

The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects

Abeer M. AlMutairi, M. Rizwan Jameel Qureshi

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: abeer_almotari@yahoo.com, anriz@hotmail.com

Abstract—Scrum of scrums is an approach used to scale the traditional Scrum methodology to fit for the development of complex and large projects. However, scaling the roles of scrum members brought new challenges especially in distributed and large software projects. This paper describes in details the roles of each scrum member in scrum of scrum to propose a solution to use a dedicated product owner for a team and inclusion of sub-backlog. The main goal of the proposed solution is to optimize the role of product owner for distributed large projects. The proposed changes will increase cohesiveness among scrum teams and it will also eliminate duplication of work. Survey is used as a research design to evaluate the proposed solution. The results are found encouraging supporting the proposed solution. It is anticipated that the proposed solution will help the software companies to scale Scrum methodology effectively for large and complex software projects.

Index Terms— Agile, Scrum, Scalability, Backlog, Large Projects.

I. INTRODUCTION

Scrum methodology was introduced to develop complex projects using agile principles. Unlike traditional Software engineering, Scrum methodology relies on collaborative work, team cohesiveness and experiences to rapidly develop high quality software following iterative approach. Many organizations are willing to take benefit of Scrum methodology because it increases productivity and reduces time in a seamless environment to change requirements to develop software as per business goals [1].

Scrum methodology is more like a framework containing processes and roles to apply. The main processes and role of Scrum are the Scrum Master, product backlog, Scrum teams, daily Scrum meetings, Sprint planning meeting, sprint, sprint review and sprint retrospective [1]. Product backlog is a document that is not necessary to be a formal document and it can be written on board. It contains all defined requirements of a

product. The functional requirements called features or user stories and the non-functional requirements are referred as stories. Sprint backlog is a set of user stories that must be accomplished in the next sprint (increment) [2].

Scrum members attend daily meeting to discuss progress, achievements and obstacles related to a sprint. Sprint review is an important process to release/reject a sprint and update product backlog. Scrum does not provide complete and detailed descriptions of the project's processes instead Scrum leaves it up to Scrum team to tailor the development processes as shown in fig. 1.

Scrum relies heavily on team structure and roles of team members. The number of members in the team is limited between 7 to 9 members in scrum methodology. Scrum assigns specific roles to Scrum Master (SM) and Product Owner (PO). SM helps team to achieve highest level of performance whereas PO guides team towards building the right product. Scrum team (ST) is responsible to deliver a working version after each spring. ST must be self-organized, cross-functional and capable to take decisions to divide tasks between members and develop a sprint.

PO plays key role to coordinate and act as mediator between business, customer needs and ST. PO priorities requirements to ensure that any change in requirements, which is requested by a customer, must be incorporated into working release regardless of development phase to achieve his satisfaction. SM is responsible to keep the processes running and he ensures that ST is doing its task in a suitable environment. SM is also responsible for organizing and facilitating the meetings during scrum.

Scrum of Scrums is an enhancement of Scrum methodology enabling Scrum to deal large projects with large team size. It allows clusters of Scrum teams to discuss their work. The use of Scrum of Scrums is even critical for large projects [1].

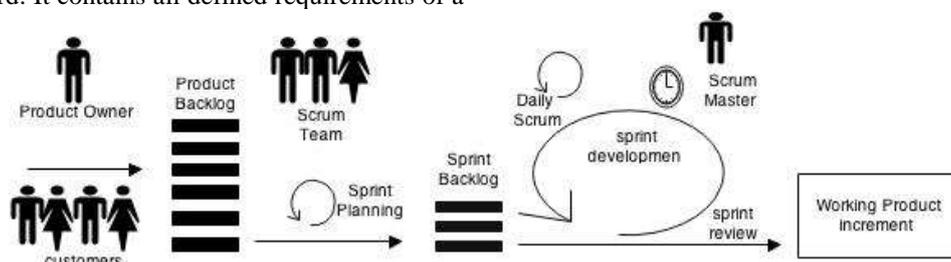


Fig. 1. Scrum Methodology

The rest of the paper is organized as follows. The related work is covered in Section II. Section III describes the problem statement. The proposed solution is illustrated in Section IV. Section V and VI discuss statistical analyze to validate the proposed solution.

II. RELATED WORK

The choice of a suitable software methodology is always a problem in software companies. Hamed and Abushama [1] discussed this issue in small to medium enterprises (SMEs). Most of software companies are seeking for rapid and agile way of development [1]. A systematic review is presented to show the appropriateness of agile methods with regard to SMEs environmental based on number of criteria [1].

Scrum is used to speed up the development of software projects in deferent area. A number of attempts are made to integrate Scrum with other development methodologies to take its advantages. Takahira et al. [2] proposed a framework to integrate Scrum and V methodologies to speed up the development of embedded software in automotive industry. Freitas et al. [3] integrate Scrum and Model Driven Development to develop an academic system [3].

Mokhtar et al. [4] adapted scrum methodology to develop an online integrated document management system to deal with academic quality assurance documents. The objectives of the system developed by Mokhtar et al. [4] are reduced redundancy and duplication of documents and time to search appropriate documents during the audit [4].

Major critical factors of Scrum are skills and experiences of scrum team [5-6]. Wangenheim et al. [5] and Scott et al. [6] agree to teach Scrum in educational courses. Wangenheim et al. [5] and Scott et al. [6] are of the opinion that learning Scrum plays an important factor while conducting projects to improve productivity, cohesiveness and coordination of teams. An educational (simulation) game is used to teach Scrum in an undergraduate program in order to improve the learning process of participating students [5]. A framework is presented to identify relationships between Scrum usage and learning style. The proposed approach is based on observing student's behavior according to the Felder-Silverman model while working on Virtual Scrum i.e., Scrum-based tool [6].

Product success significantly depends on the willingness of developers to adopt Scrum methodology [7]. Overhage et al. [7] propose a framework to support rapid delivery of high quality software in turbulent business environments. The proposed framework is based on the Technology Acceptance Model (TAM) [7]. Orłowski et al. [8] present hybrid fuzzy-ontological system to develop software process simulation modeling (SPSM) system. SPSM simulates the Scrum methodology including management processes and project roles [8].

Scrum testing is one of the responsibilities of Scrum team and many studies are aimed to find an efficient way to include best testing approach in Scrum. Parvez [9]

describes Scrum based approach to perform regression test while developing mobile application. A model is proposed to divide the sprints of Scrum into several phases and inject regression test in order to achieve an efficient process [9].

Shrivastava and Rathod [10] categorize set of risk factors that affect the performance of distributed agile development projects. There is a need identify the risk management methods to control risks related to distributed agile software development [10].

Scrum offers more management practices as compared to engineering practices. This deficiency of Scrum forces development teams to adopt engineering practices from other methodologies [11]. Khalane and Tanner [11] present theory building case study to uncover SQA concepts in Scrum environment. The study reveals that organizations need to decide how to customize development methods [11].

Vlietland and Vliet [12] adopt Scrum to empower IT development centers to deliver high quality software as per requirements of customers [12]. It is suggested to apply dedicated Scrum teams to deliver functionality swiftly [12]. Scrum facilitates to select appropriate scope of a release while considering constraints like budget, resources, dependencies and technical aspects [13]. Heikkilä et al. [13] claim that scrum can handle challenges by adopting release planning method for direct and efficient communication between the stakeholders for large-scale software development [13]. It is proposed to use Scrum for large projects by employing multiple small teams to work in parallel.

Qurashi and Qureshi [14] proposed a solution for managing large teams using Scrum Methodology. Several problems are discussed while applying scrum methodology using large team including duplication of work, communication failure, integration with other teams and dependencies between tasks [14]. A solution is proposed to overcome the problems of large teams by introducing Scrum of Scrum methodology. However, the proposed solution does not deal with situations when there are multiple product owners or Scrum masters [14]. The proposed solution also does not cover the overhead cost of backlog and story prioritization [14]. Paasivaara et al. [15] used two case studies to validate Scrum of Scrum implementation on two large globally distributed projects [15].

Project development progress is exhibited to changes due to variety of factors. However, in distributed projects, maintaining these changes by the team members become harder. This will lead to inaccurate work progress [16]. Alyahya et al. proposed a solution that supports the effective management of development progress by tightly integrating task or story progress data with the versioning data of development to keep progress consistency within team members. However, the implementation in real environment needs to be discussed and agile project management software need to be developed to help following up the work progress through single source by different members.

Identifying communication issues would help researchers to acknowledge different problems that could occur in agile global software development (AGSD) [17]. Kamaruddin et al. used literature survey to gather needed information. The authors discussed a variety of issues in GSD. These issues included different culture and language of the team members, lack of direct communication (face to face), different working hours/time zones, and unprepared communication Tools. Nevertheless, there are tools that can be suggested or developed to overcome many communication issues described previously. A web application for agile project management would enhance communication and knowledge transfer between team members.

Scrum methodologies should be tailored to meet the different needs of distributed projects [18]. Bass used a grounded theory approach to identify nine different roles of the product owner. These roles are groom, task prioritize, release master, technical architect, governor, communicator, traveler, intermediary and risk assessor. Tailoring Scrum methodology to the distributed projects is explored from the practitioners' descriptions to the previous roles. Bass focuses mainly on tailoring the product owner role in distributed environment. However, further investigation is needed to identify how to tailor other Scrum roles like development team and Scrum master.

Jalali et al. reviewed conference papers from 1999 to 2009 about the use of agile practices on the distributed agile software development [19]. The main purpose was to highlight which circumstances of agile methodologies, in which global software settings would be applied efficiently. The analysis and results revealed that agile practices and methodologies were mostly tailored to become applicable. Also, there was no systematic review that covers practicing all agile methods with all possible GSE settings. Further efforts are needed to provide detailed steps on how to tailor the agile methodology to the practitioners.

Estler et al. [20] have examined a total of 66 industry projects. The study compared agile (Scrum, XP, etc.) vs. structured (RUP, waterfall) processes to determine the

choice impact. They evaluated the correlation between process type and different measures like success, importance, team motivation, and real-time communication. The results have shown that selecting an agile rather than a structured process does not appear to be a crucial decision for globally distributed projects. However, researchers need to investigate more to identify which agile practices are common and followed in practice.

Noordeloos et al. [21] conducted a qualitative data analysis approach and found that moving from RUP to Scrum brings a positive effect in requirements engineering, communication, cost management and cross-functionality of the distributed teams. They also observed a negative change regarding the development pace and delivery time due to the lack of documentation and missing the tracing lines. As a future work, more investigations are needed on how developers communicate with each other in daily Scrum meetings and how they transfer the knowledge between each other.

Ghosh [22] listed the main challenges when implementing distributed Scrum on the projects. For example, communication factors like: distance, culture, time zone, and language.

Daneva et al. [23] used interviews as a data collection method and grounded theory for data analysis. The research results were the following:

- Requirements dependency is the key factor toward successful agile implementation in large distributed projects.
- Risk is the most important prioritization criterion in the setting of large distributed agile projects
- "Delivery stories", a new terminology added a good value to the agile development. It is the "user stories" with addition of technical implications, effort estimation and associated risk.

However, researchers did not provide a clear description of how vendors deal with agile requirements engineering in distributed and outsourcing contexts. Table 1 shows the limitations of related work in a summarized way.

Table 1. Limitations of the related work

Title	Limitations
Popular Agile Approaches in Software Development: Review and Analysis [1].	The paper only based on 7 case study of agile methodology. Need farther studies and practices experience to fully review.
Scrum and Embedded Software Development for the Automotive Industry [2].	No persistent practice and experimental study that support the proposed methodology on real product development project routine.
Applying Interdisciplinary and Agile Methods in the Development of an Embedded System [3].	The team faced issues like Impossibility of daily appointments; the physical distance between team members; and the need to follow the course schedule restrict using scrum and they failed to avoid such problems.
An integrated document management system for managing self programme accreditation using scrum approach [4].	The scope of the system is limited to manage document evidences for academic quality assurance system.
SCRUMIA—An educational game for teaching SCRUM in computing courses [5].	The validity and reliability results of the evaluation are restricted to courses in same university by the same instructor (limit generalizability)
Are learning styles useful indicators to discover how students use Scrum for the first time? [6].	Need to carry out more test and experiments with other project management tools and teaching contexts to analyze the framework.
What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance [7].	Not specified how to implement it in large projects where multiple interdependencies and higher requirements for the reporting exist.

Hybrid fuzzy-ontological project framework of a team work simulation system [8].	It is difficult to accurately evaluate its applicability before the entire SPSM design process is closed. No data was provided regarding the selection of team members.
An Efficient Model for Mobile Application Regression Test for Agile Scrum Software Development [9].	Only covers a small project developed in single environment with expert team
Categorization of risk factors for distributed agile projects [10].	18 out of 28 project work documents are not handed for collecting data for the study due to the confidentiality issues. Limitation in time and budget that restricted the scope of data collection for the study.
Software Quality Assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations [11].	add a level of complexity and overhead upon organization to go through a learning phase of additional methods, which is not be advisable for companies that are still trying to perfect their Scrum implementation. Required changing methods and learning a different method for every client project which can be risky and costly.
Towards a governance framework for chains of Scrum teams [12].	Most of the issues presented in the paper required validation and future works.
Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation [13].	Implementing the framework in large projects with interdependencies is not covered.
Scrum of Scrums Solution for Large Size Teams Using Scrum Methodology [14].	Focused on scaling the role of scrum teams in general and the meetings among them, leaving the other important factor in Scrum vague i.e. Product owner, Scrum Master, Backlog etc.
Inter-team Coordination in Large-Scale Globally Distributed Scrum: Do Scrum-of-Scrums Really Work? [15].	Only two case projects from two organizations, which limits the generalizability of the results. The data collected through interviewing teams without observing the actual SoS meetings in the organizations.

III. PROBLEM STATEMENT

There is a need of software industry to use such a process model that facilitates to deliver software with agility without compromising quality. Scrum methodology is one of the choices and it a hot area of research. However, large projects requires large teams incorporating several challenges and risks due to the team size and roles assign to each team member in Scrum [12-15]. This paper attempts to address the problems of team size and roles of scrum team members by scaling Scrum to make it suitable for the development of large distributed projects with large teams.

IV. THE PROPOSED SOLUTION

Scrum of Scrums is an approach used to develop large projects with multiple Scrum teams to complete a product. Scaling the roles in Scrum of Scrums is a critical task especially in a distributed environment. It demands extra efforts on planning in order to achieve success. Scrum of Scrums proposed to divide a large team into multiple small teams and each team works on different features of the same project. It is described to manage dependencies between tasks of different teams and handle daily and sprint meetings using Scrum of Scrums [14]. A novel solution is proposed to explain the roles of Scrum of Scrums key members i.e., Product Owner and Scrum Master to control product backlog for large distributed projects. Following goals are defined to address the problem that is taken up in this paper.

Goal 1- Optimize the role of PO in distributed large projects.

Goal 2- Ensure cohesiveness between scrum teams.

Goal 3- Eliminate duplication of works between scrum teams.

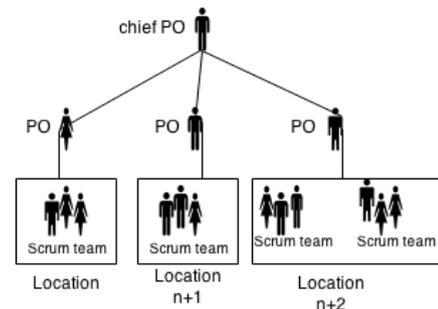


Fig. 2. The proposed team structure of Scrum of Scrums teams

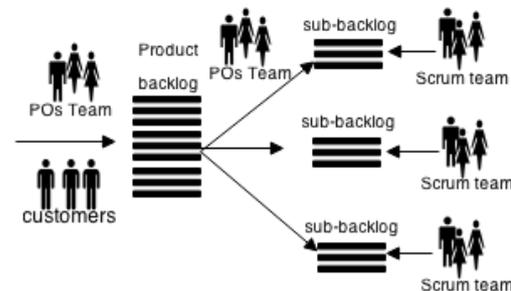


Fig. 3. The proposal of dividing a Backlog into Multiple Sub-backlogs

A. Product Owner

Product Owner is a key member in the Scrum team to interact with customers. The defined role of Product Owner is to prioritize backlog based on requirements of customers. It is proposed that PO will not only be responsible to prioritize the product backlog but also he/she will divide the product backlog into features to distribute these features among multiple teams while developing large distributed projects. It is proposed to use a dedicated local PO for each distributed Scrum team and a chief PO at the top of all local POs as shown in fig. 2.

Chief PO will interact with customers and collaborate with PO of each team to finalize product backlog. Chief PO is also responsible to decide the release date and accept, reject and modify release.

B. Scrum Master Team

Each Scrum team has a Scrum master. Scrum master is responsible to facilitate large project using a scrum of scrums approach. Scaling the role of Scrum master to make it sure that every team in Scrum are implementing Scrum in its real spirit. Scrum master is also responsible to coordinate among team. It is proposed that Scrum master ensures that stakeholders are fully utilized by adjusting the composition and frequency of scrum of scrum meetings.

C. Sub-backlog

Each PO and his team work with the costumer to construct a product backlog. Team must divide the backlog into features and address the dependencies between features. A product backlog is divided into several sub-backlogs. A sub-backlog is developed based on such features that are similar and dependent on each other. A sub-backlog is handled by an independent team. Each team has a Scrum master to lead. Each PO is responsible to priorities sub-backlogs with the respective teams as shown in fig. 3. Each scrum team works on the designated sub-backlog as it is a complete product along with the scrum master and PO. Both PO and scrum masters are responsible to ensure cohesiveness of team by addressing all conflicts among teams while conducting daily meetings. Each team integrates the sprint as it is ready. POs and their teams are responsible to integrate multiple sprints to deliver as a release. Chief PO is responsible to deploy release at customer site.

V. VALIDATION

The proposed solution is divided into three goals to validate it. Survey technique is used as a research design. A questionnaire is designed consisting of fifteen close ended questions. Likert scale is used ranging from one to five as shown in Table 2. The collected data is statistically analyzed to validate the proposed solution. Fifty three participants responded to the survey as follows.

Table 2. Likert scale used to evaluate the questionnaire

Very Low	Low	Nominal	High	Very High
1	2	3	4	5

A. Cumulative Analysis of Goal 1

Goal 1 has five questions and cumulative statistical analysis of data is shown in Table 3. Table 3 presents that 22 participants are agreed that the role of PO in the proposed solution is optimized. 9.8 software professionals are strongly agreed and whereas 4.6 respondents are disagreed and 15.2 software engineers are neither agreed nor disagreed. Only 1.4 participants are strongly disagreed the effect of the proposed solution. Fig. 4 depicts the cumulative analysis of goal 1.

Table 3. Cumulative Statistical Analysis of Goal 1

Q.No	Very Low	Low	Nominal	High	Very high
1	2.0	5.0	22.0	19.0	5.0
2	2.0	5.0	18.0	21.0	7.0
3	0.0	7.0	8.0	26.0	12.0
4	1.0	4.0	11.0	24.0	13.0
5	2.0	2.0	17.0	20.0	12.0
Total	7.0	23.0	76.0	110.0	49.0
Avg.	1.4	4.6	15.2	22.0	9.8

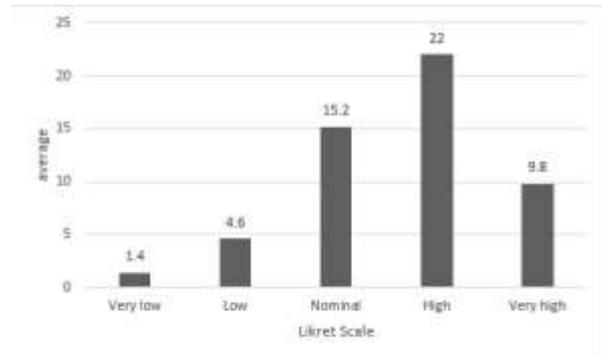


Fig. 4. Cumulative statistical analysis of goal 1

B. Cumulative Analysis of Goal 2

Goal 2 has five questions and cumulative analysis is shown in Table 4. Table 4 shows that 21.5 participants agree that the proposed solution will increase the cohesiveness among scrum teams. 14.2 respondents are highly agreed. 11.4 software professionals are neither agreed nor disagreed. 4.8 software engineers are disagreed and 1.2 participants are strongly disagreed. Fig. 5 shows the results of Table 4 graphically.

Table 4. Cumulative statistical analysis of goal 2

Q. No	very low	low	Nominal	High	Very high
6	2.0	5.0	14.0	12.0	20.0
7	1.0	8.0	10.0	22.0	12.0
8	0.0	3.0	12.0	21.0	17.0
9	3.0	3.0	8.0	25.0	13.0
10	0.0	5.0	13.0	26.0	9.0
Total	6.0	24.0	57.0	106.0	71.0
Avg.	1.2	4.8	11.4	21.5	14.2

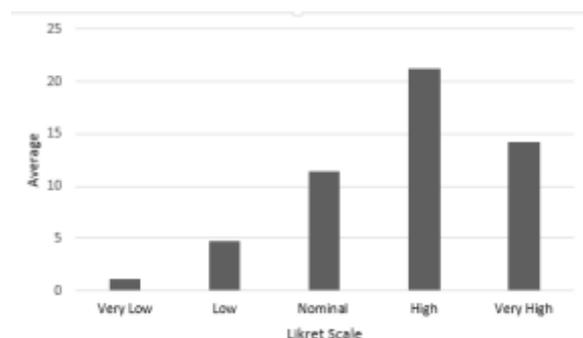


Fig. 5. Cumulative statistical analysis of goal 2

C. Cumulative Analysis of Goal 3

Table 5 shows the results of cumulative analysis of goal 3. 25 participants agree that duplication of work is disagreed and 0.8 participants are strongly disagreed. Fig. 6 shows the results of Table 5 graphically.

Table 5. Cumulative statistical analysis of goal 3

Q. No	very low	low	Nominal	High	Very high
1	1	3.0	9.0	34.0	20.0
2	2	9.0	8.0	22.0	16.0
3	0	4.0	13.0	19.0	17.0
4	0	4.0	2.0	30.0	17.0
5	1	4.0	17.0	20.0	11.0
Total	4.0	24.0	49.0	125.0	81.0
Avg.	0.8	4.8	9.8	25.0	16.2

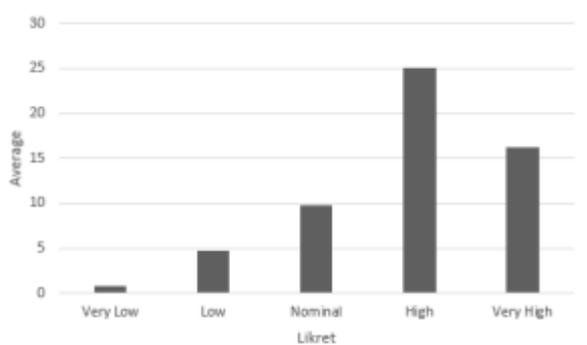


Fig. 6. Cumulative Analysis of Responses to Questions for Goal 3

D. Final Cumulative Analysis of All Goals

Table 6 shows the results of final cumulative analysis of the three goals. It is displayed in Table 6 that 22.8 participants agree that the proposed solution will increase the cohesiveness among scrum teams. 13.4 respondents are highly agreed. 12.1 software professionals are neither agreed nor disagreed. 4.7 software engineers are disagreed and 1.1 participants are strongly disagreed. Fig. 7 shows the results of Table 6 graphically.

Table 6. Final Cumulative analysis for all goals

Goal	very low	low	Nominal	High	Very high
1	1.4	4.6	15.2	22.0	9.8
2	1.2	4.8	11.4	21.5	14.2
3	0.8	4.8	9.8	25.0	16.2
Total	3.4	14.2	36.4	68.5	40.2
Avg.	1.1	4.7	12.1	22.8	13.4

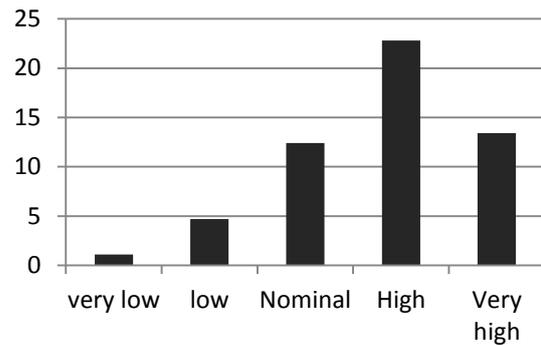


Fig. 7. Final Cumulative Analysis of 3 Goals

VI. CONCLUSION

The software industry requirement is to develop high quality products in a short time. Scrum is a good choice for software development companies to develop and deliver software in less time without compromising quality. Customers are demanding enterprise resource planning systems (ERP) to automate their processes seamlessly. ERP system falls under large projects. It requires large teams. Software companies are working with distributed locations and it is required to develop large projects utilizing distributed teams. Distributed teams face difficulties to coordinate and communicate due to different geographical locations. Scrum of scrums was introduced to overcome these difficulties to support distributed large teams using scrum methodology. There are many challenges associated to apply scrum of scrums like coordination and communication. These challenges are addressed in this paper by scaling the roles of scrum members in distributed large projects. The proposed solution is to use of dedicated/chief PO and local PO of each distributed team and divide the backlog into sub-backlogs. The proposed changes will increase coordination, communication and cohesiveness among large teams. A survey is used to evaluate the proposed solution. The results show that 36.2 out of 53 respondents are agreed with the effectiveness of the proposed solution. It is anticipated that the proposed solution will help the software companies to handle distributed projects with large teams effectively.

REFERENCES

- [1] A. M. M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," in Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, 2013, pp. 160-166.
- [2] R. Y. Takahira, L. R. Laraia, F. A. Dias, A. S. Yu, P. T. Nascimento, and A. S. Camargo, "Scrum and Embedded Software development for the automotive industry," in Management of Engineering & Technology (PICMET), 2014 Portland International Conference on, 2014, pp. 2664-2672.
- [3] C. R. de Freitas et al., "Applying Interdisciplinarity and Agile Methods in the Development of an Embedded

- System," in International Conference on Information Technology, 2014 © IEEE. doi: 10.1109
- [4] R. Mokhtar, N. H. Jaafar, N. F. Tahar, S. A. Sukiman, A. Aris, and N. F. Abu Bakar, "An integrated document management system for managing self programme accreditation using scrum approach," in Technology Management and Emerging Technologies (ISTMET), 2014 International Symposium on, 2014, pp. 102-106.
- [5] C. G. von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA—An educational game for teaching SCRUM in computing courses," *Journal of Systems and Software*, vol. 86, pp. 2675-2687, 2013
- [6] E. Scott, G. Rodríguez, Á. Soria, and M. Campo, "Are learning styles useful indicators to discover how students use Scrum for the first time?," *Computers in Human Behavior*, vol. 36, pp. 56-64, 2014.
- [7] S. Overhage, S. Schlauderer, D. Birkmeier, and J. Miller, "What makes IT personnel adopt scrum? A framework of drivers and inhibitors to developer acceptance," in System Sciences (HICSS), 2011 44th Hawaii International Conference on, 2011, pp. 1-10.
- [8] C. Orłowski, I. Bach-Dąbrowska, P. Kapłański, and W. Wysocki, "Hybrid Fuzzy-ontological Project Framework of a Team Work Simulation System," *Embedded System," in International Conference on Information Technology, Procedia Computer Science*, vol. 35, pp. 1175-1184, 2014.
- [9] A. W. M. M. Parvez, "An Efficient Model for Mobile Application Regression Test for Agile Scrum Software Development," *Advances in Computer Science and its Applications*, vol. 2, pp. 339-344, 2012.
- [10] S. V. Shrivastava and U. Rathod, "Categorization of risk factors for distributed agile projects," *Information and Software Technology*, 2014.
- [11] T. Khalane and M. Tanner, "Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations," in *Adaptive Science and Technology (ICAST), 2013 International Conference on*, 2013, pp. 1-6.
- [12] J. Vlietland and H. van Vliet, "Towards a governance framework for chains of Scrum teams," *Information and Software Technology*, 2014.
- [13] V. T. Heikkilä, M. Paasivaara, K. Rautiainen, C. Lassenius, T. Toivola, and J. Järvinen, "Operational release planning in large-scale scrum with multiple stakeholders—A longitudinal case study at F-Secure corporation," *Information and Software Technology*, 2014.
- [14] S. A. Qurashi and M. Qureshi, "Scrum of scrums solution for large size teams using scrum methodology," *arXiv preprint arXiv:1408.6142*, 2014.
- [15] M. Paasivaara, C. Lassenius, and V. T. Heikkilä, "Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work?," in *Empirical Software Engineering and Measurement (ESEM), 2012 ACM-IEEE International Symposium on*, 2012, pp. 235-238.
- [16] S. Alyahya, W. Ivins and W. Gray, "Co-ordination Support for Managing Progress of Distributed Agile Projects," in *Global Software Engineering Workshop (ICGSEW)*, Helsinki, 2011.
- [17] N. Kamaruddin, N. Arshad and A. Mohamed, "Chaos issues on communication in Agile Global Software Development," in *Business Engineering and Industrial Applications Colloquium (BEIAC)*, Kuala Lumpur, 2012.
- [18] J. Bass, "Agile Method Tailoring in Distributed Enterprises: Product Owner Teams," in *Global Software Engineering (ICGSE)*, Bari, 2013.
- [19] S. Jalali and C. Wohlin, "Agile Practices in Global Software Engineering - A Systematic Map," in *Global Software Engineering (ICGSE)*, Princeton, NJ, 2010.
- [20] H.-C. Estler, M. Nordio, C. Furia, B. Meyer and J. Schneider, "Agile vs. Structured Distributed Software Development: A Case Study," in *Global Software Engineering (ICGSE)*, Porto Alegre, 2012.
- [21] R. Noordeloos, C. Manteli and H. Van Vliet, "From RUP to Scrum in Global Software Development: A Case Study," in *Global Software Engineering (ICGSE)*, Porto Alegre, 2012.
- [22] G. Ghosh, "Challenges in Distributed Scrum," in *Global Software Engineering (ICGSE)*, Porto Alegre, 2012.
- [23] M. Daneva, E. V. D. Veen, C. Amrit, S. Ghaisas, K. Sikkel, R. Kumar, N. Ajmeri, U. Ramteerthkar and R. Wieringa, "Agile requirements prioritization in large-scale outsourced system projects: An empirical study," *Journal of Systems and Software*, vol. 86, no. 5, p. 1333-1353, 2013.

Authors' Profiles



Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree 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 the Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan in 2008.

Abeer B. ALMutairi is a master student in King Abdulaziz University, Jeddah, Saudi Arabia.

How to cite this paper: Abeer M. ALMutairi, M. Rizwan Jameel Qureshi, "The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects", *International Journal of Information Technology and Computer Science(IJITCS)*, vol.7, no.8, pp.68-74, 2015. DOI: 10.5815/ijitcs.2015.08.10