

# Challenges for Migrating to the Service Cloud Paradigm: An Agile Perspective

Stavros Stavru, Iva Krasteva, and Sylvia Ilieva

Sofia University “St. Kliment Ohridski” 5, James Bouchier Str.,  
P.B. 48 1164 Sofia, Bulgaria  
{stavross, ivak, sylvia}@fmi.uni-sofia.bg

**Abstract.** Migrating to the Service Cloud Paradigm implies the migration of legacy software systems to a service-oriented architecture with deployment in the cloud. Although this specific software modernization paradigm promises numerous strategic and operational advantages, it poses also many complex organizational and technical challenges, among which is the lack of mature processes, methods and techniques. This paper examines the questions of whether agile methods and techniques could be scaled to fit the migration to the Service Cloud Paradigm and how they could help overcoming the challenges of software modernization in this specific context. The research methodology presented here first extracts the challenges of the migration to Service Cloud Paradigm through a systematic literature review and then, using expert judgment, evaluates how different agile techniques, taken from Scrum and Extreme Programming (XP), could address the identified challenges. As a result, a ranked list of applicable agile techniques is presented and suggestions for their adoption in software modernization projects are drawn.

**Keywords:** Agile Software Development, Software Migration, Software Modernization, Cloud Computing, Service-Oriented Architecture.

## 1 Introduction

Building a modernized system from scratch often requires a huge investment in time and efforts, making this modernization approach almost unfeasible in real industrial settings. But software organizations, continuously pressured by their dynamic business and IT environment, have to modernize. Thus new and more undemanding modernization approaches are needed. With the increasing popularity of Service Oriented Architecture (SOA), the reuse of legacy system by exposing its functionality through services is identified as a feasible and very promising modernization approach. One particular modernization paradigm is the Service Cloud Paradigm, which stands for combination of Cloud Computing and SOA for development of Software as a Service (SaaS) systems and their deployment on the Cloud. Although this paradigm promises numerous strategic and operational advantages, it poses also many complex organizational and technical challenges, including but not limited to extensive business context analysis and complementary strategic efforts, steep

learning curve, external dependencies and lock-ins, complex governance, interoperability, etc. Being in its early adoption stage, the migration to the Service Cloud Paradigm still lacks mature process models, methods and techniques, so needed for providing systematic guidelines for approaching software modernization in this specific context and overcoming its challenges. Agile methods and techniques, on the other hand, has been widely adopted in the industry and successfully applied in various contexts, different from traditional software development. Thus the question of whether agile methods and techniques could be scaled to the migration to Service Cloud Paradigm becomes a question of present interest.

The study presented in the paper proposes a systematic approach for reviewing the challenges of SOA and Cloud Computing relevant to Service Cloud Paradigm. Further it evaluates, through expert judgment, various agile techniques in terms of their potential to address these challenges. These agile techniques are taken from Scrum and XP software development methods since they are the most widely adopted methods through the agile community in the recent years (in more than two thirds of the projects surveyed by VersionOne<sup>1</sup>). The results of the evaluation step are used for making suggestions on the applicability of the techniques and their inclusion in a particular agile method for migration to Service Cloud Paradigm. The design of the agile method is out of the scope of this paper since it involves an extensive research, which cannot be presented within the limits of this paper.

The present study has been carried as part of FP7 European research project for reuse and migration of legacy applications to interoperable cloud services (REMICS<sup>2</sup>). The main objective of the REMICS project is to specify, develop and evaluate a tool-supported model-driven methodology for migration to the Service Cloud Paradigm. The migration process consists of several steps: (1) understanding the legacy system in terms of its architecture and functions; (2) designing a new SOA application that provides the same or better functionality; and (3) verifying and implementing the new application in the cloud.

The reminder of the paper is organized as follows: Section 2 describes the methodology used for conducting literature review and evaluating agile techniques; Section 3 presents the challenges from SOA and Cloud Computing fields, extracted by the review process and relevant for the migration to the Service Cloud Paradigm; Section 4 discusses the results of the evaluation of agile techniques and their potential to address the identified challenges; and Section 5 concludes the paper and outlines directions for future research.

## 2 Methodology

The research methodology used in the present study contains two basic steps. During the first step a systematic literature review on challenges of SOA and Cloud Computing, which are relevant to Service Cloud Paradigm, is performed. The second step involves a particular evaluation technique to study whether agile techniques can be used to address the challenges identified on the first step. The steps are presented in details in the subsections below.

---

<sup>1</sup> <http://www.versionone.com>

<sup>2</sup> <http://www.remics.eu/>

## 2.1 Review

Articles were eligible for inclusion in the review based on their relevance to the review objectives, which are: (1) they describe the current state of research and practice in either SOA and/or Cloud Computing; and (2) they identify and discuss different challenges these areas poses to both academia and industry. The relevance was evaluated by reviewing the abstracts of the articles and grading them as either relevant or irrelevant. The inclusion was also restricted by the type of the study, including only review articles and excluding theoretical (conceptual) or empirical studies. No restrictions were made in regard to the publication year of the articles thus covering all the years available in the included electronic database at the time of the review (1 January, 2012). Other exclusion criteria used were: (1) the article does not have abstract or the abstract is not available from the included electronic database; (2) the access to the full text of the article is restricted; and (3) the full text of the article is not available in English.

The search strategy included both journals and conference papers, and was limited to the Scopus electronic database. Scopus is the largest abstract and citation database of research literature and quality web sources, which ensured the coverage of nearly 18,000 titles from more than 5,000 publishers. The titles of both journals and conference papers were searched using the following search terms:

- (1) Service-Oriented Architecture - ("Service-Oriented" AND (Challenges OR Review OR Landscape OR Roadmap OR "State of"));
- (2) Cloud Computing - ("Cloud Computing" AND (Challenges OR Review OR Landscape OR Roadmap OR "State of")).

Applying the search strategy resulted in an initial pool of 121 articles, 65 articles for SOA and 56 for Cloud Computing. Some additional articles, not covered by the search strategy, were also included as being recommended by the research community. Thus, by using the inclusion and exclusion criteria the initial pool of articles was limited to 58 articles. Their full texts were thoroughly examined in order to extract the challenges of SOA and Cloud Computing, presented in the subsequent sections.

## 2.2 Evaluation

Agile techniques were evaluated using expert's opinions through Delphi technique. Delphi is a technique frequently used for eliciting consensus from within a group of experts and has many advantages over other methods of using panel decision making [1]. Various researchers [1-3] all found that one of the major advantages of using it as a group response is that consensus will emerge with one representative opinion from the experts. Other advantages include its simplicity, anonymity, controlled feedback from the interaction and others [4]. Some limitations include that judgments are derived from the subjective opinions of experts and may not be representative, it requires adequate time and participant commitment, its validity extremely depends on the expertise and experience of the panelists, etc. [4] However, Linstone [2] recommends the Delphi technique when the examined issue does not allow the use of analytical techniques but can benefit from the subjective judgments on a collective basis, which we believe is our case.

The process followed was the one proposed by Pfeiffer [5] and included the following three steps:

- (1) Experts' recommendations - A questionnaire was sent to a panel of four experts (with an average of 9 years of both academic and industrial experience in agile software development), asking them to review the list of challenges extracted by the review process and make subjective judgment and recommendations on which agile techniques (from Scrum and XP) could be used to address these challenges. From each expert a list of agile techniques was obtained.
- (2) Experts' ratings - From the individual recommendations, a consolidated list of agile techniques was created. The list was then sent to each expert to further evaluate the relevance (on a five-point scale) of all techniques in regard to all challenges.
- (3) Experts' consensus - The consolidated list, together with experts' ratings was sent again in order to discuss big differences in ratings. At the end consensus was gained, resulting in a final list of agile techniques and addressed challenges.

### **3 Challenges in Migrating to the Service Cloud Paradigm**

The challenges identified by the review process were sorted into two categories:

- (1) Organizational challenges, including challenges from all levels of the organization as strategic challenges (e.g. process reengineering, external dependencies, etc.), managerial challenges (e.g. governance, competence acquisition) and operational challenges (e.g. lack of methodologies, tools, etc.), and which were mostly process and people oriented;
- (2) Technical challenges, which included design, implementation, verification challenges and deployment challenges, and were product and technology oriented.

As the focus of this study was on the migration to the Service Cloud Paradigm, we expected that not all of the challenges discussed by the reviewed articles would be relevant. For that reason, we limited the extraction of challenges to: (1) for Service-Oriented Architecture we included challenges relevant for both service consumers and providers, as software migration could involve both developing of new services and using external ones; (2) for Cloud-Computing we included only challenges relevant for the cloud consumers, excluding challenges covering the development of cloud infrastructure, how security should be achieved within this infrastructure and other challenges more relevant for the cloud provider.

#### **3.1 Challenges of Service-Oriented Architecture**

Total of 31 articles were thoroughly reviewed in order to extract the challenges relevant to SOA. As many challenges were found, they were further consolidated into

total of 17 challenges, 10 of which were organizational and 7 were technical. A summary of these challenges, together with their references, is presented in Table 1.

**Table 1.** Challenges of Service-Oriented Architecture

#	Challenge	Description	Ref.
<b>Organizational challenges</b>			
O1	Identification and availability of service consumers	In some cases services are first developed and then offered to real service consumers (mostly due to customer unavailability). Therefore the service provider defines and prioritizes requirements based on its assumptions of what "potential consumers" would need or how "potential usage patterns" would look like.	[6-7]
O2	Identification and reengineering of business processes / tasks	Business process modeling and analysis might be an intensive activity and a prerequisite for the specification, design and implementation of services (if top-down or domain composition approach is incorporated)	[7-14]
O3	Lack of software development methods	Due to its early adoption stage, there is still scarce availability of software development methods and techniques for guiding the migration to and the development of software systems based on Service-Oriented Architecture.	[15-17]
O4	Unclear system ownership	Service-Oriented Architecture tends to blur the boundaries of software systems ownership, so who owns what (in terms of services) might become an issue.	[8-9, 11-12, 18-19]
O5	Complex governance	Service-Oriented Architecture raises unique challenges and can be derailed unless an effective governance framework is established to clearly identify roles and responsibilities.	[9-10, 12-13, 16-22]
O6	External dependencies	By developing software systems using external services, the organization is exposed to higher risk due to third party dependencies. Thus risk mitigation becomes important issue.	[11, 15]
O7	Acquisition of competencies and expertise	For an organization adopting or migrating to Service-Oriented Architecture, thorough understanding of the underlying technologies remains highly critical. At the same time too many technologies and standards could be involved and/or not enough expertise could be available in the market.	[6, 8, 17-18, 21, 23]
O8	Addressing increased complexity	The plethora and diversity of parties involved (service consumers, service developers, integrators, infrastructure developers, etc.), technologies incorporated, diversity of types of services available as run time components, the middleware and the infrastructure required to make these components interoperable and deployable, the technical skills and expertise required etc. increase significantly the complexity of the developed or migrated software system (System of systems).	[17-19, 24]

**Table 1.** (*continued*)

O9	Immature tools and integrated development environments	Due to its early adoption stage, there are few and still under development tools and integrated development environments to assist software engineers in their migration or development efforts.	[8-9, 12, 16-19]
O10	Evolving standards	While the core web services standards (i.e. XML, SOAP, WSDL, and UDDI) are relatively mature and stable, many of the additional standards that address important issues such security and reliability (e.g. WS-Coordination, WS-Atomic Transaction, WSDM, WS-Reliability, etc.) are still under development.	[8-9, 12, 16-19, 25]
<hr/> Technical challenges <hr/>			
T1	Addressing service design	Service design needs to determine what constitutes a service and its operations (or interface), and make decisions about the level of service aggregation (or service granularity). There is no straightforward approach (neither design methodology) and often the design process becomes challenging and cumbersome.	[6, 9, 11-12, 14-19, 21-23, 25-28]
T2	Securing reconfiguration and composition	As services could be used by different consumers to perform different business tasks (covering various business scenarios), they should be reconfigurable and facilitate composition and orchestration.	[7, 9, 12, 16, 18, 22, 28-31]
T3	Securing compatibility and standards compliance	Service interfaces cannot be changed very often due to issues such as backward and forward compatibility and compliance with standards.	[11, 16, 18, 22, 26, 32]
T4	Addressing security, interoperability and other quality aspects	There could many challenges related to security (due to publicized interfaces, distributed and no trusted network and channel, unintended orchestration, the use of open standards, etc.), interoperability (due to different standards, middleware and service infrastructure, service semantics, etc.), conducting performance and reliability analysis, ensuring correctness and reliability of test cases, provenance, etc.	[6, 8, 18-19, 23, 28, 30-35]
T5	Simulation of deployment environment	Test instances of the services and simulated deployment environment could be required during implementation as real services could be unavailable or too critical to be executed (e.g. ordering purchase).	[8-9, 11-12, 16]
T6	Testing services	Testing in Service-Oriented Architecture might require complex logistics, test instances of services, more levels of testing (due to diversity of parties involved), new techniques for issues localizations and troubleshooting, greater and more diverse exception handling, more complex route cause and impact analysis, etc.	[6, 9-11, 15-16, 18, 23, 26, 33, 36]
T7	Securing SLA and QoS contracts	Fulfillment of Service Level Agreement (SLA) and Quality of Service (QoS) contracts during services usage could be challenging due to network connectivity issues, unpredicted load, etc.	[9-12, 18, 22-23, 30]

As seen from Table 1, most of the reviewed literature (84% of the reviewed articles) was concerned with technical challenges. This was expected as SOA has to achieve some degree of technical excellence before it is widely accepted by the industry, and with the extensive use of SOA in real industrial settings the majority of organizational challenges would emerge.

The challenges in Table 1 have various implications on the incorporation of agile methods and techniques into the Service Cloud Paradigm, especially when it comes to organizational challenges. For example, working with potential service consumers and potential usage patterns (O1) contradicts with agile methods and techniques, which are mostly customer centric and require intensive interaction and collaboration with the customer. Not involving real customers (or at least somebody who could represent them) in the early phase could result in customer feedback received late in the development lifecycle, customer negotiation and relationship issues when real customers interests are conflicting (e.g. in terms of service functionality, interface, etc.); complex impact analysis and lack of flexibility and agility because of unknown customers (especially when the services are publicly available), etc. The prerequisite to define business processes and tasks in advance (O2), before the actual design, implementation and verification of services (if top-down or domain composition approach is incorporated) also poses some limitations on the use of agile methods and techniques, which emphasize on working software and thus applies very short increments (from 2 to 4 weeks) with potentially shippable products. This might also hinder achieving organizational agility and responding to change, because it limits the possible introduction of changes in the predefined business models late in the development life cycle. Challenges as unclear system ownership (O4, e.g. who should test external services, included in the current iteration; who should be responsible for the infrastructure required; etc.) and the need for complex service governance (O5, e.g. requiring adherence to codes, policies and standards in order to secure process and services compliance) set the focus on defining formal processes (incl. roles, responsibilities, activities, artifacts, etc.) and extensive usage of tools (incl. integrated development environments), while agile emphasize on individuals and interactions. In terms of technical challenges, there were also many implications. Coding and testing (mostly unit and integration) are not the only major activities in SOA. Considerable attention is paid also to service design, composition and orchestration (T1, T2), which in some cases could even replace traditional coding (e.g. when the system is built from external services only). In terms of testing, system testing (T5, T6) becomes crucial as there could be many parties involved (e.g. service developers, integrators, consumers, infrastructure developers, etc.). Software quality (T3, T4, T7) is much more emphasized because security, interoperability, reliability and compatibility, etc. are real life concerns in the SOA. There are also other implications to be considered, including increased complexity (O8, O10), evolving standards, etc.

### **3.2 Challenges of Cloud Computing**

A total of 27 articles in the area of Cloud Computing were reviewed. The extracted challenges were further consolidated into 12 challenges, including 8 organizational challenges and 4 technical. They are shown in Table 2.

**Table 2.** Challenges of Cloud Computing

#	Challenge	Description	Ref.
<b>Organizational challenges</b>			
O1	Trust in the cloud provider	Trust becomes an issue when the data and computation are decentralized and distributed beyond the boundaries of the organization.	[37-47]
O2	External dependencies and vendor lock-ins	Exit strategies and lock-in risks (ability to switch cloud providers) are important concerns for organizations exploiting Cloud Computing, as well as vendor dependencies (e.g. only using services the provider is willing to offer).	[37-42, 46, 48-52]
O3	Security and privacy	The success or failure of Cloud Computing is highly dependable on how confident consumers feel that the services of the cloud provider are reliable and available, secure and safe, and that privacy is protected.	[37-38, 40-49, 51-60]
O4	Delegating data governance	By moving data into the Cloud, organizations might lose some capabilities to govern their own data, including data creation and receipt, distribution, use, maintenance and disposition.	[37-38, 43, 46-47, 49, 51-54, 57, 60-61]
O5	Introduction of new roles and responsibilities	The responsibilities for integrating in-house IT with the cloud infrastructure, adapting the in-house software systems with the cloud platform, etc. should be clearly defined and appropriate roles assigned (e.g. cloud infrastructure integrators, cloud platform integrators).	[61]
O6	Acquisition of competencies and expertise	Cloud Computing involves new technologies and standards, requiring the acquisition of new competences and expertise.	[61]
O7	In-house integration	Integration of in-house IT to the Cloud infrastructure could be hard and time consuming undertaking.	[37, 42]
O8	Early adoption stage	Due to its early adoption stage, Cloud Computing still lacks enough major cloud service providers, lacks variety of mature tools (incl. integrated development environments) and standards are continuously evolving (and even competing).	[49-50, 61-62]
<b>Technical challenges</b>			
T1	Addressing architectural and technical constraints	Cloud Computing poses some architecture constraints on the way software systems are build, incl. decomposition, decoupling, componentization, etc. and some technical constraints as what should be the type of database, etc.	[52]
T2	Maintenance and troubleshooting difficulties	Defects detection, localization and troubleshooting could be problematic as there might be no access to or sufficient control over the cloud infrastructure.	[51, 54, 63]
T3	Lack of cloud provider support	There might be lack of provider support or it might be provided as a paid service.	[54]
T4	Cloud interoperability	Cloud providers are not using any kind of common open standards, which makes cloud interoperability a serious concern.	[50, 61, 63]



The challenges mostly studied by the research community are related to security and privacy (81% of the reviewed articles), and trust (60%). This is expected since organizations tend to be extremely sensitive when crucial business assets (as business data and computation) are delegated to external parties. In cloud environment there might be no control (and even visibility in some cases) over these assets and no guarantees (even if there are strong service level agreement, strict standards, etc.) that the cloud provider would act in accordance to the interests of the organization and no external parties (or events) would significantly interfere in their relations (e.g. through business acquisition). Thus mistrust and lack of security and privacy might become the greatest barrier for the adoption of cloud solutions.

Cloud Computing and its challenges further affect the way agile methods and techniques could be incorporated into the Service Cloud Paradigm. Trust (O1), security and privacy (O3) of data and computation are as much important for the organization as for its customers, so the organization-customer collaboration, central to agile software development, might need to be extended to include the cloud provider (in order to increase transparency, visibility, responsiveness, etc., so needed for the building trust and confidence). Vendor lock-ins (O2) might further hinder organizational flexibility and agility (e.g. as one could not change its cloud provider effortlessly and in a timely manner), while external dependencies (O2) could decrease the business value delivered to customers (e.g. due to new requirements coming from the cloud infrastructure or the organization is pressured to use specific and expensive software licenses coming from the cloud provider, etc.) and could further decrease organizational responsiveness (e.g. due to contracting). In terms of technical challenges, the maintenance and troubleshooting difficulties (T1), together with the lack of cloud support (T3), might require more involvement from upper management (in order to assure the commitment of the cloud provider). Also, the architectural and technical constraints (T2) and the need to consider quality aspects (T2, T4, such as interoperability, security, performance, etc.) might require considerable efforts to be made for architecture and design, except for coding and testing.

4 Results

The present section discusses the results of the evaluation of agile techniques based on the Delphi technique. The results are shown in Table 3, where the techniques are sorted by the total number of challenges they are expected to address (shown in brackets next to the technique’s name).

Table 3. Evaluation of Agile techniques based on the challenges they are expected to address

Agile Technique	SOA Challenges	CC Challenges
Extreme Programming (XP)		
Small Releases (20)	O1, O6, O7, O8, O10, T1, T2, T3, T4, T5, T6, T7	O1, O2, O3, O4, O6, O8, T2, T3
Planning Game (14)	O1, O2, O4, O6, T1, T2, T3	O1, O2, O3, O4, O5, O7, T3

**Table 3.** (continued)

Pair Programming (13)	O5, O7, O8, O9, T1, T2, T3, T4, T5, T5	O6, T1, T2
Whole Team (13)	O2, O4, T1, T2, T3, T4, T5, T6	O5, O6, T1, T2, T3
Continuous Integration (12)	O5, O8, O8, T2, T3, T4, T5, T6	O3, O4, T1, T2
Test-Driven Development (11)	O5, O8, T1, T2, T3, T4, T5, T6, T7	T1, T2
System Metaphor (10)	O5, O8, T1, T2, T3, T5, T6	O6, O7, T1
Collective Code Ownership (7)	O4, O5, O7, T5	O6, O7, T2
Refactoring (6)	O5, O8, T5, T6	T1, T2
Simple Design (5)	O8, T1, T4, T5	T2
Coding Standards (4)	O5, O8, T3	T2
<b>Scrum</b>		
Sprint (20)	O1, O6, O7, O8, O10, T1, T2, T3, T4, T5, T6, T7	O1, O2, O3, O4, O6, O8, T2, T3
Sprint Planning Meetings (18)	O1, O2, O4, O5, O6, O7, O8, T1, T2, T3, T7	O1, O2, O3, O4, O5, O7, T3
Cross-Functional Teams (17)	O4, O5, O6, O7, O8, T2, T3, T4, T5, T6, T7	O5, O6, O8, T1, T2, T3
Product Backlog (15)	O1, O2, O4, O8, T1, T2, T3, T4, T7	O3, O4, O6, O8, T1, T2
Spring Backlog (15)	O1, O2, O4, O8, T1, T2, T3, T4, T7	O3, O4, O6, O8, T1, T2
Daily Scrum (12)	O4, O5, O6, O7, T2, T3, T4	O2, O6, T1, T2, T3
Scrum of Scrums (9)	O4, O5, O6, O8	O6, O7, T1, T2, T3
Scrum Master (9)	O4, O5, O6, T2, T3, T7	O2, O5, T3
Product Owner (9)	O2, O3, T1, T2, T3, T7	O2, O3, O4
Sprint Review Meeting (5)	O1, O5, T4, T3	T3
Sprint Retrospective (5)	O6, O7, O8	O6, T3
Sprint Burn Down Chart (4)	O5, O8	T2, T3

As seen from Table 3, the agile techniques recommended the most by the experts was Small Releases (from XP) and Sprints (from Scrum). Their arguments include receiving feedback quickly (e.g. identifying security, interoperability, privacy, etc. issues earlier in the development lifecycle), increasing responsiveness to change (e.g. limiting the impact of evolving standards on software delivery and cost/time overruns, overcoming vendor lock-ins by deploying increments on different cloud infrastructures, etc.) and issues (e.g. broken SLAs or QoS contracts), building trust and confidence (e.g. through frequent communication, increased visibility and traceability, etc.), effective competency acquisition (through learning by doing), early delivery of business value, etc. Next, Planning Poker (from XP) and Sprint Planning

Meeting (from Scrum) were rated second. The motivation for that was the active involvement of customers or customers' representatives in the development process and enhanced collaboration (e.g. resulting in more effective business process modeling and analysis by taking into consideration also the technical aspects, whether architectural or technical, of the migrated software system, better identification and prioritization of customer requirements based on customer value and collective estimations, effective identification of possible composition/choreography workflows, etc.), early detection of risks due to external dependencies and their timely mitigation (e.g. gaining support and commitment by customers and upper management to remove impediments coming from external parties), clarification of team responsiveness and service ownership (e.g. by identifying external services to be used by third parties, who will integrate them, who will test them, etc.), early escalation of quality concerns (e.g. compatibility, standards compliance, etc.), etc. The third most rated agile techniques were Pair Programming (from XP) and Cross-Functional Teams (from Scrum). Among the reasons for recommending these techniques were more effective acquisition of competencies and expertise (e.g. through daily knowledge transfer between programming pairs, homogeneous distribution of knowledge and expertise within the cross-functional team, etc.), managing increased complexity (e.g. through highly collaborative and knowledgeable workers), increased team responsiveness and support (e.g. team support for issues troubleshooting and resolution, improvement of used tools and procedures, identifications of impediments and external dependencies, etc.), secured quality in terms of security, interoperability, performance, etc. Other agile techniques, highly rated were Whole Team and Continuous Integration (from XP) and Product Backlog, Spring Backlog and Daily Scrum (from Scrum).

Based on the presented results and following the Pareto principle (80% of the effects come from 20% of the causes) [64], we would recommend that an organization, migrating to the Service Cloud Paradigm and interested in Agile Software Development, should start with small releases (or sprints), incorporate planning meetings similar to either Planning Game or Sprint Planning Meeting, and encourage Cross-Functional Teams. Afterwards, it might continue with Product / Sprint Backlogs, Daily Scrums, Continuous Integration and On-Site Customer in order to further address the challenges of the migration process. Finally, as all examined agile techniques have the potential to contribute to the Service Cloud Paradigm, an organization might also consider full implementation of either XP or Scrum (or a hybrid), as this will ensure cohesiveness and will allow the organization to take full advantage of these methods.

## 5 Conclusions

Agile methods and techniques have the potential to address many of the challenges possessed by the Service Cloud Paradigm. Thus an agile service- and cloud-oriented modernization method could be reasonable not only because the target organization might have already adopted the agile values and principles, but also because it

promises to overcome the challenges of migrating to the Service Cloud Paradigm. The future work is the identification of challenges from other related areas as Model-Driven Development and Software Modernization, and the development of a complete agile method to support organizations in the adaptation of their legacy systems to the Service Cloud Paradigm.

**Acknowledgments.** The research presented in this paper was partially supported by FP7 project REMICS, contract No. 257793 and by National Scientific Fund, Bulgarian Ministry of Education and Science, Research Project agreement n. DO-02-182.

## References

1. Helmer, O., Helmer-Hirschberg, O.: Looking forward: a guide to futures research. Sage Publications (1983)
2. Linstone, H.A., Turoff, M.: The Delphi method: techniques and applications. Addison-Wesley Pub. Co., Advanced Book Program (1975)
3. Dalkey, N.C., Corporation, R.: Delphi: Rand (1967)
4. Yousuf, M.I.: Using Experts' Opinions through Delphi Technique. Practical Assessment Research & Evaluation 12 (2007)
5. Pfeiffer, J.: New look at education: systems analysis in our schools and colleges. Odyssey Press (1968)
6. Tilley, S., et al.: Migrating to SOA: approaches, challenges, and lessons learned. Presented at the Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research, Toronto, Ontario, Canada (2010)
7. Bano, M., Ikram, N.: Issues and Challenges of Requirement Engineering in Service Oriented Software Development. In: 2010 Fifth International Conference on Software Engineering Advances (ICSEA), pp. 64–69 (2010)
8. Mahmood, Z.: Service oriented architecture: potential benefits and challenges. Presented at the Proceedings of the 11th WSEAS International Conference on Computers, Agios Nikolaos, Crete Island, Greece (2007)
9. Lewis, G.A., et al.: Effects of service-oriented architecture on software development lifecycle activities. Software Process: Improvement and Practice 13, 135–144 (2008)
10. Kontogiannis, K., et al.: A research agenda for service-oriented architecture. Presented at the Proceedings of the 2nd International Workshop on Systems Development in SOA Environments, Leipzig, Germany (2008)
11. Kontogiannis, K., et al.: The Landscape of Service-Oriented Systems: A Research Perspective. In: International Workshop on Systems Development in SOA Environments, SDSOA 2007, ICSE Workshops, p. 1 (2007)
12. Lewis, G.A., et al.: Common Misconceptions about Service-Oriented Architecture. In: Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, ICCBSS 2007, pp. 123–130 (2007)
13. Zheng, L., et al.: Facing Service-Oriented System Engineering challenges: An organizational perspective. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–4 (2010)

14. Hutchinson, J., et al.: Evolving Existing Systems to Service-Oriented Architectures: Perspective and Challenges. In: IEEE International Conference on Web Services, ICWS 2007, pp. 896–903 (2007)
15. Nasr, K.A., et al.: Realizing service migration in industry—lessons learned. *Journal of Software Maintenance and Evolution: Research and Practice*, n/a–n/a (2011)
16. Papazoglou, M., et al.: Service-Oriented Computing: A Research Roadmap. *International Journal of Cooperative Information Systems* 17, 223 (2008)
17. Kokko, T., et al.: Adopting SOA: Experiences from Nine Finnish Organizations. In: 13th European Conference on Software Maintenance and Reengineering, CSMR 2009, pp. 129–138 (2009)
18. Mahmood, Z.: The Promise and Limitations of Service Oriented Architecture. *International Journal of Computers* 1, 74–78 (2007)
19. Becker, A., et al.: Value Potentials and Challenges of Service-Oriented Architectures. *Business & Information Systems Engineering* 3, 199–210 (2011)
20. Maurizio, A., et al.: Service Oriented Architecture: Challenges for Business and Academia. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, p. 315 (2008)
21. Bhallamudi, P., Tilley, S.: SOA migration case studies and lessons learned. In: 2011 IEEE International Systems Conference (SysCon), pp. 123–128 (2011)
22. Papazoglou, M.P., et al.: Service-Oriented Computing: State of the Art and Research Challenges. *Computer* 40, 38–45 (2007)
23. Tilley, S.: Report from the 5th and 6th international workshops on adoption-centric software engineering: Migrating to SOA. In: 2011 IEEE International Systems Conference (SysCon), pp. 135–139 (2011)
24. Nigul, L., et al.: The SOA programming model: challenges in a services oriented world. Presented at the Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, Ontario, Canada (2009)
25. Feuerlicht, G.: Enterprise SOA: What are the benefits and challenges? In: *Systems Integration*, pp. 36–43 (2006)
26. Lewis, G.A.: SMART: The Service-oriented Migration and Reuse Technique: Carnegie Mellon University, Software Engineering Institute (2005)
27. Kulkarni, N., Dwivedi, V.: The Role of Service Granularity in a Successful SOA Realization A Case Study. Presented at the Proceedings of the 2008 IEEE Congress on Services - Part I (2008)
28. Issarny, V., et al.: Service-oriented middleware for the Future Internet: state of the art and research directions. *Journal of Internet Services and Applications* 2, 23–45 (2011)
29. Brown, P.C.: Succeeding with SOA: realizing business value through total architecture. Addison-Wesley (2007)
30. Choudhury, P., et al.: Deployment of Service Oriented architecture in MANET: A research roadmap. In: 2011 9th IEEE International Conference on Industrial Informatics (INDIN), pp. 666–670 (2011)
31. Yi, W., Blake, M.B.: Service-Oriented Computing and Cloud Computing: Challenges and Opportunities. *IEEE Internet Computing* 14, 72–75 (2010)
32. Balasubramaniam, S., et al.: Challenges for assuring quality of service in a service-oriented environment. In: ICSE Workshop on Principles of Engineering Service Oriented Systems, PESOS 2009, pp. 103–106 (2009)
33. Simanta, S., et al.: Information assurance challenges and strategies for securing SOA environments and web services. In: 2009 3rd Annual IEEE Systems Conference, pp. 173–178 (2009)

34. Venters, C.C., et al.: Provenance: Current directions and future challenges for service oriented computing. In: 2011 IEEE 6th International Symposium on Service Oriented System Engineering (SOSE), pp. 262–267 (2011)
35. Phan, C.: Service Oriented Architecture (SOA) - Security Challenges and Mitigation Strategies. In: Military Communications Conference, MILCOM 2007, pp. 1–7. IEEE (2007)
36. Canfora, G., Di Penta, M.: Testing services and service-centric systems: challenges and opportunities. *IT Professional* 8, 10–17 (2006)
37. Verma, A., Kaushal, S.: Cloud Computing Security Issues and Challenges: A Survey. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. CCIS, vol. 193, pp. 445–454. Springer, Heidelberg (2011)
38. Xiu-ping, Z.: Study on the opportunities and challenges of the cloud computing for Chinese medium-sized and small enterprises. In: 2011 International Conference on E - Business and E -Government (ICEE), pp. 1–3 (2011)
39. Habib, S.M., et al.: Cloud Computing Landscape and Research Challenges Regarding Trust and Reputation. In: 2010 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), pp. 410–415 (2010)
40. Dillon, T., et al.: Cloud Computing: Issues and Challenges. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 27–33 (2010)
41. Al-Qirim, N.: A Roadmap for success in the clouds. In: 2011 International Conference on Innovations in Information Technology (IIT), pp. 271–275 (2011)
42. Kim, W., et al.: Adoption issues for cloud computing. Presented at the Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, Kuala Lumpur, Malaysia (2009)
43. Oh, T.H., Lim, S., Choi, Y.B., Park, K.-R., Lee, H., Choi, H.: State of the Art of Network Security Perspectives in Cloud Computing. In: Kim, T.-h., Stoica, A., Chang, R.-S., et al. (eds.) SUCoS 2010. CCIS, vol. 78, pp. 629–637. Springer, Heidelberg (2010)
44. Roberts II, J.C., Al-Hamdani, W.: Who can you trust in the cloud?: a review of security issues within cloud computing. Presented at the Proceedings of the 2011 Information Security Curriculum Development Conference, Kennesaw, Georgia (2011)
45. Hay, B., et al.: Storm Clouds Rising: Security Challenges for IaaS Cloud Computing. In: 2011 44th Hawaii International Conference on System Sciences (HICSS), pp. 1–7 (2011)
46. Timmermans, J., et al.: The Ethics of Cloud Computing: A Conceptual Review. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 614–620 (2010)
47. Takabi, H., et al.: Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy* 8, 24–31 (2010)
48. Ovadia, S.: Navigating the Challenges of the Cloud. *Behavioral & Social Sciences Librarian* 29, 233–236 (2010)
49. Mathisen, E.: Security challenges and solutions in cloud computing. In: 2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST), pp. 208–212 (2011)
50. Petcu, D.: Portability and interoperability between clouds: Challenges and case study. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssi re, J. (eds.) ServiceWave 2011. LNCS, vol. 6994, pp. 62–74. Springer, Heidelberg (2011)
51. Mathur, P., Nishchal, N.: Cloud computing: New challenge to the entire computer industry. In: 2010 1st International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 223–228 (2010)

52. Chang, H., Choi, E.: Challenges and security in cloud computing. In: Kim, T.-H., Vasilakos, T., Sakurai, K., Xiao, Y., Zhao, G., Ślęzak, D. (eds.) FGCN 2010. CCIS, vol. 120, pp. 214–217. Springer, Heidelberg (2010)
53. Kossmann, D., Kraska, T.: Data Management in the Cloud: Promises, State-of-the-art, and Open Questions. *Datenbank-Spektrum* 10, 121–129 (2010)
54. Joshi, K.R., et al.: Dependability in the cloud: Challenges and opportunities. In: DSN, pp. 103–104 (2009)
55. Morrell, R., Chandrashekar, A.: Cloud computing: new challenges and opportunities. *Network Security* 2011, 18–19 (2011)
56. Al-Azzoni, I., et al.: Abstract only: performance evaluation for software migration. *SIGSOFT Softw. Eng. Notes* 36, 42 (2011)
57. Roberts, J.C., Al-Hamdani, W.: Who can you trust in the cloud?: a review of security issues within cloud computing. Presented at the Proceedings of the 2011 Information Security Curriculum Development Conference, Kennesaw, Georgia (2011)
58. Lar, S.U., et al.: Cloud computing privacy & security global issues, challenges, & mechanisms. In: 2011 6th International ICST Conference on Communications and Networking in China (CHINACOM), pp. 1240–1245 (2011)
59. Zhao, W.: An Initial Review of Cloud Computing Services Research Development. In: 2010 International Conference on Multimedia Information Networking and Security (MINES), pp. 324–328 (2010)
60. Zhang, Q., et al.: Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications* 1, 7–18 (2010)
61. Dudin, E., Smetanin, Y.: A review of cloud computing. *Scientific and Technical Information Processing* 38, 280–284 (2011)
62. Loutas, N., et al.: Cloud Computing Interoperability: The State of Play. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 752–757 (2011)
63. Grobauer, B., Schreck, T.: Towards incident handling in the cloud: challenges and approaches. Presented at the Proceedings of the 2010 ACM workshop on Cloud Computing Security Workshop, Chicago, Illinois, USA (2010)
64. Pareto, V.: *Manual of political economy: Scholars Book Shelf* (1971)