Agile Process for Integrated Service Delivery

Marjana Shammi
AGILE PROCESS FOR INTEGRATED SERVICE DELIVERY (APISD)

THESIS
submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
MANAGEMENT OF TECHNOLOGY
by
MARJANA SHAMMI
Student ID: 4046277

Chairman: Prof. Dr. Yao-Hua Tan
Co-Chairman: Dr. ir. M.F.W.H.A. Janssen
First Supervisor: Dr. S.J. Overbeek
Second Supervisor: Dr. R.M. Verburg

TU Delft
Section of Information and Communication Technology
Faculty of Technology, Policy and Management
DELFt UNIVERSITY OF TECHNOLOGY
Delft, The Netherlands
August 2011
Preface

"Just as energy is the basis of life itself, and ideas the source of innovation, so is innovation the vital spark of all human change, improvement and progress" - Ted Levitt

Innovation: it has been the principle drive for this research, to bring forth a different perspective in the management of Integrated Service Delivery. The production of new ideas and implementations makes it possible for organizations to create competitive advantage and manage its technological fundamentals. This research is primarily focused on Integrated Service Delivery; however it is applicable for managing the development processes within an organization in general, in an iterative and incremental way. Technology management can be defined as the integrated planning, design, optimization, operation and control of technological products, processes and services. Therefore, this research is built on the foundations of this field and deals with such factors.

This thesis presents the work I have done for my graduation project at Delft University of Technology, which I performed during the academic year of 2010-2011 under the direct supervision of Sietse Overbeek. With this thesis, I will complete my master's degree program in Management of Technology at the TPM faculty of the TU Delft.

I am grateful to the people who have contributed their efforts in bringing this work together. Without the support of Delft University of Technology and organizations supporting to provide feedback for the case study research, this research would have not been possible. I, therefore, would like to thank Yao-Hua Tan, Marijn Janssen and Robert Verburg for their support and valued feedback. I especially want to express my gratitude to Sietse Overbeek, for his encouragement and time to keep me going in the right direction.

Last but absolutely not the least, my heartfelt thanks goes to my parents, Rob and Lipi, who have enlightened me throughout the years and for whom I was inspired to do my Masters; and to my husband, Adil, who has relentlessly kept me energized and motivated.

Marijana Shammi
Delft, The Netherlands
August 1, 2011
Executive Summary

The primary purpose of this study is to investigate “How can Agile management and service development principles be incorporated together for effective collaboration between parties and coordination of activities in Integrated Service Delivery?” In order to answer this question, the following sub-questions have been formulated:

- SQ1: What are the requirements for incorporating Agile management and service development principles in Integrated Service Delivery?
- SQ2: What is the artifact resulting from this research that represents the amalgamation of service development and Agile management principles to portray collaboration of parties involved and coordination of activities in Integrated Service Delivery?
- SQ 3. How is this artifact validated to evaluate the collaboration between parties and the coordination of activities in Integrated Service Delivery?
- SQ 4. How can this artifact be used in practice in management of Integrated Service Delivery?

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. As today's web is primarily designed on the basic foundation of user interaction, service providers who emerge as winners will be those who master the complex skills of customer intimacy and converged customer management. Companies seek to make use of the market competencies and deliver customer-centric services to gain the competitive advantage by providing a one-stop mall where customers can experience several kinds of integrated services. With the opportunity of Integrated Service Delivery (ISD), companies can support clients in an integrated environment possibly reducing cost and time. Doing so, service providers face a number of challenges related to organizational integration, resistance towards change and being customer-centric. With effective collaboration and coordination of activities, ISD can be managed efficiently. To support on-demand services, companies look for agility in their business for flexibility and adaptation to changing environments. Research has shown adoption of Agile methodologies has reduced complexities in software development and focused on collaboration and coordination to achieve performance gain. To the best of our knowledge, there has not been research on how to manage the service lifecycle of ISD in a holistic view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile 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. It is a guide to practitioners because currently, organizations are using the Agile methodologies but only in the construction phase. Through this research, they can further get an insight on 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).

To conduct this research, a design science research methodology has been employed as the research approach. The approach has been broken down to a number of steps, which are followed in each of the chapters in this thesis. A case study research has been conducted as the research strategy with data collection methods used such as interviews, documentation and questionnaire.

In order to answer the sub-questions, literature analysis, case study research, interviews, documentation and practical experiences were considered the knowledge and data sources. To answer the first sub-question, a literature survey has been done on exploring the theoretical aspects of ISD and Agile methodologies. Among the Agile methodologies, Scrum was chosen
as the appropriate method due to the fact that Scrum concentrates on how team members should function to produce a system adaptively in a constantly changing environment. Understanding the foundations of ISD and Scrum, the necessary requirements were identified. For each of these concepts, the lifecycle and involved roles were looked into. After following the literature survey, the necessary components to construct the initial model were defined. Next, to produce the artifact of this research, which SQ2 addresses, the initial conceptual model for APISD was developed and elaborated on its lifecycle, roles, and artifacts. The components within this model have been derived from literature analysis and author's practical experiences. To answer SQ3, a case study research was conducted to evaluate the initial conceptual model. Three cases were looked at in three different organizations to test the artifact. Each of them, different in their nature, has compared the APISD model with their current organizational process. Data was analyzed on the basis of differences in processes, missing factors and enhancements for the APISD model. The case study findings were validated with four validity tests: internal validity, reliability, external validity and construct validity. These findings have been incorporated in the initial model of APISD resulting in a final version of the model. The final model for APISD has been detailed upon and materialized in an illustrative scenario of how it can be implemented within an organization, which answers SQ4.

Finally, conclusions are drawn based on the research findings. Future researchers may wish to investigate on the recommendations provided at the end of this research for further enhancement of this model. Although, this research has only covered the development of the conceptual model, organizations interviewed have shown interest in implementation of this process within their own organization. This enlightens the relevance of this research and its importance in the industry for implementation.
# Table of Contents

**CHAPTER 1:** ................................................................. 1

1 INTRODUCTION AND PROBLEM DEFINITION ........................................... 2
  1.1 INTEGRATED SERVICE DELIVERY ...................................................... 3
  1.2 AGILE SCRUM PROCESS .................................................................. 3
  1.3 PROBLEM DEFINITION AND RESEARCH OBJECTIVE .............................. 3
  1.4 SCOPE AND ASSUMPTIONS ............................................................... 5
  1.5 THESIS OUTLINE .............................................................................. 6
  1.6 SUMMARY ......................................................................................... 7

**CHAPTER 2:** .............................................................................. 8

2 RESEARCH METHODOLOGY AND APPROACH .......................................... 9
  2.1 METHODOLOGY ............................................................................... 9
  2.2 RESEARCH APPROACH .................................................................. 10
  2.3 SUMMARY ......................................................................................... 13

**CHAPTER 3:** .............................................................................. 14

3 INTEGRATED SERVICE DELIVERY ....................................................... 15
  3.1 BACKGROUND ............................................................................... 15
  3.2 CHARACTERISTICS OF SERVICES .................................................. 17
  3.3 SERVICE LIFECYCLE ...................................................................... 18
  3.4 ROLES IN SERVICE DEVELOPMENT .............................................. 21
  3.5 SUMMARY ......................................................................................... 21

**CHAPTER 4:** .............................................................................. 23

4 AGILE METHODOLOGY - SCRUM ..................................................... 24
  4.1 BACKGROUND ............................................................................... 24
  4.2 CHARACTERISTICS OF SCRUM ...................................................... 25
  4.3 SCRUM LIFECYCLE ....................................................................... 26
  4.4 ROLES IN SCRUM ......................................................................... 27
  4.5 SUMMARY ......................................................................................... 27

**CHAPTER 5:** .............................................................................. 28

5 INITIAL CONCEPTUAL MODEL FOR APISD ........................................... 29
  5.1 DETERMINING THE NECESSARY COMPONENTS ............................... 29
  5.2 THE APISD INITIAL CONCEPTUAL MODEL ...................................... 34
  5.3 ARTIFACTS OF APISD ................................................................. 41
  5.4 SUMMARY ......................................................................................... 42

**CHAPTER 6:** .............................................................................. 43

6 EVALUATION OF APISD ............................................................. 44
  6.1 CASE SELECTION ......................................................................... 44
  6.2 CASE STUDY PROCEDURE ............................................................ 45
  6.3 VALIDITY OF THE CASE-STUDY FINDINGS ..................................... 46
  6.4 CASE LSP-A .................................................................................. 47
  6.5 CASE LSP-B .................................................................................. 51
  6.6 CASE PSP ...................................................................................... 55
  6.7 SYNOPSIS OF ANALYSES ............................................................. 59
  6.8 SUMMARY ......................................................................................... 60

**CHAPTER 7:** .............................................................................. 62

7 FINAL CONCEPTUAL MODEL FOR APISD ........................................... 63
# List of Figures

**Figure 1.** Thesis Outline ................................................................. 7  
**Figure 2.** DSRM Used for Constructing APISD .................................................. 11  
**Figure 3.** Chapters following DSRM Steps ..................................................... 12  
**Figure 4.** Scrum Lifecycle ............................................................................. 26  
**Figure 5.** Roles in Phases of Initial Conceptual Model ................................. 31  
**Figure 6.** Initial Model of APISD Lifecycle ..................................................... 34  
**Figure 7.** Planning Phase of Initial Model ..................................................... 35  
**Figure 8.** Service Modeling Phase of Initial Model ....................................... 36  
**Figure 9.** Service Construction Phase in Initial Model ................................. 38  
**Figure 10.** Provisioning Phase of Initial Model .............................................. 40  
**Figure 11.** Deployment & Execution Phase in Initial Model .......................... 41  
**Figure 12.** Service Management Phase in Initial Model ............................... 41  
**Figure 13.** Multiple-case Study Design .......................................................... 47  
**Figure 14.** Final Model of APISD Lifecycle .................................................... 66  
**Figure 15.** Roles in Phases of Final Model .................................................... 67  
**Figure 16.** Planning Phase of Final Model ...................................................... 70  
**Figure 17.** Service Modeling Phase of Final Model ....................................... 72  
**Figure 18.** Service Construction Phase of Final Model ............................... 74  
**Figure 19.** Provisioning Phase of Final Model .............................................. 76  
**Figure 20.** Deployment and Execution Phase of Final Model ....................... 77  
**Figure 21.** Service Management Phase of Final Model ............................... 78

**Appendix II Figure 1:** Planning Phase ......................................................... 108  
**Appendix II Figure 2:** Service Modeling Phase - Analysis ............................. 109  
**Appendix II Figure 3:** Service Modeling Phase - Design ................................ 110  
**Appendix II Figure 4:** Service Construction Phase - Development ................ 111  
**Appendix II Figure 5:** Service Construction Phase - Release Planning ........... 112  
**Appendix II Figure 6:** Provisioning Phase ................................................... 113  
**Appendix II Figure 7:** Deployment & Execution Phase ............................... 114  
**Appendix II Figure 8:** Service Management Phase ...................................... 115
Chapter 1

Introduction & Problem Definition
1 Introduction and Problem Definition

The ability to rapidly create, deliver and manage new and compelling customer-centric services has become more important than ever. Companies must become more customer-centric to survive the increasing competition: 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 (Accenture). As today's web is primarily designed on the basic foundation of user interaction, service providers who emerge as winners will be those who master the complex skills of customer intimacy and converged customer management. In recent years, service delivery platforms have emerged as a critical, strategic tool for service providers looking for an edge when it comes to driving faster speeds to market for new products and services (Accenture). Companies seek to make use of the market competencies and deliver customer-centric services to gain the competitive advantage by providing a one-stop mall where customers can experience several kinds of integrated services. Key elements to provide these integrated services to customers include service integration, organizational integration and collaboration (NWT, 2004). With the growing complexity and size of the software systems to enable these integrated services, Service-oriented Architecture (SOA) has become prevalent to improve integration among heterogeneous systems, to enhance reusability, and to increase verifiability (Erl, 2005). More and more, companies look to agility in building these systems to provide the services. As companies have come to depend increasingly on IT to interact with clients, execute financial transactions, and support business processes, the issue of business agility has become one of IT agility (Christopher & Towill, 2000). Here by agility, we mean the ability to quickly react to changing conditions (Vernadat, 2007).

IT service delivery models play a critical role in helping companies meet these challenges and through years, deliver great value. The current challenge is to build the agile interoperable enterprise systems, i.e., business entities that can work together, even if they belong to different legal entities, and that can be easily tailored to fast changing conditions (e.g., markets, economic conditions, political regulations, energy crisis) (Vernadat, 2007). Moreover, to build such systems in a short period of time with changing requirements, apt coordination and collaboration is required. In such a service structure, the development of dynamic, personalized and adaptable applications in a cost-effective and high-quality manner demands for a dedicated process which is proposed in this thesis as: Agile Process for Integrated Service Delivery - APISD. The process supports the delivery of integrated services by applying the service development and agile management principles on flexible, integrative, and reusable application building blocks. The process is a blend of the Service Lifecycle supporting service development and management, and Scrum, an Agile methodology that supports short iterative development and customer focused practices. The rationale for combining these two concepts is because they share some common characteristics and can be complemented by each other’s intentions; such as collaborating closely with the client, incorporating simplicity throughout the development process, ensuring sustainable development, adopting best practices welcoming change in requirements and dealing with them continuously. These characteristics are further explained in section 3.2 and section 4.2 respectively. The Service Lifecycle is incorporated in Integrated Service Delivery (ISD), which is briefly discussed in section 1.1 and is more elaborated in chapter 3. The Scrum Lifecycle is incorporated in the Agile Scrum process, which is briefly discussed in section 1.2 and is elaborated in chapter 4.

Thus, this research discusses the motivation of incorporating the service development and Agile management principles in ISD and a conceptual framework is developed to establish such a process. The research is conducted by following a design science research methodology, performing a literature survey and evaluating it with case study research as a research strategy. The organization of this chapter is as follows. First, a brief theoretical background will be given on ISD and Scrum. Following that, the research problem will be discussed with the main research question and sub questions. Third, the scope for this research and assumptions made will be defined. Finally, a thesis outline will be given mapping the thesis
chapters with the research questions, which will provide an insight what this thesis will consist of.

1.1 Integrated Service Delivery

Integrated Service Delivery concerns 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. With ISD concerning multiple service providers, clients perceive a bundle of services provided by various service providers as a whole and do not have to deal with each single provider. While developing integrated services, the alignment of multiple services and process structure is crucial to have the services running smoothly and fulfilling the business requirements, as well as meeting continuous client demands. Therefore, understanding the service characteristics and the process of developing these services in a structured manner is important. ISD is explained further in details in chapter 3.

1.2 Agile Scrum Process

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' (Agile Manifesto, 2001a). One of the methodologies followed in the Agile software development is Scrum. The Scrum approach is basically focused 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. Scrum is further explained in chapter 4.

1.3 Problem Definition and Research Objective

To compete effectively, companies need to respond quickly to changes in market conditions with innovative new products and services, so that they can maintain client satisfaction and expand market share. To the best of our knowledge, available service delivery models have tended to leave companies on their own -by requiring them to spend significant time and effort, shopping around for the right mix of services. With less time and fewer resources to expend today, companies seek greater efficiency and value from their IT service engagements. More and more, companies are adopting a pervasive mindset by taking service orientation into account, which is basically a way of thinking about things – that can help to transform IT responsiveness and agility (Holmes, 2005). Through integrated services, companies can support clients in an integrated environment arguably by reducing cost and time. With the opportunity of ISD, meeting client demands becomes more important and thus comprises of complex development processes. Public Sector Service Delivery Council (PSSDC) (2003) has undertaken as a survey to analyze existing ISD initiatives across Canada. Their analysis looks closely at the challenges faced by ISD practitioners across Canada. PSSDC has investigated and identified the most significant challenges in ISD which are themed in: overcoming organizational cultural differences, complexities in partnerships, resources, technology, resistance to change and change fatigue, leadership, citizen-centered services and marketing. The success factors in the ISD initiatives are themed in: visible and ongoing leadership, governance and accountability, relationship management, citizen-centered, culture, demonstrating value, communication campaign, and sufficient investment in technology. As previously mentioned in this chapter, the key elements to provide integrated services to customers include service integration, organizational integration and collaboration. In order to overcome the challenges related to organizational differences, complexities in partnerships, resistance to change, effective collaboration, adequate communication and coordination of activities are crucial factors. These challenges and factors will further be discussed in section 3.1.

The fact that companies want to meet the client demands and respond quickly to the changes in the market, necessitate flexibility and dynamicity in delivering the services. This
is also expressed by Ibbotson et al. (2007), who advocate the need for an agile SOA that is more dynamic and flexible than existing business implementations and postulates as such. In addition, Kroghal et al. (2005) discuss that the service model itself is one of the central aspects in SOA that drives Agile development. That central aspect can be mapped with the lean principles behind the Agile development. Along with flexibility and dynamicity, companies also seek efficiency and performance gain keeping the cost and time in mind (Ibbotson et al., 2007). In recent literatures, it is shown that Agile software development realized advantages such as, hyper-productivity and significant performance improvement and reduced the complexity of software development (Beavers, 2007; Sutherland et al., 2008; 2009). Adopting Agile methodologies has reduced complexities in development and has strongly focused on collaboration, communication and coordination to achieve performance gain (Sutherland et al., 2009), which arguably is beneficial to the management of delivering the integrated services. Moreover, service development and Agile methodologies share some commonalities in core principles that vindicate the reason for incorporating both. These commonalities will be explained in section 5.1.

With the motivation behind the adoption of Agile principles given above, there is indeed a need to combine the core principles of Agile management and service development and to create a framework that will help companies to overcome the challenges in managing the process of delivering integrated services. Here by complexities, we refer to the intricacies related to the collaboration and coordination of activities surrounding the different phases in the software development lifecycle in ISD. By collaboration, we refer to a process of participation through which people, groups, or organizations work together to achieve desired results with a shared vision and jointly build an interdependent system to serve targeted customers (Hunter). By coordination, we refer to a consciously organized relation between activities and forces (Hatchuel, 2001), work tasks are divided over actors and the act of making different agents (i.e. people or things) work together (Smite et al., 2010). There have been many studies conducted on the technological aspects of developing integrated services. Additionally, there have been several studies on the management of the development process of constructing the services. Moreover, there have been studies on the governance of multiple parties collaborating together to provide shared services. However, to the best of our knowledge, there has not been any research done on how to manage the service lifecycle of ISD in a holistic view focusing on the collaboration of parties involved in the process and coordination of activities by working in an Agile approach. Investigating on this aspect can provide an insight in the iterative perspective of the process and help companies to incorporate and benefit from the best practices out of it to effectively collaborate and coordinate. It is a guide to practitioners because currently, organizations are using the Agile methodologies but only in the construction phase. Through this research, they can further get an insight on 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 aspects of the management of ISD. Therefore, discussing the challenges in the beginning of this section, the main objective for this research is to support Integrated Service Delivery by incorporating the service development and Agile management principles in a structured process for effective collaboration and coordination. Finally, the desired aim is to reach key benefits by implementing this process in the organization where the benefits are:

- smooth collaboration and integration between service providers themselves and between service provider and client;
- better understanding and management of requirements;
- easy adaptation to changes;
- having an iterative and prioritized process of using and re-using services;
- handling implementation complexities in iterative development and meet demands of clients as needed;
Thereby, the main research question that is to be answered is:

\[
\begin{align*}
\text{How can Agile management and service development principles be incorporated together for effective collaboration between parties and coordination of activities in Integrated Service Delivery?}
\end{align*}
\]

As the main research question is formulated in a highly prescriptive manner, to get deeper and get a steering towards acquiring knowledge, there are four sub-research questions that are needed to be addressed. The first sub-question, SQ 1, is to understand the mechanisms of the two concepts, ISD and Agile methodology, Scrum, and investigate the requirements required to develop the initial conceptual model. SQ 2 aims to derive the artifact of this research, i.e., to develop the APISD model. SQ 3 addresses how this artifact is evaluated in this research. Finally, SQ 4 looks at how the artifact can be used in practice and contributes in supporting ISD. They are the following:

**SQ 1. What are the requirements for incorporating Agile management and service development principles in Integrated Service Delivery?**

**SQ 1.1 What are the challenges faced in Integrated Service Delivery and what are the characteristics, phases and roles that can be derived from theories on service development?**

**SQ 1.2 What are the characteristics, phases and roles that can be derived from Agile management?**

**SQ 2. What is the artifact resulting from this research that represents the amalgamation of service development and Agile management principles to portray collaboration of parties involved and coordination of activities in Integrated Service Delivery?**

**SQ 3. How is this artifact validated to evaluate the collaboration between parties and the coordination of activities in Integrated Service Delivery?**

**SQ 4. How can this artifact be used in practice in management of Integrated Service Delivery?**

By acquiring the required knowledge from the above sub-research questions, the main research question can be answered fully and understood how the Agile process can be incorporated in the ISD process. To obtain a visualization of where the questions are answered in this thesis, an outline is provided in section 1.5.

### 1.4 Scope and Assumptions

In order to proceed with the details of this research, some assumptions have been made that are relevant for this research and are listed in this section. Moreover, the scope is also defined based on what is covered in this research and not.

**Assumptions**

- This research is for organizations that do not have a stable process running but wants to incorporate one or have a stable process running and want to improve their process, overcoming the complexities of development in the changing conditions of the environment. It is not applicable for an organization that has dynamic processes running for dynamic service bundles.

- By the term 'integrated services', this research assumes focusing on the organizational process of building integrated services and delivering them and not on building web services and focusing on the technicalities of the construction.
In Scope

- The process developed in this research focuses only on the workflow of the different stakeholders involved in respective activities in an iterative development.
- This research focuses only in developing a final conceptual model for APISD. Case study research is used to evaluate the process in comparison to existing organizational process to integrate and deliver services.

Out of Scope

- This research does not focus on the technical management of developing integrated web services but rather, on the management of the process to realize ISD.
- Implementation of this process in an organization and evaluating it is not in scope of this research.
- Key Performance Indicators (KPIs) are not in scope of this research because, as mentioned in the previous point, it is not implemented in the organizations to identify KPIs and for the whole process it is difficult to determine as the phases vary in measuring the indicators. However, there are retrospectives that are done in the construction phase to measure the progress and improvements of the development teams.

1.5 Thesis Outline

![Thesis Outline Diagram]

Figure. 1. Thesis Outline

Figure. 1 illustrates the research questions mapping to the chapters/sections they are answered in. The thesis starts with chapter 1, where and introduction to this thesis was given and the problem was identified. From that, main research question was defined, which is shown by the arrow ‘h’. With the research approach followed in chapter 2 and shown in arrow ‘u’, the research question is answered. The arrow ‘s’ indicates that sub questions follow. For example, the main research question ‘Main RQ’, has sub-questions ‘SQ 1’, ‘SQ 2’, ‘SQ 3’ and ‘SQ 4’. The arrow ‘a’ indicates that the research question is answered in the corresponding chapter or section. Here, chapter 3 and chapter 4 discuss on the theoretical background of ISD and Scrum respectively. Chapter 6 answers the SQ 3 on the industrial impression and evaluation of APISD. The arrow ‘r’ signifies that the chapter results in developing the initial conceptual model for APISD which is chapter 5. This model is used for evaluation in chapter 6 which is denoted by arrow ‘u’. Following the evaluation, additional factors are identified and the final conceptual model is developed in chapter 7, denoted by arrow ‘?’. From that,
arrow 'u' signifies how the final APISD model can be used in practice. Finally, conclusions are drawn from several chapters portrayed by the arrow 'c'.

1.6 Summary

This introductory chapter inaugurated with a description regarding the need for delivering integrated services to gain competitive advantage where agility is needed for the flexible and dynamic service structure. The key challenges of delivering these integrated services were identified, which are related to organization integration and collaboration, change adaptation and bringing customer satisfaction. With the motivation behind the incorporation of Agile principles due to its given record of reducing development complexities and gaining increased performance by focussing on collaboration and coordination, it is argued that adopting Agile practices with service development can overcome these challenges and result in effective collaboration between involved parties and coordination of activities. Therefore, a brief introduction was given regarding the two main concepts of this research, Integrated Service Delivery (ISD) and the selected Agile Methodology, Scrum. The key objective was defined: to investigate 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 aspects of the management of ISD. The main research question to be investigated in this study was given with sub questions. Assumptions were made and the scope for this research has also been specified. Finally, the section ended with a thesis outline mapping the thesis chapters with the research questions, which will provide an insight what the rest of this thesis will consist of. The following chapter will elaborate on the research methodology followed in this study and an approach to conduct the study will be discussed.
Chapter 2

Research Methodology & Approach
2 Research Methodology and Approach

This chapter first provides the choice of research method for this proposed study and explains its nature. Following from that, the research approach will be discussed in reference to the theoretical model. In addition, the main information sources and data gathering techniques that are considered, will be explained. Elaborating on the research approach, the considered research strategy will be explained.

2.1 Methodology

A methodology is "a system of principles, practices, and procedures applied to a specific branch of knowledge (DMreview)." With support of such a methodology, it might help researchers to generate research that is valuable, rigorous and publishable in information systems research. For the development of APISD, the problem-centered approach of the design science research methodology (DSRM) presented by Peffers et al. (2007) is chosen while aligning the research with the seven guidelines for design science research by Hevner et al. (2004). A design science approach is 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.

"Design science...creates and evaluates IT artifacts intended to solve identified organizational problems." (Hevner et al., 2004) According to Hevner et al., it involves a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences and such artifacts may include constructs, models, methods, and instantiations (2004). March and Smith (1995) similarly also view design science as a prescriptive brand of research, which is focussed on rigorous building or examining artifacts. The aim is to serve human goals and contribute upon research outputs concerning guidance and expectations from design science outcomes. The methodological choices within the design science research process are much influenced by the guidelines provided by March and Smith (1995) and Hevner et al. (2004) (Peffers et al., 2007).

Thus, by following the guidelines, Peffers et al. (2007) has created DSRM, which includes the following six steps.

- **Problem identification and motivation.** This step defines the specific research problem and justifies the value of a solution. Since the problem definition will be used to develop an artifact that can effectively provide a solution, it may be useful to atomize the problem conceptually so that the solution can capture its complexity. Justifying the value of a solution accomplishes two things: it motivates the researcher and the audience of the research to pursue the solution and to accept the results and it helps to understand the reasoning associated with the researcher’s understanding of the problem. Resources required for this activity include knowledge of the state of the problem and the importance of its solution.

- **Definition of the objectives for a solution.** Inferring the objectives of a solution from the problem definition and knowledge of what is possible and feasible is the aim of this step. The objectives can be quantitative, e.g., terms in which a desirable solution would be better than current ones, or qualitative, e.g., a description of how a new artifact is expected to support solutions to problems not hitherto addressed. The objectives should be inferred rationally from the problem specification. Resources required for this include knowledge of the state of problems and current solutions, if any, and their efficacy.

- **Design and Development.** This step focuses on creating the artifact. Such artifacts are potentially constructs, models, methods, or instantiations (each defined broadly) (Hevner et al. 2004). Conceptually, a design research artifact can be any designed object in which a research contribution is embedded in the design. This activity
includes determining the artifact's desired functionality and its architecture and then creating the actual artifact. Resources required moving from objectives to design and development include knowledge of theory that can be brought to bear in a solution.

- Demonstration. This step is about demonstrating the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity. Resources required for the demonstration include effective knowledge of how to use the artifact to solve the problem.

- Evaluation. Observing and measuring how well the artifact supports a solution to the problem is the objective of evaluation. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration. It requires knowledge of relevant metrics and analysis techniques. Depending on the nature of the problem venue and the artifact, evaluation could take many forms. It could include such items as a comparison of the artifact's functionality with the solution objectives from activity two above, objective quantitative performance measures, such as budgets or items produced, the results of satisfaction surveys, client feedback, or simulations. It could include quantifiable measures of system performance, such as response time or availability. Conceptually, such evaluation could include any appropriate empirical evidence or logical proof. At the end of this activity the researchers can decide whether to iterate back to step three to try to improve the effectiveness of the artifact or to continue on to communication and leave further improvement to subsequent projects. The nature of the research venue may dictate whether such iteration is feasible or not.

- Communication. The final step is to communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences, such as practicing professionals, when appropriate. In scholarly research publications, researchers might use the structure of this process to structure the paper, just as the nominal structure of an empirical research process (problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion) is a common structure for empirical research papers. Communication requires knowledge of the disciplinary culture.

2.2 Research Approach

The approach, in which this research is conducted, is illustrated in Figure 2. In this section, using the DSRM steps mentioned in previously, the techniques used and the associated activities that are performed in this research are given and discussed in detail. We have already explored the first two DSRM steps: Problem Identification and Research Objective Definition in section 1.3; after identifying the problem and objective of this research. In this section, we will discuss on the rest of the DSRM steps. In addition, an illustration of the steps mapped to the chapters/sections in the thesis is given. Afterwards, the research strategy chosen for this research is elaborated.
— Design and Development: In order to derive the requirements for developing the model, the theory behind ISD and Agile processes, specifically Scrum, will be looked at through literature analysis. In order to retrieve information, it is important to recognize the appropriate sources first, both data and knowledge sources. For knowledge sources, information will be retrieved from books, articles where previous background exists. In addition, literature will be used as a knowledge source for theoretical insights and as a data source to understand the objective descriptions of the practices in reality. Moreover, practical experience of author also will provide as a knowledge source in order to describe the activities defined within the model. Techniques for information retrieval are primarily through library research that is based mainly on review of existing literature and literature analysis. Focal areas of both consist of the nature of the definition of concept, significant characteristics, the lifecycle involved around them with the various phases and the contributing roles that perform a set of activities. After analyzing both individually, the common attributes will be investigated. Blending in the common factors together, a set of phases and roles will be defined, serving as the requirements for developing the model. With the literature analysis and author’s practical experience, an initial conceptual model, which is the artifact of this research, will be developed and further defined.

— Demonstration: The process will be demonstrated through the use of case studies. The case studies will be selected based on a set of criteria that fits the requirements of the process. The purpose of the case studies is to demonstrate the applicability and generalizability of the defined process as well as to analyze if further enhancements can be made to the initial conceptual model. The techniques used for case study setup will be done by reviewing literature and analyzing organizational data provided by companies. In order to acquire the necessary information regarding the process, the sources needed are individuals who will be interviewed to contribute to testing and
solidifying this model. The data collection techniques used are: questionnaire, semi-structured interviews, audio-recordings, website documentation and email correspondence.

- **Evaluation**: The Evaluation phase consists of two main objectives. The first objective is to test the model with the case study, as explained in the Demonstration phase, and analyze the data based on the missing criteria and differences between the artifact and the organization’s process. The second objective is to analyze those data and derive additional factors to expand the initial conceptual artifact and result in a final artifact. Following the identification of these categorical data, iteration to two or three steps back of the DSRM steps will be done. The reason for this iteration is to revisit the original research questions and alignment, and enhance the artifact or in this case, the model. Moderations will be done to the expansion of the initial conceptual model and a final model will be concluded with its necessary explanations. The final model will also contain highlights of the additional factors derived in the evaluation to distinguish the moderations.

- **Communication**: Once the model has been evaluated and a final conceptual model is derived, the final step is to communicate on the practice of the model. This will be done by an illustration of the artifact on how the model will work within the organization. Finally, conclusions will entail communicating the importance and novelty of the model, and its effectiveness to other research and future work.

The following figure illustrates the mapping of the DSRM steps to the chapters of the thesis where the steps are followed.

![Figure 3. Chapters following DSRM steps](image)

**Research strategy**

In management science and related disciplines with practice-oriented research, the research strategy that is selective in nature and qualitative, is case study research (Verschuren & Dooreward, 1999). This would be an appropriate approach to gain in-depth insights in the
proposed framework provided in this research. Other reasons for choosing this strategy compared to other, are because it would allow an open observation and provide depth in the knowledge retrieved from the small domain in which the data can be analysed more closely. This strategy though has a drawback, which is that the external validity of the results is limited. This is because the fewer cases studied, also needed to get depth knowledge, the more difficult it is to apply the results to a broader population (Verschuren & Doorward, 1999). Despite the drawback, this strategy is much more flexible compared to survey or experiment and it offers to obtain a general picture of the research.

Case studies are most useful to perform research to answer 'how' and 'why' questions, which do not require control over behavioural events, and which focus on contemporary events. When performing a research in a practical context, case studies are preferred and are performed when considering purposes of identifying phenomena in practice and observing causal relationships and checking whether certain theories are correct (Yin, 2002). The validity concerning the case study findings will be discussed more in detail in chapter 6.

2.3 Summary

In this chapter, first the design science research methodology was introduced and discussed in details. The rationale behind the selection of this for choosing this methodology is because it addresses important problems that can be solved in an effective way with the help of an innovative artefact, the APISD model, which is provided in this research. Based on the DSRM, the steps to be followed were discussed in details. The main information sources and data gathering techniques that were considered are as follows: literature review, website documentation, digital documents and interviews. In addition, the steps of the research approach have been mapped with the corresponding chapters they are followed in this thesis. Finally, the selected research strategy was specified, which is a case study research. To develop an initial conceptual model, the next two chapters will provide a theoretical background on the two concepts acting as the foundation of this research. Therefore, the following chapter discusses on ISD, the service lifecycle and its components.
Chapter 3

Integrated Service Delivery
3 Integrated Service Delivery

To understand the concepts behind this study and to address the research area, a background of ISD based on various literatures is described in this chapter. Recall from the previous chapter that one of the steps in DSRM is designing of artifact (shown in Figure 2). In order to design this artifact, the theoretical basis needs to be understood. Therefore, to grasp one of the foundations of this research, in this chapter first we investigate the definition of service and ISD. Then we analyze the challenges faced in ISD, which are relation of organizational integration, change adaptation and customer satisfaction. To understand how to overcome these challenges, we first try to understand the basic characteristics of services. Then, in order to comprehend the development process of services, the various phases of the service lifecycle will be discussed along with the different roles involved in the final sections.

3.1 Background

When defining a service, there are many definitions that are based on technology. 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 (O’Sullivan et al., 2002). In terms of web services, a service can also be termed as a discoverable software resource which has an advertised service description available in a repository, called a registry (Sabucedo et al., 2009). Thus, a service can be invoked, published and discovered. In economics, a service is defined as “activities, benefits and satisfactions, which are offered for sale or are provided in connection with the sale of goods” (Committee on Definitions of the American Marketing Association, 1960). Keeping the concept in mind, a service can be thought of as something that encapsulates a business process or an information area in the business (Rossberg & Redler, 2005), which can be provided to serve clients. The key is that the user of the service, the consumer, does not have to know what goes on behind as the processing of the service is hidden from the consumer, and the consumer only requires knowing how to call the service (Rossberg & Redler, 2005). With respect to the above definition by Rossberg and Redler, we consider another definition of service by Janssen et al. (2009) for this research, which is “a series of interactions between the service provider and clients that result in an observable output”.

A service can be assembled by combining a number of existing services or new services. These multiple services can be delivered by a single service provider or by various service providers collaborating together. According to Sabucedo et al. (2009), Integrated Service Delivery or ISD concerns a bundle of services offered by more than one service provider that matches variable client needs and deals with environmental changes. Intergovernmental Advisory Board defines ISD as providing an interface accessible to the citizens where they could obtain multiple services via a single access ‘window’ (1999). A similar definition by the Institute for Citizen-Centered Service (ICCS) is that ISD is a single-window service, one-stop delivery, one-stop access, one-stop shopping, service clustering, multiple service delivery channels and overall an innovative way of delivering services to citizens (2003). ICCS also mentions that sometimes successful integrated service delivery often requires organizations to partner or amalgamate. Therefore, with respect to the mentioned definitions, we consider ISD 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’. When considering multiple service providers, clients perceive a bundle of services provided by various service providers as a whole and do not have to deal with each single provider. This concept is further explored by Janssen et al. (2009) with respect to ‘shared services’, which is defined as “the concentration of dispersed service provisioning activities in a single organizational entity”. They have developed simulation models to evaluate the impact of shared services and show that shared services can result in improved efficiency and service levels. Other literatures discuss the aspects surrounding shared services. Turle (2010) examines the legal issues thrown up by shared services and how to manage them. Grant et al. (2007) explore the options and issues to consider when selecting and implementing shared services governance including accountability, management of resources and of day-to-day
operations. As developing integrated services requires a shared understanding among service providers, there exists an uncertainty that they may not understand each other and are unable to realize ISD. To solve this, foundations for an ISD language have been proposed and rules have been specified for enabling ISD (Overbeek et al., 2010). Other companies build up their own models when considering integrated services. BT and HP (2006) have jointly developed a service model with which they tackle the combined level of services that is required to support the business. They mention that the difficulty lies in defining the level of service from each supplier in such a way that when are brought in together, it results in a combined level of service that is required to support the business process. Alignment of multiple services when developing and process structure is crucial to have a smooth running orchestration of the services while meeting continuous client demands. Here, in case of integrated services, service orchestration and composition is important. Service composition involves the development of customized services often by discovering, integrating, and executing existing services. With discovered and composed services, all the services can be integrated together to fulfill the business requirements, which is termed as service orchestration (Gu & Lago, 2007).

When attempting to develop the services, a Service-Oriented-Architecture (SOA) is used. SOA is about designing and building IT systems using heterogeneous network addressable software components that can be directly invoked by business users or executed as steps of business processes (Vernadat, 2007). They can be combined, modified, or reused quickly to meet business needs. In order to build SOA, there are many mechanisms that are used. Produced by the World Wide Web Consortium (W3C), web services are one mechanism that are a set of standards using standard protocols and exchange formats to perform functions or execute business processes and accessing interfaces (Holmes, 2005; Vernadat, 2007). But web services are not a must for SOA where other means can be used to expose services, such as - message queues, where the sender and receiver of the message do not need to interact with the message queue at the same time where the message is stored in the queue until the receiver picks it up (Esl, 2005). Other technologies that can be used for implementing SOA are: Simple Object Access Protocol or SOAP – which is a protocol specification for exchanging structured information in the implementation of web services (Curbera et al., 2002); Remote Procedure Call or RPC – which is an inter-process communication that allows a program to cause a procedure to execute in another address space without specifically coding the details for this remote interaction (Birrell & Nelson, 1984): Representational State Transfer or REST – which is a style of architecture where requests made by client system and responses given by server system are built around a resource, whose state is captured by a document (Jakl); and Windows Communication Foundation or WCF which is application programming interface used for building connected service oriented applications (Mackey, 2010). For multiple service providers delivering integrated services and requiring service orchestration, Business Process Execution Language for Web Services (BPEL4WS or BPEL) can be used. BPEL is a standard business process language used by several application server vendors (Vernadat, 2007). The role of BPEL is to define the web service interactions between two parties participating in a business process and generating a baseline implementation of those interactions (Microsoft Corp, 2004).

To develop the integrated services, service providers face a number of challenges, which have been introduced in section 1.3. The challenges are related to the following:

- Organizational integration: Successful integrated service delivery, which in the cases of different departments working together in one company or multiple organizations collaborating together, require amalgamation. Challenges include addition of staff working under different work processes, standards or different collective agreements in case of multiple organizations (ICCS, 2003). 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. It is also important that partners responsible for the maintenance operation of an ISD initiative once it is up and running, be
identified and agreed upon in the provisioning stage. Effective communication, a shared understanding of roles and responsibilities, and a collaborative method of resolving issues are considered to be key factors to a successful partnership (ICCS, 2003). Furthermore ICCS also mention, to ensure successful partnership discussion, it is important to discuss on topics, which are: decision-making protocols, change management processes, issue management processes, information sharing protocols, timelines, and performance review and monitoring.

— Embracing change: The reality is 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. In one case study, it was noted that many managers felt a general lack of comfort when dealing with things outside their particular span of control although having a clear vision. By embracing change and integration, companies can innovate and advance rapidly (ICCS, 2003).

— 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 (ICCS, 2003). 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 and decline their growth in the market.

Thus, to overcome the above challenges, organizations need to effectively collaborate to establish a shared understanding and coordinate their activities by setting processes. Doing so, the service providers will be able to adapt to change in a flexible way and meet the client demands accordingly.

3.2 Characteristics of Services

Service-oriented design and development is based on an iterative and incremental process (Papazoglou & Van Den Heuvel, 2006). According to the authors, this approach is one of continuous invention, discovery and implementation with each iteration. In order to understand services and SOA, some basic characteristics are needed to be looked at. Rosberg and Redler (2005) discuss such characteristics cited from different other authors in their book:

— Abstracted: a service is abstracted from its implementation where a consumer should not have to worry how the service is implemented but only how to invoke it.
— Published: the implementation should be hidden by a precise published specification of the service functionality.
— Formal: a formal contract between provider and consumer should exist.
— Relevant: a service should be presented with the functionality at a level of understanding that could be recognized by the user.
— Reusable: services should be able to be reused.
— Autonomous: a service must always assume that nobody is in control of the entire system. There could be individual services that are deployed as a unit but the whole system is dispersed.

Grönroos (2001) has identified three main characteristics of services. As services are processes that consist of series of activities rather than things, often service providers and their clients have to interact frequently in the course of the service delivery process. Second, services are, at least to some extent, produced and consumed simultaneously, and third, customers in one way or another participate in the service delivery process.

With integrated services, other characteristics (Papazoglou & Van Den Heuvel, 2006) also have to be taken into account such as:
— the entire service lifecycle should be manageable and maintained;
— a platform and programming model should be established which includes connecting, deploying and managing services within a specific runtime platform;
— adoption of best practices for architecting service-oriented solutions in ways that deal with changing business needs;
— implementing solutions with security, compliance and standards for interoperability and designing for change;
— interfaces should be implemented to the capabilities which have to be composed in a logical way, based on business requirements using standards such as the Web Service Business Process Execution Language (WS-BPEL)\(^{(\text{Kaponig})}\);
— creating leverage of existing investments by allowing existing applications to be encapsulated as services \(^{(\text{Ibbotson et al.}, 2007)}\);

### 3.3 Service Lifecycle

There are several models of service lifecycles used by various companies and according to Gu and Lago (2007), there is no commonly agreed service lifecycle model in the literature. To the best of our knowledge, most service lifecycles are related to the management of web services and not the management of the process itself. Bianchini et al. (2009) propose a methodology to support the designing of a business process in the identification of services that compose the process itself, where the collaboration should be facilitated, by guaranteeing a homogeneous description of services at the right level of granularity. The methodology proposes the following phases: **semantic process annotation**-with concepts extracted from shared ontologies to figure out semantic similarities between terms for an uniform adopted terminology; **identification of candidate services** - to determine a first set of candidate component services to figure out service invocations and value exchanges between requesters and providers; **evaluation of service cohesion/coupling** - to better define service structure and granularity; and **refinement of process decomposition** - to detect multiple invocations of the same service throughout the process workflow by means of proper co-efficients. Such phases support the business process designer in the identification of component services. Overbeek et al. (2009) investigated the possibility of using events to coordinate demand-driven services across a network of organizations by developing an event-driven, service-oriented architecture (EDSOA). They provided an ontology which provides the basis for orchestrating events between services provided by public and private organizations that adapt the ontology. By sharing this ontology, complex software applications can meaningfully communicate to exchange data and thus make such data transactions interoperate independently of their internal technologies.

Another study by Chaves et al. (2006) proposed an infrastructure for service lifecycle management of smart items technologies such as sensor networks, Radio-frequency identification tags or embedded systems. They adopted the following service lifecycle management tasks: **service registration** (in directory), **service discovery** (according to specific characteristics), **service deployment** (plan for deployment and installation of services), and **service removal** (removal of unused services). Similarly, Autili et al. (2008) presents the work on service lifecycle management carried out within projects of the Mobile Services Platform cluster but the activities are related to the technicalities of managing the services rather than process flows. The cluster covers all work providing elements of platforms, which facilitate the development and deployment of mobile services. The lifecycle is divided in two aspects. First, design-time aspects, which include **service modeling, development and packaging**, Second, run-time aspects includes, among others, the **deployment, discovery and execution of these services**. Other studies have taken a perspective towards the organizational side when managing services but still have not covered the theoretical aspects of the whole process. Delgado et al. (2010) proposed a framework, which applies Model Driven Development (MDD) and Service Oriented Computing (SOC) paradigms to business processes for the continuous business process improvement in organizations and supporting the different stages defined in the business process life cycle from modeling to evaluation of its execution. In the framework, they have defined a number of elements or activities required to fulfill the needs.
These elements or activities are: Assess the target organization (to obtain the organization map, its processes and technologies), Identify Business Processes (to model business processes, flow, involved roles, associated functionalities), Identify and categorize services (to define and classify the services to carry out the business processes and functionalities), Specify services (to define contract of services, operations, parameters, etc.), Investigate existing services (to reuse services, components, implemented functionalities), Assign services to components (to define service implementation), Define services interaction (to define sequences of invocation of services to carry out the business processes), and Implement services (to build the services as they were designed). Another study by Schepers et al. (2008), in which a lifecycle based approach for executing SOA governance is identified, which consists of defining a SOA strategy, aligning the organization, managing the service portfolio, controlling the service lifecycle, enforcing policies and managing service levels. In another research, Overbeek et al. (2011), proposes a governance method to identify possible life cycle related scenarios when trying to integrate web services and to coordinate and track changes in those service life cycles. When attempting to realize an integrated service, the method helps to determine which of the scenarios a service provider is confronted with. These scenarios are: (1) two different services are in the same phase of their life cycles and the life cycles are also applications of the same methodology (2) two different services are in different phases of their life cycles and the life cycles are again applications of the same methodology (3) two different services are in different, but semantically similar phases of their life cycles that are based on two different methodologies (4) two different services are in semantically distinct phases of life cycles that are based on two different methodologies.

From the above analysis of various studies, it is seen that there is no commonly agreed service lifecycle model in literature that covers the organizational process flow of service lifecycle with a relation between stakeholders and service lifecycle stages. With reference to some service lifecycle models, Gu and Lago (2007) have proposed the lifecycle in three phases, design, runtime and change. Within these phases certain sub-phases or activities are mentioned, which are service lifecycle phases themselves defined by Papazoglou and Heuvel (2006). In the lifecycle proposed by Gu and Lago (2007), they refer design time to the lifecycle of a service before it is available for use. During the runtime stage, services are put into production and the implementations start to work. The change time stage comes after runtime. It focuses on the life cycle of a service when adjustments have to be made when business requirements change. They have followed the phases according to the service-oriented design development methodology defined by Papazoglou and Heuvel (2006) where the lifecycle is divided into eight phases, namely, planning, analysis and design (APSD), construction and testing, provisioning, deployment, execution and monitoring. In order to develop the necessary components for APISD, it is important to understand the span of the service lifecycle as a whole and the interactions between the stakeholders involved in the service lifecycle phases. Therefore, in terms of understanding the service lifecycle, the high level phases defined by Gu and Lago (2007) will be followed but still taking into account the sub phases similar to those of Papazoglou and Heuvel (2006). Thus, the following phases and sub phases are described below running in an iterative process, which are derived from the two sources mentioned above.

— Design

  o Market scan and planning: It is important to understand the business environment and analyzing the business needs and market requests. It is also needed to determine the feasibility, nature, and scope of service-solutions in the context of an enterprise. In this sub-phase, reviewing the market trends and analyzing the financial cost and benefits, it will provide service provider’s insights to understand and decide on which services to produce.

  o Requirements Engineering: Business goals and objectives need to be reviewed to analyze potential return on investment. Once the types of services are decided upon, requirements can be defined. Upon deciding on the services that will be produced, service providers can examine the
existing service portfolio to check which can be reused or requires to be newly implemented.

- Business Modeling: “After the requirements engineering is ready, service providers have a clear goal about what services they are going to produce and what are their target markets. The next step is to model the business process. During this phase, service providers have to capture all the requirements gathered in the requirements engineering activity and model business processes into low level processes which can be defined without going into the technical details. Usually business analysts participate only in this activity. Since services are actually units of business functionalities, adequate business modeling is therefore crucial to guarantee services can be well implemented by IT developers without requiring deep knowledge on business processes.” (Gu & Lago, 2007)

- Service design: After the processes are modeled, services can be designed that comply with all the functional and non-functional requirements. While designing the new services, it is important to keep into account whether the existing services, if existing, can be reused or modified. The various rules and policies that concern the different services should be included in the design for the developers to implement them. Furthermore, the different interfaces of the services should also be designed so that external services can connect.

- Service Development and Testing: Once the design is completed, development can be commenced in terms of coding and testing. Here, for the development of services, a SOA can be used to support the development of services using web services (see section 3.1). Integration of services can be further tested to check if all requirements are met and are according to the quality standards.

- Runtime

- Service Publishing: Once development and testing of the services are completed, the services should be ready to be published to a service registry and to become available to service consumers.

- Service Provision: In this sub-phase, before the services are actually executed, the rules for invoking services in terms of billing, service governance, service certification and service accessibility should be defined. More details on these activities can be found in (Papazoglou & Van Den Heuvel, 2006). Such rules can be defined in Service Level Agreement (SLA) for the various organizations involved. Once the SLAs are signed, it enables the authorization to use the services deployed.

- Service Deployment: Once the SLAs are agreed upon, services are ready for deployment and executed. Services can be discovered either manually or dynamically by matching its characteristics with the service interface descriptions. After service orchestration, the services can be executed and are ready for access.

- Service Monitoring: Following the publishing and execution of the services, it is important to monitor the services to ensure that the services are running according to the rules and regulations set in the SLAs. Moreover,
the monitoring will also help to analyze the availability, reliability and performance of the services and improve them if required.

---

**Change**

- **Service Management:** As much as it is important to develop the services and execute them, it is also crucial to ensure that the services are maintained and managed for successful business operation. If client requests change, the previous phases need to be followed and ensured that the applications are maintained smoothly without causing any interruption of the existing services. Here, the service registry needs to be maintained so that updates of any services do not break the invocation of existing services. Overall, the main objective of this phase is to manage the services well when changes occur and impact is minimal.

### 3.4 Roles in Service development

Following SOA, there are three roles/stakeholders associated: service provider, service consumer and service broker (Gu & Lago, 2007). These roles are further elaborated below according to the descriptions provided by Gu and Lago.

---

**Service Provider:**

A service provider is a development party who is the owner of services and responsible for implementing them as well as maintaining them. Their activities are:

- gathering requirements in the market and scanning phase;
- analyzing the objective, functionality, interface and quality of services in the requirement engineering phase;
- capturing the requirements and modeling of business processes into low level processes;
- designing the services, developing and testing them. Once done, they make them available to publish them;
- defining SLAs with service consumers;
- monitoring services for performance and maintaining them in the future.

---

**Service Broker:**

A service provider is an intermediary role acting between a service provider and a service consumer whose main role is to provide service location information which is contained in a service registry. The broker uses the registry to publish the services of service providers and service consumers use it to look up services. A service broker is also responsible for updating the registry with correct information and maintains it well so that any updates do not break the invocation of existing services.

---

**Service Consumer:**

A service consumer is a party that integrates services into an application which eventually fulfill the requirements of the end user. The party also tries to find services in the service industry and execute them. During service provisioning, a service consumer negotiates with a service provider in order to reach agreements regarding the access control of services set by the service provider. A service consumer can also monitor the services and check with the policies set in the SLAs, whether they meet or not.

### 3.5 Summary

In this chapter, a theoretical background of ISD was covered through a literature survey. First, we have tried to look into various definitions of the term 'service' and 'ISD' and have agreed upon the considered definition for this research. Then we have tried to explore the challenges of ISD that are to be reflected upon in the rest of this thesis. In order to understand the nature of service development in ISD, the main characteristics of services were looked at. In addition, to develop a conceptual model for APISD, the service lifecycle was
required to be understood. Thus, the various phases within the service lifecycle were elaborated. As any process requires identification of the stakeholders involved in it, the key roles in service development were delved into. The following chapter discusses on the second foundation of this research, Agile methodology - Scrum, which is required for constructing the conceptual model.
Chapter 4

Agile Methodology-Scrum
4 Agile Methodology - Scrum

In the previous chapter, we have discussed the concepts behind ISD. In order to elaborate on the DSRM step 'Design of Artifact', it is also important to understand the concepts behind the Agile Scrum process (shown in Figure 2). 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' (Agile Manifesto, 2001a). This section will entail a theoretical groundwork of the Agile process in general and delve further into the basics of Scrum, a development process based on the Agile principles that is selected for this research. Following this, the characteristics of the Scrum process will be discussed. The final sections will be elaborated on the Scrum lifecycle and the various roles involved in it.

4.1 Background

In Agile software development, requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. The main valued aspects in this lightweight concept compared to the traditional methods are: individuals and interactions over processes and tools, working software over comprehensive documentation, client collaboration over contract negotiation, and responding to change over following a plan. There are different understandings and perceptions on the term 'Agile'. Abbas et al. (2008) (2008) have discussed on the origins of the agile methods and discuss the reasons for it. However, the main principles of the Agile concept remain constant (Agile Manifesto, 2001b), which are the following:

- Client satisfaction by rapid delivery of useful software;
- Welcome changing requirements, even late in development;
- Working software is delivered frequently (weeks rather than months);
- Working software is the principal measure of progress;
- Sustainable development, able to maintain a constant pace;
- Close, daily cooperation between business people and developers;
- Face-to-face conversation is the best form of communication (co-location);
- Projects are built around motivated individuals, who should be trusted;
- Continuous attention to technical excellence and good design;
- Simplicity;
- Self-organizing teams;
- Regular adaptation to changing circumstances.

This development framework consists of several methodologies such as Extreme Programming (XP), Scrum and Test Driven development (Lindvall et al., 2002; Stober & Hansmann, 2010). Others methodologies are: Dynamic Systems Development Method (DSDM), Crystal family of methodologies, Rational Unified Process (RUP) and Adaptive Software Development (ASD). Abrahamsson et al. (2002) compare the different methodologies and lists their strengths and shortcomings. XP has evolved from the problems caused by the long development cycles of traditional development models and is a collection of ideas and practices drawn from already existing methodologies (Beck, 1999). Abrahamsson et al. (2002) analyze that XP provides individual practices that are suitable for many situations but lacks attention in overall view and management practices. Scrum has been developed for managing the systems development process and is an empirical approach applying the ideas of industrial process control theory to systems development resulting in an approach that re-introduces the ideas of flexibility, adaptability and productivity (Schwaber & Beedle, 2001). Scrum details in specific how to handle the cycles but the integration and acceptance tests are not detailed (Abrahamsson et al., 2002). TDD is a development method to ensure that the developer also focuses on the development of the test cases (Stober & Hansmann, 2010). This approach focuses only in the development of test cases and not on the other aspects of software development lifecycle. Abrahamsson et al. (2002) also discuss the following methods and analyze them accordingly. DSDM focuses on fixing time and resources then adjusting the amount of functionality accordingly. This methodology requires an active consortium to steer...
the method development but lacks transparency in the availability of information to other members and is fixed on resources. The Crystal family of methodologies includes a number of different methodologies for selecting the most suitable methodology for each individual project and includes principles for tailoring the methodologies to fit the varying circumstances of different projects. This methodology provides the ability to select suitable methods but it is not complete as only two of four suggested methods exist today. RUP is an iterative approach for object-oriented systems and embraces use cases for modeling requirements and building the foundation for a system. It is great tool for business modeling but it does not limit the scope of use and the criteria on how to tailor according to needs. ASD focuses on the problems in developing complex, large systems and encourages incremental, iterative development with constant prototyping. Its aim is to provide a framework with enough guidance to prevent projects from falling into chaos. However, ASD is more about the concepts and culture rather than the software practice.

Surrounding the basic principles and among the development methods, Scrum is chosen to be used for this research, fitting the needs to support APISD. The reason behind this is analyzing the various methods given in above paragraph, compared to the other methodologies, Scrum focuses on the management of the system development process rather than delving into a part of the process and defining development practices. Another rationale is that since Scrum does not require any specific engineering practice; it can be adopted to manage whatever engineering practices are used in an organization both in existing projects and new projects (Schwaber & Beedle, 2001). This underlying principle provides a strong foundation for APISD as organizations delivering integrated services may vary in their own practices. Adopting the principles of Scrum allows the flexibility of adapting to conditions as required. The following sections are intended to provide a deeper understanding of Scrum. First, basic characteristics of Scrum will be discussed. Following that the lifecycle within the Scrum approach will be described. Finally the roles within Scrum will be mentioned along with their distinguished responsibilities.

4.2 Characteristics of Scrum

The term 'Scrum' originally derives from a strategy in the game, rugby, which involves "getting an out of play ball back into the game" with teamwork (Schwaber & Beedle, 2001). The Scrum approach is basically focused 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 (Abrahamsson et al., 2002). As development involves several environmental and technical variables (also explored later in the Development phase below), it makes the process unpredictable and complex, which requires flexibility of the development process to be able to respond and adapt to these changes. Through this process, a system is produced which is useful when delivered (Schwaber, 1995). Scrum helps to improve existing engineering practices in an organization. Schwaber enlists the following characteristics of Scrum.

- **Flexible deliverable**: the environment dictates the content of the deliverable.
- **Flexible schedule**: the deliverable may be required sooner or later than initially planned.
- **Small teams**: a team may consist of no more than 6 members (although it could elongate to a maximum of 9 members). There also can be multiple teams within a project.
- **Frequent reviews**: according to the conditions of the environmental complexity and risks, team progress can be reviewed frequently.
- **Collaboration**: intra and inter-collaboration is expected in the project.
- **Object Oriented**: each team will address a set of related objects along with clear interfaces and behavior.
4.3 Scrum Lifecycle

The Scrum process includes three phases: pre-game, development and post-game according to Schwaber and Beedle (2001). These phases along with some sub-phases or activities are described below, which entails the process shown in Figure. 4.

![Scrum Lifecycle Diagram]

**Figure. 4. Scrum Lifecycle**

— **Pre-game**

This phase consists of two activities, Planning and Architecture.

In **Planning**, the system being developed is defined. A product backlog list is created containing the requirements that can originate from the client, sales, and marketing divisions or software developers. The requirements are prioritized and effort needed for their implementation is estimated. Planning also includes definition of the project team, tools, and other resources, risk assessment and training needs. At every iteration, the updated product backlog is reviewed by Scrum Team(s) so as to gain their commitment for the following iteration.

In **Architecture**, based on the items in the Product Backlog (repository that defines the work to be done in the project containing business and technical requirements; contains items such as features, functions, bug fixes, defects, change requests and technology upgrades) the high level design of the system along with the architecture is planned. In the case of changes required on the existing system, the items are identified along with the impact they may cause.

— **Development**:

This phase is highly agile where different environmental and technical (time frame, quality, requirements, resources, implementation technologies, development methods) are observed and controlled through various Scrum practices during the sprints (described below shortly). Scrum aims at controlling these constantly in order to be able to flexibly adapt to the changes occurring in the project lifespan instead of only considering them at the beginning of the project.

Describing as mentioned above, sprints are iterative cycles where the functionality is developed or modified to produce new increments. Each sprint includes the traditional phases of the software development lifecycle: requirements, analysis, design, evolution and delivery phases. Both the architecture and design of the system evolve during the sprint development. A sprint is planned to continue from one week to one month. To complete a system and execute, it could take for example, three to eight sprints in one development process. There also could be multiple teams building the increments.

— **Post-game**:

This phase consists of the closure of the release. Once agreements have been made regarding completion of requirements and no more issues exist or can be found, the system is ready for the release and enters this phase to do further integration, system testing and documentation.
4.4 Roles in Scrum

Scrum consists of six specific roles that have different purposes and tasks: Scrum Master, Product Owner, Scrum Team, Client, User and Management. The roles are further described below as presented by (Schwaber & Beedle, 2001; Stober & Hansmann, 2010).

- **Scrum Master:**
  The scrum master is a management role who is responsible for ensuring that the project is carried out according to the practices, values and rules of Scrum and according to the planning. Scrum Master interacts with the project team, client and management during the project. This actor is also responsible for making sure that impediments are removed in order to keep the team working productively.

- **Product Owner:**
  The product owner is responsible for the project, managing, controlling and making the product backlog visible. This actor makes final decisions of the tasks related to the backlog, participates in estimation of the development effort required and converts issues posted in the backlog into features that are to be developed in future sprints.

- **Scrum Team:**
  The scrum team is a project team that has the authority to decide on the necessary actions and to organize itself in order to achieve the goals of each sprint. They are involved in the estimation, creating the sprint backlog (list of product backlog items that are selected to be implemented in the following sprint), reviewing the product backlog and suggesting impediments to be removed from the project.

- **Client:**
  The client participates in the tasks related to the product backlog items for the system being developed or enhanced.

- **Management:**
  Management is responsible for the final decision making of the project standards. They are responsible for setting goals and requirements according to the business aspects.

- **User:**
  A user is the final actor that engages in using the system developed and deployed.

4.5 Summary

This chapter briefly introduced Agile software development in general along with the various methodologies. Scrum was selected among the Agile methodologies because it concentrates on how team members should function to produce a system adaptively in a constantly changing environment. The characteristics of this process were listed subsequently. Similar to ISD, in order to develop a model for APISD, it is important to understand the lifecycle of Scrum. Therefore, the Scrum lifecycle was explored entailing its phases and the roles involved in it. To provide customer-centric services and understanding the need for agility in business processes to deal with changing environments, the common characteristics of both service development and Agile management are presented in the next chapter, motivating the need in amalgamation of both. Therefore, with the concepts of ISD and Scrum defined in the previous sections, the following chapter discusses the necessary components required for developing a conceptual model for APISD and results in deriving the conceptual model.
Chapter 5

Initial Conceptual Model for APISD
5 Initial Conceptual Model for APISD

In completion of understanding the theoretical concepts of ISD and Scrum in the previous two chapters, the step in DSRM, ‘Design of Artifact’ can finally be completed (shown in Figure 2). Therefore, the initial conceptual model for APISD is constructed in this chapter. First, the necessary components are identified that need to be addressed in the conceptual model under construction in order to combine the components of ISD and Scrum. Second, the initial conceptual model of APISD is presented and described in detail.

5.1 Determining the Necessary Components

With the explication of service development and Agile management principles covered in the previous two chapters, certain common factors can be determined to motivate the need of the union. Analyzing the characteristics of both described in section 3.2 and 4.1, several common intentions can be determined, which are as follows:

- Service development suggests that the service lifecycle should be manageable and maintained where the platform should be established to maintain stability. This characteristic is also similar to an Agile principle suggesting that there should be sustainable development that can be maintained at a constant pace;
- Service development suggests adoption of best practices that allows dealing with changing business needs. This characteristic is crucial for APISD to address the challenge of embracing change and is complimented with the Agile principles for welcoming changing requirements, continuous attention to good design and most importantly regular adaption to changing circumstances;
- Service development suggests that a consumer should not have to worry about how service is implemented, which is similar to a principle of Agile is to incorporate simplicity throughout the development and process;
- ISD requires organizations to be customer-centric, which is one of the core Agile principles of meeting client satisfaction;
- A final principle that Agile contains and is very important to this research is the incorporation of intra and inter-collaboration, which is important to resolve issues in the organizational integration.

The above objectives show a clear vindication to incorporate the two areas in constructing this APISD model. Therefore, to do so, the following sections concentrate on determining the necessary components required to build the initial conceptual framework. First, the phases required in the APISD lifecycle will be determined. Then, the necessary roles to perform the respective activities will be covered.

5.1.1 Determining phases

In order to reach towards the conceptual model, it is important to identify the phases in which the process will occur. It is necessary to understand and look at both the Service Lifecycle and the Scrum Lifecycle individually. Analyzing the phases of both aspects described in section 3.3 and 4.2, a set of phases are defined, which consists of a good blend of both serving each other’s purposes. The reason for deriving the APISD phases is to map the activities crucial for ISD. With a flow of activities with clear phase division, organizations will be able to set clear decision points, manage changes and issues occurring during the planning, designing, development, execution and management stages in the process. Moreover, these phases are similar to the traditional phases followed in the traditional software development methodologies such as Waterfall, Prototyping, Incremental, Spiral and Rapid Application Development methodologies (CMS, 2005). These methodologies contain more or less the basic activities related to planning, requirements specification, design, development and testing. The differences exist in the way certain practices are conducted while performing those activities. As in practice, various companies are operating in various methods, having similar activities in adjustable phases will be easier to grasp and adapt to. Therefore, to support
APISD, the following phases and activities of service lifecycle and scrum lifecycle are given in Table 1, which are mapped together to correspond the APISD phases. The resulting phases will be incorporated in the APISD conceptual model and described.

<table>
<thead>
<tr>
<th>APISD phase</th>
<th>Service lifecycle phase</th>
<th>Scrum lifecycle phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Market Scan &amp; Planning</td>
<td>Pre-game</td>
</tr>
<tr>
<td>Service Modeling</td>
<td>Requirements Engineering,</td>
<td>Pre-game</td>
</tr>
<tr>
<td></td>
<td>Business modeling, Service</td>
<td>Pre-game</td>
</tr>
<tr>
<td></td>
<td>Design</td>
<td>Pre-game</td>
</tr>
<tr>
<td>Service Construction</td>
<td>Service Development &amp; Testing</td>
<td>Development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Development</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Service Provision</td>
<td>Post-game</td>
</tr>
<tr>
<td>Deployment and Execution</td>
<td>Service Deployment</td>
<td>Post-game</td>
</tr>
<tr>
<td>Service Management</td>
<td>Service Monitoring, Service</td>
<td>Post-game</td>
</tr>
<tr>
<td></td>
<td>Management</td>
<td>Post-game</td>
</tr>
</tbody>
</table>

In the table above, it is observed that there is a significant difference in the phases between Service Lifecycle and Scrum Lifecycle. The reason behind this is because Scrum is a simplification of project management and is a rough outline of a process based on iterative development (Stober & Hansmann, 2010). Whereas, the Service Lifecycle deals with an iterative and incremental process that comprises one preparatory and eight distinct main phases that concentrate on business processes (Papazoglou & Van Den Heuvel, 2006). Therefore, Scrum being a project management approach deals with the process and progress of the project in consideration of the team function. Thus, in comparison to Service Lifecycle, for the Pre-game phase it is involved in the Requirements Engineering, Business modeling and Service design phase; for the Development phase, it is involved similarly in the service development and testing; for Post game, it is involved in the deployment to production and closure of the iteration. It is not involved in the Market scan and Planning, Service Provision, Service Monitoring and Management phases because, the activities within these phases are not involved in the project management activities and are planned in different departmental teams and roles. The details of the mapped phases, combining the necessary components of both Service Lifecycle and Scrum Lifecycle will further be discussed in section 5.2.

5.1.2 Determining Roles Involved

In order to facilitate the process for effective collaboration and coordination of ISD management, it is important to specify the roles required and define clear responsibilities. By a shared understanding of roles and responsibilities, parties are more likely to work effectively and manage their activities accordingly, which reduces the lack of clarity in task divisions and as a result, increases productivity. The roles are inspired from the roles defined in the Scrum process and mapped accordingly to support ISD as described previously in section 3.4 and 4.4. Similar to the mapping of phases in the previous section, the APISD roles are defined according to the traditional roles that traditional methodologies have. The responsibilities of roles in the traditional methodologies are more or less the same in nature but are defined in various names according to the methodology and even by the need of organizations. Comparing to the roles defined in Scrum, the APISD roles are broader because APISD entails the process from the conception and initialization of planning services to the management of services while Scrum is focused on the management processes of developing the services. Compared to the roles defined in Service development, APISD roles are distinguished according to the individual responsibilities of the roles, whereas, the service roles are defined in terms of identifying a boundary based on the nature of the activities. The APISD roles are defined and distinguished to serve the necessary roles required to fulfill the activities in the APISD process. Taking the activities themselves into account from both the concepts, the following table shows the APISD roles mapped to the two concepts and their functionalities are subsequently discussed in details. The roles are also illustrated in Figure. 5 with their involvement in the respective phases.
### Table 2. Roles in APISD

<table>
<thead>
<tr>
<th>API SD roles</th>
<th>Service roles</th>
<th>Scrum roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Board</td>
<td>Service Provider</td>
<td>Management</td>
</tr>
<tr>
<td>Domain Manager</td>
<td></td>
<td>Product Owner</td>
</tr>
<tr>
<td>Service Analyst</td>
<td></td>
<td>Scrum Master</td>
</tr>
<tr>
<td>Development Manager</td>
<td>Service Broker</td>
<td>Scrum Team</td>
</tr>
<tr>
<td>Chief Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer/Architect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Developer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Tester</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Service</td>
<td>Customer Service</td>
<td></td>
</tr>
<tr>
<td>Client</td>
<td>Service Consumer</td>
<td>Client</td>
</tr>
<tr>
<td>User</td>
<td></td>
<td>User</td>
</tr>
</tbody>
</table>

**Figure 5. Roles in Phases of Initial Conceptual Model**

The *Service Board* is the committee that governs the set of processes, organizational structure and decides on the services to be developed that are essential for delivering on the ISD promise. The service board members vary for the following scenarios:

- For a single service provider delivering integrated services, the service board consists of members of that single organization governing the rollout of services and is central to decision making.
- For multiple service providers delivering integrated services, the service board consists of executives of the various service providers collaborating together and governing the rollout of services and central to decision making.

The service board is responsible for understanding the business environment and making sure that all the necessary controls are incorporated in the APISD framework. Activities such as identifying services required in discussion with the client, analyzing the business needs, analyzing financial aspects and making decisions as required for the ongoing development of integrated services (Papazoglou & Van Den Heuvel, 2006). Key involvement that the service board is involved in is also shown in Figure. 5, performing all activities in Planning phase,
decision making in Service Modeling phase, setting release dates in Service Construction Phase, formalizing Service Level Agreements in Provisioning Phase, and decision making if required in Execution and Service Management phase.

A Service Analyst is an analyst who considers the business objectives and defines the services identified by the service board. The service analyst is responsible for creating a service backlog (see section 5.3). An organization may have a single service analyst or a team of analysts. The team may be of either of the following nature:

- For a single service provider delivering integrated services, the service analyst team consists of analysts of that single organization working together to perform the necessary activities.
- For multiple service providers delivering integrated services, the service analyst team consists of analysts of the various service providers collaborating together to perform the necessary activities.

While working closely with the service design team, the service analyst team is responsible for specifying functional and non-functional requirements for every service from the service backlog. This team is solely responsible for maintaining the service backlog. Other responsibilities include: communicating with clients for requirement specification, clarifying requirements to other developers and testers, participation in service demonstration to clients, and communicating with the service board for prioritization of services and alignment to business needs. Key involvement that the service analyst is involved in is performing all activities in Service Modeling phase, clarifying requirements in Service Construction Phase.

The Development Manager is a role that is involved in both the Service Modeling and Service Construction phase. In the Service Modeling phase, this role is responsible for working together with the chief service developer/architect to produce technical designs. For requirement clarification, this actor communicates with the service analyst team. Finally, creating a feature backlog (see section 5.3) with the chief service developer/architect is also an output of the Service Modeling phase. The development manager is solely responsible for maintaining the feature backlog and assigning features to the development teams for developing the services. This role is responsible for introducing the feature backlog to the respective team, coordinating the integration process, handling team issues and dependencies, and ensuring that the development and testing process runs smoothly in the Service Construction phase. Apart from the various activities, the role acts as a Scrum Master (Schwaber & Beedle, 2001; Stoer & Hansmann, 2010) for the construction team. Furthermore, this role is responsible for verifying that the functionalities of the features are developed according to the specifications and arranging a preview meeting with the client. Therefore, it is important that the development managers are both technical and business oriented. With the release manager, the development manager sets release dates and therefore, it is also crucial that the development manager is aware of the progress of the iteration in terms of time and planning. Key involvement that the development manager is involved in is performing all activities in Service Modeling phase, managing the development in Service Construction Phase, monitoring deployment in Deployment & Execution.

A Chief Service Developer/Architect is a member of the Service design team in the Service Modeling phase. This role is responsible for deriving technical designs per service from the service backlog along with the development manager. In the Design sub-phase, the role is initially responsible for the high-level estimates of development and handing them over to service analysts. Then other responsibilities include high level designing and estimating the components for each of the services, such as portal, integration, infrastructure, data, policy, and business (logical) services (Durvasula et al., 2006). For requirement clarification, this actor communicates with the service analyst team. Once the high-level architecture is completed, the architect along with the development manager produces features per service and creates a feature backlog. If required, high-level designs are produced for the features. This role is also responsible for taking part in the development of the integrated services in the Service Construction phase. If there are multiple service providers involved, this role collaborates with other Chief Service Developers/Architects of the other providers.
The Domain Manager is responsible for ensuring the alignment between the service analyst and the service design team in the service modeling phase. This role can be appointed by the service board or selected among the service modeling teams. As the service analyst team will be involved in defining requirements and service design team for deriving the technical designs, it is important that both teams are in-sync with what should be implemented. Therefore, the domain manager is responsible for holding meetings for alignment; to handle any conflicts that may occur in the service modeling team; and ensuring the quality of the output products. The domain manager is also responsible for holding retrospective meetings for the service analyst and service design team, in order to review the previous iterations and to identify key improvements to be done in the future iterations.

The Service Developer is responsible for performing activities related to the realization of services in the Service Construction phase. Responsibilities include reviewing feature and service backlog and understanding the requirements, providing an estimation for development, developing and executing according to planning, coordinating with the service testers for testing and solving bugs and issues in the services. In the case of multiple service providers, if required, the service developer collaborates with other service developers of other providers during integration of the services. For any conflicts or dependency impediments, the service developer communicates and reports to the development manager. Key involvement that the service developer is involved in is performing all activities in Service Construction phase, fixing any issues in Deployment & Execution phase and fixing urgent incidents in Service Management phase.

The Service Tester is responsible for performing activities related to the testing of services in the Service Construction phase. Responsibilities include reviewing features and the service backlog and understanding the requirements, providing estimation for testing, testing according to planning, and coordinating with the service developers for bugs and issues reporting and tracking in the services developed, signing off service construction. For any conflicts or dependency impediments, the service tester communicates and reports to the development manager. Key involvement that the service tester is involved in is performing all activities in Service Construction phase, testing in Deployment & Execution phase and testing urgent incidents solved in Service Management phase.

The Release Manager is responsible for managing releases of the ISD. In the Service Construction phase, this role discusses with service board and development manager to set release dates and the services to be deployed. Together with the development manager, the release manager is responsible for incorporating a ‘freeze’ after development and testing of the integrated services, so that no more development or testing is done that could potentially defect the ‘ready’ system. This role also decides if other releases can be included with the planned release. In the Service Management phase, the release manager can go through the change request backlog (see section 5.3) and give a primary indication for implementation. This set can be further analyzed by the service board to plan accordingly for the next set of services to be developed.

The System Administrator is responsible for all the activities required in deployment of the integrated services and management of systems and services. Responsibilities include: establishing systems environment consisting of development, system integration testing, performance testing, user acceptance, and product environments; assisting service development teams in systems and application configuration, periodic builds, and capacity planning; tracking and managing dependencies among services and assets; deploying and managing business services in production and providing application support for business services based on business priority (Durvasula et al., 2006).

The Customer Service is responsible for supporting the client/user’s engagement with the integrated services. This team is involved in activities related to client support, handling change requests and logging in incidents. They are solely responsible for maintaining the change request backlog and incident backlog (see section 5.3).
The *Client* is the potential buyer of the integrated services that the service provider(s) provide. This actor is responsible for discussing with the service provider(s) about the desired needs, which are later translated to functional services, and specifying the desired requirements for those services. Once services are developed, the client previews the solution and either accepts or rejects. The client is also responsible for reviewing the SLAs produced by the service provider(s) and to agree or disagree with the SLAs. Usage of the services is either by purchasing or renting the services based on the charging mechanisms set in the SLAs. The client can report back to the service provider(s) regarding any change requests or incidents.

The *User* is someone who actually uses the integrated services. This role could be someone from the client’s organization or as customers of the client. The user uses the integrated services and can report via the client or directly to the service provider(s) regarding any incidents or change requests.

### 5.2 The APISD Initial Conceptual Model

With all the necessary components determined from the Service Lifecycle and Scrum Lifecycle in the above section, a conceptual process model of APISD has been formed, which is illustrated below in Figure. 6. This model is the artifact of this research entailing the lifecycle of APISD and will be further discussed in this section. Knowledge of activities within the phases of this lifecycle is acquired from literature analysis and work experience of the author.

![Figure 6: Initial model of APISD Lifecycle](image)

The APISD lifecycle consists of six phases, which also have been determined in section 5.1.1: Planning, Service modeling, Development and Testing, Provisioning, Deployment and Execution and Change management. Each of these phases consists of a certain number of activities performed by certain roles and is described extensively in this section. The lifecycle first starts with the Planning phase, where the planning and feasibility study of the service delivery is done. Following the Planning phase, the Service Modeling begins. As depicted in the figure, this phase consists of two sub-phases: Analysis and Design, in which each of the sub-phases goes through an iterative development. Analysis deals with the analysis of new and existing services and specification of functional and non-functional requirements. The Design sub-phase deals with the designing of the architecture, and resource planning for the development of services. The phase transitions to the Service Construction phase where the actual development and testing occurs. After the services are integrated, the system goes through a user acceptance test. Once approved, the Provisioning phase starts where the
Service Level Agreements (SLAs) are formalized. Following the confirmation of the client on SLAs, the integrated service package is deployed in a production environment and made available for clients to use. In the process of usage, services are monitored by the service providers and change requests/incidents are handled by the client service. These are logged in a backlog, where the service board can once again analyze and plan the next set of services to deploy. The phases of this lifecycle are further elaborated in the following sections, discussing the processes and communication among the roles.

5.2.1 Planning Phase

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.

![Figure 7. Planning Phase of initial model](image)

As shown in Figure 7, the service board meets with the client and discusses various aspects of the services to be developed. As both service development and Agile development encourages customer involvement, it is important to maintain close collaboration with the client for better understanding their needs and providing the desired services. Along with the activities mentioned in Market scan and planning phase in section 3.3, they also determine the nature of the services, the scope and the feasibility of developing them. The service board drafts a project document (see section 5.3). It is important to produce this document at an earlier stage so that there is a common understanding of the business objectives within the organization. They deliver this document to the service analyst team who will start with the analysis of the project. In this phase, the service board also drafts rough SLAs that deal with the service expectations. It is important to start with defining the SLAs in the beginning because the company can already think about the critical aspects of providing the services to the client and begin early negotiations. They can further formalize it in the Provisioning phase once service package is completed. In the case of multiple service providers in the 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, lack of coordination will arise throughout the lifecycle. As a result, service providers will not be able to collaborate smoothly or gain trust, which is required for sustainable development.

5.2.2 Service Modeling Phase

The main objective of the service modeling phase is primarily to describe the services identified in the planning phase. The phase consists of two sub-phases, Analysis and Design (a combination of defining and design activities mentioned in Requirements Engineering, Business Modeling and Service Design phases in section 3.3 and Planning in section 4.3) and is carried out by the service modeling team. The team comprises of service analysts, development managers, and chief service developers of the corresponding service providers associated in the project. In the team, 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 designing are aligned. The importance of this role is primarily because having separate iterations, where the service analyst team and the service design team are independent of their activities in this phase, problems may rise in the alignment and grouping of the work produced. The sub-phases are executed in form of separate iterations. As mentioned earlier in section 1.3, there have been several researches on only the management of the construction phase where iterations are incorporated but to the best of our knowledge iterative development have not been used in earlier phases in ISD. It is acknowledged that as time progresses, cost of change increases (Ambler, 2009). Therefore, if requirements are well specified and designs are accurately modeled, then there is a less probability of bottlenecks or major issues occurring in the system at a later phase than if not done well at all. Thus, we feel the importance of introducing the iterative development from an earlier stage because it is equally important to invest time and effort in proper requirement analysis and designing just as development is. Incorporating the iteration allows the respective teams to work based on priority and produce usable artifacts in short periods of time. Also, when development teams start constructing, the service modeling team does not have to sit around. They can continue their work in parallel to the development teams. This results in maximized utilization of resources by the organization. Therefore, just as construction of the services is considered a development period, requirement analysis and designing are also considered as development periods in iterations. With respect to the motivation, this phase is depicted in the Figure 8, and is further described in the corresponding sub-phases.

**Figure. 8. Service Modeling Phase of initial model**

**Analysis:**

In this sub-phase, the service analyst team is responsible for reviewing project document, analyzing services and defining requirements. As shown in Figure 8, as soon as the team has retrieved the project document and reviewed the business goals and objectives of the projects, they try to identify services and describe them. Furthermore, they also investigate to see what existing services exist and whether they can append or enhance by adding features or create new services (Papazoglou & Van Den Heuvel, 2006). Once they identify and describe the services by requirements engineering, they produce a *service backlog* (see section 5.3), which is a database of the services to be developed. Inspired from Scrum, the service backlog is similar to a *product backlog* where the requirements are specified. Having this service backlog, the entire organization is able to know what services exists or will be implemented; also it helps in the organization of managing the reference and traceability of service description. The service backlog is given to the service design team who provides a draft estimation of time and resources required to develop the services. This service backlog along with the estimation is given to the service board, where they analyze potential return on investment. If the investment is profitable and fitting with the business needs, they decide on the services and
finalize the service backlog with prioritization. The service analyst team then specifies requirements for the prioritized services and delivers to the service design team for creating technical designs. The team then sends these requirements to the client. If modifications are required, then the requirements are revised. Else, if approved, the service analysts deliver the service backlog to the service design team for creating technical designs. It is important that the client is involved in this phase because if they are not involved and services are implemented, there is a chance that the client may not agree to what is developed. Therefore, with client participation, which is also something that Agile and service development encourages, there is a better chance of understanding by the organization and trust gained.

This process involving the analysis phase is done in the first iteration, lasting for 2 – 4 weeks as depicted in the APISD lifecycle in Figure. 6. The iteration follows as a sprint does (see section 4.3 and Figure. 4) in the development phase of Scrum. The service analyst team meets weekly to coordinate and take status of their work and ultimately finalizes the service backlog. Next, the analysts elaborate on the prioritized services by producing requirements of both functional and non-functional nature. Once produced, they deliver the service backlog to the service board or meets with the client to finalize requirements. After confirmation, the analysts provide the service backlog with detailed requirements to the service design team for the design phase, also shown in Figure. 8. While the service design team is busy with modeling the services in their first iteration, this iteration is counted as the second iteration for the service analyst team. As mentioned earlier in this section, having continuous iterations allow maximized resource utilization. Here, the service analyst team commences their work on the next set of prioritized services. They meet with the service design team to coordinate and understand the required components of technical requirements to be aligned, whenever it is required. These meetings are organized by the domain manager to ensure the alignment between the two teams and to handle any conflicts that may occur in the service modeling teams. The domain manager also holds retrospective meetings for the service analyst and service design team, in order to review the previous iterations and to identify key improvements to be done in the future iterations. This is encouraged in Scrum, where frequent reviews help to grow and improve existing services. By this way, performance is reviewed and monitored as suggested in overcoming organizational integration issues (see section 3.1).

**Design:**

The design phase entails the formation of technical designs and producing features from the service backlog. This phase consists of two sub-activities. First, as mentioned above and as shown in Figure. 8, the service design team analyzes the service backlog and creates a draft estimation and second, with the given requirements of the prioritized services, they produce the high level architecture and the necessary high level technical designs required for development for the first development iteration. Services can be designed to comply with all the functional and non-functional requirements. While designing the new services, it is important to keep into account whether there are existing services that provide similar functionality (Papazoglou & Van Den Heuvel, 2006). This investigated helps in re-using existing services, which is encouraged in service development. In accordance to the previous phases, the service design team comprises of either the following scenarios:

- For a single service provider delivering integrated services, the service design team consists of the chief developer/architect and development manager of that single organization working together to perform the above activities.
- For multiple service providers delivering integrated services, the service design team consists of the chief developer/architect and development manager of the various service providers collaborating together to perform the above activities.

As shown in the APISD lifecycle in Figure. 6, the service design team performs this process iteratively for 2-4 weeks similar as a sprint does (see section 4.3 and Figure. 4) in the development phase of Scrum. They meet weekly for 1 hour or daily for a short 15 minute
status update to further clarify or decide on any issues regarding the designing of the service models. If required, they meet with the service analyst team for requirement clarification or modification. At the end of each iteration, they produce a Feature backlog (see section 5.3) of the prioritized set of services. The development manager is responsible for managing this backlog. This backlog is divided according to features and assigned to the respective construction teams. For the case of multiple service providers, features are assigned to the corresponding construction teams of the various providers. In this case, that particular team is responsible for the implementation of the features of the service or even multiple services. To the best of our knowledge, it could possibly reduce the conflicts of dependency, maintenance in later phases, to develop feature-wise rather than component-wise (database, user interface etc.) and therefore, division of work by feature is chosen for this process. This advocates the need of coordination in activities by which it helps the service provider to effectively organize the work forces (Hatchuel, 2001). Once the construction team is busy developing and testing the set of assigned features in their first iteration, this iteration is counted as the second iteration for the service design team where they commence work on defining the next set in the feature backlog for the next set of prioritized services in the service backlog.

5.2.3 Service Construction Phase

The service construction phase consists of the actual development and ongoing testing that agile methods suggest. First, the construction team(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 shown in the APISD lifecycle in Figure 6. According to the assigned features that are taken to develop in the iteration, an ‘in-progress’ update is made in the feature backlog by the development manager. For each construction team, together with the development manager, service developers (including the chief service developer) and service testers, they go through the services’ and features’ requirements and technical designs. By reviewing the service and feature backlog, they are able to have a shared understanding and by continuous communication, a collaborative environment is created between the team(s). The team communicates with the service modeling team for clarification if needed. This meeting is arranged by the development manager.

![Figure 9: Service Construction Phase in initial model](image)

Depicted in Figure 9, from steps 1 to 4, once development begins testers create their test plans and test cases. Development of the services can be done following standards and
protocols using technologies such as web services, BPEL or others (see section 3.1). When
development of each team ends, the application is deployed locally by the system
administrator and is tested by the respective test team. If issues are found, they are logged in
a bug tracker system that is available for team(s) involved and distinguished by features.
Issues are fixed and regression tests are performed. By regression tests, we mean testing
existing features as well to test whether they are affected by the new implementation or not.
Once tested and approved by the testers, all the applications are deployed by the system
administrator and all the features are integrated in one system in step 9. Next a system test is
performed that includes testing of the system as a whole. This test ensures that all the
interfaces are running smoothly between the services and the system in general. Should there
be any issues, they are logged in the bug tracking system and solved by the teams who have
developed that corresponding feature. 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 (see
section 3.1). In this demonstration, the service analysts and development/testing team
representatives can attend. Once approved by the client, the iteration ends. If not approved,
the set of improvements goes through the steps 1 to 4 and 9 again.

During the iteration, in step 5, the release manager meets with the service board to discuss
releases. In step 6, the release manager then meets with the development managers and sets
release dates. After a release date has been set and development and testing is completed, a
set of features for the corresponding completed services are deployed in the system. The
system goes through a final test after solving issues and packaged to a frozen state, which is
ready for the provisioning phase. The team can proceed with the next set of features from the
feature backlog accordingly.

The development manager handles any conflicts between the development and test teams.
In the case of multiple service providers, each construction team has one development
manager. In order to coordinate effectively among the teams, the development managers
meet weekly. In this meeting, they discuss any impediments, dependency related issues and
further planning of iterations. They also discuss on changes that may occur. With a shared
understanding, they can project a clear vision to their respective teams to adapt those
changes efficiently, thus embracing changes and coordinating the implementations accordingly
(see section 3.1). Towards the end, the development managers also arrange a retrospective
meeting of their own to discuss results, lessons learned and improvement points. Then each of
them holds a retrospective meeting with their own team to discuss the same and take two or
three improvement points to implement in the next iteration. This whole process continues in
iterations with the corresponding set of prioritized features and is implemented accordingly.

5.2.4 Provisioning Phase

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. 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. In the case of multiple service providers, this phase also involves in
settling service management and governing of the services. In this phase, the service board
decides on the service governance, and billing according to usage.

Papazoglou and Van Den Heuvel (2006) have mentioned central and distributed
governance. Central governance has one governing body from each service domain that does
not have direct responsibility for any of the service domain but reviews the type of services
before implementation. In distributed governance “each business unit has autonomous control
over how it provides the services within its own enterprise” and is suited for distributed
teams. When considering the governance type for the agile ISD, once again, it varies on the
scenario of the organizations involved. On one hand, for a single service provider, a central
governance suits best as the organization is making its own decisions about the integrated
services they are to provide. On the other hand, for multiple service providers, distributed
governance is the most suitable approach. As there are multiple organizations involved and
services are executed and maintained by the service providers themselves, they have the
autonomous control as Papazoglou and Van Den Heuvel (2006) mentioned. The service board
is then responsible for defining the rules and regulations regarding process, domain
responsibilities and service development responsibilities among the various teams for effective
steering and control of the integrated service delivery process. This is done at the beginning of
the APISD lifecycle, in the form of internal SLAs. However, even though distributed
governance exists for the execution and management of services, the service board in this
scenario does decide together and governs on the decisions. Considering this argument, during
the planning and decision making throughout the APISD lifecycle, the service board acts
according to a central governance mechanism.

Setting apart the scenarios, the service board in both cases is also responsible for creating
SLAs for clients that assures the deployed services. Through this contract, the service
provider(s) are able to charge their clients for what they offer or to endure security (Gu &
Lago, 2007). Here the service board needs to decide on how they plan to charge the service
usage and draw the SLA for the client. Available options may be: payment on a per use
basis, payment by subscription, payment by leasing, lifetime services, and free services
(Papazoglou & Van Den Heuvel, 2006). In addition, the following options are also available:
per service request or delivery (e.g. a fixed price local telephone call); (2) by unit of measure
and granularity (e.g. by length, volume, weight, area or time); and (3) on a percentage or
ratio basis of some aspect of the service (e.g. by commission) (O’Sullivan et al., 2002). Along
with the payment option, the service board also needs to define in SLAs how well the services
are delivered in terms of cost, availability, performance (Gu & Lago, 2007). Once the service
board drafts the SLAs and sends to the client, the client reviews and either agrees or
disagrees. Further discussion between these two parties occurs and further alterations are
made in case of disagreements. Only when all the SLAs are confirmed and signed, the
accessibility of the services are performed where the authorization of using the services are
permissible.

![Figure. 10. Provisioning Phase of initial model](image)

5.2.5 Deployment and Execution Phase

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. This means that the web services of the system are fully
deployed and operational. Service requesters are now able to find service definition and invoke
all the service operations (Papazoglou & Van Den Heuvel, 2006). Once the integrated services
are deployed in the production, a quick test is run to check that the services are running
properly. If there are any problems, they are fixed within a short time and re-deployed. The
customer service informs the client regarding the deployment of the services.
5.2.6 Service Management Phase

Once in production and used by the users, the integrated services can be monitored and ensured that all the services are running according to the rules and regulations set in the SLAs. Monitoring the services will allow the service providers(s) to analyze the service delivered and gather improvements. 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. As previously mentioned earlier in service lifecycle (see section 3.3), it is very important that changes are managed well in order to ensure a smooth operation of business. Following the usage of services, clients can request changes, which are logged in a change request backlog (see section 5.3), also shown in Figure 12. While using the services, various incidents may occur which are logged in an incident backlog (see section 5.3). Depending on the urgency, the incidents are immediately passed over to the respective construction team to solve it. If not urgent, they are logged in based on severity and priority which are taken in iterations. Similarly change requests are logged in. Both the backlogs are reviewed by the Release Manager and based on the primary prioritization, the chosen sets are then passed to the planning phase where the service board reviews it and the whole cycle progresses accordingly as shown in the APISD lifecycle depicted in Figure 6.

Figure 12. Service Management Phase in initial model

5.3 Artifacts of APISD

In the APISD model explained in the previous sections, several artifacts have been mentioned that are produced and used by the various roles in different phases. Each of these artifacts will be described in this section.

— Project Document
The project document consists of the identified services and their purposes, scope and feasibility of the project, financial analysis of costs and benefits including a budget. This document is produced by the service board.
— **Service Backlog**

The service backlog is a collection of finalized services. Each service in the service backlog consists of service number, description, priority, detailed functional and non-functional requirements, and estimated duration of completion. The service backlog also contains a high level architectural design that illustrates the relations between components, infrastructure and logical explanations. This backlog is produced and managed by the service analyst team.

— **Feature Backlog**

The feature backlog is a collection of features referred to corresponding services. Each feature in the feature backlog consists of a feature number, service reference, priority, technical design, estimated duration of completion, and assigned to team description. This backlog is produced by the service design team and managed by the development manager.

— **Change Request Backlog**

The change request backlog is a collection of change requests made. Each change request in the backlog consists of a change request number, description, service reference, feature reference, and priority. This backlog is produced and maintained by the customer service.

— **Incident Backlog**

The incident backlog is a collection of incidents reported. Each incident in the backlog consists of an incident number, description, service reference, feature reference, priority, and severity. This backlog is produced and maintained by the customer service.

### 5.4 Summary

This chapter first determines the phases and roles required in APISD by comparing and combining the phases and roles in ISD and Scrum. After derivation of the components, the APISD lifecycle is introduced and described in details. The phases identified in the APISD lifecycle include: Planning, Service Modeling, Service Construction, Provisioning, Deployment and Execution and finally Service Management. Following that, each phase in the APISD lifecycle is elaborated upon with the corresponding activities associate with the collaboration of various parties interacting with each other. Finally, the artifacts resulting from the workflows are described in details. In order to enhance the initial conceptual model, a case study was conducted in organizations to test the model and gather input regarding the practicalities and complexities compared to their organizational process. The following chapter elaborates on this outlook and evaluates the initial conceptual model. From the evaluation, additional enhancements are identified, which are incorporated in the final conceptual model that is addressed later in this thesis.
Chapter 6

Evaluation of APISD
6 Evaluation of APISD

Following the constructs of case study research given by Yin (2002), the framework will be 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 will be explored. The exploration of case study and evaluation of the initial conceptual model for APISD attributes to the DSRM steps 'Demonstration of Artifact (APISD) application' and 'Evaluation of Artifact (APISD)' (see Figure 2). The objective of these two steps is to demonstrate the artifact (i.e. the APISD model) to the organizations and test with their own process. The testing of the artifact will result in identifying differences between the artifact and organization process and deriving additional factors to enhance the artifact. An attempt has been made to select cases that are spread over the various development variations, to make the results of this research broadly valid. To the best of our knowledge, the more diverse the cases are, the better the external validity of the result becomes as the model can serve various organizations in general. The model was demonstrated to the organizations and evaluated according to comparison of their current running process. Feedback was retrieved on the similarities, difference and improvements that can be made in the model. This chapter first discusses the selection of cases in the case study. Second, the case study procedure followed to conduct the case study research is described. Third, the validity of the case study findings is elaborated. Fourth, each case is explored on the evaluation of the initial conceptual model. Finally, the analysis of the case study is summarized on the information that is valuable and essential for enhancing the initial conceptual model.

6.1 Case Selection

The following selection criteria for the case study are considered because in order to evaluate the APISD model, an organization needs to have these characteristics. In other words, the cases were selected based on these entry criteria to ensure a proper evaluation of the model.

- In order to consider a service provider for the case study, it is important that the service provider delivers integrated services. The service provider can deliver the integrated services themselves or together with other partners.
- Due to the aim of achieving key benefits mentioned in section 1.3 by implementing APISD, it is important that the service provider considers quality, delivery time and reusability of services as important objectives for the development of integrated services.
- It is important that the service provider already has a process working in order to develop the services. By having an existing way of working, the cases can evaluate APISD comparing to their own and thus provide feedback that are essential in practice.
- As involvement of clients a crucial aspect in Service development and Agile, it is important that the service providers involve their client in requirement specification, clarification, and preview and feedback of the demonstration of the solutions.

Three cases were conducted in three different organizations. These cases are based on the unit of analysis, namely the process that is followed within the organization to deliver integrated services. First, a brief background of the organizations will be given. The details on the case or in this case the process itself will be described in detail in the respective case analysis in the sections 6.4, 6.5 and 6.6. The first case is a logistic service provider 'LSP-A', who provides multiple integrated services and develops themselves. They provide these services acting as a hub for all logistical information in ports for providing simple and effective information exchange between companies and between the companies and governments. Customer service is very important to this service provider where the development of new services start only when there is a clear market need. Acting as the central hub and making the services available allows lower costs, greater transparency and
optimal re-use of information. This organization involves her clients in requirement specification, clarification and product demonstration.

The second case is another single logistic service provider ‘LSP-B’, who also delivers integrated services focused on the air cargo industry. This service provider delivers comprehensive supply chain solutions needed to use electronic information exchange between airlines, forwarder, handling agents, transporters, shippers and customers. To briefly explain, a forwarder is a company that organizes shipments for individuals or other companies and may also act as a carrier; handling agents is basically managing agents or a network of agents associated with the corresponding services; transporters are vehicles in air, rail or road, that transport material. LSP-B provides these solutions for specific customer groups or individual companies. Just as LSP-A, they also define projects depending on the need of the industry and how they can further improve processes. Customer service is very important for this organization and therefore, they involve the customer service department in testing their solutions. Quality, delivery time, reusability of services, trust and providing neutral services within the industry are considered very important objectives. This organization involves her clients in requirement specification, clarification and product demonstration.

The third case is a platform service provider ‘PSP’ who delivers integrated services by themselves and also with other service providers. This case fulfils the case scenario of multiple service providers collaborating together to provide integrated services. This service provider provides an Agile platform to rapidly plan, develop, deploy and manage custom business applications & services that easily extend existing systems throughout the entire agile application lifecycle. They use visual models to capture business requirements, translates them automatically into working business applications where programming is not required. Similar to the other cases, quality, delivery time, cost and reusability of services are important objectives to this provider. Furthermore, this provider also involves her clients in requirement specification, clarification and product demonstration. When collaborating with other service provider, they are involved in the development of the applications and provide support in the latter phases.

In this section, first the procedure for conducting the case study will be given. Second, the validity of the case findings will be discussed based on the validity tests explored in section 6.3. Third, each case will be explored with their background and their impression and evaluation on APISD will be analyzed. For anonymity purposes, the cases will be denoted as ‘LSP-A’, ‘LSP-B’ and ‘PSP’, with LSP for Logistic Service Provider and PSP for Platform Service Provider. In presentation of the initial conceptual model of APISD, cases were compared and analyzed with APISD. From the analysis, a certain number of differences were identified. The following evaluation by the informants will be discussed per phase with respect to:

- Current Process: how it is in the current organizational process;
- Similar To APISD: common factors between cases APISD;
- Missing Factor in Case: missing factors existing in APISD but not in the case;
- Distinction in APISD: what is different in APISD from the current process and contributing to the missing factors in case;
- Additions for APISD: essential factors to be taken in consideration to enhance the APISD artifact.

6.2 Case Study Procedure

In order to obtain the necessary information for this research, the following three procedures were applied:

1. Perform semi-structured interviews with both a set of questionnaire and an explorative discussion.
2. Analyze the questionnaire, interview discussion, audio recordings, web site documentation, digital documentation, e-mail correspondence.
3. Involve informants for reviewing the draft case study report.

First a questionnaire has been sent to the key informants consisting of a representative of each company who is knowledgeable on the process that is followed within the organization when developing integrated services. The questionnaire consists of both questions of fixed response categories and open-ended questions that may vary in cases. Each representative answered the questions listed in Appendix I: Interview Questions. From the closed questions, it is analyzed on what are the main factors that each case considers to be important and to look for commonality. With the open set of explorative questions also listed in the appendix, information regarding the constructs of the model was analyzed and certain input and differences were identified. Following that, the complementary information retrieved from website documentation and digital content was looked at. After conducting the case, the results were sent to the informants for review. The reason for involving the informants at this stage was to increase the construct validity, which is further explained in the next section.

The motive behind choosing a semi-structured interview approach was because this research required some precise answers and also an open discussion on opinions of the model derived. In contrast to structured and unstructured interview, the semi-structured interview method was chosen because structured interview deals with strict adherence of questions and instructions, with the aim to achieve the highest degree of objectivity and uniformity in procedure, whereas, unstructured interview employs open-ended questions where the structure is flexible and restrictions are minimal (Yin, 2002). Since, this research required responses that are in fixed categories to reduce interviewer biasness to a minimum and maintain uniformity for all cases, some elements of the structured interview was used. Also, since this study required an open discussion on the model to analyze and compare with the organization’s model, open-ended questions were asked. The nature of open-ended questions comes from elements of the unstructured interview method. Therefore, consisting elements of both, the semi-structured interview was chosen for this qualitative research.

6.3 Validity of the Case-study Findings

When considering the validity, it usually is used to test the ‘quality’ of research results. The types of validity that are considered in this research are (Yin, 2002):

- **Construct Validity**: establishing correct operational measure for the concepts being studied.
- **Internal Validity**: establishing causal relationships, whereby certain conditions are shown to lead to other conditions, to distinguish them from false relationships.
- **External Validity**: establishing the domain to which the findings of a study can be generalised.
- **Reliability**: demonstrating that the operation of a study can be repeated with the same results.

As there are multiple cases, therefore the study is classified as a multiple-case design. Both case studies are based on the same design. 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. With the various sources used to collect the data for this research from website documentation, interview, digital notes and e-mail correspondences, acting as the evidence, the types of validity are looked at. The multiple-case design is shown in Figure 13.
Construct validity is relevant, because in order to retrieve information, multiple sources of evidence were looked at. As multiple data sources provided evidence and the chain of evidence was established to determine the relations, construct validity is recognized. The validity has further increased due to informants reviewing the draft case study report.

Internal Validity is irrelevant for this research because the nature of the case study was explorative instead of explanatory or causal (Yin, 2002). In this research, the model was only evaluated in terms of matchmaking and understanding how to further improve. There were no causal relations examined or addressing logic models.

For the validation of theory with multiple cases, two types of case selection are distinguished that influence the borders of external validity: cases with literal replication and cases with theoretical replication (Yin, 2002). Theoretical replication is chosen as the case selection method because compared to literal replication, contradictory results are allowed to occur, but under explicitly stated reasons and predicted results. This constructs generalisability for a wider scope of cases, thus expanding the domain in which the results are externally valid. The model has been replicated for the three cases 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 in section 6.1 and can be performed for other case studies. These procedures are stored in a case study database and used to retrieve the data following the steps mentioned.

6.4 Case LSP-A

LSP-A is a single service provider who develops integrated services on their own. The service provider identifies, defines, develops, governs, executes and manages the integrated services by their own organizational members. They follow Agile Development- Scrum for the development of their services. They operate on an integrated collaboration environment where issues are managed, system build are run and releases are managed. Their main focus on development is on quality control. They are moving towards a Service Oriented Architecture (SOA) for better supporting and managing the services.

As discussed in the introduction of this section, the analysis of the comparison between LSP-A’s process and APISD will be discussed per phase of APISD and according to the list below. Each phase will consist of the factors below that are applicable to the analysis.

- Current Process: how it is in the current organization process;
- Similar To APISD: common factors between LSP’s process and APISD;
- Missing Factor in Case: missing factors existing in APISD but not in the case;
- Distinction in APISD: what is different in APISD from the current process and contributing to the missing factors in case;
- Additions for APISD: factors to be taken in consideration to enhance the APISD artifact.

Planning Phase:

A. Current Process:

In LSP-A, the process starts with defining the needs of the business, aim and objectives, which are set by the strategic board. The strategic board is also
responsible for making decisions. Keeping the business needs in mind the functional analysts are responsible for creating an Impact Analysis. The impact analysis is about analyzing the feasibility of services that meet the needs of the business. The functional analysts only look at the functional part and derive a rough idea of the service relations. In this activity, there is a tactical board who is involved in discussing the feasibility of such services and discussing the service landscape. The board consists of an Enterprise Architect, Service Portfolio Manager, Business Representatives and IT manager. In completion of the impact analysis, the tactical board communicates with the strategic board for a decision. Once the strategic board approves, the tactical board creates a Project Initiation Document (PID), which consists of the impact analysis and preliminary planning of the project. After reviewing the PID, the project manager begins to gather requirements from the client. Often, the project manager creates prototypes. Once requirements are gathered, the project manager creates a project plan for the development of the services.

B. Similar To APISD:

1) The common step that APISD also follows in comparison to LSP-A is that there is also an impact analysis done. This impact analysis is done by the service analysts in the first iteration of Analysis (sub-phase).
2) APISD has a service board that performs similar activities as the strategic board of LSP-A.

C. Distinction in APISD:

1) The impact analysis done in the planning phase of LSP-A is actually done in Service Modeling: Analysis phase.
2) APISD has only one planning board, naming the service board, in comparison to two kinds of board in LSP-A, the strategic and tactical board. The strategic board maintains the business needs and direction while the tactical board is involved in the projects to fulfill those needs. The current APISD model has the service board who does those combined activities.
3) In APISD, the service analysts are responsible for collecting and specifying requirements, in comparison to the project manager in LSP-A.
4) In APISD, there is no official project plan in comparison to the project plan created by the project manager in LSP-A. The resources and estimation are given by the service modeling teams in the form of the Service Backlog.

D. Additions for APISD:

1) Looking at the need of differentiating the responsibilities of the strategic board and tactical board, in APISD the service board can be divided in two sub boards with distinction in responsibilities. Having this distinction will allow a clear division and understanding of responsibilities according to the strategic and operational nature.

Service Modeling Phase:

A. Current Process:

In LSP-A, there is no phase existing that is distinct from the planning and construction of the services. The activity of collecting high level requirements is done in the planning phase and the detailed functional requirement specification is done in the construction phase, namely the first sprint/iteration. Producing the technical designs is also done in the construction phase in the following sprint/iteration. However, having these activities in the construction phase, LSP-A does not incorporate the requirements engineering and designing in the form of iterations themselves. Only the development is performed in iteration. With these activities in the construction phase it causes a gap between requirement specifications, designing and development and thus causes the developers to wait for a significant period that can vary. LSP-A mentions that “The connection with these phase-related activities was
difficult because the analyst and developers start at same time and when the analysts started, the developers had to sit for some time." When producing technical designs, LSP-A has one architectural design entailing the different components within the whole system which "... in time creates technical debt by lack of design and building on top of the design every time". The detail technical designs are made in the first sprint done in construction phase. According to LSP-A, it "...is not good thing in my opinion. I would like this activity placed in this approach. This gives space to think about design". By ‘this approach’, it has been observed that LSP-A pointed out to the iteration of the service modeling phase.

B. Missing Factor in LSP-A’s process:
1) A separate phase for requirements engineering and designing of services.
2) The requirements engineering and designing is not done in iterations.
   a. LSP-A mentions that the requirement engineering is done in a "...waterfall approach and then handed over to the development team, which is done in Scrum approach". Since, in later parts both analysts and developers work at the same time in different approach, LSP-A mentions "what goes wrong there in the development teams, is that some of the things that are grouped by the analysts will be lost; it loses the promise of reduced time of the project." From this, it is observed that LSP-A understands the need for having a consistency between the phases and recognizes the importance of iterations allowing a connection between the natures of the activities.

C. Distinction in APISD:
1) A separate phase exists for service analysis and design, namely the Service Modeling phase.
   a. LSP-A lacking this phase understood the importance and need of incorporating within their process. They have expressed the need by mentioning; "This phase is something that we are starting to look into right now but the activities are not done in iteration but a part of construction phase. It might be better to see it in a different phase like it is in here."
2) Different technical designs but the designs are not too detailed.

D. Additions for APISD:
1) The service design team can create high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team.

Service Construction Phase:

A. Current Process:
In the development phase of LSP-A’s process, first the business analysts specify requirements for the user stories. Following that, designs are created for the user stories. LSP-A follows Scrum for the development phase with sprints as their iterations. The construction team consisting of developers and testers together reviews user stories from the product backlog and takes it in the sprint backlog. Each sprint results in the completion of the sprint backlog. There is a Scrum Master who is responsible for monitoring the progress of the sprint and solving any impediments. After completion of the sprint, the Scrum Master holds a sprint preview for the client, where the team/team representative and functional analysts are also involved. Following that, the Scrum Master holds a retrospective meeting with the team to review the sprint and identify any improvements that can be done in the following sprints.

B. Similar To APISD:
1) The Service Construction phase of APISD is similar to that of LSP-A’s process. The development and testing also goes through an iteration of 2-4 weeks.
2) The development manager takes the role similar to Scrum Master in LSP-A's process, with additional involvement in Service Modeling phase.

C. Additions for APISD:
1) The demo given to the client after each iteration in APISD can be recorded and shown to the construction team if they are not involved in it. This will increase the satisfaction and gives an impression how the client behaves towards the system.

Provisioning Phase:
A. Current Process:
LSP-A does not have the Provisioning phase as a distinctive phase. The Project Manager and functional analysts department create SLAs that contains a very rough description after deployment phase. LSP-A recognizes the importance of having detailed SLA's and mentions "This phase is completely new for us. The SLAs are created at the beginning but they are not detailed. It’s a good idea to have them done before execution."

B. Missing Factor in Case:
1) A separate phase for structuring and formalizing SLAs with the client.

C. Distinction in APISD:
1) A separate phase exists for the provisioning of the services where the service board produces SLAs and formalizes with the client. The SLAs are detailed with non-functional requirements.

Deployment and Execution Phase:
A. Current Process:
LSP-A does have this process with a long integration sub-phase. The long duration is due to incidents occurring after deployment to the production environment. This causes delay to the release. Once problems are fixed, the services go through a small post-production phase lasting about 2 weeks where monitoring and testing occurs. LSP-A mentions the need of involving the system administrator at a high level of acquiring knowledge on the technical designs and understanding what a system does at an earlier phase. Having this knowledge will allow the system administrator "...to know what to do to keep the system healthy" and "... know what the system is for to better support the service management".

B. Similar To APISD:
1) APISD also goes through a quick post-production test once the integrated services are deployed.

C. Distinction in APISD:
1) Having the service modeling phase with the infrastructure and architecture designs helps reduce the problems occurring in deployment.

D. Additions for APISD:
1) The system administrator should be included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and provide input. Furthermore, the system administrator should also review the corresponding technical designs required for deployment and administering the system. Having an idea about the services will help speed up the problem solving.

Service Management Phase:
A. Current Process:
LSP-A has a management phase where the services are monitored and managed. The system administrator maintains the system. However, the system administrator is not
aware of the user stories being deployed and thus sometimes, it takes longer to pinpoint problems and solve them. The customer service is responsible for communicating with the users of the system and handling change requests and incidents in the corresponding systems. As the customer service is involved in knowing the functional requirements of the services, they are able to support the users much better. In LSP-A, the functional analysts are also involved in visiting the work-floor of the users and observe their system usage. Through this process, they acquire a much better understanding of the users' needs and translate them to change requests for further development.

B. Similar To APISD:
1) APISD follows the same procedures in comparison to LSP-A's process. However, in APISD, the customer service is not aware of the features being deployed in the system.
2) Change requests and incidents are also similarly logged in Change Request Backlog and Incident Backlog.

C. Additions for APISD:
1) In APISD, the customer service should be involved in the Service Modeling phase to review the service backlog produced by the service analysts. They should also input regarding possible concerns of users for the refinement of the customers.
2) In APISD after the construction phase, a high level demo should be 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.
3) There should be a workflow created for the service analyst to also visit the users' work-floor and observe their interaction and engagement with the integrated services.

6.5 Case LSP-B

LSP-B is a single service provider who provides integrated services and solutions. The service provider identifies, defines, governs, executes and manages the integrated services by their own organizational members. They outsource the development of the services to one sub-contractor selected for each project they plan. They do not have a structured process running in the organization but follow a combination of Waterfall approach and some iterative development in terms of releases and feedback loops with the sub-contractor which works smoothly. Although there are several versions of the Waterfall approach, the basic steps were originally described by Winston W. Royce (1970), which is a sequential design process flowing steadily downwards, starting with System Requirements, then Software Requirements, Analysis, Program Design, Coding, Testing and Operations. Their steering board consists of a client representative which indicates that they are in continuous communication with the client. Also, their customer service involved in testing their solutions indicates that customer support is crucial.

Similarly, as discussed in the previous case, the analysis of the comparison between LSP-B's process and APISD will be discussed per phase of APISD and according to the list below. Each phase will consist of the factors below that are applicable to the analysis.

- Current Process: how it is in the current organization process;
- Similar To APISD: common factors between LSP's process and APISD;
- Missing Factor in Case: missing factors existing in APISD but not in the case;
- Distinction in APISD: what is different in APISD from the current process and contributing to the missing factors in case;
- Additions for APISD: factors to be taken in consideration to enhance the APISD artifact.
Planning Phase:

A. Current Process:

In LSP-B, the process starts with the organization members together with the representatives of the industry defining the needs of the businesses and clients. Together with the stakeholders, they discuss the possible solutions and make agreements on what should be done. Each project has one steering group who makes the decisions of the project. Once the needs are defined, the functional designers of the organization create a Project Plan. In this project plan, the basic needs, solutions, estimation and resources are given with the definition of services to be implemented. The steering group reviews this project plan and gives a Go/No Go decision. Once approved, the functional designers create functional designs of the services. The functional designs are reviewed and decided upon by the steering group as well. Requirement collection by the functional designers does not require visiting the clients because there is always a client representative in the steering group. The specification of the requirements depend on the project whether they are specified at one time or split in phases. For larger projects such as governmental projects, the requirements are defined in specific different periods according to the project plan. Meanwhile, the steering group decides on sub-subcontractors for the implementation. Among a choice of 2 to 3, the steering group chooses one. Once all requirements and functional designs are approved by the steering group, the functional designers give the requirements of the services to the sub-contractor. They clarify any questions that the sub-contractor may have.

B. Similar To APISD:

1) Functional designs and requirement specification is done by service analysts in APISD similar to the functional designers in LSP-B. However, such activities are done in the Service Modeling phase.

C. Distinction in APISD:

1) In APISD, there is always one planning group, namely the service board which contains only the members of the service providers in comparison to LSP-B where the steering group consists of client representatives as well.

2) In APISD, the project document is created by the service board in comparison to LSP-B’s process, where the project plan is created by the functional designers.

3) In APISD, the project document consists of the business needs and identification of services with Service backlog defining the services in detail which is created by the service analysts and designers; in comparison to LSP-B’s process, where the project plan consists of the combination of both the project document and service backlog.

4) In APISD, the requirements are defined in the analysis sub-phase in iterations of 2-4 weeks based on a prioritized set of services decided by the service board in comparison to LSP-B’s process, where requirements are defined based on one single span of time or in phases for larger projects.

Service Modeling Phase:

A. Current Process:

LSP-B does not have a separate modeling phase. Once the functional designs are defined by the organization’s functional designers, they are sent to the sub-contractors for deriving the technical designs. Once the technical designs are approved by the functional designs, the construction of the services begins. Although there is no separate phase, this process acts as a sub-phase of the Construction phase where there is communication between the two parties regarding clarification and decision making. LSP-B has this process integrated with Construction phase because they outsource their development and is not associated with how the outsourcing party implements. However, there is continuous communication between the functional designers and sub-contractors through feedback loop.
B. Missing Factor in LSP-B’s process:
   1) A separate phase for requirement specification and designing.

C. Distinction in APISD:
   1) A separate phase exists for the requirement specification and designing is done.

Service Construction Phase:

A. Current Process:
In LSP-B, the construction phase starts by producing the technical designs by the sub-contractor and asking for approval from the functional designers. Once approved, the development begins. The project is wholly managed by the organization’s project manager who maintains the communication with the sub-contractor. This phase is done in a Waterfall approach where deliveries are done in release milestones and not following sprints as in Scrum. The milestones are defined according to the deliveries set in the project plan. The sub-contractor operates in her own process and delivers the services once completed. During the development, the organization can view and monitor the progress and issues through a shared system. If there are significant changes during the development, the steering group is involved in decision making. The sub-contractor has test databases of the organization, which is used for testing. Once completed, the sub-contractor deploys the first release of integrated services in the organizational system where checks are done corresponding to the project plan ensuring qualitative deliverables. The test reports are communicated to the steering group. Releases depend on the project size. Some release can be after 2 to 3 months or 1.5 to 1 year. Once deployed, testing is also done by the customer service department. This practice is to educate the customer service department for providing better support to the users. After approval of the release, the integrated services are presented to the client. For large projects, the demo days are defined where group presentations are given. For a set of common services to be given to multiple clients, the clients are invited in those group presentations so that they can discuss with each other. For small projects, some client’s representatives are invited for the demo held in small sessions. In either of the scenarios, the clients give feedback and decide to approve or reject the release. If rejected, feedback on improvements are given, which are given to the sub-contractor to develop them.

B. Similar To APISD:
   1) In APISD during the development of the services, if there are any changes, the service board is communicated for any decision making as well when compared to LSP-B’s process.
   2) In APISD, the checks of the integrated services are based on the service and feature backlog, similar to LSP-B’s process where the checks are based on the project plan.
   3) In APISD, the integrated services are deployed in a system of the appointed service provider (in case of multiple service providers), similar to LSP-B’s process, where the integrated services developed by the sub-contractor are delivered in the service provider’s system.

C. Missing Factor in LSP-B’s process:
   1) A separate phase for requirement specification and designing is not there in LSP-B’s process.
   2) The requirements engineering, designing and development are not done in iterative development. In LSP-B, as the requirements are specified first and then construction begins, which is of a nature of waterfall approach, LSP-B does not incorporate iterations but rather a sequential milestone approach.
D. Distinction in APISD:

1) In APISD, the development manager of the service provider manages the designing and development phase in comparison to the project manager managing the whole project.

2) In APISD, the Service modeling and Service Construction phase is done in a short iteration of 2-4 weeks in comparison to LSP-B’s process, where the duration of the milestone varies according to the project.

3) In APISD, the service design team reviews the functional requirements and derives the technical designs based on the prioritized services from the service backlog; in comparison to LSP-B’s process, where the sub-contractors receive the services set for the project at one time-span or in release milestones.

4) In APISD, the releases are set based on the prioritized services and vary according to the release manager in comparison to LSP-B’s process, where the releases are pre-set in the project plan in the beginning.

5) In APISD, there is a possibility to involve the clients in a mid-demo to give feedback on the ongoing development of services, in comparison to LSP-B’s process, where the clients are only involved in providing feedback once the release is presented.

Provisioning Phase:

A. Current Process:

In LSP-B’s process, during the provisioning phase, as soon as the first release has been decided upon, SLAs are created on what is expected from the service. The SLAs are formalized with the client. In the planning phase, rough drafts are created and detailed versions are agreed upon in this phase.

B. Similar To APISD:

1) In APISD, this phase is identical for LSP-B’s process.

Deployment and Execution Phase:

A. Current Process:

In LSP-B, once the first release is finalized, the integrated services are deployed in the production environment. A period of checking in post-production exists in this phase where the services are monitored. If any problems occur, they are fixed immediately. If for any particular reason, big incidents occur where the client declines the release; the process is evaluated by the service provider and analyzed for potential reasons and solutions.

B. Similar To APISD:

1) In APISD, this phase is identical for LSP-B’s process. However, the possibilities of the client rejecting the release once it is in production are less because the client is involved in mid-previews and end-previews. Also, potential problem areas of the production environment are defined in the technical designs that are produced in the Service Modeling phase.

C. Additions for APISD:

1) In case of rejection by the client after post-production, an activity should be included in APISD for analyzing the problems and producing possible solutions.

Service Management Phase:

A. Current Process:

In LSP-B, service management is done by the service provider’s own IT department under the leadership of the project manager. The project manager is responsible for tracking the change requests submitted by the users through the customer service. If
changes are required to be implemented, the sub-contractors go through the process of the construction phase. The service management follows ITIL standards.

B. Distinction in APISD:
1) In APISD, the customer service is responsible for maintaining change requests and incidents and the release manager filters according to priority for further planning of the implementation of services; in comparison to LSP-B's process where the project manager is responsible for tracking change requests and managing the implementation.

C. Additions for APISD:
1) APISD should incorporate the practice of service provider(s) engaging in adopting best practices and standards for service management.

6.6 Case PSP

PSP is a service provider who develops integrated services on their own and with other service providers. For internal development, they identify, define, develop, govern, execute and manage the integrated services by their own organizational members. For collaborative development with other service providers, they develop, execute the integrated services with the other organizational members. The other service provider(s) are then responsible for identifying, defining, developing, governing, executing and managing the integrated services by their own organizational members throughout the phases. In PSP, there is a collaborative environment where requirements and change requests are captured and translated into user stories, issues are managed and client feedback is managed. This environment is managed by the host provider and available for the other provider based on the projects involved and sometimes to clients involved in corresponding projects. The methodology followed by this service provider is Scrum. Scrum is only applied in the development of the applications within the teams. It is also followed when other service providers are involved in the development if agreed upon. If other partners do not choose to follow Scrum, they tailor the process according to the way that is needed to work.

As discussed in the introduction of this section, the analysis of the comparison between PSP's process and APISD will be discussed per phase of APISD and according to the list below. Each phase will consist of the factors below that are applicable to the analysis.

- Current Process: how it is in the current organization process;
- Similar To APISD: common factors between PSP's process and APISD;
- Missing Factor in Case: missing factors existing in APISD but not in the case;
- Distinction in APISD: what is different in APISD from the current process and contributing to the missing factors in case;
- Additions for APISD: factors to be taken in consideration to improve the APISD artifact.

Planning Phase:

A. Current Process:
In PSP, the process starts with the figuring out the aim and objectives and assessing the opportunities. The process consists of 4 phases which share similar activities with the APISD Planning phase. Therefore, these four phases will be described in this phase. First, the opportunities are identified in the Sales department. Here the client performs a market analysis and defines what is needed. Second phase involving scoping the needs of the business and the feasibility of developing the requirements of the clients. Here, first the project manager assigned to the specific project communicates with the customer to gather requirements. In agreement with the project manager, the customer defines the high-level user stories/features to be built in applications and plans release dates. In case of multiple service providers, the project manager always belongs to the partner organization. Third phase is the
financial negotiations made with the clients regarding the services to be provided. Once that is negotiated, the project manager drafts the project plan containing necessary deliverables and duration and a project kick-off is arranged.

B. Distinction in APISD:
1) APISD contains one phase, Planning Phase, which includes the activities of the PSP’s process phases Sales, Scoping Financial Negotiations and Project Kick-off.
2) The market analysis in APISD is done by the client together with the service board, whereas in PSP’s process, the market analysis is done by only the client.
3) APISD has a service board which is responsible for the planning of the projects, whereas in PSP’s process, there is no board but rather a project manager who plans and manages the project.
4) In case of multiple providers working together, in APISD, the service board consists of representatives of providers. Whereas in PSP, the governing is done by the project manager of the partnering organization and not the host organization.
5) In APISD, the service analysts are responsible for collecting and specifying requirements in the Service Modeling phase, whereas in PSP, the project manager is responsible for collecting and clients for specifying high-level requirements in form of user stories in the Scoping phase.

Service Modeling Phase:

A. Current Process:
In PSP, the process that serves the service modeling phase is sprint preparation where the project manager tries to determine the requirements in detail for the first sprint back-log. Once the requirements are set, the sprint begins directly with the construction of the applications. The reason that they do not require a separate construction phase is because this phase consists of producing domain models in UML that are executable with business logic with BPEL. In other words, they produce executable applications automatically from models created. When the executables are completed as prototypes, clients are required to test the prototypes within the sprint and provide feedback using their collaborative system. If there are feedbacks to be implemented, one team member is assigned to implement the feedback while others engage in the application itself. As soon as the applications are built, the project manager gives a product demonstration to the client.

When collaborating with other partners in this phase, as the project manager belongs to the partner organization, the developers also known as business modelers in PSP’s process are involved in the development of applications together with the development team of the partner. The development phase followed by all partners in this phase is done according to Scrum development if agreed upon in the initial phase.

B. Similar to APISD:
1) The common step that APISD also follows in comparison to PSP’s process is that there is also detailing of the requirements done for the first set of prioritized services as PSP also does for the first sprint.
2) APISD also involves client to test the features developed and provide feedback when demonstrated.

C. Missing Factor in PSP’s process:
1) The sprint preparation in PSP is not done in iterations as service modeling is done in APISD.

D. Distinction in APISD:
1) The service modeling phase is separate from the construction part in APISD, however it is understandable for PSP since they directly convert models into executable applications without any effort in programming or implementing the services.
2) In case of multiple service providers collaborating together, in APISD, the domain manager is responsible for looking after the service analyst and service design team. The development managers of each provider is responsible for managing and looking after their own construction teams and meeting each other weekly for solving dependency issues and the service board is responsible for the overall planning and management of the projects. Whereas in PSP, the project manager of the partner is responsible for managing and looking after the development of the applications.

E. Additions for APISD:
1) As PSP involves one team member to focus on the feedback implementation allowing the rest of the team members to focus on the rest of the development, APISD can also adopt this practice.

Service Construction Phase:

A. Current Process:
1) There is no separate phase for construction in PSP, as previously explained, the sprint preparation and sprint are executed directly from modeling to applications without programming it. It can be seen however the sprint preparation behaves as the service modeling phase and sprint as the construction phase. However, PSP does not distinguish these two as separate phases because as a sprint is going on, the project manager prepares for the next sprint.

B. Distinction in APISD:
1) The construction phase is treated as a separate phase to distinguish the nature of the activities and responsibilities of parties involved.

Provisioning Phase:

A. Current Process:
PSP does not have a separate phase for provisions. The negotiations are done in the beginning of Financial Negotiation phase. Once the applications are ready to be deployed, the Professional service department formalizes the SLAs following some templates. These templates are adapted to the client’s needs and demands. Depending on the availability and the size of the project, the SLA is altered by the delivery manager from the Professional Service department.

B. Missing Factor in Case:
1) A separate phase to conduct all formalization of the SLAs is missing in the PSP’s process. PSP feels the importance of this phase: “We don’t have a separate phase and although we do negotiate in the beginning, I feel as our company is growing there is need to have detailed SLAs to be formalized at a later stage.”

C. Distinction in APISD:
1) A separate phase exists to conduct all formalization of the SLAs with the clients by the service board as they look after the whole process.

Deployment and Execution Phase:

A. Current Process:
1) PSP has a similar phase, namely, Delivery phase, where they deliver the applications to the client. In this phase, the applications are deployed in their own cloud or other cloud systems. Once deployed, the Sales department informs the clients.

B. Similar to APISD:
1) APISD has similar phase where deployment and execution of the services are done.
C. Distinction in APISD:
   1) In APISD, the customer service informs the client regarding the deployment of the services, in comparison to PSP’s process, the Sales department informs.

   **Service Management Phase:**

   A. Current Process:
      PSP calls this phase as Support phase or Phase 2 of the project. For change requests, the client provides feedback or request in their collaborative system. The client translates their requirements into the user stories which are later discussed with the project manager to be implemented in the sprints. In case of incident management, PSP has a support manager who processes all incidents. These indices are assigned to one of the developers from the original project team.

      When collaborating with multiple providers, usually the host organization takes the lead to fix issues. As their partners are slower in fixing or adapting to change, the host organization takes the lead to change rather than waiting.

   B. Similar to APISD:
      1) Regarding client feedback, APISD also incorporates the ability to allow client to provide feedback to the customer service for issues or change requests.
      2) Similar to PSP, in case of fixing issues, the issues are assigned to the original project team in APISD.

   C. Distinction in APISD:
      1) In APISD, the service analysts are responsible for translating the change requests and feedback to requirements for services, in comparison to PSP’s process, where the client defines and translates.
      2) In APISD, the customer service and release manager is responsible for managing incidents with the release manager responsible for filtering them for further development; in comparison to PSP’s process where there is one support manager performing those activities.
6.7 Synopsis of Analyses

From the above evaluation of the three cases, various aspects have been investigated. Though the above analyses include valuable information for this research, it is further required to distinguish the level of information and deepen the overall analysis. Therefore, this section summarizes the information gathered above, on which are valuable and essential to this research.

To recapitulate, LSP-A’s process consists of the phases: Planning, Construction, Deployment and Service management. There is no separate phase for Service modeling and Provisioning, although the activities are spread out in other phases. Also, there is no iterative development followed in requirements analysis compared to APISD’s Service Modeling phase. LSP-A recognizes the need for these phases and concur that these phases are important to incorporate within their organization. From the evaluation of this case, additional factors have been identified that will further enhance the initial model for APISD. They are:

- Looking at the need of differentiating the responsibilities of the strategic board and tactical board, in APISD the service board can be divided in two sub boards with distinction in responsibilities. This alteration is essential for this process because having this distinction will allow a clear division and understanding of responsibilities according to the strategic and operational nature, which supplements the organizational integration activities;

- The service design team can create high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team. This input is valuable for this process because it further enhances and creates distinction of the activities;

- The demo given to the client after each iteration in APISD can be recorded and shown to the construction team if they are not involved in it. This enhancement is a nice-to-have for this model because by this practice, it will increase the satisfaction of the teams and gives an impression how the client behaves towards the system. It is however not so crucial to perform always in the process;

- The system administrator should be included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and provide input. Furthermore, the system administrator should also review the corresponding technical designs required for deployment and administrating the system. This input is essential to the process because having an idea about the services will help the system administrator speed up the problem solving;

- In APISD, the customer service should be involved in the Service Modeling phase to review the service backlog produced by the service analysts. They should also input regarding possible concerns of users for the refinement of the customers. This input from the case study is essential to the enhancement of the model because including this input will allow the organization to have a consistent understanding of the services deployed;

- In APISD after the construction phase, a high level demo should be 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. This input is valuable for the process because providing a demo will allow both actors to gather a deeper understanding of the releases;

- There should be a workflow created for the service analysts to also visit the users’ work-floor and observe their interaction and engagement with the integrated services. This input is essential to the process, because it will allow organizations to gain a competitive advantage by meeting client needs better.

LSP-B’s process consists of the following phases: System Requirements, then Software Requirements, Analysis, Program Design, Coding, Testing and Operations. LSP-B relates to the APISD model in similar activities but in different naming convention. However, there is no separate phase for requirements analysis in iterations, although the activities are performed in the planning phase compared to the Service Modeling phase. During the interview, LSP-B
has mentioned the services and products they provide. These services and products are not discussed in this research because this information is not valuable for this process itself. LSP-B indicates that the process followed is iterative development and is similar to the activities followed in APISD. However, from the interview and information gathered, it has been observed that the organization follows a Waterfall approach with slight moderation in feedback loops incorporated in the Program Design and Coding phase. In these phases the sub-contractor and the functional designers communicate frequently. Although, this case does not follow iterative development, APISD can still be incorporated for similar cases given that the organization is interested in iterative development. From the evaluation of this case, additional factors have been identified that will further enhance and solidify the APISD initial model. They are:

- In case of rejection by the client after deploying the service package in Deployment and Execution phase, an activity should be included in APISD for analyzing the problems and producing possible solutions. This input is valuable for the process because having a retrospect allows the organization to understand and analyze the problems which can be avoided in the future;
- APISD should incorporate the practice of service provider(s) engaging in adopting best practices and standards for service management. This input is valuable for the process because incorporating best practices, organizations can become more efficient.

PSP’s process consists of the following phases: Sales, Scoping, Financial negotiations, Project Kick-off, Sprint preparation, Sprints, Delivery and Support. Comparing to APISD model, PSP does not have a separate Service Modeling phase because the requirements are directly converted to executable functionalities. Thus, these activities are done in their Sprints, similar to APISD’s Construction phase. However they do have a Sprint Preparation phase to specify requirements for their user stories but this is not done in iterations. Furthermore, there is no separate Provisioning phase but the activities are roughly done in Financial negotiations phase. PSP recognizes the need of detailed SLAs and a formalization of the governance of services. In addition, PSP encourages this research by mentioning: “It is good that the roles are well described and process in defined in detail. Perhaps, we can make use of it in our organization”. PSP has demonstrated a system that they use for collecting user stories and managing the sprint planning. This information is not discussed in this research as it is not valuable information for the process itself. From the evaluation of this case, additional factors have been identified that will further enhance and solidify the APISD initial model. They are:

- As PSP involves one team member to focus on the feedback implementation allowing the rest of the team members to focus on the rest of the development, APISD can also adopt this practice. This input is nice-to-have for the process and not so essential because organizations can vary on how they operate. However, having this practice allow division of attention and focus to continuously provide on-going services.

6.8 Summary

In this chapter, a case study had been conducted that evaluates the initial conceptual model of APISD. The case study consists of three cases that were related to the processes followed at three different organizations. The first case, LSP-A, has been conducted at a single service provider following Agile development method, Scrum, for the development of the integrated services. The second case, LSP-B, has been conducted at another single service provider who has outsourced their development of services. They follow a mix of Waterfall and iterative development throughout the whole process excluding development. The third case, PSP, has been conducted at a service provider who delivers services by themselves and also in collaboration with other partners. They follow Scrum for their own developed projects and also in projects with other partners if agreed to.
The case study has been classified as a multiple-case study design because three cases have been distinguished based on the same design. The multiple case study design has one single unit of analysis, the process. Different sources of evidence were used to collect the data for this research. Sources are from website documentation, interviews including audio recordings, a questionnaire and e-mail correspondence. The validity of the case study findings have been explored with relevance in construct, external validity and reliability.

The cases were analyzed in comparison to APISD where particular factors were looked at: the current organization process, similarity with APISD, missing factors existing in APISD but in the case, distinction in APISD. Through the evaluation, additional factors have been identified to further enhance the APISD model. Finally, the chapter concludes with a synopsis of the cases analyzed and summarizes the information valuable and essential to the enhancement of the initial conceptual model. This chapter completes the DSRM step ‘Evaluation of Artifact’. The next chapter will highlight the additional factors referring to the cases and re-model APISD as the final model with its necessary explanations.
Chapter 7

Final Conceptual Model for APISD
7 Final Conceptual Model for APISD

In the previous chapter, through the case study research, the initial conceptual model for APISD has been evaluated and the validity of the study findings was explored. With the completion of the DSRM step ‘Evaluation of Artifact’, this chapter begins with the final DSRM step ‘Communication’ (shown in Figure 2) where the results are discussed defining the final conceptual model. This final step consists of three parts discussed in three chapters (shown in Figure 3). The first part will be followed in this chapter deriving the final conceptual model. The subsequent parts will be entailed in the later chapters. Recall the evaluation of the case study research in the previous chapter, where several differences and additional factors have been identified crucial to the re-modeling of APISD. This section will incorporate those additions and create a final conceptual model for APISD firstly. Secondly, the different phases of the lifecycle will be discussed in detail. Chapter 5 will be revisited regarding the details with the improvements distinguished. This section serves as the final version of the model.

7.1 Enhancements for APISD Lifecycle

The additional factors for the APISD lifecycle identified in the evaluation in section 6 are listed below according to phases. Each enhancement also contains the insight from the cases explored and the difference with APISD.

A. Planning Phase

1) In APISD, the service board should be divided in two sub boards with a distinction in responsibilities serving a strategic and tactical nature, namely the Executive Board and Service Board.

   - This enhancement is retrieved from an insight of LSP-A, where it consists of two separate boards, a strategic and tactical board (see section 6.4-Planning phase-A and C2). Previously in the initial conceptual model, APISD consisted of only one planning board, the service board, which is responsible for looking after all the planning activities. After understanding the need of the strategic and tactical nature, an additional board, the executive board, is added to the model. This distinction is made so that the executive board can look after the objectives of the business while the service board looks after the services to be implemented to fulfill the business needs.

B. Service Modeling Phase

1) The service design team should create high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team.

   - This change is retrieved from an insight of LSP-A, where there was a suggestion for creating multiple high level designs for various components, deployment and infrastructure (see section 6.4-Service modeling phase -A and C2). APISD consisted of different high level designs but too detailed, so from the suggestion, the level of detailing is changed in the final conceptual model.

2) In APISD, the customer service should be involved in the Service Modeling phase to review the service backlog produced by the service analysts. They should also input regarding possible concerns of users for the refinement of the customers.

   - This enhancement is retrieved from an insight of LSP-A, where they involved customer service in knowing the functional requirements of the services and thus suggestion was made to involve them in reviewing the service backlog (see section 6.4-Service Management phase-A and C1).
Previously in the initial conceptual model for APISD, the customer service was not at all involved until service management phase.

3) The system administrator should be included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and provide input. Furthermore, the system administrator should also review the corresponding technical designs required for deployment and administrating the system. Having an idea about the services will help speed up the problem solving.
   o This change is retrieved from an insight of LSP-A, where they suggested involving the system administrator in reviewing the service backlog (see section 6.4-Deployment Execution phase D1). Previously in the initial conceptual model for APISD, the system administrator was not at all involved until deployment and execution phase.

C. Service Construction Phase

1) The demo given to the client after each iteration in APISD can be recorded and shown to the construction team if they are not involved in it. This will increase the satisfaction and give an impression how the client behaves towards the system.
   o This change is retrieved from an insight of LSP-A, where they involved the team in production demonstration to the client. A suggestion was made that if team members cannot attend, the meeting can be recorded (see section 6.4-Service Construction phase-A and C1). Previously in the initial conceptual model for APISD, it was optional for the team members but this suggestion is added in the final conceptual model as a best practice.

2) In APISD after the construction phase, a high level product demonstration should be 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.
   o This enhancement is retrieved from an insight of LSP-A, where they suggested involving the customer service and system administrator for a high level product demonstration to familiarized with the services to be delivered (see section 6.4-Service Management Phase-A, B 1) and C2). This way, they can support the customers better. Previously in the initial conceptual model for APISD, the system administrator and customer service was not at all involved until deployment and execution phase and service management phase. Involving them in this activity can result in better support and therefore is added in the final conceptual model.

3) When there is issues to be solved that are urgent given by the client for existing services, one team member should be involved in implementation of that issue, while the rest of the team members should focus on the rest of the service development.
   o This enhancement is retrieved from an insight of PSP, where only one team member to implement the feedback from the client (see section 6.6-Service Modeling Phase-A and E1). Previously in the initial conceptual model for APISD, it was only defined that the original team is assigned to implement the feedback. However, since the benefit is seen from the case where the other team members can focus in the development instead of every team member focusing on the feedback implementation, this is adapted in the final conceptual model.

D. Deployment and Execution Phase

1) In case of rejection by the client after post-production, an activity should be included in APISD for analyzing the problems and producing possible solutions.
This enhancement is retrieved from an insight of LSP-B, where once the client is exposed to the deployed services, they can reject the release and this requires the organization to analyze the problem and produce possible solutions (see section 6.5-Deployment and Execution Phase-A, B1) and C1). This way, the organization has a possibility to analyze if something goes wrong in post-production. Previously in the initial conceptual model for APISD, the possibility of rejection is less because the customer is heavily involved during the analysis and development of services. However, it is good practice to consider worse-situations where the client may reject and thus this activity is adapted in the final conceptual model.

E. Service Management Phase

1) There should be a workflow created for the service analysts to also visit the users’ work-floor and observe their interaction and engagement with the integrated services.
   - This enhancement is retrieved from an insight of LSP-A, where they send their functional analysts to the user’s work-floor in order to understand their needs better (see section 6.4-Service Management Phase-A, B1) and C2). This way, the organization can support the customers better. Previously in the initial conceptual model for APISD, there was only one way to provide feedback, which is users requesting the customer service. To have a competitive advantage, organizations can adopt this activity to get a firsthand look at user’s needs. Therefore, it is adapted in the final conceptual model.

2) APISD should incorporate the practice of service provider(s) engaging in adopting best practices and standards for service management.
   - This enhancement is retrieved from an insight of LSP-B, where they follow ITIL standards (see section 6.5-Service Management Phase- A and C1). Previously in the initial conceptual model for APISD, there was no particular standards mentioned that organization can follow. It can be a best practice to follow standards to manage services in a structure manner and therefore, it is adapted in the final conceptual model

The identified enhancements listed above will be incorporated in the APISD phases in details in the following section.

7.2 Final Conceptual Model for APISD

With the enhancements listed in the previous section, the final conceptual model for APISD will be re-modeled in this section. The supplements will be distinguished in the details and illustrations referencing to section 7.1. In order to fulfill the final model, section 5 will be revisited.

With all the necessary components determined from the section 5.1, the initial conceptual process model of APISD was formed, and it is again illustrated below serving as the framework for the final conceptual model for APISD. The model entails the lifecycle of APISD and will further be discussed.
The APISD lifecycle consists of six phases, which also have been determined in earlier: Planning, Service modeling, Development and Testing, Provisioning, Deployment and Execution and Change management. After the evaluation of case study findings, there is no significant change in the lifecycle itself. The phases remain the same but the inner-workings of the process entail improvements, which will be discussed in their own respective areas. Each of these phases consists of a certain number of activities performed by certain roles and is described extensively in this section. The lifecycle first starts with Planning phase where the planning and feasibility study of the service delivery is done. Following the Planning phase, the Service Modeling begins. As depicted in the figure, this phase consists of two sub-phases: Analysis and Design, in which each of the sub-phases goes through iterative development. Analysis deals with the analysis of new and existing services and specification of functional and non-functional requirements. The Design sub-phase deals with the designing of the architecture, and resource planning for the development of services. The phase transitions to the Service Construction phase where the actual development and testing occurs. After the services are integrated, the system goes through user acceptance test. Once approved, the Provisioning phase starts where the Service Level Agreements (SLAs) are formalized. Following the confirmation by the client on SLAs, the integrated service package is deployed in the production environment and made available for clients to use. In the process of usage, services are monitored by the service providers and change requests/incidents are handled by the client service. These are logged in a backlog, where the service board can once again analyze and plan the next set of services to deploy. The phases of this lifecycle are further elaborated in the following sections, discussing the process flows and communication among the roles.

7.2.1 Roles in APISD

Following the differences in the case study findings, the need to include certain roles in certain phases was realized. In correspondence to the improvements listed earlier in this chapter, a new role, Executive Board, has been added in APISD and other roles have been introduced in some phases. In comparison to the roles illustrated in the initial conceptual model (shown in Figure. 5), the following figure displays the finalized version of the roles involved in respective phases with the addition of new and existing roles distinguished in a red color. The roles are further described subsequently.
The *Executive board* (enhancement in section 7.1: A1) is the committee that governs the set of processes, organizational structure. The executive board members vary for the following scenarios:

- For a single service provider delivering integrated services, the executive board consists of members of that single organization defining the organization’s mission and vision.
- For multiple service providers delivering integrated services, the service board consists of executives of the various service providers coming together and defining the consortium’s mission and vision.

The executive board is responsible for understanding the business environment and making sure that all necessary controls are incorporated in the APISD framework. They define the organization’s mission, vision and objectives in a vision document (see section 7.3), and delivers to service board. This committee is responsible for providing the service board a direction to what should be achieved and how to approach future possibilities. Key involvement that the executive board is involved in is setting up the vision document in Planning phase, reviewing Service Level Agreements in Provisioning Phase, and decision making if required in any point in the APISD lifecycle.

The *Service Board* is the committee that governs and decides on the services to be developed that are essential for delivering on the ISD promise. The service board members vary for the following scenarios:

- For a single service provider delivering integrated services, the service board consists of members of that single organization governing the rollout of services and is central to decision making within the APISD lifecycle.
- For multiple service providers delivering integrated services, the service board consists of members of the various service providers collaborating together and governing the rollout of services and central to decision making within the APISD lifecycle.

The service board is responsible for understanding the business environment and making sure that all the necessary controls are incorporated in the APISD framework. Activities such as identifying services required in discussion with the client, analyzing the business needs, analyzing financial aspects and making decisions as required for the ongoing development of integrated services (Papazoglou & Van Den Heuvel, 2006). Key involvement that the service board is involved in, is performing all activities in Planning phase, decision making in Service
Modeling phase, setting release dates in Service Construction Phase, formalizing Service Level Agreements in Provisioning Phase, and decision making if required in Execution and Service Management phase. For decision making regarding potential investments, they are responsible for communicating with the executive board.

A Service Analyst is an analyst who considers the business objectives and defines the services identified by the service board. The service analyst is responsible for creating a service backlog (see section 7.3). An organization may have a single service analyst or a team of analysts. The team may be of either of the following nature:

- For a single service provider delivering integrated services, the service analyst team consists of analysts of that single organization working together to perform the necessary activities.
- For multiple service providers delivering integrated services, the service analyst team consists of analysts of the various service providers coming together to perform the necessary activities.

While working closely with the service design team, the service analyst team is responsible for specifying functional and non-functional requirements for every service from the service backlog. This team is solely responsible for maintaining the service backlog. Other responsibilities include: communicating with clients for requirement specification, clarifying requirements to other developers and testers, participation in service demonstration to clients, and communicating with service board for prioritizing of services and alignment to business needs. Key involvement that the service analyst is involved in is performing all activities in Service Modeling phase, clarifying requirements in Service Construction Phase. Another activity of the service analyst is to visit the work-floor of the users in the Service Management phase, to better understand the engagement with the services and translate their needs to change requests (enhancement section 7.1 E1).

The Development Manager is a role that is involved in both the Service Modeling and the Service Construction phase. In the Service Modeling phase, this role is responsible for working together with the chief service developer/architect to produce technical designs. For requirement clarification, this actor communicates with the service analyst team. Finally, creating a feature backlog (see section 7.3) with the chief service developer/architect is also an output of the Service Modeling phase. The development manager is solely responsible for maintaining the feature backlog and assigning features to the development teams for developing the services. This role is responsible for introducing the feature backlog to the respective team, coordinating the integration process, handling team issues and dependencies, and ensuring that the development and testing process runs smoothly in the Service Construction phase. Apart from the various activities, the role acts as a Scrum Master (Schwaber & Beedle, 2001; Stober & Hansmann, 2010) for the construction team. Furthermore, this role is responsible for verifying that the functionalities of the features are developed according to the specifications and arranging a preview meeting with the client. Therefore, it is important that the development managers are both technical and business oriented. With the release manager, the development manager sets release dates and therefore, it is also crucial that the development manager is aware of the progress of the iteration in terms of time and planning. Key involvement that the development manager is involved in is performing all activities in Service Modeling phase, managing the development in Service Construction Phase, monitoring deployment in Deployment & Execution.

A Chief Service Developer/Architect is a member of the Service design team in the Service Modeling phase. This role is responsible for deriving technical designs per service from the service backlog along with the development manager. In the Design sub-phase, the role is initially responsible for the high-level estimates of development and handing them over to service analysts. Then other responsibilities include high level designing and estimating the components for each of the services, such as portal, integration, infrastructure, data, policy, and business (logical) services (Durvasula et al., 2006). For requirement clarification, this actor communicates with the service analyst team. Once the high-level architecture is completed, the architect along with the development manager produces features per service
and creates a feature backlog. If required, high-level designs are produced for the features. This role is also responsible for taking part in the development of the integrated services in the Service Construction phase. If there are multiple service providers involved, this role collaborates with other Chief Service Developers/Architects of the other providers.

The Domain Manager is responsible for ensuring the alignment between the service analyst and the service design team in the service modeling phase. This role can be appointed by the service board or selected from the service modeling teams. As the service analyst team will be involved in defining requirements and service design team for deriving the technical designs, it is important that both teams are in-sync with what should be implemented. Therefore, the domain manager is responsible for holding meetings for alignment; to handle any conflicts that may occur in the service modeling team and ensuring the quality of the output products. The domain manager is also responsible for holding retrospective meetings for the service analyst and service design team, in order to review the previous iterations and to identify key improvements to be done in the future iterations.

The Service Developer is responsible for performing activities related to the realization of services in the Service Construction phase. Responsibilities include reviewing features and the service backlog and understanding the requirements, providing an estimation for development, developing and executing according to planning, and coordinating with the service testers for testing and solving bugs and issues in the services. In the case of multiple service providers, if required, the service developer collaborates with other service developers of other providers during integration of the services. For any conflicts or dependency impediments, the service developer communicates and reports to the development manager. Key involvement that the service developer is involved in is performing all activities in Service Construction phase, fixing any issues in Deployment & Execution phase and fixing urgent incidents in Service Management phase.

The Service Tester is responsible for performing activities related to the testing of services in the Service Construction phase. Responsibilities include reviewing features and the service backlog and understanding the requirements, providing estimation for testing, testing according to planning, coordinating with the service developers for bugs and issues reporting and tracking in the services developed, signing off service construction. For any conflicts or dependency impediments, the service tester communicates and reports to the development manager. Key involvement that the service tester is involved in is performing all activities in Service Construction phase, testing in Deployment & Execution phase and testing urgent incidents solved in Service Management phase.

The Release Manager is responsible for managing releases of the ISD. In the Service Construction phase, this role discusses with service board and development manager to set release dates and the services to be deployed. Together with the development manager, the release manager is responsible for incorporating a ‘freeze’ after development and testing of the integrated services, so that no more development or testing is done that could potentially defect the ‘ready’ system. This role also decides if other releases can be included with the planned release. In the Service Management phase, the release manager can go through the change request backlog (see section 7.3) and give a primary indication for implementation. This set can be further analyzed by the service board to plan accordingly for the next set of services to be developed.

The System Administrator is responsible for all the activities required in deployment of the integrated services and management of systems and services. Responsibilities include: establishing systems environment consisting of development, system integration testing, performance testing, user acceptance, and product environments; assisting service development teams in systems and application configuration, periodic builds, and capacity planning; tracking and managing dependencies among services and assets; deploying and managing business services in production, and providing application support for business services based on business priority (Durvasula et al., 2006). The system administrator is also responsible for reviewing the technical designs related to deployment and infrastructure
focusing on the non-functional requirements from the feature and service backlog defined in Service Modeling phase (enhancement section 7.1B3). Once acceptance has been given by the client regarding a release, the system administrator should be involved in participating in a high-level demo arranged by the development manager to gain knowledge on the integrated services to be released (enhancement section 7.1C2).

The Customer Service is responsible for supporting the client/user's engagement with the integrated services. This team is involved in activities related to client support, handling change requests and logging in incidents. They are solely responsible for maintaining the change request backlog and incident backlog (see section 7.3). The customer service department should also be involved in reviewing the functional requirements of the service Backlog in Service Modeling phase, in order to input possible concerns of the users (enhancement section 7.1B2). Furthermore, they should be involved in participating in a high-level demo arranged by the development manager to gain knowledge on the integrated services to be released to better support the users (enhancement section 7.1C2).

The Client is the potential buyer of the integrated services that the service provider(s) provide. This actor is responsible for discussing with the service provider(s) about the desired needs, which are later translated to functional services, and specifying the desired requirements for those services. Once services are developed, the client previews the solution and either accepts or rejects. The client is also responsible for reviewing the SLAs produced by the service provider(s) and to agree or disagree with the SLAs. Usage of the services is either done by purchasing or renting the services based on the charging mechanisms set in the SLAs. The client can report back to the service provider(s) regarding any change requests or incidents.

The User is someone who actually uses the integrated services. This role could be someone from the client's organization or as customers of the client. The user uses the integrated services and can report via the client or directly to the service provider(s) regarding any incidents or change requests.

7.2.2 Planning Phase

The Planning phase consists of activities that allow businesses to analyze the business needs and market requests, and to determine the vision and objectives, and identify the services required to fulfill them. 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 executive board and service board (enhancement 7.1 A1), shown shaded in Figure. 16. This addition is a significant difference from the initial conceptual model's planning phase.

![Figure. 16. Planning Phase of final model](image)

As shown in the figure above, the executive board meets the service board to explain their mission, vision and objectives. Furthermore, they provide a direction to what should be achieved and how to approach future possibilities by providing them a vision document (see section 7.3). The service board meets with the client and discusses various aspects of the services to be developed. As both service development and Agile development encourages
customer involvement, it is important to maintain close collaboration with the client for better understanding their needs and providing the desired services. Along with the activities mentioned in Market scan and planning phase in section 3.3, they also determine the nature of the services, the scope and the feasibility of developing them. The service board drafts a project document (see section 7.3) and delivers to the executive board. They make moderations according to the feedback given by the executive board. Once approved, they deliver this document to the service analyst team who will start with the analysis of the project. In this phase, the service board also drafts rough SLAs that deal with the service expectations. It is important to start with defining the SLAs in the beginning because the company can already think about the critical aspects of providing the services to the client and begin early negotiations. They can further formalize it in the Provisioning phase once service package is completed. In the case of multiple service providers in the serving the board, the executive board drafts SLAs for their own governing responsibilities together with the service board. 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, lack of coordination will arise throughout the lifecycle. As a result, service providers will not be able to collaborate smoothly or gain trust, which is required for sustainable development. For a detailed understanding, a sequence diagram of this phase is given in Appendix II Figure 1.

7.2.3 Service Modeling Phase

The main objective of the service modeling phase is primarily to describe the services identified in the planning phase. The phase consists of two sub-phases, Analysis and Design (a combination of defining and design activities mentioned in Requirements Engineering, Business Modeling and Service Design phases in section 3.3 and Planning in section 4.3) and is carried out by the service modeling team. The team comprises of service analysts, development managers, and chief service developers of the corresponding service providers associated in the project. In the team, 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 designing are aligned. The importance of this role is primarily because having separate iterations, where the service analyst team and the service design team are independent of their activities in this phase, problems may rise in the alignment and grouping of the work produced. The sub-phases are executed in form of separate iterations. As mentioned earlier in section 1.3, there have been several researches on only the management of the construction phase where iterations are incorporated but to the best of our knowledge iterative development have not been used in earlier phases in ISD. It is acknowledged that as time progresses, cost of change increases (Ambler, 2009). Therefore, if requirements are well specified and designs are accurately modeled, then there is a less probability of bottlenecks or major issues occurring in the system at a later phase than if not done well at all. Thus, we feel the importance of introducing the iterative development from an earlier stage because it is equally important to invest time and effort in proper requirement analysis and designing just as development is. Incorporating the iteration allows the respective teams to work based on priority and produce usable artifacts in short periods of time. Also, when development teams start constructing, the service modeling team does not have to sit around. They can continue their work in parallel to the development teams. This results in maximized utilization of resources by the organization. Therefore, just as construction of the services is considered a development period, requirement analysis and designing are also considered as development periods in iterations. With respect to the motivation, this phase is depicted in Figure. 14, and is further described in the corresponding sub-phases shown in Figure. 17.
Analysis

In this sub-phase, the service analyst team is responsible for reviewing project document, analyzing services and defining requirements.

As shown in Figure. 17, once the team has retrieved the project document and reviewed the business goals and objectives of the projects, they try to identify services and describe them. Furthermore, they also investigate to see what existing services exist and whether they can append or enhance by adding features or create new services (Papazoglou & Van Den Heuvel, 2006). Once they identify and describe the services by requirements engineering, they produce a service backlog (see section 7.3). Inspired from Scrum, the service backlog is similar to a product backlog where the requirements are specified. Having this service backlog, the entire organization is able to know what services exists or will be implemented; also it helps in the organization of managing the reference and traceability of service description. The service backlog is given to the service design team who provides a draft estimation of time and resources required to develop the services. This service backlog along with the estimation is given to the service board, where they analyze potential return on investment. If the investment is profitable and fitting with the business needs, they decide on the services and finalize the service backlog with prioritization. The service analyst team then specifies requirements for the prioritized services. These requirements are reviewed by the customer service and they input possible concerns of the users (enhancement section 7.1B2; shaded in the middle of the Figure. 17), helping to refine the requirements based on users’ needs. This addition is a difference from the initial conceptual model’s Analysis phase. The service analyst team then sends these requirements to the client. If moderations are required, then the requirements are revised. Else, if approved, the service analysts deliver the service backlog to the service design team for creating technical designs. It is important that the client is involved in this phase because if they are not involved and services are implemented, there is a chance that the client may not agree to what is developed. Therefore, with client participation, which is also something that Agile and service development encourages, there is a better chance of understanding by the organization and trust gained.

This process involving the analysis phase is done in the first iteration, lasting for 2 – 4 weeks as depicted in the APISD lifecycle in Figure. 14. The iteration follows as a sprint does (see section 4.3 and Figure. 4) in the development phase of Scrum. The service analyst team meets weekly to coordinate and take status of their work and ultimately finalizes the service backlog. Next, the analysts elaborate on the prioritized services by producing requirements of both functional and non-functional nature. Once produced, they deliver the service backlog to
the service board or meets with the client to finalize requirements. After confirmation, the analysts provide the service backlog with detailed requirements to the service design team for the design phase, also shown in Figure 17. While the service design team is busy with modeling the services in their first iteration, this iteration is counted as the second iteration for the service analyst team. Here, they commence their work on the next set of prioritized services. They meet with the service design team to coordinate and understand the required components of technical requirements to be aligned, whenever it is required. These meetings are organized by the domain manager to ensure the alignment between the two teams and to handle any conflicts that may occur in the service modeling teams. The domain manager is also holds retrospective meetings for the service analyst and service design team, in order to review the previous iterations and identify key improvements to be done in the future iterations. This is encouraged in Scrum, where frequent reviews help to grow and improve existing services. By this way, performance is reviewed and monitored as suggested in overcoming organizational integration issues (see section 3.1). A detailed view of this phase can be looked at in the sequence diagram of Analysis phase in Appendix II Figure 2.

**Design**

The design phase entails the formation of technical designs and producing features from the service backlog. This phase consists of two sub-activities. First, as mentioned above and shown in Figure 17. Service Model. The service design team analyzes the service backlog and creates a draft estimation and second, with the given requirements of the prioritized services, they produce the high level architecture and the necessary high level technical designs required for development for the first development iteration. Services can be designed to comply with all the functional and non-functional requirements. They create high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team (improvement section 7.1 B1); shown shaded in the middle of the Figure. 17). While designing the new services, it is important to keep into account whether there are existing services that provide similar functionality (Papazoglou & Van Den Heuvel, 2006). Just as the other phases, the service design team comprises of either the following scenarios:

- For a single service provider delivering integrated services, the service design team consists of the chief developer/architect and development manager of that single organization working together to perform the above activities.
- For multiple service providers delivering integrated services, the service design team consists of the chief developer/architect and development manager of the various service providers collaborating together to perform the above activities.

As shown in the API(SD) lifecycle in Figure 14, the service design team performs this process iteratively for 2-4 weeks similar as a sprint does (see section 4.3 and Figure. 4) in the development phase of Scrum. They meet weekly for 1 hour or daily for a short 15 minute status update to further clarify or decide on any issues regarding the designing of the service models. If required, they meet with the service analyst team for requirement clarification or modification. At the end of each iteration, they produce a feature backlog (see section 5.3) of the prioritized set of services. The development manager is responsible for managing this backlog. Once the feature backlog is completed, the system administrator reviews the non-functional requirements and provides input. Furthermore, the system administrator also reviews the corresponding technical designs required for deployment and administering the system (enhancement section 7.1 B3); shown shaded in the middle of the Figure 14). This addition is a difference from the initial conceptual model’s design phase.

Once reviewed and refined, the feature backlog is divided by the development manager according to features and assigned to the respective construction teams. For the case of multiple service providers, features are assigned to the corresponding construction teams of the various providers. In this case, that particular team is responsible for the implementation of the features of the service or even multiple services. To the best of our knowledge, it could possibly reduce the conflicts of dependency, maintenance in later phases, to develop feature-wise rather than component-wise (database, user interface etc.) and therefore, division of work
by feature is chosen for this process. This advocates the need of coordination in activities by which it helps the service provider to effectively organize the work forces (Hatchuel, 2001). Once the construction team is busy developing and testing the set of assigned features in their first iteration, this iteration is counted as the second iteration for the service design team where they commence work on defining the next set of Feature Backlog for the next set of prioritized services in the Service Backlog. A detailed transaction flow of this phase is shown in the sequence diagram in Appendix II Figure 3.

7.2.4 Service Construction Phase

The service construction phase consists of the actual development and ongoing testing that agile methods suggest. First, the construction team(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 shown in the APISD lifecycle in Figure 14. According to the assigned features that are taken to develop in the iteration, an ‘in-progress’ update is made in the feature backlog by the development manager. For each construction team, together with the development manager, service developers (including the chief service developer) and service testers, they go through the services’ and features’ requirements, technical designs. The team communicates with the service modeling team for clarification if needed. This meeting is arranged by the development manager.

![Service Construction Phase of final model](image)

**Figure.** 18. Service Construction Phase of final model

Depicted in Figure 18, from steps 1 to 4, once development begins testers create their test plans and test cases. Development of the services can be done following standards and protocols using technologies such as web services, BPEL or others (see section 3.1). When development of each team ends, the application is deployed locally by the system administrator and is tested by the respective test team. If issues are found, they are logged in a bug tracker system that is available for team(s) involved and distinguished by features. Issues are fixed and regression tests are performed. By regression tests, we mean testing existing features as well to test whether they are affected by the new implementation or not. Once tested and approved by the testers, all the applications are deployed by the system administrator and all the features are integrated in one system in step 8. Next, system test is performed that includes testing of the system as a whole. This test ensures that all the interfaces are running smoothly between the services and the system in general. Should there be any issues, they are logged in the bug tracking system and solved by the teams who have developed that corresponding feature. 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 (see section 3.1). In this demonstration, the service analysts and development/testing team representatives can attend. Once approved by the client, the iteration ends. If teams cannot attend the demo, the meeting can be recorded and later played for the team to see (enhancement section 7.1 C1); shown shaded in the lower part of the figure 14). This addition is a difference from the initial conceptual model’s service construction phase. This will increase the satisfaction and give an impression how the client behaves towards the system. Once the solutions are approved by the client, the iteration ends. If not approved, the set of improvements goes through the steps 1 to 4 and 8 again.

During the iteration, in step 5, the release manager meets with the service board to discuss releases. In step 6, the release manager then meets with the development managers and sets release dates. After a release date has been set and development and testing is completed, a set of features for the corresponding completed services are deployed in the system. The system goes through a final test after solving issues and packaged to a frozen state, which is ready for the provisioning phase. The team can proceed with the next set of features from the feature backlog accordingly. A detailed flow of release planning can be seen in Appendix II Figure 5.

The development manager handles any conflicts between the development and test teams. In the case of multiple service providers, each construction team has one development manager. In order to coordinate effectively among the teams, the development managers meet weekly. In this meeting, they discuss any impediments, dependency related issues and further planning of iterations. They also discuss on changes that may occur. With a shared understanding, they can project a clear vision to their respective teams to adapt those changes efficiently; thus embracing changes and coordinating the implementations accordingly (see section 3.1). Towards the end, the development managers also arrange a retrospective meeting of their own to discuss results, lessons learned and improvement points. Then each of them holds a retrospective meeting with their own team to discuss the same and take 2 or 3 improvement points to implement in the next iteration. This whole process continues in iterations with the corresponding set of prioritized features and is implemented accordingly. When a release has been set, the development manager gives a high-level demo to the system administrator and customer service so that they are acknowledged of the integrated services to be deployed in that release (enhancement section 7.1 C2). This addition is a difference from the initial conceptual model’s service construction phase. A detailed transaction flow of this phase is shown in the sequence diagram in Appendix II Figure 4.

7.2.5 Provisioning Phase

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. 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. In the case of multiple service providers, this phase also involves in settling service management and governing of the services. In this phase, the service board decides on the service governance, and billing according to usage.

Papazoglou and Van Den Heuvel (2006) have mentioned central and distributed governance. Central governance has one governing body from each service domain that does not have direct responsibility for any of the service domain but reviews the type of services before implementation. In distributed governance “each business unit has autonomous control over how it provides the services within its own enterprise” and is suited for distributed teams. When considering the governance type for the agile ISD, once again, it varies on the
scenario of the organizations involved. On one hand, for a single service provider, a central governance suits best as the organization is making its own decisions about the integrated services they are to provide. On the other hand, for multiple service providers, distributed governance is the most suitable approach. As there are multiple organizations involved and services are executed and maintained by the service providers themselves, they have the autonomous control as Papazoglou and Van Den Heuvel (2006) mentioned. The service board is then responsible for defining the rules and regulations regarding process, domain responsibilities and service development responsibilities among the various teams for effective steering and control of the integrated service delivery process. This is done at the beginning of the APISD lifecycle, in the form of internal SLAs. However, even though distributed governance exists for the execution and management of services, the service board in this scenario does decide together and governs on the decisions. Considering this argument, during the planning and decision making throughout the APISD lifecycle, the service board acts according to a central governance mechanism.

Setting apart the scenarios, the service board in both cases is also responsible for creating SLAs for clients that assures the deployed services. Through this contract, the service provider(s) are able to charge their clients for what they offer or to endure security (Gu & Lago, 2007). Here the service board needs to decide on how they plan to charge the service usage and draw the SLA for the client. Available options may be: payment on a per use basis, payment by subscription, payment by leasing, lifetime services, and free services (Papazoglou & Van Den Heuvel, 2006). In addition, the following options are also available: per service request or delivery (e.g. a fixed price local telephone call); (2) by unit of measure and granularity (e.g. by length, volume, weight, area or time); and (3) on a percentage or ratio basis of some aspect of the service (e.g. by commission) (O’Sullivan et al., 2002). Along with the payment option, the service board also needs to define in SLAs how well the services are delivered in terms of cost, availability, performance (Gu & Lago, 2007). Once the service board drafts the SLAs and sends to the client, the client reviews and either agrees or disagrees. Further discussion between these two parties occurs and further alterations are made in case of disagreements. Only when all the SLAs are confirmed and signed, the accessibility of the services are performed where the authorization of using the services are permissible. A detailed flow of the process in this phase can be seen in Appendix II Figure 6.

![Figure 19. Provisioning Phase of final model](image)

7.2.6 Development and Execution Phase

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. This means that the web services of the system are fully deployed and operational. Service requesters are now able to find service definition and invoke all the service operations (Papazoglou & Van Den Heuvel, 2006). The customer service informs the client regarding the deployment of the services. Once the integrated services are deployed in the production, a quick test is run to check that the services are running properly. If there are any problems, they are fixed within a short time and re-deployed. Once redeployed and client is able to access the integrated services, if somehow there arises a
situation where the client rejects the release, a rollback is done returning the system back to its old state. After this, the service board reports back to the executive board. Then they analyze the problems and produces possible solutions (enhancement section 7.1 D1); shown shaded in figure 16). This addition is a difference from the initial conceptual model's development and execution phase. A detailed flow of this phase is given in Appendix II Figure 7.

Figure. 20. Deployment and Execution Phase of final model

7.2.7 Service Management Phase

Once in production and used by the users, the integrated services are monitored and ensured that all the services are running according to the rules and regulations set in the SLAs. Monitoring the services will allow the service providers(s) to analyze the service delivered and gather improvements. Regarding the management of the technicalities, the system administrator is responsible for configuring, managing and troubleshooting the servers. The service provider can manage the services according to standards and best practices (improvement section 7.1 E2). This addition is a difference from the initial conceptual model’s service management phase. This phase also consists of change management. As previously mentioned earlier in service lifecycle (see section 3.3), it is very important that changes are managed well in order to ensure a smoothly run business. Following the usage of services, clients can request changes, which are logged in a change request backlog (see section 7.3), also shown in Figure. 21. While using the services, various incidents may occur which are logged in an incident backlog (see section 7.3). Depending on the urgency, the incidents are immediately passed over to the respective construction team to solve it. Here, instead of the whole team engaging, one team member of that respective team solves it (enhancement section 7.1 C3). This addition is also a difference from the initial conceptual model’s service management phase. This way the other team members are focused on the ongoing iteration development. If not urgent, they are logged in based on severity and priority which are taken in iterations. Similarly change requests are logged in. Both the backlogs are reviewed by the Release Manager and based on the primary prioritization, the chosen sets are then passed to the planning phase where the service board reviews it and the whole cycle progresses accordingly as shown in the APISD lifecycle depicted in Figure. 14.

With the ongoing usage of the services, the service analysts visit the work-floor of the users to observe their interaction with the services (enhancement section 7.1 E1); also shown shaded in Figure. 21). This activity will allow a better understanding of the users' needs. They later translate the needs into change requests and inform the customer service to log them in the change request backlog. A detailed view of this phase can be seen in the sequence diagram in Appendix II Figure 8.
7.3 Artifacts for APISD

In the APISD model explained in the previous sections, several artifacts have been mentioned that are produced and used by the various roles in different phases. Each of these artifacts will be described in this section.

— *Vision Document*

The vision document consists of the vision, mission, objectives and business needs of the organization. This document is produced by the executive board. This artifact is a new addition to the change made in section 7.1A(1). This addition is a difference from the initial conceptual model’s artifacts.

— *Project Document*

The project document consists of the identified services and their purposes, scope and feasibility of the project, financial analysis of costs and benefits including a budget. This document is produced by the service board.

— *Service Backlog*

The service backlog is a collection of finalized services. Each service in the service backlog consists of service number, description, priority, detailed functional and non-functional requirements, and estimated duration of completion. The service backlog also contains a high level architectural design that illustrates the relations between components, infrastructure and logical explanations. This backlog is produced and managed by the service analyst team.

— *Feature Backlog*

The feature backlog is a collection of features referred to corresponding services. Each feature in the feature backlog consists of a feature number, service reference, priority, technical design, estimated duration of completion, and assigned to team description. This backlog is produced by the service design team and managed by the development manager.

— *Change Request Backlog*

The change request backlog is a collection of change requests made. Each change request in the backlog consists of a change request number, description, service
reference, feature reference, and priority. This backlog is produced and maintained by the customer service.

- **Incident Backlog**
  The incident backlog is a collection of incidents reported. Each incident in the backlog consists of an incident number, description, service reference, feature reference, priority, and severity. This backlog is produced and maintained by the customer service.

7.4 **Summary**

In this chapter, the conceptual model for APISD has been finalized. To finalize it, first the additional factors were differentiated and identified from the case study findings in chapter 6 and were highlighted as enhancements to the model. Each enhancement was placed under an APISD phase and also consists of an insight from the cases that it was analyzed from. The difference from the initial conceptual model of APISD is also distinguished for that enhancement. To recap, the enhancements are:

- The service board has been divided in two separate boards, namely, *Executive board* (serving the strategic activities) and *Service board* (serving the tactical activities);
- In the Service Modeling phase, the service design team creates high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team;
- The customer service is involved in the Service Modeling phase to review the service backlog produced by the service analysts;
- The system administrator is included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and provide input;
- An optional activity exists where the demo given to the client after each iteration in APISD can be recorded and shown to the construction team if they are not involved in it;
- After the construction phase, a high level product demonstration is given to the customer service and system administrator;
- For feedback implementation, one team member is responsible for implementing the feedback while the rest of the team can focus on the iteration;
- In case of rejection by the client after post-production, an activity is included in APISD for analyzing the problems and producing possible solutions;
- There is a workflow created for the service analysts to also visit the users' work-floor and observe their interaction and engagement with the integrated services;
- The practice of service provider(s) engaging in adopting best practices and standards for service management is included in the final model.

The initial conceptual model described in chapter 5 was revisited with the above supplements appended in the corresponding phases. Within the model description, the enhancements are both highlighted in the description and in the figures. With the APISD conceptual model finalized, the next chapter will entail an illustrative example of how APISD can be implemented within an organization.
Chapter 8

Illustrating APISD
8 Illustrating APISD

In the previous chapter, the conceptual model of APISD has been finalized. In order to understand how APISD can be implemented within an organization, an illustrative scenario of an implementation is described in this chapter, and thus completing the second part of the DSRM step 'Communication'. First, APISD will be presented in details for a single service provider called 'Alpha'. Then, the model will be presented for multiple service providers collaborating together called 'Alpha', 'Beta' and 'Gamma'.

8.1 Single Service Provider's Case

Alpha is a single service provider which produces integrated services. They are planning to grow as an organization and emerge in new markets.

8.1.1 Planning Phase

In the planning phase, the executive board of Alpha explores future possibilities for growth. After understanding the current market conditions, they define the organization's mission, vision and objectives. In order to communicate with the rest of the organization, they produce a vision document that contains all information necessary to understand what Alpha is aiming for and how to achieve the objectives defined.

Alpha has a number of members who looks over the services developed in the organization. This group is Alpha's service board. The service board has received notification of the new changes by the executive board. Once delivered, the vision document is reviewed by the service board, which depicts step 4 in Appendix II Figure 1: Planning Phase. The service board grasps what is written on the vision document and searches for potential clients. As Alpha is reaching different markets, she needs to meet with the client to understand the services required. Therefore, according to step 5 in the figure, the service board recently has met a promising client, 'Sigma'. Sigma is a client which is in need of supporting their internal processes to better manage and automate. The processes are related to management of their products, content and customer relationships. Analyzing the potential of Alpha and her recognition in the industry, Sigma decides to meet with Alpha. Thus, the service board discusses the type of services related to those processes Sigma needs and financial aspects. Afterwards, Sigma decides to have a collection of services in integrated content management modules which are related to news management, events management, product management, contact management, document management and membership management.

The service board returns back to Alpha producing a Project document where they define the identified services and their purposes and scope. They also define the feasibility of development of those services and provide a financial analysis of costs and benefits including a budget. This financial analysis is done by calculating the profitability of implementation of these services by calculating ratios such as Return on equity, profit margin and Return of Investment. In step 8 shown in the figure, they deliver this document to the executive board. The executive board reviews this project document and checks whether it meets the organizational needs as well. As Sigma is an important client, the executive board decides to go ahead with this project. Once the project is confirmed, the service board delivers the project document to the service analyst team in order to specify requirements for the services. In the meantime, as shown in step 13, the service board drafts a Service Level Agreement (SLA) for Sigma, that consists of the commitment of Alpha's services based on how well the services are delivered. This is done with respect to the availability and performance of the services delivered. Alpha guarantees a 24 hour service with exception to the major government holidays and guarantees a 99.99% availability and running criteria for the services. The service board also analyzes the type of charging mechanism to incorporate for the services. This SLA is further detailed in the Provisioning phase.
8.1.2 Service Modeling Phase

As shown in Appendix II Figure 2: Service Modeling Phase - Analysis step 1, Alpha has a team of service analysts who first review the project document provided by the service board. After understanding the nature and type of services required by Sigma, the service analysts begin producing the service backlog. In it, they identify 5 services and define them in high level requirements, which are required and will be provided to Sigma. These services are: News Portal (NP), Events Portal (EP), Product Administration (PA), Customer Relationship Management Service (CRMS), Document Management Service (DMS). They investigate Alpha’s existing services and realize that two of Alpha’s existing services can support three of Sigma’s new services. Alpha’s Client Service can support the creation of the CRMS and the Content Service can support the NP and EP. Therefore, in the service backlog for those three services, they attach a reference and brief description of the Client Service and Content Service and how they can be incorporated in the integrated service delivery for Sigma. As shown in step 3 in the sequence diagram of Appendix II Figure 2: Service Modeling Phase - Analysis, in order to investigate the feasibility of the development, they deliver the service backlog to the service design team. Alpha has a chief architect who looks over Alpha’s service structure and architecture. Together with the development manager, they determine the resources required in terms of people and infrastructure. They also determine a high level estimation of developing those 10 services. Then, they provide this estimation and feasibility concerning resources to the service analysts. The service analysts update the service backlog with the information and deliver it to the service board. As shown in step 6 and 7, the service board analyzes the service backlog and decides on the investment. Then, with a quick call with Sigma, the service board prioritizes the services. They finalize the prioritized service backlog and inform the service analysts (see section 5.1.2).

Given the prioritization, the domain manager, who is responsible for managing the alignment between the Analysis and Design sub-phase in Service Modeling phase, groups the services according to iterations. The first analysis iteration begins in the service modeling phase, namely I-A1 which is short for Iteration-Analysis 1, and is scheduled to continue for 3 weeks. I-A1 consists of Priority 1 service which is CRMS. Here by Priority, it is divided to three divisions based on the urgency of developing them. Priority-1 indicates: service must be completed; Priority-2 indicates: service should be completed; Priority-3 indicates: service might be completed before shipping the service package if time allows. The service analysts begin specifying detailed requirements for this prioritizes sets. Every Monday, they sit together to go through the status of the work and discusses what is needed to complete the rest of the work. During the week, they occasionally meet to discuss the dependencies within the services so that they can incorporate the interfaces within the requirements. Once completed, they send this service backlog to the customer service department, as shown in step 9, to review and to provide feedback. Customer service manager, who is responsible for the customer service department, reviews the requirements for those three services and from previous experiences he/she provides feedback on usability aspects. If applicable, the service analysts incorporate the feedback in the requirements and refine them. They send these requirements to Sigma for review. Sigma reviews and provides feedback if necessary. After incorporating Sigma’s feedback, the service analysts finalizes the service backlog. Then, the domain manager informs the design team to begin their design iteration, namely I-D1. As I-D1 commences, I-A2 begins for the service analyst team and they proceeded with Priority 2 service, which is DMS. The following iterations behave similarly until all services in the service backlog are completely specified.

In I-D1, the chief architect and development manager first review the service backlog on Priority 1 services, as shown in step 1 in Appendix II Figure 3: Service Modeling Phase - Design. If they have any questions regarding the requirements, they request a meeting to the domain manager. The domain manager arranges a meeting for them and the service analysts. After clarification, the chief architect and development manager begin producing the technical architecture that encompasses the services to be delivered as a whole in step 3. I-D1 is scheduled to last for 3 weeks. They meet daily for a 15 minute status update on the work done, future work and identify dependencies. They create designs for interfaces in
components, deployment and infrastructure. For the development, they divide the CRMS into small features such as Contact Management, Membership Management, User Activity Management. For these features, they also create high-level code designs and define the non-functional requirements for those services related to security of user access, usability of the services, availability of services, and performance of the services. As the new services are supported by Alpha's existing two services, they create interfaces for connection as well as update the existing architecture to incorporate the interfaces to the new services. Once the designs are completed, the service design team delivers them to the system administrator to review the designs for being familiarized with the new systems and provide feedback on the designing for deployment issues and infrastructure such as whether to load empty databases or with client data and what considerations to take in order to load after deployment of the services. With this feedback, the design team finalizes the service backlog. A problem arises that these non-functional requirements are not incorporated in the service backlog since the service analysts are responsible for maintaining it. Therefore, it is important that the domain manager maintains alignment between the iterations. The domain manager informs the service analysts regarding the non-functional requirements and later together with the design team, they update the service backlog. Towards the end of I-D1, the development manager creates the feature backlog according to the features divided earlier and connects each feature with the service by referencing the corresponding Service ID in the Feature. He/she inputs the corresponding requirements for those features accordingly and assigns the service developers to build them. With this, I-D1 completes and I-D2 begins after the completion of I-A2 with performing similar tasks as they did in I-D1. The following iterations behave similarly until all services in the service backlog are completely designed.

8.1.3 Service Construction Phase

Alpha has a development team consisting of 15 service developers and 3 service testers. These members are divided into three construction teams: CT-A, CT-B and CT-C, with 5 developers and 1 tester each. As shown in Appendix II Figure 4: Service Construction Phase - Development, the first construction iteration, I-C1, begins with the first Priority 1 services. As shown in step 1, all teams go through the feature backlog and understand the requirements and designs. If they have any questions, they are first collected and then clarified by the service analysts in a meeting arranged by the development manager. Following that in step 3, the developers in each team begin developing the features they are assigned to by the development manager. In this iteration and others, the members of a team meet daily for 15 minutes, just like a Daily Standup meeting in Scrum, to discuss the status of what they have completed, what they will complete and address any impediments that affect that day's work. This occurs with every team individually. Each of the meetings is held in separate times, so that the development manager can attend all three meetings and keep note of the progress of the iteration. While they are developing, testers begin creating test scripts to test the system later on. Using best practices, testers and developers collaborate together to ensure that all requirements are implemented and all scenarios regarding non-functional requirements are also incorporated. During this iteration, a number of issues may arise. Such as: (1) dependencies with other services where other teams are working in (2) conflicts with testers in order to incorporate the test scenarios in the implementation (3) timing issues of completion of developing components that other developers build upon (4) need of extra resources that development cannot continue without. These issues are reported to the development manager and thus solved accordingly. Once a team has completed developing a feature, they request the system administrator to deploy the features in a test environment, as shown as in step 4. The system administrator deploys and notifies the tester. The tester then begins testing the implemented features according to the test scripts, comes across faults in the system and logs them in their bug-tracking system, Jira. These faults are then reported to the developers in step 7 and they start fixing them. As soon as the faults are all fixed, they request the system administrator to re-deploy the services in the test environment. The tester then tests the faults fixed and performs regression tests to check for inconsistencies caused by the new fixes, as shown in step 11; for example, re-testing the features initially tested in the first time and checking for any abnormal occurrences. The steps from 6 to 11 can occur until the testers are
satisfied according to the accepted fault benchmark set. The steps from 3 to 11 occur for each team as well. This ends I-C1. Similarly, I-C2 starts with the next prioritized set of services. After each iteration, the development manager arranges a short demonstration meeting with the client. Together with a member of each team and service analyst (optional), the development manager presents the developed services to the client. The client provides feedback for improvements if needed. In this case, in the following iteration, one service developer from the team that has developed those features where the improvements are required, that member implements the feedback while other developers develop the features assigned for I-C2. For every iteration, a different developer takes turn to implement the feedback. Also, at the end of each iteration, the development manager holds a retrospective meeting with all three teams to discuss and analyze the iteration passed and identifies improvement points for the team that they can all realize in the following iterations. This learning process helps to grow stronger and competent teams.

Meanwhile, Alpha’s release manager discusses possible release dates with the service board, as shown in Appendix II Figure 5: Service Construction Phase - Release Planning. When the possible dates are decided on, the release manager discusses with the development manager on the feasibility of delivering the integrated services on those dates. Once the date has been decided upon and after all the iterations complete with the services defined in the service backlog, the development manager requests the system administrator for deploying all integrated services. As shown in step 13 in Appendix II Figure 4: Service Construction Phase - Development, the system administrator deploys the service package in the test environment where the service testers can begin performing a system test. In a system test, testing related to integration, performance, load, compatibility, stress, security, installation, and usability are considered. If there are any issues followed by the tests, they are fixed by the developers and tested by the testers. A re-deployment is done and regression tests are performed. Once the system is ready, the development manager arranges a meeting to demonstrate the integrated service package to Sigma. The client tests and provides feedback if required. This feedback is implemented and tested. Based on their review of the services, Sigma accepts the package and gives approval to proceed with deployment in the production environment. As the preparation begins, meanwhile the development manager gives a high-level demonstration of the service package to the customer service and the system administrator to familiarize so that they can support Sigma much better.

8.1.4 Provisioning Phase

In this phase, the service board finalizes the SLA drafted during the Planning phase. They finalize on availability and performance of the CRM, DMS, NP, EP services they will provide to Sigma. As mentioned earlier, Alpha guarantees a 24 hour service with exception to the major government holidays and guarantees a 99.99% availability and running criteria for the services. The service board decides to charge Sigma by subscription use. With other inclusions, they draft up a SLA and delivers to the executive board to review it. As shown in Appendix II Figure 6: Provisioning Phase, the executive board reviews the SLA and gives feedback in step 4. The service board finalizes the SLA and delivers to Sigma. Sigma reviews the SLA and signs off the contract.

8.1.5 Deployment & Execution Phase

As soon as the SLA is signed off, Alpha begins to deploy the service package in production environment. The system administrator deploys and notifies the development manager. The development manager notifies the customer service which notifies Sigma. As shown in Appendix II Figure 7: Deployment & Execution Phase in step 9, Sigma tests the service and approves. The service board sends a status report to the executive board.
8.1.6 Service Management Phase

As Sigma’s users use the service package, the system administrator monitors the various aspects of the services. In case of incidents, Sigma reports to the customer service. Example of incidents that may occur, the user is unable to retrieve content of a customer that was saved earlier, creation of 100 customer data crashes the module or the DMS becomes unavailable when 150MB of data is stored. As shown in Appendix II Figure 8: Service Management Phase step 2, the customer service logs the incident in the incident backlog. Based on the urgency and severity of the incident, the customer service either requests the service construction team to solve or keeps it logged to fix it later. In case of urgent fixes, a service developer fixes the incident and the tester tests it. A deployment is made for the fix. If Sigma requires a change request on a feature of a particular service, they contact the customer service regarding it. The customer service logs this request in a change request backlog. Every two weeks, the release manager goes through the incident and change request backlog. Based on priority, the release manager selects a bundle of the two and provides them to the service board for further planning. Based on the needs, the service board analyzes them and the whole APISD lifecycle commences again.

Alpha is very keen in supporting Sigma as much as possible and therefore, to serve their needs, Alpha sends the service analysts to Sigma’s office and promotes observing how Sigma’s users interact with the delivered integrated services. Based on the observation and also discussing with the users, the service analysts are able to translate them to change requests. These changes are reported to Alpha’s customer service where they are logged in the service backlog and further prioritized by the release manager.

8.2 Multiple Service Providers’ Case

Alpha, Beta and Gamma in general are single service providers that delivers integrated services by themselves. Recently, they have come together to deliver integrated services combining competences of all three to a specific targeted client, ‘Theta’. For identification purposes, the following text will recognize this consortium as ABG.

8.2.1 Planning Phase

Since the three companies have decided to collaborate together and deliver integrated services to Theta, they have formed an executive board. This board consists of the executives from each company and together they analyze the current market conditions and define the consortium’s mission, vision, strategy and objectives. In order to communicate with the rest of the organization, they produce a Vision document that contains all information necessary to understand what ABG is aiming for.

ABG has formed a group of members from each company who will look over the integrated services developed for Theta. This group is ABG’s service board. The service board has received notification of the new changes by the executive board. Once delivered, the vision document is reviewed by the service board, which depicts step 4 in Appendix II Figure 1: Planning Phase. The service board grasps what is written on the vision document and searches for potential clients. As ABG is reaching different markets, they need to meet with the Theta to understand the services required. Therefore, according to step 5, the service board meets with Theta. They discuss the type of services Theta needs and financial aspects. Afterward, Theta decides to have a collection of services.

The service board returns back to ABG producing a Project document where they define the identified services and their purposes and scope. They also define the feasibility of development of those services and provide a financial analysis of costs and benefits including a budget. This financial analysis is done by calculating the profitability of implementation of these services by calculating ratios such as Return on equity, profit margin and Return of Investment. In step 8, they deliver this document to the executive board. The executing board
reviews this project document and checks whether it meets the consortium's needs as well. As Theta is an important client, the executive board decides to go ahead with this project. Once the project is confirmed, the service board delivers the project document to the service analyst team in order to specify requirements for the services. In the meantime, as shown in step 13, the service board drafts a Service Level Agreement (SLA) for Theta, that consists of the commitment of ABG's services based on how well the services are delivered. This SLA is further detailed and finalized in the Provisioning phase. In addition to that, the service board drafts an internal SLA that defines the contractual agreements between Alpha, Beta and Gamma. These agreements are based on the responsibilities, financial investments and profit sharing aspects. They send these agreements to the executive board and the executive board reviews and signs off.

8.2.2 Service Modeling Phase

As shown in Appendix II Figure 2: Service Modeling Phase - Analysis step 1, ABG has a team of service analysts from each company who first reviews the project document provided by the service board. After understanding the nature and type of services required by Theta, the service analysts begin producing the service backlog. In the service backlog, they identify 10 services and define them in high level requirements, which are required and will be provided to Theta. As shown in step 3, in order to investigate the feasibility of the development, they deliver the service backlog to the service design team. This design team comprises of chief architects/developers and development managers from each company and will look over ABG's service structure and architecture. The service design team determines the resources required in terms of people and infrastructure. They also determine a high level estimation of developing those 10 services. Then, they provide this estimation and feasibility concerning resources to the service analyst team. The service analyst team updates the service backlog with the necessary information and delivers it to the service board. As shown in step 6 and 7, the service board analyzes the service backlog and decides on the investment. Then, with a quick call with Theta, the service board prioritizes the services. They finalize the prioritized service backlog and inform the service analyst team.

ABG appoints a domain manager who is responsible for managing the service modeling phase. Given the prioritization, the domain manager groups the services according to iterations. The first analysis iteration begins in the service modeling phase, namely I-A1, and is scheduled to continue for 3 weeks. I-A1 consists of three services with Priority 1. Here by Priority, it is divided to three divisions based on the urgency of developing them. Priority-1 indicates: service must be completed; Priority-2 indicates: service should be completed; Priority-3 indicates: service might be completed before shipping the service package if time allows. The service analyst team begins specifying detailed requirements for the prioritized sets. Every Monday, they sit together to go through the status of their work and discuss what is needed to complete the rest of the work. During the week, they occasionally meet to discuss the dependencies within the services so that they can incorporate the interfaces within the requirements. Dependency issues can be related to a set service components that are used in multiple services. If coordination of these components is not specified correctly, issues of redundancy and lack of proper resource utilization may occur. Once completed, they send this service backlog to the customer service department of the three companies, as shown in step 9, to review and to provide feedback. Customer service managers from each company review the requirements for those three services and from previous experiences they provide feedback on usability aspects. If applicable, the service analyst team incorporates the feedback in the requirements and refines them. They send these requirements to Theta for review. Theta review and provides feedback if necessary. After incorporating Theta's feedback, the service analyst team finalizes the service backlog. Then, the domain manager informs the service design team to begin their design iteration, namely I-D1. As I-D1 commences, I-A2 begins for the service analyst team and they proceed with Priority 2 services. The following iterations behave similarly until all services in the service backlog are completely specified.
In I-D1, the chief architects and development managers from each company first review the service backlog on Priority 1 services, as shown in step 1 in Appendix II Figure 3: Service Modeling Phase - Design. If they have any questions regarding the requirements, they request a meeting to the domain manager. The domain manager arranges a meeting for them and the service analysts. After clarification, the design team begins producing the technical architecture that encompasses the services to be delivered as a whole in step 3. I-D1 is scheduled to last for 3 weeks. They meet daily for a 15 minute status update on the work done, future work and to identify dependencies. Due to geographical boundaries, ABG promotes use of virtual communication. This daily meeting is therefore held virtually with the use of audio and video via the internet. Furthermore, with desktop sharing tools such as TeamViewer or WebEx, they collaborate remotely and design together if needed using collaborative diagramming tool, LucidChart. They create designs for interfaces in components, deployment and infrastructure. For the development, they divide the three services into small features. For these features, they also create high-level code designs and define the non-functional requirements for those services related to security of user access, usability of the services, availability of services, and performance of the services. As the new services are supported by Beta's and Gama’s existing services, they create interfaces for connection as well as update the existing architecture to incorporate the interfaces to the new services with technologies such as Simple Object Access Protocol or Remote Procedure Call. Once, the designs are completed, the service design team delivers them to the system administrators of each company in order to review the designs to be aware of the new systems to be implemented and provide feedback on the designs. With this feedback, the design team finalizes the service backlog. The domain manager informs the service analyst team regarding the non-functional requirements and later together with the design team, they update the service backlog. Towards the end of I-D1, the development managers create the feature backlog according to the features divided earlier and connect each feature with the service by referencing the corresponding Service ID in the Feature. They input the corresponding requirements for those features accordingly and assign the service construction teams to build them. With this, I-D1 completes and I-D2 begins after the completion of I-A2 with performing similar tasks as they did in I-D1. The following iterations behave similarly until all services in the service backlog are completely designed.

8.2.3 Service Construction Phase

All three companies have their own construction teams. These teams are: CT-A, CT-B and CT-G. As shown in Appendix II Figure 4: Service Construction Phase - Development, the first construction iteration, I-C1, begins with the first Priority 1 services. As shown in step 1, all teams go through the feature backlog and understand the requirements and designs. If they have any questions, they are first collected and then clarified by the service analyst team in a meeting arranged by the development managers. This meeting consists of 1 member per construction team, 1 service analyst and the development managers. Following that in step 3, the developers in each team begin developing the features they are assigned to by the development managers. The developers builds the services in such a way that they create web services to be able to connect to other services once all the services are integrated in one system. In this iteration and others, the members of a team meet daily for 15 minutes, just like a Daily Standup meeting in Scrum, to discuss the status of what they have completed, what they will complete and address any impediments that affect that day’s work. This occurs with every team individually. The development manager attends this meeting and keeps note of the progress of the iteration. While they are developing, the testers begin creating test scripts to test the system later on. Using best practices, testers and developers collaborate together to ensure that all requirements are implemented and all scenarios regarding non-functional requirements are also incorporated. During this iteration, a number of issues may arise. Such as: (1) dependencies with other services where other teams are working in (2) conflicts with testers in order to incorporate the test scenarios in the implementation (3) timing issues of completion of developing components that other developers build upon (4) need of extra resources that development cannot continue without.
These issues are reported to the development manager responsible for that team. The development manager solves the impediments occurring for his/her team. In case of dependencies with other teams, the development manager brings these issues to the weekly meeting with other development managers. In this weekly meeting, they discuss on the status of the iteration and analyze solutions for any impediments. Once a team has completed developing the feature, they request their own system administrator to deploy the features in test environment, as shown as in step 4. The system administrator deploys and notifies the tester. The tester then begins testing the implemented features according to the test scripts, comes across faults in the system and logs them in their bug-tracking system, Bugzilla. These faults are then reported to the developers in step 7 and they start fixing them. As soon as all the faults are fixed, they request the system administrator to re-deploy the services in the test environment. The tester then tests the faults fixed and performs regression tests to check for inconsistencies caused by the new fixes, as shown in step 11; for example, re-testing the features initially tested in the first time and checking for any abnormal occurrences. The steps from 6 to 11 can occur until the testers are satisfied according to the accepted fault benchmark set. The steps from 3 to 11 occur for each team as well. This ends I-C1. Similarly, I-C2 starts with the next prioritized set of services. Towards the end of the iteration, each company’s system administrator deploys their developed features in a test-integrated environment. ABG’s service testers begin performing an integration test. If there are any integration issues, they are fixed by the corresponding developers assigned and tested by the testers. A re-deployment is done and regression tests are performed. Once the system is ready, the development manager arranges a short meeting to demonstrate the service package to Theta. The client tests and provides feedback if required. This feedback is implemented and tested. In this case, in the following iteration, one service developer from the team that has previously developed the feature that is related to the feedback, implements while other developers develops the features assigned for I-C2. For every iteration, a different developer takes turn to implement the feedback. Also, at the end of each iteration, the development managers meet with each other and discuss on the iteration passed. Then each development manager holds a retrospective meeting with their own team to discuss and analyze the iteration passed and identifies improvement points for the team that they can all realize in the following iterations. This learning process helps to grow stronger and competent teams.

Meanwhile, ABG appoints a release manager who discusses possible release dates with the service board, as shown in Appendix II Figure 5: Service Construction Phase - Release Planning. When the possible dates are decided on, the release manager discusses with the development managers on the feasibility of delivering the integrated services on those dates. Once the date has been decided upon and after all the iterations complete with the services defined in the service backlog, a final iteration is done to integrate all the services developed in one system. Just as mentioned before, the integration involves an integration test and if issues found, they are fixed and tested. Once the system is ready, the development manager arranges a final meeting to demonstrate the integrated service package to Theta. The client tests and provides feedback if required. This feedback is implemented, tested and redeployed. Based on their review of the services, Theta accepts the package and gives approval to proceed with deployment in the production environment. As the preparation begins, meanwhile the development manager gives a high-level demonstration of the service package to the customer service and the system administrators appointed to manage ABG’s service package in order to familiarize so that they can support Sigma much better.

8.2.4 Provisioning Phase

In this phase, the service board finalizes the SLA drafted during the Planning phase. They finalize on availability and performance of the services they will provide to Theta. ABG guarantees a 24 hour service with exception to the major government holidays and guarantees a 99.99% availability and running criteria for the services. The service board decides to charge Theta by subscription use. With other inclusions, they draft up a SLA and delivers to the executive board to review it. As shown in Appendix II Figure 6: Provisioning Phase, the
executive board reviews the SLA and gives feedback in step 4. The service board finalizes the SLA and delivers to Theta. Theta reviews the SLA and signs off the contract.

8.2.5 Deployment & Execution Phase

As soon as the SLA is signed off, ABG begins to deploy the service package in production environment. The appointment system administrator deploys and notifies the development managers. One of the development managers notifies the customer service which also notifies Theta. As shown in Appendix II Figure 7: Deployment & Execution Phase in step 9, Theta tests the service and approves. The service board sends a status report to the executive board.

8.2.6 Service Management Phase

As Theta’s users use the service package, the system administrators of ABG monitor the various aspects of the services. In case of incidents, Theta reports to the customer service. As shown in Appendix II Figure 8: Service Management Phase step 2, the customer service logs the incident in the incident backlog. Based on the urgency and severity of the incident, the customer service either requests the service construction team to solve or keeps it logged to fix it later. In case of urgent fixes, a service developer responsible for the development of the respective feature fixes the incident and the tester of that team tests it. A re-deployment is made for the fix by the system administrator. If Theta requires a change request on a feature of a particular service, they contact the customer service regarding it. The customer service logs this request in a change request backlog. Every two weeks, the release manager goes through the incident and change request backlog. Based on priority, the release manager selects a bundle of the two and provides them to the service board for further planning. Based on the needs, the service board analyzes them and the whole APISD lifecycle commences again.

ABG is very keen in supporting Theta as much as possible and therefore, to serve their needs, ABG sends a member from the service analyst team in rotation to Theta’s office and promotes observing how Theta’s users interact with the delivered integrated services. Based on the observation and also discussing with the users, the service analyst team is able to translate them to change requests. These changes are reported to ABG’s customer service where they are logged in the service backlog and further prioritized by the release manager.

8.3 Summary

In this chapter, an illustration of how APISD can be implemented in an organization is presented. Two scenarios have been undertaken. First, how APISD is implemented within a single service provider delivering integrated services and second, how it is implemented within multiple service providers. With this demonstration, an understanding can be obtained how APISD materializes in practice and thus provides a concrete overview. The next chapter will provide conclusions regarding this thesis.
Chapter 9

Conclusions
&
Recommendations
9 Conclusions and Recommendations

The research objective of this thesis was to develop a conceptual model of Agile Process for Integrated Service Delivery (APISD). The conceptual model was introduced in chapter 5 and enhanced in chapter 7 based on the case study findings of chapter 6. Furthermore, a practical demonstration of the implementation of the model was presented in chapter 8. This chapter focuses on the final step 'Communication' in the DSRM framework (shown in Figure 2), where the conclusions and importance of this research are communicated to the community.

Therefore, in this chapter, the conclusions of this research are described as follows. First, the conclusions of the research are given answering the main research question and sub-research questions. Afterwards, the limitations, recommendations for this model and opportunities of further research are discussed.

9.1 Conclusions of APISD Research

The main idea of the APISD model presented in this thesis was the following: as organizations are now more involved in building agile interoperable systems where the business entities can work together, even if they belong to different organizations, the need for a flexible and integrative approach is needed to effectively collaborate and coordinate. Since, such a structure concerns the management of information space, seamless integration and interoperation between parties, the need of adopting a flexible approach to support fast changing conditions in this environment has become extremely important.

As companies have increasingly become dependent on IT to interact with clients, the issue of business agility has become one of IT agility. Companies look for agility in their business for flexibility and adaptation with respect to changing environments. They seek to make use of the market competencies and deliver customer-centric services to gain the competitive advantage by providing a one-stop mall where customers can experience several kinds of integrated services. With the opportunity of Integrated Service Delivery (ISD), companies can support clients in an integrated environment possibly reducing cost and time. Doing so, service providers face a number of challenges related to organizational integration, resistance towards change and being customer-centric. With a structured process, collaboration and coordination of activities in ISD can be efficient, reducing the complexities in those challenges. Research has shown adoption of Agile methodologies has reduced complexities in software development and focused on collaboration and coordination to achieve performance gain. Therefore, in this case, adopting Agile management principles will be highly beneficial in ISD. To the best of our knowledge there has not been research on how to manage the service lifecycle of ISD in a holistic view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile approach. Thus, in service development, specifically focusing on ISD, amalgamating the service development and Agile management principles in a structured process can support ISD by reducing complexities in the challenges mentioned above and provide a guideline to effectively collaborate with parties involved and coordinate activities. Here, by collaboration we have referred to a process of participation through which people, groups, or organizations work together to achieve desired results with a shared vision and jointly build an interdependent system to serve targeted customers. By coordination, we referred to a consciously organized relation between activities and forces, work tasks are divided over actors and the act of making different agents (i.e. people or things) work together. Thus, the aim of this thesis revolved around the main research question, which was:

\[ How\ can\ Agile\ management\ and\ service\ development\ principles\ be\ incorporated\ together\ for\ effective\ collaboration\ between\ parties\ and\ coordination\ of\ activities\ in\ Integrated\ Service\ Delivery? \]
In order to resolve this research question, there were four sub-questions to be answered. The sub-questions are mentioned in the following sections. Before providing a concrete solution to the main research question, it is important that we examine whether the sub-questions are answered in this thesis. First, SQ1 is investigated and determined if it is answered to derive the requirements for incorporating Agile management and service development principles in ISD. Second, conclusions to answer SQ2 are given to developing the artifact of this research. Third, conclusions regarding the evaluation of the artifact of this research is provided and investigated on whether SQ3 is answered. Finally, it is determined whether SQ4 is answered regarding the contribution of this research in practice.

9.1.1 Groundwork for Developing the Conceptual Model

This section discusses on the groundwork that was established to develop the initial conceptual model answering following sub-question:

SQ1. What are the requirements for incorporating Agile management and service development principles in Integrated Service Delivery?

The answer to SQ1 has been formulated by performing a literature survey on service development and the Agile management answering the following questions:

SQ 1.1 What are the challenges faced in Integrated Service Delivery and what are the characteristics, phases and roles that can be derived from theories on service development?

SQ 1.2 What are the characteristics, phases and roles that can be derived from Agile management?

The following text summarizes to what is covered chapter 3 and chapter 4 encompassing the elucidation of the above questions.

Integrated Service Delivery

Before finding out the requirements to incorporate in ISD, we first have looked at the definition and meaning of Integrated Service Delivery (ISD). With respect to several definitions from literature, we consider ISD 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’. When developing these integrated services, studies have shown that service providers face a number of challenges. These challenges are related to:

- Organizations integration: Successful integrated service delivery, which in the cases of different departments working together in one company or multiple organizations collaborating together, require amalgamation. Challenges include addition of staff working under different work processes, standards or different collective agreements in case of multiple organizations. 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. It is also important that partners responsible for the maintenance operation of an ISD initiative once it is up and running, be identified and agreed upon in the provisioning stage. Effective communication, a shared understanding of roles and responsibilities, and a collaborative method of resolving issues are considered to be key factors to a successful partnership. Furthermore, to ensure successful partnership discussion, it is important to discuss on topics, which are: decision-making protocols, change management processes, issue management processes, information sharing protocols, timelines, and performance review and monitoring.

- Embracing change: The reality is 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. In one case study, it was noted that many managers felt a general lack of comfort when dealing with things outside their particular span of control although having a clear vision. By embracing change and integration, companies can innovate and advance rapidly.

— **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. 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 and decline their growth in the market.

Thus to overcome the above challenges, it is important that organizations effectively collaborate among the parties involved and coordinate activities accordingly to adapt to changing conditions of the market. Understanding the challenges and nature of ISD, we then have delved in to the basics of service development that surrounds ISD. Looking into the characteristics of services in general, we have studied the service lifecycle and roles involved in it, serving the requirements for answering SQ1.1.

With the characteristics identified, the service lifecycle has been investigated. Among the analyzed several lifecycles, a combination of the lifecycles proposed by Papazoglu & Heuvel (2006) and Gu & Lago (2007) have been chosen based on the theoretical aspects of management of services. The phases within these lifecycles share commonalities, which were later grouped in to three main phases: design, runtime and change. The activities within these phases have been described in details. The roles involved within the service lifecycle have been elaborated upon, defined by Gu & Lago (2007) and are the following: service provider, service broker and service consumer.

**Agile Methodology - Scrum**

In order to find out how an Agile approach can support ISD, we first have looked at the background of Agile software development methodologies. Understanding the differences, we have chosen Scrum as the methodology to support the needs of APISD. The reason behind this is, compared to the various methodologies, Scrum focuses on the management of the system development process rather than delving into a part of the process and defining development practices. Another rationale is that since Scrum does not require any specific engineering practice, it can be adopted to manage whatever engineering practices are used in an organization both in existing projects and new projects. Adopting the principles of Scrum allows the flexibility of adapting to conditions as required, which is a crucial factor in ISD. Looking into the characteristics of Scrum in general, we studied the Scrum lifecycle and roles involved in it, and whether it can serve the requirements for answering SQ 1.2. The Scrum lifecycle consists of three phases: pre-game, development and post-game which were elaborated in chapter 4. In addition to the lifecycle, the main roles in Scrum were explored. These main roles were defined in details and are: scrum master, product owner, scrum team, client, management and user.

Hence, deriving the characteristics, phases and roles required in ISD and in Scrum, SQ1 is answered by determine the corresponding requirements required to develop the initial conceptual model. These requirements have been used to develop the model, which is further discussed in the following section.

**9.1.2 Development of a Conceptual Model**

To answer the sub-question, SQ 2, ‘What is the artifact resulting from this research that represents the amalgamation of service development and Agile management principles to
portray collaboration of parties involved and coordination of activities in Integrated Service Delivery?; a conceptual model has been developed following the necessary constructs and activities within this model has been described comprehensively.

With the groundwork of literature, the intentions for developing the conceptual model was understood. Both service development and Agile management shared some common characteristics, which vindicated the amalgamation in producing a process to support ISD and portray the collaboration of parties involved and the coordination of activities within the lifecycle, covered in chapter 5.

Following the necessity of the fusion of both these concepts to support ISD, a set of phases required for the development of the conceptual model was derived. The phases resulted from blending the service lifecycle and Scrum lifecycle. These phases are:

- Planning
- Service Modeling
- Service Construction
- Provisioning
- Deployment and Execution
- Service Management

Each of these phases was further detailed following the initial conceptual model for APISD. In addition to phases, roles have been defined as well, which are required to fulfill the responsibilities and perform activities in APISD. They are:

- Service board
- Domain Manager
- Service Analyst
- Development Manager
- Chief Service Developer/Architect
- Service Developer
- Service Tester
- Release Manager
- System Administrator
- Customer Services
- Client
- User

Each of these roles has been defined based on their responsibilities and involvement in different phases.

Following the derivation of the components, a conceptual model has been developed, which displays the phases mentioned above in a lifecycle. This conceptual model is the artifact of this research and is titled Agile Process for Integrated Service Delivery (APISD). Each phase has been described with the necessary activities performed by the involved roles given above. The details for this lifecycle have been elaborated along with the components produced from the process.

The APISD model enfolds the coordination of various activities that are required to deliver integrated services and discusses how different parties collaborate and communicate with each other. To tackle issues regarding the challenges related to organizational integration, change adaptation and customer satisfaction in ISD, APISD explicates the interaction of the service provider and the client in two different situations:

- a single service provider delivering integrated services to the client;
- multiple service providers delivering integrated services to the client, collaborating together to perform the necessary activities.

In either of the situations, parties involved are the client and the members of the service provider(s). As both intra and inter-collaboration is required in these cases, which Agile
supports, the APISD model addresses the primary challenges with the incorporation of the common characteristics of service development and Agile management. They are:

- **Multiple service provider collaboration:** in APISD common goals, assumptions are establish to build trust in the beginning of the lifecycle, the Planning phase. A detailed description of roles and responsibilities has been specified that can provide a shared understanding, enriching the collaboration between the parties.

- **Change adaptation:** due to the nature of the APISD model, which is iterative development, the process allows adaptation to change flexibly. The introduction of iterations before the construction of the services, demonstrates that changes occurring during requirement specification and designing can be easily adapted in the subsequent iterations. By incorporating iterative development in the Service Modeling phase, it reduces the complexities of having to define requirements and designs in one phase (as suggested in Waterfall approach) and not being able to re-iterate back just in case change is required. Similarly, comprising of the iterative development in the Service Construction phase, client feedback can easily be adapted to. In changing conditions of the market, the service board can easily change priorities of the services in a shared system, the service backlog, which is easily accessible throughout the organization to monitor.

- **Client collaboration:** APISD enables service provider(s) to have close interaction with the client from the beginning so that they a continuous focus on the main desire of ISD, increasing customer service by being customer-centric. For example, understanding what the client needs in the Planning phase, involving the client to understand their requirements in Service Modeling phase, involving them frequently during iterations in the Service Construction phase and negotiating with them formally in the Provisioning phase via SLAs.

Furthermore, for effective collaboration between the internal members of the service provider(s), the following concerns of the challenge are met by coordinating the necessary activities:

- **decision points:** 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, a change backlog artifact was introduced by which corresponding service change requests are made by client, prioritized by release manager, and implemented by the development team(s);

- **issue management processes:** to manage the issues, an incident backlog was introduced by which incidents reported by users are logged into, prioritized by the release manager and later implemented by the development team(s);

- **information sharing:** to have a consistent flow of information, teams are able to retrieve the requirements set in the service backlog and feature backlog;

- **performance review and monitoring:** to monitor the progress, APISD consists of retrospective meetings that are held in the teams once the iterations are completed. These retrospective meetings comprises of lessons learned and identification of improvements to be implemented in future iterations.

In conclusion, SQ2 is answered by developing a model called APISD, which incorporates Agile management and service development principles portraying the collaboration of stakeholders involved and coordination of activities to meet the primary challenges of ISD. The following section will answer SQ3, which will result in conclusions regarding the evaluation of this model.
9.1.3 Evaluation of the APISD Model

By keeping two objectives in consideration, the evaluation of the APISD conceptual model answers the sub-question: 'SQ 3. How is this artifact validated to evaluate the collaboration between parties and the coordination of activities in Integrated Service Delivery?' The first objective was to test the model with the case study, analyze the data based on the missing criteria and differences between the artifact and the organization's process. The second objective was to analyze those data and derive additional factors to expand the initial conceptual artifact and producing another enhanced version. This model was not tested in terms of implementing it in an organization and observing the impacts created and issues found. The answer to the above question has been formulated by performing a case study research and evaluating the model in comparison to the case-organizations model. Analyzing their feedback, the differences from existing organizational process and APISD have been identified along with the derivation of the additional factors. The following text summarizes to what has been covered in chapter 6 and chapter 7.

A case study has been conducted that consists of three cases and is related to the development of integrated services in three different organizations. This case study validates and evaluates the constructed initial conceptual model for APISD. Data has been collected by analyzing documentation both from web-site and provided by the companies, retrieving feedback from questionnaire, conducting interviews and by letting key informants review the case study analysis. The cases were selected based on a few selection criteria. In order to validate the gathered data, four validity tests have been applied: construct validity, internal validity, external validity and reliability. Construct validity was relevant and increases due to multiple data sources acting as the chain of evidence and as informants have reviewed the case analysis. The internal validity of the model was irrelevant as the nature of case study was explorative instead of explanatory or causal. External validity has been recognized since the research allows theoretical replication and the model can be replicated for evaluation in other similar case study. Finally, the reliability of the case study has been obtained by allowing the case study procedure to be followed and stored in a case study database to retrieve data.

Interviews were conducted in three organizations and three cases were analyzed. Each case was analyzed based on five factors: current organizational process, common factors between case and APISD, missing factor in case in comparison to APISD, distinction in APISD in comparison to case and additions for APISD. Key findings of the case study research were that organization representatives compared with their existing processes and commended the APISD model in the following aspects:

- Importance of a separate phase regarding provisioning;
- Importance of division of the design phase from 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.

With respect to the analysis of the feedback, a number of factors have been identified, which are considered to enhance the initial model. These findings are summarized in the following and have been highlighted in the beginning of chapter 7:

- 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.
- The service design team creates high level designs that are for logical components, deployment, infrastructure and code designs, while the detailed designs can be created by the development team.
— In APISD, the customer service was involved in the Service Modeling phase to review the service backlog produced by the service analysts.

— The system administrator was included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and provide input.

— For team acknowledgement and satisfaction, an activity was incorporated in APISD. The demo given to the client after each iteration in APISD can be recorded and shown to the construction team if they are not involved in it. This will increase the satisfaction and give an impression how the client behaves towards the system.

— 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.

— When there is issues to be solved that are urgent given by the client for existing services, one team member is involved in implementation of that issue, while the rest of the team members should focus on the rest of the service development.

— In case of rejection by the client after post-production, an activity is included in APISD for analyzing the problems and producing possible solutions.

— There is a workflow created for the service analysts to also visit the users’ work-floor and observe their interaction and engagement with the integrated services.

— APISD incorporated the practice of service provider(s) engaging in adopting best practices and standards for service management.

Meeting the two objectives of the evaluation of the artifact, APISD model, SQ3 is answered by testing the model, analyzing the differences between the model and organization’s processes and determining additional factors that have been used to enhance the APISD model. The following section summarizes on the finalization of the model and on how it can be used in practice drawing conclusions to answer SQ4.

9.1.4 Finalizing the Conceptual Model

From the analysis of the case study findings, the additional factors have been distinguished per phase encompassing the differences in the cases and the initial conceptual model. The factors are further discussed on how they can be incorporated in the model. Therefore, with the inclusion of the factors, the initial conceptual model developed in chapter 5 has been revisited and enhanced in chapter 7. The improved version of the APISD model corroborates SQ 2 answered in section 9.1.2. It complements the incorporation of Agile and service development principles in the activities of APISD concerning the challenges by focusing more on the collaboration and coordination aspects.

Contribution of this research

In order to get the picture of how this research can contribute to the implementation within an organization, chapter 8 has been included in the thesis to materialize the APISD model via an illustrative case. This chapter answers the final sub-question: ‘SQ 4. How can this artifact be used in practice in management of Integrated Service Delivery?’ The model was materialized in two scenarios: for a single service provider and for multiple service providers collaborating together to deliver integrated services. This research can guide practitioners, since currently organizations are using the Agile methodology but are following a traditional way of working in the integrated service development (observed from case study). By having a holistic view on how to manage the service lifecycle of ISD with a focus on the collaboration of parties involved in the process and coordination of activities, working in an Agile approach can bring performance gain to the companies.

Finalizing the model has enabled us to answer the main research question addressed in chapter 1 and in the beginning of this chapter. By implementation of this process, ISD can possibly be achieved in a structured manner and support on-demand services in a fast changing environment with effective collaboration between parties and coordination of activities with the support of Agile and service development principles rooted in. Also, as the
model was illustrated to demonstrate the activities applying to a practical case, the key benefits by implementing this process in the organization can be achieved are:

- smooth collaboration and integration between service providers themselves and between service provider and client;
- better understanding and management of the requirements;
- an iterative and prioritized process of using and re-using services;
- handling the implementation complexities in the iterative development and meet the demands of the clients as needed;

The importance and attainment of this research is further grasped by the informants interviewed. Much appreciation is expressed by the following quotes:

"I think it is interesting and good that you define this process. Maybe you can send this to us?"

"Can I take this process with me?"

"We can think of this by ourselves or now that I know someone who is already working on it... we can use some part of it?"

This section concludes with answering the sub-questions, which covers the necessary knowledge to satisfy the main research question: 'How can Agile management and service development principles be incorporated together for effective collaboration between parties and coordination of activities in Integrated Service Delivery?' It is answered as follows:

For the effective collaboration between parties and the coordination of activities in Integrated Service Delivery, the Agile management and service development principles have been amalgamated together in an integrated and iterative process called Agile Process for Integrated Service Delivery or APISD. This process tries to overcome the challenges of ISD that are related to the organizational integration, change adaptation and customer satisfaction by adopting agile practices in ISD and by illustrating the necessary coordination of activities and interactions of the different involved stakeholders in the ISD lifecycle.
9.2 Limitations & Opportunities for Further Research

This research has been conducted to a certain extent. Given the aim of this research to develop a conceptual model for APISD, there are several limitations of this research that can be investigated upon in the future. This section addresses these limitations and discusses on opportunities for further research to overcome them.

The APISD model has been developed as a conceptual model but not implemented within an organization due to time constraints of conducting this research. Therefore, the limitation of this research is that it cannot be empirically concluded that this process will result in efficient collaboration between the parties involved and coordination of activities in the APISD lifecycle. In order to solidify the model, evaluation is required in terms of implementing this model in an organization and observing the impacts. By implementing it, further findings can be gathered. These findings can help to identify the gaps in the model and make it stronger. Using it in several organizations will help to increase the generalizability of the model. Internal validity can be achieved by understanding the causal relationships within the process once it is implemented.

From the usage of this model in various organizations, the practicalities within the APISD process can be refined. The inner constructs such as artifacts, meetings and activities can be investigated further and adapted to the needs. Furthermore, guidelines and best practices can be incorporated in the process.

The cases conducted in the case study research provided a sufficient amount of information but is limited to a general selection. 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.

Adopting Scrum as the Agile approach, the process inherits the problems faced in Scrum such as project scaling and team scaling. Scrum teams suggest not having more than 10 members involved because the management of conducting daily meetings is complex. Therefore, in APISD, it is suggest having weekly meetings to reduce this problem. In addition, since the number of service providers may be multiple, causing team scale and project scale to increase, Scrum is difficult to maintain. However, moderations to Scrum are made to support such situations by managing the complexities and uncertainties by adopting a context-adaptive agility (Little, 2005), which determines what process practices are sufficient for a particular scale of project. In the projects using SCRM, the vision of the client highly influences development. Highsmith and Cockburn (2001) show that if the client does not have a clear sense of the product’s direction, the members of the development team will tend to behave in the same way, and the final product can be significantly different to what is expected. Therefore, one of the main weaknesses of Scrum is precisely one of its strengths: client involvement in the development process.

Another limitation of this research is that it does not cover the political context sensitivity of organizations accepting this model. Issues regarding resistance to accepting this model for implementation, commitment of multi-actors and their accountability are not addressed in this research. Opportunity exists 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.

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. Without establishing such norms for sound scientific results in the community, it is likely to remain an area that lacks knowledge on the impact of its own approach.
9.3 Epilogue

The objective of this research was to develop a conceptual model for supporting Integrated Service Delivery by overcoming its challenges by having effective collaboration and coordination with help from Agile management and service development principles. This chapter has concluded with an answer to the main research question and solutions on how to overcome the challenges. We believe that the findings presented in this research will help to provide new insights for everyone who delivers integrated services. The approach presented in this thesis is not fully a recipe book yet and does not prescribe all the steps required. However, an illustrative case is provided on how the process can work out in a real implementation. Optimistically, the concepts explained and process elaborated will be used in practice and improved over time in delivering integrated services.
Thesis Reflection

After completing this thesis, there are several things that I have learned and have reflected back upon. With this reflection, I have realized that my researching skills, argumentation and thinking capabilities have improved significantly.

I have tried to work with something new in this research. I feel innovation is very important to sustainability and through research the production of new ideas and implementations, it makes it possible to advance. With the knowledge gathered from my period spent studying in Management of Technology, I have tried to create a bridge between the technology and the organization of it. Being inspired from courses such as ‘Technology and Strategy’, ‘Innovation Management’ and ‘E-business’. I was driven to explore the aspects of combining new trends of methodologies that organizations are approaching towards. I believe that with this research, both the industrial and scientific community can acquire an insight on the holistic view 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.

This thesis, in my opinion, was one of my best research work. My previous researches were followed in a pragmatic and unstructured way. Doing my master’s helped me to structure my work and to create a theoretical groundwork for my assignments. In previous research papers, I used a few main sources to quote or get information from. But now that I had used a larger variety of sources, I had a better understanding of the topic that I argued about. Understanding more about the topic helped me to analyze better and to make a better argument. In addition, using this large variety of sources also allowed me to get information about my argument from many different perspectives. With these different perspectives, I was able to create a conclusive argument that set the tone for the entire thesis.

Although working in a structured approach, following the Design Science Research Methodology helped in iterating to some steps back to further refine the work done. For example, after completion of the case study analysis, I have reiterated back to finalize the initial conceptual model. I have also re-iterated to refine the research questions and draw conclusions accordingly. It is possible to conduct this research in small iterations where I could have derived high level requirements at first and then try to start with the conceptual model. However, I felt the structured flow of steps in the methodology helped me first to create a vision, second to gather knowledge in broader sense to fully understand what my research is all about and then narrow down to produce the artifact; then to validate the artifact with case study research and finally to conclude in a final model. With some iterative steps, it helped to modify and redesign the content accordingly.

Moreover, I have learned how to better argue a point. A reader takes my arguments more seriously when I have facts to back them up, especially facts researched from credible sources. Using facts and examples from theory, my arguments to arrive to a good model became better. When commencing with the actual research, I only had an idea about what I was going to be working on. I did not have a solid, conclusive argument. As time passed and as I delved into the topic, I was able to develop this topic.
One specific experience that I never had before was interviewing for my research following a case study research strategy. Initially, I found it challenging to prepare and conduct the interviews. However, much of the difficulty was diminished with the immense support of my supervisors. Their guidance throughout the thesis helped me to steer in the right direction. Also, the theory behind interviews and data collection methods learned in the courses 'Internal organization of firms' and 'Special methods and techniques in research', taught in TU Delft, have helped me in gathering knowledge to apply in this thesis. I have learned much more on how to interview, how to analyze the interview feedback and how to incorporate it in my research.

Apart from my learning, there are some aspects that I believe I could have done better. In the beginning of my research, just as important as it was to focus on the theory, it is equally important to pay attention to the case study research. If I had given more awareness to the case selection, the choices of my cases would have been different and refined, although the cases that I have conducted are sufficient enough for the preliminary study and analysis of this research; as this research is a new endeavor to combine the Agile management and service oriented principles in ISD. In this thesis, I have set the criteria towards the development of the conceptual model. However, I have learned that with a strong understanding of the theory, the aim of the research and only after the completion of the conceptual model, if the case selection criteria could be set, it would have been more specific to fit the needs of the research. Nevertheless, it helped me realize the importance of it and provided me a learning that I can take into account for my future research, so that I can build a firmer work.

In spite of all the experiences involved in the duration of my research, it was an exciting period. I have been very motivated throughout and enjoyed doing all of it. Two highlights of this research have stimulated my interest in this research. First, seeing the motivation and support of my supervisor to go through with this topic and second, when two companies were interested in the implementation of this process within their organization. With this enthusiasm, I hope that I can bring forth this research to the scientific community and industry to enable its materialization and future work enhancements.
References


Appendix I: Interview Questions

— How important do you rate the following objectives when providing services?
  o Quality, delivery time, cost, reusability of services, other important objectives

— Does your organization communicate with her clients directly?
  o If answer to the question is ‘No’, does your organization communicate with her clients via intermediary client representatives?

— Your organization collaborates with her clients regarding: Requirements specification, requirement clarification, product demonstration, all mentioned or none of them.

— Is your organization involved in collaborating with other organizations to develop integrated services?
  o If answer to the question is ‘No’, what process is used within the organization?
    ▪ What are the phases involved within that process?
    ▪ What are the distinguished roles in that process?
  o If answer to the question is ‘Yes’,
    ▪ How is the communication with the various organizations arranged around the process?
    ▪ What are the phases involved within that process?
    ▪ What are the distinguished roles in that process?

— What are some typical bottlenecks faced when collaborating with the different organizations when implementing integrated services?

In the explorative interview, first the model was presented explaining the various constructs within it. Then the following questions were asked in an open form, resulting discussions and bringing out the variations.

— To what extent would this model fit your process?
— Would you like to implement this process within their organization?
  o If answer to the question is ‘No’:
    ▪ Why do you think it will not fit?
    ▪ What are the missing criteria required in this model to fulfil the needs?
  o If answer to the question is ‘Yes’:
    ▪ If this process was to be implemented in your organization, how would it work?
Appendix II: Sequence Diagrams

Appendix II Figure 1: Planning Phase
Appendix II Figure 3: Service Modeling Phase - Design
Appendix II Figure 4: Service Construction Phase - Development
Appendix II Figure 5: Service Construction Phase - Release Planning

1. Discusses release date
2. Sets possible release date
3. Discusses feasible release dates
4. Finalizes service release date
5. Prepares necessary release activities
Appendix II Figure 6: Provisioning Phase
Appendix II Figure 7: Deployment & Execution Phase
Appendix II Figure 8: Service Management Phase