Agile Process for Integrated Service Delivery

Marjana Shammi, Sietse Overbeek, Robert Verburg, Marijn Janssen, and Yao-Hua Tan

Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands, EU
mshammi@gmail.com,
{S.J.Overbeek,R.M.Verburg,M.F.W.H.A.Janssen,Y.Tan}@tudelft.nl

Abstract. Companies want to become more customer-centric and embrace Integrated Service Delivery (ISD) to provide a single shop. The integration of services of different organizations results in the creation of dependencies among services which are in different stages of the life-cycle. Only with effective collaboration between the parties and coordination of development activities, ISD can be managed efficiently. With the adoption of Agile methodologies, performance can be gained reducing the complexities of software development and focus on collaboration and coordination aspects. Therefore, this research proposes a model on how to manage the service lifecycle of ISD in a top-down view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile Scrum approach. The method is different from existing ones as it uses agile principles applied to life-cycle management and incorporating iterative development from the commencement of requirement analysis until the completion of the development of services.

Keywords. Integrated Service Delivery, Agile, Scrum, service, development process

1 Introduction

Companies are becoming more and more customer-centric: understanding and anticipating the needs of customers, designing what customers want, and then aggregating and managing the components and suppliers to rollout products and services quickly and cost-effectively to meet ever-changing customer needs. With the opportunity of Integrated Service Delivery (ISD), companies can support clients in an integrated environment possibly reducing cost and time. ISD can be defined as *'a bundle of services provided by a single service provider or multiple service providers collaborating with each other through a single interface accessible to clients*' [1-3]. Providing these services, service providers face a number of challenges related to organizational integration, resistance towards change and managing the dependencies among services. With effective collaboration and coordination of activities, ISD can arguably be managed efficiently. To support the development of ISD and the updating of services, companies look for agility in their development process of ISD. Research has shown that adoption of Agile methodologies has reduced complexities in software development and there is an increasing focus on collaboration and coordination to achieve performance gain [4]. Agile development processes are characterized by

incremental and iterative software development by teams closely collaborating together[5]. To the best of our knowledge, there has not been research on how to manage the service lifecycle of ISD in a top-down view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile development approach. Researching this aspect can provide an insight on the iterative perspective of the process and help companies to incorporate and benefit the best practices out of it, to effectively collaborate and coordinate. This research aims at understanding how the *Agile management principles* can further be blended with the *service development principles* and be incorporated throughout the lifecycle to focus on the collaboration and coordination in ISD management. Thus, this research proposes such a process - *Agile Process for Integrated Service Delivery (APISD).* Compared to traditional software development models such as Waterfall, Spiral and incremental development models, the APISD model introduces the iterative development from an earlier stage. This is because it is equally important to invest time and effort in proper requirement analysis and designing just as in development. Early iterative development allows adapting the changes flexibly compared to adapting them at a later stage. Moreover, incorporating the iteration allows the respective teams to work based on priority and produce usable artifacts in short periods of time. Furthermore, this model is different and extends from existing Agile development methodologies, because it envisions a wider focus at the entire lifecycle instead of focusing only on the development phase; methods such as Extreme Programming, Test Driven Development, Feature Driven Development and Scrum itself does [6].

To conduct this research, a design science research methodology has been employed as the research approach and case study research as a research strategy. The design science approach [7] was chosen since it addresses important problems that can be solved in an effective way with the help of an innovative artifact provided in this research. Case studies were investigated by reading reports and conducting three interviews with three organizations. Six steps have been followed, which are: Problem Identification and motivation, Definition of objectives, Design and development, Demonstration, Evaluation, and Communication [7]. The structure of this paper is as follows. Section 2 consists of a literature background on the concepts followed, section 3 consists of the derivation and description of the developed conceptual model and finally section 4 provides conclusions for this research and future work.

2 Literature Background

The following section briefly discusses the theoretical groundwork that was covered on the two concepts of ISD and Agile methodologies.

2.1 Integrated Service Delivery

When defining a service, there are many definitions that are based on technology or originate from the marketing literature. Some definitions are of electronic services, some thought of as web services, others are viewed as abstractions of business processes and some are considered to be an aggregation of other services [8]. Considering the various aspects surrounding the meanings of ‘service’, for this research, we contemplate the definition of service. [9], which is “*a series of*

interactions between the service provider and clients that result in an observable output".

As far as multiple service providers are concerned to provide the integrated services, clients perceive a bundle of services provided by various service providers as a whole and do not have to deal with each single provider. The essence of this problem of ISD is that these services need to be integrated; however, they are often heterogeneous and not designed for this purpose. Therefore, understanding the challenges faced in ISD, service characteristics and the process of developing these services in a structured manner is important. To develop the integrated services, service providers face a number of challenges which are related to organizational integration, embracing change and customer satisfaction. In the case of *organizational integration*, challenges include addition of staff working under different work processes, standards or different collective agreements in case of multiple organizations [2]. Therefore, there is a need of a common language and vision. For effective collaboration, it is important for parties to agree and to set common goals, establish common assumptions and build trust in the beginning of the development lifecycle. Effective communication, a shared understanding of roles and responsibilities, and a collaborative method of resolving issues are considered to be key factors in a successful partnership [2]. When concerning *embracing change*, the reality in ISD is about change and that change requires a certain level of risk. To deal with the risks and adapt to changes, working in this type of environment requires extensive communication and coordination of activities to manage those changes accordingly. By embracing change and integration, companies can innovate and advance rapidly [2]. As for *customer satisfaction* - ISD must be driven by a common desire to increase customer service. ISD partners should seek to satisfy stakeholders by determining how to meet their needs and then actually meeting them [2]. To be a customer-centered organization, the organization should consult the customers and other key stakeholders on an ongoing basis. As the nature of ISD is customer service oriented, not addressing to customer needs will cause organizations to lose the competitive advantage [2] and decline their growth in the market.

In this research, we have studied the service lifecycle of services suggested by several authors. The purpose of these service lifecycle models are either to introduce a new approach to deal with the lifecycle management, which consists of new roles and new development tasks as opposed to the ones of traditional software engineering[10], [11], or to deal with the heterogeneity challenge in platform specific or independent functionalities[12]. There are also several models of service lifecycles used by various companies and according to Gu and Lago [10], that covers the organizational process flow of a service lifecycle with a relation between stakeholders and service lifecycle stages. From the investigation, these models have allowed us to understand and follow a theoretical perspective of the service lifecycle provided by Gu and Lago [10] and the phases suggested by Papazoglou and Heuvel [11]. The lifecycle consists of three phases, *design, runtime* and *change* [10]*;* where design refers to the lifecycle of a service before it is available for use; runtime refers to when services are put into production and the implementations start to work; change focuses on the life cycle of a service when adjustments have to be made when business requirements change. Within these phases, sub-phases mentioned by Papazoglou and Heuvel [11] exist: *planning, analysis and design (A&D), construction*

and testing, provisioning, deployment, execution and monitoring. The roles involved throughout the service development are service provider, service broker and service consumer. These roles along with the phases were explored.

2.2 Agile-Scrum Methodology

Agile software development is a group of software development methodologies based on iterative and incremental development, which was termed and introduced by 'The Agile Manifesto' [13]. Some important characteristics of this manifesto are: (a) client satisfaction by rapid delivery of useful software; (b) welcome changing requirements; (c) working software is delivered frequently (weeks rather than months); (d) sustainable development, able to maintain a constant pace;(e) close, daily cooperation between business people and developers;(f) continuous attention to technical excellence and good design; and (e) regular adaptation to changing circumstances. One of the methodologies followed in the Agile software development is Scrum. The Scrum approach basically focuses on managing the system development process. It does not define specific software development techniques for implementation but rather concentrates on how team members should function to produce a system adaptively in a constantly changing environment. The characteristics of Scrum have been provided by Schwaber [14]. These are: flexible deliverables, flexible schedules, small teams, frequent reviews, inter and intra-collaboration, object oriented development. According to Schwaber and Beedle [15], the lifecycle consists of three phases: *Pre-game, Development and Postgame*. The roles involved in this lifecycle are: scrum master, product owner, scrum team, client, management and user; who were described in details.

3 Defining the Model

After understanding the characteristics and lifecycle of ISD and Scrum, we developed a conceptual model. The following section briefly elaborates on the model itself, how we evaluated it and finally how the model can be used in practice.

3.1 Model Construction

In order to construct the model we looked into the commonalities of ISD and Scrum. We tried to determine the phases for APISD by amalgamating the phases of ISD and Scrum creating a mapping between them. Similarly, the roles required were defined, which were required for APISD, and were inspired from ISD and Scrum. With the necessary components derived the model was developed. As shown in Figure 1, the model is a lifecycle consisting of six phases derived from the amalgamation of ISD and Scrum given in section 2.1 and 2.2: *Planning, Service Modeling, Service Construction, Provisioning, Deployment and Execution and Service Management.* Activities within each phase were described on how the process will be performed and focused on how to overcome the challenges. As explained in the Introduction, this model includes practices from the Agile principles but is different from other Agile development methods because first, it does not focus only on the development of the services but instead on the whole lifecycle and second, unlike the other methods, this method incorporates iterative development starting from the requirement analysis. The early iterative development allows the requirement analysis and the designing to be considered a development process

themselves in its nature, thus enhancing the clarification and the adaptation of changes at an early stage.

The *planning* phase consists of activities that allow businesses to analyze the business needs and market requests, and to determine the vision and objectives. With that knowledge, businesses are able to identify the type of services required and to be provided. The planning phase is carried out by the *service board*. The service board meets with the *client* and discusses various aspects of the services to be developed. The service board drafts a project document. They deliver this document to the *service analyst team* who will start with the analysis of the project. In the case of multiple service providers in serving the board, they also draft SLAs for their own governing responsibilities. This activity is crucial, because if the responsibilities and understanding between the parties are not addressed or agreed upon, several problems related to miscommunication, lack of ownership, and lack of coordination will arise throughout the lifecycle[11]. As a result, service providers will not be able to collaborate smoothly or gain trust, which is arguably required for sustainable development.

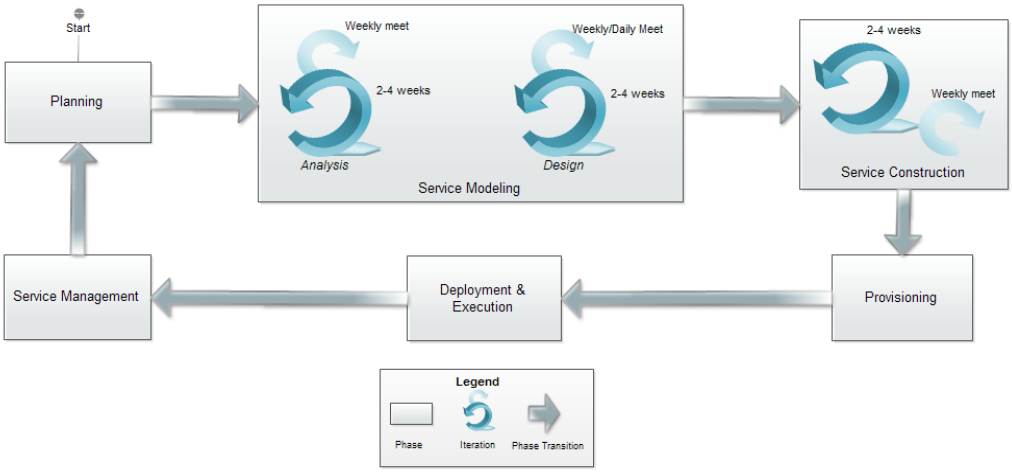

Figure. 1. APISD Lifecycle.

The main objective of the *service modeling* phase is primarily to describe the services identified in the planning phase consisting of two sub-phases, Analysis and Design. The team comprises of *service analysts* in Analysis, *development managers, and chief service developers* of the corresponding service providers associated in the project in Design. A *domain manager* exists which is appointed by the service modeling team who manages this phase to ensure that the different activities of analysis and design are aligned. The artifacts produced from this phase are: service backlog consisting of the services required and their requirements; feature backlog consisting of features derived from each service and their requirements and assigned to construction teams; and technical designs produced by the design team required for the implementation of the services. This whole process continues in separate iterations with the corresponding set of prioritized services. This phase is different from current practices because the iterative process begins at an earlier stage than the actual development. Moreover, the distinction between the requirement specification

and technical designing from the actual development of the services in an iterative form, allows adapting to changes flexibly and focusing on prioritized work rather than implementing at one attempt.

The *service construction* phase consists of the actual development and ongoing testing that Agile methods suggest. First, the construction team(s) (*service developers, service testers, development manager(s)*) of either a single service provider or multiple service providers views the feature backlog set for the first iteration which lasts for 2-4 weeks. According to the assigned features, services are developed and tested. Considering the scenario with multiple service providers, solving integration issues will require each organization's developer to communicate with each other and solve. By communicating with each other, they are able to gather knowledge (which promotes collective growth) and coordinate effectively to solve the issues. Once all issues are solved, a demonstration of the integrated services is given to the client by the *development manager*. By involving the client at this stage, the service provider is able to acknowledge their needs and ensure those needs are met, as a result satisfying the client. The development manager handles any conflicts between the development and test teams. In order to coordinate effectively among the teams, the development managers of each provider meet weekly. In this meeting, they discuss any impediments, dependency related issues and further planning of iterations. The *release manager* meets with the service board to discuss releases and finalizes them with the development manager. Towards the end, the development managers also arrange a retrospective meeting of their own to discuss results, lessons learned and improvement points. This whole process continues in iterations with the corresponding set of prioritized features and is implemented accordingly.

As soon as the service package is ready to be deployed, the *provisioning* phase deals with settling on the various rules and regulations surrounding the service delivery which are defined by the service board together with the client in the form of Service Level Agreement (SLA). This phase is required before making the services available to the client, because for effective collaboration between the service provider and the client, there needs to be an understanding and agreement regarding the usage and charges of the services. After the completion of the provisioning phase, the services are ready to be *deployed and executed*. The *system administrator* performs the necessary activities and deploys the system in the production environment. Once in production and used by the *users*, in the *service management* phase, the integrated services can be monitored and ensured that all the services are running according to the rules and regulations set in the SLAs. Regarding the management of the technicalities, the system administrator is responsible for configuring, managing and troubleshooting the servers. This phase also consists of change management which is very important so that changes are managed well in order to ensure a smooth operation of business; these changes are logged in a *change request backlog* which is later prioritized by the release manager for further planning and implementation. Changes are logged in by the *customer service*. They also report incidents in the *incident backlog* which are also prioritized by the release manager.

3.2 Evaluation of the Model

Following the constructs of case study research given by Yin [16], the model was applied to organizations that develop software providing integrated services and are looking for a faster, flexible and structured way to produce their products. For this

research three cases were explored. Two cases were performed on a single service provider scenario and the third case on a multiple service provider scenario. The multiple service provider and one of single provider have just commenced in following Scrum only in the development phase. The other single provider follows a Waterfall approach with moderations in the development phase. Semi-structured interviews were performed with a questionnaire where data was analyzed based on the answers provided in the questionnaire, interview discussion, audio recordings, website documentation and email correspondence.

In the interviews first the model was demonstrated to the organizations and evaluated with their current process. The comparison resulted into identifying differences between the two processes. From the analysis of the case study findings, additional factors were identified that have been used to enhance the model. As the *unit of analysis* is the implementation process to be investigated, which is a single unit of analysis, the case study takes towards a holistic view. Types of validity were looked at towards the case study findings. Construct validity was relevant because multiple sources of evidence were looked at providing a chain of evidence and further increased due to informants reviewing the draft case study report. Internal validity was irrelevant for this research because the nature of the case study was explorative instead of explanatory or causal. External validity was relevant because the cases were different and the model was replicated for all three which resulted in findings that can be generalized for other similar case studies. Finally, the reliability can be determined by following the case study procedure that was followed.

From the evaluation of the model, six key factors were identified and later appended in the model. (1) The service board was divided in two sub boards with a distinction in responsibilities serving a strategic and tactical nature, namely the *Executive Board* and *Service Board.* (2) The customer service was included in the Service Modeling phase to review the service backlog produced by the service analysts. (3) The system administrator was included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and to provide input. (4) In APISD after the construction phase, a high level product demonstration is given to the customer service and system administrator. This way, these roles are acknowledged of how the services work and can better support the service management. (5) In case of rejection by the client after post-production, an activity was required included in APISD for analyzing the problems and producing possible solutions. (6) A workflow was required for the service analysts to also visit the users' work-floor and observe their interaction and engagement with the integrated services. In comparison to the existing methodologies followed in the cases, they have identified some advantages of the APISD model which are: incorporation of iterative development earlier in the phases from construction; creation of a separate phase regarding *provisioning;* division of the design phase from construction phase allowing focus on architectural decisions; detailed description of roles and responsibilities explicating the collaboration between the parties involved in the process; and coordination of ongoing activities between the service analysts for requirement specification and construction teams for service development.

3.3 Illustration of the Model

With the case study research findings appended to the model, the model was finalized. In addition, an illustration of the model was given on how this model can be

practically implemented in an organization. Here, the implementation was based on two scenarios: for a single service provider and for multiple service providers collaborating together to deliver integrated services. Both scenarios were presented with a real life staging of the service provider and client and the type of services required. In each scenario, the activities within the APISD lifecycle phases were elaborated and the iterations were described on how team members can follow the iterations one after another maintaining synchronous information flow with other members. Examples were provided of how the service provider(s) collaborate(s) with the client and among themselves, what type of complications are faced and how they can deal with them using the constructs provided in the APISD model. Here, the concerns of the challenge of coordination were met by coordinating the following necessary activities in each of the scenarios: *decision points* were set- for example, once the service analysts produce the service backlog, only the service board decides and prioritizes the services. In the case of multiple-service providers, conflicts within the different teams are resolved by the development managers coming together to solve the dependencies or impediments; *change management processes*- to manage the changes, where a change backlog artifact is prioritized by the release manager, and implemented by the development team(s); an *issue management process* exists to manage the issues, where an incident backlog was introduced by which incidents reported by users are logged in, prioritized by the release manger and later implemented by the development team(s); *information sharing* was given importance, in order to have a consistent flow of information where teams are able to retrieve the requirements set in the service backlog and feature backlog and daily/weekly meetings were given within the iterations to share status and discuss impediments; finally, *performance review and monitoring* are portrayed- to monitor the progress, retrospective meetings were held by the company's development manager once the iterations are completed. These retrospective meetings comprises of lessons learned and identification of improvements to be implemented in future iterations. Moreover, other activities were detailed on what type of tools or artifacts the stakeholders can use while performing those activities. For example, in the scenario of multiple service providers, distributed teams can collaborate using virtual sharing tools such as TeamViewer or WebEx and designers can use collaborative diagramming using LucidChart. In order to fully understand the flow of activities among the different parties in the different phases, the implementation has also been demonstrated through sequence diagrams. Due to space limitations, these sequence diagrams are not part of this paper but can be looked at in [17]. These sequence diagrams provide a better understanding of how the various stakeholders interact with each other via the detailed activities per phase.

4 Conclusions and Future Work

The research objective was to provide an answer to the main research question on how Agile management and service development principles can be incorporated together for effective collaboration between parties and coordination of activities in Integrated Service Delivery. This was done by developing a conceptual model of Agile Process for Integrated Service Delivery (APISD).

With a literature analysis and from the author's experience, the model was developed to manage the heterogeneous services that are bundled to create integrated

services. The model portrays *multiple service provider collaboration* where common goals and assumptions are established to build trust in the beginning of the lifecycle: the Planning phase. A detailed description of roles and responsibilities were specified that can provide a shared understanding, enriching the collaboration between the parties. The model also allows *change adaptation*, due to the nature of the APISD model having iterative development, demonstrating that the changes occurring during requirement specification and designing can be easily adapted in the subsequent iterations. Finally, the model promotes *client collaboration*. APISD enables service provider(s) to have close interaction with the client from the beginning so that they a continuously focus on the main desire of ISD, increasing customer service by being customer-centric.

From the evaluation of the model, key findings from the comparison and commended parts of the model were: importance of a separate phase regarding *provisioning;* importance of division of the design phase from the construction phase; detailed description of roles and responsibilities explicating the collaboration between the parties involved in the process; incorporation of iterative development earlier in the phases from construction; coordination of ongoing activities between the service analysts for requirement specification and construction teams for service development.

Finalizing the model has enabled us to answer the main research question. Moreover, both the industrial and scientific community can acquire an insight on the perspective of managing Integrated Service Delivery with Agile practices. The industrial community can develop an understanding of the iterative perspective of the process and incorporate the activities to collaborate with stakeholders and coordinate accordingly. Furthermore, they can try to adapt the process within their organization and incorporate customized practices to benefit their needs. The scientific community can critically analyze the intention of this research, the challenges that are dealt with, the method this research was conducted in, the model itself and perceive an understanding of the research findings. From the critical analysis, they can try to empirically test the model and identify improvements that can make the model stronger to benefit the organizations in the management of their technologies. Moreover, with that comprehension, they can try to investigate other methods of conducting this research for more efficiency and effectiveness.

The current research has several limitations and opportunities exist for solidifying the model. First the model can be actually implemented within an organization and empirically conclude that this process will result in efficient collaboration between the parties involved and coordination of activities in the APISD lifecycle. From the usage of this model in various organizations, the practicalities within the APISD process can be refined. Opportunities exist for delving in the acceptability of this model after implementation in organizations through extensive case study research and investigating furthermore into the accountability and governing mechanisms. Moreover, for further research, it can be investigated in the future of how a better case selection can be made specific to the type and number of service providers working together and the industry they are in. The evaluation of such specific selection criteria will be beneficial to empirically conclude on the effectiveness of the model focusing on collaboration and can be further generalized to broader sense of applicability. Finally, further research is necessary on the investigation of the research questions

scientifically and it is recommended for researchers to publish new techniques and methods to implement this model and verify the effectiveness and efficiency of the implementation.

References

[1] L. M. Á. Sabucedo, L. E. A. Rifón, R. M. Pérez, and J. M. S. Gago, "Providing standard-oriented data models and interfaces to eGovernment services: A semantic-driven approach," *Computer Standards & Interfaces*, vol. 31, no. 5, pp. 1014-1027, Sep. 2009.

[2] ICCS, "Integrated Service Delivery: A critical analysis."Institute for Citizen-Centred Service, 2003.

[3] IAB, "Integrated Service Delivery Governments Using Technology to Serve the Citizen."1999.

[4] J. Sutherland, G. Schoonheim, N. Kumar, V. Pandey, and S. Vishal, "Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India," in *2009 Agile Conference*, Chicago, USA, 2009, pp. 277-282.

[5] B. Boehm, "Get ready for agile methods, with care," *Computer*, vol. 35, no. 1, pp. 64-69, undefined.

[6] P. Abrahamsson, O. Salo, and J. Ronakainen, *Agile software development methods : review and analysis*. Espoo [Finland]: VTT, 2002.

[7] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, p. 45–77, Dec. 2007.

[8] J. O'Sullivan, D. Edmond, and A. Ter Hofstede, "What's in a Service?," *Distributed and Parallel Databases*, vol. 12, p. 117–133, Sep. 2002.

[9] M. Janssen, A. Joha, and A. Zuurmond, "Simulation and animation for adopting shared services: Evaluating and comparing alternative arrangements," *Government Information Quarterly*, vol. 26, no. 1, pp. 15-24, 2009.

[10] Q. Gu and P. Lago, "A stakeholder-driven service life cycle model for SOA," in *2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting*, New York, NY, USA, 2007, p. 1–7.

[11] M. P. Papazoglou and W.-J. Van Den Heuvel, "Service-oriented design and development methodology," *International Journal of Web Engineering and Technology*, 2006. [Online]. Available: http://inderscience.metapress.com/app/home/contribution.asp?referrer=parent&backto=issue,7,7;journal,15,21;linkingpublicationresults,1:110898,1.

[12] L. W. F. Chaves, L. M. Sá de Souza, J. Müller, and J. Anke, "Service lifecycle management infrastructure for smart items," in *Proceedings of the international workshop on Middleware for sensor networks*, New York, NY, USA, 2006, p. 25–30.

[13] Agile Manifesto, "Manifesto for Agile Software Development," 2001. [Online]. Available: http://agilemanifesto.org/.

[14] K. Schwaber, "SCRUM Development Process," *Proceeding of the 10th annual ACM Conference on Object Oriented Programming Systems, Languages and Applications (OOPSLA)*, p. 117--134, 1995.

[15] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Prentice Hall PTR, 2001.

[16] R. Yin, *Case Study Research: Design and Methods, Third Edition, Applied Social Research Methods Series, Vol 5*. Sage Publications, Inc, 2002.

[17] M. Shammi, "Agile Process for Integrated Service Delivery," Delft University of Technology, 2011. Available: http://resolver.tudelft.nl/uuid:a3eab3e2-29c3-4d19-8208-6f12a7111d8c