





+ Journal Menu

PDF



Article Sections

Special Issue

Scientific Programming Approaches to Deep Learning for Source Code Transformation

View this Special Issue

Research Article | Open Access

Volume 2021 | Article ID 6611407 | https://doi.org/10.1155/2021/6611407

Show citation

Analysis of Service-Oriented Architecture and Scrum Software Development Approach for IIoT

Yanqing Cui № 0,¹ Islam Zada,² Sara Shahzad,² Shah Nazir № 0,³ Shafi Ullah Khan,⁴ Naveed Hussain,³ and Muhammad Asshad⁵

Show more

Academic Editor: Sikandar Ali

Published: 23 Jan 2021

Abstract

Flexibility and change adoption are key attributes for service-oriented architecture (SOA) and agile software development processes. Although the notion of agility is quite visible on both sides, still the integration of the two diverse concepts (architectural framework and development process) should be well thought of before employing them for a software development project. For this purpose, this study is designed to analyze the two diverse software architectural framework and development approaches, that is, SOA and Scrum process model, respectively, and their integrated environment in software project development setup perspective for Industrial Internet of Things (IIoT). This study also analyzes commonalities among Scrum process model and SOA architectural framework to identify compatibility between Scrum and SOA so that the Scrum process can be constructively used for SOA based projects. This study also examines the proper design and setup of Scrum process suitable for large-scale SOA based projects. For this purpose, an SOA







+ Journal Menu

PDF



Article Sections

network accessible and executable [1]. Flexibility and change adoption are key attributes for SOA and agile software development processes. Although the notion of agility is quite visible on both sides, still the integration of the two diverse concepts (architectural framework and development process) should be well thought of before employing them for a software development project [2]. Therefore, the use of an appropriate agile process for the SOA based application development, to adopt major requirements modifications and changes even during application building along with the conservation of software superiority and quality, is essential [3]. Agile processes (based on agile manifesto [4]) tend to focus on iterations and client suggestions to improve performance and allows for the predictability of varying requirements. Agile software development (ASD) is the development process through which a system is developed efficiently and rapidly by means of regular, frequent, and complete releases permitting the participants and stakeholders to get their hands on the application [5, 6]. The application and process are reviewed and tested through agile retrospective meetings. In this way, a prototype is developed into useful progressive and creative system by means of an iterative and incremental process, whereby feedback is given by the stakeholders, based upon the rapid successive releases of the software.

Iteration stresses on the delivery of working software that provides grade and value to customers as well as to project. It also produces other concerned artifacts which are valuable to both customer and project. The aim of agile process is to provide workable deliverables in a dynamic way. Proponents believe that in an agile process the main concern is to provide deliverables and working products in a dynamic way without emphasizing on design models and documentation perspective [7, 8]. Scrum is one of the agile methodologies, which is a standard way of introducing agility due to its flexibility and straightforwardness [9]. It is a popular agile management method in industry. Agile development of applications in an enterprise surrounding can be challenging because of the compound nature of team members and their environment [10]. Scrum process facilitates discovering better ways of developing software by promoting individuals as well as teams [11]. Scrum divides the development process into iterations, called sprints, where a sprint stresses on the delivery of working product that provides value to both the project and customer [12]. Scrum, as the most used agile process, highlights empirical feedback, team self-management, and struggle to build properly tested product increments within short iterations [13]. The product owner, Scrum team, and Scrum master are the main three roles in Scrum. The responsibilities of the traditional project manager role are split among these three Scrum roles [14]. Scrum master is the one who maintains the process of the team, leads Scrum meetings, and makes sure that the process esteems all rules and principles of Scrum. Scrum team is a cross-functional team (having different expertise in different areas) involving developers, designers, testers, and analysts. This team is responsible for the project as a whole and is also involved in the development. Product owner is the one who represents the interest of end users and others interested in the product parties.

Scrum is based on regular meetings in which Scrum masters, product owner(s), the developers, and third parties discuss different issues concerning the development process. These meetings are the following: sprint planning meeting, daily Scrum meeting, sprint review meeting, and sprint retrospective meeting. Sprint planning meeting is held at the beginning of every sprint cycle. During this meeting, the product owner and







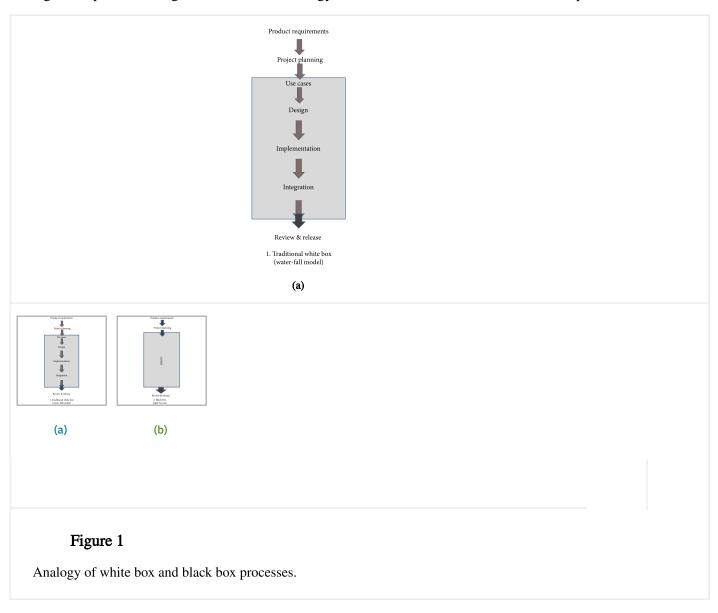
during the enrint. In enrint retrochective meeting teem members review the way the teem works and

Scientific Programming

+ Journal Menu PDF

Article Sections

overcoming any blockage in any phase or process during development. In ague terminology, the postmortem meeting known as retrospective is important for looking at the processes and creating knowledge base through best practices. Figure 1 shows the analogy of white box and black box nature of processes models.



Nonagile development methodologies are heavy-weight processes for developing software. These methodologies are based on a successive sequence of steps, for example, requirements explanation, solution structure, testing, and deployment. Traditional methodologies impose heavy documentation and define a fixed set of requirements at the start of a project. There are various traditional methodologies such as







monatitionare have also started to adopt acids approaches for these projects [16]

Scientific Programming

+ Journal Menu

PDF



Article Sections

irom scraten. This feature of reusability in SOA based applications maximizes economic benefits for the organizations [14]. Each service in SOA performs autonomously but is not isolated from the whole. Each service encapsulates a specific logic in the problem domain. The other features of SOA are loose coupling, service contract, autonomy, abstraction, discoverability, and statelessness [17].

Although SOA and agile approaches are generally viewed with related concerns, still there is no clear definition of organization and setup of both approaches in a single environment for IIoT. Truly little information is provided as to what will be the impact of this integrated implementation on the important factors such as productivity, quality, agility, and innovativeness.

Based on the above claims and current market needs, this study analyzes commonalities among Scrum process model and SOA architectural framework to identify compatibility between Scrum and SOA so that the Scrum process can be constructively used for SOA based projects. This study also examines appropriate design and setup of Scrum process suitable for large-scale SOA based projects. For this purpose, an SOA based research and development project is selected as a case study using Scrum as the software development process. The project development and deployment perspective includes eight core modules that constitute the overall project framework.

This study is organized in seven sections. The current section introduces major terminologies used in this study. The second section presents analysis of existing research in Scrum and SOA approach for an in-depth understanding. The third section is dedicated to studying design and setup, analysis of SOA and Scrum, and integration of Scrum and SOA. The fourth section is about the integration analysis; the fifth section is the discussion about SOA and Scrum integration, while the conclusion and future work of overall study are presented in the sixth and seventh sections, respectively.

2. Literature Study

Agile processes are planned to upkeep early and fast development of software applications. This is made possible by working in iterations [9]. Iterations stress on the delivery of working software that provides value to customers as well as to the project. An iteration is a short development cycle that produces working software and other artifacts which are valuable to both customer and the project [9]. Main concern of agile processes is to provide deliverables and working products in a dynamic way without emphasizing on design models and documentation perspective [7].

Experts of traditional development process conclude that when documentation and analysis process is neglected, it leads to corporate memory loss, especially when complex and large-scale software are developed. But the experts of agile confirm that, by following agility for software development global software delivery, it has led to better outcomes in gaining concerned objectives [18, 19]. It is clear that organizations will have to face some sort of challenges in coordinating and integrating agility and global





Ague software development methodologies are introduced to provide answers to the concerned business

Scientific Programming

+ Journal Menu

PDF



Article Sections

Scrum highlights empirical feedback, team self-management, and struggling to build thoroughly tested product increments within short iterations [23, 24]. Scrum is an agile methodology which is the most standard way of introducing agility due to its flexibility and straightforwardness and a popular management agile method in industry. Agile software development process facilitates better ways of developing software by promoting individual work as well as teamwork [11]. Agile processes are planned to maintain early and fast development of software application. This is made possible by dividing the development process into sprints, where sprint stresses on the delivery of working product that provides value to both the project and customer [25]. Agile development of applications in an enterprise surrounding can be challenging because of the compound nature team members and their environments [11].

SOA agrees to the integration of applications, users, and existing system into a flexible architecture that can easily accommodate changes when they are needed in a system [14]. SOA is regarded as one of the best approaches for distributed application development [26]. SOA allows reusing the functionality of existing systems rather than building again from scratch [27]. This feature of reusability in the SOA based applications maximizes economic benefits for the organization. Each service in SOA performs autonomously but is not isolated from the whole system. Each service encapsulates a specific logic in the problem domain. The main features of SOA are reusability, loose coupling, service contract, autonomy, abstraction, discoverability, and statelessness [28].

Researchers and professionals have a mixed opinion about similarity and compatibility of the two approaches. Critics emphasize differences among SOA and agile approaches, arguing that SOA and agile are standing at different development direction: SOA is architecture and agile is a methodology [24, 26]; SOA works in a top-down manner, while agile is inherently a bottom-up approach [29, 30]. SOA is an architectural framework and follows a set of principles, whereas agile is a process model and works at practice level [31]. Based on this view, advocates of SOA and agile integration suggest that one way of combining them is to use SOA framework and guiding principles of agile practices for service development [32, 33]. Some researchers also claim that SOA based systems are developed and deployed differently from traditional Systems [21]. Also, there are many challenges like stakeholders' involvement, business and IT alignment, and reuse of assets. To overcome such type of problems, agility and service orientation are better integrated [34, 35]. It is notable that Scrum and SOA share similar concerns, such as responsiveness to change, new ways of working, flexibility, and business understanding [36].

SOA and Scrum assessment and specification need attention of technology factors along with their relationships to IT related business organization [37] and to individual information consumers. Other requirements such as flexibility and ease of use are defined in terms of such type of system features that can be measured and designed [38, 39].







+ Journal Menu

PDF



Article Sections

development methodology for the development of an SOA based software system. The SOA based project is named M4S (Mineral resource, Mapping, Modeling, and Management System). The project development and deployment perspective includes eight core modules that constitute the overall project framework. The system is developed following the systematic steps which are discussed in detail in the following paragraphs.

Comme as a development appears and COA as an application from account deal with similar appearance

This study is evaluated through identifying, determining, and using one of the best combinations in software engineering, which is designed based on SOA, using highly productive open-source frameworks and tools, development through agile development methodologies, and deployment as a combination of client and cloud applications.

Scrum has been used as an agile process for continuous development and improvement to following changing requirement to refine working prototypes in iterations [42, 43]. The application is designed using SOA to ensure that the case study system modules can be used as independent subsystems through service interface. SOA facilitates exploiting the characteristics and capabilities of both Scrum and SOA for integration, customization, and binding of the desired designed frameworks and tools, the Mineral resource, Mapping, Modeling, and Management System (M4S); ultimately the deployment phase of this system has infrastructure having properties of interactive communication, scalability, and dependability of services.

These deployment requirements are achieved through cloud infrastructure and distributed computing. Modules of the developed application are designed to act as interoperable services and can be used independently and recombined in other modules resulting in an integration of multiple coherent modules. Data banks for mineral, resources, and application to manage those data banks are specifically designed for M4S system. These databanks, libraries, and applications are designed to be used as open-source utilities.

Scrum methodology is used to ensure having a transparent process of analysis and design of M4S and engagement of stakeholders and also ensures a dynamic interaction among the different components of the project. The use of agility and Open-Source System (OSS) principles for the development and project management has permitted targeting a diverse community of stakeholders through multilateral liaison of academia, government, and industry. A collaborative work of these diverse actors played a prime role in M4S development, along with a minor role of individual contributors.

As discussed in ICT R&D project named M4S [https://www.openm4s.org], the core framework of M4S utilizing open-source libraries, services, servers, applications, database management system, and an open-source Enterprise resource planning (ERP) is shown in Figure 2. Client-server architecture and browser-based clients are based on the principles of interoperability and open standards. The figure also shows components and information flow within proposed M4S.

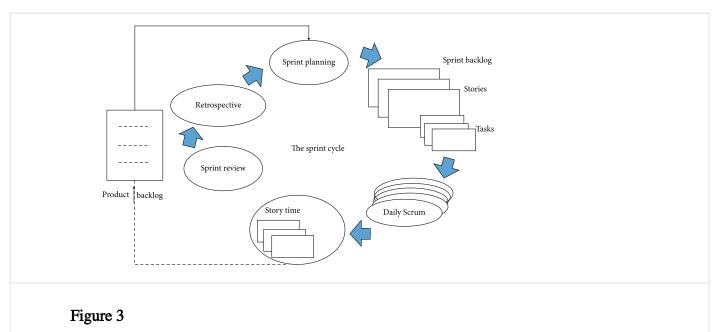


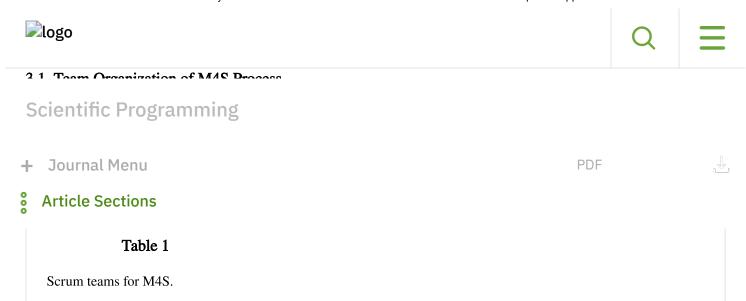
Figure 2

Proposed frameworks of M4S [https://www.openm4s.org].

Scrum-driven agile software development methodology is used for managing and organizing the development of modules [15]. Scrum used in the context of this case study along with specified roles and team organization is given in the following section: The process setup is decided according to the specification of agile Scrum. The following is a brief description of roles as these are present and work in a Scrum process environment.

- (a) Scrum master is the one who leads Scrum meetings, maintains the processes of team, and makes sure that the process respects all principles of Scrum [44, 45].
- (b) Scrum coach also facilitates meetings, helps the team to reach consensus, and takes care of quality aspects of the development environment and team.
- (c) Scrum team is a cross-functional team consisting of developers, testers, architects, and analysts. This team is responsible for project results as a whole and is fully involved in development. The team takes care of developing features according to quality and functional requirements [46].
- (d) Product owner is a person who represents the interests of end users and the stockholders. The product owner takes care of quality aspects related to user requirements. A typical sprint cycle is presented in Figure 3.





Key roles and responsibilities of each member in the project are as follows:

PD (project director) was responsible for planning, monitoring, and managing project activities.

Co-PD (coproject director) was responsible for monitoring project activities, establishing liaison, and coordination with government and industrial partners.

Software and computer systems experts were responsible for directing software architecture design and quality assurance.

Engineering management expert was responsible for directing project management activities and quality assurance.

Team leads were responsible for scheduling and execution of design, development, and deployment tasks.

Developer teams were responsible for development, testing, verification, and documentation of modules.

Coproject director was responsible for maintaining liaison and coordination with government and industrial partners.

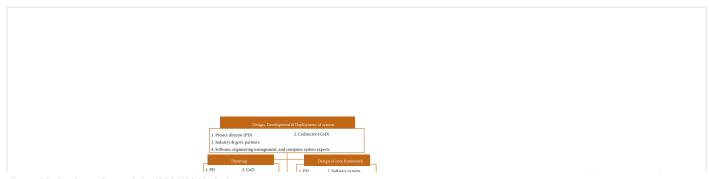
Mining and software experts (part time) were responsible for testing and validation of M4S and its modules.

Mentors (student supervisors) were responsible for supervision and guidance to UG and PG students, working on M4S related projects.

Part-time developers were responsible for development, testing, and verification of modules.

Internees were working on various features of M4S as their research projects and developed and documented the assigned projects.

Work breakdown structure (WBS), along with the responsible personnel for each block of activity, of M4S project is shown in Figure 4.









+ Journal Menu

PDF



Article Sections

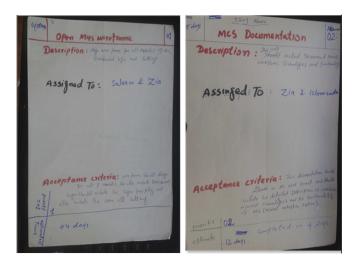


Figure 5

Story cards used in the project.

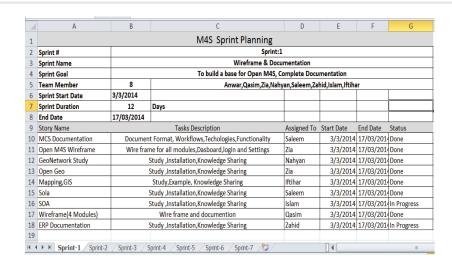


Figure 6

First sprint of M4S project designed in MS excel.

The following are the details of the Scrum process and its different features which are applied in the project.







starias accountance tast and is propored by the product owners (project director and addirector) and

Scientific Programming

+ Journal Menu

PDF



Article Sections

members are working on only one story card at a time. Each story or task is assigned priority according to its importance. Team members select stories on their priority base to finish. Figure 5 depicts two stories that are finished before the selection of next story.

(b) Sprint

According to Scrum process, development is organized into two- or four-week-long periods called sprints. A sprint is a time-boxed development iteration within which selected tasks are implemented [37]. Tasks are assigned to sprints during a sprint planning session at the beginning of each sprint based on the status of the backlog. A one-week sprint (also called sprint 0) is recommended before starting with the M4S project development. This allows the developers to get acquainted with the development environment, technological solutions, and the existing components that are reused. The first module of M4S is completed in seven sprints. The first sprint consists of total nine stories, in which seven stories were completed within sprint time box, while the remaining two stories were included in the second sprint to be completed next. Each sprint is designed in Microsoft Excel sheet, which contains all the information about each story, team members who worked on a particular story, time required for a particular story to be completed, and so forth. Figure 6 depicts the first sprint as a

Scrum methodology is based on regular meetings where product owner, Scrum team, third parties, and Scrum master discuss different issues faced/accrued during development time. The meetings are held for the purpose of ensuring smooth running of the project and preserving/maintaining the overall quality phases of the developed system. Scrum process model employs five types of meetings which are daily Scrum meeting, sprint retrospective meeting, sprint review meeting, sprint planning meeting, and sprint release meeting. These five meetings are also held and maintained during the M4S implementation in different situations. These meetings are discussed one by one in the following subsection.

3.1.1. Sprint Planning Meeting

sample as follows.

A total of seven sprint meetings have been held to plan the work to be finished in a sprint. Normally a meeting is held after every two weeks. During this meeting, Scrum team and the product owner (M4S project director & codirector) define the sprint goals. The achievement of the sprint is later evaluated in the sprint review meeting. Product owner works to prioritize the story cards, to describe the highest priority story points to the Scrum team. Finally, Scrum team needs to prepare the sprint plan and sprint backlog that detail the time it takes to do the work [48].

3.1.2. Daily Scrum







calculated at avant regular time. During this meeting each member of the team will ensure the succeeding

Scientific Programming

+ Journal Menu

PDF



Article Sections

the same place, at the same time, and for the same duration.

3.1.3. Sprint Review Meeting

This meeting is held after each sprint. In this meeting, the team members show what they have done and achieved during the completed sprint. Normally, this meeting takes place in the form of demonstration of the recently developed features. In M4S project, Scrum team has finished at least 95% of product backlog story points fetched to the sprint, and the team achieved most of the sprint goals. In each module of the M4S project, 98% of the features that meet the "done" criteria are accepted and marked as complete. The remaining 2% of the features that are not complete in the specified sprint are rescheduled for future sprints. The project director (product owner) is satisfied by the 98% completion; then the features are closed, and next backlog is updated to include the features that need further development.

- (a) The completed stories for each sprint are accepted by M4S project director and coproject director (product owners).
- (b) The finished features are accepted by project director during the meeting.
- (c) All the acceptance tests run for the stories in concerned sprint.
- (d) The whole code has been passed through code reviewing process.
- (e) No critical bugs are found in the bug backlog.
- (f) All finished story points for release are accepted.
- (g) The products are tested properly and have not found any serious bugs.
- (h) A successful backup is made.
- (i) Most of the deployment documents are up to date.

3.1.4. Sprint Retrospective

In this meeting, M4S team members reread the approach team workings, interrelations, behavioral features, and refining methodological skills, so that the ensuing sprint is more rapid, and so forth. Scrum team and team leader (Scrum master) discussed three things: what went well, what did not go well, and what improvements can be considered in the next sprint. The overall performance of the team is evaluated (in the form of metrics) and improvement strategies (in the form of KPIs) are identified, which are adjusted in our previous article. It is suggested that team members should have to share technical skills and knowledge, should keep in continuous communication with product owner, and so on. This meeting is held at the end of each sprint and lasts approximately two to three hours.

3.1.5. Release Planning







+ Journal Menu

PDF



Article Sections

- (b) Establish velocity, i.e., determine how many story points are completed in each sprint; the details are given in our other article.
- (c) Compute forecast, i.e., date-based release is estimated to complete (velocity × number of sprints) story points. Functionality-based releases are estimated to complete in (total story points ÷ velocity) sprints.

The highest priority stories, whose sum is no more than the number of story points (computed as mentioned above), are selected. For a functionality-based release, if the estimated completion date (computed above) is acceptable, all the stories are selected for the release.

4. SOA and Scrum Integration

This section analyzes the integration of Scrum, a software development process, and SOA, an architectural style. SOA introduces an organized and well-defined software design, where change is supported through principles of agility along with the application of design patterns and standards to achieve the quality, efficiency, and productivity [9]. Service reusability and abstraction are applied for designing flexible and adaptable systems [16]. SOA uses agility at design level as new services are designed and old ones are evolved on the underlying architecture and not by changing the architecture itself. Scrum allows development in increments, facilitating faster feedback among customer(s) and development team. Scrum and SOA both support the corporate process that promotes setting up aligned businesses and development strategies.

The analysis of Scrum and SOA process in M4S project development environment showed that Scrum and SOA use similar principles to reach the joint goal of required software development. Even though SOA runs in an organized and well-controlled setup, it can benefit from agile Scrum at the same time, without any unsuited overlapping. A clear strategy to introduce change in increments should, however, be designed.

Use of Scrum for traditional applications or conventional architectures is also possible by describing requirements into a backlog and incremental deliverables for the customers. Nevertheless, it is expected to get increased developmental efforts and overall project cost [49].

To get proposed benefits and maximum output through an agile process, it should be kept in mind that the underlying software architecture and IT organization environment also support the process in terms of responsiveness and costs. Otherwise, the organization may end up getting only minimal benefit from the agile process.

As already discussed, SOA and Scrum are approaches that follow different directions. In the services development scenario, SOA approach follows from top-down approach (services are building in the top of SOA system), while Scrum follows bottom-up approach (starting from initial planning to prototype delivery) as using a process development methodology. The following question arises: how are these different







This section refers to those questions and also discusses SOA and Somm matrice having commonclities. The

+ Journal Menu

PDF

Article Sections

Figure 7

Table 2 identifies most important metrics of SOA and Scrum, which can provide more value to business and to those whose aim is to use SOA and Scrum together in a software development venture.

Typical developmental framework of SOA and Scrum.

Table 2 Scrum and SOA common metrics.

Some of Scrum and SOA metrics are used for common purpose, which are integrated to be used for Scrum and SOA combination. Theses metrics are integrated form Scrum and SOA metrics which are used for the same purpose but using different terminologies. Through these metrics, development process can be streamlined when a proper measure is taken, which will provide more business agility, flexibility, and compatibility for SOA and Scrum environment.

The "completed stories vs. planned stories" metric of Scrum and "new service created and used as percentage of total service" metric of SOA measure the production of product using ratio and percentage as a scale device. They both are used for service or work measurements but using different terminologies. Also, the three metrics which are team velocity, development time, and average development time to develop a service measure the team progress in terms of sprint and time required for a service which is to be completed in a particular sprint within time. Due to common goals of these metrics, these could be combined into team velocity metrics which will be considered as a metric for SOA and Scrum integration. For quality measurement, Scrum and SOA used separate metrics, but the purpose was the same; therefore these two metrics can be used as quality assurance metric for measuring the services' quality using Scrum process model.







aammuniaatian hatusaan taam mamhara will alwaya ha nacitiya whan thay wark aagarky dua ta hanninaca

Scientific Programming

+ Journal Menu

PDF



Article Sections

collaboration, and feedback [13, 14]. As already discussed, Scrum and SOA are generally viewed with similar concerns, but still there exist some diversities and incompatibility issues between the two approaches. The compatibility and commonalities of these two approaches are discussed in order to make a ground for the integration of both development approaches.

5.1. Compatibility of Scrum and SOA

It may sound confusing as to why there is a need to find similarities and compatibilities among Scrum and SOA. Scrum and SOA are two different paradigms as one is a process and the other is an architectural style. Still, it is logical and relevant to find compatibilities of both when used together, for example, to develop an SOA based system using Scrum as a software development methodology, while most SOA teams are subconsciously aware of the way of design and development of services. The whole focus revolves around service design policies. SOA's nature inspires specific team makeup and style of communication within teams according to policies just like Scrum practices [15, 51]. It can be said that Scrum is like the human hands that work using gloves, while SOA is like that glove. It is understood that most of Scrum and SOA principles are not in conflict with each other because they both are adaptive approaches [2]. Meanwhile SOA with a traditional approach, for example, waterfall model, becomes predictive, as predictive methods completely depend on the requirement analysis and careful planning at the beginning of the cycle. Any change that is to be included will go through a strict change control management and prioritization.

The agile model uses an adaptive approach where there is no detailed planning and only clear future tasks are those related to the characteristics that must be developed. The team adapts to dynamic changes in the product requirements. The product is frequently tested, minimizing the risk of major faults in the future. Interaction with the clients is the strong point of agile methodology and open communication and minimal documentation are typical characteristics of the agile development environment. Teams collaborate closely and are often located in the same geographical space. Therefore, an agile model is better suited for the rapid development of adaptive services in SOA projects development. Application development through Scrum without a clear and strong idea of the aims of the organization will be useless. SOA without a clear image of how exactly to design and build using Scrum process model rules is a waste of resource and time. Also using SOA, as a pervasive strategy, for developing software application is increasing, since it focuses on the ability to respond to changes. Since adapting to changes is the indispensable concept of SOA development as well as agile development, it seems that using agile methodology is a natural fit to develop SOA applications.

Hence, Scrum and SOA are about agility that can be used together by applying a number of rules and principles, which are not in conflict with each other. This way they maintain each other in balance.

5.2. Scrum and SOA Commonalities







and than these convices are integrated in an application [40, 52]. Magnithile Commission exile process model

Scientific Programming

+ Journal Menu

PDF



Article Sections

Although Scrum and SOA are compatible, still there are some major distinctions among the two approaches, which should be understood and handled while working with SOA and Scrum together in a project [53]. One of the leading causes is that the two approaches have different origins and diverse directions. Scrum is traditionally small-project-based, although with process improvement and experience of practitioners have gain knowledge and learned experience to adapt the rules of the Scrum manifesto can also be applied to large software development projects. SOA is a top-down and divide-and-conquer approach to applications development. The "divide" part naturally results in low communication between teams. There are three main areas where Scrum and SOA clash with each other.

- (1) SOA encourages that architecture be designed upfront, while in Scrum community big design upfront (BDUF) is considered as an antipattern.
- (2) SOA encourages teams to split along functional lines, while Scrum encourages cross-functional teams.
- (3) SOA does not have any formal feedback and communication among development teams, while Scrum is focused on frequent feedback at both a technical level and personal level.

When using SOA in a large setup, it may be complicated to effect any change [54]. There may also be other issues beyond software agility and cost, for example, lack of team communication and discoverability of services. So instead we should strive for closer cooperation and openness among teams. Each team will make available an easy modifiable version of its service for internal test. The service team will separate test data from service, so it survives even if the code is messed up. The service team will provide clean and simple documentation on how to add new calls to the service stub.

6. Conclusion

Service-oriented architecture (SOA) allows reusing the functionality of existing systems rather than building again from scratch. This feature of reusability in the SOA based applications maximizes economic benefits for organizations. Scrum process model, on the other hand, tends to focus on iterations and client suggestions to improve performance and allow for the predictability of varying requirements. The study analyzes a basis for identifying commonalties and compatibilities in Scrum and SOA process to achieve maximum benefits of the organized Scrum management process for SOA based applications development. The study establishes that most of the Scrum and SOA principles are not in conflict with each other. Both Scrum and SOA are about agility which can be applied using rules and principles that do not clash with each other. To confirm this compatibility, the performance of an integrated Scrum and SOA development environment can be tested through formal KPIs based on the individual Scrum and SOA metrics commonly used by Scrum and SOA practitioners in the software development industry. These KPIs will provide a formal approach to measure







integrated project

Scientific Programming

+ Journal Menu

PDF



Article Sections

m a distributed chynomicht.

(2) To define a formal process for the evaluation of identified metrics and KPIs in nondistributed and distributed Scrum and SOA integrated projects.

Data Availability

No data are available.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- 1. N. Bieberstein, M. Fiammante, K. Jones, S. K. Bose, and R. Shah, *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*, FT Press, Indianapolis, IN, USA, 2006.
- 2. M. Lankhorst, *Agile Service Development: Combining Adaptive Methods and Flexible Solutions*, Springer Science & Business Media, Berlin, Germany, 2012.
- 3. F. Rago, "Self-organizing business networks, SOA and software maintenance," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, Hong Kong, China, March 2008.

View at: Google Scholar

4. A. Manifesto, "Agile manifesto," *Haettu*, vol. 14, p. 14, 2012.

View at: Google Scholar

5. R. Kumar, P. Maheshwary, and T. Malche, "Inside agile family software development methodologies," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 6, pp. 650–660, 2019.

View at: Publisher Site | Google Scholar

6. V. Jiménez, P. Afonso, and G. Fernandes, "Using agile project management in the design and implementation of activity-based costing systems," *Sustainability*, vol. 12, no. 24, p. 10352, 2020.

View at: Publisher Site | Google Scholar



C



1ccmology, vol. 73, no. 3, pp. 301-323, 2013.

Scientific Programming

+ Journal Menu

PDF



Article Sections

development," in *Proceedings of the 2019 IEEE Conference on Games (CoG)*, IEEE, London, UK, August 2019.

View at: Publisher Site | Google Scholar

10. P. Kutschera and S. Schäfer, "Applying agile methods in rapidly changing environments," in *Proceedings of the 1st RTO Symposium on Technology for Evolutionary Software Development*, Citeseer, Bonn, Germany, September 2002.

View at: Google Scholar

- 11. M. Awad, A Comparison between Agile and Traditional Software Development Methodologies, University of Western Australia, Perth, Australia, 2005.
- 12. W. M. Theunissen, A. Boake, and D. G. Kourie, "In search of the sweet spot: agile open collaborative corporate software development," *ACM International Conference Proceeding Series*, vol. 150, pp. 268–277, 2005.

View at: Google Scholar

- 13. S. W. Ambler, *The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments*, IBM Corporation, Armonk, NY, USA, 2009.
- 14. S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, IBM Press, Indianapolis, IN, USA, 2012.
- 15. A. Bhagwat, "Role of beacon architecture in mitigating enterprise architecture challenges of the public sector," in *Advances in Government Enterprise Architecture*, p. 502, IGI Global, Hershey, PA, USA, 2008.

View at: Publisher Site | Google Scholar

- 16. H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, John Wiley & Sons, Hoboken, NJ, USA, 2017.
- 17. D. Gabioud, E. László, G. Basso et al., *D3.2–Overall System Requirements and Functional Specifications*, 2015, Scalable Energy Management Infrastructure for Aggregation of Households (SEMIAH) project.
- 18. C. Verma, *The Effects of Organizational Culture on Risk Management during Software Development*, The British University in Dubai (BUiD), Dubai, UAE, 2009.



C



Software recimology, vol. 112, pp. 00-02, 2019.

Scientific Programming

+ Journal Menu

PDF



Article Sections

and analysis," 2017, http://arxiv.org/abs/17/09.08439.

View at: Google Scholar

- 22. M. Singh, N. Chauhan, and R. Popli, "A technique for transitioning of plan driven software development method to distributed agile software development," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020*, Delhi, India, February 2020. View at: Publisher Site | Google Scholar
- 23. M. Perkusich, L. Chaves e Silva, A. Costa et al., "Intelligent software engineering in the context of agile software development: a systematic literature review," *Information and Software Technology*, vol. 119, p. 106241, 2020.

View at: Publisher Site | Google Scholar

24. S. Moyo and E. Mnkandla, "A novel lightweight solo software development methodology with optimum security practices," *IEEE Access*, vol. 8, pp. 33735–33747, 2020.

View at: Publisher Site | Google Scholar

- 25. M. S. Nazare, *Evaluation of Agile Software Development Methodologies and its Applications*, California State Polytechnic University, Pomona, CA, USA, 2019.
- 26. O. Philipp, K. Stefan, and S. Eric, "Model-based resource analysis and synthesis of service-oriented automotive software architectures," in *Proceedings of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, IEEE, Munich, Germany, September 2019.

View at: Publisher Site | Google Scholar

27. O. Goubali, A. Idir, L. Poinel, L. Boulhic, D. Kesraoui, and A. Bignon, "Service-oriented control-command components for designing complex systems," in *Proceedings of the International Conference on Human-Computer Interaction. 2019*, Springer, Orlando, FL, USA, July 2019.

View at: Google Scholar

28. M. Rehan and G. A. Akyuz, "Enterprise application integration (EAI), service oriented architectures (SOA) and their relevance to e-supply chain formation," *African Journal of Business Management*, vol. 4, no. 13, pp. 2604–2614, 2010.

View at: Google Scholar



Q



view at. Fublisher ofte i Google ocholar

Scientific Programming

+ Journal Menu

PDF



Article Sections

View at: Publisher Site I Google Scholar

32. J. T. F. Chaves, "Service-oriented architecture (SOA), agile development methods and quality assurance (QA): a case study," University of Brasília, Brasilia, Brazil, 2019, Master's Thesis.

View at: Google Scholar

33. S. Abidi, M. Fakhri, M. Essafi, and H. Ben Ghazela, "A comprehensive framework for evaluating web services composition methods," *International Journal of Web Information Systems*, vol. 15, no. 3, pp. 324–345, 2019.

View at: Publisher Site | Google Scholar

34. M. Fischer, F. Imgrund, C. Janiesch, and A. Winkelmann, "Directions for future research on the integration of SOA, BPM, and BRM," *Business Process Management Journal*, vol. 25, no. 7, pp. 1491–1519, 2019.

View at: Publisher Site | Google Scholar

35. A. Plugge, S. Nikou, and H. Bouwman, "The revitalization of service orientation: a business services model," *Business Process Management Journal*, 2020.

View at: Publisher Site | Google Scholar

36. J. Paramanathan, "Security of lightweight-and heavyweight-IT in a high-tech hospital," University of Oslo, Oslo, Norway, 2019, Master Thesis.

View at: Google Scholar

37. M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using Scrum in distributed agile development: a multiple case study," in *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering*, IEEE, Limerick, Ireland, July 2009.

View at: Publisher Site | Google Scholar

38. M. Farsi, M. Badawy, N. Abdelmoneim, H. Arafat Ali, and Y. Abdulazeem, "QoS-aware framework for performance enhancement of SOA in enterprise IT environments," *IEEE Access*, vol. 7, pp. 107699–107715, 2019.

View at: Publisher Site | Google Scholar

39. J. Bloomberg, *The Agile Architecture Revolution: How Cloud Computing, Rest-Based SOA, and Mobile Computing Are Changing Enterprise IT*, John Wiley & Sons, Hoboken, NJ, USA, 2013.





IEEE/ACM Jul International Workshop on Conducting Emphrical Studies in Industry (CESI), IEEE,

Scientific Programming

+ Journal Menu

PDF



Article Sections

requirements engineering management," in *Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering*, Beijing, China, May 2020.

View at: Google Scholar

- 43. S. Al-Saqqa, S. Sawalha, and H. AbdelNabi, "Agile software development: methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, p. 246, 2020. View at: Publisher Site | Google Scholar
- 44. K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, Boston, MA, USA, 2012.
- 45. K. Schwaber, Agile Project Management with Scrum, Microsoft Press, Redmond, WA, USA, 2004.
- 46. G. R. King, "Using the agile development methodology and applying best practice project management processes," Naval Postgraduate School, Monterey, CA, USA, 2014, Master's Thesis. View at: Google Scholar
- 47. C. O'hEocha and K. Conboy, "The role of the user story agile practice in innovation," in *Proceedings* of the International Conference on Lean Enterprise Software and Systems, Springer, Helsinki, Finland, October 2010.

View at: Publisher Site | Google Scholar

- 48. K. Beck, J. V. Sutherland, K. Schwaber et al., The Agile Manifesto, 2001.
- 49. A. Andriyanto and R. Doss, "Problems and solutions of service architecture in small and medium enterprise communities," 2020, http://arxiv.org/abs/2004.10660.

View at: Google Scholar

- 50. Z. Stojanović and A. Dahanayake, *Service-Oriented Software System Engineering: Challenges and Practices*, IGI Global, Hershey, PA, USA, 2005.
- 51. X. Wang, Agile and Lean Service-Oriented Development: Foundations, Theory, and Practice: Foundations, Theory, and Practice, IGI Global, Hershey, PA, USA, 2012.
- 52. S. Graham, G. Daniels, D. Davis et al., *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, SAMS Publishing, Indianapolis, IN, USA, 2004.







view at. Google Scholar

Scientific Programming

+ Journal Menu

PDF



Article Sections

Copyright © 2021 Yanqing Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

More related articles

Multicriteria Decision Making to Continuous Software Improvement Based on Quality Management, Assurance, and Metrics

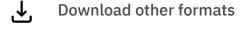
Lei Sun | Shah Nazir | Anwar Hussain

Complex Service Process Optimization Based on Service Touchpoint Association and the Design Structure Matrix

Zhonghang Bai | Chang Liu | ... | Man Ding

Structure, Dynamics, and Applications of Complex Networks in Software Engineering Weifeng Pan | Hua Ming | Chunlai Chai

Service Discovery and Selection Based on Dynamic QoS in the Internet of Things Naiheng Zhang















Follow us:







+ Journal Menu PDF

Article Sections

Journals

Article Processing Charges

Print editions

Authors

Editors

Reviewers

Hindawi XML Corpus

Open Archives Initiative

Fraud prevention

Privacy Policy | Terms of Service | Responsible Disclosure Policy | Cookie Policy | Copyright