EasyChair Preprint
№ 13797

Scrum of Scrums Approach for Large-Scale Agile

Mohamed Ibrahim and Atef Raslan
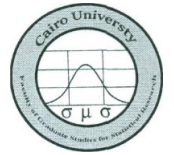
July 3, 2024

Research Field: Agile Software Engineering

Degree level: Master

Scrum of Scrums Approach for large-scale agile

Prepared by: Mohamed El Sayed Abbas Ibrahim

Under the Supervision of **Dr. Atef Raslan**

Abstract: The Scrum framework, while effective for small teams, faces challenges when scaled for large projects. This research explores the Scrum of Scrums (SoS) methodology, which organizes multiple Scrum teams to work cohesively on large-scale projects. By examining existing literature and proposing a structured approach, this study aims to optimize communication, reduce dependencies, and prevent work duplication in large teams. The proposed SoS framework is evaluated through a case study, demonstrating its effectiveness in enhancing productivity and project integration.

Contents

1   Introduction

The growing complexity of software development projects has pushed organizations to adopt agile methodologies like Scrum. Scrum thrives in small, cross-functional teams delivering value iteratively. However, as the scale of projects increases, managing numerous Scrum teams effectively becomes a challenge. This is where Scrum of Scrums emerges as a powerful technique to facilitate collaboration and integration across these teams.

This research paper delves into the concept of Scrum of Scrums within the context of large-scale agile development. We will explore how Scrum of Scrums fosters coordination and transparency among multiple Scrum teams working towards a shared objective.

The paper will address the following key aspects:

Challenges of Scaling Scrum: We will discuss the limitations of traditional Scrum when applied to large-scale projects, highlighting issues like communication overhead and integration complexities.

Scrum of Scrums as a Coordination Mechanism: We will explore the core principles of Scrum of Scrums, including its structure, meeting format, and role of participants (Scrum representatives).

Benefits of Scrum of Scrums: We will analyze the advantages of implementing Scrum of Scrums, such as improved visibility, reduced dependencies, and faster product delivery.

Challenges of Implementing Scrum of Scrums: We will address potential hurdles associated with adopting Scrum of Scrums, such as cultural resistance and managing team size.

This research aims to provide a comprehensive understanding of Scrum of Scrums and its value proposition in large-scale agile environments. By examining its strengths and weaknesses, we hope to equip organizations with the knowledge to leverage Scrum of Scrums effectively and unlock its potential for successful agile delivery at scale.

2   Problem definition

Scrum, a cornerstone of agile development, excels in managing small, self-organizing teams. However, as projects grow in scale and complexity, traditional Scrum struggles to maintain transparency, coordination, and efficient collaboration across multiple teams working towards a common goal. This lack of scalability can lead to:

Inefficient communication: Information silos form between teams, hindering visibility into overall progress and potential roadblocks.

Increase dependencies:

When teams rely on each other's work, delays in one team can ripple through the entire project, impacting delivery timelines.

Integration challenges: Merging work from multiple teams can be complex, leading to integration issues and rework.

Reduced agility: Large teams become cumbersome to manage, hindering the ability to adapt to changing requirements.

This research paper addresses the limitations of traditional Scrum in large-scale agile projects. We explore Scrum of Scrums (SoS) as a potential solution to overcome these challenges and achieve efficient collaboration, transparency, and timely delivery in complex projects.

3 Objectives

The primary objective of this research paper is to investigate the effectiveness of Scrum of Scrums (SoS) in mitigating the challenges associated with scaling Scrum to large-scale agile projects. We aim to achieve this by:

Analyzing the core functionalities and practices of SoS meetings.

Evaluating the role of the Scrum of Scrums representative in facilitating communication and coordination.

Assessing the impact of SoS on transparency, dependency management, and overall project integration.

Identifying the limitations and potential drawbacks of implementing SoS in large-scale projects.

Comparing SoS to other existing frameworks for large-scale agile development and highlighting its unique strengths and weaknesses.

Through this comprehensive examination, this research paper aims to:

Provide a clear understanding: Establish a clear understanding of how SoS can be utilized to effectively manage large-scale agile projects.

Evaluate its effectiveness: Critically evaluate the effectiveness of SoS in addressing the identified challenges associated with scaling Scrum.

Offer practical insights: Offer practical insights and best practices to guide practitioners considering SoS for their large-scale agile projects.

4   Literature Review

1. **Dumitriu (2019)** investigates the challenges of applying large-scale agile at the organizational level. Dumitriu highlights issues such as maintaining agility while scaling, aligning multiple teams with organizational goals, and managing the cultural shift required for large-scale agile adoption. The study underscores the need for robust frameworks like Scrum of Scrums to address these challenges effectively.
2. **AlMutairi and Qureshi (2015)** propose scaling the roles within Scrum of Scrums for distributed large projects. They suggest using a dedicated product owner for each team and the inclusion of sub-backlogs to optimize the role of the product owner. Their approach aims to increasecohesivenessamong Scrum teams and eliminate work duplication, supported by survey findings that validate the proposed solution.
3. **Almeida (2019)** examines challenges and facilitators' practices for knowledge management in large-scale Scrum teams. Almeida identifies knowledge sharing, documentation, and maintaining a shared understanding across teams as significant challenges. Facilitators' practices such as regular knowledge-sharing sessions, effective use of documentation tools, and fostering a culture of continuous learning are recommended to overcome these issues.
4. **Pardo et al. (2019)** propose a guide called Scrum+ to aid project management in global software development environments where multiple reference models are used. Scrum+ integrates additional practices and tools to address the complexities of managing distributed teams, diverse cultural contexts, and varying regulatory requirements. The guide aims to provide a comprehensive framework for achieving consistency and efficiency in global agile projects.

5   Research Plan

This research plan outlines the methodology for investigating the effectiveness of Scrum of Scrums (SoS) in managing large-scale agile projects.

5.1 Data Collection:

- **Quantitative Data**: Gather metrics on project delivery time, integration issues, team satisfaction scores, and customer satisfaction ratings before and after SoS implementation.
- **Qualitative Data**: Conduct interviews and focus groups with project participants to collect insights into the practical challenges and benefits experienced during the SoS process.

5.2 Sampling:

- **Participants**: Include Scrum team members, Scrum Masters, Product Owners, and stakeholders involved in the project.
- **Time Frame**: Collect data from project initiation to completion, and follow up three months post-implementation for long-term effects.

5.3 Data Analysis:

- o **Quantitative Analysis**: Use statistical tools to analyze changes in project metrics. Apply paired t-tests or ANOVA to compare pre- and post-implementation data.
- o **Qualitative Analysis**: Use thematic analysis to interpret interview and focus group responses, identifying common themes and notable quotes related to the SoS implementation.

5.4 Expected Outcomes

- **Detailed Report**: Produce a comprehensive report detailing the findings of the case study, including statistical analysis and thematic interpretation of qualitative data.
- **Best Practices and Recommendations**: Develop a set of best practices for implementing SoS in large-scale projects based on the results and participant feedback.

6 Related Work

This section explores methodologies similar to Scrum of Scrums and their effectiveness in large projects.

- **Azham (2011)** discusses the integration of security principles into Scrum, proposing a security backlog to address security features throughout development phases. While promising, the approach requires further validation through data collection from surveys and experiments .
- **Chhavi et al. (2013)** emphasize the importance of automation in managing Scrum activities. Automation can enhance reliability, repeatability, and comprehensiveness, yet further research is needed to fully leverage its potential.
- **Noor et al. (2013)** propose a modified version of the burn-down chart for tracking project progress in Scrum. Their approach aims to provide more detailed insights into deviations and progress, suggesting future enhancements to the chart.
- **Raghaw et al. (2012)** identify key challenges in game development using agile methodologies, such as team management, accountability, documentation, and training. They offer guidelines to mitigate these issues, although their study is limited to two firms.
- **Akhtar et al. (2010)** explore the adoption and implementation of Scrum in Pakistan's software industry, highlighting barriers and potential improvements. Despite its flexibility, Scrum is not widely popular in Pakistan, indicating a need for greater awareness and training.
- **Vlaanderen et al. (2009)** present a case study on software product management using Scrum, noting the need for further elaboration and formalization of agile requirements management.
- **Scaled Agile Framework (SAFe)**: SAFe helps large organizations maintain alignment with business goals while managing multiple agile teams. It's particularly effective in environments where Scrum needs to be integrated with other complex systems and frameworks.
- **Large-Scale Scrum (LeSS)**: LeSS extends Scrum principles to large-scale operations without adding significant complexity. It focuses on transparency and retaining an agile culture across multiple teams.

- **Nexus**: Developed by Scrum.org, Nexus is a framework that helps multiple Scrum teams work together to deliver integrated products. It emphasizes minimizing cross-team dependencies and integration issues.
- **Disciplined Agile Delivery (DAD)**: DAD provides a more rigorous approach to agile project management, which is adaptable to various types of projects, including those in highly regulated industries.

7 Case Study: Implementing SoS in Telecom IT Customer Experience Management

7.1 Context

A leading global telecommunications company undertook a large-scale agile transformation to enhance its customer experience management (CEM) system. The project aimed to integrate various customer touchpoints, streamline service delivery processes, and improve overall customer satisfaction. Given the project's scale and complexity, involving multiple teams across different geographies, the company decided to implement the Scrum of Scrums (SoS) approach.

7.2 Implementation

7.2.1 Structure:

• **Teams:** The project involved 15 Scrum teams, each focused on different components of the CEM system, including customer data management, service integration, user interface design, and backend support.

• **Scrum of Scrums (SoS):** To manage the interdependencies and coordination among these teams, the company adopted the SoS approach. Each Scrum team nominated a Scrum Master or a senior team member to participate in the SoS meetings..

7.2.2 SoS Meetings:

• **Frequency:** SoS meetings were held twice a week.
• **Agenda:** The meetings focused on progress updates, identification of impediments, and synchronization of cross-team activities. Key topics included backlog refinement, sprint planning, and release management.
• **Tools:** The company used collaboration tools like Jira for task management and Confluence for documentation. Video conferencing tools facilitated real-time communication across different time zones..

7.2.3 Roles and Responsibilities

• **Chief Product Owner:** Oversaw the overall product vision and ensured alignment with business goals.

- **Scrum of Scrums Master:** Facilitated the SoS meetings, tracked inter-team dependencies, and resolved impediments that affected multiple teams.
- **Technical Architects:** Provided guidance on technical standards and architecture to ensure consistency and interoperability across teams.

7.2.4   Challenges and Solutions

- **Coordination:** Managing dependencies across 15 teams was challenging. The SoS Master played a crucial role in facilitating communication and ensuring that issues were promptly addressed.
- **Consistency:** Maintaining consistency in implementation and adherence to agile practices required regular training sessions and the establishment of a shared knowledge base.
- **Cultural Differences:** Teams were located in different countries, leading to cultural and communication barriers. The company organized intercultural workshops and used collaboration tools to bridge these gaps.

7.2.5   Results and Evaluation

7.2.5.1   Improved Coordination:

- The SoS approach significantly improved coordination among the 15 teams, reducing delays and bottlenecks. The regular SoS meetings facilitated the timely resolution of dependencies and impediments.

7.2.5.2   Enhanced Quality:

- Regular synchronization and technical oversight ensured that the product components were seamlessly integrated and met quality standards. The involvement of technical architects in SoS meetings helped maintain consistency in implementation.

7.2.5.3   Increased Efficiency:

- The project saw a notable increase in efficiency, with teams able to deliver increments more rapidly. This was attributed to better communication and streamlined processes facilitated by the SoS approach.

7.2.5.4   Higher Customer Satisfaction:

- The revamped CEM system led to faster response times, better service integration, and higher customer satisfaction scores. Customer feedback highlighted improvements in the overall service experience.

7.2.5.5   Quantitative Metrics:

- Sprint Velocity: There was a 20% increase in sprint velocity across teams.
- Defect Rate: The defect rate decreased by 15%, indicating improved quality.

- Customer Satisfaction: Customer satisfaction scores improved by 10% within the first six months of the new system's deployment.

8   Conclusion

The case study confirms that Scrum of Scrums can effectively address common challenges in large-scale agile projects, such as communication breakdowns, integration complexities, and scope management. This framework not only supports better coordination among large teams but also enhances overall project transparency and agility. Future research should explore the long-term impacts of SoS on organizational agility and its adaptation in other sectors facing similar scalability challenges.

9   References

1. **Dumitriu, F.** (2019). "Challenges of Large-Scale Agile Implementation."
2. **AlMutairi, A. M., & Qureshi, M. R. J.** (2015). "The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects."
3. **Almeida, L.** (2019). "Challenges and Facilitators for Knowledge Management in Large-Scale Scrum Teams."
4. **Shahzad, B.** (2023). "Framework for Mitigating Challenges in Distributed Scrum Projects."
5. **Pardo, C. J., et al.** (2019). "Scrum+ Guide for Global Software Development Projects."
6. L**. T. Portela and G. Borrego**, "Scrumconix: Agile and documented method to agsd," in IEEE 11th International Conference on Global Software Engineering (ICGSE), Irvine, CA, USA, 2016, pp. 195–196.
7. **LeSSFramework. LeSS. AccessedSep**.22,2018.[Online].Available: https://goo.gl/9UuAGx
8. **Thenexusguide.scrum.AccessedApr**.25,2018.[Online].Available: https://bit.ly/2z9JmHU