

I-Scrum: An Improved Scrum Process Model

Sara Ashraf, Shabib Aftab

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

Received: 27 May 2017; Accepted: 18 June 2017; Published: 08 August 2017

Abstract—Resolving a wide domain of issues and offering a variety of benefits to software engineering, makes the Agile process models attractive for researchers. Scrum has been recognized as one of the most promising and successfully adopted agile process models at software industry. The reason behind vast recognition is its contribution towards increased productivity, improved collaboration, quick response to fluctuating market needs and faster delivery of quality product. Though Scrum performs better for small projects but there are certain challenges that practitioners encounter while implementing it. Experts have made some efforts to adapt the Scrum in a way that could remove those drawbacks and limitations, however, no single effort addresses all the issues. This paper is intended to present a tailored version of Scrum aimed at improving documentation, team's performance, and visibility of work, testing, and maintenance. The proposed model involves adapting and innovating the traditional Scrum practices and roles to overcome the problems while preserving the integrity and simplicity of the model.

Index Terms—Scrum, Improved Scrum, Tailored Scrum, Customized Scrum, Agile Model, Software Process Improvement

I. INTRODUCTION

Scrum is a light-weight software development framework embodies adaptive, evolutionary, and cooperative attributes [23] [24]. Abrahamsson [24] defines Scrum as a project management approach proven effective for co-located and small development teams. For managing complex and dynamic software projects, Scrum acquainted with the concept of empirical process control. Keeping in view the empirical reality of the project, the plans are consistently inspected and adapted [25]. According to Alliance [27], 62% of the projects employing Scrum have been delivered successfully. Agile Scrum has minimized the overall software development time [26]. It is a customer-centered and value-driven approach, therefore, it embraces changing demands of customers to achieve their satisfaction and involvement.

Scrum focuses on project management, for circumstances where planning is troublesome initially,

the principle component, the feedback-loops are used as tool for empiric process control. Scrum is employed through three key roles: product owner, scrum master and development team. The product is created by a self-organizing and cross-functional team in short-term stages called sprints (usually of 2 to 4 weeks). Each sprint begins with a sprint planning meeting and closes with retrospective evaluation. Product owner maintains a product backlog i.e. a list of prioritized features that must be implemented. The development team chooses which tasks are to be executed in the following sprint and develops a sprint backlog. Daily Scrum is a stand-up meeting of about 15 minutes, conducted at the start of the day to coordinate the team activities. While Scrum master tackles issues that might cause hindrances for the Scrum team. To acquire feedback, potentially shippable product is presented in sprint review meeting. A retrospective meeting is held at the end of sprint, for evaluation of process and its improvement [29] [30] [41]. The Scrum process model overview can be seen in Fig. 1.

Scrum works well for project management but has certain limitations in technical engineering aspect. There are a number of such weak areas in Scrum, a few of them are being discussed here in the following.

We have to learn principles and values of Scrum, not only the practices [34]. Many problems arise due to the reason that Scrum team is not Scrum trained [31]. Scrum certification and training is recommended by 32%, while 11% of the members of the organizations need them for effective implementation of Scrum, since it has significantly improved the process and practices of Scrum [35] [36]. If product owner and customer are not scrum trained that may cause project failure. The development is driven mainly by product owner so if he is not well trained, it might lead the development team to a wrong track [32].

There is a lack of documentation in Scrum. Sprints are too short to manage time for a strict formal documentation along with accomplishment of other sprint tasks. It has been found that Product backlog is not updated properly as most of the requests for change in requirements are received directly through emails and phone calls; that may lead to traceability concerns later. Different team members are documenting that may result in lack of standardization and increase in bug rate [37] [38].

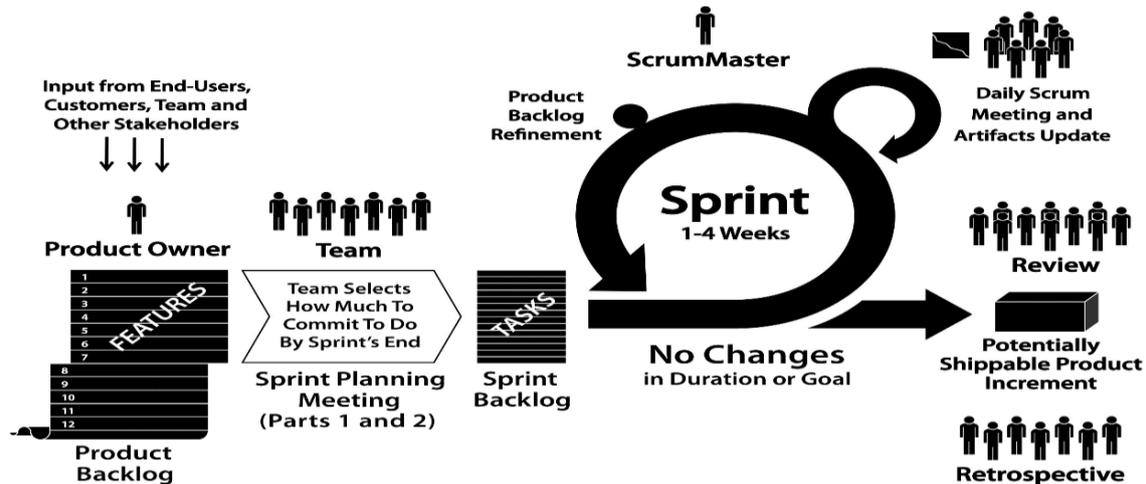


Fig.1. Scrum process overview [30]

During sprint, changes cannot be made. No customer or product owner intervention is allowed during sprint. Lack of visibility of work is also encountered in Scrum. (1) It is hard to perceive what alternate groups are doing at any given time, multiple tasks are running in parallel, which makes it hard to know who is doing what undertaking; (2) a considerable measure of undertaking functionalities are not in a state of coherence since conditions from different groups are not unique; and (3) there is not much time for everyone in the team to experience all tasks and request data about their assigned task inside and outside the group [33].

The quality of software might suffer due to inadequate communication between development team and QA engineers. Moreover, there is lack of explicit Regression testing in Scrum, reason is again lean practices. Also during a sprint, the code written should be debugged and tested.

It is difficult to implement Scrum model in maintenance area. Since, maintenance can't be sorted out into sprints; and involves separation of tasks rather than their interaction and a common goal like in development phase. Maintenance groups regularly work with various clients who are far from site, and rarely communicate face-to-face [28].

Rest of the paper is structured as: Section II covers contemporary researcher's work to evolve the Scrum or to integrate it with other software models. Section III defines the problem statement. Section IV includes the proposed model. And, Section V concludes the paper.

II. RELATED WORK

Keeping in view the strengths and weaknesses, the researchers keep on experimenting with the Scrum. Some introduced new practices and new roles in it, some have evolved the existing one while others have integrated other software process models with it.

Sharma and Hasteer [1] reviewed the literature and analyzed the current state of Scrum during past five years (i.e. 2010-2015) in terms of popularity, adoption, and evolution. They referred to survey reports and reviewed

thirty papers published during this period. They found that the researchers and software industry are more inclined towards integration and adoption of Scrum respectively as compared to any other software process models.

Larusdottir *et al.* [2] and [39] proposed to incorporate the user-centered systems design (UCSD) activities and practices with Scrum to improve collaboration with users, users experience and usability throughout the software development. Gupta *et al.* [3] have transformed the Scrum by integrating some of the innovative practices into it. They studied the globally distributed product and analyzed the impact of evolved practices on code quality, cost, product early reach to market, and scaling up of new users.

Wangenheim *et al.* [4] explored the benefits of teaching Scrum in academic institution for undergraduate program. They developed a simulated learning game intended to develop the skill and experience among the participating students. The authors established that education of Scrum is vital for improving collaboration among teams, cohesiveness, and their productivity.

Jha *et al.* [5] implemented a hybrid model in which they integrated waterfall model with traditional Scrum as well as proposed some practices to beat the challenges confronted during implementation of hybrid model.

Park *et al.* [6] proposed a mechanism to structure the practices of Scrum model. They discussed how Scrum practices can be expressed using Essence kernel and language that is accepted as a standard for making and endorsing software engineering methods by Object Management Group.

Darwish and Megahed [7] proposed how Requirements Engineering (RE) practices and techniques can be fused into Scrum, and also discussed the challenges encountered in this regard. There is no mechanism or tool in Scrum that can assure process conformance through concrete feedback, to deal with this challenge

Matthies *et al.* [8] presented a tool ScrumLint that analyses the development artifacts. According to the authors the feedback collected through such tool can improve the workflows and overall process measurement.

Table 1. Summary of the related work

| Title | Limitations |
|---|---|
| A Comprehensive Study on State of Scrum Development [1]. | Reviewed the research of evolving Scrum but not analyzed the work based on integrating Scrum with other software process models. |
| A license to kill—Improving UCSD in Agile development [2]. | The proposed guidelines are too generic and don't specify how to employ them. Providing no practical implications. |
| Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey A case study in a multi-location legacy software product development team [3]. | Proposed model lacks the practices for improving the visibility of tasks to the management. No way to deal with issues arising during sprint. Furthermore, the practices for improving team performance are also neglected. |
| SCRUMIA—An educational game for teaching SCRUM in computing courses [4]. | Results of the assessments lack generality, hence are applied to a restricted domain (courses by a single instructor within same university). |
| Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market [5]. | Though have better results for large and distributed projects but not optimal for small projects. |
| Scrum powered by Essence [6]. | No practical evidence is provided to analyze the results, the graphical representation of work at each stage of process may increase documentation overhead. |
| Requirements Engineering in Scrum Framework [7]. | The paper doesn't address how to resolve the issues that arise during RE phases when applied in Scrum. |
| ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts [8]. | The tool is implemented in controlled environment with limited performance measuring parameters and metrics. |
| Enhancements in scum framework using extreme Programming practices [9]. | The proposed model is not tested and validated in real practical settings. |
| An experience in blending the Traditional and Agile methodologies to assist in a small software development project [10]. | The blended approach increased the time required for planning and testing. Also, do not provide the evidence for applying with diversity of teams. |
| Towards an Agile Requirements Engineering Process Combining HERMES 5 and SCRUM [11]. | The combined approach solves RE problems but drops the agility in this phase while dealing with complex projects. |
| An integrated document management system for managing self programme accreditation using Scrum approach [12]. | For academic quality assurance system, support for document management is quite limited. |
| A Synchronous Agile Framework Proposal Combining Scrum and TDD [13]. | The proposed model needs to be validated in industry for confirming the claimed benefits. It does not address the team composition and structure. Tools and techniques for testing are also not specified. |
| A Hybrid Agile model using SCRUM and Feature Driven Development [14]. | The model SCR-FDD is validated only with a single controlled setting, large-scale implementation is required to justify the potential. |
| Software Quality Assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations [15]. | Recommended techniques and training may increase overhead and complexity for organization. |
| Hybrid fuzzy-ontological project framework of a team work simulation system [16]. | The applicability of the given model can be assessed accurately only if the whole SPSM design process is first closed. Data for team members' selection was not provided. |
| Agile for large scale projects—A hybrid approach [17]. | For deployment phases, better approaches are desirable. Range of implementation is limited, validated in a single organization. |
| Scrum and Embedded Software Development for the Automotive Industry [18]. | No experimental evidence that could support the said approach in real product development project. |
| ScrumS: a model for safe agile development [19]. | Involves documentation overhead and intensive testing to test security. Scrum team training with modified infra-structure is pre-requisite. |
| Review-Scrum (R-Scrum) Introduction of Model Driven Architecture (MDA) in Agile methodology [20]. | Risk factor is not mitigated, testing process requires further refinement. |

Scrum lacks engineering practices that is why researchers are motivated to incorporate such practices from other models. Ramadan [9] presented an Enhanced Scrum framework by combining the best practices of XP into Scrum. The author elaborated the framework with a comprehensive set of guidelines for accomplishing each activity of traditional Scrum starting from preparing product backlog to delivering an increment and sprint retrospective. Also, elaborated how to incorporate the selected XP practices and validated through a survey.

Singhto and Denwattana [10] implemented a hybrid model, a blend of traditional waterfall model and agile Scrum for a small software development project. They achieved success in terms of user satisfaction, meeting deadlines and timely delivery. It also solved poor design problems. Schar et al. [11] implemented a combination of HERMES 5 a sequential process model and the Agile Scrum model with the principle focus on requirements engineering. The Scrum process model lacks explicit security practices.

Mokhtar *et al.* [12] evolved the Scrum with the emphasis on reducing duplication and redundancy of documents so that during the audit, desired document can be searched in minimum time. They implemented this model for developing an online integrated document management system.

Maria *et al.* [13] proposed a synchronous framework integrating the Scrum and TDD by merging a series of good practices of Scrum for team management and some features of TDD regarding development and testing. This model involves incorporating management team of Scrum throughout the software process and the TDD team that develops the lines of code and tests them while ensuring feedback all the way thereby, increasing quality of both process and product.

Tirumala *et al.* [14] presented a hybrid model SCR-FDD, a combination of Scrum and FDD aimed at improving quality and time for development. Since the FDD model is more inclined towards achieving quality but not meeting deadlines in most of the cases. On the other hand, Scrum is more focus on strict time line than the quality. Keeping in view these facts, the SCR-FDD starts with identification of features and then modules comprising of multiple features are developed iteratively. The authors validated the model by implementing FDD, Scrum, and SCR-FDD in a real-time project with three small teams.

Khalane and Tanner [15] revealed SQA aspects in Scrum by presenting theory building case study. They found that it is the organization that decides how to customize development processes.

Orlowski *et al.* [16] proposed a hybrid fuzzy-ontological system for developing software process simulation-modeling SPSM system. Scrum model including project roles and management processes are all simulated by SPSM.

Tanveer [17] proposed a hybrid model combining the best practices of RUP with Scrum to overcome the weaknesses of both models and to improve the predictability, communication, and management. Takahira *et al.* [18] integrated V methodologies into Scrum framework for rapid development of embedded software in automotive industry. The Scrum process model lacks explicit security practices.

The Scrum process model lacks explicit security practices. Maria *et al.* [19] presented a model for safe agile development by adding specific security techniques to the conventional Scrum lifecycle without affecting its originality and mapped the Risk Analysis method.

Iqbal and Javed [20] proposed a Model Driven Architecture MDA in Scrum and named it as Review-Scrum. This model helps resolving issue like lack of pictorial representation of work and risk handling. It is evident that no single work embraces all the major problem areas of maintenance, team performance, documentation, testing and work visualization during

Sprint. Table 1 shows summary of related work highlighting the limitations.

III. PROBLEM DEFINITION

Scrum's project management practices strongly advocate its application in projects with increasing complexity, but as far as its engineering areas are concerned, practices are not explicitly stated in the process model.

As discussed in Section II, researchers have been evolving the Scrum in different ways to cope up the challenges confronted while applying it for the projects varying in size and complexity. It is observed that all such efforts that have been made revolve around one or two problem areas at a time. However, there is no single solution that can cover most of the key problem areas simultaneously. Also, customizing Scrum may involve the risk of no longer adhering to the values, practices, and principles of the methodology ensued losing the integrity. Hence, the research question is:

How to tailor the Scrum framework to improve documentation, team's performance, visibility of work during a sprint, testing, and maintenance while preserving the integrity and simplicity of Scrum?

IV. THE PROPOSED ISCRUM PROCESS MODEL

The proposed IScrum model is intended to deliver a high-quality software in minimum time for small to medium sized projects with co-located teams.

The proposed model adapts the traditional Scrum by tailoring the practices involved in requirements engineering. Also introducing a role of Technical Writer for appropriate documentation. Moreover, Scrum Master's role is customized to analyze the Stakeholder and team, also to conduct the training.

Firstly, we need to establish an infrastructure for Scrum to execute the process in full spirit. It encompasses identification of phases, events, roles and also the practices involved to undergo them. The improved process model includes the core components of traditional Scrum i.e. Product Backlog development, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective except Scrum Training. Also, the major roles remain same including Scrum Master, Product Owner, and the Team except for Technical Writer a new addition in Scrum Team. The ceremonies of traditional Scrum are performed in a way not violating its ambiance but with a little tweak to the practices to overcome the impediments.

There are several activities and events in the presented model, their workflow is shown in Fig. 2.

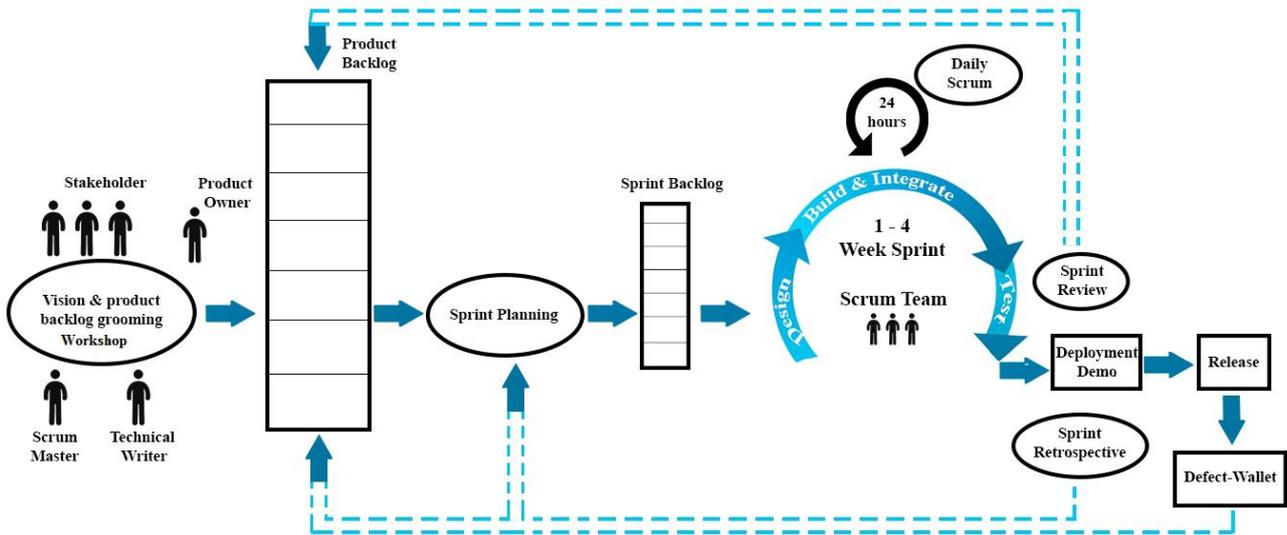


Fig. 2. IScrum process model

A. Preliminary stakeholders' Analysis

Product Owner is the representative of customers and users of the product. According to Scrum, Product Owner is solely responsible for requirements management. Project requirements usually cover a number of aspects e.g. functional, technical, and business. There are no such criteria defined for requirements content in Scrum. Therefore, it is a challenging task for the Product Owner to cover all dimensions of requirements efficiently. Furthermore, the Product Owner has to play two important roles of Business Analyst and Functional Analyst. Business Analyst is the one who imparts requirements i.e. what the business organization wants to achieve from the users of this product. The Functional Analyst divulges the requirements pertaining to what functionality customer desires from the product.

For that Product Owner should have a knowhow of such delicacies. Here, a preliminary analysis of Product Owner is necessary for two reasons:

Firstly, to evaluate that what level of training is required to the Product Owner for advancing through the process efficiently.

Secondly, to understand the nature of the upcoming project for planning and designing the task force and other resources.

B. Team Formation

Taken into account the nature of the project (its size and complexity), constraints (time and cost), available tools and technology and also the human resources a team will be designed having all the needed skill set. Here, a team member will play a new role of Technical Writer. The team size will be chosen according to the Scrum rules.

C. Scrum Training

Next step, after organizing team and analysis of Product Owner, is conducting training sessions. This training can be of two types: the Scrum Training and the

Technical Skill training. Former is mandatory while latter can be conducted if required. An external resource person can be hired for Scrum training if feasible. Otherwise, only Train the Scrum Master first and then he can provide the Scrum training to the rest of the team by conducting in-house training sessions.

As all the events in Scrum are strictly time-boxed, so is the training practice. Training sessions can be of 04 hours per day for 2-3 days.

D. Role of Technical Writer

Technical Writer can be presented as an integral player in the Scrum team to meet the challenge of insufficient documentation. He goes along with the process starting from the project planning all the way to Sprint retrospective and even after shipping the product to the maintenance.

For example, Technical Writer will take part in Sprint Planning for identifying stories and tasks that need further documentation. He may write test suites that will serve as technical documentation. Technical Writer will stay in touch with the developers and testers throughout the Sprint to achieve the team's collaboration over those deliverables that they are going to develop during the iteration.

E. Role of Quality Assurance QA Engineer

It has been found that QA engineer lags one Sprint behind because he has to wait for the release for testing. Role of QA engineer in Scrum team remains same but the practice is adapted in such a way that QA personnel remain engaged with the developers' team throughout the Sprint and they mutually decide the internal release. This release may comprise of half the features built and developers hand over the QA for testing. So that when the external release is ready the QA would need less time for testing. QA Engineer will also review the test results and test coverage for evaluating their adequacy. Also he would monitor the progress of testing with continuous feedback daily. During Sprint QA member would be

involved throughout the development, thereby reducing the time for testing and ensuring the quality as he will be more informed about product.

F. Role of Scrum Master

The Scrum Master will play the same role as is in traditional Scrum except with the added responsibility of Scrum training to the team. He will hold in-house training sessions for Scrum Team.

G. Workshop for Product Backlog grooming

Arrange a workshop where the Product Owner, customers and other stakeholders together with Scrum Master and Technical Writer generate the Product vision, identify business needs, and decide high-level release timelines while grooming the product backlog in parallel. Both the customers and the team have a close collaboration that may lead to building mutual trust.

H. Documentation

Scrum doesn't ask for either more comprehensive or no documentation. According to Cockburn & Highsmith [22], they are the people who drive agile projects, if they are co-located then they do their best, moreover, their collaboration and face-to-face communication largely replace document preparation and dissemination. Keeping in view this fact, the researchers typically go for intensive communication to fill the gap caused due to reduced documentation. But here we need to keep the balance between communication and documentation.

Documentation practices of traditional Scrum will not be completely rejected rather improved. In order to have effective and adequate documentation, we need to decide first that what and when to document, for that following criteria are being proposed here.

The documentation will be done only if:

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

During the Product Backlog grooming workshop, requirements analysis, their specification, and other technical documentation will be done by Technical Writer following the above-mentioned criteria. In the subsequent stages of Sprint planning and executing the Sprint, Technical Writer will document whatever values for the team and customers. Code will be as much self-documenting as possible.

I. Standup meeting at the end of the day

Unlike traditional Daily Scrum that is held at the start of the day, it is suggested to hold this meeting at the end of the working day. All the formalities for the meeting like time duration and place will remain same as are in

traditional Scrum. Each individual in the team will answer the two questions:

- a) What he/she has accomplished today?
- b) Any impediment?

After this, the tasks till next standup meeting will be re-planned such that in case of any impediment or bug that task is set to top priority. All the work in progress and not done yet will be made visible on a task board.

This way each member of the team will be accountable for his /her task accomplishment. Also, it keeps everyone on task and collaborating.

J. Testing

By using tests, 1) requirements, architecture, and design can be specified, and 2) our work can be validated. QA personnel will involve throughout the Sprint will perform early product acceptance tests.

Regression testing will be performed at the end of each Sprint by testers to ensure that new features developed in the current Sprint have not generated any unwanted effects or changed the previously developed functionality a.k.a. ripple effects in the entire product developed so far. White box testing will be performed by QA engineers if required.

It is suggested to use test suits that may or may not be platform dependent and should be completely automated with minimal human intervention needed. It will reduce the cost both in terms of human effort and time. Hence, improve the quality of the product.

K. Visibility of work during a sprint

Active participation of customers in projects is desirable hence it lets them control the project. They acknowledge if they are updated and development work is visible to them [21]. The work is not visible during sprint to the stakeholders. For this, it is recommended to put all work that the team has to do in the Product backlog. This way the team's work will be made visible. Though, all the planned work is available in the product backlog still there are various unplanned items crawling into the Sprint and originating from different directions; it may be a manager, in some cases a Product owner, and sometimes from corporate-side. Place all the work in Product Backlog to make it visible and transparent, however, try not to interrupt the teams so that they can better achieve their Sprint goal without losing their focus.

The most upsetting thing for a team is new/changing requirements appearing amid a Sprint. In order to deal with them, some criteria can be set such that a new or change in requirement can only be allowed to a Sprint if:

- a) All requirements/tasks from the Sprint are done.
- b) It is a Priority item/critical for our users, or a kind of blocker.

Teams do not need any other Key Performance Indicator (KPI) rather they can better determine their

performance patterns through visualizing their progress using some Time-in-Process Chart.

L. Maintenance

In order to improve the product quality, configuration management and defect management are crucial [40]. For keeping track of defects that may appear in last Sprint Release, use a defect-wallet on the task board. It will serve the purpose of defects logging. Initially, the defect-wallet will contain no subtasks, but only some points assigned. In this way, it is ensured that even the task of fixing a defect is also included in the Sprint backlog with values assigned to it. In some cases, the product backlog can also be updated, if required. The scope of the Sprint can easily be updated to the Product Owner, in case it is being affected, the team can protect the Sprint goal. It would not only help keeping track of time but also ensure that team is conforming the burndown. This is how maintenance tasks and the development both run in parallel.

V. CONCLUSION AND FUTURE WORK

The proposed model adheres to the principles and values of original Scrum while retaining its simplicity. The presented model is expected to improve the team performance and produce the quality product. Scrum team's training will help improving the team performance. By introducing the role of Technical Writer, unified and quality documentation can be accomplished. It would help reducing traceability concerns. Also, the issue of inadequate documentation will be resolved. Visibility of work during Sprint will help improving customer's trust. The QA's interaction throughout the Sprint may lead to enhanced product quality. Automated testing will reduce time and effort, and maintenance will be a parallel part of the product development Sprint.

For future, it is intended to refine the model by validating it for different practical settings. Furthermore, the domain of research will be expanded by adding other problem areas of Scrum.

REFERENCES

- [1] 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.
- [2] 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.
- [3] R. K. Gupta, P. Manikreddy, & 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, Feb 2017.
- [4] C. G. von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA—An educational game for teaching SCRUM in computing courses," *Journal of Systems and Software*, vol. 86, pp. 2675-2687, 2013
- [5] 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 Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*, pp. 84-88, IEEE, August 2016.
- [6] J. S. Park, P. E. McMahon, & B. Myburgh, "Scrum powered by essence," *ACM SIGSOFT Software Engineering Notes*, vol. 41, pp. 1-8, 2016.
- [7] N. R. Darwish, & S. Megahed, "Requirements Engineering in Scrum Framework," *Requirements Engineering*, vol. 149, 2016.
- [8] C. Matthies, T. Kowark, K. Richly, M. Uflacker, and H. Plattner, "ScrumLint: identifying violations of agile practices using development artifacts," *In Cooperative and Human Aspects of Software Engineering (CHASE), 2016 IEEE/ACM*, pp. 40-43, May 2016.
- [9] N. R. Darwish, "Enhancements In Scrum Framework Using Extreme Programming Practices," *International Journal of Intelligent Computing and Information Sciences (IJICIS)*, Ain Shams University, vol. 14, pp.53-67, 2014
- [10] W. Singhto, & N. Denwattana, "An experience in blending the traditional and Agile methodologies to assist in a small software development project," *In Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on IEEE*, pp. 1-5, July 2016.
- [11] B. Schär, S. Jüngling, & B. Thönsen, "Towards an Agile Requirements Engineering Process Combining HERMES 5 and SCRUM," *In Enterprise Systems (ES), 2015 International Conference on IEEE*, pp. 98-109, October 2015.
- [12] R. Mokhtar, N. H. Jaafar, N. F. Tahar, S. A. Sukiman, A. Aris, and N. F. Abu Bakar, "An integrated document management system for managing self programme accreditation using Scrum approach," *in Technology Management and Emerging Technologies (ISTMET), 2014 International Symposium on*, pp. 102-106, 2014
- [13] Savoine, M. Maria, V. F. Rocha, C. A. C. Bezerra, A. M. C. de Araújo, and J. K. M. Matias, "A Synchronous Agile Framework Proposal Combining Scrum and TDD," *ICSEA*, pp.350, 2016
- [14] 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.
- [15] T. Khalane and M. Tanner, "Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations," *in Adaptive Science and Technology (ICAST), 2013 International Conference on*, pp. 1-6, 2013.
- [16] C. Orłowski, I. Bach-Dabrowska, P. Kaplanski, and W. Wysocki, "Hybrid Fuzzy-ontological Project Framework of a Team Work Simulation System Embedded System," *in International Conference on Information Technology, Procedia Computer Science*, vol. 35, pp. 1175-1184, 2014.
- [17] M. Tanveer, "Agile for large scale projects—A hybrid approach," *In Software Engineering Conference (NSEC), 2015 National IEEE*, pp. 14-18, December 2015,
- [18] R. Y. Takahira, L. R. Laraia, F. A. Dias, A. S. Yu, P. T. Nascimento, and A. S. Camargo, "Scrum and Embedded Software development for the automotive industry," *in Management of Engineering & Technology (PICMET), 2014 Portland International Conference on*, pp. 2664-2672, 2014.

- [19] R. E. Maria, Jr, L.A. Rodrigues, and N. A. Pinto, "ScrumS: a model for safe agile development," *In Proceedings of the 7th International Conference on Management of computational and collective intelligence in Digital EcoSystems ACM*, pp. 43-47, October 2015.
- [20] U. Iqbal, and A. Javed, "Review-Scrum (R-Scrum) Introduction of Model Driven Architecture (MDA) In Agile Methodology," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 3, pp. 296-302, 2014.
- [21] K. Petersen, & C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," *Journal of Systems and Software*, vol. 82 no. 9, pp. 1479-1490, 2009.
- [22] A. Cockburn, & J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34 no. 11, pp. 131-133, 2001.
- [23] A. Cockburn, "Agile Software Development," *Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.*, 2002.
- [24] 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.
- [25] V. Szalvay, "An introduction to agile software development," *Danube technologies*, pp. 1-9, 2004.
- [26] D. Mougouei, N. F. M. Sani, & M. M. Almasi, "S-Scrum: a secure methodology for agile development of web services," *World of Computer Science and Information Technology Journal (WCSIT)*, vol. 3, no. 1, pp. 15-19, 2013.
- [27] S. Alliance, "The 2015 State of Scrum Report," *Download unter www.Scrumalliance.org/why-Scrum/state-of-Scrumreport/2015-state-of-Scrum* 2015. [Accessed: 05 01 2017]
- [28] I. Ghani, Z. Azham, and S. R. Jeong, "Integrating Software Security into Agile-Scrum Method," *TIIS*, vol. 8, pp. 646-663, 2014.
- [29] K. Schwaber, "Agile project management with Scrum," *Microsoft press*, 2004
- [30] 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]
- [31] P. Bootla, O. Rojanapornpun, and Mongkolnam, P., "Necessary skills and attitudes for development team members in Scrum: Thai experts' and practitioners' perspectives," *In Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on* pp. 184-189, IEEE. July 2015.
- [32] C. D. O. Melo, V. Santos, E. Katayama, H. Corbucci, R. Prikladnicki, A. Goldman, and F. Kon, "The evolution of agile software development in Brazil," *Journal of the Brazilian Computer Society*, vol. 19, no. 4, pp. 523-552, 2013.
- [33] M. O. Ahmad, P. Kuvaja, M. Oivo, and J. Markkula, "Transition of software maintenance teams from Scrum to Kanban," *In System Sciences (HICSS), 2016 49th Hawaii International Conference on* pp. 5427-5436, IEEE. 2016, January.
- [34] J. López-Martínez, R. Juárez-Ramírez, C. Huertas, S. Jiménez, and C. Guerra-García, "Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review," *In Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in*, pp. 141-148, IEEE, April, 2016.
- [35] S. Alliance, "The state of Scrum: benchmarks and guidelines," F. L. Orlando, D. Kim. 2013.
- [36] Mann, C. and Maurer, F., "A case study on the impact of Scrum on overtime and customer satisfaction," *In Agile Conference, 2005, Proceedings IEEE*, pp. 70-79, July, 2005.
- [37] F. Ghafoor, I. A. Shah, & N. Rashid, "Issues in Adopting Agile Methodologies in Global and Local Software Development: A Systematic Literature Review Protocol with Preliminary Results." *International Journal of Computer Applications*, vol. 160, no. 7, 2017.
- [38] Kapitsaki, M. Georgia and M. Christou, "Where is Scrum in the current Agile world?," *In Evaluation of Novel Approaches to Software Engineering (ENASE), IEEE, International Conference on*, pp. 1-8, 2014.
- [39] 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.
- [40] R. Noor, M. F. Khan, "Defect Management in Agile Software Development," *IJMECS*, vol.6, no.3, pp.55-60, 2014.
- [41] 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.

Authors' Profiles

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



Shabib Aftab is working as a Lecturer in Computer Science Department, Virtual University of Pakistan. He got his MS degree in Computer Science from 'COMSATS Institute of Information Technology, Lahore. Previously he has completed M.Sc Information Technology from 'Punjab University College of Information Technology (PUCIT), Lahore. His areas of research are Data Mining and Software Process Improvement.

How to cite this paper: Sara Ashraf, Shabib Aftab, "IScrum: An Improved Scrum Process Model", International Journal of Modern Education and Computer Science(IJMECS), Vol.9, No.8, pp.16-24, 2017.DOI: 10.5815/ijmecs.2017.08.03