Hindawi
Scientific Programming
Volume 2022, Article ID 3556809, 14 pages
https://doi.org/10.1155/2022/3556809

Research Article

Analytical Evaluation of SOA and SCRUM Business Process Management Approaches for IoT-Based Services Development

Fahad H. Alshammari

College of Computing and Information Technology, Shaqra University, Shaqraa, Saudi Arabia

Correspondence should be addressed to Fahad H. Alshammari; fahad.h@su.edu.sa

Received 21 April 2022; Accepted 13 May 2022; Published 30 May 2022

Academic Editor: Fazli Wahid

The SCRUM approach and Service-Oriented Architecture (SOA) framework are critical in assessing the factors that influence the efficiency of a business process and ensuring that business objectives are fulfilled, and the process is on track to meet those objectives. Flexibility and change adoption are critical features for both SCRUM and SOA approaches. Even though both sides encourage agility, the integration of the two independent concepts (SOA is the architectural framework while SCRUM is the development process) should be considered before being used in software management and development projects. This study assessed and analyzed both SCRUM and SOA's diverse and different software architectural frameworks and development methodologies as well as their environment, which is integrated with the context of software project management and development setup for the software development industry. In addition, this study explores the similarities between the SCRUM process model and the SOA architectural framework to see if they are compatible and, if so, how they may be combined to enhance SOA-based projects. This research also looks at how to build and use a SCRUM methodology for large-scale SOA projects. As a result, SCRUM was chosen as the software development methodology for a research and development project based on SOA. In terms of project development and implementation, the complete project structure is made up of eight main parts.

1. Introduction

It is essential to the success of any organization in today's modern, dynamic world for them to be able to welcome, and adapt to, change smoothly and efficiently. SOA is a design method for constructing, developing, managing, and deploying software applications and infrastructure in which all applications are arranged into network-accessible and executable business services [1]. SOA and agile software development approaches necessitate flexibility and change acceptance. It is recommended to thoroughly investigate (before implementing the architecture framework and development process in a software development project) the integration of these two concepts [2] [Dias, 2022 #232]. As a result, using an effective agile approach for SOA-based application development, it is critical to have a system in place that allows for substantial needs changes that occur even during application development while preserving software superiority and quality [3]. The client's requirements are enhanced and predictability is improved by prioritizing iterations and user stories in accordance with agile principles [4]. ASD (agile software development) involves the rapid and efficient development of products through frequent and complete releases that allow their members and sponsors to assess them before they are released [5, 6]. Agile retrospective meetings are used to examine and test the application and process. In this manner, a mockup is transformed into an iterative and incremental approach, a functioning progressive and creative system in which stakeholders provide feedback based on frequent software releases.

Iteration emphasizes the delivery of working software that is of high quality and value to both clients and the project. Besides creating useful artifacts, it also generates others that are beneficial to both the user and the project. Agile processes are designed to deliver high-quality functional deliverables in a timely manner. The main purpose of an agile process is not to focus on designing models and documentation but rather to deliver products and deliver them as quickly and efficiently as possible [7, 8]. SCRUM is one of the agile approaches that, because of its flexibility and straightforwardness, has become a common way of introducing agility [9]. In the business world, it is a well-known agile management technique. Due to the diverse nature of group members and their surroundings, agile application development in an enterprise setting can be complex [10]. By promoting both people and groups, the SCRUM process assists in the discovery of better ways to develop software [11]. With SCRUM, the development process is broken down into sprints in order to focus on delivering a useful product and a project that benefits everyone [12].

SCRUM is the most frequently used agile methodology, stresses group self-management, empirical feedback, and the goal of rapidly improving completely high-quality products [13]. The key shareholders in SCRUM are the product owner, often known as the client, the SCRUM team, and the SCRUM master. The project manager's tasks are shared among these three SCRUM roles [14]. The SCRUM master's job is to make sure that the software development team's process goes well by facilitating SCRUM meetings and ensuring that all SCRUM rules and ideas are adhered to. The SCRUM team is a cross-functional group with specific expertise. Sprint planning meetings are held at the start of each sprint cycle. The product owner and SCRUM team decide on the sprint target, as well as which and what the sprint will strive to accomplish, at this meeting. During the sprint review meeting, the success of the sprint will be measured against its goal. Each sprint ends with a sprint review meeting, during which the team displays everything they have completed. Members of the team discussed how they collaborate, interact, deal with behavioral challenges, and improve their technical abilities. Therefore, among other things, the following sprint is faster at the sprint retrospective meeting [15].

SCRUM is a black-box methodology for building and finding a project's team and processes while working on it. White box process models like the waterfall model are nonagile methodology from a management viewpoint [14]. The sprint's objectives and outcomes are defined during the planning and postmortem phases. Careful preparation is necessary to overcome any growth hurdles in every step or process. Retrospective meetings, also known as postmortems in agile, are necessary for examining operations and establishing a best practices knowledge base. Figure 1 illustrates the properties of process models on a white box and black-box basis.

Nonagile development methods are a time-consuming procedure for creating software. These methods are based on a series of iterations that include identifying requirements, structuring solutions, testing, and deploying them. Traditional procedures involve a lot of documentation at the outset of a project and provide a preset set of requirements. There are many traditional approaches, such as the Waterfall, Spiral model, Win-Win, and Unified Process. A precise upfront strategy, extensive documentation, and a large-scale upfront design are all required for heavy-weight processes. Nonagile development methods are a time-consuming procedure for creating software. These methods are constructed through a series of iterations that include establishing requirements, structuring solutions, testing, and deploying them. Traditional procedures involve a lot of documentation at the outset of a project and provide a preset set of requirements. The Spiral model, Waterfall, Win-Win Spiral model, and Unified Process are examples of traditional approaches. A precise upfront strategy, extensive documentation, and a large-scale upfront design are all required for heavy-weight processes. Large-scale projects with a well-defined and relatively consistent set of needs benefit from traditional methodologies. However, some practitioners have begun to employ agile approaches for specific projects [16].

SOA is a design, development, management, and deployment framework and strategy for software applications. It is a software architecture within which applications are grouped into services, each with network-executable and accessible business rules. SOA is also defined as a combination of applications, clients, and existing systems that can instantly respond to changes in those systems [13]. SOA is often considered a candidate for the most effective methods for developing distributed applications. Instead of starting from scratch, SOA allows you to reuse the functionality of existing systems. In SOA-based systems, this property of reusability boosts organizational economic benefits [14]. Each SOA service is self-contained, but not isolated from the rest of the system. In the problem domain, each service encapsulates a specific logic. In addition to loose coupling, service contracts, autonomy, abstraction, discoverability, and statelessness, SOA has a number of other properties that makes it a powerful structure [17].

Even though SOA and agile methodologies are usually associated with similar concerns, there is still no clear description of how to organize and set up both of these methodologies in one platform for IoT. There is very little information on how this integrated implementation would affect crucial criteria like quality, productivity, agility, and innovativeness.

According to the following statements and actual market needs, the following study examines the similarities between the SCRUM process and the SOA architecture framework to determine if SCRUM and SOA are compatible. This research also looks at how to build and use a SCRUM methodology for large-scale SOA projects. As a result, SCRUM was chosen as the software development methodology for a research and development project based on SOA. The entire project structure is comprised of eight basic modules in the project development and deployment perspective.

This research is classified into seven different sections. The current section introduces the many important concepts employed in this research. The second stage discusses existing research on SCRUM and SOA techniques, as well as their interpretation based on analysis. The research design and setup, as well as the analysis of SOA, SCRUM, and the joint use of SCRUM and SOA, are all covered in the third

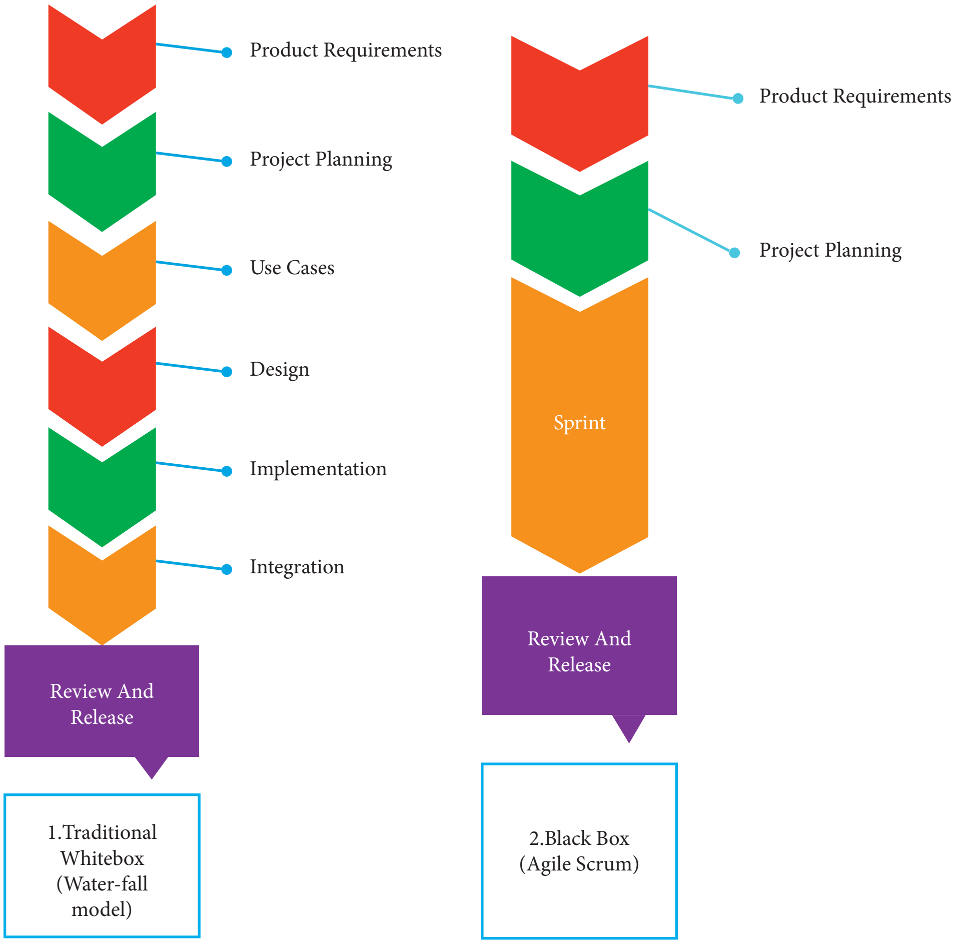

Figure 1: Analogy of white-box and black-box software development processes.

step. The integration analysis is offered in the fourth step, the discussion of SOA and SCRUM integration is presented in the fifth section, and the 6th and 7th steps, respectively, present the entire research's conclusion and recommendation.

2. Literature Study

Agile techniques are designed to ensure that software products are developed quickly and efficiently. Working in iterations [9] makes this possible. As part of an iterative process, working software is delivered to both the project and the customer. During iterations, software and other artifacts are created that benefit the client and project in the same way they benefit the client [9]. Agile processes are primarily concerned with dynamically delivering deliverables and working products, rather than focusing on design models and documentation [7].

When constructing sophisticated and huge software, traditional development experts agree that skipping the paperwork and evaluation process leads to corporate loss of memory. Agile experts, on the other hand, agree that applying agility to software building and worldwide software release has resulted in better results in reaching the required outcomes [18, 19]. It will be difficult for organizations to coordinate and integrate agility with worldwide service provision, but it will also allow them to provide business software that is fast and error-free [20].

Agile software building strategies were created in response to a need from the business sector for lighter, faster, and additional flexible software building methods [21]. As a result, agile building approaches are focused on agility, speed, and willingness to develop [22]. Agile processes, according to the Oxford dictionary, emphasize the importance of being agile, as well as willingness to move, nimbleness, activity, and dexterity in motion [12]. Client satisfaction can be obtained as a result of the lightness of concern procedures, according to several agile approaches such as extreme programming and SCRUM [12].

SCRUM is a popular agile management methodology. Three previous surveys of SCRUM adoption have revealed a focus on actual review, group awareness, and the effort to develop and validate appropriate product increments during a small cycle [23, 24]. SCRUM is a widely used agile approach in the industry, and it is the most common method of bringing agility because of its flexibility and simplicity. By encouraging both individuals and teamwork, the agile software development method supports better software development [11]. Agile processes are designed to ensure that software products are developed quickly and efficiently.

SOA admits to the integration of programs, clients, and current systems into a versatile framework capable of quickly adapting system changes when they are required [14]. SOA is recognized as one of the most effective ways of developing distributed applications [25]. Instead of starting from scratch, SOA allows you to reuse the functionality of existing systems [26]. In SOA-based systems, the attribute of reusability boosts the organization's economic benefits. SOA services are self-contained, yet they are not separated from the rest of the system. In the problem domain, each service encapsulates a specific logic. The primary characteristics of SOA are reusability, loose coupling, service contracts, independence, transparency, search-ability, and statelessness [27].

Researchers and experts are mixed on whether the two approaches are similar and compatible. Critics point out the disparities between SOA and agile methodologies, claiming SOA and agile are moving in opposite directions: SOA is a framework, and agile is a technique [24, 25]. In general, SOAs are top-down approaches, whereas agile is a bottom-up approach [28, 29]. A process paradigm such as agile works at the level of practices, whereas SOA functions at an architectural level [30]. In order to integrate SOA and agile strategies for service creation, advocates of SOA and agile integration argue that the SOA architecture and the guiding principles of agile approaches are complementary [31, 32]. Some experts believe that SOA-based systems are not created like traditional ones [21]. There are several obstacles to overcome, including stakeholder involvement, alignment between the IT and business sectors, and asset reuse. Agility and service orientation are better related to tackling such challenges [33, 34]. SOA and SCRUM share many characteristics when it comes to issues like business understanding, new ways of working, and change responsiveness [35].

When evaluating and specifying SOA and SCRUM, technology challenges, as well as their ties to IT-related business organizations [36] and consumers of personal information, must be taken into account. Other requirements, such as usability and flexibility, are expressed as measurable and designable system characteristics [37, 38].

3. Process Set Up of SOA and SCRUM Approaches

SCRUM as a developmental methodology and SOA as a platform for architecture both address comparable issues, such as flexibility and quick responsiveness to change while maintaining a strong business focus. Even though this inherent operational resemblance is clear, there has not been enough research done to determine the best strategy for integrating these two methods to reap the benefits in combination [39, 40]. There is also a need to look into the consequence of adopting SOA in terms of increasing productivity and making better use of time and resources while using the SCRUM approach.

For analyzing this scenario, a case study using the SCRUM model has been chosen to build SOA-based software applications. M4S is a software application (Mining, Mapping, Modeling, and Management System) for mining resources. The overall project structure is comprised of eight main parts relevant to project development and implementation. The system is built through a series of iterations, which are discussed in the sections below.

The research's effectiveness is measured by identifying, determining, and implementing one of the most effective software engineering combinations: SOA-based architecture, highly effective open-source architectures and solutions, agile development processes, and client and cloud application deployment.

A structured agile approach called SCRUM was applied to develop and refine functional prototypes in an iteration as requirements changed [41, 42]. The case study system components are built with SOA to ensure that they can be used as stand-alone modules via a service gateway. This system's architecture contains aspects of interpersonal connection, sustainability, and service dependability in the natural resource, mapping, simulation, and system management (M4S) stages, as well as the delivery stage.

Cloud infrastructure and distributed computing are used to meet these deployment needs. The developed application's modules are developed to function like services that are interconnected that may be utilized alone and reassembled with additional modules to create a coherent system. Mineral and resource data banks, as well as applications to manage those databanks, are built expressly for the M4S system. They were developed for open-source use as databanks, libraries, and programs.

SCRUM approaches are utilized to guarantee that the transparent process of M4S evaluates and developed, as well as stakeholder involvement and dynamic interaction between the project's various components. As a result of the adoption of open-source methodologies for engineering and project governance, academics, government, and enterprises have formed a multilateral interface to address a range of partners.

[ICT R&D program called M4S: https://www.openm4s.org], as addressed in [ICT R&D program entitled M4S: https://www.openm4s.org].org] Figure 2 depicts the M4S basic infrastructure, which includes open-source libraries, services, hosts, programs, a DBMS, and free software business resource management. Interoperability and open standards are the foundation of client-server architecture and browser-based clients. Within the planned M4S, the graphic also illustrates the components and information flow.

The development of modules is managed and organized using the SCRUM-driven agile software development process [15]. The following section explains how SCRUM was implemented in this case study, as well as the responsibilities and team structure that were assigned: the procedure is set up by agile SCRUM specifications. The roles that exist and work in the SCRUM process are briefly summarized below.

(a) SCRUM Master is responsible for leading SCRUM meetings, maintaining team processes, and ensuring that the process complies with all SCRUM principles [43, 44].

(b) SCRUM Coach facilitates meetings, assists teams in reaching consensus, and ensures that the development environment and team are of high quality.

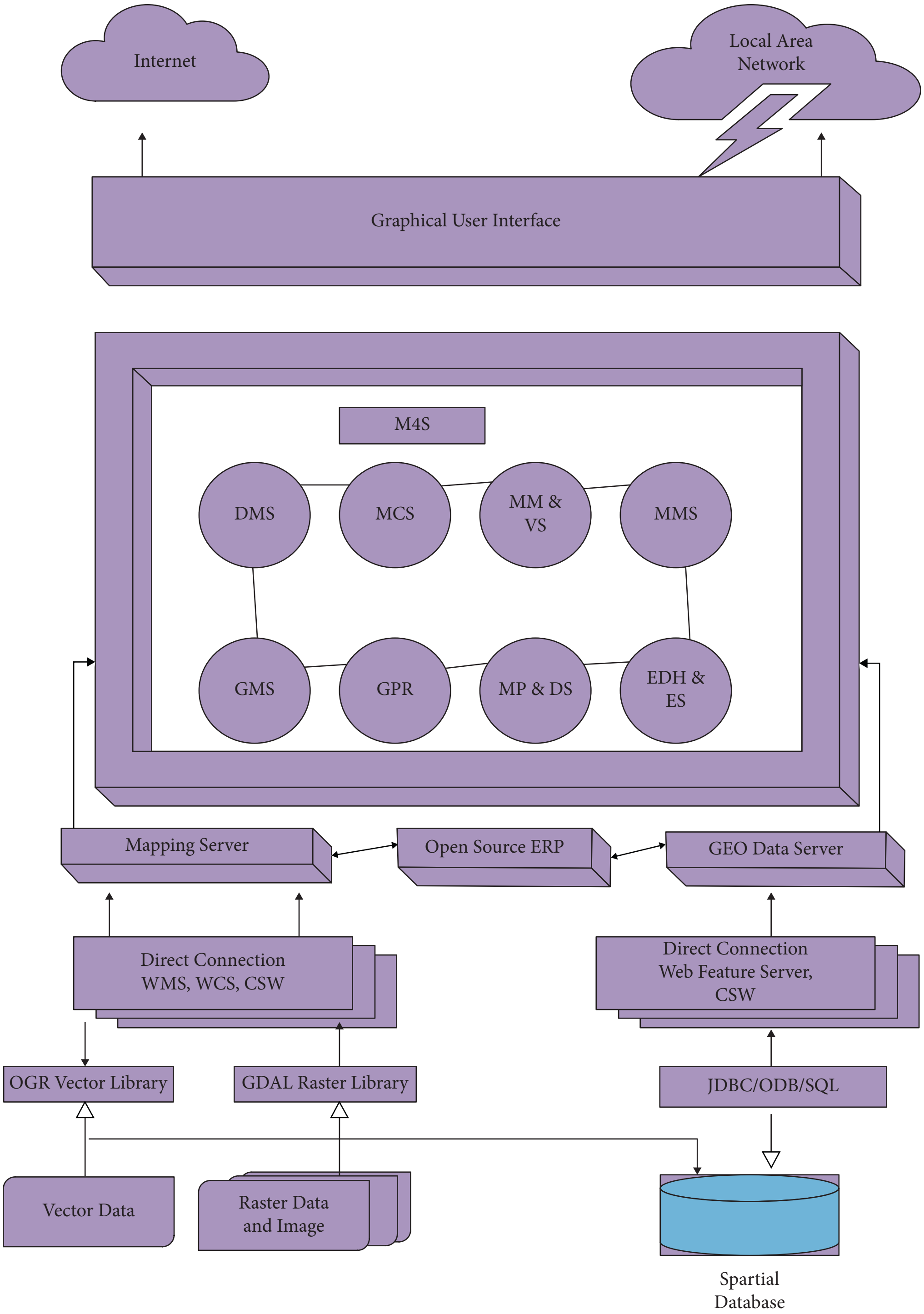

Figure 2: M4S (https://www.openm4s.org) suggested architectures.

(c) SCRUM Team make up of developers, testers, architects, and analysts, which is a cross-functional team. This group is in control of the overall project's outputs and is actively involved in its building. The group is in charge of developing characteristics that meet excellence and functionality standards [45].

(d) Product Owner is an individual who serves the objectives of both end customers and investors. The product owner is responsible for all aspects of quality that are connected to customer requirements. Figure 3 illustrates a typical sprint cycle.

3.1. M4S Process Team Organization. Project development teams are depicted in Table 1 of the M4S process. The owners of products provide the strategy, economics, delivery plan, and design guidelines. SCRUM masters are familiar with

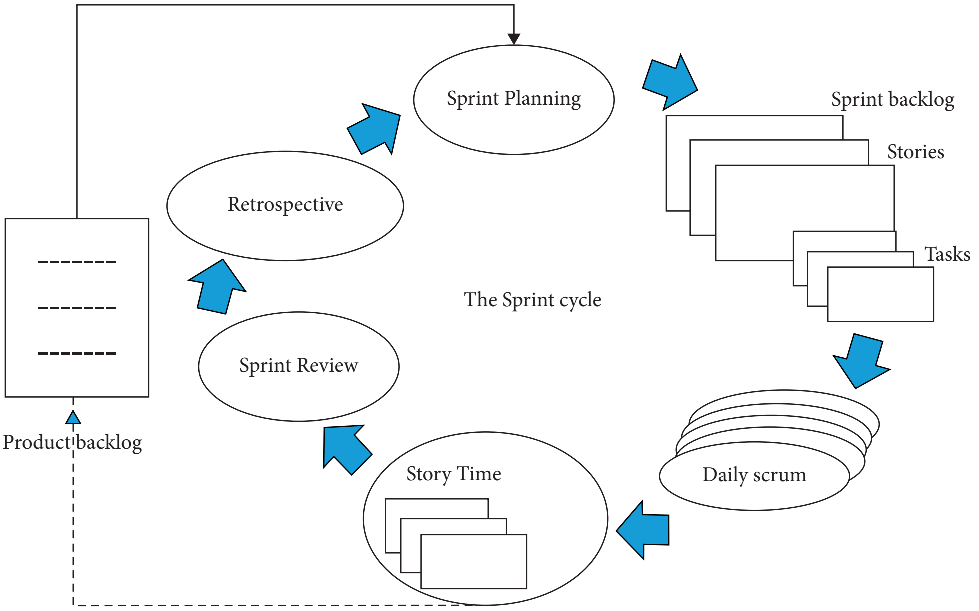

Figure 3: An example of a sprint cycle.

Table 1: SCRUM teams for M4S.

Product owner	Product director, Co-project director
SCRUM master	Leader of the group
SCRUM team	Senior programmers Volunteers (undergraduate and graduate students) For particular jobs such as debugging and performance assessment, the team includes programmers, specialists (S/W and CS professionals, technical governance specialists, miners and software designers, trainers, and part-time designers) Partners, business coordinators, public officials, and academia (analysts)

M4S' architectural framework, as well as the context in which the requirements and features are used. They were responsible for business analysis and quality assurance, as well as sprint cycle management.

The following are the main duties and tasks of each project participant:

Project director (PD) duties are formulation, supervision, and administration of projects.

Co-PD (co-project director) was in charge of coordinating and controlling project operations, as well as maintaining interaction with public and industry partners.

Software and Computer Systems Experts were in charge of supervising system design and performance assessment.

Engineering Management Expert was in duty of overseeing project governance and quality assurance.

Team Leads were in charge of arranging and carrying out design, implementation, and delivery activities.

Unit implementation, debugging, validation, and reporting were the job of developer teams.

Co-project director's duties are handling interaction and collaboration with public and commercial partners.

Part-time Mining and Software Experts are on duty of verifying and confirming M4S and its modules.

Advisers (student monitors) were on the duty of monitoring and advising UG/PG students who were studying on M4S projects.

Part-time programmers were on the duty of module building, debugging, and verification.

Interns worked on different components of M4S projects as a research and implemented and documented the tasks they were allocated.

Figure 4 illustrates the M4S project's work breakdown structure (WBS), as well as the relevant individuals for each block of activity.

Below is a description of the SCRUM method and its many aspects as they apply to the project.

3.1.1. User Stories. A client story encapsulates the user's goals for using the system. From the client's perspective, this goal is converted into a user story [46]. The product owners (project director and codirector) design an approval test card based on the user stories, and the engineers build those stories. The approval test card consists of a series of test instances that are used to evaluate the completed client stories. Only when the approval tests pass will a user story be declared successful. The user stories are then converted into system characteristics [17]. A characteristic is a unit of functionality that has to be incorporated in the system or "task that a character is a unit of functionality that has to be incorporated in the system" or "task that requires to be performed".

Individual features are divided into smaller units in the M4S project. Then, for each specific assignment, story cards are constructed. The average working day was eight hours

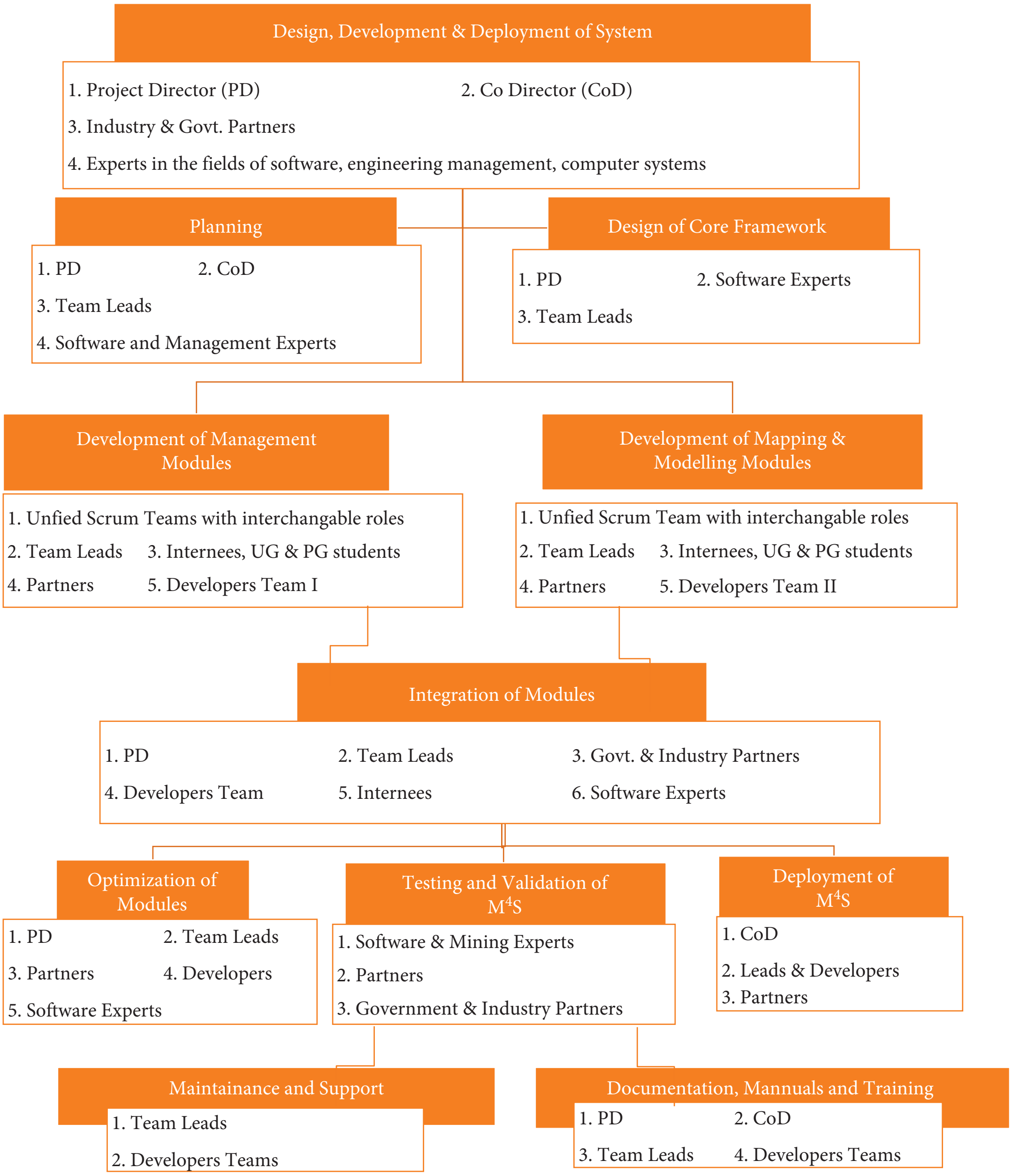
Design, Development & Deployment of System
1. Project Director (PD)
2. Co Director (CoD)
3. Industry & Govt. Partners
4. Experts in the fields of software, engineering management, computer systems
Planning
1. PD
2. CoD
3. Team Leads
4. Software and Management Experts
Design of Core Framework
1. PD
2. Software Experts
3. Team Leads
Development of Management Modules
1. Unfied Scrum Teams with interchangable roles
2. Team Leads
3. Internees, UG & PG students
4. Partners
5. Developers Team I
Development of Mapping & Modelling Modules
1. Unfied Scrum Team with interchangable roles
2. Team Leads
3. Internees, UG & PG students
4. Partners
5. Developers Team II
Integration of Modules
1. PD
2. Team Leads
3. Govt. & Industry Partners
4. Developers Team
5. Internees
6. Software Experts
Optimization of Modules
1. PD
2. Team Leads
3. Partners
4. Developers
5. Software Experts
Testing and Validation of M^4S
1. Software & Mining Experts
2. Partners
3. Government & Industry Partners
Deployment of M^4S
1. CoD
2. Leads & Developers
3. Partners
Maintainance and Support
1. Team Leads
2. Developers Teams
Documentation, Mannuals and Training
1. PD
2. CoD
3. Team Leads
4. Developers Teams

Figure 4: The M4S project's work decomposition framework and resource persons.

daily, and the completion time for tasks was between five and fifteen days. Each story card is usually worked on by one or two members of the team at a time. Assignments are prioritized according to their importance. Each member of the team decides which stories to complete first.

Figure 5 illustrates two stories that are performed until the next narrative is picked.

3.1.2. Sprint. According to the SCRUM process, programming is split over 2 or four-week sprints. Sprints are periodic iterations of development that consist of specific tasks completed [36]. At the beginning of every sprint, throughout a sprint discussion session, tasks are assigned to sprints based on the backlog status. Creating the M4S project should also be preceded by a one-week sprint (named sprint 0). This gives the programmers more freedom to learn about the working platform, scientific solutions, and reusable materials.

In seven sprints, M4S's initial module has been finalized. 1st sprint comprises nine stories, seven of which were finalized inside the sprint time frame and the final two stories were added to the 2nd sprint that will be finished later. Every sprint is documented in an MS Excel file that comprises all of the details about each story, including the team members who worked on it, the amount of time it will take to complete it, and so on. The initial sprint is visualized in Table 2 as an example.

As part of the SCRUM process, the product owner, team, third parties, and SCRUM master meet regularly to address

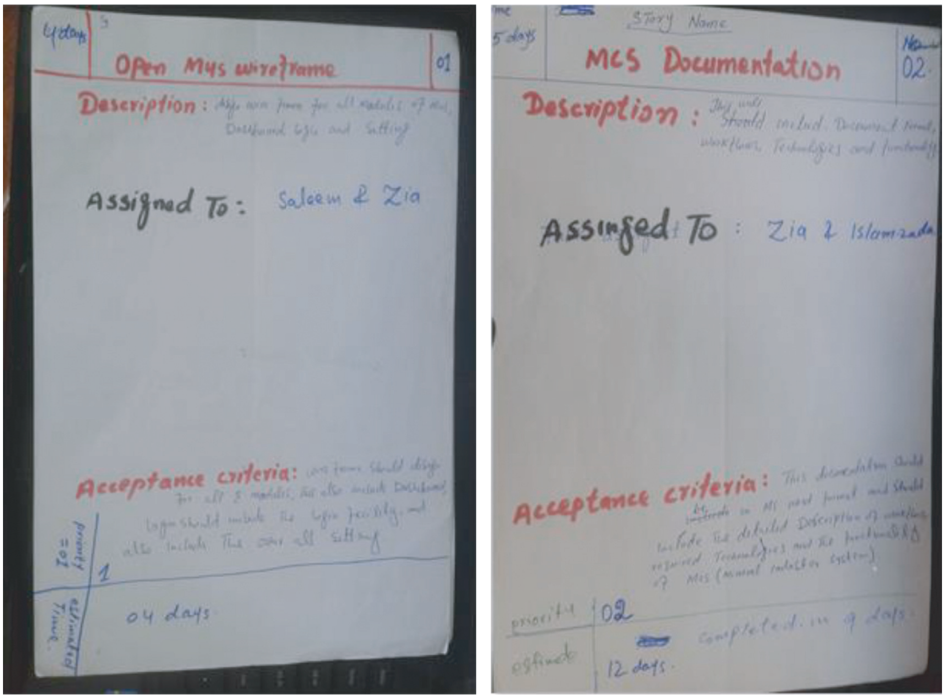

Figure 5: Story cards used in the project.

Table 2: The initial sprint of the M4S project, which was developed in Microsoft Excel.

M4S sprint planning					
Sprint#	Sprint:1				
Sprint name	Wireframe & documentation				
Sprint goal	To build a base for open M4S, complete documentation				
Team member	8	Anwar, Qasim, Zia, Nahyan, Saleem, Zahid, Islam, Iftihar			
Sprint start date	3/3/2014				
Sprint duration	12	Days			
End date	17/03/2014				
Story name task description assigned to start date end date status					
MCS documentation	Document format, workflows, technologies, functionality	Saleem	3/3/2014	17/03/2014	Done
Open M4S wireframe	Wireframe for all modules, dashboards, login, and settings	Zia	3/3/2014	17/03/2014	Done
GeoNetwork study	Study, installation, knowledge sharing	Nahyan	3/3/2014	17/03/2014	Done
Open geo	Study, installation, knowledge sharing	Zia	3/3/2014	17/03/2014	Done
Mapping GIS	Study, example, knowledge sharing	Iftihar	3/3/2014	17/03/2014	Done
Sola	Study, installation, knowledge sharing	Saleem	3/3/2014	17/03/2014	Done
SOA	Study, installation, knowledge sharing	Islam	3/3/2014	17/03/2014	In progress
Wireframe(4 modules)	Wire frame and documentation	Qasim	3/3/2014	17/03/2014	Done
ERP documentation	Study, installation, knowledge sharing	Zahid	3/3/2014	17/03/2014	In progress

issues that have arisen during the process. The gatherings take place to guarantee that the project runs smoothly and that the overall quality phases of the produced system are preserved and maintained. Every day SCRUM meetings, sprint retrospective meetings, assessment meetings for sprints, meetings to plan sprints, and sprint release meetings are all part of the SCRUM process paradigm. Such 5 meetings are conducted and monitored throughout the M4S deployment process in various scenarios. The following subsections go over each of these meetings one by one:

(1) Meeting to Plan the Sprint. There will be 7 sprint sessions to make a work plan that would be completed in a sprint. Every two weeks, a session is usually conducted. The Project Owner and the SCRUM Group (M4S project director and codirector) identify the sprint objectives during this meeting. At the sprint evaluation session, the sprint's success is assessed. The product owner prioritizes the story cards, which means he or she describes the most important story point to the SCRUM team. Finally, the SCRUM Team must create a sprint plan and sprint backlog

that specify how long it will take to complete the assigned work [47].

(2) SCRUM Each Day. There is a regular SCRUM session with the primary goal of ensuring that the application's features are implemented in the sequence in which they were planned. This meeting usually takes place at the start of each sprint and lasts between 20 and 35 min. These meetings will be conducted at the allotted hour. During this meeting, each team member will respond to the following three questions, such as:

(i) How much and what kind of work have you performed until yesterday?

(ii) What are your work objectives regarding today?

(iii) What are the hurdles that would restrict you from achieving the objective?

If an issue comes, it is the role of the team manager (officially known as the SCRUM team) to simplify and tackle the issues. Unsettled problems are highlighted again at a later meeting. This meeting will take place in the same venue, at the same time, and for the same length of time as the previous one.

(3) Sprint Review Meeting. After each sprint, there is a session arranged. Participants of the team present what they have achieved and accomplished during the finished sprint during this session. Typically, this meeting takes the shape of a demo of newly invented features. A majority of the sprint goals were met by the M4S SCRUM team, and 95% of the backlog items were fetched for the sprint, as well as the majority of the product backlog items. As a result, 98% of the elements of a project like M4S meet the "performed" standard, are approved, and have a "done" label. The leftover 2 percent of elements not implemented at the sprint are planned for upcoming sprints. The elements were closed whenever the project manager (product owner) is satisfied with the 98% delivery record, and the next backlog is modified to highlight the features that require additional implementation.

(a) The M4S project director and co-project director approve all finished stories for every sprint.

(b) During the meeting, the project director accepts the completed features.

(c) For each story in a sprint, all acceptance tests are run.

(d) A code assessment was performed on the whole code.

(e) In bug backlog contains no bugs and issues.

(f) All completed story points are approved for publication.

(g) The products were thoroughly examined and no serious errors were detected.

(h) A successful backup has been created.

(i) The majority of installation documents are updated.

(4) Retrospective on the Sprint. Members of the M4S team went over the approach team's procedures, interrelate, behavioral characteristics, refining methodological skills, etc. during this meeting. The team leader (SCRUM Master) and team members discussed three things during the SCRUM meeting: what went well, what did not go well, and what improvements can be made at the next sprint. The team's all performance (in the form of metrics) is assessed, and methods for growth (within the shape of KPIs) are highlighted, as already explained in our earlier research paper. It is also recommended that members of the team exchange technical skills and knowledge and that they maintain constant communication with the product owner, among other things. This meeting takes place after each sprint and lasts about 2 and 3 hours.

(5) Planning for Release. In a release backlog, a release strategy meeting is a method of defining, prioritizing, and separating software features in an ongoing and undisturbed way. This entails identifying and committing to the following:

(a) Release goals

(b) Prioritized set of user stories (features to be developed)

(c) Rough estimates of user stories

(d) A release dates

The following are the primary input steps in the M4S project that are held during this meeting:

(a) The user stories are evaluated by the members of the team, i.e., making intelligent guesses on the story's size.

(b) Find your sprint velocity, that is, the amount of story points you finish in a sprint; you can find more information in our previous article.

(c) Use the forecast to calculate story points that will be finished as part of a date-based release (velocity x number of sprints). (Total story points $*$ velocity) sprints are expected to be needed to complete performance-based releases.

Story points (calculated above) are used to select the most important stories. A functionality-based release will be selected if all of the stories will be completed by the expected completion date (calculated above).

4. Integration of SOA and SCRUM

In this section, we will explore how SCRUM, a process for developing software, and SOA, an architectural style, interact. An SOA architecture provides an organized and well-defined technology architecture that incorporates notions of agility and design patterns to enable quality, efficiency, and productivity. To construct flexible and adaptive systems, service reusability and abstraction [16] are employed. By building new services and modifying existing ones on top of our underlying architecture, SOA enables agility at the design level rather than modifying the architecture. SCRUM enables incremental development and quicker client feedback. SCRUM facilitates incremental development by allowing customers and the development team to provide feedback more quickly. SCRUM and SOA both help companies construct business and development strategies that are aligned.

The M4S project development environment uses concepts that are similar to the SCRUM and SOA processes to reach the common goal of developing necessary software. The SOA can still take advantage of agile SCRUM while operating in a well-managed environment with no undesirable overlap. A clear strategy for bringing change in modest steps, on the other hand, should be devised.

SCRUM can also be used for traditional software or designs by specifying needs in a backlog and providing customers with incremental outcomes. Nonetheless, additional development expenditures and overall project costs are projected [48].

When it comes to cost-effectiveness and responsiveness, IT organizational design plays a crucial role in helping agile to achieve the claimed benefits. Otherwise, the agile approach may only provide a minor value to the organization.

SOA and SCRUM, as previously said, are two techniques that pursue different paths. SCRUM is a bottom-up approach to process development (begins with basic design and ends with prototype delivery). Services are developed based on the SOA system (a top-down approach). Does the question arise as to how these various methodologies are compatible with one another when used in a development process? Another debate is if SOA follows the agile approach in the same way as SCRUM does? If the answer is yes, how should you proceed? Finally, how might these two approaches be combined to maximize the benefits afforded by each separately? This section addresses these concerns, as well as the similarities between SOA and SCRUM metrics. Figure 6 illustrates the standard SOA and SCRUM development paradigm.

Table 2 shows the most essential SOA and SCRUM metrics that can help enterprises and consumers who want to employ SOA and SCRUM combined in a software engineering project (Table 3).

Some SCRUM and SOA metrics have been combined to be used in both SCRUM and SOA environments. These metrics integrate SCRUM and SOA metrics, which have similar goals but differ in terminology. The development process can be expedited when these KPIs are appropriately measured, resulting in enhanced business agility, adaptability, and compatibility for SOA and SCRUM settings.

Ratio measurement of completed stories versus planned stories in SCRUM and percentage measurement of new services versus total services in SOA evaluate product performance using ratios and percentages. They are both measuring service and work, but they are utilizing distinct terminology. Team velocity, development time, and average development time are three metrics that the team uses to track its progress and the time needed to complete a service. This metric can be combined with team velocity measurements, which can be used as an integration measurement for SOA and SCRUM. Since SCRUM and SOA use different metrics for assessing quality, but their intent is the same, both of these metrics can be used as quality assurance metrics to gauge the quality of services under a SCRUM model.

Are team members ready to work when it comes to the team enthusiasm metric? Members of the free-planned architecture team will work gladly and obey all laws and rules if they are satisfied and are working in a pleasant environment. They will collaborate in a team setting. When team members are working eagerly since they are happy, communication between them will always be pleasant.

5. SCRUM and SOA Compatibility Discussion

The major goal of SOA and SCRUM is to make the entire organization agile by utilizing services like software application building blocks [49]. Also, using the SCRUM process paradigm to develop software involves increasing organizational agility by combining SCRUM principles that can improve collaboration, communication, and feedback [13, 14]. Although there are some similarities and incompatibilities between SCRUM and SOA, as previously noted, there are certain distinctions and incompatibilities between the two techniques To design a framework for their integration, the compatibility and commonality of these two development approaches are investigated [50, 51].

5.1. Compatibility of SCRUM and SOA. It may be unclear why commonality and compatibilities between SCRUM and SOA are required. The two concepts (process and architectural style) are completely separate from one another. Although most SOA teams are subconsciously aware of how services are planned and developed, identifying compatibilities of both when used together is sensible and valuable. The focus is entirely on policies related to service design. SOA, like SCRUM, stimulates certain team composition and communication patterns among teams based on policies [3]. SCRUM is similar to human hands working using gloves, whereas SOA is similar to that glove. The majority of SCRUM and SOA principles are thought to be compatible because they are both adaptable approaches [52]. Although SOA that utilizes a traditional method, for example, waterfall design, turns predictive, predictive techniques depend mainly on requirement assessment and careful scheduling at the beginning of every cycle. Any change which has to be made will be subjected to a comprehensive modification monitoring and prioritization procedure. An agile methodology employs an adaptive technique in which no extensive preparation is done but only specific upcoming jobs associated with the features which must be achieved are assigned. The squad responds to changing product needs regularly. The product is thoroughly evaluated on the regular basis, reducing the possibility of serious errors mostly in the future. The strength of the agile approach is customer interaction, and the agile manufacturing setting is characterized by open dialogue and minimal paperwork. Teams work closely together and are frequently based in the same geographical area. As a result, in the creation of SOA projects, an agile paradigm is best suitable for the quick implementation of adaptable services [53]. SCRUM application implementation would be ineffective lacking a solid understanding of the organization's goals. Absent a clear

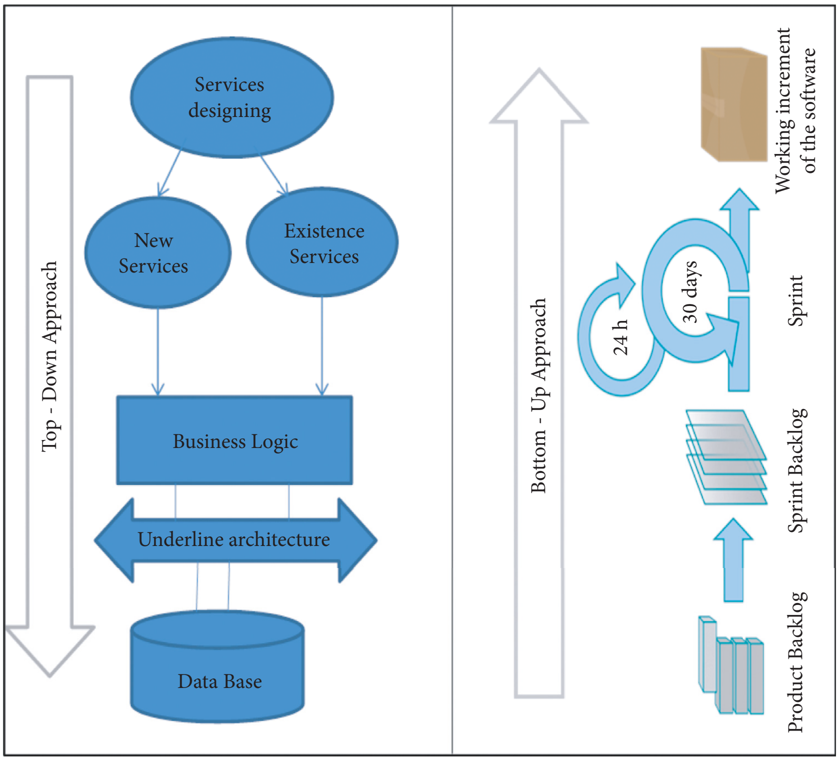

Figure 6: A basic SOA and SCRUM development procedure.

picture about how to develop and create utilizing SCRUM process concept guidelines, SOA is a loss of time and money. Because it emphasizes the capacity to react with modifications, the adoption of SOA as a widespread technique for designing software apps is also on the growth. Because SOA development and agile development both need adjusting to changes, it appears that utilizing the agile approach to build SOA apps is a logical fit.

Thus SCRUM and SOA are both regarding agility, and they may be utilized combined by adopting a series of principles and ideas that are not mutually exclusive. In this manner, they keep each other in balance [54].

5.2. SCRUM and SOA Commonalities. SOA appears to be an architectural method that highlights the necessity for commercial organizations to be intelligent and capable of responding to rapid commercial changes, as previously indicated. The goal of reusing software is indefinable; developing services can assist in reaching that goal. Teams build services as a result of the SOA method that can then be combined into applications [48, 55]. A SCRUM process emphasizes change response in application development [56]. Teams can help enterprises become more agile by incorporating both technological and nontechnical approaches. SCRUM and SOA are complementary and have similar goals. Change is flexible in both circumstances, and companies must be able to deal with it successfully. As a result, SCRUM might be a viable option for developing SOA-based apps.

5.3. SOA and SCRUM Adversity. Whereas SCRUM and SOA seem to be compatible, there are some significant differences between the two methodologies which should be acknowledged and managed when using SOA and SCRUM in the same project [57]. One of the main reasons is that the two techniques originate from different directions and go in opposite directions. SCRUM has typically been used for small projects, but as processes have improved and practitioners have gained knowledge and experience, the SCRUM manifesto's standards could now be used for massive software projects. The SOA approach to application development is top-down and divide-and-conquer [4]. The "divide" part of the approach inevitably leads to a lack of communication across teams. There seem to be three primary areas wherein SCRUM and SOA are in conflict with one another.

(1) SOA promotes architecture to be upfront; however, the SCRUM community does not encourage upfront design and considers Big Design Upfront (BDUF) to be an antipattern.

(2) SOA promotes teams to be divided along functional lines, whereas agile supports the formation of cross-functional teams.

(3) SOA lacks formal feedback and communication within development teams, whereas agile emphasizes frequent assessment on both technical and personal levels.

It can be difficult to make changes when employing SOA in a large setup [53]. Other difficulties, such as a lack of team communication and service discoverability, may exist in addition to software agility and cost. Rather, we should strive for robust collaboration and transparency. Every team will

Table 3: Typical SCRUM and SOA metrics.

	SCRUM	SOA	Commonalities
1	Ones that have been finished vs. stories that have been scheduled	As a percentage of overall services, new created services and utilized	The proportion and percent of finished tasks (services) performed in one sprint are determined to use these two indicators.
2	Team velocity	The length of time it takes to build a service and the average length of time it takes to develop a service	Such three factors track the team's efforts for a service that should be delivered within a sprint in regard to per-sprint and duration. These metrics could be combined to determine SOA and SCRUM integration by developing a team velocity metric.
3	Customer satisfaction	Confirmation of quality of service	When adopting the SCRUM developmental methodological framework, certain metrics can be employed to assess the efficiency of service. Both measures have quality as a common attribute, and they can be utilized to provide the metric for evaluating SOA and SCRUM integration.
4	Team enthusiasm	Violation of architecture policies	The team members will interact in a friendly, collaborative fashion when they are happy, content, and working in a pleasant environment. They will devote their whole attention and concentration to product development, ensuring that product quality remains standard. They will also obediently observe the predetermined architectural policies and rules. We can also produce a huge number of high-quality services in a short length of time if the SCRUM team is happy and in a relaxing environment. As a result, we can say that the SCRUM metrics "team excitement & communication", "violations of architecture policies", and "median period to service creation" is dependent and also can affect the project when they were not targeted. Within the development platform, such metrics are employed to assess how employees adhere to rules and processes.
5	Team communication	Average time to service development	
6	Advancement of the process in retrospect	Availability and usefulness of the service	A single SCRUM and SOA metric can be created by combining the two metrics. Because all SCRUM procedures conclude with a retrospective meeting, it is the last opportunity to revive the overall activities. Whenever these services get completed in a sprint, an evaluation meeting will be conducted to evaluate the produced service's performance and usefulness, such as how to access the service & how it operates.
7	Control of technological debt	Project and maintenance costs are reduced	The major goal of these two measures is to lower product development costs by making the maximum use of resources and team members' abilities. These two measures can be combined into a single metric to allow SCRUM and SOA techniques to be used together.

commit an easily customizable version of their service available for internal testing. Test data will be segregated from service by the service team so that it can continue to function even if the code is corrupted. The customer care team will advise clear and straightforward instructions for adding additional calls to the service stub.

6. Conclusion

Instead of starting from scratch, service-oriented architecture (SOA) allows you to reuse the functionality of existing systems. This reusability attribute in SOA-based systems enhances financial benefits for companies. In contrast, the SCRUM process model focuses on cycle and customer feedback to increase functionality while also enabling for predictability of evolving requirements. This study establishes a baseline for finding commonalities and compatibilities between SCRUM and SOA processes to ensure the most effective use of the structured SCRUM management process in SOA-based application development. The majority of SCRUM and SOA rules are not mutually incompatible, according to this research. Neither SCRUM nor SOA is mutually exclusive. They both leverage rules and principles that apply to agility. Using formal KPIs based on specific SCRUM and SOA indicators, it is possible to determine the success of a combined SCRUM and SOA development environment. All organizations that wish to apply SCRUM and SOA within an integrated context should be able to evaluate their agility, complexity, effectiveness, and worth based on such KPIs. Even though the study's output is accurate, the researcher highlights several constraints which must be carefully evaluated before implementing the conclusions to a SCRUM and SOA integrated project in the enterprise [58, 59].

7. Future Work

This research work has been done according to the study scope and domain but some work is remaining in a domain that could be future research directions. For better understanding, SCRUM can be used as a process model in a distributed environment. In addition, to determine what effect SCRUM and SOA combination will have on project performance when it is employed in a distributed development environment, a system for evaluating and identifying metrics and KPIs should be established in all integrated SCRUM and SOA projects, whether distributed or not.

Data Availability

There is no data available.

Conflicts of Interest

The author claim that this paper does not include any conflicts of interest.

Acknowledgments

This study could not be started nor be achieved without Shaqra University's encouragement, and its continued support.

References

[1] N. Bieberstein, B. Sanjay, F. Marc, J. Keith, and S. Rawn, *Service-oriented Architecture Compass: Business Value, Planning, and enterprise Roadmap*, FT Press, NJ, USA, 2006.

[2] M. Lankhorst, *Agile Service Development: Combining Adaptive Methods and Flexible Solutions*, Springer Science & Business Media, Germany, 2012.

[3] F. Rago, "Self-organizing business networks, SOA and software maintenance," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2008.

[4] A. Manifesto, "Agile manifesto," *Haettu*, vol. 14, p. 2012, 2001.

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

[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, Article ID 10352, 2020.

[7] S. Mohanarajah and M. A. Jabar, "AN improved adaptive and dynamic hybrid agile methodology to enhance software project success deliveries," *Journal of Theoretical and Applied Information Technology*, vol. 75, no. 3, 2015.

[8] S. Shahzad and S. Nazir, "Issues and implications of scrum on global software development," *Bahria University Journal of Information & Communication Technology*, vol. 8, no. 1, 2015.

[9] N. Naik and P. J. Relax, "It'sa game: utilising gamification in learning agile scrum software development," in *Proceedings of the 2019 IEEE Conference on Games (CoG)*, IEEE, London, UK, August, 2019.

[10] P. Kutschera and S. Schäfer, "Applying agile methods in rapidly changing environments," in *RTO Symposium on Technology for Evolutionary Software Development,* Citeseer, 2002.

[11] M. Awad, *A Comparison between Agile and Traditional Software Development Methodologies*, p. 30, University of Western Australia, Nedlands, 2005.

[12] W. M. Theunissen, A. Boake, and D. G. Kourie, "In search of the sweet spot: agile open collaborative corporate software development," in *Proceedings of the ACM International Conference Proceeding Series*, 2005.

[13] S. W. Ambler, "The agile scaling model (ASM): adapting agile methods for complex environments," *Environments*, 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, Indiana, 2012.

[15] A. Bhagwat, "Role of beacon architecture in mitigating enterprise architecture challenges of the public sector," *Advances in Government Enterprise Architecture*, p. 56, IGI Global, Hershey, PA, USA, 2008.

[16] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, John Wiley & Sons, 2017.

[17] D. Gabioud, *D3. 2–Overall System Requirements and Functional Specifications*.

[18] C. Verma, *The Effects of Organizational Culture on Risk Management during Software Development*, The British University in Dubai (BUiD), NJ, USA, 2009.

[19] G. Borrego, A. L. Morán, R. R. Palacio, A. Vizcaíno, and F. O. García, "Towards a reduction in architectural knowledge vaporization during agile global software development," *Information and Software Technology*, vol. 112, pp. 68–82, 2019.

[20] J. McGovern, S. Oliver, J. Ashish, and L. Mark, *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations*, Springer Science & Business Media, Germany, 2006.

[21] P. Abrahamsson, S. Outi, R. Jussi, and W. Juhani, "Agile Software Development Methods: Review and Analysis," 2017, arXiv preprint arXiv:1709.08439.

[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.

[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, Article ID 106241, 2020.

[24] S. Moyo and E. Mnkandla, "A novel lightweight solo software development methodology with optimum security practices," *IEEE Access*, vol. 8, Article ID 33735, 2020.

[25] 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.

[26] O. Goubali, I. Abdenour, P. Line, B. Laurianne, K. Djamal, and B. Alain, "Service-oriented control-command components for designing complex systems," in *Proceedings of the International Conference on Human-Computer Interaction*, Springer, Orlando, FL, USA, July, 2019.

[27] 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.

[28] Y. Baghdadi, "A comparison framework for service-oriented software engineering approaches," *International Journal of Web Information Systems*, 2013.

[29] F. Kamoun, "A roadmap towards the convergence of business process management and service oriented architecture," *Ubiquity*, vol. 2007, no. April, 2007.

[30] Z. Dragičević and S. Bošnjak, "Agile architecture in the digital era: trends and practices," *Strategic Management*, vol. 24, no. 2, pp. 12–33, 2019.

[31] J. T. F. Chaves, "Service-oriented architecture (SOA)," *Agile Development Methods and Quality Assurance (QA): A Case Study*, University of Brasılia, Brasilia, Brazil, 2019.

[32] S. Abidi, E. Mehrez, F. Myriam, and B. G. Henda, "A comprehensive framework for evaluating web services composition methods," *International Journal of Web Information Systems*, vol. 15, no. 3, pp. 324–345, 2019.

[33] M. Fischer, I. Florian, J. Christian, and W. Axel, "Directions for future research on the integration of SOA, BPM, and BRM," *Business Process Management Journal*, vol. 25, no. 7, pp. 1491–1519, 2019.

[34] A. Plugge, S. Nikou, and H. Bouwman, "The revitalization of service orientation: a business services model," *Business Process Management Journal*, vol. 27, 2020.

[35] J. Paramanathan, *Security of Lightweight-And Heavyweight-IT in a High-Tech Hospital*, University of Oslo, Oslo, Norway, 2019.

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

[37] 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, Article ID 107699, 2019.

[38] J. Bloomberg, *The Agile Architecture Revolution: How Cloud Computing, Rest-Based SOA, and mobile Computing Are Changing enterprise IT*, John Wiley & Sons, NJ, USA, 2013.

[39] C. Ardito, T. B Maria, C Danilo, and L Rosa, "Integrating a SCRUM-based process with human centred design: an experience from an action research study," in *Proceedings of the 2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*, IEEE, Buenos Aires, Argentina, May 2017.

[40] H. Yang, *In A Quest to Solve Information System Agility Problems: A SaaS Experience*, Victoria University of Wellington, Wellington, New Zealand, 2018.

[41] A. Alhazmi and S. Huang, "Integrating design thinking into scrum framework in the context of requirements engineering management," in *Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering*, Beijing, China, May 2020.

[42] S. Al-Saqqa, S. Sawalha, and H. AbdelNabi, "Agile software development: methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, 2020.

[43] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, Boston, 2012.

[44] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, Germany, 2004.

[45] G. R. King, *Using the Agile Development Methodology and Applying Best Practice Project Management Processes*, DTIC Document, Virginia, 2014.

[46] 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.

[47] K. Beck, J. V. Sutherland, and K. Schwaber, *The Agile Manifesto*, 2001.

[48] A. Andriyanto and R. Doss, "Problems and Solutions of Service Architecture in Small and Medium Enterprise Communities," 2020, arXiv preprint arXiv:2004.10660.

[49] Z. Stojanović and A. Dahanayake, *Service-oriented Software System Engineering: Challenges and Practices*, IGI Global, Hershey, Pennsylvania, 2005.

[50] A. Onan, H. Bulut, and S. Korukoglu, "An improved ant algorithm with LDA-based representation for text document clustering," *Journal of Information Science*, vol. 43, no. 2, pp. 275–292, 2017.

[51] Y. Cui, Z Islam, S Sara et al., "Analysis of service-oriented architecture and scrum software development approach for IIoT," *Scientific Programming*, vol. 2021, Article ID 6611407, 14 pages, 2021.

[52] K. A. Abdelouhab, D. Idoughi, and C. Kolski, "Agile & user centric SOA based service design framework applied in disaster management," in *Proceedings of the 2014 1st International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, IEEE, Algiers, Algeria, March 2014.

[53] A. Kumar, R. Liu, and Z. Shan, "Is blockchain a silver bullet for supply chain management? technical challenges and research opportunities," *Decision Sciences*, vol. 51, no. 1, pp. 8–37, 2020.

[54] J. P. Dias, A. Restivo, and H. S. Ferreira, "Designing and constructing internet-of-Things systems: an overview of the ecosystem," *Internet of Things*, vol. 19, Article ID 100529, 2022.

[55] S. Graham, D. Doug, S. Simeon et al., *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, SAMS publishing, Carmel, Indiana, 2004.

[56] S. S. Islam Zada and S. Nazir, "Issues and Implications of Scrum on Global Software Development," *Bahria University Journal of Information & Communication Technologies*, vol. 8, 2015.

[57] A. S. Dadras, *IT Agility through Service-Oriented Architecture*, University of New South Wales, Kensington, Australia, 2016.