

Latest Transformations in Scrum: A State of the Art Review

Sara Ashraf, Shabib Aftab

Department of Computer Science, Virtual University of Pakistan
Email: {sara22.ashraf, shabib.aftab}@gmail.com

Abstract—Owing to a big deal of benefits that Agile process models offer to the software industry, they have been the center of attention for a couple of decades for researchers. Scrum has emerged as one of the most prevalent contemporary Agile approaches. It's adaptive and versatile nature makes it appropriate for adoption. Experts have been experimenting and tweaking the practices for last many years to enrich the Scrum. This paper is intended to provide the latest insightful understanding of how the Agile Scrum tailored and adapted in different areas for software process improvement that in turn lead to increased productivity and product quality. A research strategy has been designed to extract the literature since 2016, based on pragmatic transformations of Scrum, subsequently gaining the in-depth perception that is presented in the paper as a comprehensive review and the outcomes are discussed. This work will contribute a state-of-the-art objective summary from which advance research activities can be planned and carried out.

Index Terms—Agile, Scrum, Tailored Scrum, Improved Scrum, Transformations in Scrum, Customized Scrum, Systematic Literature Review.

I. INTRODUCTION

Adapting to inconsistent environments is essential. Thus, Agile philosophies emerged as a remedy for the orthodox customs of development [12] [13]. Agile software development models have proven their claim valid for better handling of the dynamic business environment [14, 15, 16, and 18]. Agile Software Development Ecosystems (ASDEs) [16] comprising of Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Crystal Clear, Test Driven Development (TDD) etc. They share the common philosophy based on 04 values and 12 principles collectively named as Agile Manifesto [9] [10]. Hence, Scrum is not the only Agile, though it has won the Agile war [11]. According to Schwaber et al. [17], the formal definition of Scrum is:

“A framework within which people can address complex adaptive problems while productively and creatively delivering products of the highest possible value”.

With increased productivity, reduced time to market the product, better collaboration, continuous feedback,

and ability to embrace change, Scrum is recognized as most used agile model currently [42]. According to [23], 61% of respondents who were from 76 countries use Scrum. For dealing with growing complexity of projects, Scrum offers an extensive range of management practices [45]. However, in case of engineering practices, it gives the liberty of adaptations by not defining them explicitly. Configuring a standard software process by adapting it in compliance with a project or organization's particularities is called tailoring [37] [38]. Projects are unique in nature, so same method fits all philosophy doesn't exist [35]. Hence, tailoring of Scrum process model is required for accomplishing a project successfully and meeting the objectives. Planning, design, software quality assurance, testing, team performance, and communication are some key areas where practitioners confront challenges that need to be tackled prudently as Scrum process practices do not support them formally. Different researchers and practitioners have been making efforts to evolve Scrum for last decade to address these challenges. Most of them tried to optimize the traditional Scrum with a more pragmatic adoption of agile philosophy.

In this paper, a comprehensive analysis of the latest enhancements in Scrum process model is being presented in such a way that Section II abridges some related work, Section III elaborates the protocol of Systematic Literature Review (SLR), Section IV presents the critical review of the selected empirical studies of tailored Scrum versions, Section V discusses the results, and Section VI will conclude the paper.

II. RELATED WORK

Different secondary and tertiary studies have been conducted within the domain of Agile Software Development (ASD) during last years. Some of these tertiary studies emphasized on investigating Agile practices along with their corresponding effects on project constraints [31], some premeditated the role of Agile for Global Software Engineering (GSE) and current trends [41], while others identified several research areas within ASD [33].

Certain systematic reviews examined the pieces of evidence that support the benefits and weaknesses of agile methods [24, 29, and 47]. Some explored their integration with User-Centered-Design (UCD) [28] [49] and Requirements Engineering (RE) [32]. Karvonen et al. [30] examined the area of Agile Release Engineering

(ARE). Pedreira et al. [37] presented a literature review for identification of software process tailoring approaches.

Various secondary studies were mainly focused on Scrum, its evolution, adoption, and usage trends in the industry [42] [44]. Some consolidated empirical evidence to reveal the association between Scrum and increased productivity [29]. Pino et al. [34] studied Scrum in context of Software Process Improvement (SPI) for small organizations, similarly, they analyzed the prevailing SPI approaches for SMEs [26]. Hossain et al. [25] conducted an SLR that seeks to identify challenges that are encountered while using Scrum in GSD. Jalali et al. [27] addressed the similar area by discussing agile practices in GSD. Communication is one of the major challenges for GSD [48], [36] has identified communication practices in this context. Some investigated on how agile transformations can fulfil the needs of large-scale distributed development [39] [40]. It has been found that no study is evident yet which has systematically reviewed the latest pragmatic transformations in Scrum.

III. RESEARCH PROTOCOL

In a systematic literature review, the evidence-based-paradigm supports objective assessment and empirical result generation in relevance to a specified research question [20].

The research protocol specifies the procedure to be followed for review; it is a detailed plan to be adopted along with any boundary conditions or any criteria to apply while selecting primary studies, quality measures, etc. [19].

Taking into account the guidelines presented by Kitchenham and Charters [21], [22] for SLR in SE, a research protocol has been designed. Different steps of our SLR are: i) formulating research questions, ii) identifying keywords to form the Query string, iii) identifying the search space, iv) establishing selection criteria, v) applying these criteria for literature extraction, vi) quality assessment, vii) data extraction and synthesis, and viii) presenting the outcomes, as can be seen in fig. 1.

A. Research Questions

At this step of SLR, research objectives are defined in terms of research questions. There are following Research Questions (RQ) of our study:

- **RQ1:** What are the improved or tailored models of Scrum?
- **RQ2:** Do tailored models provide any practical evidence?
- **RQ3:** What are the recent transformations in Scrum?
- **RQ4:** What are the key areas in which Scrum process model has been improved?
- **RQ5:** What are the practices, roles, or events that have been introduced or innovated in Scrum?

B. Search Space and Query String

Some key terms are derived from our research questions formulated in the preceding step. The keywords have been connected using Boolean operators to design the search string: (agile AND (scrum AND (process OR method) AND (improved OR tailored OR adapted OR customization)))

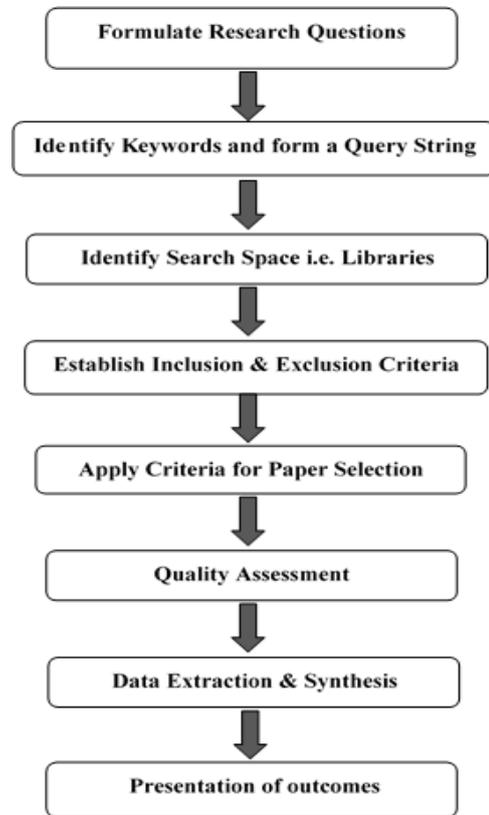


Fig.1. Steps of SLR

As a next step of the process, search space is identified i.e. actually the selection of databases or libraries to consult from. In our case, we've opted for IEEE Xplore, ACM Digital Library, and SpringerLink as can be seen in Table 1. It is important to note that each of these libraries has different characteristics of their search engines. For that reason, slight adjustments were made into query string to make appropriate searches in each of these libraries.

Table 1. Search Space

Sr. No.	Digital Library	Search Scheme	Date Searched
1	IEEE Xplore	title, abstract and keywords	2017-03-20
2	ACM DL	title, abstract and keywords	2017-03-22
3	SpringerLink	title, abstract and keywords	2017-03-25

C. Selection Criteria

We established the selection criteria for extracting the related literature. Selection of papers is based on the following inclusion and exclusion criteria:

1) *Inclusion Criteria (IC):*

Inclusion criteria are based on the following propositions:

- **IC1:** Papers published in 2016 till to date.
- **IC2:** All literature available in journals, conferences, proceedings of conferences, and workshops.
- **IC3:** Papers that include variants of Scrum with a figure of the improved model.
- **IC4:** Papers that have empirical evaluations of the customized Scrum.

2) *Exclusion Criteria (EC):*

Exclusion criteria are based on the following propositions:

- **EC1:** Papers not written in English.
- **EC2:** Papers published before 2016.
- **EC3:** Papers published in a journal other than ACM, IEEE, or Springer.
- **EC4:** Papers under peer review, thesis reports, and those of which full text is not available.
- **EC5:** Specific book chapters or full books.
- **EC6:** Papers that contain reviews, surveys, or analysis of previous work.
- **EC7:** Papers including agile Scrum but not improved process models.
- **EC8:** Papers that do not include any figure of the tailored Scrum.
- **EC9:** Papers based on proposals or just ideas, guidelines or recommendations, and lessons learned.
- **EC10:** Papers that introduced tools to improve agile Scrum process.
- **EC11:** Papers that improve Scrum by integrating other models into it or proposing hybrid models.
- **EC12:** Papers that involve empirical studies of Scrum in other disciplines.

As a result of an initial search based on keywords, we found a large number of articles i.e. aggregate results = 26920. Then, search results were narrowed down by applying selection criteria. A step-wise screening was carried out as shown in fig. 2.

D. *Quality Assessment*

In order to be effective, quality assessment needs to be ascertained. A quality standard has been acquired for comprehending crucial results of pertinent studies.

1. Highly-reputed scientific databases have been selected and most recent work and researches have been included to infer upon.

2. Data extraction and assessment has been done without biases and obscure assertions.

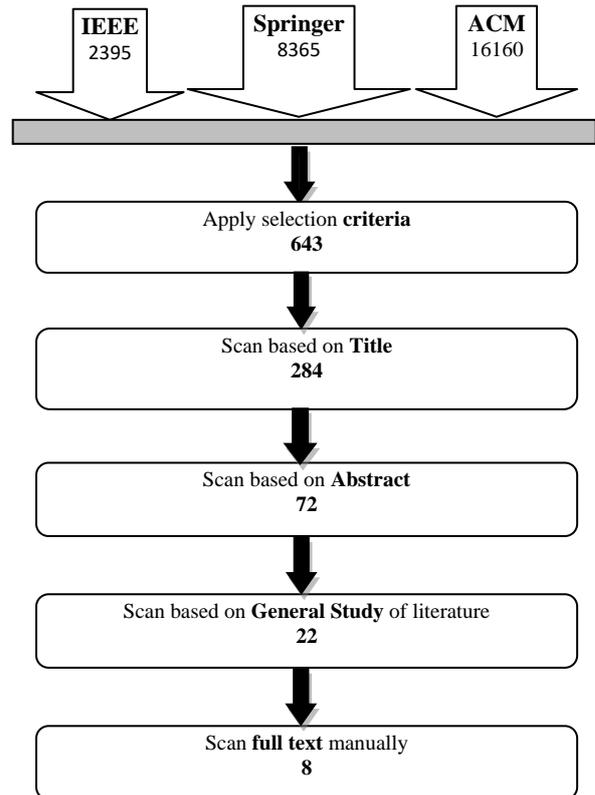


Fig.2. Search Process

E. *Data Extraction & Synthesis*

After applying selection criteria eight most relevant papers have been retrieved, given in Table 2. Extraction of data is carried out in line with the template given in Table 3.

Table 2. Distribution of Research Works by Source

Sr. #	Scientific Database	Type	Selected Research Works	No. of Researches
1.	IEEE	C, C, J	[4], [7], [8]	3
2.	ACM	C, C	[2], [3]	2
3.	SPRINGER	W, C, C	[1], [5], [6]	3
J: Journal, C: Conference, W: Workshop.				

Table 3. Data Extraction & Synthesis Format

Sr. #	Description	Details
1.	Bibliographic Details	Publication year, author, title, type, (journal, conference, workshop), publisher details
Extraction of Data (Objective Results of Study)		

2.	Methodology	(case-study, survey, experience)
3.	Nature of case-study	small, large / academic, industrial
4.	Model	Model title/ name
5.	Project	Name of Project
Synthesis of Data (Objective Results of Study)		
6.	Roles/ Artifacts/ events evolved/ introduced	PO, SM, QA, etc. / Product Backlog, Sprint Backlog, etc. / Daily-standup, Sprint Review, etc.
7.	Areas addressed	Identification of Areas where improvements are made.
8.	Outcomes	Success factors achieved.

IV. TRANSFORMED OR TAILORED VERSIONS OF SCRUM

As far as the communication channel is concerned, the models were published in scientific journals, workshops or conference proceedings. Out of eight, six of the included models were published in conference proceedings, one in a scientific journal and one appeared in a workshop. A comprehensive review of the selected eight transformed Scrum models is given as under:

Scrum-Hero is a gamified Scrum process for project monitoring and planning. The customized Scrum is aimed at achieving customer satisfaction, on-time releases, and efficiency and motivation of the team. Souza et al. [1] customized the Scrum framework by mapping the process as a game based on Role Playing Game RPG. The authors proposed the model to increase people involvement by making their tasks amusing so that they would get enough motivation to complete their tasks. They introduced a new taxonomy of game terminologies in Scrum e.g. characters, scores, rewards, challenges, and medals etc. The roles of typical Scrum are mapped as players: Development team as Warriors, Product Owner PO as Product Oracle, Scrum Master SM as Scrum Healer, and Scrum team as Clan’s life where the customers interact as mystical entities. Impediments are termed as Stone Monsters and Sprints as Quests. In Scrum-Hero, there are 3 types of scores: a player has Skill score/ Points SP, Clan has no score, Score points are earned when players accomplish their daily tasks. The players gain Experience Points XP by fulfilling their achievements. There are certain rewards such as medals for related skills, trophies for achievements, and there are real awards given at the end of game or Quest. By achieving these rewards, the players grow in their levels like Hero, Ninja, Knight, Guardian, and Newbie. Scrum-Hero maps the artifacts as: Product backlog is now a wish list of Mystical entities, Sprint backlog is termed as Task-list of Quests, Increment refers to Offerings, Burn down chart remains same, validated increment is Offering accepted. Similarly, the ceremonies of Scrum are mapped as: Quests are the

Sprints such that each with a goal and tasks that Warriors perform. Sprint Planning is now a Quest Planning. Daily Scrum is Daily Challenge, Sprint Review and Sprint Retrospective are now Quest Challenge and Clan-Improvement meeting respectively. The gamified model was then evaluated in real practical settings with 4 Quests and 4 team members i.e. 1 Product Oracle, 2 Warriors, and 1 Scrum Healer. A software prototype named as Scrum-Hero Manager supports data management including Wishes, Skills, Scores, Quests, etc. of players. In order to assess the impact of gamification on team’s motivation, on-time releases, historical data and results of the company are compared with that of the current feedback gathered through the questionnaires and Scrum-Hero Manager. It has been found that in 75% of the Quests, the releases were on-time which was previously 55%.

Limitations:

Scrum-Hero model needs prior training and learning for appropriate adaptation. No explicit guidelines regarding product quality are mentioned within the proposed model. It needs to have a friendly environment for employing this model. It is for small projects and teams only. Results for customer satisfaction and team motivation are not delivered in the paper.

Gupta et al. [2] presented a transformed Scrum model by incorporating innovative practices for any legacy product’s development team employing agile. They analyzed their model through the case study of a multi-location software product. The model was tailored in a way to cope up the challenges of scaling business, technical debt, collaboration, communication, and testing principally. The product under consideration is very complex and has a history of ten years, with huge investments but no improvement in output. The development is distributed mainly at three sites India, Germany, and US. The authors identified three critical factors along with technical competency needed for the product i.e. communication, collaboration, and culture of high performance and continuous improvement. They customized the traditional roles and responsibilities for the adoption of Scrum e.g. Chief Product Owner (CPO), Part Product Owner (PPO), Chief Scrum Master (CSM), Scrum Master-cum-Part-Product Owner (SMPO), and Test Coach. Additional responsibilities of Scrum Masters and Test Coaches are defined. They defined some new events designed to improve collaboration each with a time limit like daily, weekly, and bi-weekly Scrum-of-Scrums. Similarly, some other events are dedicated daily collaboration, virtual testing Stand-up, and peer feedback. In order to repay the technical debt, the authors adapted a 4-stage pragmatic approach of [46]. Identifying technical debt, defining strategy, executing action plan, and validating or handling the technical debt are the 4 stages of this approach. The paper contributes an agile testing strategy that extends the traditional testing (limited to system, integration, and regression testing) to load, unit, API, and performance testing. The pragmatic studies revealed that the modified taxonomy and ceremonies of

transformed Scrum have positive impact on product quality and team's performance. The pragmatic approach helped to decrease technical debts. The time-boxed ceremonies resolved conflicts, improved collaboration that in turn improved team's competency and motivation. The testing strategy has improved product's stability and quality. Also, users scaling, early product delivery, code quality, and customer satisfaction are some other benefits found.

Limitations:

The management needs to have better visibility of tasks rolling from one end to the other. Practices are needed to incorporate in order to improve visibility and transparency. There is no shortcut to deal with critical issues appearing in between the Sprints except waiting for the next sprint to get ranked. Also, practices are needed to improve the team's performance.

Scrum has been scaled by different researchers for distributed development and teams but these variants of Scrum are not competent to manage the synchronized releases across different platforms i.e. same application needs different front-ends across different platforms simultaneously. Sithole and Fritz [3] customized the Scrum for synchronized cross-platform releases. The proposed model is, in fact, an extension of the model Scrum at Scale by Brown and Sutherland [43] which is a modular framework with ten such modules that offer flexibility by supporting context-based solutions. The authors introduced a set of optional practices that may target the client application with synchronized development. The synchronized Scrum model was validated through a case study based on Android and iOS client applications development for the South African government department. The client demands for a concurrent release of the above-mentioned applications. There were 2 teams each of 3 developers working on each client application. The authors of the model tweaked nine modules out of ten to make them work for synchronized releases. Different practices customized among these modules are: empowered product owner in strategic vision module, product-owner team partnering in backlog prioritization module, requirement analysis teams in backlog decomposition and refinement module, and the practices of flexible synchronized releases and tightly managed deliverables in release planning and management module. In team level process module, different Scrum Masters were assigned to the client and server-side development teams. In feedback module, the practice of product owner filtration was selected so that synchronized teams and customers can have a single communication channel. Continuous improvement module uses sprint retrospective, and impediments removal module includes daily Scrum, team level resolutions, center of excellence, and executive level practices. Cross-team coordination module enforces norms, shared standards and guidelines among teams. Shared dashboard serves the transparency module and metrics module uses ad hoc practice so that team may choose metrics and tools randomly. By applying

Mulder's success criterion [44] the synchronized Scrum model was evaluated for its compliance and found 82.22% rating that is quite satisfactory. Different metrics were used to measure the performance of the teams categorized under 4 different levels: a) the input b) activity c) output d) outcome. During this case-study certain challenges were encountered that may put synchronized releases at risk, some of them are team dynamics and difference in team profiles, and lack of team interference regulation.

Limitations:

The current study is confined to a limited set of principles so can be extended further with more practices and parameters. Moreover, it is evaluated for two small teams with synchronized releases, so needs to be validated for large-scale synchronized developments.

Rahayu et al. [4] enriched the Scrum by incorporating usability testing into it. The proposed model is intended to cope up the 3 key challenges encountered in Scrum process that are: a) longer user feedback loops, b) no user involvement in testing, and c) lack of quality product. A research methodology comprising of 5 stages was designed to investigate if Scrum combined with usability testing can handle the above-mentioned challenges. According to this strategy, at first, identify the research problem, then review the literature, develop and gather data, then analyze the solution and finally infer the results. The authors integrated the usability testing within the Sprint Review event of Scrum, thereby inviting both the users and stakeholders for collecting the feedback from them. In this way, the results of usability testing will guide the development team and in turn, lead to improved product quality. This feedback can help in refining product backlog and in next Sprint planning meeting. Similarly, the output of next Sprint will have better customer satisfaction. For tailored Scrum model's evaluation, a case-study based on 'Lembaga Asisten' information system was conducted. Testing Strategy was supported by 2 instruments i.e. Post-Study System Usability Questionnaire PSSUQ and usability scenario. Firstly, the users followed a set of activities planned in usability testing scenario then they are asked about PSSUQ. It has been found that feedback obtained from users is more comprehensive than the key stakeholders' feedback. Moreover, the usability testing response is more structured. Rather than waiting for the end product and then making changes, the development team gets an earlier insight of the product and ensures its success. It has not only improved the user involvement but also builds their trust and confidence in the product. Also, the team gets motivated through the user's acknowledgment. Thus, the model has successfully achieved most of its goals.

Limitations:

The assessment tool PSSUQ for usability testing needs to be improved by adding more indicators into it. Automated testing tools should be used.

Hanssen et al. [5] presented a variant of Scrum, named as SafeScrum, which supports the development of a safety-critical software by borrowing some XP practices. The SafeScrum ascertains that the developed software is in accordance with the safety standard IEC61508. The model maintains two backlogs, one for functional requirements and the other for security requirements. Also, analyzes the dependence of former on the latter. By satisfying the needs of safety-critical software development an additional overhead is introduced that may compromise the concept of light-weight of an agile framework. Automated tools were required to be introduced in the process for making it efficient and effective. Therefore, the authors have designed a tool-chain to support the SafeScrum that includes the tools for documentation, quality management, workflow, and process traceability. The authors have been employing and adapting the SafeScrum for an industry project of fire and gas detection system for almost two years, with the intention to attain a Safety-Integrity-Level SIL3 of the aforementioned standard. The data was collected through meetings' observations, interviews of the team, and analysis of documentation. Moreover, a TUV organization played a role of external assessor in the process. The process was initiated simply with a team of 5 co-located members, while Sprint was spanned over 4 weeks. The process was refined gradually with each Sprint, after a few Sprints automated tools replaced the manual effort for managing workflow, team collaboration, documentation, and quality assurance. By configuring the tool-chain in the process the team was successfully producing and maintaining the required information. But after a few months, there was a need to reinforce the QA function as it was overloaded and needs to be addressed. Hence, authors recommended adding the QA role in the Scrum team explicitly to avoid adding more work to the method. Furthermore, QA is supposed to perform four simple and specific tasks for checking test coverage, documentation coverage, code-traceability, and code-metrics values.

Limitations:

The role of QA is very crucial for carrying out the SafeScrum process successfully in compliance with to the standard so needs to be refined and streamlined. The project has not completed yet therefore, the developed software is not certified.

Ardito et al. [6] improved the Scrum framework by integrating Human-Centered Design HCD practices into it. This tailored HCD-SCRUM is aimed at minimizing the limitations of Scrum by systematically incorporating stakeholder's feedback and input into the development. The authors discovered four common principles between HCD and Agile software development which are: i) user involvement, ii) prototyping, iii) continuous testing, and iv) iterative design. The HCD-SCRUM was evaluated through a case study in SER&P, an Italian Small-Medium Enterprise SME with the main focus on rapid code development, iterative refinement, constant customer involvement, and high frequency of releases. The key

features of the HCD-SCRUM are: a) Customer Committee comprising of 1 PO selected from Scrum Team, 2 or more people from customers, and a business-domain expert, b) Inception: evaluate and rank the features to be built, c) Sprint n.0: a high-level prototype is developed, d) SCRUM island: it's a working desk of 4 seats, e) 1 week time-boxed Sprint, f) (IN)Sprint Review: a Verification & Validation V&V continues throughout the Sprint, g) Project Retrospective contains Sprint Retrospective. For the Scrum-based process management, SER&P employs Application-Lifecycle Management ALM tool. Moreover, Kiuwan tool is being used for product quality assessment. The tailored Scrum was applied to develop a web portal related to Public-Accounts Project in SER&P. It's an 18-month project with 5 people working on it. Seventeen Sprints have been completed within 10 months of the project, to this end the quality evaluation results show that all the quality characteristics are satisfactory except maintainability and portability. Findings of the experiment reveal that model improves the customer involvement and their feedback is vital for HCD.

Limitations:

The tailored model HCD-SCRUM doesn't provide portability and maintainability. It is appropriate only for small teams. Larger and distributed teams cannot use it without further alterations.

Harvi and Agah [7] contributed a novel approach to modify Scrum through inspiration from mission command. According to authors, mission command and software engineering both have the same complex and persistently varying nature. They hypothesized that planning, communicating, and prioritizing of requirements in a software development process can be improved through incorporating the military command based approach. Targeted Scrum is a variant of Scrum software process model and reflects 03 inspirations from Mission-Command i.e. Targeting, Line of Effort, and End State. Targeted Scrum targets the challenges of preliminary planning and overall design architecture. Before creating Product Backlog, an initial-planning meeting is introduced in the tailored model. Afterward, an architecture of the program is planned and assessed to develop Product backlog along with the prioritization of requirements to develop further Sprint Backlogs. Here, Line of Effort (LOEs) plays a vital role in communicating project's prioritization and architecture and lead ultimately to the end state. Authors claimed that Targeted Scrum is better in 04 ways: 1) by establishing Product Design framework, 2) by grooming Product Backlog through Product Design Meeting and LOEs, 3) by improving Sprint Planning through LOEs, and 4) by communicating the progress of Product to stakeholders to synchronize the efforts. Agenda for the preliminary Product Design Meeting and subsequent Product Design Meeting is described by the authors. Former is no more than 02 hours, and latter is limited to 01 hour. To evaluate the contribution of Targeted Scrum, authors conducted a case-study through undergraduate level students of

software engineering. Two projects of almost same difficulty level were developed by students. There were total 26 divided into 3-4 per team, 04 of the groups followed Targeted Scrum and rest of the 03 groups employed Traditional Scrum. They had a 01-month time-box to complete and 02-week sprint. Authors through a survey assessed the overall impact of these methodologies on the software development team. Planning, software architecture development, requirements prioritization, and communicating progress were analyzed for both the methodologies. Similarly, developed products were analyzed along with their source code. Results were in favor of Targeted Scrum for the planning and requirements prioritization. On the other hand, Targeted Scrum could not improve product quality plus internal and external communications of development teams. However, product quality assisted by Targeted Scrum, revealed improvement signs in the middle of the continuum.

Limitations:

The limited dataset, academic calendar, and academic environment were the limitations of this empirical study. Due to military setup, more meetings and more formalism are there that may seem to affect the process agility. The experiment was conducted with 26 students for the period of one semester. Although it provided a great insight into Targeted Scrum but the same transformed model should be evaluated through real-time experiments with more software developers. A survey with a large set of questions can be helpful in the analysis of robust data set.

Padmini et al. [8] acquainted Scrum framework with the concept of User Acceptance Testing (UAT) to increase efficiency, productivity, and faster delivery. As most of the UAT happenings are actually an integral part of agile practices, therefore, it was easy to incorporate the UAT in Scrum as well as in the field. Business users conduct UAT to confirm product fitness, while this testing involves both functional as well as non-functional aspects of testing. The paper mainly focuses the functional testing. The authors systematically gathered and analyzed the data to improve the process and streamline UAT into it. They identified the major causes of less productivity and low level of efficiency of the quality assurance team. Therefore, the proposed approach stresses on team-wise transformation and subsequently improving the process. Authors assessed the tailored model to a complex, large-scale Revenue Management System (RMS) with 100 domain experts as UAT team. The system comprising of a web portal with multiple back-end applications planned in 3 phases. The case study is confined to phase-1 encompassing development of 16 modules. Initially, training sessions and workshops were conducted to enhance the team's knowledge about their roles and responsibilities. Separate backlogs were maintained as Release backlogs, Module backlogs and Scenario backlogs for each release and modules in that release. There were 2-week sprints defined within a release sprint of 4-week. The ceremonies of the Scrum process were followed with slight changes. The process

elaborated defect management tracking. The quality of UAT process was also evaluated in 4 steps. According to the follow-up survey findings, there was a significant increase in team's performance, analyzed by using Defect-to-Remark ratio (DRR). The process gradually got streamlined. Authors recommended introducing UAT earlier in the process to avoid project failure. Providing early feedback, reduces the risk of failure to fulfil user needs and managing change in technology as well as user requirements, reduces time and cost of rework, helps improving client's trust, satisfies business needs. Also, team performance was improved in terms of their motivation, efficiency, and productivity.

Limitations:

The project has not completed yet, the findings of assessment are taken from the first release only. Rest of 2nd and 3rd releases are under process. The testing procedure needs to be refined further.

V. RESULTS AND DISCUSSION

We have found 08 papers through the search process based on the inclusion and exclusion criteria stated in Section III. By in-depth exploration and analysis of the selected papers, we have found the following answers to the given Research Questions (RQs):

RQ1: What are the improved or tailored models of Scrum?

The retrieved 08 [1-8] papers are the most appropriate and are strictly in compliance with our selection criteria. These researches have proposed improvements and tailored the traditional Scrum process model (Table 4 & Table 5).

RQ2: Do tailored models provide any practical evidence?

All of these research works are validated through case studies either conducted in controlled settings or in industry.

RQ3: What are the recent transformations in Scrum?

Table 4 shows an overview of the recent transformations in Scrum.

RQ4: What are the key areas in which Scrum process model has been improved?

It has been observed that researchers have adapted Scrum model in the following areas: planning, design architecture, team performance, customer involvement, testing, product quality, time to market the product, critical software development, communication etc. as shown in Table 5.

RQ5: What are the practices, roles, or events that have been introduced or innovated in Scrum?

According to outcomes of this SLR, each tailored model [1-8] has contributed either by introducing new roles, events, and artifacts or have tailored them to address the challenges. Table 4 can clearly represent these roles, events, and artifacts.

Limitations of Research:

There are following limitations of our research:

1. Despite the fact that all the published literature are retrieved through a precise and rigorous search process that ascertains the completeness of our study, even then there is a probability to miss out some relevant work.
2. The transformed models are mostly evaluated through a single case-study, therefore, their results might not be generalized to other practical settings. It might affect the interpretation of our results.

Table 4. Summary of the Selected Transformed-Scrum Models

Models	Scrum-Hero	Transformed-Scrum	Synchronized-Agile Scrum	Usability Testing in Scrum	SafeScrum	HCD-Scrum	UAT into Scrum	Targeted Scrum
Areas addressed	Team performance & motivation, Customer satisfaction, On-time releases.	Scaling business, Technical debt, Collaboration, Communication, and Testing.	Cross-team coordination, Synchronized cross-Platform releases.	Longer User feedback loops, User involvement in Testing, and Quality product.	Quality assurance for safety-critical software development, achieving SIL-3, traceability of requirements	Rapid code development, iterative refinement, constant customer involvement, and high frequency of releases.	Team's efficiency, productivity, and faster delivery	Preliminary planning & overall design architecture, communication, prioritization,
Nature of Case-study	Small-scale, industrial	Complex, large-scale, industrial	Large-scale, industrial	Small-scale, academic	Large-scale, industrial	Large-scale, industrial	Complex, large-scale, industrial	Simple, small-scale, academic
Project	--	GCP Global Configurator Project	2 client applications for iOS, and Android platform	Lembaga Asisten information system	Fire-&-gas Detection system SIL-3	Public-Accounts Project	Revenue-Management System (RMS)	i) POS for restaurant ii) veterinary clinic
Roles added/ adapted	Warriors, Product Oracle, Scrum Healer, Clan's life	PPO, Test Coach, CPO, CSM, SMPO	Single SM for all teams, PO team-partnering	usability tester (user)	QA engineer	Customer Committee	UAT planning team (Domain experts), UAT execution team,	PO & SM
Practices/ events introduced/ adapted	Quests, Quest Planning, Daily Challenge, Quest Challenge and Clan-Improvement	Daily, weekly, bi-weekly Scrum-of-Scrums, dedicated daily collaboration, virtual testing Stand-up, and peer feedback.	Joint daily stand-ups, PO filtration, shared sprint retrospective, shared standards	Usability testing during Sprint Review	tasks involving QA function	Sprint n.0, Scrum Island, no Daily-standup, IN-Sprint Review, Project Retrospective, weekly Sprint review	Daily and weekly meeting, Sprint planning, trainings,	Preliminary & subsequent Product Design Meetings, LOEs,
Artifacts added/ adapted	Wish-List, Task-List, Offerings	Common Product Backlog	Common Product Backlog, shared dashboards	PSSUQ and usability scenario	Functional-Product Backlog, Security-Product Backlog	prototype	Release Backlog, Module Backlog, Scenario Backlog, training plans, review repot, defect log	Line-Of-Effort (LOE)
Team size	4 persons	7-8 members	2 distributed teams, each of 3 developers	15 usability testers	5 persons	5 persons	100 domain experts as a UAT team	3-4 per team, total team members 26, 07 teams
Team location	co-located	distributed	distributed	co-located	co-located	co-located	co-located teams	co-located teams
Project organization	Brazil	India, Germany, USA	Govt. department of South Africa	University of Indonesia	Autronica Fire & Securitym Norway	SER& Practices, Italy	public organization of Srilanka	University of Kansas, KS
Sprint duration	--	3 week	--	--	4 week	1 week	2-week Sprint within a 4-week Sprint	2-week
Outcomes/ Success factors	On-time release	Product quality, team's performance, users scaling, early product delivery, code quality, and customer satisfaction	Customer satisfaction, teams' high-level of synchronization, team's performance	User involvement, customer trust in team and product, quality product, team motivation	Quality product, safety standard, better documentation, traceability	Usability, Security, efficiency, portability	Team morale, efficiency, productivity, and faster delivery	Progress visualization, requirements planning and prioritization,

Table 5. Areas improved in Scrum/ Challenges Addressed

Year	Papers	Areas improved / Challenges Addressed
2017	Scrum Hero: Gamifying the Scrum Framework (Springer) [1]	<ul style="list-style-type: none"> ▪ Team performance and motivation, ▪ Customer satisfaction, and ▪ On-time releases.
2017	Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey A case study in a multi-location legacy software product development team (ACM) [2]	<ul style="list-style-type: none"> ▪ Scaling business, ▪ Technical debt, ▪ Collaboration, Communication, and ▪ Testing.
2016	Synchronized Agile (ACM)[3]	<ul style="list-style-type: none"> ▪ Cross-team coordination, ▪ Synchronized cross-Platform releases.
2016	Applying Usability Testing to Improving Scrum Methodology in Develop Assistant Information System (IEEE) [4]	<ul style="list-style-type: none"> ▪ User feedback loops, ▪ User involvement in Testing, and ▪ Quality product.
2016	Quality Assurance in Scrum Applied to Safety Critical Software (Springer)[5]	<ul style="list-style-type: none"> ▪ Quality assurance for safety-critical software development.
2016	Integration of Human-Centred Design and Agile Software Development practices: experience report from a SME (Springer)[6]	<ul style="list-style-type: none"> ▪ User involvement, ▪ Testing, and ▪ Design.
2016	Applying Agile Practices to Avoid Chaos in User Acceptance Testing: A Case Study (IEEE)[7]	<ul style="list-style-type: none"> ▪ User Acceptance testing ▪ product quality ▪ customer satisfaction
2016	Targeted Scrum: Applying Mission Command to Agile Software Development (IEEE)[8]	<ul style="list-style-type: none"> ▪ communication, ▪ prioritization, ▪ Preliminary planning, and ▪ Overall design architecture

VI. CONCLUSION

A Systematic Literature Review (SLR) has been conducted to find out the most recent transformations in Scrum process model while keeping intact the quality of research protocol and the literature included. Eight papers are identified as most relevant to our criteria. A comprehensive analysis of these studies revealed the areas where adaptations in practices and infrastructure of Scrum may enhance its adoption and applicability. Furthermore, the outcomes of these transformations introduced in Scrum process are presented.

It is evident that practitioners have made efforts to optimize the traditional Scrum process with a more pragmatic adoption of agile and lean principles. It is concluded that Scrum transformations have been undergone in pursuit of delivering better, faster and cheaper software product. However, more empirical studies are needed for the evaluation of these transformations under various project settings to explore new dimensions of research and improvements.

In future, it is intended to further investigate the transformations and their corresponding outcomes that are introduced through integration of other agile process models with Scrum.

REFERENCES

- [1] J. P. Souza, A. R. Zavan, and D. E. Flôr, "Scrum Hero: Gamifying the Scrum Framework," In *Brazilian Workshop on Agile Methods*, pp. 131-135, Springer, Cham, November 2016.
- [2] R. K. Gupta, P. Manikreddy, and K. C. Arya, "Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey A case study in a multi-location legacy software product development team," In *Proceedings of the 10th Innovations in Software Engineering Conference*, pp. 147-156, ACM, February 2017.
- [3] V. Sithole, and F. Solms, "Synchronized Agile," In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, p. 39, ACM, September 2016.
- [4] P. Rahayu, D. I. Sensuse, W. R. Fitriani, I. Nurrohmah, R. Mauliadi, and H. N. Rochman, "Applying usability testing to improving Scrum methodology in develop assistant information system," In *Information Technology Systems and Innovation (ICITSI), 2016 International Conference on*, pp. 1-6, IEEE, October 2016.
- [5] G. K. Hanssen, B. Haugset, T. Ståhane, T. Myklebust, and I. Kulbrandstad, "Quality Assurance in Scrum Applied to Safety Critical Software," In *International Conference on Agile Software Development*, pp. 92-103, Springer International Publishing, May 2016.
- [6] C. Ardito, M. T. Baldassarre, D. Caivano, and R. Lanzilotti, "Integration of Human-Centred Design and Agile Software Development Practices: Experience Report from a SME" In *Integrating User-Centred Design in Agile Development*, pp. 117-135, Springer International Publishing, 2016.
- [7] K. J. Padmini, I. Perera, and H. D. Bandara, "Applying agile practices to avoid chaos in User Acceptance Testing: A case study," In *Moratuwa Engineering Research Conference (MERCCon)*, 2016, pp. 96-101, IEEE, April 2016.

- [8] D. P. Harvie, and A. Agah, "Targeted scrum: Applying mission command to agile software development," *IEEE Transactions on Software Engineering*, vol. 42 no. 5, pp. 476-489, 2016.
- [9] A. Alliance. 2001 "Agile manifesto," [Online]. Available: <http://agilemanifesto.org/> [Accessed 25 03 2017]
- [10] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, and J. Kern, "Manifesto for agile software development," 2001.
- [11] P. O. Koivisto, "New agile process errors in software development," 2010.
- [12] C. Andres, and K. Beck, "Extreme Programming Explained: Embrace Change," Reading: Addison-Wesley Professional, 2004.
- [13] K. Conboy, "Agility from first principles: Reconstructing the concept of agility in information systems development," *Information Systems Research*, vol. 20 no. 3, pp.329-354, 2009.
- [14] A. Cockburn, "Agile Software Development Joins the "Would-Be" Crowd," *Cutter IT Journal*, vol. 15 no. 1, pp.6-12, 2002.
- [15] K. Schwaber, "Scrum development process," In *Business Object Design and Implementation*, pp. 117-134, Springer London, 1997.
- [16] J. A. Highsmith, "Agile software development ecosystems," vol. 13, Addison-Wesley Professional, 2002.
- [17] K. Schwaber, J. Sutherland, and M. Beedle, "The definitive guide to scrum: The rules of the game," Recuperado de: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>, 2013. [Accessed: 15 03 2017]
- [18] B.W. Boehm, R. Turner, "Balancing Agility and Discipline: A Guide for the Perplexed," Addison-Wesley Longman, Boston, 2003.
- [19] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80 no. 4, pp. 571-583, 2007.
- [20] D. L. Sackett, S. E. Staus, W. S. Richardson, W. Rosenberg, and R. B. Haynes, "How to practice and teach EBM," Edinburgh: Churchill Livingstone, 2000.
- [21] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on software engineering*, vol. 28, no. 8, pp.721-734, 2002.
- [22] B. A. Kitchenham and S. Charters, "Procedures for Performing Systematic Literature Review in Software Engineering," EBSE Technical Report version 2.3, EBSE-2007-01, Software Eng. Group.
- [23] <https://www.scrumalliance.org/why-scrum/state-of-scrum-report/2016-state-of-scrum> [Accessed 20 03 2017]
- [24] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review. *Information and software technology*, vol. 50, no. 9, pp.833-859, 2008.
- [25] E. Hossain, M. A. Babar, and H. Y. Paik, "Using scrum in global software development: a systematic literature review," In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pp. 175-184, IEEE, 2009, July.
- [26] F. J. Pino, F. Garca, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," *Software Quality Journal*, vol. 16 no. 2, pp. 237-261, 2008.
- [27] S. Jalali, and C. Wohlin, "Global software engineering and agile practices: a systematic review," *Journal of software: Evolution and Process*, vol. 24 no. 6, pp. 643-659, 2012.
- [28] T. S. Da Silva, A. Martin, F. Maurer, and M. Silveira, "User-centered design and agile methods: a systematic review," In *Agile Conference (AGILE)*, 2011, pp. 77-86, IEEE, August 2011.
- [29] E. S. Cardozo, J. B. F. A. Neto, A. Barza, A. C. C Fran ca, and F.Q. da Silva, "SCRUM and Productivity in Software Projects: A Systematic Literature Review," In *EASE*, April 2010.
- [30] T. Karvonen, W. Behutiye, M. Oivo, and P. Kuvaja, "Systematic literature review on the impacts of agile release engineering practices," *Information and Software Technology*, 2017.
- [31] I. Nurdiani, J. Brstler, and S. A. Fricker, "The impacts of agile and lean practices on project constraints: A tertiary study," *Journal of Systems and Software*, vol. 119, pp.162-183, 2016.
- [32] E. M. Schon, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Computer Standards & Interfaces*, vol. 49, pp.79-91, 2017.
- [33] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Information and Software Technology*, vol. 85, pp. 60-70, 2017.
- [34] F. J. Pino, O. Pedreira, F. Garca, M. R. Luaces, and M. Piattini, "Using Scrum to guide the execution of software process improvement in small organizations," *Journal of systems and software*, vol. 83 no. 10, pp. 1662-1677, 2010.
- [35] D. Firesmith, "Creating a project-specific requirements engineering process," *Journal of Object Technology*, Vol. 3, No. 5, pp.31-44, 2004.
- [36] T. Dreesen, R. Linden, C. Meures, N. Schmidt, and C. Rosenkranz, "Beyond the Border: A Comparative Literature Review on Communication Practices for Agile Global Outsourced Software Development Projects," In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pp 4932-4941, IEEE, January 2016.
- [37] O. Pedreira, M. Piattini, M. R. Luaces, and N. R. Brisaboa, "A systematic review of software process tailoring," *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 3, pp.1-6, 2007.
- [38] J. A. Hurtado Alegre, M. C. Bastarrica, A. Quispe, and S. F. Ochoa, "An MDE approach to software process tailoring," In *Proceedings of the 2011 International Conference on Software and Systems Process*, pp. 43-52, ACM, May 2011.
- [39] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87-108, 2016.
- [40] J. M. Bass, "Large-scale offshore agile tailoring: exploring product and service organisations," In *Proceedings of the Scientific Workshop Proceedings of XP2016*, p. 8, ACM, May 2016.
- [41] G. K. Hanssen, D. Šmite, and N. B. Moe, "Signs of agile trends in global software engineering research: A tertiary study," In *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*, pp. 17-23, IEEE, August 2011.
- [42] S. Sharma, and N. Hasteer, "A comprehensive study on state of Scrum development," In *Computing, Communication and Automation (ICCCA), 2016*

- International Conference on*, pp. 867-872, IEEE, April 2016.
- [43] A. Brown and J. Sutherland, "Scrum at scale - go modular for greater success," <https://www.scruminc.com/wp-content/uploads/2014/07/Scrum-at-Scale-A-Modular-Approach.pdf>, 2014. [Accessed: 22 03 2017]
- [44] S. Mulder, "An action research study on the use of scrum to provide agility in data warehouse development," (Doctoral dissertation), 2011.
- [45] K. Schwaber, and M. Beedle, "Agile software development with Scrum," Vol. 1, Upper Saddle River: Prentice Hall, 2002.
- [46] T. Sharma, "Pragmatic Technical Debt Management," Techie, Researcher, Consultant, and Author at Siemens R&D, Slide Share, Apr 17, 2015.
- [47] F. Anwer, S. Aftab, U. Waheed, and S. S. Muhammad, "Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey," *International Journal of Multidisciplinary Sciences and Engineering*, vol. 8, no. 2, 2017.
- [48] S. Asiri, M. R. J. Qureshi, "The Proposed GSD Model to Solve Coordination and Communication Problems," *IJMECS*, vol.6, no.11, pp.25-30, 2014.
- [49] H. Iqbal, M. F. Khan, "Assimilation of Usability Engineering and User-Centered Design using Agile

Software Development Approach," *IJMECS*, vol.6, no.10, pp. 23-28, 2014.

Authors' Profiles

Sara Ashraf is a student of MS Computer Science with specialization in Software Engineering at Virtual University of Pakistan. Her areas of interest include Agile Development Models and Software Process Improvement.



Shabib Aftab received MS Degree in Computer Sciences from COMSATS Institute of Information Technology Lahore, Pakistan. He is serving as a Lecturer Computer Sciences at Virtual University of Pakistan. His research areas include Data Mining and Software Process Improvement.

How to cite this paper: Sara Ashraf, Shabib Aftab, "Latest Transformations in Scrum: A State of the Art Review", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.9, No.7, pp.12-22, 2017.DOI: 10.5815/ijmeecs.2017.07.02