

Pragmatic Evaluation of IScrum & Scrum

Sara Ashraf, Shabib Aftab

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

Received: 05 September 2017; Accepted: 13 October 2017; Published: 08 January 2018

Abstract—Scrum has emerged as a most adopted and most desired Agile approach that provides corporate strategic competency by laying a firm foundation for project management. Scrum, being more of a framework than a rigid methodology, offers maximum flexibility to its practitioners. However, there are several challenges confronted during its implementation for which certain researchers not only adapted, but also augmented Scrum with other Agile practices. One such effort is IScrum, an Improved Scrum process model. In this paper an empirical study has been conducted for analyzing the two models i.e. classical Agile Scrum model and IScrum process model. There are two goals of this study: first is to validate the IScrum and the second goal is to evaluate it in comparison with the traditional Scrum model. Subsequently, the study will describe and highlight which characteristics of Scrum are enhanced in IScrum. Furthermore, a survey is used to investigate the teams' experience with both models. The results of survey and case-study have been examined and compared to find out if IScrum performs well than Scrum in software development. The outcomes advocate that the improvements were quite effective in resolving most of the problem areas. The IScrum can thus be adopted by industry practitioners a best choice.

Index Terms—IScrum, Scrum, Agile process model, Empirical evaluation, Validation, Effort estimation, Story points, Scrum metrics, Backlog

I. INTRODUCTION

Agile Software Development (ASD) refers to a paradigm of processes that is in line with the concepts of Agile Manifesto [2]. A group of 17 leading experts shared their experience about software development approaches that what does and doesn't work, and agreed upon 4 values and 12 principles [3]. ASD offers mechanism to embrace constantly changing requirements, thus, allowing a defter way to deal with complex project undertakings [4]. Agile methods are aimed at minimizing the cost of inevitable change [5], providing business value rapidly, and delivering working software frequently [6].

Agile philosophies are simple, adaptable, and suitable for fulfilling the needs of contemporary software development. For projects with high degree of variation in tasks, in the technology being utilized, and in the

capacities of individuals, Agile methodologies are the ideal option [7]. Feature Driven Development (FDD) [8], eXtreme Programming (XP) [9], Adaptive Software Development (ASD) [10], Test Driven Development (TDD) [11], Dynamic System Development Method (DSDM) [12], Scrum [13], and Crystal methods [15], etc. are part of the Agile Software Development Ecosystem (ASDE) [14].

Within ASDE, Scrum is the most studied model among researchers, as well as it is the most widely adopted framework in software industry [18]. Scrum is a light-weight approach having collaborative, adaptive, and evolutionary attributes [15] [16]. According to the survey [17], more than half of the projects involving Scrum were delivered successfully. Owing to versatility, adaptability, and an extensive range of best practices that it offers, Scrum has been seeking the attention of researchers for last many years. Practitioners experimented with Scrum for improvement and better adoption. They made transformations in different extents of development process either by adopting or adapting various practices within Scrum [19]. Different variants of Scrum have been introduced to achieve product quality, and increased productivity. Ashraf and Aftab [20] made an attempt in this context and presented an improved version of Scrum named as IScrum. The paper in hand is intended to evaluate IScrum in comparison with the traditional Scrum through a case-study and a survey.

This paper is divided into following sections: Section II discusses the related work, and elaborates both the models Scrum and IScrum. Section III defines the research method that has been followed for evaluation of the models, then, Section IV presents the results, and lastly, Section V concludes the paper.

II. BACKGROUND & RELATED WORK

Scrum framework contains a variety of generic project management practices [21] due to which it is popular in other disciplines as well [17]. Despite all the benefits that Scrum provide, several areas are still challenging for practitioners. Scrum lacks explicit guide for product engineering while it offers a comprehensive framework for project management. Also, [22] and [23] reported various challenges and issues in adoption of Scrum. Researchers proposed numerous variants of Scrum in quest of improving the development process. Ali et al. [24] customized the Scrum by incorporating different risk

assessment, and mitigating practices. Rahayu et al. [25] introduced usability testing into traditional model of Scrum. Some researchers [26] brought game mechanics into Scrum framework and presented a new taxonomy for roles and practices in it. Others [27] tried to evolve the Scrum in a way that it can support the development of safety-critical products. Some [28] [29] experts made efforts to improve the collaboration with users by enriching the Scrum with User-Centered Systems Design (UCSD) tasks. Darwish and Megahed [30] emphasized on improving the phase of Requirements Engineering (RE) in Scrum framework by fusing various RE practices. Another stream of research is blending traditional models with Scrum. Singhto and Denwattana [31] and Jha et al. [32] made an attempt in this context, by combining waterfall with Scrum and achieved user satisfaction, and on-time delivery. Tirumala et al. [33] proposed a blend of Scrum and FDD, and incorporated the feature development practices into Scrum. They achieved improved quality and on-time delivery simultaneously through implementation of this hybrid. Similarly, other researchers also integrated Scrum with other Agile members [41].

For better understanding, both the models Scrum and IScrum are being elaborated below.

A. SCRUM Process Model

Being a member of Agile family, Scrum is an iterative and incremental framework, where solutions evolve through collaboration between self-organizing cross-functional teams. Through a disciplined project management process, Scrum promotes frequent inspection and adaptation. The philosophy of leadership in Scrum, encourages teamwork, accountability, and self-organization. A collection of practices are intended to deliver a high quality product. Likewise, Scrum's business approach aligns the development with the organization's goals and customer needs. According to [10], as a project management approach, it works well for small co-located development teams. Schwaber et al. [34] define the Scrum as:

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

Empiric process control through feedback-loops makes Scrum feasible for dynamic and complex projects. Three roles, three artifacts, and four ceremonies are contained within Scrum framework (see fig. 1).

Scrum Phases

According to Schwaber [1], there are 3 phases of Scrum activities: Pregame, game, and postgame.

- 1) *Pregame*: Initially, a vision is defined, and Product Backlog is organized by Product Owner (PO). Time and cost estimation plan is prepared. High-

level system architectural design is also created or modified. Moreover, funds approval, team formation, risk assessment, and validation of development tools are also accomplished in this phase.

- 2) *Game*: Also known as concurrent engineering. Actual development of product is done in iterative cycles. Develop (analysis, design, develop), wrap, review, and adjust are the activities that may be performed inside a cycle until a product is ready [35].
- 3) *Postgame*: Also known as closure phase. This phase involves preparing the final documentation i.e. user manuals, training materials etc., final integration testing, and then releasing the final product.

Scrum Roles

There are 3 formal roles in Scrum: Product Owner (PO), Scrum Master (SM), and Development Team.

- 1) *Product Owner*: PO represents the customers' view point and responsible for creating and prioritizing Product Backlog as well as for maximizing the ROI. Budget and schedules are planned by him.
- 2) *Scrum Master*: SM is a facilitator who helps the team in achieving cohesiveness, and self-organization, also, ensures that Scrum principles are followed [17].
- 3) *Development Team*: It comprises of individuals who are experts in their job. The team size should be between 3 and 9 [13].

Scrum Artifacts

There are following main artifacts in Scrum:

- 1) *Product Backlog*: A list of prioritized items/ user stories.
- 2) *Sprint Backlog*: A list of tasks to be done in the current Sprint.
- 3) *Product Increment*: A potentially shippable working software.

Scrum Ceremonies

In Scrum, there are 4 formal ceremonies:

- 1) *Sprint Planning Meeting*,
- 2) *Daily Scrum*,
- 3) *Sprint Review*, and
- 4) *Sprint Retrospective*.

Sprint is a small time-boxed iteration or cycle of development activities that may span over 1 to 4 weeks. Rest of the ceremonies are contained within it.

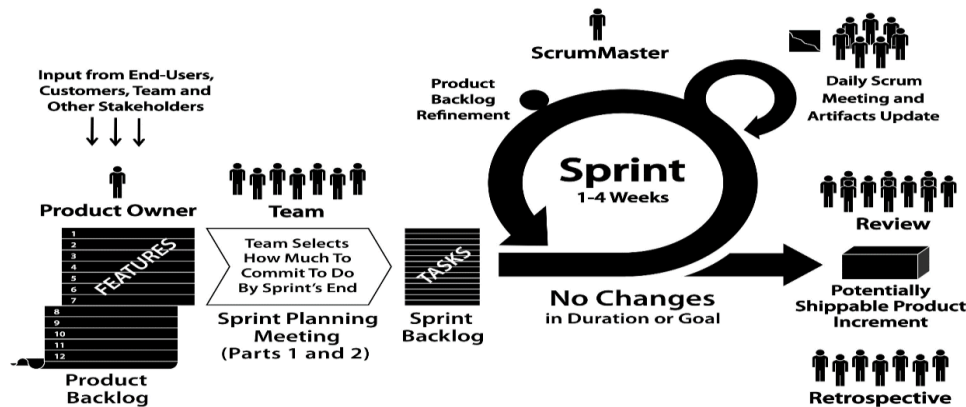


Fig.1. Overview of Scrum Process Model [36]

Scrum Process Overview

PO manages Product Backlog by deciding which features are to be implemented first. There is a Sprint Planning meeting at the start of each Sprint, where team decides on the tasks from Product Backlog and puts them into Sprint Backlog. In order to coordinate the team activities, a 15 minutes meeting is held daily at same time (mostly at the start of the day), same place under the supervision of Scrum Master. Team members discuss the work they have done, to be done, and problems if any. SM helps team in resolving those issues. After each Sprint, a release is presented before stakeholders, and management to capture their feedback, this is Sprint Review meeting. Subsequently, a Retrospective meeting is held to review the process. What did work and what didn't, suggestions are gathered from the team, reviewed, and improved the process accordingly [34] [36].

B. ISCRUM Process Model

IScrum model's main goal is to deliver in-time a high-quality product. It provides support for both small and medium scale projects. The IScrum framework [20] follows strictly all the major ceremonies, roles, artifacts, and practices of traditional Scrum. Along with these aforementioned core components, new roles are also introduced i.e. role of Technical Writer, and QA Engineer. Similarly, an event of training is added before initializing the development work. Rest of the ceremonies are performed with slight changes to their practices.

A preliminary stakeholder's analysis is conducted i) to evaluate the PO's prior relevant knowledge and experience, ii) to get familiar with the needs and scope of the project. After analyzing the project and its variables i.e. size, complexity, time, cost, and resources etc., a team is designed with the addition of new roles of Technical Writer and QA Engineer. After team formation, training sessions are conducted. There may be a Technical Skill training (if required) and Scrum training that is mandatory. In-house training sessions, each of 4 hours per day are conducted for 2-3 days.

IScrum Roles

Scrum Master will have same traditional set of responsibilities. Furthermore, he will analyze the Product Owner and Scrum team's need for training. He may conduct training sessions himself.

QA Engineer will play the same role with few variations. Rather than waiting for a Sprint release for testing, start collaborating with the development team from the beginning of Sprint. Mutually decide a half built internal release. QA will require less time for testing when external release will be ready at the end of a Sprint.

Technical Writer will participate in all the modules of Scrum from vision and product backlog grooming workshop to Sprint retrospective. He needs to be involved throughout the sprint to coordinate with developers and testers to make sure that the team is communicating and coordinating about documentation requirements and deliverables that must be baked into the iteration.

Criteria for documentation: Before documenting the process, technical writer and other team members need to use the following comprehensive criteria:

The documentation will be done only if:

- There is no or little chance of having discussion.
- It clearly imparts the immediate goal of the project.
- That document can be turned into executable specifications i.e. requirement, architecture, and design specifications in the form of tests.
- That item/ concept/ requirement is stable.
- That is required to the customer.
- It is industry regulation or contractual obligation

IScrum Ceremonies

Product Backlog grooming Workshop This meeting is held for developing Product vision, deciding high-level release timelines, identifying business needs. Also, Backlog grooming is underway in parallel. Participants of this workshop are PO, other stakeholders, SM, and Technical Writer.

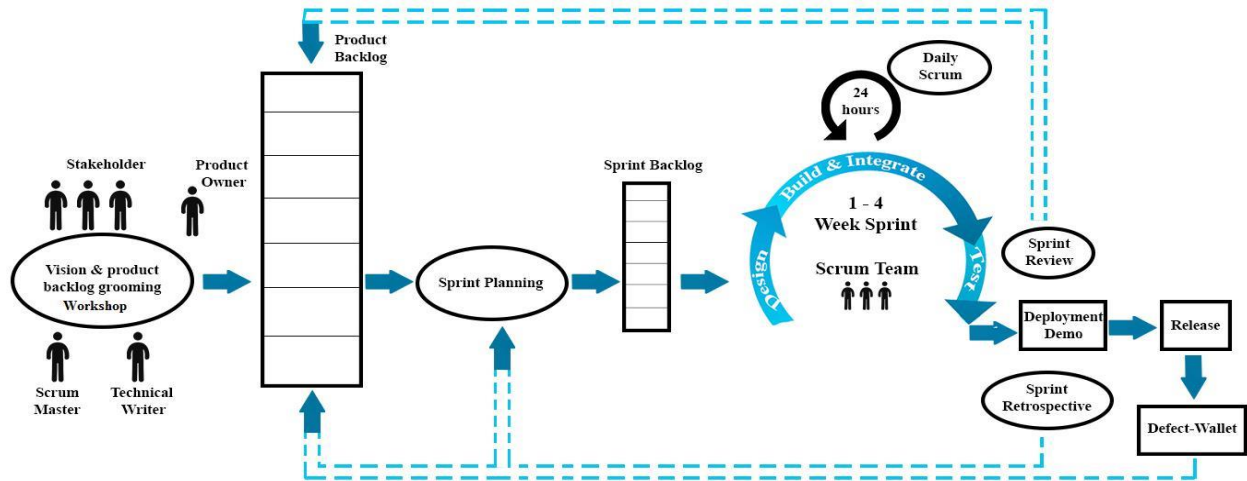


Fig.2. Overview of IScrum Process Model [20]

Daily Scrum/ Standup meeting: This meeting will be held at the end of working day, same time and same place daily for about 15 minutes. Each member will answer:

- a) What he has done since last meeting?
- b) Any obstacle or problem he is facing?

Then, next day tasks will be re-planned accordingly. Problems or bugs will be set as top priority tasks for next day.

IScrum Practices

Testing: Developers will perform unit testing. QA engineer will perform acceptance tests earlier in the Sprint. Regression testing will be performed for each release. White box testing will be performed by QA if needed. Use automated testing tools and test suits.

Work visibility: All the planned to-do work will be in Product Backlog. Changing requirements and unplanned items that appear during Sprint will be placed in Product Backlog, and plan next Sprint accordingly. Allow a new or change in requirement to a Sprint if:

- a) It is a Priority item/critical for stakeholders.
- b) All the Sprint tasks are done.

Maintenance/ Change management: Use a Defect-wallet on the task board. Log the defects/bugs that appear in the last release. These bugs will be sent to product backlog where they are analyzed for assigning points and priority. Then define sub-tasks for fixing these bugs and put them in the next Sprint.

III. RESEARCH DESIGN AND METHODS

Main concern of our research is to validate and evaluate the IScrum process model [20] in comparison with traditional Scrum. We employed a mixed research method that includes both the case study and the survey. We conducted a controlled case study for small to medium scale projects to prove the worth of IScrum.

A. Research Hypothesis

First of all, we need to define our research hypothesis that would steer the process of investigation through case study and survey. Here, we have following hypotheses regarding how IScrum can improve the overall software engineering process better than the traditional Scrum:

- H1: IScrum improves the team's productivity,
- H2: IScrum improves the quality of overall software product.
- H3: IScrum improves the visibility of work during Sprint.
- H4: IScrum improves the documentation.
- H5: IScrum improves the change management/maintenance.

B. Case Study

The objective of this empirical study is validation i.e. to find out whether the hypotheses H1-H5 are true. The National University of Computer and Emerging Sciences NUCES Islamabad, Faculty of Computer Science, volunteered for this research endeavor. This case study was carried out by 2 teams, each team comprising of 4 members. Being students of the final year undergraduate program, all team members have the same level of prior knowledge and experience. The project was assigned to both teams in such a way that Team A developed the software using traditional Scrum, while Team B employed IScrum to accomplish it. A brief introduction of this project is given below.

Description of Project

This project required the teams to develop a learning tool that helps in demonstrating the behavior of basic C++ code through animations. Teachers can use it to teach students. Table 1 briefly describes the project.

Students were familiar with the tools and technology and have ample knowledge of software development, however, they hadn't have any prior experience of deploying Agile methodologies. For that reason, an extensive training program was conducted to introduce

the team members with both the process models i.e. Scrum and IScrum. This case study was spanned over 2 months, and 4 Sprints. There were 4 team members in Team A such that 2 performing the role of Developers, 1 Technical Writer, and 1 QA Engineer. In Team B, there were 3 Developers and 1 QA Engineer. To balance the workload, the Technical Writer and QA Engineer performed 40% of their work as a Developer other than their dedicated roles. The supervisor served as a Product Owner for the project, while co-supervisor played the role of Scrum Master for both teams.

Table 1. Description of Project

Project Parameters	Description
Size	Small to medium
Type of Case Study	Controlled
Iterations	4
Programming Approach	Object Oriented
Language	Java
Documents	MS Office
Testing	J-Unit
Web Server	Apache Tomcat
Product Type	Learning tool
Project Type	Average
Project Duration	2 months
Team size	4 members
Feed back	Twice a week
Development Environment	Eclipse IDE

C. Surveys

Success of a project can also be evaluated through the surveys both quantitatively and qualitatively. A survey might help in collecting the opinions of individuals impacted through the process and product. In our case, two such surveys were conducted, one for process evaluation and the other for overall assessment of product and process.

The process evaluation survey was conducted at the completion of project through team members. They rated their overall experience with the pertinent methodology and its different areas, and attributes over a 5-point ordinal scale, where 1 representing 'poor' to 5 expressing 'excellent'.

The product evaluation was carried out by project supervisor, who also played the role of a customer, especially in our case the product was a learning tool, therefore, the supervisor can best assess it. At the end of each Sprint, supervisor assessed the release, performance of both teams, process parameters, and rated them over a 10-point scale.

D. Product & Process Metrics

While planning for a project, we need to plan for data collection as well, so that it can be decided what is to be measured, how to measure, and which metrics will be used, and when and what data will be gathered [37]. Software metrics are aimed at improving the software development as well as maintenance in software

engineering process [38]. As, they mostly deal with the benchmarking [39]. Software metrics measure the effectiveness of tools, methods, and technologies employed for software development [40]. Therefore, metrics help the teams to acquaint with the development progress, as well as to discover the unattended areas that need improvement. Moreover, metrics improve the understanding of stakeholders by increasing the transparency and visibility of end-to-end flow in a complex system [41]. Conventional metrics are not appropriate for ASD, so there are some specific metrics used to track the progress, team's performance, and product quality. Some of the metrics that have been used in this case-study are presented in Table 2.

Table 2. Metrics Adopted for Case-study

Sr. #	Metrics
1	Total Estimated work effort (hrs.) = [No. of members * No. of working hours per day * No. of days in a week * No. of weeks]
2	Total Actual work effort (hrs.) = [No. of members * No. of hours worked per day * No. of days in a week * No. of weeks]
3	Schedule Variance = (Actual effort - Estimated effort) / Estimated Effort * 100
4	User Stories Planned = [stories planned + carry forward from previous sprint]
5	Story Points Planned = [story points planned + carry forward from previous sprint]
6	Test Coverage (%) = Test LOC / Total LOC (OR) Automation Coverage = (Total No. of Test Case Automated / Total No. of Manual Test Cases) * 100 (OR) Test-case Coverage = Total No. of test-cases or scripts prepared / total No. of test requirements * Total no. of test scenarios
7	Defect Density = Post Release Defects per KLOC
8	Defect Removal Efficiency (DRE) = No. of Pre-release defects removed / (No. of Pre-release defects removed + No. of Post-Release Defects)
9	Team productivity = Total Lines Of Code / Actual Effort in hours
10	Team Productivity= KLOC/ Person-month
11	Rework Effort = No. of hours to fix Change Requests/ Actual Effort in hours * 100
12	Customer Participation= Time spent by customers for each release (hrs.) / Actual Effort in hours * 100

IV. EMPIRICAL ASSESSMENT OF RESULTS

During Sprint Planning meeting, the teams decided to use story pointing technique for planning and effort estimation. Both teams were given same time period for each Sprint. The performance of teams was keenly observed and tracked during meetings and appropriate guidance was provided timely. Team A's Technical Writer started collaborating with Developers to generate the documents, while in Team B it was a distributive effort. It was evident that Team A produced appropriate and standard documentation including flow diagram, requirement document, technical document, and Functional test document. On the other hand, Team B didn't generate sufficient documentation that ensued

decrease in productivity and over-commitment to maintenance. In order to assess the change management of both process models, Product Owner (supervisor) asked the teams for certain changes in the product during each Sprint. The detailed pragmatic results for IScrum and Scrum are presented in Table 3.

A. Comparative Analysis

We evaluated the outcomes by tracking the effort, velocity, productivity, quality, and rework effort estimations for both process models. A comparative analysis is being presented to show that if results of this case-study and survey support our hypotheses.

For successful and in-time delivery of a quality product, accurate estimation is required in the project planning phase. Story pointing technique has been applied for this purpose. This technique doesn't emphasize on man-hours, and/ or personnel's velocities but task complexity. For each Sprint, same number of user stories were assigned to both teams. Nonetheless, the story pointing was done by teams independently for each Sprint. After assigning story points, tasks were defined for each user story. In order to get better understanding of the project, same work effort (hrs.) was estimated for Team A & Team B.

Effort variance for IScrum is shown in Fig 3 i.e. difference between estimated and actual effort time. For first two Sprints, a positive effort variance is observed, it is due to lack of experience and understanding about the product. Afterwards, in Sprint 3 and 4, it gradually decreased for IScrum but increased for Scrum. Fig. 5 demonstrates the effort trend of IScrum, it is obvious that actual effort was high for first 2 Sprints, and low for next 2 Sprints. In fig. 6, effort estimates were less than the actual effort throughout the 4 Sprints.

We can see the team's velocity for IScrum and Scrum in fig. 7 and fig. 8 respectively i.e. the story points delivered in a Sprint. The velocity of Team A gradually increased, while, it decreased for Team B.

Scrum Backlog is very important for tracking the progress, fig. 9 and 10 present the comparison of IScrum and Scrum Backlog respectively, i.e. user stories planned vs. delivered per Sprint. User stories planned are the sum of stories planned for current Sprint and carry over from preceding Sprint. There was a Sprint carry over for IScrum, observed during first two sprints. This is due to documentation, and defect overhead. Team A proceeded smoothly afterwards. Sprint carry forward or moved is the number of user stories either partially completed or not even started in the sprint in which they were committed. It is also noteworthy, that story points for such stories were moved to the Sprints where they were 'Done'. There were more tasks defined for same number of stories with IScrum than that with Scrum (see item 7

Table 3 & 4). The reason is additional tasks of documentation and testing for each user story.

Productivity can be defined in terms of Lines Of Code (LOC) divided by actual effort in hours. Fig. 11 shows Productivity trend analysis. For IScrum, team's productivity was less for 1st small Sprint due to documentation and communication overhead. But, it increased gradually with each Sprint, hence, H1 is true. However, productivity remained steady and eventually waned for Scrum. There were multiple factors behind it such as lack of documentation, relatively less number of integrations, and more accumulation of defects.

Number of pre-release defects (i.e. item 18 in Table 3) depends on inspections and testing coverage, these defects are assumed to be removed as and when found prior to release. Number of post-release defects (i.e. item 19 in Table 3) depends on Change Requests (CRs) and/ or impacted functionality. Defect density, test coverage, and Defect Removal Efficiency (DRE) can effectively indicate the product quality. The quality achieved through IScrum and Scrum is presented in fig. 12 and fig. 13 (see item 16, 20, 21, 22, 23 of Table 3). It is obvious that for IScrum defect density decreased significantly while test coverage and DRE increased that implies that IScrum has delivered improved quality than Scrum. This is mainly because of parallel testing and development, and use of automated testing tool. Thus, it is proved that H2 is true.

Rework effort is an indicator of effective change management for a process model. Fig. 14 reveals that IScrum offers better change management and less rework effort than Scrum for which it increased considerably. The reason could be lack of quality implementation, lack of documentation, some CRs caused major changes in already done work, and/or misperception of tasks by Developer. So, H4 and H5 are true, as they support each other.

The teams' survey responses were collected and analyzed. Surprisingly, IScrum performed better in all the items, and excelled in 4th, 7th, 8th, 9th, and 10th items, as shown in Table 4 and fig. 15. The findings of survey affirm the quantitative results of case-study i.e. outcomes for: item 3 verifies H1, item 4 and 8 confirms H2, & item 7 and 9 prove H4 and H5 true respectively. Ratings for item 5 indicate the significance of training. Team A experienced better communication and collaboration as compared to Team B as per scores for item 6. IScrum achieved 89.6% customer satisfaction higher than that of Scrum i.e. 73.06% as shown in fig. 16. Visibility of work was also better achieved with IScrum. These two parameters were assessed through customer survey conducted through supervisor. Thus, H3 is also true.

Table 3. Empirical Data from both Teams for IScrum & Scrum

ID	Item	Team A					Team B				
		Sprints				Total	Sprints				Total
		1	2	3	4		1	2	3	4	
1	Completion time duration (weeks)	1	2	3	2	8	1	2	3	2	8
2	*Total Estimated work effort (hrs.)	40	80	120	80	320	40	80	120	80	320
3	**Total Actual work effort (hrs.)	50	90	115	70	325	48	98	125	100	371
4	Schedule variance	25	12.5	-4.167	-12.5	1.562	20	22.5	4.167	25	15.94
5	User Stories Planned = [stories planned + carry forward from previous sprint]	10	16 = 15+1	25 = 24+1	9	60	10	15	27 = 24+3	13 = 9+4	65
6	Story Points Planned = [story points planned + carry forward from previous sprint]	16	30 = 28+2	44 = 42+2	32	122	18	30	51 = 44+7	38 = 32+6	137
7	Tasks defined for User Stories	80	90	110	40	320	64	77	96	40	277
8	User Stories Delivered	9	15	25	9	58	10	12	23	13	58
9	Story Points Delivered	14	28	44	32	118	18	23	45	38	124
10	User Stories Moved	1	1	0	0	2	0	3	4	0	7
11	Story Points Moved	2	2	0	0	4	0	7	6	0	13
12	Total Lines of Code	1050	2330	3044	1980	8404	1102	2265	2970	2090	8427
13	Total KLOC	1.05	2.33	3.044	1.98	8.404	1.102	2.265	2.97	2.09	8.427
14	Unit test scripts/ test cases	27	35	48	20	130	17	27	36	14	94
15	Functional test scripts / test cases	58	64	78	76	276	49	51	69	28	197
16	***Test Coverage (%)	58.51	63.09	58.29	71.61	62.89	57.6	42.19	53.2	30.2	45.79
17	Code Integrations	17	18	27	24	86	13	19	23	20	75
18	Pre-Release Defects	7	8	6	5	26	5	7	5	4	21
19	Post Release Defects	2	3	2	1	8	2	6	7	3	18
20	Pre-Release Defects per User Story delivered	0.778	0.533	0.24	0.556	0.448	0.278	0.304	0.111	0.105	0.169
21	Post-release defects per User story delivered	0.22	0.2	0.08	0.111	0.138	0.2	0.5	0.304	0.231	0.310
22	Post Release defects / KLOC	1.905	1.288	0.657	0.505	0.952	1.815	2.649	2.357	1.435	2.136
23	Defect Removal Efficiency	77.78	72.73	75.00	83.33	76.47	71.43	53.85	41.67	57.14	53.85
24	Post Release Proposals (Sprint Retrospective)	4	2	1	0	7	3	2	3	2	10
25	Pre-release Change Requests due to change in business, user, functional and non-functional requests	2	2	3	1	8	1	2	3	2	8
26	Total Change Requests per KLOC	1.905	0.858	0.986	0.505	1.063	0.907	0.883	1.010	0.957	0.939
27	Time to manage total Change Requests (hrs.)	3.5	4.5	5	2	15	3.5	7	10	8.5	29
28	Rework Effort	7	5	4.348	2.857	19.21	7.29	7.14	8	8.5	30.93
29	Productivity= LOC / Actual effort hrs.	21	25.89	26.47	28.29	25.86	22.96	23.11	23.76	20.9	22.71
30	Productivity= KLOC/ Person-month	1.050	1.294	1.323	1.414	1.293	1.148	1.156	1.188	1.045	0.148
31	Customer participation (sprint hrs. per week)	35%	30%	25%	20%	28%	30%	30%	20%	20%	25%
32	Customer Satisfaction	86.5 %	88.9%	90.5%	92.6%	89.63%	86.12 %	85.63 %	62.75 %	57.75%	73.06 %

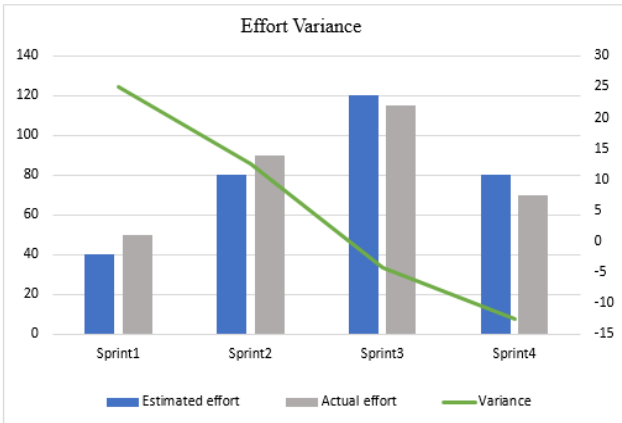


Fig.3. Effort Variance for IScrum

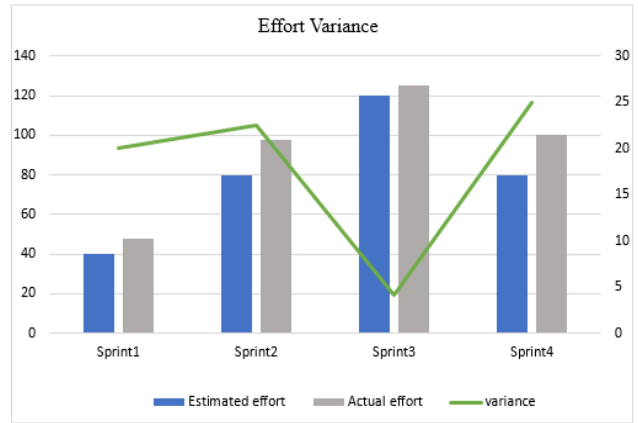


Fig.4. Effort Variance for Scrum

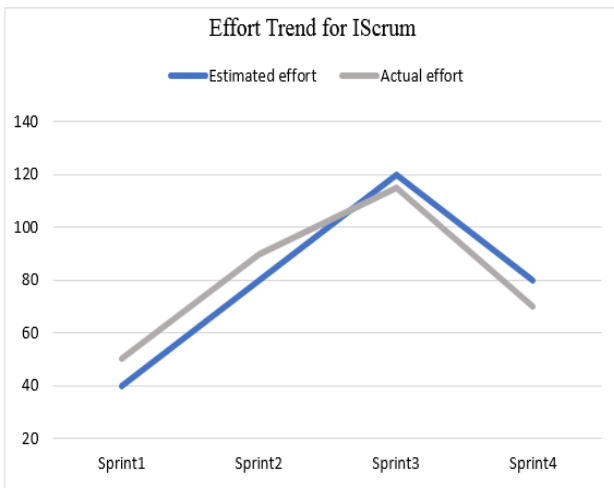


Fig.5. Effort Trend for IScrum

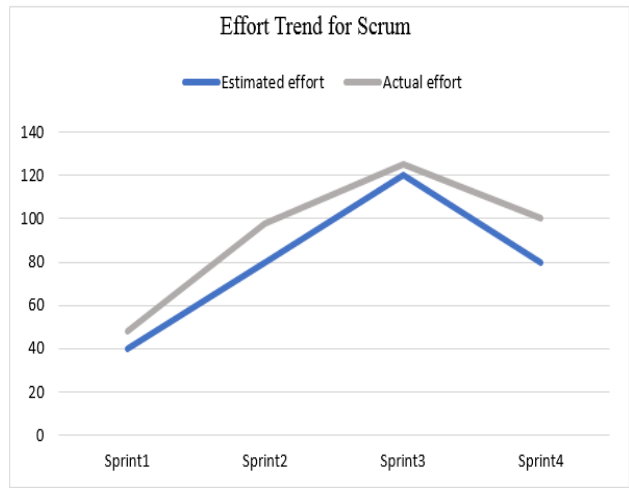


Fig.6. Effort Trend for Scrum

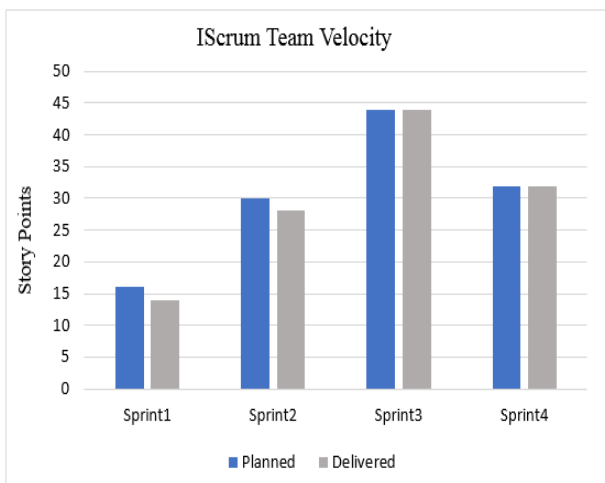


Fig.7. Velocity for IScrum Team

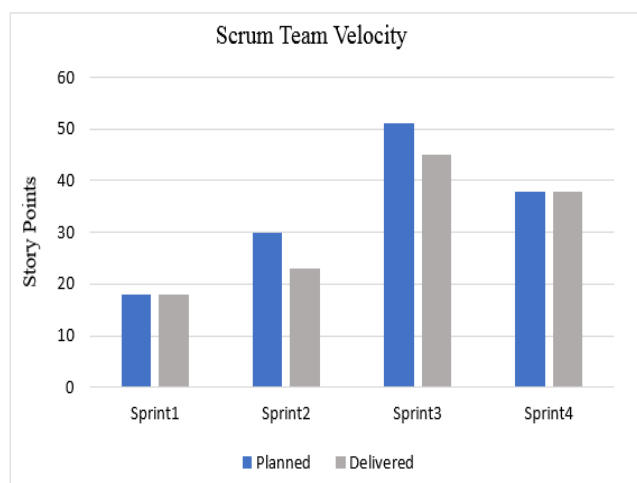


Fig.8. Velocity for Scrum Team

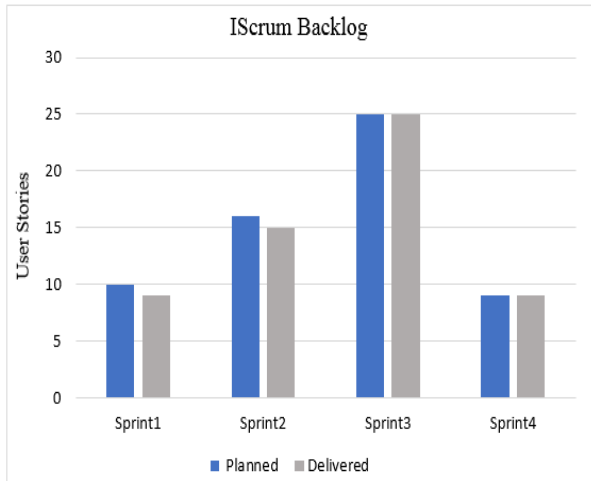


Fig.9. Backlog of IScrum

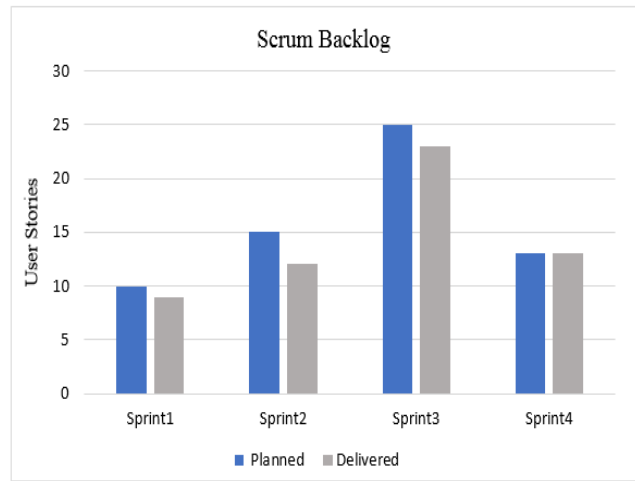


Fig.10. Backlog of Scrum

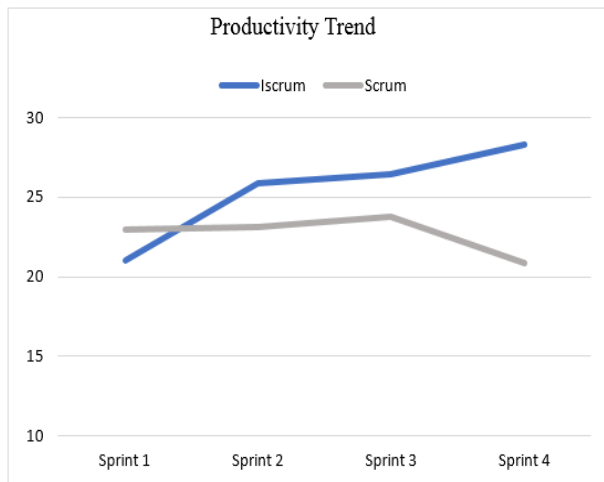


Fig.11. Productivity Trend of IScrum & Scrum

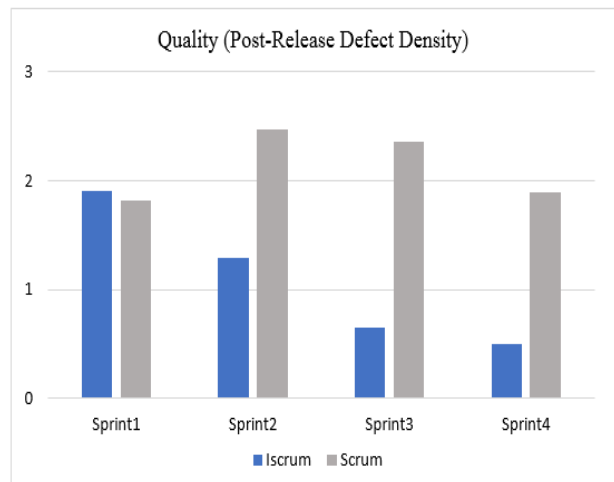


Fig.12. Quality of IScrum & Scrum

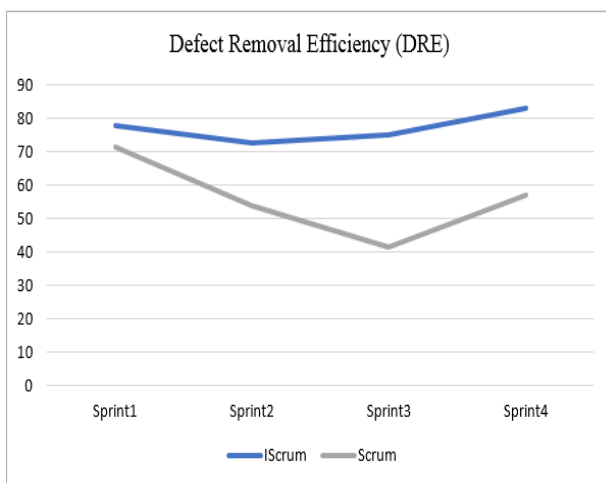


Fig. 13. DRE of IScrum & Scrum

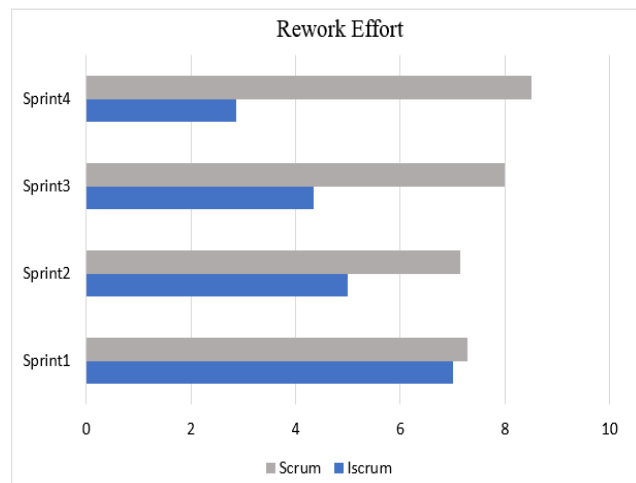


Fig.14. Rework Effort of IScrum & Scrum

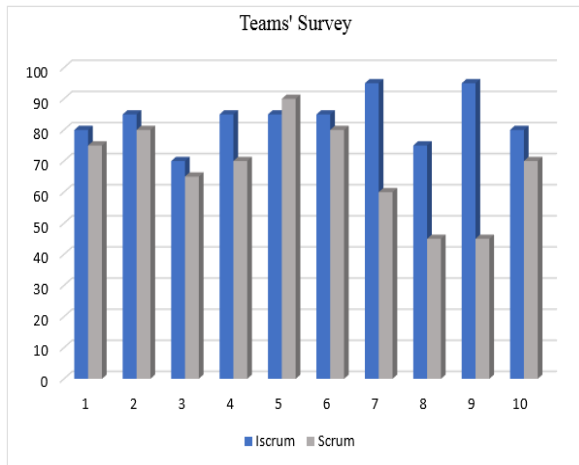


Fig.15. Teams' Survey for IScrum & Scrum

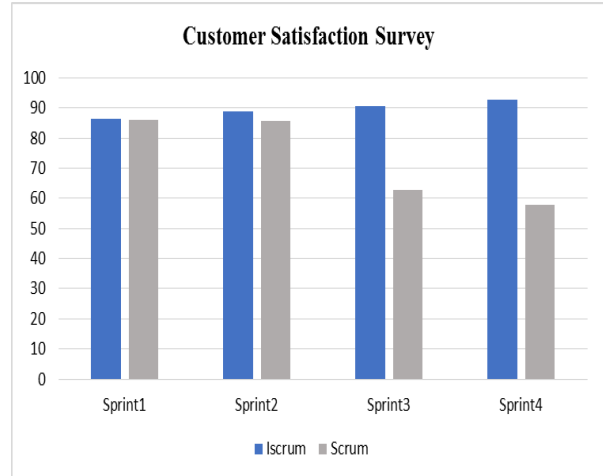


Fig.16. Customer Satisfaction

Table 4. Survey findings from Teams

Item#	Description	Response of Members of	
		Team A	Team B
1	You had the required understanding of Sprint Scope and Goal	80%	75%
2	Team has implemented the actions suggested in retrospectives	85%	80%
3	You were more productive	70%	65%
4	You delivered a quality product	85%	70%
5	Prior training has improved your knowledge about process model and your performance	85%	90%
6	Team collaboration and communication was frequent	85%	80%
7	Documentation criteria helped in managing documents/ you achieved required documentation	95%	60%
8	Automated Testing improved test coverage/ testing was done effectively	75%	45%
9	Defect-Wallet helped managing changes/ change was managed effectively	95%	45%
10	How would you rate your overall experience with IScrum/ Scrum model? (Bold the one you used)	80%	70%

Table 5. Customer Survey Items

Item #	Statements	Item #	Statements
1	Completion rate of Sprint Items	5	Team productivity
2	Delivered desired functionality	6	Change Request managed and delivered
3	Easy to use and interactive	7	Defect rate and Severity
4	Testing coverage offered in current Sprint	8	Visibility of work

V. CONCLUSION

We successfully tested our research hypotheses. Moreover, from observations and exploratory data gathered through case-study and survey, following improvements have been experienced and limitations are identified.

A. Findings

- Prior training helped in synchronizing the team members and making them aware that how the process goes. It helped team members to gain a better understanding of the process. Moreover, it enhanced the team's process abilities.

- We had a better quality control with IScrum. Moreover, with well-maintained documentation, we improved process with less dependence on resource/personnel replacement during the project life cycle.
- With improved and concrete documentation and communication, IScrum provides accurate information to complete tasks professionally and with improved eminence.
- Defects management also improved the maintenance and user experience, in a nutshell, these all factors contribute in improving team's productivity. Better quality gives the better customer experience and satisfaction since IScrum delivers more business value.

- Some collaboration time gets consumed between Technical Writer and developers to synchronize documentation that can slightly reduce the productivity.
- Standup meetings at the end of working day didn't prove to be a good idea, in the start of day seems more suitable as this refreshes the information for the day and any impediments can be addressed during day. Energy levels for team members are high at the start as compared to end of day.
- Timely and precise communication between Developers and Technical Writer is important to avoid time lags and team members to be on the same page and remain on progress track. It is believed that it will improve with more usage and experience on IScrum methodology. Overall IScrum has shown better results and this suggests valuable improvements in the traditional methodology.

B. Limitations of the study

This study was conducted under controlled settings of an academic environment, therefore, its scope is limited and so is the data set. Similarly, survey responses were confined to 8 team members and one supervisor only. Students have no prior experience of Agile software development. Also, testing automation couldn't be achieved upto the mark due to lack of tools and skill required to use them. Moreover, students were over-committed within limited calendar time of 2 months as other academic activities were also there. Though, the supervisor enacted as a customer but there was no typical customer involved.

C. Future Work

For further improvement, it is intended to work on the above-mentioned limitations. Also, the IScrum model should be analyzed on industrial level for multiple projects to generalize the results. More Agile practices can be incorporated to improve other problem areas of Scrum. The scope of the proposed model can further be extended to scale it for large-scale distributed software development.

ACKNOWLEDGMENT

The authors would like to thank Saba Rasheed Assistant Professor, Faculty of Computer Sciences, Hina Ashraf Lecturer, Faculty of Electrical Engineering NUCES, for their support and effort to accomplish this research endeavor, and all the team members for their cooperation in collection of exploratory data.

REFERENCES

- [1] K. Schwaber, "Scrum development process," In *Business Object Design and Implementation* Springer London, pp. 117-134, 1997.
- [2] A. Alliance. 2001 "Agile manifesto," [Online]. Available: <http://agilemanifesto.org/> [Accessed 25 05 2017]
- [3] 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.
- [4] D. E. Strode, S. L. Huff, B. Hope, & S. Link, "Coordination in co-located agile software development projects," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1222–1238, 2012.
- [5] J. Highsmith, and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp.120-127, 2001.
- [6] O. P. Timperi, "An overview of quality assurance practices in agile methodologies," In *T-76.650 Seminar in Software Engineering*, 2004.
- [7] J. Highsmith, "Cutter Consortium Reports: Agile Project Management: Principles and Tools 4", Cutter Consortium, Arlington, MA. Feb. 2003
- [8] S. R. Palmer, and M. Felsing," A practical guide to feature-driven development," Pearson Education, 2001
- [9] K. Beck, "Extreme Programming Explained," Addison Wesley, Reading, MA, 1999
- [10] J. Highsmith, "Adaptive software development." Dorset House, 2000.
- [11] K. Beck, "Test-driven development: by example," Addison-Wesley Professional, 2003
- [12] J. Stapleton, "DSDM, dynamic systems development method: the method in practice," Cambridge University Press, 1999
- [13] K. Schwaber, A. Beedle, "Agile Software Development with SCRUM," Prentice-Hall, Upper Saddle River. NJ, 2002.
- [14] J. A. Highsmith," Agile software development ecosystems," Addison-Wesley Professional, vol. 13. 2002.
- [15] A. Cockburn, "Agile Software Development," Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [16] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New directions on agile methods: A comparative analysis," In *Proceedings of the 25th International Conference on Software Engineering, ser. ICSE '03. Washington, DC, USA: IEEE Computer Society*, pp. 244–254, 2003.
- [17] S. Alliance, "The 2015 State of Scrum Report," Download unter www.Scrumalliance.org/why-Scrum/state-of-Scrumreport/2015-state-of-Scrum 2015. [Accessed: 07 05 2017]
- [18] S. Sharma, and N. Hasteer, "A comprehensive study on state of Scrum development," In *Computing, Communication and Automation (ICCA), 2016 International Conference on*, pp. 867-872, IEEE, April 2016.
- [19] S. Ashraf, S. 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.
- [20] S. Ashraf, S. Aftab, "IScrum: An Improved Scrum Process Model," *International Journal of Modern Education and Computer Science (IJMECS)*, vol.9, no.8, pp.16-24, 2017.
- [21] F. Anwer, S. Aftab, S. S. M. Shah, and U. Waheed, "Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum," *International Journal of Computer Science and Telecommunications*, vol. 8, no. 2, March 2017.

- [22] R. Akif, and H. Majeed, "Issues and challenges in Scrum implementation," *International Journal of Scientific & Engineering Research*, vol. 3, no. 8, pp. 1-4, 2012.
- [23] H. Hajjdiab, and A.S. Taleb, "Adopting agile software development: issues and challenges," *International Journal of Managing Value and Supply Chains (IJMVSC)*, vol. 2 no. 3, pp. 1-10, 2011.
- [24] M. Ali, Y. Hafeez, and B. Hamid, "An Empirical Study and a Framework for Effective Risk Management in Scrum," In *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, vol. 53 no. 4, pp. 417-429, 2016.
- [25] 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 *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2016, pp. 1-6, IEEE, October 2016.
- [26] 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.
- [27] 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.
- [28] M. Larusdottir, J. Gulliksen, & A. Cajander, "A license to kill-Improving UCSD in Agile development," *Journal of Systems and Software*, vol. 123, pp. 214-222, 2017.
- [29] H. Iqbal, M. F. Khan, "Assimilation of Usability Engineering and User-Centered Design using Agile Software Development Approach," *International Journal of Modern Education and Computer Science (IJMECS)*, vol.6, no. 10, pp. 23-28, 2014.
- [30] N. R. Darwish, & S. Megahed, "Requirements Engineering in Scrum Framework," *Requirements Engineering*, vol. 149, 2016.
- [31] W. Singhto, & N. Denwattana, "An experience in blending the traditional and Agile methodologies to assist in a small software development project," In *13th International Joint Conference on Computer Science and Software Engineering (JCSSE) IEEE*, pp. 1-5, July 2016.
- [32] M. M. Jha, R. M. F. Vilardell, & J. Narayan, "Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market," In *IEEE 11th International Conference on Global Software Engineering (ICGSE)*, 2016, pp. 84-88, IEEE, August 2016.
- [33] S. Tirumala, S. Ali, B. G. Anjan, "A Hybrid Agile model using SCRUM and Feature Driven Development," *International Journal of Computer Applications*, vol. 156 pp. 1-5, 2016.
- [34] 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: 07 03 2017]
- [35] D. Cohen, M. Lindvall and P. Costa, "An introduction to agile methods," In *Advances in Computers*, vol. 62, 2004, pp.1-66.
- [36] P. Deemer, G. Benefield, C. Larman and B. Vodde, "The Scrum primer V 1.2," *Scrum Alliance*, <http://www.brianidavidson.com/agile/docs/Scrumprimer121.pdf>. [Accessed: 05 01 2017]
- [37] N. E. Fenton, and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach: Brooks," 1998.
- [38] S. H. Kan, *Metrics and models in software quality engineering*. Addison-Wesley Longman Publishing Co., Inc. 2002.
- [39] C. Jones, "Applied Software Measurement", McGraw Hill, 1991.
- [40] N.E. Fenton, and M., Neil, *Software metrics: roadmap*. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 357-370). ACM, May 2000.
- [41] S. Ashraf, S. Aftab, "Scrum with the Spices of Agile Family: A Systematic Mapping," *International Journal of Modern Education and Computer Science (IJMECS)*, vol.9, no.11, pp. 58-72, 2017.

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, "Pragmatic Evaluation of IScrum & Scrum", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.10, No.1, pp. 24-35, 2018.DOI: 10.5815/ijmeecs.2018.01.03