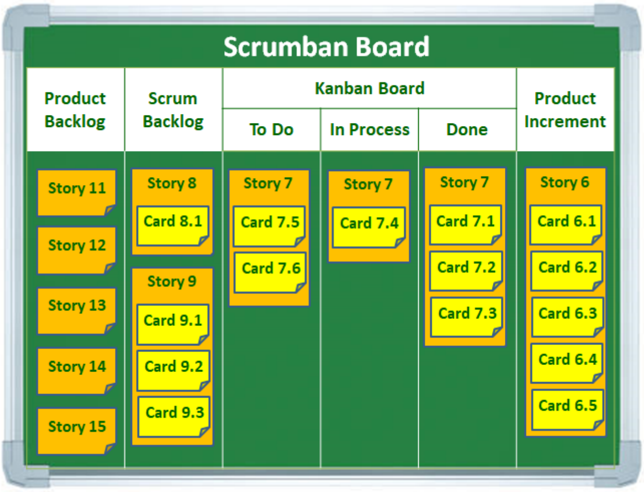


Fig. 4. Combnation of Scrum and Kanban

Product backlog prioritization is common process for both Scrum and Kanban which has been inherited into Scrumban.



Scrumban Board

Product Backlog	Scrum Backlog	Kanban Board			Product Increment
		To Do	In Process	Done	
Story 11 Story 12 Story 13 Story 14 Story 15	Story 8 Card 8.1 Story 9 Card 9.1 Card 9.2 Card 9.3	Story 7 Card 7.5 Card 7.6	Story 7 Card 7.4	Story 7 Card 7.1 Card 7.2 Card 7.3	Story 6 Card 6.1 Card 6.2 Card 6.3 Card 6.4 Card 6.5

Fig. 5. Scrumban Board

Figure 4 represent a visualization of Scrumban board, along with state of sample work item (known as card in Kanban) for each story of Scrum Backlog (SB) and Sprint. Flow of work item, according to proposed Scrumban framework, will be follows:

- Initial requirement will be collected in the pool of Product Backlog (PB), in the form of User Stories.
- During the Sprint Planning meeting, User stories will be pulled into Scrum Backlog (SB) by the Scrum Team members.
- Development team members will generate a Kanban card for each story point by converting story point into list of tasks for them to be accomplished during the Sprint.
- Each team member will select only one user story in its ‘To Do’ state of Kanban board.
- Only one Kanban card will be go ‘In process’ state from the selected User Story a time.
- The Kanban card will be moved to ‘Done’ state, once the all the tasks listed in a card will be completed.
- If entire user story if moved to ‘Done’ state, it can be pushed to product increment of Sprint, as a part of CICD, if required.
- Sprint will end, if all the user stories of Sprint Backlog will be (completed) moved to Sprint Review.

Other work flows will be followed according Scrum rules and guideline [1], [2].

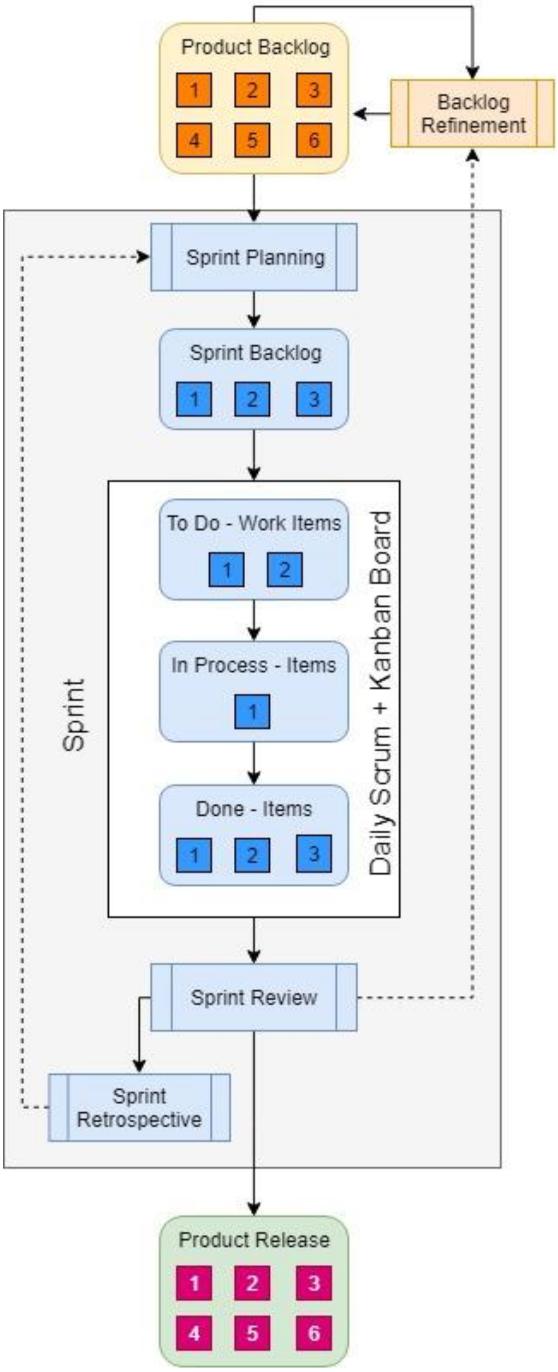


Fig. 6. Proposed Scrumban Flow

Figure 6 represents system flow for proposed Scrumban framework by adopting core elements and processes of Scrum and Kanban. Product development items are collected in the pool of Product Backlog, in the form user stories, where they go under refinement as often as the changes are requested by client or stake holders. Refinement process includes prioritization of user stories and each story is pulled into Scrum Backlog during Sprint Planning event according to priority.

Role of Kanban begins from this stage in Scrumban. User stories should be converted into a Kanban cards in the Sprint Backlog by development team members. Each contains one or more technical task specification with accurate estimation for each task and passes through each state of work items.

Completed items that confirms DoW (Definition of Workflow), should be move to Scrum Review events where completed stories are inspected by Scrum team along with stake holders. User stories confirms DoD (Definition of Done), will be considered for product release. Unlike Scrum, the Kanban system does not have time boxed events and that's why Scrumban borrows Sprint Planning, Daily Scrum and Sprint Review from the Scrum [28].

VI. EMPIRICAL ANALYSIS

The word 'Empiricism' asserts that knowledge comes from the experience and direct or indirect experience is evidence to truth and reliability. The result of this research is carried out by a qualitative empirical analysis of objectives of this research. In 2011, Ikonen et al [34] argued that the Kanban motivates development team members by controlling project activities using Lean Agile thinking principles and it has great implication in gaining momentum in the field of software development. During the empirical investigation in 2012, Yilmaz et al [24] evaluated that Scrum and Kanban are most popular and widely used Agile based frameworks in the technology industries. Nikitina et al [46] performed a case study in 2012, on transition of Scrum to Scrumban. They evaluated two pivotal parameters, which should be considered by the organizations are, CICD establishment and well trained resources, can improve processes and sustainable result. In 2012, Terlecka [47] implemented Scrumban in the system maintenance project with the team of system administrators which improved the result of the project, after unsuccessful attempts with first Scrum only and then after Kanban. Ahmad et al [4] suggested in 2014, that combination of agile based model with traditional SDLC can resolve Scrum limitations and improve quality of the product. Bougroun et al [33] proposed an integration of Scrum, Kanban and XP in 2014 with the aim of comprising such software development methodologies into CMMi to improve cost and budget optimization and they gained 58% software product quality improvements. In 2016, Alqudah and Razali [18] suggested that scaled Agile methodologies are essential for the large scaled software project development where large team with requirement specific expertise are essential to overcome the limitations [43] of existing agile methodologies like XP, DSDM, Scrum and Kanban. Yilmaz and O'Connor [23] performed a cross section survey in technical research and development organizations for the gamification project, as part of their empirical case study in 2016, to get an opinion of technology experts about adoption of a hybrid software development methodology - Scrumban and they carried out as a result that Scrumban helps in improving productivity of individual and organization. Salah et al [9] recommended a hybrid form of agile based frameworks in 2017, to overcome the challenges with Scrum like budget estimation, project delivery and goal achievement. In 2017, Ashraf and Aftab [7] proposed with insightful understanding that the Scrum requires to be plugged in with different Agile models to enhance the productivity and quality of the product. Hanslo and Mnkandla [6] performed narrative reviews in 2017, on

Scrum challenges like structure of organization and its suitability with Scrum, lack of expertise in Scrum team members and proposed Scrum variants by adopting traditional software development model. In 2018, Patil and Neve [5] observed that Kanban is stepping into the world of Scrum but Kanban as a standalone framework, can't provide full SDLC development and management support. They proposed the mixture of Scrum and Kanban to improve competency of both Scrum and Kanban. Alqudah and Razali presented in 2018 [11], that Scrumban has been found appropriate framework in reducing waste of time, improving productivity and quality, compared to Scrum and Kanban. In 2018, Albarqi and Qureshi [30] proposed L-Scrumban framework integrating Lean, Scrum and Kanban to achieve a comprehensive agile approach using survey result which confirms the proposed L-Scrumban improves that efficiency of software redevelopment practices. In 2018, Plengvittaya and Sanpote [49] surveyed the implementation of a Scrumban as a mixture of Scrum and Kanban to improve the efficiency of software project development for the students of University of Phayao, and analyzed that experience and perception of the team members influences the success factors of the project. Robinson [48] presented an experience report on implementation of Scrum and Scrumban in 2019 at SIE (Sony Interactive Entertainment) where they were facing issues with Scrum and their Sprints were almost failed, which could be solved with the help of Scrumban. Mohan et al [12] conducted a case study in 2019, on the procedure of pension payment for a small employee section with the purpose of facilitation and guidance using Scrumban, which resulted into increase in completion of accreditation process of pension.

VII. RESULTS AND RECOMMENDATIONS

Scrumban is derived from the characteristics of Scrum and Kanban. Each and every methods have their own pros and cons. Along with strengths, the limitations of both methodologies are also inherited into Scrumban that are represented in Table I. Table I represent the result of this research by indicating Scrum limitation parameters and the support of Kanban in the formation of Scrumban. It shows limitations of Kanban and Scrumban against the parameters Scrum challenges [3]. The keyword 'Yes' (limitation) and 'No' (strength) indicates agreement with limitation or strength for the mentioned challenging parameter, while 'NA' represents that challenging parameter is not applicable or there no specification for particular framework.

A. Strengths of Scrumban

Scrumban helps in improving agile practices of Software Engineering Management by overcoming following limitations of Scrum with the help of Kanban.

- JIT: Scrum has no lead time management specification, which can be implemented with the help of JIT, a concept of Kanban; to reduce lead time for all activities and improvement productivity.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- WFM (Work Flow Management): Scrumban allows internal workflow management by assigning state (like 'To-Do', 'In-Progress' and 'Done') to each work item. Scrum does not have internal work flow management system within the Sprint.
- Work in Process: Scrum does not provide any specification about visualization of work items which are active now and in process.
- WIP Limit: Scrum specifies WIP control mechanism at Sprint Backlog level but there is no specification about WIP at any time for the developer. Scrumban restrict overload of work item by implementing WIP limit in Kanban board on each work item.
- CICD: Iterative approach Scrum does not support continuous integration of development activities and continuous deployment of releasable features, which can be implemented with the help of Kanban.

Table I: Limitations of Scrum and support of Kanban in formation of Scrumban

Parameter	Limitations			
	Scrum Limitation Description	Scrum	Kanban	Scrumban
Just-In-Time	No lead time management.	Yes	No	No
External Stakeholder	Restricted involvement of external stakeholders as a team member.	Yes	NA	Yes
Team Size	Restricted team size.	Yes	NA	Yes
Role	Limited roles in a team	Yes	NA	Yes
Project Tracker	Progress view at project level.	Yes	Yes	Yes
WFM	No workflow management within the Sprint.	Yes	No	No
Work In Process	No vision about internal state of work items.	Yes	No	No
WIP Limit	Explicit limit indicating work items in process at a time	Yes	No	No
Product Vision	Unclear product vision.	Yes	Yes	Yes
Estimation	Full product development estimation is not feasible.	Yes	Yes	Yes
Distributed Environment	Collaboration issue in distributed environment.	Yes	NA	Yes
Skill and Expertise	Unavailability of specific skilled resource.	Yes	NA	Yes
Documentation	No or minimum documentation.	Yes	Yes	Yes
CICD	Limitation of Iterative approach	Yes	No	No

B. Weaknesses of Scrumban

Kanban has improved strengths of Scrum in the form of Scrumban but still there are some limitations of Kanban against Scrum rules, which are ultimately weaknesses of Scrumban, represented as follows:

- Team Size: Scrum limits size of team members between 3-9 for a single Scrum team which is the best phenomena for small or mid-size of project. But it limits progress for large scaled project and Kanban has no specifications about team size.
- External Stakeholder: Scrum restricts direct involvement of external stakeholders into Sprint, which is direct indicator of issue with progress of Sprint and its DoD. Kanban has no specification about involvement of external stakeholders.
- Role: Scrum limits role of team members between Product Owner, Scrum Master and Development Team Member. Task specific roles are essential within development team members and it defines their involvement into team. Kanban has no specification about roles of team members.
- Skill and Expertise: According to Scrum, development team is cross functional [42] team and they have all the expertise to accomplish required goal, during the Sprint. But in real world situation [40], development team members are dependent on skilled experts to complete their tasks.
- Distributed Environment: Collaboration and project management are the common issues in distributed environment and typically not easy to resolve.
- Documentation: Scrum and other agile based methodologies do not support project requirement and specification documentation, which is essential for project signup process and demonstration to client.
- Estimation: Estimation of project is dependent on strong documentation of full project. While Scrum does not support project documentation, it's challenging to provide complete estimation [44] of project to client.

C. Recommendations

The result of this research highly recommends Scrumban for the software development and management practices, compared to Scrum or Kanban as the ideology of Scrumban is formed by the strengths of Scrum and Kanban both. The result also represents that Scrumban inherits some of the limitations of Scrum, which needs further investigation and recommends that software development organization should consider the challenges of Scrumban, prior to adopting it as a standard Software Engineering Management (SEM) for their software development practices. Krunal et al have proposed Scrumbanfall [8] as an Agile Business Process Reengineering to overcome, some of the limitations of Scrumban.

VIII. CONCLUSION

Software development organizations, those have adopted Scrum as a standard Agile framework and implemented in their software product development and management practices, are struggling with some of the challenges, (that we discovered in previous research [3]); are looking for the resolution of such issues and with the same goal, we have proposed Scrumban by integrating Scrum with Kanban. The result of this result concludes that Scrumban is an Agile Business Process Reengineering (ABPR) approach for the Scrum and Kanban practitioners as it can resolve some of the challenges of Scrum like work item flow management and visualization of its progress, reduction of lead time with the help of JIT concept that helps in CICD (Continuous Integration and Continuous Delivery) of the releasable work items rather depending upon the Sprint iteration.

The proposed framework Scrumban can overcome some of the limitations of the Scrum with the help of Kanban. But still there are some issues, which we highlighted in previous research [3] but could not be covered all of them in this research and needs to explore, the feasibility of integration of Scrumban with most popular traditional framework Waterfall as an hybrid SDLC framework called Scrumbanfall [8], as an attempt to resolve some of the limitations of Scrumban like complete estimation and documentation of the project without altering its base rules.

IX. FUTURE ENHANCEMENT

Scrum is the most popular agile framework in the software development organizations for their Software Engineering Management (SEM) practices. But we also, highlighted that there are some limitation of Scrum and, not all, but some the issue of them can be resolved with the help of Kanban. It is still required to continue our research by targeting those specific issues, which could not be covered (resolved) using Scrumban; in the center of next phase of research. Our focus for the next stage of research is Scrumbanfall, a hybrid of traditional SDLC Waterfall and Scrumban.

X. ACKNOWLEDGEMENT

As a part of my academic affiliation with Indus University for the doctorate degree program in Computer Science and Engineering, through this research study, we have proposed Scrumban as a resolution of some of the Scrum challenges and Scrumbanfall will be our next goal of research to overcome remaining challenges.

XI. ABBREVIATIONS

A

ABPR – Agile Business Process Reengineering
AI - Artificial Intelligence
AM – Agile Methodology
ASEM – Agile Software Engineering Management Framework

B

BPR – Business Process Reengineering

C

CICD – Continuous Integration and Continuous Delivery

D

DoD – Definition of Done
DoW – Definition of Workflow

E

ERP – Enterprise Resource Planning
ESE - Empirical Software Engineering

G

GSD – Global Software Development

J

JiT –Just in Time

M

ML – Machine Learning

O

OOP – Object Oriented Programming

P

PB – Product Backlog
PBI – Product Backlog Item
PLCF – Process Life cycle Framework

S

SB – Sprint Backlog
SDLC – Software Development Life Cycle
SE – Software Engineering
SEM – Software Engineering Management
SPM – Software Project Management

W

WFM – Work Flow Management
WIP – Work in Process

X

XP – Extreme Programing

REFERENCES

1. K. Schwaber and J. Sutherland, "The Kanban Guide for Scrum Team", Scrum.org, 2019
2. K. Schwaber and J. Sutherland, "The Scrum Guide", Scrum.org, 2017
3. K. Bhavsar, V. Shah and S. Gopalan, "Scrum Challenges: An Agile Process Reengineering in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 4, 2020. ISSN: 2278-3075. (Submitted. Under Review.)
4. G. Ahmad, T. R. Soomro and M. N. Brohi, "XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP)", International Journal of Soft Computing and Engineering (IJSCE), vol. 2, issue 3, pp. 126-130, 2014. ISSN: 2231-2307
5. S. Patil and J. Neve, "Productivity Improvement of Software Development Process through Scrumban: A Practitioner's Approach", International Conference on Advances in Communication and Computing Technology (ICACCT), pp. 314-318, 2018. DOI: 10.1109/ICACCT.2018.8529405
6. R. Hanslo and P. E. Mnkandla, "Scrum Adoption Challenges Detection Model: SACDM", 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), Poznan, pp. 949-957, 2018.
7. S. Ashraf and S. Aftab, "Scrum with the Spices of Agile Family: A Systematic Mapping", International Journal of Modern Education and Computer Science (IJMECS), vol. 9, issue 11, pp. 58-72, 2017. ISSN: 2075-017X, DOI: 10.5815/ijmecs.2017.11.07
8. K. Bhavsar, V. Shah and S. Gopalan, "Scrumbanfall: An Agile Integration of Scrum and Kanban with Waterfall in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 4, 2020. ISSN: 2278-3075
9. A. Salah, N. R. Darwish and H. A. Hefny, "Towards a Hybrid Approach for Software Project Management using Ontology Alignment", International Journal of Computer Applications (IJCA), vol. 168, issue 6, pp. 12-19, 2017. ISSN: 0975 – 8887

10. M. Beedle, J. Kern, A. Bennekum, K. Schwaber, A. Cockburn, S. Mellor, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, B. Marick, K. Beck, R. C. Martin, D. Thomas and J. Sutherland, "Manifesto for Agile Software Development", AgileManifesto.org, 2001
11. M. Alqudah and R. Razali, "An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices", International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), vol. 8, issue 6, pp. 2315-2322, 2018. ISSN: 2088-5334, DOI: 10.18517/ijaseit.8.6.6566
12. A. Mohan, Devisree A. S. and C. V. Prasanna Kumar, "Rationalized Scrumban Based Methodology for Improving the Slackening Pension Scheme of Employee Sectors in India", International Journal of Recent Technology and Engineering (IJRTE), vol. 8, issue 1, pp. 1525-1530, 2019. ISSN: 2277-3878
13. K. Bhavsar, V. Shah and S. Gopalan, "Scrum: An Agile Process Reengineering in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 3, pp. 840-848, 2019. ISSN: 2278-3075, DOI: 10.35940/ijitee.C8545.019320
14. R. V. Anand and Dr. M. Dinakaran, "Popular Agile Methods in Software Development: Review and Analysis" International Journal of Scientific and Technical Advancements, vol. 2, issue 4, pp. 147-150, 2016. ISSN: 2454-1532.
15. K. Godbole and J. Neve, "Kanban: A Definite Answer for Effective Project Execution in Uncertainties", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 6, issue 6, pp. 561-565, 2017.
16. J. Sutherland and K. Schwaber, "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework", Scrum Inc., draft: 29 Jan 2011, Paris.
17. K. Bhavsar, V. Shah and S. Gopalan, "Business Process Reengineering: A Scope of Automation in Software Project Management using Artificial Intelligence", International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, issue 2, pp. 3589-3594, 2019. ISSN: 2249-8958, DOI: 10.35940/ijeat.B2640.129219
18. M. Alqudah and R. Razali, "A Review of Scaling Agile Methods in Large Software Development", International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), vol. 6, issue 6, pp. 828-837, 2016. ISSN: 2088-5334, DOI: 10.18517/ijaseit.6.6.1374
19. T. Ohno, "Toyota Production System - beyond large-scale production", Productivity Press, pp. 29, 1988. ISBN 0-915299-14-3.
20. N. Nikitina and M. Kajko-Mattsson, "Developer-driven big-bang process transition from Scrum to Kanban", International Conference on Software and Systems Process, 2011, pp. 159-168.
21. H. Takeuchi and I. Nonaka, "The New Product Development Game", Harvard Business Review, 1986.
22. K. Bhavsar, V. Shah and S. Gopalan, "Machine Learning: A Software Process Reengineering in Software Development Organization", International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, issue 2, pp. 4492-4500, 2019. ISSN: 2249-8958, DOI: 10.35940/ijeat.B4563.129219
23. M. Yilmaz and R. V. O'Connor, "A Scrumban Integrated Gamification Approach to Guide Software Process Improvement: A Turkish Case Study", Technical Gazette, vol. 23, issue 1, pp. 237-245, 2016, ISSN 1330-3651 (Print), ISSN 1848-6339 (Online), DOI: 10.17559/TV-20140922220409
24. M. Yilmaz, R. V. O'Connor and P. Clarke, "A systematic approach to the comparison of roles in the software development processes", Software Process Improvement and Capability Determination. Communications in Computer and Information Science, vol. 290. Springer, Berlin, Heidelberg, vol. 290, pp. 198-209, 2012, ISSN: 1865-0929, ISBN 978-3-642-30439-2 (Online), DOI: 10.1007/978-3-642-30439-2_18
25. K. Schwaber, "Agile Project Management with Scrum", Microsoft Press, Redmond, WA, USA. 2004, ISBN:073561993X
26. C. Larman, "Agile and iterative development: A manager's guide", Addison-Wesley Professional, Massachusetts, 2004.
27. M. Cohn and D. Ford, "Introducing an Agile Process to an Organization", IEEE Computer, vol. 36, issue 6, pp. 74-78, 2003. DOI: 10.1109/MC.2003.1204378
28. C. Ladas, "Scrumban-essays on Kanban Systems for Lean Software Development". Modus Cooperandi Press, USA, 2009
29. K. Bhavsar, V. Shah and S. Gopalan, "Process Life Cycle Framework: A Conceptual Model and Literature Study of Business Process Re-Engineering for Software Engineering Management", CiiT International Journal of Software Engineering and Technology, vol. 11, issue 6, pp. 096-100, 2019. ISSN: 0974-9748.
30. A. A. Albarqi and R. Qureshi, "The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development", International Journal of Information Engineering and Electronic Business, vol. 10, issue, 3, pp. 23-35, 2018. ISSN: 2074-9031 (Online), DOI: 10.5815/ijieeb.2018.03.04
31. M. Kohlbacher, E. Stelzmann and S. Maierhofer, "Do Agile Software Development Practices Increase Customer Satisfaction in Systems Engineering Projects?", 2011 IEEE International Systems Conference, Montreal, QC, pp. 168-172. 2011, EISBN: 978-1-4244-9493-4, DOI: 10.1109/SYSCON.2011.5929091
32. A. Janes, "A Guide to Lean Software Development in Action", 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Graz, pp. 1-2, 2015. ISBN: 978-1-4799-1885-0, DOI: 10.1109/ICSTW.2015.7107412.
33. Z. Bougroun, A. Zeaaraoui and T. Bouchentouf, "The projection of the specific practices of the third level of CMMI model in agile methods: Scrum, XP and Kanban", 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, pp. 174-179, 2014. EISBN: 978-1-4799-5979-2, DOI: 10.1109/CIST.2014.7016614
34. M. Ikonen, E. Pirinen, F. Fagerholm, P. Kettunen and P. Abrahamsson, "On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation", 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, NV, pp. 305-314, 2011. EISBN: 978-0-7695-4381-9, DOI: 10.1109/ICECCS.2011.37
35. A. C. Jasti, J. Medapati and R. TV, "An Empirical Software Reliability Growth Model for Identification of True Failures", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, issue 10, 2019, ISSN: 2278-3075, DOI: 10.35940/ijitee.I8756.0881019
36. A. Rengarajan, S. Hariharan and R. Prem Kumar, "Scrum based Scaling using Agile Method to Test Software Projects and its Future Solutions using in Artificial Neural Networks", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, issue 9, pp. 223-230, 2019, ISSN: 2278-3075, DOI: 10.35940/ijitee.H6853.078919
37. T. Vijayakumar and V. Ram, "Effects of Agile adoption on Trust, Knowledge Sharing and Collaboration in IT Organizations", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, issue 12, pp. 2652-2655, 2019. ISSN: 2278-3075, DOI: 10.35940/ijitee.K2219.1081219
38. E. O. C. Mkpojiogu, N. LailyHashim, A. Al-sakkaf and A. Hussain, "Software Startups: Motivations for Agile Adoption", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, issue 8S, pp. 454-459, 2019, ISSN: 2278-3075.
39. S. Kulkarni, N. Takawale and J. Bachhav, "To Monitor and Improve Project Development of Final Year Students using Scrum Model", International Journal of Engineering and Advanced Technology (IJEAT)", vol. 9, issue 1, pp. 953-955, 2019, ISSN: 2249–8958
40. S. Al-Ratrout, "Practical Implementation of Agile Approaches in Teaching process", International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, issue 4, pp. 278-284, 2019. ISSN: 2249 – 8958
41. R. C. Narayanan and N. Ganesh, "Challenges Faced in the Enterprise Resource Planning Material Management Section when Transitioning Towards Agile Software Development", International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, issue 6, pp. 3472-3475, 2019. ISSN: 2249–8958
42. D. N. A. Jawawi, A. F. B. Arbain, W. M. N. B. W. Kadir and I. Ghani, "Requirement Traceability Model for Agile Development: Results from Empirical Studies", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, issue 8S, pp. 402-405, 2019. ISSN: 2278-3075
43. A. K. Rai, S. Agarwal, M. Khaliq, A. Kumar, "Quantitative Analysis of Development Environment Risk for Agile Software through Machine Learning", International Journal of Recent Technology and Engineering (IJRTE), vol. 7, issue 6, pp. 83-89, 2019. ISSN: 2277-3878
44. V. Jayaraj and N. A. Bhaskaran, "A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD)", International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, issue 1, pp. 1078-1087, 2019, ISSN: 2249–8958
45. S. K. Yadav, R. A. Mahajan and S. A Mahajan, "Development and Integration of Scrum Tree Algorithm with K-means Data Clustering", International Journal of Engineering and Advanced Technology (IJEAT), vol. 8 issue 6, pp. 4248-4251, 2019, ISSN: 2249 – 8958

46. N. Nikitina, M. Kajko-Mattsson and M. Strâle, "From scrum to Scrumban: A case study of a process transition," 2012 International Conference on Software and System Process (ICSSP), Zurich, pp. 140-149, 2012. DOI: 10.1109/ICSSP.2012.6225959
47. K. Terlecka, "Combining Kanban and Scrum -- Lessons from a Team of Sysadmins", 2012 Agile Conference, Dallas, TX, pp. 99-102, 2012. DOI: 10.1109/Agile.2012.20
48. P. T. Robinson and S. Beecham, "TWINS - This Workflow Is Not Scrum: Agile Process Adaptation for Open Source Software Projects", 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), Montreal, QC, Canada, pp. 24-33, 2019. DOI: 10.1109/ICSSP.2019.00014
49. C. Plengvittaya and D. Sanpote, "Scrumban for teaching at undergraduate program: A case study from software engineering students, University of Phayao, Thailand" 2018 International Conference on Digital Arts, Media and Technology (ICDAMT), Phayao, pp. 109-114, 2018. DOI: 10.1109/ICDAMT.2018.8376505

AUTHORS PROFILE

Krunal Bhavsar, is a research scholar and corresponding author for this research paper. He is pursuing his PhD in the stream of Computer Science and Engineering from Indus University (Ahmedabad, India). Krunal has received his MBA in Information Technology from Indus University (Ahmedabad, India) in 2018 and MSc in Mobile Application and Technology (Gold Medalist) from Gujarat University (Ahmedabad, India) in 2014 and BSc in Computer Science from Gujarat University (Ahmedabad, India) in 2012. He is an enthusiastic research scientist with 14+ years of corporate experience in software business process management, software project management, software engineering and data analytics. He is certified Scrum Master by Scrum.org for certifications ‘Professional Scrum Master - I’ and ‘Professional Scrum with Kanban - I’. He is a Technical Project Manager in software research and development organization and he has professional expertise in data science and analysis also. He has been serving as a visiting lecturer at Computer Science department at Gujarat University also. Teaching, Training, Learning are his hobbies and he is passionate about research and development in Artificial Intelligence and Machine Learning technologies.

Dr. Vrutik Shah, is a research guide and contributor author to this research work. He has received his PhD from Karpagam University in the stream of Computer Science and MCA in the stream of Computer Application from Nirma University. He is a Technical Project Manager in multinational well-known software development organization – Streebo and he has been leading software process redesign and redevelopment with the help of trending automation technologies. He is a certified Agile Scrum Master and having expertise in software project and product management, business analysis and business process reengineering.

Dr. Samir Gopalan, is a research co-guide and contributor author to this research work. He is a director and a head of department of management, external relation manager and dean of faculty (OSD Research) at Indus University. He has achieved Ph.D., M.Phil. (Computer Science), MBA (Finance) and BSc degrees in his academic career. His domain research areas are marketing management, business intelligence, training & development, skill & performance enhancement with 15+ years of experience.Krunal Bhavsar is the author to whom all the correspondence regarding this research paper should be addressed; E-mail: krunalbhavsar@engineer.com; krunalbhavsar.rs@indusuni.ac.in. Tel: +91-9737007007