

Chapter 19

Agile Scrum Issues at Large– Scale Distributed Projects: Scrum Project Development At Large

Ayesha Khalid


University of Lahore, Lahore, Punjab, Pakistan

Shariq Aziz Butt

 <https://orcid.org/0000-0002-5820-4028>

University of Lahore, Lahore, Punjab, Pakistan

Tauseef Jamal

 <https://orcid.org/0000-0003-4965-0322>

PIEAS University, Islamabad, Pakistan

Saikat Gochhait

*Symbiosis Institute of Digital and Telecom Management, Constituent of Symbiosis International
(Deemed University), Pune, India*

ABSTRACT

The agile model is a very vast and popular model in use in the software industry currently. It changes the way software is developed. It was introduced in 2001 to overcome deficiencies of software development in a workshop arranged by researchers and practitioners who were involved with the agile concept. They introduced the complete agile manifesto. The agile model has main components that make it more viable for use in well-organized software development. One of these is scrum methodology. The reason for the agile-scrum popularity is its use for small-scale projects, making small teams and allows change requests at any stage of a project from the client. It works for client satisfaction. Instead of so much popularity and distinctive features, agile-scrum also has some limitations when used for large scale projects development that makes it less efficient for development. This article discusses the agile-scrum methodology and its limitations when using for large-scale project organization.

DOI: 10.4018/978-1-6684-3702-5.ch019

1. INTRODUCTION

The Agile model is modern and vastly in use model in the software development. It was introduced in 2001 as a complete agile manifesto. The aim of introducing was to overcome the deficiencies from the efficient software development models and make the software development easier and more efficient. The agile model develops any software in iterations and allows change request at any stage of the project and at any iteration. It always gives priority to customer satisfaction and involves the customer in software development (Balaji & Murugaiyan, 2012; Ruparelia, 2010). In the agile model, the customer is directly involved in development and requirement elicitation. Agile model completes with its components such as Scrum, this is the most known and popular agile component. The agile model works with the small teams and use for the small size project development. In the agile model, scrum has a scrum master who arranges a review meeting session on a daily basis to manage the project and gets the project's update from developers. It takes the software development at an extreme level. Due to agile distinctive features, many software industries are moved toward agile development. Instead of these features agile model also has some limitations in software development (Plonka et al., 2014; Kaisti et al., 2014; Dingsøy & Moe, 2014). These limitations include its only use for small-scale project organizations, daily Scrum meeting sessions, Team communication, and customer involvement, less applicable for distributed development, Team Conflicts, less resourceful for complex and large systems (Dingsøy & Moe, 2014; Sekitoleko & Evbota, 2014; Lalsing & Kishnah, 2012; Sykov et al., 2018; Tiako, 2009).

The article is designed in the following sections, section 2 explains the literature of agile model including scrum, section 3 explains the limitations of agile model while using in large software development in large organizations, section 4 explains the research methodology of paper, section 5 explains the result discussion and suggestions and section 6 explains conclusion.

2. AGILE MODEL

The agile model is now very popular and vast in use SDLC (software development life cycle model). First, the term agile was used in the 1990's for the first time in many publish papers by different researchers and practitioners. In these papers, a new idea was discussed that people (researchers and practitioners) looking for new creative, efficient and attractive approach to develop the software application in a good manner. The Jim Smith and Bob Martin were involved in the agile concept and arranged a workshop in this regard. In the workshop, they arranged ideas about the agile with others who also involved in the agile concept. In the results of that workshop, the complete agile manifesto came up in 2001. Now the agile model is very trendy in use in many software industries. The agile model design to response the particular challenges of the software industry such as no physical deliverable, short development cycle etc. An agile model is an evolutionary approach that produces high-quality software products in a cost-effective and within time (Balaji & Murugaiyan, 2012; Vijayasathy & Turk, 2008; Moe et al., 2010; Butt, 2016). The popularity of the agile model between different software industries rather than other life cycle model is due to its unique features such it works with the small team members with 3-9 members to develop a particular software application. It gives many progressive results when using for small-scale projects in small organizations. The most owing feature of the agile model is to allow change request at any stage of the project and at any sprint/iteration. The study shows that usual times scale spends on the planning of sprints in agile software development. The agile model works in sprints as shown below in

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/agile-scrum-issues-at-large-scale-distributed-projects/294474

Related Content

Assigning the Test Case Priorities Using Butterfly Optimization Algorithm for Software Test

Nagaraj V. Dharwadkar, Srikant Shetgar, Manoj Patil and Abhijeet P. Shah (2022). *International Journal of Software Innovation* (pp. 1-15).

www.irma-international.org/article/assigning-the-test-case-priorities-using-butterfly-optimization-algorithm-for-software-test/303577

IDS Using Reinforcement Learning Automata for Preserving Security in Cloud Environment

Partha Ghosh, Meghna Bardhan, Nilabhra Roy Chowdhury and Santanu Phadikar (2017). *International Journal of Information System Modeling and Design* (pp. 21-37).

www.irma-international.org/article/ids-using-reinforcement-learning-automata-for-preserving-security-in-cloud-environment/205594

Using DRAM as Cache for Non-Volatile Main Memory Swapping

Hirota Kawata, Gaku Nakagawa and Shuichi Oikawa (2016). *International Journal of Software Innovation* (pp. 61-71).

www.irma-international.org/article/using-dram-as-cache-for-non-volatile-main-memory-swapping/144142

Hybrid Technique for Complexity Analysis for Java Code

Mohammad Subhi Al-Batah, Noh Alhindawi, Rami Malkawi and Ahmad Al Zuraiqi (2019). *International Journal of Software Innovation* (pp. 118-133).

www.irma-international.org/article/hybrid-technique-for-complexity-analysis-for-java-code/230927

Code Clone Detection and Analysis in Open Source Applications

Al-Fahim Mubarak-Ali, Shahida Sulaiman, Sharifah Mashita Syed-Mohamad and Zhenchang Xing (2014). *Handbook of Research on Emerging Advancements and Technologies in Software Engineering* (pp. 494-509).

www.irma-international.org/chapter/code-clone-detection-and-analysis-in-open-source-applications/108633