An Empirical Investigation of Applying Lean Production to Software Development and IT Services: Outcomes, Challenges, and Lessons Learned

by

Osama Al Baik

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering & Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Lean principles for software development are relatively new, and their applications in software organizations are yet to be established. Software-centric organizations should be warned that transferring Lean principles and best practices from other industries is not a straightforward process; it is complex and requires reinvention and reinterpretation of practices in these new contexts. What are the applicable Lean principles to software organizations? What do they mean in the context of software development and IT services? What is required to implement these principles? What are the application strategies that practitioners have reported in the available literature? How can these strategies be better implemented? Many more questions have been answered throughout real-life field studies that spanned over more than a 5-year period, where various and mixed methodological approaches were employed, including action research and ethnographic methods. The principle goal of this thesis has been to empirically investigate how software organizations can realize the value that could be brought by the application of the Lean principles in software-centric and IT services contexts.

Preface

Collaborative Research Work. This thesis is an original work by Osama Hamdi Al Baik. Following is a list of published chapters included in this thesis. I was the primary investigator and was responsible for research design, data collection, and analysis, literature reviews, as well as manuscript composition. Professor James Miller, as the supervisory author, contributed in and directed the concepts formulation, research design, data analysis and validation, as well as manuscript composition.

1) Chapter 2 of this thesis was published as Al-Baik, O., & Miller, J. (2014). Waste identification and elimination in information technology organizations. *Empirical Software Engineering*, 19(6), 2019-2061.
2) Chapter 3 of this thesis was published as Al-Baik, O., & Miller, J. (2015). The Kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, 20(6), 1861-1897
3) Chapter 4 of this thesis has been submitted for publication in IEEE, Transactions on Software Engineering, first draft: October 14, 2016, second draft: April 19, 2017.
4) Chapter 5 of this thesis has been submitted for publication in Empirical Software Engineering Journal (ESEJ) , first draft: June 16, 2017, second draft August 26, 2017

Chapter 5 in this thesis involved an international collaboration with Dr. Bahae Samhan, an assistant professor at Illinois State University. The research project was led by Professor James Miller from the University of Alberta, while I was the primary investigator for this research undertaking. Professor Miller and Dr. Samhan helped in the data analysis and validation. I was responsible for research design, data collection, and manuscript composition. Problem implications, design, and execution of remedies, as well as the literature review; these were all my original work, except the subsection of knowledge management in the literature review, was a contribution of Dr. Samhan.

The following are published conference articles, but not included in this thesis:

1) Al-Baik, O., & Miller, J. (2016, January). Kaizen Cookbook: The Success Recipe for Continuous Learning and Improvements. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on* (pp. 5388-5397). IEEE.
2) Al-Baik, O., Miller, J., & Greening, D. (2017, January). Towards an Innovative Validation-Driven Approach to Lean Product Development. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Research Ethics Approvals. The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board. Details of the ethics approval are as follows: Study name: "Kanban-based Project Management for Software Development Organizations", No. Pro00038648, Date: July 3, 2013. Renewed: June 12, 2014, June 26, 2015, and June 28, 2016. The study included the following projects: **1)** Project Name: "Experiments with a Lean-based product development process", No. RES0020338, was approved on June 12, 2014, renewed: June 26, 2015, and June 28, 2016**, 2)** Project Name: "Validation Driven Software Product Development", No. RES0020338 was approved on June 28, 2016.

Acknowledgment

I would like to express my sincere gratitude and appreciation to my advisor, Professor James Miller. I cannot thank you enough for your endless support over the past several years.

I would also like to acknowledge the financial support by NSERC.

Finally, I would like to present my special thanks to my parents, Hamdi and Shadia, my wife, Yasmeen, and my daughters, Rozana, Juana, and Eliana!

Thank you all!

Table of Contents

List of Tables

List of Figures

1.0 Introduction

1.1 Overview of Lean Philosophy

Software development and IT services have evolved significantly since the start of the new millennium. Increasingly, scholars and practitioners have been trying to develop methods to improve software development processes, advance quality, enhance productivity, and augment predictability of their development efforts. These newer methods, however, are not without risk. Recent studies show increasing interest in the application of Lean philosophy to software-centric organizations. Womack, Jones, and Ross (1990) has defined Lean as:

> *Lean manufacturing, lean enterprise, or lean production, often simply, Lean, is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination*.

The House of Lean (Dennis, 2007), the visual representation of the Lean principles, practices, concepts, and tools, illustrates the essential foundation for successful Lean implementation. Figure 1.1 represents the elementary form of the Lean house. It suggests that to realize the ultimate goal of Lean, delivering "value to customers," the two main pillars of Lean: Just-in-time (JIT) and Jidoka must be built on top of a strong foundation, stability and standardization. This can be better achieved through involving and developing the manpower within an organization.

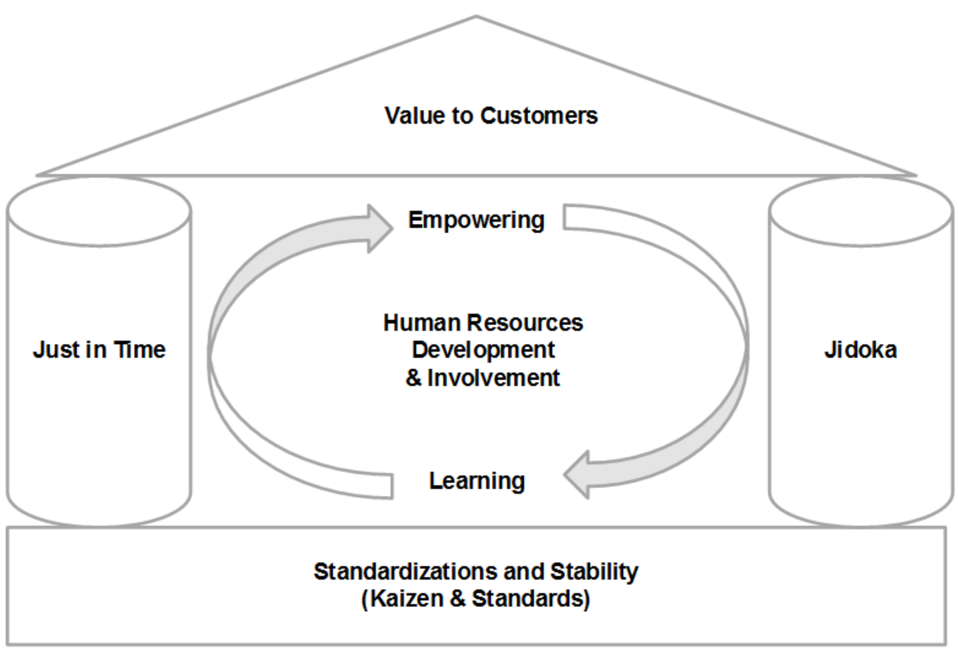

Figure 1.1 The Basic Lean House [*modified version, adopted by (Dennis, 2007)]*

Lean emphasizes that creating *value for the customer* should be the principal driver for work activities. Value-based engineering is a more accepted viewpoint outside of software development and IT maintenance; hence by considering other engineering fields, organizations in the software domain can hypothetically realize tremendous improvements and materialize many benefits. Applying and implementing the Lean philosophy, however, is not an overnight process. Lean implementation is an ongoing process. According to Womack and Jones (2003), the Lean implementation should encompass five core pillars, these are:

1) ***Value***: specifically, the value from the customer's point of view, and value should be expressed regarding a specific product or service.
2) ***Value Stream***: identify the set of actions or steps required to transform the *value* from concept into finished product or service in the customer's hand.
3) ***Flow***: once the *value stream* is well specified, and the wastefulness steps have been eliminated, make the *value*-adding actions, or steps, *flow* without interruption or delay.
4) ***Pull***: when the *flow* is realized, a period required to transform a concept into a finished *value* (product or service). Hence, the *value* has to be only introduced per the customer's demand. The *value stream* is triggered when the client pulls the *value* when needed.
5) ***Seeking Perfection***: when an organization implements the previous four principles, it realizes that there is no end for continuous improvement to seek *perfection.*

In order to amplify the likelihood of implementing Lean successfully, enough time must first be allocated to analyze and understand the nature of the organization, its capabilities, its business processes, and the environment in which the organization operates. This analysis would help understanding where the organization stands, where it would like to be, and what is required to realize that transition. Several research studies on the application of lean philosophy have shown encouraging results, e.g., (Fraser, Boehm, Jarkvik, Lundh, & Vilkki, 2006; Mujtaba, Feldt, & Petersen, 2010; Staats & Upton, 2009). However, scholars have agreed on that significant changes to the underlying culture of the organization, and its way of doing work is essential to implement lean successfully (M. Poppendieck & Poppendieck, 2003; Shah & Ward, 2007a).

1.2 Research Motivation

The software industry faces a major problem – how to define features that people are willing to pay for and want to use? Many software systems contain unnecessary functionality. Johnson (2002) reports that 45% of features in analyzed systems were never used (Juergens et al., 2011). On the usage of an industrial business information system showed that 28% of its features were never used. A study by DuPont revealed that 75% of commercial software features were not needed (Shelton, 2010). Also, during a speech, Jim Johnson, the CEO of Standish Group, stated that the results of the DuPont study showed that “On average, 45 percent of software features are never used, and only 20 percent of features are used always or often” (Shelton, 2010).

While the accuracy of the quoted percentages can be debated, they demonstrate that capturing the real requirements for a product is a poorly understood process. This problem is especially true when constructing for the mass marketplace. Lean has been promoted as a silver-bullet for improving bottom-line of organizations in whatever industry. This claim has been challenged by Dybå and Sharp (2012), they have called into question the ostensible claims about the Lean production by James Womack in his book: “*The Machine That Changed the World*,” which was published in 1990 and articulated the principles and practices of *Lean Production*.

Despite the many years of study, research, and development of crafting Lean principles and practices to be more suitable for software development and IT services, they still need considerable investigation (Staats, Brunner, & Upton, 2011; Wang, Conboy, & Cawley, 2012). This problem is not new to the field of software engineering. For example, in 1975, Lucas, as cited in Dybå (2002), suggests that “*the primary cause of system failure has been organizational behavior problems*” (Dybå, 2002, p. 388). After more than a decade, (Basili, 1989) emphasizes that engineering practices in the software context should be different from conventional manufacturing or hardware engineering; Basili (1989) asserts that processes, methods, models, or thinking approaches used in manufacturing should not be assumed appropriate for use in software engineering organizations.

After more than four decades, since Lucas underscored the importance of dealing with organizational problems, Viana et al. (2012); Virtanen, Pekkola, and Päivärinta (2013); and Calderon et al. (2015) have concluded that the low success rate of improvement initiatives, in the software engineering domain, is primarily related to the fact that process improvement initiatives focus on the technical aspects of the product development, whereas the underlying social and organizational aspects are often disregarded.

The improvement initiatives are generally more complex and challenging to implement in the software development context, due to the fact that software development process encompasses creativity efforts, combining social interactions and human intelligence (Abrahamsson, 2002), as such the social and human interactions can neither be resolved using engineering practices nor it can be standardized in a process model (Basili, 1989; Dybå, 2002).

Shah and Ward (2007b) call into question the level of understanding possessed by different scholars regarding “Lean thinking.” Several scholars, such as Dybå (2002), Petersen and Wohlin (2010), and Poppendieck and Poppendieck (2003, 2006), suggested that instead of focusing on imitating practices and technical procedures, software organizations should understand the essence of *Lean Thinking* and process improvements, which requires developing context-dependent practices that facilitate the achievement of “sustainable performance improvement”.

Some of the other major challenges that have been repetitively reported, in the available literature, are related to the lack of awareness, lack of understanding, lack of clarity, and absence of vision on "how-to" implement improvement initiatives and their impact on the organization culture and policies, management practices, and employees' behavior (Calderon et al., 2015; Dybå & Sharp, 2012). Dybå (2002) criticized software organizations for attempting to apply more rigorous engineering principles to resolve organizational problems. Motivated by this notion, this thesis tries to address these problems by developing a different perspective on what do lean principles mean to software development organizations? How can they be tweaked to better implemented in software contexts? What are the socio-cultural and organization behavioral methods to combine with lean principles to achieve the "sustainable performance improvement"?

1.3 Summary of Outcomes

To answer the questions mentioned above, the lean principles, concepts, and tools need to be empirically investigated as to how they can be reinterpreted and redesigned to fit the software engineering contexts better. Hence, the research group has conducted several field studies with an industrial partner, that is an internal medium-sized IT department that provides a broad range of IT-service offerings through 27 teams, including application hosting, system administration, networking, project management, desk-side support, physical infrastructure and data centers, and software application development.

At the time of writing this manuscript, the IT department had over 400 staff, an estimated annual budget of approximately $60 million, and provided services for 39,000+ end users. The research investigators and the IT department, as a research partner, agreed to keep the identity of the organization anonymous, hence, for the remainder of this thesis, we will refer to the industrial partner by an arbitrary name "ORGUS."

In 2012, the first research study was initiated when the overarching executive team of the organization had challenged ORGUS's senior management to reduce operational costs and service response time while increasing quality and customer satisfaction. Process improvement initiatives were deemed necessary to overcome these difficulties and to sustain the future performance of ORGUS. After careful analysis of the organization's capabilities and reviewing the trending improvement methods, ORGUS decided to proceed with implanting Lean Thinking and started with Value Stream Mapping and Waste elimination;

Reviewing available literature on the topic revealed that, previous attempts to implement lean initiative did not go beyond using software-related terminologies to re-label mimicked practices, from the manufacturing industry. For instance, waste taxonomy in the software engineering domain was mapped from the originally devised wastes categorization by Toyota. The waste "Over-processing" in manufacturing is called "extra development steps" in a software-centric environment. Similarly, the manufacturing term "inventory" is simply translated into "requirements" (Schyff, 2011).

Unfortunately, the terms reflect a mapping of terms from manufacturing environment rather than being discovered from within software-centric settings. To understand what the wastes in the software engineering industry are, research is needed to analyze the activities within the software/IT operations. A Lean initiative is not a "one size fits all" project. Hence, each chapter in this thesis provides a narrative of how different practices have evolved from within a software-centric context to implement one or more of the lean core principles, concepts, or methods.

The remainder of this thesis is organized as follows; Chapter 2 provides a narrative of how a new waste taxonomy model, to classify wastes in software/IT organizations, has evolved from within the software-centric settings. Also, a discussion of suitable elimination strategies is presented along with statistical indications of realized performance improvements. Chapter 3 sheds some light on what Kanban means to software organization. Using systematic review method, the findings show that scholars and practitioners considered and reported 20 different elements as part of the Kanban approach based upon considerations of being an Agile approach or a Lean principle; these elements, however, realized significant benefits and improvements to the software development teams. These benefits along with the challenges have been reported in this chapter.

Chapter 4 investigates the Lean's continuous improvement philosophy, that is Kaizen. The chapter looks into why do not traditional improvement methods deliver the desired results? And, how do we maintain the improvements after they are realized? For almost three years of ethnographic action research with 231 participants, the outcome is an integrated improvement model that combines social and organizational learning methods with Kaizen, that is an Integrative Double Kaizen Loop (IDKL), which has proven successful in promoting continuous learning and improvement within the context of ORGUS.

Chapter 5 reports a case study that was triggered as a result of observing a high rate of employees' turnover rate in ORGUS. Software organizations still struggle with the fact that knowledge is mainly locked away in the minds of their employees. In an attempt to resolve the issues related to turnover, this chapter emphasizes the need to rethink the way organizations handle turnover by retaining organizational knowledge, rather than retaining staff. This chapter reports how improvements can be realized through implementing changes to the cultural and social aspects of the organization. In summary, the overall contributions of this thesis can be summarized as follows:

- It provides a "Waste Taxonomy" that was discovered from within an IT Services settings, which is better suitable for identifying and eliminating wastes in IT Service processes.
- It illuminates the guiding principles and the prime elements of the Kanban approach
- It develops a simple model that has proven successful in integrating Kaizen with sociocultural and organization behavioral methods.
- It provides empirical evidence of the applicability of re-invented lean concepts and principles in IT Services context.

2.0 Waste Taxonomies and Elimination Strategies

2.1 Introduction

Waste identification and elimination are important and integral principles to Lean production. Waste elimination steps have proven successful in increasing the productivity of manufacturing organizations. These measures have helped to reduce both, time-to-market and product development cost; Evidence has been reported by Toyota, Japan. Through the use of Toyota Production System (TPS) to identify and eliminate manufacturing wastes (Liker & Hoseus, 2008; Magee, 2007). Having gone through many decades of development and enhancement, the Lean philosophy has been proven successful in the manufacturing industry (Magee, 2007)however, the feasibility of its application in different settings, particularly in software-centric organizations, has yet to be established (Hopp & Spearman, 2004; Upton & Staats, 2011)

Based upon results of research studies, produced by Shelton (2010), DeMarco (2011), Wang et al. (2012), and Schwaber and Sutherland (2012), a new waste classification model was derived by the reasons for high costs in certain phases of IT projects life cycle, IT operations, and the reasons behind IT project failures. A study by DuPont showed that 75 percent of commercial software features were not needed (Shelton, 2010). Also, during a speech, Jim Johnson, the CEO of Standish Group, stated that the results of the DuPont study revealed that "On average, 45 percent of software features are never used, and only 20 percent of features are used always or often" (Shelton, 2010, p. 4)

This chapter seeks to demonstrate validity and value of the proposed wastes classification model to identify and eliminate wasteful processes by applying it to the operations of a "medium-sized" IT department. The model has proven successful and has helped ORGUS in reducing operational costs and service delivery lead-time by 56 to 60 percent; on the other hand, ORGUS realized a higher customer satisfaction rate by 15.7 percent. During the lean initiative in ORGUS, the investigators identified twenty duplicate services provided by different teams, more than one hundred services rendered by inappropriate teams, and about 15 services that should not be offered at all. The main contributions of this paper can be summarized as follows:

- The model has been empirically evolved based upon IT work in a real-life project.
- The model has considered both, software development and IT service operations.
- To the best of our knowledge, none of the scholars in the existing literature has described "how to" identify and eliminate wasteful processes and activities. This report provides a detailed narrative of waste identification and elimination strategies that have proven successful in ORGUS.
- The report helps IT organizations in their decisions when implementing a Lean initiative in general, and waste identification and elimination in particular, as well as illustrating the challenges that come with a decision to implement it.

The lean initiative in ORGUS was tackled as a standalone project that lasted for 18 months, during which the project team was composed of two groups: 1) a core project team of 8 members, and 2) a supporting group, where other resources were engaged when necessary. The total number of employees involved in this project was 93 members, who worked at various periods of times ranging from one day to 6 months. The investigators have been involved in around 100 meetings that vary from 1 hour to 3 hours. The study included review and analysis of a total of 130 projects carried out in ORGUS, composed of 70 small projects, 35 medium projects, 15 large projects, and 10 extra large projects.

The remainder of this chapter is structured as follows: Section 2.2 summarizes the motivation behind this study. Section 2.3 sheds some light on the available literature on the seven types of waste, the core of Toyota Production System (TPS) and the previous literature review about types of waste in IT. Section 2.4 provides a detailed experience report in which we proposed a new taxonomy for wastes in IT organizations; the new taxonomy model has been evolved while being implemented in cooperation with the industrial partner, ORGUS. Section 2.5 shows statistical evidence of the performance improvements achieved within ORGUS, and finally, in Section 2.6 a summary of the chapter and recommendations are provided based upon both, the literature reviews as well as our experience from the organization we studied.

2.2 Study Motivation

Applying and implementing Lean philosophy is not an overnight process. In order to have a successful Lean journey, enough time must first be allocated to analyze and understand the nature of the organization, its business processes, and the environment in which the organization operates. A significant amount of research about applying Lean has shown effective results (M. Poppendieck & Poppendieck, 2003; Upton & Staats, 2011). However, these scholars have not described or explained how the wasteful processes have been identified and which waste elimination strategies have been used or were useful, evidence can be found in Wang et al. (2012). Several scholars, e.g., Hopp and Spearman (2004) and Staats et al. (2011), have acknowledged that waste identification is not an easy task, and others point to a lack of "how-to" methods for identifying and eliminating waste without further explanation.

In preparation for the Lean implementation in ORGUS, and in an attempt to provide cheaper, faster and higher quality IT services, the literature was reviewed for published empirical studies and experience reports that included narrative of the initiatives to eliminate wastes in IT organizations; see (Wang et al., 2012) for the actual reports. It has been concluded that:

1) Out of thirty experience reports, Wang et al. (2012) claimed that eight companies implemented the "Eliminate Waste" principle, and five of the eight organizations operate in IT-related business domains. After reviewing the actual reports, it was noticed that one of the studies reported development of a software application to eliminate wastes in a restaurant (Rand & Eckfeldt, 2004); another report did not try to implement "Eliminate Waste" at all (Nottonson, 2011); and the remaining three reports (De-Ste-Croix & Easton, 2008; Packlick, 2007; Yap, 2006) used Poppendieck's classification of wastes and simply stated that wastes were discovered and improvement opportunities were identified.

2) The waste classifications specified by Poppendieck and Poppendieck (2003, 2006) were mapped from Ohno's waste categorization. To the best of our knowledge, no waste categorizations or classifications have evolved purely based upon IT work, except two types of wastes: "Unnecessary Work" and "Avoidable Rework," which were identified by Petersen and Wohlin (2010) as part of interview questions.

3) Petersen and Wohlin (2010), and Poppendieck and Poppendieck (2003) emphasized that significant changes to the underlying culture of the organization and its way of doing work be essential to implement lean successfully. Also, Poppendieck and Poppendieck (2003) reported that software development organizations have not been successful in implementing lean, as the required changes to the organizations' culture and habits are not easy to apply.

4) Poppendieck and Poppendieck (2003) emphasized that the organizations that understood "Lean thinking," were able to develop context-dependent practices that facilitate the achievement of "sustainable performance improvement." When imitating proven techniques from other disciplines to software development, the problems most likely arose because of imitating the practices (context-specific) rather than the principles (M. Poppendieck & Poppendieck, 2003).

5) Shah and Ward (2007b) calls into question the level of understanding possessed by different scholars about "Lean thinking." Hence, the investigators started reviewing the original Lean essence from the manufacturing perspective to have an in-depth understanding of Lean principles. Waste elimination is one of the Lean principles; however, waste classification and identification are practices to implement the principle in specific settings. That is, context-dependent should differ from one environment to another (M. Poppendieck & Poppendieck, 2003; Wang et al., 2012).

In comparison to manufacturing, IT processes do not produce tangible wastes (Staats et al., 2011); thus, the wastes in IT are better identified and categorized based upon IT work rather than being mapped from manufacturing. This study identified nine distinct categories of wastes based upon a real-life project in IT organization, which supports our claim in the first bullet point of the reported contributions list. Successful implementation of lean in general, and waste identification, in particular, is challenging; this chapter reports the implementation of waste identification and elimination to both, software development and IT services through the use of ethnographic methodologies, where the primary investigator was involved in a real-life project for 18 months.

This study shed some lights on how the practices were determined based upon several factors in ORGUS, including but not limited to: culture, resources, budget constraints, and time constraints, which supports the claim in the fourth bullet point of the reported contributions list.

2.3 Types of Waste

Wastes are defined as additional activities that add cost or time without adding value (Dennis, 2007; Tapping, Luyster, & Shuker, 2002). The goal of most organizations is to provide highest quality product, in the shortest time, and at the lowest price. Customers, however, require additional attributes beyond product or service quality, price, and time to delivery; therefore, organizations implementing Lean need to focus on a broader set of customer expectations. For example, many Lean organizations have added safety, environment, and morale to their goals (Dennis, 2007). In order to provide a comprehensive overview of wastes, and to emphasize the differences between IT and manufacturing processes, this section is decomposed further into two sections: Section 2.3.1 provides a description of wastes in manufacturing, and Section 2.3.2 discusses the wastes in IT organizations, and the differences between the nature of IT wastes and manufacturing wastes.

2.3.1 Wastes in Manufacturing

The founder of Toyota Production System (TPS), Taiichi Ohno identified seven general types of waste in manufacturing (Womack & Jones, 2003), where products vary significantly from industry to industry. The wastes that are found in manufacturing environments are, however, quite similar. The main reason for this similarity lies in the nature of the physical and tangible wastes. The seven most common types of waste found in manufacturing environments are:

1. ***Overproduction***. Overproduction occurs when producing an item that is not intended for immediate use or sale is manufactured (Tapping et al., 2002). Overproduction is the source of many other types of waste. In fact, Ohno declared overproduction to be the root cause of other kinds of wastefulness and manufacturing evil as cited in (Dennis, 2007).
2. ***Waiting***. Waiting waste occurs when a worker has to wait for the work to be released from another employee, another manufacturing process, or for material to be delivered. Goldratt (1992) has explained that one hour lost in a bottleneck process is one hour lost to the entire manufacturing output and one which can never be recovered (Goldratt & Cox, 1992).
3. ***Transportation***. A transportation waste results when the material is moved more than is necessary; this type of waste is usually caused by an inefficient and poor workplace layout. Although transportation is a non-value adding activity, it is a necessary process. However, organizations should strive to minimize material movement.
4. ***Over-processing***. Over-processing waste relates to spending more time, efforts, and resources to produce a higher quality product than is required by the customer, or producing these products using inappropriate tools. Many organizations use high precision equipment where simpler tools would be sufficient (Tapping et al., 2002). Over-processing is considered the most difficult type of waste to identify and eliminate.
5. ***Inventory***. Inventory waste is defined as the retention of unnecessary raw materials, “Work in Progress” (WIP), and finished products. WIP is a direct result of overproduction and waiting. Unnecessary inventory tends to hide problems that must be identified and resolved in order to improve an organization’s performance. Unnecessary inventory increases lead times, consumes otherwise productive floor space, delays the identification of problems, and inhibits communication (Dennis, 2007).
6. ***Excess Motion***. Excess Motion Waste is any human “motion” that is unnecessary to the successful completion of an operation. This waste is related to workplace ergonomics (Dennis, 2007) and includes all instances of bending, stretching, walking, lifting, twisting, and searching for parts or tools. Jobs that need excessive human movement should be analyzed and redesigned to reduce the required amount of motion.

7. ***Defect or Rework*** Waste. Defective products need to be fixed or reworked; this waste category also includes productivity losses and waiting waste associated with disrupting the production flow to deal with the defects or rework. A large body of literature has been developed on the cost of quality, and specifically the cost of repairing defects and scrap (Besterfield, 2009).

Lean Thinking theorists and Lean Manufacturing practitioners have recently added underutilization of employees as the eighth type of waste (Dennis, 2007). Organizations should invest in utilizing employees' creativity so that they can work towards eliminating the other seven wastes and continuously improve their performance.

2.3.2 Wastes in IT Organizations

IT organizations have not experienced the same level of success with Lean implementations as has been achieved by organizations in the automobile industry. The lack of strategies that have been used to identify and eliminate waste is one of the important factors responsible for this disappointing performance.

IT organizations continue to fail to deliver reliable systems within scope and budget, and on schedule (DeMarco, 2011), thus the software industry is still considered to be a relatively young industry. The success stories of Lean Manufacturing in the automobile industry have prompted IT organizations to begin learning and implementing Lean principles and strategies.

As Toyota and other world-class organizations have come to realize, customers will pay for value-added work, but never for waste (Hino, 2002). In order to successfully eliminate waste, it is important to understand what the wastes are, where they exist, and how to attack them. The key to success in IT Lean initiatives is to understand the differences between IT and manufacturing environments: The Lean initiative is not a “cut and paste” project: rather, it must be developed based upon the nature of a specific business.

As reported by Wang et al. (2012), there were various attempts to apply Lean in IT: such as De-Ste-Croix and Easton (2008), Yap (2006), and Packlick (2007). These attempts, however, have been dependent on identifying waste based upon the proposed models by Poppendieck and Poppendieck (2003; 2006) that were mapped from manufacturing rather than being evolved from IT work. The mapped types of waste found in software development and IT operations are illustrated below in Table 2.1.

Table 2.1 – Mapping of wastes to IT operations and software development

The Seven Wastes of Manufacturing	The Eight Wastes of IT Operations	The Seven Wastes of Software Development	
Ohno (1988)	Waterhouse (2008)	Poppendieck & Poppendieck (2003)	Poppendieck & Poppendieck (2006)
Inventory	Inventory (excess)	Partially Done Work	Partially Done Work
Extra Processing	Non-Value Processing	Extra Processes	Relearning

Overproduction	Overproduction	Extra Features	Extra Features
Transportation	Transportation	Task Switching	Handoffs
Waiting	Waiting	Waiting	Delays
Motion	Motion (excess)	Motion	Task Switching
Defects	Defects	Defects	Defects
	Employee Knowledge		

Table 2.1 shows that the wastes in IT operations by Waterhouse (2008) and the wastes in software development by Poppendieck and Poppendieck (2003, 2006) have been mapped from the waste categorization that was established purely for manufacturing, specifically, the automobile industry. Poppendieck and Proppendieck have revised their mapping only after 3 years of publishing their first waste categorizations, where the Extra Processes, Task Switching, and Waiting have been replaced, respectively by Relearning, Handoffs, and Delays. It is noticeable that Task Switching has not been completely disappeared from the new mapping as it has replaced the Motion waste. This revisiting of the waste classifications is a clear indication of the need to develop more stable and suitable model for IT organization that evolves based upon IT core work.

Almost every attempt in IT has focused on the administrative aspects of the work done in IT, ignoring the actual core work that is done within IT core processes, this can be concluded from recent articles (Staats et al., 2011; Wang et al., 2012). For example, Upton and Staats (2011) identified the location of a printer in a different floor as a waste in an IT organization. Despite the fact that the far-site of the printer is a clear motion waste, however, it is not an IT core waste; rather, it is an administrative waste. To understand what the wastes are, where they exist, and how to attack them, we have developed a new model to classify IT-specific waste.

2.4 Case Study

This case study aims to answer the following questions:

- RQ.1 Are the waste categories identified by Toyota appropriate for IT service organizations?
- RQ.2 Can we develop a better understanding of the types of waste produced by IT service organizations?
- RQ.3 How can we identify wasteful processes in IT service organizations? And,
- RQ.4 How can we eliminate these wasteful processes?

The remainder of this subsection is structured as follows: Section 2.4.1 describes the study context. Section 2.4.2 presents an overview of the action research methodology used in this project and describes the main methodology phases.

Section 2.4.3 outlines the project approach that has been used during the implementation of the waste identification and elimination initiative. It also describes the techniques used to develop and visualize high-level process flows. Section 2.4.4 describes how the model was evolved and provides a detailed description of the identified nine categories of wastes, in addition to examples of wastes that were identified under each category along with their elimination strategies. Section 2.4.5 reports the improvements resulting from implementing the waste elimination tactics.

2.4.1 Study Context

ORGUS is an internal "medium-sized" IT department that provides services for 35,000+ end users, has a staff exceeding 250, and an estimated annual budget of approximation of $30 million. ORGUS provides a wide set of IT services divided and categorized into twenty-four groups, including, but not limited to: Application and website hosting, workgroup applications, system administrations, networking, project management, IT service management, desk-side support, physical infrastructure and data centers, and in-house application development. The services in ORGUS are divided into two broad categories based on ITIL framework (Addy, 2007):

1) *Request for Change*: this type of service requires a change that usually affects more than one individual and takes a long time to implement, typically weeks or even months, as it needs more effort to analyze the impact of the change regarding cost, benefits, and risks. *Requests for Change* are concerned with the services that could be seen as projects, i.e., deployment of a video streaming system.
2) *Service Request:* this type of service usually affects one individual; it should take only a short time to implement, typically on the same day and takes hours to implement. *Service Requests* are concerned with the maintenance and daily operations of the IT services provided, i.e., reset password or requesting access.

The twenty-four teams in ORGUS were organized and decomposed into eight strata based upon the type of service provided by each team. The eight strata are outlined in the table below. However, the team names have been omitted to keep the confidentiality of ORGUS.

Table 2.2 – The Decomposition of Teams into Strata

Stratum	Description
Financial Management	Includes a team of 9 members responsible for providing financial services, such as accounting, budgeting, and procurement.
Human Resources Management	Includes a team of 7 members responsible for providing human resources services, such as recruitment, payroll, and employees advancement.
Product Design and Development	Includes 3 teams of 37 members responsible for product development, design, architecture, standards and project management
Infrastructure Operations	Comprises 4 teams of 47 members responsible for networking and cabling, data centers, and hardware repairs (laptops, projectors, and other devices)
Service and Application	Includes 6 teams of 61 members responsible for security, server and storage upgrades, system administration, and application hosting.
Client Services	Comprises 3 teams of 32 members responsible for providing helpdesk services, printing services, call center and receptionists.
Onsite Services	Includes 4 teams of 41 members responsible for providing support for desk side (not running on the server-side) applications and hardware issues requiring onsite visits.
Relationship Management	Includes 2 teams of 18 members responsible for marketing and managing client relations, in addition to continuous evaluation and improvements to the existing services through the Information Technology Services Management (ITSM) team.

Within each stratum, we used *Purposive Sampling* (Easterbrook & Singer, 2008) and identified eight teams as a starting point. The sample included three teams from Product Design and Development and one team from each of the remaining strata except for financial management and human resources. The *purposive sampling* was helpful to choose the appropriate teams for the study. The teams were selected based upon several factors, including but not limited to: the importance of the core services they provide, the resource availability, the team willingness to participate in the waste identification initiative, and the number of different workflows performed by the group (the lower the number, the more the preference to include in the study).

2.4.2 Research Methodology

The action research methodology was implemented during this study. However, to increase the level of scientific rigor, we imposed *Cooperative Method Development* (Yvonne Dittrich, Rönkkö, Eriksson, Hansson, & Lindeberg, 2007), which implies implementing the *Action Research Cycle* by its three phases: *Understanding Practice*, *Deliberate Improvements*, and *Implement and Observe Improvements*. The three phases were chronologically implemented as described below:

Phase I: Within the *Understanding Practice* phase, the ethnographic method was espoused to understand the practice better, as this phase is considered the base for the second and third phases of the *Action Research Cycle*. The primary investigator was found to be a *normal participant* rather than an *observing participant* (Easterbrook & Singer, 2008; P Runeson & Höst, 2009), which helps to identify the gaps between what the people were actually doing and what they said they were doing within ORGUS.

As described in section 2.2 of this chapter, the successful implementation of Lean initiative requires underlying changes to organization's culture and habits (K Petersen & Wohlin, 2010; M. Poppendieck & Poppendieck, 2003), to understand ORGUS' culture, as a starting point, we analysed the ORGUS' policy documents, and in an attempt to understand ORGUS's employees, we reviewed the "personality analysis reports" that were prepared by third party professionals.

The analysis supported the decision to use a *Double-Loop Learning* approach (Argyris, 2004) This method was used as it helps to detect and to correct the problems by implementing the required changes not only to *action strategies* (the practices of the employees) but also to the human being's values that govern their *theory-in-use* that originates their actions. For more details on double-loop learning the related theories, see (Argyris, 2004).

Support from senior management and buy-in from employees were both necessary to influence the changes to the employees' *theory-in-use* and transform their *action strategies* from *Defensive Reasoning* to *Productive Reasoning* type of employees, which in turn should help in transforming ORGUS into a learning organization where the employees were encouraged to detect and correct the problems they create, see (Argyris, 2004). This transformation was required to facilitate the implementation of the necessary changes and encourage employees to detect the wastes and eliminate them.

Phase II: During the *Deliberate Improvements* phase, the second phase of the *Action Research Cycle*, the design of possible improvements was carried out in cooperation with the *research participants*, mainly, the team leads and directors. Eight team leads, four directors, two business analysts; one project manager (the primary investigator) and a change manager were involved through the *Deliberate Improvements* process.

Phase III: The final phase of *Action Research Cycle* in this project was *Implement and Observe Improvements* where the strategies and recommendations of the *Deliberate Improvements* phase have been accepted and agreed upon to generate commitment to start implementing them (Argyris, 2004; Y Dittrich & John, 2007). In addition, the investigators during this phase collected data about the performance of the organization using auto-generated reports and customers' satisfaction data to be the base for the improvements measurements (see section 2.4.5).

To illustrate action research process, consider the following example: during the *Understanding Practice* phase, we found that ORGUS had eight different active directories (AD) that had resided in various servers. During the *Deliberate Improvements* phase, recommendations were to build a central AD, while in the *Implement and Observe Improvements* phase, a project to consolidate ADs was implemented successfully, and a central AD has been constructed. ORGUS invested a considerable amount of resources to build a central AD, however, on the long run, ORGUS has definitely saved time, effort, and cost associated with maintaining several ADs and servers; in addition to the saving on customization and integration required for each application and system that needs authentication through ADs.

2.4.3 The Project Approach

A successful Lean initiative usually requires the support of senior management and buy-in from operational level personnel (Dennis, 2007). The first step towards getting senior management support was to convince senior management that applying Lean principles in general, and waste identification and elimination, in particular, can help ORGUS in providing higher quality products and services in shorter cycle time, while the operational cost can be decreased dramatically.

We developed a better understanding of the current problems in ORGUS through observations and informal discussions with operational staff. We began each discussion by pointing out the problems we noticed and asked employees for their personal perceptions. The problems were discussed with both, staff and senior management, which encouraged staff to raise the problems and concerns to the surface. This was a key to creating organizational buy-in.

We then had to convince senior management that we understood their problems, and that we have had a comprehensive understanding of Lean principles and the experience on how to apply them. We started by giving a presentation about the Lean philosophy to the Board of Directors. We focused on their viable problems, giving examples of how they might be addressed by applying Lean principles, concepts, and tools, i.e., *Create Value, Just-In-Time* and *Kanban*, *Poka-Yoke* and *Built-In Quality*, *Standardization* (Dennis, 2007). However, we mainly focused on *Eliminating Wastes*.

The first step in eliminating wastes was to ensure that each team conducted only the tasks that were necessary (value added) for their processes and eliminated all others, unnecessary (non-value adding) tasks. To achieve this goal, the team leads were asked to identify the services they provided, and to mark the ones they believed should be carried out by other teams. In addition, they have been invited to suggest which team should undertake these services.

The team leads were then invited to meet and discuss the identified services to reach consensus on who should run which of the services. The preliminary result of this initiative was impressive; we identified approximately twenty duplicate services provided by different teams, more than one hundred services rendered by the inappropriate team, and about 15 services that should not be offered at all. During the evaluation of the services under transition, the new service providers were asked for their feedback on how to improve the service quality, reduce the response time (lead time), and reduce cost. The feedbacks from the new service providers were deemed to be insightful; they provided the "fresh pair of eyes" which is often needed to determine wasteful components within existing processes.

As this project was not intended for scientific research only, the use of the technique to represent the value stream was based on several factors, the most important were cost and time, the options to represent the value stream (current and future) were: *Value Stream Mapping (VSM)* (Dennis, 2007), *Product Development Value Stream Mapping (PDVSM)* (Morgan & Liker, 2006), or SIPOC and Process Flowchart (Przekop, 2006). During the project, the investigators have had the chance to discuss the options with two of the Lean Product Development pioneers, Dr. Jeffery Liker and Mr. Ron Mascitelli, both of them explicitly stated that VSM is a powerful tool for manufacturing but not for IT/Engineering processes as the nature of the value stream in both industries are different. Dr. Liker stated "Jim Morgan developed this at Ford and had a lot of time and resources. Generally, I have found simpler versions of mapping for IT or PS are better".

At the end, both Dr. Liker and Mr. Mascitelli commended the idea of using process flowchart as far as it covers the metrics of waste elements, i.e., takt-time, the available production time divided by the customer demand (Dennis, 2007), waiting time, cycle time, lead time, and *Non-Value Added* (NVA) versus *Value Added* (VA) time. We used a straightforward approach to identify the wasteful processes that were "not required," by simply asking three questions:

I. Does the activity, or the process increase customer satisfaction?
II. Does the activity, or the process reduces cost, time, or effort?
III. Is the activity, or the process mandated by a legal authority? Is it a rule that we need to follow?

If the answers to all of the questions were "No," we would analyze the process and conduct a process improvement discovery procedure, using the SIPOC that stands for "Supplier, Input, Process, Output, and Customer." The modeled process is usually represented as a block diagram that also includes several activities (Przekop, 2006). An example is shown below in Figure 2.1.

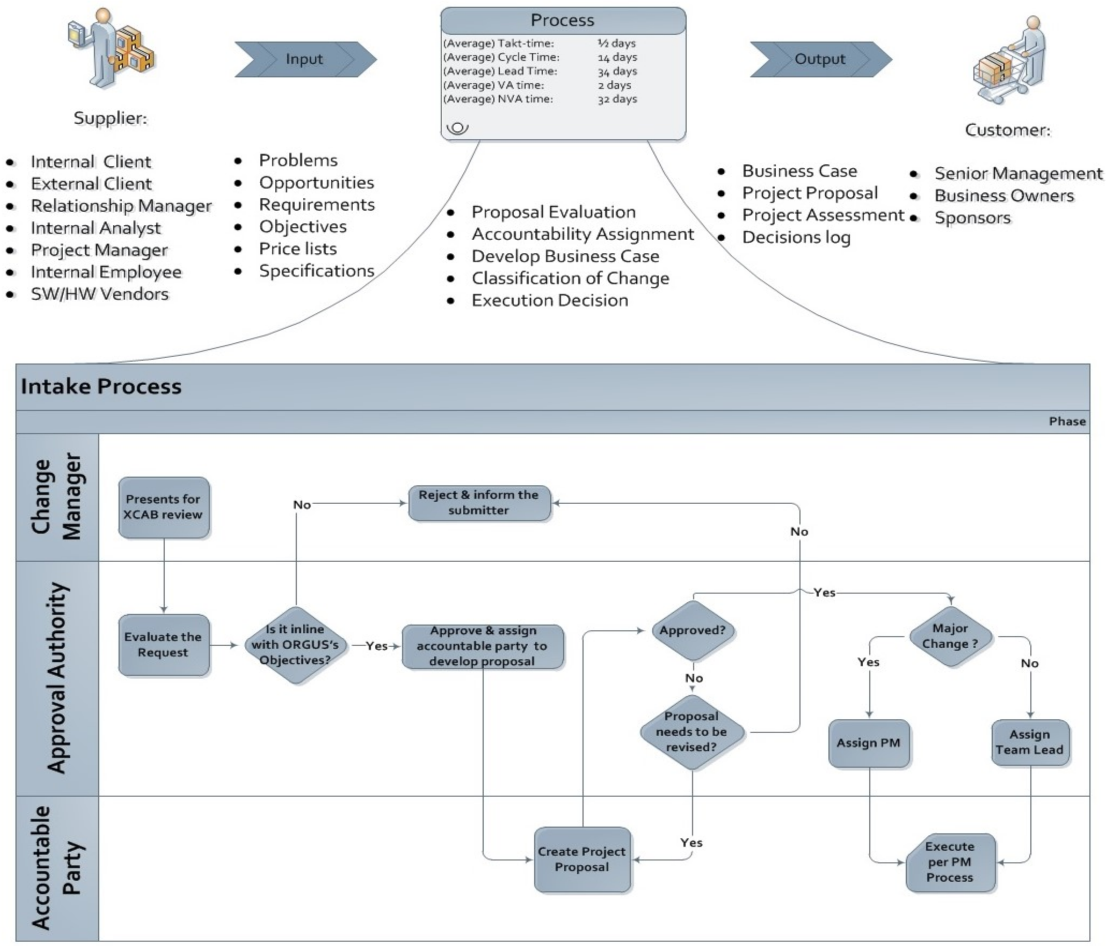

Figure 2.1– SIPOC and Process Flowchart

In the example of the "Project Intake Process" represented in Figure 2.1, we used the process flowchart to visualize the block diagram process within SIPOC. We first identified "who" are the suppliers, "what" do they supply, "how" is the process done which is presented using the flowchart, "what" are the outcomes, and "who" are the customers.

This identification has given us the details required to pinpoint the potential wastefulness activities in the workflow by confirming the requirements and expected outcomes with the right stakeholders who were identified by the "who" questions. Processes with high lead and NVA time were broken down and were presented with a greater level of details, going down to the activity level for which we identified more opportunities for improvements. The team leads also discussed and agreed to deadlines for the setup, configuration, and required training that would be necessary to transition the services from one team to another team(s). The transition process and activities are shown in Figure 2.2.

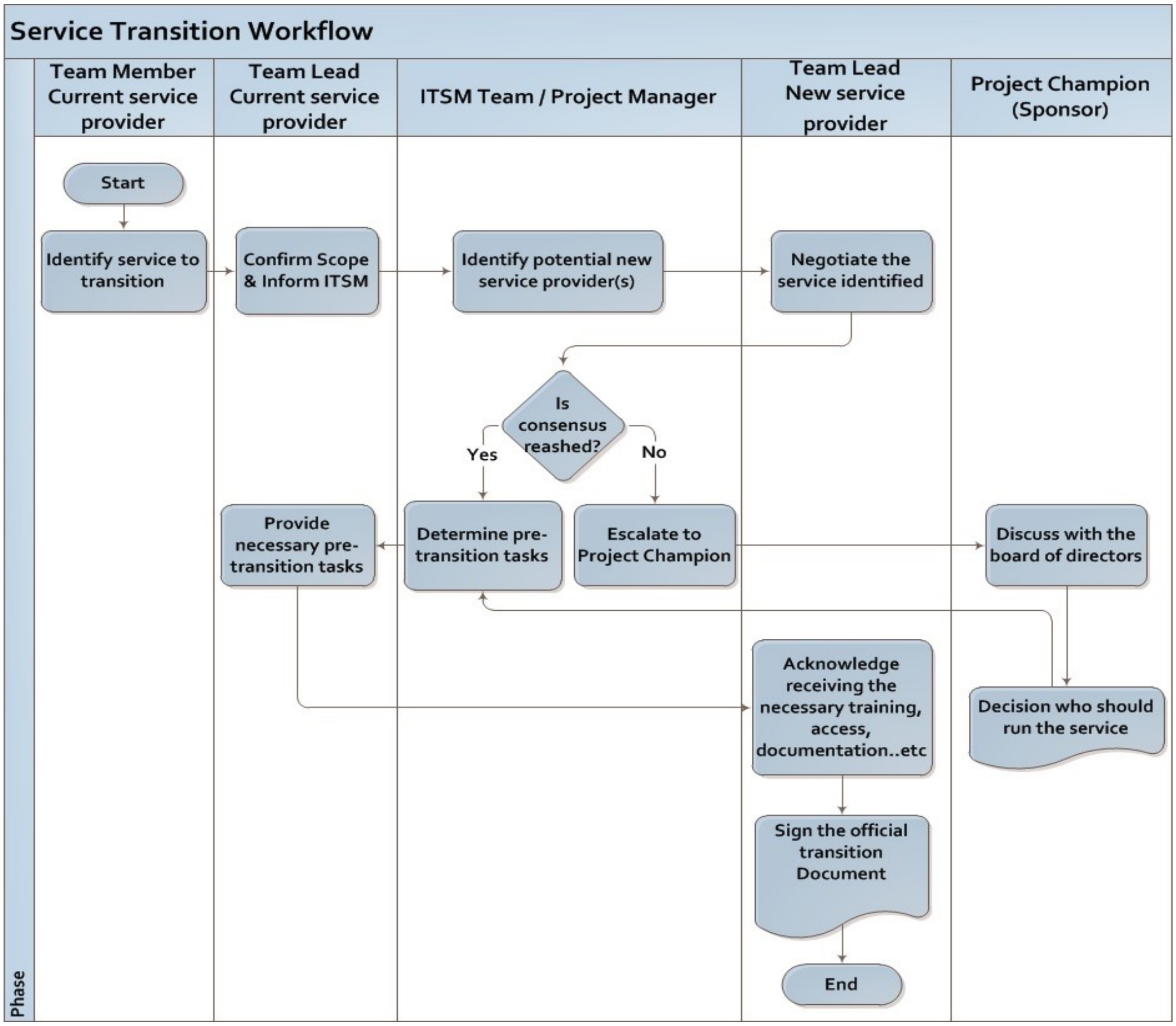

Figure 2.2 – Service Transition Workflow

As illustrated above in Figure 2.2, four-stage transition process was used during the transition process. First, we identified the service(s) to transition. The team member who was responsible for running the service was to determine the services that were to be transitioned. The current team lead– "current service provider"–was to work with the project manager to confirm the scope of the service under transition and inform the IT Service Management (ITSM) team. Secondly, transitioned services were negotiated and assigned.

The ITSM team lead worked with the project manager to identify the potential new service provider and meet with the potential service provider team lead to discuss and negotiate the service to be transitioned. If a consensus was reached, stage three was triggered. Otherwise, and if the newly identified team lead refused to accept the responsibility, the issue was escalated to the project champion–the sponsor–who would discuss the matter with the Board of Directors to decide which team was to accept responsibility for the service.

During the third stage "the transition," the activities that would guarantee a smooth transition of the services were determined and delivered, including training, configuration, granting privileges, and any other necessary activities to facilitate the transition. At the end of this stage, the services would be transitioned to new teams. A formal sign-off would be obtained during the fourth stage, and the service queue would be updated to reflect the changes, and the clients for each service would be informed. Several challenges arose during the transition process; the most frequent and problematic of these difficulties can be summarized as follows:

- Scheduling meetings; finding the time to put several individuals in one room was challenging.
- Reaching a consensus, more than 25% of identified services were escalated for final decisions on them (total number of services identified was 351 services).
- Building the transition plan and schedule as the new teams complained about the lack of resources to accept new services.
- The implementation of waste identification, determining, and depicting the value stream were challenging, as none of the employees have the experience or the knowledge to use VSM.
- The communications were difficult and required tremendous efforts to coordinate.
- The toughest challenge was changing the culture; For example, it took ORGUS' employees awhile to follow SPOC policy of not commencing a work without an assigned work order.

2.4.4 Waste Identification and Taxonomy Evolution

The identification of waste was an ongoing process conducted throughout the transition process, the taxonomy was developed, and wastes were categorized as they were revealed. During the evaluation of the services under transition, the new service providers contributed their feedback on how to improve the service quality, reduce the response time (lead time), and/or reduce cost. The feedbacks were usually obtained through face-to-face discussions between the research group and the new service provider and his/her staff, based on specific interview questions. The questions that were asked included:

1) If you are given a chance to change the way the service is offered, what would you like to change? Why?
2) How would like to implement the change? Why?
3) What are the benefits/advantages of implementing the change?
4) What are the drawbacks of NOT implementing the change?
5) Do we have the skills and experience to apply the change?
6) Who will be involved in implementing the change?
7) How long does it take? How much does it cost? (Rough estimations)

The feedback from the new service providers was deemed insightful; they provided the "fresh pair of eyes" which is often needed to determine wasteful components within existing processes. The feedbacks saved money and resources for ORGUS as the new service providers identified several services as duplications. These duplications had been identified during the development of the SQM catalog that specifies only one team responsible for each service. In an instance, during a follow-up meeting, a new service provider team lead discussed the licensing agreement for an application for which the licensing is based on the number of CPUs. For the licensing purposes, the application provider considers the following:

- Each physical CPU equals 4 virtual CPUs
- User Acceptance Testing (UAT) CPU should be licensed for half price
- Development (Dev.) CPU is free of charge
- Physical CPU license costs $30,000

ORGUS was licensed for four physical CPUs. Two CPUs were used for production, one for testing, and the fourth for development. The team lead suggested building a clustered environment of two nodes to increase the availability with four virtual CPUs on each node to increase the performance for a total of eight virtual CPUs. In addition, two virtual CPUs were dedicated for UAT on each node for a total of four virtual CPUs. The total licensing should come to three physical CPUs in which one of them was devoted for UAT. The overall results, enhanced availability, enhanced performance, increased customer satisfaction, lower efforts in monitoring the application, and saving of $45,000 per annum.

Organizations need to encourage a culture in which employees continuously identify root causes of wasteful effort and develop strategies to eliminate them. Instead of spending effort, time, and resources to implement *correction actions*. Organizations should strive to design and implement either *preventive* or *corrective action* (this is a required to transform employees from type *Model I to Model II*, for more details, see (Argyris, 2004). Organizations can avoid the effort and cost of correcting actions by eliminating root cause sources of waste, which implies that subsequent wastes are also eliminated.

Wastes that were identified during the transition project in ORGUS are highlighted in this section, organized by the waste classifications of the proposed model that was created and revised during the identification of the wastefulness processes in ORGUS. Some wastes could be categorized under more than one category; however, we identified the waste underneath one classification that was determined to be its root cause. The *5 Whys* technique was employed to determine the root-cause (Besterfield, 2009), this technique was developed and used by Toyota during the evolution of the TPS (Murugaiah, Benjamin, Marathamuthu, & Muthaiyah, 2010), to illustrate the *5 Whys* technique, consider the following real example that took place during the project:

1. *Why do* the employees in ORGUS implement workarounds?
   For the reason that they try to find quick fixes for problems.
2. *Why* do the employees in ORGUS try to find quick fixes for problems?
   For the reason that they try to avoid waiting for customer's feedback.
3. *Why* do the employees in ORGUS try to avoid waiting for customer's feedback?
   Because getting the feedback of the client is time-consuming.
4. *Why* is taking customer's feedback time-consuming?
   Because contacting the customer requires the relationship manager's approval.
5. *Why* does contacting the customer require the relationship manager's approval?
   For the reason that it is a company policy!

The root-cause of the identified waste was not the "lack of customer involvement" (see Section 2.4.4.3); the root-cause was centralized decision-making (see Section 2.4.4.5). The root-cause could be something else if the answers were different (Murugaiah et al., 2010), however, to validate the analysis of the root cause, we brainstormed the questions with fifteen different participants in three separate meetings. The brainstorming sessions resulted in nine participants (60 percent) providing answers that led to the identified root cause, while six participants (40 percent) responded to the questions in ways that resulted in a lack of customer involvement as a consequence of the absence of well-established customer involvement strategies.

In the proposed categorization, we identified the waste mentioned above under the lack of customer involvement/Inappropriate assumptions, as this particular waste could be found in decentralized organizations. However, we referred to the centralized decision-making as the root-cause to maintain the research honesty and integrity.

2.4.4.1 Waste Taxonomy 1: Gold Plating

Gold plating in software engineering relates to the development of unnecessary features or refinements. Gold plating is considered to be a poor project management practice: it results in the addition of functions that are not strictly within scope, or "scope creep" (PMI; Project Management Institute, 2008) that leads to cost overruns.

Developers are fascinated by new technology and about keen to develop new features ("bells and whistles") that, in their view, will enhance the software product; as well, developers often assume that their customers will be happy to see additional features in their products. Customers, however, might be disappointed by the results (McConnell, 1996).

In addition, the effort required to design, implement, test, document and support features that were not necessary lengthens the schedule and incurs additional cost (McConnell, 1996) and must be seen as a waste. *Gold plating* in its generic and broader definition may include the provision of redundant information, using additional tools or technologies, or providing a complex solution rather than the simplest one. The identified waste and elimination strategies, as part of the "Gold Plating" taxonomy, were:

Unnecessary tools, technologies, or methodologies. Likewise identifying the wasteful processes, we used the same approach to identify required tools, technologies, or methodologies that were "not required," by simply asking three questions:

i. Does the use of the tool, technology, or methodology increase customer satisfaction?
ii. Does the use of the tool, technology, or methodology reduce cost, time, or effort?
iii. Does a legal authority mandate the utilization of the tool, technology, or methodology?

If the answers to all of the questions were "No," we would analyze the process to support the recommendations to stop or minimize using the tool, technology, or methodology. Before evaluating the feasibility of the recommendations, we reported our findings to senior management for their review and approval. The ITSM team led the feasibility analysis phase, during which the effects of the recommended process changes were assessed in terms of cost, risk, and benefits.

A MoSCoW matrix is a prioritization technique. MoSCoW categories are: M-Must do it, S-Should do it, C-Could do it, W-Won't do it (IIBA; International Institute of Business Analysis, 2009). Senior management reviewed the resulting proposals during their regular biweekly meetings and used a MoSCoW matrix to prioritize the proposals. The highest priority proposals were placed in the appropriate queue for implementation.

Producing and distributing reports that are not required, and providing more information than is necessary. Providing more information than is necessary generates "time and effort" waste for both the report's author and its reader(s). An analysis of the reporting scheme in ORGUS revealed that, on average, 60 percent of the recipients did not have any significant interest in the reported item; moreover, these recipients did not know why they were receiving the reports. On average, 70 percent of the report recipients indicated that the reports provided useless or supplementary information that the recipients said they did not need to know.

The analysis of the reporting scheme was conducted through semi-structured interviews that included 20 directors and team leads. The questions were asked during the interviews included:

1) What reports you usually receive? Why?
2) What report(s) you should have received? Why?
3) For the report(s) you receive, do they provide sufficient details?
    a. If not, what are the missing details?
4) Do they provide more information than you need to know?
    a. If yes, what are the extra details?
5) What would you like to see in status/performance/weekly/incident report?
6) What is your preference for report presentation and format?
    a. Presentation (oral) Vs. written? Why?
    b. Word, PDF, PowerPoint, Prezi...etc.? Why?
    c. Graphs / charts vs. written description? Why?
7) For the current reports, what (you think) should be reviewed and revised? What changes would you like to see? Why?
8) How often should you receive each report?
9) How do you receive the report? Email attachment, hardcopy, etc.? How would you like to receive the report?
10) How much time do you spend on reading reports? (Per Week)
11) How much time (you think) you should spend reading reports?
12) On a scale from 0-10 (0 is the minimum, and 10 is the maximum), how do you evaluate the overall reporting system, format, details, presentation, distribution, etc.?
13) Do you have any question, comment, or suggestion?

The solution was to create a report template that contained the information that the recipients had deemed to be necessary based upon the requirements collected during the analysis phase. The new report template was revised based upon the comments and the feedback that had been provided by the directors and team leads. After four months of using the new template, we asked the question number 12 again. The overall evaluation was increased from 3.65 to 8.45.

Unnecessary cosmetic functionality designed and developed into the software. New technology tends to fascinate some developers, and its use can lead to the development of unnecessary new features or refinements without corresponding business requirements. To limit the resulting waste of effort, time, and resources, we introduced a Kanban board (Middleton & Joyce, 2012); items to be developed were to be clearly identified and visually represented on this board. In addition, we introduced the concept of "stand-up" meetings to ensure the team was focused on delivering only needed functionality, thereby reducing feedback cycle to one working day.

The research investigators were closely involved during the setup of Kanban and the early stage of the standup meetings; they attended the meetings for four weeks and provided guidance and feedback on how to run the meetings. Based upon the feedback from the Software Development team lead, Kanban has helped to keep the developers focused on the features they had to develop through the use of the Kanban board.

Striving to design perfect systems, rather than designing systems that simply are adequate. Our analysis showed that the system designers and architects in the ORGUS aimed to develop perfect system designs and architectures, rather than the simplest and "sufficient" designs and architectures. These employees tended to work in isolation, without taking into consideration other external factors, such as budget availability, lack of resources, and lack of required skills.

Initially, we tried to solve this problem by applying the *Quality by Design* philosophy through which design flaws and pitfalls are avoided by identifying the risks and limitations of the design (Besterfield, 2009). However, we noticed that the problem was, in fact, more related to the designers' and architects' mindsets: we reviewed these employees' "personality analysis reports," we found that 100 percent of the system designers and architects in ORGUS were categorized as perfectionists. Therefore, we needed to change the values that govern the *theory-in-use* through implementing the *double-Loop Learning* (Argyris, 2004) and promoted the idea of inquiring. Daily stand-up meetings formed a solid ground to build a collaborative environment and reminded the designers to ask themselves: Why do we develop it this way? Can we develop a simpler design?

2.4.4.2 Waste Taxonomy 2: Over-specifications

Developers and analysts sometimes define requirements and specifications that are not strictly required. For example, performance is stated as a requirement more often than it needs to be, and this can unnecessarily lengthen deployment schedules (McConnell, 1996). Over-specification wastes may also cause gold plating wastes. The identified waste and elimination strategies, as part of the "Over-specifications" taxonomy, were:

Excessive analysis that does not support decision-making. Analysis requires a substantial investment of time and effort. We observed that the analysts in ORGUS focused on specific problems and began "digging deep" into these problems during the early stages of the project lifecycle. Consequently, the analysis process produced a significant amount of data that needed further analysis, leading to what sometimes became an almost infinite cycle to "gather information and analyze further".

For example, in a project to implement a centralized, secure printing solution, the business analyst had found that the clients were using different printing station models, hence, he started "digging deep" to explore how the printing solution could be integrated with these different models, which led to more analysis of this specific requirement. The business analysis should have reported the findings and left the solutions architects and designers to look into how these pieces should be integrated together.

The elimination strategy for this identified waste was straightforward; however, time was needed to implement it correctly and to ensure consistent execution. First, we suggested that the team focus only on what is required at the specific stage of the project lifecycle. In this case, we recommended that a high-level analysis, to identify the problems be undertaken within the predefined time frame, without detailed analysis of any one problem.

The solution was realized by emphasizing the use of a template that we had prepared earlier based upon senior management's requirements. We used the same interview protocol and questions to analyze the reporting scheme as described in Section 2.4.4.1; each part of the new requirements template contains a precise description of what should be included in that section, which encouraged the analysts to provide only the information needed and not go beyond these requirements, thereby avoiding unnecessary work. Finally, we monitored the analysts' results to ensure they were producing the desired outcomes and adjusting their plans during the standup meetings.

Searching for information that is hard to find. Searching for information that is hard to find is a project bottleneck that contributes to "waiting waste." In ORGUS, we set out a "half day" rule for searching for information, if the search to find the information takes more than half a day of effort, the information is marked as difficult to find. Having identified this waste in the ORGUS, we propose two options:

1) If "enough" information existed to complete the task in hand, proceed further.

2) If more information was required, the analysts should work with subject matter experts to state all assumptions and document the assumptions as potential risks. We emphasized that assumptions should be seen as potential source of waste, and should only be stated based upon a consultation with subject matter experts and, if and only if, the information is difficult to find.

Overdesign of software or application. Generally, in the "real world," the time between preliminarily system design and the final product does not shrink if more analysis takes place up front. Overdesign wastes effort, time and resources, and may result in a design that does not fulfill the customer's requirements. The application of customer feedback model allows for the validation of the design and guarantees that the focus is on what really matters to the client. This recommendation was just left in the *Internal Implementation* stage. For the complete list of possible solutions to eliminate wastes and their implementation status in ORGUS based upon resources availability, see Table 2.4.

2.4.4.3 Waste Taxonomy 3: Lack of Customer Involvement and Inappropriate Assumption

In many IT organizations, making assumptions is a habit that results from a lack of customer involvement. Thinking on behalf of the client is not uncommon in IT organizations. As well, developers make assumptions about the future production environment, and although these assumptions may be relevant, for example, for a developer's workstation, they may be less relevant or completely inappropriate in a users' "production" environment (Humble & Farley, 2010).

Inappropriate assumptions can lead to the creation of new defects, which are hard to discover and fix, especially after the service, or system has been deployed into a production environment (Humble & Farley, 2010). Since the probability of introducing defects is high, end users will be significantly affected which leads, in turn, to low customer satisfaction. The identified waste and elimination strategies, as part of the "Lack of Customer Involvement..." taxonomy, were:

Misunderstanding of user requirements and developing software based on assuming what the client wants. we reviewed some previously completed projects carried out by ORGUS, focusing our review on those projects that had not met the needs or that had exceeded the time and/or budget. The shared team behavior that we identified in these projects was that of "thinking on the customer's behalf." Failure to spend "sufficient" time in the requirements gathering phase, and limited communication with end users resulting in project failure. Hiring competent business analysts was a relatively simple solution to eliminate the above-mentioned waste.

ORGUS hired a skilled business analyst who subsequently made a noteworthy contribution by eliciting more than 40 major functional, non-functional and transitional requirements, successfully delivering what the customers needed by focusing on requirements elicitation techniques. One large project was put on hold for about 30 days to gather customers' requirements: as a result, the project's scope, duration, cost, and completion criteria changed significantly. The major drawback to this solution was the cost associated with the hiring; however, senior management considered the "cost of quality" that would have resulted from not implementing the recommended solution.

An inflexible and lengthy approval process. Several reasons may lead employees to solve problems by implementing workarounds, some of these reasons are, unavailability of the customer to answer questions or to provide feedback and the desire to finish the task in hand as soon as possible. Subsequently, they try to find a quick fix which most likely will be a workaround. Generally speaking, eliminating a root cause waste will remove subsequent waste. However, the reverse is not true: elimination of "downstream" waste does not necessarily result in eliminating the root cause waste.

An inflexible and lengthy approval process that encourages workarounds (subsequent waste) is the direct consequence of centralized decision-making (root cause waste) that has been identified as a stand-alone classification, to eliminate this waste, ORGUS has changed some of its policies to empower the employees to overpass some approval processes, which reduced the waiting time and increased the staff morale.

An example of the policies that ORGUS changed was related to procurement for which the requester had to go through 4 approvals levels: 1) Team Lead or Project Manager, 2) Direct manager/supervisor of the team lead or project manager, 3) Director of the unit/department, and 4) Procurement Manager. ORGUS has eliminated the second and third approval steps and changed the policy to place the purchase order if the team lead or the project manager, and the procurement manager have approved it.

Work performed earlier than needed because of the absence of prioritization and dependencies. Due to the considerable number of projects that the IT department needed to manage, as well as the new project proposals arriving every day for review, a systematic approach to evaluate the importance, priority, and dependencies between projects was necessary. We developed a *MoSCoW* matrix to assign project priorities based on factors that ORGUS considered to be important, as shown in Figure 2.3.

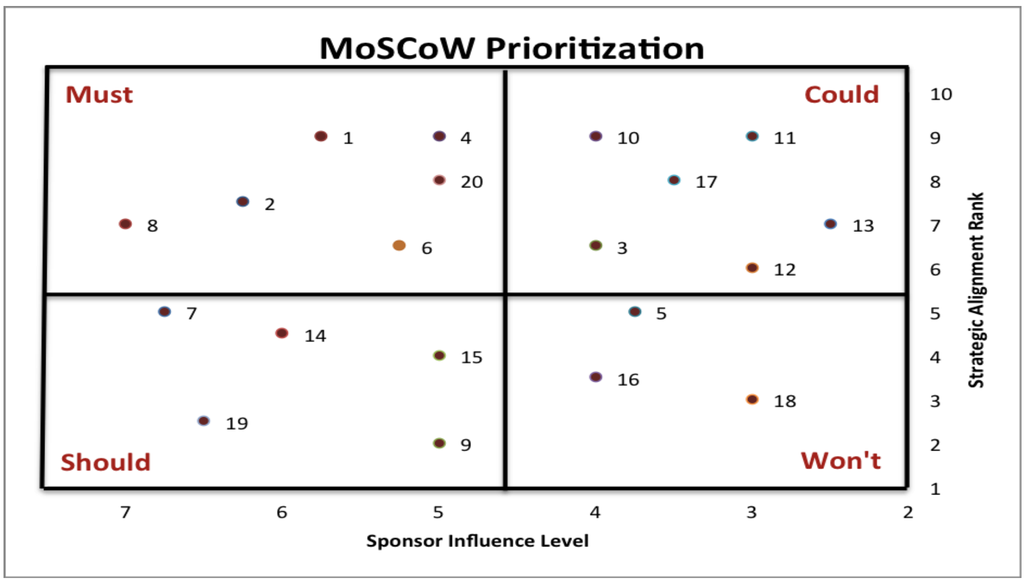

Figure 2.3– MoSCoW Prioritization Matrix

The use of MoSCoW prioritization helped senior management to decide which projects to initiate and which ones to put on hold. As well, the matrix contributed to clarify and manage the dependencies between projects, and prioritizing the projects contributed to fact-based decisions and enhanced resource management effectiveness and efficiency.

Lost information, thus, developer starts making assumptions. We noticed that each team in the IT Department had its own data repository; as well, very few teams had a backup mechanism in place for their data, while the others had multiple repositories for backup. We found that only one team used the department's version control system. Ironically, the organization has an enterprise content management system (ECMS) and a collaborative shared document environment. To eliminate the waste associated with missing information, we needed to eliminate its root cause. Our proposed solution was to increase awareness of the existing and available backup, version control, and document management systems and to mandate and regulate their use.

Limited access to the customer. The reasons behind the restricted access to the client were varied; however, the most common reason was the use of improvised or ad hoc customer access models. We recommended the adoption of more formal customer involvement methodologies, depending on the need for the feedback of the client. These involvement methods ranged from daily customer participation – that is, the customer as part of the team, using the tactic of the Extreme Programming (XP) methodology – to periodic feedback only, using prototyping methods (for more details, see (Martin, Biddle, & Noble, 2009)).

This recommendation has been just left in the *Internal Implementation* stage in which neither commitment nor approval has been given toward implementing the recommendations. For the complete list of possible solutions to eliminate wastes and their application status in ORGUS based upon resources availability, see Table 2.4 in the summary section.

Onsite visits to solve problems that could be solved remotely. Onsite visits to resolve problems are costly and time-consuming. We encouraged the use of monitoring systems and remote access software to undertake initial problem investigations as a simple strategy to eliminate this waste.

2.4.4.4 Waste Taxonomy 4: Double handling / Duplicate Processes

In today's business environment, Internet connection failures, email problems, and non-responding systems are among the biggest nightmares for business leaders. Non-IT leaders panic when they experience these IT-related problems and often escalate this panic randomly. In turn, IT personnel react immediately, especially if the person who is going through the problem is a senior manager or a lead customer. In many cases, more than one IT technician–sometimes an entire team–begins working to solve the problem. This duplicated effort magnifies the cost to address the problem, and the work done by one person or team to fix the problem might conflict with the work that the other person or team is doing to repair the problem, thereby creating more problems. To a great IT organizations or IT departments providing services to large organizations, it is usually difficult to eliminate these duplicate processes completely. Reducing these duplicated processes, however, is possible by analyzing the services provided using "Responsible, Accountable, Consult, and Inform" (RACI) matrix, illustrated in Figure 2.4.

Providing the same services by more than one team. We identified duplication as one of the major wastes in the ORGUS. The clients typically approached multiple employees in different teams to report a problem. Because of the high volume of daily requests and limited communication between various teams, it is hard to identify who was working on what. After spending some time to determine a possible solution, the study team concluded that the organization should implement a "Single Point of Contact" (SPOC) and develop a service queue map that would be used to send the problem to the appropriate team once it was received.

Process Name / Description:	*Cloud Printing Project*
Created On:	*12-01-29* Revision: *12-04-01*
Created by:	*Owen Jordan (PMO Mgr), Osama Al Baik (Project Mgr), James Miller (Project Sponsor)*

Activity / Task \ Role	PMO Mgr	Sponsor	Technical Lead	Technicican	Project Mgr
Identify a minimum of three printing solution providers (Vendor)	C	-	-	-	R
Arrange for vendor visits and quotes	I	-	-	-	R
Review quotes and references, make vendor selection	A	I	I	-	R
Review and finalize contract	I	I	-	-	R
Communicate project to stakeholders, make sure all users are aware of the go live date!	I	I	R	I	I
Provide access to users to be able to print by June 15	I	-	A	R	I
Oversee the project, ensure it is completed on time	A	I	I	-	R

R = Responsible, A = Accountable, C = Consulted, I = Informed

Figure 2.4 – RACI Matrix [Template: racichart.org]

As a result of this recommendation, the organization initiated two new projects, namely, a SPOC project and a "Service Queue Mapping" (SQM) project; both have been established as stand-alone and separate improvement projects in the organization. SQM aimed to prepare a service catalog to be used by the SPOC team to route and dispatch the *Service Request* or the *Request for Change* to the appropriate team. The service catalog included the service name, description, technology used, list of supported clients, escalation levels, and the team responsible for solving the problems.

During the identification of a team in charge of each service, we noticed that several teams had claimed responsibility for providing 21 duplicate services, the team leads were invited to make sure that the services rendered were identical, then a transition process was initiated to identify the first and the second level of support. The SPOC and SQM have realized several improvements, including, but not limited to:

1) Increased customer satisfaction;
2) Reduced response time, as it has become easier to route the reported issue to the appropriate team; and
3) Increased productivity and reduced cost, as one employee is guaranteed to work on the issue rather than having multiple employees working on the same work item. This has been realized through assigning work orders by SPOC and creating a policy in ORGUS that no work should be started without an assigned work order.

Having redundant Data in the same or different systems. One of the many problems associated with storing duplicate data is the amount of effort and time needed to maintain and update the same data in multiple locations. Within the same system, the problem of duplicated data can be simply addressed by normalizing the data in the database (Xingjian, 1992). However, if the data is duplicated in different systems, the solution lies in establishing a central data repository. The situation was fully described as an example in section 2.4.2.

Multitasking, in which people move between unfinished projects because of the absence of project prioritization. We noticed that employees stopped undertaking their daily operational tasks to undertake project related tasks when a project manager pushed them to work on a project with a deadline; the employees began missing deadlines of their operational tasks because of this movement between tasks. As mentioned previously, a waste elimination strategy can eliminate more than one waste at a time. In fact, waste elimination strategies themselves should be developed based upon the Lean principles. MoSCoW prioritization is one good example, which can be used to eliminate several wastes at once and can be utilized to prevent the organization from creating other wastes, the waste resulting from multitasking could be eradicated by prioritizing the work on a team level using a MoSCoW prioritization matrix.

Repetitive, unnecessary, and ineffective meetings. Avoiding unnecessary meetings is a time management "best practice" (Walsh, 2008). As well, the best strategy to prevent unnecessary meetings is not to schedule a meeting in the first place! A simple estimate of the time spent in meetings at ORGUS revealed a rather significant amount of money was being devoted to ineffective meetings. Top management in the organization was acutely aware of this problem. We identified several reasons why repetitive and inefficient meetings were scheduled and held. These included: lack of an agenda, ineffective meeting facilitation, lack of a meeting review and conclusion summary, and lack of formal adjournment. The investigators reviewed and analyzed the documentations of around 200 meetings to assess their effectiveness. The assessment was based upon several factors that were formed as questions, which are listed below

1) Did the meeting have an agenda?
2) Did the facilitator distribute the agenda prior to the meeting? If yes, how long before the meeting?
3) Did the meeting start and finish on time?
4) Did the agenda items have specific time limits?
5) Did the agenda clearly define the meeting objectives?
6) Did the facilitator assign a meeting scribe? If not, who was documenting the minutes?
7) Did the facilitator run the meeting according to the agenda?
8) Did the facilitator give time for the attendees to express their thoughts? Did the moderator/scribe take notes and document these ideas?
9) Did the facilitator review and summarize the meeting outcomes and action items (if any) at the end of the meeting?
    a. Did the facilitator follow up on action items (if any)?
10) Did the moderator/scribe distribute the meeting minutes? If yes, how long after the meeting?
11) Did the meeting achieve its objectives?
    a. If not, what were the obstacles that prevented the team from achieving the objectives?
    b. What were the steps taken to remove the obstacles?

The strategy we recommended to minimize the organization's meeting waste was to develop and implement meeting rules. These rules included: defining the meeting's goal; sending invitations, along with a clear agenda, at least two business days in advance; setting time limits for discussion; effective meeting facilitation by engaging attendees and encouraging them to participate; ensuring that any important off-track points should be noted and "parked" for further discussion in another meeting(s); writing clear meeting minutes; identifying action items with specific due dates and follow-up plans; and finally, conducting post-meeting evaluations and reviews. The ORGUS established a campaign to emphasize the importance of effective meeting management, and senior management highlighted the rules to the employees at periodic department meetings.

The investigators had been involved in about 100 meetings related to the waste identification and elimination project, where they used the rules mentioned above. In addition, 3 months after applying the rules, the investigators reviewed the documentations of around 200 meetings. The quality of the meetings results and documentations were dramatically improved.

Troubleshooting to solve the single problem instead of root causes. Conducting root cause analysis takes more time and effort than simply addressing and resolving a single problem (Ammerman, 1998); however, the summation of time required to troubleshoot and treat each single problem that results from the same root cause is much greater in the long term. The elimination strategy for such waste must be Lean as well, that is, root cause analysis must be employed. In ORGUS, we made it clear to the teams that, by spending more time to analyze the problem and implementing the appropriate corrective action(s), the cost, in the long run, would be much cheaper and better than finding workarounds to solve each problem individually.

Focusing on team optimization, not overall departmental performance. Team optimization processes can improve the speed and quality of a team's performance; however, such optimization may result in the team missing the "bigger picture" (Campanello, 2004), and not having the necessary information or knowledge of the organization's long-term goals. As well, teams should not work in isolation to optimize their performance, since organizations cannot produce more than the capacity of their bottleneck operations (Goldratt & Cox, 1992).

The optimized performance and increased production speed of any one team are irrelevant if the input operation to that team is a bottleneck (Antanovich, Sheyko, & Katumba, 2010). Therefore, optimization should be carried out on the level of the organization as a whole, rather than at the individual team level. Senior management took a significant step forward by ensuring that the organizational long-term goals and strategy were transparent to all employees.

Furthermore, management adopted a waterfall approach to communicate strategic decisions and plans to all employees on a weekly basis. The organization also implemented an open-door policy so that upper management could receive input directly from operational-level employees. All of these changes to the *organizational defensive routines* (Argyris, 2004) contributed in shifting the focus in ORGUS from individual team optimization, to the organization overall optimization.

2.4.4.5 Waste Taxonomy 5: Centralized Decision Making

Under a centralized decision-making model, senior management makes all of the decisions and lower organizational levels implement these decisions (Bhagat & Steers, 2009). Decisions that are made by lower-level employees are subject to approval by senior management. Centralized decision-making results in longer lead-times. The approval process may take longer than necessary because, in many cases, senior management will require more information in order to evaluate the situation before making a decision (Bhagat & Steers, 2009). By the time senior management evaluates the situation and makes or approves a decision, the information that was provided as background might be obsolete; this is true especially for time-sensitive decisions.

Centralized decision-making, therefore, leads to waste and, in turn, creates other types of waste, most noticeably, waiting waste. The best strategy for eliminating this waste is to implement a decentralized decision-making model. Senior management should establish the strategic goals for the organization and make key decisions, but trust and empower their employees to take the actions and make the decisions that are necessary to achieve these objectives. The identified waste and elimination strategies, as part of the "Centralized Decision Making" taxonomy, were:

Unnecessary approval processes and unclear responsibility and authority. ORGUS had undertaken actions to restructure its business processes, which resulted in the main organizational changes. Senior management needed to retain decision-making authority, especially during the early stages of a restructuring that was undergoing. As time passed, the organization chart became more robust, and the responsibilities, accountabilities, and authorities have become apparent using the RACI matrix. This enabled ORGUS to eliminate unnecessary approval steps.

Complex reporting systems that add extra effort without adding value. Employees in ORGUS were required to prepare different reports for different purposes. A previously created report template was modified based upon stakeholders' requirements. The project management team experienced additional types of reporting problems that created extra overhead. For example, project managers were asked to create different reports, such as reports for immediate managers, reports for project's sponsors, and summary reports for senior officers. On average, by considering managing multiple projects, time to prepare the reports was estimated at 25% of PM's time.

We examined the time required to develop these different reports as pure waste since they did not add any value from the customer's point of view. We recommended an automated reporting mechanism using an MS Excel spreadsheet. Project managers would complete a weekly report template by entering data into the worksheet; the different reports would then be populated and produced from this data.

Making decisions that lock in resources too early and produce inflexibility. Poppendieck and Poppendieck (2003) proposed, *"Decide as late as possible"* as one of the Lean software principles. Some people confuse planning and commitment (Schyff, 2011), planning is required; however, flexibility in planning is desired. As well, effective resource management distinguishes the "good managers" (Campanello, 2004).

Project Management Institute (2008) has recommended the adoption of a matrix organization structure to help project managers, in collaboration with functional managers/team leads, to plan and build a project schedule according to resource availability. As well, we recommended the use of MoSCoW prioritization matrix, which can be a beneficial technique to negotiate the availability of required resources for prioritized projects so that these can be delivered on time.

Lengthy approval processes that discourage innovation and creativity. ORGUS has applied ITIL (Information Technology Infrastructure Library) best practices to improve productivity, improve customer satisfaction, reduce cost, and improve the IT services. Adopting ITIL implies adopting its set of approved methods (Addy, 2007), thereby, eliminating lengthy approval processes. For example, ORGUS has a pre-approved list of 73 standard changes that do not need to go through the approval process. Based on the reports generated from the ticking tool, on average, ORGUS receives 200 standards changes on a daily basis; eliminating the review and approval process showed a saving of around $235,000 per annum and reduced waiting time by two working days.

Lack of delegation to empower the employees resulting in technical decisions made by non-technical people. Successful leaders coach, rather than direct, their subordinates (Besterfield, 2009). Employee empowerment is one of the Lean best practices that can save time and minimize rework; moreover, empowered employees are more confident and loyal (Womack & Jones, 2003). Employees are usually more committed to decisions that they either make themselves or that they participate in making (Bhagat & Steers, 2009). The employees who have the largest capacity to add value are not necessarily those who are at the top of the organization hierarchy (Campanello, 2004). Rather, the employees who add the most value are usually those who are more intelligent; at the same time, those are the employees who are typically not easy to manage (Younker, 2009).

"Smart" employees do not usually accept the superficial value of a decision made by management; this explains why it is necessary to get their buy-in to decisions. These employees are not impressed with titles; they tend to understand the rationale behind the decision and believe in expertise rather than simply accepting a decision because of the authority level of the decision-maker (Harvard Business Review, 2011)

The project management team was experiencing high overhead for being involved in every initiative tackled by ORGUS; senior management seemed to see each initiative as a project, while in fact, the project management experts in ORGUS had a different definition of what to consider a project. Project managers were expected to be involved very early in the process and spend time developing project plans for *Request for Change* (for more details see section 2.4.1) that was still "not approved." In cooperation with the project management group, we developed a "Project Intake Process" to define what should be considered a project. The intake process is shown in Figure 2.5.

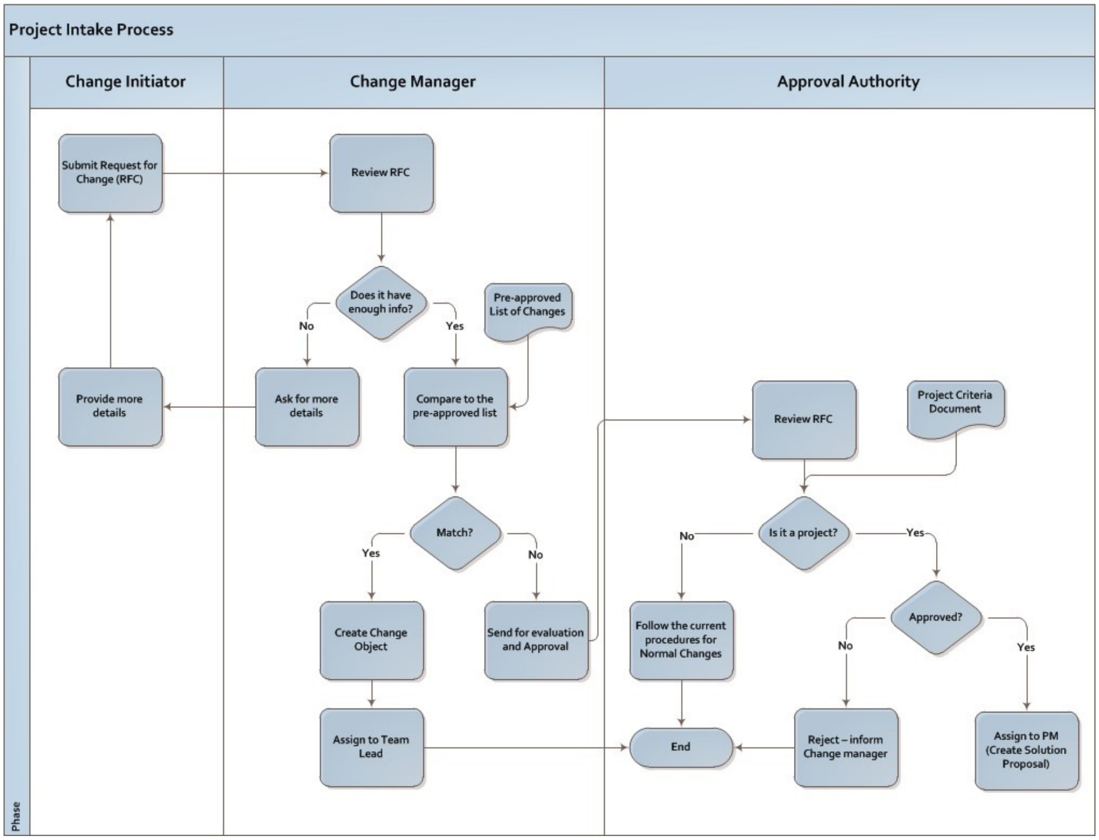

Figure 2.5 – Project Intake Process

2.4.4.6 Waste Taxonomy 6: Waiting

The waste of waiting in software-centric processes is similar to the waiting in manufacturing. The nature of the waiting waste, however, is different. *Waiting* waste in IT occurs when an employee has to wait for work to be released from the predecessor operation (Dennis, 2007). The main causes of the waiting waste in IT are: waiting for approval on action items, for example as a result of centralized decision making, fixing defects, or waiting for customer feedback.

In general, personnel always try to avoid waiting—not because it is wasteful, but because it is human nature. For example, clients who are waiting in a queue watch the clock every second while they are waiting to be served. However, they will not pay attention to time while they are being served, regardless of the time that it takes, or a number of other customers in queue.

In order to avoid waiting, IT personnel would initiate an avoidance strategy and start working based on assumptions. These assumptions may not be correct and may produce other types of waste such as the "inappropriate assumptions" waste discussed earlier in this paper. The identified waste and elimination strategies, as part of the "Waiting" taxonomy, were:

Waiting for review, and approval on action items. The implementation of ITIL in ORGUS resulted in a faster approval process for the action items and corresponding change requests that were on the pre-approved list. However, major action items that triggered major *Request for Change* (for more details, see Section 2.4.1) still needed to be reviewed and approved.

The solution for the waste that is produced by waiting for review and approval of the major action items is embedded within the ITIL best practices and involves the implementation of a *Change Advisory Board* (CAB) that is responsible for reviewing and approving these major changes (Addy, 2007). This review process could have a “time box” of 24 hours to minimize the waiting time. For details on improvement resulted from introducing “time box,” see section (2.4.5).

Waiting for information that is needed to complete a task. The above waste is an excellent example of a “push versus pull” system. Lean supports the pull system as it creates a continuous flow (Jaisingh, 2012). By using the knowledge management system to store and access shared documents and project workspaces, employees in the ORGUS have the ability to get up-to-date information at any time, retreieving the necessary information from this repository whenever it is needed reduced the waiting time by 2-3 business days based on a survey with 23 participants.

Using shared resources; waiting to assign specialists to accomplish tasks. Collaborative decision-making enables better resource management (Power & Sohal, 2000). The investigators held 3 focus groups in three separate venues with 23 participants, including 7 directors, 8 team leads, and 8 project managers. This arrangement offered a more convenient environment for participants to express their thoughts and had provided freedom to speak up without the presence of their supervisors. The investigators in cooperation with the participants identified three main strategies to eliminate the waste of waiting for shared resources: 1) specialists should be scheduled in advance, 2) slack time should be considered, and 3) contingency plans should be developed.

Developer wait tester to finish before starting work on the next piece of code. Lean suggests using WIP to create a continuous flow and minimize idle time (Tapping et al., 2002). In order to effectively and efficiently manage their WIP, developers in ORGUS have used a Kanban board to visualize the work items’ progress and continue to work on the tasks that have been assigned to, within the WIP limit.

When a testing engineer pulls a work item for testing, the developer should not be idle and wait for the testing results. Once the testing is done, and if further coding is required, the testing engineer should add the work item to the Kanban backlog with an assigned priority. The developer, according to the work item priority and the WIP limit, either puts one of the current work items on hold and pulls the returned one from the backlog, or just finishes the current work item currently “in hand” and pulls the returned work item by the testing engineer on completion of the current work item (Boeg, 2011).

Waiting for a physical signature. Waiting time for a paper-based signature is a pure waste; the elimination strategy for this waste was identified to use a digital signature based upon a survey with 11 participants including 7 directors, 3 managers, and 2 executive directors. The use of digital signatures was proposed to senior management at ORGUS; by the time this paper was written, the investigation was underway to identify and select a digital signature technology and online signature service provider.

Waiting to determine the team for providing problem resolution. Identifying responsible team for resolving a specific issue is no longer an issue for ORGUS. As mentioned earlier in this chapter, ORGUS has initiated SQM system. This system enabled employees to search for the responsible team, by the service name, by the technology used, and/or by part of the service description.

Clients waiting for calls to be transferred to the appropriate team. Clients, of ORGUS, use a variety of communication channels for reporting problems or for making information inquiries; these channels include email, online chatting, an online ticketing system, and telephone. Regardless of the communication channel, a "ticket" is to be created and assigned.

Using email, chat, or online ticketing system is much faster and efficient than "calling" for these requests, for obvious reasons. The telephone is the only communication channel that requires someone to be dedicated to the client as part of the communication task; as well, it is the only channel that requires the client to wait idly (i.e., on the phone) while waiting to get an answer.

We recommended automated call messaging to eliminate telephone waiting time and to minimize effort and cost; this system uses automated messages and voice-to-text technology to create a "ticket." This is an example of where the recommendation was made during the *Deliberate Improvements* phase and left on the *Internal Implementation* stage as no actual effort was made to put the recommendation into practice, hence moving to the *Implement and Observe Improvements* phase to measure and validate the result of the *External Implementation* stage.

2.4.4.7 Waste Taxonomy 7: Deferred Verification and Validation

In the context of a system development life cycle, deferring verification and validation (V and V) activities means that the V and V are expected to be carried out in later stages of the project. In the waterfall development methodology, this is called the "big-bang" approach. As can be seen in Figure 2.6 below, the cost of fixing defects increases dramatically as the it progresses to later stages of the project (Humble & Farley, 2010; Watkins, 2009). The identified waste and elimination strategies, as part of the "Deferred V and V" waste taxonomy, were:

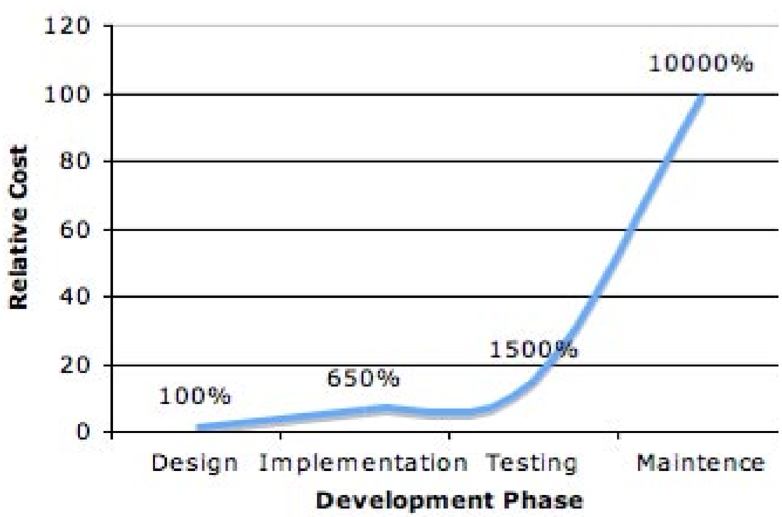

Figure 2.6 – Relative Cost of Fixing Defects [IBM Research]

"Lack of system integration between legacy systems (Waterfall instead of Agile). The best solution to reduce the risk of project failures from inadequate V and V is to employ "Continuous Integration" (CI) that integrates the verified features into existing system continuously. Starting V and V in early stages of project life cycle guarantees higher quality and lower cost to fix bugs than doing the V and V in later stages. Continuously integrating the tested features into the system eliminates the time, effort, and cost required for integration testing (Watkins, 2009)

As well, continuous integration helps in validating design assumptions and reducing the feedback cycles. The "big-bang" approach of the waterfall methodology has been proven to be the most expensive development approach, in terms of fixing defects (Pries & Quigley, 2011). Agile techniques in software development reduce the cost of fixing defects significantly by aiming to reveal and repair the defect during its development (Humble & Farley, 2010; Watkins, 2009). In summary, waiting until the later stages of development to test the entire development at once is extremely wasteful.

Excessive standards to follow that encourage avoiding testing. We identified the use of checklists and coding standards as one of the main root causes of waste in the ORGUS. The IT employees were essentially unwilling to follow any standards written by others. Generally speaking, people prefer to have options; they do not like the fact that they have no other choice except to work according to the prescribed standards. The research investigators introduced the concept of "self-accessed policies" for the first time in a presentation that we delivered to a different IT organization. The logic behind this concept is that a team is more willing to follow rules that it has established itself. The rules that projects' team members were asked to establish were to cover three coding aspects: 1) typographical conventions; 2) naming conventions; and 3) structural conventions (Younker, 2009)

Not enough training on, or awareness of, the importance of testing. ORGUS delivered a series of training sessions on the change processes associated with, and as part of, ITIL implementation. The importance of testing was emphasized as one of the primary training topics during these sessions. Major action items—those that create and lead to change—require a test plan amongst other schemes, before they can be approved. Based on several interviews with more than 50 participants, the importance of testing has become apparent for 85 percent of the participants who became more confident of the features they developed or the services they offered.

Insufficient funds for testing based on budget underestimation. As mentioned previously, the testing cost should be considered when estimating the overall project budget. The accuracy of these testing cost estimates will improve with experience. The use of the collaborative decision-making and historical data from similar projects that are available in the shared environment makes it easier to estimate project costs more accurately. As well, applying a bottom-up estimation strategy has proven to be effective for projects that lack the existence of historical information for similar projects (PMI; Project Management Institute, 2008).

As part of the employees' development program, members of the Project Management group in ORGUS delivered a series of training sessions on costing and estimation to support this requirement, in addition, to refresh sessions every quarter. The investigators reviewed 35 small projects for which the same personnel estimated the cost (17 prior to the training and 18 after the training). The results showed that over 66 percent of the projects were delivered within 15 percent cost tolerance in comparison to about 41 percent with same cost tolerance.

Limited time to release. An agile team should work toward providing a bug-free release; the team should have ready-to-ship code at the end of any iteration (Watkins, 2009). The item with the highest priority in the subsequent iteration should be to fix any remaining defects from the previous iteration (Stober & Hansmann, 2010). Limited time to release could be a major factor that motivates the team to defer testing to the next release: the closer the release date, the greater the pressure on the team members (Humble & Farley, 2010). Introducing automated testing, along with continuous delivery of smaller releases, saves time and effort, and reduces the need to work under pressure and stress (Humble & Farley, 2010). Unfortunately, the automated testing was another example of the waste elimination strategies that were made during the *Deliberate Improvements* phase and left on the *Internal Implementation* phase.

2.4.4.8 Waste Taxonomy 8: Defects

In IT context, a defect is defined as the unintended behavior of a computer program or system. It is a common term that is used to describe an error, failure, or fault that produces an incorrect or unexpected result. Every system produced contains defects (Pries & Quigley, 2011). Defects are considered to be the main cause for rework, and they also are an indication of product quality (Besterfield, 2009; Upton & Staats, 2011). The elimination of defects depends upon the effectiveness and efficiency of the software V &V process.

A correlation exists between V&V and defects: the more frequent the V & V, the fewer the number of defects. Eliminating deferred V & V waste by introducing them in the early stages of a system development project will reduce the waste associated with defects. In addition, the cost of fixing the defects will be much lower if the defect is fixed at, or close to, the point where it was originally introduced. The identified waste and elimination strategies, as part the “Defects” taxonomy, were:

Inaccurate and out-of-date specifications and conflict between supervisor directions and client’s requirements. Software quality may be measured by the number of defects in customer acceptance testing (Staats & Upton, 2009). The number of defects detected depends on system specifications. Ironically, some people working in the IT service filed, do not know the difference between user requirements and system specifications. Evidence for this exists in the online programmer and developer forums.

In one example, a discussion about differences between requirements and specifications in IT has seven answers and about 4000 views; the discussion ended with no agreement on what the difference between requirements and specifications are (Programmers Forum, 2012). This discussion raised an interesting and important question: if people within the IT industry cannot differentiate between requirements and specifications, how can we expect them to develop appropriate specifications?

In short, user requirements describe what the user needs—what the system or finished product must deliver (PMI; Project Management Institute, 2008). User requirements are measured, or “validated”, against the user’s acceptance criteria (Besterfield, 2009)and answer the question, “Are we building the right product?” (Watkins & Mills, 2011). System specifications, on the other hand, are the criteria that guide the actual system development work;

By other words, they are the admissible or tolerance limits established by the project team members and supervisors (Besterfield, 2009; PMI; Project Management Institute, 2008). These criteria are used to determine the quality of the product under development by means of the verification process and answer the question, “Are we building the product, right?” (Watkins & Mills, 2011).

The directions from supervisors should contribute to the specifications while the customers requirements should be determined in collaboration with the client, in case of a conflict, the issue should be discussed with the stakeholder (supervisor and customer) to reach a consensus, if no agreement has been achieved, the matter should be escalated to the higher level.

For example, a customer requirement was to have the ability to maintain security groups within a specific system. However, the security specifications indicated that only system administrators were privileged to this functionality amongst other configuration privileges that should be retained within ORGUS. After negotiation, the customer accepted to pay the cost of developing a customized role to manage the security groups only and banned from the other configuration privileges.

Excessive quantitative productivity standards for evaluation. The employee performance appraisal system in ORGUS contributed to creating defects that required rework. The employee was being measured against such criteria as the number of tickets/problems resolved, the number of telephone calls answered, the number of lines of code written, the number of pieces of equipment/hardware repaired, it was evident that every criterion asked for volume, thereby promoting a focus on quantity over quality. Changing the evaluation process to one that focuses on quality of employee performance must start by understanding the employees, their experience, and abilities (McGovern & Shelly, 2008)

Senior management in ORGUS established a new approach for performance evaluations and introduced one-on-one (“O3”) meetings between the employee and the supervisor to build stronger relationships and remove barriers. Supervisors started to have a better understanding of their employees’ experience and abilities.

Questions that are asked, as part of the performance appraisal process, have been restructured to focus on how well the work was done and how the results could be improved. The criterion for performance evaluation has changed to address such things as how cost can be reduced while increasing the quality of the service. As well, customer evaluations for the services that are provided were implemented to assess the employees’ qualitative performance.

Unclear customer requirements, resulting in defective software. The root causes of this waste, specifically, the lack of customer involvement resulting in inappropriate assumptions; this issue was addressed previously in Section 2.4.4.3. This is just an example of how the elimination of root causes wastes (lack of customer involvement and inappropriate assumptions) lead to eliminating the subsequent waste (unclear customer requirement, resulting in defective software).

2.4.4.9 Waste Taxonomy 9: Outdated information / Obsolete Working Version

Traditional approaches to software development do not typically incorporate the continuous delivery philosophy. Developers are accustomed to having a long development cycle time; the problems associated with using outdated information or out-of-date working versions, therefore, the only surface when the developers start to commit their changes and integration testing start (Humble & Farley, 2010). The waste and elimination strategies that we identified as part of the "Outdated information / Obsolete working versions" waste classification, were:

Developers do not integrate frequently enough. Unexpected defects are usually discovered because of the recently committed work by other developers since the last retrieval of the working version or because of changes to the environmental settings amended to the latest available working version used by the developer. This requires rework on the changes made by the developer to fix the breakage to the trunk – the parent working version – this might be a never-ending cycle if the developers do not change their mindset and start operating in a continuous development environment.

We recommended the use of a version control system and a continuous integration server for the software development team, and we increased the organization's awareness of the importance of using a continuous delivery philosophy. By reviewing the version control commit log messages, the developers showed a higher keenness towards committing their changes, which has reached 4 commits per day per developer, which in turn helped in reducing the defect rate from 14 bugs per thousand lines of code (KLOC) to 8 bugs per KLOC.

Missing, inaccurate, or incomplete documentation for complex systems. During the period of time that services were transitioning in ORGUS, we identified the deliverables that were necessary for the new team(s) to run the service efficiently. These included configuration documents, training, and service and system documentation. Unfortunately, based on a survey questionnaire with more than 50 participants, we found that about 70 percent of the services and systems lacked adequate and up-to-date documentation. The survey included questions, about documentation structure, clarity, tone, consistency, terminologies and unambiguity, the questions included:

1) Does the documentation include a table of contents or indexes?
2) Is the documentation structure simple and clear?
3) Does it include titles and subtitles?
4) Does it indicate where the physical components are located? (Datacenter location, physical server location, cabinet and rack number, etc.)
5) Does it indicate how to get access? (How to get credentials, IP addresses, etc.)

6) Are all links clear enough to tell where they will take the user?
7) Are the links functional? Or are they broken and not found?
8) Does the documentation provide clear, numbered step-by-step instructions?
9) Are technical terms used consistently?
10) Are sentences short and straightforward? Can non-native speakers understand them?

The Lean philosophy "Do it right the first time" (Dennis, 2007; Hino, 2002; Womack & Jones, 2003) has a flip side of "Do it right, or do not do it at all!". Since one of the pre-transition requirements was to provide full and up-to-date documentation of the service under transition, the schedule, cost, and effort for the transitioning project were revised to include the development of the documentation to the project scope.

Lack of documentation raised another issue. What would happen if the current employee responsible for the IT service were to leave ORGUS? This was the main concern for those services that were more complex, and that lacked proper documentation: in fact, we identified some services that ORGUS cannot provide, if the employee who runs them leaves. In order to mitigate this risk, the employees responsible for running these services were required to train at least two additional employees to provide these services. knowledge sharing strategies could have also been applied.

Distributing reports too early, with information that is quickly obsolete. The new reporting structure and template were helpful in eliminating the waste associated with this category. As well, different types of weekly reports are now being populated automatically based on the stakeholders' pre-identified requirements. The use of a shared collaborative document environment meant that reports no longer had to be sent as attachments by email. Stakeholders can now access up-to-date project reports at any time.

2.4.5 Performance Observations from the Lean initiative

As part of the *Implement and Observe Improvements* phase to validate the results of the study, the investigators assessed the performance of ORGUS using a Business Intelligence (BI) reporting and analysis tool integrated with the ticketing system. Furthermore, customers' satisfaction survey was also considered to be the base for the improvements measurements.

For the *Request for Change (RFC)* type of services, this study included 1,684 requests for change between January 2012 and December 2012. The lead-time dropped from an average of 58 days, and 59 days for the first and second quarters, respectively, to an average of 33 and 23 days, for the third and fourth quarters, respectively. On average, ORGUS reduced lead-time by 60 percent and achieved the targeted level performance of 30 days lead-time. Figure 2.7 illustrates the realized improvements progress.

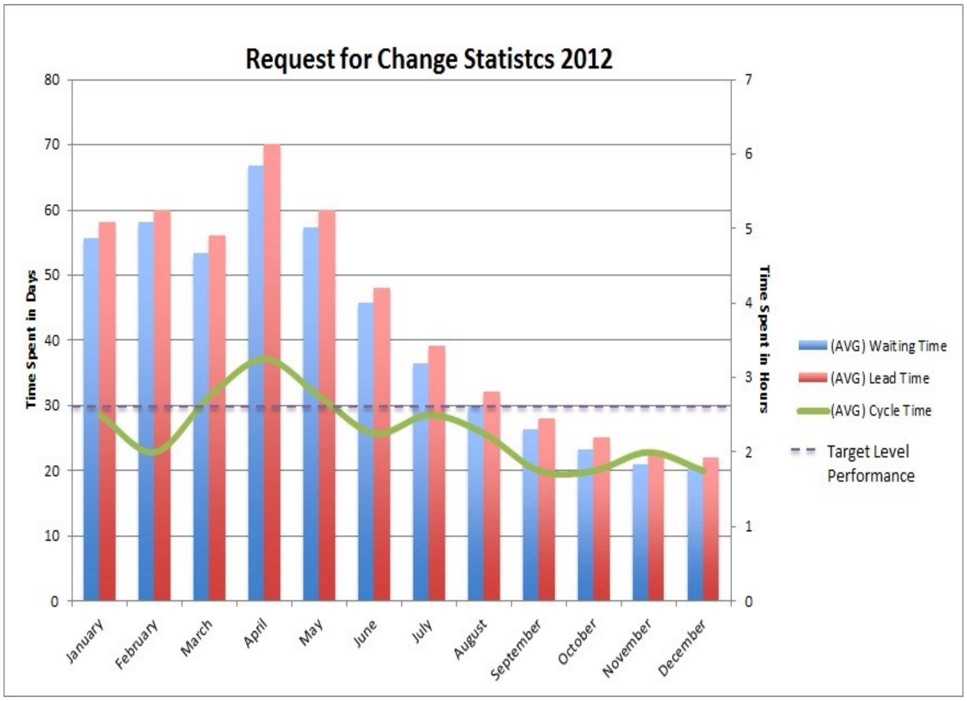

Figure 2.7 – Request for Change Improvements

For the *Service Request (SR)* type of services, this study included 67,546 service requests between January 2012 and December 2012. The lead-time dropped from an average of 5.5 hours and 6 hours for the first and second quarters, respectively, to an average of 3.3 hours and 2.6 hours for the third and fourth quarters, respectively. On average, ORGUS reduced its lead time by 56 percent and achieved the targeted level performance of 3 hours lead-time. Figure 2.8 illustrates the improvements achieved.

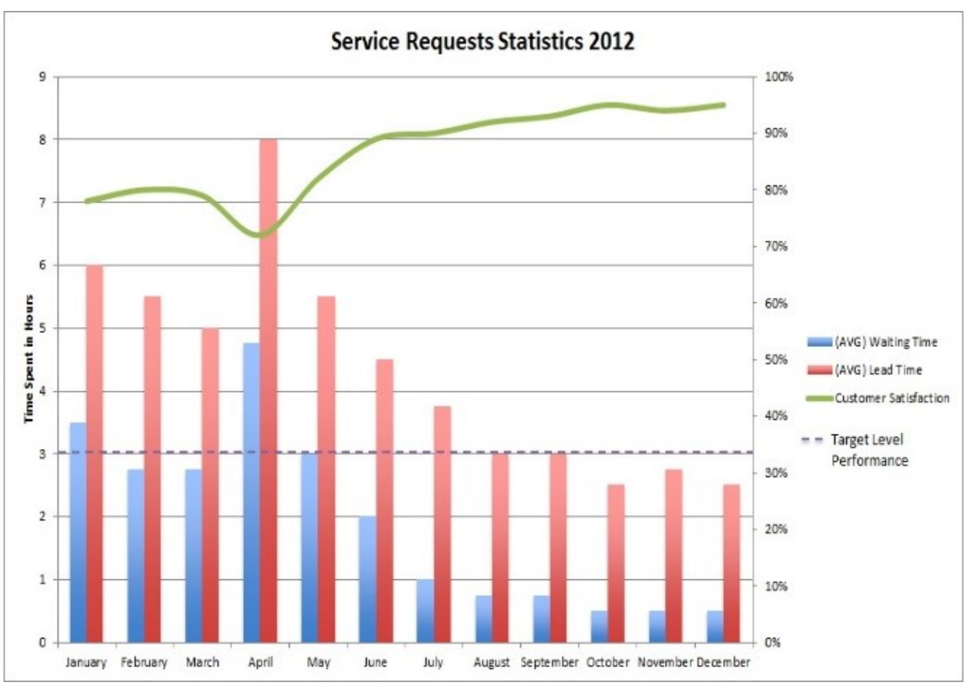

Figure 2.8 – Service Request Improvements

In addition, the customer surveys showed an increment in the level of satisfaction throughout the year. On average, the customer satisfaction rate increased from an average of 79 percent and 81 percent for the first and second quarters, respectively, to an average of 91.7 percent and 94.7 for the third and fourth quarters, respectively. The number of participants in the survey was 7,654 participants. The survey consisted of three simple questions:

1) Are you satisfied with the service(s) offered?
2) Did ORGUS provide the service(s) in a timely manner?
3) In the future, how can ORGUS serve you better? (Comments, suggestions, thoughts?)

Surveys were sent only once at the end of the month in which the service(s) were provided. For instance, if customer ABC submitted 24 SRs during the month of July, assume that the 18 SRs were closed during the same month, while the remaining 6 SRs were closed in August, the customer would receive only one survey for the SRs that were closed in July and only one survey for SRs that were closed in August.

2.5 Validity Threats

Care needs to be taken when interpreting figures 2.7 and 2.8, as clearly no causal link can be guaranteed to exist between the Lean initiative and the performance measures. Nor can we discount the impact of confounding variables or local factors, such as the skills of the new hires or the improvements realized by other projects. These factors could have significant impacts on the presented results, which comprises a threat to the *Internal Validity*.

This study summarizes 18 months of work in one company, studying its culture, practices, policies and most importantly, its employees (*study participants*), the *External Validity* threat to our study is that our observations on this company will not generalize to other settings. In an effort to reduce the *Reliability* threat, the research study was undertaken by two researchers who made independent observations, as such the possibility of having incorrect results full of bias was reduced, however, the possibility of having a biased interpretation of observations still a valid threat.

In an attempt to reduce the *Construct Validity* threat, field studies data collection techniques were used during the three phases of the *Action Research Cycle* to triangulate data in order to limit data misinterpretation and to validate the findings (Y Dittrich & John, 2007; Easterbrook & Singer, 2008). The investigators had written the observations, and asked 35 observed participants to read and validate them. *Direct techniques* were also applied, which included semi-structured interviews, but mainly observational: *Think-aloud* and *participant observer* protocols (Singer, Sim, & Lethbridge, 2008);

As per ORGUS' policies, all meetings and group activity sessions should be video recorded for future references; the researchers had been given access to review the video recording, which could be categorized under Fly on the wall, indirect source of data (Singer et al., 2008). The independent techniques that were employed to collect data included analysis of tool logs and documentation analysis (Singer et al., 2008). The application of these techniques to the data triangulation from multiple resources has strengthened the validity of the study.

2.6 Conclusions

Previous research has created IT-specific terms and phrases to represent the seven types of waste that have been identified in manufacturing environments. For example, the waste "Over-processing" in manufacturing is called "extra development steps" in an IT environment. Similarly, the manufacturing term "inventory" is simply translated into "requirements" (Schyff, 2011). Unfortunately, the terms reflect a mapping of terms from manufacturing environment rather than being discovered from an IT environment.

To understand what the wastes in the IT industry are, research is needed to analyze the activities within IT operations. A Lean initiative is not a "one size fits all" project. The following table provides a summary of the proposed classification model with related examples identified in ORGUS; some examples are followed by notes to explain the rationale behind classifying the identified waste under a specific proposed classification.

Table 2.3– Summary of the proposed classification model with examples

ID	Classification	Identified Waste
1.1	**Gold Plating**	Unnecessary tools, technologies, or methodologies
1.2		Producing and distributing reports that are not required, and providing more information than necessary
1.3		Unnecessary cosmetic functionality designed and developed into software
1.4		Striving to design perfect systems, rather than designing systems that simply are adequate
2.1	**Over-specifications**	Excessive analysis that does not support decision-making
2.2		Searching for information that is hard to find
2.3		Overdesign of software or application
3.1	**Lack of Customer Involvement and Inappropriate Assumptions**	Misunderstanding of user requirements
3.2		An inflexible and lengthy approval process that encourages workarounds.
3.3		Work performed earlier than needed because of the absence of prioritization and dependencies (result of assumptions)
3.4		Lost information, thus, developer starts to make assumptions
3.5		Limited access to the customer
3.6		Onsite visits to solve problems that could be solved remotely

4.1	**Double handling / Duplicate Processes**	Providing the same services by more than one team
4.2		Having redundant Data in the same or different systems
4.3		Multitasking, in which people move between incomplete projects because of the absence of project prioritization
4.4		Repetitive, unnecessary, and ineffective meetings
4.5		Troubleshooting to solve single problem instead of root cause
4.6		Focusing on team optimization, not overall departmental performance
5.1	**Centralized Decision Making**	Unnecessary approval processes and unclear responsibility and authority
5.2		Complex reporting systems that add extra effort without adding value
5.3		Making decisions that lock in resources too early and produce inflexibility
5.4		Lengthy approval processes that discourage innovation and creativity
5.5		Lack of delegation to empower the employees resulting in technical decisions made by non-technical people
6.1	**Waiting**	Waiting for review and approval on action items
6.2		Waiting for information that is needed complete a task
6.3		Using shared resources; waiting to assign specialists to accomplish tasks
6.4		Developer waits for testing to finish before starting work on the next piece of code
6.5		Waiting for a physical signature
6.6		Waiting to identify the team for providing problem resolution
6.7		Clients waiting on the phone for their calls to be transferred to the appropriate team
7.1	**Deferred Verification and Validation**	Lack of system integration between legacy systems (Waterfall instead of Agile)
7.2		Excessive standards to follow that encourage avoiding testing
7.3		Not enough training on, or awareness of, the importance of testing
7.4		Insufficient funds for testing based on budget underestimation
7.5		Limited time to release

8.1	**Defects**	Inaccurate and obsolete specifications and conflict between supervisor directions and client's requirements
8.2		Excessive quantitative productivity standards for evaluation
8.3		Unclear customer requirements, resulting in defective software
9.1	**Outdated information / Obsolete Working Version**	Developers do not integrate frequently enough
9.2		Missing, inaccurate, or incomplete documentation for complex systems
9.3		Distributing reports too early, with information that is quickly obsolete

Waste identification and elimination strategies may be done in an iterative fashion. "Thinking lean" when applying the Lean principles is the smartest and shortest path for a successful Lean journey. Making use of proven methodologies and techniques to eliminate waste is a Lean way to implement Lean; for instance, employing continuous and incremental improvements have proven to be effective in reducing cost and increasing quality (Kai Petersen & Wohlin, 2009). A successful Lean initiative needs senior management support and employee buy-in since a waste identification process will trigger the need to start new projects and these, in turn, will trigger significant changes at all levels of the organization. Senior management should be aware that to successfully implement Lean, a significant investment may be required to carry out different projects initiated throughout the Lean journey.

The following table provides a summary of the possible solutions that were developed during the *Deliberate Improvements* phase along with the decision that was taken either to move to *Implement and Observe Improvements* phase or just to leave the recommendations in the *Deliberate Improvements* phase for future considerations based upon the resources available in ORGUS:

Table 2.4– Summary of the proposed techniques to eliminate wastes

ID	Possible Solution	Phase
1.1	Using SIPOC-Flowchart and MoSCoW Prioritization	I
1.2	Creating a report template that contains the information that the recipients deem to be necessary	**I**
1.3	Introducing Kanban board and Stand-up meetings	**I**
1.4	Implementing *Double-loop learning* and Stand-up meetings	**I**
2.1	Implementing *Double-loop learning,* Stand-up meetings, and reporting	**I**
2.2	Stating all assumptions and document them as potential risks	**I**

2.3	Establishing customer involvement/feedback model	**D**
3.1	Hiring competent business analysts and customer feedback	**I**
3.2	Changing the *organizational defensive routines* (Argyris, 2004)	**I**
3.3	Establishing prioritization process i.e. MoSCoW prioritization	**I**
3.4	Using backup, version control, and document management systems – in ORGUS, tools were existed but not utilized!	**I**
3.5	Establishing customer involvement/feedback model	**D**
3.6	Using remote access software tools	**I**
4.1	Establishing SPOC and SQM	**I**
4.2	Establishing central data repository or document management system - in ORGUS, tools were existed but not utilized!	**I**
4.3	Establishing prioritization process i.e. MoSCoW prioritization	**I**
4.4	Developing and implementing meeting rules	**I**
4.5	Implementing root-cause analysis	**I**
4.6	Changing the *organizational defensive routines* (Argyris, 2004)	**I**
5.1	Establishing RACI matrix and Changing the *organizational defensive routines* (Argyris, 2004)	**I**
5.2	Using automated reporting tool – in ORGUS MS Excel is used	**I**
5.3	Implementing collaborative decision-making environment	**D**
5.4	Changing the *organizational defensive routines* (Argyris, 2004) to empower employees – in ORGUS, ITIL framework helped in reducing/eliminating some unnecessary approval steps	**I**
5.5	Define/Revise processes with domain experts	**I**
6.1	Changing the *organizational defensive routines* (Argyris, 2004) to empower employees – in ORGUS, ITIL framework helped in reducing/eliminating some unnecessary approval steps	**I**
6.2	Establishing document management system to create knowledge base- in ORGUS, tools were existed but not utilized! In addition, ITIL helped in creating knowledge bases as its one of the required systems in ITIL	**I**
6.3	Implementing collaborative decision-making environment	**D**
6.4	Introducing Kanban board and Stand-up meetings	**I**
6.5	Using Digital Signature	**D**

6.6	Establishing SQM	**I**
6.7	Establishing automated call messaging system	**D**
7.1	Employing the Continuous Integration technique	**D**
7.2	Establishing "self-accessed policies."	**D**
7.3	Changing the *organizational defensive routines* (Argyris, 2004)– in ORGUS, ITIL framework helped in increasing the awareness testing viability as well as ITIL enforced preparing test plans prior final approval of *Major Changes*	**I**
7.4	Changing the *organizational defensive routines* (Argyris, 2004) - PM group in ORGUS delivered a training program on costing and estimation	**I**
7.5	Employing the automated testing and continuous integration technique	**D**
8.1	Changing the *organizational defensive routines* (Argyris, 2004)- Establishing specifications that meet the quality level required by the customer and make sure that the specifications do not conflict with requirements. Empowering employees to define specifications along with establishing robust customer involvement model realize the changes.	**D**
8.2	Changing the *organizational defensive routines* (Argyris, 2004)– ORGUS changed employee's evaluation process to one that focuses on quality, not quantity, customer feedback is used now as part of the employee's annual evaluation	**I**
8.3	Establishing customer involvement/feedback model	**D**
9.1	Establishing of a version control system and a continuous integration server for the software development team	
9.2	Employing the Continuous Integration technique	**D**
9.3	Using automated reporting tool and document management system	**I**
(**D**) Deliberate Improvement (**I**) Implement and Observe Improvements		

ORGUS has had several tools and systems in place that were considered when developing the improvement strategies, which helped in reducing the cost to implement these strategies and led to maximizing senior management support. The most challenging factor we faced during the project was the changes to the *organizational defensive routines* and the changes to the employees' *theory-in-use.* The change process was not easy to implement; however, it was achievable by implementing *double-loop learning*. Showing quick wins, in order to develop employees', buy-in and management support, was a critical factor to success. We began with the small changes that would have eliminated transparent wastefulness processes to achieve large performance improvements.

Lean practitioners need to take into consideration the limitations and challenges of the organization when developing the strategies to identify and eliminate wastes. The investigators worked collaboratively with *research participants* to design and develop improvements strategies and recommendations to identify and eliminate wastes based upon the time and resources available in ORGUS. The model developed during this project could be used as starting point for IT organizations that would like to tackle a lean initiative. However, these organizations should analyze their own process to discover the wastefulness processes within their core IT operations.

3.0 Kanban approach, between Agility and Leanness

3.1 Introduction

In recent years, the lean approach to software development and its concepts have become increasingly popular. The lean approach was first applied in the manufacturing industry at Toyota, where it was founded and formulated as the Toyota Production System (TPS). The lean approach is aimed at delivering value to customers more effectively and efficiently through the process of finding and eliminating waste, which is a huge impediment to the productivity and quality offered by an organization (Liker & Hoseus, 2008; Magee, 2007)

One of the most popular principles of the lean approach is Kanban, which is a tool for controlling the logistical chain from a production point of view and is a method by which just-in-time (JIT) is achieved (Ohno, 1988). Kanban is one of the two pillars in the Lean house that was developed by Toyota. Since 2003, David Anderson (2003) has attempted to tailor the Kanban system to software development, formulating the *Kanban method*, an approach to the application of incremental, evolutionary process and systems changes in organizations.

Anderson (2010; 2003) differentiates the *Kanban method* from kanban—the pull system—by capitalizing the word Kanban. He identifies five elements to the successful implementation of the Kanban method: 1) visualize the workflow, 2) limit work-in-progress (WIP), 3) manage flow, 4) make policies explicit, and 5) implement feedback loops. Anderson’s contribution to establishing a comprehensive Kanban approach is evident; however, the approach is not based on a clear framework and therefore needs further development.

The purpose of the study in this chapter is to illuminating the guiding principles and elements of the Kanban approach based upon the objective of increasing the likelihood of implementing it successfully to software product development by 1) Defining the prime elements of the Kanban approach, and 2) Outlining the benefits and drawbacks of using the Kanban approach to help IT organizations as they decide whether to apply the approach.

The findings of this systematic literature review were derived from 37 primary studies that were selected per the research scope. An analysis of these studies showed that they could be classified into four main categories: 1) Kanban as an element of lean (beyond agile), 2) Kanban as an agile methodology, 3) Hybrid of Kanban and Scrum, and 4) Explanatory Kanban. Each classification identifies several sets of elements related to the Kanban approach that scholars in different categories—and sometimes within the same category—have adopted, which added another dimension to the complexity of gaining a common understanding and a precise definition of the Kanban approach among practitioners.

Furthermore, based on the findings of this review, the investigators propose areas that need to undergo further analysis and investigations. This will ensure that software development organizations have adequate information about all that is involved in applying the Kanban approach to their software development processes.

The chapter is organized as follows: Section 3.2 provides background related to the research area and emphasizes the motivation behind this study, it also underscores the differences between this systematic review and previous systematic reviews in the domain. Section 3.3 outlines the research methodology, which takes the form of a systematic review. Section 3.4 describes the process used during the search for primary studies; it briefly outlines the sources of the primary studies as well as the search terms used during the search process. Section 3.5 describes the selection process and inclusion criteria. Section 3.6 discusses the data extraction procedures, while Section 3.7 discusses the *case survey* approach for data synthesis as well as the coding scheme mechanism.

The results of the study are given in Section 3.8, together with their analysis. Section 3.9 provides some insights into the empirically reported benefits and challenges by the primary studies. Section 3.10 provides discusses the inconsistencies and gaps that were identified in within the primary studies. Section 3.11 presents a discussion of the study, while Section 3.12 discusses proposed guidelines to implement the Kanban approach. Section 3.13, summarizes the threats to validity , and Section 3.14 provides conclusions of this chapter as well as recommendations for extending the research study.

3.2 Background and Motivation

The value that the lean approach has brought to the manufacturing industries has inspired scholars and practitioners in the domain of software engineering to start exploring and implementing the lean concepts and principles to the software development process. In 2003, Tom and Mary Poppendieck (2003) proposed key elements of lean principles and outlined the challenges to implement lean successfully in IT organizations. The main difficulty has been related to the underlying changes to the organization culture that the lean thinking brings to an organization.

Poppendieck and Poppendieck (2003), on the other hand, claimed that understanding lean thinking has helped IT organizations to achieve sustainable performance improvement. However, Shah and Ward (2007) questioned the level of understanding possessed by scholars of lean thinking in the software engineering domain. The hypothesis is that practitioners are usually simulating practices of proven approaches from other disciplines or industries to software development, where these practices are generally context-dependent and should be different from one setting to another. Instead, practitioners and scholars should understand the underlying principles in order to develop best practices around the application strategies of these principles (Al-Baik & Miller, 2014; M. Poppendieck & Poppendieck, 2003; Wang et al., 2012).

Although the Kanban approach has been further developed in the past few years, no specific practices for its implementation exist, as illustrated by the different perceptions held among scholars about the approach itself. This has created the need for more study and analysis. The Kanban approach fundamentally depends on the lean principles that have been developed over the past 60 years; yet even these principles have no clear definitions in the literature available on the subject (Al-Baik & Miller, 2014; Shah & Ward, 2003). For instance, Womack and Jones (2003) mention only 5 principles, while Liker (2004) holds that there are 14 principles.

Wang et al. (2012) put credible efforts towards identifying the lean concepts, principles, and practices, where they, on the one hand, identified five principles related to the Kanban approach based on Anderson's definition of the Kanban method. On the other hand, they identified a set of practices related to implementing these principles to software development. The application strategies of Kanban to software product development are defined as practices to implement the principle in specific settings. That is, a context-dependent.

This study has started as an attempt to shed some light on the practices of implementing the Kanban approach to software product development. The results of this study are expected to help establish knowledge of the different elements of the Kanban approach. The study also attempts to offer a first step towards developing guidelines for practitioners when implementing the Kanban approach in software development organizations. Anderson's initiative has created a growing interest in the Kanban approach and how it can be deployed and set up in the IT industry. Even though Kanban has been used for many years by Toyota in the TPS, the concept is relatively new in IT and still requires considerable study and investigation (Womack & Jones, 2003). As a result, many IT organizations have been shying away from its use, while those who are using it continue to struggle with the challenges it presents.

The scope of this review is specific to the implementation of the Kanban approach for product development processes in software development. In an attempt to develop a comprehensive Kanban approach, the investigators, therefore, define the Kanban approach like the following: *a set of concepts, principles, practices, techniques, and tools for managing the product development process with an emphasis on the continual delivery of value to customers, while promoting ongoing learning and continuous improvements.*

The scope and coverage of this systematic review differ significantly from previous reviews. For example, Dybå and Dingsøyr (2008) performed a systematic literature review of various empirical studies conducted on agile methods and lean software development up to the year 2005, and they were able to identify 36 relevant studies. Of those 36 studies, only a single study reported on the application of lean practices to software development. Further, they found that these empirical studies focused on only one development method such as XP. They also concluded that the implementation of the lean approach was only carried out on a small scale with only three papers investigating organizations with more than 50 employees.

Another systematic review conducted by Wang et al. (2012) was limited to experience reports that had been published in agile-related conferences between 2000 and 2011. The scope of Wang's review was broad and generic; it investigated the application strategies of more than 30 lean concepts, practices, and principles in agile software development. The review illustrated that organizations were mainly focusing on two lean concepts, "value" and "eliminating waste." Wang et al. (2012) reported that organizations were moving from agile processes to lean processes, and this shift has led Wang to commend the development of operational guidance for each specific lean process, which was suggested as a significant area for future research and investigation. In short, Wang et al. (2012) do not specifically address the question of "what is Kanban?" in any detail. Table 3.1 emphasizes the differences between this systematic review and the Wang et al. (2012) systematic review:

Table 3.1– Comparison with Wang et al. (2012)

Comparison element	**The Kanban Systematic Review Al-Baik & Miller (2015)**	**Leagile Systematic Review Wang et al. (2012)**
Purpose	1. Illuminating the guiding principles and elements of the Kanban approach to increase the likelihood of implementing it successfully by:	"Providing a better understanding of the application strategies of lean approaches in agile software development, and

	2. Defining the prime elements of the Kanban approach, and 3. Outlining the benefits and challenges of using the approach within IT organizations.	Demonstrating how these strategies are being implemented in practice" Wang et al. (2012, p. 1288)
Years included	1990 – 2013	2000 – 2011
Scope	Specific to the Kanban approach	More than 30 different lean concepts, principles, and practices
Sources of primary studies	Journal articles, and Grey Literature, including conference proceedings and digital libraries Scopus, Scirus, Google Scholar, ACM Digital Library, IEEE Xplore, Science Direct (Elsevier), Web of Science, InterScience (Wiley), Science.gov, and InfoQ.com	Limited to experience reports that were published in agile-related conferences "XP Conference series, Agile Conference series, and XP/Agile Universe series" Wang et al. (2012)

The results have been analyzed using the broad lean perspective relating to lean product development rather than lean software development. The systematic review approach of this study will also provide software development teams with an assessment of the methodological quality, and the strength of the evidence given will deliver the necessary information prior to the adoption of the Kanban approach.

3.3 Research Methodology

This research study has been undertaken by following the guidelines produced by Kitchenham and Charters (2007). The study has been conducted in three phases, 1) Planning the review, 2) Conducting the review, and 3) Reporting the review. Each phase has specific steps and outcomes, which are outlined in Table 3.2.

Table 3.2– Steps and Artifacts of Systematic Literature Review

Phase	**Steps**	**Artifact Outcomes**
Planning Phase	- Identifying the objectives of the review - Formulating the research questions - Developing and Evaluating the review protocol	- Review Protocol document - Data Extraction Form
Conducting Phase	- Developing a *search strategy* and *search strings* - Identify sources of *primary studies*	- *Search Strategy* document - *Quality Instrument*

	- Assessing quality of, and selecting *primary studies* - Extracting data from the selected *primary studies* - Synthesizing the extracted data	- *Data Extraction* form (revised) - Result summary
Reporting Phase	- Presenting the results - Writing and formatting the systematic review - Evaluating the systematic review	- *Systematic Review* report

3.3.1 Objectives of Systematic Literature Review

The purpose of this paper is to illuminate the guiding principles and elements of the Kanban approach to increase the likelihood of implementing it successfully by defining the principal elements of Kanban and outlining the advantages and benefits of using it within IT organizations. The paper uses a systematic review methodology to look at the associated literature on the Kanban approach and its application in industrial I.T. settings. There is a need to empirically show the advantages that the Kanban approach can bring to IT organizations, which may increase confidence in the approach that in turn may lead to its wider adoption.

As recommended by Kitchenham and Charters (2007), the research group has framed the research questions based upon Petticrew and Roberts's (2006) criteria (*Population*, *Intervention*, *Comparison*, *Outcomes*, and *Context*) ***PICOC***. The *population* for our study is the application area, of the Kanban approach, that is the IT settings for software product development; the *intervention* for this research is the implementation of the Kanban approach in software product development, while the main *comparison* is with agile methodologies in general and Scrum in particular.

The *outcome* of this study aims to illuminate the guiding principles to implement the Kanban approach in IT organizations to reduce both, cost and product cycle time, while increasing both, product quality and customer satisfaction. The *context* of this study targets industrial IT settings for software product development without a restriction on the size. The industrial settings might not be a full empirical study in one particular organization or project; for instance, Poppendieck and Poppendieck (2003) use several examples from diverse projects in different organizations throughout their book. The primary questions of this study are as follows:

RQ.1. What elements of the Kanban approach are being discussed by the available literature as providing evidence of perceived benefits?
RQ.2. What are the perceived benefits and challenges of using the Kanban approach, according to the available literature?
RQ.3. Are the traditional guiding principles of the Kanban elements by the available literature well defined and applicable?
RQ.4. Are agile techniques being relabeled as elements of the Kanban approach?
The first research question seeks to investigate Kanban's primary elements that the past studies have covered. It answers the following questions:

RQ 1.1 What are the principal elements of the Kanban approach that have been discussed by the available literature?

RQ 1.2 What are the relationships between the Kanban key elements and the five core pillars of lean thinking?

Womack & Jones (2003) defined the five core pillars of Lean Thinking as:

1) Value: specifically, the value from the customer's point of view, and value should be expressed in terms of a specific product or service.
2) *Value Stream*: identify the set of actions or steps required to transform the *value* from concept into finished product or service in the customer's hand.
3) *Flow*: once the *value stream* is well specified, and the wastefulness steps have been eliminated, making the *value*-adding actions, or steps, *flow* without interruption or delay.
4) *Pull*: when the *flow* is realized, a period time required to transform a concept into a finished *value* (product or service). Hence, the *value* has to be only introduced per the customer's demand. The *value stream* is triggered when the client pulls the *value* when needed.
5) *Perfection*: when an organization implements the previous four principles, it realizes that there is no end for continuous improvement to seek *perfection.*

The second research question (RQ.2) aims to investigate the perceived benefits and challenges that the available studies have reported. It answers the following questions:

RQ 2.1 What are the perceived benefits of using the Kanban approach, according to the available literature?

RQ 2.2 What are the perceived challenges of using the Kanban approach, according to the available literature?

The third research question (RQ.3) investigates the clarity and applicability of the established guiding principles as they have been undertaken by past studies to lead the Kanban initiatives in IT organizations. It answers the following question:

RQ 3.1 Have the practitioners and scholars distinguished between the Kanban (capital K) method as a comprehensive change catalyst for software development and the kanban system (small k) as a pull system?

Finally, the fourth research question (RQ.4) assesses the relevance of results that have been reported in the past studies by examining whether the agile-related techniques were just relabeled as being the Kanban techniques. It considers the following question:

RQ 4.1 Are Scrum boards being relabeled as Kanban boards?

3.4 Search for Primary Studies

In an attempt to perform a comprehensive scholarly and scientific search, and to minimize the bias into this systematic review, the search for *primary studies* was not only limited to published journal article, but also targeted *Grey Literature,* including conference proceedings, digital libraries, and theses *as recommended by* York University, Centre for Reviews and Dissemination (CRD) Database of Abstract of Review of Effects (DARE).

The research has a planned, systematic approach and specific sources to search for the *grey literature* to overcome the difficulties associated with finding articles to make it easier for the reader to verify the resources used in this research. The research contributions are limited from the year 1990 onwards, as Womack, Jones, & Roos (1990) introduced the concept of Lean in 1990. The search was only applied to abstracts, keywords, and titles.

3.4.1 Sources of Primary Studies

As recommended by Kitchenham & Charters (2007), a combination of automatic search and manual search was conducted to minimize the threat of missing relevant studies. The Kanban approach and Lean have been used by IT organizations in different business domains, including but not limited to: IT Retailing, IT Quality Management, and Software Development; thus, the related studies may be located in any IT related source or digital library; hence, the need has become apparent to search different electronic sources and digital libraries, as there is no one authoritative source covering all the IT fields. In an attempt to cover as many related sources as possible, the search covers the digital libraries and data sources as recommended by Brereton, Kichenham, Budgen, Turner, & Khalil (2007) in Kitchenham & Charters (2007). Table 3.3 provides a list of sources, which have been electronically searched to find *primary studies*

Table 3.3– Sources of Primary Studies

Source	Rationale	Liter. Type
Scopus	Claims to be the world's largest database of abstracts and citations, with 49 million records including 5.3 conference papers	Monographic, Grey
Scirus	Claims to be the most comprehensive scientific search tool, with 545 million items including repositories and web information	Monographic, Grey
Google Scholar	Indexes the text of scholarly literature across many disciplines, and sources. It covers theses, books, and publications from professional societies and online repositories	Monographic, Grey
ACM Digital Library	Claims to be world's largest education and scientific computing society, it covers books, periodicals, reports, and theses	Monographic
IEEE Xplore	A major resource for scientific and technical content. It claims that it expands its database by 25,000 new documents each month	Monographic
Science Direct (Elsevier)	Claims to contain more than 11 million full-text articles from more than 2,500 journals and 11,000 books	Monographic
Web of Science	Provides access to cross disciplines, and multiple databases. It provides a citation index to reference any other literature referencing the work of the searched study.	Monographic
InterScience (Wiley)	Claims to be the largest publisher for professional and scholarly societies. Provides access to over 4 million articles, from 1,500 journals, 9,000 books, and reference works and databases.	Monographic
Science.gov	A gateway to over 2,100 scientific websites, provides search of over 55 scientific databases and 200 million pages of science information and research results	Grey
InfoQ	An online practitioner-driven community providing news, articles, presentations, and interviews to spread the knowledge and innovation in the software development community	Grey

Specific journals, such as "Empirical Software Engineering," "The Journal of Systems and Software" and "The Information and Software Technology Journal," were searched manually, as they are known to have either empirical studies or literature reviews and have been used by other *Software Engineering* systematic reviews. The full list of the manually searched journals and conference proceedings has been declared in the *research protocol*.

3.4.2 Search Strings

A list of terms was created earlier during the development of the *research protocol*; the list has been used to develop the search strings. The terms have been combined using the Boolean expressions (AND, OR) while being searched so as to reduce the number of irrelevant results that might not relate to the research area. That is, Google Scholar returned 32,200 results for searching (Kanban), 15,400 results for (Kanban AND Software) and 2,290 results for (Kanban AND "Software Development"). This is due to the fact that individually, the terms are also applicable to other disciplines. The search strings were constructed based upon the structured research questions by following the *PICOC* method (see section 3.3.1), major synonyms for each element were combined using the *Boolean* expression *OR,* then the final research string was assembled using the *Boolean* expression *AND* as shown in Table 4.

Table 3.4– Search Strings

Element	Monographic	Systematic Literature
Population	"Software Engineering" OR "Software development" OR "Information technology" OR "Computer Science" OR "IT project management" OR "Software project management" OR "Information technology project management" OR "Software product development" OR "IT Product development."	"Software Engineering" OR "Software development" OR "Information technology" OR "Computer Science" OR "IT project management" OR "Software project management" OR "Information technology project management" OR "Software product development" OR "IT Product development."
Intervention	Kanban OR Lean OR "Continuous flow" OR "Pull system" OR "Lead time" OR "Cycle time" OR "Work in progress" OR "Work in process" OR WIP	Kanban OR Lean OR "Continuous flow" OR "Pull system" OR "Lead time" OR "Cycle time" OR "Work in progress" OR "Work in process" OR WIP
Comparison	Agile OR Agility OR Scrum OR Leagile OR Scrumban	Agile OR Agility OR Scrum OR Leagile OR Scrumban
Outcome	"Process improvement" OR "Cost reduction" OR "Reducing Cost" OR "Reduce cost" OR "High quality" OR "Higher quality" OR "Quality improvement" OR "Improving quality" OR "Improve quality."	"Process improvement" OR "Cost reduction" OR "Reducing Cost" OR "Reduce cost" OR "High quality" OR "Higher quality" OR "Quality improvement" OR "Improving quality" OR "Improve quality."
Context	Empirical OR Experiment OR "Evidence-based" OR "Industrial setting" OR "Case study" OR "Action research."	"Literature Review" OR Overview OR "Research review" OR "Research synthesis" OR "Research integration" OR "Systematic review" OR "Integrative research review" OR "Integrative review."

The strategy for constructing the search strings was unified regardless of literature type, (*Monographic Literature*, *Grey Literature*, or *Systematic Literature Review*); however, the search terms were not the same. For instance, the *context* to search for primary studies in the *monographic literature* was composed of alternative terms to IT industrial settings or empirical evidence; however, the *context* to locate *systematic reviews* consisted of synonyms given to the systematic review as defined by Biolchini, Mian, Natali, and Travassos (2005). The investigators have also examined the cited resources in the primary studies and reviewed these citations to include the studies that satisfy the inclusion criteria within the scope of the research area.

3.5 Selection of Primary Studies

The selection of candidate primary studies was derived from the inclusion criteria that had been defined in the research protocol. Due to the enormous number of candidate studies and the limited time and resources available, the screening process consisted of three main steps. First, depending on the inclusion criteria as the sole determinant to whether to include or to exclude the paper, the authors screened the titles and keywords to determine the relevance to the study area. In case of uncertainties, the second step would be triggered, which involved screening the abstract of the paper, after which the investigators included or excluded the paper. If doubt still existed, step three would be carried out, reading the introduction and conclusion of the manuscript, then the investigators included or excluded the paper. The selection process that was followed is illustrated in Figure 3.1.

3.5.1 Inclusion and Exclusion Criteria

For the purpose of this research study, *exclusion criteria* have not been established; the only *exclusion criterion* was "*If the study does not meet the inclusion criteria, it must be excluded,"* the following inclusion criteria have been established for this study:

- The study should be written in English.
- The study should be published between 1990 and 2012.
- The study should include empirical evidence derived from industrial settings, regardless of research methodology espoused in the study, i.e., *exploratory* or *explanatory.*
- The study should clearly state that it has its focus on software product development through the use of the Kanban approach or Lean.
- The study should describe the elements and the method used to implement Kanban.
- If the study has been published in more than one journal/conference, the most recent version of the manuscript has to be included.

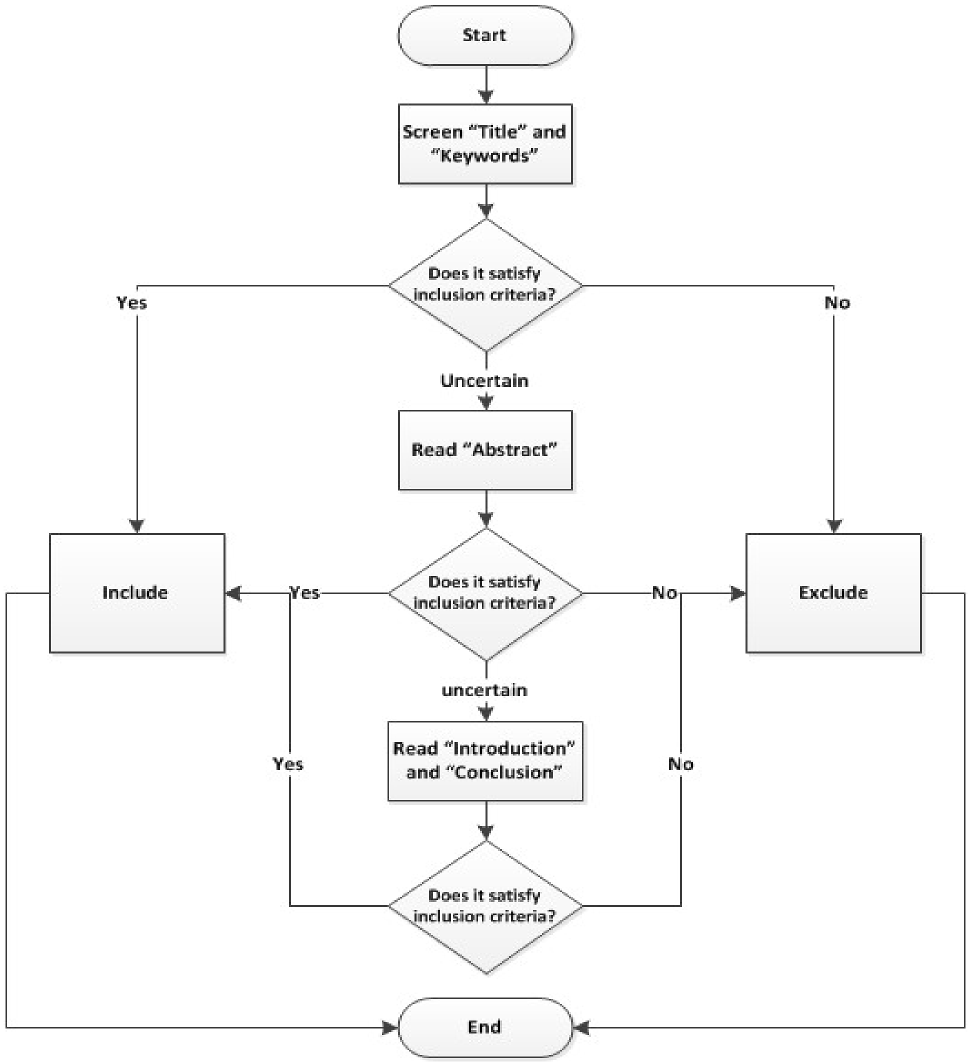

Figure 3.1– Selection Process of Primary Studies

3.5.2 Study Quality Assessment

Each of the studies that had been selected previously was subjected to a full quality assessment; the quality evaluation process was developed to accommodate the three types of literature covered by this study; *monographic literature, systematic literature review,* and *grey* literature. As recommended by Kitchenham and Charters (2007), the research investigators developed a *quality instrument* to be the basis for the quality assessment (Appendix B describes the instrument). Monographic quality was assessed by investigating the study's rigor and relevance using the guidelines proposed by Ivarsson and Gorschek (2010), while *systematic literature review* has been evaluated as per the criteria suggested by York University, Centre for Reviews and Dissemination (CRD) Database of Abstract of Review of Effects (DARE). As there are no clear criteria to assess the quality of the *grey literature*, the investigators followed a systematic approach to evaluate the quality using two attributes: 1) the number of citation and 2) the *h-index* of the author(s). The studies that had not succeeded to meet a minimum level of quality threshold were excluded (see Appendix C for the list of excluded papers).

3.5.3 Reliability of Inclusion Decisions

As the research was conducted by two investigators, disputes had occurred. The disputes were addressed scientifically by adhering to/and following the *primary study* selection process presented in Figure 3.5. Agreements between the investigators have been measured using Cohen's Kappa *(k)* method to estimate the *interrater reliability* (Banerjee, Capozzoli, McSweeney, & Sinha, 1999). Scholars have not determined the level of *interrater reliability* that must be reached to consider the reliability measurement acceptable. Banerjee et al. (1999), however, suggested that for the interpretation of kappa values *"For most purposes, values greater than 0.75 or so may be taken to represent excellent agreement beyond chance… and values between 0.40 and 0.75 may be taken to represent fair to good agreement"* (Banerjee et al., 1999, p. 604).

Hence, the research investigators decided to adopt the rigorous requirements of achieving greater than 0.75 on Kappa's measurements. The *inclusion criteria* were pretested twice, after the first pretest, the *inclusion criteria* were revised, and the pretest was conducted again on the same studies to assess the clarity of the *inclusion criteria*. The *interrater reliability* has been increased by 20 per cent as a result of resolving one disagreement between the research investigators. After the second pre-test, the criteria have been considered to be adequately reliable to proceed further. Table 3.5 shows the results of *interrater reliability* estimations.

Table 3.5 – Pretest Interrater Reliability

Pre-test	Number of Studies Included	Interrater Reliability
Pre-test 1-1	10	0.60
Pre-test 1-2	10	0.80
Pre-test 2	15	0.85

3.6 Data Extraction

The data extraction stage aims to design the extraction form, which helps the research investigators to report the information that they will gain from the *primary studies* to answer the research questions. In an attempt to reduce the bias and as recommended by Kitchenham & Charters (2007), the extraction form has been defined and piloted during the *planning the review* phase as part of the *systematic review protocol.* The pilot process has not only helped in reducing the bias but also, has given the investigators a chance to familiarize themselves with the data extraction process and to revise the form as required.

3.6.1 Data Extraction Procedures

Multiple data extraction methods were applied to ensure data extraction consistency. The methods were *test-retest* and *cross checking*, in which the primary investigator and the coauthor, respectively, performed the second extraction*, and* cross-checked twelve randomly selected *primary studies.* After applying the *inclusion criteria* on more than 3,000 studies, candidate studies for inclusion had been reduced to 54 studies that went through quality assessment (section 3.5.2). As the investigators had to read the full-text of the published studies to determine its quality, the data extraction process was undertaken simultaneously. However, for the studies from the *grey literature,* quality assessment and data extraction processes were totally separated as reading the full-text was not necessary to estimate the quality of the *primary study*.

3.7 Data Synthesis

In software engineering, data synthesis concerns with summarizing the results of the primary studies in order to help practitioners to adopt the appropriate technology (Biolchini et al., 2005). Two types of synthesis have been espoused by scholars in the field of Software Engineering, 1) *Descriptive* (qualitative) *Synthesis*, and 2) *Quantitative Synthesis*. Several methods for the synthesis are explained by Dixon-Woods, Agarwal, Young, Jones, & Sutton (2005), these methods are categorized into three broad categories:

1) *Theory-led approaches*, which are intended to find relationships between findings from different studies;
2) *Analytical approaches*, which aim to analyze the textual data reported by various studies; and
3) *Triangulation approaches*, which aim to draw more evidence by comparing qualitative and quantitative data from different studies.

For the purpose of this research and the type of data presented in the primary studies, the descriptive synthesis using the *analytical approaches* is considered the most appropriate type of the data synthesis. *Analytical approaches* include but not limited to, *grounded theory, meta-ethnography, Bayesian's meta-analysis, Miles and Huberman's data-analysis,* and *content survey*. Given the research questions of this study, the latter method is determined to be the most appropriate method to synthesize the data to answer the research questions.

3.7.1 Case Survey Approach

We employed the extended version of the *case survey* proposed by Jensen & Rodgers (2002) for categorizing and coding related data from qualitative studies for quantitative analysis. The data is summarized in a *cross-case* summary table, in which each row represents a *primary study,* and each column accounts for a specific element related to the Kanban approach, the cell entry is a color-code indicating that whether a particular study supported the given variable. The *case survey* method is a compelling method to synthesize both qualitative and quantitative evidence (Dixon-woods et al., 2005).

3.7.2 Coding Mechanism

The *coding scheme* for categorizing the components of Kanban was determined by using *reciprocal translation* suggested by Kitchenham and Charters (2007), specifically the *Meta-ethnography* (Dixon-woods et al., 2005). The investigators examined the recurring themes in the articles identified as primary studies in the current literature to draft the preliminarily *coding scheme*. Each category was well defined and communicated to the research panel (peer-reviewers' familiar with the research area) that provided feedback, and then the *coding scheme* was revised (see Appendix D for a description of the coding scheme).

Likewise, the inclusion criteria, the *coding scheme* was pretested twice, following the first pretest, the *coding scheme* was revised and the second pretest was conducted on the same studies to validate the *coding scheme* clarity and applicability. The co-author applied the *cross-checking* technique and coded fifteen of the primary studies independently using the coding scheme. The *inter-rater reliability* of the coding is shown below in Table 6.

Table 3.6– Coding Interrater Reliability

Pretest	Number of Studies Included	Interrater Reliability
Pretest 1	10	0.73
Pretest 2	10	0.86
Cross-Check	15	0.92

3.8 Results

This section is a presentation of the results of the systematic literature review derived from 37 papers that were finally selected as per the research questions specified earlier. In addition to the findings related to the Kanban approach, this section provides an insight into the results of the search process as well as the quality assessment of the *primary studies*.

3.8.1 Results of Search and Quality Assessment

The search generated 7,809 articles that were written between 1990 and 2012. Of those, 4,568 articles were identified as duplications through the use of the bibliography management software package, Mendeley, which reduced the final number to 3,242 manuscripts for screening. After applying the inclusion criteria, 3,117 articles were found to be irrelevant or written in languages other than English, which limited the number of candidate articles to 126 articles. Finally, 72 candidate studies were excluded because there was a more recent or complete copy, the full text of the article was not available, or the focus of the article was not on software or Kanban.

As a result, a total of 54 articles were left for full-text reading, quality assessment, and data extraction. After analyzing, estimating, and assessing the quality of the candidate studies, 17 were excluded for either being irrelevant or not meeting the minimum quality threshold (see Appendix C for the list of excluded studies). Hence, the primary study list has 37 studies (S1–S37; see Appendix A for the list of included studies). The distribution of the primary studies per study type (i.e., Journal, Conference, Web Article, Thesis, or Book) is illustrated in Figure 3.2.

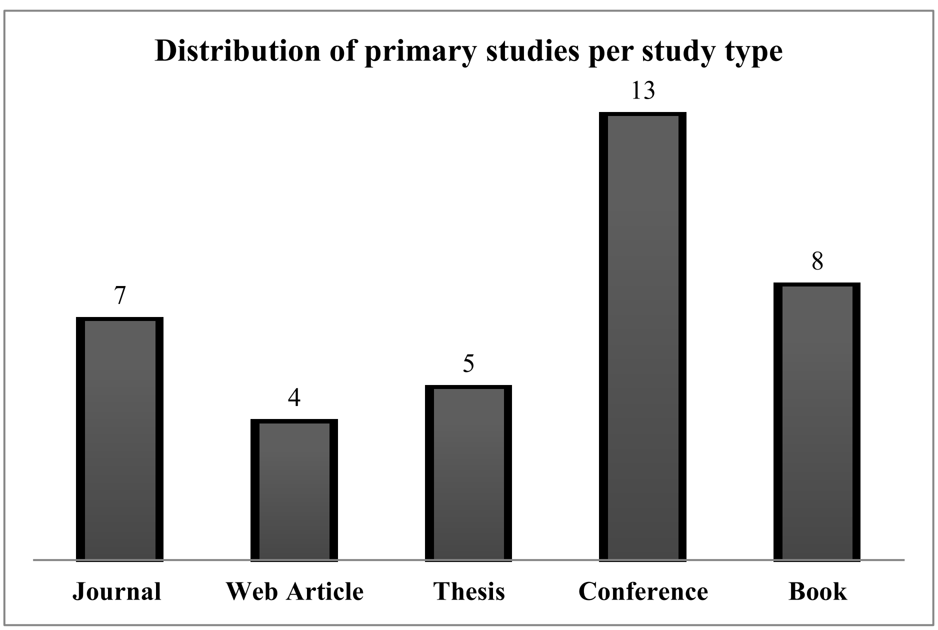

Figure 3.2– Distribution of primary studies per study type

3.8.2 Research Outcomes

Given the variety of primary study types and contexts, the primary studies were divided into four categories based upon similarities of study type and context. Table 3.7 shows the four categories; this categorization has helped in justifying the scholars' position towards each element of the Kanban approach within each category. For example, Type A scholars see *planning* as a source of waste as it does not add value from the customer's point of view; Type B scholars, however, advocate the use of *retrospective planning* to facilitate the achievement of continuous flow.

Table 3.7– Categories of Applying Kanban to Software Organizations

Type	Category Name	Category Description
A	Kanban as an element of lean (beyond agile)	This category considers Kanban one of the primary elements of lean. Kanban here is used as a starting point or a first step towards transforming the organization from agile to lean.
B	Kanban as an agile methodology	This category considers Kanban as an agile methodology. Kanban is used here to realize the agile manifesto.
C	Hybrid of Kanban and Scrum	This category is considered a combination of both Kanban and Scrum. Kanban has been used to overcome the limitations and leverage the advantages of Scrum.
D	Explanatory Kanban	This category explains Kanban and related and elements as a standalone tool without direct linkage to either agile or lean.

Figure 3.3 below shows the distribution of primary studies per category per type. The distribution revealed some critical conclusions based upon the type of the conducted research. For example, theses, in general, seek to prove specific phenomena. In this particular study, theses seek to show that Kanban extends beyond agility as a stand-alone methodology, which indicates that Kanban is not yet a well-defined methodology. On the other hand, as shown in Figure 3, conferences tend towards Type C, which means that agile practitioners and scholars are looking into enhancing the existing agile methodologies by implementing the Kanban approach.

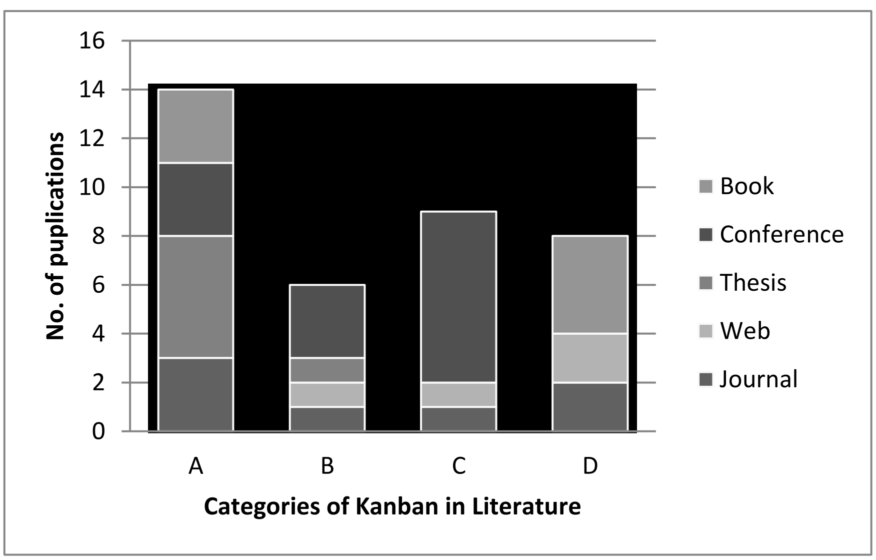

Figure 3.3 – Distribution of primary studies per category per type

Apportionment of the primary studies per category per year is illustrated in Table 8. It shows that the Kanban approach is relatively new in IT and still needs considerable research and investigation. The focus on tailoring the Kanban approach to software development was started in 2003. However, it has increased since 2009 with no clear direction regarding the elements of the Kanban approach as it offers a less rigid element as compared to agile.

Table 3.8 – Distribution of the primary studies per category per year

Year	Type A	Type B	Type C	Type D
2003	[S5]			[S23]
2007		[S2]		[S36]
2008		[S28]		
2009	[S6] [S31]		[S11]	[S35]
2010	[S14][S18][S25]		[S15]	[S19]
2011	[S7][S21][S24] [S26][S43]	[S3] [S9] [S32]	[S4] [S17] [S32]	
2012	[S22][S27][S29]	[S30]	[S10] [S8] [S12] [S36]	[S1] [S13] [S16] [S20]

3.8.3 Kanban Elements

The analysis of the 37 studies (S1–S37) provided 20 different concepts, principles, and techniques related to Kanban that has been identified as the Kanban elements. This large set of elements resulted from the various perspectives based upon the categories explained above. In Figures 3.4–3.7, the headers (C1–C20) represent the 20 elements of the Kanban approach, which are described below. A three-color scale was used to represent the 20 elements visually. The green color represents concepts that were discussed and thoroughly described; as such, we refer to them as "Discussed." The amber represents those concepts that were merely mentioned; for consistency, we refer to them as "Stated." The red represents the concepts that were not discussed at all; therefore, we refer to them as "Not Mentioned."

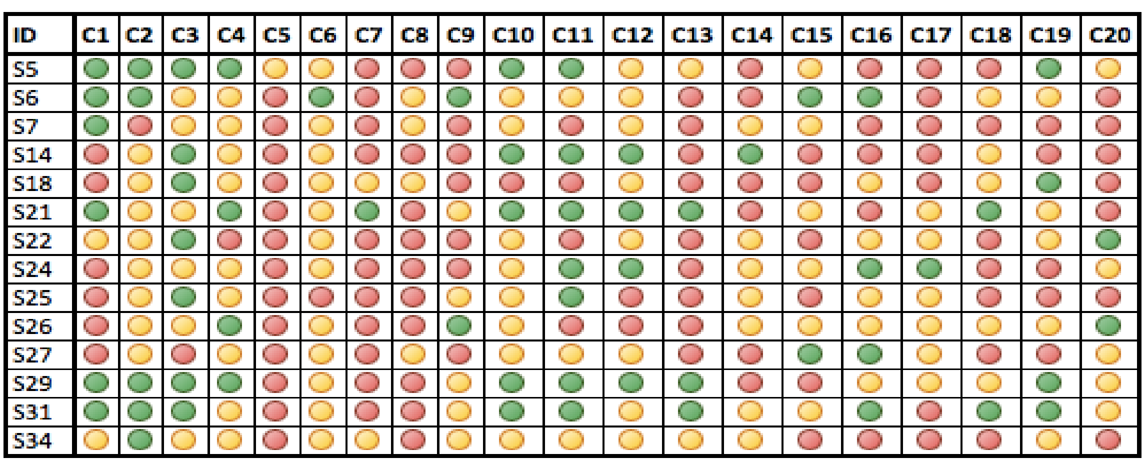

Figure 3.4 – Cross-Case summary of the Kanban elements for Type A

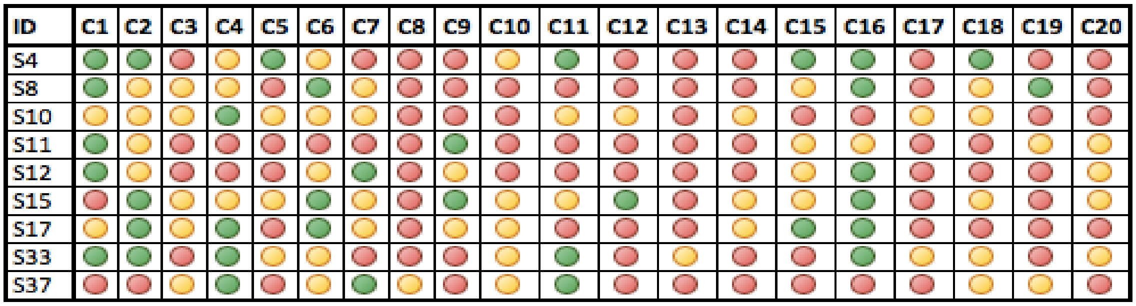

Figure 3.5 – Cross-Case summary of the Kanban elements for Type B

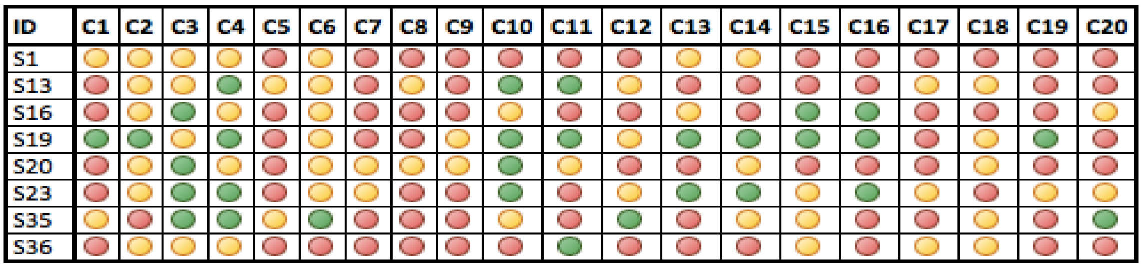

Figure 3.6– Cross-Case summary of the Kanban elements for Type C

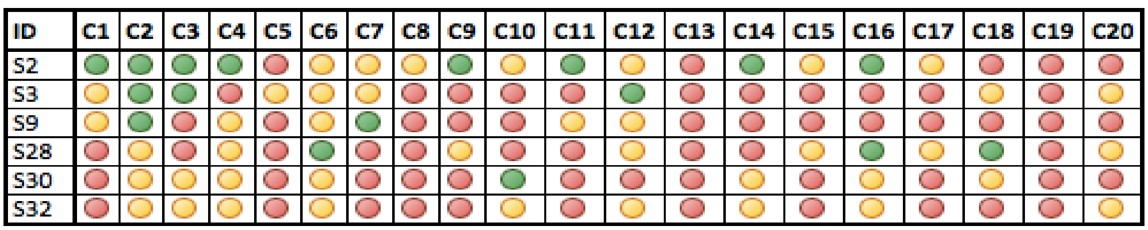

Figure 3.7– Cross-Case summary of the Kanban elements for Type D

From Figures 3.4 – 3.7, it is evident that some Kanban elements were established and discussed more thoroughly than other elements regardless of the study context and type. Scholars in a specific category discussed some elements that were not mentioned at all or were touched lightly by scholars in other categories. A summary of the 20 elements is outlined below:

C1: *Kanban method,* describes the methodology to implement Kanban in an organization. Kanban should be applied in an incremental and evolutionary fashion, the method should also be reviewed and continuously improved [S2, S4–S8, S11, S12, S19, S21, S29, S31, S33], which is referred to as "*Kaizen*" by lean practitioners.

C2: *Kanban board,* the tool to visualize the workflow. It represents the activities to be performed from product/service conceptualization until it reaches the customer's hand. Some scholars recommended a design board based upon the current activities before improving the process evolutionarily [S2, S5, S7], whilst other scholars [S3, S6, S22, S29] used the traditional waterfall approach as the basis for the board design. Still, other scholars [S12, S21, 34] used a simpler form of board design to include three main queues (i.e., To Do, In Progress, and Done).

C3: *Pull system*, the approach of starting the development process based upon a request from a customer [S5, S6]. It also requires *work-in-progress* (WIP) limit to be set out, as a work item should be pulled only if the WIP limit is not reached [S2, 29]. In addition, it is recommended to pull the highest priority items, if possible [S24].

C4: *Prioritized queue,* which refers to the list of ordered requirements or work items. Several priority criteria were set by different scholars, [S29] suggested that organizations prioritize work items based upon their value, urgency, and importance. [S2] set priority based upon the *cost of delay* or resources availability. [S26, S4] defined several queues for each process in the flow by placing a queue for each process on the Kanban board.

C5: *Inclusion criteria,* one of the major elements of the Kanban approach. These criteria guarantee that the work will be of value to the customer before being added to the board. None of the studies have attempted to define the inclusion criteria, except [S4] which shed some light on how they prioritize the work items and what were the criteria to add the work items to the waiting queue.

C6: *Work-in-progress (WIP),* the concurrent number of work items that are allowed for each process in the workflow. WIP has been identified as a major characteristic of the Kanban approach and the main objective of Kanban (Wang et al., 2012). The concept, however, has not been clearly defined. For instance, majority of the studies suggested that organizations set WIP limit by experiment [S1–S5, S7, S9, S10, S12–S14, S16, S18–S24, S26, S27, S29–S34, S37], while some studies [S11, S25, S36] did not mention WIP at all.

C7: *Done Item,* refers to when a work item is considered completed. This element should have been investigated more systematically as it is regarded as one of the major contributors to the continuous flow. Ironically, the majority of the existing primary studies [S1, S4–S7, S11, S13, S14, S16, S19, S22, S24–S26, S28–S33, S35–S37] did not mention this concept. [S9, S12, S27] claim that they have established a *done item*; however, none of the studies offered further explanation of what that definition was or how it was established. In [S17], the *done item* was established as checklists based upon code quality guidelines and standards.

C8: *Reverse item,* which is a work item that is being moved backward on the Kanban board instead of moving forward, for instance, a bug fixing work item, which needs further development. This element has not been considered a major issue as the scholars who shed some light on this element suggested that the work item flows in loops [S5, S35]; Lundgren, as cited in Norrmalm (2011) suggested putting the item back to the backlog. Hence, there is no such backward movement.

C9: *Validated learning,* which is a validation process for a feature that has been completed to measure its value from a business perspective (Ries, 2011). The scholars in [S26] suggested that monitoring and learning is one of the key performance indicators to measure the project success.

C10: *Cycle time/Lead time.* This element is arguably the determinant for the process efficiency and effectiveness as it is used in measuring the performance of Kanban. Nevertheless, Scholars have not reached an agreement on its definition [S30]. Scholars used *cycle time* and *lead time* interchangeably [S14]. Some scholars [S5, S19, S20, S30], defined cycle time as the time elapsed from work start to end, to produce a feature; however, some other scholars defined it as the time between deliveries of items [S23, S35].

C11: *Performance measurement tool*. This element describes how the performance of Kanban is measured. Some studies [S2, S5, S14] indicated the use of a *cumulative flow* diagram based upon cycle time and WIP. Other studies [S21, S25, S29] encouraged tracking the progress on a daily basis using *burn-down* charts.

C12: *Bottleneck*, a point of congestion where a processing point cannot keep pulling work items at the same pace of production from the preceding processing point. Bottlenecks can be identified by visualizing the workflow [S14, S21] or by establishing a measure to detect bottlenecks [S24] that were adopted from the *theory of constraints*. Zhang (2010) in [S18] suggested decomposing the bottlenecks into smaller chunks and realizing their expected value in increments.

C13: *Slack* or *Buffer.* The buffer is needed to determine the *waiting time* [S14]. A work item stays in the queue before being processed, to handle a process's cycle-time variation [S29, S31], and to reduce the creation of bottlenecks. An appropriate *buffer* size should be introduced in front of the bottleneck process; the buffer size should be reviewed periodically and adjusted as needed [S29].

C14: *Waste identification*. In summary, waste is anything that does not contribute to the value creation for the customer; hence, it should be removed (Al-Baik & Miller, 2014). Consensus has been reached on this element; several scholars [S2, S14, S19, S23, S24] agreed that Kanban helped in visual communication and identification of wasteful activities so that wasted energy and productivity were reduced and more energy was put towards a quick response to customer requests.

C15: *Team collaboration.* Many software development teams lack collaboration, cohesion, and cohesiveness. IT teams are still perceived to operate as islands, and bridges between technical people and business people are needed [S29]. Oostvogels (2012) in [S16] argued that the minute a team is involved in a Kanban flow, their cooperation mode begins to change. Kanban increases communications between team members themselves on the one hand, and team members and senior management on the other hand.

C16: *Meeting structure*. This element is one of the most questionabl elements. On the one hand, some scholars [S30] identified meetings as sources of waste that should be eliminated. On the other hand, other scholars [S2, S14, S24, S27, S31] recommended a daily standup meeting for 15 minutes to discuss constraints, while Kniberg (2011) in [S21] argued that each team should be given the freedom to decide meeting length and structure.

C17: *Avatars*, the visual representation of the task owner who is responsible for getting the work done. This visual demonstration makes Kanban a powerful tool in making informed decisions. The data presented on the Kanban board makes it easier to make a factual-based decision [S14, S29]. Management can get information on resource capacity and availability that helps in resource assignment and scheduling by just looking at the Kanban board.

C18: *Planning and estimation*. This element is similar to C16, as there is a contradiction of views about its importance. Several scholars have identified planning as a source of waste, and have suggested that reducing this waste will increase both, productivity and customer satisfaction. For instance, the authors of [S30] claimed that by reducing the planning activities in their sprint planning, the *lead-time* was reduced significantly. Zhang (2010) in [S18] reported that *planning* took two-thirds of the production process *lead-time*. However, Zhang (2010) in [S18] as well justified that planning and retrospective activities are necessary, as they provide an opportunity for improvement through continuous feedback from team members.

C19: *Policies,* which govern the Kanban approach, they are the dominant guidelines that outline what work should be done and how it should be done. Anderson (2003) in [S5] claimed that *policies* help in producing a healthier environment to solve disputes and reach consensus. He also argued that policies contribute to moving a discussion of problems from an emotional and subjective discussion to a rational and objective discussion. Boeg (2012) in [S29] suggested having the team members involved in developing the Kanban policies.

C20: *Feedback loop*. Getting feedback from customers is the primary goal of delivering a product through short cycles and frequent builds. Several scholars [S22, S26, S35] underlined the importance of a *feedback loop* in providing facts rather than guessing about how the customers perceive the product; however, no sufficient effort was put towards defining how Kanban would help in getting customers involved to provide their feedback.

3.9 Benefits and Challenges

The primary studies reported various benefits and challenges associated with the use of the Kanban approach, and the investigators have summarized the benefits and challenges that were reported by at least five primary studies. Most of the studies reported advantages and difficulties related to the management of the software development process as well as the software development process itself. Few studies covered the human aspect of the software development process. Other areas of software development reported by fewer than five studies included the configuration management, design, and maintenance.

Table 3.9 summarizes the benefits of using the Kanban approach to software product development. The most frequently reported benefits, which were reported by about 46 percent of the primary studies, were both, enhancing visual control to facilitate the management decisions and developing strategies for continuous improvements. While the least-reported benefit, with 16.7 percent, was related to the team development and cohesiveness.

Table 3.9– Benefits of the Kanban Approach

Benefits	Reporting Studies	% Reported Benefit
Enhancing visual control that facilitated and supported the decision-making process	[S2][S4][S10][S11][S13][S14][S15][S20][S21][S23][S24][S26][S29][S31][S35][S36][S37]	45.9%
Facilitating the coordination of cross-functional teamwork and imposing self-organization	[S1][S2][S4][S5][S10][S14][S15][S16][S17][S19][S21][S22][S33][S34]	37.8%
Empirically introducing quality circles and kaizen events	[S2][S5][S11][S14][S17][S19][S24][S25][S29][S34][S35]	29.7%
Reducing the cycle time/lead time	[S2][S4][S15][S17][S18][S22][S24][S25][S29][S32][S37]	29.7%
Increasing customer satisfaction and realizing high value	[S1][S11][S14][S15][S23][S24][S25][S26][S29][S35]	27.0%
Decreasing market and technical risks of the product	[S2][S11][S16][S18][S24][S26][S30][S34][S35]	24.3%
Developing continuous improvements strategies	[S1][S5][S7][S12][S13][S14][S15][S17][S19][21][S22][S23][S24][S27][S29][S33][S37]	45.9%

Increasing the predictability in the delivery of the final products with the constraint of changing customer requirements	[S2][S4][S11][S14][S15][S23][S24][S26][S27][S30][S32][S33] [S35]	35.1%
Ensuring team skills development and team cohesiveness	[S4][S5][S11][S15][S21][S33]	16.2%
Driving and facilitating organizational change management and culture changes	[S2][S4][S5][S8][S13][S14][S19][S20][S21][S26][S33][S34]	32.4%
Enhancing the quality of the product, indicated by decreasing defects rate, increasing quality assurance pass rate, and reducing the number of bugs.	[S11][S14][S24][S25][S30][S34]	16.2%

As reported in the available literature, the implementation and use of the Kanban approach can significantly improve the development process; however, many teams are faced with challenges when using the approach. There are issues of the team working together to deliver a software product successfully. The team should also understand the Kanban board and its different elements to ensure that they use them effectively to improve their software development process.

Table 3.10 emphasizes the challenges that accompanied the decision to use the Kanban approach. Of the primary studies, 51.4 percent perceived a challenge with the absence of guidelines for using the Kanban approach within IT settings. On the other hand, only 13.5 percent reported the use of the Kanban approach in combination with agile techniques as one of the main challenges.

Table 3.10– Challenges with the Kanban Approach

Challenges	**Reporting Studies**	**% of Reported Challenge**
Defining the measurement metric, for instance, lead time has been deemed problematic to assess the performance	[S2][S5][S6][S16][S19][S24][S29][S30][S35][S36]	27.0%
Implementing the Kanban approach requires a deeper understanding of lean concepts, principles, and practices.	[S3][S7][S10][S18][S22][S23][S25][S28]	21.6%
Unclear definitions of the Kanban prime elements are used, such as the definition of "Done."	[S7][S8][S10][S12][S15][S17][S18][S23][S32][S37]	27.0%
Guidelines to implement the Kanban approach and guiding principles on introducing the Kanban elements, such as WIP limits, are absent.	[S3][S4][S5][S6][S7][S8][S11][S12][S13][S15][S16][S17][S18][S23][S25][S27][S29][S32][S35]	51.4%
IT organizations have difficulties in digesting some elements of the Kanban approach, such as flow.	[S3][S7][S8][S10][S12][S14][S16][S24][S25][S35]	27.0%
Development teams need to be reorganized and restructured.	[S3][S4][S7][S10][S27][S33]	16.2%

Priming the Kanban approach to software development process and integrating it with existing agile techniques is complicated, expensive, and time-consuming.	[S6][S9][S10][S12] [S18]	13.5%

The dominant challenge reported by the studies is related to the absence of the guidelines on how the Kanban approach can be used in IT organizations, as well as the lack of the guiding principles on how the different Kanban elements can be introduced. The Kanban approach fundamentally depends on the lean principles that have been developed over the past 60 years; yet even these principles have no clear definitions in the available literature on the subject (Al-Baik & Miller, 2014; Shah & Ward, 2007b).

3.10 Inconsistencies and Gaps within Primary Studies

As stated previously, in an attempt to establish guidelines for the Kanban approach, Anderson has differentiated the *Kanban method* from kanban—the pull system—by capitalizing the word Kanban. Anderson distinguished between the Kanban method and the kanban pull system as follows: "*Kanban (capital K) as the evolutionary change method that utilizes a kanban (small k) pull system, visualization, and other tools to catalyze the introduction of the Lean ideas into technology development and IT operations.*" (D. Anderson, 2003, p. 6)

Anderson clearly stated that Kanban is neither a software development life cycle nor a project management methodology; Kanban is used to improve an existing process incrementally. Kanban suggests customizing the process template to suit the organization's needs; it does not recommend the adoption of a defined method or template; hence, Kanban has become a controversy in the agile software community (D. Anderson, 2010).

The analysis of the primary studies has revealed some reasons behind the dilemma that practitioners and scholars have been experiencing with the established guiding principles in the domain. In fact, a few studies have followed Anderson's definitions of small "k" and capital "K" as reported in [S6] and [S18] where they distinguished between Kanban as a change method and kanban as a pull system, Kanban was defined as a learning tool to incrementally introduce lean principles (pull, flow, and value). [S4] reported the use of Kanban to modify Scrum by eliminating the iterative development and burndown chart and introducing several prioritization queues.

Several studies ([S1][S7][S8][S14][S22][S24]) have theoretically defined kanban (small k) as the pull system to realize the continuous production flow; however, [S1][S8][S22], in fact, have reported the practices of using Kanban (capital K) as a change method. Terlecka (2012) in [S8], for example, introduced Kanban to amend Scrum by eliminating sprint planning and time boxing, which are more related to the definition of the Kanban method, not the pull system.

Another example was derived from [S1], wherein the background section of the study, kanban (small k) was defined as a pull system that signals an upstream process to start production when a downstream process is ready to accept more work. Conversely, the reported findings were more related to the Kanban method (capital K). The authors of [S1] deemed the changes imposed by Kanban on the Scrum practices as an explicit rejection of Scrum. They also considered Kanban as a stand-alone lean methodology, which contradicts Anderson's statement that Kanban is not a software development life cycle.

> *Due to the nature of the work in HR, the feasibility of implementing Scrum was rejected, including the concepts of time boxing, scheduled releases, story point estimation, sprint burndown, etc. Instead, kanban was seen a better fit for the HR department.* (Wang et al., 2012, p. 1292)

Several studies ([S3][S9][S12][15][S17][S18][S26][S27]) have agreed on three prescription rules for Kanban: 1) visualize the workflow, 2) limit WIP, and 3) measure the flow. In addition to these prescribed rules, [S21] and [S29] have added two guiding principles as predecessors for the prescriptions: 1) start by doing what you are doing right now, and 2) map the value stream. This tends more towards the Kanban (capital K). However, [S3] and [S9] have described Kanban as an agile methodology and not as a method for change and improvement, which contradicts Anderson's description of the Kanban method.

In [S9], the authors reported what they claimed was a transition from Scrum to Kanban, *"Process transition was made overnight. However, the Process Transition stage lasted for two weeks"* [p. 160]. They claimed that the goal was to improve the software development process, but they reported a big bang transition from Scrum to Kanban, which violates the Kanban rule of incremental change as set by Anderson.

The authors in [S30] also reported a transition from Scrum to Kanban, *"in 2010; the company switched from Scrum to Kanban"* [S30, p. 48]. They reported amending some changes to Scrum by introducing Kanban: *"Reduced sprint planning activities (and abandoned cross-functional teams) by the end of 2009. Two employees mentioned this relaxation of the Scrum rules as an explanation for why the lead time reduced from 2009 to 2010"* [S30, p. 52]; however, they insisted on calling it a transition, *"after replacing Scrum with Kanban, SI almost halved its lead time, reduced the number of weighted bugs by 10 percent, and improved productivity"* [S30, p. 52].

The use of small "k" and capital "K" in the manuscript of the primary studies is another point worth mentioning. In addition to Anderson, only two studies ([S16] and [S34]) distinguished between the Kanban method and the kanban pull system. Another two studies ([S15] and [S37]) interchangeably used small "k" and capital "K." Surprisingly, [S15] was forwarded by David Anderson himself.

More than 40 percent of the primary studies ([S3] [S4] [S8] [S9] [S12] [S14] [S17] [S22] [S24] [S27] [S28] [S30] [S32] [S33] [S35]) did not use small "k" in their manuscripts at all. Other studies ([S6][S23][S25][S31]) followed the formal English writing by using capital "K" at the beginning of a new sentence and small "k" in the text. The authors of [S26] made their own way of using small "k" and capital "K": the former referred to Kanban in manufacturing while the latter referred to Kanban and its elements in software development.

3.10.1 Kanban Board

The Kanban board is a tool in the Kanban approach that is used in visualizing and coordinating the work of the software development teams. It has columns that illustrate a flow of activities with sticky notes representing a work item. For every activity in the process, there are limits placed on the number of work items that will obtain an overall limited WIP (Ladas, 2009).

The Kanban board can be used to enhance communication and flow of information within an organization. Employees will pin their feedback to the board while management will be able to respond and make changes within the organization (Pink, 2011). Boeg (2012) in [S29] argued that the Kanban board is also used to illustrate issues, challenges, and bottlenecks. However, there is a poor illustration of how the Kanban board can be used to help teams in identifying these challenges and bottlenecks.

The analysis of the primary studies showed that the Kanban board is not different from a Scrum board, or Scrum task board. The primary studies have sometimes explicitly stated this finding. For example, Kniberg and Skarin (2009) in [S15] imposed the question, *"What's the difference between a Scrum board and a Kanban board?"* (Kniberg & Skarin, 2009, p. 15). They then answered the question with, *"the little 2 in the middle column on the kanban board. That's all."* The number refers to the WIP limit. A similar answer was provided by Norrmalm (2011) in [S3]:

> *In Scrum, tasks are placed on boards for time-boxed sprints. Kanban is a different way to approach this. Instead of being time-boxed, tasks are pulled at any time, only limited by a work-in-progress (WIP) limit that restricts the allowed number of tasks in every workflow state.* (Norrmalm, 2011, p. 15)

The authors of [S12] reported that the Kanban board emerged from the changes that they had made to the Scrum board, without stating these changes. Poppendieck and Cusumano (2012) in [S22] suggested: *"Kanban board might start out in a software development environment, but can easily expand to include more steps in the value stream, such as marketing and operations."* (M. Poppendieck & Cusumano, 2012, p. 31)However, it has not gone beyond being a suggestion. The explanation of what the Kanban board is and what it involves is not enough, and more research and analysis needs to be carried out.

There are no specific rules on how a Kanban board should look, although there is a proposed Kanban board layout. The Kanban board has columns that show the activities as they are organized. The columns will then hold different activities and illustrations. There is a need to clearly illustrate how the various components of the Kanban board work alongside each other and the role of the team members in ensuring that this is made possible. The study and explanation of what the Kanban board is and what it encompasses are not sufficient, and more research and analysis needs to be carried out.

3.10.2 Interrelation Between Kanban Elements and Lean Pillars

None of the scholars discussed the interrelations between the Kanban elements and the Lean pillars or the agile manifesto. As such, there is a gap in information on how the concepts interrelate as the Kanban approach is implemented in an organization. In an attempt to reduce this gap, the concepts are discussed together to give practitioners a better understanding of the relationships between the Lean five pillars and the elements of the Kanban approach. The associations were identified based upon the results of the *Thematic Analysis* that were espoused to specify the coding scheme. Two types of associations were identified; positive association and negative association, the latter was determined if at least one study reported a negative impact or disadvantage resulted from the implementation of that specific element of Kanban. Figure 3.8 below depicts these relationships.

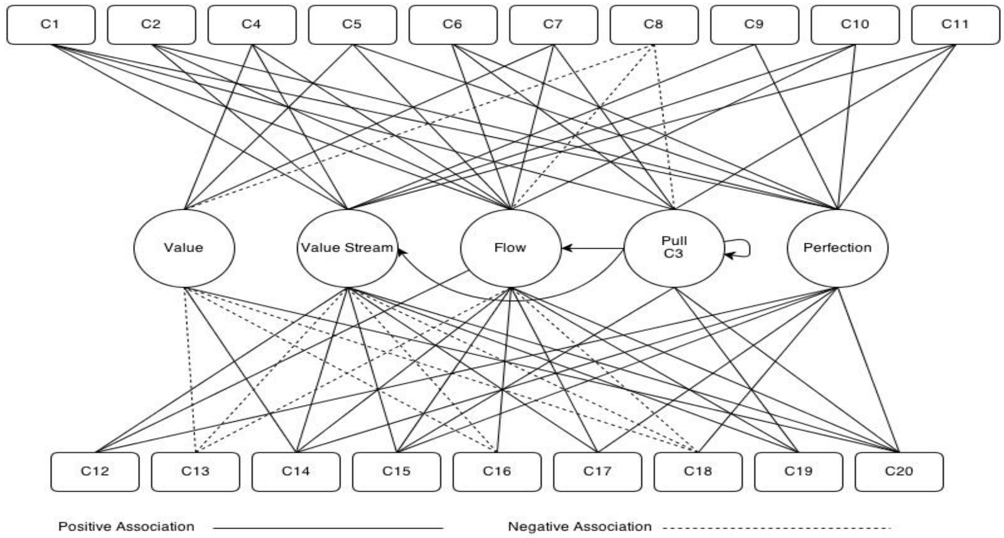

Figure 3.8 – Associations of Kanban Concepts and Lean Five Pillars

Based upon the findings that were reported by the included primary studies, the figure above shows that four out of five Lean pillars (i.e., Value, Values Stream, Flow, and Pull) may be impacted negatively by implementing any of the following elements: C8, C13, C16, or C18. Care must be taken when recognizing the necessity of applying these elements.

3.11 Discussion

Kanban is heavily influenced by Anderson (2003) work on Lean product development. Researchers have identified that it is a good fit to be used in the software development just as it has been used for maintenance and operation (Hibbs, Jewett, & Sullivan, 2009). There is a misconception that Kanban and Scrum are similar with minor differences. In actuality, Kanban and Scrum are individual concepts as Kanban acts as a change agent and the principles of Scrum can be used to optimize the Kanban workflow (M. Poppendieck & Poppendieck, 2003)

IT practitioners have started to use Kanban in the software development process as a mean to optimize the workflow, whereby they aim to reduce wastes and to add value to the product development (D. Anderson, 2010a). However, the analysis of the primary studies revealed major issues in the existing literature that limit practitioners from implementing Kanban appropriately. The issues can be classified into three broad taxonomies: 1) definition disagreement, which includes the elements that lack standard definition in the available literature, 2) vague guidelines, which consists of the items that have poor instructions and guidance on how to be designed and implemented, and 3) *contradiction and conflict*, which classifies each element that contradicts or conflicts with another element or with one of the Lean five pillars.

The classification of issues to these taxonomies was not a simple and straightforward task to accomplish as one issue could be categorized under more than one taxonomy. For instance, C10 (cycle-time/lead-time), the analysis of the available literature has shown that scholars have not reached an agreement on a standard definition for any of the terminologies, which comprises a *definition disagreement,* consequently, there were no agreed-upon instructions on how to design and measure either cycle-time or lead-time, which can be categorized as *vague guidelines*.

Likewise, *contradiction and conflict* issue was derived from the definition of *cycle-time* in [S30] in which the *cycle-time* was defined as the time elapsed from work start to end to produce a feature; the authors argued that *cycle-time* should not start from the minute a customer asks for a new feature because of two reasons, the first reason that a feature would have to wait for a while before start working on that feature, hence, waiting time in the backlog should not be counted towards the *cycle-time*; this reason contradicts with *waste identification* and violates the continuous *flow* pillar of lean. The second reason that any person other than the customer might propose a feature to be developed, which conflicts with pull, a prime element of Kanban and another pillar of lean.

One key discussion point is the Kanban board design. The analysis of the included studies showed a theme of designing the board based upon the classical waterfall approach, which was clearly noted by Lundgren (as cited in [S3]). We believe that the recommendations of scholars in [S2, S5, S7] would be more applicable by representing the current activities and improve evolutionary, which is very much aligned with lean core pillars, *value stream,* and *perfection.*

Another interesting argument is related to normalizing the work items in terms of size. Several scholars [S10, S15, S33] reported the use of cadence and team velocity to determine the amount of work that could be accomplished during a given period of time, which implies the standardization of work items size. Kniberg and Skarin (2009) in [S15] clearly reports the decomposition of work into pieces of roughly the same size based upon the Minimum Marketable Feature (MMF). On the other hand, Reinertsen (2009) in [S35] argues that spending time and efforts to reduce variances of tasks instead of handling and managing these deviations, is actually a wastefulness activity rather than a value adding activity to the product development lifecycle.

3.12 Kanban Three Core Principles and Guidelines

Based on the review of the findings, it was evident that the team members contribute a great deal to the implementation of the Kanban approach in an organization. The Kanban approach has been used by Toyota for many years now through the JIT approach. Consequently, due to its success in the TPS, IT practitioners have been using the approach as a means to optimize the workflow of the software development process whereby they aim to reduce waste and to add value to product development (D. Anderson, 2010a); however, the approach has not been fully understood. Three core principles have been identified by more than 30 percent of the primary studies, which are discussed below in greater details.

3.12.1 Visualize the Workflow

The recommendation of this paper is to start implementing the Kanban approach by visualizing the workflow; one has to start with understanding how the current system works. One of the bases of the Kanban approach is that there should be an emphasis on the current system because the approach is that of a change agent not of overturning the whole process. Several studies ([S5][S7][S21][S29]) recommended visualizing the existing development process and its activities and then modifying the board as the process is improved and wastefulness activities are eliminated.

There are different ways to visualize the workflow to improve it; one of the approaches as recommended by more than 24 percent of the primary studies ([S1] [S3] [S7] [S14] [S15] [S20] [S22] [S23] [S24]) is through value stream mapping. Value stream mapping is the process through which the team identifies all the steps in the software development process in order to make a product that is desirable to the client. The term "stream" here refers to the fact that there is a smooth, unbroken flow between varieties of steps. A continuous smooth flow where there are valuable new features and elements will lead to a desired final product.

For a value stream to result in successful software product development, it is crucial for everyone to be considered, from the customers to the developers and support engineers not merely the development team (Womack & Jones, 2003). To assess the efficiency of the value stream that composes the development process, the team can employ the element of metrics to investigate the effectiveness of the process and its various activities and tasks. The value stream should be carried out at several stages and include such elements as quarterly value stream mapping to reassess the whole value stream process (Hibbs et al., 2009).

This is an important consideration because it incorporates adding value to the various processes of the software development and ensures that the client is presented with a high-quality product (M. Poppendieck & Poppendieck, 2003). Lean thinking has been argued by several scholars to be more concerned with doing the right work at the right time rather than concentrating on who is doing the right work (Broza, 2012).

3.12.2 Limit and Visualize WIP

Following visualizing, the workflow understands and puts limits on the work in progress (WIP). Work in progress refers to the concurrent number of work items allowed in each process. The correct use of the Kanban board ensures that there is minimal WIP, which highlights the constraints and teamwork coordination. This will enable the team, as well as other stakeholders, to keep track of their development and distinguish between what is in progress and what is done in the board. This visualization gives the team insight on how the development process is moving along.

Despite the fact that several studies ([S3][S5][S9][S12][15][S17][S18][S21][S26][S27][S29]) have explicitly identified limiting WIP as one of the core elements of the Kanban approach, only 13 percent of the primary studies reported their experiences on how to set them. The value of WIP has been outlined in almost 92 percent of the primary studies; for example, Shinkle in [S6] cited a statement made by a developer during the implementation of Kanban: *WIP limits seem to be the worst understood part of the kanban system. When used properly, it exposes bottlenecks and reduces lead time for individual work items. Used improperly, it can starve developers for work or result in too many people working on the same work items.* (Shinkle & Shihkle, 2009, p. 188)

Oostvogels (2012) in [S16] suggested that there should be set rules about the limits of the WIP along the development process. However, he stated, *"There is no exact science on determining WIP limits"* (Oostvogels, 2012, p. 39). This fact among others has led more than 75 percent of primary studies to suggest constraining WIP limits by experiments. Anderson (2003) in [S5] has established a general rule to define WIP limits: *"the work-in-progress limits should be agreed upon by consensus with up and downstream stakeholders and senior management"* (D. Anderson, 2003, p. 113). He argued that consensus creates a commitment to adhering to the WIP limit.

A few studies endeavored to establish formulas to generalize setting WIP limits based upon their own experiences. Anderson (2003) in [S5], for example, suggested that the WIP lies between 1 and 3 work items per person at any given time. Hence, if the team is composed of 10 members, and the consensus has been reached on having 2 work items per person, then the WIP limit is agreed to be 20 for the team. Other studies reported WIP as ranges; for example, Terlecka (2012) in [S8] reported that

> *The team reached a conclusion (later empirically checked by other teams) that the recommended limit of cards is between the number of team members plus 1, to number of team members times 2 minus 2, so for a 7-person team, it would be between 8 and 12* (Terlecka, 2012, p. 8)

Kniberg and Skarin in [S15] suggested, *"The first WIP limit we used was 2n-1 (n= number of team members, -1 to encourage cooperation) … By monitoring the Kanban board, it is easy to figure out the right limits along the way"* [S15, p. 71]. They also provided three pages of example on limiting WIP, see (Kniberg & Skarin, 2009, pp. 19–22). Limiting WIP in [S6], [S31], and [S35] was suggested to be determined using Little's law, which is one of the principles of queuing theory. Little's equation is presented here:

WIP Limit = Average Processing Rate (Throughput) × Cycle Time (Flow Time)

A combination of Little's law and an innovative approach for controlling the flow and determining WIP limits has been discussed by several primary studies ([S10] [S13] [S33] [S35] [S37]). *Differential service* is an approach to differentiate the quality of service by work stream. The approach suggests dividing the work items into categories with different classes of services. Then, the WIP limits can be set for each category (Reinertsen, 2009); in that sense, the Kanban approach becomes more flexible and responsive to the rapid customer changing requirements.

The classification of work items helps in prioritizing the work activities whereby it will lead to effective management and high- quality products. Work items will be prioritized to avoid delays within the software development process. Tasks are not equal, and, as such, they will be dealt with differently. This means that activities and tasks will be treated differently per their specific characteristics. The type of work will determine the class given to the development process.

As suggested by Norrmalm (2011) in [S3], the work items can be divided into categories of high, medium, and low. Another categorization was suggested by Turner and Lane (2013) in [S13], where work items were divided into a class of services "Normal Class of Service," "Special Class of Service," and "Expedite Class of Service." Table 3.11 provides an example of how the combined approach of Little's law and *differential service* can be used to specify the WIP limits per category.

Table 3.11– Example of Specifying WIP Using Little's Law and Differential Service

Category	Throughput per Week	Weekly Time per Work Item	Targeted Cycle Time per Work Item	WIP Limit
Expedite	3	2.5	1	3
Special	5	3.5	2	10
Normal	11	5	3	33

The example considers having two developers whom each one works 40 hours per week; with the given throughput and the weekly time that is spent on each work item, the total working hours comes to 80 hours per week. The targeted cycle time per work item represents the time that is required for a work item to continuously flow throughout the development process without interruption, where the WIP limit for each category of the class of service is calculated using Little's formula by multiplying the throughput by the cycle time.

3.12.3 Measure the Flow

Measuring metrics is an essential element in the Kanban approach. Based on the definition of the Kanban approach, there should be an emphasis on the continual delivery of value to customers while promoting continuous learning and improvements. Hence, the primary goal of measuring flow is to pinpoint the improvement opportunities leading to a smoother flow to deliver value to customers quickly. Under the measuring of flow, it is important to remember one idea: the system has a capacity that should not be exceeded. If the system is overloaded, the quality of the development will be low. WIP limits can be utilized to overcome the capacity overloading problems. The less the WIP limits, the shorter the cycle time, presuming the throughput is constant as there is a linear relationship between WIP and cycle time as shown in Little's law.

Anderson (2003) in expressed the relationships among WIP, lead time, and defects. He claimed that based upon industrial evidence, *"there is a correlation between increased lead time and poorer quality. Longer lead times seem to be associated with significantly poorer quality… Longer average lead times result from greater amounts of work-in-progress"* (D. Anderson, 2003, p. 27).

Hence, measuring flow depends mainly on measuring the amount of WIP, lead time, and defect rates to gauge the quality of the product under development. After identifying what should be measured and why it should be measured, the instruments to measure the flow can be narrowed down to the following tools: cumulative flow diagrams, lead time/cycle time, and defect rate. This was in line with what has been reported by about 30 percent of the primary studies ([S2] [S3] [S13] [S15] [S20] [S24] [S26] [S27] [S29] [S33] [S35]). Figure 3.9 provides an example of how the cumulative flow diagram is depicted and interpreted.

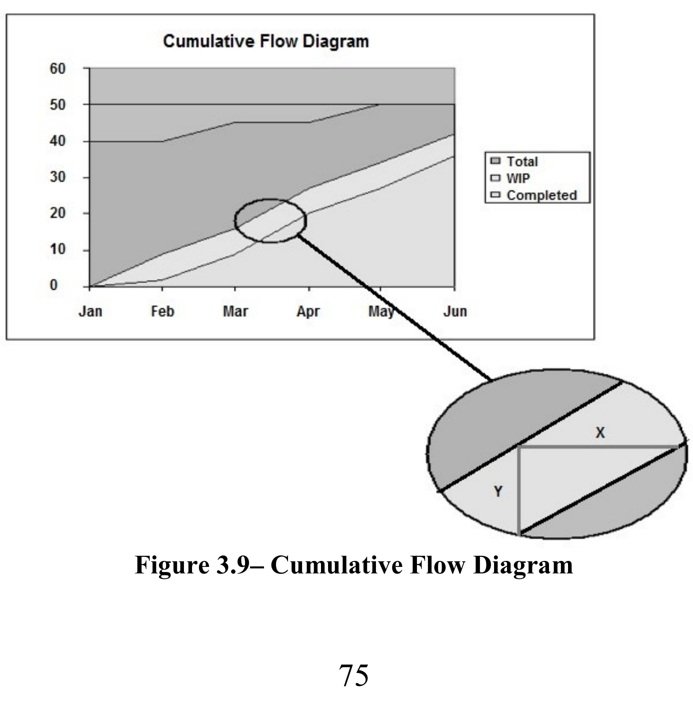

Figure 3.9– Cumulative Flow Diagram

The figure shows three types of queues 1) Total, which represents the amount of work waiting in the queue to be pulled for development, 2) WIP, which represents the amount of work that is currently under development, and 3) completed, which represented the amount of work that was completed. The horizontal axis – X provides a good projection of the time required to complete the work item at hand, while, the vertical axis – Y, specifies the number of WIP.

Figure 3.9 represents a simple cumulative flow diagram followed by a naïve explanation; measuring the flow in fact is significant area to investigate and study, Petersen and Wohlin (2011) in [S24] produced a full research paper on the topic where they empirically evaluated defined measures that had been applied to the visualized cumulative diagrams to increase throughput and reduce lead-time. Furthermore, Reinertsen (2009) in [S35] discussed in an entire book the principles of product development flow.

3.13 Validity Threats

An overall challenge experienced when undertaking this research study was the definition of the scope, as the area under study is multidisciplinary and covers a broad range of fields. Searches covering all the involved disciplines would be challenging and complex, as terminology may refer to a different notion in each area; this comprises a threat to *construct validity*. This threat was considered during the definition of the search criteria and has led to the avoidance of a high risk of errors. However, we could not discount the impact of confounding and local variables in having a significant impact on the search results, which is considered a threat to *internal validity.*

Another limitation of this study forms a threat to *content validity*, as we had no control over the quality and level of details presented in the primary studies. Since most papers lacked adequate information for classifying them, it would have been preferred to have parallel data extraction and to *cross-check* the information on all the study papers. Nevertheless, a lack of resources made this impossible.

Another threat is a selection bias that comprises a *reliability threat*. To protect the study against this menace, the research took three strategies. First, to balance precision and comprehensiveness of the search string, the trial searches were performed using Google Scholar where several alternative keywords were used, and keywords were also combined.

Second, the publications were selected from different sources such as the Scopus, Scirus, and InfoQ among others. Third, measures were taken to avoid and limit built-in bias on the selection that was based on the title, keywords, and abstract. It is possible that we may have missed relevant papers that would have provided great insight to the implementation of the lean-Kanban approach. This likely omission comprises a threat to *external validity*.

3.14 Conclusions

This systematic review incorporated 37 primary studies in which the context ranged from large to complex projects in various companies and also individual projects. The majority of the studies reported the implementation of lean practices along the side with the Kanban approach. The studies, however, reported the Kanban approach as the most frequently used lean practice. Further, a number of studies published recently (from 2006 onwards) illustrated a specific use of the

Kanban approach. This is very much in line with what Wang et al. (2012) reported; the trend to use more specific Lean practices, in particular, the Kanban software development approach.

The Kanban approach to software development is a more direct implementation of lean product development. Lean in software development refers to the lean manufacturing and lean elements translation to the software development application. Kanban offers a less rigid element as compared to agile and has ended up being a popular extension to traditional agile methods (i.e. Scrum and XP) (Hibbs et al., 2009).

The lean approach can be argued to be the fastest growing product development approach in the recent decade, and this is through the use of the Kanban approach. However, despite the interest in the approach, there is neither a standard definition of the Kanban approach nor a clear definition of its elements to software development as indicated by 27 per cent of the primary studies. This lack of definition causes a series of challenges to implementing the Kanban approach.

The use of the Kanban approach comes with several challenges, as reported by more than 50 percent of the primary studies, the conspicuous challenge is related to a lack of details and absence of the guidelines on how the Kanban approach can be used in IT organizations, as well as the lack of the guiding principles on how the different Kanban elements can be introduced. In an attempt to address these challenge, this study provided an analysis of the included 37 studies to investigate the reported elements of the Kanban approach as well as its uses and presentations.

As derived from the available literature, more than 32 per cent of the primary studies agreed that the Kanban approach has three prime principles 1) Visualize the workflow, 2) Set WIP limits, and 3) Measure the flow. It is important to have a consensus by the development team on how these three principles should be implemented and used. Some guidelines have been synthesized from the primary studies, where they were used and proven successful. Team cohesiveness and team building are critical elements in ensuring a successful Kanban approach.

Software development teams in the various firms in the IT industry are considered to still operating as islands, and they should move to working together to ensure quality and consistency of the system (Oostvogels, 2012; Pink, 2011). More than 16 percent of the primary studies reported that the use of the Kanban approach drops the use of isolated teams and led teams to work together to ensure high quality as well as limited use of resources. The team members should participate in developing the Kanban board to make sure that there is a feeling of ownership and pride.

Despite the reported challenges, the Kanban board, as reported by almost 50 percent of the studies, is an efficient visualization tool and should be used effectively to ensure that the development process takes place as expected. This is not the only benefit of the Kanban approach; our findings show that more than 32 percent the primary studies used the Kanban approach to effectively deriving and leading organizational changes as well as facilitating the cross-functional teamwork.

It is recommended to have the Kanban elements to be discussed together based around the Lean five pillars so as to minimize the risk of evolving contradicted elements and to reduce the gap in the literature by creating a common ground to define these elements, which in turn facilitates the establishment of guidelines and instructions on how to setup the Kanban approach to give practitioners an overall framework that increases the likelihood to successfully implement the Kanban approach in IT organizations.

The paper concludes that there is limited research that gives guidelines to the practitioners of the software development in applying the Kanban approach. It is also evident that there is a strong need for systematic studies on the benefits of the lean-Kanban approach. This includes the analysis of what needs to be changed or improved within the lean-Kanban practices to their application in the IT industry. Further research is also necessary to provide formulas to help practitioners in defining the primary elements of Kanban, such as inclusion criteria and defining the “done” items.

The correct use of the Kanban approach ensures that there are realized benefits to IT organization, including but not limited to, minimal work in progress; give a highlight on the constraints and a coordination of the teamwork. The lean approach is more than the Kanban method, and it has been proposed that the Kanban approach should use more approaches from the lean principle so as to take the full advantage of the lean principles in the software development. The Kanban approach shares principles of the agile methodologies on the basis that the requirements are explained in atomic features, which are then implemented incrementally.

4.0 Kaizen, Ubiquitous Improvement Philosophy

4.1 Introduction

For almost three decades, researchers and practitioners have been trying to develop methods and models to improve software development practices, advance quality, enhance productivity, and augment predictability of their development efforts. These methods, models, and mechanisms have been coined in the software engineering field as Software Process Improvement (SPI) ((Virtanen et al., 2013). Process improvement initiatives are generally more complex and challenging to implement in the software development and maintenance context. This is mainly because software development efforts involve various properties, which entail technical aspects, social dimensions, and organizational behavior (Dybå, 2002; Viana et al., 2012; Virtanen et al., 2013).

Despite the many years of study, research, and development of SPI methods, they still fail (Abrahamsson, 2002). One of the major reasons is primarily related to the fact that most SPI methods utilize engineering practices, which mainly deal with the technical aspects, whereas the underlying social and organizational factors are often disregarded (Virtanen et al., 2013). In order to improve the success rate of implementing SPIs, a method should improve all facets, technical, social and organizational aspects (Calderon et al., 2015; Dybå, 2002; Viana et al., 2012; Virtanen et al., 2013). Recent studies show an increasing trend of implementing improvement efforts in agile and lean-transforming organizations using Kaizen Events (Farris, 2006). Kaizen events are defined as short-term improvement activities that target the business performance and the employees' knowledge, skills, and attitudes (Farris, 2006).

Kaizen has been referred to as "change for better" by the majority of researchers (Al-Baik & Miller, 2016). Despite the fact that Kaizen does not explicitly denote the notion of "continuity", it is implicitly embedded in the context of Lean. The application of Kaizen in the manufacturing industry has been claimed to introduce substantial benefits because of its ability to continuously improve processes, products, and people(Farris, 2006; Middleton & Sutton, 2005; Singh & Singh, 2009). Kaizen, as a philosophy and an approach to continuous improvements, goes far beyond its definition. In fact, Kaizen can be applied as an overall thinking philosophy; it is a binding agent for many other techniques, methods, and tools that empower employees and encourage them to continuously re-think the way they do their jobs in a more efficient fashion (Imai, 1986). Kaizen as a philosophy can be seen as a ubiquitous approach to overhaul all aspects of organizations.

Dybå (2002) emphasized that "software organizations that want to prosper in the 21st century should synergistically combine technology with social collaboration to become learning software organizations." (Dybå, 2002, p. 389). A method to promote learning and understanding the work processes and procedures have been devised in social sciences by Argyris and Schön (1978); that is, Double-loop learning (Argyris & Schön, 1978). Double-loop learning requires an individual to deeply consider the reasons why a problem exists (Hollnagel, 2010). Furthermore, it demands an individual to not only identify the problem, but also to find the reason for occurrence in the organization. Consequently, individuals will have a greater ability to establish systems that would effectively solve the root cause of problems and possible obstacles (Patton, 2011)

Practitioners in the domain of software engineering should be warned that transferring Kaizen and other methods from the manufacturing and other industries to software development and maintenance contexts is not a straightforward process; it is complex and requires the reinvention and the reinterpretation of practices in the new environment. In order to achieve improvements on an ongoing basis, based on the available literature and the researchers experience in implementing Kaizen, the researchers suggest not to depend only on Kaizen Events, but also to attempt practicing Kaizen as ubiquitous philosophy. By definition, an "event" is a temporary endeavor, not continuous. Hence, in order to create continuity and realize the benefits of Kaizen, it is recommended to look at ubiquitous philosophy integrated into organization's culture to become a habit embedded within the daily operations of the organization.

The research study presented in this chapter resulted from the need of ORGUS to sustain the benefits that it has realized from implementing the fundamental Lean principle, "Eliminating Waste" (Al-Baik & Miller, 2014). The organization needed a simple, yet a powerful approach to realize continuous improvements. Over almost 3 years of using the proposed Kaizen model, the industrial research partner has achieved noticeable and steady improvements. The results have shown statistically significant correlations between organizational learning and both, team productivity and customer satisfaction. The major contributions of this report can be summarized as follows:

1) This study attempts to offer a first step towards defining Kaizen as ubiquitous improvement philosophy in the context of IT services and software development organizations;
2) This study combines social and organizational learning methods, tools, and techniques with Lean principles to design and measure a simple model to realize the Kaizen philosophy;
3) This study discusses an application strategy of Kaizen using essential assets, capabilities, and skill sets that exist within each software-centric organization; and
4) Provides a high-level statistical analysis of the results discovered during this study.

This chapter is organized as follows. Section 4.2 provides background on the topic that discusses the main problems as outlined in the literature. Section 4.3 provides a precise statement of objectives and subsequent research hypotheses. Section 4.4 provides an overview of Kaizen as ubiquitous approach and summarizes its key features in light of the available literature.

Section 4.5 describes how Kaizen philosophy can combine social and organizational learning methods to establish the Integrative Double Kaizen Loop (IDKL) model; It also describes the different methods from social sciences and organization theories that have been used in developing the IDKL model. Section 4.6 briefly touches on how the various elements of the IDKL model can be integrated and embedded into the culture to produce one cohesive model.

Section 4.7 is a case study; it provides descriptions of the research context, scope, and methodology; Section 4.8 gives an insight into how the model was designed and measured, it provides detailed statistical analysis of the findings. Section 4.9 provides an insight into the validity threats within the study, while Section 4.10 provides conclusions and recommendations from this study.

4.2 Background

Researchers and practitioners have been trying to develop SPI models and provide systematic approaches to tackle SPI initiatives (Salo & Abrahamsson, 2007). Since the 1970s, SPI methods have evolved to improve the capabilities of the software development processes. In 1976, the Software Engineering Laboratory (SEL) was established as a joint-venture between NASA, University of Maryland, and Computer Sciences Corporation (Basili, McGarry, Pajerski, & Zelkowitz, 2002). One of the primary goals of SEL was to "understand the development process, the software product itself, the effect of various 'improvements' on the process with respect to the methodology" (Basili et al., 2002, p. 69). It was also reported that between late 1970's and early 1980's, the understanding of the software development process was systematically tuned-up through the development and use of Goal-Question-Metric (GQM) (Basili, 1992). GQM aimed to collect -only- the necessary data for the purpose of improving the development process.

In the 1980s, GQM formed the basis towards the development of the Quality Improvement Paradigm (QIP) (Basili, 1989). QIP was developed as a cyclic six-step analysis approach to improve the software development process by ensuring that the development process contributed to the project and organization goals, which had already been defined through the use of GQM (Basili et al., 2002; van Solingen, Basili, Caldiera, & Rombach, 2002).

As data had been collected to measure and analyse the performance of the development process, there was an identified need to scientifically store the available data from previous projects in a useful format so that the data can be reused for the benefits of future projects, which then led to the development of both, Tailoring A Measurement Environment (TAME) (Basili et al., 2002), and the Experience Factory in the 1990's (Basili, 1992). Since the beginning of the new century, the interest in the SPI has become very persistent. Many SPI methods have been developed over the past two decades; most of these SPIs have a cyclic-based approach with 4 basic steps:

1) Evaluate the current situation,
2) Plan for improvement,
3) Implement the improvement, and
4) Assess if improvement has been realized.

The conventional model of Plan-Do-Check-Act (PDCA) (Pettersson, Ivarsson, Gorschek, & Öhman, 2008). Recent studies in the field of software development show an increasing trend of adopting various methods, tools, and techniques from Lean production, which focus on improving the production process (Staats et al., 2011) These improvement efforts have usually been implemented using the Lean promoted principle, Kaizen, which is more often referred to and applied through the use of Kaizen Events. Kaizen events are defined as short-term improvement activities that target, the business performance and the employees' knowledge, skills, and attitudes (Farris, 2006).

Many SPI methods have been developed over the years; however, they were all based on either software development methodologies or internationally recognized standards. For instances, Capability Maturity Model (CMM) ((Paulk, Curtis, Chrissis, & Weber, 1993), Software Process Improvement and Capabilities dEtermination (SPICE) (SO/IEC 15504:5, 2012), and Six Sigma-DMAIC process (Biehl, 2004). Other methods that combined international standards and software development methodologies include, but are not limited to: Iterative Improvement Process (Salo

& Abrahamsson, 2007), which is agile and QIP-based, PRISMS (P. Allen, Ramachandran, & Abushama, 2003) that is CMM-based, and SPI-LEAM (Kai Petersen & Wohlin, 2010), which combines QIP and lean principles.

4.2.1 SPI Limitations and Problem Scope

Despite the many years of research and development on SPI methods, they still regularly fail. Process improvement initiatives are generally more complex and challenging to implement in the software development context. This can be related to several reasons, as inferred from the available literature, a summary of the limitations and problems unique to the software development processes are described below:

SPI methods have applied the principles of engineering to solve non-engineering problems: According to the available literature (Dybå, 2002; Viana et al., 2012; Virtanen et al., 2013), software development efforts involve various properties, which entail technical, social, and organizational aspects. One of the major reasons that SPI are likely to fail, is because that SPI methods utilize engineering practices to deal with the technical aspects mainly; whereas the underlying social and organizational factors are often disregarded (Basili, 1989; Calderon et al., 2015; Dybå, 2002; Virtanen et al., 2013). Several scholars have emphasized that the software development process encompasses creativity efforts combining social interactions and human intelligence (Abrahamsson, 2002), as such the social and human interactions can neither be resolved using engineering practices nor it can be standardized in a process model (Basili, 1989; Dybå, 2002).

SPI methods have been imitating practices from other industries rather than developing more relevant software-context practices: Dybå (2002) suggested that “rather than trying to imitate technical procedures, software organizations should focus their SPI efforts on creating an organizational culture within which these procedures can thrive … software organizations that want to prosper in the 21st century should synergistically combine technology with social collaboration to become learning software organizations.” (Dybå, 2002, p. 389). In addition, Basili (1989) for example, illustrated that engineering practices in the software context should be different from typical manufacturing or hardware engineering, and no one should assume that processes, methods, models, or thinking approaches used in manufacturing should be appropriate for use in the software engineering organizations.

Lack of awareness and understanding of both, what is needed to implement SPI initiatives; and SPI impact on the organization, management, and employees. Researching the available literature on topics that are related to the challenges and barriers of implementing SPI methods, whether on small-to-medium or on large organizations, revealed a wide number of problems and issues. For example, some of the biggest challenges that have been repetitively reported, in the available literature, are related to the lack of awareness, lack of understanding, lack of clarity, and absence of vision on “how-to” implement SPI initiatives and their impact on the organization culture and policies, management practices, and employees’ behaviour (Calderon et al., 2015; Dybå & Sharp, 2012; Virtanen et al., 2013)

4.2.2 Limitations of Lean Adoption by Software Development Organizations

Lean related principles and practices to software development are relatively new, and their applications to software-based organizations are yet to be established (Al-Baik & Miller, 2014; Staats et al., 2011; Wang et al., 2012). Lean production has been promoted as a silver-bullet for whatever problem in any industry; this has been challenged by Dybå and Sharp (2012), they have called into question the ostensible claims about Lean production by James Womack in his book: "The Machine That Changed the World", which was published in 1990 and articulated the principles of Lean Production.

Kaizen is an integral part of Lean philosophy; similar to the other Lean principles; no absolute Kaizen definitions are available in the literature (Al-Baik & Miller, 2016). Albeit the majority of the Lean proponents agree on the importance and criticality of Kaizen, there is a major unawareness of the term in influential books and publications that are concerned with the application of Lean to software development; for example, Poppendieck and Poppendieck (2003) and Shah and Ward (2007a).

The available literature on the application of Lean to knowledge-based industries in general, and in software development in particular, has not gone beyond defining Kaizen as Lean's terminology for continuous improvement, evidence can be found in Wang et al. (2012). Hence, the research investigators started reviewing the original Lean essence from the manufacturing perspective to have an in-depth understanding of the application of Kaizen and its related principles as an overarching and ubiquitous philosophy.

A major risk associated with reviewing Lean principles from a manufacturing perspective, however, is to attempt imitating the practices, rather than the principles. Practices are context-specific (M. Poppendieck & Poppendieck, 2003, 2006); thus, it is impractical and not recommended to duplicate practices "as-is" from a manufacturing context into a software development context (Calderon et al., 2015; Dybå, 2002; Virtanen et al., 2013). Transferring Kaizen to a software context is complex and requires the reinvention and the reinterpretation of practices in the new environment.

4.3 Study Objectives

This research attempts to be a first step towards shedding some light on defining Kaizen as a ubiquitous philosophy, a vehicle for learning and continuous improvements. Additionally, this study attempts to develop Kaizen practices that have more relevance to software development and maintenance organizations in an industrial context. Since 1975, Lucas as cited in Dybå (2002) suggested that "the primary cause of system failure has been organizational behavior problems" (Dybå, 2002, p. 388). Dybå (2002) has also criticized software organizations for attempting to apply more rigorous engineering principles to resolve organizational problems

This research attempts to integrate sociocultural and organizational behavior methods with the philosophy of Kaizen to empirically explore practices that try to overcome the problems identified in sections 4.2.1 and 4.2.2. Based upon the objectives of the study and by considering the reported problems in the available literature, the researchers have formulated the following hypotheses:

H1: Kaizen practices will heavily depend on individual skill sets and involvement in the Kaizen implementation,
H2: Realized improvements can be sustained and positively correlate to the learning habits and learning activities of employees, and
H3: Teams that adopt Kaizen philosophy for an extended period of time will experience a greater boost to productivity than teams that practice Kaizen for only a relatively shorter time.

4.4 Kaizen Overview

Kaizen application in the manufacturing industry has been claimed to introduce substantial benefits as a result of its ability to continuously improve processes and products. The benefits of Kaizen have been realized by the wide variety of concepts and tools that are useful in improving both, individual skill sets as well as organizational capabilities. Kaizen touches on people and processes, simultaneously. This means that Kaizen's techniques can overhaul all aspects of an organization. Kaizen allows organizations, within their unique context, to select suitable methods to apply based on the available resources. However, this has presented some challenges for many organizations, considering that implementing Kaizen approaches and tools may require reinvention and reinterpretation in different contexts, this can be a complex endeavor.

Kaizen has been described in the current literature as a beneficial tool that has been used between 1995 and 2008 to introduce continuous improvement (Singh & Singh, 2009). Kaizen has been claimed to help to map the corporate values onto the employee's values, which has emerged as an important idea for the industry. It can improve attitudes and change management approaches. In regard to the product, Kaizen can reduce development costs, errors, and defects. Kaizen also reduces wastefulness activities within the processes of software development (Middleton & Sutton, 2005). It emphasizes the importance of validating outcomes with customers (Al-Baik & Miller, 2014).

A review of the most cited literature on Kaizen has unveiled that, after more than three decades of existence and continuous development, Kaizen in the manufacturing industry itself has not been well established, and it lacks a precise definition of what it means and what it entails. The review of the literature revealed three most frequently cited research studies: Berger (1997), Imai (1986), and Sing and Singh (2009). The latter is a literature review on Kaizen philosophy.

4.4.1 Kaizen Characteristics

Berger (1997), based upon the book of Imai (1986), attempts to define Kaizen and all that pertains to it as a holistic approach. Berger (1997) also sought to outline the reasons behind why the adoption of Kaizen in North America and Europe has not been as successful as it has been in Japan. The investigators have extracted and synthesized the characteristics of Kaizen from Imai (1986) and Berger (1997). Below is a summarized list of the key characteristics of the Kaizen philosophy:

- Kaizen offers an incremental approach, for small improvements;
- Kaizen is process-oriented and people-oriented rather than a results-oriented;
- Kaizen suggests to solve problems as they arise;
- Kaizen is characterized as a self-discipline approach;
- Kaizen promotes learning by doing, organization learning, and shared mental knowledge;

- Kaizen needs standards to serve as baselines and targets to measure the realized improvements against;
- Kaizen activities can be implemented at either the individual or the organizational level;
- Kaizen promotes questioning of the work procedures;
- Kaizen is developed in a suggestion-based approach;
- Kaizen should be embedded as a work habit.

Recognizing Kaizen as an embedded habit within the organization's culture rather than a small project with start and end dates can develop a very dynamic and changing environment. Based upon this theory, the research investigators attempted to establish a model that considers Kaizen as an overarching and ubiquitous philosophy that consists of methods and techniques to support the realization of each of the abovementioned characteristics.

It is also notable that there have been several existing studies to the application of Kaizen in other industries. For example, the application of Kaizen in Morris Electronics Limited (Chaudhuri, 1997), resulted in extensive improvements in the areas of corporate values. Another example of applying Kaizen at Allied Signal Inc. showed that low productivity and waste of space could be alleviated; specifically, Kaizen led to reducing the Work-in-processes (WIP) by 89% and increasing overall productivity by 88.5% (Daniels, 1995). These two case studies revealed that Kaizen could unlock potential in both technical areas and areas of leadership and organizational culture. This can attest to the diverse nature of Kaizen philosophical approaches. In other words, Kaizen as an improvement philosophy can overhaul different organizational aspects depending on the method of implementation and on the intentions of management.

4.5 Model Development and Evolution

The Kaizen model was incrementally developed as a practical model for continuous learning and sustainable improvements; it has gone through several iterations of refinements based upon feedback from the research participants. While refining the model, it was noted that the preliminary model, as presented in Al-Baik and Miller (2016), was missing essential measurement criteria to assess if improvements have been realized. The modified model is composed of the four original elements (Al-Baik & Miller, 2016) and the additional fifth element, *evaluation methods*, these elements are depicted in Figure 4.1 and explained in more details in subsections (4.5.1 – 4.5.5):

1) Reflective practice, which represents a method to learn from previous experiences to improve the current and the future course of actions;
2) Double-loop learning as an approach to challenging and ultimately change the underlying organizational culture;
3) The analysis technique, which is used to identify the root-causes of the problems;
4) Policies and Standards, which are intended to serve as the baseline to measure future improvements and to promote self-discipline; and
5) Evaluation Methods.

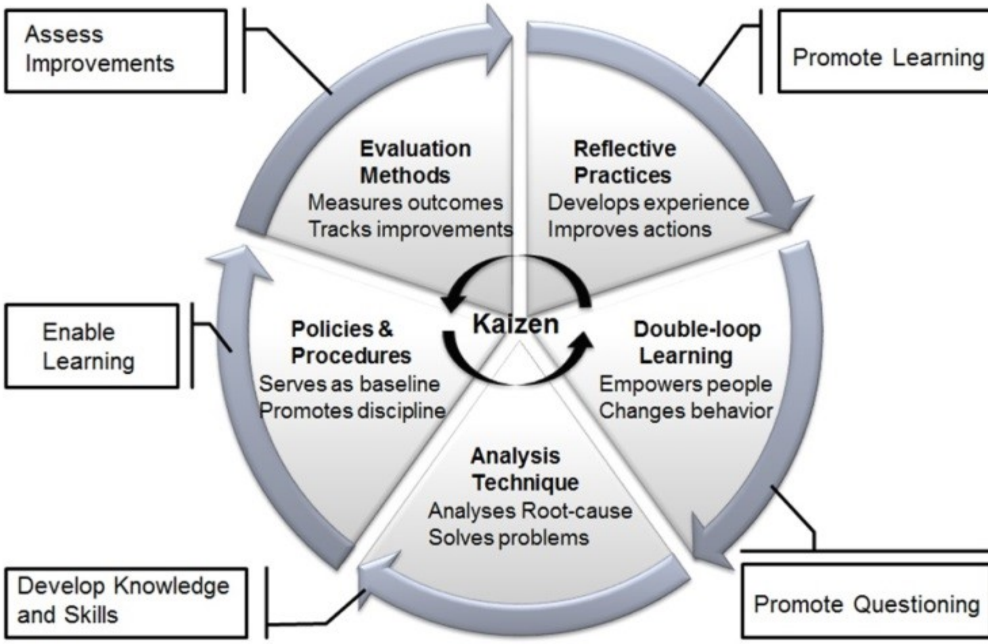

Figure 4.1– Modified Kaizen Model

It is important to note that the sequence of implementing the techniques is not necessarily the same order as illustrated in Fig. 4.1. It is also worth mentioning that Kaizen in the middle is shown in a different track to the model's loop direction; this is to emphasize the flexibility of going back and forth between two or more techniques without the necessity to go sequentially throughout the model. The efforts should be focused on achieving the main objectives of the model, which are stated in the rectangular boxes: promote and enable learning and questioning, develop knowledge and skills, and assess improvements.

In order for Kaizen to be successful, software development teams need to have the capability to periodically conduct self-analysis and explore ways to continuously learn how to improve their development processes (Dingsøyr, 2005). This results in promoting learning and ultimately leads to establishing a learning organization. Dybå, as cited in Dingsøyr (2005), defines a learning organization and suggests that an organization has to "promote improved actions through better knowledge and understanding" (Dingsøyr, 2005, p. 294).

Establishing learning organization in software-centric contexts is challenging; business processes, workflows, and the nature of work is complex to analyze, as they depend on human behavior and commitment as well as technical aspects (Viana et al., 2012). This requires analysis of both: 1) human beliefs, values, and actions, and 2) business functions, processes, policies, and procedures.

Argyris and Schön (1978), recognize single and double-loop learning as mechanisms to provide the capacity to change and question the primary beliefs and values of action. Single-loop, on one hand, the most basic form of learning, results in changes in the behaviors and actions that an individual carries out within a system. On the other hand, double-loop learning is referred to as reframing (Spitzer, 2007), which encourages questioning the functions and the purposes of the work while it is being carried-out (Argyris & Schön, 1978).

4.5.1 Double-Loop Learning

According to Argyris (Argyris, 2002), and as illustrated in Figure 4.2, in double-loop learning, an individual executes the cycle of single-loop learning and complements it by reflecting on whether certain rules need to be altered. Specifically, the rules that are associated with corrective actions of a detected deviation. Watson and Gallagher (2005) argue that double-loop learning plays a crucial role in the success of organizations, particularly during the period of instant change. In this respect, double-loop learning enables the organization to improve its learning capabilities and helps to facilitate changing rules, policies, and practices that are not likely to add value to the end product or service, which contributes to waste elimination, the essence of lean thinking.

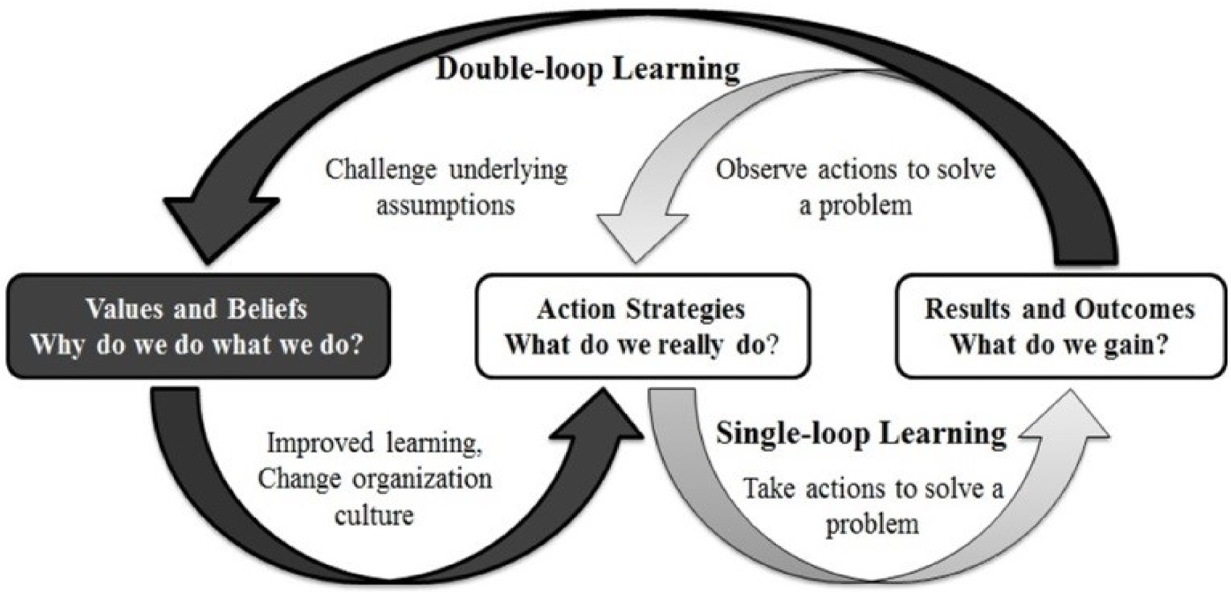

Figure 4.2 – Learning Models [*modified version, adapted by Donald Clark, Double-loop Learning, 2009*]

According to Spitzer (2007), double-loop learning is referred to as reframing, which is characterized by questioning the function and purpose of the type of work that is carried out in an organization. Double-loop learning deals with knowing the kind of work that is conducted, instead of focusing on efficient processes to complete the tasks. Double-loop learning entices an individual to "think outside the box"; hence, it requires critical and creative thinking. Double-loop learning requires an individual to deeply consider the reason why a problem exists (Hollnagel, 2010; Malik & Choudhary, 2008).

Argyris and Schön (1978) suggest that people have mental maps that determine how they act in different situations; these maps guide human actions, which are categorized into two types, theories-in-use and espoused theory. The theories-in-use, on one hand, explains the control of real behavior and act as tacit structures. Theories-in-use explains assumptions related to environment, self, and others, which are inferred from observing human actions to understand how they behave.

The espoused theory, on the other hand, is the rationale that individuals use to justify an action. Intrinsically, if a person is asked: "how would you act under this situation?"; the responses provided are the espoused theories that the individual is more likely to communicate to others (Brockbank & McGill, 2007). Interestingly, it does not necessarily represent the actions that will actually be taken. Argyris (2002) proposed three components of theory-in-use, which includes governing values, action strategies and results (Argyris, 2002, 2004; Rahim, 2010). Governing values and beliefs are the mental maps that individuals use to plan, execute, and review their actions in a given situation.

These actions when implemented can produce discrepancies between the desired outcomes and the actual results. The process of detecting and correcting these errors is the learning process. When a problem is detected, the primitive reaction is to solve the problem by finding an action strategy that would work within the governing values and beliefs. Double-loop learning occurs when the action is observed, and the underlying assumptions of the values and beliefs are questioned (Argyris, 2002). Kim (1998) asserts that double-loop learning brings individuals to a higher level of thinking by incorporating their ideas into a shared organizational thinking model where individuals would build on each other's ideas, which in turn leads to improve the overall performance of an organization.

4.5.2 Reflective Practices

Double-loop learning can be realized through accessing the organization memory, which is defined as: "The means by which knowledge from the past is brought to bear on present activities resulting in higher or lower levels of organizational effectiveness." (Croasdell, Jennex, Yu, & Christianson, 2003, p. 3). Reflecting on past experiences can be achieved by employing reflective practices (Argyris, 2004); these practices are divided into two main methods:

1) ***Reflection-in-action*** is applied to facilitate immediate improvements to current course of action as it is happening; it is done instinctively while drawing on previous experiences;
2) ***Reflection-on-action*** is used to facilitate the achievement of future learning and improvement objectives by reflecting on previous experiences to avoid pitfalls and to incorporate successes in the future course of actions.

On the one hand, the concept of reflection-in-action proposes a way to use reflection in practice by looking at past experiences, emotions, and other instinctive sources of insight (Seibert & Daudelin, 1999). This approach is better defined or expressed by the phrase 'thinking on your feet.' When a practitioner is faced with a challenging situation, this kind of reflection can help aggregating past knowledge and current information to make it real-time ((Lyons, 2010). On the other hand, reflection-on-action enables the practitioner to analyze their past experiences that are similar to the current situation. This further enables a review of the lessons learned on what could have improved to increase chances of success. As synthesized from (Harrison & Reeve, 2002), reflection-in-action aims at:

1) Creating knowledge through the analysis of past actions;
2) Understanding the reasons that resulted in the actions; and
3) Identifying the outcomes of these specific actions.

It is recommended to provide enough and clear details about the context in which the actions were taken and to ensure that the new knowledge has been stored (Koper, 2009).

4.5.3 An Analysis Technique: "5 Whys"

In order for the improvements to be achieved by preventing problems from reoccurring again, the root cause of the problem must be identified. The 5 Whys is a root-cause-analysis technique

(Besterfield, 2009) that was devised by Toyota during the evolution of the Lean production system. It is commonly used in lean enterprises, as it provides a factual-based approach to identify the causal link between the problems and their root causes (Murugaiah et al., 2010).

The 5 Whys approach basically suggests asking "why?" five times when a problem arises, which gives the ability to question the course of action while it is being implemented; hence, it realizes the objectives of reflection-in-action. The following is a real-life example that was used to identify the main reasons for not following the standards in the organization under study:

1) *Why were the standards not followed?* Because the standards were not documented.
2) *Why were the standards not documented?* Because of the resistance to document them.
3) *Why there was resistance?* Because the engineers were not aware of the standards importance.
4) *Why were not the engineers aware of the importance of standards?* Because there was a lack of communicating the benefits of standards.
5) *Why was there a lack of communicating the importance and benefits of standards?* Because it was assumed that developers were aware of these benefits!

This example revealed two root causes, the first was a lack of communication, and the second was an invalid assumption that there was adequate awareness of the importance of standards to the organization. These issues have now been resolved at the organization under study.

4.5.4 Policies and Standards

Standards play a major role in establishing a unique way of executing the daily operations of an organization by taking into consideration the resources, knowledge, and the objectives of that organization. Standards should be documented and updated when new knowledge is obtained. Policies and Standards, however, are more than just documented procedures on how to get the work done. Standards represent the best way an organization knows on how the work should be accomplished at any given point of time (Kondo, 2000).

Hence, the standards should be flexible enough to be altered once a better way, to get the job done, is identified (Wickens, 1990). The main goal of work standards is to reduce variances and defects, which ultimately lead to a higher quality product or service (Kondo, 2000; Narayanan & Chen, 2012). Higher quality, however, cannot be achieved if the engineers are enforced to follow the standards without understanding the work itself, its nature, and objectives (Kondo, 2000; Moe, Dingsoyr, & Dyba, 2009).

The employees need to take part in developing organizational standards, as employees better understand how the work is conducted in greater detail. Therefore, work standards should consist of the goal of the work, policies, pre-conditions, post-conditions, and constraints on performing the work. However, the actual procedural level and steps on the "how-to" should be self-determined by the engineers themselves (Daniels, 1995). In addition to improving quality, the self-determined standard serves as a baseline for future improvements (Marcum, 1993).

For example, consider there is an agreed upon self-determined process for converting a paper-based form to an electronic one. When an engineer finds a better approach for the conversion process, they should document it and propose it to be the new standard process. The differences on resource consumptions or lead time for example between the old and the new approach are the realized improvements. Additionally, self- determined standards contribute to organizational memory, and hence enables learning as part of reflective practices.

4.5.5 Evaluation Methods

The evaluation methods are designed as a structured approach to detect variations beyond the acceptable tolerance level as set by the product specifications. The evaluation methods may vary from process to process; regardless of the process importance to an organization, whether critical or incidental, there should be a systematic approach to evaluate its outcome. Commonly, the outcomes are assessed against a baseline or benchmark (Besterfield, 2009).

Organizations in the software development and IT service field, have applied several methods to evaluate the performance of their products and service offerings. Some of these methods are well-defined, such as ISO, CMM, and TQM, while some others are just ad-hoc. Regardless of the evaluation methods employed, they are recommended to be designed to assess the outcomes based around specific performance objectives, which are often referred to as, Key Performance Indicators (KPI), such as cycle or lead time, and defect rate.

4.6 Embedding Kaizen into Culture

Successful transformation towards becoming a learning organization that is open to adopting Lean principles and continuous improvements requires changes to the organization's culture and habits (M. Poppendieck & Poppendieck, 2003). Cultural changes in organizations start with their employees (Burgess, 2007; D. Kim, 1998; Pascale, Millemann, & Gioja, 1997); double loop learning and reflection-in-action of the IDKL model outlines the basis for implementing changes to the employees' theory-in-use, as it establishes a mindset of inquiry.

This inquiry culture encourages employees to question the motivation behind the way the work is conducted. This can be seen as an opportunity to analyze the current processes and enhance their efficiencies by eliminating wastefulness activities. The entire waste elimination process, a core lean principle, is promoted through the use of inquiry questions, which helps in realizing and sustaining the continuous improvements.

Figure 4.3 depicts the fully integrated double-loop learning method combined with other techniques and tools, which were believed to fulfill the Kaizen characteristics as defined in subsection 4.4.1. The analysis technique in IDKL requires achieving two major objectives: (1) Challenge value estimations and internalized beliefs, and (2) Question policies and work procedures. These objectives have been achieved through employing the analysis technique, 5 Whys (Besterfield, 2009; Murugaiah et al., 2010)

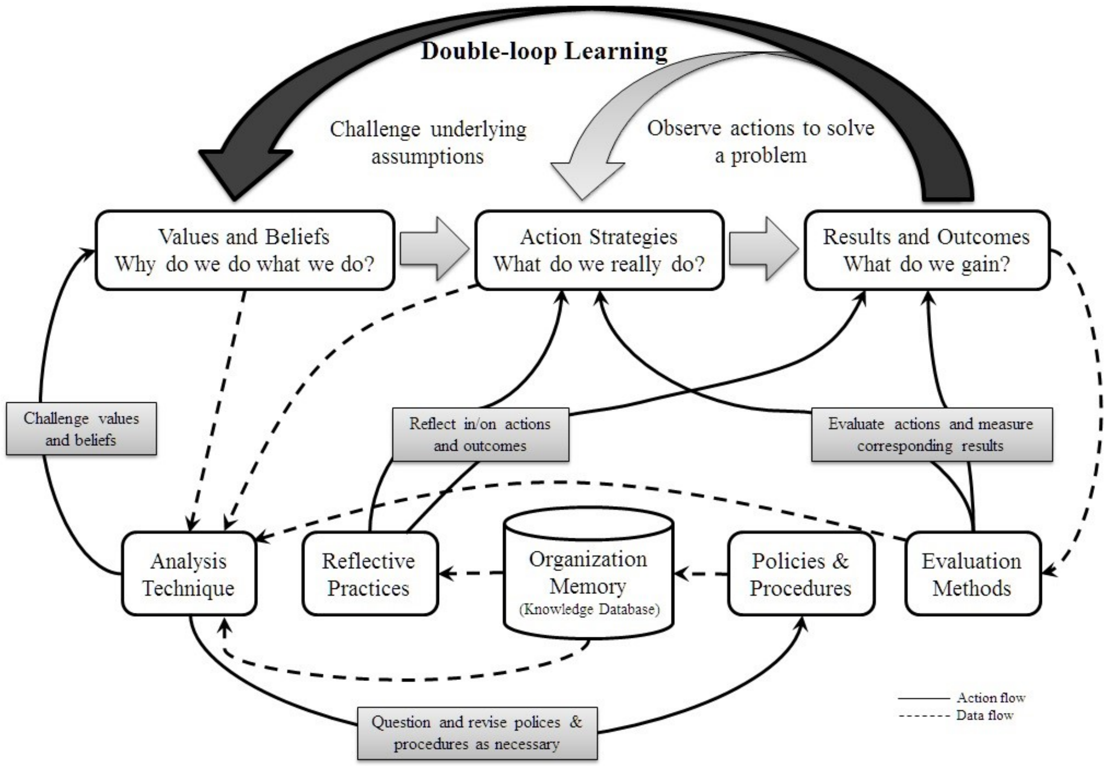

Figure 4.3 – IDKL: Kaizen philosophy Integrated with Double-loop Learning

The 5 Whys technique was used for its easiness and simplicity, as it does neither need special skills nor superior knowledge to apply. The 5 Whys enables an individual to question the course of action, which contributes to the objectives of reflection-in-action; that is, grasping "in-spot" and immediate improvements to the current course of actions.

For employees to be more productive, they should understand why they do their duties in that way! They should start questioning the work procedures, look for deficiencies in the work instructions, and suggest better ways to get the task accomplished. Propositions for more productive and efficient procedures could be inspired by investigating organization memory, in search of related lessons learned.

Lessons learned are to be concluded using reflective practices that utilize both, (1) action strategies and (2) results and outcomes. The actions strategies, on the one hand, undergo further investigation by challenging the underlying assumptions that govern values and beliefs, which may lead to the questioning of organization policies or modifying work instructions or procedures. In the other hand, results and outcomes trigger the evaluation methods to measure the corresponding results and assess if improvements have been realized.

Evaluation methods, using the analysis technique, may trigger the learning process to begin if variances from the desired results are deducted. Hence, the loop will start again by challenging the values and beliefs, which ultimately will lead to the questioning of organizational policies and/or work processes and procedures.

4.6.1 IDKL: A Suggestion-based Approach

Improvements can be realized when there are new comprehended changes to the employees' values and beliefs; this will encourage them to start the practice of questioning policies and work procedures, it will also empower them to experiment with other mechanisms in an attempt to achieve better outcomes and results by altering policies or procedures. If improvements have been observed, employees would then suggest changes to policies or work procedures.

The outcomes are then evaluated by subject matter experts, who are recommended by other colleagues who have hands-on experience and deep knowledge of the process and its desired outcomes. The subject matter experts should have the capabilities and power to determine if the changes are incidental and can be implemented immediately or substantiate that needs to be reviewed and approved by a higher official body. This attests to the fact that Kaizen philosophy proposes improvements through a suggestion-based system.

As an illustration, consider the following example, it provides a critique of how the IDKL model was implemented in a real-life situation. It is important to note that the suggestion-based system and the review process in the organization under study were applied through an electronic voting mechanism. Once a suggestion is proposed by an employee to improve a policy or work procedure, the employee's (direct) supervisor and the change manager proceed to evaluate the outcomes. If agreed upon, the proposed changes must undergo further investigation; the suggested changes have proven to be the source of improvements, they would then be officially adopted.

If disagreement exists between the direct supervisor and the change manager, the suggested changes to the policy or work procedure should then be escalated to a higher official committee; that is, a Change Advisory Board, for more details, see (Al-Baik & Miller, 2014). In the organization under study, there has been a process for monitoring the performance of several enterprise-wide systems. The response time of retrieving "records" by the Enterprise Records Management System (ERMS) was unstable.

When the end-user conducts a "search," a query is triggered and run over and over again as the user enters each folder, and does not stop until the query is completed within the directory underneath that folder, even if the end-user closes the search page! As the search goes into folders, when not required, it consumes a lot of the ERMS server resources. Hence, the response time keeps growing with every new search process, which negatively affects the consumption of server resources.

The ERMS software development team was asked to use the IDKL in an attempt to solve the problem described above. This issue has triggered the learning process to begin! A deviation from the desired result was detected. The ERMS team has a prior knowledge of IDKL as the team was involved in developing the IDKL model. The primary investigator in this study was the leader of the ERMS team; he delivered 3 workshops to the ERMS team on the various components of the IDKL model and how they were expected to be applied. In the beginning, the team did not follow the proposed model; the team was used to the culture of rushing into implementing a solution (Al-Baik & Miller, 2014). The team, however, started to change their behavior incrementally and began to apply the IDKL model.

The following shows how the team was progressing with implementing the model. Instead of starting with 5 Whys, they just went with 3 Why questions:

1) *Why is not the response-time stable?* Because of the heavy consumption of server resources.
2) *Why are the server resources heavily consumed?* Because the search query runs, and does not stop.
3) *Why does the search query run and does not stop?* Because it runs until completion, each time an end-user changes their current location in the directory tree.

A process resolution was proposed, developing a "Search Guide" – a tutorial that explains how to perform a search, and to best retrieve and work with records. The team prepared the "Search Guide," where they explicitly wrote: "Due to performance issues, we request users to not go into a folder if not required. This should provide them with a faster system response time and help balance system resources on the server". Obviously, this resolution was not approved to proceed further. The team then continued with a 4th Why question:

4) *Why does the search query run each time end-user changes their location in the directory tree?* Because the underlying technology has been designed this way.

Several technical solutions were attempted, for example, terminating the search process when the end-user changes their location in the directory tree, or when they close the search page. Another resolution was to ignore searching within the contents, but limit the search to records' metadata, which is pre-defined mandatory data about the record, such as author, creation date, and record type. These proposed resolutions were found to require enormous efforts and time to implement. The 5th Why question, was then imposed:

5) *"Why does ERMS software use that specific technology?"* This question was the bottom-line.

The team concluded that the main issue lies in the employed technology; that is, Apache Lucene, an indexing engine, and information retrieval software library. This question has challenged the underlying assumption that the team has always believed in, that is, the system should use the Apache Lucene. As the underlying assumption has been challenged; the team has implemented reflective practices on the actions to be taken, they have reviewed the organization memory looking for suitable alternatives to Lucene.

The team found that another enterprise application had replaced Lucene by Solr, which is an open source enterprise search platform that uses Lucene as its core but includes faceted searches that enable users to apply multiple filters and navigational searches that allow users to narrow the search scope iteratively.

The team explored the applicability and compatibility of Solr; then they proceeded with the implementation, which has stabilized the ERMS response-time. The changes to the underlying technology were escalated to the change advisory board for voting, along with reports showing the improvements that have been realized. Fig. 4.4 shows a data chart that compares the response time over 8 weeks, where for the first 4 weeks, the ERMS was using the Lucene indexing engine, and for the 4 following weeks, it has been using the Solr enterprise search framework. This example applies the IDKL model in a real-life situation. The realized improvements have encouraged the Enterprise Records Management Development team to reuse the model over and over again. It has ultimately become a daily habit for each identified improvement opportunity.

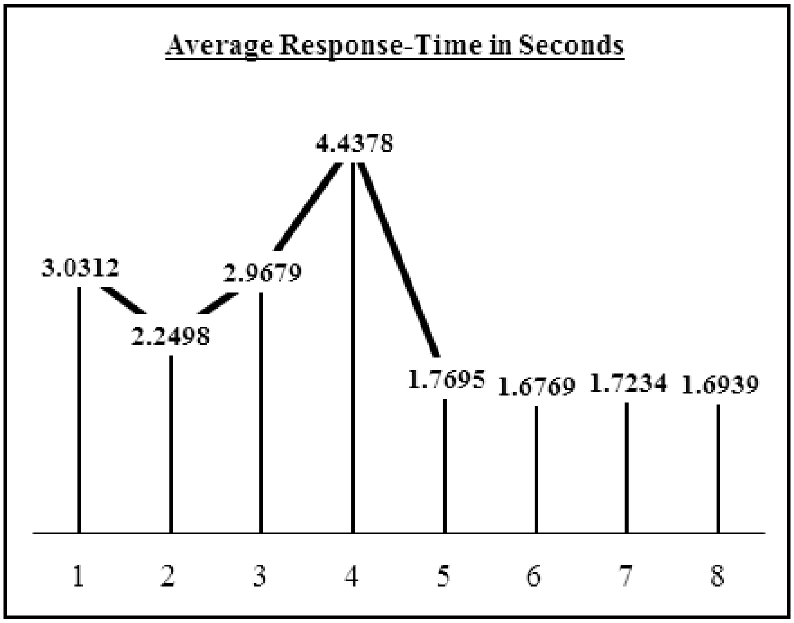

Figure 4.4 – ERMS Response Time Comparison

4.7 IDKL in-action: Case Study

This case study seeks to demonstrate the validity and value of the proposed IDKL model on improving the learning habits of a medium-sized IT organization, through integrating basic, yet well-developed, practices into the organization's underlying culture. The organization under study and the research group agreed to keep the identity of the organization anonymous. Hence, for the remainder of the paper, we will refer to the company by the arbitrary name ORGUS.

4.7.1 Research Context

ORGUS is an internal medium-sized IT department that provides a broad range of IT services through 27 teams, including Application hosting, system administrations, networking, project management, desk-side support, physical infrastructure and data centers, and software application development. At the time of writing this report, ORGUS had over 380 staff, an estimated annual budget of approximately $46 million, and provided services for 38,000+ end users.

The executive team challenged ORGUS senior management to reduce operational costs and service response times while increasing quality and customer satisfaction. Process improvement initiatives were deemed necessary to overcome these difficulties and to sustain the future performance of ORGUS. Further details can be found in Al-Baik and Miller (Al-Baik & Miller, 2014). In this chapter, however, the focus is on implementing the IDKL model to realize continuous learning and sustaining the performance of the improvement initiatives.

The improvement initiative to introduce the learning habits and culture lasted for 18 months and employed a core project team and a supporting project group. The core group included a change manager, a project coordinator, two business analysts, four team leads, and the investigators; the primary investigator was the practice lead, and the co-investigator acted as an external consultant. The working group was composed of various resources engaged throughout the project as required; it involved a total of 93 participants.

As the scope of this study goes beyond the initial project work, the working group has been extended to include participants from those teams who have been using the IDKL model. A total of 231 participants from distinct 17 teams were involved in the initial project work and the development of the IDKL model. The teams were categorized into three strata based upon the maturity level of the teams in applying and using the IDKL model.

Table 4.1– Groups of Participants

Stratum	**ID**	**# of members**	**Maturity Criteria**	**Kaizen Experience**
Beginner Teams (BegTms)	BegT1	14	Metrics for performance assessment did not exist.	6 months or less
	BegT2	11		
	BegT3	8		
	BegT4	12		
	BegT5	13		
	BegT6	14		
Participation from BegTms			Total: 72 Ration: 31.2%	
Intermediate Teams (IntTms)	IntT1	12	Metrics for performance assessment existed; however, they were not realistic, difficult to measure, or non-sense.	More than 6 months but less than 1 year
	IntT2	14		
	IntT3	14		
	IntT4	7		
	IntT5	14		
	IntT6	18		
	IntT7	13		
Total participants from IntTms			Total: 92 Ration: 39.8%	
Advanced Teams (AdvTms)	AdvT1	18	Metrics for performance assessment existed, and they were appropriate to quantify the performance results.	More than 1 year
	AdvT2	11		
	AdvT3	22		
	AdvT4	16		
Total participants from AdvTms			Total: 67 Ration: 29%	

Within each stratum, we used *Purposive Sampling* (Easterbrook & Singer, 2008) and identified 17 teams as a starting point. The purposive sampling was helpful to choose the appropriate teams for the study. The teams were selected based upon several factors, including, but not limited to: the importance of the core services they provide, the resource availability, the team's willingness to participate in the improvement initiative, the period for which the IDKL model has been used by that team, and the maturity of the team in terms of performance assessment readiness.

4.7.2 Research Method

Multiple methods have been used during this study, including surveys, focus groups, ethnographic, and action research. The latter was implemented throughout the entire duration of this study. The investigators and industry participants worked together to define problems, identify possible alternatives, apply the optimal solution, and observe the results. In an attempt to increase the scientific rigor of the study, a Cooperative Method Development approach was imposed. Dittrich et al. (2007) recommend implementing the Action Research Cycle in three phases – note that phases were overlapping and running in parallel:

Understanding Practice, in phase, the ethnographic method was espoused to understand the practice better. This phase has been considered the base for the second and third phases. The primary investigator has been working in ORGUS for more than 5 years; hence, he was considered as a normal participant rather than observing participant (Easterbrook & Singer, 2008; Per Runeson & Höst, 2009), Understanding Practice helped identifying gaps in the participants' action strategies. Determine the gaps between theory-in-use (what individuals were actually doing) and espoused theory (what they said they were doing) was essential to understand the culture, which ultimately helped in specifying remedy strategies to introduce improvements.

Deliberate Improvements, in this phase, the team worked to design the potential improvements. Several workshops and focus groups were held to discuss the existing policies and procedures that were related to software processes, including management processes. These discussions were deemed insightful, especially the inputs from the participants who were actually implementing the procedures as they were directly affected by the existing policies. Each workshop started with an icebreaker event. For example, the first brainstorming workshop aimed at identifying the challenges that the group had in ORGUS, it began with an opposite question: "If you were given a chance to make your colleague's life miserable at work, what kind of limitation would you impose on him?". Subsequently, a question was asked to explore what limitations the team actually had: "Which of these limitations do really exist in ORGUS?"

Implement and Observe Improvements, during this phase, the strategies and recommendations of the Deliberate Improvements phase have been accepted and agreed upon to generate commitment to start implementing them (Yvonne Dittrich et al., 2007). During this phase as well, the investigators collected data about the performance of the organization using surveys and auto-generated reports to be the basis for improvement measurement (Easterbrook & Singer, 2008; Per Runeson & Höst, 2009). During the three phases, the investigators attended tens of meetings; the primary investigator has been given access to all the resources needed to collect, analyze, and reformat the data as he has been working as a Senior Project Manager in ORGUS and was the practice lead for the improvements project. The investigators then transcribed their observations that were then validated by the research participants.

4.7.3 Use of G-Q-M

Multiple methods have been used during this study, including surveys, focus groups, ethnographic, and action research. The latter was implemented throughout the entire duration of this study. The research investigators had worked with ORGUS and identified critical performance and quality attributes (CPA), and specific targeted quality goals (TQG). These attributes were then transformed to reflect Lean's quality measures (LQM) by aggregating, redefining, or decomposing them further so that the LQM can be applied (M. Poppendieck & Poppendieck, 2003; Shah & Ward, 2003). As an illustration, consider the following example that shows different transformation techniques. ORGUS has a Helpdesk team, representing the first-line of support that is responsible for routine maintenance and daily operations of the IT services provided by ORGUS.

The helpdesk team was composed of 3 sub-teams, an in-person service team, an online service team, and a telephone service team. The performance of the team was assessed based around three established CPAs: 1) number of served customers, 2) duration of service call/visit, and 3) a customer survey after the call/visit. The review of the current CPAs showed that they suffered different flaws, to mention a few:

- The established CPAs were not consistent as they were not used by the three sub-teams, for instance, call/visit duration was marked as "Not Applicable" by the online-service team;
- The established CPAs did not provide qualitative indicators; they were all quantitative;
- The established CPAs did not contribute to the overall strategic goals of ORGUS;
- The collected CPAs data were not validated; they could have provided misleading conclusions.

The research investigators worked with senior management to crystallize a simplified list of the strategic objectives of each team; then the list was vetted with mid-level managers to develop more suitable CPAs. In order to achieve both: (1) providing a systematic approach for other teams in ORGUS to facilitate reviewing and revisiting their CPAs; and (2) avoiding nested risks of solving the CPAs' flaws by risking introducing a different set of flaws, a scientific and systematic approach was needed, that was GQM.

GQM is one of the several methods that have been widely used by software-centric organizations to define and evaluate measurable goals. Details of GQM are beyond the scope of this study, readers interested in GQM and related topics should refer to the work of Basili (1992), Solingen et al. (2002), as well as Berander and Jönsso (2006). However, it is critical to note that the transformed CPAs were formulated based upon GQM containing the following three levels:

1) *The conceptual level,* was re-labeled as *Identifying Goals*, which contains measurement goals and specifies the objectives of the measurement. At this level, senior management identified strategic goals, whereas the middle management decomposed them to more tactical goals.
2) *The operational level*, was re-labeled as *Formulating Questions*, which contains a set of questions used to characterize how the goals should be accomplished. At this level, middle management defined the questions in cooperation with team leads; and,
3) *The quantitative level*, was re-labeled as *Defining Metrics*, which contains objective quantitative metrics that were collected in order to answer the questions. At this level, team leads worked with operational staff and defined suitable metrics that quantify the answers for the formulated questions in the Formulating the Questions level.

This top-down arrangement has made the CPAs more aligned with the organization's direction and strategic goals. It has also helped in engaging the employees at every level in ORGUS, operational level employees have been more involved, and they have become more aware of "how" they can contribute to the overall performance of the organization.

4.7.4 Quantitative Metrics

The research investigators worked with the research participants to develop generic, organizational wide metrics. These metrics were required to accommodate any given job type. The metrics needed to provide evidence whether improvements were materialized. The metrics were defined based around LQM as follows:

1) *Processing Time (ProcTime):* the time that elapsed from the beginning of a process until its completion (Tapping et al., 2002), excluding all non-value-added activities. This was computed per hour per month.
2) *First-Pass Process Yield (FPPY):* the ratio of non-defective products or completed services that were produced without further rework to the total number of goods and / or services that were produced at any given time (Tapping et al., 2002)
3) *Value Added Time (VaAdTime):* the time spent on creating progress towards delivering the final product or service that the customer is willing to use and pay for (Tapping et al., 2002). This represent the average number of hours per service/feature delivery.
4) *Waiting Time (WaitTime):* the time elapsed from when a source was idle waiting for a work-order to flow from a downstream activity (Tapping et al., 2002). This represents the average number of hours per month waiting for service/feature delivery.
5) *Lead Time (LeadTime):* the time elapsed to deliver a service or product since the order is triggered (Reinertsen, 2009); including non-value adding activities, i.e. waiting time. This represent the average number of hours per month.
6) *Process Cycle Efficiency (PCE)*: the ratio of the Value-Added Time to the total Lead Time (Reinertsen, 2009). This was computed as the average number of hours per month.
7) *Employee Average Learning Time (EARLT):* time spent by an employee to learn about the work or the customer base, which results in an increase in productivity and/or knowledge. This was computed as the average number of hours per month.
8) *Customer Satisfaction (CustSat):* an assessment of customers' perception of how well a product or a service has met their expectations. This was collected from the customer on monthly basis; then average was aggregated and calculated for every quarter.

Auto-generated reports provided data with regard to the performance of each participating team, based around established LQM, which were determined to include FPPY, EARLT, and PCE, where the PCE was calculated as per Eq. 1:

$$PCE = \frac{VaAdTime}{WaitTime+ProcTime} = \frac{VaAdTime}{LeadTime} \qquad \text{Eq. 1}$$

Data on the LQM were collected and triangulated from different sources using various techniques (Easterbrook & Singer, 2008), including customer satisfaction surveys and auto-generated reports produced by business intelligence software that is connected to the ORGUS' ticketing system.

4.7.5 Data Collection

In an attempt to limit data misinterpretation; direct and indirect data collection techniques were used to validate the findings (Easterbrook & Singer, 2008). The direct techniques included surveys and semi-structured interviews, focus groups through brainstorming workshops, mainly observational. This study employed surveys to collect data with regard to Kaizen Events mainly. Farris (2006) concluded that surveys were appropriate for studying Kaizen event effectiveness; on a six-point Likert-like scale, a Kick-Off survey had been conducted at the beginning of the workshop, and a Report-Out survey was performed at the end of the seminar.

The scale was encoded as, 1 for "Strongly Disagree" throughout to 6 for "Strongly Agree." The surveys were intended to measure:

1) The ***group's buy-in*** to the goal of the workshop by measuring the *Goal Clarity* (GCL), *Goal Difficulty* (GDF), and Affective Commitment to Change (ACC).
2) The ***appropriateness of the elements*** of the IDKL model, which were measured by the perceived *Appropriateness of Tool* (APT), and *Quality of Use* (QOU).
3) The ***perceived benefits*** of the workshops in terms of facilitating the gains from *Understanding of Continuous Improvement* (UCI), *Skills* (SKL), *Attitude* (ATT), *Impact on Work* (IMW), and *Overall Success* (OAS).

Over a period of 3 years, a total of 609 Kick-Off surveys and 673 Report-Out surveys were collected from 64 different Kaizen events. A data dump file was created, and all survey results were manually entered by the primary investigator. It should be noted that the scale of Goal Difficulty (GDF) was reverse scored using formula Eq. 2, where X $x$scale of GDF:

$$f(x) = 7 - xf(x) = 7 - x \qquad \text{Eq. 2}$$

Kick-Off and Report-Out surveys were then matched using a unique survey identifier. 91 Kick-off surveys and 106 Report-Out surveys were excluded as the survey ID was either missing or unmatched. A total of 556 surveys were matched. Further data screening resulted in excluding 34 Kick-Off surveys and 23 Report-Out surveys, for strong evidence of response bias (e.g., all answers were marked as 4, 5, or 6). This resulted in excluding 49 more pairs of surveys.

A total number of 507 were included for Kick-Off and Report-Out survey-pairs. In order to ensure data consistency, multiple manual data extraction techniques were applied, namely, test-retest and cross-checking. The primary investigator re-extracted the survey results, compared the two dump files, and compiled a final master file. The master file was then shared with the co-investigator, who cross-checked 34 surveys, two surveys for each of the 17 participating teams, chosen on a random basis. Table 4.2 shows the case processing summary.

The focus groups and brainstorming sessions were adopted to facilitate a number of workshops, where principles and theories related to Lean, reflective practices, and double-loop learning was introduced to the participants. The workshops were also forums to discuss and analyze business processes, discuss current issues with existing policies and work procedures, and to explore improvement opportunities.

Table 4.2 – Case Processing Summary

		Total No. of Surveys		Excluded due to Unmatched ID		Excluded due to Reponse-bias		Included Cases			
		Kick-Off	Rep-Out	Kick-Off	Rep-Out	Kick-Off	Rep-Out	No. Cases	No. Valid Cases	% of Combined	% of Total
Group	BegTms	108	90	9	6	38	43	148	127	28.5%	25.0%
	IntTms	243	269	12	8	33	39	203	179	40.1%	35.3%
	AdvTms	258	314	8	6	20	16	156	140	31.4%	27.6%
Cases Total/Ratio		**609**	**673**	**29**	**20**	**91**	**98**	**507**	**446**	**100.0%**	**88.0%**

The investigators were involved in over 100 workshops, which included 12-15 participants and varied from one to three hours. For each of these workshops, the goals and scope were communicated clearly and revised based upon input and feedback from the participants, who also reviewed and validated the findings and collected data from the workshops. The researchers have also sought expert opinion and feedback to validate the interpretation of critical findings. The indirect techniques used to collect data, which included analysis of tool logs and documentation analysis (Singer et al., 2008).

4.7.6 Kaizen Events Effectiveness

Measuring the effectiveness of Kaizen was harder than expected, this difficulty lies in the heart of Kaizen, as its main principle suggests, “Kaizen offers an incremental approach, for small improvements”; hence, each improvement by itself is likely to be unobtrusive. The incremental improvements can, however, be accumulated over time and be seen as a visible change.

In previous Kaizen research, Al-Baik and Miller (2016), the investigators had adopted a multi-level approach to report research results. The first level, Short-term measures, targeted the “effectiveness of Kaizen events”, which were used to assess and report immediate improvements as perceived by the research participants. This level of assessment was developed merely to satisfy senior management in ORGUS. In this study, however, the short-term assessment was deemed appropriate for an entirely different purpose, that was, developing commitment, interest, and buy-in to the changes introduced by the IDKL model. If the short-term results were significantly positive, they should motivate the team to proceed further to the *Implement and Observe* phase

Short-term measures utilized a systematic, empirical-based constructed scale that has been claimed to be adequately tested and validated in industrial settings to evaluate the effectiveness of Kaizen events (Doolen, Worley, Van Aken, & Farris, 2003; Farris, 2006). Kaizen events are defined as short-term improvement activities that target business performance and the employees’ knowledge, skills, and attitudes (Farris, 2006). In this regard, brainstorming sessions and focus group workshops from this research study have been determined to fit under this definition. Hence, the sessions that were dedicated for continuous improvements were considered Kaizen events, and thus they can be measured using the instrument that was developed to assess the Kaizen events effectiveness.

It is noteworthy to clarify that focus groups and brainstorming workshops that had been determined to be Kaizen events, were the ones that were designed, prepared, and facilitated for the sole purpose of improving the business processes. The goal of the session has been either to continuously improving specific business processes or to design the IDKL model to sustain improvement results.

In different settings, the brainstorming sessions and focus groups may not be considered one-to-one with Kaizen events. Multiple workshops could be regarded as one Kaizen event, or one workshop could be decomposed into multiple Kaizen events depending on the workshop or event goals and expected outcomes. The criteria to measure the results of the Kaizen events have been formulated around specific objectives, which included: service cycle time, defect removal efficiency, statistical analysis of accumulated organizational knowledge, and finally the most important measure, customer satisfaction.

The second level of reporting was on the performance of the team's productivity – long-term measures. This reporting level, however, was revisited and entirely revamped to account for, and unify the different measuring factors that were used by the various teams in ORGUS to assess performance objectives for each team. These unified factors were previously described as LQM (see section 4.7.4). In order to assess if improvements have been achieved by designing and using the IDKL model, the investigators collected data during multiple phases of the Action Research cycle. Specifically, the focus has been on collecting data during the final reporting period (*Implement and Observe Improvements*) to gather data on the overall long-term performance of the IDKL model. Performance data were triangulated from several sources (see chapter 4.7.5).

4.7.7 Results of Applying IDKL Model

Following Farris' model (2006), the groups' buy-in factor was extracted from the Kick-Off surveys, while the appropriateness of Kaizen and perceived benefits were both extracted from the Report-Out surveys. Responses of scales 1 and 2 aggregated to represent negative responses towards the corresponding attribute, scale 3 and 4 were aggregated to represent neither negative nor positive responses; this representation was deemed reasonable to avoid false consensus bias. Scales 5 and 6 were aggregated to represent positive responses towards the related attribute. The Grouped frequency distribution (GFD) along with the frequency ratio (%) were determined to be more appropriate to analyze Likert-like data and was calculated as per Eq. 3:

$$GFD = L + {}^{\frac{N}{2} - Cfl}\!\big/_{Fm} \times Rw \qquad Eq.\,3$$

Where *L* is the lower-class boundary of the median; *N* is the population size; *Cfl* is the cumulative frequency of the groups before the median group; *Fm* is the frequency of the median group, and *Rw* is the width of the group range. As shown in Table 4.3. the results confirm that research participants have been very committed to change events with a GDF mean of more than 5 for each of the "buy-in" factors: GCL, GDF, and ACC.

Overall, the respondents' perceptions with regard to the success of the Kaizen events have exceeded 89%. One of the interesting findings to emphasise was related to the appropriateness of the Kaizen's tools, methods, and techniques. The respondents in the BegTms group showed inadequate positive responses of 29.9% for both, APT and QOU. The perceptions, however, have changed noticeably by the IntTms to 82.9% and 76%, respectively. It increased to 85.5% and 82.1%, respectively, by the AdvTms, who have practiced Kaizen through the IDKL model for an extended period of time and had relatively extensive awareness training on Lean, Kaizen, and the associated IDKL components.

Table 4.3– Kaizen Events Survey results

		GCL	GDF	ACC	APT	QOU	UCI	SKL	ATT	IMW	OAS
BegTms	**GFD Mean**	**4.96**	**4.74**	**4.76**	**3.72**	**3.77**	**4.24**	**5.28**	**4.19**	**4.27**	**5.17**
	Negative Resp. %	0.00%	0.79%	1.57%	18.90%	16.54%	0.00%	0.00%	0.00%	0.79%	0.00%
	Non Neg/Pos %	26.77%	36.22%	33.86%	51.18%	53.54%	62.99%	11.02%	65.35%	59.84%	16.54%
	Positive Resp. %	73.23%	62.99%	64.57%	29.92%	29.92%	37.01%	88.98%	34.65%	39.37%	83.46%
IntTms	**GFD Mean**	**5.14**	**5.09**	**5.12**	**5.09**	**4.93**	**5.08**	**5.27**	**5.16**	**5.20**	**5.31**
	Negative Resp. %	0.00%	0.00%	0.59%	3.61%	4.47%	0.00%	0.00%	0.00%	0.56%	0.00%
	Non Neg/Pos %	17.88%	20.67%	17.91%	13.59%	19.55%	21.23%	11.73%	16.76%	13.97%	9.50%
	Positive Resp. %	82.12%	79.33%	81.60%	82.80%	75.98%	78.77%	88.27%	83.24%	85.47%	90.50%
AdvTms	**GFD Mean**	**5.24**	**5.20**	**5.17**	**5.19**	**5.07**	**5.16**	**5.36**	**4.97**	**4.77**	**5.36**
	Negative Resp. %	0.00%	0.00%	0.71%	2.78%	3.57%	0.00%	0.00%	0.00%	0.00%	0.00%
	Non Neg/Pos %	12.86%	15.00%	15.00%	11.71%	14.29%	17.14%	7.14%	26.43%	36.43%	7.14%
	Positive Resp. %	87.14%	85.00%	84.29%	85.51%	82.14%	82.86%	92.86%	73.57%	63.57%	92.86%
Overall	**GFD Mean**	**5.12**	**5.02**	**5.03**	**4.72**	**4.64**	**4.86**	**5.30**	**4.83**	**4.80**	**5.28**
	Negative Resp. %	0.00%	0.22%	0.90%	7.62%	7.62%	0.00%	0.00%	0.00%	0.45%	0.00%
	Non Neg/Pos %	18.83%	23.32%	21.52%	23.54%	27.58%	31.84%	10.09%	33.63%	34.08%	10.76%
	Positive Resp. %	81.17%	76.46%	77.58%	68.83%	64.80%	68.16%	89.91%	66.37%	65.47%	89.24%

The reason behind the better-perceived understanding of the appropriateness of Kaizen tools seems to be related to the experience of the team members with the different Kaizen tools. The more the participants become familiar with the tools and the techniques, the more they can relate to how and when they can be used to improve a process' outcomes and results.

Overall, the perceptions towards Kaizen have been positively increasing as the teams advance in their familiarity and knowledge with Kaizen philosophy and the related learning methods. For every factor, the positive perceptions were trending for better, except for Attitude (ATT), and Impact on Work area (IMW).

These two factors seem to reach the peak when the teams have used Kaizen for at least one year. Table 4 shows the aggregated perceived results for buy-in, appropriateness of tools, and perceived benefits for each group. After using the IDKL model for almost three years, the results show notable changes in the different gauged attributes as the teams progressed from the BegTms to the IntTms level. Steady improvements have also been noted as the teams progressed towards the Advanced level (AdvTms).

The most noticeable change was the level of understanding how the concepts, tools, and techniques of the IDKL model can fit and work together so that continuous learning and improvements can be better understood and realized. The results reported in Table 4.4. have been synthesized based solely upon the perceived views of the participants on the overall success of the Kaizen events. Figure 4.5 shows a parallel view of the aggregated perceived attributes; it also provides a visual presentation of how the gauged attributes have been better perceived as teams have progressed from Beginner (BegTms) level, through Intermediate (IntTms) level, to the advanced level (AdvTms).

Table 4.4– Aggregated Attribute per Group

		Participants Buy-in	Appropriateness of Tool	Perceived Benefits	Overall Scuccess
BegTms	Non-Positive Resp. %	32.30%	17.70%	50.00%	16.54%
	Positive Resp. %	66.93%	29.92%	50.00%	83.46%
IntTms	Non-Positive Resp. %	18.99%	19.27%	16.06%	9.50%
	Positive Resp. %	81.01%	80.73%	83.94%	90.50%
AdvTms	Non-Positive Resp. %	14.52%	17.50%	21.79%	7.14%
	Positive Resp. %	87.10%	85.00%	84.30%	92.90%

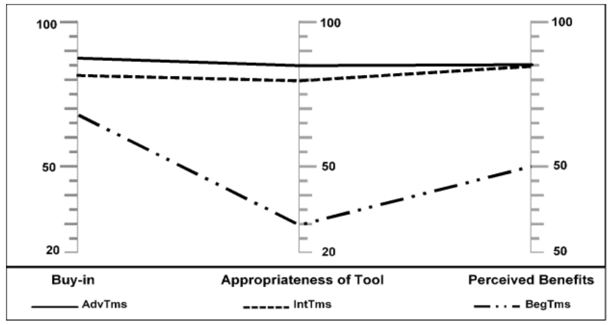

Figure 4.5 –Gauged Attributes Parallel View

Over a period of 33 months, from October 2013 to June 2016, inclusive, data was aggregated to show the results per quarter. Table 4.5. provides a summary of the reported results. Table 4.5 shows the total number of New KBAs (NewKBA) and the total number of Updated KBAs (UpdKBA) in each quarter. Table 4.5. also lists the averages in hours, for WaitTime, ProcTime, ValAdTime, LeadTime, and EALT. Finally, it provides the ratios for FPPY, PCE, and CustSat.

As noted in Table 4.4, the creation of new KBAs has dropped over time, while the number of updated KBAs has amplified. This indicates a successful adoption of double-loop learning, which has changed the employees' behaviors. The new trend has become to review the KBAs, look if a better way exists to perform the task in hand, and update existing KBAs rather than creating a new KBA for every assigned work-order.

Table 4.5 – Performance Data Per Quarter

Quarter	NewKBA	UpdKBA	WaitTime	ProcTime	VaAdTime	LeadTime	FPPY	PCE	EARLT	CustSat	KaizEvent
Q4/2013	57086	80	8.884	10.238	1.252	19.122	0.5019	0.0653	6.309	0.71	4
Q1/2014	61723	569	8.916	9.559	1.450	18.476	0.5121	0.0791	7.701	0.76	7
Q2/2014	34977	2499	8.063	7.817	1.787	15.880	0.5219	0.1129	9.062	0.72	7
Q3/2014	23685	5681	6.604	7.238	1.614	13.842	0.5394	0.1161	9.806	0.75	6
Q4/2014	23132	8228	7.273	6.568	1.701	13.841	0.5699	0.1162	10.667	0.77	7
Q1/2015	19285	12207	7.482	5.298	2.030	12.780	0.6295	0.1616	11.728	0.80	9
Q2/2015	15145	17010	7.607	4.135	1.636	11.743	0.6305	0.1378	13.051	0.83	6
Q3/2015	9082	23375	7.771	3.845	1.569	11.617	0.6334	0.1386	14.779	0.84	5
Q4/2015	9333	31441	7.074	3.853	1.542	10.926	0.6365	0.1408	16.117	0.87	4
Q1/2016	8985	34909	6.635	3.779	1.550	10.414	0.6383	0.1489	16.776	0.89	4
Q2/2016	8617	48649	6.474	3.851	1.587	10.325	0.6396	0.1535	18.470	0.89	5

Over 33 months, the EARLT and PCE have both realized impressive and steady improvements. EARLT has increased from an average of 6.31 hours to an average of 18.47 hours spent on reviewing and updating KBAs. This is on average, an increase of 192.7%, which indicates noteworthy changes in the employees' behavior, which ultimately has contributed to transforming ORGUS towards becoming a learning organization.

PCE has also increased from 6.5% to 15.4%; this is, on average, an increase of 136.9%. This indicates better utilization of employees' capabilities. This has been the result of realizing an increase in the average value-added time by 26.8%, and a drop in the LeadTime by an average of 46%. This indicates an improved response time and a faster delivery of the product/service.

FPPY and CustSat have as well increased, on average, by 27.4% and 25.4%, respectively, which indicates a boost to productivity and better customer satisfaction. In order to validate the results, the quality attributes have been examined for dependency correlations with organizational knowledge and learning. These were represented by both: 1) newly accumulated knowledge – NewKBAs, and 2) the learning process, that is reviewing KBAs and contributing to the existing knowledge, and was indicated by UpdKBA.

4.7.8 Correlation Analysis

The non-parametric analysis was employed to calculate the correlation coefficient and the statistical significance (p-value) for each of the quality attributes presented in Table 4.6. Since the situation is highly complex, we are unable to produce a robust estimation of the statistical power of the correlation analysis, and hence selecting a Type I error value is ill-defined. Hence, we have elected to provide two commons, but arbitrary values, to illustrate this uncertainty. A single asterisk (*) was used to note a correlation that is significant at p-value < 0.05, while double asterisk (**) was used to denote a correlation that is significant at p-value < 0.01. Statistically significant positive correlations, "beyond" the higher Type I error value, have been marked in bold.

Table 4.6 – Correlation Coefficient

		NewKBA	UpdKBA	WaitTime	ProcTime	VaAdTime	LeadTime	FPPY	PCE	EARLT
UpdKBA	Correlation	-.751**								
	P-Value	0.000								
WaitTime	Correlation	**.724****	-.594**							
	P-Value	0.000	0.000							
ProcTime	Correlation	**.941****	-.826**	**.653****						
	P-Value	0.000	0.000	0.000						
VaAdTime	Correlation	-.375*	0.029	-0.279	-0.331					
	P-Value	0.032	0.872	0.116	0.060					
LeadTime	Correlation	**.948****	-.820**	**.814****	**.971****	-0.342				
	P-Value	0.000	0.000	0.000	0.000	0.052				
FPPY	Correlation	-.880**	**.827****	-.546**	-.965**	0.309	-.911**			
	P-Value	0.000	0.000	0.001	0.000	0.080	0.000			
PCE	Correlation	-.836**	**.657****	-.697**	-.853**	**.720****	-.872**	**.827****		
	P-Value	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
EARLT	Correlation	-.862**	**.966****	-.637**	-.923**	0.138	-.907**	**.905****	**.750****	
	P-Value	0.000	0.000	0.000	0.000	0.443	0.000	0.000	0.000	
KiazEvent	Correlation	0.064	-0.304	0.064	0.076	**.618****	0.078	-0.090	-0.090	0.200
	P-Value	0.724	0.086	0.725	0.676	0.000	0.666	0.619	0.619	0.264

Table 4.6 shows that there have been significant positive correlations, at p-value < 0.01, between the learning-variable indicator, UpdKBA on one side, and the three performance attributes: PCE, FPPY, and EARLT on the other side. It is also noticeable that there are significant negative correlations between the learning-variable indicator and the unfavorable factors: (1) WaitTime, (2) ProcTime, and (3) LeadTime. The negative correlations here produce positive results, as reducing values of the unfavorable factors produces positive results. The number of Kaizen Events has no significant correlation with any of the variables, except VaAdTime.

4.8 Discussion

Software process improvements have evolved over the years. Many scholars have emphasized that sociocultural and organizational problems have been acknowledged as important facets that accompanying the improvements to the software development processes, yet the social and organizational aspects were generally disregarded in many of the SPI methods (Calderon et al., 2015; Dybå, 2002; Virtanen et al., 2013); hence in order to increase the likelihood of achieving sustainable results from the process improvement initiatives, several scholars, such as Dybå (2002), Virtanen et al. (2013), and Calderon et al. (2015) have underlined the need for software development organizations to consider social and organizational related problems.

The Integrated Double Kaizen Loop (IDKL) model is composed of a set of processes, methods, tools, and techniques that allow software-centric organizations to combine the core Lean's principle, Kaizen, with social, cultural and organizational learning methods. This was proven successful within the context of ORGUS to improve the overall performance of the software development and service maintenance processes.

A considerable amount of time was allocated to analyzing and understanding the nature of the organization, its operational procedures, its business processes, its culture and capabilities, and the environment in which the organization operates. This analysis enabled the research team to narrow down the processes and procedures that have the greatest contributions to the identified quality and performance attributes.

The daily operations of ORGUS are heavily dependent on its ticketing system. In order for a task to begin, a work-order must be created in the ticketing system and assigned to only one engineer at any given time. Once the task is completed, the work order should be linked to a Knowledge Base Article (KBA) that describes how a task was fulfilled. Figure 4.6 illustrates a modified process flowchart that starts when a service is requested, or a problem is encountered and ends when a resolution is implemented and documented.

Each KBA by itself represents the materialization of the self-determined standards on how the work was accomplished. Figure 6 also shows how the proposed practices support realizing the different characteristics, principles, and concepts of the developed Kaizen philosophy (see subsection 4.4.1). It is worth mentioning that KBAs collectively represent the organization's knowledge and memory, which has been used as measuring factors and an indicator to the performance of the IDKL model.

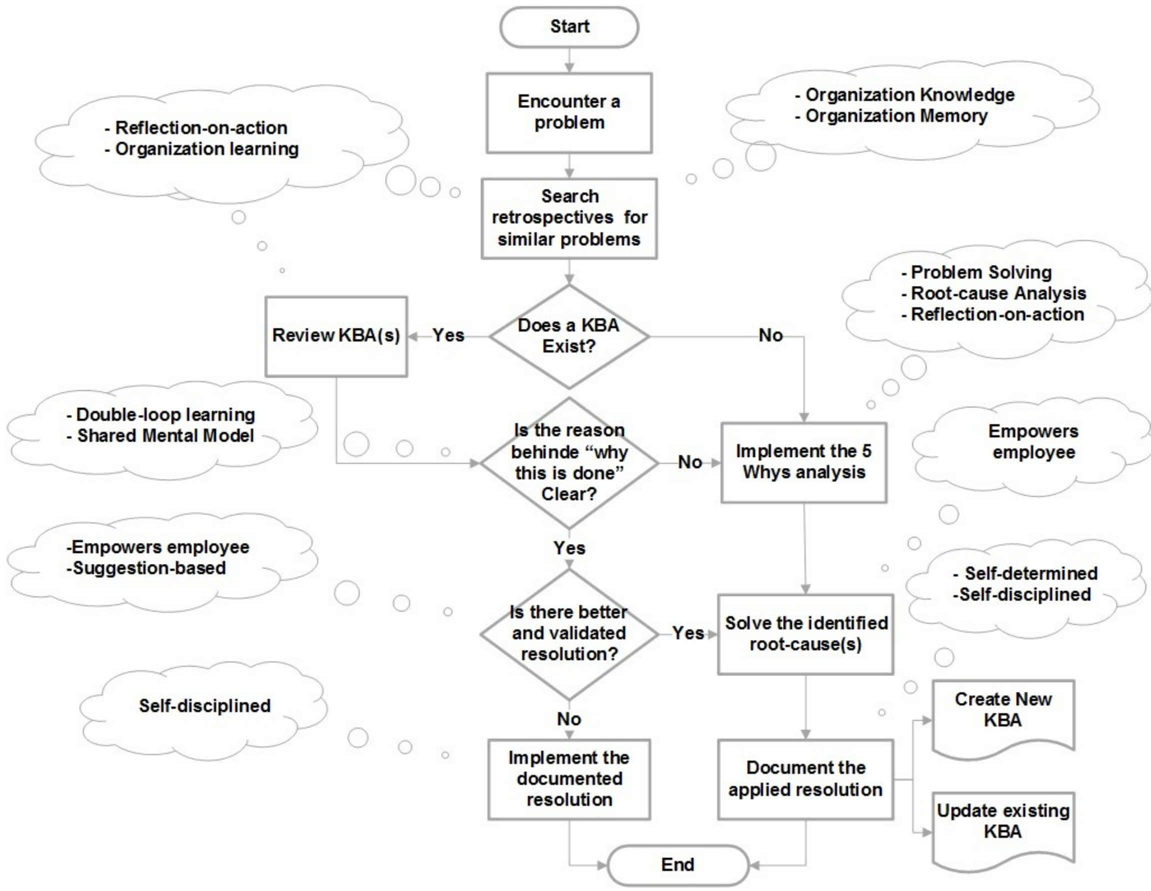

Figure 4.6–Work-order Lifecycle showing Kaizen Characteristics

The newly created KBAs indicated accumulated organizational knowledge, whereas, updates to existing KBAs indicated the process of learning and improving current standards. The practices in Figure 6 were designed during the focus group workshops and agreed to be appropriate for use within ORGUS. The workshops served several purposes, in addition to their primary objectives of empowering the employees and involving them in the decision-making process.

The workshops were considered a platform to collect and analyze performance data that provided early assurance on the fulfillment of the IDKL model and its effectiveness. Understanding the daily operations of ORGUS has help informing the overall direction on what social and organizational learning methods should be combined in the IDKL. The ultimate goal has been to embed the IDKL within the daily operation of ORGUS so that a culture of learning and continuous improvement can be developed at all levels and within every team in ORGUS.

In order to assess whether realized improvements have impacted the desired quality attributes. An assessment model was needed, however, there were no clear criteria for measuring Kaizen effectiveness (Al-Baik & Miller, 2016; Farris, 2006) The IDKL model is composed of several elements, measuring each element of the model by itself will result in focusing on a micro level of Kaizen, while it is the researchers believe that measuring Kaizen effectiveness should, in fact, be at a macro level. Hence, the causal relationships between organizational knowledge and learning and different quality and performance attribute needed to undergo further analyses and investigation.

4.8.1 Causal Relationships Modeling

Despite the fact that the correlation analysis has provided strong evidence of significant correlations, it does neither provide explanations nor guarantees for causation; that is, correlation does not imply causation. Correlations between NewKBA and UpdKBA on one end and the performance attributes on the other end needed to undergo further causality analysis, as they are considered the backbone and the core of this research. A causation analysis has been executed using the Temporal Causal Model (IBM, 2016), the causality diagram is illustrated in Figure 4.7

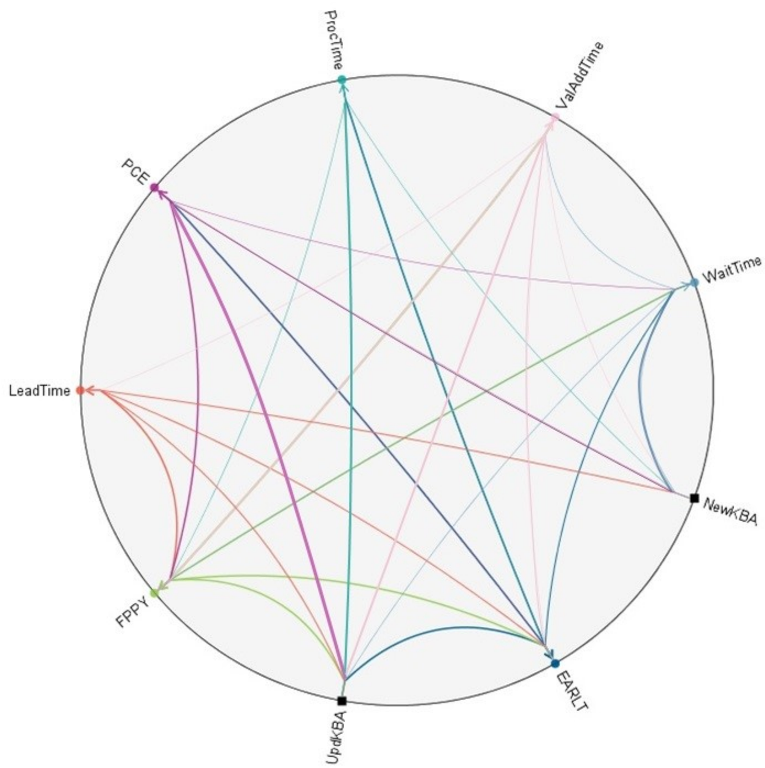

Figure 4.7 – Temporal Causality Diagram

As demonstrated in Figure 4.7, all the variables were connected to those other variables with which they have causal relationships. The arrowhead at the end of the line represents the direction of the impact, starting from the influential variable (cause) towards the impacted ones (effect), whereas the thickness of the line shows the significance of the relationship.

For instance, UpdKBA, NewKBA, and WaitTime were three determined influential variables to PCE as the arrowheads point towards it. However, by looking at the thickness of the lines, it is noticeable that UpdKBA has been the most influential variable, WaitTIme has been the second most influential, followed by WaitTime which is the least influential variable on PCE.

In an attempt to limit the number of influential variables, several procedures were taken: 1) Removal of variables with significance value of more than 0.05 from the model; 2) Removal of dependent variables that showed strong evidence of multicollinearity; and, 3) Removal of the influential variables that have an impact on any non-performance variables.

The third procedure has led to the exclusion of one variable, FPPY, which has a negative correlation with WaitTime. Although FPPY was excluded from being an influential variable, we can draw a logical conclusion on its influence. Taking into consideration that negative correlations with WaitTime produces positive results, the logical analysis of this relationship can be described in a plain language as: “When the employees achieve a higher FPPY rate, they basically realize a reduction in the amount of rework, which will free up more of their time that could be directed towards getting more work done, hence, the overall WaitTime decreases”. This logical conclusion was inferred based on perceptual feedback from team leads in ORGUS.

Statistically, as illustrated in Figure 4.8, applying the foregoing three procedures results in limiting the most influential variables to NewKBA, UpdKBA, and EARLT. It is evident from Figure 4.8 that the performance variables PCE, FPPY, and EARLT have been most influenced by UpdKBA. While NewKBA has a “mildly” negative impact on PCE; in addition, it has a significant impact on WaitTime. It is also evident that EARLT has a significant positive impact on one PCE.

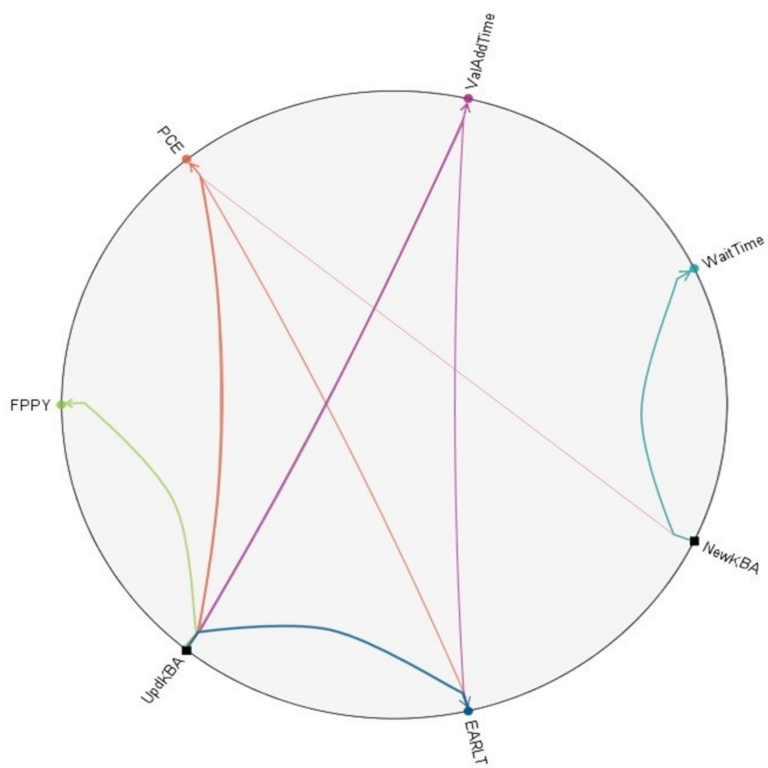

Figure 4.8 – Most Influential Variables

Hence, further investigation was conducted to assess the impact of other variables on EARLT. Figure 4.9 illustrates the impact analysis diagram (IBM, 2016) of the variables impacting EARLT. In order to quantify the influence of each of the four main variables used to predict EARLT, a regression analysis was conducted. Based upon the analysis, UpdKBA has been found to be the most important determinant of EARLT at 43% followed by ProcTime at 31%.

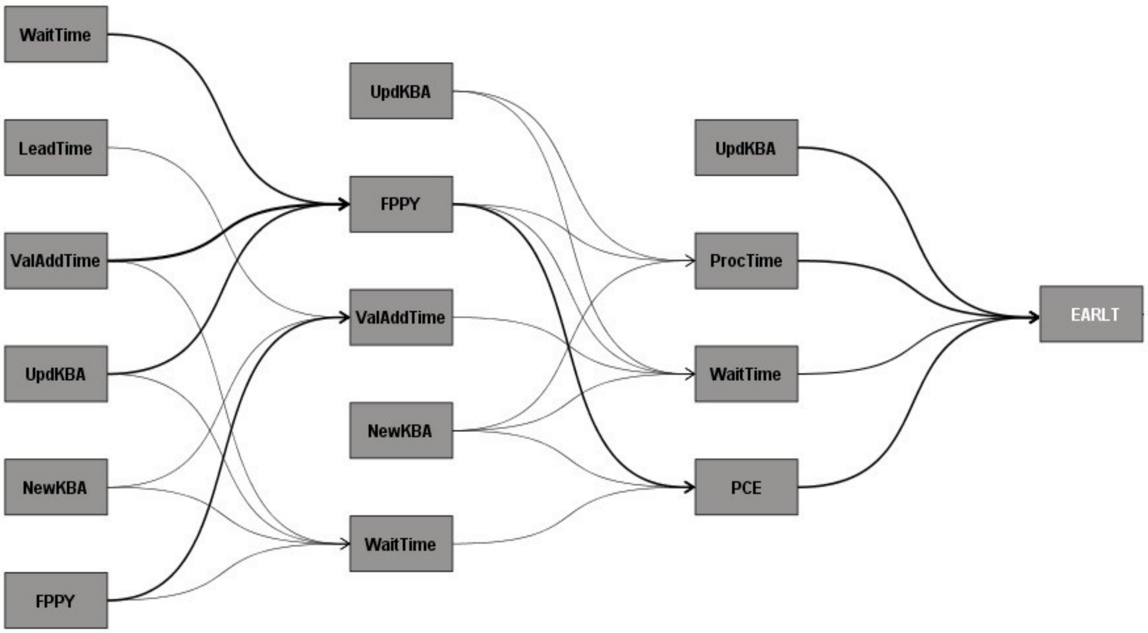

Figure 4.9 – Impact Analysis Diagram – EARLT Focus

Further analysis to determine the most influential and determinant variable to ProcTime was conducted, where FPPY was found to be the most important determinant of ProcTime at 44%. A similar process was also exercised on FPPY, where UpdKBA was the most influential and determinant variable at 54%. The objective of this analysis has been to determine whether transitive relationships exist between UpdKBA, EARLT, and PCE. In addition to the direct impact of UpdKBA on both EARLT and PCE, the impact of EARLT on PCE has been found to have a transitive relationship with UpdKBA as summarized herein: UpdKBA → FPPY → ProcTime → EARLT → PCE.

As can be inferred from the analysis, EARLT, PCE, and FPPY have been primarily influenced by the organizational knowledge and learning, which are indicated by NewKBA and UpdKBA, respectively. Within the context of ORGUS, the proposed model has proven to be successful in enhancing the learning habits of the research participants. Utilization of the IDKL model has almost tripled the average number of hours spent on learning activities, from an average of 6.31 hours to an average of 18.47 hours.

This change of employees' behavior translates to realized improvements on Process Cycle Efficiency (PCE), which has increased from 6.5% to 15.4%; this is, on average, an increase of almost 140%, which indicates a faster delivery of service. First-Pass Process Yield (FPPY) has also enhanced by 27.4%. Higher FPPY results in saving of employees' efforts and time, which has been directed towards service excellence that realized higher quality. The improvements on FPPY and PCY have ultimately improved the overall experience of the customers and increased the satisfaction indicator by an average of 25.4%.

While we cannot scientifically prove that the IDKL has been the primary reason for this change, it is the researchers believe that the changes that were introduced by the IDKL model over a period of 3 years has at least enticed the employees of ORGUS to re-think the way they do their jobs and encouraged them to look into the benefits that they may realize from using the IDKL tools and techniques, such as reflective practices, 5 whys, and double loop learning. For instance, a survey of 36 project managers in ORGUS was conducted in the early stages of developing the IDKL model to understand the issues related to reflective learning and how ORGUS was implementing them. Over 88.9% reported that reviewing the lessons learned was perceived as a waste of time and effort. The following assumptions and espoused theories were identified:

- Lessons learned are poorly documented.
- Nobody reads lessons learned unless there is a disaster in the project, where stakeholders may need to hold someone accountable or "to cover his or her back."
- Lessons learned are only recorded because they are one of the project's deliverables.
- Lessons learned are usually technical and specific to each project per se; therefore, are not deemed useful for other projects.
- Searching lessons learned is time-consuming, too many irrelevant results.
- Project team members are usually assigned to new projects immediately after the completion of the previous one; therefore, there is a lack of time to capture lessons learned in a useful manner.
- Lessons learned do not provide detailed descriptions of project contexts, which affect the decision-making process. What works for one project might not work for another, depending on the project context.
- During the project, stakeholders are interested in finding solutions for current issues, not in documenting lessons learned for future use.
- Lessons learned are captured at the end of the project where most of the details have been forgotten.

In a recent survey of 32 projects managers, which included 26 project managers who participated in the first survey, 84.4% reported that reviewing KBAs to learn from past experiences has been one of the unique cultural advantages in ORGUS! Considering the original survey results, this would translate to worst case scenario of 65% of survey participants whose perception, with regard to reviewing and utilizing the lessons learned, has been entirely reversed over a period of 33 months.

This change in perspective is not an overnight process; it required time, effort, and the commitment of both, employees and senior management. Organizations need to encourage employees to start questioning policies and work procedures (Senge, 2006). Peter Senge (2006) suggested that organizations may even need to reward employees for challenging policies and finding better ways to get the task in-hand completed.

The IDKL model attempts to define Kaizen as a ubiquitous and overarching philosophy for continuous learning and improvement. It can be seen as a binding agent that connects several methods, techniques, and tools to realize continuity. It is the recommendation of the researchers to deeply understand and consider the organization capabilities, assets, and policies as well as employees' knowledge, skillsets, and experience in order to design a practical Kaizen model suitable for that particular context. In summary, the IDKL model has been found to be an efficient and successful approach in the context of ORGUS.

4.9 Threats to Validity

The improvements and results reported in the previous sections must be interpreted cautiously. We cannot claim that the improvements resulted solely from implementing the IDKL model and its related practices. Nor can the impact of potential confounding variables or local factors be discounted; such as the successful implementation of other improvement initiatives. These actors could have significant impacts on the presented results, which comprises a threat to the *Internal Validity.*

The *External Validity* threat to our study is that our observations on this organization may not generalize to other settings, as this study summarizes over 3 years of work in one organization and is unique to its culture, practices, policies, and employees (study participants). The context of each organization per se might dramatically impact the results. In an attempt to reduce the *Construct Validity* threat, various data collection techniques were used during the three phases of the action research cycle, in order to limit data misinterpretation and to validate the findings (Yvonne Dittrich et al., 2007; Easterbrook & Singer, 2008). Additionally, the investigators produced a report outlining their observations and asked the research participants to validate them, thus further validating the findings.

4.10 Conclusions

Organizations in the software engineering field must continuously seek improvements to sustain their competitive advantages and survive in rapidly changing market conditions. One way of achieving continuous improvements is through the application of Kaizen philosophy. Kaizen is an umbrella and the binding agent for many other tools, methods, and techniques that all work in harmony to realize a set of Kaizen characteristics, principles and concepts.

One important concept is that "changing the culture starts with changing the way each employee thinks, the way they behave, and the way they get the task in-hand accomplished!". The synthesized Kaizen principles – as summarized in subsection 4.4.1 – can be useful for software-centric organizations; however, these principles are not recommended to be implemented "as-is". Imitating Kaizen from other industries is not a straightforward process. It requires innovation and creativity in crafting practices that can change the fundamental values and beliefs of individuals.

Kaizen promotes organizational learning, which facilitates improving the overall performance of an organization; it also improves team productivity and increases customer satisfaction. In summary, the overall results and quality of an organization are not a matter of just constructing a gigantic organization memory, documenting lessons learned, or just having retrospectives. Accumulating knowledge or building great lessons learned database does not necessarily help organizations in their learning or improvement efforts. In order to achieve better performance, an organization needs mechanisms to enable learning and to utilize the accumulated organizational memory.

The implementation of the IDKL model has been incremental and gradual. In order to be sustainable, the model required to be part of the organization culture. The research investigators worked with the research participants to design and roll out a model that is embedded within the daily operations of ORGUS. Several research methods were employed, including workshops, which served as forums where the IDKL model has been gradually injected into the organizational culture. The knowledge of ORGUS' policies and work practices has helped to utilize them to derive a model that works within ORGUS' context.

In this chapter, the intention has been to develop a simple and improved practical model to implement the Kaizen philosophy that can help software-centric organizations to improve their operations. Over time, the skill sets of employees have been polished, their knowledge has been empowered, and the way they conduct their daily work has been enhanced. Furthermore, this study investigates the unique nature of software-centric contexts, and a case study of ORGUS was presented to elicit ideas and data on how ubiquitous Kaizen philosophy could be implemented in similar contexts. In the proposed IDKL model, an element of evaluation methods was introduced to capture the need for continuous evaluation to determine whether improvements were realized.

The study also underscores the unique nature of software firms as compared to other companies in the manufacturing industries. This is because of the nature of the business, which is largely dependent on human behavior, personal commitment, creativity, and unique working cultures. In this study, the results have demonstrated the importance of embedding Kaizen within the organizational culture. The results have also shown that the enhanced Kaizen approach can help to improve the daily operations; however, its implementation needs to be done with caution.

In terms of the lessons learned, the unique nature of industries has been recognized, and this has implications for the implementation of paradigms such as Kaizen. Paradigms meant to improve the operations and quality in organizations imply they need to consider all aspects of the development processes; it is also necessary to further enhance improvement approaches themselves to be suitable for implementation in new contexts. Our future research plan includes replicating this research with different industrial partners, in various settings, and within various cultures and organizational structures. This may help in developing a theoretical model that can be generalized and reused by software development organizations with minimal customizations.

We also need to look into a more systematic and rigorous approach to assess the results of implementing SPI methods in general and the IDKL in particular. Our future research plan includes developing a measurement scale that takes into consideration the social, technical, and organizational aspects of SPI initiatives. We finally encourage researchers to conduct empirical studies to explore the applicability of the IDKL model in other software settings to either validate or refute the analysis of the reported results. More empirical studies are generally needed on the implementation of Lean thinking and the related methods such as Kaizen, Kanban, and continuous flow in software-centric contexts.

5.0 Rethinking Resolutions of Employees Turnover

5.1 Introduction

In every organization, regardless of title, authority level, or position in the organization hierarchy, each employee has unique knowledge that no other employees may have. Employees working on the same project or product may have shared and general knowledge about the project or the product that they are involved in, however, there will always be pieces of knowledge about a specific subject matter that no other employee will have. What happens when the employees with critical knowledge leave? Employee turnover is considered one of the main sources of draining institutional knowledge (Armstrong, 2009; Reib, 2014). What are the impacts of losing knowledge because of turnover? How do organizations mitigate the risks associated with such knowledge loss?

This chapter provides analysis from a case study of ORGUS, as a high employee turnover rate was observed over the years. The case study is divided into four phases. Each phase represents a period of time in which parts of the study were conducted to investigate our research questions. We focused on the following research questions: 1) what are the missing factors that lead to the loss of knowledge at the organization under study? 2) What are the main implications of knowledge loss due to employee turnover? 3) What are potential remedies to the problems? 4) How effective are these remedies in addressing the problems of the loss of knowledge due to employees' turnover? This research, however, aims at providing a valuable addition to the research by explaining how to sustainably implement systems that organizations can use to prevent issues related to employee turnover in conjunction with maintaining institutional knowledge.

The remainder of the chapter is organized as follows; Section 5.2 provides an overview of the available literature on various topics and subject areas that were examined during the different phases of this research study. Section 5.3 provides a precise statement of the research motivation. Section 5.4 gives an overview of the research context. Section 5.5 describes the research method and provides details on the four different phases of this research study, which started with an exploration and data collection phase, followed by a phase dedicated to analysing implications of turnover, then the remedies were designed in Phase III, while implementation remedies and assessing if improvement has been realized were conducted in Phase IV. Section 5.6 provides insights to the threats of the research validity. While Section 5.7 summarizes the conclusions from the study.

5.2 Literature Review

First, a review of the available literature on employees' turnover, the impact of turnover on organization's productivity and efficiency, and what were the possible resolutions to the identified problems resulting from turnover. Second, a review of the literature on knowledge management, specifically knowledge transfer, and retention. Third, a review of the available literature on the onboarding of new employees, the importance of successful onboarding, and best practices for smooth and quick onboarding.

There is evidence of a growing literature investigating the implications of employee turnover on the loss of knowledge (Shankar & Ghosh, 2013). In prior studies, one research stream focused on concepts of knowledge flow between competing firms because of employee turnover (Combes & Duranton, 2006; Duranton & Puga, 2004; Gerlach, Rønde, & Stahl, 2009). The second stream of research focused on the external factors influencing employees' turnover such as patterns of wage, tenure, and promotion (Gibbons & Katz, 1991; Greenwald, 1986; Shankar & Ghosh, 2013). Another approach in the literature explains internal factors leading to turnover decisions such as the fit of employees to current jobs or positions (Burdett, 1978; Jovanovic, 1979; Mortensen & Pissarides, 1999; Rogerson, Shimer, & Wright, 2005). While these efforts are directed towards the investigation of different parts of employee turnover concepts, they all identified employee turnover as the main contributor to the loss of knowledge and a major threat to the internal knowledge transfer within organizations.

5.2.1 Employees Turnover

Turnover does not only refer to the employees who leave their current employing organization when the employment relationship ends, but also retired employees, promoted employees, and those who depart on long-term leaves and may not return to work after the leave is over, such as, Long-term Disability or Maternity leave. Research studies reported that the cost of hiring and training a replacement worker for a lost employee is approximately 50 percent of the employee's annual salary (J. T. Johnson, Griffeth, & Griffin, 2000).

As for ramifications of employees turnover, it has been reported that it negatively affects organizations' productivity and overall performance by various means, including but are not limited to: disrupting team collaboration (Šmite & Solingen, 2016), reduced employees' morale (Ampoamah & Samuel K. Cudjor, 2015; Durbin, 2000), and incurred high tangible costs for hiring, training, and mentoring of new hires as well as intangible costs that difficult to estimate, such as team cohesiveness and team collaboration (D. Allen & Bryant, 2012). Ampoamah and Cudjor (2015) also reported poor quality and the loss of skilled manpower amongst the conveyed negative impacts of employees' turnover.

Griffeth and Hom (2001) attributed causes of turnover to employees' dissatisfaction resulting from lack of self-esteem, lack of being seen as a valued member of the organization, lack of career development and advancement opportunities, and reporting to a demanding and impersonal supervisor. (Armstrong, 2009) argued that underpayment or even the feeling of being underpaid as one of the main reasons for employees' turnover. In attempts to resolve the turnover problem, most research emphasized the need to reduce the rate of staff turnover by dissolving the causes of turnover through recognizing good performance, (e.g. Šmite and Solingen(2016)), investing in training programs, establishing career planning and development programs (e.g. Kinicki and Kreitner (2003)), and building a trustworthy and mutually respectful work environment (e.g. Ampoamah and Cudjor (2015))

5.2.2 Knowledge Transfer

The scope of knowledge management is wide; its research topics have been gaining the attention of scholars since the late 1980s (Bender & Fish, 2000). Prior studies have investigated the opportunities, practices, challenges, and benefits of knowledge management (Davenport & Prusak, 1998; Gayawali, Stewart, & Grant, 1997; Nonaka, 2008; Sveiby, 1997). It is found that organizations across major industries constantly show interest in implementing new knowledge management systems (Alavi & Leidner, 1999) and that managers are becoming aware of the importance of knowledge transfer within their firms (North & Hornung, 2003). Similarly, the lack of sound *Knowledge Management* at an organization is found to be contributing directly to the loss of knowledge, especially in situations of employee turnover (Urbancová & Linhartová, 2011).

Bhatt (2001) suggests that organizations willing to capitalize on knowledge must balance its knowledge management activities such as knowledge creation, knowledge validation, knowledge presentation, knowledge distribution, and knowledge application. In general, to achieve this kind of balance, organizations need to focus on interactions between technology, techniques, and employees to manage its knowledge effectively. By creating a "learning-by-doing" kind of environment, an organization can sustain its competitive advantages (Bhatt, 2001). Motivated by this notion, this study suggests resolutions to the problem of knowledge loss due to employee turnover using a learn-by-doing lens.

When organizations lose employees due to retirement or because of employees' decisions to work for other competing firms, organizations are impacted by the loss of the tacit knowledge employees maintained. Therefore, it becomes imperative for organizations to be able to manage knowledge, especially tacit types, to overcome problems of knowledge transfer due to turnover (Urbancová & Linhartová, 2011).

5.2.3 New-hire Onboarding

Immediately after the recruitment process is completed, organizations should strive to improve the performance and productivity of their new hires through the efficient use of onboarding strategies, which has also been referred to as organizational socialization (Bauer, Todd Bodner, Erdogan, & Tucker, 2007). Regardless of what it is called, the main concept is that the quicker the new hires adapt and adjust to the organization's culture, the faster they become productive and contribute towards the overall objectives of the organization. Onboarding refers to the actions taken to implement a combination of tactics, mechanisms, and tools to familiarize and adjust new-hires into the social and cultural aspects of the organization (Rollag, Parise, & Cross, 2005).

The maturity level of onboarding programs varies across organizations, for example, one organization may have a one-day "ad-hoc" orientation program, while a different organization may have a formal written onboarding plan that spans over several months. Researchers show that there are two types of onboarding (Zahrly and Tosi (1989), as cited in Bauer (2010)):

1) *Formal onboarding*, which indicates that the organization has recorded policies, procedures, and manuals to help new-hires adapt to both, the organization culture and the work tasks; and
2) *Informal onboarding*, which refers to the activities that new-hires take to learn about the related job duties and/or organization's social and cultural contexts without an explicit or formal organizational plan.

According to research studies (Bauer et al., 2007; T.-Y. Kim, Cable, & Kim, 2005; Saks, Uggerslev, & Fassina, 2007), new-hires who go through formal onboarding programs that explain the necessary social and cultural norms of the organization, and equip new-hires with the necessary knowledge of how to behave in accordance with the organization's policies and guidelines, adjust faster and become more productive than those who do not attend to a formal onboarding programs. In fact, Bauer (2010) reported that in one study conducted in 1985, it was estimated that 60% of unsuccessful new-hires in managerial positions reported the primary reason for their failure to their inability of establishing effective working relationships. Integrating, adapting, and socializing with colleagues and management of the organization would positively affect performance and satisfaction, and hence reduce the rate of turnover.

Onboarding Practice Guide published by the Society for Human Resources Management (SHRM) (Bauer, 2010), suggests to build an onboarding program that covers four main criteria; These are: 1) Self-efficacy, which refers building self-confidence with regard to getting the job done "well"; 2) Role clarity, which relates to introducing the new-hire to the role's duties and responsibilities as well as expectations and other role-related details; 3) Social integration, which refers to presenting the new-hire to the social norms of the organization and makes them feel accepted and welcomed by their colleagues; and 4) Cultural knowledge and fit, which refers to the level of support provided to the new-hire in order to understand the organizational culture, politics, values, and goals.

5.3 Study motivation

Loss of institutional knowledge may cause severe damage to the organization; this applies in particular when the loss results from the turnover of the key resource, or exp erienced employees who have worked for the organization for many years (Branham, 2005; Katcher & Snyder, 2007; Somaya & Williamson, 2008). Research studies reported that, in Fortune 500 companies, more than half million managers leave their positions each year, and for management positions, the average employment duration is 3 years (Rollag et al., 2005). This high turnover rate means that employees will take all of their tacit knowledge and some of the explicit knowledge they have hoarded - during the term of their employment when they leave.

Therefore, it is important for organizations to cope with the potential departure of their employees (Urbancová & Linhartová, 2011). Many organizations spend millions of dollars to develop and purchase solutions to mitigate the risks associated with knowledge loss (Koudsi, 2000; McCUNE, 1999). It has been estimated that the cost of hiring and training a replacement worker for a lost employee is approximately 50 percent of the employee's annual salary (J. T. Johnson et al., 2000). In addition, each time an employee leaves the firm, productivity normally drops because of the learning curve involved in understanding the new job (Stovel & Bontis, 2002).

Despite today's technological advancements that make documenting and storing knowledge easier and affordable, organizations still struggle with the fact that knowledge is mainly locked away in the minds of employees (Koudsi, 2000). Additionally, this knowledge is rarely shared with other employees before the employee with this knowledge departs (Droege & Hoobler, 2003). Therefore, senior management is encouraged to mitigate the risks of lost institutional knowledge as well as creating plans for organizations to prevent further loss, protect themselves from knowledge attrition, and sustaining previous experiences by codifying knowledge using strategies of knowledge transfer to preserve and retain that knowledge within the organization.

One of the gaps present in the literature is that insufficient researches have been conducted on developing practical models that aim at sustaining and retaining knowledge, and when they existed, they lack the "how-to". Most of the previous research studies have looked at what institutional knowledge to be stored, e.g., (Majchrzak, Rice, King, Malhotra, & Ba, 2014), why the institutional knowledge gets lost, e.g., (Manhart & Thalmann, 2015), and what methods organizations have used to prevent employees' turnover within specific contexts (Bryant & Allen, 2013).

5.4 Context of Case Study

The case study is for an internal medium-sized IT department that provides a broad range of IT services through 27 teams, including application hosting, system administration, networking, project management, desk-side support, physical infrastructure and data centers, and software application development. At the time of writing this manuscript, the IT department had over 400 staff, an estimated annual budget of approximately $60 million, and provided services for 39,000+ end users. The research investigators and the IT department, as an industrial partner, agreed to keep the identity of the organization anonymous, hence, for the remainder of this manuscript, we will refer to the industrial partner by an arbitrary name "ORGUS."

In 2012, the overarching executive team of the organization had challenged ORGUS's senior management to reduce operational costs and service response times, while still increasing quality and customer satisfaction. Process improvement initiatives were deemed necessary to overcome these challenges and to sustain the future performance of ORGUS. After careful analysis of the organization's capabilities and reviewing the trending improvement methods, ORGUS decided to proceed with implanting Lean Thinking and started with Value Stream Mapping and Waste elimination; for further details, see (Al-Baik & Miller, 2014).

In 2013, the implementation of lean thinking continued with a focus on sustaining the realized improvements that ORGUS has had after implementing waste elimination strategies. The outcome of this research project was an Integrative Double-Kaizen Loop model (IDKL) (Al-Baik & Miller, 2017:Forthcoming), which has proven to be successful within the context of the ORGUS, where continuous learning and sustaining performance of the improvement initiatives were realized. IDKL and the improvement initiative has transformed the ORGUS' culture into a learning organization and has successfully enhanced the learning habits in the software development teams.

Since 2012, the research investigators have observed a high turnover rate, which was reported to senior management as one of the biggest problems that ORGUS needed to mitigate and accounted for. In 2014, senior management had made a decision to look into the turnover problems as it hit the highest rate of 32.28% of total employees in ORGUS. The investigator witnessed key resources (research participants) on the research project leaving ORGUS for various reasons. Hence, this served as an important motivation triggering this research project.

Employee turnover is not just a problem because organizations need to hire new employees, it is also a crucial issue because of the loss of institutional knowledge that accompanies employee turnover. In an attempt to help ORGUS resolving these issues, the researchers emphasized the need to re-think the resolution for a turnover. The concept is to retaining organizational knowledge, rather than retaining staff. Organizations need to respond to situations where staff depart from their current positions. Organizations need to adapt to keep operating with less knowledge, but not knowledge-less when employees leave the organization.

During this research study, the research participants were decomposed into core team and supporting team. The core group included a change manager, a project coordinator, two business analysts, four team leads, and the investigators; the primary investigator was the practice lead, while the co-investigators acted as external consultants. The supporting team was composed of various resources engaged throughout the project as required; it included 278 participants who were involved during different phases of the research project.

5.5 Research Methodology and Study Phases

In this study, we used mixed methods to investigate the research questions at four different phases. At the beginning of our investigation, it was important to know what factors that facilitate knowledge transfer between employees were missing from ORGUS. These factors have a direct influence on the process of knowledge transfer (Bresman, Birkinshaw, & Nobel, 1999). Next, we evaluated the impacts of losing knowledge due to employees' turnover on ORGUS. This helped us better understand the current situation of the department and the struggles it is going through due to the loss of knowledge based on high employee turnover. After that, we invistegated possible remedies that may help address this problem. Finally, we applied these remedies and evaluated the improvement in the department's performance because of the aforementioned remedies.

5.5.1 Phase I: Exploration and Data Collection

To uncover the missing factors of knowledge transfer between employees that lead to the problem of knowledge loss at ORGUS after employees' turnover, we gathered data from individuals currently employed by ORGUS. We wrote interview questions based on prior studies investigating factors facilitating knowledge transfer within firms internally (Bresman et al., 1999; Buono & Bowditch, 2003; Cohen & Levinthal, 1990; Haspeslagh & Jemison, 1991; Owen, Koskela, Henrich, & Codinhoto, 2006; Van Maanen & Schein, 1979). The researchers met virtually four times for a total of 10 hours to discuss suitability and validity of the questions, and any modifications or enhancements that they felt were appropriate before conducting the interviews. This process resulted in several substantive changes to the questionnaire.

The sample consisted of 278 employees of ORGUS. However, 57 were excluded from this study due to their minimum historical knowledge about the situation at ORGUS prior to their placement in the last year. Hence, 221 responses were eventually recorded. After collecting all of the responses, the data were coded by two researchers. Because the data was independently coded, the inter-rater reliability was assessed. When complete, the coding results were compared for agreement. The analysis of responses revealed a number of knowledge transfer facilitators being missed –the lack of communications, minimum interactions, and documentation problems.

Lack of Communication. Respondents were asked to indicate the frequency of their internal communication with their colleagues, and for each to distinguish between face-to-face communication and other types of contact (fax, phone, etc.). In addition, the respondents were asked to specify the general purpose behind the face-to-face communication, for example, a coffee meeting for socialization or a work meeting to discuss progress. While we expected to see some differences in patterns between face-to-face and other types of contact, the analysis indicated very little communication took place between the employees. Our findings show that only 40% of employees met with their colleagues on a regular basis. These meetings were averaged to once a month, and less than 10% of the time the meeting involved some type of knowledge transfer.

The knowledge transfer process relies heavily on communication; often it involves several months of interaction between stakeholders (Szulanski, 1996). This notion is supported by evidence from prior studies, e.g., (Ghoshal, 1986; Ridderstrale, 1996), which suggest that higher levels of communication between stakeholders are likely to be associated with elevated levels of knowledge transfer. Cohen and Levinthal (1990) used the term "absorptive capacity" that refers to the ability to utilize new knowledge; they found communication to be a pre-requisite for the development of this capability. Furthermore, communication between participants in the knowledge transfer process helps in the creation of a supportive environment in which the transfer of knowledge can be facilitated easily (Kogut & Zander, 1992).

Minimum Interactions. While communication between employees is essential for the process of knowledge transfer, there are also varieties of interaction modes that can be used to enhance the process of knowledge transfer. These include technical meetings, extended visits, and joint training programs (Bresman et al., 1999). In general, it is suggested that the more such interactions were encouraged, the higher the level of knowledge transfer, e.g., (Haspeslagh & Jemison, 1991). Additionally, these interactions include social components that enhance normative integrations within the organization (Ouchi, 1980; Van Maanen & Schein, 1979). In this study, respondents were asked to indicate how often they traveled to technical meetings with other employees from the same department, how often they visited other employees' offices to discuss problems related to the projects they are working on, and how often they received visits from other employees in their offices for similar reasons.

We found that only visits and meetings between employees were reported as additional means of communication. However, these interactions were very limited, only 7% of respondents reported having meetings that were undertaken to address specific tasks or problems. Because there has been clear evidence of a lack of communication between employees of ORGUS, we identified having a lack of communication and interaction as a potential major contributor to the loss of knowledge after employees permanently departure the organization.

Documentation Problems. It is important to understand that the nature of the knowledge can have an impact on the knowledge transfer process. If the knowledge is considered to be tacit, which means it is not readily communicated in written or symbolic form (Reber, 1989), then it will be difficult to be transferred (Zander & Kogut, 1995). Bresman et al. (1999) suggest that tacit knowledge can be facilitated by intense interactions between the two parties. On the other hand, documented knowledge such as that found in patents and blueprints is likely to be easier to transfer between stakeholders, because it does not rely on a strong social bond between the parties.

Respondents were asked to indicate the extent to which the knowledge in the department was documented, and how easily can they acquire particulars to get the job done by studying a complete set of available blueprints. Our analysis of the responses shows that the documentation of knowledge at ORGUS was mainly done by individual efforts and only for certain tasks in a project. This led to having documentation for only small parts of the projects; and in addition, this documentation was in a non-uniformed fashion and scattered between employees who worked on the different tasks. Additionally, the documented knowledge in more than 25% of the reported cases was illegible by others. Undocumented knowledge in this case study was behind the loss of more than 45% of potentially transferable knowledge.

5.5.2 Phase II: Analyzing Implications of Turnover

After investigating the main contributors to the loss of knowledge, we evaluated the impact of that loss of knowledge in the department. In this phase, we want to reveal the consequences of knowledge loss because of turnover. We relied on intensive interviews using open-ended questions. For this phase, we interviewed the majority of the employees at ORGUS as well as few of their clients. The total number of partticipating employees that we interviewed was 279, and the total number of clients was 43. While exploring new emergent core categories, whenever possible, subsequent interviews were initiated with open-ended questions (Sedano, Ralph, & Péraire, 2016).

After collecting data from all 322 respondents, data analysis began with line-by-line coding as recommended by Charmaz (Charmaz, 2014). We reviewed the initial codes while reading the transcripts. Codes were then recorded into a spreadsheet, and we used the constant comparison to generate focused codes. Initial categories were formed from these focused codes which represented expressions made by multiple interviewees.

As responses were coded distinctively by two researchers, inter-rater reliability was assessed. When complete, the coding results were compared for agreement. Agreements between the investigators have been measured using Cohen's Kappa (k) method to estimate the interrater reliability (Banerjee et al., 1999). The scientifically acceptable level of inter-rater reliability has not been clearly determined. However, Banerjee et al. (1999) suggested that for the interpretation of kappa values, *"For most purposes, values greater than 0.75 or so may be taken to represent excellent agreement beyond chance"* (Banerjee et al., 1999, p. 604).

Therefore, the research investigators adopted the rigorous reliability measure of achieving higher than 0.75 on Kappa's scale. The coding scheme was pretested twice and cross-checked once. After the first pretest, the researchers revised the coding scheme and retested on the same set of data. This pre-testing approach assured clarity of revised scheme, as sources of a previous disagreement were spotlighted and resolved. When the second pretest was conducted, the third investigator cross-checked the tested set of data; as kappa value exceeded 0.75, the coding scheme was considered adequately reliable. Table 5.1 shows the results of the inter-rater reliability pre-tests.

Table 5.1– Inter-rater Reliability Estimates

Pretest	No. of Test Cases	Inter-rater Reliability
Pretest 1-1	15	0.52
Pretest 1-2	15	0.70
Pretest 2	20	0.89
Cross-check	20	0.92

We constantly compared new codes to existing categories to refine them and eventually generate emerging categories. We periodically audited each category for cohesion by comparing their codes (Sedano et al., 2016). The categories basically represent the classification of the responses into groups based upon the subject matter and the core message of each response. The grouped responses essentially represented trends. Within these patterns, the researchers also noticed that there were related categories. The researchers attempted to determine what broader underlying categories were present in the types of responses; the research participants were invited to group meetings, the categories and discussion led to better understanding and refined categories. Table 5.2 includes the refined emerging categories with examples of quotes resulting in each category.

Table 5.2 – Emerged Categories of Turnover Knowledge Loss Implications

Turnover Knowledge Loss Implications	Quotes/Examples
Higher Development Cost	*"We had to pay huge amount of money on overtime"* [Manager] "*The extra cost and time of fixing some bugs were unavoidable*" [Product Manager]
Loss of Expertise	*"Some defects could've been avoided if [xxx] did not leave us in the middle of testing"* [Team Member] *"We were counting on [xxx]'s extensive experience and knowledge of workflow automation to get some forms and workflows developed in short time. Leaving us before getting these workflows developed put us in unfavorable position with our client"* [Product Manager]
Irritated Client	*"We felt abandoned when [xxx] departed. She was very responsive and getting things done for us"* [Client] *"It was brought to our attention that our existing clients had started to look for alternative solutions"* [Manager]

Reduced Staff Morale	*"[xxx]'s departure broke the team spirit. She always spreader positive energy everywhere she went"* [Team Member] *"When [xxx] left, I had to work double shifts and did not take a single day off for more than 20 days. I would never accept doing this ever again"* [Team Member]
Delay of Service / Product Delivery	"*The extra cost and time of fixing some bugs were unavoidable"* [Product Manager] *"It took two months to develop a simple form that we needed; they promised to have the form ready within three business days for our review"* [Client]
Lower Service / Product Quality	*"We were focusing on getting the product delivered on-time, we did not pay much attention to technical debts"* [Team Member] *"We knew that we would produce more defects as a natural result of new members joining the team in the middle of the project without enough time to familiarize themselves with the code-base"* [Product Manager]
New-hire Onboarding	*"Getting a new-hire to a productive state requires us to spend a tremendous amount of time and effort."* [Product Manager] *"It took me some time to understand the nature of work, every time I ask a question, the typical answer is: Figure it out!"* [new-hire Team Member]

The responses, as charted in Table 5.2, were encoded into seven primary categories regarding the implications of knowledge loss from employee turnover. The researchers noted the frequency of interviewees who provided responses that contributed to the development of each code across the various categories. This allowed the researchers to understand better and formulate a set of specific issues related to knowledge loss from employee turnover. The respondents were invited to three more open group meetings, where the identified implications were presented and discussed with the research participants.

As shown in Figure 5.1, using Pareto Analysis (Besterfield, 2009), more than 86% of the participants reported "higher development cost" as a major implication of turnover, followed by "loss of expertise" as reported by more than 77% of the research participants, while the third highest reported implication was "Onboarding of New-hire" as indicated by more than 74% of the research participants. The less frequent responses were "reduced staff morale" reported by slightly over 46%, "lower quality of service" by just over 53%, "irritated client" by 57%, and "delay of service delivery" by slightly over 71% of research participants.

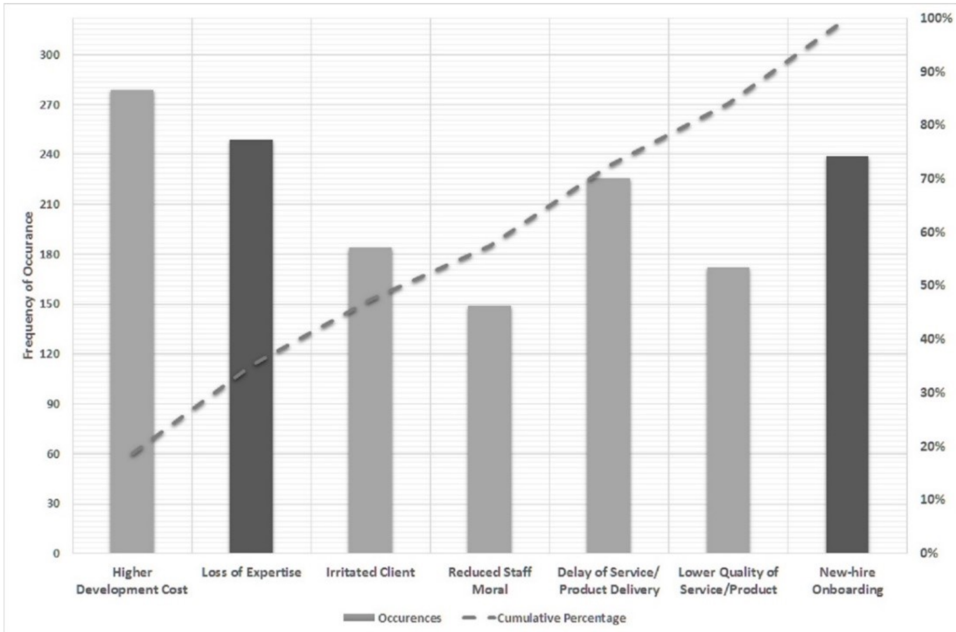

Figure 5.1 – Pareto Analysis of Categories

The outcomes suggested that there were interrelations between some of the emerged categories. Considering the narrow-focus of investigating the implications of knowledge loss that results from turnover, the "Irritated Client" and "Reduced Staff Morale" were both attributed to other categories. In other words, the core existence of these categories was based on influences of the remaining categories. Evidence can be concluded from the interview responses, for example, a client stated: "*We felt abandoned when [xxx] departed. She was very responsive and getting things done for us*". This was a clear response that supports "Irritated Client," however, as a result of "Loss of Expertise" who was getting things done for that particular customer.

Likewise, the core existence for each of these categories: "Higher Development Cost," "Delay of Service/Product Delivery," and "Lower Quality of Service/Product" was in fact influenced by the other remaining two categories, those were, "Loss of Expertise" and "New-hire Onboarding." As portrayed in Figure 5.2, the relationships between the seven categories are presented, and the identified major two categories have been presented in dark-grey boxes.

Our analysis of research participants' responses highlighted the possibility of having "New-hire Onboarding" and "Loss of Expertise" as the main sources of the other implications of knowledge loss from employees' turnover. The existence of problem-impact categories: higher development costs, irritated clients, reduced staff morale, delay of service/product delivery, and lower quality of service/product were not only found to be influenced by the loss of knowledge and new-hire onboarding, but they were also influencing the existence each other. As an example of such evidence, consider this quote from a product manager: "*The extra cost and time of fixing some bugs were unavoidable*." This statement shows the substantial impact on core existence between "higher development cost," "delay of product delivery," and "lower quality of product."

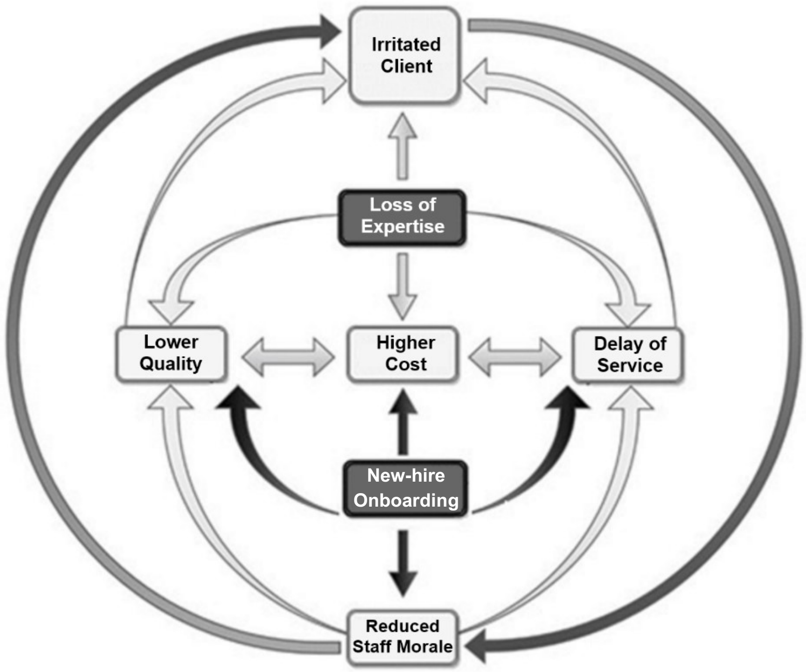

Figure 5.2– Analysis of Relationships between Categories

The discussions of the implications with the research participants and their inputs and feedback did not only help in narrowing down the categories related to employee turnover, but also was a forum to discuss how to use these implications to derive remedies and strategies that would articulate "how-to" create a system that sustains and continuously retains knowledge. The researchers theorized two main ways to address the aforementioned major problem areas. The first theorized-resolution was suggested to deal with "New-hire Onboarding" by developing practical onboarding strategies, and the second conceived-resolution was to deal with "Loss of expertise" by developing practical Knowledge Sharing Strategies that retains knowledge within the organization. The following phase provides more details about each of these strategies.

5.5.3 Phase III: Designing Remedies

After revealing the two key problem categories that are influencing other implications related to the loss of knowledge after employees' turnover at ORGUS, we suggested theoretical remedies to these two problem categories. First, to overcome the issue of onboarding new hires, we developed practical onboarding strategies. Second, to overcome issues of loss of expertise, we theorized a practical knowledge-sharing model.

5.5.3.1 Onboarding Strategies

Onboarding aims to move new-hires to a productive state quickly; this can be achieved by facilitating the acquisition of knowledge, skills, and behavior to fit in faster with the organization culture (Bauer, 2010). The knowledge and skills to be acquired do include not only the contextual and behavioral aspects but also the technical expertise and skills that are required by an employee to function effectively towards the organization's goals.

ORGUS claimed that a formal onboarding process existed. However, there was no written onboarding plan. In fact, the organization had a one-day orientation program where the new-hire would be handed a checklist that outlines basic setup tasks, such as setup of an email address, get an access card, and equipment configuration. The orientation program also has an "ad-hoc" tour for the new-hire to be introduced to the management team; a colleague from the same team would be asked to take the new-hire and go around visiting the offices of the management team members. The findings were reported to senior management during an official review meeting, where they requested further investigations to the challenges that the new hires were experiencing.

The research investigators, in cooperation with the Human Resource (HR) team of 5 employees, surveyed new hires who were working in the organization for less than 12 months as well as their immediate supervisor to investigate further the challenges they have faced during the early employment phases. The survey questions adopted from the SHRM's onboarding guide, the surveys were specifically focusing on assessing the four onboarding criteria (see 2.3 for descriptions of these criteria), these are: 1) Self-efficacy, 2) Role clarity, 3) Social integration, and 4) Cultural knowledge and fit.

The HR team sent two Likert-type voluntary surveys, the first survey targeted new hires and was sent to 113 candidates with a purpose of assessing the above mentioned onboarding criteria. The second survey targeted the immediate supervisor of new hires and was sent to 29 candidates with a purpose of evaluating the degree that the immediate supervisor feels their new hires meet these onboarding criteria. The reason behind surveying both, the new hires and their immediate supervisors, was that the reported self-perception by a new-hire on the degree that they feel they meet the criteria might not necessarily represent how the supervisor feels about how well the new hire has been performing with regard to the onboarding criteria.

Out of 144 email-survey requests, 99 completed surveys were received from 92 participants, decomposed into 81 new hires, and 18 immediate supervisors, where 7 employees had dual roles and responded to both surveys, once as a new hire of fewer than 12 months and once as a direct supervisor of the newly-hired employee(s). Responses "Strongly agree" and "Agree" on the surveys were aggregated and reported as positive signs towards each onboarding criteria per se. In addition, the reported results made a clear distinction between new hires in supervisory positions and new hires in nonsupervisory positions, because there were significant variances of perceptions between the two new-hire groups. Table 5.3 shows results of the perception surveys.

Table 5.3 – Onboarding Criteria Perceptual Results

		Self-efficacy	Role Clarity	Social Integration	Cultural Fit
Perception about self	New-hire	66.2%	86.5%	51.4%	21.6%
	Newly-hired Supervisor	71.4%	85.7%	100.0%	85.7%
Perception about subordinates	Manager	27.3%	72.7%	81.8%	36.4%
	Newly-hired Supervisor	42.9%	71.4%	100.0%	57.1%

As shown in Table 5.3, the results of the surveys show that new hires in supervisory positions have scored higher than new hires in non-supervisory positions in the four onboarding criteria; 100% of the new hires in supervisory positions feel more comfortable and socially accepted by their colleagues. They also feel that their newly hired subordinates were socially accepted and integrated; on the contrary, almost half of the new hires, in nonsupervisory positions, feel the opposite. The results also show that new hires in supervisory positions reported a higher degree of self-confidence than other new hires. This may be partially influenced by the levels of power and confidence that supervisory positions have over other positions. It was also interesting to note that only 27.3% of supervisors think their new hire subordinates feel confident with regard to self-efficacy. The cultural knowledge and fit were also repeatedly reported as poorly understood by the new-hires and their supervisors.

Table 5.3 shows that the organization had major deficiencies in at least two of the four onboarding criteria. Supported by survey results and internal feedback from the HR team, the decision was agreed upon by senior management to develop a formal onboarding program. This initiative was carried out as a separate project that lasted for 10 months. During this project, the primary investigator was involved as the project lead; he was also responsible for researching and benchmarking the onboarding best practices. The survey results have helped in scoping the project and directing the focus of the project to be on the onboarding criteria. Based upon the best practices as delineated by SHRM's onboarding practice guide (Bauer, 2010), a comprehensive onboarding program was successfully developed that moved the organization from being at the lowest rank with a passive onboarding level to the highest rank with a proactive onboarding level (Bauer, 2010). Table 5.4 shows some examples of strategies, tactics, building blocks, and associated level of the onboarding program.

Table 5.4 – Examples of Onboarding Implementation Strategies and Tactics

Strategy	Tactic	Onboarding Criteria	Onboarding level
Better motivation	Certificates have been awarded during the newly established organization's president bi-	Self-efficacy	Passive onboarding

	monthly meeting for high performers of new hires		
Boost confidence	Public recognition; ORGUS has used "Golden Name Plate" in recognition of high performers of new hires	Self-efficacy	Passive onboarding
Quicker adjustment	ORGUS introduced new onboarding checklist and orientation program (available from authors upon request) as well as a formal onboarding program.	Self-efficacy, cultural fit	High-potential onboarding
Boost job attitude	ORGUS has updated their Job Fact Sheets (JFS) and developed new ones for the new positions	Role-clarity	High-potential onboarding
Reducing role conflict	ORGUS promoted and encouraged the use of RACI matrix (Al-Baik & Miller, 2014)	Role-clarity, Social integration	High-potential onboarding
Social Enrichment	ORGUS established a *Social Committee* to start arranging social activities, such as lunches and coffee meetings for new hires	Social integration	High-potential onboarding
Communication & Involvement	ORGUS' senior management initiated "World Tour" where the CEO and Deputy CEO meet with employees (with focus on new hires) to get feedback on various work and non-work topics.	Self-efficacy, Social integration, Culture fit	Proactive onboarding
Communication & Openness	ORGUS implemented one-on-one (O3) meetings between employees and members of the management team. O3 is a 25-minute weekly meeting where each party has 10 minutes to talk about any non-work-related topic, then 5 minutes dedicated for Q/A. O3 aims to build and strengthen personal connections	Self-efficacy, Social integration, Culture fit	Proactive onboarding

Orientation Program	ORGUS started a program for new hires where they have several orientations sessions with key personnel. The most noteworthy orientation is the "Cultural Orientation", 2-hour session with either the CEO or the Deputy CEO for new hires, where they talk about the strategic goals, mission, vision, norms, values and provide advice on how to quickly and smoothly adapt with ORGUS' culture	Self-efficacy, Role Clarity Social integration, Culture fit	Proactive onboarding

Table 5.4 provides a short list of examples of strategies and tactics that have been used to develop a strong onboarding program. While some of these strategies and tactics were not exclusively developed or established for the sole purpose of serving the onboarding program, the program emphasized the importance of some of the tactics that were developed but were not used, for example, the JFS. Some other strategies and tactics were only drawn up for the sole purpose of the onboarding program, however, they were not mentioned for the sake of brevity, as they need contextual background and other details to be useful for the reader, for instance leveraging the existing organizational portal and developing online orientation and training programs to adjust new hires quickly.

5.5.3.2 Knowledge-Sharing Strategies.

The onboarding strategies are mostly related to ensuring that new hires are able to adjust, adapt, and settle into the organization. By contrast, the existing team members themselves need to find mechanisms to share the knowledge they accumulated over the lifetime of the projects they have been involved in. This would allow retaining the knowledge within the organization. Therefore, when one employee from the team leaves, there is a much lower likelihood that institutional knowledge will be lost; in fact, it will be reduced, but not totally lost. Two primary approaches were adopted to achieve this goal, the first approach aims at transferring knowledge between team members through adopting overlapping pair rotation (Sedano et al., 2016), and the second approach aims at systematically structuring the knowledge transfer sessions and documenting the knowledge through employing knowledge transfer Kaizen events (Al-Baik & Miller, 2017 :Forthcoming).

Overlapping Pair Rotation. The primary tactic of this approach is by ensuring that there is a rotation of the software development teams during various phases and processes of the product lifecycle. That is, different team members will have greater exposure to, and hence understanding of, the work on the project rather than having single team member working on the project. The result of having different employees working on a particular project task is that no one individual has all of the knowledge regarding that project-task. This type of structure is particularly critical within a team where losing one employee with exclusive knowledge could completely derail the entire team's ability to function.

The general process of *overlapping pair rotation* suggests that the employees working on a project rotate over time. Each employee will pair with another employee – within the same project team as well as cross-teams – and cooperate with him or her for a specific period of time to learn as much as possible about the codebase, architecture, and other artifacts. As illustrated in Figure 5.3, and for the purpose of simplification, only one member (arbitrarily named "Alex") from Project B team was presented to cross-pair with each other member of the Project A team. The individual names in figure 3 are all arbitrary and do not represent real team member names.

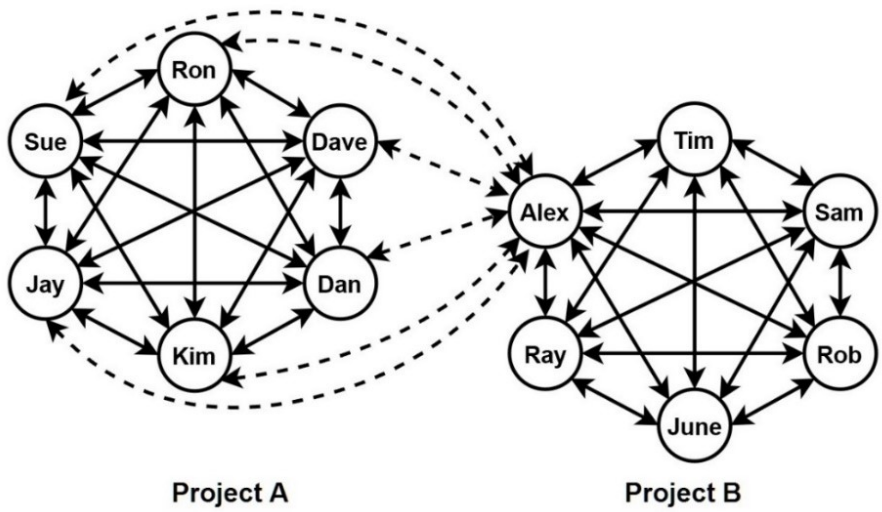

Figure 5.3 – Pair Rotation within and Cross Teams

The overlapping pair rotation has been claimed to decrease mistakes and produce a higher quality of work (Sedano et al., 2016). However, it does not provide a systematic mechanism on how knowledge transfer could be realized or conducted. Hence, the transfer of knowledge between the paired-employees has been systematically structured through implementing Knowledge Transfer Kaizen Events (Farris, 2005). Hence, combining both, overlapping pair rotation and the knowledge transfer Kaizen events would contribute to ensuring that everybody within the team knows what is happening with every project within the organization. Thus, when one individual leaves, the team will be better prepared to take responsibility for completing the tasks of that departed employee with minimum defects and high quality.

Knowledge Transfer Kaizen Events. Kaizen events are defined as short-term improvement activities that target the business performance and the employees' knowledge, skills, and attitudes (Farris, 2006). In this context, the Kaizen events were designed and arranged to discuss both, product and process issues, this design allows us to achieve two objectives: 1) transfer knowledge, and 2) explore improvement opportunities. It is recommended to start with an icebreaker activity at the beginning of each Kaizen event. The paired-employees would start by introducing themselves to the other party with a focus on their experience and skills. Then, each employee debriefs the other party on the tasks that he or she has been assigned to. Figure 5.4 depicts a flowchart for the full process.

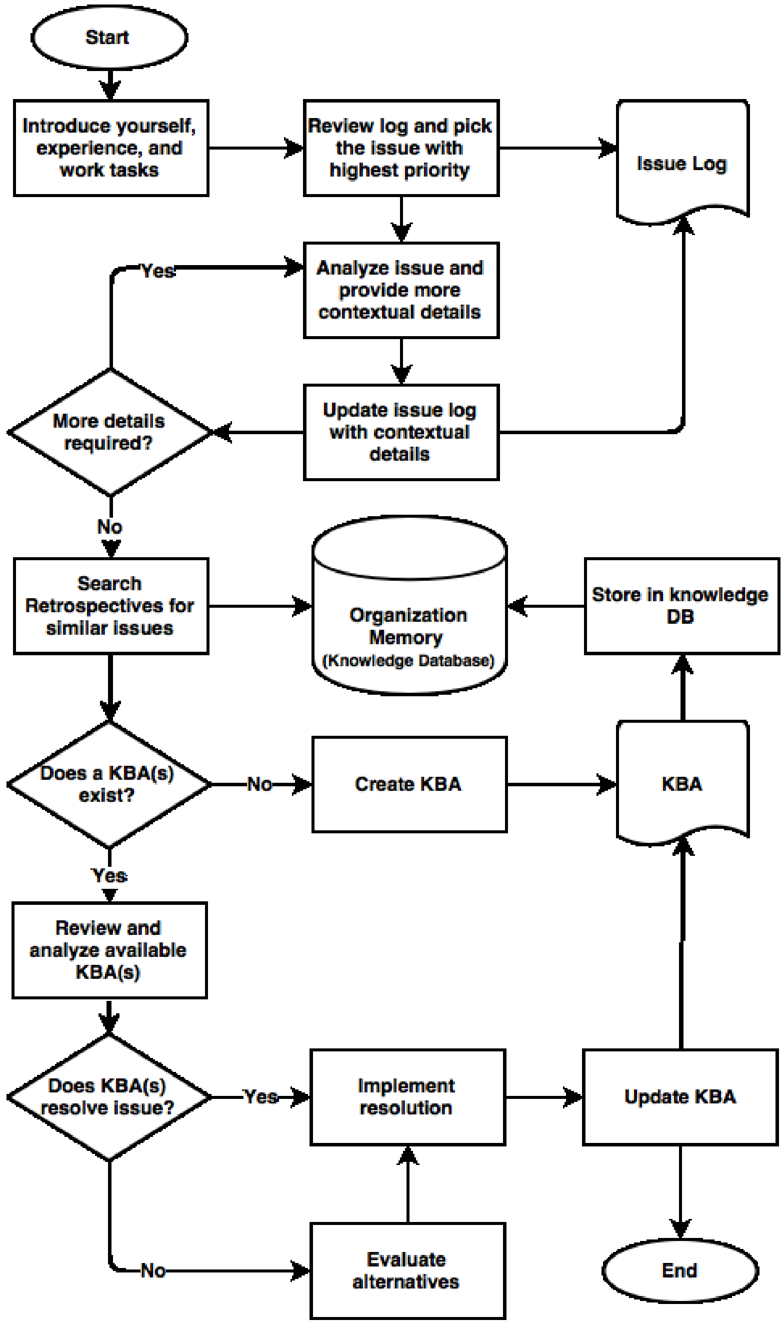

Figure 5.4– Knowledge Transfer Kaizen Event Flowchart

The event would then proceed with reviewing the issues log, picking up the highest priority issue to start working on as a team to resolve it. The employee responsible for that issue would explain it and provide more contextual details while the other party updates the log with the additional details. The transcriber may ask questions to understand the issue better; this approach entails a deeper analysis of the issue. Once the issue is better understood, the paired-employees search ORGUS' retrospectives for similar problems to shed some lights on previously implemented resolutions. The retrospectives would be stored in ORGUS' ticketing system in the form of a Knowledge Base Article (KBA).

KBA contains a succinct description of an issue, problem, or a task to be completed. KBA also includes significance and ramifications of the issue, a description of the implemented resolution, step-by-step procedures that guide the implementation process, cost and timelines, a list of concerned stakeholders, and lessons learned (what worked and why? What did not work and why?). If a KBA exists, the KBA would be further analysed to assess the validity of the documented resolution of the current issue and work would proceed to implementation; otherwise, a new KBA would be created with a detailed description of the issue, and as work progresses, KBA gets updated with any additional steps, modifications, or lessons learned.

The knowledge sharing strategies, overlapping pair rotation and knowledge transfer Kaizen events are beneficial for preparing successors when employees depart the organization. According to research studies, more than 25% of the workforce in North America transition from their positions on a yearly basis (Rollag et al., 2005). It has also been reported that 50% of new hires in supervisory positions from outside the organization fail within 18 months (Smart, 2005), while new hires in non-supervisory positions leave within 4 months (Krauss, 2010).

Finding successors for projects is critical to reducing institutional knowledge loss, as well as maximizing the likelihood of success of the new hire. Accordingly, monitoring performance during the execution of the knowledge transfer strategies in specific projects shows who are the best-performing individuals; this would then help to identify who is the most qualified employee to start working on a project if someone leaves. Invariably, this decreases the period of delays and increases product quality (Al-Baik & Miller, 2017 :Forthcoming; Sedano et al., 2016).

5.5.4 Phase IV: Implementing Remedies and Assess Results

Prior to applying the remedies described in phase III, we collected data on a number of metrics to be used evaluate the effectiveness of the suggested remedies. Data were generated from an intelligent business application that is connected to ORGUS' ticketing system. These metrics were identified as evidence of improving the learning and improvement habits within the context of ORGUS (Al-Baik & Miller, 2017 : Forthcoming). These metrics are:

1) Customer Satisfaction (CS), this refers to the customers' perception of how well a product or service has met their requirements and expectations.
2) Processing Time (PT), refers to the average time taken to develop a specific product feature or complete a requested service. PT does not yield delivery of feature or service to a client.
3) Lead Time (LT), refers to the time spent to produce a working feature or deliver a service, it includes the processing time of all the activities required from production to delivery.
4) Employee Average Learning Time (EALT), this is the time spent by an employee to learn about organization context, client, or the work itself.

Data for customer satisfaction were gathered using a five-point Likert-like scale, and the *grouped frequency distribution (GFD)* was calculated on a monthly basis; while the other three metrics (PT, LT, and EALT), were collected through auto-generated reports produced by business intelligence software that is connected to the ORGUS' ticketing system. As the suggested remedies are ongoing processes, we collected data for the aforementioned metrics, since before applying the remedies, starting in October 2013, and then aggregated the average for each quarter between January 2014 and September 2016, inclusive. By the end of each quarter onwards from Q1 2014, we compared data from the previous quarter to check for improvements based on each of the fours metrics. We conducted a nonparametric test (Mann-Whitney, U-test) to provide the effect size for the comparison.

The remaining metrics (PT, LT, and EALT) represent time series. In time series analysis, the values are realizations of random processes and hence can be auto-correlated, which implies that these values lack independence (Morell & Fried, 2009). Furthermore, time series show seasonal effects and so the values are not normally distributed, even if there is no monotonic trend (Morell & Fried, 2009). Therefore, we used the U-test for nonparametric analysis of the difference between quarters, and the Cliff's Delta to provide the effect size (Macbeth, Razumiejczyk, & Ledesma, 2011).

Cliff's Delta statistic is a nonparametric *effect size measure* that quantifies the amount of difference between two groups (Macbeth et al., 2011). Cliff's Delta values fall between -1 and +1. An effect size of +1.0 indicates that the first group observations are greater than the second group observations (no overlap). On the other hand, an effect size of -1.0 indicates that the second group observations are larger than the first group observations (Again, no overlap). An effect size of approaching 0.0 indicates that group distributions overlap completely and that there is no difference between groups. Whereas a non-significant p-value will result in the Cliff's Delta statistic to approach 0.0 (Macbeth et al., 2011). Table 5.5 summarizes findings of the analysis of the difference between first quarter (Q4 2013) and last quarter (Q3 2016) for the four metrics. Significant test results at a p-value of less than 0.05 have been marked bold.

Table 5.5– Customer Satisfaction Analysis of Difference between Q4-2013 and Q3-2016

Quarter	**CS**	**PT**	**LT**	**EALT**
Q4 2013	GFD = 4.13	Mean = 10.24 SD =2.29	Mean = 19.12 SD = 4.39	Mean = 6.31 SD = 2.32
Q3 2016	GFD = 4.87	Mean = 3.78 SD = 1.75	Mean = 10.41 SD = 2.18	Mean = 16.78 SD = 3.98
Mann Whitney U-test	U = 16380.5 p =.26	U = 16380.5 **p < .00**	U = 9017 **p < .00**	U = 24468 **p < .00**
Cliff's Delta	0.018 → 0	+1	+1	-1

The results show that all PT, LT, and EALT had the statistically significant difference between observations made before applying the remedies and the observations from the last quarter of collecting data (after 3 years). Cliff's Delta shows that both PT and LT decreased and that EALT increased over time. However, customer satisfaction showed no improvement over the same period. Figure 5.5 depicts the metrics improvement over time.

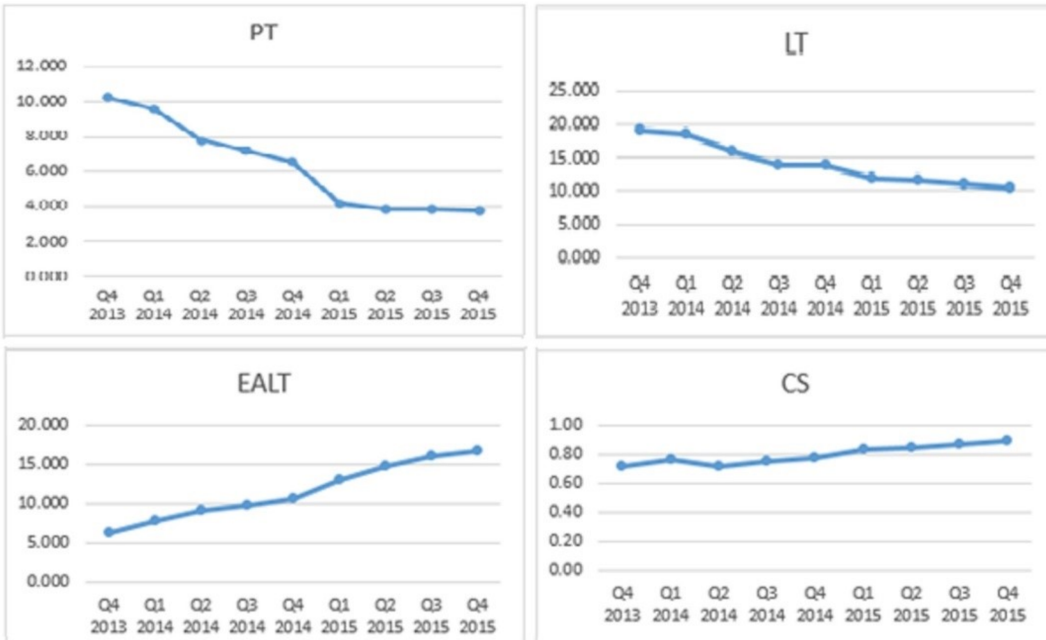

Figure 5.5 – Metrics Improvement over Time

Additionally, we tested for the improvement in the three factors mentioned in Phase I of this case study. These are employee communications, interactions, and documentation. We collected data for these factors once before applying the remedies and once at the end of the study (after 3 years). The *Grouped Frequency Distribution* (GFD) was determined to be more appropriate to analyze Likert-like data (I. E. Allen & Seaman, 2007). GFD was calculated as per Eq. 1:

$$GFD = L + \frac{N}{2} - Cfl \Big/ Fm \times Rw \qquad \text{Eq. 1}$$

Where *L* is the lower-class boundary of the median; *N* is population size; *Cfl* is the cumulative frequency of the groups before the median group; *Fm* is the frequency of the median group, and *Rw* is the width of the group range. Table 6 summarizes the findings on the improvement factors; significant test results at a *p-value* of less than 0.05 have been marked bold.

Table 5.6 – Test Analysis for Factors Data Q4-2013 and Q3-2016

Quarter	Employees' Communication	Employees' Interactions	Documentations

Q4 2013	GFD = 4.37	GFD = 3.38	GFD = 3.82
Q3 2016	GFD = 4.73	GFD = 4.47	GFD = 4.16
U-test	U = 15987 **p < .00**	U = 17211 **p < .00**	U = 16724 **p < .00**
Cliff's Delta	-1	-1	-1

All three factors showed statistically significant difference between observations from the beginning of this study and the end. Cliff's Delta indicates that observations of all three factors have increased over time and by results from 2016 are now completely distinct from the 2013 results. This evidence supports the effectiveness of the suggested remedies to overcome problems of employees' communication and interactions, as well as the documentation of knowledge throughout the process of working on a new service or product.

5.6 Threats to Validity

This study has its limitations. First, we cannot claim that the improvements resulted solely from implementing the proposed remedies, onboarding strategies and knowledge sharing strategies. The impact of other confounding factors could have significantly impacted the presented results, such as the successful implementation of other improvement initiatives and the development and application of the other learning models, such as IDKL (Al-Baik & Miller, 2017 :Forthcoming). This encompasses a threat to the *internal validity* of the study.

Second, the case study findings are difficult to generalize to other similar settings or different industries (Baxter & Jack, 2008). The study summarizes 3 years of work in one organization and is unique to its culture, practices, policies, and employees (study participants). The context of each organization per se might dramatically impact the results of similar undertakings. This constitutes a threat to the *external validity*. We encourage future researchers to conduct investigations in different settings to determine levels of finding generalizability.

Third, our study focused on knowledge loss due to employee turnover only while overlooking other sources of knowledge loss such as the loss of structural capital (Massingham, 2008). This pertains to a threat to the *construct validity;* and forth, the suggested remedies were based merely on the revealed two major implications of employees' turnover, this relates to the generalizability limitation of case studies. However, we suggest that issues related to employee turnover within software companies are similar in nature, and research studies addressing these issues have found similar problems within different organizations.

5.7 Conclusions

This case study allowed us to conduct an intensive analysis of the situation for a reasonably extended period of time. This does not only allow the researchers to define the problem, analyze the causes, and develop remedies strategies, but it also allows to implement the remedies and evaluates whether improvements have been realized. While many researchers have studied employee turnover and knowledge transfer distinctively, few have considered investigating predictors and impacts of knowledge loss from employee turnover. Further, there are limited efforts towards providing specific practices that can mitigate the problems stemming from the loss of knowledge due to employee turnover in knowledge-based organizations in general, and in IT/Software organizations in particular.

While additional research is needed to generalize our findings across organizations and industries, the current study does show the importance in understanding predictors, implications, and potential remedies of knowledge loss due to employee turnover. The case study provided significant findings throughout the different phases of the research. These results are expected to help practitioners by providing them with precautionary measures that can be taken to reduce the impact of knowledge loss when employees leave the organization.

It is the recommendation of this research for managers to encourage team collaborations, increase the frequency of technical meetings, and ensure the legibility of documenting knowledge at their organizations. This case shows that strong relationships and connections between the individual employees involved as well as the documentation of knowledge are also critical to successful knowledge transfer. Additionally, the suggested after-the-effect remedies may be beneficial to organizations that are undergoing similar problems of knowledge loss due to employee turnover.

6.0 Summary of Conclusions

In this thesis, the application of various fundamental Lean principles, concepts, and methods have been empirically and extensively investigated. The practices, for how these concepts and methods to be implemented in software development and IT services contexts, have been devised and evolved from within industrial software-centric settings. The application of lean to software development is not an overnight process. It requires reinterpretation and reinvention of the principles, methods, and practices to be better-suited in the new context.

Several research projects have been initiated that span over more than 5 years. Mixed research methods have been employed, but mainly action research and field studies. The advantage of implementing the research projects in one organization is that it allowed the research team to conduct an intensive analysis of the organization capabilities, culture, and process. This has helped not only to define problems, analyze the causes, and develop working models but it also allowed implementing the proposed models for a reasonably extended period of time to assess whether improvements have been realized.

Another advantage was the thorough understanding of the organization culture, politics, business process and key resources. This level of detail and understanding has helped in crafting working models by utilizing the organization's available resources and assets. Furthermore, the most significant advantage was the common knowledge and understanding of what has been implemented so far, and build on top of it. This would not be the case if the studies were to be implemented in various organizations.

The implementation of lean methods was, however, not without challenges. In the beginning, there had been a major resistance from the majority of the operational staff and some of the middle managers. This problem was alleviated by mainly emphasizing the advantages that lean can realize. Most of the projects were first piloted with one or two teams, specifically, the software development team. When proved to be successful, the initiative would then be promoted for organization-wide implementation and adoption.

Another major challenge that has been repetitively reported by several scholars and practitioners for more than 4 decades that software engineering and IT maintenance organizations have a tendency to disregard the sociocultural and organizational attributes of the improvement and development processes. Lean philosophy has implied the importance of the organizational, cultural, and human-side in software organizations. Lean methods emphasize the learning and advancement of the development teams, encourages empowerment and "on-spot" decision making, which implies a delegation of decision-making authority to operational-level staff.

Specifics to the principles and methods that have been investigated and reported in this thesis. The fundamental lean principle, eliminating waste, as declared in chapter 1, the resulting outcome of this research study has been a new taxonomy model to classify wastes in the software and IT contexts. The taxonomy proposes nine different categories, one of these categories is "Centralized Decision-Making," which may take various forms, however, yet another evidence is what has been identified in cooperation with the industrial partner "***Lack of delegation to empower the employees resulting in technical decisions made by non-technical people***."

Moving to the Lean's improvement philosophy, Kaizen, which requires employees to apply small improvements to the daily routines that they have been doing. This also implies an empowering aspect. However, it has even a stronger relation to the organization behavioral methods. The small improvements cannot be suggested and thought of without encouraging a culture of inquiry. Asking self and others "why" do we do what we do? How do we improve? How can we change? Challenging the norms, questioning policies and procedures, and even challenging the underlying values and beliefs, are just a few examples of what motivates the improvement processes and gear it towards success.

Organizations in the software engineering field must continuously seek improvements to sustain their competitive advantages and survive in rapidly changing market conditions. Lean touches on all aspects of the organization, sociocultural and organizational properties are crucial for the success of lean implementation. Methods of double-loop learning, reflective practices, and social communications and integrations are the missing ingredients from the success recipe for successful implementation of software process improvements.

Our recommendation for practitioners is to understand the organization's assets and capabilities to utilize them in developing practices that contribute to realizing Lean's principles and concepts. This should start with empowering and involving the employees. Organizations should encourage creativity and inquiry in the working environment to trigger the learning process that helps in realizing benefits and sustaining competitive advantages.

As for researchers, we suggest conducting more empirical studies to either validate or refute the analysis of the results and findings of this thesis. We conclude that lean promotes organization learning, which in turn, facilitates improving the overall performance of the organization; it can improve team productivity and increase the customer satisfaction. Additional research is needed to generalize our findings across organizations and industries, as well as to promote the need to consider the social and organizations properties during the implementation of the lean initiatives.

6.1 Future Research

Lean philosophy encompasses many principles and methods that will require decades to study and investigate. Our future research plan includes empirical investigations of the second pillar of the Lean House, that is "Jidoka". Jidoka has been translated to "automation with human touch". From the manufacturing perspective, the primary goal of Jidoka is to produce zero defects. While this may not be possible in the context of software development and IT maintenance, however, it could be feasible to introduce some mechanisms to reduce the number of defects.

Jidoka suggests stopping the production line as soon as a defect is detected, whereas some signal is triggered to stop the production process. In the meanwhile, the responsible engineer/employee works merely on eliminating the cause of the problem. If the problem persists, the entire team will start working on getting the issue resolved. The crucial question here is how Jidoka can be implemented in the context of software development? How will we objectively assess whether Jidoka produces positive results and hence is worthy of further investigation?

Our future research plan also includes developing a measurement scale that takes into consideration the social, technical, and organizational aspects of improvement initiatives. We finally encourage researchers to conduct empirical studies to explore the applicability of the models and methods presented in this thesis to other software settings. This would help to either validate or refute the analysis of the reported findings and results. More empirical studies are generally needed on the implementation of Lean thinking and the related methods such as Kaizen, Kanban, Jidoka, and continuous flow in software-centric contexts.

Bibliography

Abrahamsson, P. (2002). *The role of commitment in software process improvement*. Oulun yliopisto.

Addy, R. (2007). *Effective IT service management: to ITIL and beyond!* Springer-Verlag New York Inc.

Al-Baik, O., & Miller, J. (2014). Waste identification and elimination in information technology organizations. *Empirical Software Engineering*, *19*(6), 2019–2061.

Al-Baik, O., & Miller, J. (2015). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, *20*(6), 1861–1897.

Al-Baik, O., & Miller, J. (2016). Kaizen Cookbook: The Success Recipe for Continuous Learning and Improvements. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE.

Al-Baik, O., & Miller, J. (2017). Integrative Double Kaizen Loop (IDKL): Towards a Culture of Continuous Learning and Sustainable Improvements for Software Organizations. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*.

Alavi, M., & Leidner, D. (1999). Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS*, *1*(2es), 1.

Allen, D., & Bryant, P. (2012). *Managing employee turnover: myths to dispel and strategies for effective management*. New York, USA.

Allen, I. E., & Seaman, C. A. (2007). Likert scales and data analyses. *Quality Progress*, *40*(7), 64.

Allen, P., Ramachandran, M., & Abushama, H. (2003). PRISMS: an approach to software process improvement for small to medium enterprises. In *Quality Software, 2003 Proceedings. Third International Conference on* (pp. 211–214). IEEE.

Ammerman, M. (1998). *The root cause analysis handbook: A simplified approach to identifying, correcting, and reporting workplace errors*. SteinerBooks.

Ampoamah, P., & Samuel K. Cudjor. (2015). The Effect of Employee Turnover on Organizations (Case Study of Electricity Company of Ghana, Cape Coast). *Asian Journal of Social Sciences and Management Studies*, *2*(1), 21–24.

Anderson, D. (2003). *Agile management for software engineering: applying the theory of constraints for business results*. Prentice Hall Professional.

Anderson, D. (2010a). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

Anderson, D. (2010b). Panel: Why Agile, Why Lean? In P. and O. Abrahamsson N (Ed.), *LEAN ENTERPRISE SOFTWARE AND SYSTEMS* (Vol. 65, pp. 159–161).

Anderson, D. J. (2003). *Agile management for software engineering: Applying the theory of constraints for business results*. Prentice Hall Professional.

Antanovich, A., Sheyko, A., & Katumba, B. (2010). *Bottlenecks in the Development Life Cycle of a Feature - A Case Study Conducted at Ericsson AB*.

Argyris, C. (2002). Double loop learning, teaching, and research. *Academy of Management Learning and Education*.

Argyris, C. (2004). *Reasons And Rationalizations: The Limits to Organizational Knowledge*. OUP Oxford.

Argyris, C., & Schön, D. A. (1978). *Organizational Learning: A Theory in Action Perspective*. Addison-Wesley.

Armstrong, M. (2009). *Armstrong's handbook of human resource management practice*. *Human Resource Management*. London: Kogan Page.

Banerjee, M., Capozzoli, M., McSweeney, L., & Sinha, D. (1999). Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, *27*(1), 3–23. https://doi.org/10.2307/3315487

Basili, V. R. (1989). Software development: a paradigm for the future. In *Proceedings of the 13th Annual International Computer Software and Applications Conference, COMPSAC 89* (pp. 471–485). https://doi.org/10.1109/CMPSAC.1989.65127

Basili, V. R. (1992). Software modeling and measurement: the Goal/Question/Metric paradigm. *Quality*. https://doi.org/10.1016/S0164-1212(99)00035-7

Basili, V. R., McGarry, F. E., Pajerski, R., & Zelkowitz, M. V. (2002). Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory. In *Software Engineering, 2002. ICSE 2002. Proceedings of the 24th International Conference on* (pp. 69–79).

Bauer, T. N. (2010). Onboarding new employees: Maximizing success. *SHRM Foundation*. Retrieved from http://www.shrm.org/about/foundation/products/pages/onboardingepg.aspx

Bauer, T. N., Todd Bodner, B., Erdogan, D. M. T., & Tucker, J. S. (2007). Newcomer adjustment during organizational socialization: a meta-analytic review of antecedents, outcomes, and methods. *Journal of Applied Psychology*, *26*(3), 707.

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The Qualitative Report*, *13*(4), 544–559.

Bender, S., & Fish, A. (2000). The transfer of knowledge and the retention of expertise: the continuing need for global assignments. *Journal of Knowledge Management*, *4*(2), 125–137.

Berander, P., & Jönsson, P. (2006). A goal question metric based approach for efficient measurement framework definition. In *Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06*.

Berger, A. (1997). Continuous improvement and kaizen: standardization and organizational designs. *Integrated Manufacturing Systems*. https://doi.org/10.1108/09576069710165792

Besterfield, D. (2009). *Quality Control*. Pearson Education.

Bhagat, R. S., & Steers, R. M. (2009). *Cambridge handbook of culture, organizations, and work*. New York: Cambridge University Press.

Bhatt, G. D. (2001). Knowledge management in organizations: examining the interaction between technologies, techniques, and people. *JOURNAL OF KNOWLEDGE MANAGEMENT*, *5*(1), 68–75.

Biehl, R. E. (2004). Six Sigma for software. *IEEE Software*, *21*(2), 68–70.

Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005). Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES, 679(05).*, (May).

Boeg, J. (2011). Priming Kanban: A 10 Step Guide to Optimizing Flow in your Software Delivery System. Retrieved from https://www.infoq.com/minibooks/priming-kanban-jesper-boeg

Boeg, J. (2012). *Priming Kanban: A 10 step guid to optimizing flow in your software delivery system* (Second Edi). Chronografisk A/S.

Branham, L. (2005). *The 7 hidden reasons employees leave*. New York: American Management.

Brereton, P., Kitchenham, B. a., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583. https://doi.org/10.1016/j.jss.2006.07.009

Bresman, H., Birkinshaw, J., & Nobel, R. (1999). Knowledge transfer in international acquisitions. *Journal of International Business Studies*, *30*(3), 439–462.

Brockbank, A., & McGill, I. (2007). Facilitating Reflective Learning in Higher Education. *British Journal of Educational Technology*. https://doi.org/10.1111/j.1467-8535.2007.00720.x

Broza, G. (2012). *The Human Side of Agile - How to Help Your Team Deliver*. 3P Vantage Media.

Bryant, P. C., & Allen, D. G. (2013). Compensation, benefits and employee turnover: HR strategies for retaining top talent. *Compensation & Benefits Review*, *45*(3), 171–175.

Buono, A. F., & Bowditch, J. L. (2003). *The human side of mergers and acquisitions: Managing collisions between people, cultures, and organizations*. Beard Books.

Burdett, K. (1978). A theory of employee job search and quit rates. *The American Economic Review*, *68*(1), 212–220.

Burgess, K. (2007). *The role of the social factors in generating innovation within mature* *industry supply chains: a case study*.

Calderon, Natalja Nikitina, M., Kajko-Mattsson, & Nolan, A. J. (2015). Successful process improvement projects are no accidents. *Journal of Software: Evolution and Process*, *27*(11), 896–911.

Campanello, R. J. (2004). *HR optimization: from personnel administration to human and organizational capital development*. Massachusetts, USA: HRD Press, Inc.

Charmaz, K. (2014). Grounded theory in global perspective: Reviews by international researchers. *Qualitative Inquiry*, *20*(9), 1074–1084.

Chaudhuri, S. (1997). Kaizen at Morris Electronics: key to competitive success. In *Innovation in Technology Management. The Key to Global Leadership. PICMET '97*.

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, 128–152.

Combes, P.-P., & Duranton, G. (2006). Labour Pooling, Labour Poaching and Spatial Clustering. *Regional Science and Urban Economics*, *36*(1), 1–28.

Croasdell, D. T., Jennex, M., Yu, Z. Y. Z., & Christianson, T. (2003). A meta-analysis of methodologies for research in knowledge management, organizational learning and organizational memory: five years at HICSS. *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. https://doi.org/10.1109/HICSS.2003.1174253

Daniels, R. C. (1995). Performance measurement at the sharp end: driving continuous improvement on the shop floor. *Engineering Management Journal*, *5*(5), 211–214.

Davenport, T. H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Harvard Business Press.

De-Ste-Croix, L., & Easton, A. (2008). The Product Owner Team. In *Agile Conference, 2008. AGILE '08.* (pp. 274–279). IEEE. https://doi.org/10.1109/Agile.2008.94

DeMarco, T. (2011). All Late Projects Are the Same. *IEEE Software*, *28*(6), 103–104.

Dennis, P. (2007). *Lean Production simplified: A Plain-Language Guide to the World's Most Powerful Production System* (Second Edi). CRC Press.

Dingsøyr, T. (2005). Postmortem reviews: Purpose and approaches in software engineering. *Information and Software Technology*, *47*(5), 293–303. https://doi.org/10.1016/j.infsof.2004.08.008

Dittrich, Y., & John, M. (2007). Editorial: for the special issue on qualitative software engineering research. *Information and Software.*

Dittrich, Y., Rönkkö, K., Eriksson, J., Hansson, C., & Lindeberg, O. (2007). Cooperative method development. *Empirical Software Engineering*, *13*(3), 231–260. https://doi.org/10.1007/s10664-007-9057-1

Dixon-woods, M., Agarwal, S., Jones, D., Young, B., & Sutton, A. (2005). Synthesising qualitative and quantitative evidence : a review of possible methods. *Journal of Health Services Research & Policy*, *10*(1), 45–53.

Doolen, T. L., Worley, J., Van Aken, E. M., & Farris, J. (2003). Development of an Assessment Approach for Kaizen Events. *IIE Annual Conference. Proceedings*.

Droege, S. B., & Hoobler, J. M. (2003). Employee turnover and tacit knowledge diffusion: A network perspective. *Journal of Managerial Issues*, *15*(1), 50–64.

Duranton, G., & Puga, D. (2004). Micro-foundations of urban agglomeration economies. *Handbook of Regional and Urban Economics*, *4*, 2063–2117.

Durbin, A. J. (2000). *Applying psychology: Individual and organizational effectiveness*. Prentice Hall Professional.

Dybå, T. (2002). Enabling Software Process Improvement : An Investigation of the Importance of Organizational. *Enabling Software Process Improvement*, *7*(4), 387–390.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859.

Dybå, T., & Sharp, H. (2012). What's the Evidence for Lean? *IEEE Software*, *29*(5), 19–21.

Easterbrook, S., & Singer, J. (2008). Selecting empirical methods for software engineering research. *... Software Engineering*, 285–311.

Farris, J. A. (2006). *An Empirical Investigation of Kaizen Event Effectiveness: Outcomes and Critical Success Factors*. *ProQuest Dissertations and Theses*.

Fraser, S., Boehm, B., Jarkvik, J., Lundh, E., & Vilkki, K. (2006). How do Agile/XP development methods affect companies? In P. and M. Abrahamsson M and Succi G (Ed.), *EXTREME PROGRAMMING AND AGILE PROCESSES IN SOFTWARE ENGINEERING, PROCEEDINGS* (Vol. 4044, pp. 225–228).

Gayawali, D. R., Stewart, A. C., & Grant, J. H. (1997). Creation and utilization of organizational knowledge: an empirical study of the roles of organizational learning on strategic decision-making. *Academy of Management Proceedings*, *1997*(1), 16–20.

Gerlach, H., Rønde, T., & Stahl, K. (2009). Labor pooling in R&D intensive industries. *Journal of Urban Economics*, *65*(1), 99–111.

Ghoshal, S. (1986). *The innovative multinational: a differentiated network of organizational roles and management proceesses*. Harvard University.

Gibbons, R., & Katz, L. F. (1991). Layoffs and lemons. *Journal of Labor Economics*, *9*(4), 351–380.

Goldratt, E., & Cox, J. (1992). *The Goal: A Process of Ongoing Improvement*. Massachusetts, USA: North River Press.

Greenwald, B. C. (1986). Adverse selection in the labour market. *The Review of Economic Studies*, *53*(3), 325–347.

Griffeth, R. W., & Hom, P. W. (2001). *Retaining valued employees*. Sage Publications.

Harrison, R., & Reeve, F. (2002). *Supporting Lifelong Learning: Perspectives on learning* (Vol. 1). Psychology Press.

Harvard Business Review. (2011). *Management Tips: From Harvard Business Review*. Massachusetts, USA: Harvard Business Press.

Haspeslagh, P. C., & Jemison, D. B. (1991). *Managing acquisitions: Creating value through corporate renewal* (Vol. 416). New York: Free Press.

Hibbs, C., Jewett, S., & Sullivan, M. (2009). *The art of lean software development: a practical and incremental approach*. O'Reilly Media.

Hino, S. (2002). *Inside The Mind Of Toyota*. New York, USA: Productivity Press.

Hollnagel, E. (2010). *The ETTO Principle: Efficiency-Thoroughness Trade-Off. Why Things That Go Right Sometimes Go Wrong*. *Risk Analysis*. https://doi.org/10.1111/j.1539-6924.2009.01333.x

Hopp, W. J., & Spearman, M. L. (2004). To pull or not to pull: what is the question? *Manufacturing & Service Operations Management*, *6*(2), 133–148.

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.

IBM. (2016). Case Studies: Uncovering causal relationships in business metrics. Impact Diagrams. IBM Help Manual.

IIBA; International Institute of Business Analysis. (2009). *A Guide to the Business Analysis Body of Knowledger*.

Imai, M. (1986). *Kaizen: The Key to Japan's Competitive Success*. *Becoming lean Inside stories of US manufacturers*.

Ivarsson, M., & Gorschek, T. (2010). A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, *16*(3), 365–395. https://doi.org/10.1007/s10664-010-9146-4

Jaisingh, S. (2012). Lean Manufacturing [Class handout]. Edmonton: Department of Mechanical Engineering, University of Alberta.

Jensen, J., & Rodgers, R. (2002). Cumulating the intellectual gold of case study research. *Public Administration Review*, 235–246.

Johnson, J. (2002). ROI, It's Your Job! *Published Keynote Third International Conference on Extreme Programming*, *4*(3), 48.

Johnson, J. T., Griffeth, R. W., & Griffin, M. (2000). Factors discriminating functional and dysfunctional salesforce turnover. *Journal of Business & Industrial Marketing*, *15*(6), 399–415.

Jovanovic, B. (1979). Job matching and the theory of turnover. *Journal of Political Economy*, *87*(5), 972–990.

Juergens, E., Feilkas, M., Herrmannsdoerfer, M., Deissenboeck, F., Vaas, R., & Prommer, K.-H. (2011). Feature profiling for evolving systems. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on* (pp. 171–180).

Katcher, B. L., & Snyder, A. (2007). *30 Reasons Employees Hate Their Managers: What Your People May Be Thinking and What You Can Do About It*. New York, USA: AMACOM.

Kim, D. (1998). The link between individual and organizational learning. *The Strategic Management of Intellectual Capital*, *35*(1), 41–62. https://doi.org/10.1016/B978-0-7506-9850-4.50006-3

Kim, T.-Y., Cable, D. M., & Kim, S.-P. (2005). Socialization tactics, employee proactivity, and person-organization fit. *Journal of Applied Psychology*, *90*(2), 232.

Kinicki, A., & Kreitner, R. (2003). *Organizational behavior: Key concepts, skills & best practices*. McGraw-Hill/Irwin.

Kitchenham, B., & Charters. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Vol. EBSE Techn).

Kniberg, H. (2011). *Lean from the Trenches: Managing Large-Scale Projects with Kanban*. *The Pragmatic Bookshelf*. Pragmatic Bookshelf.

Kniberg, H., & Skarin, M. (2009). *Kanban and Scrum-making the most of both*. *Chemistry & …*. InfoQ. Retrieved from http://www.infoq.com/minibooks/kanban-scrum-minibook

Kogut, B., & Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *ORGANIZATION SCIENCE*, *3*(3), 383–397.

Kondo, Y. (2000). Innovation versus Standardization. *The TQM Magazine*, *12*(1), 6–10. https://doi.org/10.1108/09544780010287177

Koper, R. (2009). *Learning Network Services for Professional Development*. *Learning Network Services for Professional Development*. https://doi.org/10.1007/978-3-642-00978-5

Koudsi, S. (2000). Actually, it is like brain surgery. *Fortune*, *141*(6), 233–234.

Krauss, A. D. (2010). Onboarding the hourly workforce. *Poster Presented at theSociety for Industrial and Organizational Psychology (SIOP)*.

Ladas, C. (2009). *Scrumban-essays on kanban systems for lean software development*. Modus Cooperandi Press.

Liker, J. K. (2004). *The Toyota Way: 14 management principles from the World's greatest* *manufacturer*. New York, USA: McGraw-Hill.

Liker, J. K., & Hoseus, M. (2008). *Toyota culture: the heart and sould of the Toyota Way*. New York, USA: McGraw-Hill.

Lyons, N. (2010). Reflection and Reflective Inquiry: What Future? In *Handbook of Reflection and Reflective Inquiry*.

Macbeth, G., Razumiejczyk, E., & Ledesma, R. D. (2011). Cliff's Delta Calculator: A non-parametric effect size program for two groups of observations. *Universitas Psychologica*, *10*(2), 545–555.

Magee, D. (2007). *How Toyota Became #1: leadership lessons from the world's greatest car company*. Penguin.

Majchrzak, A., Rice, R. E., King, N., Malhotra, A., & Ba, S. (2014). Computer-mediated inter-organizational knowledge-sharing: Insights from a virtual team innovating using a collaborative tool.

Malik, K., & Choudhary, P. (2008). *Software Quality: A Practitioner's Approach*. New Delhi: Tata McGraw-Hill.

Manhart, M., & Thalmann, S. (2015). Protecting organizational knowledge: a structured literature review. *JOURNAL OF KNOWLEDGE MANAGEMENT*, *19*(2), 190–211.

Marcum, J. W. (1993). Kaizen Teian 2: Guiding Continuous Improvement Through Employee Suggestions. *National Productivity Review*.

Martin, A., Biddle, R., & Noble, J. (2009). XP Custo omer Practices : A Grounded Theory. In *Agile Conference, 2009. AGILE '09*. IEEE. https://doi.org/10.1109/AGILE.2009.68

Massingham, P. (2008). Measuring the impact of knowledge loss: more than ripples on a pond? *Management Learning*, *39*(5), 541–560.

McConnell, S. (1996). *Rapid development: taming wild software schedules*. Washington, USA: Pearson Education.

McCUNE, J. C. (1999). Thirst for knowledge. *Management Review*, *88*(4), 10.

McGovern, J., & Shelly, S. (2008). *The Happy Employee*. Adams Business.

Middleton, P., & Joyce, D. (2012). Lean software management: BBC worldwide case study. *IEEE Transactions on Engineering Management*, *59*(1), 20–32.

Middleton, P., & Sutton, J. (2005). *Lean software strategies: proven techniques for managers and developers*. CRC Press.

Moe, N. B., Dingsoyr, T., & Dyba, T. (2009). Overcoming Barriers to Self-Management in Software Teams. *IEEE SOFTWARE*, *26*(6), 20–26.

Morell, O., & Fried, R. (2009). On nonparametric tests for trend detection in seasonal time series. In *Statistical Inference, Econometric Analysis and Matrix Algebra* (pp. 19–39). Physica-Verlag HD.

Morgan, J., & Liker, J. K. (2006). *The Toyota Product Development System: Integrating People, Process, and Technology*. Taylor & Francis.

Mortensen, D. T., & Pissarides, C. A. (1999). New developments in models of search in the labor market. *Handbook of Labor Economics*, *3*, 2567–2627.

Mujtaba, S., Feldt, R., & Petersen, K. (2010). Waste and Lead Time Reduction in a Software Product Customization Process with Value Stream Maps. *2010 21st Australian Software Engineering Conference*, 139–148. https://doi.org/10.1109/ASWEC.2010.37

Murugaiah, U., Benjamin, S. J., Marathamuthu, M. S., & Muthaiyah, S. (2010). Scrap loss reduction using the 5-whys analysis. *International Journal of Quality & Reliability Management*, *27*(5).

Narayanan, V. K., & Chen, T. (2012). Research on technology standards: Accomplishment and challenges. *Research Policy*, *41*(8), 1375–1406.

Nonaka, I. (2008). *The knowledge-creating company*. Harvard Business Review Press.

Norrmalm, T. (2011). Achieving Lean Software Development Implementation of Agile and Lean Practices in a Manufacturing-Oriented Organization.

North, K., & Hornung, T. (2003). The Benefits of Knowledge Management - Results of the German Award "Knowledge Manager 2002." *J. UCS*, *9*(6), 463–471.

Nottonson, K. (2011). Yahoo ! Sports : Sprint 100 & Beyond. In *Agile Conference (AGILE), 2011* (pp. 252–255). IEEE. https://doi.org/10.1109/AGILE.2011.42

Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. CRC Press.

Oostvogels, N. (2012). *Kanban for skeptics: Clear answers to Kanban in software development*. Leanpub. Retrieved from https://leanpub.com/kanbanforskeptics

Ouchi, W. G. (1980). Markets, bureaucracies, and clans. *Administrative Science Quarterly*, 129–141.

Owen, R., Koskela, L., Henrich, G., & Codinhoto, R. (2006). Is agile project management applicable to construction? nderstanding and Managing the Construction Process: Theory and Practice. In *U-14th Annual Conference of the International Group for Lean Construction, IGLC-14* (pp. 51–66).

Packlick, J. (2007). The Agile Maturity Map A Goal Oriented Approach to Agile Improvement. In *AGILE conference (Agile), 2007* (pp. 266–271). IEEE.

Pascale, R., Millemann, M., & Gioja, L. (1997). Changing the way we change. *HARVARD BUSINESS REVIEW*, *75*(6), 126+.

Patton, M. Q. (2011). *Developmental evaluation: Applying complexity concepts to enhance* *innovation and use*. *Developmental evaluation: Applying complexity concepts to enhance innovation and use*. https://doi.org/10.1002/casp

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model, version 1.1. *IEEE Software*, *10*(4), 18–27.

Petersen, K. (2010). An empirical study of lead-times in incremental and agile software development. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Blekinge Institute of Technology, Sweden.

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2009.03.036

Petersen, K., & Wohlin, C. (2010). Software process improvement through the Lean Measurement (SPI-LEAM) method. *Journal of Systems and Software*, *83*(7), 1275–1287. https://doi.org/http://dx.doi.org/10.1016/j.jss.2009.03.036

Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study. *Empirical Software Engineering*, *15*(6), 654–693.

Petersen, K., & Wohlin, C. (2011). Measuring the flow in lean software development. *SOFTWARE-PRACTICE \& EXPERIENCE*, *41*(9, SI), 975–996. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/spe.975/full

Pettersson, F., Ivarsson, M., Gorschek, T., & Öhman, P. (2008). A practitioner's guide to light weight software process assessment and improvement planning. *Journal of Systems and Software*, *81*(6), 972–995. https://doi.org/http://dx.doi.org/10.1016/j.jss.2007.08.032

Petticrew, M., & Roberts, H. (2006). *Systematic reviews in the social sciences: A practical guide*. Blackwell Publishing Ltd.

Pink, D. (2011). *Drive: The surprising truth about what motivates us*. Penguin.

PMI; Project Management Institute. (2008). A Guide to the Project Management Body of Knowledge. PMI.

Poppendieck, B. M., Poppendieck, T., & Wesley, P. A. (2003). *Copyright Many of the designations used by manufacturers and sellers to distinguish their products*.

Poppendieck, M., & Cusumano, M. A. (2012). Lean Software Development: A Tutorial. *Software, IEEE*, *29*(5), 26–32.

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison Wesley.

Poppendieck, M., & Poppendieck, T. (2006). *Implementing lean software development: From concept to cash*. Addison-Wesley.

Power, D., & Sohal, A. S. (2000). Human resource management strategies and practices in Just-In-Time environments: Australian case study evidence. *Technovation*, *20*(7), 373–387. https://doi.org/http://dx.doi.org/10.1016/S0166-4972(99)00151-0

Pries, K. H., & Quigley, J. M. (2011). *Testing complex and embedded systems*. New York, USA: CRC Press.

Programmers Forum. (2012). What is the difference between requirements and specifications? Retrieved October 20, 2012, from http://programmers.stackexchange.com/questions/121289/what-is-the-difference-between-requirements-and-specifications

Przekop, P. (2006). *Six Sigma for Business Excellence: A Manager's Guide to Supervising Six Sigma Projects and Teams*. New York, USA: McGraw-Hill.

Rahim, M. A. (2010). *Managing conflict in organizations*. Transaction Publishers.

Rand, C., & Eckfeldt, B. (2004). Aligning strategic planning with agile development: Extending agile thinking to business improvement. In *Proceedings of the Agile Development Conference, ADC 2004* (pp. 78–82). IEEE. https://doi.org/10.1109/ADEVC.2004.5

Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology*, *118*(3), 219.

Reib, C. F. (2014). [E-text type]. Retrieved June 4, 2017, from http://www.personaler-online.de/typo3/nc/personalthemen/suche-in-artikeln/detailansicht/artikel/fluktuation.html

Reinertsen, D. G. (2009). *The principles of product development flow: second generation lean product development*. Celeritas Publishing.

Ridderstrale, J. (1996). *Global Innovation: Managing International Innovation Projects at Abb and Electrolux*. Stockholm: Institute of International Business, Stockholm School of Economics.

Ries, E. (2011). *The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Random House Digital, Inc.

Rogerson, R., Shimer, R., & Wright, R. (2005). Search-theoretic models of the labor market: A survey. *Journal of Economic Literature*, *43*(4), 959–988.

Rollag, K., Parise, S., & Cross, R. (2005). Getting new hires up to speed quickly. *MIT Sloan Management Review*, *2*(46), 35.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*. Retrieved from http://link.springer.com/article/10.1007/s10664-008-9102-8

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164. https://doi.org/10.1007/s10664-008-9102-8

Saks, A. M., Uggerslev, K. L., & Fassina, N. E. (2007). Socialization tactics and newcomer adjustment: A meta-analytic review and test of a model. *Journal of Vocational Behavior*, *70*(3), 413–446.

Salo, O., & Abrahamsson, P. (2007). An iterative improvement process for agile software development. *Software Process Improvement and Practice*, *12*(1), 81–100. https://doi.org/10.1002/spip.305

Schwaber, K., & Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. John Wiley & Sons.

Schyff, A. (2011). Lean Software Development.

Sedano, T., Ralph, P., & Péraire, C. (2016). Practice and perception of team code ownership. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering* (p. 36). ACM Press.

Seibert, K. W., & Daudelin, M. W. (1999). *The role of reflection in managerial learning: Theory, research, and practice*. Quorum Books.

Senge, P. (2006). *The fifth discipline: The art and practice of the learning organization*. Crown Pub.

Shah, R., & Ward, P. T. (2003). Lean manufacturing: context, practice bundles, and performance. *Journal of Operations Management*, *21*(2), 129–149.

Shah, R., & Ward, P. T. (2007a). Defining and developing measures of lean production. *Journal of Operations Management*, *25*(4), 785–805.

Shah, R., & Ward, P. T. (2007b). Defining and developing measures of lean production. *Journal of Operations Management*, *25*(4), 785–805.

Shankar, K., & Ghosh, S. (2013). A theory of worker turnover and knowledge transfer in high-technology industries. *Journal of Human Capital*, *7*(2), 107–129.

Shelton, C. (2010). *Business Process Reengineering with Commercial Off-the-Shelf Software*.

Shinkle, C. M., & Shihkle, C. M. (2009). Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP). In Y. and D. Dubinsky T and Adolph S and Sidky A (Ed.), *Agile Conference, 2009. AGILE '09.* (pp. 186–191).

Singer, J., Sim, S. E., & Lethbridge, T. (2008). Software engineering data collection for field studies. In *Guide to Advanced Empirical Software Engineering* (pp. 9–34).

Singh, J., & Singh, H. (2009). Kaizen Philosophy: A Review of Literature. *ICFAI Journal of Operations Management*.

Šmite, D., & Solingen, R. van. (2016). What's the True Hourly Cost of Offshoring? *IEEE SOFTWARE*, *33*(5), 60–70.

SO/IEC 15504:5. (2012). *Information technology -- Process assessment -- Part 1: Concepts and vocabulary*. *ISO/IEC*. Geneva.

Somaya, D., & Williamson, I. O. (2008). Rethinking the “war for talent.” *MIT Sloan Management Review*, *29*(4), 29–34.

Spitzer, D. R. (2007). Transforming performance measurement: Rethinking the way we measure and drive organizational success (pp. 160–176). AMACOM Div American Mgmt Assn.

Staats, B., Brunner, D. J., & Upton, D. (2011). Lean principles, learning, and knowledge work: Evidence from a software services provider. *Journal of Operations Management*, *29*(5), 376–390.

Staats, B., & Upton, D. (2009). *Lean Principles , Learning , and Software Production : Evidence from Indian Software Services*. *... Business School Technology & Operations Mgt.*

Stober, T., & Hansmann, U. (2010). *Agile Software Development: Best Practices for Large Software Development Projects*. Springer Berlin Heidelberg.

Stovel, M., & Bontis, N. (2002). Voluntary turnover: knowledge management–friend or foe? *Journal of Intellectual Capital*, *3*(3), 303–322.

Sveiby, K. E. (1997). *The new organizational wealth: Managing & measuring knowledge-based assets*. Berrett-Koehler Publishers.

Szulanski, G. (1996). Exploring internal stickiness: Impediments to the transfer of best practice within the firm. *Strategic Management Journal*, *17*(S2), 27–43.

Tapping, D., Luyster, T., & Shuker, T. (2002). *Value Stream Managemenet: Eight Steps to Planning, Mapping, and Sustaining Lean Improvements*. Taylor & Francis.

Terlecka, K. (2012). Combining Kanban and Scrum -- Lessons from a Team of Sysadmins. In *Agile Conference (AGILE), 2012* (pp. 99–102).

Turner, R., & Lane, J. A. (2013). Goal-question-Kanban: Applying Lean Concepts to Coordinate Multi-level Systems Engineering in Large Enterprises. *Procedia Computer Science*, *16*, 512–521. https://doi.org/10.1016/j.procs.2013.01.054

Upton, D., & Staats, B. (2011). Lean Knowledge Work. *Harvard Business Review*, (October). Retrieved from http://eureka.bodleian.ox.ac.uk/2097/

Urbancová, H., & Linhartová, L. (2011). Staff turnover as a possible threat to knowledge loss. *Journal of Competitiveness*, *3*(3).

Van Maanen, J., & Schein, E. H. (1979). Toward a theory of organizational socialization. In *BM Staw (Ed.), Research in organizational behavior* (pp. 209–264). Greenwich, CT: JAI Press.

van Solingen, R., Basili, V., Caldiera, G., & Rombach, H. D. (2002). Goal Question Metric (GQM) Approach. In *Encyclopedia of Software Engineering*. https://doi.org/10.1002/0471028959.sof142

Viana, D., Conte, T., Vilela, D., de Souza, C. R. B., Santos, G., & Prikladnicki, R. (2012). The influence of human aspects on software process improvement: Qualitative research findings and comparison to previous studies. In *Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on*. https://doi.org/10.1049/ic.2012.0015

Virtanen, P., Pekkola, S., & Päivärinta, T. (2013). Why SPI initiative failed: Contextual factors and changing software development environment. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on* (pp. 4606–4615).

Walsh, R. (2008). *Time Management: Proven Techniques for Making Every Minute Count*. Adams Business.

Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, *85*(6), 1287–1299. https://doi.org/10.1016/j.jss.2012.01.061

Waterhouse, P. (2008). Improving IT Economics: Thinking "Lean." CA White Paper.

Watkins, J. (2009). *Agile testing: how to succeed in an extreme testing environment*. Cambridge University Press.

Watkins, J., & Mills, S. (2011). *Testing IT: an off-the-shelf software testing process* (2nd Editio). New York, USA: Cambridge University Press.

Watson, G. E., & Gallagher, K. (2005). *Managing for Results*. CIPD Publishing.

Wickens, P. D. (1990). Production management. Japanese and British approaches. *IEE Proceedings A: Physical Science. Measurement and Instrumentation. Management and Education. Reviews*.

Womack, J. P., & Jones, D. T. (2003). Part I: Lean Principles. In *Lean Thinking: Banish Waste and Create Wealth in Your Corporation* (pp. 15–98). NY: Free Press.

Womack, J. P., Jones, D. T., & Roos, D. (1990). *The Machine That Changed the World*.

Xingjian, Y. (1992). A database design method for finite element analysis. *Computers & Structures*, *44*(4), 911–914.

Yap, M. (2006). Value based extreme programming. In *Agile Conference, 2006* (p. 8). IEEE.

Younker, J. (2009). *Foundations of agile python development*. Apress.

Zander, U., & Kogut, B. (1995). Knowledge and the speed of the transfer and imitation of organizational capabilities: An empirical test. *ORGANIZATION SCIENCE*, *6*(1), 76–92.

Zhang, Y. (2010). *Kanban Re-engineers Production Process In Åkers Sweden AB*. Mälardalen University, School of Innovation, Design and Engineering.

Appendices

Appendix (A): List of Included Studies

[S1] Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6), 1287–1299.

[S2] Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

[S3] Norrmalm, T. (2011). Achieving Lean Software Development Implementation of Agile and Lean Practices in a Manufacturing-Oriented Organization. Retrieved from http://uu.diva-portal.org/smash/record.jsf?pid=diva2:400627

[S4] Polk, R. (2011). Agile & Kanban in coordination. *In Proceedings - 2011 Agile Conference, Agile 2011* (pp. 263–268).

[S5] Anderson, D. J. (2003). *Agile management for software engineering: Applying the theory of constraints for business results*. Prentice Hall Professional.

[S6] Shinkle, C. M., & Shihkle, C. M. (2009). Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP). *In Agile Conference, 2009. AGILE '09.* (pp. 186–191). IEEE.

[S7] Betz, C. T. (2011). Chapter 1 - IT in a World of Continuous Improvement. *Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children*, 1–31.

[S8] Terlecka, K. (2012). Combining Kanban and Scrum -- Lessons from a Team of Sysadmins. *In Agile Conference (AGILE), 2012* (pp. 99–102).

[S9] Nikitina, N., & Kajko-Mattsson, M. (2011). Developer-driven big-bang process transition from Scrum to Kanban. *In Proceedings - International Conference on Software Engineering* (pp. 159–168).

[S10] Turner, Richard, Ingold, D., Lane, J. A., Madachy, R., & Anderson, D. (2012). Effectiveness of kanban approaches in systems engineering within rapid response environments. *Procedia Computer Science*, 8(0), 309–314.

[S11] Fisher, K. G., & Bankston, A. (2009). From Cradle to Sprint: Creating a Full-Lifecycle Request Pipeline at Nationwide Insurance. *In Agile Conference 2009, AGILE '09* (pp. 223–228).

[S12] Nikitina, N., Kajko-Mattsson, M., & Stråle, M. (2012). From scrum to scrumban: A case study of a process transition. *In 2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings* (pp. 140–149).

[S13] Turner, Richard, & Lane, J. A. (2013). Goal-question-Kanban: Applying Lean Concepts to Coordinate Multi-level Systems Engineering in Large Enterprises. *Procedia Computer Science*, 16, 512–521.

[S14] Petersen, Kai. (2010). *Implementing Lean and Agile Software Development in Industry*. Retrieved from Belkinge Institute of Techology's Dissertations and Theses Database (UMI No. 2010:04)

[S15] Kniberg, H., & Skarin, M. (2009). *Kanban and Scrum-making the most of both*. InfoQ. Retrieved from http://www.infoq.com/minibooks/kanban-scrum-minibook

[S16] Oostvogels, N. (2012). *Kanban for skeptics: Clear answers to Kanban in software development*. Leanpub. Retrieved from https://leanpub.com/kanbanforskeptics

[S17] Seikola, M., Loisa, H., & Jagos, A. (2011). Kanban Implementation in a Telecom Product Maintenance. *In Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference* (pp. 321–329).

[S18] Zhang, Y. (2010). *Kanban Re-engineers Production Process In Åkers Sweden AB*. Retrieved from Mälardalen University's Dissertations and Theses Database. (UMI No. 10064)

[S19] Anderson, D. J., & Garber, R. (2007). *A Kanban System for Sustaining Engineering on Software Systems*. Retrieved April, 01, 2013 from http://www.lean.org/FuseTalk/Forum/Attachments/Kanban%20for%20Software%20Development-Corbis.pdf

[S20] Medinilla, Á. (2012). Lean and Agile in a Nutshell. *In Agile Management SE - 2* (pp. 19–52). Berlin, Heidelberg: Springer Berlin Heidelberg.

[S21] Kniberg, H. (2011). *Lean from the Trenches: Managing Large-Scale Projects with Kanban*. The Pragmatic Bookshelf. Pragmatic Bookshelf.

[S22] Poppendieck, M., & Cusumano, M. A. (2012). Lean Software Development: A Tutorial. *Software, IEEE, 29(5)*, 26–32.

[S23] Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.

[S24] Petersen, K, & Wohlin, C. (2011). Measuring the flow in lean software development. *Software-Practice & Experience*, 41(9, SI), 975–996.

[S25] Heidenberg, J., & Porres, I. (2010). Metrics Functions for Kanban Guards. In *Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops* on (pp. 306–310).

[S26] Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the impact of Kanban on software project work: An empirical case study investigation. In *Proceedings - 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2011* (pp. 305–314).

[S27] Han, B., & Xie, J. (2012). Practical Experience: Adopt Agile Methodology Combined With Kanban For Virtual Reality Development.

[S28] Kinoshita, F. (2008). Practices of an agile team. In *Agile, 2008. AGILE'08. Conference* (pp. 373-377). IEEE.

[S29] Boeg, J. (2012). *Priming Kanban: A 10 step guide to optimizing flow in your software delivery system* (Second Edi.). Chronografisk A/S.

[S30] Sjøberg, D. I. K., Johnsen, A., & Solberg, J. (2012). Quantifying the Effect of Using Kanban versus Scrum. *Software, IEEE*, 29(5), 47–53.

[S31] Ladas, C. (2009). *Scrumban-essays on kanban systems for lean software development*. Modus Cooperandi Press.

[S32] Cocco, L., Mannaro, K., Concas, G., & Marchesi, M. (2011). Simulating Kanban and Scrum vs. Waterfall with System Dynamics. In *Agile Processes In Software Engineering And Extreme Programming* (Vol. 77, pp. 117–131).

[S33] Greaves, K. (2011). Taming the Customer Support Queue: A Kanban Experience Report. *In Agile Conference (AGILE), 2011* (pp. 154–160).

[S34] Ericsson, R., & Granlöf, A. (2011). *The effects of Kanban in software development teams : a study of the implementation at Sandvik*. Retreived from Dalarna University's Dessertation and Theses Database (http://du.diva-portal.org/smash/record.jsf?pid=diva2:519094)

[S35] Reinertsen, D. G. (2009). *The principles of product development flow: second generation lean product development*. Celeritas Publishing.

[S36] Hiranabe, K. (2007). *Visualizing Agile Projects using Kanban Boards*. InfoQ. Retrieved April 01, 2013, from http://www.infoq.com/articles/agile-kanban-boards

[S37] Turner, R, Madachy, R., Ingold, D., & Lane, J. A. (2012). Modeling kanban processes in systems engineering. In *2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings* (pp. 78–82).

Appendix (B): Quality Instrument

Quality Instrument

Study Identifier (Unique ID)

1 What is the study Type? ☐ ML ☐ SLR ☐ GL

If the type is Monographic, complete section (A), if the type is SLR, complete section (B), otherwise, complete section (C)

Section (A): Monographic

1	Does the study report any empirical research?	☐ Yes	☐ No	
2	Is there a clear description of the aim of the research?	☐ Yes	☐ No	☐ Partially
3	Is there a clear description of the study context?	☐ Yes	☐ No	☐ Partially
4	Is there an adequate justification for the research design?	☐ Yes	☐ No	☐ Partially
5	Is there an adequate description of the studied sample?	☐ Yes	☐ No	☐ Partially
6	Is there an adequate justification for the selected sample?	☐ Yes	☐ No	☐ Partially
7	Is there a clear description of the data collection?	☐ Yes	☐ No	☐ Partially
8	Is there a clear description of the data analysis?	☐ Yes	☐ No	☐ Partially
9	Are the findings of the study clearly stated?	☐ Yes	☐ No	☐ Partially
10	Is there sufficient data presented to support the findings?	☐ Yes	☐ No	☐ Partially
11	Are the research contribution and value adequately discussed?	☐ Yes	☐ No	☐ Partially
12	Are validity threats of the research adequately described?	☐ Yes	☐ No	☐ Partially

If any of the answers is "No" for any of the first two questions, exclude the paper. The scoring procedure is Yes=2, Partial=1, and No or Unknown=0, calculate total from 4-12, if the score is 12 or more, include. For more details about scoring, see "Ivarsson and Gorschek."

Section (B): Systematic Literature Review (SLR)

1	Are the review's inclusion and exclusion criteria described and appropriate?	☐ Yes	☐ No	☐ Partially
2	Is the literature search likely to have covered all relevant studies?	☐ Yes	☐ No	☐ Partially
3	Did the reviewers assess the quality/validity of the included studies?	☐ Yes	☐ No	☐ Partially

4	Were the basic data/studies adequately described?	☐ Yes ☐ No ☐ Partially

The scoring procedure is Yes=2, Partial=1, and No or Unknown=0, calculate total from 1-4, if the score is 6 or more, include. For more details about scoring see "Kitchenham - Guidelines."

Section (C): Gray Literature

Websites

1	Is it an official website for Lean/Kanban/Agile/Software organizations/societies?	☐ Yes ☐ No
2	Does it provide references for what has been written/described on the website?	☐ Yes ☐ No
3	Is the data on the website quoted/cited from one or more of the included primary studies?	☐ Yes ☐ No

If any of the answers is "No" for any of the first two questions or "Yes" for the third question, exclude the website.

Non-Websites (eBooks, Articles on InfQ)

1	Does the study report any empirical research?	☐ Yes ☐ No
2	Has the study been peer reviewed by an expert in the domain?	☐ Yes ☐ No
3	Does the study have "Known" author?	☐ Yes ☐ No
4	Does the author has h-index > 5? (using Google Scholar gadget)	☐ Yes ☐ No
5	Does the study has been cited by > 5 studies? (using Google Scholar gadget)	☐ Yes ☐ No

If the answer is "No" for the 1st questions, exclude the study. If the answer is yes, proceed to question 2, if the answer is "Yes", include the study, if the answer is "No," proceed to question 3 and 4, if the answer is "No" for both of the questions, exclude the study.

Appendix (C): List of Excluded Studies

Akbar, R., Hassan, M. F., & Abdullah, A. (2011). A Review of Prominent Work on Agile Processes Software Process Improvement and Process Tailoring Practices. In *Software Engineering And Computer Systems, PT 3* (Vol. 181, pp. 571–585).

Antanovich, A., Sheyko, A., & Katumba, B. (2010). *Bottlenecks in the Development Life Cycle of a Feature - A Case Study Conducted at Ericsson AB*.

Cockburn, A. (2006). *Agile software development: the cooperative game*.

Corona, E., & Pani, F. (2012). An investigation of approaches to set up a Kanban board, and of tools to manage it. In *SITE'12 Proceedings of the 11th international conference on Telecommunications and Informatics, Proceedings of the 11th international conference on Signal Processing,* pp. 53-58.

Hibbs, C., Jewett, S., & Sullivan, M. (2009). *The art of lean software development: a practical and incremental approach*.

Hiranabe, K. (2008). *Kanban Applied to Software Development: from Agile to Lean*. InfoQ. Retrieved April 01, 2013, from http://www.infoq.com/articles/hiranabe-lean-agile-kanban

Leffingwell, D. (2007). *Scaling software agility: best practices for large enterprises*.

Pettersen, J.-A., & Segerstedt, A. (2009). Restricted work-in-process: A study of differences between Kanban and CONWIP. *International Journal of Production Economics*, *118*(1), 199–207.

Sanders, A. (2007). Kanban Ground Rules Example for a Specific Team. Retrieved April 01, 2013, from http://aaron.sanders.name/agile-fashion/kanban-ground-rules-example-for-a-specific-team

Sanders, A. (2007). Naked Planning Explained – Kanban in the Small. Retrieved April 01, 2013, from http://aaron.sanders.name/agile-fashion/naked-planning-explained-kanban-in-the-small

Scotland, K. (2009). *Kanban, Flow and Cadence*. EMC Corporation. Retrieved from http://availagility.files.wordpress.com/2008/04/kfc-development-accu2009.pdf

Scotland, K. (2011). *Crystallising Kanban with Properties, Strategies and Techniques.* Retrieved from http://availagility.co.uk/2011/08/03/crystallising-kanban-with-properties-strategies-and-techniques/

Staats, BR., & Upton, D. (2009). Lean Principles , Learning , and Software Production : Evidence from Indian Software Services. *Harvard Business School*

Staats, BR., Brunner, D., & Upton, D. (2011). Lean principles, learning, and knowledge work: Evidence from a software services provider. *Journal of Operations Management*.

Takahashi, K. (2003). Comparing reactive Kanban systems. *International Journal Of Production Research*, *41*(18), 4317–4337.

Vilkki, K. (2010). When Agile Is Not Enough. In *Lean Enterprise Software And Systems* (Vol. 65, pp. 44–47).

Weber, B., & Wild, W. (2004). Application of lean and agile principles to workflow management. In *Extreme Programming And Agile Processes In Software Engineering, Proceedings* (Vol. 3092, pp. 258–261).

Appendix (D): Coding Scheme

Category	**Description**	**Coding Rule**
Kanban method	It refers to the method of incremental and evolutionary process change for organizations and institutions.	Any description of the steps that has been undertaken to implement the Kanban approach into an organization.
Kanban Board	This refers to visual tools in project management application that have a basis on Kanban for the visualization and optimization of the workflow. It also enables real-time teamwork and team collaboration	Any tool or means used to visualize the processes, tasks, or the workflow of the software product development.
Pull System	This refers to trigger the process of working on producing only what the customer needs by the quantity that is necessary and only when it is needed	The approach of starting the development process based upon a request from the customer.
Work Item	It refers to a task that needs to be visualized and added to the Kanban board.	Any feature, user stories, MMF or any other item that needs to be visualized and has to go through different phases through the development process.
Priority Queue	It refers to the list of ordered requirements or work items	Any term indicates the use of queuing, ordering or prioritizing tool. i.e. backlog or to-do items.
Inclusion Criteria	It refers to the rules that are used to make the decision of adding the work item to the visualization tool. Ideally, it should consider the value from a customer perspective.	Any norm, rule, or standard that is used to determine what work-item should be added to the board, i.e. through the use of MMF or Story point.
Work In Progress (WIP)	It refers to the concurrent number of work items that is allowed in each development phase.	Any rule, technique, to determine the number of concurrent work items.

Done Items	These are the items that have been completed on a particular phase and are ready to be released to the next stage.	The work items that have been through a specific development process and ready to be pulled by the next development process.
Reverse Items	It refers to items that are being moved backward on the Kanban board instead of moving forward.	Any work item that needs to go back to the previous development process, i.e. bug fixing.
Validated Learning	This variable refers to the process of learning what works and what does not operate for a specific organization. Its intent is to measure the work item's value from a business perspective	Validate learning is whereby an organization over time obtains a formula for obtaining, qualifying, and selling to customers in a particular target area. This can refer to the P-D-C-A cycle.
Cycle Time	Cycle time is the actual time required to complete one cycle of an operation for one work item from start to finish while the work item flows seamlessly through the development process.	Cycle time is a calculated based upon the time spent on producing a work item from the beginning until delivery without delays.
Lead Time	This is the time of completing one work item including delays between the initiation of a process and its execution, i.e. the time between making an order and the order being delivered.	Lead-time is calculated from the moment of requesting the work item until the moment that item is delivered to the customer.
Measurement tool	This usually a diagramming tool illustrates the cycle taken by a work item as it goes through the system.	Any measurement diagram is used to represent the performance of the Kanban.
Bottleneck	Bottlenecks are the constraints that are faced during the product development process.	Any work item that is slacking and slowing the process down.
Slack / Buffer	This refers to designed quantity of time applied to task schedule to protect the successful implementation of the required deliverable on time.	Any spare time put aside to help in delivering the work item on time.
Kanban Principles	These principles offer a framework that is chosen in the software development process by using the Kanban approach.	These are the guiding concepts of the Kanban approach.
Kanban techniques	This refers to production control system that is used to overcome the problems once occurred.	Any technique used to allow no surplus production and gives the workers the right to stop if they cannot keep up.
Avatar	A visual representation of work item details.	Any visual representation of the details related to a work item, this includes sticky notes, symbols or any other means.