

Scrum: An Agile Process Reengineering In Software Engineering

Krunal Bhavsar, Vrutik Shah, Samir Gopalan

Abstract: *BPR (Business Process Re-engineering) is an organizational mechanism leading organization towards change management as and when it is required. In the form BPR, Agile practices have had a remarkable impact on Software Engineering Management (SEM) in software development organizations across the world. It has enhanced collaboration and productivity of Software Engineering (SE) teams and improved the level of quality of software products. Nowadays Scrum has been trending into software development organizations as a standard SDLC approach. Scrum is the framework of the Agile methodology, built on empiricism control theory, asserts that experience brings knowledge and increases decision making capabilities. The empiricism control theory is built on three pillars: Adaptation, Inspection and Transparency. This research study presents Scrum as a trending SDLC framework using empirical analysis. We have analyzed literature reviews, case studies, and research surveys; and implemented Scrum in our software development unit and carried out the reasons, why Scrum is trending in software development organizations. How the scrum artifacts, events and values play a vital role in upholding the Scrum pillars and strengthen the capabilities of Scrum team members to address the software engineering management challenges; compared to traditional software development approaches.*

Keywords: *Scrum, 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*

I. INTRODUCTION

Software development organizations have been transforming from the traditional project planning based approaches to agile software engineering, to answer the question about how a software development can be organized to deliver better and faster solutions. Since two decades many remedies were suggested to form a standardization of software process measurement with the help of comparative analysis and empirical reviews. Software development organizations which are focusing on customer values, product quality, team collaboration and individual adoptions using fascinating agile methodologies, have been gaining productivity for their software product development activities. Agile approaches reduce the effort of

Revised Manuscript Received on January 5, 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. Sameer Gopalan, Research Co-Guide, Business Administration & Management, Indus University, Ahmedabad, India.

documentation of project and claims that the result of developed source code should be itself a project document, representing features and system flow of the software. Sjoberg et al [19] presented an outlook of Empirical Software Engineering (ESE) research that the software development organizations follows different types of practices, software engineering standards, technologies and roles to achieve the organizational goals. Amongst them, agile methodologies constitute a set of practices that have been found revolutionary and could bring change in software engineering processes and results. According to Dyba [20], relying and following on traditional software development theories is like an assumption that all the problems in the software world are predicated and can be resolved. But in the real software world, there are new unpredicted challenges at each step that have been never seen and discussed. Agile processes address the unpredictable issues by allowing people to use their own creativity and problem solving skills to overcome the challenges rather than relying on system processes to handle the issues. Object Oriented Programming (OOP) software engineering practice provides a viable solution to design a robust coding structure for the software. The iterative and incremental lightweight agile methodologies claim that OOP can be implemented at program level only. The step by step development and increment of releasable features should be cognizant to the agile philosophy of the stakeholders involved into software development, [22] argued by Cockburn and Highsmith. The agile approaches are about iterative feedback, analysis, change and implement the new increment into practice. It's not about the rejection even if the level of change is very high. Embrace the change; stated by Williams and Cockburn [23] in 2003. The high level of change management leads towards BPR (Business Process Reengineering) in organizational processes. Morien reviewed [28] in 2014, that the agile approaches are preferred by many software development and management experts and adopted as an SEM (Software Engineering Management) approach in their software development practices.

II. RESEARCH OBJECTIVES

Primary goal of this research study is focus on SEM (Software Engineering Management) for the software development organizations seeking BPR; with the help of empirical case studies and literature analysis. Following are core objectives of this research study:

- Focus on the fascinating agile framework – Scrum
- An empirical analysis for the reasons, why Scrum is trending in SEM.

- Scrum as an ABPR approach for the software development organizations

This research highlights Scrum as the present and future of SEM. The empirical analysis of traditional agile methodologies and BPR using Scrum in software development organizations based on case study reviews is the approach of this research.

III. BPR AND SEM

Today’s dynamic technological environment has raised emergent change in the software business world that they should consistently adapt new policies, structure and strategies to compete the industry requirements, [21] observed by Nerur et al. The growing interest of Global Software Development (GSD) in implementation of agile practices in Software Engineering Management (SEM) has been leading software development organizations towards Business Process Reengineering (BPR). Krunal et al [5] have proposed a new PLCF (Process Life Cycle Framework) as a life cycle model of BPR (Business Process Reengineering) for software product development, to improve capabilities of organization. AI (Artificial Intelligence) has becoming a trending technology for software development organizations in the form of BPR [14]. ML (Machine Learning), a sub domain of AI, is scientific and statistical analysis of data that can play a vital role in SPM (Software Project Management) as a processing unit of SDLC project planning to produce accurate output [37]. In 2019, Medapati et al. [38] presented that procedural based software layout reduced failures by identifying them at early stages.

IV. AGILE METHODOLOGY (AM)

Agile software development methods were introduced in late 1990, with the approach of iteration and increment in software development process. The core principles of agile methodology were defined by the software process methodologist, known as The Agile Manifesto [2], in February 2001. It uncovers the understanding of software development requirement using following statements:

- Collaboration with people is more important than the system process.
- A deliverable software product release is more important than the documentation of the process.
- Customer satisfaction is more important than the negotiation.
- Always ready to accept change in system, where and when it is required

The Agile Manifesto is very concise and simple to understand and implement but complex to master (manage). The manifesto is built on empiricism of software development practice and management processes. The vision behind the manifesto clearly focuses on the real world scenarios of software engineering management and proposed new ways of thinking from the development point of view that self-organizing, responsibility, accountability and creativity should be managed by development team members rather than relying on supervisor or management. This led invention

of the Scrum framework. The Scrum is a framework of AM (Agile Methodology) built on principles of manifesto. The agile approach and its basic principles are also applicable in creative and cultural business sectors like architecture design and media advertisements [33].

V. SCRUM

Scrum is a software product development and management framework, founded by Jeff Sutherland and Ken Schwaber in 1993 [8], derived from the principles of agile methodology and built on the base of empiricism. Takeuchi and Nonaka [7], who are the pioneer of the keyword Scrum, announced in 1986 that the new Agile based framework has helped global transformation of software development and ready to be adopted by software development organizations. They derived the strategy of Scrum from the game of Rugby with the holistic and self-organizing structure in the team during the game. Scrum [1] is a lightweight framework which enables capability of team to address complex adaptive challenges for development and delivery of high value products by improving collaboration, creativity and productivity.

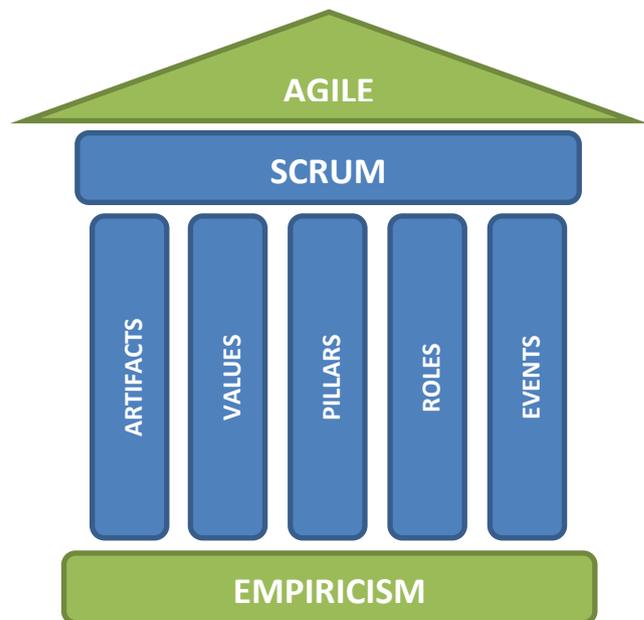


Fig. 1. Base and Roof of Scrum

As represented in figure 1, Scrum is built on empiricism control theory which is managed by: Artifacts, Values, Pillars, Roles, and Events. Adaptation, Inspection and Transparency are core pillars of Scrum that uphold empiricism theory of Scrum. Successful implementation of Scrum is dependent on its core five values: Courage, Commitment, Openness, Respect and Focus. The team members learn and explore these values, as soon as they are involved into self-organization. Product Owner, Scrum Master and Development team (members) are only roles in Scrum that understands, executes and manages self-organization throughout the Sprint as they are considered to be cross functional and having all necessary competencies to achieve the sprint goal.

The main events defined in Scrum are Sprint, Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective, as life cycle events of a Scrum for a single Sprint. Product Backlog, Sprint Backlog (SB) and Increment are artifacts of Scrum, designed as a base of empiricism of pillars, values, roles and events of the Scrum. Scrum is flexible across the Sprint cycle by providing a control mechanism that allows limiting (WIP) Work in Process for the development by the development team members; so that they can focus on specific sprint goal to produce an increment and that product increment can be released at any point of time, if the developed feature meets DoD (Definition of Done) of Sprint. The Daily Scrum allows development team members to share their thought in the form of suggestions. They discuss all the impediments they have had or will have, which require input from other team members. Daily Scrum is a great example of self-organization. Sprint review provides an opportunity for inspection of result of Sprint by Scrum team along with stakeholders. Sprint Retrospective allows an opportunity for improvements, for the next Sprint.

VI. EMPIRICAL ANALYSIS

Rising and Janoff derived from their experiments in 2000 [24], Scrum limits number of members in a Scrum team because the collaboration is one of the core values of Scrum, while a large team (more than 9 development team members in a single team) increases difficulties in communication with each other and becomes resistance for proper implementation of Scrum. Limited members in a team, reduces unnecessary discussion and conflicts with each other and increases development velocity. Duration per Sprint is limited from 1 to 4 weeks, and if the releasable feature of product will not be ready even after a week, there will be possibilities that product goal may become obsolete or changed and the increment of product will no longer satisfy product vision. Highsmith and Cockburn concluded in their experiment in 2001 [22] that the specific reason of limiting Sprint duration is reduction of the significant loss due to change request in product structure or vision by the stakeholders. Another benefits of incremental approach of Scrum is that the client can see the product and share feedbacks on it, which will direct Scrum team in considering changes, that are essential for the DoD (Definition of Done) should meet client expectation; as well as client can also get a direction of future of product vision. Ken Schwaber [9], the co-founder of Scrum, argued in 2004 that the three pillars of Scrum employed by empirical control theory that confirms any aspect of the process which affects definition of done must be visible to all team members, it should be investigated and refined frequently to ensure that unacceptable variables are no more affecting. Chatzoglou et al [11] surveyed IT sectors in a Greek in 2007 and reported that IT/IS projects are relies on the parameters - commitment, knowledge and experience. All of these parameters are base of empiricism. In 2008, Cho [15] represented benefits of agile practices over traditional SDLC methodologies and introduces core challenges like user engagement, communication, documentation, working environment that can be addressed with the help of implementation of Scrum. Dyba and Dingsoyr [18] studied

literature reviews in 2008 and investigated perceptions of benefits and limitations of agile methods using comparative analysis. They also identified that Scrum is very popular in the software industry compared to other agile methodologies.

Hossain et al conducted a systematic literature review in 2009 [16], about execution of Scrum framework practices in GSD projects, with the goal of determination of challenging factors as a resistance to use of Scrum. They reviewed impact of such factors on GSD in terms of coordination, collaboration and communication. The base concept and strategies of Scrum are built on such factors and can overcome impact of the factors; they derived in their comprehensive survey. In 2010, Cardozo et al [12] systematically reviewed future scope of Scrum and its role to improve productivity in software projects and concluded that the use of agile based frameworks will increase productivity in software development organization. According to scientific evidence from their research outcome, Scrum will play a vital role in improvement in productivity of software projects. Harleen et al [6] surveyed in 2014 in mobile application development organization and evaluated that agile software engineering management methodologies like Scrum, Lean and XP are best suited for their mobile application software development unit and improved utilization of resources. Scrum has proven itself as a product creation framework by stakeholder involvement for improvement in customer values. In 2014, Kapitsaki and Christou [39] emphasized from the result of their study, that the ratio of successful adoption of Scrum is 80%, compared to other agile methodologies. In 2015, Permana [3] concluded during his analytical study that Scrum helps in visualization of quality and risk in the project using early customer feedback by highlighting needs for change. Santos et al [17] empirically studied integration of CBL (Challenge-Based Learning) with Scrum framework in 2015. They found that the integration of Scrum with CBL for mobile application development increased productivity and quality of software application. Ramos et al observed in 2016 [26], in their case study for the implementation of Scrum control in the research institute project, that regular collaboration between team members as a part of Daily Scrum, improved pro-activities, respect and accountabilities and the opinion survey represents quality improvements in their scientific work. Khmelevsky et al recommended [31] that the Scrum value 'focus' helps in delivering qualitative product and improved collaboration among distributed Scrum teams, as result of their case study on application of Scrum and Agile practices in distributed project in 2017. Noll et al [27] performed systematic literature study along with an embedded case study in 2017, to evaluate empirical results on the role of the Scrum Master as servant leader of Scrum team and the role of the Project Manager as a project leader of Project team. They hypothesized both roles and carried out the traditional project manager is well suited for the Product Owner role in Scrum Team; he/she should not be the Scrum Master; as core responsibility of product owner to maximize the product value with the help of other stakeholders, is just like a traditional project manager.

In 2019, Hanslo et al conducted [29] a qualitative

study on challenges for adoption of Scrum and carried out factors proposing conceptual framework using a questionnaire survey. They concluded that Scrum is a trending phenomenon of agile approach in South African organizations and recommended that the organizations adopting Scrum in their software development practices, should consider the factors like management of Sprint, complexity and relative advantages. Korimbocus et al proposed Scrum as a standard framework to cater the challenges for the organizations implementing agile practices. During their survey in Mauritian software development organizations in 2019 [30], they found that knowledge management is crucial factor and the main challenge for the organizations and got recommendations that it is required to improve knowledge management practices like capture, store, share and update the knowledge being discussed in their Scrum meetings. In 2019, Al-Ratrout conducted a case study [32] in web software engineering course teaching using agile approaches like Scrum and XP, to bridge the gap between real world industry experience and academic methodology. He evaluated using the result of case study that agile approach has enabled communication, responsibility, respect and motivation in between students. In 2019, Mathur and Satapathy [34] analyzed comparative study on mobile application development using hybrid agile practices like Scrum and XP, and discovered that agile approaches reduces waste of time and strengthen skills mobile software development modeling techniques. In 2019, Sharma and Hasteer investigated [35] the adoption and popularity of Scrum on the basis of industrial survey and found that the software industry has been migrating from traditional to agile approach. Sisomboon et al proposed the implementation of game theory with Scrum in 2019 [36], to change behavior software development activities and make it fun and motivating by engagement of team members with higher interaction. Al-Ratrout conducted a case study in 2019 [40], to analyze the impact of agile approach in software engineering education to bridge the gap between real world practical environment and education system. She surveyed and investigated that implementation of Scrum and XP both influenced teaching process and motivated the students. Ganesh and Narayanan conducted a research in 2019 [41], on implementation of Scrum for the development of ERP (Enterprise Resource Planning) material management feature and concluded that Scrum is fit for quality product delivery with an incremental approach. Takawale et al in 2019 [10], implemented Scrum as a project development model in academic institute to monitor improvement in the progress of final year project and carried out in their report that the pillars of scrum brought change in teaching and learning methodology of team members and improved the quality of education in the academic institute. In 2019, Hariharan et al concluded during they research [43], that traditional SDLC like Waterfall and other project management practices have limitation which could be resolved with the help of implementation Agile Scrum. Vijaykumar and Ram [44] measured mean score of key parameters of Agile like Collaboration, Knowledge Sharing and Trust, in their Scrum team survey and they evaluated that Scrum has improved score of knowledge sharing and collaboration. While score

for Trust was lower. Mkpojiogu et al measured [46] positive and negative impact of agile Scrum in startup software development organization in 2019, in Saudi Arabia. Along with one negative, they evaluated several positive impacts also like collaboration and motivation in team members for startup and its direct impact on increase in productivity of software product.

VII. SCRUM CASE STUDIES

A case study is known as qualitative research strategy which involves empirical analysis that investigates the careful observation of requirement and result in real life context phenomenon. Following are several case studies of software development organizations that implemented Scrum as an ABPR (Agile Business Process Reengineering) framework for their SEM (Software Engineering Management) processes.

- A. LRN – “Scrum Loaded”
- B. TRUMP – “Scrum Drive”
- C. Scrum Initiative

A. LRN – “Scrum Loaded”

LRN [4] initiated with Scrum to deliver a value to its organization by training their teams and delivered software product 150% more frequently using Scrum. The cross location teams at LRN, using Waterfall along with Scrum, were inconsistent in their Scrum increment and deliverable release. The teams were under direct super vision of project managers who were directly supervising all SEM operations likes planning, scheduling, task and resource allocation, supervised communication and execution; that led overload tasks on project managers and it was the potential point of failure.

The Chief Product Officer was seeking BPR for the teams to align their product development philosophy of high quality value and more frequent software product release by scaling development team collaboration strategy. As a part of ‘Scrum Loaded’, they enabled self-organization for their development teams. They allowed development team members to discuss their mistakes openly in Daily Scrum and ask for suggestive corrections. They identified several key parameters to focus on implementation of change management towards ideal use of Scrum.

As a result of meticulous implementation of Scrum, they ignite culture change in organization. According to Scrum Master at LRN, it has been an incredible opportunity for learning and adapting change and they could see significant improvements up to 150% in qualitative deliverable release of business value to the customers. This has been the start of new phase for LRN towards the target of 200% increment for the customer value deliverables.

B. TRUMPH – “Scrum Drive”

The software development unit at TRUMPH [13] underwent agile transformation, for their software development practices to explore the challenges they had and overcame with the implementation of Scrum.

The head of development division at TRUMPH, initiated a BPR to transform traditional practices of software development into agile approach. They investigated that lack of collaboration was the major issue between the teams and team members; even they already had been using Scrum and resulted inconsistencies about how to use Scrum in their software product development.

They conducted training of team members as an opportunity for them to learn fundamental concepts of Scrum and enable shared understanding, by the Professional Scrum Trainer (PST). This transformation resulted into self-organization, cross functionality and improved collaboration in development team members. Some of the development team members adopted swarming for their daily Scrum practices that helped them in focusing high priority Product Backlog Item (PBI) rather than their own tasks and strengthened the shared responsibility and accountability in the team. The team had also begun applying Scrum values to improve stakeholder involvement into the product vision. Now they are consistent in implementation of Scrum.

LRN and TRUMPH are latest case study reviews available on Scrum.org that represent how Scrum has successfully transformed their SEM (Software Engineering Management) processes and improved customer deliverable values.

C. Scrum – An Agile BPR Initiative

We have had also problems in software product development unit in our software development organization. We had been following traditional approach for software development processes for internal stakeholders' products. We had more than 50% deliverable were always in delay by 2-8 weeks and sometimes projects were went into scrap due to late delivery as our internal stakeholders had already been moved to alternate market place software because of late delivery. We were strongly seeking BPR in software engineering processes.

By seeing the success of LRN and TRUMPH we initiated implementation of Scrum in our software development unit. Initially we faced an issue with matching traditional roles with Scrum roles and we decided as a traditional project manager (Krunal) may be best fit in the role of Scrum Master and the Operational Head as a Product Owner. After completion of several Sprints, we observed improvements in our result but with very low significant of success ratio. We re-analyzed our Sprints that were completed and found out that there were no transparent collaboration and suggestions from development, as they could not discuss their problems openly in Daily Scrum due to presence of a Scrum Master. They could see the shadow of a Traditional Project Manager in a Scrum Master. We redefined the roles in our Scrum team. Traditional Project Manager was assigned a role of a Product Owner; and a Technical Team Lead was assigned a role of a Scrum Master. Also, we decided that only development team members will be present in Daily Scrum meeting.

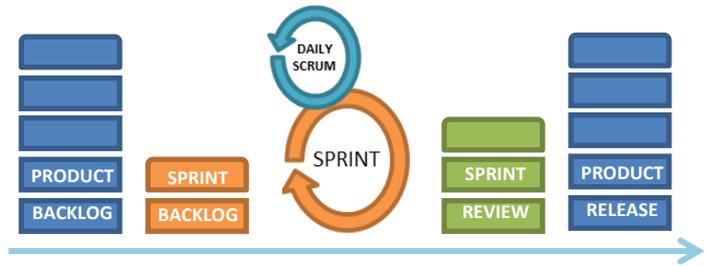


Fig. 2. Scrum Flow

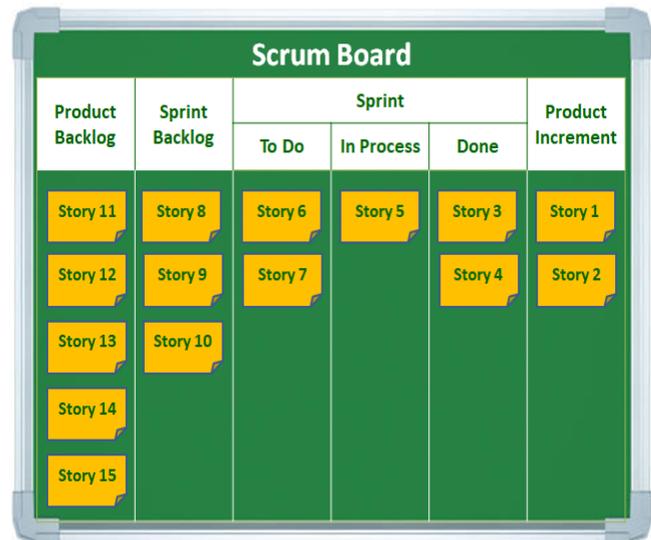


Fig. 3. Work Item Flow using Scrum Board

Figure 2 represents a Scrum process flow for work item. Initially work items are pushed in Product Backlog. Development team pulls work items into Sprint Backlog aiming towards DoD of Sprint. During Sprint Review completed work items are verified and then delivered as a Product increment. Figure 3 shows flow of work items in the Scrum board for each artifact and the state of each work item during the Sprint.

After completion of 8 Sprints of duration of 2-3 weeks; we could achieve the significant improvement in our software product deliverable by reducing delay ratio 50% to 0%. We felt that our decision for implementation of Scrum was correct as Scrum could bring ABPR (Agile Business Process Reengineering) in our Software Engineering practices and Scrum has proven it with the help of its characteristics. We have determined that traditional project manager can be fit into Product Owner role, while technical team leader can be fit into the role of a Scrum Master. We also analyzed and concluded that non-development team member should not be present in the Daily Scrum as it impacts on transparent collaboration of development team.

VIII. RESULTS AND RECOMMENDATIONS

Scrum is a software engineering management framework under the umbrella of Agile and on the base of empiricism and that are core reasons the Scrum is trending as a standard software development approach in software development organizations that we carried out as a result of this research.

The empiricism analysis and its implementation in our software development practices confirm that Scrum is an agile BPR approach for the software development organizations, which are seeking change in software engineering processes.

The case study analysis shows that a proper implementation of Scrum in software development practices has improved the productivity and quality of the developed software product and its features. LRN has carried out 150% to 200% significant improvement for the qualitative deliverable customer values with the help of Scrum. TRUMP has transformed their software development practices using Scrum that enabled self-organization, self-accountability, self-responsibility and improved collaboration in between Scrum team members, which help them in consistent Scrum implementation and remarkable productivity increase during each Sprint. We have reduced delay in deliverable from 50% to 0% with the help of Scrum. We have also analyzed that the openness; collaboration and transparency in development team members have direct impact on productivity.

A. Strength of Scrum

Scrum is easy to use but complex to master. We have determined that following characteristics (represented in tables, followed by the explanation for each) are strength of Scrum and makes it the very suitable for software engineering practices.

Table I: Scrum Artifacts

Artifacts	Characteristics
Product Backlog	Transparency, Priority Order
Sprint Backlog	WIP Limit, Work Items
Increment	Releasable Feature

- Table I represents Scrum Artifacts and their characteristics.
 - Product backlog is a transparent ordered list of requirements to be considered to shape and build a product.
 - Sprint backlog is a list of work items to be done during the Sprint and result is known as DoD.
 - Increment is a small unit of working feature of work item, which is ready to release on production environment.

Table II: Scrum Values

Values	Characteristics
Commitment	Self-decision
Courage	Motivations Capability
Focus	Target Product Goal
Openness	Visibility and Transparency
Respect	Virtual Bonding and Team Building

- Table II represents Scrum Values and their characteristics.
 - Commitment is a self-decision of each team member of team towards achievement of product goal for each Sprint.
 - Courage enables capability of team members to solve all types of impediments during the Sprint.
 - Focus improves strength of team collaborate to achieve the product goal.
 - Openness enables transparency and trust in team by sharing thoughts, issues or impediments with team

members.

- Respect is the base of team building structure with virtual bonding of respect for each other.

Table III: Scrum Pillars

Pillars	Characteristics
Inspection	Validation and Rectification
Adaptation	Improvement and Opportunity
Transparency	Visibility of all aspects of Scrum

- Table III represents Scrum Pillars and their characteristics.
 - Inspection is an opportunity for a Scrum team to validate result of Sprint events.
 - Adaptation allows adjustment to be accepted and made as soon as possible to minimize deviation.
 - Transparency enables significant visibility for all aspects and work item states to each team member of Scrum, involved and responsible for the outcome of Sprint.

Table IV: Scrum Roles

Roles	Characteristics
Product Owner	Product Backlog Management
Scrum Master	Servant Leadership
Development Team	Self-organizing and Cross functional

- Table IV represents Scrum Roles and their characteristics.
 - Product Owner is responsible for managing visibility of product and its priority for development.
 - Scrum Master accountable for supporting and promoting implementation of Scrum in organization.
 - Development Team is self-organizing for resolution of any impediments during the Sprint.

Table V: Scrum Events

Events	Characteristics
Sprint	Work-In-Process Limit
Sprint Planning	Estimation and Definition of Done
Daily Scrum	Collaboration and Team Bonding
Sprint Review	Inspection and Rectification
Sprint Retrospective	Opportunity of Increment

- Table V represents Scrum Events and their characteristics.
 - Sprint controls pulling mechanism which limits overload of inflow.
 - Sprint planning allows self-organizing estimation of each state of work item that enables transparency and progress towards DoD (Definition of Done).
 - Daily scrum improves collaboration between development team members and share their thoughts with each other.
 - Sprint review allows inspection of the development work and feedback by Scrum team along with other stockholders.
 - Sprint retrospective provides an opportunity to analyze impediments for the current and adopt increments for the next Sprint.

B. Recommendations

The result of this research strongly recommends that the ideology and characteristics of Scrum makes it very suitable for software engineering management processes. We also recommend that Scrum is the better framework than traditional.

But we also suggest that software development organizations should consider about the limitations of organization and its structure, while adopting Scrum in software development practices. The critical analysis of case studies represented in this research, opens the next door of the research to focus on how organizations can successfully adopt Scrum framework. Krunal et al [25] have stepped into this direction; to discover and elicit the challenges with Scrum in Software Engineering Management (SEM) and concluded that it's time for an Agile Business Process Reengineering (ABPR) to improve the productivity of Software Engineer (SE) practices by integration of Scrum with other agile based framework like Kanban in the form Scrumban [42] as a hybrid integration of Scrum and Kanban as well as an integration of traditional SDLC like Waterfall with Scrum and Kanban in the form of Scrumbanfall [45] as a hybrid framework.

IX. CONCLUSION

Business Process Reengineering (BPR) has becoming essential in Software Engineering Management (SEM) for software development organization nowadays and the same we have discussed with the help of empirical literature reviews and case studies about Scrum. This research has carried out the reasons: why Scrum is trending as an agile SEM framework in software development organizations; and that are the key parameters and characteristics of Scrum like: Pillars, Values, Roles, Events and Artifacts; and all of these characteristics are the base of empiricism control theory.

We have also concluded that Scrum has been helping organizations in transformation of their business by initiating responsibilities and accountabilities; by creating courage and respect; by improving collaboration and self-organization; in between the team members, which has resulted in increase of core values of business, quality of deliverable software product features and customer satisfaction.

X. FUTURE ENHANCEMENT

It's time for a deep dive into the Scrum and needs investigation, what are the issues with Scrum or challenges for understanding Scrum in Software Development Organizations, which aren't discovered and discussed yet. Also need to find out what will be the challenges for Scrum in the era of AI (Artificial Intelligence) and ML (Machine Learning) and how the Scrum will overcome these challenges. Is Scrum ready for integration with advanced technologies and their benefits?

ACKNOWLEDGEMENT

As a part of my academic affiliation with Indus University for the doctorate degree program in Computer Science and Engineering, through this empirical analysis we have

attempted to highlight the positive characteristics of Scrum that have improved productivity of software product development and management practices. The challenges for SEM (Software Engineering Management) with Scrum will be our next goal.

ABBREVIATIONS

- A**
ABPR – Agile Business Process Reengineering
AI - Artificial Intelligence
AM – Agile Methodology
- B**
BPR – Business Process Reengineering
- D**
DoD – Definition of Done
- E**
ERP – Enterprise Resource Planning
ESE - Empirical Software Engineering
- G**
GSD – Global Software Development
- M**
ML – Machine Learning
- O**
OOP – Object Oriented Programming
- P**
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**
WIP – Work in Process
- X**
XP – Extreme Programming

REFERENCES

1. K. Schwaber and J. Sutherland, "The Scrum Guide™", Scrum.org, 2017
2. K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development", AgileManifesto.org, 2001
3. P. A. G. Permana, "Scrum Method Implementation in a Software Development Project Management", International Journal of Advanced Computer Science and Applications, vol. 6, issue 9, pp. 198-204, 2015.
4. R. Kirpane and G. Vora, "LRN Transitions to Scrum to Deliver More Value with its 'Scrum Loaded' Initiative", Scrum.org, 2019.
5. 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
6. H. K. Flora, S. V. Chande and X. Wang, "Adopting an Agile Approach for the Development of Mobile Applications", International Journal of Computer Applications, vol. 94, issue 17, pp. 43-50, 2014.
7. H. Takeuchi and I. Nonaka, "The New Product Development Game", Harvard Business Review, 1986.
8. J. Sutherland and K. Schwaber, "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework", Scrum Inc., draft: 29 Jan 2011, Paris.

9. K. Schwaber, "Agile Project Management with Scrum", Microsoft Press, Redmond, WA, USA. 2004, ISBN:073561993X
10. N. Takawale, S. Kulkarni 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
11. P. D. Chatzoglou, N. Theriou, E. Demetriades and V. Aggelidis, "Software project management and planning: the case of the Greek IT sector", International Journal of Applied Systemic, vol. 1, issue 3, pp. 305-316, 2007, DOI: 10.1504/IJASS.2007.017713
12. E. Cardozo, B. Araújo, J. F. Neto, A. Barza, C. França and F. Q. B. da Silva, "SCRUM and Productivity in Software Projects : A Systematic Literature Review", Conference: International Conference on Evaluation & Assessment in Software Engineering (EASE), April 2010, DOI: 10.14236/ewic/EASE2010.16
13. T. Schissler, "How TRUMPF Used Scrum to Drive its Agile Transformation", Scrum.org, 2019.
14. 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, 2019. ISSN: 2249-8958
15. J. Cho, "Issues and Challenges of Agile Software Development with Scrum", Issues in Information Systems, vol. 10, issue 2, pp. 188-195, 2008.
16. E. Hossain, M. A. Babar and H. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review", Fourth IEEE International Conference on Global Software Engineering, IEEE Computer Society, pp. 175-184, 2009. DOI 10.1109/ICGSE.2009.25
17. A. Santos, A. Sales, P. Fernandes and M. Nichols, "Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development", 15th annual conference on Innovation and Technology in Computer Science Education (ITiCSE), Vilnius, Lithuania, vol. 1, 2015
18. T. Dyba and T. Dingsoyr, "Empirical studies of agile software development: A systematic review", Information and Software Technology, vol. 50, issue 9, pp. 833-859, 2008.
19. D. I. K. Sjoberg, T. Dyba and M. Jorgensen, "The Future of Empirical Methods in Software Engineering Research", Future of Software Engineering (FOSE '07), Minneapolis, MN, 2007, pp. 358-378. DOI: 10.1109/FOSE.2007.30
20. T. Dyba, "Improvisation in Small Software Organizations", in IEEE Software, vol. 17, issue 5, pp. 82-87, 2000. DOI: 10.1109/52.877872
21. S. Nerur, R. Mahapatra and G. Mangalaraj, "Challenges of migrating to agile methodologies", Communications of the ACM, vol. 5, issue 5, pp. 73-78, 2005.
22. A. Cockburn and J. Highsmith, "Agile Software Development: The Business of Innovation", in IEEE Computer, vol. 34, issue 9, pp. 120-122, 2001. DOI: 10.1109/2.947100
23. L. Williams and A. Cockburn, "Agile Software Development: It's about Feedback and Change", in IEEE Computer, vol. 36, issue 6, pp. 39-43, 2003. DOI: 10.1109/2.947100
24. L. Rising and N. S. Janoff, "The Scrum Software Development Process for Small Teams", IEEE Software, vol. 17, issue 4, pp. 26-32, 2000, DOI: 10.1109/52.854065
25. 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 3, 2020. ISSN: 2278-3075. (Submitted. Under Review.)
26. D. B. Ramos, I. M. M. Ramos, W. S. Viana, G. R. Silva and E. H. T. Oliveira, "On the use of Scrum for the management of research oriented projects", New Ideas in Educational Informatics, vol. 12, pp. 589-594, 2016
27. J. Noll, M. A. Razzak, J. M. Bass and S. Beecham, "A Study of the Scrum Master's Role", International Conference on Product-Focused Software Process Improvement, 2017, DOI: 10.1007/978-3-319-69926-4_22
28. R. Morien, "An Agile Software Project Management Manifesto - A Reference Disciplines Framework for Agile Development", International Journal of Advances Soft Computing and its Application, vol. 6, issue 1, pp. 01-19, 2014
29. R. Hanslo, E. Mnkandla and A. Vahed, "Factors That Contribute Significantly to Scrum Adoption", 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 2019, pp. 813-821. DOI: 10.15439/2019F220
30. M. A. Korimbocus, T. S. Towokul and S. D. Nagowah, "Scrum Meeting Tool for Knowledge Capture and Sharing", 2019 Conference on Next Generation Computing Applications (NextComp), Mauritius, 2019, pp. 1-6. DOI: 10.1109/NEXTCOMP.2019.8883672
31. Y. Khmelevsky, X. Li and S. Madnick, "Software development using agile and scrum in distributed teams," 2017 Annual IEEE International Systems Conference (SysCon), Montreal, QC, 2017, pp. 1-4. DOI: 10.1109/SYSCON.2017.7934766
32. S. Al-Ratrou, "Impact of using Agile Methods in Software Engineering Education: A Case Study", 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 2019, pp. 1986-1991. DOI: 10.1109/CoDIT.2019.8820377
33. K. Angelov, "Research for Usefulness of Agile Methods in Creative Business", 2019 International Conference on Creative Business for Smart and Sustainable Growth (CREBUS), Sandanski, Bulgaria, 2019, pp. 1-5. DOI: 10.1109/CREBUS.2019.8840091
34. B. Mathur and S. M. Satapathy, "An Analytical Comparison of Mobile Application Development using Agile Methodologies", 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1147-1152. DOI: 10.1109/ICOEI.2019.8862532
35. S. Sharma and N. Hasteer, "A comprehensive study on state of Scrum Development", 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 867-872. DOI: 10.1109/CCAA.2016.7813837
36. W. Sisomboon, N. Phakdee and N. Denwattana, "Engaging and Motivating Developers by Adopting Scrum Utilizing Gamification", 2019 4th International Conference on Information Technology (INCIT), Bangkok, Thailand, 2019, pp. 223-227. DOI: 10.1109/INCIT.2019.8911976
37. 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 3, 2020. ISSN: 2249-8958. (Submitted. Under Review.)
38. J. Medapati, A. C. Jasti 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.l8756.0881019
39. G. M. Kapitsaki and M. Christou, "Where is Scrum in the current Agile world?", 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Lisbon, 2014, pp. 1-8.
40. S. Al-Ratrou, "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. N. Ganesh and R. C. Narayanan, "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. K. Bhavsar, V. Shah and S. Gopalan, "Scrumban: An Agile Integration of Scrum and Kanban in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 4, 2020. ISSN: 2278-3075. (In process)
43. S. Hariharan, A. Rengarajan 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
44. V. Ram and T. Vijayakumar, "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
45. K. Bhavsar, V. Shah and S. Gopalan, "Scrumbanfall: An Agile Integration of Waterfall, Scrum and Kanban in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 4, 2020. ISSN: 2278-3075 (In process)
46. E. O. C. Mkpjojiogu, 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.

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 paper. 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 paper. 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.

¹Krunal 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