LEAN ACCOUNTING COMES TO LEAN SOFTWARE DEVELOPMENT

A Dissertation
Submitted to the
Temple University Graduate School

In Partial Fulfillment
of the Requirements for the Degree
Executive Doctorate of Business Administration

by
Thomas W. Stone
Temple University, Fox School of Business
May, 2018

Review Committee Members:

Sudipta Basu, Advisory Chair, Accounting
Paul A. Pavlou, Office of Research, Doctoral Programs, and Strategic Initiatives
Stuart M. Schmidt, Human Resources Management
Mark E. Gershon, External Reader, Marketing and Supply Chain Management

ABSTRACT

I argue that lean software development firms become more productive if they adopt and align lean managerial accounting systems with lean software development processes. I conduct two experiments on retraining and coaching of software development teams that have used lean and agile software development practices, demonstrating that these practices significantly improve productivity compared to control groups that did not receive this retraining and coaching. In a third experiment, I expand on this theme by introducing lean accounting productivity metrics to a treatment group of software developers. Team leaders actively use these metrics as quantitative “retrospectives” in team meetings to review past performance and identify areas for process improvement. Four months after these metrics are introduced, I measure their impact on the treatment group productivity and also survey the group to determine how these metrics affect employee attitudes and productivity compared to a control group that was not trained in use of these metrics for team meetings. The results indicate that introduction of lean accounting metrics does not impact employee attitudes and understanding of processes and metrics, nor does it improve productivity in the near term. Discussions with management indicate that retraining and coaching immediately improve productivity since they are directed at remedying specific operational and process issues. Using lean accounting metrics to impact team productivity and employee attitudes is more foundational and likely requires a longer period of exposure and learning. The experimental site is a large publicly traded software firm that uses lean and agile software development practices.

Key Words: Lean Accounting, Training, Coaching, Software Development, Productivity

DEDICATION

I dedicate this dissertation to my wife Suzanne, who has given me every means of support and encouragement during my three-year journey toward completion of the Executive Doctorate in Business Administration. This degree is truly a joint undertaking.

When I switched careers from the private sector to academia, Suzanne encouraged me to pursue a doctoral program so that I could realize the full potential of this new profession. Being out of school for so many years, I started the program with some intrepidation, but her confidence in my abilities gave me the courage to take the plunge. She has been by my side, patiently enduring the endless evenings when I needed to study or write, and providing the emotional and financial security that allowed me to pursue this degree. I could not have had a better partner for this endeavor, or indeed for the wonderful life we have built together.

I also want to acknowledge the support of my three sons, Tom, Matt and Eric, who have continually expressed their pride in my pursuit of this degree. Their own academic successes led me to believe that there must be some of those same abilities in their own father. I am enormously proud of all of them. A special note of recognition goes to Tom, who will also receive his doctorate this spring from Temple University. I know we inspired each other, especially during the dissertation-writing phase.

ACKNOWLEDGMENTS

I thank several members of Cerner Corporation's management team who enthusiastically supported this research and graciously invested substantial amount of time and expertise, allowing me to investigate a topic I have been passionate about for the past several years.

John Glaser provided my entry into Cerner while Max Rienig and Shahzar Zafar helped me identify those managers best able to help me with the research. Howard Meyers, Nick Conti, Bob Marini and Charlie Villare were instrumental in providing me access to the data, acting as liaisons with the development teams, helping to refine the metrics and the survey, and providing key insights into the results of the analysis. They were available anytime I asked to meet, and frequently hosted me at Cerner's offices. Their support has been invaluable. I hope to continue to work with them on extensions of this analysis.

I also thank Dan Vacanti who generously allowed me to use his consultancy firm's Actionable Agile tool to analyze the Cerner data. Dan was also available anytime I had questions about the tool. I first met Dan while working with Bennet Vallet at Siemens Healthcare IT (now Cerner), and I must also acknowledge Bennet's early interest in involving me in this topic as he saw the need to make productivity measures in software development more transparent to both staff and senior management. Dan also recognized this need. Both have been very supportive of this research.

Finally, I recognize the faculty and staff of the Fox School of Business who have guided me through the doctoral program. Their support has been remarkable. In particular, I want to thank my dissertation committee, Paul Pavlou, Stuart Schmidt and

my external reader, Mark Gershon. Most importantly, I want to acknowledge the Sudipta Basu, my dissertation advisor. His detailed review of my innumerable research paper drafts and this dissertation have greatly improved both the content and readability of my work. He was always available for a meeting and his insights made this a substantially stronger dissertation.

TABLE OF CONTENTS

ABSTRACT	ii
DEDICATION.....	iii
ACKNOWLEDGMENTS	iv
LIST OF FIGURES	viii
LIST OF TABLES.....	x
CHAPTERS	
1. INTRODUCTION	1
2. RESEARCH MOTIVATION	4
Emergence of Lean Production in Manufacturing.....	5
Emergence of Lean Production in Software Development.....	12
Transition of Lean Accounting: Manufacturing to Software Development.....	16
3. LITERATURE REVIEW	19
Lean Software Development	19
Training and Coaching.....	20
Lean Accounting	25
4. CONCEPTUAL MODELS AND HYPOTHESES	29
Retraining and Coaching.....	29
Lean Accounting	32
5. RETRAINING AND COACHING EXPERIMENTS.....	37
Description of Necessary Data	37
Methodology for Data Collection and Analysis	39

Discussion of Results	40
Retraining Experiment Results	40
Coaching Experiment Results	53
6. LEAN ACCOUNTING EXPERIMENT	61
Description of Necessary Data	61
Methodology for Data Collection and Analysis	62
Discussion of Results	67
Productivity Results	68
Survey Results	77
7. CONCLUSIONS	81
8. LIMITATIONS OF STUDY	85
9. EXPECTATION OF RESEARCH CONTRIBUTION	87
BIBLIOGRAPHY	90

APPENDICES

A. SURVEY QUESTIONS AND RESPONSES	95
B. STUDY PROTOCOL AT RESEARCH SITE	96
C. BASIS FOR SURVEY QUESTIONS	98
D. IRB SUBMISSION	100

LIST OF FIGURES

Figure 1: The Traditional Mass Production Process.....	7
Figure 2: Lean Manufacturing	8
Figure 3: Value Stream and Costing.....	10
Figure 4: Flow Efficiency	11
Figure 5: Cycle Time, Work-In-Process, Throughput	12
Figure 6: Traditional Waterfall Software Development Process.....	14
Figure 7: Lean Software Development Process	15
Figure 8: Timeline: Lean Accounting – Manufacturing and Software Development	17
Figure 9: Conceptual Model with Measurements	34
Figure 10: Research Protocol: Retraining/Coaching.....	37
Figure 11: Experiment One: Treatment Group Scatterplot.....	41
Figure 12: Experiment One: Control Group Scatterplot	42
Figure 13: Experiment One: Treatment Group CFD - Pre-Retraining	45
Figure 14: Experiment One: Treatment Group CFD - Post-Retraining	46
Figure 15: Experiment One: Control Group CFD - Pre-Retraining.....	47
Figure 16: Experiment One: Control Group CFD - Post-Retraining	47
Figure 17: Experiment Two: Treatment Group Scatterplot	54
Figure 18: Experiment Two: Control Group Scatterplot.....	55
Figure 19: Experiment Two: Treatment Group CFD - Pre-Coaching	57
Figure 20: Experiment Two: Treatment Group CFD - Post-Coaching.....	57
Figure 21: Experiment Two: Control Group CFD - Pre-Coaching	58
Figure 22: Experiment Two: Control Group CFD - Post-Coaching.....	58

Figure 23: Research Protocol: Lean Accounting	61
Figure 24: Cycle Time Report.....	64
Figure 25: Flow Efficiency Report.....	65
Figure 26: Throughput Report	65
Figure 27: Forecast Accuracy Report	66
Figure 28: Work-In-Process Report (via monitor)	67
Figure 29: Experiment Three: Treatment Group Scatterplot	69
Figure 30: Experiment Three: Control Group Scatterplot.....	69
Figure 31: Experiment Three: Treatment Group CFD - Pre-Metric Rollout.....	70
Figure 32: Experiment Three: Treatment Group CFD - Post-Metric Rollout	71
Figure 33: Experiment Three: Control Group CFD - Pre-Metric Rollout.....	71
Figure 34: Experiment Three: Control Group CFD - Post-Metric Rollout	72

LIST OF TABLES

Table 1: Construct Definitions: Retraining, Coaching Productivity	31
Table 2: Construct Measurement: Retraining, Coaching Productivity.....	31
Table 3: Construct Definitions: Lean Accounting Productivity.....	35
Table 4: Construct Measurement: Lean Accounting Productivity	35
Table 5: Construct Measurement: Lean Accounting Practices	36
Table 6: Construct Measurement: Employee Understanding of Lean Practices.....	36
Table 7: Experiment One: Summary of Results - Cycle Time.....	43
Table 8: Experiment One: Summary of Results - Flow Efficiency, Throughput, WIP	48
Table 9: Experiment One: Results by Team.....	51
Table 10: Experiment Two: Summary of Results - Cycle Time	55
Table 11: Experiment Two: Summary of Results - Flow Efficiency, Throughput, WIP ...	59
Table 12: Experiment Two: Results by Team	60
Table 13: Experiment Three: Summary of Results - Cycle Time	73
Table 14 Experiment Three: Summary of Results - Flow Efficiency, Throughput, WIP:..	74
Table 15: Experiment Three: Results by Team	76
Table 16: Summary of Survey Results	78

CHAPTER 1

INTRODUCTION

Throughout the 1980's and 1990's, many U.S. manufacturers changed from centralized mass production processes that “pushed” work through factories, reducing per unit cost, to decentralized lean processes that “pulled” work just-in-time to reduce work-in-process and inventories. Whereas traditional manufacturing views work-in-process as an asset and a necessary buffer in the production process, lean practices consider work-in-process as inefficient. As these production processes changed, the old centralized accounting systems used to support mass manufacturing no longer supported the new decentralized lean processes. As a result, lean accounting methods were developed that focused on value contribution at the factory floor level and made this data transparent, graphic, accessible, and actionable for workers.

Since the early 2000's, the software development industry has undergone a similar change in production processes, moving from highly centralized “waterfall” development processes to nimble, decentralized, and highly iterative lean/agile processes. However, also similar to manufacturing, reporting systems and related metrics have not kept pace with operational changes. My dissertation examines how productivity in lean software development can be improved. First, I analyze how much retraining and coaching development teams about lean practices improves software development productivity. Second, I introduce lean accounting measures into the workplace to evaluate their impact on team performance and employee attitudes.

I study a natural experiment of a software development team that was re-trained in lean software development practices and compare productivity before and after the

retraining. I find that cycle time fell by 81% after the retraining; cycle time for a control group dropped by 23% over a comparable period. When I analyzed a second natural experiment on coaching, the treatment group realized a 52% reduction in cycle time, while the control group decreased cycle time 15%. Both experiments indicate that greater focus on lean practices improves software development productivity considerably.

I also conduct an experiment where I introduce lean accounting metrics to a group of software development teams. These metrics are highly visible, relevant to the employees' work, understandable, graphic, and actionable. The reports are actively used in team meetings as "retrospectives"—reviews of past performance to decide what practices to maintain and what to change. Previously, these meetings had been largely qualitative. The new lean accounting metrics provide quantitative data for the team to assess performance and modify processes and resource allocation. The treatment and control groups were surveyed before the rollout of the new metrics. Following a four-month exposure to the metrics, I re-surveyed teams to compare changes in adoption of lean practices, in employee attitudes toward work and teammates, and understanding of the value of operational metrics. I also measured changes in productivity for both the treatment and control groups. I found a 32% increase in cycle time for the treatment team compared to a 12% decline for the control team, which is opposite my prediction. Management indicates that a favorable productivity impact is likely to take longer than the four-month period studied. Also, the survey revealed no material change in employee attitudes and understanding and use of lean practices when compared to a control group.

Through these experiments, I have sought to establish the value of lean/agile software development practices and study how lean accounting practices can improve

software developers' attitudes and productivity. Study results indicate that reinforcing existing lean development skills through timely interventions of retraining and coaching can provide immediate benefits in improved software development productivity.

However, introducing new lean accounting metrics into the educational process does not improve productivity or employee attitudes in the short term. Discussions with management suggest that this new data requires more time and consistent exposure before it can add value. Indeed, some of the lean accounting metrics identify needed changes that may not be implemented until subsequent development cycles.

CHAPTER 2

RESEARCH MOTIVATION

As venture capitalist Marc Andreessen (2011) noted, “software is eating the world.” The software industry has grown rapidly and disrupted many industries. Shapiro (2014) estimates that 15% of labor productivity gains in the U.S. from 2004 through 2012 are attributable to the impact of software, and business investment in software has grown twice as fast as all fixed investments from 1990 through 2012. Customers increasingly demand better and faster software development. These improvements can be achieved by shorter, quicker production cycles using decentralized management, reduced work-in-process, and shorter queues through just-in-time (JIT) practices, otherwise known as lean production methods. In order to speed up production, the software industry is changing just like the manufacturing sector did in the late 20th century.

The growth of lean software production is attributable to an increasing demand for adaptable software solutions serving a variety of smaller industries and businesses. These software products differ from earlier monolithic solutions that served large commercial and governmental customers, and as such require faster production. Delivery has sped up with evolving software solutions such as Software as a Service (SaaS) that automatically “pushes” frequent updates to centrally hosted websites, letting the customer access the system directly from the software firm’s web-based platform (Bennett, 2000).

Toyota and other lean manufacturers have found that lean practices need continual reinforcement, and now deliver training in a continual flow rather than as a batch process with long but infrequent periods of formal training. Training takes place in short, iterative cycles, much like the lean production process itself. Learning is “just-in-time” and highly

tailored to immediate workplace needs. Short but regular training exercises are vital for growing productivity in lean organizations and as a defense against inevitable skills erosion when training is deferred. This paper examines the value of lean skills retraining by examining the impact of this training on the productivity of lean software development teams.

Lean manufacturers have also incorporated lean accounting metrics to complement and reinforce lean practices. These metrics are visual, accessible, understandable and actionable to the production staff. As employees are exposed to and understand these metrics, they are better able to adjust processes, prioritize tasks and align resources to improve productivity. Lean accounting measures provide a critical component of the lean training and coaching process.

Just as software development organizations have benefitted from adapting lean production processes in recent years, lean accounting metrics are likely needed to fully realize the potential of lean process productivity gains. However, realizing these gains may require more extensive exposure to these metrics than tested in this analysis.

Emergence of Lean Production in Manufacturing

Mass automobile production using a moving assembly line was introduced by Ford Motors in 1913 at its Highland Park plant and improved by General Motors in the 1930's to handle multiple automotive models (Womack, Jones & Roos, 2007). Mass production typically starts with raw or relatively unfinished materials that move through several production stages in which highly standardized machine and labor processes create intermediate work-in-process. These streams of work-in-process are merged to

assemble interchangeable and identical components into finished goods. At each step, work-in-process is usually built up so that the production line never stops for lack of needed inputs. For complex products, many intermediate components stay as work-in-process that requires considerable financing. Also, due to a long production cycle and long times to switch between products, complex goods are batch-produced based on noisy forecasts of future demand, requiring maintenance of finished goods inventory to meet any unexpected demand. Even when quality issues arise, large-scale production flow is maintained because halting production disrupts the integrated mass production process. Therefore, quality issues are resolved after production, which causes greater waste and complications than if the issues were resolved when discovered. Although mass production increases efficiency by having unskilled workers perform unvarying tasks as extensions of machines, it can lead to repetitive motion injuries and create an unrewarding work environment.

Except when restricted by union contracts or labor law, mass manufacturers treat unskilled labor as a variable cost that is easily added, removed or substituted from the production process with minimal investments in training. Figure 1 illustrates the traditional mass production process.

Figure 1. The Traditional Mass Production Process

Due to post-WWII labor shortages, limited capital, and low domestic demand that made mass manufacturing uninviting, Toyota developed a new manufacturing method that was later labeled “lean.” Since Toyota employees had lifetime employment contracts, in part due to the post-WWII U.S. government support of Japanese unions, Toyota considered labor to be just as important as machinery for productivity. Toyota trained workers to perform a variety of tasks and delegated manufacturing decisions to allow greater self-management, continuous improvement, and quality control, partly because of the influence of W. Edward Deming.¹ The lean manufacturing process, elements of which were present at Toyota since 1938, meets current customer demand using Just-In-Time

¹ https://en.wikipedia.org/wiki/W._Edwards_Deming

(JIT) supplies of parts that eliminates the need for raw materials, work-in-process, and finished goods inventories, as well as the related working capital financing.² Figure 2 presents the lean manufacturing process.

Figure 2. Lean Manufacturing.

Because of Toyota's competitive success, other manufacturers adopted lean techniques in the 1980's and 1990's. For example, Dell produces computers on demand, maintaining no inventory. As lean processes spread, firms realized that the traditional centralized managerial accounting techniques failed to capture and clearly communicate

² Ford's earlier (and much smaller) Highland Park factory incorporated many features of lean manufacturing, including a crude "pull" system, limits to work-in-process, and a nearly continuous flow of process from raw materials to finished product, all possible since a single automobile style, the Model T, was manufactured on a small scale. The size, complexity and variety of automobiles produced at the Rouge plant required Ford to abandon these early "lean" processes (Womack et al., 2007, p. 288).

the operational efficiencies of lean processes. Lean production focuses on process flow and elimination of waste in labor time and materials. Waste is identified by mapping value-adding activities and isolating and eliminating production steps that do not add value. Value-added processes are those for which the customer is willing to pay. All else is waste.

Since inventories are not a factor in lean production, period production costs match revenues, greatly simplifying the accounting. Expenses necessary to the business but not of direct benefit to the customer (generally, R&D, marketing, sales and administrative expenses) are captured as business value-added expenses and recorded separately from customer-related or product expenses. There are no or minimal allocations other than fixed expenses that have a direct customer value-added benefit.

Figure 3 maps the value stream flow and related cost for each step of the software development process (requirements, design, develop, etc.). Each of these steps represents a direct customer value-added function. This mapping process allows non-value-added activities to be identified and eliminated.

Figure 3. Value Stream Flow and Costing

Figures 4 and 5 identify the productivity metrics used for this research: flow efficiency, cycle time, work-in-process, and throughput. Figure 4 shows the calculation of flow efficiency for the value stream with the amount of time (resources) assigned to each step identified. The productive time is the time used to contribute directly to customer value summed across all the steps. The full elapsed time needed to complete production is called the cycle time and includes the waiting time between steps. Flow efficiency is the ratio of productive time over cycle time. Figure 5, provided by Vacanti (2015), shows the relationship between cycle time, work-in-process, and throughput, known as Little's Law (Little, 1961). The figure illustrates a continuous flow diagram (CFD). If the arrival of work into a process exceeds the departure of work (throughput), then work-in-process increases and the total time needed to process work increases as well (cycle time). A traditional push approach typically forces work into the process without regard to the

departure rate, resulting in an increase in work-in-process and cycle time. A lean approach pulls work into the process only when the next production stage has capacity to process the work, resulting in consistent levels of work-in-process and throughput.

Figure 4. Flow Efficiency

At any given point between the lines of the CFDs in Figure 5, the horizontal distance between the lines indicates the cycle time and the vertical distance represents the queue or work-in-process. The steeper the arrival or departure line, the faster the items are entering or leaving the queue. If the top arrival line has a greater slope than the bottom departure line, the queue or work-in-process (the vertical distance) is increasing. This occurs when work is “pushed” into the next stage of production before that stage is ready to process the work. When the lines are parallel, the process is steady, predictable and in control. Work items are pulled into the next stage of the process only when that stage is ready for the work. This is considered to be the ideal state of flow. If the slope of the departure line is greater than that of the arrival line, it represents a batch process

where a queue or backlog of work is processed until completion when the next batch is introduced.

$$\text{Cycle Time} = \frac{\text{Work-in-Process}}{\text{Throughput}}$$

Figure 5. Cycle Time, Work-in-Process and Throughput

Emergence of Lean Production in Software Development

Lean practices in the software development industry surfaced in the early 2000s (Beck et al., 2001, Poppendieck & Poppendieck, 2003). Since then, software development has progressed from traditional techniques that involve large, inflexible, sequential phases of work that build upon each other (described below), to the nimbler and highly iterative *agile* approach. In the agile approach, software is developed in small, self-organizing scrum teams (six to eight members) that rely heavily on customer feedback and are highly responsive to change.

There are strong parallels between manufacturing and software development processes. Traditional software development, known as the “Waterfall” approach, is structured like mass manufacturing (Bell & Thayer, 1976). A massive scale of software development flows like a waterfall from one level of development to the next using specialized development processes that are rigidly controlled by a centralized management structure. Any changes made to the software mid-process require starting over from the beginning since each stage heavily depends on earlier stages of the process. Also similar to mass production in manufacturing, software work is “pushed” through the development process so that queues form between successive stages: this creates work-in-process and lengthens the development cycle. The U.S. Department of Defense software projects, which are required to meet the standards captured in DoD-STD-2167A (DoD, 1988), are examples of large waterfall projects for which the requirements are stable and known in advance. These standards require formal reviews and audits of each sequential stage of the development process, which means that each stage had to be completed and tested before work could begin on the next stage. Figure 6 depicts the waterfall development process.

Figure 6. Traditional Waterfall Software Development Process

In contrast, the lean software development process “pulls” the software through each stage so that there is no queue. Further, lean software developers maintain continuous contact with the customer so that the end product is always aligned with customer demands. Since the process is highly iterative, changes are incremental and are made quickly and easily with minimal disruption. Modular development and object-oriented programming facilitate such arrangements. Additionally, defects in the software are typically identified earlier and are remedied promptly. Figure 7 depicts the lean software development process. However, software developers found that the highly decentralized and self-controlled group processes were not well managed and backlogs with attendant delays were occurring. As a result, agile lean processes have more recently been introduced that focus on predictability of completion as the key metric for software

developers (Vacanti, 2015). Creating a highly predictable process means minimizing backlogs and work-in-process queues, and maintaining a continuous flow of production. This also requires that work is pulled into the next stage of the development process only when it is ready to be received and that the next stage has the capacity to complete the work.

Figure 7. Lean Software Development Process

Lean software development practices are often called agile software development. The two approaches share many attributes, are highly complementary, and the terms are often used interchangeably (Wang, Conboy & Cawley, 2012), differing chiefly in points of emphasis. For example, lean focuses on elimination of waste, queue time, and inventory, while agile refers to highly iterative and rapid customer-centric software

development using self-organizing teams. In this paper, I refer mainly to lean practices but will use attributes emphasized by agile practices as well.

Transition of Lean Accounting: Manufacturing to Software Development

Firms try to align their managerial reporting systems and related metrics to their operational practices. Firms in dynamic industries, such as software development, find that their management reporting systems and metrics often lag operational changes. As a result, both management and operational staff cannot use the reporting systems and related metrics to understand and improve operations. Lean and agile practices have significantly altered software development processes over the past fifteen years, and reporting systems are generally misaligned in this new environment. Today's development staff and their managers lack the data needed to improve operations and senior managers cannot see the connection between lean/agile processes, productivity, and profitability.

The software development industry is in the same position as U.S. car makers were more than a decade ago at the start of the Lean Accounting Summit for Manufacturing in 2005, as depicted in Figure 8. Lean practices were introduced in the software industry fifteen years ago and developers now demand more robust and appropriate measures of software development productivity. For comparison, lean manufacturers took thirty years to formalize lean management accounting practices. As noted earlier, these improved metrics are vital to the continued profitable growth and market responsiveness of the software industry.

80% of the issues related to the adoption of lean production in the software industry are due to inappropriate financial and operational measurement systems (Vacanti, 2015). I will explore how lean accounting affects software development productivity.

Figure 8. Timeline of Lean Accounting -- Manufacturing and Software Development

Research in lean manufacturing has established the value of lean accounting practices in improving operational efficiency and profitability. However, no equivalent research exists for the software development industry, despite the integral role of software as a part of virtually every other industry. My dissertation tries to fill this gap by examining the impact of lean accounting on the productivity of the software development industry, software writers' attitudes and their understanding of lean/agile processes. The

study employs an experimental approach and is based on the experiences of approximately 430 software developers in the U.S., India and Romania working at a major healthcare software firm.

CHAPTER 3

LITERATURE REVIEW

Lean Software Development

Although there has been much research on the link between lean manufacturing and productivity, there is little rigorous empirical research on lean software development productivity. Uniquely, Petersen and Wohlin (2010) connect lean manufacturing and lean software development, addressing the measurement techniques common to both, including measurement of process bottlenecks, continuous flow requirements, and graphically visual measurements.

A review of the agile software development literature through 2005 finds 1,996 articles of which 36 are empirical (Dyba & Dingsoyr, 2008). Of these 36, only Middleton (2001) examines lean processes. This article examines two three-person development teams (one highly experienced, the other not) and tracks their error rates in software development after being trained in lean techniques. Based on a time-series analysis, the study finds that the experienced team made more errors because they were not committed to the process, but the less experienced team regressed after a few days of improved quality (fewer errors). The study concludes that lean concepts need full organizational alignment if they are to succeed. Subsequently, Middleton, Flaxel, and Cookson (2005) examined the lean software development processes of Timberline Corporation and estimated percentage improvements in productivity by surveying Timberline's senior managers. No baseline productivity data were collected, so statistical comparisons were not conducted.

The most rigorous extant quantitative study is an analysis of the introduction of lean practices to the agile development processes at Siemens Health Services (Vacanti, 2015). This case study documents a 42% reduction in cycle time and approximately 50% reduction in defects by establishing limits to the amount of work-in-process. However, the study includes only the 85% of the work items with the lowest cycle time. The 15% of work items that are “outliers” were excluded. As such, a true mean of cycle times is not reported.

Poppendieck and Poppendieck (2003) emphasize the importance of using many short, iterative learning cycles that focus on refactoring, or improving the design as the system develops. This is more effective than long, infrequent “batch” approaches to learning that rely more on documentation than actual work production and extends the lean “just-in-time” approach to learning as well as production.

Training and Coaching

The literature on the effects of training on productivity is generally survey-based or uses panel data from industry databases. The earliest analyses rely on survey data to determine the relations between training and productivity. Barron, Black, and Lowenstein (1987) survey 28 firms that applied for U.S. Department of Labor Employment Opportunity Pilot Project funding of training programs. The study seeks to determine the relationship between firm size and training investments, hypothesizing that large firms have higher employee monitoring costs that can be controlled through increased investments in training. The results indicate that a 10% increase in firm size is associated with a 1.7% (1.6%) increase in formal (informal) training.

Holzer, Block, Cheatham, and Knott (1993) survey 498 firms that applied for Michigan Job Opportunity Bank grants to cover training costs for manufacturing firms with less than 500 employees. Of the 157 firms that responded, 66 had received grants and 91 had not. Firms that received grants had substantial increases in training for that year, reaching a level twice that of firms that did not receive a grant. Firms that received the grants and invested in training realized a reduction in scrap rates of 0.5% to 0.7% of sales, worth about \$30,000 to \$50,000 in savings per year. Grants used to fund the training averaged \$12,000 per firm in 1988 and \$20,000 in 1989. Training investments returned to the pre-grant levels following the conclusion of the state funding, which is surprising given the positive returns on the investment. Continued investments in training would have substantiated whether there is a ceiling effect to additional training or if the positive results continue.

Bishop's (1994) study uses telephone interviews conducted with 3,412 randomly selected firms, 88% with less than 200 employees, and finds that formal on-the-job training through a previous employer increases initial productivity by 9.5% and reduces training requirements by 17.3%. However, formal training received off-the-job increases productivity by 15.9%. Bishop concludes that there are significant externalities from both on-the-job and off-the-job training. Furthermore, formal training has a long-term impact, while informal training generates externalities only for the first year of employment.

Using Compustat data on 495 manufacturing business lines over a three-year period (1983-1986), Bartel (1994) finds that firms with below expected productivity levels improved to the level of peer firms as a result of increased investment in training,

representing an 18.9% increase in productivity. Productivity is measured as sales per employee less cost of purchased materials to determine value-added per employee.

Zwick (2006) notes that endogeneity adversely affects many analyses of training's impact on productivity. Firms do not decide randomly on when to invest in training or on which employees to train, so training is not strictly an exogenous event. Productivity is also impacted by numerous other internal firm-level factors, such as management quality and personnel policies, which may vary from period to period and firm to firm. External factors such as technological change and labor market conditions may affect productivity as well. Zwick provides an analysis of training's impact on productivity based on an annual survey of German firms by the Institute for Employment Research, covering a range of firms, from 8,917 in 1997 to 15,537 in 2001. The firms represent all industries and firm sizes. Zwick attempts to control for this endogeneity bias and indicates that a 1 percent increase in training in the first half of 1997 increases average productivity by 0.76 percent in the period 1998 through 2001.

Black and Lynch (1996) use 1994 survey data from the National Center on the Educational Quality of the Workforce, which is designed to determine the impact of human capital investments on business productivity. An analysis of telephone interviews conducted with 2,945 businesses (55% manufacturers, 45% services) indicate that a 10% increase in average education level results in an 8.5% and 12.7% increase in productivity for manufacturing and non-manufacturing, respectively. The evidence is less straightforward for training which, similar to other capital investments, has little impact in the current year, but has a lagged effect over 1 to 4 years. In manufacturing, off-the-job training has a strong impact, perhaps due to the use of non-work time for training.

Further, off-the-job training may be more intense and enable higher skill development. In services, the content of the training, specifically computer training, increases productivity. Most interesting is that training in Total Quality Management (TQM) and Benchmarking caused no immediate impact on productivity. It appears that these are cultural changes that require longer implementation and acceptance to affect productivity.

Dearden, Reed, and Van Reenen (2006) use panel data from a sample of 119 British manufacturing firms from 1983 through 1996 along with the UK Labor Force Survey and the Annual Census of Production. The study finds that a 1% increase in training results in a 0.6% increase in productivity and a 0.3% increase in wages. The authors indicate that this gap between productivity and wage impact is indicative of the spillover effects of training on industry in general. This effect, identified using a combination of industry-level and household-level data, represents a substantial improvement over prior studies that cite the impact of training on wage growth as an indicator of productivity.

Training has a positive impact on employee attitudes as well. In a multi-case analysis of the sources of motivation for software engineers at three lean agile software development firms, 11 motivating factors were found, including technically challenging work, teamwork, and employee participation, among others. However, the existence of training opportunities to widen skills as well as specialize was the only attribute identified across all three firms and is cited as the most important factor in employee motivation (Melo, Santana & Kon, 2012). Specifically, the recommended type of training is “constant learning and knowledge sharing,” suggesting that the most effective training

is not a formal, structured, and infrequent exercise but an ingrained part of the corporate culture. This echoes the learning theme in Poppendieck and Poppendieck (2003).

This form of continuous training is similar to the coaching techniques embedded in the “flow” culture of lean processes at Nordea Bank, a Scandinavian-based bank that is one of the 25 largest in the world. Starting in 2007, all Nordea managers had to complete at least a basic coaching course in order to incorporate lean coaching skills as part of their daily leadership practice. Coaching processes were designed to support the lean process through three phases: initial implementation, growth and sustainability. Coaching consisted of both structured formal sessions as well as spontaneous sessions. Following introduction of the coaching program, management indicated that coaching was key to the successful implementation of lean processes at the bank. In fact, 67% of surveyed employees indicated that the coaching increased employee engagement (Gørtz, 2012).

In a case study of Wipro, an Indian software services provider, it was found that implementation of lean processes improved project productivity compared to non-lean projects. The analysis included both qualitative and quantitative factors. Deviations from projected schedules, expected effort, and prior defect rate (quality) were used to measure performance. Lean projects performed better than non-lean projects on all three measures, and the results were statistically significant with the exception of defect rate. Both training and mentoring/coaching are integral to the success of the lean practices program at Wipro (Staats, Brunner & Upton, 2011).

Lean Accounting

The definitive book on lean practices in the software industry, *Lean Software Development: An Agile Toolkit* (Poppendieck & Poppendieck, 2003), only mentions the term “accounting” three times, but in each instance it invokes how accounting falls short when capturing the productivity of lean principles. To date, there is virtually no research that addresses this issue. However, several recently published articles examine the application of lean accounting in the manufacturing sector. In addition, researchers have studied the relationship between accounting systems and strategy as well as culture, which applies to this topic.

Rosemary Fullerton, Frances Kennedy, and Sally Widener jointly published four articles from 2008 to 2014 on lean accounting in manufacturing. Kennedy and Widener (2008) develop a theoretical framework based on a literature review regarding the implementation of lean manufacturing as a change in operational strategy. They find evidence of many reporting practices such as visualization of metrics, standard operating procedures, employee empowerment, training and peer pressure, all incorporated into a framework that has lean manufacturing as the core concept. They subsequently validate these concepts through an interview-based case study of a manufacturer that has converted from traditional to lean manufacturing processes.

Fullerton et al. (2013) examined 244 U.S. companies with an interest in lean manufacturing and found a direct correlation with four management accounting practices: simplified strategic reporting systems, value stream costing (capturing direct costs in a process rather than using cost allocations), visual performance measurement, and employee empowerment. These practices must be employed together as a package for

optimal results. Similarly, Shah and Ward (2007) identify ten lean operational factors that are associated with lean manufacturing and note that full implementation of all ten factors facilitates achieving multiple performance goals and generating sustained competitive advantage but that it is difficult to implement all ten factors.

Fullerton, Kennedy and Widener (2014) used the same 244 firms from their prior study to examine the relationship between specific lean accounting practices and financial performance. They find that three lean accounting practices associated with lean manufacturing (value stream costing, visual performance measures, simplified strategic management accounting) are correlated with better operational performance, which, in turn, is positively related to financial performance.

Minimization of inventory levels is a key characteristic of lean manufacturing, and inventory was found to be a mediating variable between lean production systems and financial performance based on 229 survey respondents (Hofer, Eroglu & Hofer, 2012). Furthermore, similar to the findings of Fullerton et al. (2013), this article notes that the concurrent implementation of a bundle of lean practices is more effective than selective implementation of such practices.

Fullerton and Wempe (2009) found that non-financial manufacturing performance (NFMP) measures mediate the relationship between lean manufacturing and financial performance based on a survey of 121 U.S. manufacturing executives. The researchers note that these NFMP measures may broadly be considered managerial accounting measures, as they track waste and other variables that influence financial results. However, the article does not provide specific examples of the NFMP measures. Similarly, Vacanti (2015) notes that the use of non-financial lean metrics used to improve

the predictability of software development cycle time at Siemens Healthcare IT could enable tracking and forecasting of costs for a software release.

Banker, Potter, and Schroeder (1993) demonstrate the strong relationship between distributing manufacturing shop performance measures to line personnel and successful implementation of just-in-time (JIT) practices and Total Quality Management (TQM), both predecessors to lean processes. A survey of 362 workers at 40 firms found positive results when employees are given feedback from the reporting systems. The reporting information helped employees learn the system and direct their efforts to improvements in areas requiring worker involvement. Worker morale was also found to be improved.

Uniquely, Petersen and Wohlin (2010) connect lean manufacturing and lean software development, addressing measurement techniques common to both, such as measurement of process bottlenecks, continuous flow requirements and visualization of measures. However, managerial accounting concepts are not addressed directly.

Although not specific to agile/lean methodologies, Banker and Kauffman (1991) propose automated means to evaluate software development productivity using function point analyses and code re-use.³ The measurement process itself is typically very manual, but has been automated to a degree by integrated computer-aided software engineering (CASE) which, in turn, has been facilitated by object-oriented programming and a

³ Function point analysis was developed by Allan Albrecht of IBM (1979) and is computed by measuring the functionality delivered by the software to the user. User functionality is measured based on the number of inputs, outputs, internal files, external queries and interfaces, further adjusted for complexity.

centralized data repository.⁴ These methodologies can certainly be useful as non-financial/accounting performance metrics within the agile/lean approach to development.

On a broader note, Busco, Riccaboni, and Scapens (2000) look at the required congruence between organizational culture and managerial accounting systems. Their case study examines the rapid change of a 174-year-old Italian industrial products manufacturer that has no “culture of measurement” but has been acquired by GE (called ALPA in the case) and is transformed into a leading Six Sigma firm in a few years. The case study describes how the change in accountability systems helped to bring about the transformation of the organizational culture. This case clearly illustrates the need for alignment between lean accounting and lean software development. Birnberg (2011) and Chenhall and Langfield-Smith (1998) observe that the literature makes a strong and consistent case that organizational strategy and managerial accounting practices should be aligned to ensure operational effectiveness of the organization.

⁴ Similarly, code re-use is facilitated by CASE in conjunction with object-oriented development methods that allow objects to be maintained and retrieved for re-use in future projects.

CHAPTER 4

CONCEPTUAL MODELS AND HYPOTHESES

Retraining and Coaching

The preceding observations from the literature review inform the theoretical framework for this dissertation:

Hypothesis 1a: *Retraining of product development teams on lean production processes improves lean software development productivity as measured by reduced cycle time.*

Hypothesis 1b: *Retraining of product development teams on lean production processes improves lean software development productivity as measured by increased flow efficiency.*

Hypothesis 1c: *Retraining of product development teams on lean production processes improves lean software development productivity as measured by reduced daily work-in-process.*

Hypothesis 1d: *Retraining of product development teams on lean production processes improves lean software development productivity as measured by increased daily throughput.*

Hypothesis 2a: *Coaching of product development teams on lean production processes improves lean software development productivity as measured by reduced cycle time.*

Hypothesis 2b: *Coaching of product development teams on lean production processes improves lean software development productivity as measured by increased flow efficiency.*

Hypothesis 2c: *Coaching of product development teams on lean production processes improves lean software development productivity as measured by reduced daily work-in-process.*

Hypothesis 2d: *Coaching of product development teams on lean production processes improves lean software development productivity as measured by increased daily throughput.*

My analysis of lean retraining and coaching builds upon research on implementation of lean practices in a software development firm (Vacanti, 2015). As previously noted, this article is the only rigorous quantitative analysis of the relationship between lean software development and productivity of which I am aware.

Tables 1 and 2 provide detail on the construct definitions and measurement.

Table 1 <i>Construct Definitions: Retraining, Coaching Productivity</i>		
Construct	Definition	Relevant Literature
Lean Software Development	Software development processes that follow lean principles: - Eliminate waste - Amplify learning - Decide as late as possible - Deliver as fast as possible - Empower the team - Build integrity in - See the whole	Poppendieck and Poppendieck, 2003
Productivity	The rate of output per unit of input	Case, Fair and Oster, 2017 Mateer and Coppack, 2014
Lean Retraining	Multi-day training at work site focused on retraining teams on lean software development processes	Poppendieck and Poppendieck, 2003 Kennedy and Widener, 2008
Lean Coaching	Monthly coaching sessions conducted over three months	Gørtz, 2008

Table 2 <i>Construct Measurement: Retraining, Coaching Productivity</i>			
Measure	Instrument	Method used to Measure	Relevant Literature
Work-in-Process	ActionableAgile tool	All discrete units of customer value that have entered a given production process but have not exited	Vacanti, 2015 Reinertsen, 1997, 2009
Cycle time	ActionableAgile tool	Number of workdays it takes for a work item to complete the production process	Vacanti, 2015 Reinertsen, 1997, 2009 Little, 1961
Throughput	ActionableAgile tool	Number of work items completed per time period	Vacanti, 2015 Little, 1961
Flow efficiency	ActionableAgile tool	Ratio of the total lapsed time a work item was actively worked ÷ the total lapsed time that the work item took to complete (cycle time).	Vacanti, 2015 Reinertsen, 1997, 2009 Little, 1961

Lean Accounting

The preceding observations from the literature review inform the theoretical framework for this research proposal:

Hypothesis 3a: *The degree of lean accounting implementation impacts lean productivity in software development as measured by reduced cycle time.*

Hypothesis 3b: *The degree of lean accounting implementation impacts lean productivity in software development as measured by increased flow efficiency.*

Hypothesis 3c: *The degree of lean accounting implementation impacts lean productivity in software development as measured by decreased daily work-in-process.*

Hypothesis 3d: *The degree of lean accounting implementation impacts lean productivity in software development as measured by increased daily throughput.*

Hypothesis 4: *Lean management accounting practices help team employees understand lean operational metrics, resulting in improved lean practices and greater employee work satisfaction as measured by employee survey results.*

I argue that the impact of lean software development processes on productivity is favorably modified by lean management accounting practices. Lean practices must permeate the entire organization; they represent a philosophy of how to do business. Managerial accounting systems bind together the disparate parts of the organization (Jensen, 1983) and any misalignment on measurements, vocabulary, standards, and access will likely result in sub-optimal performance. The more comprehensively lean accounting is implemented, the more effective lean practices will be.

These lean management accounting practices also reinforce and enhance employee acceptance of lean processes. As employees more fully understand their

operational impact, they can initiate actions that strengthen lean processes and improve attitudes toward their work. Finally, the positive impact of lean accounting practices on employee attitudes and understanding of lean practices further improves productivity, although I will not be testing this argument here.

Figure 9 outlines the conceptual model and its measurement, illustrating that lean software development retraining (H1) and coaching (H2) will improve production through reduced cycle time and work-in-process while increasing flow efficiency and throughput. The conceptual model also represents that use of lean accounting practices (H3) will improve software productivity by decreasing production cycle time and work-in-process and increasing flow efficiency and throughput. Also, frequent exposure of employees to visible and accessible accounting metrics (H4) will further enhance productivity and employee use of lean practices. It may be argued that any improvement in productivity is achieved through the employee's realization of lean practices. However, the design of this experiment does not allow for the direct measurement of the employee's productivity.

Figure 9. Conceptual Model with Measurements

Table 3 provides the definitions of the conceptual model constructs, and Tables 4 through 6 identify how the constructs are measured. Although lean software development is not a construct in the conceptual model, it is a core concept in the analysis and is also defined.

Table 3 <i>Construct Definitions: Lean Accounting Productivity</i>		
Construct	Definition	Relevant Literature
Lean Software Development	Software development processes that follow lean principles: - Eliminate waste - Amplify learning - Decide as late as possible - Deliver as fast as possible - Empower the team - Build integrity in - See the whole	Poppendieck and Poppendieck, 2003
Productivity	The rate of output per unit of input	Case, Fair and Oster, 2017 Mateer and Coppack, 2014
Lean Accounting Management Practices	Metrics with these qualities: Visual, simplified, decentralized, strategically aligned, operationally-oriented, identifies value stream costs and resources, and empowers employees through actionable information	Fullerton, Kennedy and Widener, 2013, 2014 Kennedy and Widener, 2008
Employee realization of lean practices	Employee understanding and use of lean software development processes	Poppendieck and Poppendieck, 2003

Table 4 <i>Construct Measurement: Lean Accounting Productivity</i>			
Measure	Instrument	Method used to Measure	Relevant Literature
Work-in-Process	ActionableAgile tool	All discrete units of customer value that have entered a given production process but have not exited	Vacanti, 2015 Reinertsen, 1997, 2009
Cycle time	ActionableAgile tool	Number of workdays it takes for a work item to complete the production process	Vacanti, 2015 Reinertsen, 1997, 2009 Little, 1961
Throughput	ActionableAgile tool	Number of work items completed per period of time	Vacanti, 2015 Little, 1961
Flow efficiency	ActionableAgile tool	Ratio of the total lapsed time a work item was actively worked / the total lapsed time that the work item took to complete (cycle time).	Vacanti, 2015 Reinertsen, 2009, 1997 Little, 1961

Table 5 <i>Construct Measurement: Lean Accounting Practices</i>			
Measure	Instrument	Method used to Measure	Relevant Literature
Frequent, Visual, Actionable	Graphic, one page report reports on work-in-process, flow efficiency, throughput, cycle time, work-in-process and predictability	Daily review of and adjustments to WIP; Bi-weekly distribution and of other reports review in team meetings	Fullerton, Kennedy and Widener, 2013, 2014 Kennedy and Widener, 2008

Table 6 <i>Construct Measurement: Employee Realization of Lean Practices</i>			
Measure	Instrument	Method used to Measure	Relevant Literature
Team adoption of lean / agile practices	Survey	6 questions related to adoption of lean/agile practices using a 5 point Likert scale: Strongly agree (1) to Strongly disagree (5)	Poppendieck and Poppendieck, 2003 Fullerton, Kennedy and Widener, 2013, 2014 Kennedy and Widener, 2008
Satisfaction with work at the personal and team level	Survey	4 questions related to work satisfaction using a 5 point Likert scale: Strongly agree (1) to Strongly disagree (5)	Poppendieck and Poppendieck, 2003 Fullerton, Kennedy and Widener, 2013, 2014 and Kennedy and Widener, 2008
Clarity, quality and usefulness of operational reporting	Survey	4 questions related to the value of operational reporting using a 5 point Likert scale: Strongly agree (1) to Strongly disagree (5)	Poppendieck and Poppendieck, 2003 Fullerton, Kennedy and Widener, 2013, 2014 Kennedy and Widener, 2008

Survey questions are provided in the Appendices

CHAPTER 5

RETRAINING AND COACHING EXPERIMENTS

Description of Necessary Data

I test the retraining and coaching hypotheses using two experiments. Figure 10 describes the research protocol for both experiments.

Figure 10. Research Protocol: Retraining/Coaching

In experiment one, productivity was measured for both the treatment group (MedRec development team) and the potential control group (32 clinical software development teams) from August 25, 2015 through December 28, 2016. The treatment team developed 227 stories (the smallest unit of software production) during this time. The control group developed 1,337 stories during the same timeframe. However, 18 of the 32 control group teams did not produce stories after the retraining phase, and one team (EU Ext) was excluded due to data anomalies, so the analysis for the control groups is limited to the 13 teams that produced stories both before and after the treatment. Table 9 summarizes the productivity measures for the control teams included and excluded

from the analysis. The treatment group undertook a multi-day lean retraining program in January 2017. Productivity was measured again from February 9, 2017 through March 22, 2017 for the treatment group and from January 1, 2017 through April 30, 2017 for the control group. The timing differences are due to different development cycle start and end times for the development teams. During this post-training period, the treatment team developed 51 stories. The 13 control group teams produced 202 stories.

In experiment two, the treatment group consisting of three teams received three monthly coaching sessions beginning in late May 2017 and ending in July 2017. The control group was the remaining 16 teams in the clinical development organization; however, four teams were dropped because they did not develop stories either before or after the coaching phase. Productivity is measured for 6 months before and 5 months after the coaching sessions for both the treatment and control groups. The treatment (control) group developed 112 (255) stories before and 85 (222) stories after initiating the coaching session treatment.

Stories in the pre-treatment phase comprise those completed up to the start of treatment regardless of when the story was initiated. Stories in the post-treatment phase comprise those started and completed after the initiation of treatment. Stories started before/during the treatment but completed during/after the treatment are not included in this study. This approach is used for all experiments in this study.

The site for the study is a publicly-traded healthcare IT company (\$5.1 billion in sales, FY17). The firm uses lean software development but has not consistently implemented lean processes throughout the organization. Non-Disclosure Agreements (NDA) with the firm allow access to the firm's internal data.

Methodology for Data Collection and Analysis

I conducted informal and semi-structured interviews of managers in development, operations, and finance to determine the most effective metrics and to identify the development teams to include in the study. In experiment one, a development team, MedRec, had recently concluded a lean retraining exercise, so I selected it as the treatment group. The balance of the clinical application development organization served as the control group, excluding those control teams that did not produce stories after retraining. Development managers provided story production data for all teams over the evaluation period (August 25, 2015 through April 30, 2017).

In experiment two, three monthly lean-coaching sessions were conducted with three teams, Orders Blue, Orders Green and Orders White. The remaining clinical application software teams served as the control group. The evaluation covered January 1, 2017 through November 13, 2017.

Managing work-in-process within accepted limits is the key to a successful lean development process. While the concept and financial calculation of work-in-process is well-established in manufacturing, it is not yet widely recognized in software development. Perhaps this is due to the visibility of work-in-process on the shop floor in manufacturing. However, some lean/agile software development organizations are starting to measure work-in-process in terms of days in production (Vacanti, 2015). Although software work-in-process may not be visible, its costs are as important as in manufacturing.

Discussion of Results

Retraining Experiment Results

The analysis is represented in two formats. First, I visually analyze cycle time improvements for the MedRec treatment group relative to the control group and how they were obtained by evaluating scatterplots and cumulative flow diagrams. Second, I test for the statistical significance of the improvements using a difference-in-difference analysis. The reduction of cycle times around lean software retraining is evidence that retraining improves software development productivity.

The results of the lean retraining are evident in the Figure 11 scatterplot which depicts the cycle times for each story, represented by a blue dot, over the study period. The Y-axis represents days to complete a story and the X-axis represents the date on which the story was completed. Before retraining, the treatment group (MedRec team) had produced 227 stories during two development cycles that ran between August 25, 2015 and December 28, 2016.

The retraining in January 2017 dramatically changed the cycle time of the development process. Before retraining, the MedRec group varied a lot in how long it took to deliver stories. Several work items took significantly longer than 100 days to complete. Figure 11 shows that there were two distinct development projects before retraining. The second development project has a wider dispersion of cycle times towards the end, indicating that the development process was becoming less predictable and lengthening. After retraining, the treatment group eliminated all long-cycle time stories, increased predictability in story completion and reduced cycle time drastically.

Figure 11. Experiment One: Treatment Group Scatterplot
Out-of-control process for developing stories now in control following retraining

Figure 12 depicts the cycle times for the control group, constituting the balance of the clinical software development organization, evaluated over the same time period. Dispersion of story production is initially reduced after treatment, since this includes only stories started after treatment. However, cycle time quickly ramps up, approaching 100 days, while the treatment group remains relatively stable and below 30 days.

Table 7 summarizes the cycle time statistics. For the treatment group, average cycle time to complete these stories before retraining was 38.3 days. After retraining in January 2017, this treatment group completed 51 stories with an average cycle time of 7.1 days, representing an 81.3% improvement. With a t -statistic of 11.60, the difference in average cycle times is statistically significant at $p < 0.001$.

Figure 12. Experiment One: Control Group Scatterplot
Increasing cycle time trend for stories following retraining

The treatment team was re-trained to improve its predictability and reduce its variability. The team completed all stories within 30 days, eliminating the “long tail” of stories that before retraining took an inordinately long time, as seen in the Figure 11 scatterplot. This process improvement sharply reduced standard deviation, from 37.8 before retraining to 6.8 after retraining, an 82% drop, which is statistically significant at $p<0.001$. Similarly, skewness in cycle times fell 34.7% from 1.61 to 1.05, and this change is statistically significant at $p<0.001$.

During 2015-2016, the control group completed 1,337 stories with an average cycle time of 36.7 days. During a comparable 2017 period, the control group completed 202 stories with an average cycle time of 28.4 days, representing a 22.7% improvement. With a t -statistic of 4.19, the difference in cycle times between the pre and post training control group results rejects the null hypothesis that the results are due to chance. These results suggest that there were some company or other environmental forces that led to reduced cycle times throughout the firm. Skewness also decreased from 2.82 to 0.84, a

decline of 70.2%. This suggests that the proportion of long cycle-time stories decreased instead of increasing over time, and the change is statistically significant.

Table 7

Experiment One: Summary of Results - Cycle Time

Treatment Group - Retrained					Control Group - No Retraining				
	Work Items	Mean Cycle Time (a)	Cycle Time Std Dev (b)	Cycle Time Skewness (b)		Work Items	Mean Cycle Time (a)	Cycle Time Std Dev (b)	Cycle Time Skewness (b)
Pre-Retraining					No Retraining				
8/25/15 - 12/28/16	227	38.27	37.80	1.61	8/25/15 - 12/28/16	1,337	36.74	46.45	2.82
Post-Retraining					No Retraining				
2/9/17 - 3/22/17	51	7.14	6.79	1.05	1/1/17 - 4/30/17	202	28.41	21.75	0.84
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		-31.13	-31.01	-0.56	Differences		-8.33	-24.7	-1.98
% Change		-81.3%	-82.0%	-34.7%	% Change		-22.7%	-53.2%	-70.2%
Significance tests:					Significance tests:				
Statistic		11.604	31.012	10.424	Statistic		4.188	4.557	74.509
P-Value		0.000	0.000	0.000	P-Value		0.000	0.000	0.000
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		-22.80	-6.31	1.42					
% Point Change		-58.7%	-28.9%	35.5%					
Significance tests:									
Statistic		-7.348	NA	NA					
P-Value		0.000	NA	NA					

Significance Tests: a) T-test; b) F-test

The last panel in Table 7 compares cycle time changes for the treatment and control groups. The results show that the retraining on lean software development practices has improved software productivity around retraining compared to simultaneous changes in the control group. The 31-day reduction in the treatment-group cycle time is more than three times the 8.3 day improvement of the control-group cycle time and is statistically significant at $p < 0.001$. The standard deviation fell 28.9% points more for the treatment group compared to the control group. Skewness fell for both groups, although it fell more for the control group. The treatment group completed about a quarter of the number of stories that the control group finished after the retraining period, but was able

to complete all stories within 30 days. These results are consistent with lean retraining reducing cycle times and their variability by an economically significant magnitude.

The difference-in-difference analysis shows a 22.8-day difference in cycle time differences between the treatment group and the control group, statistically significant at $p < 0.005$. To the extent that company and external events affected both treatment and control teams similarly, and that both groups were subject to the limited 4-month post-retraining evaluation period, the significant difference-in-difference of the means is more confidently attributable to retraining rather than other events.

A cumulative flow diagram (CFD) helps in reporting and explaining lean practices. The CFD is the most useful tool to visualize the flow of work through a process (Reinertsen, 2009). With the Y-axis representing quantity of items produced and the X-axis representing time, the diagram plots cumulative arrivals as an arrival line (top line) and cumulative departures as a departure line (bottom line).⁵

The development period before retraining (7/22/16 to 12/28/16) for the treatment group is depicted in the cumulative flow diagram (CFD) in Figure 13. In this graphic, work items are “pushed” to the next development stage beyond current capacity. The various bands represent different stages of development process: “specifying” requirements, “developing” stories and “acceptance testing” of the story. Each stage is further divided into an “active” working stage and a “done” stage where completed work

⁵ The CFD visually conveys Little’s (1961) Law which states that the *Average Number of Items in a system = Average Arrival Rate * Average Waiting Time* (Little, 2011). Little’s Law has been adopted by many industries after customization. In the software industry, it takes the form *Cycle Time = Work-in-Progress / Throughput*, with throughput being the average departure rate of the system.

sits for the next stage. Stories are entering the queue more quickly than they are exiting, resulting in a build-up of WIP and lengthening of cycle time.

Figure 13. Experiment One: Treatment Group CFD - Pre-Retraining

7/16 through 12/16 - Increasing cycle time and work-in-process

As illustrated in Figure 13, when throughput declines and/or WIP increases, cycle time increases and the demands on staff to work overtime increases as well. This percentage increase in staff utilization can strongly affect the percentage increase in time needed to complete the project (Thomke & Reinertsen, 2012). More stories take longer to complete, as seen in Figure 11. The process is more variable and less predictable.

After retraining in January 2017, the arrival and departure rates are parallel as shown in Figure 14. This is typically achieved by WIP limits that “pull” work through the system only when the next development stage is ready to accept the work. Development work becomes much more predictable and cycle times and WIP are in control. Work “flows” with minimal interruption. Employees need not work overtime or weekends.

Figure 14. Experiment One: Treatment Group CFD - Post-Retraining
2/17 through 3/17 - Stable and predictable work-in-process and cycle time

CFDs for the control group are provided in Figures 15 and 16. As already seen in the scatterplot in Figure 12 and statistics in Table 7, the average cycle time and its variability improved for the control group. The CFD in Figure 16 shows that a relatively stable flow process exists in the control group before the retraining period. Although there are fluctuations in the amount of work introduced to the process, cycle time and work-in-process are generally consistent. This pattern remains unchanged after the retraining period for the control group shown in Figure 16. Differences in the two graphs are primarily due to scale.

Figure 15. Experiment One: Control Group CFD - Pre-Retraining

3/16 through 12/16 - Fluctuating levels of work but stable cycle time and work-in-process

Figure 16. Experiment One: Control Group CFD - Post-Retraining

1/17 through 4/17 - Stable cycle time and work-in-process

In addition to cycle time, I analyze other measures to further substantiate the productivity improvements for the treatment group. Daily throughput is the average number of stories produced in a day. Flow efficiency is the amount of productive time over the total cycle time (wait time plus productive time) as described in Figure 4. Higher flow efficiency means less wait time and more productive time. Work-in-process represents the average number of stories that are being processed in the system over a period of time. Fewer work-in-process stories indicate greater efficiency as stories are moving faster through the system. Results of tests using these additional productivity measures are reported in Table 8.

Table 8

Experiment One: Summary of Results - Flow Efficiency, Throughput, WIP

Treatment Group - Retrained					Control Group - No Retraining				
	Work Items	Flow Efficiency	Daily Through-put	Daily WIP		Work Items	Flow Efficiency	Daily Through-put	Daily WIP
Pre-Retraining					No Retraining				
8/25/15 - 12/28/16	227	70.9%	0.46	16.53	8/25/15 - 12/28/16	1,337	52.8%	2.71	88.73
Post-Retraining					No Retraining				
2/9/17 - 3/22/17	51	76.5%	1.21	7.45	1/1/17 - 4/30/17	202	46.6%	1.74	47.73
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		5.6%	0.75	-9.08	Differences		-6.2%	-0.97	-41.0
% Change			163.0%	-54.9%	% Change			-35.8%	-46.2%
Significance tests:					Significance tests:				
T-test			-1.647	9.650	T-test			3.403	15.501
P-Value			0.107	0.000	P-Value			0.001	0.000
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		11.8%	1.7	31.9					
% Point Change		NA	198.8%	-8.7%					
Significance tests:									
Statistic		NA	NA	NA					
P-Value		NA	NA	NA					

Note that, to a large degree, the absolute number of stories reflected in throughput and work-in-process is a function of the number of developers on a team and the length of time in the evaluation period. More developers can handle more stories, leading to

higher throughput and work-in-process. A longer time frame also allows for more story production. However, this study measures daily work-in-process and throughput, so the number of days available to process work should not matter. Similarly, the number of workers per team/group varies between teams/groups, but not significantly for a given team/group before and after retraining. Therefore, for throughput and work-in-process, the *percentage change* is the key to understanding and comparing productivity either between periods or among teams. Comparing the absolute difference in daily work-in-process and throughput between teams and groups is not valid. Management notes that each team is relatively unique. However, cycle time and flow efficiency are comparable in terms of absolute numbers and percentage change.

Average daily work-in-process, declined 46.2% and 54.9% for control and treatment teams, respectively. Changes for both groups are statistically significant, but this may be attributable to the difference between the pre and post-retraining periods, with the latter having fewer work days and therefore less likely to have stories with extended period of work. The difference-in-difference analysis indicates an 8.7% reduction in daily work-in-process for the treatment group.

Daily throughput of the treatment group improved from 0.46 to 1.21 stories per day, a 163% improvement. The control group declined from 2.71 to 1.74 stories per day, a 35.8% reduction, and the result is statistically significant. However, due to the small sample after retraining in the treatment group, the improvement is not statistically significant, with a *p*-value of 0.107. Therefore, for the treatment group, the null hypothesis that the difference in average daily throughput is attributable to chance cannot be rejected. However, the control group with a larger sample both before and after the

retraining period has a result that is statistically significant, with a t -statistic of -3.40 and p -value of 0.001. The difference-in-difference analysis shows that the treatment group has a much larger and more favorable percentage point change of 198.8 % points in daily throughput when compared to the control group.

Treatment group flow efficiency improved slightly from 70.9% to 76.5%. For the control group, flow efficiency dropped from 52.8% to 46.6%. The treatment group already had a strong flow efficiency, so improvement was harder for this group to achieve. Since these are single flow measures over time rather than specific data points in time, I could not statistically test for differences.

Daily work-in-process improved from 16.53 to 7.45 stories per day for the treatment group, a 54.9% improvement, while the control group had a similar improvement of 46.2%, dropping daily WIP from 88.73 to 47.73 stories. Discussions with management indicate that there was a system-wide emphasis on improving flow efficiency and reducing work-in-process during late 2016, and the results seem to bear this out. Both the treatment and control groups have statistically significant reductions of work-in-process, with t -statistics of 9.65 and 15.50 respectively.

As shown in Table 9, MedRec's post-retraining cycle time of 7.14 days was lower than all but 2 of the 13 teams in the control group after retraining and all of the control teams before retraining. It was also lower than the cycle time of teams that did not produce stories after the retraining period (non-common teams).

Table 9

Experiment One: Results by Team

Common Teams	Pre-Retraining (8/25/15-12/28/16)					Post-Retraining (1/1/17-4/30/17)					% Change			
	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency
ADT	0.29	14.93	52.89	23.1	144	0.31	11.53	35.72	47.9	36	6.9%	-22.8%	-32.5%	107.4%
ASCH	0.35	10.95	40.43	66.5	141	0.29	6.99	18.68	90.2	22	-17.1%	-36.2%	-53.8%	35.6%
Content	0.79	12.23	16.55	73.8	261	0.10	0.49	6.75	100.0	5	-87.3%	-96.0%	-59.2%	35.5%
Critical Care	0.20	5.12	33.59	78.6	46	0.50	2.50	36.00	54.6	3	150.0%	-51.2%	7.2%	-30.5%
DA	0.73	13.14	21.93	77.2	352	0.21	2.97	15.33	81.5	6	-71.2%	-77.4%	-30.1%	5.6%
ED	0.04	3.50	100.80	93.1	15	0.44	10.15	16.41	71.6	44	1000.0%	190.0%	-83.7%	-23.1%
Implement	0.08	2.00	26.53	71.1	38	0.07	1.41	20.00	87.5	2	-12.5%	-29.5%	-24.6%	23.1%
Orders Blue	0.25	10.94	44.36	43.6	53	0.21	7.74	36.50	62.9	24	-16.0%	-29.3%	-17.7%	44.3%
Order Green	0.44	24.57	56.84	41.1	176	0.20	5.06	25.36	62.7	22	-54.5%	-79.4%	-55.4%	52.6%
Orders White	0.21	11.53	56.33	61.2	30	0.17	10.26	38.45	65.9	20	-19.0%	-11.0%	-31.7%	7.7%
Prt & Rpt	0.06	5.77	120.00	27.7	31	0.02	1.65	87.00	55.7	2	-66.7%	-71.4%	-27.5%	101.1%
RE	0.11	3.19	31.29	39.8	47	0.04	0.94	26.00	15.4	2	-63.6%	-70.5%	-16.9%	-61.3%
Sensai	0.14	2.86	146.67	23.0	3	0.16	5.94	4.50	80.4	14	14.3%	107.7%	-96.9%	249.6%
Total Control	2.71	88.73	36.74	52.8	1337	1.74	47.73	28.41	46.6	202	-35.8%	-46.2%	-22.7%	-11.7%
MedRec	0.46	16.53	38.27	70.9	227	1.21	7.45	7.14	76.5	51	163.0%	-54.9%	-81.3%	7.9%
Total Treatment	0.46	16.53	38.27	70.9	227	1.2	7.5	7.1	76.5	51	163.0%	-54.9%	-81.3%	7.9%

Non Common Teams	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency
AIS	0.63	40.03	65.29	34.7	286	NA	NA	NA	NA	0	NA	NA	NA	NA
CCDA API	0.20	1.92	61.04	80.8	55	NA	NA	NA	NA	0	NA	NA	NA	NA
CCDA CCDS	0.49	17.46	39.56	49.9	173	NA	NA	NA	NA	0	NA	NA	NA	NA
Clin Doc	0.19	5.75	35.34	43.3	89	NA	NA	NA	NA	0	NA	NA	NA	NA
DPC-QSmed	0.22	3.27	25.56	57.4	98	NA	NA	NA	NA	0	NA	NA	NA	NA
FHV1-Enh	0.66	23.19	141.58	47.1	113	NA	NA	NA	NA	0	NA	NA	NA	NA
Immun	0.30	9.09	81.39	80.5	75	NA	NA	NA	NA	0	NA	NA	NA	NA
KDI-CMW	0.13	3.72	29.00	74.6	29	NA	NA	NA	NA	0	NA	NA	NA	NA
KDI-Found	0.32	7.50	52.80	42.2	46	NA	NA	NA	NA	0	NA	NA	NA	NA
Mobility	0.38	7.80	29.50	30.5	33	NA	NA	NA	NA	0	NA	NA	NA	NA
Orders 2015	0.34	11.63	38.15	75.3	89	NA	NA	NA	NA	0	NA	NA	NA	NA
Orders Red	0.04	1.17	47.19	99.2	16	NA	NA	NA	NA	0	NA	NA	NA	NA
P Doc Infra	0.20	8.91	47.30	74.7	81	NA	NA	NA	NA	0	NA	NA	NA	NA
Performance	0.12	6.17	54.42	70.0	57	NA	NA	NA	NA	0	NA	NA	NA	NA
Pharmacy	NA	NA	NA	NA	0	0.40	9.7	25.3	66.9	16	NA	NA	NA	NA
SEDA	0.07	5.38	79.48	58.7	23	NA	NA	NA	NA	0	NA	NA	NA	NA
SPW	NA	NA	NA	NA	0	0.15	6.8	50.7	100.0	11	NA	NA	NA	NA
VisitEnd	0.47	15.37	36.12	61.1	227	NA	NA	NA	NA	0	NA	NA	NA	NA
Total Non Common	2.26	120.99	54.62	53.6	1490	0.25	8.3	33.8	81.6	27	NA	NA	NA	NA

EU Ext	0.81	2.11	3.61	100.0	46	0.05	3.3	34.0	100.0	7	NA	NA	NA	NA
--------	------	------	------	-------	----	------	-----	------	-------	---	----	----	----	----

Management discussion about the dramatic reduction in cycle time for MedRec revealed some possible causes other than retraining. The pre-retraining period had some novel work requirements that required longer cycle time, possibly causing relatively lower cycle time after retraining as the nature of the work normalized. However, before retraining MedRec still had a cycle time lower than 20 of the 31 other teams producing stories, so the tasks were not atypical of software development during that period.⁶

The training exercise examined story size since large stories increase process complexity. All else equal, a smaller story requires less work resulting in shorter cycle time and less time in the system. For the same reason, throughput will be higher with smaller stories. Management could not determine how much story size changes affected productivity improvements compared to retraining but it is clear to management that the retraining had significant impact.

The hypotheses related to retraining are generally consistent with the results of the analysis:

The substantial reduction in treatment team cycle time following retraining compared to the modest control team improvement is consistent with the retraining hypothesis 1a.

Modest improvements in treatment team flow efficiency after retraining compared to the decline for the control group improvement are consistent with the retraining hypothesis 1b.

⁶ Team data that is categorized as NA (Not Applicable) after retraining did not have any material story production during that time period. Details for control teams with minimal story production before retraining are also not displayed since the percentage changes would be exaggerated. EU Ext team is also excluded due to the non-comparability of results between the pre- and post-retraining periods and the unrealistic 100% flow efficiency in both periods.

Improvements in daily work-in-process are comparable to those of the control group. This result is not consistent with the retraining hypothesis 1c.

The substantial increase in daily throughput for the treatment group and the decline for control group is consistent with the retraining hypothesis 1d.

Coaching Experiment Results

Similar to Experiment One, the impact of lean coaching on software productivity is evident in the scatterplots of the treatment (Figure 17) and the control teams (Figure 18), although both appear to improve since only those stories started after treatment period are included for the later period.

The treatment teams received three monthly coaching sessions from late May through July 2017. During the five months after the initiation of coaching, story production exhibited greater control than in the six months before the coaching. Before coaching, several stories took significantly longer than 100 days to complete; after coaching, virtually all stories took 72 days or less to complete. Furthermore, the improvements were most evident in the last two months of the evaluation period, from mid-September through mid-November, reflecting the cumulative impact of the coaching sessions. The control teams also improved but continued to have a large number of stories above the 70 day range. Additionally, improvement in the last two months is far less evident than in the treatment group.

Analysis of the statistical data supports this visual evidence. As shown in Table 10, the coached treatment group reduced cycle time from 53.5 days to 25.8 days, a 51.8% decrease. In contrast, the control group had a much smaller decrease in cycle time from

37.0 days to 31.5 days, a 14.7% decrease. The number of stories before and after coaching is similar for both the treatment and control teams and the timeframes are similar, with 144 days before coaching and 173 days after coaching. However, there are fewer stories completed after coaching for both the treatment and control groups since I only include stories started and completed after coaching. The pre-coaching period includes stories starting at any time but completed before any coaching. The p -values for changes in cycle time for the treatment group is statistically significant at 0.000, however, this is not the case for the control group with a p -value of 0.077. For the treatment group, the null hypothesis that the difference in means for each group is due to chance is rejected. It is not rejected for the control group.

Figure 17. Experiment Two: Treatment Group Scatterplot

Large reduction in variability in post-coaching phase, especially in last two months

Figure 18. Experiment Two: Control group Scatterplot
Initial reduction in variability, but consistent ramp-up in cycle time

Table 10

Experiment Two: Summary of Results - Cycle Time

Treatment Group - Coached					Control Group - Not Coached				
	Work Items	Mean Cycle Time (a)	Cycle Time Std Dev (b)	Cycle Time Skewness (b)		Work Items	Mean Cycle Time (a)	Cycle Time Std Dev (b)	Cycle Time Skewness (b)
Pre-Coaching					No Coaching				
1/1/17 - 5/24/17	112	53.54	36.98	0.97	1/1/17 - 5/24/17	255	36.96	40.20	2.65
Post-Coaching					No Coaching				
5/25/17 - 11/13/17	85	25.79	21.85	0.94	5/25/17 - 11/13/17	222	31.52	26.22	1.21
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		-27.75	-15.1	-0.02	Differences		-5.44	-14.0	-1.44
% Change		-51.8%	-40.9%	-2.2%	% Change		-14.7%	-34.8%	-54.4%
Significance tests:					Significance tests:				
Statistic		6.574	2.863	1.378	Statistic		1.772	2.350	5.152
P-Value		0.000	0.000	0.756	P-Value		0.077	0.000	0.560
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		-22.31	-1.15	1.42					
% Point Change		-37.1%	-6.1%	52.1%					
Significance tests:									
Statistic		-7.348	NA	NA					
P-Value		0.000	NA	NA					

Significance Tests: a) T-test; b) F-test

Cycle time standard deviation also improved for the treatment group, dropping from 37.0 days to 21.9 days, a reduction of 40.9%. The control group also decreased in cycle time standard deviation from 40.2 days to 26.2 days, a 14.7% decrease. This means

that control of the process and predictability of cycle time improved for both groups, but more substantially for the treatment group. The results are statistically significant for the treatment group with an F statistic of 2.863 and $p < 0.001$. As a result, the null hypothesis of no change is rejected. For the control group, the F -statistic of 2.350 is also above the critical value, and with $p < 0.001$ the null hypothesis is also rejected. The reduction in cycle times for both groups is not due to chance. Changes in cycle time skewness are minimal for the treatment group (-0.2, -2.2%), but the control group dropped by more than half, from 2.65 to 1.21, although neither result is statistically significant.

The difference-in-difference analysis reveals a 22.3-day difference in cycle time reduction between the treatment and control groups, statistically significant at $p < 0.001$. Thus, we reject the null hypothesis that the difference-in-difference of the mean cycle times is due to chance. Furthermore, the reduction in cycle time for coaching in Experiment Two (27.8 days) is virtually the same as retraining in Experiment One (31.1 days) but has the added benefit of no required shutdown of production.

Examining the cumulative flow diagrams in Experiment Two, productivity improved in the treatment group following the coaching sessions from May through July 2017. As shown in Figure 19, before coaching the treatment group exhibited a batch process similar to that of the control group in Experiment One, Figure 13, where work items are loaded up at the start of the development cycle and worked down through the period. Departures exceed arrivals until the backlog of work is completed.

Figure 19. Experiment Two: Treatment Group CFD - Pre-Coaching
1/17 through 5/17 - Batch process

As shown in Figure 20, after the coaching sessions, there is less evidence of a batch process. By early September, shortly after the last coaching session, work-in-process virtually disappeared and the process seemed to be very stable. Throughout the period after coaching, both cycle time and work-in-process were shorter than before coaching.

Figure 20. Experiment Two: Treatment Group CFD - Post-Coaching
5/17 through 11/17 -Diminished batch process, lower WIP, shorter cycle time

For the control group, the cumulative flow diagrams show little significant change from before to after the coaching period. In both phases, the control group exhibits a stable process with a consistent level of completion of the work items through the period. Departures parallel arrivals both before (Figure 21) and after coaching (Figure 22).

Figure 21. Experiment Two: Control Group CFD - Pre-Coaching

1/17 through 5/17 - Stable cycle time and work-in-process

Figure 22. Experiment Two: Control Group CFD - Post-Coaching

5/17 through 11/17 - Stable cycle time and work-in-process

Table 11 indicates a slight improvement in flow efficiency for the treatment group, increasing from 55.3% to 62.7%, with no change for the control group at approximately 65% both before and after the coaching period. Daily throughput declined similarly for the treatment and control groups, with both experiencing reductions in throughput of 30.8% and 27.7% respectively. The statistics for change in daily throughput were also not significant with p -values in excess of 0.05, resulting in failure to reject the null hypotheses that the differences are caused by chance.

Table 11

Experiment Two: Summary of Results - Flow Efficiency, Throughput, WIP

Treatment Group - Coached					Control Group - Not Coached				
	Work	Mean	Cycle Time			Work	Mean	Cycle	
	Items	Cycle Time	Std Dev	Cycle Time		Items	Cycle Time	Time Std	Cycle Time
		(a)	(b)	Skewness (b)			(a)	Dev (b)	Skewness (b)
Pre-Coaching					No Coaching				
1/1/17 - 5/24/17	112	53.54	36.98	0.97	1/1/17 - 5/24/17	255	36.96	40.20	2.65
Post-Coaching					No Coaching				
5/25/17 - 11/13/17	85	25.79	21.85	0.94	5/25/17 - 11/13/17	222	31.52	26.22	1.21
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		-27.75	-15.1	-0.02	Differences		-5.44	-14.0	-1.44
% Change		-51.8%	-40.9%	-2.2%	% Change		-14.7%	-34.8%	-54.4%
Significance tests:					Significance tests:				
Statistic		6.574	2.863	1.378	Statistic		1.772	2.350	5.152
P-Value		0.000	0.000	0.756	P-Value		0.077	0.000	0.560
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		-22.31	-1.15	1.42					
% Point Change		-37.1%	-6.1%	52.1%					
Significance tests:									
Statistic		-7.348	NA	NA					
P-Value		0.000	NA	NA					

Significance Tests: a) T-test; b) F-test

The results for changes in work-in-process indicate substantial improvement for the treatment group, and to a lesser degree, the control groups as well. Following the coaching sessions, average daily work-in-process declined from 27.3 stories to 13.3 stories, a 51.1% reduction. The control group also declined 19.1%, from 48.4 to 39.2 stories. Changes for both the treatment and control groups were statistically significant,

with t -statistics above the critical values and $p < 0.001$. In both cases, the null hypothesis that the reduction of work-in-process is due to chance is rejected. However, it is clear that the improvements are significantly stronger for the treatment team compared to the control teams.

Table 12

Experiment Two: Results by Team

	Pre-Coaching (1/1/17 - 5/24/17)					Post-Coaching (5/25/17-11/13/17)					% Change				
	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories
Common Teams															
ADT	0.31	10.61	37.05	66.0	42	0.15	2.87	20.24	68.5	21	-51.6%	-73.0%	-45.4%	3.8%	-50.0%
ASCII	0.20	4.61	30.63	86.6	27	0.02	0.72	36.33	32.1	3	-90.0%	-84.4%	18.6%	-62.9%	-88.9%
SC 3D	0.12	4.53	84.14	48.6	7	0.19	10.89	59.10	59.3	30	58.3%	140.4%	-29.8%	22.0%	328.6%
Pharm	0.43	10.15	24.42	71.0	26	0.23	4.74	21.53	70.6	15	-46.5%	-53.3%	-11.8%	-0.6%	-42.3%
ED	0.40	7.36	20.86	80.7	57	0.24	5.20	22.67	70.3	36	-40.0%	-29.3%	8.7%	-12.9%	-36.8%
HTML	0.06	0.41	8.00	88.9	6	0.15	6.76	46.33	56.8	24	150.0%	1548.8%	479.1%	-36.1%	300.0%
Content	0.22	2.44	28.23	46.3	31	0.32	6.48	21.31	89.5	52	45.5%	165.6%	-24.5%	93.3%	67.7%
DA	0.17	2.33	59.20	95.6	10	0.02	1.80	100.00	100.0	2	-88.2%	-22.7%	68.9%	4.6%	-80.0%
EU Extensions	0.10	1.96	24.00	100.0	12	0.06	1.26	20.86	76.4	7	-40.0%	-35.7%	-13.1%	-23.6%	-41.7%
SPW UI	0.15	9.94	68.60	74.1	20	0.07	5.52	74.67	23.2	6	-53.3%	-44.5%	8.8%	-68.7%	-70.0%
Implement	0.07	0.91	37.00	70.3	6	0.22	4.70	22.15	71.3	20	214.3%	416.5%	-40.1%	1.4%	233.3%
Prt & Rpt	0.08	6.00	112.00	39.5	11	0.05	0.75	15.67	76.3	6	-37.5%	-87.5%	-86.0%	93.2%	-45.5%
Total Control	1.77	48.43	36.96	65.0	255	1.28	39.16	31.52	65.8	222	-27.7%	-19.1%	-14.7%	1.2%	-12.9%
Orders Blue	0.17	8.04	36.50	58.90	24	0.24	6.60	28.18	74.6	33	41.2%	-17.9%	-22.8%	26.7%	37.5%
Order Green	0.23	18.25	63.19	46.00	53	0.52	12.25	24.76	52.6	50	126.1%	-32.9%	-60.8%	14.3%	-5.7%
Orders White	0.24	9.18	50.63	71.10	35	0.03	0.32	12.00	NA	2	-87.5%	-96.5%	-76.3%	NA	-94.3%
Total Treatment	0.78	27.26	53.54	55.30	112	0.54	13.34	25.79	62.3	85	-30.8%	-51.1%	-51.8%	12.7%	-24.1%
Non-Common Teams															
DPC-QSMD	NA	NA	NA	NA	NA	0.25	2.52	11.08	89.6	13	NA	NA	NA	NA	NA
Sensai	0.13	6.30	48.25	79.10	16	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MedRecFormulary	NA	NA	NA	NA	NA	0.22	6.44	30.47	49.7	19	NA	NA	NA	NA	NA
ProblemList	NA	NA	NA	NA	NA	0.38	9.36	25.95	72.3	63	NA	NA	NA	NA	NA
Total Non-Common	0.13	6.30	48.25	79.10	16	0.57	13.47	23.41	67.8	95	NA	NA	NA	NA	NA

The results by team in Table 12 show that an increase in the number of stories processed by a team generally results in an increase in the average daily throughput and work-in-process.⁷ Teams such as SC 3D, HTML, Content, and Implement all increased in number of stories processed and all also increased in daily throughput and work-in-process. Conversely, teams with lower story volumes are lower on these same measures. This

⁷ Non-common Control teams did not produce stories both before and after the coaching period. As a result, they are excluded from the comparison to the treatment team.

suggests that it is the increase in the volume of stories that is responsible for the increase in daily work-in-process as well as daily throughput.⁸

The hypotheses related to coaching are generally consistent with the results of the analysis:

Treatment group cycle time is substantially lower after coaching, and the improvement is considerably greater than the modest improvement in the control group. The difference-in-difference analysis also shows greater productivity improvement for the treatment group. These results are consistent with the coaching hypothesis 2a.

Treatment group flow efficiency is slightly improved while the control group remained unchanged. These results are somewhat consistent with the coaching hypothesis 2b.

Daily work-in-process improved significantly more for the treatment group. The results are consistent with the coaching hypothesis 2c.

Daily throughput declined for both the treatment group and the control group by comparable percentages. These results are not consistent with the coaching hypothesis 2d.

⁸ The treatment group improved in cycle time, flow efficiency and WIP, but realized lower throughput, which seems contra-indicated. An examination of individual treatment team results reveals that two of the three treatment teams did substantially improve throughput (41.2% and 126.1%), but the lower throughput of the remaining team negatively impacted the overall weighted average.

CHAPTER 6

LEAN ACCOUNTING EXPERIMENT

Description of Necessary Data

I introduced lean accounting measures in the work environment and assessed how these measures influence productivity and employee attitudes. Figure 23 describes the research process used for the lean accounting experiment.

Figure 23. Research Protocol: Lean Accounting

The initial survey was administered in August and September 2017. The lean accounting measures were introduced to the treatment group in October 2017 shortly after the survey was completed. At the conclusion of the development cycle in late February 2018, the treatment and control groups were surveyed again. The survey results and productivity measures were compared to those before lean accounting introduction. The treatment and control groups contain approximately 110 and 330 developers, respectively.

The site for the study is the same as that used in the retraining and coaching experiments. Discussions with the firm's management indicates that their lean operations and managerial accounting systems are not fully aligned.

Methodology for Data Collection and Analysis

I conducted informal and semi-structured interviews of managers in development, operations, and finance. Interview questions were used to assess the current status of lean software development and operational practices and to identify which lean management accounting metrics would be most valuable as productivity measures. These questions are listed in the Appendix. Working with management, we jointly agreed upon the most appropriate metrics and the graphical presentation of these metrics for ease in understanding. The selected metrics are work-in-process, cycle time, flow efficiency, and throughput. Some lean/agile software development organizations are starting to measure work-in-process in terms of days in production (Vacanti, 2015). Software development has a highly complex build process, with many stages of production and significant costs tied up in work-in-process.

On a bi-weekly basis, the treatment group received metrics on cycle time, throughput and flow efficiency. Additionally, the teams received reports indicating how closely they are performing compared to forecasts of throughput developed using a Monte Carlo simulation. Work-in-process is reviewed at daily team meetings using a video monitor. Work-in-process is very dynamic and is controlled on a daily basis. It is fundamental to improvements in the other productivity metrics. These metrics are actively used by the development teams in the treatment group to prioritize tasks and

resources, and modify processes. Before the rollout of the metrics, scrum masters who are responsible for the development process within the teams were trained on the use of the metrics in meetings to ensure consistency in the exposure.

Use of certain metrics is sensitive. The development managers are reluctant to share data with the staff that focuses on productivity per full time equivalent position (FTE). Managers believe that a focus on productivity per person may drive competitive pressures and force higher volumes with less staff, ultimately resulting in worse quality and adverse downstream consequences. In lean production, the objective is to maximize flow and minimize work-in-process; thus, any improvement of productivity per FTE is a secondary consideration and can distract from the larger goals.

Figures 24 through 27 provide the distributed lean accounting reports that include all of the agreed-upon metrics. Figures 28 represents the work-in-process data viewed daily on the video monitors and Figure 5 earlier explained the work-in-process data.⁹

Consistent with the literature on use of lean accounting measures in manufacturing, the measures in this report are visual, easy for workers to understand, operationally based, and identify flow and bottlenecks (Kennedy & Widener, 2008; Fullerton et al., 2013, 2014; Petersen & Wohlin, 2010).

⁹ One of the treatment group teams, Edge, did not have access to these reports due to different software development platforms. Although this team used reports from the Actionable Agile tool, including scatterplots, continuous flow diagrams, flow efficiency and throughput reporting, and daily work-in-process monitors, I excluded them from the analysis since they were not subject to the same treatment as the other treatment teams.

Figure 24. Cycle Time Report

In Figure 24 the Cycle Time report identifies the average cycle time for 100% of the stories completed each week and the best 85th percentile of stories completed each week, leaving out the worst performing stories. Additionally, this report identifies the cycle time for the 85th percentile story and the 100th percentile story (the longest cycle time story). This graphic provides both the average performance and impact of outliers on the average.

Figure 25. Flow Efficiency Report

Figure 26. Throughput Report

Figures 25 and 26 identify the status of flow efficiency and throughput by week over the preceding 10 weeks so that trends are immediately recognizable.

Figure 27. Forecast Accuracy Report

Figure 27 illustrates how well the team is performing against a Monte-Carlo generated forecast of story throughput by week. As performance varies from forecast, explanations of the cause of the variance (e.g. change in staffing levels, change in requirements, etc.) can be used to adjust the forecast inputs to generate a new forecast.

Figure 28 displays the work-in-process information reviewed daily on video monitors by the teams. Each block represents a story at a stage in the production cycle, and the number at the top of the column represents the work-in-process limit that should not be exceeded at that stage. The number of blocks (stories) on the video board represents all work-in-process, throughput is represented by the blocks exiting the process, and cycle time is the period required to pass from the initial stage of the process

to the final stage. Cycle time days spent in “done” column status reduce flow efficiency because they represent undesirable queueing delays.

Figure 28. Work-in-Process Report (via monitor)

These metrics are easily understood by the development staff and they specify which parts of the development process are impeding efficiency and lengthening cycle time, enabling remedial action. The reports are issued bi-weekly at the production unit level and reviewed at each staff meeting. Work-in-process reports are reviewed daily.

Discussion of Results

Similar to the retraining and coaching experiments, the analysis of lean accounting is represented in two formats. First, I visually analyze the productivity impact by evaluating scatterplots and cumulative flow diagrams. Second, I analyze the statistical impact of lean accounting on software development cycle times, comparing treatment

group and control group results before and after introduction of the lean accounting metrics, including a difference-in-difference analysis. I also examine the statistical evidence for improved productivity in flow efficiency, work-in-process, and throughput.

Productivity Results

The impact of the lean accounting metrics is evident in the Figure 29 scatterplot which depicts the cycle times for each story, represented by a dot, over the study period. The Y-axis represents days to complete a story and the X-axis represents the date on which the story was completed. Before distribution of the lean accounting reports, the treatment group had produced 297 stories during the development cycles that ran between January 1, 2017 and October 21, 2017. Following distribution of the reports, 93 stories were completed by the treatment group. For the control group displayed in Figure 30, 1,218 and 534 stories were completed, respectively, before and after metric distribution. A visual review of the scatter plots indicates greater dispersion in the treatment teams both before and after distribution of metrics. However, following distribution of metrics, the maximum cycle time for the treatment group remains relatively consistent while the control group increases, exceeding 200 days (Figures 29 and 30, respectively). The maximum story cycle time for the treatment group remains between 170 and 180 days before and after metric distribution.

Figure 29. Treatment Group Scatterplot

No change in cycle time distribution for stories

Figure 30. Control Group Scatterplot

Slight increase in cycle time distribution for stories

The CFD for the treatment group prior to metric rollout is provided in Figure 31. The arrival and departure lines indicate a process in relative control, with work-in-process generally stable through the period.

Figure 31. Experiment Three: Treatment Group CFD - Pre-Metric Rollout
1/17 through 10/17 - Stable cycle time and work-in-process

After metric introduction, work-in-process also appears to be in relative control in Figure 32, with a modest batch process indicated as the arrival and departure lines slightly converge. Note that the scale of the two figures is different, with Figure 31 having almost 300 cumulative stories over almost 11 months compared to 93 stories for four months in Figure 32.

Figure 32. Experiment Three: Treatment Group CFD - Post-Metric Rollout
10/17 through 2/18 - Modest batch process

For the control group, the CFDs reveal much the same pattern as for the treatment group. Work-in-process appears to be stable and in control throughout the pre-metric rollout period in Figure 33, as well as following the rollout of metrics in Figure 34. Again, there are considerable differences in scale between the two figures.

Figure 33. Experiment Three: Control Group CFD – Pre-Metric Rollout

1/17 through 10/1 - Stable cycle time and work-in-process

Figure 34. Experiment Three: Control Group CFD – Post-Metric Rollout

10/17 through 2/18 - Stable cycle time and work-in-process

In summary, there do not appear to be meaningful differences between the treatment and control groups with respect to CFDs. In the pre-metric period, the treatment group story volumes appear to have significant variation while the control group is more consistent, as seen in the relatively smooth upward trend for both arrivals and departures. After metric introduction, the treatment group appears to be slightly converging, indicating a batch process similar to that seen in the control group CFD before and after retraining (Figures 15 and 16). The CFDs indicate no material improvement in productivity following introduction of the lean accounting metrics.

Table 13

Experiment Three: Summary of Results - Cycle Time

Treatment Group - Exposed to Metrics					Control Group - No Metric Exposure				
	Work Items	Mean Cycle Time (a)	Std Dev (b)	Cycle Time Skewness (b)		Work Items	Mean Cycle Time (a)	Cycle Time Std Dev (b)	Cycle Time Skewness (b)
Pre-Rollout					Pre-Rollout				
1/1/17 - 10/21/17	340	30.68	30.33	2.27	1/1/17 - 10/21/17	1,311	22.00	22.16	2.54
Post-Rollout					Post-Rollout				
10/22/17 - 2/14/18	93	40.61	34.40	1.64	10/22/17 - 2/14/18	427	19.42	16.63	1.42
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		9.93	4.07	-0.63	Differences		-2.58	-5.52	-1.11
% Change		32.4%	13.4%	-27.8%	% Change		-11.7%	-24.9%	-43.9%
Significance tests:					Significance tests:				
Statistic		-2.528	0.777	6.974	Statistic		2.548	1.774	9.757
P-Value		0.012	0.057	0.000	P-Value		0.011	0.000	0.000
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		12.51	9.59	0.48					
% Point Change		44.1%	38.3%	16.2%					
Significance tests:									
Statistic		-7.348	NA	NA					
P-Value		0.000	NA	NA					

Significance Tests: a) T-test; b) F-test

There does not appear to be statistical support for reduced cycle times for the treatment group in the post-metric rollout period as represented in Table 13. In fact, average cycle time for the treatment group actually increased by almost ten days (32.4%). Cycle time standard deviation also increased by over four days (13.4%), indicating that

predictability and control of the software development process deteriorated. Skewness fell (-27.8%) reflecting improvement in the consistency of the variation in cycle times. The cycle time and skewness results are statistically significant, indicating that the null hypotheses - that any difference between the pre- and post-rollout period are due to chance - can be rejected. However, the direction of the change in cycle time does not support the conclusion that the introduction of metrics improves software development productivity.

After rollout, the control group experienced a modest decline in cycle time of 2.6 days (11.7%) as well as decreases in cycle time standard deviation (5.5 days, 24.9%) and skewness (1.11, 43.9%). Again, similar to the treatment group, cycle time and skewness measures are statistically significant. The 24.9% reduction in cycle time standard deviation is also statistically significant and contrasts with the treatment group increase of 13.4%.

The treatment group experienced a productivity decline compared to the control group, as indicated in the difference-in-difference analysis. Cycle time change for the treatment group is 12.5 days (44.1%) greater than the change for the control group. The results are statistically significant with $p < 0.001$. Similarly, changes in cycle time standard deviation and skewness are also less favorable for the treatment group relative to the control group.

Table 14

Experiment Three: Summary of Results - Flow Efficiency, Throughput, WIP

Treatment Group - Exposed to Metrics					Control Group - No Metric Exposure				
	Work	Flow	Daily			Work	Flow	Daily	
Pre-Rollout	Items	Efficiency	Through-put	Daily WIP	Pre-Rollout	Items	Efficiency	Through-put	Daily WIP
1/1/17 - 10/21/17	340	48.7%	1.16	28.71	1/1/17 - 10/21/17	1,311	65.8%	4.47	84.06
Post-Rollout					Post-Rollout				
10/22/17 - 2/14/18	93	62.2%	0.80	26.57	10/22/17 - 2/14/18	427	70.3%	3.71	68.39
Change in Productivity Measures - Treatment Group					Change in Productivity Measures - Control Group				
Differences		13.5%	-0.36	-2.14	Differences		4.5%	-0.76	-15.7
% Change			-31.0%	-7.5%	% Change			-17.0%	-18.6%
Significance tests:					Significance tests:				
T-test			2.107	1.847	T-test			1.449	5.110
P-Value			0.036	0.066	P-Value			0.149	0.000
Change in Productivity Measures - Treatment vs Control Group									
Difference in Difference		9.0%	0.4	13.5					
% Point Change		NA	-14.0%	11.2%					
Significance tests:									
Statistic		NA	NA	NA					
P-Value		NA	NA	NA					

Examining other productivity measures in Table 14, the treatment group realized a 13.5% point increase in flow efficiency. The control group also improved 4.5% points and remained higher than the treatment group in absolute terms. For the treatment group, daily throughput declined -0.36 (-31.0%) after metric introduction, as did daily work-in-process, declining -2.14 (-7.5%). Results for daily throughput are statistically significant with the t -statistic of 2.107 above the critical value and $p < 0.05$, rejecting the null hypothesis that these differences are due to chance. The daily work-in-process results for the treatment group is not statistically significant so differences due to chance cannot be ruled out.

The conclusion that the treatment group results do not support the hypothesis that lean accounting metrics improve software development productivity is also borne out in the individual team results. As noted Table 15, virtually all of the control teams

experienced reductions in cycle time but neither of the treatment teams did. Average daily work-in-process also declined for all but one of the 11 control teams but only one of the two treatment teams had this reduction. Improvements in flow efficiency were experienced by all control and treatment teams; however, daily throughput and work-in-process declined for virtually all teams.

Table 15

Experiment Three: Results by Team

Common Teams	Pre-Metrics (1/1/17 - 10/21/17)					Post-Metrics (10/22/17-2/14/18)					% Change			
	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency	# Stories	Average Daily Through put	Average Daily WIP	Cycle Time	Flow Efficiency
Bronze/Copper	0.25	5.61	26.05	86.2%	73	0.40	10.67	27.80	89.2%	45	60.0%	90.2%	6.7%	4.8%
Brown	0.67	9.34	16.62	73.6%	195	0.41	6.60	17.29	79.3%	45	-38.8%	-29.3%	4.0%	3.1%
Burgundy	0.15	2.86	24.00	60.6%	43	0.31	2.61	9.29	34.7%	28	NA	NA	NA	NA
Crimson	0.24	6.01	27.36	91.3%	67	0.18	2.54	15.50	94.8%	10	-25.0%	-57.7%	-43.3%	4.6%
CTM	0.87	15.84	20.76	71.9%	256	0.77	15.07	20.56	77.8%	84	-11.5%	-4.9%	-1.0%	6.8%
Magenta	0.04	2.12	109.38	36.7%	8	0.02	0.98	45.00	60.0%	1	-50.0%	-53.8%	-58.9%	24.9%
Maroon	0.48	12.40	33.72	59.6%	135	0.37	7.59	21.73	74.2%	41	-22.9%	-38.8%	-35.6%	2.2%
Orange (Core)	0.61	10.41	18.48	50.9%	179	0.65	10.06	16.38	67.5%	68	6.6%	-3.4%	-11.4%	18.1%
Regs	0.37	6.83	19.84	47.6%	107	0.21	2.85	14.74	65.7%	23	-43.2%	-58.3%	-25.7%	16.6%
Sapphire	0.49	7.45	16.68	62.7%	144	0.50	10.57	22.33	65.1%	57	2.0%	41.9%	33.9%	1.8%
Yellow	0.36	6.79	21.68	66.4%	104	0.22	3.81	18.36	74.1%	25	-38.9%	-43.9%	-15.3%	13.2%
Total Control	4.47	84.06	22.00	65.8%	1311	3.71	68.39	19.42	70.3%	427	-17.0%	-18.6%	-11.7%	9.4%
Blue	0.35	11.31	38.94	57.1%	102	0.22	8.85	51.24	73.0%	25	-37.1%	-21.8%	31.6%	20.1%
Gold	0.81	17.40	27.14	43.5%	238	0.59	17.72	36.71	56.6%	68	-27.2%	1.8%	35.3%	10.0%
Total Treatment	1.16	28.71	30.68	48.7%	340	0.80	26.57	40.61	62.2%	93	-31.0%	-7.5%	32.4%	8.6%
Non-Common Teams														
Blue 2	1.00	7.74	18.15	65.6%	27	NA	NA	NA	NA	NA	NA	NA	NA	NA
Brown (Perf)	0.05	0.25	4.50	96.6%	6	NA	NA	NA	NA	NA	NA	NA	NA	NA
Gold 2	0.40	7.35	10.54	61.9%	92	NA	NA	NA	NA	NA	NA	NA	NA	NA
Olive	NA	NA	NA	NA	NA	0.41	5.72	14.96	72.6%	32	NA	NA	NA	NA
Total Non-Common	2.91	31.16	13.1	48.5%	125	0.41	5.72	14.94	72.6%	32	-85.9%	-81.6%	14.2%	4.8%
Indigo	2.49	23.37	11.66	38.8%	732	2.65	3.42	2.29	89.0%	305	6.4%	-85.4%	-80.4%	46.4%

One treatment team, Indigo, was removed from the treatment group as it undertook an atypical project involving a platform change after treatment phase that allowed use of smaller story sizes. This resulted in dramatic reductions in cycle time (-78.4%) and improvement in flow efficiency (+46.4% points) that distorted the treatment group results. Indigo results are provided at the bottom of Table 15. Indigo is also the largest team by story volume, constituting approximately 30% of total story production. These strong favorable results would suggest that Indigo benefitted from the introduction of lean accounting metrics. However, additional discussion of these results with development managers indicates that Indigo has recently been processing stories for a platform change, and that these stories are much smaller than typically experienced.

Based on the results of this analysis, it is difficult to attribute any productivity improvement to the introduction of lean accounting metrics.

The hypotheses related to the introduction of lean accounting metrics are generally not consistent with the analysis:

Cycle time increased for the treatment group while it slightly declined for the control group. These results are not consistent with the lean accounting metrics hypothesis 3a.

Flow efficiency improved three times the percentage point difference for the treatment group compared to the control group (13.5% point improvement compared to 4.5% point improvement). These results are consistent with the lean accounting metrics hypothesis 3b.

Daily work-in-process improved slightly for the treatment group (7.5% reduction) but substantially less than the improvement in the control group (18.6% decrease). These results are not consistent with the lean accounting metrics hypothesis 3c.

Daily throughput declined for both the treatment and control groups, and the decline is substantially greater for the latter. These results are not consistent with lean accounting metrics the hypothesis 3d.

Survey Results

Before and after the lean accounting metrics were distributed, team members were surveyed regarding their perceptions of lean/agile software development practices, employee satisfaction and teamwork, and operational reporting. The survey uses a 5-point Likert scale and the questions are on lean software development characteristics

(Poppendieck & Poppendieck, 2003) as itemized in Appendix C. The Pre-Treatment survey was concluded on September 27, 2017; the Post-Treatment survey was completed on March 7, 2018. Table 1 below summarizes the responses by major question category. Detailed data on the responses to each of the 14 questions are in Appendix A. Original email dates with follow-up emails are also in the Appendix A.

Table 16

Summary of Survey Results

Survey questions:							
On a scale of 1 to 5, responses are: Strongly Agree (1), Agree (2), Neutral (3), Disagree (4), Strongly Disagree (5):							
	Treatment			Control			Point change in variance: Treatment vs Control
	Pre-Metric	Post-Metric	% Variance	Pre-Metric	Post-Metric	% Variance	
My organization/team uses lean practices (6 questions)	2.51	2.53	0.8%	2.72	2.73	0.4%	0.4%
I am satisfied with my work and my team (4 questions)	2.19	2.04	-6.8%	2.28	2.23	-2.2%	-4.7%
Operational reporting is understandable and useful to my work (4 questions)	2.52	2.63	4.4%	2.83	2.96	4.6%	-0.2%
Response rate:							
Total surveys sent:	90	90	0.0%	320	320	0.0%	
Total responses	61	32	-47.5%	108	105	-2.8%	
Total effective responses	47	26	-44.7%	95	99	4.2%	
Total response %	67.8%	35.6%	-32.2%	33.8%	32.8%	-0.9%	
Total effective responses %	52.2%	28.9%	-23.3%	29.7%	30.9%	1.3%	
Note: An effective response provided answers to the questions. Ineffective responses just provided background information. The 90 treatment surveys sent excludes 20 surveys sent to the Edge team excluded from the analysis since Edge was not fully subject to the treatment.							

The initial survey draft was significantly more detailed, consisting of 46 questions. Suggestions from Temple University Fox School of Business faculty were incorporated into the survey.¹⁰ However, the development managers at the research site argued that the revised survey was too long and detailed. Managers explained that staff

¹⁰ Contributions from Lynne Andersson, Susan Mudambi and David Schuff are greatly appreciated.

are frequently surveyed and are more likely to complete brief surveys of 14-15 questions that take 5 minutes to complete. The shortened survey was piloted among 19 development managers and revised further by adding open-ended questions to let employees comment on current metrics.

As noted on Table 16, developers did not materially change in their understanding of lean practices, operational metrics and attitudes toward work and teammates between the pre- and post-metric periods. Ironically, one measure, “Provides the team with measurements of progress” declined 6.2% (from 2.09 to 1.96) for the treatment group but improved 31.1% for the control group (2.28 to 2.99).¹¹ It is interesting to note that during this same period the treatment team actually experienced increased cycle times, so that progress was not evident.

Effective response rates (those for responding and completing all of the questions) were in the 30% range except for the initial treatment group survey which was just above 50%. However, given the population, response rates should have been 75% for the treatment group and 46% for the control group for a 90% confidence level with a 5% margin of error.¹²

Discussion of these survey results with managers at the site indicate that the four-month period (late October, 2017 through mid-February, 2018) is probably insufficient time for the use of metrics to have a meaningful impact on employees. Metrics were reviewed bi-weekly, generally 12 to 15 times over the period, and there were staff absences over the holiday, so overall exposure was somewhat limited. Furthermore, if the

¹¹ See question 5 in Appendix A.

¹² Power analysis is based on the tool at <https://www.qualtrics.com/blog/calculating-sample-size/>, and indicates the minimum sample size needed for statistical support that the survey responses are representative of the population.

metrics are not clearly introduced to the team, developers may become further confused and find the reporting to be distracting and unnecessary. A clear linkage between changing and improving processes and seeing the results in the metrics is needed for developers to give the reports credibility. Given the length of software development cycles, such a change is likely to take several months to have a meaningful impact.

The hypotheses related to employee understanding of lean practices and metrics, and attitudes toward work are not supported:

Results of the employee survey indicate no material change in employee work satisfaction, understanding of lean practices and operational metrics.

CHAPTER 7

CONCLUSIONS

The experimental results are generally consistent with my hypotheses on retraining (H1a-d) and coaching (H2a-d). These are targeted interventions, effective in improving productivity in the near term. Significant improvement in most of the productivity measures is observed.

However, the experiment on lean accounting metrics produced results that were largely inconsistent with my hypotheses (3a-d). There is modest improvement in flow efficiency but other productivity measures are unaffected or even decline. Furthermore, based on the survey results, the display of lean accounting metrics has no observable impact on team use of lean practices, work satisfaction or understanding of operational reporting (H4). Table 17 summarizes the results of the hypotheses tests.

Discussions with management indicated that a longer timeframe is needed to properly evaluate the use of lean accounting metrics, ideally a full development cycle (several months). This study covered late October, 2017 through mid-February, 2018, interrupted by holiday absences in December. Both team managers and developers need more time to fully understand how to apply knowledge from the metrics to daily operations. Introduction of these new lean accounting metrics may result in short-term declines in productivity as teams experiment with work processes to understand how to increase the reported metrics. Finally, some learnings early in the cycle from the metrics may only be implementable in subsequent development cycles.

Table 17

Summary of Hypotheses Results

Hypotheses	Measurements				
	Cycle Time	Flow Efficiency	Through put	WIP	Survey
1a-d	Consistent	Consistent	Consistent	Not Consistent	Not Applicable
2a-d	Consistent	Consistent	Not Consistent	Consistent	Not Applicable
3a-d	Not Consistent	Consistent	Not Consistent	Not Consistent	Not Applicable
4	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Consistent

The impact of this study on productivity is illustrated in the simplified conceptual model in Figure 35 below. The solid lines indicate that the hypotheses are consistent with the results of the experiments, while the dotted line indicates a lack of consistency. Furthermore, this conceptual model excludes the reference to Hypotheses 4 since this did not have a hypothesized relationship to the independent variable, productivity, and the hypothesis was not consistent with the results of the survey.

Figure 35. Simplified Conceptual Model for Productivity

I plan to continue to work with the subject firm to expand this analysis, covering a full development cycle and potentially incorporating financial metrics such as costs per story based on developer costs. Furthermore, data from the Monte Carlo-derived Forecast Accuracy report (Figure 27) can be used for budgeting and forecasting development expenses and understanding variances from prior forecasts. Integrating the lean accounting metrics with financial data will make the value of lean agile practices much more transparent to both developers and senior management.

I have also had discussions with a leading software firm (approximately \$1 billion in sales) that utilizes software as a service (SaaS) as its delivery mode for software features and functions. Unlike the site-based software delivery used by the subject firm of this study, SaaS delivery has a direct connection between completion of software development and revenue generation, since the features and functions are readily available to the customer. With site-based software delivery, intermediate steps and

extended time frames are needed to get the software operational at the customer site. The relationship between software development and revenue realization is far less direct than that realized by the SaaS model. Extending this analysis to a SaaS firm would allow for greater understanding of the impact of lean agile practices on revenue as well as expense. Discussions with the SaaS-based firm indicate strong interest in better understanding this relationship.

Finally, it would be of value to examine the combined effects of training/coaching and the subsequent introduction of metrics. I planned on including this analysis, but the treatment team that received the additional training did not get the same metrics exposure as the rest of the treatment group.

CHAPTER 8

LIMITATIONS OF STUDY

There is always the danger that specific productivity measures become the object or target of improvement rather than reflect broader improvement to processes, as noted in Goodhart's Law or Campbell's Law (Goodhart, 1976; Campbell, 1979). In my experimental setting, the productivity measures have a relatively low profile and have not been emphasized in the organization; therefore, the potential problem of this corruption is somewhat abated. However, as these metrics gain in visibility, and senior management starts to rely on these measures for making management decisions, the issue may need to be addressed more directly. These productivity measures are inputs to the Monte Carlo forecasting model used by the company. These inputs can be manipulated to forecast the impact of improvements in work-in-process and other variables on cycle time and delivery dates (Magennis, 2012).

Further, as previously noted, story size is a variable that impacts throughput and cycle time. While discussions with management indicate that story size does not generally differ materially from period to period within a team, the results for the Indigo team indicate that sudden changes in productivity results must be examined in detail to determine if story size is a factor in the analysis.

As noted in the conclusion, my review of the lean accounting productivity results with management indicates that the impact on team performance is likely to take longer than the evaluation period used in this study. These reports provide new data that must be digested, internalized and translated into decisions by developers. These decisions need time to have a measurable impact on productivity results and developer attitudes.

The survey used to assess developer attitudes had a lower response rate than expected, especially for the treatment group's post-treatment survey which was almost half the pre-treatment survey response rate (35.6% versus 67.8%). The low overall response rate may arise from survey design or employee disinterest from having to complete the same survey twice. Also, responding to a survey request from an unknown outside party may have reduced the willingness to participate in the survey. As a remedy for this problem, future studies can incorporate structured or semi-structured interviews of both treatment and control teams prior to and following the treatment period. This may prove more effective in getting meaningful insights into changes in developer attitudes.

The disparities in length of time before treatment versus after treatment should be reduced in future studies. Comparing a long pre-treatment period to a short post-treatment period will cause shorter story development times after treatment than before. Since both the treatment and control groups are using the same pre- and post-treatment timeframes, performance differences between the two groups reveal relative improvements in productivity. However, using the same full development cycle of several months before and after treatment would permit measurement of productivity changes around the treatment for both groups. The visual depictions of performance in the scatterplots would also benefit from using similar pre and post full development cycles as a basis for the analysis.

CHAPTER 9

EXPECTATION OF RESEARCH CONTRIBUTION

Given the lack of prior empirical research on the productivity improvements using lean software development processes, this study is likely to be significant for practitioners and academics. A review of the literature on training and productivity suggests that experiments using treatment and control groups have not been previously used. Furthermore, the analysis of training and coaching is based upon the results of two experiments, which both produced similar improvements in cycle time and other productivity measures. As noted, most of the prior research relies on surveys or analyses of panel data. Both experiments use a control group from the same organization over the same time periods to reduce the effect of unmodeled events. Furthermore, many studies on training and productivity use an increase in wages as an indicator of a productivity increase. This analysis uses improvements in cycle time, flow efficiency, throughput and work-in-process that are all direct measures of productivity.

Additionally, there is virtually no research on the need to align managerial accounting practices to operational approaches for software development. The misalignment between senior management controls and the culture of lean/agile practices in software development teams was identified as a major hurdle at the 2015 Agile Alliance Conference, impacting 70% to 90% of agile/lean organizations (Denning, 2015). This research is the first to document these lean accounting practices as applied to a major software development firm and will therefore establish a basis for comparison to practices of other firms in this industry, especially if longer evaluation periods are used in the analysis. Replication of this research with other firms over full development cycles

may result in a common set of best practices, similar to what was identified in manufacturing through the Lean Accounting Summit in 2005. The need for understanding, documenting and promulgating the best practices in this area has become even more compelling due to the market demands for continually updated solutions as seen in SaaS.

This research also attempts to identify the impact of transparent, accessible, actionable and visual metrics on employee engagement with and understanding of work processes at the work unit level. Extensions of this research using longer periods of analysis and similar metrics at a consolidated company-wide level may strengthen senior management's appreciation of lean/agile contributions, with the added dimension of integrating financial and resource-based information of interest to senior management.

This analysis has value beyond the software development industry. Software is now a substantial asset in many non-software firms, and the breadth of software's impact through all industries continues to grow. As an example, 32% of Levi Strauss Inc.'s gross PP&E consists of software, much of which is internally developed (\$384 million of \$1,202 million of PP&E, Levi Strauss 10K, 2015). Therefore, measuring the productivity of software development is broadly a business-wide issue, impacting all firms that utilize software as a key asset in operations.

Finally, it is worth noting that although this paper addresses lean production in software development, there are many other processes involved in delivering software products to customers which must also be incorporated if the lean approach to production is ultimately to improve the productivity of the firm. As noted in *The Machine that Changed the World* (Womack et al., 2007), a text that examines the automobile

manufacturing process, all steps necessary to the production process must be incorporated into lean processes, including market assessment, product design, engineering, supply chain, and sales and service. Similarly, many software development firms need to host and install the software, and all software firms need to maintain extensive customer support operations, and develop strong relationships with current and potential customers. Ultimately, the productivity of software development firms requires a consistent lean approach across the whole enterprise. Use of similar metrics throughout the full software production value chain, including software delivery, support, hosting and consulting, will significantly enhance the firm's productivity. Developing software is simply the first stage in delivering value to the end user. To fully realize the value of lean practices, it is necessary to extend lean practices and the supporting metrics throughout the entire enterprise.

BIBLIOGRAPHY

- Andreessen, M., "Why Software is Eating the World", *The Wall Street Journal*, August 20, 2011. Accessed June 5, 2016.
- Albrecht, A. J., (1979). Measuring application development productivity. In *Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium* (pp. 83–92).
- Banker, R. D. & Kauffman, R. J., Automated software metrics, repository evaluation and software asset management: New tools and perspectives for managing Integrated Computer Aided Software Engineering (I-Case) (May 1991). Information Systems Working Papers Series, NYU Working Paper No. IS-91-08, 1991. Available at SSRN: <http://ssrn.com/abstract=1289032>
- Banker, R. D., Potter, G., & Schroeder, R. G. (1993). Reporting manufacturing performance measures to workers: An empirical study. *Journal of Management Accounting Research*, 5, 33-55.
- Barron, J. M., Black, D. A., & Loewenstein, M. A. (1987). Employer size: The implications for search, training, capital investment, starting wages, and wage growth. *Journal of Labor Economics*, 5(1), 76-89.
- Bartel, A. P. (1994). Productivity gains from the implementation of employee training programs. *Industrial relations: a journal of economy and society*, 33(4), 411-425.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Kern, J. (2001). Manifesto for agile software development. www.agileAlliance.org.
- Bell, T. E., & Thayer, T. A. (1976), Software requirements: Are they really a problem? *Proceedings of the 2nd international conference on Software engineering*. IEEE Computer Society Press.
- Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., & Munro, M. (2000). Service-based software: The future for flexible software. *Software Engineering Conference (APSEC) Proceedings*, pp. 214-221. IEEE.
- Birnberg, J. G. (2011). A proposed framework for behavioral accounting research. *Behavioral Research in Accounting*, 23(1), 1-43.
- Bishop, J. (1994). The impact of previous training on productivity and wages. L. M. Lynch (Ed.), *Training and the private sector: International comparisons* (pp. 161-200). University of Chicago Press.

Black, S. E., & Lynch, L. M. (1996). Human-capital investments and productivity. *The American Economic Review*, 86(2), 263-267.

Busco, C., Riccaboni, A., & Scapens, R. W. (2000). Management accounting systems and organizational culture: An institutional framework for interpreting their linkages and processes of change. Paper presented at the *2nd Workshops on Management Accounting Change, Oslo, Norway*, 11-14.

Campbell, D. T. (1979). Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1), 67-90.

Case, K. E., Fair, R. C., Oster, S. M. (2017). *Principles of Microeconomics*, Pearson.

Chenhall, R. H., & Langfield-Smith, K. (1998). The relationship between strategic priorities, management techniques and management accounting: An empirical investigation using a systems approach. *Accounting, Organizations and Society*, 23(3), 243-264.

Dearden, L., Reed, H., & Van Reenen, J. (2006). The impact of training on productivity and wages: Evidence from British panel data. *Oxford Bulletin of Economics and Statistics*, 68(4), 397-421.

Denning, S., Jefferies, R., Ballao, M., Girard, A., Grant, T., Newkirk, J. (August, 2015). Agile Alliance 2015, Industry analyst panel discussion: Agile trends and future discussions. www.agileAlliance.org.

Department of Defense, United States (1988). Military Standard: Defense System Software Development. *DoD-STD-2167A, US Department of Defense*.

Dybå, T., & Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9), 833-859.

Fullerton, R. R., Kennedy, F. A., & Widener, S. K. (2013). Management accounting and control practices in a lean manufacturing environment. *Accounting, Organizations and Society*, 38(1), 50-71.

Fullerton, R. R., Kennedy, F. A., & Widener, S. K. (2014). Lean manufacturing and firm performance: The incremental contribution of lean management accounting practices. *Journal of Operations Management*, 32(7), 414-428.

Fullerton, R. R., & Wempe, W. F. (2009). Lean manufacturing, non-financial performance measures, and financial performance. *International Journal of Operations & Production Management*, 29(3), 214-240.

Goodhart, C. (1976). *Monetary Relationships: A View from Threadneedle Street*. University of Warwick.

Gørtz, K. (2008). Coaching, Lean Processes and the Concept of Flow. D. Drake, D. Brennan, K. Gørtz (Ed.), *The Philosophy and Practice of Coaching: Insights and Issues for a New Era* (pp. 299-315). San Francisco: Jossey-Bass.

Hofer, C., Eroglu, C., & Hofer, A. R. (2012). The effect of lean production on financial performance: The mediating role of inventory leanness. *International Journal of Production Economics*, 138(2), 242-253.

Holzer, H. J., Block, R. N., Cheatham, M., & Knott, J. H. (1993). Are training subsidies for firms effective? The Michigan experience. *ILR Review*, 46(4), 625-636.

Jensen, M. C. (1983). Organization theory and methodology. *The Accounting Review*, 58(2), 319-339.

Kennedy, F. A., & Widener, S. K. (2008). A control framework: Insights from evidence on lean accounting. *Management Accounting Research*, 19(4), 301-323.

Little, J. D. (1961). A Proof for the Queuing Formula: $L = \lambda W$. *Operations Research*, 9(3), 383-387.

Little, J. D. (2011). Little's Law as viewed on its 50th anniversary. *Operations Research*, 59(3), 536-549.

Magennis, T. (2012). Managing Software Development Risk using Modeling and Monte Carlo Simulation. In *Proceedings of the Lean Software & Systems Conference* (pp. 32-53).

Mateer, D. & Coppack, L. (2014). *Principles of Microeconomics*. W. W. Norton, New York.

Melo, C. D. O., Santana, C., & Kon, F. (2012, September). Developers motivation in agile teams. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference* (pp. 376-383). IEEE.

Middleton, P. (2001). Lean software development: two case studies. *Software Quality Journal*, 9(4), 241-252.

Middleton, P., Flaxel, A., & Cookson, A. (2005, June). Lean software management case study: Timberline Inc. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 1-9). Springer Berlin Heidelberg.

- Petersen, K., & Wohlin, C. (2011). Measuring the flow in lean software development. *Software: Practice and Experience*, 41(9), 975-996.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.
- Reinertsen, D. G. (1997). *Managing the Design Factory*. The Free Press, New York.
- Reinertsen, D. G. (2009). *Principles of Product Development Flow* (Vol. 1). Redondo Beach: Celeritas Publishing.
- Shah, R., & Ward, P. T. (2007). Defining and developing measures of lean production. *Journal of Operations Management*, 25(4), 785-805
- Shapiro, R. J. (2014). The U.S. software industry as an engine for economic growth and employment, Sonecon LLC White Paper, September
- Staats, B. R., Brunner, D. J., & Upton, D. M. (2011). Lean principles, learning, and knowledge work: Evidence from a software services provider. *Journal of Operations Management*, 29(5), 376-390.
- Thomke, S., & Reinertsen, D. G. (2012). Six myths of product development. *Harvard Business Review*, 90(5), 84-94.
- Vacanti, D. S. (2015). *Actionable Agile Metrics for Predictability: An Introduction*. Leanpub.com/actionableagilemetrics.
- Vacanti, D. S. (2016). Personal communication, February 22.
- 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.
- Womack, J. P., Jones, D. T., & Roos, D. (2007). *The Machine that Changed the World*. Free Press, New York.
- Zwick, T. (2006). The impact of training intensity on establishment productivity. *Industrial Relations: A Journal of Economy and Society*, 45(1), 26-46.

Graphic images in the paper were partially adapted from:

Traditional manufacturing (Figure 1):

Background Slides on Lean Manufacturing and the Toyota Product Development System. John Cleveland. john@in4c.net. Accessed April 22, 2016.

Lean manufacturing (Figure 2):

Images on conceptdraw.com. Accessed April 22, 2016.

Cycle Time, Work-in-Process and Throughput (Figure 5):

Vacanti (2015).

Waterfall development (Figure 6):

W. W. Royce (1970). "Managing the Development of Large Software Systems" in: In: *Proceedings of IEEE WESCON 26* (August): 1–9.

Scatterplots and Cumulative Flow Diagrams have been created courtesy of Actionable Agile Inc., (<http://www.actionableagile.com>).

APPENDIX A

SURVEY QUESTIONS AND RESPONSES

Survey questions:							
On a scale of 1 to 5, responses are: Strongly Agree (1), Agree (2), Neutral (3), Disagree (4), Strongly Disagree (5):							
	Treatment			Control			Point change in variance: Treatment vs Control
	Pre-Metric	Post-Metric	% Variance	Pre-Metric	Post-Metric	% Variance	
My organization/team uses lean practices (6 questions)	2.51	2.53	0.8%	2.72	2.73	0.4%	0.4%
I am satisfied with my work and my team (4 questions)	2.19	2.04	-6.8%	2.28	2.23	-2.2%	-4.7%
Operational reporting is understandable and useful to my work (4 questions)	2.52	2.63	4.4%	2.83	2.96	4.6%	-0.2%
My organization or team:							
1. Uses reporting to make waste visible such as partially done work, extra processes, extra features, task switching, defects, and waiting.	2.94	2.62	-10.9%	2.99	3.02	1.0%	-11.9%
2. Reduces cycle time using small batches with steady arrival rate and departure rate throughout the workflow.	2.55	2.77	8.6%	2.83	2.81	-0.7%	9.3%
3. Allows team members to choose the best means for them to accomplish their work and make their own commitments.	2.49	2.58	3.6%	2.63	2.65	0.8%	2.9%
4. Motivates employees by helping them understand the purpose of their work.	2.66	2.73	2.6%	2.82	2.83	0.4%	2.3%
5. Provides the team with measurements of progress.	2.09	1.96	-6.2%	2.28	2.99	31.1%	-37.4%
6. Continuously and systematically improves how work is done so that processes stay healthy and productive over time.	2.32	2.54	9.5%	2.77	2.71	-2.2%	11.6%
I am satisfied with:							
7. My ability to do interesting work in my role.	2.38	2.08	-12.6%	2.35	2.73	16.2%	-28.8%
8. My ability to apply my skills and experience in my current role.	2.28	2.19	-3.9%	2.27	2.62	15.4%	-19.4%
9. The ability of my team to communicate effectively with each other.	2.09	1.96	-6.2%	2.28	2.26	-0.9%	-5.3%
10. The ability of my team to work effectively with each other.	2.02	1.92	-5.0%	2.20	2.21	0.5%	-5.4%
Regarding the operational performance reporting of the software development process, I am satisfied with:							
11. The regular and frequent availability of the reporting for my team.	2.38	2.35	-1.3%	2.68	2.82	5.2%	-6.5%
12. The clarity and ease of understanding the reporting that my team receives.	2.32	2.62	12.9%	2.83	2.94	3.9%	9.0%
13. The usefulness of reporting that my team receives to help us improve operations.	2.64	2.81	6.4%	2.82	3.02	7.1%	-0.7%
14. How reporting explains the efficiency of delivering client value with available resources.	2.74	2.73	-0.4%	3.00	3.06	2.0%	-2.4%
Response rate:							
Total surveys sent:	90	90	0.0%	320	320	0.0%	
Total responses	61	32	-47.5%	108	105	-2.8%	
Total effective responses	47	26	-44.7%	95	99	4.2%	
Total response %	67.8%	35.6%	-32.2%	33.8%	32.8%	-0.9%	
Total effective responses %	52.2%	28.9%	-23.3%	29.7%	30.9%	1.3%	
Note: An effective response provided answers to the questions. Ineffective responses just provided background information. The 90 treatment surveys sent excludes 20 surveys sent to the Edge team excluded from the analysis since Edge was not fully subject to the treatment.							

The pre-metric survey email was sent on August 11, 2017. An alert was sent out on August 4, 2017 describing the study and informing the teams that the survey would be sent out shortly. A reminder email to complete the survey was sent out on August 25, and again on September 20, 2017. The survey was closed on September 27, 2017.

The post-metric survey was sent out on February 14, 2017 and a reminder email was sent out on February 26, 2018 and again on March 5, 2017. The survey was closed on March 7, 2018.

APPENDIX B STUDY PROTOCOL AT RESEARCH SITE

Requested access:

At each site, access to software development and finance and accounting managers for initial open-ended and semi-structured interviews (60 to 90 minutes each) followed up by more in depth meetings (two to four hours) with the same managers to identify measures appropriate to the study.

Requested information:

State of implementation of lean software development processes and practices and related processes (hosting, support, implementation) from initial concept development through delivery of value to the customer (resulting in revenues).

Semi-structured questions for the development organizations:

1. What specific metrics are used to determine productivity in your organization?
2. What reports or other data are available to operations management to evaluate their productivity?
3. To what level of your organization is this data available?
4. How frequently is this data distributed?
5. Is the reporting timely?
6. Do you and your team fully understand the performance reports that you receive?
7. Does the performance reporting help you and your team improve operations?
8. Are there reports issued to you or your team that have little or no value?
9. How is this data translated into or used for company financial data?
10. How are your organization's activities prioritized, planned and budgeted?
11. Which reports are used for near term (current day, week, month), mid-term (current year) and long term/strategic decision-making (multiple years)?
12. How does the performance management reporting process contribute to (or hinder) agile/lean processes?
13. How are continuous improvement practices incorporated into operations?
14. How is employee empowerment encouraged in operations?
15. How extensive is cross training of your staff?
16. What is the management review process for selection and termination of software development projects?
17. What is the management review process for monitoring of progress on software development projects?

Semi-structured questions for the finance/accounting (F&A) organization:

1. How does the F&A organization evaluate the productivity of the various operations organizations?
2. What reports or other information are available to F&A to evaluate the productivity of the various operations organizations?
3. Who receives operational and financial reporting/tracking data and how does it impact operations?
4. Are there any lean accounting practices in place (visual performance measurements, simplified reporting, value stream mapping)?
5. Explain the annual budgeting process.
6. Explain the financial forecasting process.
7. How do the operational organizations participate in the budgeting and forecasting process?
8. Which financial/accounting reports have little or no value?
9. What reports requested by operations organizations are not being produced or delivered?
10. What is the management review process for selection and termination of software development projects?
11. What is the management review process for monitoring of progress on software development projects?
12. What is the impact of lean practices on forecasting, budgeting and strategic planning?

APPENDIX C
BASIS FOR SURVEY QUESTIONS
(Poppendieck & Poppendieck, 2003)

1. **Eliminate waste** - Does your organization:
 - a. Use reporting make visible waste: partially done work, extra processes, extra features, task switching, defects, waiting, motion?
 - b. Use value stream mapping to identify critical paths used to deliver value to the customer?
2. **Amplify learning** - Does your organization:
 - c. Provide get feedback and input from an immediate customer?
 - d. Develop software in iterations: designed, tested, integrated and delivered during a short, fixed timeframe with negotiated scope and team commitment?
 - e. Build software in small batches and regularly synchronized and stabilized?
 - f. Use set-based development that identifies constraints rather than point-based development that offers choices? (Note: Set-based development makes decisions based on known constraints rather than point-based development which provides multiple choices that may work under some circumstances but not under all conditions.)
3. **Decide as late as possible** - Does your organization:
 - g. Reduce high stakes constraints; develop software that is easy to change; use concurrent software development?
 - h. Employ options thinking, providing the flexibility to make decisions as late as possible?
 - i. Avoid losing an important alternative by failing to make a timely decision at the last responsible moment?
 - j. Encourage making decisions on breadth first rather than depth first decisions and, with experience, learn to make intuitive decisions?
 - k. Use simple rules where possible allowing for a framework for self-organization?
4. **Deliver as fast as possible; complements “decide as late as possible”** - Does your organization:
 - l. Pull customer work into the development process based on team choice?
 - m. Employ queuing theory by reducing cycle time using small batches with steady arrival rate and departure rate throughout the system?
 - n. Keep some slack in the systems for capacity?
 - o. Understand and make transparent the cumulative multi-year full cost of delay to the customer/market including market share loss?

5. **Empower the team** - Does your organization:
 - p. Allow employees to decide/choose the best means for them to accomplish their work and make their own commitments?
 - q. Motivate employees by helping them understand the purpose of their work and how it helps the customer?
 - r. Provide employees with a safe working environment, a sense of belonging, assurance of competence and measurements of progress?
 - s. Set direction, align people, enable motivation, using the concept of a *shusa*, master designer or chief engineer, who sets direction and guides development through the “fuzzy front end”?
 - t. Use Project leaders provide the operational basis for a lean organization?
 - u. Develop and foster communities of expertise that will typically create the standards (naming, build, code check-in/check-out, etc.)?

6. **Build integrity in** - Does your organization:
 - a. Promote perceived integrity: product has a balance of function, usability, reliability and economy that is valued by the customer?
 - b. Promote conceptual integrity as a prerequisite for perceived integrity, i.e., the product’s central concepts work well together as a smooth, collective whole, balancing flexibility?
 - c. Continuously improve the system as it evolves so internal systems stay healthy over time?
 - d. Employ unit, system, and integration tests *as developer tests*?
 - e. Employ acceptance tests as *customer tests*; both run as automated tests and as part of the daily build process?
 - f. Create a “testing scaffolding” process that identifies unintended consequences immediately?
 - g. Create documentation of software “as built”?

7. **See the whole** - Does your organization:
 - v. Use the 5 “whys” to uncover root systemic issues; seek and fix the current constraint and understand that it may move to another place in the system?
 - w. Optimize each sub task through collaboration and coordination between the sub tasks resulting in overall sub optimization of the system?
 - x. Avoid contracts with detailed fixed scope?
 - y. Use shared benefit contracts, time and materials contracts, multi-stage contracts or target-cost contracts, all of which make the contract a collaborative process?

APPENDIX D IRB SUBMISSION

1) Abstract of the study

During the past 15 years, software development has adopted lean practices used by manufacturing for several decades. Similar to the manufacturers, software developers will need to modify managerial accounting practices to effectively measure productivity in a lean/agile environment. This study examines the impact of introduction of lean accounting measures into two lean/agile software development companies. The companies differ in the delivery of software to the customer, one using web-based Software as a Service (SaaS), the other using a more traditional site-based delivery method.

2) Protocol Title

Lean Accounting Comes to Lean Software Development

3) Investigator

Principal Investigator: Dr. Sudipta Basu

Research Assistant: Thomas Stone, DBA student, Fox School of Business

4) Objectives

The purpose of this research is to examine the impact of lean accounting practices on the productivity of software development and organizational attitudes toward lean practices and employee engagement. There will also be an examination of the differences between software development firms using different software delivery practices, since SaaS is considered to be a more lean delivery method.

5) Rationale and Significance

In general, the literature has determined that strategic direction and operational practices must be supported by appropriate management controls in order for an organization to function optimally. Currently, misalignment between management controls and the culture of lean/agile software development adversely impacts most software development organizations. This study will examine the impact of introducing lean management accounting measures to firms where these practices are not fully implemented.

6) Resources and Setting

Beyond the required CITI training, the investigators will ensure that any persons assisting with the study are adequately informed about the protocol and their study-related duties and functions by meeting with each researcher to discuss and review role specific tasks and expectations.

The research interviews will take place at two firms, Cerner Corporation in Kansas City, MO and Malvern, PA and Ultimate Software in Weston, FL. Information will be gathered through In-person meetings, phone and WebEx meetings, and email correspondence.

7) Prior Approvals

No other non-IRB approvals are necessary in order to perform the research. Non-Disclosure Agreements (NDA) have been provided by both firms.

8) Study Design

a) Recruitment Methods

The investigators will work with each firm to determine the employees best suited to participate in this study. It is expected that those interviewed and surveyed will be from all levels of management and staff in various functional areas, including operations, R&D, finance and accounting, and product management. The study aims to recruit approximately 300 participants at each firm. A smaller subset of these participants will also be included in a pilot study prior to executing on the full examination.

b) Inclusion and Exclusion Criteria

Subjects must be employees or contractors of the firms and will participate on an entirely volunteer basis via one of the previously specified subject pools.

c) Study Timelines

The duration of a subject's participation in the study will be less than 30 minutes to complete a survey. Interviews may take longer, up to 90 minutes. Additionally, workshops to development metrics may be conducted which may take 3 to 4 hours. Managers may be asked to participate in more than one interview, depending on the study requirements. The estimated date that the investigators will complete the study is March 31, 2018.

d) Study Procedures and Data Analysis

This is a study of the relationship between use of certain performance metrics and productivity in software development firms. This will require me to survey employees on their perceptions of their organizations. The survey responses will be anonymous and participation will be voluntary. The research approach will be a systematic investigation designed to develop and contribute generalizable knowledge. No payment is received from or provided to the organization or the subjects for this research. There may be reimbursement by the firm of the investigator's travel expenses to visit the firm and conduct the research.

My research will introduce certain new performance metrics to the organization being studied and the employees will be exposed to these metrics. Managers will reference these metrics at employee meetings to re-enforce the visibility and understanding of the metrics. I will use an experimental design approach, where one treated group of employees will be exposed to the new metrics and another control group will not have this exposure.

Before and after a period of exposure to the new metrics, both the treated group and the control group will be surveyed to determine differences in perceptions of organizational performance. Additionally, I will measure actual productivity results for

both groups by examining the level of actual output by the groups both before and after the treatment.

Questions used in the survey will ask employees if exposure to these new metrics employees helps them to more fully understand the impact of their work on the performance of the organization and if this understanding motivates the employee to become more productive. Additionally, there will be interpersonal interviews with selected employees and managers to help formulate the questions and to determine performance metrics which are most critical. These interviews will be voluntary and will be recorded and notes will be taken during the interviews. The interviews will be conducted individually or in groups. If conducted in groups, others will be aware of the individual's responses. Otherwise, responses will be kept confidential.

Managers will reference these metrics at employee meetings to re-enforce the visibility and understanding of the metrics. I will use an experimental design approach, where one treated group of employees will be exposed to the new metrics and another control group will not have this exposure.

Studies will be conducted through in-depth unstructured and semi-structured interviews and surveys. Subjects will answer questions about lean software development practices, reporting processes and measures, planning and budgeting, general operational issues, and attitudes toward work. Subjects will complete a survey that includes established, validated measures and interviews for the purposes of creating and validating new construct measures. A list of sample survey questions is provided in this packet. Statistical analysis will be performed to determine any relationships in the data.

e) Withdrawal of Subjects

There are no anticipated circumstances under which subjects will be withdrawn from the research without their consent. There are no consequences of a subject's decision to withdraw from the research. The participant can exit the study at any time without consequence.

f) Privacy & Confidentiality

While the research collects demographic information such as age and gender, no identity revealing information is collected. The study will neither use nor disclose subjects' Protected Health Information (PHI).

9) Risks to Subjects

No physical or emotional discomfort should come to any of the participants. There are no short- or long-term risks to the subjects.

10) Multi-Site Human Research

This research involves multi-site studies. No problems involving risk to subjects or others are anticipated. Participant recruitment, study procedures, and data collection will follow a uniform protocol across all sites to protect subjects and ensure data confidentiality.

11) Potential Benefits to Subjects

Given the nature of the study, participants and their firms can expect to benefit from improved financial and operational metrics, greater employee engagement and increased productivity.

12) Costs to Subjects

There are no costs that subjects may be financially responsible for due to study participation.

13) Informed Consent

The participants in the face-to-face interviews will sign the Informed Consent for Minimal Risk Social and Behavioral Research form, which is provided, before the interviews or experiments.

A waiver of written document of informed consent is requested for subjects participating in the online survey as the research presents no more than minimal risk of harm to subjects and involves no procedures for which written consent is normally required outside of the research context. In order to preserve confidentiality, the informed consent form will be electronically agreed to by the participants before gaining access to the online survey. The text of the consent form for the online survey is presented on the first page of the survey.

14) Vulnerable Populations

The study does not seek to obtain respondents with any specific defining characteristics and will not include individuals who are not yet adults (infants, children, teenagers), pregnant

On-Line Consent Form used with the survey

Research study: Lean Accounting Comes to Lean Software Development

Name and Department of Investigators:

Thomas Stone, Doctoral student, Temple University Fox School of Business

Sudipta Basu, Faculty member, Temple University, Fox School of Business

The purpose of this study is to examine the impact of certain operational and financial metrics on software development productivity and employee attitudes toward lean agile software development.

Your participation in the survey will contribute to a better understanding of the value of tracking and reporting transparent metrics on productivity. The total estimated time to complete the survey is 10 minutes.

Access to the survey data detail during data collection is limited to the investigators noted above. Individual responses, along with any other personally identifiable information about you, will be stripped from the final dataset. Total results of the survey will be used in the study.

Your participation in this survey is voluntary. You may decline to answer any question and you have the right to withdraw from participation at any time without penalty.

Regarding any questions, concerns, or complaints about this study, contact Thomas Stone at 215-421-0030 or tug21815@temple.edu or contact Dr. Sudipta Basu at sudipta.basu@temple.edu or 215-204-0489.

This research has been reviewed and approved by the Temple University Institutional Review Board. Please contact them at (215) 707-3390 or e-mail them at irb@temple.edu for any of the following: questions, concerns, or complaints about the research; questions about your rights; to obtain information; or to offer input.

Confidentiality: Efforts will be made to limit the disclosure of your personal information, including research study records, to people who have a need to review this information. However, the study team cannot promise complete secrecy. For example, although the study team has put in safeguards to protect your information, there is always a potential risk of loss of confidentiality. There are several organizations that may inspect and copy your information to make sure that the study team is following the rules and regulations regarding research and the protection of human subjects. These organizations include the IRB, Temple University, its affiliates and agents, Temple University Health System, Inc., its affiliates and agents, and the Office for Human Research Protections.

By clicking on the link to the survey below I agree that I am at least 21 years old and I agree to participate in this research project.

Research Integrity & Compliance
Student Faculty Center
3340 N. Broad Street, Suite 304
Philadelphia PA 19140

Institutional Review Board
Phone: (215) 707-3390
Fax: (215) 707-9100
e-mail: irb@temple.edu

Certification of Approval for a Project Involving Human Subjects

Date: 11-Apr-2017

Protocol Number: 24436
PI: BASU, SUDIPTA
Review Type: EXEMPT
Approved On: 11-Apr-2017
Approved From:
Approved To:
Committee: A1
School/College: BUSINESS SCHOOL (1500)
Department: FSBM-ACCOUNTING (15020)
Sponsor: NO EXTERNAL SPONSOR
Project Title: Lean Accounting Comes to Lean Software Development

The IRB approved the protocol 24436.

If the study was approved under expedited or full board review, the approval period can be found above. Otherwise, the study was deemed exempt and does not have an IRB approval period.

If applicable to your study, you can access your IRB-approved, stamped consent document or consent script through ERA. **Open the Attachments tab and open the stamped documents by clicking the Latest link next to each document.** The stamped documents are labeled as such. Copies of the IRB approved stamped consent document or consent script must be used in obtaining consent.

Before an approval period ends, you must submit the Continuing Review form via the ERA module. Please note that though an item is submitted in ERA, it is not received in the IRB office until the principal investigator approves it. Consequently, please submit the Continuing Review form via the ERA module at least 60 days, and preferably 90 days, before the study's expiration date.

Note that all applicable Institutional approvals must also be secured before study implementation. These approvals include, but are not limited to, Medical Radiation Committee ("MRC"); Radiation Safety Committee ("RSC"); Institutional Biosafety Committee ("IBC"); and Temple University Survey Coordinating Committee ("TUSCC"). Please visit these Committees' websites for further information.

Finally, in conducting this research, you are obligated to submit the following:

- Amendment requests - all changes to the study must be approved by the IRB prior to the implementation of the changes unless necessary to eliminate apparent immediate hazards to subjects
- Reportable new information - using the Reportable New Information form, report new information items such

as those described in the Investigator Guidance: Prompt Reporting Requirements HRP-801 to the IRB within 5 days

- Closure report - using a closure form, submit when the study is permanently closed to enrollment; all subjects have completed all protocol related interventions and interactions; collection of private identifiable information is complete; and Analysis of private identifiable information is complete.

For the complete list of investigator responsibilities, please see the Policies and Procedures, the Investigator Manual, and other requirements found on the Temple University IRB website: : <http://research.temple.edu/irb-forms-standard-operating-procedures#POLICY>

Please contact the IRB at (215) 707-3390 if you have any questions