

Evaluating the Quality of Proposed Agile XScrum Model

M. Rizwan Jameel Qureshi

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: rmuhammd@kau.edu.sa

Received: 28 September 2017; Accepted: 16 October 2017; Published: 08 November 2017

Abstract—The software companies are practicing XP and Scrum models from last several years. XP lacks in management practices whereas Scrum is weak in engineering practices. Due to continually promising need of agile development, this research tackles the problems of XP and Scrum by integrating them to enrich the strengths of XP and Scrum and suppress their limitations. The previous attempts provide little effective empirical evidence regarding the integration of XP and Scrum. Therefore, there is a pressing need to provide empirical evidence for the integration of XP and Scrum to show its usefulness developing the software projects. The same is accomplished by proposing XScrum model. Another goal of this paper is to analyze the quality of proposed XScrum with existing XP and Scrum. The proposed XScrum is validated by performing three case studies for three industrial projects and the results are described in the paper. The results are presented using quantitative and qualitative data. The results provide empirical evidence that there is a significant improvement in quality of proposed XScrum as compared to the existing XP and Scrum.

Index Terms—Backlog, Engineering Practices, Management Practices, Quality, Scrum, XP.

I. INTRODUCTION

The traditional software (SW) development methodologies are one of the topics of discussion from decades by several researchers [1-4]. The major problems, to apply the traditional methodologies, are: misunderstanding the requirements, fail to meet the expectations of end users, changing requirements, conflicts among categories of users and difficult to involve users [5-6]. Therefore, agile development methodologies are introduced to tackle the problems of traditional methodologies. Agile manifesto is emphasized on: 1) individuals and interactions over processes and tools; 2) working software over comprehensive documentation; 3) strong customer collaboration over contract negotiation and 4) respond to changing requirements over plan driven development. Agile development methodologies include practices that focus on code centric approach in contrast to comprehensive documentation, high interaction among team and user

involvement and quick response to changing customer requirements without sacrificing quality [6].

The SW is developed in iterations following agile models [7]. The completion time of iteration is not more than two weeks. Agile methodologies invite the developers to get involved in testing, rather than a separate quality assurance team. Agile methodologies are popular because of their ability to work effectively and efficiently in changing environments. This is due to the modern practices and principles enabling development teams to complete software as per the schedule. XP methodology is widely practiced in software industry. Many case studies are available comprising successful stories of XP model for small projects. Although XP has enormous strengths but still significant number of software companies are hesitant to transfer from plan driven methodologies to XP [8].

Scrum is influencing the development community from the last few years due to its strong management practices. Scrum methodology clearly describes human roles. The success of Scrum methodology mainly relies on effective and efficient contribution of Scrum teams. Strong communication between team members will increase coordination, knowledge sharing and satisfaction of team members. The effective participation of a member effects to entire Scrum team especially in distributed software development (DSD) environment.

The main objective of Scrum is to deliver a product in increments within 2 to 4 weeks. The Scrum development believes in five core values.

- Deliver every sprint or demo.
- Team must be cross functional (self organized) i.e., team is responsible to take every action including the establishment of sprint backlog.
- Daily meetings need to organize i.e., inspects daily.
- There must be a chief impediment with the team i.e., Scrum Master.
- Product owner is responsible to set the priorities to establish product backlog.

The core Scrum activities are product backlog, sprint planning, sprint backlog, sprint, shippable product, sprint review and retrospective [5]. Product backlog is a set of prioritized user stories by product owner. Sprint planning meeting is conduct among team members to decide the sprint backlog to tackle the number of user stories in a

sprint. Sprint is workable software that is completed within 2 to 4 weeks. It is up to the team that how many sprints are completed to deliver a shippable product to customer. Sprint is reviewed by team and retrospective is used to judge that Scrum is successfully implanted by the team or it needs further improvement. Scrum lacks in engineering practices to develop software like pair programming, continuous integration and automated builds practices of XP. This paper is a proposal to integrate XP with Scrum to enrich the benefits of both models by removing their shortcomings to improve quality of the resultant software.

Further paper is arranged as: Section 2 focuses on the related work and main differences between Scrum and XP. Section 3 defines the problem in hand. Section 4 illustrates the motivation for the proposed XScrum model. Section 5 presents integration of Scrum into XP model. Section 6 describes the validation, analysis and findings of this research.

II. RELATED WORK

Noor and Kahn [9] present a study to manage defects in agile development projects. It is mentioned that software quality assurance methods must be incorporated in agile software development to ensure high quality such as inspections, testing, product metrics and refactoring. Defect prevention, removal, tolerance and forecasting are the main benefits achieved due to including software quality assurance methods. It is recommended that defect management should be applied throughout the system development life cycle of agile projects. Defect management process will also help the agile team to cater the major risks of software projects [9]. Ali et al. [10] report a study to apply defect management with respect to the software industry of Pakistan. A questionnaire is used as a research tool to conclude the results. It is inferred that defect management enables the development team to ensure high quality and customer satisfaction.

A comparison of existing attempts to improve classical eXtreme Programming (XP) is provided to show their limitations [12]. Simplified Extreme Programming (SXP) process model is proposed to overcome the limitations of classical XP model without sacrificing the agile spirit. The classical XP is proposed to deploy it for small size projects. The core objective of SXP is to adapt it for small, medium and large size projects. The main phases of SXP are Initialization, Analysis, Design, Development and Testing and Release. The proposed SXP model is not validated to check its effectiveness [12].

Moe et al. [13] present a 9 months case study to introduce Scrum in a software development company using the teamwork model. Moe et al. [13] integrate the pair programming practice of XP with Scrum to improve monitoring, feedback and backup.

Mahnic [14] describes a case study on agile estimating and planning using Scrum. The results of the study show that beginners can understand and implement Scrum practices after couple of sprints [14]. The empirical study is providing an evidence for the software industry that

Scrum brings into many benefits such as better management and improvement in productivity [14]. There is a pressing need for conducting a large industrial project to find out the impact of introducing Scrum into XP in a software development company.

XP software development process works in six phases namely exploration, planning, iterations to first release, productionalization, maintenance and death [5]. On the other hand, the study reveals that Scrum model has three phases namely pre-game, development and post-game [15]. In XP model, features (to be developed) are prioritized by the customer [16]. On the other hand, the Product Owner prioritizes the product backlog which contains everything that is needed in the final product [17]. In XP model, a Coach guides to other team members. In Scrum, Scrum Master is responsible to ensure that the project is carried through according to the practices, values and rules of Scrum [16]. Manager makes decisions in XP project and he distinguishes difficulties and deficiencies in the process. On contrary in Scrum project, management makes final decisions and sets the goals and requirements [18-20]. Scrum lacks in engineering practices and it can be adopted to manage whatever engineering practices are being used in an organization [17]. It is identified that XP focuses more on software development whereas main concentration of Scrum is on project management. Abrahamsson et al. [16] deduce that Scrum model produces more efficient software than XP model.

Caballero et al. [21] provide a case study to introduce Scrum in a very small enterprise. The objective is to analyze productivity and quality of Scrum model with a similar experience of Team Software Process (TSP). Four sprints of Scrum are compared with two release cycles of TSP. It is recommended that Scrum can be integrated with XP to improve quality. Malhotra and Chug [22] provide a case study to develop small size project by integrating XP and Scrum i.e., IXSCRUM. The results show the validation of the IXSCRUM by completing the three iterations of a case study. The results are inferred based on the defects per kilo line of code (KLOC) to show the improvement in quality without providing a comparison of the IXSCRUM case study with XP and Scrum case studies.

Sutherland et al. [23] describe that it is easy to integrate XP and Scrum practices even on large projects with distributed teams to improve productivity, reduce project risk and increase software quality. A case study, of integrating XP and Scrum, is provided by Kniberg and Farhang [24] to show its successful implementation. A comparative analysis of XP and Scrum is provided to find the subtle differences and advantages of both models [25]. There are limited empirical evidences provided in the literature to integrate XP and Scrum [23-25].

III. PROBLEM STATEMENT

There are several attempts to analyze the integration of XP and Scrum from last few years [21-25] but still further work is required [26]. According to Marchenko

and Abrahamsson [27], the Scrum process is not assessed fully in the previous literature though there is an enormous acceptance of Scrum in the software industry. It is argued that less than 5% of the previous studies provide effective evidence that agile software development addresses Scrum. There is very little attention paid about the integration of XP and Scrum and limited empirical evidences are provided in the literature [21-27]. Therefore, it is highly required to integrate XP and Scrum in industrial settings to address this issue.

IV. MOTIVATION OF THE PROPOSED SOLUTION

Table 1 shows the twelve core practices of classical XP [5]. Planning game of XP is replaced by Sprint and Sprint planning meeting of Scrum due to change the scope of team from user stories to sprint goals. Metaphor practice is not included to increase the productivity. 40 hours per week and onsite customer practices are excluded to solve the scalability problem of XP. The motivation to exclude 40 hours per week practice is to propose a model that is equally beneficial for distributed teams (for global software development) and co-located teams. Therefore, the proposed model will be equally adaptable for co-located and distributed teams. There are twelve practices of Scrum available in the existing literature [27] as shown in Table 2. Burndown chart and freeze sprint backlog practices are eliminated to increase scalability, productivity and customer satisfaction. Sprint review meeting is replaced by the post mortem analysis as it is proved to establish procedures to improve software development and research processes during a project [28-29].

Table 1. A Comparison of XP and proposed XScrum Practices

XP [5]	Proposed XScrum
Planning game	No
Small releases	Yes
Metaphor	No
Simple Design	Yes
Test first coding	Yes
Continuous integration	Yes
Pair programming	Yes
Code Ownership	Yes
Refactoring	Yes
40 hours per week	No
Onsite Customer	No
Coding standards	Yes

Table 2. A Comparison of Scrum and proposed XScrum Practices

Scrum [27]	Proposed XScrum
Sprints	Yes
Product owner	Yes
Sprint Planning meeting	Yes
Daily Scrum Meeting	Yes
Scrum of Scrum Meeting	Yes
Scrum master	Yes
Sprint Review Meeting	No
Sprint backlog	Yes
Product backlog	Yes
Self-organized team	Yes
Burn down Chart	No
Freeze Sprint backlog	No (in case of small projects)

The comparisons of classical XP and Scrum practices with the proposed XScrum are shown in Tables 1 and 2.

V. THE PROPOSED XSCRUM MODEL

The proposed phases of XScrum model are 'Plot', 'Pattern', 'Instigate' and 'Filter' as shown in fig. 1. Table 3 shows the main activities of the proposed XScrum model.

A. 'Plot' Phase.

The system development life cycle (SDLC) of XP model starts with 'Planning' phase. The main activities of this phase are collection of user requirements (stories), maintain a response mechanism, story estimations, setting criterion of acceptance test and planning to develop iteration [5]. The first phase of Scrum is 'Pregame'. Its main activities are defining a new release using backlog, estimation of time and cost, analysis, design architecture and interfaces [5]. The designing activities of Scrum are excluded in the 'Plot' phase of the proposed XScrum model to perform in the 'Pattern' phase. In this phase, the product owner describes project plan of the system to be developed after defining the requirements as sprint goals. The 'Plot' phase defines empirical estimations, product backlog, sprint backlog and communication and quality assurance activities. The product backlog contains all the business and technical requirements of the system narrated by the customer. The product backlog is constantly updated as per the new stories arriving from the customer. Product owner evaluates the backlog to update priorities as mandatory. He is a mentor and mediator of each sprint cycle. The team estimates the size of sprint goals in terms of development weeks as a sprint backlog during sprint planning meeting. Unlike product backlog, the items in sprint backlog are changed in the XScrum model in order to welcome requirements to achieve customer satisfaction.

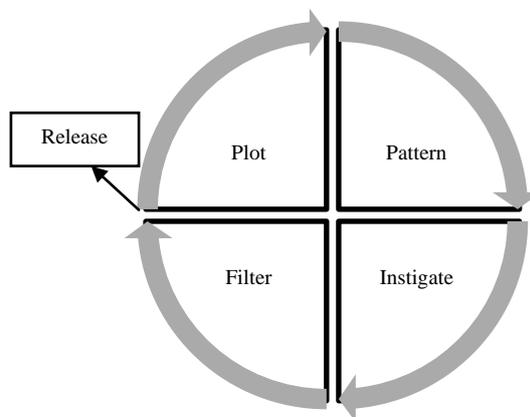


Fig.1. The proposed XScrum Model.

Daily Scrum meeting is arranged to resolve the problems of team members. Scrum of Scrum planning meeting is arranged to ensure goal achievement and successful integration of the work. XScrum master is responsible to answer the work of his team members in

daily scrum meeting like what is the progress of his team and what kind of impediments they are facing. The Sprint Planning meeting is conducted at the start of each sprint, in which the presence of whole teams is mandatory, for thirty minutes. Sprint backlog is changed only for small projects unlike the Scrum model. Each sprint is completed in a stipulated period of time.

B. 'Pattern' Phase.

The existing XP used 'Design' phase to produce metaphor and spike solution and Scrum used 'Pregame' to produce high level architecture and design [5]. Spike solution is an operational prototype. The 'Pattern' phase of the proposed XScrum model includes spike solution activity from XP and high level architecture and design activities from Scrum. These activities are incorporated into 'Pattern' phase to improve the scalability problems of XP and Scrum to develop for large size projects.

Table 3. The Main Activities of Proposed XScrum Model

Phases → Activities ↓	Plot	Pattern	Instigate	Filter
Process	1. Define sprint goals 2. Empirical estimations 3. High level planning 4. Product backlog 5. Sprint backlog	1. Architectural 2. Design 3. Interface 4. Database	1. Code 2. Test 3. Maintain 4. Release	1. Code review 2. Retrospective
Communication & Coordination	6. Sprint Planning 7. Daily Scrum 8. Scrum of Scrum	5. Simplicity	5. Pair Programming	3. Behavioral analysis
Quality Assurance	9. Team selection 10. Define roles 11. Analysis	6. Inspection	6. TDD 7. Documentation standard 8. Refactoring 9. Code ownership	4. Postmortem analysis 5. Feedback loop
Customer Satisfaction	12. Open to accept requirements at any stage (only for small projects)	7. Spike solution	10. Early and continuous delivery to customer	5. Improve

High level architecture will facilitate the development team to use design patterns. The design pattern helps to save time and cost to develop a sprint and it increases efficiency, reliability and reusability of the developed software. Interface specification (IS) is prepared in each sprint cycle. Spike solution is used to approve the sprint iterations and interfaces. Architecture is kept simple to increase the scalability of software and facilitate understanding of the development team. Inspection and refactoring cycles are also applied to improve design. Test classes are also designed during the 'Pattern' phase.

C. 'Instigate' Phase.

The existing XP model used 'Coding' phase to write unit tests and code whereas 'Game' phase of Scrum model is used to develop, wrap, review and adjust sprints [5]. 'Testing' is the final phase of the existing XP. It is used to perform unit, integration, system and acceptance tests. 'Coding' and 'Testing' stages of the existing XP are integrated to achieve agility in the 'Instigate' phase of proposed XScrum model. The proposed changes in the XScrum will enable it to deal with the development of large size projects. 'Game' phase activities of the existing Scrum model are also integrated into the 'Instigate' phase. In the proposed XScrum, developers use pair programming to complete the assigned tasks.

The Unit tests are already developed during the 'Pattern' phase of the proposed XScrum model. IS document of the next sprint is prepared in parallel while

developers are coding for the current sprint. Test-driven development (TDD) environment will be used to speed up the development and testing cycles. Unit, Integration, System tests are performed. Acceptance test is performed before a sprint is released. Refactoring technique is performed throughout the development to improve the quality of code. The product owner makes sure that software is delivered in small releases to achieve scalability, manageability, productivity, quality assurance and customer satisfaction. The release is ready to deploy followed by training and security.

D. 'Filter' Phase.

Post mortem analysis is performed at the end of a sprint. Post mortem analysis activity is added in the proposed XScrum model to increase its scalability to develop large projects as supported by other researchers [28-31]. It is similar to the idea of Retrospective practice of XP as proposed by Miller [31]. It is conducted to improve learning of team members and avoid mistakes in the upcoming sprints. The objective of post-mortem analysis is to establish a mechanism of feedback loop throughout the software development. One of the main objectives is to reorganize extra actions required to improve upcoming sprints. The lessons learned are considered at the start of upcoming sprint. The post mortem analysis is added in the proposed XScrum to act as a filter to check its effectiveness, take improvement actions, monitor the level of adoption, get feedback about

the experiences of team and recommend necessary actions for upcoming releases. According to Salo and Abrahamsson [29], post-mortem reviews proved to establish procedures to improve software development and research processes during a project. The proposed XScrum model is cyclic and evolutionary like XP and Scrum models.

VI. VALIDATION

Agile software development is iterative and there are frequent changes in both the product and process metrics as compared to plan driven development. Therefore, there is a pressing need to merge case study and action research approaches efficiently to conclude empirical results about agile software development [32-33]. Therefore, the action research and case study approaches are merged in the research setting to conclude the results [29]. A single case approach tends to provide insight details of a process or project to provide data but its unique nature is difficult to compare and generalize the results [29]. Therefore, three projects are conducted to address this problem. The three projects are conducted in three different software companies who are willing to participate. The author worked during the three projects as product owner and mediator of the post mortem analysis. One Scrum Master is selected from each team and he took certified Scrum Master course before starting the case study research. Only those team members are selected who have attitude of agile work to achieve self-organization. The author works with the team throughout the development of first four releases for the three projects and he spends more than 90% of his time at work places. An intensive two weeks training is organized with each team separately that is taken part in the proposed research. The training is focused on the main practices of the proposed XScrum model. According to the existing approaches [28-29], the author notices that 2 to 6 iterations are taken into consideration to describe the results and asserting some judgments. Therefore, the author considers it suitable to drive the results by providing the data of initial four iterations of the three cases. The description of three cases is provided in Table 4.

Table 4. Specification of the Three Projects

Attributes	Project 1	Project 2	Project 3
Team size	7	11	20
Project title	Transportation System	Financial System	Invoice Management System
Experience of the Team in Agile development	Medium	Medium	Medium
Iterations compared	4	4	4
Number of user stories completed	72	92	160
Size of the Iterations	12 KLOC	22.52 KLOC	47 KLOC

Project 1 develops a Transport Management System in a small size software company that is working globally. The top management of company shows its interest to introduce agile methodology to get its benefits. A small team is chosen for this pilot project i.e., seven members from twenty employees. The duration of pilot project is three months. The top management are committed if the pilot project is successful then the proposed model will be practiced for the rest of projects running in the company. Project 2 is conducted in a software company that is facing a critical problem while transiting from traditional methodology to XP. The author selects a team of eleven members out of hundred employees to implement the proposed XScrum model. The duration to complete the project 2 is one year. Project 3 develops an Invoice Management System in a software company. A team of twenty members are selected from five hundred employees. Team is previously using customized model that is a blend of traditional Waterfall and agile XP. The top management of company is highly supportive to integrate the Scrum into XP (i.e., the proposed XScrum model) to attain the potential benefits of both agile methodologies. Product owner decides that each project will be delivered following iterative approach. A sprint is composed of several iterations. The Product owner approves the sprint before shipment.

Quantitative data is collected in this research using the time, size and defect data sets as per the instructions of Humphrey [34]. Qualitative data is reported using research diaries based on participants' observations, direct observations, post mortem analysis sessions, interviews and surveys [28-29][35]. The research diaries are completed after interviewing with the team members based on their observations even in the absence of author.

Interview and survey questions are not included in this paper due to the confidentiality issue.

A. The Results of Three Case Studies.

Table 5 shows that the average time to complete iteration in weeks is 1.7 in Project 1, 2.2 in Project 2 and 2.5 in Project 3. The average size in kilo line of code (KLOC) per iteration is 3 in Project 1, 5.6 in Project 2 and 11.7 in Project 3. The average test coverage in percentage (%) per iteration is 89.2 in Project 1, 90 in Project 2 and 88.7 in Project 3. The average defects per KLOC, before iterations, are 2.2 in Project 1, 3.5 in Project 2 and 3.7 in Project 3. The average defects per KLOC, after iteration, are 1.2 in Project 1, 2.1 in Project 2 and 2.3 in Project 3. The pair programming (pp) is practiced extensively during the Projects 1 to 3 and the average pp per iteration (in %) is 78.4, 83.7 and 83 subsequently. The averages of post-mortem analysis time and efforts per iteration are 2.2 hours and 3.4% in Project 1, 3.2 hours and 4.7% in Project 2 and 4.3 hours and 4.8% in Project 3. The average customer satisfactions (in %) per iteration is 89 in Project 1, 88 in Project 2 and 85 in Project 3.

The results of the case study research are accomplished through the post-mortem analysis. Table 5 illustrates the results by taking the average of time (in hours) and effort

(in %) that are consumed per iteration to perform post-mortem analysis of the three projects. It is described by several agile case studies that two to four hours are minimum time consumed to conduct post-mortem analysis and the effort employed on lightweight post-mortem reviews is about 4.7% [28]. In Table 5, the average time is 2.2 hours for Project 1, 3.2 hours in Project 2 and 4.3 hours in Project 3. Table 5 shows that the average effort is 3.4% in Project 1, 4.7% in Project 2 and 4.8% in Project 3. The proposed XScrum model demonstrates its effectiveness for Projects 1 to 3 by

accomplishing the post-mortem analysis within the speculated time as recommended in existing XP for small projects [36-37]. Another indication of increase in quality of the proposed XScrum is presented because the effort in Project 1 is 3.4% that is fewer than 4.7% of the prescribed effort of existing XP for small projects. In Table 5, an average improvement in time and effort are recorded from Projects 1 to 3 due to growth in the sizes of the iterations i.e., 12 Kilo Line of Code (KLOC) in Project 1, 22.5 KLOC in Project 2 and 47 KLOC in Project 3.

Table 5. Data Analysis of 1st four Iterations of the three projects

	Time in week	Size in KLOC	Test Coverage in %	Pre Release Defects per KLOC	Post Release Defects per KLOC	Pair Programming in %	Post-mortem Analysis Time in hrs.	Post-mortem Analysis Effort in %	Customer Satisfaction in %
Project 1									
R1	2	1.6	92	3	1.4	80.7	3.2	4.3	88
R2	2	2.6	88	2	1.4	78.3	2.8	3.9	89
R3	2	3.4	87	2	1.2	75.0	1.8	2.7	90
R4	1	4.4	90	2	1.0	79.7	1.0	2.8	92
Avg.	1.7	3	89.2	2.2	1.2	78.4	2.2	3.4	89
Project 2									
R1	3	3.2	90	5	2.34	80	3.7	4.4	85
R2	2	4.5	85	3	2.30	80	3.1	4.5	88
R3	2	6.8	92	3	2.15	85	2.8	4.7	92
R4	2	8.02	93	3	2.0	90	3.0	4.8	87
Avg.	2.2	5.6	90	3.5	2.1	83.7	3.2	4.7	88
Project 3									
R1	3	10.4	90	6	2.6	85	4.5	5.3	87
R2	3	12.1	89	3	2.38	78	4.8	5.1	88
R3	2	14	89	3	2.1	79	4.9	4.9	83
R4	2	10.5	87	3	2.0	90	3.2	4.2	82
Avg.	2.5	11.7	88.7	3.7	2.3	83	4.3	4.8	85

Table 6. Defect Rate/KLOC

	1	2	3	4	Average
Project 1	1.4	1.4	1.2	1	1.2
Project 2	2.3	2.3	2.1	2	2.1
Project 3	2.6	2.3	2.1	2	2.3
Existing XP Case Study [28]	2.1	2.1	2.0	8.7	3.7
Existing Scrum Case Study [22]	2	1	1	N/A	1.3

There is regular reduction of time and effort from iteration to iteration in the three projects as shown in Table 5. This is due to the discipline learning of the teams about the projects as they progressed from iteration to iteration. The number of defects is used to infer the results those are recorded of the three projects. In Table 6, the proposed XScrum projects are analyzed with existing XP [28] and Scrum [22] projects by reporting the comparative ratios of quality assurance (QA) to demonstrate the opinion that the quality of proposed XScrum is increased or decreased in contrast to the quality of existing XP and Scrum models.

Sajid and Jongmoon [38] are of the opinion there is possibility to analyze the projects by examining the comparative ratios for QA to take proof of quality. Defect rate per KLOC (four iterations of three projects) to Plan, Pattern, Instigate and Filter as a complete for each iteration of Projects 1 to 3 consequently to estimate the average rate. In Table 6, this is achieved by matching the

average defect rate per KLOC of three projects with the average of four iterations of a case study of existing XP [28] and the average of three iterations of a case study of existing Scrum [22]. The average defect rate per KLOC is 1.25 in Project 1, 2.1 in Project 2 and 2.3 in Project 3 whereas it is 3.7 and 1.3 defect rate per KLOC in the existing XP and Scrum case studies for small size projects. The average defect rate per KLOC for the small size Project 1 is less as compared to the existing XP and Scrum models reflecting that quality of the proposed XScrum model is enhanced than the existing XP and Scrum models. The enhancement in quality of the proposed XScrum model is due to the integration of XP and Scrum models warrants its proposal.

VII. CONCLUSION

Scrum model has gained popularity among agile methodologies helping to manage projects efficiently. Scrum is kept silent that how to engineer a software. XP model has several success reports for the development of small projects. Several attempts are done to analyze XP and Scrum but little focus is paid to integrate XP and Scrum models. There is more work required to provide empirical evidences for the fruitful integration of XP and Scrum models. The same objectives are accomplished in

this research by proposing XScrum model. The research validity is evaluated by conducting three industrial projects. The results are extracted mainly using the post-mortem analysis and defect rate per KLOC. The proposed XScrum model reflects its effectiveness by accomplishing the post-mortem analysis in two to four hours as suggested for XP projects. Three XScrum projects are compared with the existing XP and Scrum projects by comparing the values of quality assurance (QA) to conclude that quality of the proposed model is higher or lower than the quality of existing XP and Scrum. It is noticed that quality of the proposed model is enhanced as compared to the existing XP and Scrum because of less number of defect rate per KLOC in case of small size projects. From the results, the author has established supports that the proposal of XScrum model meets its objectives by integrating the strengths of XP and Scrum and suppressing their limitations. However, statistical evaluation of the XScrum proposal will be dealt with comparing forthcoming releases in days to come.

REFERENCES

- [1] P. A. Beavers, "Managing a Large "Agile" Software Engineering Organization," *Proc. Conf. Agile*, pp. 296-303, 2007.
- [2] B. Boehm, "A Spiral Model of Software Development and Enhancement," *Computer Journal*, vol. 21, no. 5, pp. 61-72, 1988.
- [3] I. H. Sarker, F. Faruque, U. Hossen, A. Rahman, "A Survey of Software Development Process Models in Software Engineering," *International Journal of Software Engineering and Its Applications*, vol. 9, no. 11, pp. 55-70, 2015.
- [4] T. Johansen, T. Gilb, "From Waterfall to Evolutionary Development (Evo): How we Created Faster, More User-Friendly, More Productive Software Products for a Multi-National Market," *Proc. of INCOSE*, Roch-ester, New York, pp. 1676-1686, 2005.
- [5] R. S. Pressman, *Software Engineering*, 7th ed., McGraw Hill, USA, 2009.
- [6] A. Sutharshan, S. P. Maj, "An Evaluation of Agile Software Methodology Techniques," *International Journal of Computer Science and Network Security*, vol. 10, no. 12, pp. 68-71, 2010.
- [7] R. Green, T. Mazzuchi, S. Sarkani, "Communication and Quality in Distributed Agile Development: An Empirical Case Study," *WASET*, pp. 322-328, 2010.
- [8] D. Boland, B. Fitzgerald, "Transitioning from a Collocated to a globally-distributed software development team: a case study at analog devices inc.," *Proc. ICSE Workshop on Global Software Development*, Scotland, pp. 4-7, 2004.
- [9] T. Dyba, T. Dingsoyr, "Empirical studies of agile software development a systematic review," *Journal of Information and Software Technology*, vol. 50, no. 9-10, pp. 833-859, 2008.
- [10] R. Noor, M. F. Khan, "Defect Management in Agile Software Development", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 6, no.3, pp.55-60, 2014.
- [11] W. Ali, Zia-Ur-Rehman, A. Badshah, A. Javed, "Software Inspection in Software Industry: A Pakistan's Perspective", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 7, no. 3, pp.47-53, 2015.
- [12] F. Anwer, S. Aftab, "SXP: Simplified Extreme Programming Process Model", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 9, no.6, pp.25-31, 2017.
- [13] N. B. Moe, T. Dingsoyr, T. Dyba, "A teamwork model for understanding an agile team: a case study of a scrum project," *Journal of Information and Software Technology*, vol. 52, no. 5, pp. 480-491, 2010.
- [14] V. Mahnic, "A case study on agile estimating and planning using Scrum," *Electronics and Electrical Engineering*, vol. 111, no. 5, pp. 123-128, 2011.
- [15] K. Schwaber, M. Beedle, *Agile Software Development with Scrum*, Pearson, USA, 2002.
- [16] P. Abrahamsson, J. Warsta, M. T. Siponen, J. Ronkainen, "New Directions on Agile Methods: A Comparative Analysis," *Proc. 25th Int. Conf. Software Engineering*, Portland, Oregon, pp. 244-254, 2003.
- [17] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, USA, 2004.
- [18] B. Boehm, "Get Ready for Agile Methods With Care," *Computer*, vol. 35, no. 1, pp. 64-69, 2002.
- [19] A. Qumer, B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering," *Information and Software Technology*, vol. 50, no. 4, pp. 280-295.
- [20] A. Qumer, B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice," *Journal of System and Software*, vol. 81, no. 11, pp. 1899-1919, 2008.
- [21] E. Caballero, Jose A. Calvo-Manzano, and San Feliu Tom á, "Introducing Scrum in a Very Small Enterprise: a Productivity and Quality Analysis," *Proc. European Conference On Software Process Improvement*, Denmark, , pp. 215-224, 2011.
- [22] C. Malhotra, and A. Chug, "IXSCRUM-A Framework Combining Scrum and XP," *International Journal of Scientific & Engineering Research*, vol. 4, no. 7, pp. 1322-1328, 2013.
- [23] J. Sutherland, A. Viktorov, and J. Blount, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," *Proc. Int. Conf. Agile*, USA, pp. 274a-274a, 2006.
- [24] H. Kniberg, and R. Farhang, "Bootstrapping Scrum and XP under Crisis A Story from the Trenches," *Proc. Int. Conf. Agile*, Toronto, Canada, pp. 436-444, 2008.
- [25] N. R. Zuiderveld, "eXtreme programming and scrum a comparative analysis of agile methods," *Proc. Int. Conf. Software Engineering*, Portland, USA, pp. 1-7, 2003.
- [26] S. Sultana, Y. H. Motla, S. Asghar, M. Jamal, R. Azad, "A hybrid model by integrating agile practices for Pakistani software industry," *Proc. Int. Conf. Electronics, Communications and Computers (CONIELECOMP)*, Cholula, pp. 256-262, 2014.
- [27] A. Marchenko, and P. Abrahamsson, "Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges," *Proc. Conf. Expanding Agile Horizons*, Toronto, pp. 15-26, 2008.
- [28] P. Abrahamsson, J. Koskela, "Extreme programming: a survey of empirical data from a controlled case study," *Proc. Int. Symposium Empirical Software Engineering*, USA, pp. 73-82, 2004.
- [29] O. Salo, P. Abrahamsson, "Empirical Evaluation of Agile Software Development: The Controlled Case Study Approach," *Lecture Notes in Computer Science* (Springer Verlag, Berl ín, Heidelberg), pp. 408-423, 2004.

- [30] M. Visconti, C. Cook, "An Ideal Process Model for Agile Methods," *Lecture Notes in Computer Science* (Springer Verlag, Berlin, Heidelberg), pp. 431-441, 2004.
- [31] R. Miller, "Demystifying Extreme Programming, "XP distilled" revisited," <https://www.ibm.com/developerworks/library/j-xp0813/index.html>, accessed October 2017.
- [32] M. Pikkarainen, O. Salo, J. Still, "Deploying agile practices in organizations: a case study," *Lecture Notes in Computer Science* (Springer Verlag, Berlin, Heidelberg), pp. 16-27, 2005.
- [33] J. B. Chunningham, "Case study principles for different types of cases," *Quality and Quantity*, vol. 12, no. 4, pp. 401-423, 1997.
- [34] W.S. Humphrey, *Managing the software process*, Addison-Wesley Longman Publishing Co., Inc. Boston, 1989.
- [35] M. B. Miles, A. M. Huberman, *Qualitative data analysis: an expanded sourcebook*, Sage Publications, 2nd Edition, USA, 1997.
- [36] O. Salo, K. Kolehmainen, P. Kyllönen, J. Lähman, S. Salmijärvi, and P. Abrahamsson, "Self-adaptability of agile software processes: a case study on postiteration workshops," *Proc. Fifth Int. Conf. on Extreme Programming and Agile Processes in Software Engineering*, Germany, pp. 184-193, 2004.
- [37] T. Dingsøyr, G. K. Hanssen, "Extending Agile methods: postmortem reviews as extended feedback," *Fourth Int. Workshop on Learning Software Organizations*, Chicago, Illinois, USA, pp. 4-12, 2002.
- [38] I. H. Sajid, B. Jongmoon, "Software Quality Assurance in XP and Spiral - A Comparative Study," *Proc. 5th Int. Conf. Computational Science and its Applications*, Malaysia, pp. 367-374, 2007.

Authors' Profiles



Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree in Computer Sciences from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honoured as a best researcher in Computer Science discipline from seven campuses of COMSATS Institute of Information Technology, Pakistan in 2008.

How to cite this paper: M. Rizwan Jameel Qureshi, "Evaluating the Quality of Proposed Agile XScrum Model", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.9, No.11, pp. 41-48, 2017.DOI: 10.5815/ijmeecs.2017.11.05