A comparison of agile RUP with scrum software methods for developing system of systems:C4I system

Yasmeen Alomair
King Saud University
5 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Abdullah Sharaf Alghamdi
King Saud University
121 PUBLICATIONS **708** CITATIONS
SEE PROFILE

A Comparison of Agile RUP With Scrum Software Methods For Developing System of Systems:C4I System

Mona AlZuhair, Yasmeen AlOmair, Abdullah Alghamdi, Iftikhar Ahmad

Department of Software Engineering, College of Computer and Information Sciences, King Saud University,P.O. Box 51178, Riyadh 11543, Saudi Arabia.

E-mail: mz3220@gmail.com, yasmeen2010_cjouf@hotmail.com, ghamdi@ksu.edu.sa, iftahmad@ksu.edu.sa

Abstract

System of systems architecture has been paid a remarkable attention by the Software Engineering community and widely adopted in the field of Software design. One common characteristic of system of systems software development is that the project is often distributed over multiple teams, buildings, or countries. In this paper, we reviewed the necessary literature regarding system of systems and agile methodologies. We explored the difficulties in applying such methodologies within large system of systems. We chose two agile development methods (Agile RUP and Scrum) and performed an objective comparison of them with respect to a number of features of the development process of system of systems. We utilized the characteristics of developing and managing system of systems along with the features and tools offered by the two agile methods and found that Agile RUP is a better method for developing system of systems.

Keywords: System of Systems (SoS), agile methodologies, Software Development, C4I, Scrum, Agile RUP.

1. Introduction

There has been a growing interest in a class of complex systems whose constituents are themselves complex. These systems are sometimes called system of systems (SoS) or federation of systems (FoS) [1]. System of systems SoS considered as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. Both individual systems and System of systems conform to the accepted definition of a system in that each consists of parts, relationships, and a whole [2].

The development process of system of systems SoS is considered a critical issue due to its heterogeneous nature with changing requirements and changing domain information. Multiple implementation platforms and tools are required and high concurrency occurs within SoS which makes the development process of SoS even critical. SoS is prone to hardware and software failures frequently which require fast agile treatment techniques [3].

Traditional software development methodologies have much strength including the emphasis on tightly defined objectives, the controllable process and the clear documentation and accountability. But they are more bureaucratic and less good when it comes to coping with fast changing external environments which impact on objectives and potential solutions . Many small-scale developers have shifted from a traditional, waterfall method for developing software to lighter weight, nimble and agile methods [4]. These methods are being named as agile software development methods.

The rest of the paper is organized as follows: The Section 2 describes the related work. Section 3 describes and analyzes agile RUP software method with respect to developing SoS. Section 4 describes and analyzes Scrum agile software method with respect to developing SoS. Section 5 describes the comparison of Agile RUP with Scrum Software methods for developing System of Systems and Section 6 concludes the paper.

2. Related work

Proper amount of Literature was studied and reviewed by the researchers for the purpose of discovering the up to date information regarding SoS, including SoS features, SoS development process, C4I systems as a real-world example of SoS. Agile software development methods including Agile RUP and Scrum and using them for developing SoS were also reviewed [5],[6].

C4I systems are real world examples of implementing a SoS. These systems had been used by various agencies like defense, police, emergency systems, investigation, airports, COE and oil and gas related department [8],[7],[9]. Factors like stability, reliability, resource utilization are of pivotal importance in such architectures. In C4I, edge cases and ambiguous conditions often present the main challenges when trying to define a complete and consistent rule-set or business logic to handle the diversity of conditions that occur at the boundary. Interoperability is also a large demand of C4I systems.

Common Operating Environment (COE) systems are real world example of C4I systems and therefore are example of SoS. COE designs innovative products for surveillance, transmission and management of video over IP, fiber and hybrid video networks . COE-based family of systems, has Succeeded in delivering compatible C4I capabilities to US and coalition forces. Tolk mentioned that applying a common framework within C4I community is not only technically feasible, but necessary to increase the efficiency of the Warfighter for future military operations. Within the C4I community, the Defense Information Systems Agency is working on solutions to ensure Information Superiority in Network Centric Warfare scenarios [10].

With regard to the development of SoS, Maier noted that the SoS architect must take into consideration the specification of communication standards within the SoS structure. The reason behind that is the independence and extent of super-systems that results in a greater emphasis on interface design and interaction facilitating rather than traditional system architecting [11].

Software development process is attached to and based on many different variables. Ken Schwaber said that the Software product releases are planned based on the many variables that form the initial plan and are subject to change during the project. These variables include: customer requirements, time pressure, the competition around, required quality, the system's vision and project resources (staff and funding available). A successful development methodology must take these variables and their evolutionary nature into account [12].

Kontio mentioned that the agility has become a hot topic in development processes and process development-related issues for many reasons include: Businesses need to react faster to market changes. IT departments want to deliver issues without the formal six-month release cycle and developers like a steady development environment where they can fully concentrate on the features and quality of the software they're building [13].

One of the limitations of the agile methodologies is the project size and how they handle larger teams. The key elements of the project size are project budget, duration and project team organization. The larger the team or more budget you need, the bigger the project is. Cockburn and Highsmith concluded that “Agile development is more difficult for larger teams, as size grows coordinating interfaces become a dominant issue”. In contrast, heavyweight methodologies support this by providing plans, documentation and processes for better communication and coordination across large groups [6].

Moving to the philosophy of agile development of RUP methodology is an urgent need, Liu and Roussev noted the reasons behind such urgent need. The Rational Unified Process (RUP) is a huge (over 3,200 files), prescriptive, and generic development method rather than concise, agile, and specific. Organizations moving to RUP are often unsure how much of it to adopt, and wary of how they can adapt it to their specific software projects [15].

Banking sector is well known for using large, sometimes monolithic, legacy systems. As for applying agile methodologies in the field of banking, Christou et al said that the safe choice for a development methodology to apply in banking systems is a process framework such as the Rational Unified Process (RUP). They adopted the use of Agile Unified Process (AUP) that combines RUP with agile methods to a successful project in the banking sector. The project achieved on-time delivery within budget, integrating heavy legacy application systems with newly reengineered client user-interface applications on service-oriented architecture (SOA) platform [16].

Cho introduced a new hybrid method, which combines RUP with Scrum methods to maximize the strengths of both conventional and agile methods, while trying to suppressing their weakness. The Rational Unified Process was used as a skeleton in the hybrid method while Scrum was embedded into the Rational Unified process to offer project management and tracking mechanisms [20].

3. Agile Unified Process (AUP)

Agile Unified Process (AUP) is a simplified version of the IBM Rational Unified Process (RUP) developed by Scott Ambler. It describes a simple approach for developing business application software using agile techniques and concepts yet still remaining true to the RUP. The AUP applies agile techniques including test driven development (TDD), Agile Modeling, agile change management, and database refactoring to improve productivity. The Agile Unified Process distinguishes between two types of iterations; a development release iteration results in a deployment to the quality-assurance and/or demo area and a production release iteration that results in a deployment to the production area. This is a significant refinement to the Rational Unified Process [20],[21].

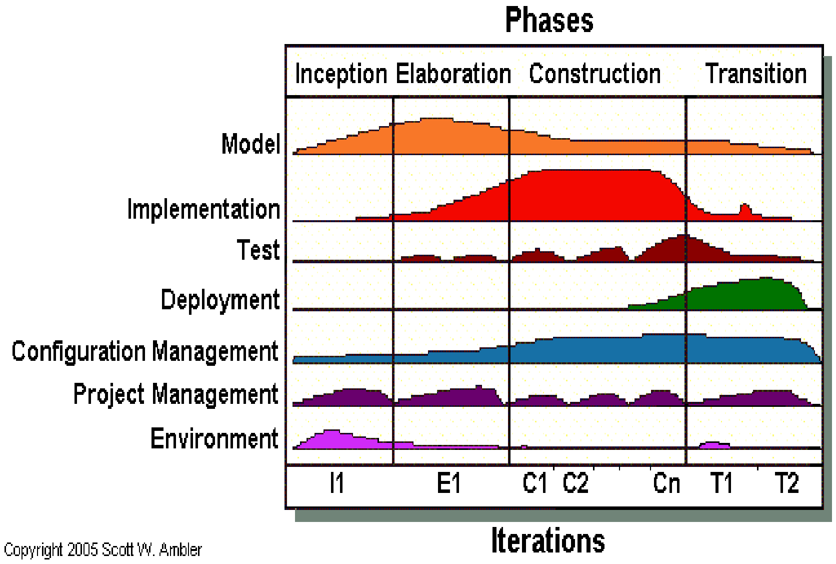

Figure 1. The Agile Unified Process (AUP) lifecycle

In general, RUP could be applied to projects of varying sizes. It is a flexible, scalable and fairly independent of type of software being developed. We can apply the Agile RUP to standalone applications, client/server applications and web applications with equal facility .

4. Scrum

Scrum is an iterative, incremental process for developing any product or managing any work [6]. Scrum is an agile, adaptive, quick and self-organizing product development process originating from Japan. The term derives form a strategy in the game of rugby where it denotes "getting an out-of-play back into the game" with teamwork [5]. SCRUM assumes that the systems development process is an unpredictable, complicated process that can only be roughly described as an overall progression [12]. In software development organization, chaos often results when a project's complexity is greater than its manger's ability. Progress in development might be made but it is often unsatisfactory. Scrum cuts through this complexity and wrests order from chaos by enabling a team to organize itself which allows a particularly productive order to emerge [20].

While RUP process can provide the method's platform: activities, roles, and artifacts. Scrum can offer the management and tracking mechanisms through these activities, roles, and artifacts [21]. In scrum, the changes happened to the system are not taken care of at the beginning of the SDLC, scrum controls them constantly and by consistently identifying deficiencies in the development process. A Scrum process is divided into three categories of Roles, Artifacts, and Time Boxes (phases). Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements.

Scrum introduces multiple roles within the development project: Scrum master, Product owner, Scrum team, Customer and management [5]. The table below describes the responsibilities of scrum roles:

Table1: Scrum roles and their responsibilities.

Role	Responsibility
Scrum master	- Ensures that the process is done according to the practices of Scrum. - Removes deficiencies, acts with team, customer and management.
Product owner	- Making visible of the product backlog list, turning issues into features to be developed and participating in the effort estimation process.
Scrum team	- Achieves the goals of each sprint, involved in effort estimation, create sprint backlog and review product backlog.
Customer	- Participate in the tasks related the product backlog items.

management	- Making the final decisions related to the charters, project standards and conventions, setting goals and requirement. - Measuring progress and reduces backlog with the master.

The Figure 2 demonstrates the main phases, artifacts and roles of Scrum process [14]:

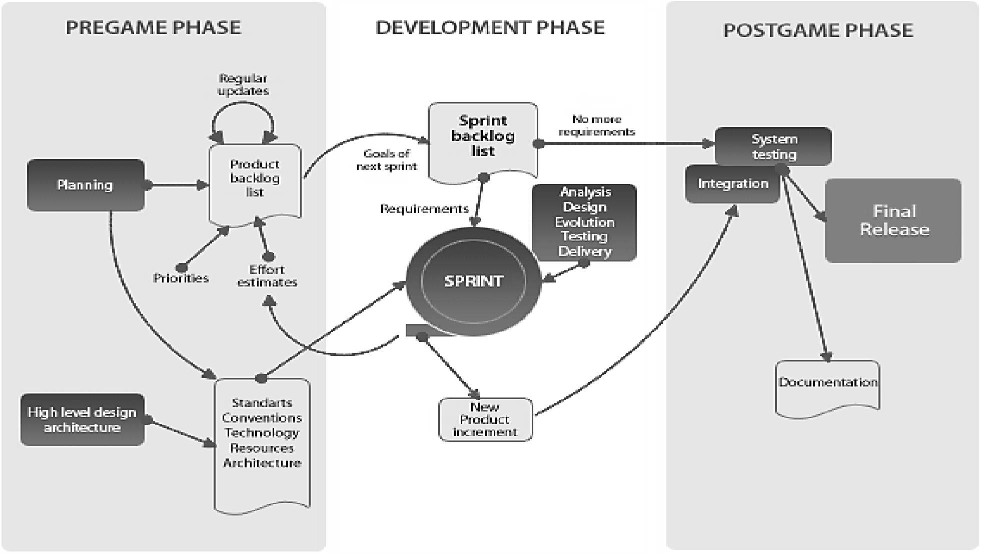

Figure 2. Scrum Process [14]

Schwaber mentioned many characteristics of scrum methodology that we utilized them for conducting the comparison of scrum and agile RUP in developing system of systems [17], [18]. Scrum method could be tailored by limiting the daily meeting to only 2 to 3 times per week; this could help achieving better results in some cases. Scrum changes job titles and descriptions in the SDLC. For example: scrum master corresponds "project manager" job but with different descriptions; Scrum Master does no longer need to organize the team but the team organizes itself and makes decisions on what to do [19].

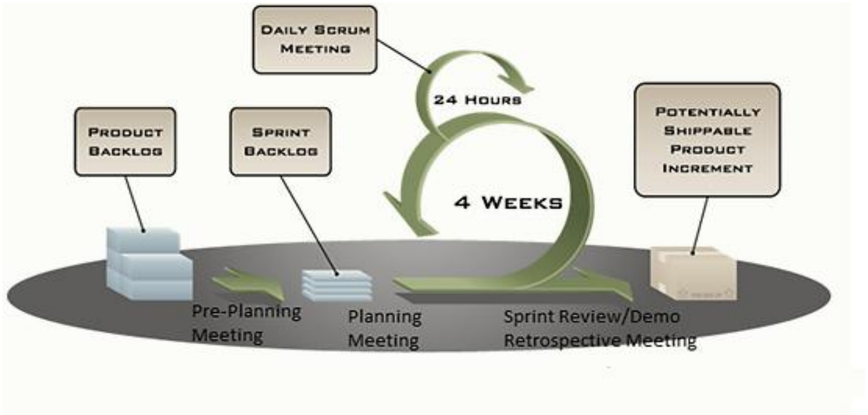

Figure 3. The Agile Based Development Model of Scrum [20]

5. Comparison of Agile RUP with Scrum methods for developing SoS

In this section, we are going to perform an objective comparison of the two agile development methods (Scrum and Agile RUP) with respect to the development process of system of systems. We set a number of ten characteristics of developing and managing SoS as the comparison criteria. Features and tools offered by scrum and agile RUP formed the basis for conducting this comparison:

This comparison shows that Agile RUP method is better than scrum method in the field of developing large system of systems for the following reasons:

1. Support for decomposition system into subsystems
2. Use collaboration tools and techniques
3. Manage dependencies, defines interfaces to support interoperability
4. Breaking the project into small and manageable tasks
5. Contain business engineering core process workflow to better understand the system
6. The concept of iterations helps discover risks quickly and reduced integration time and effort
7. Force extended teams during phases of lifecycles
8. Have architecture team to overlook and focus on the key concerns
9. Well-documented and a complete methodology
10. Have training readily available

6. Conclusion

As this field of SE is lately focused on the system of systems architecture and agile development practices, successful implementations and documentation of developing system of systems while using agile methodologies are scarce. With the distributed, heterogeneous and complex nature of SoS with large number of team members and stakeholders involved, the need for effective coordination grows with the project's complexity. The development process of such systems represents a big challenge for software engineers and domain experts. Since the development trend has shifted from the traditional long-term methods to a lighter weight and agile methods that focus intensely on the effectiveness, flexibility and maneuverability of the system delivered. However, scaling there agile methods into the development of system of systems is an interesting field to investigate. In this paper, we reviewed the necessary literature regarding system of systems and agile methodologies. We explored the difficulties in applying agile methodologies within large systems. An objective comparison of two agile development methods (Agile RUP and Scrum) was performed with respect to the development process of system of systems. We found that Agile RUP method is better for developing system of systems. As research continues and as more system of systems software development projects adopts agile methods, there should be detailed solutions to overcome issues seen in scaling to agile methods and better documented processes and practices. A suitable future work would be to consider the other features of SoS as criteria of the comparison between Scrum and Agile RUP methods.

Acknowledgement

The authors extends their appreciation to the Deanship of Scientific Reseach at King Saud University for funding the work through the research group project No. RGP-VPP-191.

References

[1] Mo Jamshidi (ed), System of Systems Engineering: Innovations for the Twenty-First Century, (May, 2008)

[2] KJ Baldwin, Systems Engineering Guide for Systems of Systems, Version 1.0, (August 2008).

[3] Project Lifecycle Services Ltd, AGILE - SCRUM Project Management (2007), http://www.projectlifecycleservicesltd.co.uk, Accessed on Dec 10, 2012.

[4] T Hafterson , Incorporating Agile Methods into the Development of Large-Scale Systems, University of Minnesota Morris CSci Senior Seminar Conference Morris.

[5] P. Abrahamsson, O. Salo, J .Ronkainen,and J Warsta, Agile software development, review and analysis, ESPOO 2002, VTT Publications 478.

[6] MA Awad, A Comparison between Agile and Traditional Software Development Methodologies. University of Western Australia, 2005

[7] J Highsmith, A Cockburn, Agile Software Development: The Business of Innovation, J Computer, (September 2001).

[8] A Alghamdi, Evaluating defense architecture frameworks for C4I system Using analytic Hierarchy process,J.Computer Scie. 5(12):1057-1081, 2009.

[9] M Chun NG, L Weng YEOH, Architecting C4I Systems, Second International Engineering Systems Symposium, 2009.

[10] A Tolk, A Common Framework for Military M&S and C4I Systems , Proceedings of the 2003 Spring Simulation Interoperability Workshop.

[11] M W Maier, Architecting Principles for Systems-of-Systems, Feb 1999

[12] K Schwaber,.Scrum development process, in OOPSLA Business Object Design and Implementation Workshop, J. Sutherland, et al., Editors. 1997, Springer: London.

[13] M Kontio , Architectural manifesto: Adopting agile development, Part 1 Is agile development right for your organization? , 8 Mar 2008.

[14] Development process, The SCRUM Phases, Accessed on Dec 21, 2012.

[15] M. Paasivaara, S. Durasiewicz, and C. Lassenius. Using scrum in distributed agile development: a multiple case study. In Proceedings – 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009, pages 195–204, 2009.

[16] I. Christou,,S. Ponis and E. Palaiologou , Using the Agile Unified Process in Banking , IEEE software , (2010), pp.72-79.

[17] M Ericsson, Developing Large-Scale Systems with the Rational Unified Process, 2000.

[18] J Garland, R Anthony, Large-Scale Software Architecture A Practical Guide using UML, 2003.

[19] N.A.Stanton Brunel University, G.H. Walker Brunel University, R.J. Houghton, The University of Birmingham and P. Salmon Brunel University , Generic Process Model of C4I Activities, January 2006**.**

[20] K Schwaber, Agile Project Management with Scrum , 2009

[21] Cho J Colorado State University, A Hybrid Software Development Method for Large-Scale Projects: Rational Unified Process with Scrum, 2009.

Mona AlZuhair ,
DSE, CCIS, King Saud University.
E-mail: *mz3220@gmail.com*