Effectiveness of Scrum for Offshore Software Development in Sri Lanka

R.K. Chandana Ranasinghe and Indika Perera
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
chandana.fit@gmail.com, indika@cse.mrt.ac.lk

***Abstract*—This paper presents the results of a comprehensive literature survey and a web based survey on the use of Scrum in offshore software development (OSD) in Sri Lankan context. The trend in the recent software development industry is to move towards OSD. Issues and challenges related to OSD have to be solved to gain success. Scrum has gained a significant attention due to its flexible approach to managing requirement volatility and emphasis on extensive collaboration between customer and development team. However, the geographical and cultural distance of offshore teams create challenges for use Scrum with OSD. Hence objectives of the research was to identify issues and challenges related to OSD in Sri Lankan context and how those issues and challenges can be overcome by using Scrum. Results of the study reveal that key factors of Scrum such as communication, people, process, and organizational factors are having positive contribution towards OSD success in Sri Lankan context. This study further reveals the importance of combining engineering practices with Scrum to achieve the success in OSD in Sri Lankan context.**

Keywords—Offshore software development, Off-shoring, Global teams, Agile, XP, Scrum

I. Introduction

Offshore software development (OSD) is the process of developing software by a supplier who locates in a different country from clients [1]. The primary motivation behind OSD is cost. Several issues and challenges related to OSD have been identified, such as lack of communication, poor team collaboration and knowledge sharing. Solutions for these issues have been proposed, for example, divide software into separate modules and make the module development more independent in order to minimize communication between global sites. However for software projects which develop genuinely novel products, making a clear modular structure and minimizing communication might be difficult or even impossible.

In Sri Lankan software industry, there are larger number of companies involve with OSD. Scrum has gained a high popularity among Sri Lankan software companies including off-shoring companies, which are practicing Agile methodologies. Quick respond to change, improved communication practices and team collaboration for iterative and incremental software development have made the popularity of Scrum in OSD industry to overcome issues and challenges. The geographical separations, time zone differences and cultural differences of offshore teams represent the challenges and issues for OSD and for practice Scrum.

The objective of this study is to identify OSD and related issues through a comprehensive literature survey. The literature survey is also used to identify the uses of Agile methodologies with off-shoring and its effectiveness to solve existing issues. Finally, it is expected to identify the effectiveness of using Scrum with OSD in Sri Lankan context through a statistical survey and provide recommendations.

The remainder of this paper is organized as follows. Section 2 presents the theoretical model. Section 3 discusses the research methodology and variable descriptions. Section 4 discusses the data analysis. The results of the study, recommendations and future research directions are presented in Section 5.

II. Theoretical Model

This section describes the OSD and reasons for using it. Also the issues are discussed and critical success factors of off-shoring are identified. Then the Agile methodologies and their effectiveness for off-shoring are discussed. Later, the Scrum methodology is discussed with its processes, roles, and contract types.

A. *Offshore Software Development (OSD)*

OSD is the process of developing software by a supplier who is from a different country from the onshore company who outsourced its development [1]. For example, a software company in USA could out-source some of its software development activities to a software supplier company in Sri Lanka. Here the Sri Lankan company is doing OSD to the onshore company in USA. There might be more than one offshore software supplier who works for a particular onshore company. Some of the software development activities are still at onshore site and some activities are done at offshore site. Major decision making is done by onshore site where origination of the project takes place and execution of high level plans are mainly done by offshore site.

In off-shoring, one software project is divided into different components and these components are developed by one or more supplier in different countries. The component module development is closely monitored and coordinated by onshore site. Finally, the software components developed at one or more offshore site are brought together and integrated into one product at onshore site as they have the bigger and holistic picture of the software product [3].

1) Moving towards OSD

Main reasons for companies to moving towards OSD are reduce cost of development with lower wages, skilled labor in off-shoring countries, 24 hour round the clock development and product support availability [18]. Further, it has been identified that factors, such as improvement in IT and telecommunications, favorable governmental policies, and less cost of setting offshore site, promote off-shoring [2].

2) Issues and challenges related to OSD

There are issues and challenges related to OSD such as poor communication, strategic issues and poor team collaboration [17]. More issues related to OSD are given below.

- *Communication issues:* Due to the geographical separation between offshore and onshore site, proper communication is challenging and it directly affects the software development process to be weaken. Time zone differences, lack of communication tools and technologies, poor telecommunication infrastructure, and poor organizational processes weaken the communication.
- *Team collaboration:* The social boundaries and physical distance of the teams have made it difficult to build good collaboration in OSD. The cultural, organizational and functional boundaries between offshore teams challenge the effective collaboration.
- *Process differences:* The probability of occurrence of issues related to the software development processes across multiple sites is high. There can be problems in process synchronization and system integration due to a one project is divided in to multiple phases and also modules and developed in multiple sites globally.
- *Knowledge management:* Since offshore teams are distributed across the globe, information and knowledge should be shared and available for everyone to work on common set of goals. Poor mechanisms of information and knowledge sharing, such as poor documentation of information, can weak the OSD process.
- *Cultural differences:* In off-shoring, there are people from different regions and countries with different languages, religions, and work practices working together in the process of software development. Hence there are lot of cultural differences can be found in OSD and people from different cultures have different behaviors and norms.

3) Critical success factors for OSD

Sudhakar [19] has done a research to identify the critical success factors for OSD. This research has identified six most important critical success factors based on literature review and number of citations in literature. Follows the six most critical success factors found.

- Trust in global teams
- Efficient communication
- Cultural understanding
- Relationship between client and vendor
- Contract type
- Efficient knowledge transfer

B. Agile Methodologies For OSD

Agile methodologies are used for cater dynamic behavior and drastic changes of software development. The Agile manifesto contains some important points, such as rapid customer satisfaction, continuous software delivery, welcome the late requirement changes, close and daily cooperation between project stakeholders, and face to face communication, to cater the dynamic changes in software development [17].

Agile based software development can be used to overcome the issues and challenges faced with plan driven software development methodologies, such as waterfall method, in dynamically changing software development environment [19]. The close collaboration between customer and developers is used in Agile to deliver the software product on time and under the budgetary constraints.

The benefits of connecting Agile methodologies with OSD as follows [17].

- *Communication:* Frequent and effective communication between teams in Agile benefits the OSD which needs proper communication.
- *Team collaboration:* Agile methodologies are promoting face to face communication and team collaboration to achieve common set of goals. This is supporting OSD, which is lack of collaboration between offshore teams.
- *Cultural awareness:* With Agile methodologies, back and forth team visits are encouraged to enhance the communication between team members. This mechanism also helps globally distributed offshore teams to get to know different cultures and avoid awkward situations during communication.
- *Process efficiency:* The continuous integration in Agile methodologies help offshore teams to integrate even a small change to the software product, which eliminate the integration issues at the end.

As stated above, Agile concepts are helpful to overcome the OSD issues and challenges. Paasivaara and Lassenius [16] have found some challenges of using Agile methodologies with OSD as follows.

- *Communication challenges:* The global distribution of teams limit the effective communication and it is required to find proper communication practices and suitable media for support Agile methodologies.
- *Integration issues:* The different tools and technologies used for integrations and version controls by offshore teams, make system integration more challenging in off-shoring.
- *Customer interaction issues:* Since the customer is located in another country, it is difficult to have frequent communication between customer and developers.

C. *What Is Scrum?*

Scrum for software development was introduced by Ken Schwaber in year 1995 to overcome issues in traditional software development processes which follow planning, estimation and development phases to build software, ultimately finish with project failures. However Scrum methodology follows the principles of Agile software development to adhere dynamic behavior of software development cycle. Scrum uses a flexible way to develop software, because the development phases cannot be predicted early. Scrum can be defined as an “iterative” and “incremental” software development method as well as an “adaptive” software development method [19]. The iterative and incremental development of software and team adaptation to the development life cycle by solving various problems, such as technical issues, requirement issues, and knowledge issues, prove the above definition for Scrum.

1) Scrum roles

There are three roles exist in Scrum such as Product owner, Scrum master and development team. Together, these three roles are known as Scrum team. Product owner is a single person who owns the product backlog. He/she is responsible for success of the project officially. Product owner commitment is required for maximizing the return on investment by identifying product features, creating a prioritized list of product backlog, selecting top priorities for the next sprint, and continually re-prioritizing and refining the product backlog [17].

Scrum master is the person who leads Scrum meetings, creates development sprint backlog and measures the progress of the product. Scrum master ensures the delivery of progress by everyone, records the decisions taken in meetings, tracks the actions need to be taken, keeps the scrum meetings short and focused. Scrum master is responsible for reduce product risk through incremental delivery of features, resolve development issues, and track the delivery of backlog items. The issues and challenges faced by development team are resolved by the Scrum master and guides team to be adhered with Scrum practices [18].

Scrum development team is responsible for build the product according to the product owner requirements. The development team is cross functional for deliver a shippable product in each sprint with experts from each area in software development. Also, the development team is self-organizing with a high degree of autonomy and accountability. While the team develops the product, new ideas are provided to the product owner to make the product great [19].

2) Scrum process

The project architecture has to be developed by the team with chief architect of the project. This initial architecture can be changed during the development of the project; however it is required initially to make the plan. After initial planning phase, other phases in software development life cycle, such as design, implementation, and integration are completed through the *sprints* to deliver the product incrementally. In Scrum, there is no predefined process within a sprint as opposed to the traditional repeatable and defined process approach [19].

Tasks done by Scrum team are tracked trough backlog which is an important instrument in the Scrum process. Two kinds of backlogs are used with Scrum such as product backlog and development sprint backlog. The *product backlog* is used to list and prioritized all the tasks relevant to the product been developed. Before every sprint, the prioritized items in the product backlog are moved to the *development sprint backlog* by the product owner with the agreement of Scrum development team in sprint planning meeting. The development sprint backlog can be used to monitor the progress of each item and the progress of developers and development teams [17].

One sprint produces a visible, usable and deliverable product that implements incrementally one or more valuable functionality of the software. *Time-boxed development* is used with sprints by preserving the end date of the particular sprint. It is possible to reduce the functionality which is delivered during the sprint; however the delivery date cannot be changed [18]. During a sprint, the team holds *daily scrum meetings*. Through daily scrum meetings, the tasks completed in backlog since last scrum meeting, what issues or blockers found that need to be resolved, and the tasks which team works on till next daily scrum meeting are basically discussed by the team [19].

At the end of each sprint, the software product is incremented from previous build by development team on promised delivery date. The developments completed during last sprint are demonstrated by the team to product owner and customer in *sprint review meeting*. The *sprint retrospective* is held to inspect the past sprint and do plan for improvements to be enacted during the next sprint. Then, *sprint planning meeting* is held to plan the next sprint. The product backlog can be re-prioritized and new tasks can be added based on customer requirements during this meeting and tasks are added to development sprint backlog to work on next sprint. The development team is giving estimates for the tasks added to the development sprint backlog, based on their previous sprints experience. The project discontinuation decision can be taken at the end of the sprint by considering deliveries or issues which take the project to closure phase to complete the project [18].

3) *Scrum contracts*

With Agile methodologies, such as Scrum, the product development approach can also be provided to invite customer to development process from the start. The product development information can be contained an overview of Scrum, how Agile works, and information on Target-cost contracts which is recommended for Agile methodologies. If customer doesn't prefer the Agile contracts, the Fixed-price traditional contract type can be selected with Scrum [16]. Follows the two contract types used with Scrum.

- *Target-cost contract type:* Here, the project delivery date and estimated budget can be predefined, however the requirements will be implemented based on the priorities. Changes are allowed through the development process, the product is incremented iteratively, and the customer feedback is encouraged. If project finishes earlier, the cost benefit will be shared between customer and vendor. If project finishes late, the additional cost will also be shared between customer and vendor.
- *Fixed-price contract type:* Total price, time of delivery and project scope have to be defined up front to start the project contract. Software requirement specification (SRS) is created by vendor and it is sent to customer along with project plan and estimated cost to start contract. The requirements in SRS are used as the stories in product backlog and then customer prioritized stories are sent to sprint backlog which development team works on.

Scrum allows early termination of contracts which is called “Money for Nothing”, for customer by paying only 20% - 30% of remaining contract value to vendor at the end of any sprint [18]. If project continuing cost is higher than the additional value received from the project, customer can decide to early terminate the project. Also Scrum contracts support the Change for Free, which enables customer to make changes to project scope without incurring any additional cost. Here new features are added to backlog and items of equal scope are removed form backlog and contract. These Scrum contract features provide benefits for both customer and vendor with lot of flexibilities.

III. Methodology

This research contains a comprehensive literature survey on OSD, current issues and challenges of OSD, critical success factors of OSD, Agile software development, and Scrum methodology. The literature survey was studied from credible sources such as electronic databases, books, journals, conferences, and internet. Findings in literature survey were used for developing the conceptual framework by identifying dependent and independent variables. A statistical analysis was performed to evaluate the critical success factors of OSD and find the effectiveness of Scrum on those critical success factors by participating OSD companies in Sri Lanka.

1) *Variable descriptions*

Four independent variables were identified based on Scrum methodology analysis through theoretical model as follows.

- Communication factors
- People factors
- Process factors
- Organizational factors

Following Table I represents the above independent variables identified from literature survey with their references.

TABLE I. Independent Variables And References

Independent Variable	Decomposed Factor	Reference
Communication	Scrum meetings	[10], [11]
	Informal communication	[10]
People	Role of Scrum master	[10], [15]
	Role of Product owner	[9], [15]
	Scrum team values	[5], [10], [15]
Process	Iterative and time-boxed development	[9], [10], [11]
Organizational	Scrum contracts	[4], [14], [16]

The success of OSD was identified as the dependent variable and its six decomposed factors of were identified based on most critical success factors identified in theoretical model as follows.

- Trust in global teams
- Client vendor relationship
- Contract type
- Communication
- Cultural understanding
- Knowledge transfer

The following “null” hypotheses were derived using above four independent variables with the dependent variable identified.

- H0: There is no significant relationship between “communication factors” of Scrum and success of OSD in Sri Lankan context.
- H0: There is no significant relationship between “people factors” of Scrum and success of OSD in Sri Lankan context.
- H0: There is no significant relationship between “process factors” of Scrum and success of OSD in Sri Lankan context.

- H0: There is no significant relationship between "organizational factors" of Scrum and success of OSD in Sri Lankan context.

2) Data Collection

According to national ICT workforce survey done by Information and Communication Technology Agency of Sri Lanka in year 2013, there was about 21,973 workforce in offshore software industry in Sri Lanka in year 2014 and it was used as the target population size of this research [8]. The following (1) developed by Cochran for large populations to yield a representative sample population, was used to identify the sample size of this research [6].

$$N = Z^2 * S * (1-S) / E^2 \qquad (1)$$

where N is the necessary sample size, Z is the Z-score, S is the standard deviation, and E is the margin of error.

The 95% confident level was used for this research with the constant Z-score of 1.96. Standard deviation of 0.5 was assumed for this research by considering the large population of OSD workforce in Sri Lanka and their anonymity for on-line survey. An expected +/-10% error rate was used for this research by considering time and cost constraints for this research. When applying expected confidence level, expected margin of error and assumed standard deviation for sample size formula, 96 number of responses were required for this research.

The sample population was software professionals working in selected Sri Lankan software development companies which involve OSD using Scrum. The company selection was based on its size and country of onshore site. Data collection was performed through an on-line questionnaire to evaluate the effectiveness of Scrum for OSD in Sri Lanka.

IV. Data Analysis

The on-line questionnaire consisted of demographic data, Scrum adherence level, independent variables and dependent variables related questions, was used to collect survey data from software professionals work in selected OSD companies in Sri Lanka. A descriptive analysis was used to analyze the demographic data and an inferential analysis was used to analyze the survey data and test hypotheses statistically.

A total of 119 responses were recorded and the reliability of survey dataset of all factors was tested using Cronbach's Alpha. According to the Cronbach's Alpha analysis, internal consistence of dataset of all factors was $\alpha = 0.89$ and found to be highly consistent.

A two-tailed Pearson correlation test was performed to test the four hypotheses and all four hypotheses have been accepted with 99% confidence interval. Hence the Scrum factors, such as communication, people, process and organizational factors, are positively effecting on OSD critical success factors, such as trust in global teams, client-vendor relationship, contract type, communication, cultural understanding, and knowledge transfer, in Sri Lankan context. Table II shows the summary of hypothesis test results.

TABLE II. Summary Of Hypothesis Test Results

Hypothesis	Correlation	Confidence level	Accepted hypothesis
Communication factors for OSD	Moderate	99%	Alternative
People factors for OSD	Strong	99%	Alternative
Process factors for OSD	Strong	99%	Alternative
Organizational factors for OSD	Moderate	99%	Alternative

V. Recommendations and Conclusion

A. Results

As per the hypothesis tests, key factors of Scrum have a positive relationship with OSD success in Sri Lankan context.

The first hypothesis test identified that communication factors of Scrum significantly relate to OSD success in Sri Lankan context. Scrum communication practices, such as daily Scrum meetings, iteration planning, retrospective meetings, product demos and informal communication methods, have effected positively on critical success factors of OSD.

The second hypothesis test identified that people factors of Scrum significantly relate to OSD success in Sri Lankan context. Role of Scrum master and related duties, role of product owner and his involvement with product development, and Scrum team values have effected positively on critical success factors of OSD.

The third hypothesis test identified that process factors of Scrum significantly relate to OSD success in Sri Lankan context. Scrum processes, such as iterative and time-boxed development and continuous integration, have affected positively on critical success factors of OSD.

The fourth hypothesis test identified that organizational factors of Scrum significantly relate to OSD success in Sri Lankan context. Scrum contract types with more contract flexibilities and cost benefits have affected positively on critical success factors of OSD such as offshore contract type.

B. Recommendations

The following improvements for Scrum practices with OSD have been identified.

- Use appropriate documentation mechanism and proper knowledge transfer mechanism to improve knowledge sharing between global sites.
- Use Scrum features, such as product backlogs, product demos, and burnt down charts, effectively to maximize benefits of Scrum and improve visibility of product development and build confident level of project stakeholders.

- Make arrangements to have frequent and timely communication with global teams to maintain a proper communication.
- Involve product owner with Scrum meetings as much as possible for get product related clarifications, proper sprint planning, and review progress to make offshore project successful with Scrum.
- Scrum methodology is lack of engineering practices in SDLC such as coding standards, continuous integration, automated integration and test automation. Therefore the engineering practices can be used effectively with Scrum processes and improve iterative and incremental software development for off-shoring.

C. Future Work

This research study can be further extended and use to support future researches focus on Scrum practices with OSD. Some of the future directives are mentioned below.

- Research to identify the effectiveness of combining Scrum with other Agile methodologies, such as XP, TDD, and ATDD, to import engineering practices to Scrum, such as coding standards, simple design, continuous integration, and test automation, in OSD.
- Research to identify effectiveness of using Scrum for research and development projects (R&D) with challenges such as broader and loosely defined requirements, user stories with harder estimations, flexible and dynamic planning, dynamic project goals and more documentation.
- The same conceptual model could be applied and validated across different countries and different cultures where the off-shoring issues and challenges could be varied, to see whether this model can be generalized.

ACKNOWLEDGMENT

The authors gratefully acknowledge the research support received from the Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. They would also like to thank software development companies who provided their fullest support in data collection.

REFERENCES

[1] A. Gopal, K. Sivaramakrishnan, M. S. Krishnan, and T. Mukhopadhyay, “Contracts in offshore software development: An empirical analysis,” Management Science, vol. 49, no. 12, pp. 1671–1683, 2003.

[2] A. Gopal, T. Mukhopadhyay, and M. S. Krishnan, “The role of software processes and communication in offshore software development,” Communications of the ACM, vol. 45, no. 4, pp. 193–200, 2002.

[3] B. Nicholson and S. Sahay, “Embedded knowledge and offshore software development,” Information and organization, vol. 14, no. 4, pp. 329–365, 2004.

[4] C. Hedin and S. Ö. Birgisson, “Introducing the Agile Requirements Abstraction Model: requirements engineering in a Scrum environment,” Department of Computer Science, Lund University, 2008.

[5] “Core Scrum | What is Scrum? | Scrum Principles - Scrum Alliance,” [Online]. Available: https://www.scrumalliance.org/why-scrum/core-scrum-values-roles. [Accessed: 20-Oct-2014].

[6] G. D. Israel, “Determining sample size,” University of Florida Cooperative Extension Service, Institute of Food and Agriculture Sciences, EDIS, 1992.

[7] G. P. Sudhakar, “A Review of Critical Success Factors for Offshore Software Development Projects,” Organizacija, vol. 46, no. 6, pp. 282–296, 2013.

[8] ICTA, “National ICT Workforce Survey 2013,” Available: http://www.icta.lk/attachments/1425_ICT%20Workforce%20Survey%20Report%202013.pdf.pdf, 2013.

[9] K. Vlaanderen, S. Jansen, S. Brinkkemper, and E. Jaspers, “The agile requirements refinery: Applying SCRUM principles to software product management,” Information and Software Technology, vol. 53, no. 1, pp. 58–70, 2011.

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

[11] M. Beedle, M. Devos, Y. Sharon, K. Schwaber, and J. Sutherland, “SCRUM: An extension pattern language for hyperproductive software development,” Pattern Languages of Program Design, vol. 4, pp. 637–651, 1999.

[12] M. Paasivaara and C. Lassenius, “Could global software development benefit from agile methods?,” in International Conference on Global Software Engineering, 2006, pp. 109–113.

[13] M. Paasivaara, S. Durasiewicz, and C. Lassenius, “Using scrum in a globally distributed project: a case study,” Software Process: Improvement and Practice, vol. 13, no. 6, pp. 527–544, 2008.

[14] M. Usman, T. R. Soomro, and M. N. Brohi, “EMBEDDING PROJECT MANAGEMENT INTO XP, SCRUM AND RUP,” European Scientific Journal, vol. 10, no. 15, 2014.

[15] P. Deemer, G. Benefield, C. Larman, and B. Vodde, “The scrum primer,” Available: http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf, vol. 1285931497, 2010.

[16] R. B. Svensson, S. Ö. Birgisson, C. Hedin, and B. Regnell, “Agile Requirements Abstraction Model–Requirements Engineering in a Scrum Environment,” Department of Computer Science, Lund University, 2008.

[17] R. Phalnikar, V. S. Deshpande, and S. D. Joshi, “Applying agile principles for distributed software development,” in International Conference on Advanced Computer Control, 2009, pp. 535–539.

[18] R. Prikladnicki, J. L. Nicolas Audy, and R. Evaristo, “Global software development in practice lessons learned,” Software Process: Improvement and Practice, vol. 8, no. 4, pp. 267–281, 2003.

[19] S. Jalali and C. Wohlin, “Agile practices in global software engineering- A systematic map,” in 5th IEEE International Conference on Global Software Engineering (ICGSE), 2010, pp. 45–54.