Spring 2010

An Agile Ba: a Case Study of the Business Analyst in Agile

Nathan Wagner
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

AN AGILE BA: A CASE STUDY OF THE BUSINESS ANALYST IN AGILE

A THESIS

SUBMITTED ON APRIL 29, 2010

TO THE DEPARTMENT OF INFORMATION TECHNOLOGY

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

COMPUTER INFORMATION TECHNOLOGY

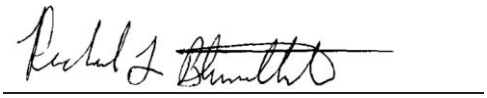
BY

Nathan Wagner

APPROVALS

Shari Plantz-Masters, Thesis/Project Advisor

Nancy Birkenheuer



Richard L. Blumenthal

Abstract

Companies adopting Agile software development methodologies are becoming a growing trend. The roles responsible for managing and participating in these types of projects must make a significant shift in focus when moving from the waterfall development process to a more adaptive way of working. Even with the growing adoption and increasing amount of study around Agile, the literature and texts prescribing the various Agile methodologies do not address the specific role of the Business Analayst. In this paper, the role of the Business Analyst in Agile is explored through a qualitative study within a large software development organization utilizing Agile development. Through the course of directly participating and observing Agile project work, five distinct themes of the BA role were identified: Communication, Agile Process, Analysis, Prioritization, and Dependency Management. The study concludes with a discussion of the limitations of this inquiry and approach as well as suggestions for further research.

Acknowledgements

I want to thank my thesis advisor Shari Plantz-Masters who provided the necessary guidance for me to shape my project and for keeping in contact with me over the long period it took for my project to complete. I want to express my appreciation for the staff and instructors at Regis whom consistently provided a professional, friendly, and challenging learning experience for me.

Finally, I want to thank my family and friends for their support and encouragement throughout this process. Special thanks to my wife Jen who helped me tremendously and without her assistance, I would not have been able to complete this project.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

Agile development methodologies such as eXtreme Programming and Scrum are being adopted more widely in organizations that want to become more adaptive to changing market and business conditions (Pikkarainen et al., 2008). Agile development methods present a stark contrast from the traditional way of developing software. The so called process "heavy" methodologies which were predominately in place from the late 1970s to 1990s emphasized intense periods of up front design captured within lengthy requirements documents followed by periods of development (Surendra, 2008). On the other hand, Agile methodologies focus on working closely with the customer throughout the entire process, incorporating changes along the way to deliver the most value in the shortest possible time.

With a significant shift required to move from the traditional plan-everything-up-front methodologies to a more adaptive way of responding to software development challenges, a number of issues can be encountered. This would especially be true for employees working at organizations where a move to Agile also translates directly into dramatic changes in the way one works and interacts with other team members. Such is the case at my organization.

At my organization, a Business Analyst is primarily responsible for performing requirements engineering which can be defined as "The process of discovering, analyzing, modeling and specifying business and user requirements for an information system…" (Nguyen & Shanks, 2009, p. 655). The International Institute of Business Analysis (IIBA) formed in 2003 to provide guidance and to promote the Business Analyst community also discusses the different named roles (including requirements engineering) that are under the umbrella of work that a Business Analyst performs (IIBA, 2009). For this discussion and the remainder of the paper, a

Requirements Engineer can also be labeled as a Business Analyst and the terms will be used interchangeably.

Cao and Ramesh (2008) found that the process of performing requirements engineering differs substantially in Agile software development compared to traditional development approaches. The importance of requirements engineering in the process of software development cannot be overstated. Research has pointed to this importance (e.g., Bhat, et al., 2006; Schreiner, 2007) arguing that the requirement engineering phase of a project is the most critical for executing a successful development project. Although there is much research in the area of Agile adoption as a whole, Cao and Ramesh (2008) note that there is little study specifically focused around the Requirements Engineer in Agile.

Furthermore, only recently has there been formal guidance provided to the Business Analyst community as to the role of the BA in Agile methodologies (IIBA, 2009). The IIBA recently released the 2.0 version of the Business Analyst Body of Knowledge (BABOK) in March of 2009 which is a guide to the skills and techniques generally required by Business Analysts working in the field. It can be considered the de facto formal resource for defining the role and tasks for Business Analysts. The IIBA is the only organization that currently certifies Business Analysts (Certified Business Analysis Professional) and the BABOK is the basis for the certification. It is important to note that until this new version of the guide, there was no mention of Agile development nor how a Business Analyst would perform his/her tasks inside an Agile development framework. Given this lack of research and reference, more investigation is warranted in this area.

While there are many challenges faced by organizations on their path to Agile adoption, the focus of this study will be specifically on the Business Analyst role. The research question to

be answered in this study is: What is the role of the Business Analyst in an Agile development process? It is the aim of this inquiry to provide an answer to this question through a qualitative study within a real-world Agile development organization. Not only will this account be a valuable addition to the Agile adoption research area, but it will also assist the Business Analyst community and the individual analysts making the transition to Agile development. This information will provide insight and suggestions into additional areas of research around the role of the Business Analyst in Agile.

The following introductory sections present background and context to this inquiry. The first section (*1.1 Agile Methodologies*) introduces some of the more common Agile concepts with a heavy emphasis on describing the Scrum development methodology in practice at the company where I am currently working. In section *1.2 Company Background*, a narrative description is presented around the organization. This section provides historical perspective, key characteristics, and examples of the decisions leading up to the organization's Agile adoption. This section also highlights why I decided to investigate the research question.

1.1 Agile Methodologies

Agile development methodologies were born from a philosophical viewpoint that placed a higher value on producing a working product as soon as possible, bringing value to the customer from their perspective iteratively through constant interaction, and embracing change as a natural exploration in the learning process of building a complex system (Agile Manifesto, 2001). These values were to be placed higher than the traditional practices of software development where all of the planning and design had been traditionally completed up front in the process with little or no room to change midstream once the project was placed in motion.

The failure of this old way of thinking about software development has been well documented and first appeared in a Standish Group study completed in 2001. That study reported that 30% of initiated software development projects were completely abandoned before completion and over 60% were considered to be failures by the project stakeholders (The Economist, 2004). As Highsmith & Cockburn (2001) have noted, "Traditional approaches assumed that if we just tried hard enough, we could anticipate the complete set of requirements early and reduce cost by eliminating change" (p. 120). They further argue that trying to eliminate change in a project means being unresponsive to the market and business opportunities, and thus experience project failure.

While the specifics of the various Agile methodologies differ in their suggested practices, they all share a set of twelve common principles that have been outlined by the Agile Manifesto (Agile Manifesto, 2001). These common principles center around teams being able to self organize, iterative development cycles with short feedback loops, embracing change driven by close customer interaction with the working product, and utilizing face to face communication when possible. An important aspect of the feedback loop is to not only show the customer a working product to get their feedback, but it also involves a team looking at their own processes to reflect on what can be improved in the next short development cycle.

There are several commonly used and accepted Agile methodologies in practice today. This discussion will focus on Scrum as that is the process being followed at the case study organization. eXtreme Programming and Crystal are also common agile approaches (for eXtreme Programming see Beck & Andres 2004; for Crystal, see Cockburn 2007; Cockburn 2005). The co-creator of Scrum, Ken Schwaber, describes Scrum as set of guidelines which help drive the process of software development rather than a traditional prescriptive methodology

where all the rules are to be followed in any given circumstance (Schwaber, 2004). However, there are a few rules that should be followed and key roles that must be in place in Schwaber's view in order for teams to successfully implement Scrum.

The first rule of Scrum involves a planning meeting that must occur at the beginning of each development cycle. The development cycle is called a sprint, and it is recommended to be time constrained to 30 days, although many organizations use shorter or longer durations. The role called the Scrum Master is responsible for planning and facilitating the sprint planning meeting. The Scrum Master is typically the project manager, but in Scrum the role is more of a coach and facilitator than that of the typical project manager (Schwaber, 2004). This sprint planning meeting is time constrained (as is most everything in Scrum). The meeting is designed to create a comprehensive list of prioritized chunks of functionality that form the whole of the product vision. This list is called the product backlog (Schwaber, 2004).

The sprint planning meeting is also used to plan the next sprint of development work. In this planning session, the team and product owner decide what items the team is willing and able to commit to completing for that 30 days of work. The product owner, who is either a customer of the product or a customer representative, must be available to the team during this meeting. The product owner is responsible for producing and prioritizing the list of items in the backlog and the development team is responsible for educating the product owner on how complex each item is to build and to commit to completing the list of items by the end of the sprint period. The goal being that at any given sprint end, the product owner can decide to ship the product as is and immediately receive value from their investment (Schwaber, 2004). This is possible because the product owner places the highest value items at the top of the backlog (constantly re-prioritizing at each sprint review meeting with new information or market changes), and the team commits to

building each item completely (QA testing, integration, etc…) by the end of the sprint. Once a commitment is made to the sprint backlog of items for that development cycle, the list of items cannot be changed until the next sprint planning meeting. This is to keep the team from losing their focus and to insulate the team from outside influences that would distract them from meeting their goals and commitments.

The next rule that must be followed for a Scrum team involves a daily stand up meeting of about 15 minutes where the team members building the product are required to answer three simple questions: What did I do yesterday? What will I do today? What is blocking me from getting my work completed? This meeting is called the daily Scrum (Schwaber, 2004). The Scrum Master is responsible for planning and facilitating these meetings, but will pay particular attention to those blocking issues identified by the development team and work toward removing those obstacles. One of the primary duties of the Scrum Master is to insulate the team from any outside distractions and to remove any barriers to the team keeping them from meeting their sprint commitments (Schwaber, 2004).

The daily Scrum standup meeting is intended to increase visibility into the daily progress of the development effort and to spur follow-up discussion between team members. The meeting itself should not stray outside of the three questions, and the Scrum Master is responsible for making sure these meetings are quick and stay on task. It is a fundamental value within Scrum that when you restrict the time (time-boxing) of activities, then time will be prioritized and properly respected by the team members (Schwaber, 2004).

After the daily standup, the next rule of Scrum is that at the conclusion of the sprint development period, there is a review meeting where the work produced in that development sprint is demonstrated to the product owner and any other interested product stakeholders. This

meeting is intended to formally showcase the working product and to illustrate the development commitments met (or not met) by the team (Schwaber, 2004).

Once the sprint review meeting is concluded, the final rule of Scrum is that the team holds a retrospective meeting (Schwaber, 2004). The retrospective meeting is a chance for the team members to reflect on what they did well in the previous iteration, and what can be improved upon in the next sprint period. This is a feedback loop setup for the team to improve upon their processes in developing working product for the next sprint, and should result in actionable items for improvement. Nothing is off-limits, but anything suggested must be agreed upon by the team and any change to the Scrum rules has to be approved by the Scrum Master (Schwaber, 2004).

After the retrospective is held, the team gets back together for their next sprint planning meeting and the process repeats until either the backlog of product features is exhausted, or the product owner decides enough functionality has been built to ship or deploy the product into the market and stop development. Ongoing maintenance can continue on the current product and / or another product can be spun up depending on how the organization is positioned. The idea being that the team can continue in this iterative development structure indefinitely as long as there are products to be built or supported.

1.2 Company Background

The setting for this study was conducted at a large software and services organization where I am currently employed. My organization's products and services are offered to our clients as a Software as a Service (SAS) model, meaning that the software is delivered and hosted by the organization and delivered to the client via the Internet over a web browser.

Another model for software delivery is a client side or install locally delivery method whereby a client installs the software on their local machine or hosts in their own environment and the software then runs in isolation. This distinction is important to highlight. In a SAS model many clients access the same version or instance of the product and therefore one change to our software affects all clients. Because of this risk, my organization has a strictly gated and regimented release process.

The switch to Agile for my organization was brought about after a lengthy update of the core delivery software to a new version with a migration strategy to move clients from the old version to the new platform. The update took over two years to complete and was plagued with defects. More significantly, after the new version was finally released the organization found that it was behind many of its other competitors in terms of features and functionality. Even though the new version was ported to a more stable and extensible development architecture, the current customer base did not perceive any real difference in features or functionality and thus they saw no benefit in going through a migration.

From this experience, the company realized that we needed to make a change in the way we produced and maintained our product. In order to maintain and grow our current customer base, we needed to be in a position to quickly respond to our market needs, and also have a way to differentiate our old version from the new to entice our existing customers to migrate. Furthermore, sustaining and maintaining two different versions of our software is costly and adds complexity to every layer in the delivery and support process (i.e., Development, Help Desk, QA, Product Prioritization, etc…).

After much research and consideration, Scrum was chosen as the Agile method to be implemented and customized within my organization. The decision to implement Agile was

adopted from the top down. Senior management was in full support of funding and making the necessary organizational changes to implement Agile. In order to assist with the transition, an Agile coach was hired from an outside consulting firm to train management and team members on the basics of Scrum and to help the projects and different groups transition into the new process. A pilot project was identified for the first phase of the Scrum implementation, and shortly after most projects and teams (including the offshore development teams) were fully transitioned to Scrum. Some of the Scrum rules were modified slightly (such as two week iterations instead of 30 days), but for the most part all of the Scrum rules continue to be followed by teams.

Almost two years after making the move to Agile, the organization has been able to start addressing the core challenges described above with the old software delivery method. We are able to release requested features more often, and we are able to quickly adjust to changing priorities or market conditions. We have also demonstrated to our current customers the advantages of migrating to a modern and extensible software architecture by giving them differentiating features in our new version faster and more reliably.

However, the shift to Agile has not been without challenges. Although it is beyond the scope of this paper to speak to all of the challenges of this company-wide Agile adoption, I can speak to how difficult it was for me to enter into this new environment and my new role as a Business Analyst. My main issue initially was that there were abundant sources of information on how a BA works within the traditional waterfall software development process, but very little information around how a BA is expected to contribute within an Agile process. At the time of my hire, the new edition of the Business Analyst Body of Knowledge had not yet been published, and there was little to no mention of a BA or a Requirements Engineering role within

Agile in anything but discussion groups or blog sites. Agile development was something completely new to me, and I was looking for a more formal and accepted set of practices and tools that a Business Analyst could refer to within an Agile process for guidance. I quickly realized that the information and resources I was looking for did not exist. It was this lack of formal information that has led me to my research question.

Chapter 2 – Review of Literature and Research

The following discussion is a review of the relevant literature taken from peer reviewed journals and conference proceedings for requirements engineering within Agile development. The review will first focus on the current state of Agile development research. Next, the limited amount of research available relevant to Agile and requirements engineering will be evaluated. The theme throughout this literature review is to highlight the lack of research around the actual practice of requirements engineering within Agile. There are a number of case studies around general Agile adoption and Agile practices (e.g. Lindvall, et al., 2004; Drobka, et al., 2004; Layman, et al., 2005; Schatz & Abdelshafi, 2005; Ceschi, et al., 2005; Hanssen & Faegri, 2006; Nottonson & DeLong, 2008). However, these studies either fail to mention requirements engineering or do not provide any specifics into the practice of performing requirements engineering within Agile. Therefore, the scope of this literature review will not include research on general Agile adoption unless it includes an emphasis on requirements engineering.

In 2008, researchers Dyba and Dingsoyr published a systematic review of empirical studies for Agile software development. Studies (including conference proceedings) published up to and including 2005 were evaluated and categorized into groups of related topics for analysis. The aim of their review was to locate and evaluate all of the current empirical research around Agile development in order to provide a discussion on the state of the research. The researchers also provided some discussion on the strengths and weaknesses of current Agile development approaches. Dyba and Dingsoyr used critical evaluation criteria to narrow down the studies to include in their discussion. For example, studies based solely on lessons learned and expert opinion alone were not considered empirically based studies and were not included for review. The results of this evaluation yielded 33 primary and 3 secondary studies for further

analysis. After their analysis, Dyba and Dingsoyr concluded that there was a need for additional empirical studies in Agile development in general, and that with the case of Scrum there was a large underrepresentation of study. They noted that only one of the 36 studies in their review spoke about Scrum compared to how popular Scrum's adoption is in the industry (Dyba & Dingsoyr, 2008). It should be noted that of the 36 studies the researchers focused on, only five of those touched somewhat on project management aspects of Agile and only one of those discussed the topic of requirements within an Agile process. The particular article discussing requirements (Dagnino, et. al., 2004) merely reported that an Agile process was able to accommodate changing and evolving requirements better than a traditional up front planning process. However, this study did not discuss in any detail the practice of performing requirements engineering within the Agile process.

The limitations of Dyba's and Dingsoyr's review and conclusions, which they freely admit, is their potential bias in narrowing their selection of studies for analysis as well as the search terms used in locating studies for review. However, by piloting the search and selection process as well as having a pre-defined method for evaluating quality for inclusion and exclusion of the final studies for analysis, the authors were able to reduce the amount of bias introduced into their study. Another limitation of the study was that the selection only included the years prior to and including 2005 which leaves a significant gap of years where the state of the research could have changed since their initial findings were reported. Given that the researchers themselves found a trend of empirical studies increasing steadily from 2001 until 2005 (Dyba & Dingsoyr, 2008), it can be reasonably assumed that there are a number of studies since 2005 that the researchers could not include in their evaluation.

While it is beyond the scope of this literature review to fill in the research gap from Dyba and Dingsoyr's 2005 systematic review, the most recent literature around requirements engineering within Agile will be evaluated. In one of those recent studies, Surendra (2007) used an ethnographic process called Strip Resolution Process (SRP) to help understand and guide the development of requirements. Surendra argued that because the software development process was less like traditional engineering disciplines and more focused on the interaction between people and their mutual understanding, an ethnographic approach can be appropriate. Surendra also provided evidence that an ethnographic approach has been suggested by previous researchers.

The Strip Resolution Process was chosen for study by Surendra because of its apparent ease of implementation so that developers would be able to apply this process without any extensive training. The implementation of this process was to first construct a basic understanding of the needs of the project stakeholder from the perspective of the developer in the form of a schema (Surendra, 2007). In the Strip Resolution Process, the schema is the researcher's attempt to articulate their understanding of the environment under study based on initial observations and interviews. The schema in Surendra's application of SRP to software development can be built upon interviews, discussions and documentation between the developer and the project stakeholders. The schema in SRP can then be equated to the software requirements for a development project in Surendra's comparison.

The next step in Surendra's application of SRP was to demonstrate the developer's understanding of the stakeholder's requirement via a working prototype to determine if the developer's understanding matches up with that of the project stakeholder. In the Strip Resolution Process, this is the Strip portion (Surendra, 2007). If there is a mismatch between

that of the developer prototype and what the stakeholder really wanted, then a breakdown occurs (Surendra, 2007). This breakdown is analyzed to determine how the mismatch occurred, and then a new schema is constructed along with another working prototype. This process continues iteratively until there is a resolution between the developer's understanding and the project stakeholder. The other key in the Strip Resolution Process is that the schema should hold up over a variety of different strip tests to ensure that it is a coherent solution (Surendra, 2007). Surendra likens this to demonstrating the prototype to a variety of different stakeholders throughout the software organization to ensure that the prototype works for everyone involved.

In Surendra's study, the Strip Resolution Process was applied to a single, small software development project in a case study organization, and the software prototype was successfully accepted after the fourth iteration of the SRP process. The study was perhaps important in illustrating that an ethnographic approach could be applied to the practice of requirements engineering to create a common language between the two disciplines. However, the value beyond that illustration was not quantified nor was it compared to current ways of performing requirements engineering. It is questionable from these findings whether or not learning and implementing a formal ethnographic approach would provide any additional value to teams over and above how they are currently performing requirements engineering.

Another case study dealing with requirements engineering within Agile was conducted around the role of physical artifacts (Sharp, et al., 2008). In this study, the authors explore two of the more common physical artifacts used by agile teams to capture and manage requirements: the story card and story wall. The authors explore the reason for these items and describe how they are physically used. The authors identify the two primary uses of these physical artifacts as being first notational in nature and second in providing a context to promote social interaction.

The researchers briefly describe what the story cards and story wall look like and how they are used by the teams that they studied. They observe that the notational conventions for these artifacts vary considerably between different teams, but the basic information and the use of these artifacts are for the most part the same. The researchers describe the story cards as a notational device while the wall is used as a visual system for processing the work. They go on to illustrate the role that the story cards play in capturing and prioritizing customer requirements and how the cards and the story wall are used in conjunction to process the day-to-day work done by the team. Taken together, the story card and wall are also used as a visual project dashboard to provide any passerby insight into the state of the project (Sharp, et al., 2008).

The authors devote the remaining bulk of their article to use a cognitive dimensions framework to describe the notational value of the artifacts and an ethnographic approach to analyze the social aspects of these artifacts. The authors believe that after analyzing these artifacts with these two complimentary approaches, they were able to show how the two artifacts complement each other appropriately with a mix of notational and social aspects to make teams successful (Sharp, et al., 2008).

Although the authors concentrate very lightly on the actual practice of using these physical artifacts in the practice of requirements engineering, they do present a compelling argument for practicing teams. Both the notational and social context of these artifacts, when taken together and in this physical form, are highly complex but their value cannot be understated for enabling the success of these teams. Therefore teams attempting to transfer these two artifacts into electronic form, absent these physical aspects, may not achieve the same success.

Pikkarainen, et al. (2008) provide another study that focuses some attention on Agile requirements engineering. Their case study was conducted at a software development organization looking at two different projects and the effect that Agile practices have on communication. Since the early Agile adopters believed that an increase in face-to-face communication is an important value enough to include as one of the twelve values in the Agile Manifesto (Agile Manifesto, 2001), the researchers wanted to study how using Agile methods impacted communication. Their primary assumption prior to collecting the data in their research was that using Agile methods "facilitates the transfer of knowledge and should beneficially affect the software development process which is based on communication" (Pikkarainen, et al., 2008, p. 309).

The researchers used coordination theory to map the dependencies between the particular roles in Agile software development and then used those mappings to evaluate the effectiveness of both internal and external communication (Pikkarainen, et al. 2008). Internal communication was defined as communication between developers and those roles directly supporting the development of the project. External communication was considered to be between the internal team and the project stakeholders driving the direction of the project but not directly working on the development. Throughout the study, the researchers mentioned the tools used to facilitate communication inside the internal team and between the internal team and external project stakeholders. Many of these tools, such as the product backlog, were defined as being used in the requirements management and engineering process. However, the researchers did not go into depth on how these tools are utilized except to describe how they facilitate either internal or external communication.

The researchers concluded that Agile practices and tools had positive effects on internal communication within the teams. Yet, there were risks uncovered in the case of external communication. Because there is more tacit knowledge and less documentation within Agile, there is a risk that external communication is reliant upon internal team members sharing that information with external project stakeholders. Furthermore, they found that when the number of external stakeholders on a project increased, the communication mechanisms of Agile (i.e., sprint planning meetings, daily Scrums) increasingly fell short in facilitating that communication. The key finding was that if teams are looking to Agile methodologies to increase communication, then care should be taken to only implement those aspects of Agile where teams will get the benefit and leave those plan-based approaches in place where appropriate (Pikkarainen, et al., 2008).

This finding by Pikkarainen, et al. certainly impacts the direction of requirements engineering, even though the actual practice of performing requirements engineering was not addressed. When the number of outside stakeholders or even remote team members is increased in a project, then the researchers are suggesting the use of more plan-based methods of managing requirements. Otherwise, a communication breakdown is likely to occur for external team members. Since this case study is isolated to one organization, it is yet to be seen whether these findings can be replicated. The researchers also suggest more study is needed in the area of communication and Agile in order to further validate their findings.

The final study to be discussed in this review was conducted in 2008 by Cao & Ramesh. In their study, they collected data from 16 organizations to evaluate and describe what practices Agile teams use to perform requirements engineering. Cao & Ramesh gathered data at these organizations by conducting interviews, through direct observation, and a review of documents

and artifacts. The researchers focused on identifying what common practices were performed across all of the organizations. They also outlined what challenges and benefits came out of using these practices from the point of view of the study participants.

Cao & Ramesh concluded that there were seven Agile requirements engineering practices that were found common across all of these organizations. These common practices were the heavy use of face-to-face communication, iterative requirements engineering, constant requirement prioritization, constant planning of requirement changes, the use of prototypes, test-driven development, and the use of review meetings and acceptance tests (Cao & Ramesh, 2008). Some of the organizations utilized the practices more or less compared to each other and not all organizations experienced the same challenges. It was noteworthy that all organizations studied listed the most common challenges as acquiring sufficient access to the customer representative for their projects, and coming to agreement on requirement decisions between different stakeholder groups. The most important of the requirement engineering practice listed by these organizations was the heavy use of face-to-face communication between the developers and customers (Cao & Ramesh, 2008).

Although the researchers did not specifically focus on the role and perspective of the requirements engineer or Business Analyst in their research, Cao & Ramesh are the only researchers that have yet to provide empirical study into the actual practice of requirements engineering within Agile development. Because Cao and Ramesh were able to collect and compare data from 16 different case study organizations, they go further in being able to generalize their findings to apply in other situations. However, considering the lack of other studies similar to Cao and Ramesh, there is certainly an opportunity to expand this body of research. It has been noted that “By performing multiple case studies and/or experiments and

recording the context variables of each case study, researchers can build up evidence through a family of experiments" (Layman, et al., 2005). It is therefore the goal of my research to build on that body of evidence through an additional qualitative study on requirements engineering within Agile development.

Chapter 3 – Methodology

It was noted that "if a concept or phenomenon needs to be understood because little research has been done on it, then it merits a qualitative approach (Creswell, 2009, p. 18)". The qualitative approach is described by Creswell (2009) as exploratory in its nature and is the appropriate design when the researcher is unclear about the variables that need to be studied. Since the research question for my study explored a topic where little study has been directed, as previously illustrated in the literature review, a qualitative study was chosen as the basis for my research methodology. The characteristics of how this study was conducted within this framework are discussed in the following sections.

3.1 Participants

The data compiled for this study was wholly collected through my direct participation as the researcher within the environment and situation under study. As a recently hired Business Analyst in an organization that had recently made the transition to Agile, I had the unique opportunity to interact, observe, and record the information I had gathered acting in a role of a direct participant within the process. As such, I was fully able to explore and develop a deep and intimate understanding of the role of a Business Analyst within an Agile centered organization. My background prior to becoming a Business Analyst was in Information Technology Support, Systems Administration, and then managing a staff of technologists responsible for maintaining the IT systems and support for an entire organization. While I had some experience in developing and supporting technology solutions, I had no prior experience or knowledge of what a Business Analyst was expected to do within Agile, nor did I have any prior experience as a Business Analyst in a traditional waterfall development organization. Although the Agile

development process was a concept that I had heard referenced in some of my previous work and studies, I had little to no knowledge about what the process specifically entailed.

When I began my role as a Business Analyst, my initial training and expectations that I developed came primarily from observing and then participating in projects already in progress and through my own research and reading about the Agile process. Since the transition to Agile was relatively new to the organization where I work, the Business Analyst role had evolved and changed over time, and continues to be in flux. I had to discover for myself what it meant to be a Business Analyst in Agile with a combination of no formal definition of my job role from my organization and a lack of any guidance from the research community. It therefore made sense to put the focus on exploring my experiences, recording my observations, and describing my approach and findings as an observer-as-participant in this process.

While other participants within my organization helped me to frame my understanding and influence my findings, I have primarily drawn from my own experiences, observations and actions. This information has come from working directly with my teams on the projects that I have been assigned and through the interaction with fellow Business Analysts and other employees.

3.2 Place

An overview of the organization in which this study has taken place was already presented in *1.2 Company Background.* This section describes the unique environmental, organizational and project characteristics of the organization. The important team member roles and interactions and the work setting within the organization where the study was conducted will also be described.

The product under development for my organization consists of a numerous set of features that have to work together to form the basis for the entire software platform. Over the years, the software platform has grown in features and complexity that no single developer or development team can become experts in every area under development. Furthermore, each feature can be in a different state of maturity as well as having a different priority for the organization. Therefore, the organization made the decision to organize teams around software features. Each team's focus could then be directed towards one or more feature sets and thus these teams could become the experts and owners of those particular features. Teams could be scaled up or down appropriately to support larger or more important features and new teams could be created or organized if new features needed to be developed into the platform.

The feature-based organization of teams also fits better into the Agile development methodology. Each team is organized around a much more manageable sized project and product vision. In practical terms, this means that short development cycles could be constructed and easily tracked within teams, and changes in direction or priority could be easily worked into the next development cycle. This also translates into a team size that is small enough to meet together in daily stand-up meetings, and the entire team is physically located together for increased face-to-face interaction and collaboration. Working within a limited scope of responsibility, teams are allowed to develop a high level of expertise in their areas by maintaining a narrow yet deep focus on their particular feature areas. Efficiencies could also be gained by having teams focus in one feature area for an extended period of time over several releases. The amount of time spent task switching is reduced and the tacit knowledge built up over many months working within the particular feature code base is not wasted by moving developers and team members from one area of the system to another.

In order to support feature-based teams, the product development organizational structure is also divided into various cross-sections of features which a Product Manager (PDM) is responsible for overseeing. Each team's feature development work is provided by this Product Manager role who is assigned to drive the vision and direction for each of their features across multiple releases. From the bottom up, the product development hierarchy then consists of a number of Product Managers whose feature teams work independently of each other but come together at the Product Director level to ensure that all features converge to form the entirety of the company's product vision.

From the top down, the VP of Product works with each Product Director to develop and maintain the company's product roadmap. This longer term product vision for the entire company is then disseminated to the individual Product Managers who can then intimately understand and maintain their feature release roadmaps. Since all product teams work in short development cycles using Agile, the entire organization can quickly move the product direction, or just certain sections of product features can be adjusted to changing market or customer demands. This feature team structure has allowed for the top down decomposition and prioritization of the entire company's product vision to be filtered and managed at the individual small team level. Much like an Agile project, the entire company has been setup to quickly respond to change while still being able to maintain a larger and longer-term product vision.

The Business Analyst role works with the Project Manager (PM) on each team to ensure that the release and product vision is executed at the day-to-day level and over the iteration and release cycles. The BA works closely with the Product Manager (PDM) to understand their vision for the product and they interact closely over the course of the iteration cycles to make sure that the most important features for the product are implemented first. This prioritization is

re-examined at the planning stages for the next iteration to ensure that when the team starts a new iteration of work, the most important items are again addressed in the proper order. Decomposing the Product Manager's high level feature requests into understandable and manageable chunks of work for the team to be able to complete in a two week iteration cycle is also the responsibility of the BA.

The remainder of the feature team is made up of a number of developers and quality assurance engineers who are fully allocated to a single team. Unlike these fully allocated resources, the Business Analyst is assigned to at least two or more teams depending on the size and complexity of the projects. The team also has shared resources that span across multiple projects that provide input and work to the team when required such as User Experience Engineers, Product Architects, Technical Writers, and Database Administrators. These resources are not required to provide work during every iteration and therefore they are shared across a number of teams.

The work setting for development projects consists of an area of cubes where as much as possible the entire team is seated together so that face-to-face interaction and spontaneous meetings and other communications can take place easily. Since many of the shared resources, such as the BA, cannot always sit close to every team they work with some team members are not as conveniently located to those they work with as other teams. However, in all cases the fully allocated developers and quality assurance engineers sit together in the same area. The entire organization consists of nearly 500 U.S. employees with Sales, Human Resources, Marketing, Client Services, Information Technology, Technical Support, Software Engineering, Product Management, and Program Management departments.

The organization also has an off-shore development operation that is similarly structured around feature teams and is about 60 employees in size. This team's location is within a time zone that is 12 hours different than the U.S. organization. One other distinction for this off-shore operation is the development and quality assurance roles for these teams are located off-shore, while the business roles (BA, and Product Manager) and primary Project Manager role is located in the U.S. The off-shore teams also have Project Managers on their end that provide the necessary organizational structure to run the teams during their time zone so that communication and hand off can occur between the off-shore Project Manager and the U.S. Project Manager. If the remote team needs information for follow-up from the U.S. side of the team, it is these two Project Manager roles that are responsible for making sure the flow of information does not impede the progress of the team.

The project team that was observed during this study consisted of six Developers, four Quality Assurance Engineers, and one Project Manager which were fully allocated to this project. In addition, the team consisted of a Product Manager and I was the Business Analyst. Both I and the Product Manager were not fully allocated to this project team, but this was our primary project team allocation. This team also shared a User Experience Engineer, Product Architect, and a Database Administrator as needed. As described above, this team was one of many responsible for ownership and maintenance of a subset of our organization's software product features. The feature teams in the organization range in size depending on the amount and complexity of the features they are responsible for maintaining. This particular team was responsible for two major feature sets and was a bit larger than most other teams.

3.3 Instruments and Materials

This section describes the tools and materials that were used in the study for data collection. The primary tool for the collection and compilation of data was Microsoft OneNote which is an electronic journaling program. Notes and observations can be recorded very quickly using this program and then images, documents, emails or any other form of electronic data relating to that note could be attached and then associated.

The bulk of the material collected was my meeting notes and project observations which were recorded chronologically and organized by project into sections using Microsoft OneNote. Email communications between me and my team members were also collected. Finally, some work products produced during the course of the data collection period including documents and images throughout the course of my projects were compiled. Therefore, all the material for this project was collected with and compiled into a single electronic storage location for analysis and archiving.

3.4 Procedure

The data collected for this study consisted of an intense period of observational and electronic artifact collection from two iterations of Agile development work. Each of the two iterations lasted two weeks for this team and spanned between March 1st – March 26th, 2010. I also collected data during the week leading up to the start of this iteration period and the week continuing after the end of the second iteration. Based upon my year of experience participating in Agile projects, I understood that much of the work conducted by a Business Analyst takes place outside of the actual development periods. Therefore, I wanted to make sure that those interactions were also included for this review. Even though as a Business Analysts I am a

shared resource on multiple teams, observations and artifacts for those additional teams were not included for this analysis. However, I did record all interactions I had outside of my study team's work but where the observation directly applied to my role as a Business Analyst. These types of interactions included discussions with my entire Business Analyst team, the coaching of other Business Analysts in the tools or process I used for my projects, or where the interaction or observation cut across every project I am a part of in some meaningful way.

The team that received the focus of my observation was the team where I was physically located within my immediate workspace. Because of the close proximity to my other team members, I was able to naturally interact with these team members and observe and participate in a greater amount of interactions. This team's close proximity and formal Agile process ceremony (i.e., Daily Scrum, Iteration Planning, Retrospective, etc…) is as close to what Schwaber (2004) refers to when describing a healthy, co-located Scrum project team.

Any interaction where I was involved directly with members of my team or interactions with external stakeholders which affected the work of my team were recorded as an individual page within a daily electronic log. Whenever possible, each interaction was recorded in the form of descriptive notes as they were occurring so that I could record as much detail immediately rather than fill in information after it has perhaps degraded from my memory. The direct recording of data was not always possible given the nature of work disruptions and because of time constraints. However, at the end of each work day, I went back and added any missing information around those interactions to ensure that I had a complete accounting of everything I participated in or worked on within my project.

Within each page of the electronic journal, I included a short title and date stamp along with a brief description of what the interaction entailed. This was followed by a series of

descriptive notes about the interaction which often included background information which could help put the information into context. When the interaction included associated artifacts such as a follow-up email conversation or other work products I produced or was a part of, these items were included as sub-pages within the electronic journal. If my interaction occurred wholly over email, those emails were attached as separate pages within the journal.

As I was concerned with exploring the role of the Business Analyst during this process, my observations of my work and interactions were focused from that point of view. However, my descriptions would not be complete or provide enough context without mentioning or describing the other roles as I was interacting with my team. While I was recording descriptions, these other roles were only identified generically with a role name, and I took care not to record or include any personally identifiable information around any of my team members. The same care was taken when I attached work products or included email interactions with other employees. In these cases, I replaced the name or email address with the generic role name of that person. For example, I used Product Manager when referring to the Product Manager on my team and used that same type of label if that person was included in email records.

3.5 Data Analysis

The data analysis began with a read through of all the individual sections and chronological pages of observations and artifacts that were collected. Since my research objective was to define the role of the Business Analyst within an Agile development process, I analyzed and coded the data from that perspective. Therefore, the areas where I organized and grouped my data were designed with my research question in mind. As each page was read, I created a “tag” within the Microsoft OneNote program that defined my first impression of how

that thought or passage could be classified into a topic area. OneNote comes with a number of default tags I could have used, but they were limited and I had the option of creating and defining my own custom tags. These tags in OneNote were custom created with a symbol, title, and a text color or text highlight. The attributes except for title were optional, but I found that the symbols next to the passage and the unique text color color made it easier for me to glance at my notes and see what I had completed and what was still to be coded.

Once I created one of these custom tags, I then highlighted the particular passage of text and applied the tag to that information using OneNote. As I was reading through each page and expanding my list of tags, I also created a new section within OneNote with a table. This section and table kept track of what tags I created with the description for the tag, the title I had defined, the symbol I had used, and the text color that I had chosen. I made changes and additions to this table as I continued to define and refine my groupings. In addition to placing a symbol next to the text and applying the text color to my data, this process automatically created an association with any data that shared the same tag. A report produced within OneNote could show me at any time all of the passages across my notes that were defined with each tag. This process continued iteratively until I had several broad groups of tags applied to all of my information.

Once I had gone through all of my data and applied tags to the information, I was able to create a report page in OneNote that outlined a grouping of the data by each tag. I then analyzed the data contained in these groupings to see if any further refinement was necessary. I quickly found that a lot of my tags were overlapping. I had some passages where it was difficult for me not to apply two tags to the information. Therefore, I continued to make refinements to my tagging until the data was able to fit better within a broader set of groupings. I continued to

iterate through this process of tag and data refinement until I had no remaining overlapping passages.

This refinement process produced several sub groupings of distinct categories under a few of my major theme areas. I allowed these sub groupings to develop and tagged them separately but included them for analysis and discussion within the broader area of the main theme. I also included a tag for background information. I found that in my observations I was providing additional contextual data that did not get grouped into a theme or category. Even though this information did not feed directly into any of my findings tied to the Business Analyst role, this information was later used to help construct the detailed narrative around the project, roles, and work setting which I have used throughout this paper.

To address the reliability and validity of my study, I employed several approaches. To maintain reliability, all of my notes, work products and other forms of data were directly recorded in a single location for analysis. There was no need to transcribe from written notes to electronic format, and therefore no risk in making mistakes moving from one form of media for data collection to another form for analysis. After each day of data gathering, the observations and other work products were reviewed thoroughly to make sure there were no mistakes in recording, major omissions, or any gaps in recorded activities.

To validate my findings and lend credibility to my themes as they developed, I used data triangulation. This approach is discussed by Yin (2003) and Creswell (2009) as one important form of ensuring qualitative study validation. This approach brings together different data sources which help the researcher point to the same finding. While I was coding and reviewing my data, I created a tag that identified when I had two or more separate sources of information coming together to demonstrate whether I had properly triangulated my findings. For example, I

had email, work products, and images along with my observational data that made triangulation possible.

A peer review of my themes and findings was another form of validation that was used. This approach is described by Creswell (2009) as using an external peer to assist the researcher in validating the accuracy and the accounting of the information put forth by the researcher. This peer review exercise both helped to validate my findings and enforced the reliability of what I had presented. I was able to execute this approach by discussing and presenting my findings with a senior Business Analyst working outside of my projects but within my organization. As my data came mainly from the perspective of how I was conducting Requirements Engineering, this peer review from another Business Analyst helped me to confirm that what I was finding was not isolated to just my projects or my perspective.

The final method I used in my study for validity was to describe in detail the setting, actors and situations I encountered while conducting my study. I used that information while presenting the discussion of my findings and have used as much of this rich detail throughout other sections of this paper whenever possible. This method of validity is listed by Creswell (2009) as one of several strategies a researcher can use for study validity.

Chapter 4 – Results

My original research question was: What is the role of the Business Analyst within an Agile development process? The results outlined in this chapter directly address this question by presenting the five major Business Analyst role themes I found through the data analysis exercise. Whenever a distinct sub grouping was discovered under any of these major themes, those sub groupings were also described. In order to provide some context for objective evaluation of these themes, I included a percentage of occurrence calculation for each theme. This percentage calculation is based upon the number of distinct observed occurrences of the theme divided by the total number of all unique observations that were coded in this study. Note that when a major theme contained sub groupings, the percentage of occurrence calculation for the major theme also included the sub grouping rolled up into the total for the major theme. The major themes found were: Communication, Agile Process, Analysis, Prioritization, and Dependency Management.

As illustrated in "*Table1:* Business Analyst Role Themes", the BA role theme labeled Communication, along with the four distinct sub groups, made up the highest percentage of observations recorded and analyzed during this study at 32%. The Communication role theme was coded from any interaction where I was responsible for communicating the team's current or future project work. The unique sub groupings under Communication included: External, Status, Team, and Planning. The sub group labeled External was coded whenever I met with any external stakeholders to convey information around what the team was currently working. The Status sub group under Communication was observed whenever I communicated directly with other team members in order to retrieve status information on their current tasks or project work. Whenever there was an instance where I was engaged with some members or the entire team in

formal, scheduled discussions regarding the team's current or future project work, the Team sub group item was used. Finally, any instance where I conducted planning discussions with members of the team, the Planning sub group under Communication was coded.

The next highest occurring Business Analyst role theme that surfaced in the data analysis was labeled Agile Process and constituted 31% of the total observations for the study. This theme, which included two additional sub groupings, was defined as any information found which related to the practice, coaching, or refinement of the Agile process by the Business Analyst. This theme included any interactions which took place inside or outside of the team. This BA role theme surfaced with two sub groups of information which included Coaching and Tools. Coaching was distinct because these interactions involved assisting other teams in their practice of Agile. The Tools sub group contained items when I was directly involved in constructing or using tools to manage the Agile process.

The third BA role theme that was uncovered during data analysis and coding was Analysis. The Analysis role theme represented 21% of the total observations in the study and was coded whenever I was asked to perform requirements analysis or a presentation of that analysis. This theme also included any requests directly to me to provide clarifications for how our software should behave in given situations. The sub group under this theme was Testing. This included any instances where I assisted with testing current functionality or with reproducing issues found in the software.

The fourth BA role theme, Prioritization, represented 10% of the total interactions coded. The Prioritization theme was observed whenever I directly intervened (questioned) or was asked to assess the team's prioritization of work, or any cases where I assisted with or directly guided the prioritization of work for our team. Only one sub group developed under this theme and that

was labeled Reviewing and Accepting. A form of affecting the team's time and therefore indirectly guiding priority, Reviewing and Accepting was observed and coded whenever I was the primary role responsible for reviewing the work our team had done and accepting it as complete.

The final BA role theme that was discovered was labeled Dependency Management. This theme constituted 6% of the observed data. This theme was a unique role area defined whenever I was responsible for the management of cross team work dependencies. Dependencies are defined whenever one feature team is dependent upon another team's feature code in order to meet their commitments.

The presentation of the five major themes, the sub groupings, and the percentage of occurrence calculation is shown in "*Table 1:* Business Analyst Role Themes*"*.

Table 1: Business Analyst Role Themes

<table>
<tr><th>Theme</th><th>Description</th><th>Percentage of Occurrence</th></tr>
<tr><td>Communication</td><td>Communication includes any general project related interactions where the discussions are initiated or led by the BA.

<u>Distinct Sub Groupings within Communication:</u>

<table><tr><th>Sub Grouping Name</th><th>Sub Grouping Description</th></tr><tr><td>External</td><td>The communication that takes place outside of the team between the BA and external stakeholders.</td></tr><tr><td>Status</td><td>Communication with team members around task or work item status.</td></tr><tr><td>Team</td><td>Any interactions where the full team or partial team is engaged in a scheduled discussion led by the BA.</td></tr><tr><td>Planning</td><td>Meetings or conversations with portions of the team with the specific purpose of planning or preparing for full team meetings and discussions.</td></tr></table></td><td>32 %</td></tr>
<tr><td>Agile Process</td><td>Information relating to the practice, refinement, or coaching of the Agile process by the BA within or outside of the team.

<u>Sub Groupings within Agile Process:</u>

<table><tr><th>Sub Grouping Name</th><th>Sub Grouping Description</th></tr><tr><td>Coaching</td><td>Agile process or tools coaching from the BA to other teams.</td></tr><tr><td>Tools</td><td>The use or development of tools by the BA to assist in the management of an Agile development project.</td></tr></table></td><td>31 %</td></tr>
</table>

<table>
<tr>
<td>Analysis</td>
<td>Information relating to analysis of the product or requirements, presenting the analysis results, or providing clarification on how things should work or how they currently work in the product.

<u>Distinct Sub Groupings within Analysis:</u>

<table>
<tr><th>Sub Grouping Name</th><th>Sub Grouping Description</th></tr>
<tr><td>Testing</td><td>Testing out product functionality or assisting with the reproduction and confirmation of issues.</td></tr>
</table>
</td>
<td>21 %</td>
</tr>
<tr>
<td>Prioritization</td>
<td>Information related to assessing, assisting with, or directly guiding prioritization of work coming into our team or deflecting work from reaching our team.

<u>Distinct Sub Grouping within Prioritization:</u>

<table>
<tr><th>Sub Grouping Name</th><th>Sub Grouping Description</th></tr>
<tr><td>Reviewing and Accepting</td><td>Reviewing the work the team has produced and accepting it as complete. This also includes making go / no go decisions around release decisions.</td></tr>
</table>
</td>
<td>10 %</td>
</tr>
<tr>
<td>Dependency Management</td>
<td>Information or activities around or the process of managing feature dependencies across teams.</td>
<td>6 %</td>
</tr>
</table>

Chapter 5 – Discussion

To summarize my results, the five major Business Analyst role areas identified were: Communication (32%), Agile Process (31%), Analysis (21%), Prioritization (10%), and Dependency Management (6%). In order to illustrate how these five themes come together to fully describe the role of a Business Analyst in Agile, I have provided a detailed discussion of each theme in the sections below.

5.1 Communication

In my organization's Agile development process, I found that the Business Analyst role acted primarily as a communication and information broker for the project team. As was illustrated in the four different sub groupings under this theme, much of the work that I performed for my project was directed toward communicating with external stakeholders and the project team in order to gather and share general project information or status. Working with other roles such as the Product Manager (PDM) and Project Manager (PM), I acted to balance the amount and quality of information flowing into and out of the team appropriately. I found that achieving this balance was essential to allow the product builders (Developers and QA) on the team to sufficiently focus on the current iteration of work without unnecessary distractions. At the same time the team received enough information to know what to build and how to build it.

Planning discussions were also sub grouped into this theme. As the BA, I organized sub sets of the team in order to plan the approach for when and what was appropriate to communicate to the whole team. Planning in this context was not around analyzing the type of work the team would be doing, but rather organizing and focusing the information presented to

the full team so that it was concise and appropriately relevant. With two week iterations Agile development is fast paced at our organization. While work was performed during the two weeks by our product developers, the BA, Product Manager, and Project Manager constantly communicated with each other to ensure that the next two weeks of work was prepared to pass to the development team by the end of the current iteration. Planning how, when, and what we would communicate to the full team was not intended to keep important information or decisions from the other team members. On the contrary, it was a targeted approach designed to give the team only the most important information they needed with a goal to reduce the number of interruptions for our development resources. Putting in the time to plan in this way was intended to make the full team interactions focused and succinct.

Two related but distinctive major themes from Communication were Prioritization and Analysis. These two themes will be discussed at length in following sections. However, it is important to note in this Communication theme that the prioritization and analysis work performed by the BA often informed the general communication passed on (or not passed on) to the team or external stakeholders. For example, it was often important to fully analyze and correctly prioritize the work before communicating that work to the full team or passing on specific work to a developer. On more than one occasion, the analysis performed by me was enough information for the Product Manager to decide not to have our resources take on particular work items.

In a study referenced in my literature review, Cao & Ramesh (2008) found a similar communication theme in practice at Agile organizations. They describe the importance of face-to-face communication between the customer and the team over the use of written documentation for requirements engineering. My Communication theme described here is both

more broadly defined to include more than face-to-face communication and is specifically targeted to the role of the BA in brokering that communication. However, I believe my observations generally support the findings by Cao & Ramesh (2008) that the extensive use of communication over documentation is a common practice in Agile development organizations. Where my findings can add value above the research conducted by Cao & Ramesh is in providing a specific perspective on how that communication is managed by a Business Analyst.

5.2 Agile Process

In my Agile development project, I often played the role of an Agile process champion for my project team and coach or advisor to other teams. Whenever there were questions about the use of our Scrum process, or the tools we were using to manage our development process, these questions came to me to advise or guide the team. I also found myself interjecting my suggestions for improvement or process direction even when unprompted but where I felt the team needed guidance. These suggestions were often well received. The two passages below came from the Senior Director of Software Engineering when I suggested an approach to conducting planning meetings before iteration planning.

> "Awesome. I like the plan, Nate. I will help however I can."
> "Very cool. Looking forward to some very productive pre- and actual- iteration planning meetings".

One sub group under the Agile Process theme surfaced around the use and construction of Tools that assisted in managing our team's Scrum Agile process. This sub group was distinctive from the major Agile Process theme because the tools helped manage the development process, but the Agile process was the umbrella under which the tools were given context and meaning. One such tool that I used extensively with the team was a physical task

board built on a white board. The board was by constructed with columns and descriptions where the team's work was visually displayed as individual items of work. As the team members pulled work from one side of the board starting on the left as "unassigned", the progress of that item was tracked across the board as the work moved through various stages including: assigned, dev complete, ready for QA, and finally, ready for release.

I found that utilizing the task board served the team on various levels. From outside our team, any product stakeholder could walk by our area and see how we were tracking towards our release goal and iteration commitments. One concrete example of this in use was with our Production Release Manager. The Production Release Manager, who was responsible for managing our entire organization's production release process and who was not on our team, was often observed standing at our task board to see how we were tracking for our release. From the team's perspective, we utilized the task board every day during our daily Scrum to have each member of the team speak to their assigned work. Each person provided their updated information to the full team on their progress, any roadblocks or issues as we worked through the iterations. To illustrate this task board, see "*Figure 1*: Agile Task Board" below which was a photograph taken during one of our iterations using this tool.

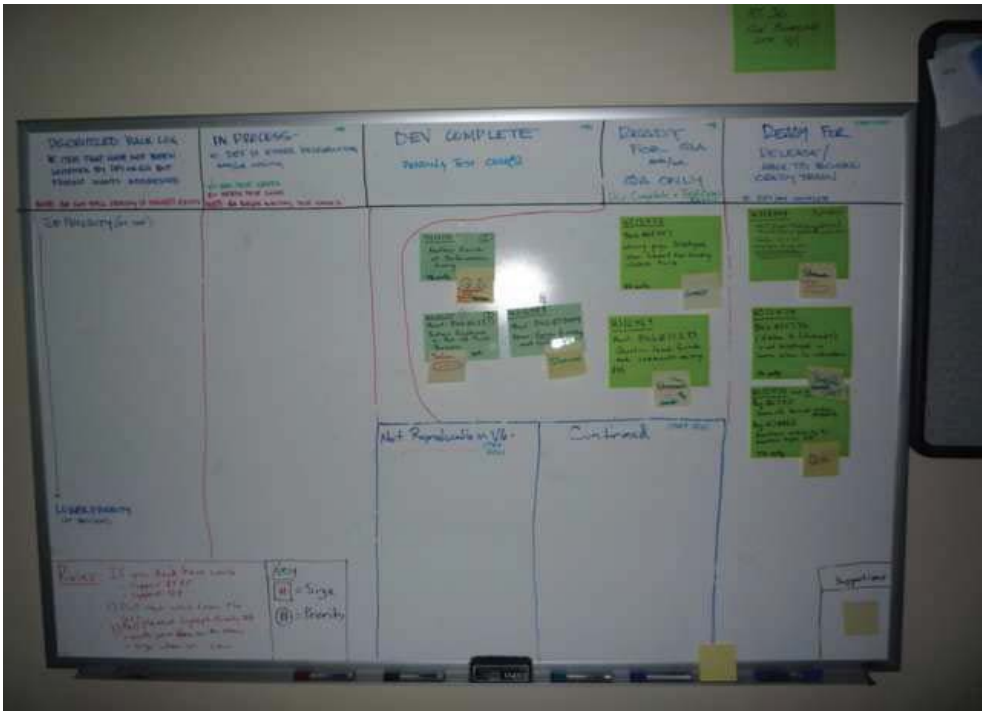


Figure 2: Agile Task Board

As I referenced in my literature review, the use of tools on Agile teams has been presented by Sharp, et al., (2008) in their study on physical artifacts. What I have described in the previous passage and illustrated with the above photo align with the study findings. What Sharp, et al., called the story wall and I what I have called the task board was used both for notational and social purposes for the team (Sharp, et al., 2008). Further, the authors described the story wall as a visual dashboard used to inform any passerby the status of the team's project work (Sharp, et al., 2008). I also found this to be the case as my example above with the Production Release Manager using our task board to get our team's release readiness status by using only the information on the wall. What I cannot infer from the study by Sharp, et al. is what role the Business Analyst played, if any, in constructing or managing the physical artifacts they studied. I can only report that in my observations, I played the primary role in constructing and managing our task board throughout the project.

As I conclude the interpretation of the Agile Process theme, which I have described as Agile process champion or coach, it was somewhat surprising to me that this theme showed up second only to Communication in the number of observations I recorded. If I look to the review of literature, there was no guidance around this role being applied to the Business Analyst. However, I do understand from my research within the general Agile literature that the role I found myself acting in for this theme was closer to what Schwaber refers to as the Scrum Master (Schwaber, 2004). As described in the introduction chapter under section *1.1 Agile Methodologies*, Schwaber (2004) described the Scrum Master as more of a process coach and project facilitator. Schwaber points to this role traditionally being filled by the Project Manager (Schwaber, 2004). On my project, we did have the Project Manager role which I observed performing a mix of the traditional project management tasks (project metrics, status reports, etc.) and some of those tasks described by Schwaber including leading the daily Scrum, and facilitating the team's work (Schwaber, 2004).

While I do not want to completely undermine the importance of this theme surfacing in my research as one major focus for the BA role, I do postulate that this theme is very specific to my given situation and background. Agile methodologies were a primary focus of my research and interest for the past year. By the time I started collecting data for this study, I had already been researching Agile methodologies for some time. I also had gained significant experience as a BA on several Agile projects using our organization's implementation of Scrum over the past year. If not for my specific interest, research and experience, it is unclear if the Agile process champion theme would have surfaced as such a large part of my role.

5.3 Analysis

During the course of my Agile development project, I was asked or I volunteered on many occasions to provide analysis into a product requirement or clarification around expected product feature functionality. This analysis was always followed up with a communication of my findings. The results of this analysis were most often used to drive product decisions that directly affected the work coming into my team. As work was being queued for the team into our next iteration, I worked closely and often with our Product Manager to not only understand the work being asked of us as feature requirements but also the importance of this work to our clients or to our product vision. If anything was unclear, further analysis or discussion was conducted. This allowed me to wholly represent that issue or item of work to the rest of the team. More importantly, I was then able to understand the relative importance of each item of work in relation to each other. Based upon a complete analysis of all the items of work proposed for our team to work on, I felt comfortable and often did challenge our work prioritization.

One distinct sub group that I identified in my coding under Analysis was Testing. Testing was distinct from general Analysis in that when I was asked to test, I was being tasked with trying to walk through a specific scenario in order to reproduce a particular issue. While I believe that testing was a form of Analysis, it was not the same as looking at a product requirement or issue and deciding a direction based upon business rules or customer requirements. Testing is a very structured activity where the outcome is either confirmed or unconfirmed. Testing fed into analysis on more than one occasion in my observations when after an issue was discovered, I had to help our team and Product representative decide what the expected behavior should be based upon my knowledge of the product.

Although the Analysis theme is probably most expected item to emerge for the role of the Business Analyst, the literature review supports this finding in the context of Agile requirements engineering. It is described by Cao & Ramesh, (2008) as being a process that occurs iteratively during the course of the development process. As I have described, I observed the Analysis theme occurring often with the input of new potential work items and requests from the Product Manager. As I've mentioned in other themes, the research does not specifically describe the role that the Business Analyst plays under this theme. However, one can infer that Analysis would be a major role for the BA whether or not an Agile process is in place.

5.4 Prioritization

Prioritization was the fourth most common theme. Specifically, I found that often I was not only assisting with but I was also directly guiding the prioritization of work for our team. This sometimes meant that work which was immediately scheduled for our team to work on was deferred or abandoned. In some cases, this meant that the order in which our team was to complete the work was adjusted based on my suggestions. Although the Product Manager or Client Services role always has the final say on what our team worked on, I had several interactions where my suggestions were taken as the final prioritization decision.

In one such interaction, I had analyzed a series of issues put in initial priority order by the Client Services Manager based on a short description of each issue. These items were going to be resourced by our team to work on in order to release a new version of our product. After performing analysis on each of these issues, I presented my findings to the Product Manager and the Client Services manager along with a clear direction of what I thought the priority order should entail. The interaction around this example spanned over several meetings and emails,

but the result was an email message sent by the Client Services Manager which read “Thanks again Nate for putting this together. I greatly appreciate your help on this and prioritization! Please let me know if there is anything else you need from me.”

I discovered and labeled one sub grouping under this Prioritization theme as Reviewing and Accepting. This sub grouping was placed under the Prioritization theme because performing work under this sub group meant that I was indirectly affecting the prioritization of the team’s work time. During the course of the project work, this sub group was observed in two different ways. As was the case on a couple of different occasions, absent the Product Manager or another business role, I had to make the call whether or not the team member had completed the work assigned and could therefore move onto another item of work.

The other way this sub grouping was observed was on our release day. In this particular situation, it was late in the day and we had to have our work completed in order to hand our code over to the release team in time to release as scheduled. There was one small issue found in testing around one of our work items that could not be addressed by a developer and then tested in time for release. I met with the Lead QA Engineer and the Developer still supporting our release readiness, but the conversation ended with me making the call to not hold the release for this small item. The Lead QA Engineer later sent the following to sum up our discussion: “Per Nate, this scenario is not worth holding the train for so we’re not going to wait for dev to look at this and get a fix, especially since it’s so late in the afternoon of our boarding day”.

As I found in my research and through discussing the themes, Prioritization and Analysis conducted by the BA role were often closely related. Without my analysis or experience working in my project area for over a year, I would not have been in a position to guide priority.

Unless I already understood the problem or work item under review because of prior experience, I always had to conduct some amount of analysis in order to guide priority.

There was no specific mention of how the Business Analyst role directs priority within the research literature. However, the way in which prioritization was conducted within Agile was mentioned as one of the themes Cao & Ramesh (2008) found across their study of 16 Agile development companies. Cao & Ramesh described that all participants in the study reported that prioritization is done a number of different times during the development process instead of a single time up front (Cao & Ramesh, 2008). I also found this to be true during the course of my project and data analysis. While Cao & Ramesh point to the same type of prioritization that I found being performed on my team, there was no mention of what actors in that process managed or guided the prioritization.

5.5 Dependency Management

The final Business Analyst role theme to discuss is Dependency Management. As described in "Chapter 3 – Methodology" section *3.2 Place,* the organization where I work and conducted this study had a feature team model where small teams are responsible for developing and maintaining their feature code. This team structure inevitably led to one team needing changes done or additional features from another team within their feature code in order to meet a commitment. I found that the identification and management of these feature dependencies was one of the responsibilities that fell to the Business Analyst role. This was a natural fit for the BA role as our BA team met regularly to discuss our projects. Because of the intimate knowledge we all had of our own product backlogs and features, we were in a position to identify any dependencies we would need from other teams (or required by our team from other

teams). During the course of the data gathering for this project, our BA group conducted a dependency mapping exercise. This exercise was used to discuss or update each other on dependencies we knew we had on each other's team. While this exercise cut across every team I worked on, the current project team under study was also discussed.

I found the idea around Dependency Management for an Agile development organization was not unique to our company. In a study by Babinet & Ramanathan (2008), they outlined the process and steps that a large scale Agile development organization goes through to manage dependencies. During the dependency identification exercises described by Babinet & Ramanathan, they report that the Scrum Master or Product Owner from each team got together in a room to discuss and identify dependencies (Babinet& Ramanathan, 2008). However, they do leave it open that the role can be anyone from each team as long as each team is represented (Babinet& Ramanathan, 2008).

5.7 Conclusion

The role of the Business Analyst within the Agile development process in my organization was not defined by a structured set of assigned responsibilities and clear boundaries. Through the process of observation and analysis, I have determined that five distinct yet related themes surfaced during the course of an Agile development project. The role I played as a Business Analyst for my team contained a mix of all the themes: Communication, Agile Process, Analysis, Prioritization, and Dependency Management. While the Agile literature and research points in some way or another to my themes as expected within an Agile development process, the research fails to address how those themes apply to the Business Analyst role.

Where I have made a contribution to the growing field of Agile research was presenting one perspective from the point of view of a practicing Business Analyst.

However, I do understand that my work has some important limitations. One of the larger limitations of my study is that it was conducted with a narrow and specific case and it cannot be generalized to other groups. Another limitation is the short time period in which I conducted the data gathering. Although the amount of observations I gathered were extensive, it would have been interesting to see whether or not other themes developed over a more extended period of observation. Another limitation is that my study was presented from the point of view of one observer acting within the role which was under study. Even though I went through great lengths to ensure reliability and validity in my findings, my bias and what I thought was important to observe certainly helped to shape the study and the results.

For further research, it would be beneficial to study the themes I have identified and apply them in a survey or large case study across a broad range of Business Analysts in other Agile development organizations. Findings from studies such as these could determine if the themes I have identified are common or transferable to other BA settings. Further qualitative studies observing the activities of practicing Business Analysts in other Agile organizations would also be important. These types of studies could either validate my themes existing in other settings, or also uncover additional themes for further exploration.

References

Agile Manifesto, (2001). Manifesto for Agile Software Development. Retrieved July 9, 2009, from http://www.agilemanifesto.org

Babinet, E., & Ramanathan, R. (2008). Dependency Management in a Large Agile Environment. *Proceedings of the Agile 2008 Conference,* Toronto, Canada. 401-406.

Beck, K., & Andres, C. (2004). EXtreme Programming Explained: Embrace Change. San Francisco, CA: Addison-Wesley.

Bhat, J. M., Gupta, M., & Murthy, S. N. (2006). Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing. *IEEE Software*, 23(5), 38-44.

Cao, L. & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, 25(1), 60-67.

Ceschi, M., Sillitti, A., Succi, G., & Panfilis, S. (2005). Project Management in Plan-Based and Agile Companies. *IEEE Software*, 22(3), 21-27.

Cockburn, A. (2007). Agile Software Development: The Cooperative Game (2nd ed.). Boston, MA: Addison-Wesley.

Cockburn, A. (2005). Crystal Clear: A Human-Powered Methodology for Small Teams. Boston, MA: Addison-Wesley.

Creswell, J. W. (2009). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (3rd ed.). Thousand Oaks, CA: Sage Publications.

Dagnino, A., Smiley, K., Srikanth, H., Anton, A., & Williams, L. (2004). Experiences in Applying Agile Software Development Practices in New Product Development. Retrieved July 20, 2009, from http://collaboration.csc.ncsu.edu/laurie/Papers/IASTED.pdf

Drobka, J., Noftz, D., & Raghu, R. (2004) Piloting XP on Four Mission-Critical Projects. *IEEE Software*, 21(6), 70-75.

Dyba, T., & Dingsoyr, T. (2008). Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, 50 833-859.

Hanssen, G. K., & Faegri T. E. (2006). Agile Customer Engagement: A Longitudinal Qualitative Case Study. *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Engineering*, Rio de Janeiro, Brazil, 164-173. Retrieved July 22, 2009 from http://www.idi.ntnu.no/grupper/su/publ/geirkjetil/isese06-agile.pdf

International Institute of Business Analysis (IIBA). (2009). A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 2.0. Toronto, Ontario, Canada: Author.

Layman, L., Williams, L., & Cunningham, L. (2005). Motivations and Measurements in an Agile Case Study. Foundations of Software Engineering. New York: NY: ACM

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al. (2004). Agile Software Development in Large Organizations, *Computer*, 37(12), 26-34.

Nguyen, L., & Shanks, G. (2009). A Framework for Understanding Creativity in Requirements Engineering. *Information and Software Technology*, 51, 655-662.

Nottonson, K., & DeLong K. (2008). Baby Steps: Agile Transformation at BabyCenter.com. *IT Professional*, 10(5), 59-62.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The Impact of Agile Practices on Communication in Software Development. *Empirical Software Engineering*, 13(3), 303-337.

Schatz, B., & Abdelshafi, I. (2005). Primavera Gets Agile: A Successful Transition to Agile Development. *IEEE Software*. 22(3), 36-42.

Schreiner, K. (2007). The Bridge and Beyond: Business Analysis Extends Its Role and Reach. *IT Professional*, 9(6), 50-54.

Schwaber, K. (2004). Agile Project Management with Scrum. Redmond, WA: Microsoft Press.

Sharp, H., Robinson, H., & Petre, M. (2008). The Role of Physical Artefacts in Agile Software Development: Two Complementary Perspectives. *Interacting with Computers*, 21, 108-116.

Surendra, N. C. (2008). Using an Ethnographic Process to Conduct Requirements Analysis for Agile Systems Development. *Information Technology and Management*, 9(1), 55-69.

The Economist Newspapers Ltd. (2004). Managing Complexity – Software Development; The Software-development Industry. *The Economist*, 373, 71. Retrieved July 9, 2009, from Gale Academic OneFile database (A12537892).

Yin, R. K. (2003). Case Study Research: Design and Methods (3rd ed.). Thousand Oaks, CA: Sage Publications.