OSCRUM: A Modified Scrum for Open Source Software Development

Sheikh Shah Mohammad Motiur Rahman, Saiful Alam Mollah, Shikha Anirban, Md. Habibur Rahman,
Mostafijur Rahman, Md. Maruf Hassan, Md. Hasan Sharif

Department of Software Engineering
Daffodil International University
Dhaka, Bangladesh
motiur.swe@diu.edu.bd, saiful.alam15@gmail.com, anirban@daffodilvarsity.edu.bd, mdhabibur.r.bd@ieee.org,
mostafijur.cse@gmail.com, maruf.swe@diu.edu.bd, sharif.swe@diu.edu.bd

***Abstract* - The Open Source Software Development (OSSD) is a movement, challenges many traditional and commercial theories of software development. A group of developers, programmers, and other community members develop the Open Source Software (OSS) in a collaborative manner. Community and contributors provide great support to make the source code of the software easily understandable and modifiable. However, there are insignificant such standard model or methodologies for OSSD has yet been established. Currently, researchers are proposing methodologies in this area. This paper proposes a new model, OScrum by modifying the scrum to make it applicable for OSSD. The proposed model has been constructed after analyzing the key metrics, pillars, and values of Scrum and OSSD. The model has been evaluated through comparing implementation process and working procedure of OSSD. The result shows that the implementation process of OSCRUM has a very close relationship with the process of OSSD and therefore, it fits well in such software development.**

Keywords - Open Source; Scrum; OSSD; Comparison Metrics; Agile.

I. INTRODUCTION

Open Source Software Development has become rapidly growing and widely preferable term in the field of software development. Open Source Software has become much popular among developers, organizations etc. There are various kinds of places on the web in which Open Source Software has hosted such as Source Forge, GitHub [1,2], Google Code [3], Bitbucket [4] etc. Some of those platforms have the power of version control management. From all of those platforms, SourceForge [5], was one of the first platforms which offer to host projects for free to Open Source projects. According to the official report of Sourceforge, they claimed to host millions of registered users and host over 500,000 projects and also connects more than 30 million users per month to their directory. All of the Open Source projects and serves which are hosted on SourceForge has more than 4 million downloads per day [6]. But now, many users have migrated their projects to GitHub because of extra facilities. Open Source Software Development create a community which has huge interaction and intercommunication among the world's top developers and also various kinds of users. That's why Open Source Software has a large number of testers, developers, contributors also. On the other hand, Scrum also plays a vital role as an agile approach in the field of traditional software development. Scrum has greater acceptance in software development industry and it is also rapidly growing and widely used software development framework. In recent a research claimed that the most adopted agile approach in the software development industry is Scrum [7,8].Scrum was originally designed for use in small team members of whose are collocated, face-to-face working [9,10]. Open Source Software [11] is developed with multiple teams or contributors distributed to several geographical locations. During the adoption of the scrum in the field of Open Source Software Development [12,13] there are much more similarities found between the process of Open Source Software Development and the Scrum. Some conflicts also happen with the time-boxed concept of Scrum and some other issues. This study proposes a scrum based model for open source software development after identifying and comparing the similar and dissimilar metrics or terms of Scrum and Open Source Software Development.

This paper is organized into six sections. Introduction and proposed model generation are discussed in section I and II respectively, a comparison between scrum and OSSD are explained in section III. Proposed model of this research is discussed in section IV where OScrum model has been described. Implementation process and evaluation of proposed model are presented in section V. The paper is concluded with the outcome of the study, limitation, and future work in section VI.

II. GENERATION OF PROPOSED MODEL

To construct the proposed model, the key metrics of open source and scrum have been identified. The similarities and conflicts between Scrum and OSSD have also compared in this study. The properties of Scrum and OSSD are discussed below.

A. Open Source Software Development

Open Source Software Development (OSSD) has been successfully implemented and has received huge attention in the last decade. OSSD not only throws a challenge but also threatens to develop the software faster, better and cheaper than the traditional software development [14,15]. The term Open source is not only just limited to the access to the source code, but also follows some criteria. A software will be termed as an open source software if it complies with the criteria of Open Source Definition (OSD), provided by the Open Source Initiative (OSI) which derived from the Debian Free Software Guidelines (DFSG) [16,17].

The criterion or the conditions of Open Source Definition outlined in Table I [18].

TABLE I. CRITERION OF OPEN SOURCE DEFINITION.

Criteria/Conditions	Description
Free of Cost	The license of the software or project will be given with full rights to reproduce, redistribute and sell the software or project to any party, either free or for a fee that means reproducible, redistributable and also sell-able without any restrictions or fees to the users.
The source code must be open to all users.	The distribution of the software of projects will include the original and runnable source code of the all released versions or the latest stable versions, and the source code must be available publicly so that users can download the codes freely.
No restrictions are allowed to modify and derivative works	Modification of the software or produce the derivative works will be permitted. Distribute the derivative works under the same conditions or license as original software has must be allowed.
Restriction of source codes from being distributed may be allowed if the distribution of "patch files" with the author's source code for modifying the program or projects at build time	The license may have the restriction to the source codes from being distributed by the form of modification only if the license or terms of the program or project allows the "patch files" distributions with the source code for modification purpose of the program during build time. The terms must openly permit distribution of the software which is built from modified source code. To use a different software name or version number the license may require derivative works.
Discrimination against any individual, group of individuals or field of exertion are not allowed.	The license must not discriminate against any individual, group of individuals. Even also must not apply any restriction to anyone from making the use of the program in a specific field of effort
The license distribution of the software or projects criteria	The license distribution of the software or projects should not be product specific also not restricted to the other software which is distributed along with the licensed software. The license must not force to all other programs or software or projects which is distributed on the same medium must be open source software.
The license must not be any specific technology oriented	The license must not apply any limitation on any particular technology or any styles, it should be technology independent or technology neutral.

B. Pillars of Open Source

From above study, the pillars of Open Source Software has derived as outlined in Table II.

TABLE II. PILLARS OF OPEN SOURCE SOFTWARE

Pillars	Meaning
Restriction Free	No restriction to use, to share, to update or to modify the software according to specific needs.
Open Participation	Open for all. Anyone can participate.
Openness	Completely open for academic research, peer review, participate, contribution and so on.
Collaboration	Collaboration means collective engagement and sharing knowledge.
Modularity	Modularity characterizes the Open Source Development style such as peer review control, limited information hiding to another developer etc.
Community	Communities bring together diverse ideas and share work. A community can create beyond the capabilities of any one individual and multiplies the effort.
Transparency	An Open Source company or individual should be transparent that means frank, honest and operating with integrity.
Early and Often	Making the information available with the earliest period of time so that others can check and review.
Releases Versioning	To maintain a versioning for every milestone of releases.

C. Scrum

Scrum is not just a definitive method or technique but a framework in which various techniques and processes can be applied. Scrum helps to achieve a clear view of the relative efficacy of various techniques and product management [19]. It is a framework which consists of roles, events, artifacts, and the rules that bind them together. Scrum is an iterative and incremental process for developing any software. Scrum concentrates on how the team members should act in a constantly changing environment in order to build the system flexibility. Scrum helps to make a real-time decision based on actual events and information. This requires teams which are well-trained, specialized, and capable of self-management, communication, and decision-making [20]. Key metrics of Scrum practices are outlined in Table III

TABLE III. KEY METRICS OF SCRUM PRACTICES.

Scrum Roles	Events	Artifacts
The Product Owner The Development Team The Scrum Master	Sprint Planning Daily Scrum Sprint Review Sprint Retrospective	Product Backlog Sprint Backlog Increment

D. Scrum Theory and Scrum Values

According to the founder's guides of the scrum, Scrum is founded based on empiricism or empirical process or industrial [21] control theory.

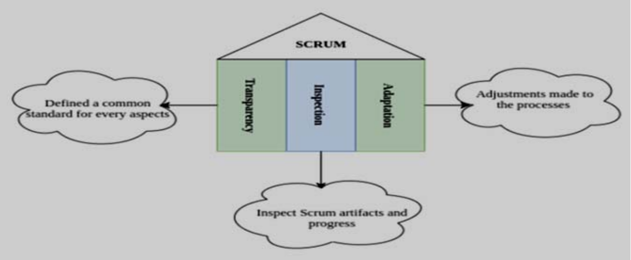
SCRUM
Transparency
Inspection
Adaptation
Defined a common standard for every aspects
Adjustments made to the processes
Inspect Scrum artifacts and progress

Figure 1. Three pillars of Scrum.

Empiricism possess the knowledge comes from experience and based on the known knowledge makes decisions. Scrum ensure an incremental and iterative approach to optimize the ability to predict and control the risk. For the implementation of empirical process control, there are three pillars [22],[23] exist transparency, inspection, and adaptation (See the above Fig.1).

The pillars of scrum come to life and build trust for everyone when the values of commitment, courage, focus, openness, and respect are lived by the Scrum Team. The team members learn and explore those values as they work with the Scrum roles, events, and artifacts.

Successful use of Scrum depends on teams becoming more proficient to the line in five values.

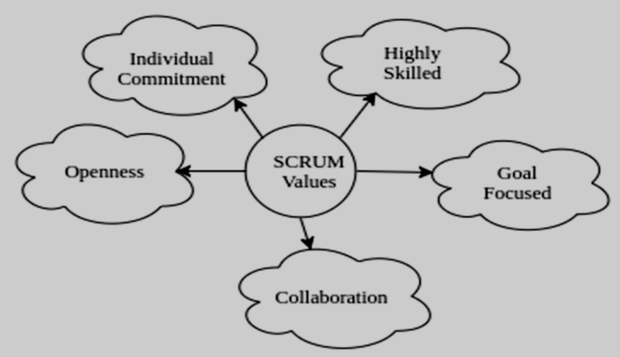

Figure 2. Key values of Scrum.

III. Comparison Between scrum and open source software development

A study expressed Scrum as an "Agile based software development process design or develop to add strength, attention, evidence, visually, and transparency to development teams during developing the software systems"[24]. The Scrum processes follow the agile approach principles. The principles of Agile approach mainly focus on: ensure customer satisfaction through early delivery of the software, changing requirements are welcome, the developer and customer collaboration, release working software after every milestone as a customer can visualize and measure the progress, simplicity, making the development teams more effective during the development [25]. An analysis found that at least partially agile principles are present in open source projects [26]. He considers the main principles of agile and open source development like highly skilled individuals in a self-organizing development team, continuously changing environment based on feedback or review, frequently release a working version of software, integration, and collaboration etc. There is a comparative study of similarities and conflicting issues or terms during applying scrum in OSSD. Considering the key metrics, pillars and values of the scrum from scrum guides provided by scrum founders and the key metrics, pillars and values of open source software development gathered from Open Source Initiative and from Raymond's description [27], the similarities and conflicting issues are compared.

A. *Similarities*

The terms "Scrum" and "Open Source" are based on a similar set of pillars and values or principles such as openness, early and often, transparency, collaboration etc. Scrum has a milestone of sprints which also exist in Open Source Development as a milestone of versions. Scrum has a small team of Open Source core contributors or core developers. Scrum team and contributors of Open Source project both are self-managing. The key similarities are as outlined in Table IV.

TABLE IV. Similarities between Scrum and Open Source Software Development

Scrum	Open Source
Collaboration between a team and business persons	A collaboration of contributors and users
Transparency among the developer's team, organization and product owner	Transparency among main maintainer, contributors, and users.
Milestone of sprints	Milestone of versions.
Scrum Team	Developer Team or Contributor
Team members are highly skilled individuals	Contributors are highly skilled individuals
Team members are self-organizing	Contributors are self-motivating
Continuous Feedback Gathering	Continuous Reviews and Issues from Users
Requirements Changing Environment	Features change based on user demand.
Scrum Master	Main Maintainer
Acceptance Testing (Sprint Review)	Customer Reviews
Product backlog	Feature Lists
Sprint Backlog	Release-wise Feature Separation
Community consisting product owner, scrum master, developer team, management and stakeholders	A community consisting the main maintainer, contributors and anyone for a short period of time who wants to be
Sprint burndown chart	Progress chart, Commit Chart

B. *Conflicting Issues*

In Open Source perspective, there are some conflicting issues exist in traditional Scrum methodology [28,29].

Generally, Scrum methodology has sprint lengths around 2-4 weeks [30]. But in Open Source development sprint lengths is based on an assumption. In Scrum, the development team should be physically located close to each other. But in Open Source development the core developer team may locate close to each other or may not. In Scrum, there is a community consists of a product owner, scrum master, developer team, management and stakeholders which are fixed. But in Open Source development the community is not fixed and may engage for a shorter period of time. In Scrum, acceptance testing is done by the specific community. But in Open Source Development anyone can do it. In Scrum, "Product Backlog" has a timeline for implementation. But in Open Source Development it may have or not.

The key conflicting issues are as outlined in Table V.

TABLE V. CONFLICTING ISSUES BETWEEN SCRUM AND OPEN SOURCE SOFTWARE DEVELOPMENT

Scrum	Open Source
Sprint Length is fixed	Release Length is counted based on an assumption.
Team members physically located, close to each other	Contributors may locate close to each other or may not
Community member are fixed ex: product owner, scrum master, developer team, management and stakeholders	Community members are not limited or fixed. Open to All.
Community and stakeholders are specified for acceptance testing	Community and stakeholders are not specified. Everyone is allowed to participate.
Face to face communication(Scrum events)	Communication to each other may be face to face or distributed using an electronic medium.
Fixed timeline for implementation	Fixed implementation timeline may have or may not.
The short meeting daily basis	Daily basis meeting may happen or not. But still, happen for commercial open source development.
A review is basically performed by stakeholders including product owner for accepting the completed features whose are fixed	A review is performed by the community as accepting and bug reporting of the features.
Stakeholders are targeted users or consumers who are not directly involved but support the development process	Anyone can be a part of stakeholders. Open to all.
Product Owner is one individual, not a committee	The main maintainer, contributors, and users also can be classified as a product owner. Mainly main maintainer and contributors have performed this role.
Almost all activities are time boxed	All activities may be time-boxed or not, it depends.

IV. PROPOSED MODEL

From table IV and Table V, the proposed model has been constructed. From Table IV, it identifies the similarities between scrum and open source software development. From table V, all possible conflicts are identified that may happen in case of implementing scrum in open source software development. However, modification of some rules in scrum can be adopted in OSSD.

The modified Scrum methodology is defined as the proposed model named “OScrum”. OScrum is described by comparing with Scrum in Table VI.

TABLE VI. COMPARATIVELY, DESCRIBE OSCRUM WITH SCRUM

Scrum	OScrum
Product Backlog	***Product Feature Backlog***
Requirements list which should be implemented	Feature list from main maintainer and core-contributors. It can be from others which are approved by the main maintainer. It includes requirements and design documentation if possible.
Sprint Planning	**Sprint Planning**
A meeting for selecting features which will be done during the upcoming sprint	Same. The main difference is “This should be public for extend team who has skills and interest”.
Sprint Length	***Sprint Length***
Sprint length is time bounded usually 2-4 weeks.	Sprint length is not fixed.
Sprint Retrospective	***Sprint Retrospective***
A meeting with the product owner, scrum master, developer team and management about the previous sprint	Same. A meeting which performs by main maintainer and core-contributor about the decision of developed features of the previous sprint
Product Owner	***Main Maintainer and Core-Contributor***
A responsible person for the product backlog	Basically, this role is filled by main maintainer and core-contributor.
Scrum Master	***Main Maintainer***
The responsible person for tracking daily update	This role is performed by the main maintainer
Scrum Team	***Core-Contributors and Extended-Contributors***
Developers team	Core-contributor and Extended-contributor including the main maintainer
Stake-holders	***Community***
Stake-holders are targeted users or consumers who are not directly involved but support the development process	Open to all
Story	***Story***
Describes the features of the software.	Same.
Sprint burndown chart	***Sprint burndown chart***
Progress chart of the sprint.	Same.
Sprint Backlog	***Sprint Feature Backlog***
The features which will be done in a sprint committed by scrum team	Same. But sometimes it may update if new contributions are accepted by main maintainer or core-contributors.
Daily Meeting	***Daily Meeting***
Short meeting about the daily progress of the development	May happen or not. But it should happen for commercial Open Source projects.
Sprint Review	***Sprint Review***
It is basically performed by stakeholders including product owner for accepting the completed features.	Performed by the community as accepting and bug reporting of the features

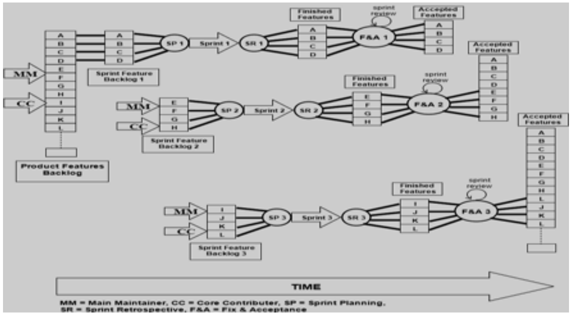

Figure 3. Overview of OScrum

In figure 3 A, B, C, D are an assumption of product features which are defined as product features backlog in OScrum. For selected features for one sprint will repeat in a loop. A sequence of sprints including fix and acceptance testing and also sprint review after each sprint retrospective in which stakeholders or consumers are including for feedback that happens after the end of the Sprint. The flexibility of changing requirements during development is also including. After one sprint completed there is an option for next sprint planning meeting.

It represents Main Maintainer and core contributors as a product owner and sometimes users also can be the part of product owned as user satisfaction is a primary priority here. The main Maintainer will also act as a scrum master.

V. IMPLEMENTATION PROCESS AND EVALUATION OF PROPOSED MODEL

Evaluation of the proposed model of OSSD has been conducted after comparing the implementation process of OSCRUM and the OSSD process.

TABLE VII. COMPARISON OF THE IMPLEMENTATION PROCESS OF OSCRUM AND THE OPEN SOURCE SOFTWARE DEVELOPMENT PROCESS

Criteria	OScrum	Open Source	Status
Problem Discovery and Finding volunteers	The main maintainer will generate an idea and write a description of that and publish to the web for gathering the core-contributors.	Started with an idea by an individual or a small group and published a short story on the web for self-motivated and self-interested contributors [4].	Similar
Communication	Arrange a meeting with core-contributors through the mailing list or VOIP tools like Skype for generating product features backlog and publish on the web.	The main maintainer and core contributors use mailing list.	Similar
Initial Release Planning Meeting	After generating product features backlog they will decide about the sprint length and create sprint features backlog which is called sprint planning meeting.	Open source software development consists of various activities which are described by using a story of the activities of OSS development from planning to release.	Similar
Release Plan and Status	They will complete the sprint features backlog.	The activities of OSS development are listed in release plan.	Similar
Features Update	During this sprint, if any new features get approved by the main maintainer or core contributors, the sprint features backlog will be updated.	There are so many alternative solutions are founded by the contributors. They choice reliable solutions and when get approved by all then it's being updated.	Similar
Source Code Test	The main maintainer and core-contributors will review the completed features which are called sprint retrospective and publish the project with documentation or screenshot on the web.	The contributors start coding to a local copy of the source code in his or her own environment. And after completion of their tasks committed to the main maintainer and core contributors using a continuous integration and version controlling platform like Bitbucket, Github, Gitlab etc for review. Once the solution is tested and approved by the main maintainer, it will be published to the web.	Similar
Error Report	The community will engage with the project and review the features and will create an issue if find any bugs or error in the project which is called sprint review.	By this time if users start to use the solution and get any problem then they create an issue or bug report under the project .	Similar
Independent to Join	By this time extended contributors will be added to the project and obviously approved by the main maintainer.	Contributors can make interest to join any time but the main maintainer will decide to add.	Similar
Fixing	The created issues or errors are fixed by contributors.	The contributors fix the issue or bug and commit to publishing on the web.	Similar
Approval	After committing the solved features, the community will recheck and accept	By solving bugs contributors will commit and main maintainer will	Similar

	the features which called fixed and acceptance review.	review and once getting approved, it will be published to the web.	
Iteration	This process will be continued according to the set of product features backlog.	The process is a continuous process which continued iteratively .	Similar

It is observed from the Table VII that the implementation process of OSCRUM has much more similarity with the open source software development process. Therefore, Oscrum has better perfection in Open Source Development.

VI. Conclusion

Scrum and Open Source Development have been broadly debated and find out the similarities and conflicts based on key metrics, main pillars and key values of both. It was found from the study that maximum conflicts occur with time boxed concept of scrum, specified stakeholders, individual product owner and some rules of scrum. At the same time, similarities among two have also been found such as the characteristics of developer team (i.e. highly skilled, self-managing, cross-functional), acceptance of feedback within a short loop, frequently release the working version, integration and collaboration, customer engagement and so on. As this paper represents a broad comparative study the similarities and conflicts between Scrum and Open Source Software Development and proposed a scrum based model OScrum. This research is a continuous process, the future research will implement OScrum on OSSD life cycle.

References

[1] O. Jarczyk, S. Jaroszewicz, A. Wierzbicki, K. Pawlak, and M. Jankowski-Lorek, “Surgical teams on GitHub: Modeling performance of GitHub project development processes,” *Information and Software Technology*, 2018.

[2] J. Jiang, D. Lo, X. Ma, F. Feng, and L. Zhang, “Understanding inactive yet available assignees in GitHub,” *Information and Software Technology*, vol. 91, pp. 44–55, 2017.

[3] *Google Open Source – opensource.google.com*. [Online]. Available: https://opensource.google.com/. [Accessed: 31-Mar-2018].

[4] “The Git solution for professional teams,” *Bitbucket*. [Online]. Available: https://bitbucket.org/. [Accessed: 31-Mar-2018].

[5] E. A. V. Hippel and G. V. Krogh, “Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science,” SSRN Electronic Journal, 2009.

[6] “About SourceForge,” About. [Online]. Available: https://sourceforge.net/about. [Accessed: 31-Mar-2018].

[7] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” Journal of Systems and Software, vol. 85, no. 6, pp. 1213–1221, 2012.

[8] T. Streule, N. Miserini, O. Bartlomé, M. Klippel, and B. G. D. Soto, “Implementation of Scrum in the Construction Industry,” Procedia Engineering, vol. 164, pp. 269–276, 2016.

[9] M. Villavicencio, E. Narvaez, E. Izquierdo, and J. Pincay, “Learning scrum by doing real-life projects,” 2017 IEEE Global Engineering Education Conference (EDUCON), 2017.

[10] V. T. Faniran, A. Badru, and N. Ajayi, “Adopting Scrum as an Agile approach in distributed software development: A review of literature,” 2017 1st International Conference on Next Generation Computing Applications (NextComp), 2017.

[11] J. Gamalielsson and B. Lundell, “Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved?,” Journal of Systems and Software, vol. 89, pp. 128–145, 2014.

[12] S. Lee, H. Baek, and J. Jahng, “Governance strategies for open collaboration: Focusing on resource allocation in open source software development organizations,” International Journal of Information Management, vol. 37, no. 5, pp. 431–437, 2017.

[13] K. Wei, K. Crowston, U. Eseryel, and R. Heckman, “Roles and politeness behavior in community-based free/libre open source software development,” Information & Management, vol. 54, no. 5, pp. 573–582, 2017.

[14] C. DiBona , and S. Ockman, “Open sources: Voices from the open source revolution,”. O'Reilly Media, Inc, 1999.

[15] A. Mockus, R. Fielding, and J. Herbsleb, “A case study of open source software development: the Apache server,” Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium.

[16] P. D. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G. Thallinger, D. J. V. Horn, and C. F. Weber, “Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities,” Applied and Environmental Microbiology, vol. 75, no. 23, pp. 7537–7541, Feb. 2009.

[17] “The Open Source Definition,” The Open Source Definition | Open Source Initiative. [Online]. Available: https://opensource.org/osd. [Accessed: 31-Mar-2018].

[18] L. Rising and N. Janoff, “The Scrum software development process for small teams,” IEEE Software, vol. 17, no. 4, pp. 26–32, 2000.

[19] “The Scrum Guide,” Scrum.org. [Online]. Available: https://www.scrum.org/resources/scrum-guide. [Accessed: 31-Mar-2018].

[20] P. Deemer, G. Benefield, C. Larman, and B. Vodde,”The scrum primer,”http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121. pdf, pp. 15, 2010.

[21] K. Schwaber, “Agile project management with Scrum,” Microsoft press, 2004.

[22] R. Handfield, “The scientific theory-building process: a primer using the case of TQM,” Journal of Operations Management, vol. 16, no. 4, pp. 321–339, 1998.

[23] L. J. Trevino, D. E. Thomas, and J. Cullen, “The three pillars of institutional theory and FDI in Latin America: An institutionalization process,” International Business Review, vol. 17, no. 1, pp. 118–133, 2008.

[24] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, “Distributed Scrum: Agile Project Management with Outsourced Development Teams,” 2007 40th Annual Hawaii International Conference on System Sciences (HICSS07), 2007.

[25] E. Hossain, M. A. Babar, and H.-Y. Paik, “Using Scrum in Global Software Development: A Systematic Literature Review,” 2009 Fourth IEEE International Conference on Global Software Engineering, 2009.

[26] S. Koch, “Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion,” Extreme Programming and Agile Processes in Software Engineering Lecture Notes in Computer Science, pp. 85–93, 2004.

[27] E. Raymond, “The cathedral and the bazaar,” Philosophy and Technology, vol. 12(3), pp. 23, 1999.

[28] “A statistical analysis of the effects of Scrum and Kanban on software development projects,” *Robotics and Computer-Integrated Manufacturing*, 17-Dec-2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584515301599. [Accessed: 31-Mar-2018].

[29] “Is Scrum and XP suitable for CSE Development?,” *Procedia Computer Science*, 01-Jun-2010. [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S1877050910001699. [Accessed: 31-Mar-2018].

[30] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, and J. Kern, "The agile manifesto," 2001.