2016

Exploring Organizations' Software Quality Assurance Strategies

Scott Underwood
Walden University

Walden University

College of Management and Technology

This is to certify that the doctoral study by

Scott Underwood

has been found to be complete and satisfactory in all respects,
and that any and all revisions required by
the review committee have been made.

Review Committee
Dr. Jamie Klein, Committee Chairperson, Doctor of Business Administration Faculty

Dr. Alexandre Lazo, Committee Member, Doctor of Business Administration Faculty

Dr. Cheryl Lentz, University Reviewer, Doctor of Business Administration Faculty

Chief Academic Officer
Eric Riedel, Ph.D.

Walden University
2016

Abstract

Exploring Organizations' Software Quality Assurance Strategies

by

Scott Underwood

MBA, University of Fredericton, 2010

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Business Administration

Walden University

September 2016

Abstract

Poor software quality leads to lost profits and even loss of life. U.S. organizations lose billions of dollars annually because of poor software quality. The purpose of this multiple case study was to explore the strategies that quality assurance (QA) leaders in small software development organizations used for successful software quality assurance (SQA) processes. A case study provided the best research design to allow for the exploration of organizational and managerial processes. The target population group was the QA leaders of 3 small software development organizations who successfully implemented SQA processes, located in Saint John, New Brunswick, Canada. The conceptual framework that grounded this study was total quality management (TQM) established by Deming in 1980. Face-to-face semistructured interviews with 2 QA leaders from each organization and documentation including process and training materials provided all the data for analysis. NVivo software aided a qualitative analysis of all collected data using a process of disassembling the data into common codes, reassembling the data into themes, interpreting the meaning, and concluding the data. The resulting major themes were Agile practices, documentation, testing, and lost profits. The results were in contrast to the main themes discovered in the literature review, although there was some overlap. The implications for positive social change include the potential to provide QA leaders with the strategies to improve SQA processes, thereby allowing for improved profits, contributing to the organizations' longevity in business, and strengthening the local economy.

Exploring Organizations' Software Quality Assurance Strategies

by

Scott Underwood

MBA, University of Fredericton, 2010

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Business Administration

Walden University

September 2016

Dedication

For Alex and Leah. Thank you for sharing daddy's time and motivating me beyond comprehension. In the winter, may your footsteps never meet.

Acknowledgments

This doctoral study is a shared accomplishment with everyone who offered words of encouragement or gave their time to my family while I was working. I appreciated all of these gestures. I thank my Chair, Dr. Klein, for being a source of encouragement and support. I enjoyed working under your guidance. Thank you to Dr. Lazo and Dr. Lentz, for your help and feedback throughout this process.

I would specifically like to acknowledge my family. Each of you has inspired me to achieve this accomplishment in unique ways and allowed me to lean on your different strengths throughout the process. To Dad, who set the academic standard and offered unwavering support. To Mom, for your strength, resolve, and love. To Lisa, for your focus on family. To the Justasons, for your support and helping to keep our house sane during this journey. Most importantly to Tracy, for your unquestioning love, support, and understanding. We did it.

Table of Contents

Section 1: Foundation of the Study

The purpose of this study was to explore strategies quality assurance (QA) leaders in small software development organizations can use for successful software quality assurance (SQA) processes. In the subsections of Section 1, I lay the foundation to discuss why conducting this study may lead to improving SQA processes in small organizations. A background of the study precedes a description of the problem statement, purpose statement, nature of the study, and the central research question. Next, a discussion regarding the research method and design leads to an overview of total quality management (TQM) as the conceptual framework for this study. A review of the academic and professional literature on TQM regarding this study then transitions into Section 2.

Background of the Problem

Managing SQA is important because of the cost of integrating bug fixes or new features into an application after software development organizations launch an application (Nord, Ozkaya, Kruchten, & Gonzalez-Rojas, 2012). The concept of technical debt is a cost related to poor software quality measured at $3.61 per line of code (Curtis, Sappidi, & Szynkarski, 2012). Organizations fail to budget technical debt costs resulting in spending more money than expected to build applications (Nord et al., 2012). Therefore, these cost overruns are affecting organizations' profitability.

TQM, Capability Maturity Model Integration (CMMI), and International Standards Organization (ISO) 9001 are manufacturing industry QA frameworks used to control quality (Morris, 2012; Pillai, Pundir, & Ganapathy, 2012). Additionally, the

software development industry uses project management methodologies, such as Six Sigma and Agile, for managing quality (Baldassarre, Caivano, Pino, Piattini, & Visaggio, 2012; Rafique & Mišić, 2013). QA leaders of software development organizations require a competitive edge in their industry by lowering costs and reducing time to market (Singh & Kannojia, 2013). Understanding how SQA processes align with industry standards may help improve quality and efficiency, which may allow for lower development costs. The unknown capability of an organization's SQA strategies is a business problem that led to the need for the research on this case study.

Problem Statement

Poor software quality leads to lost profits and even loss of life (Harter, Kemerer, & Slaughter, 2012). U.S. organizations lose $60 billion annually because of poor software quality (Harter et al., 2012). The general business problem is that inadequate SQA processes negatively affect some small software development organizations' profits. The specific business problem is that some quality assurance QA leaders within small software development organizations lack strategies for successful SQA processes.

Purpose Statement

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The target population group was the QA leaders of small software development organizations. The geographic location was Saint John, New Brunswick, Canada. The implications for positive social change include the potential to provide QA leaders with the strategies to improve SQA processes, thereby allowing for improved

profits, contributing to the organizations' ability to stay in business, and strengthening the local economy. The sustained longevity of small organizations becomes important to larger social communities because of contributions to economic prosperity and employment opportunities, especially when considering small organizations employ half of all workers in the United States (Taneja, Pryor, & Hayek, 2016). Therefore, the potential for the sustained longevity of small software development organizations through improved SQA processes can contribute to positive social change.

Nature of the Study

I chose to use a qualitative approach and an exploratory, multiple case study design for this study because the purpose of the research was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The exploratory nature of understanding a phenomenon within qualitative research made this research method the best option (Kapoulas & Mitic, 2012). A quantitative research study of assigning dependent and independent variables to examine their relationships allowed for obtaining actual numerical values of SQA (Yoshikawa, Weisner, Kalil, & Way, 2013). Discovering the organizations' SQA strategies required achievement before measuring the strategies' effectiveness, eliminating a quantitative design (Yoshikawa et al., 2013). The timeliness researchers require to collect data and complete a mixed method study eliminated the mixed method approach as a viable option for this study (Yoshikawa et al., 2013).

More specifically, this study was an exploratory multiple case study. Researchers use case studies to focus on organizational and managerial processes (Yin, 2014). SQA is

a continuous process throughout the software development lifecycle (Galinac Grbac, Runeson, & Huljenić, 2013), making a case study the most appropriate design for this study. Other research designs did not align with the purpose of the study. Narrative researchers focus on individual experiences to lay a chronological order to events, which does not lead to the understanding of phenomenon as related to business practices (De Loo, Cooper, & Manochin, 2015). A phenomenological researcher places a phenomenon as the subject of the research when viewed through the participant's experiences (Giorgi, 2012). However, human experience was not the focus of this study, so this design was not appropriate. The principle focus of grounded theory research is to design a new theory, making grounded theory incompatible with the purpose of this study (Urquhart & Fernández, 2013). Ethnographic researchers focus on participants who share experiences and interactions within a culture (Murthy, 2013; Zilber, 2014). The demographic of the study was not a particular culture leading to the elimination of an ethnographic design as an option for this study.

Research Question

The primary objective of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The central research question, problem statement, and purpose statement align with each other. To accomplish the primary objective, the central research question for the study was: What strategies can QA leaders within small software development organizations use for successful SQA processes?

Interview Questions

The interview questions aligned with the central research question to allow for the participants the share their experiences with their organizations' SQA processes. This alignment helped guide the data collection process. My analysis of the answers led to understanding the central research question. Some questions required additional probing questions to encourage the interview participants to share more information. The interview questions were as follows:

1. How would you describe the strategies you use for managing software quality?
2. What strategies do you use that are most effective for managing software quality?
3. What strategies do you use that are least effective for managing software quality?
4. How would you describe the critical success factors you use to measure your software quality?
5. What types of data do you track and store with relation to SQA?
6. How would you describe the benefits and constraints in adopting a SQA process?
7. How would you describe the project management methodology you use for your software development?
8. How would you describe the strategies you use for testing during the software development lifecycle?

9. What processes do you use for requirements gathering and validation?
10. How would you describe your process for including accessibility and usability when establishing your requirements?
11. What training strategies do you use for your SQA?
12. How would you describe your change management for feature integration in your software development lifecycle?
13. How would you describe your root-cause analysis (RCA) processes?
14. What is an example of an incident where poor software quality lead to decreased profits for your organization?

Conceptual Framework

The conceptual framework for this study was TQM. Deming introduced TQM in 1980 to improve process control and resource management within the manufacturing industry (Petersen, 1999). The U.S. government adopted TQM in military practice with the goal of reducing costs through increased efficiency (Petersen, 1999). TQM is a process of thinking about the organization as a whole to achieve excellence, with the most important principles being leadership and customer focus (Burli, Bagodi, & Kottursheettar, 2012). Organizations use TQM to focus on customer satisfaction, continuous improvement, treating the organization as a total system, and using statistical process control (Malik & Blumenfeld, 2012).

The purpose of this qualitative exploratory multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. SQA is a component of TQM frameworks; therefore, TQM as a

conceptul framework was in alignment with the purpose of the study (A. Brown, 2013; Lin, Chuang, & Shih, 2012; Morris, 2012). The holistic view of TQM contains industry best practices frameworks for quality management (Malik & Blumenfeld, 2012). As a conceptual framework, TQM allowed the ability to compare and contrast the organizations' SQA to highlight areas for improvement to QA leaders (Majstorovic & Sibalija, 2015).

Assumptions, Limitations, and Delimitations

Assumptions, limitations, and delimitations are important considerations when understanding the scope and perspective of the study (Nenty, 2009). Each assumption, limitation, and delimitation sets the context for how the researcher focused the study (Nenty, 2009). The following subsections include the assumptions, limitations, and delimitations for this study.

Assumptions

Research assumptions are the unverifiable items a researcher must hold to be true (Nenty, 2009). Lindgren and Packendorff (2009) noted the importance of detailing assumptions to improve the quality of qualitative research. Yin (2014) identified that detailing research assumptions provided context for the study and offered potential alternatives for future studies. I made three assumptions concerning this study. The first assumption was that the participants were truthful and forthcoming with their answers during the interviews. The second was that the participating QA leaders would willingly share documentation as a secondary data source. Finally, I assumed that the number of selected participants was enough to reach data saturation as defined for this study.

Limitations

Nenty (2009) defined research limitations as the factors that confine and constrain a researcher's study. Simon (2011) indicated limitations could be a source of possible weakness in the study. Yin (2014) noted that ethical qualitative research included divulging research limitations.This study contained three limitations. The first limitation was that the accuracy of the results was reflective of the information shared by the participating organizations. The next was the case study sample size was small and may not be representative of the population. The final limitation was The participants limited their experiences to within the participating organization.

Delimitations

Research delimitations are the factors a researcher uses to focus the scope and explain the restrictions of the study (Nenty, 2009; Simon, 2011; Yin, 2014). This study contained three delimitations. First, this study focused only on software development processes, not the hardware or networking architecture of the software. Second, this study did not compare and contrast one QA tool against another as an exploration of the benefits of one tool over another. Third, the description of software included those applications either installed locally on a user's computer device or accessible through the Internet.

Significance of the Study

This study is of value to businesses through the identification of strategies that organizations need to improve and manage their SQA. Quality frameworks exist as industry best practices established by the International Institute for Software Testing

(IIST) body of knowledge such as TQM, CMMI, ISO 9001, Six Sigma, and Agile that organizations can leverage ("Certified Test Manager", 2015). The freedom of QA leaders to leverage a combination of quality frameworks that best meet the organizations' needs (Burli et al., 2012) led to the purpose of the research of this study. The identification of the existing SQA strategies for the organizations' QA leaders through TQM as a conceptual framework highlighted the industry best practice strategies that may be missing or working (Majstorovic & Sibalija, 2015). In assessing the significance of the study, a review of how the study may help contribute to business practice and the implications for social change follows.

Contribution to Business Practice

This study may contribute to the effective practice of business by strengthening the SQA strategies of the selected organizations of the study (Majstorovic & Sibalija, 2015). The participating organizations' QA leaders may use SQA processes causing significant and relevant concerns that hinder efficiency and profitability (Burli et al., 2012). U.S. organizations lose $60 billion annually because of poor software quality (Harter et al., 2012). Understanding SQA strategies and their applicability become significant to the participating organizations to improve their business practice (Singh & Kannojia, 2013).

Implications for Social Change

This study may contribute to positive social change by strengthening the SQA knowledge of QA leaders and making these leaders better practitioners in their industry of software development (Majstorovic & Sibalija, 2015). If QA leaders become more

mindful regarding SQA management, then that may lead to reducing the cost of software development, increasing efficiency, and helping to make organizations more profitable and competitive (Singh & Kannojia, 2013). Efficient profitability allows for the growth of the organizations and contributes to the organizations' ability to stay in business (Masa'deh, 2012), thus contributing to social change through strengthening organizations' local economies. Small organizations employ half of all U.S. workers, making the sustained longevity of small organizations important to larger social communities (Taneja et al., 2016). Small organizations contribute to economic prosperity and provide employment opportunities. The improved SQA processes of small software development organizations can contribute to positive social change through the sustained longevity of the organizations.

A Review of the Professional and Academic Literature

I used the Walden University Library and Google Scholar academic databases to retrieve the literature for this study. The academic databases from the Walden University Library included Business Source Complete, ABI / INFORM Complete, Emerald Management, SAGE Premier, ACM Digital Library, IEEE Xplore Digital Library, ProQuest Central, and ScienceDirect. The keywords used when searching were *CMMI, CMMI-DEV, ISO 9001, total quality management, software quality assurance, Agile, Six Sigma,* and *technical debt*. The filters I applied to the search results were peer-reviewed articles and dissertations from 2012 and newer. Once retrieved, I gathered and organized the literature in Zotero software. The total number of articles contained in the literature

review is 87, of which 98% are peer reviewed, and 94% are within 5 years of expected Walden University Chief Academic Officer approval.

The central research question was: What strategies can QA leaders of small software development organizations use for their SQA processes? In this section, I will discuss topics that cover industry standards that start with a more holistic system approach to quality and narrows down towards a standard metric used to monitor software quality. The subsections include (a) SQA–the subject of software quality best practices, (b) TQM–a discussion on the holistic system thinking of quality selected as the conceptual framework, (c) statistical process control–the industry best practice for measuring and monitoring SQA, and (d) continuous improvement in SQA.

Software Quality Assurance (SQA)

The increasing integration of information technology (IT) into people's lives makes the correct execution of software important (Leveson, 2004). Customers demanding better quality force organizations to improve product quality (Al-Dhaafri & Al-Swidi, 2016). Software failures can range from minor inconvenience to catastrophic loss of finances or life (Harter et al., 2012). Examples of software failure consequences include automotive recalls for over 68,000 vehicles, energy blackouts affecting millions of lives, and failed aeronautical satellite launches (Harter et al., 2012; Leveson, 2004). U.S. organizations lose $60 billion annually because of poor software quality, of which $21.2 billion is directly realized by software development organizations (Harter et al., 2012). In comparison, the U.S. software industry was worth $425 billion in 2012 (Shapiro, 2014).

The quality of the system should be organizations' focus instead of a product's quality (Ilkay & Aslan, 2012). SQA must include varying standards, plans, management systems, policies, and procedures because none of the industry models covers all the perspectives on software quality improvement (Masa'deh, 2012). Quality frameworks exist as industry best practices established by the IIST body of knowledge such as TQM, CMMI, ISO 9001, Six Sigma, and Agile that QA leaders can leverage for SQA (Baldassarre et al., 2012; A. Brown, 2013; "Certified Test Manager", 2015; Lin et al., 2012; Morris, 2012; Pillai et al., 2012; Rafique & Mišić, 2013). Hinojo (2014) contended that industry best practices for SQA are CMMI and Capability Maturity Model Integration for Development (CMMI-DEV), ISO 9001 is the industry best practice for quality management measurement, and project frameworks, such as Agile, exist to execute the SQA frameworks.

Managing SQA is important to QA leaders for many reasons. Omar and Murgan (2014) highlighted customer loyalty as a benefit to having high quality products and services. The cost of integrating bug fixes or new features is more expensive after launching an application than an early detection in the software development lifecycle (Nord et al., 2012). Heger, Happe, and Farahbod (2013) also identified the increased cost of fixing bugs the later the bug discoveries occurred and supported the findings of Nord et al. (2012) when discussing the importance of early bug detection. Heger et al. explained that increased complexity, code, and development experience is required to correct bugs later in the software development lifecycle. Quality software can be a competitive advantage directly related to the processes that development teams use (De

Castro, Braga, & Soares, 2013). Singh and Kannojia (2013) noted the benefits of SQA include increasing profitability. Singh and Kannojia also explained that QA leaders of software development organizations benefited from a competitive edge in their industry by lowering costs and reducing time to market.

Several researchers defined SQA within the context of TQM. Morris (2012) defined SQA as a subset of the holistic system approach of TQM. A. Brown (2013) identified the system's holistic view of TQM contains industry best practices frameworks for quality management, which aligned with Morris. Galinac Grbac et al. (2013) identified that SQA is a continuous improvement process throughout the software development lifecycle. According to Masa'deh (2012), organizations achieved SQA by measuring applicable process descriptions, standards, and procedures against industry best practices; identifying and documenting noncompliance issues; providing feedback to work group staff and managers on the results of QA activities, and addressing noncompliance issues. Masa'deh supported other researchers when identifying the important practices of SQA contained statistical process control and continuous improvement through release planning, RCA, testing techniques, and requirements tracing (Cotroneo, Pietrantuono, & Russo, 2013; Dalal & Chhillar, 2013; Ghabi & Egyed, 2012).

QA leaders face the challenge of selecting appropriate SQA processes. Masa'deh (2012) argued that organizations have to choose the right approach to SQA based on their specific business needs. Leopoulos and Chatzistelios (2014) agreed with Masa'deh and suggested QA leaders must implement SQA best practices in a cost-effective way to

realize the benefits. Vierhauser, Rabiser, and Grünbacher (2014) noted very large software systems (VLSS) required a different level of SQA than smaller software applications do. Burli et al. (2012) argued that TQM contains no standard framework, giving QA leaders the freedom to leverage a combination of quality frameworks that best meet the organizations' needs.

In summary, SQA manages the quality of the software product (Morris, 2012). The important practices of SQA are statistical process control and continuous improvement through release planning, RCA, testing techniques, and requirements tracing (Masa'deh, 2012). When used as a continuous improvement process, SQA is aligned within TQM (Galinac Grbac et al., 2013). Statistical process control also aligns SQA within TQM (Morris, 2012). The freedom of QA leaders to leverage a combination of quality frameworks that best meet the organizations' needs (Burli et al., 2012) led to the purpose of the research of this study.

Total Quality Management (TQM)

Deming originated the TQM theory in 1980 to improve process control and resource management within the manufacturing industry (Petersen, 1999). The U.S. government adopted TQM in military practice with the goal of reducing costs through increased efficiency (Petersen, 1999). TQM is a process of thinking about the organization as a whole to achieve excellence (Burli et al., 2012). Majstorovic and Sibalija (2015) measured TQM usage as high as 92% in organizations based on one or more certified systems. Al-Dhaafri and Al-Swidi (2016) noted organizations required a management framework such as TQM to help improve overall quality. However, Wang

(2014) demonstrated that QA leaders who invest in TQM see a positive effect on operational performance until reaching a certain investment threshold, after which the TQM investment became sunk-cost.

TQM has eight to 16 principles, depending on the researcher and industry (Talib, Rahman, & Qureshi, 2011). According to Talib et al. (2011), the most common principles in TQM frameworks were top management commitment, customer focus, training and education, continuous improvement and innovation, supplier management, and employee involvement. Burli et al. (2012) studied TQM in ISO certified Indian organizations and determined that leadership and customer focus are the most important principles of TQM. Malik and Blumenfeld (2012) and Maskara (2014) agreed with Burli et al. and included additional TQM principles of continuous improvement, treating the organization as a total system, and using statistical process control as the most important principles. Al-Dhaafri and Al-Swidi (2016) added to TQM research when studying TQM impact on organizational performance and identified the alignment of TQM with customers' needs as an important principle of TQM. Shokri, Waring, and Nabhani (2016) also noted that leadership is a key factor in the implementation of TQM frameworks within organizations.

Previous research linked TQM to improved operational performance and financial benefits. Ahmad, Zakuan, Jusoh, Ariff, and Takala (2013) demonstrated the strong positive impact TQM had on operational performance when studying the relationship between TQM and operational performance. Bu, Liu, and Peng (2013) also discovered the positive financial impact TQM had on organizations. Bu and Cao (2015) furthered the

research of Bu et al. (2013) and discussed how QA leaders that implemented TQM helped lead organizations towards increased profits through improved customer satisfaction, reduced operating costs, and improved return on total assets. The financial benefits TQM can offer organizations makes TQM an important consideration for QA leaders.

The implementation of TQM can be challenging for QA leaders. The relationship between policy and strategy act as references for all factors of TQM, and must be in place (Burli et al., 2012). QA leaders should seek to higher employees with personal high quality standards to build an effective TQM framework (Shokri et al., 2016). Majstorovic and Sibalija (2015) contended that moving towards TQM requires adhering to standardized management systems and using quantitative analysis of the systems to monitor efficiency and defects. Majstorovic and Sibalija's findings supported Talib et al. (2011), Burli et al. (2012), Malik and Blumenfeld (2012), and Maskara (2014). Lin et al. (2012) studied the relationship between management information systems (MIS) and TQM to determine that organizations moved through five stages attempting to implement a TQM framework, similar to maturity levels found in CMMI. Each stage discovered by Lin et al. can guide QA leaders through successful TQM implementation.

Wang (2014) noted how TQM could provide a source of competitive advantage for QA leaders who implement the framework. Additionally, Wang demonstrated the need for organizations to invest simultaneously in innovation and TQM as a source of competitive advantage. Omar and Murgan (2014) also discussed TQM as a source of competitive advantage, supporting Wang. Al-Dhaafri and Al-Swidi (2016) further

supported TQM as a source of competitive advantage when implemented in organizations.

In summary, the common themes found in previous TQM research are continuous improvement, leadership, customer focus, treating the organization as a total system, and using statistical process control (Burli et al., 2012; Dhaafri & Al-Swidi, 2016; Malik & Blumenfeld, 2012; Maskara, 2014; Majstorovic & Sibalija, 2015; Talib et al., 2011). These common TQM themes align with the themes of continuous improvement and statistical process control highlighted through previous research on SQA. Leadership is a key factor in the successful implementation of TQM frameworks within organizations (Shokri et al., 2016). QA leaders who incorporated the important TQM practices into SQA may lead their organizations towards achieving TQM (Majstorovic & Sibalija, 2015) and positive organizational performance and innovation (Wang, 2014).

Statistical Process Control

Organizations need to measure all aspects of software development to execute quality monitoring (Masa'deh, 2012). Azizi (2015) recommended statistical process control as an important TQM practice. The concept of a sustainable monitoring framework is important to SQA because a sustainable monitoring framework gives a view of how QA leaders monitor the quality of the software (Lami, Fabbrini, & Fusani, 2013). A sustainable monitoring framework also provides developers an opportunity to correct quality earlier in the software lifecycle (Lami et al., 2013).

Data-driven decision making strengthens quality and process management (Raza & Faria, 2014). The difficulty of gathering enormous amounts of data to sift through and

utilize in a meaningful way complicates performing data measuring and monitoring during the software development lifecycle (Bijlsma, Correia, & Visser, 2012). QA leaders must understand customer usage metrics to make better decisions regarding the prioritization of bug fixing against new feature releases (Khalane & Tanner, 2013). QA leaders only realize accurate performance metrics after customers use the software (Vierhauser et al., 2014). Waiting for customers to provide feedback on defects is too late in the software development lifecycle (Heger et al., 2013).

Defining metrics. Defining the metrics to monitor SQA varied greatly within previous research, providing QA leaders with several options to choose from. Baggen, Correia, Schill, and Visser (2012) decomposed quality within SQA into metrics of volume, redundancy, unit size, complexity, unit interface size, and coupling. In contrast, Song and Kim (2012) found 15 metrics for measuring software quality included test coverage, requirements coverage, software inspections, and software error density. From a different view, Raza and Faria (2014) defined the process quality index as the measurement of defect density using ratios of the design time to coding time, design review time to design time, code review time to coding time, complete defects to a size measure, and unit test defects to a size measure.

Traditional project managers measure a project's triple constraints of scope, schedule, and budget using the best tools and techniques available (McCann, 2013). Wright (2013) found that earned value measurement, cost-benefit analysis, critical path method, activity on arrow, activity on node, Gantt charts, and work breakdown structures are methods that project managers can use to monitor the triple constraints, especially

project costs. Wright noted that the triple constraints inaccurately measured a project's success. Williams (2013) agreed with Wright and determined the cost of a project does not have an effect on the measured success rate of a project, nor did the length of the project. Cost measurements included only the cost to build the software, not the costs of quality (Wright, 2013).

Wang (2014) warned of the increased costs to implement and maintain TQM within organizations. Measuring the financial metrics of SQA included organizations' payroll, purchasing/invoicing, payments, collection, financial statements, and cost analysis (Sedevich-Fons, 2014). Omar and Murgan (2014) demonstrated that QA leaders could measure the costs of quality through cost of control and cost of failure of control using a simulation approach. However, intangible costs made quality cost measurement an inexact science that required organizations to define specific metrics for cost measurements (Omar & Murgan, 2014).

Technical debt is a measurable dollar value directly related to SQA (Seaman et al., 2012). Nord et al. (2012) noted that organizations failed to budget technical debt costs resulting in spending more money than expected to build applications. Seaman et al. (2012) suggested the technical debt overruns were affecting organizations' profitability. Curtis et al. (2012) conducted a quantitative study of 160 organizations using a set rule of architectural coding best practices that measured the quality of the applications. Setting a labor rate of $75/hour, the average cost of technical debt across all programming languages analyzed was $3.61 per line of code using the following formula (Curtis et al., 2012):

∑ High severity violations x .5 x $75 +

∑ Medium severity violations x .25 x $75 +

∑ Low severity violations x .1 x $75

Seaman et al. (2012) discussed the causes of technical debt were development teams putting aside a feature for implementing another feature, having a lack of time to implement the feature, unclear requirements, and changing requirements. Allman (2012) found another contributing factor to increased technical debt within organizations was the developer creating technical debt not having to be the person who paid the technical debt. Allman also identified changing requirements as a contributing factor to technical debt. Nord et al. (2012) discussed that application developers try to code quickly, thereby placing code refactoring, testing, documentation, architecture issues, and known defects as examples of technical debt aside to meet their deadlines. Codabux and Williams (2013) classified several types of technical debt existed including testing debt, design debt, and defect debt.

Siebra et al. (2012) suggested not all technical debt was detrimental if QA leaders strategically accepted technical debt, benefiting organizations' attempts to be first to market with a product. From a different view, Codabux and Williams (2013) identified the implications of technical debt can be large cost overruns, quality issues, or the complication of feature integration because of breaking existing features. Therefore, if not managed properly, then technical debt may cause substantial future issues for development teams (Codabux & Williams, 2013). However, the formula established by Nord et al. (2012) provided a tool for QA leaders to measure technical debt.

Six Sigma. Six Sigma is a process improvement methodology for the elimination of waste through statistical measures and controls (Pillai et al., 2012). Six Sigma is a set of iterative processes that includes define, measure, analyze, improve, and control for existing product improvement (Kastelic & Peer, 2012). The Six Sigma process for new product development is define, measure, analyze, design, and verify (Kastelic & Peer, 2012). The measure and analyze processes of Six Sigma include heavy statistical control over the processes, making Six Sigma an excellent fit with CMMI and ISO 9001 (Majstorovic & Sibalija, 2015).

Tlapa, Limon, García-Alcaraz, Baez, and Sánchez (2016) discovered successful implementation of Six Sigma required top management support, an implementation strategy, and a collaborative team. Pillai et al. (2012) identified the benefits of Six Sigma as improved process control and enhanced financial performance through improved quality. Kastelic and Peer (2012) discussed benefits of reduced development cycle time, decreased defect density, improved customer satisfaction, improved overall software quality, improved business to IT integration, improved process efficiencies, minimized costs, improved focus on items critical to quality, and improved data analysis. Malik and Blumenfeld (2012) noted that Six Sigma is not difficult for organizations to implement, although certified practitioners (Master Black Belts, Black Belts, and Green Belts) should be used to implement the processes.

Lean is an organizational process improvement framework that fits within TQM (Shokri et al., 2016; Nicholas, 2016). Lean adopts the Japanese word *kaizen* translated as continuous improvement as the main principle (Drohomeretski, Gouvea da Costa,

Pinheiro de Lima, & Garbuio, 2014). Lean encourages QA leaders to focus on kaizen through cost, process time, value and efficiency, whereas Six Sigma encourages sustainable and incremental process improvement (Shokri et al., 2016). Lean combined with Six Sigma to become Lean Six Sigma (LSS) by QA leaders, because of the benefits each framework contributes to the combination (Drohomeretski et al., 2014). LSS aligns with continuous improvement models because of the incremental process improvements (Shokri et al., 2016).

Despite the benefits lean can bring to organizations including competitiveness (Drohomeretski et al., 2014), internal resistance and the availability of resources are two leading factors that cause LSS implementation to fail within organizations (Shokri et al., 2016). To aid lean adoption, Nicholas (2016) established a lean assessment tool QA leaders can use to determine the organizational readiness for the adoption of lean principles. Nicholas used over 50 metrics within the assessment tool including time effectiveness, quality, process, cost, and human resources (HR).

Capability Maturity Model Integration (CMMI). CMMI is a tool that organizations can use to perform SQA with the goal of achieving a TQM system (Hinojo, 2014). The U.S. Department of Defense established CMMI in 1991 to manage the quality of software suppliers (David, 2013). The Software Engineering Institute (SEI) assumed the responsibility of the CMMI soon after that (David, 2013). In 1997, SEI branched the focus areas of the CMMI to include definitions specific to software development known as the CMMI-DEV (David, 2013). SEI evolved CMMI-DEV with different versions, the most recent being CMMI-DEV v1.3 (2010) (David, 2013).

The CMMI-DEV is a set of 22 QA process areas organized into categories called maturity levels (ML;SEI, 2010). Each process area has specific or generic measurable goals. The goal measurements are incomplete (0), performed (1), managed (2), and defined (3) (David, 2013). A process reaches a maturity level when an organization meets all of the specific and generic goals. The five CMMI-DEV ML are (a) ML1 initiated, (b) ML2 repeatable, (c) ML3 defined, (d) ML4 managed, and (e) ML5 optimized (David, 2013).

ML1 is the beginning stage when organizations set out to achieve higher levels (SEI, 2010). ML2 contains process areas for configuration management, measurement and analysis, project monitoring and control, project planning, process and product QA, requirements management, and supplier agreement management (David, 2013; Hinojo, 2014; SEI, 2010). ML3 contains all of the ML2 process areas plus decision analysis and resolution, integrated project management, organizational process definition, organizational process focus, organizational training, product integration, requirements development, risk management, technical solution, validation, and verification (David, 2013; Hinojo, 2014; SEI, 2010).

ML4 contains all of the ML2 and ML3 process areas plus organizational process performance, and quantitative project management (David, 2013; Hinojo, 2014; SEI, 2010). ML5 contains all of the ML2, ML3, and ML4 process areas plus causal analysis and resolution, and organizational performance management (David, 2013; Hinojo, 2014; SEI, 2010). High maturity organizations achieve ML4 or ML5 (David, 2013; Hinojo,

2014; SEI, 2010). An organization appraised at ML5 demonstrated tight control of their processes (Bass, Allison, & Banerjee, 2013).

McKnight (2013) discussed that achieving a CMMI rating requires time, resources, and the analysis of process and product data. McKnight also identified that senior management involvement is critical to compliance. Bass et al. (2013) noted that organizations needed to be aware of the increased costs from the extra governance required in a CMMI environment. Tuan and Thang (2013) contended that higher maturity leads to higher quality. Falessi, Shaw, and Mullen (2014) identified many U.S. organizations asked for CMMI maturity level ratings from their consultants.

Organizations can achieve an ML through a process of assigning a dedicated internal resource, establishing the stability of key leaders, encouraging an organizational culture that fuels change, rigorously selecting the lead appraiser, leveraging external expertise, and using small iterations (Falessi et al., 2014). Organizations use the CMMI and ISO 9001 frameworks to guide what needs to be architected into SQA processes, whereas organizations use the Six Sigma framework to execute SQA processes (Baldassarre et al., 2012). David (2013) contended the CMMI had the process areas defined but did not instruct QA leaders how to achieve the process areas. David also identified that QA leaders must understand and define what each of the CMMI process areas means, which further supported the purpose of this study.

The majority of organizations seeking CMMI accreditation achieve their highest level at ML2 or ML3 (Swinarski, Parente, & Kishore, 2012). Within the 5,500 rated CMMI organizations worldwide, only 344 achieved Level 5, and only 3% of the 344

were small organizations (Falessi et al., 2014). Müller and Nielsen (2013) argued that more formalized, internally focused, goal oriented, and results driven organizations used the CMMI processes. Falessi et al. (2014) noted that achieving and maintaining a high maturity level requires considerable effort. Shih, Shaw, Fu, and Cheng (2013) argued that adopting the CMMI is a significant undertaking requiring complicated change management, another reason organizations fail to adopt CMMI.

Almost 80% of U.S. IT jobs are within small organizations, and global organizations spend billions of dollars on CMMI process improvements (Swinarski et al., 2012). Overcoming the perception that industry processes such as CMMI-DEV is too bureaucratic and prohibits flexibility was the challenge for small organizations (Swinarski et al., 2012). Mora, O'Connor, Raisinghani, and Gelman (2013) found that financial and people resources to implement the industry process are other challenges to small organizations. From a different view, Mora et al. discussed how QA leaders overcome the implementation challenges with a CMMI-DEV knowledge management system that simplified the process area definitions and reduced the dependence on consultants.

Using CMMI metrics such as walkthroughs, inspections, reviews, and audits through verification and validation practices within ML3 accomplished SQA (Sharma & Vishwakarma, 2012). Defect tracking including severity within the verification and validation processes was important in SQA (Sahin, Kaynak, & Sencan, 2013). Bug fixes are less expensive if identified in the starting phases of development rather than later

testing phases, making requirements management a key component of CMMI (Sharma & Vishwakarma, 2012; De Vasconcelos, de la Vara, Sanchez, & Pastor, 2012).

Improved software quality is the main benefit to organizations through the utilization of SQA models such as CMMI (Mora et al., 2013). Organizations achieving the CMMI Maturity Level 3 rating experienced a reduction in the number of bugs per requirement, allowing the development teams to release more functionality to customers and increase profits per employee (Falessi et al., 2014). A reduced defect density (number of defects delivered in the software divided by the size of the software) is also a benefit to implementing CMMI-DEV (Shah, Morisio, & Torchiano, 2012).

ISO 9001. ISO 9001 series is a tool similar to the CMMI that organizations can use to achieve SQA with the goal of performing in a TQM system (Ilkay & Aslan, 2012). Wang (2014) highlighted the effect ISO 9001 had on operational performance. The ISO 9001 series is a collection of quality management best practices that organizations user to help establish quality control systems (Ilkay & Aslan, 2012). QA leaders have successfully leveraged ISO 9001 despite ISO 9001 not being a tool specific to the software development industry (Baldassarre et al., 2012). ISO 9001 contains process definitions for quality management system, management responsibility, resource management, product realization, and measurement, analysis, and improvement (Baldassarre et al., 2012). Organizations demonstrate compliance with the process areas through an audit by an accredited ISO auditor. The ISO 9001 series contains process areas of customer focus, leadership, the involvement of people, process management,

system approach to management, continuous improvement, factual approach to decision making, and mutually beneficial supplier relationships (O'Mahony & Garavan, 2012).

Baldassarre et al. (2012) demonstrated the process areas that aligned ISO 9001 with TQM were customer focus, leadership, process management, system approach to management, continuous improvement, measurement, and analysis. Baldassarre et al. further mapped CMMI process areas to ISO 9001 process areas, demonstrating that organizations were seeking CMMI appraisal by obtaining an ISO 9001 certification first because ISO 9001 certification was a larger undertaking than CMMI accreditation. Ilkay and Aslan (2012) identified financial support and resource availability were reasons for the difficulty of small and medium-sized enterprises (SMEs) to undergo the 1 to 2 year certification process for ISO 9001, similar to the results from Mora et al. (2013) regarding CMMI accreditation. However, the quality management of ISO 9001 certified organizations improved from non-certified organizations (Ilkay & Aslan, 2012). Wang (2014) discussed TQM frameworks implemented within organizations provided an advantage in innovation and the adoption of more structured processes such as ISO certifications.

Organizations combine multiple SQA frameworks to achieve the benefits of all frameworks (Baldassarre et al., 2012). Many of the industry quality frameworks have overlapping processes that can lead to inefficiencies and redundancies if organizations are not strategic during implementation (Baldassarre et al., 2012). Organizations can gravitate towards using an industry standard SQA, because of the discussion heard regarding the benefits of using one (Müller & Nielsen, 2013). Wang (2014) noted the

positive relationships between quality certification and organizational performance, customer satisfaction, and innovation.

However, not all SQA frameworks are a good match for an organization, and an organization's culture plays a part in the selection process (Müller & Nielsen, 2013). Using an SQA framework verbatim might mean the failure of the SQA framework because of the incompatibility of the strict standards and a loose or innovative culture (Müller & Nielsen, 2013). QA leaders should not expect quality improvements purely by implementing an SQA framework. Quality improvements come from everyone following the SQA framework (Ilkay & Aslan, 2012). Wang (2014) further discussed how the implementation of a strict process framework such as ISO 9000 series might negatively affect organizational innovation, yet argued the need for QA leaders to be mindful of the strong non-linear relationship between innovation and quality. An organization needs to be able to see how to adjust the SQA to suit their needs (Müller & Nielsen, 2013), which further supports the purpose of this study.

Continuous Improvement

Various industries use the concept of continuous improvement to drive operational efficiency and improve the quality of products and services (Schmidt, Elezi, Tommelein, & Lindemann, 2014). QA leaders often exercise continuous improvement through a plan-do-check-act (PDCA) cycle (Schmidt et al., 2014). The PDCA cycle originated by Deming as a part of the TQM work for Motorola in 1980 (Petersen, 1999). The PDCA cycle is called Kaizen in Lean Six Sigma (Schmidt et al., 2014). Abdulmouti (2015) demonstrated the operational improvements of an organization using Kaizen

through the reduction of labor hours, output increases, inventory reduction, and the reduction of required capital.

Schmidt et al. (2014) identified continuous improvement as a fundamental piece in TQM. According to Schmidt et al. continuous improvement should involve organizations as a whole, a similar concept to TQM as identified by Burli et al. (2012). Abdulmouti (2015) noted that business leaders should involve all employees in the PDCA cycle. The continuous cycle of PDCA allows organizations constantly to improve quality across the system (Schmidt et al., 2014).

As previously discussed by Galinac Grbac et al. (2013), SQA is a continuous improvement process throughout the software development lifecycle. Masa'deh (2012) identified the important practices of SQA were statistical process control and continuous improvement through release planning, RCA, testing techniques, and requirements tracing. Khalane and Tanner (2013) discovered continuous improvement was a strategy that development teams used to address the concerns of the stakeholders. De Castro et al. (2013), Hinojo (2014), and Singh and Kannojia (2013) all identified continuous improvement as an important strategy in SQA. Hinojo further identified Agile project management as a framework best suited for continuous improvement initiatives.

Agile methodology. The trend for IT project management since 2001 has been to move away from traditional methodologies towards the Agile methodology (Williams, 2013). In the Agile methodology, project managers only plan the project in short intervals of usually two weeks at a time, engaging customers to trial any features built within that time (Pedersen, 2013). This iterative practice encourages customers to communicate

changes early and often to allow development teams to be more responsive to those changes (Pedersen, 2013; Wolfe, 2013).

The more traditional Waterfall framework to software development encourages the completion of all project planning at the start of the project, then the development of the application occurs, followed by customer deployment (Pedersen, 2013). Waterfall does not allow for the easy integration of changes requested by the customer because all development activities take place at the same stage (Pedersen, 2013). Project managers use Agile practices and hybrid methods of project management rather than traditional top-down approaches to realize more successful project execution (Williams, 2013). Agile practices such as test-driven development produce better quality software (Rafique & Mišić, 2013). Regardless of the methodology, the project manager should include documentation, team reviews, and product demonstrations with key stakeholders, short release cycles, and regular team communication (Wolfe, 2013).

A project management methodology focused on productivity as one of its values would help to lower the cost of software development, thereby increasing quality and lowering the cost of technical debt (Seaman et al., 2012). Flumerfelt, Anna, and Kahlen (2012) discussed the benefits of Agile practices included increased productivity, which leads to reduced costs. M. E. Brown (2013) suggested the pursuit of increased productivity is another decision development teams considered when selecting Agile practices. M. E. Brown also noted that improved productivity decreased the cost of software development.

Agile processes for managing change make the Agile methodology a more attractive project management methodology than the more traditional Waterfall approach (Allman, 2012). The decision to use Agile practices over Waterfall for the change management gains help to alleviate the accrual of technical debt in Waterfall, but does not eliminate technical debt (Allman, 2012). One of the attractive aspects of using Agile practices is the speed of application development and the frequent product releases, making Agile practices attractive for teams with tight deadlines (Nord et al., 2012). However, development speed is a direct contributor to accumulated technical debt by rewarding teams for working software and not long term maintainability (Allman, 2012). Nord et al. (2012) also identified the speed of development as a contributing factor to technical debt.

Wolfe (2013) discussed choosing the right software development methodology was an important decision that can lead to project success, supporting Burli et al. (2012) who argued regarding the freedom of QA leaders to leverage a combination of quality frameworks that best meet the organizations' needs. Young (2013) also discussed organizations needed to understand the benefits of using Agile practices. Wolfe noted that project managers simply selecting Agile practices for every project would be a mistake that can lead to project failure. Young identified teams selecting Agile practices most likely had a high level of technical expertise and had previous Agile experience. Young argued that the benefits one team realized by using Agile practices may be different from what another team achieves, helping to support the purpose of this study.

Release planning. Release planning evolution from the mid-1990s caused organizations' SQA complexity to increase (Roche, 2013). In the 1990s, organizations released software on CDs, meaning organizations had one chance to get the software right, and therefore focused on QA as a more integrated process within the software development lifecycle (Roche, 2013). The introduction of the Internet and cloud computing equipped organizations with new tools to allow for release planning (Jeffery, 2012). The Internet allowed organizations to relax on SQA, instead focusing on releasing new features and relying on software patches to correct poor quality (Roche, 2013). Between the years 2010 and 2015 organizations uncoupled their release planning of software from customers' expectations, meaning customers always had the latest release of the software via a continuous release synchronization through the Internet (Khalane & Tanner, 2013).

Cloud computing is on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service (Jeffery, 2012). Software development organizations began a cloud computing release planning strategy by 2010 (Jeffery, 2012). Jeffery (2012) noted that the software release planning method of cloud computing by organizations introduced SQA issues to consumers. Planning of infrastructure to meet the global growth of data and processing capacity challenged IT architects, making SQA a high priority (Jeffery, 2012). Cloud computing caused new challenges to software development organizations of elastic scalability, security and trust, manageability, and usability (Jeffery, 2012). Nord et al. (2012) argued that if the decision-making process of

release planning included technical debt management, organizations would realize a reduction in the total cost of application development and improve software quality.

Release planning by QA leaders requires the inclusion of customer satisfaction and time to market (Agarwal, Karimpour, & Ruhe, 2014). Agarwal et al. (2014) recommended theme-based release planning as the optimal method of delivering value and improving customer satisfaction. Agarwal et al. discussed how theme-based release planning helped QA leaders with the decision-making process of what features to include in a release by categorizing the features into common themes instead of individuals. Theme-based release planning supports the Agile practice of goal completion application development Pedersen (2013) discussed. In contrast, QA leaders integrating release planning with third parties experience a higher level of complexity than QA leaders that controlled the release (Naciri, Idrissi, & Kerzazi, 2015). Third party maintenance requires QA leaders to align release planning with third party scheduling, removing control over the release planning (Naciri et al., 2015).

Root-cause analysis (RCA). RCA is an investigation into an issue with the goal of understanding what caused the issue so preventative actions can occur (Dalal & Chhillar, 2013). Industry standard RCA involves data collection, RCA format and template design, RCA tool/method being applied, identifying events of variance, brainstorming and discovery among team members, defining corrective and preventative actions, developing and then implementing the action plan, conducting post-implementation analysis, and updating a knowledge repository (Dalal & Chhillar, 2013). Galinac Grbac et al. (2013) used the Pareto principle to demonstrate 20% of the

application caused 80% of the defects, making product knowledge important to RCA. Masa'deh (2012) identified statistical process control through measurement can help QA leaders in the RCA process.

Teams that understand and follow the RCA will be more effective at correcting defects and producing better quality (Dalal & Chhillar, 2013). Nieminen and Räty (2015) discussed model-based testing as a testing technique to aid QA leaders in automated testing when performing RCA. Heger et al. (2013) also recommended RCA through automated testing to be a best practice throughout the software development lifecycle and noted the different methods available to QA leaders to execute RCA existed. The increasing complexity of applications and data directly affects QA leaders' ability to execute RCA (Nieminen & Räty, 2015).

Testing techniques. The software testing techniques used by organizations have a direct impact on the quality of software (Cotroneo et al., 2013). Heger et al. (2013) recommended the use of testing techniques through both regression and unit testing. Heger et al. also recommended Software Performance Engineering as a technique to guide high quality software. Machado, McGregor, Cerqueira Cavalcantic, and Santana de Almeida (2014) recommended the creation of strategic use cases to help QA leaders perform testing. Zein, Salleh, and Grundy (2016) identified the testing techniques QA leaders used depends on the type of application being created such as mobile, web, or desktop.

Selecting the right techniques to use in a given situation can be troublesome. Testing is commonly based on development teams' knowledge of the software instead of

the best techniques that will result in more stringent testing metrics that capture more defects (Cotroneo et al., 2013). Machado et al. (2014) recognized the difficulty in testing every application feature because of the complexity of inputs and features. QA leaders who use the same techniques repeatedly based on past success will not be adaptive enough to the ever-changing complex requirements within software development (Cotroneo et al., 2013). Nieminen and Räty (2015) recommended model-based testing as an accurate testing technique for QA leaders.

Zein et al.(2016) contended that successful mobile application testing techniques required the inclusion of early testing requirements, development environments matching production environments, and security and usability testing. Machado et al. (2014) recommended reducing testing requirements through feature integration coverage and performing testing and two key testing techniques. Bae, Rothermel, and Bae (2014) recommended eliminating non-executable functionality from Graphical User Interface (GUI) testing to strengthen the effectiveness and reduce costs.

Bae et al. (2014) noted that manual testing is costly to execute, and therefore QA leaders should find opportunities to implement automated testing through model-based or dynamic GUI testing. Heger et al. (2013) established an automated testing technique and recommended the use of the technique throughout the software development lifecycle. Nieminen & Räty (2015) discussed the limitation of automated testing, because of the differing functionality of the testing tools QA leaders used. Huang, Peng, and Huang (2012) discussed how QA leaders' prioritization of test cases based historically on sequence should be restructured during regression testing to account for test cases failure

history, allowing for QA leaders to assign a higher priority to historically unsuccessful cases to reduce testing costs. From a different view, Cotroneo et al. (2013) argued that QA leaders need to mix and match testing strategies of singularly, random criteria, and full testing at each criterion to find the strategies that will ultimately result in the best defect detection. The recommendation of Cotroneo et al. to mix and match testing strategies added to the purpose of this study.

Requirements tracing. Proper requirements tracing plays an important role in SQA (Ghabi & Egyed, 2012). The CMMI maturity level 3 requires requirements tracing (SEI, 2010). Masa'deh (2012) also recommended requirements tracing as a part of organizations' continuous improvement. Developers who include non-functional requirements for accessibility and usability early in the software development lifecycle help to improve the functionality and quality of the software (Dias, Pontin de Mattos Fortes, & Masiero, 2012). Ghabi and Egyed (2012) identified that requirements tracing helped reduce software maintenance costs by educating the developer who needs to correct a defect, therefore reducing effort. Ghabi and Egyed further noted that requirements tracing also pinpointed where the defect occurred, thus reducing the possibility of the developer implementing defects into the system designed to correct the original defect.

Transition

QA leaders want to meet user expectations (Khalane & Tanner, 2013). The important practices of SQA are statistical process control and continuous improvement through release planning, RCA, testing techniques, and requirements tracing (Masa'deh,

2012). SQA as a continuous improvement process aligns SQA within TQM (Galinac Grbac et al., 2013). The common themes found in previous TQM research are continuous improvement, leadership, customer focus, treating the organization as a total system, and using statistical process control (Burli et al., 2012; Malik & Blumenfeld, 2012; Maskara, 2014; Majstorovic & Sibalija, 2015; Talib et al., 2011).

QA leaders used TQM practices in SQA through processes of continuous improvement, statistical process control, release planning, RCA, testing techniques, and requirements tracing (Masa'deh, 2012). CMMI, ISO 9001, Agile project management, Six Sigma, and technical debt calculations all include SQA processes for QA leaders (Baldassarre et al., 2012; A. Brown, 2013; "Certified Test Manager", 2015; Hinojo, 2014; Lin et al., 2012; Morris, 2012; Pillai et al., 2012; Rafique & Mišić, 2013). TQM investment requires close measurement to understand the optimal investment level for organizations (Wang, 2014).

Small organizations can struggle with adopting large industry standards because of a lack of resources or financial capital to do so (De Castro et al., 2013). Additional adoption challenges include selecting the wrong framework, incorrectly implementing the framework, or the organization being a poor environment for a framework (Mosadeghrad, 2014). However, SME benefit from having these strategies and need to find a way to adopt industry standards (De Castro et al., 2013). Organizations need a low cost and time to implement TQM system (Leopoulos & Chatzistelios, 2014). Adding to the complexity of adopting a TQM system are the new challenges caused by cloud

computing such as elastic scalability, security and trust, manageability, and usability (Jeffery, 2012).

Adhering to standards and being innovative are contradictions to each other, making TQM adoption challenging for QA leaders (Fried, Gey, Pretorius, & Günther, 2013). Communication between the development stakeholders of customers, the organization, and the development team is a critical success factor for both producing software of high quality and the adoption of SQA processes (Robillard & Lavallée, 2012). A common taxonomy between the stakeholders will help improve communication and help to focus requirements (Robillard & Lavallée, 2012).

The research in this study included gaps for QA leaders trying to implement SQA strategies, which added to the purpose of this study. This study addressed the need for QA leaders assessing their SQA strategies. Section 1 included the problem statement, purpose of research, research problem, the nature of the study, and the literature review. The literature review contained the analysis of TQM research and the application of SQA best practices. Section 2 will contain the purpose statement, the role of researcher, details of participants, research methods and design, population and sampling, data collections, data analysis, and the data validation techniques. Section 3 will contain the results of the study.

Section 2: The Project

Section 2 contains a reintroduction of the purpose of the study. In Section 2, I will also describe the role of the researcher, the participants, the research method, the research design, the population and sampling, the ethical research considerations, the data collection instruments, the data collection technique, the data organization techniques, the data analysis, and the reliability and validity. Section 3 will include a discussion of the results.

Purpose Statement

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The target population group was the QA leaders of the organizations. The geographic location was Saint John, New Brunswick, Canada. This study's implications for positive social change include the potential to provide QA leaders with the strategies to improve SQA processes, thereby allowing for improved profits, strengthening the local economy, and contributing to the organizations' ability to stay in business (Day, 2011; Masa'deh, 2012).

Role of the Researcher

The role of the researcher in a qualitative exploratory case study is to act as the data collection instrument (Yin, 2014). The researcher must also be able to build a rapport with the interview participant by asking good questions, being a good listener, and being able to adjust to changes (Yin, 2014). In this study, I followed all ethical guidelines for protecting research participants (U.S. Department of Health and Human

Services, 1979; Yin, 2014). The ethical guidelines are respect, beneficence, justice, informed consent, risks and benefits assessment, the selection of subjects of research, and avoiding bias (U.S. Department of Health and Human Services, 1979; Yin, 2014).

I have over 20 years of professional experience with software development as a website developer, project manager, and team manager, and I was not employed by the participating organizations during this study. My previous experience with SQA increased the potential for bias in this study. However, ethical considerations for data collection and analysis eliminated the potential for bias (Yin, 2014).

Interviews acted as the primary data collection method because they allowed the participants to share their experiences in an open-ended method (Yin, 2014). Interviews allow researchers to observe the participants while collecting the data, leading to more in-depth data collection (Yin, 2014). Before conducting this study, I had experience interviewing at a professional level for hiring employees and conducting requirements gathering for projects.

Participants

The participating organizations selected for inclusion in this study were from the available small software development organizations located in Saint John, New Brunswick, Canada. The three participating organizations were from those available based on willingness to participate. The eligibility criteria for interview participants was through the purposeful sampling (Comi, Bischof, & Eppler, 2014; Shaw, 2012; Yin, 2014) of QA leaders responsible for SQA within the participating organizations (Comi et

al., 2014; Shaw, 2012; Yin, 2014). Selected participating organizations were not my work peers or family members (M. E. Brown, 2013; Maskara, 2014; Shaw, 2012).

The business leaders of the participating organizations introduced me to the interview participants. I then collected letters of consent from the participants, as well as a nondisclosure agreement signed at the request of the organizations. Participants attended the interviews on a voluntary basis (Comi et al., 2014; Maskara, 2014; Yin, 2014). Participants had the option of removing themselves from the study at any time (Comi et al., 2014; Maskara, 2014; Yin, 2014). No data collected for this study identified the participating organizations or interview participants, protecting their anonymity (Comi et al., 2014; Maskara, 2014; Yin, 2014).

Research Method

Researchers have three research methods available to them: qualitative, quantitative, and mixed method (Venkatesh, Brown, & Bala, 2013). In this study, I used a qualitative research method. Qualitative research explores and understands a phenomenon through the experiences of the research participants, using interviews as a data collection method (Kapoulas & Mitic, 2012; Rennie, 2012; Venkatesh et al., 2013). The exploratory nature of the qualitative method aligned with the purpose of the study to explore the strategies QA leaders in small software development organizations can use for successful SQA processes. The focus of quantitative research is to understand the relationship between variables through the testing of the hypothesis (Yoshikawa et al., 2013), which was not the focus of this study. The mixed method is a combination of both qualitative and quantitative research (Venkatesh et al., 2013; Yoshikawa et al., 2013).

The elimination of quantitative research, the time commitment required, and the lack of existing data to analyze eliminated mixed method research as appropriate for this study (Venkatesh et al., 2013; Yoshikawa et al., 2013).

Research Design

This study had a multiple case study design. The five designs in qualitative research are a case study, narrative, phenomenological, ethnographic, and grounded theory. A multiple case study allowed me to focus on specific organizations' processes, making a case study the best design aligned with this study's purpose statement (Yin, 2014). This study did not use a narrative research design because the chronological ordering of events did not align with understanding organizations' processes (De Loo et al., 2015). A phenomenological study focuses on the lived experiences of the participants and does not allow for the exploration of organizations' SQA processes and so was not appropriate for this study (Giorgi, 2012). Ethnographic research as a design for this study would place a cultural focus on the research participants, which was not in my interest as the researcher (Zilber, 2014). Grounded theory was an unsuitable research design for this study because the aim of grounded theory is to establish a new theory, which contrasts with the purpose of this study (Urquhart & Fernández, 2013).

Population and Sampling

The population of the study included all QA leaders active in the SQA processes within the participating organizations. To achieve the appropriate sampling, interview participants were purposefully selected to align their experience as rich sources of information with the purpose of the study (Cleary, Horsfall, & Hayter, 2014; DeFeo,

2013; Maskara, 2014). The interview participants were QA leaders who worked on a software development project within the participating organizations and had the responsibility for SQA (M. E. Brown, 2013; Shaw, 2012; Yin, 2014).

The business leaders of the participating organizations provided me with a list of recommended participants (M. E. Brown, 2013; Shaw, 2012; Yin, 2014). I scheduled the interviews with the participants through e-mail. The interview setting was comfortable, familiar, and at a convenient time for the participants (Ando, Cousins, & Young, 2014; Maskara, 2014; Yin, 2014). Documentation such as processes, policies, and reports gathered enabled additional analysis (Kapoulas & Mitic, 2012; Rowley, 2012; Thomas, 2015).

Data saturation is the point in qualitative research when data contains no new themes or concepts (Ando et al., 2014). Ando et al. (2014) demonstrated that while performing data collection, 76.5% of discovered main codes occurred in the first six interviews, including all major themes. To ensure enough data for the analysis in this study, I achieved data saturation through the selection of two interview participants per participating organization for a total of six interview participants (Ando et al., 2014; Cleary et al., 2014; DeFeo, 2013). Six participants were a small number that aligned with the findings of qualitative research literature and fit within the population of the participating organizations (Ando et al., 2014; Cleary et al., 2014; DeFeo, 2013).

Ethical Research Considerations

The design of this study reduced the potential ethical risks. Researchers should not contact the interview participants before receiving Institutional Review Board (IRB)

approval (M. E. Brown, 2013; Maskara, 2014; Shaw, 2014). The IRB approval number for this study was 04-22-16-0484850. I provided a consent form to the participants, which included background information on this study and detailed how the participants' involvement was voluntary, that participants could ask to be removed at any time either verbally or in writing, and that the data would not include personal information.

All data collected for the study will be stored for 5 years using a secured folder in a password-protected and encrypted cloud-based storage system (M. E. Brown, 2013; Maskara, 2014; Shaw, 2014). Additionally, I stored the collected data in a password-protected Universal Serial Bus (USB) flash drive locked in a safe in my house. This will also be destroyed at the 5-year expiry (M. E. Brown, 2013; Maskara, 2014; Shaw, 2014). The password-protected and encrypted cloud-based storage system stored all relevant data during the study, as well as a password protected USB flash drive. I also did not reference participating organizations' names at any time during this study (M. E. Brown, 2013; Maskara, 2014; Shaw, 2014).

Data Collection Instruments

The primary instrument for data collection was face-to-face, semi structured interviews, which have been determined by the literature to be an appropriate instrument to use for qualitative case studies (Kapoulas & Mitic, 2012; Rowley, 2012; Yin, 2014). Member checking during each interview ensured the reliability and validity of the data collection process (Belin, 2015; Higdon, 2016; Scott, 2015). The interview questions were the primary tool for collecting information from the participants to answer the central research question: What strategies can QA leaders within small software

development organizations use for successful SQA processes? The interview questions aligned with the central research question to allow for data analysis. The interview questions I asked were as follows:

1. How would you describe the strategies you use for managing software quality?
2. What strategies do you use that are most effective for managing software quality?
3. What strategies do you use that are least effective for managing software quality?
4. How would you describe the critical success factors you use to measure your software quality?
5. What types of data do you track and store with relation to SQA?
6. How would you describe the benefits and constraints in adopting a SQA process?
7. How would you describe the project management methodology you use for your software development?
8. How would you describe the strategies you use for testing during the software development lifecycle?
9. What processes do you use for requirements gathering and validation?
10. How would you describe your process for including accessibility and usability when establishing your requirements?
11. What training strategies do you use for your SQA?

12. How would you describe your change management for feature integration in your software development lifecycle?
13. How would you describe your RCA processes?
14. What is an example of an incident where poor software quality lead to decreased profits for your organization?

Interview Question 1 established the participants' overall experiences regarding their SQA. Questions 2 and 3 allowed the participants to expand their answers from Question 1. Questions 4 and 5 established the critical success factors used to execute quantitative measures within the participating organizations' processes. Question 6 allowed the participants to explain the barriers to adopting an established industry framework for SQA. Question 7 established the project management methodology for comparison to industry best practices. Question 8 allowed the participants to explain their testing processes, an established industry best practice for SQA. Questions 9, 10, 11, and 12 allowed the participants to explain best practices from across all of the TQM frameworks that lead to high-quality software. Question 13 established RCA processes, which are a key component of SQA. Lastly, Question 14 helped set the context for how poor SQA affected the organization's profits.

The secondary instrument for data collection was the review of documents provided by the organizations (Kapoulas & Mitic, 2012; Rowley, 2012; Thomas, 2015). The documents were records, policies, procedures, diagrams, and other sources of data related to the organizations' SQA processes to complement the semi structured interviews. My analysis of the documentation separate from the interview data provided

additional insight to help answer the central research question (Kapoulas & Mitic, 2012; Rowley, 2012; Thomas, 2015).

Data Collection Technique

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The process for data collection followed the Walden IRB requirements. All electronic data was in a password-protected and encrypted cloud-based storage system, as well as a password-protected USB flash drive locked in a safe in my house (Maskara, 2014; Spengler, 2015; Wilkinson, 2012). The conversion of all physical data into electronically scanned data, later returning or destroying the original physical versions enabled data retention in the storage systems (Maskara, 2014; Spengler, 2015; Wilkinson, 2012). Walden IRB requires the destruction of all data for this study after 5 years of CAO approval (Maskara, 2014; Spengler, 2015; Wilkinson, 2012).

Study participants received consent forms. After collecting the consent forms from the participants, the participants received meeting requests from a Walden e-mail account. A reminder e-mail sent 1 day in advance of the interview notified the participants of the interview appointment. The face-to-face interviews consisted of open-ended questions with the goal of lasting 30 to 60 minutes (Maskara, 2014; Spengler, 2015; Wilkinson, 2012). The interviews were at a convenient time and location for the participants.

The interviews were audio recorded on an iPad, and additional notes recorded on a laptop using Microsoft OneNote software, deleting the audio file from the iPad after

confirming the interview audio data file is in the cloud-based storage system. Dragon 13 software aided interview audio file transcription. I validated for accuracy once completed by listening to the audio files while reading the transcripts. Redacting personal identifying information from the data files ensured anonymity (Maskara, 2014; Spengler, 2015; Wilkinson, 2012). The secondary data source was documentation electronically collected from the organization through a Walden e-mail account or physically collected from the interview participant (Maskara, 2014; Spengler, 2015; Wilkinson, 2012).

Reliability and validity of the data collection process used member checking during each interview (Belin, 2015; Higdon, 2016; Scott, 2015). Member checking included taking detailed notes during each interview and asking the participants clarifying questions to validate the recording of accurate responses (Belin, 2015; Higdon, 2016; Scott, 2015). Member checking allowed for the identification of any missing information (Belin, 2015; Higdon, 2016; Scott, 2015).

Data Organization Technique

Electronic folders divided all data stored in the cloud-based storage system organized on how the data collection occurred as one folder for interviews and one folder for documentation as the secondary data source. All interview audio files received a naming convention Organization1-Participant 1, Organization1-Participant 2, etc. The transcribed files for each interview audio file were stored in a separate folder using the same naming convention. The Microsoft OneNote file was sectioned based on the interview participants and used the same naming convention for each interview as Participant 1, Participant 2. The conversion of all hard copies of documentation as the

secondary data source to an electronic format endures for 5 years, followed by deletion. The hard copy sources followed the process of conversion to electronic format through scanning, then destruction.

Data Analysis

This study answered the research question: What strategies can QA leaders of small software development organizations use for their SQA processes? Data analysis followed the inductive strategy of (a) compiling the data, (b) disassembling the data into codes, (c) reassembling the data into themes, (d) interpreting the meaning of the data by applying critical thinking, and (e) concluding the data (M. E. Brown, 2013; Thomas, 2015; Yin, 2014). In the following subsections, I will discuss the data analysis inductive strategy used in this study.

Compiling the Data

I used NVivo 10 software to compile all interview data and documentation (M. E. Brown, 2013; Shaw, 2012; Thomas, 2015). Compiling the data into a central resource allowed for the data analysis, holistically looking for commonality across the data (Lee, 2014; Thomas, 2015; Yin, 2014). Organizing the data through categorizations of participant, interview question, or documentation source enabled the analysis from different perspectives.

Disassembling the Data into Codes

Data analysis includes searching for reoccurring codes grouped into common themes (Lee, 2014; Thomas, 2015; Yin, 2014). Each interview question from each participant required analysis for common codes. Documentation required the same

analysis to establish additional codes. I performed all data analysis using NVivo10 software.

Reassembling the Data into Themes

Reassembling the data into themes involves analysis to find commonality in the established codes, and then categorization into themes (Lee, 2014; Thomas, 2015; Yin, 2014). NVivo 10 software was the tool I used to tabulate these themes, categorized for each interview question and documentation source. Reassembling the data established the themes that occurred while analyzing the interviews and documentation sources.

Interpreting the Meaning of the Data by Applying Critical Thinking

Analysis of the data themes using critical thinking provides an interpretation of what the themes mean (Lee, 2014; Thomas, 2015; Yin, 2014). I used NVivo 10 for categorizing all the recorded notes with the established themes. Recorded notes were the source material for describing the themes and conclusions in the final presentation of the study.

Concluding the Data

The final write up of the study included the conclusions based on the data. These conclusions included the required sections of the Walden University doctoral dissertation rubric. The conclusion contains a discussion of an alignment between the data and the problem statement, purpose statement, and central research question of this study.

Reliability and Validity

Dependability, credibility, transferability, and confirmability demonstrate the concepts of reliability and validity in qualitative research (Peltzer & Teel, 2012).

Reliability is the ability of a researcher to reproduce the results of a previously conducted study if provided the same conditions and speaks to the dependability of a qualitative study (McCann, 2013; Thomas, 2015). Member checking, documenting the processes, and recording any changes that may occur demonstrated reliability and validity for the study (Belin, 2015; Higdon, 2016; Scott, 2015).

My prolonged contact with this case study and triangulation of data achieved credibility (Maskara, 2014; Shaw, 2012; Spengler, 2015). Data triangulation was through the alignment of the interview data, documentation, and research notes taken during the interviews (Yin, 2014). Member checking also helped ensure creditability (Belin, 2015; Higdon, 2016; Scott, 2015). Researchers cannot transfer qualitative multiple case studies, which was true for this study (M. E. Brown, 2013).

Validity is the demonstrated accuracy of the analysis results related to confirmability (M. E. Brown, 2013; Maskara, 2014). The processes used for data collection and analysis were standard for qualitative case studies and led to the confirmability of this study. Achieving data saturation for this study through the selection of six interview participants fit within the size of the participating organizations, aligned with qualitative research, and ensured enough data to analyze (Ando et al., 2014; Cleary et al., 2014; DeFeo, 2013). Member checking allowed for data saturation to occur through a process with the participants of follow-up questions until the participants shared no new information (Belin, 2015; Higdon, 2016; Scott, 2015).

Transition and Summary

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The two sources of data for the study were face-to-face, semi structured interviews and documentation (M. E. Brown, 2013; Maskara, 2014; Yin, 2014). Reliability and validity of the data collection process used member checking during each interview (Belin, 2015; Higdon, 2016; Scott, 2015). This study followed the Walden IRB process. The six purposefully selected interview participants had the responsibility for SQA from the participating organizations (M. E. Brown, 2013; Shaw, 2012; Yin, 2014). The data analysis involved searching for reoccurring codes grouped into common themes to answer the central research question (Lee, 2014; Thomas, 2015; Yin, 2014). Section 3 contains the presentation of findings, application to professional practice, implications for social change, recommendations for action, recommendations for further research, reflections, and a conclusion.

Section 3: Application to Professional Practice and Implications for Change

Introduction

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. I analyzed the QA processes from three software development organizations in Saint John, NB, Canada from data collected through face-to-face, semistructured interviews with two QA leaders from each organization, for a total of six interviews. Each organization also shared documentation including processes and training material that provided a deeper understanding of the QA processes. Data were analyzed and coded into themes, and Section 3 contains a presentation of the results.

The major themes in the results were Agile practices, documentation, testing, and lost profits. The results were in contrast to the main themes discovered in the literature review, which were statistical process control, release planning, RCA, testing techniques, requirements tracing, leadership, customer focus, and treating the organization as a total system, although there is some overlap. Section 3 will include the presentation of the findings, applications to professional practice, implications for social change, recommendations for action, recommendations for further research, reflections, and the conclusion.

Presentation of the Findings

To accomplish the primary objective, the central research question for this study was: What strategies can QA leaders within small software development organizations use for successful SQA processes? The interview questions explored the central research

question through the lens of the best practices discovered in the literature review: statistical process control, release planning, RCA, testing techniques, requirements tracing, leadership, customer focus, and treating the organization as a total system. The interview questions were open-ended to allow the participants to explore what strategies worked and did not work. The interview participants also provided an example of when their organization may have lost profits because of inadequate SQA processes. My analysis of all interview responses and documentation collected from the organizations led to major themes of Agile practices, documentation, testing, and lost profits. In the following subsections, I discuss each major theme as a comparison and contrast to the main literature review themes.

Agile Practices

Agile practices emerged as a major theme from each of the participants. However, none of the participants used Agile practices in whole as a project management methodology. Organization 1-Participant 1 stated, “We use the Agile approach. We move rapidly.” Organization 2-Participant 1 stated, “I say we don't do Agile; we try to do Agile.” Finally, Organization 3-Participant 1 stated, “We're not Agile or not waterfall we're somewhere in between. We're iterative.” Each participant used Agile concepts molded to the unique SQA processes. Common Agile practices were iterative work through sprints, frequent code check-in, release planning, and a customer focus. The duration of the sprints and frequency of releases varied between the participants.

Participants from one organization used short daily meetings, while the other two organizations’ participants found daily meetings to be too much overhead and relied on

the regular communication of the team members. None of the participants exercised paired programming or performed code reviews in a retrospective meeting. Each participant worked off a ticket tracking system; however, only one participant configured tickets into sprint backlogs.

The release planning of the participants varied from several weeks to several months. Each participant identified the struggle with coordinating the release of defect patches with the scheduled releases of new features or enhancements, which demonstrated change management challenges. The change management of release planning relied on the QA leaders of each organization to be aware of what was happening at the development level. All of the participants discussed their process of reading the ticket system and participating in meetings with development leadership. Organization 2-Participant 1 stated, “In the morning that is what I do; it is the first thing I do. I look through all the queues, and I stick my nose in everyone's business.”

Statistical process control is important in Agile project management (Rafique, & Mišić, 2013). Each organization’s QA leaders used Agile statistics sparingly, with only one participant mentioning the use of burndown charts, tracking velocity, and variance from time estimates. When asked if Organization 1-Participant 1 tracked velocity the reply was, “Yeah, we track it. And then we track the variance, and how close they were to their estimations based on their final hours logged. We track all of that.”

Customer focus is an important aspect of Agile practices (Pedersen, 2013). Several participants discussed a customer focus as an important concept. Organization 2-Participant 2 exclaimed, “Communication with the customer is pretty big here.”

Organization 3-Participant 2 mentioned, “If we have an escalation from customer care it trumps anything we have to be doing on a project.” When asked about customer inclusion on validating requirements, Organization 2-Participant 1 explained:

> We test it before we bring it to training because not only is it a training environment for our customer, it is also a place where they will see changes that they've asked for. Sometimes they want to review it on their side before it goes.

Documentation

The consistent discussion of documentation by all participants as a SQA process was a discovery that extended the knowledge of the literature review. All interview participants discussed the need for clear documentation to improve SQA processes. However, all participants expressed that their SQA processes lacked adequate documentation for coding, but QA leaders documented the SQA processes. The participants all stressed undertaking initiatives to improve documentation. Organization 2-Participant 2 stated, “[developers] are lacking on the documentation, but we're building them up, and they are definitely building up by the day.”

The participants’ responses varied in the degree of documentation used for test cases. The participants from one organization did not use test cases and relied on the system knowledge of one participant as a guide to remember what needed testing despite the complexity of the application. However, the QA leaders included the customer support representatives in the SQA process as test users. Conversely, the other two organizations’ participants fully documented test plans and cases based on the

requirements documentation. Each participant documented defect resolution within the ticketing system.

All participants discussed proper requirements documentation as an important strategy for SQA. Each organization elicited requirements from the customer in different capacities. One organization's participants relied on customers to call customer service for feature requests in addition to internal innovation to drive requirements, whereas the other two organizations' participants employ business analysts as customer service representatives to gather customer requirements. None of the participants considered usability and accessibility to be primary concerns when establishing requirements documentation. Each participant thought of usability during testing, but not during requirements gathering. However, all participants indicated that product usability was excellent. Organization 1-Participant 2 remarked, "The application itself is very easy to use," echoing similar comments from the other participants.

The level of requirements validation varied between the participants. Only one organization's participants included the customer to validate requirements of new features before building a feature. One organization's participants relied on internal employees to represent the customer and validate requirements after gathering the requirements. Another organization's participants would fully build new features first, and then demo the features with customers to understand if the requirements were valid.

Participants did not perform requirements tracing. Each participant discussed the use of Agile user stories in varying degrees of complexity. However the code deployed was not always aligned with the user story documentation. Organization 3-Participant 1

explained “our technical components do not map directly back to our business requirements because the business requirement doesn't have, doesn't care what the ‘how’ is with how you're building it.”

Testing

Testing techniques varied between the participants. However, the common practice between each participant was multistage testing. Each participant progressed code through a development environment, a QA environment, and a production environment. The QA environment served different purposes for each participant’s SQA processes including a place for customers to do functional testing before production releases. Each participant used a ticketing system to track the progress of unit testing, functional testing, and regression testing. Only one organization’s participant used automated testing software to allow for the execution of regularly required regression testing. Organization 2-Participant 2 explained, “It's not anything fancy it just kind of goes through and makes sure basic functionality works. So we run that every time we go to [QA environment]. It's normally the first thing we do.”

The participating QA leaders use statistical process control sparingly. All participants monitored the number of tickets in the SQA system. One organization’s participants monitored the time to complete a ticket, which is beneficial for technical debt calculations. One organization’s participants did monitor some Agile statistics. However, the low usage of statistical process control by each participant contrasted with the possible metrics that could be monitoring such as defect density ratios or cost measurements including technical debt. Additionally, no participant discussed the use of

Six Sigma's statistical process control benefits, CMMI process maturation, or ISO 9001 best practices as a part of SQA processes.

Each participant performed RCA in a similar method. A tiered customer support system was in place that could allow for the escalation of the defect to be ultimately corrected by the developer who originally wrote the code. The participants' processes aligned with industry standards of RCA established by Dalal and Chhillar (2013).

A common challenge faced by the participants was the existence of a development environment that was not scaled the same as production and not synchronized with production data. Each participant indicated the difficulty to test some features adequately because of a lack of production data, the development environment not matching production hardware, or production integration complexity with third party applications. Additionally, the unique customer environment configurations also made the software difficult to test adequately. The inability to adequately test is also a discussion point for lost profits.

Lost Profits

The theme of lost profits was common between each participant when referring to wasted time and the inefficiency of repeatedly correcting the same defects. None of the participants were able to point to a major failure of their product that led to lost profits, sales, or life. The participants highlighted additional time wasting activities resulting from poorly defined requirements; requirements that changed, but inadequately updated for the other team members; and missing communication between team members leading to poor change management.

Each participant discussed the inability to test unique customer environments. The participants all expressed the challenge to replicate customers' environments for adequate testing before releasing functionality to production. Therefore, the complexity of replicating every customer's environment was a cause of defects. All participants discussed the wasted time to correct defects post release. Participants also explained that the inability to test different customer configurations led to organizations' reputations suffering from relying on customers to identify defects rather than QA. Defects that remain in the system without being corrected was identified as a potential for lost profits by Organization 2-Participant 1, who explained "it's never really worked well. I know for sure that there's two main customers that have complained a lot, and one that actually threatened to leave because of it."

Of note was one participant's mention of team morale as an important benefit to having QA. Organization 3-Participant 2 shared "The team, it's morale. The team felt very unsatisfied with their work if it's shoved out the door before they feel they've put their stamp of quality on it. So there's a morale aspect there that we've seen." Employee turnover is a cause of lost profits for organizations (Subramony & Holtom, 2012) and this perspective from Organization 3-Participant 2 extends the knowledge of QA lost profits.

Misaligned customer focus with an organization's QA processes by rushing functionality into production to make a sale was a point made by both participants of Organization 3. The participants who discussed this form of technical debt identified the strategic choice to secure the sale over having working software. Organization 3-Participant 1 explained:

> Even if they think it's a product we have to, and we should be working with them like they're a beta customer, and you know trying to figure out what the real requirements are and how they're going to use it. Otherwise, we're going to spend a lot of money and miss the mark.

Organization 3-Participant 2 confirmed rushing functionality as an issue when stating, “having a piece of software that they can deploy somewhere sitting in their inbox sometimes takes precedent over completing the testing.”

Applications to Professional Practice

Tailoring SQA strategies unique to the organization was a common discovery in this study. Each participant had strategies that worked for their SQA needs. QA leaders can use this study to form successful strategies addressing SQA process concerns. The results of this study may help QA leaders by highlighted similar SQA process between the participating organizations as a confirmation of effective SQA. The analysis of the SQA processes used in the industry may add value to QA leaders through new strategies that could strengthen SQA processes if implemented. In addition to what SQA processes worked well, the analysis of what SQA processes performed poorly and led to lost profits may highlight areas for improving SQA processes. Maintaining a strong organization reputation was a discovery from the results of this study, becoming an important consideration for applications to professional practice.

This study added to existing literature on SQA. This study adds to the body of knowledge on TQM and the relation to SQA. The emerging theme of this study was the unique SQA processes each participating organization used based on existing knowledge

and experience of the QA leaders. There are areas of opportunity for QA leaders to improve SQA strategies through the use of additional industry best practices not currently used in the participating organizations. Participating organizations received a copy of this study to use as a reference for SQA processes.

Implications for Social Change

Strategies for improving SQA lead to social change through the improved knowledge and execution of better quality software by QA leaders (Majstorovic & Sibalija, 2015). If QA leaders become more mindful regarding SQA management then that may lead to reducing the cost of software development, increasing efficiency, and helping to make organizations more profitable and competitive (Singh & Kannojia, 2013). Employee morale achieved through the pride of work was an important consideration for positive social change discovered in the results of this study (Geiman, 2016). Meeting customers expectations was an important discussion point in the results of this study when considering business leaders' objectives to operate profitably and is demonstrated as a focal point of TQM (Burli et al., 2012). This study may contribute to social change through the identification of poor SQA processes that led to lost profits, enabling QA leaders to improve employee morale and strengthen SQA processes to remain in business and protect organizations' reputations.

The implications for positive social change include improved employee morale (Geiman, 2016), improved profits, contributing to organizations' ability to stay in business, and strengthening the local economy (Singh & Kannojia, 2013). Larger social communities benefit from the sustained longevity of small organizations, because small

organizations employ half of U.S. workers (Taneja et al., 2016). Improved SQA processes can impact positive social change through the potential for the sustained longevity of small software development organizations.

Recommendations for Action

The analysis of this study leads to recommendations for action in categories that apply to all QA leaders of software development organizations. Statistical process control is an important aspect of SQA (Masa'deh, 2012). QA leaders should define and closely monitor relevant metrics within SQA processes. Suggested metrics extend beyond the number of tickets in the system and should include defect density ratios and cost measurements when strategically accepting technical debt. Accurately measuring time spent correcting defects will highlight lost profits, because of wasted time for QA leaders. An investigation into industry best practices of Six Sigma, CMMI, and ISO 9001 may provide insight to QA leaders that will strengthen existing SQA processes whether adopted in whole or in part.

QA leaders should consider employee morale when thinking of the system as a whole. Additionally, organizations as a TQM structure should consider SQA when engaging customers to align expectations of quality software with employee’s ability to deliver quality. Organizations should validate new features with customers before writing code to avoid wasting time on non value add enhancements. The inclusion of the customer closely during the software development lifecycle will also enable the functionality to be more valuable and defect free. Frequent requirements validation with the customer will help avoid wasted time when the requirements change. Additionally, a

change management process should be in place to allow for the easy communication of changed requirements and release planning between all stakeholders.

Agile practices focused on iterative design should be a consideration in SQA processes. The more frequent the code check-in within a SQA process, the more frequent the testing can occur. QA leaders should use a multi staged testing environment that resembles production to allow for accurate testing techniques. Deploying automated regression testing should be considered in the testing strategies to speed test execution and reduce redundancy of test cases. Test cases should be well documented and executed by multiple SQA resources to improve process knowledge and growth of SQA skills.

Accurate documentation should flow from requirements through to defect resolution. Requirements and defect resolution documentation should align to provide the context of why features existed, and why certain corrective actions for defects occurred. Appropriate documentation should be available to both internal and external stakeholders to increase product knowledge and improve defect detection.

Recommendations for Further Research

The purpose of this qualitative multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The results of this study highlighted the strategies some organizations in Saint John, NB Canada used for SQA. This study could be replicated for different geographic areas to extend the knowledge in this industry.

A continuation of this study could be applied to the participating organizations to understand SQA from a quantitative perspective. QA leaders could deploy the

recommended statistical process controls to establish SQA metric baselines. Each recommendation for action could be individually deployed and monitored for a duration of time, enabling the establishment of a second baseline and the measurement of the effectiveness of the changes deployed between the baselines.

Reflections

The opportunity to conduct research in the industry of SQA exceeded all expectations. The thought that the participating organizations may find value in this research added to the reward of contributing to the existing literature on TQM and SQA. The possibility to contribute to the software development industry with the potential for positive social change as an outcome was a motivating factor throughout the process of this study, enriching the educational experience.

I learned despite organizations' unique SQA processes there was a commonality between all processes. The QA leaders were open to discussing their SQA strategies which allowed for the exploration to occur without hesitation. The QA leaders were prideful of their work and were willing to learn more about SQA through the participation of this study.

Conclusion

The purpose of this qualitative, exploratory, multiple case study was to explore strategies QA leaders in small software development organizations can use for successful SQA processes. The industry research conducted identified best practices of statistical process control, release planning, RCA, testing techniques, requirements tracing, leadership, customer focus, and treating the organization as a total system. The themes

identified in the data analysis of this study were the use of Agile practices, documentation, and testing. An analysis of the themes identified in this study included comparing and contrasting to the industry best practice themes, highlighting the participating organizations' overlaps and gaps.

Recommendations for action may help strengthen QA leaders' strategies for SQA. Recommendations in this study include using statistical process control, considering employee morale when thinking of the system as a whole, including customers throughout the SQA processes, using Agile practices focused on iterative design, and maintaining accurate documentation. Recommendations for further research include the replication of this study in different geographic areas and further quantitative research within the participating organizations to understand the impact of implementing any recommendations.

This study may enable QA leaders to strengthen SQA processes through the identification of poor SQA processes that led to lost profits. The implications for positive social change include improved employee morale, improved profits, contributing to organizations' ability to stay in business, and strengthening the local economy. Software development organizations may remain sustainable by strengthening their SQA strategies through the use of this study.

References

Abdulmouti, H. (2015). The role of kaizen (continuous improvement) in improving companies' performance: A case study. *2015 International Conference on Industrial Engineering and Operations Management (IEOM),* 1–6. doi:10.1109/IEOM.2015.7093768

Agarwal, N., Karimpour, R., & Ruhe, G. (2014). Theme-based product release planning: An analytical approach. *2014 47th Hawaii International Conference on System Sciences*, 4739–4748. doi:10.1109/HICSS.2014.582

Ahmad, M. F., Zakuan, N., Jusoh, A., Ariff, M. S. M., & Takala, J. (2013). Relationship amongst TQM, business performance, tools and techniques: Qualitative study result. *2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC)* 22–27. doi:10.1109/BEIAC.2013.6560120

Al-Dhaafri, H. S., & Al-Swidi, A. (2016). The impact of total quality management and entrepreneurial orientation on organizational performance. *International Journal of Quality & Reliability Management, 33*(5), 597–614. doi:10.1108/IJQRM-03-2014-0034

Allman, E. (2012). Managing technical debt. *Communications of the ACM, 55*(5), 50–55. doi:10.1145/2160718.2160733

Ando, H., Cousins, R., & Young, C. (2014). Achieving saturation in thematic analysis: Development and refinement of a codebook. *Comprehensive Psychology, 3*(4), 1–7. doi:10.2466/03.CP.3.4

Azizi, A. (2015). An integrated framework of critical techniques in implementation of Total Quality Management. *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, 1–7. doi:10.1109/IEOM.2015.7228096

Bae, G., Rothermel, G., & Bae, D. H. (2014). Comparing model-based and dynamic event-extraction based GUI testing techniques: An empirical study. *Journal of Systems & Software*, *97*, 15–46. doi:10.1016/j.jss.2014.06.039

Baggen, R., Correia, J. P., Schill, K., & Visser, J. (2012). Standardized code quality benchmarking for improving software maintainability. *Software Quality Journal*, *20*(2), 287–307. doi:10.1007/s11219-011-9144-9

Baldassarre, M. T., Caivano, D., Pino, F. J., Piattini, M., & Visaggio, G. (2012). Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: From a theoretical comparison to a real case application. *Software Quality Journal, 20,* 309-335. doi:10.1007/s11219-011-9154-7

Bass, J. M., Allison, I. K., & Banerjee, U. (2013). Agile method tailoring in a CMMI level 5 organization: Addressing the paradox. *Journal of International Technology and Information Management, 22*(4), 77-98. Retrieved from http://www.iima.org/index.php?option=com_phocadownload&view=section&id=11&Itemid=77

Belin, D. (2015). *Police perceptions and decision making related to domestic minors trafficked through prostitution* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3705130)

Bijlsma, D., Correia, J. P., & Visser, J. (2012). Automatic event detection for software product quality monitoring. *2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC)*, 30-37. doi:10.1109/QUATIC.2012.22

Brown, A. (2013). Quality: Where have we come from and what can we expect? *TQM Journal*, *25*, 585–596. doi:10.1108/TQM-04-2013-0045

Brown, M. E. (2013). *Data-driven decision making as a tool to improve software development productivity* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 1435602894)

Bu, X. Z., & Cao, L. (2015). Study on the relationship between TQM implementation and corporate financial performance. *2015 12th International Conference on Service Systems and Service Management (ICSSSM)*, 1–6. doi:10.1109/ICSSSM.2015.7170339

Bu, X. Z., Liu, S., & Peng, T. (2013). The impact of the TQM implementation on corporate financial performance 2014: Based on empirical data of Chinese listed companies. *2013 10th International Conference on Service Systems and Service Management*, 6–11. doi:10.1109/ICSSSM.2013.6602581

Burli, S., Bagodi, V., & Kottursheṭtar, B. (2012). TQM dimensions and their interrelationships in ISO certified engineering institutes of India. *Benchmarking*, *19*(2), 177–192. doi:10.1108/14635771211224527

Certified test manager. (2015). Retrieved from http://testinginstitute.com/ctm.php

Cleary, M., Horsfall, J., & Hayter, M. (2014). Data collection and sampling in qualitative research: Does size matter? *Journal of Advanced Nursing, 70*, 473–475. doi:10.1111/jan.12163

Codabux, Z., & Williams, B. (2013). Managing technical debt: An industrial case study. *Managing Technical Debt (MTD), 2013 4th International Workshop on Technical Debt*. doi:10.1109/MTD.2013.6608672

Comi, A., Bischof, N., & Eppler, M. J. (2014). Beyond projection: Using collaborative visualization to conduct qualitative interviews. *Qualitative Research in Organizations and Management: An International Journal, 9*(2), 110–133. doi:10.1108/QROM-05-2012-1074

Cotroneo, D., Pietrantuono, R., & Russo, S. (2013). A learning-based method for combining testing techniques. *Proceedings of the 2013 International Conference on Software Engineering,* 142–151. doi:10.1109/icse.2013.6606560

Curtis, B., Sappidi, J., & Szynkarski, A. (2012). Estimating the size, cost, and types of Technical Debt. *2012 Third International Workshop on Managing Technical Debt (MTD).* doi:10.1109/MTD.2012.6226000

Dalal, S., & Chhillar, R. S. (2013). Empirical study of root cause analysis of software failure. *SIGSOFT Software Engineering Notes*, *38*(4), 1–7. doi:10.1145/2492248.2492263

David, M. (2013). Organising, valuing and improving the engineering design process. *Journal of Engineering Design, 24*, 524-545. doi:10.1080/09544828.2013.776214

Day, L. E. (2011). *The integration of software quality into software project management* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3440422)

De Castro, R. M., Braga, J. L., & Soares, L. S. (2013). Selection of good practices for small software development teams: A knowledge-based approach. *SIGSOFT Software Engineering Notes*, *38*(6), 1–15. doi:10.1145/2532780.2532804

De Loo, I., Cooper, S., & Manochin, M. (2015). Enhancing the transparency of accounting research: The case of narrative analysis. *Qualitative Research in Accounting & Management*, *12*(1), 34–54. doi:10.1108/QRAM-02-2013-0007

De Vasconcelos, A. M. L., de la Vara, J. L., Sanchez, J., & Pastor, O. (2012). Towards CMMI-compliant business process-driven requirements engineering. *2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC),* 193-198. doi:10.1109/QUATIC.2012.61

DeFeo, D. J. (2013). Toward a model of purposeful participant inclusion: Examining deselection as a participant risk. *Qualitative Research Journal*, *13*(3), 253–264. doi:10.1108/QRJ-01-2013-0007

Dias, A. L., Pontin de Mattos Fortes, R., & Masiero, P. C. (2012). Increasing the quality of web systems: By inserting requirements of accessibility and usability. *2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC)*, 224–229. doi:10.1109/QUATIC.2012.33

Drohomeretski, E., Gouvea da Costa, S. E., Pinheiro de Lima, E., & Garbuio, P. A. da R. (2014). Lean, Six Sigma and Lean Six Sigma: An analysis based on operations strategy. *International Journal of Production Research*, *52*, 804–824. doi:10.1080/00207543.2013.842015

Falessi, D., Shaw, M., & Mullen, K. (2014). Achieving and maintaining CMMI maturity level 5 in a small organization. *IEEE Software*, *31*(5), 80–86. doi:10.1109/MS.2014.17

Flumerfelt, S., Anna, B. S., & Kahlen, F. (2012). Are agile and lean manufacturing systems employing sustainability, complexity and organizational learning? *Learning Organization, 19*(3), 238–247. doi:10.1108/09696471211219976

Fried, A., Gey, R., Pretorius, A., & Günther, L. (2013). Decoupling from standards: Process management and technical innovation in software development organizations. *International Journal of Innovation Management*, *17*(4), 1–34. doi:10.1142/S1363919613500126

Galinac Grbac, T., Runeson, P., & Huljenić, D. (2013). A second replicated quantitative analysis of fault distributions in complex software systems. *IEEE Transactions on Software Engineering*, *39*, 462–476. doi:10.1109/TSE.2012.46

Geiman, M. (2016). *A multiple case study of the influence of positive organizational behavior on human resources* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 10129957)

Ghabi, A., & Egyed, A. (2012). Code patterns for automatically validating requirements-to-code traces. *Proceedings of the 27th IEEE / ACM International Conference on Automated Software Engineering*, 200–209. doi:10.1145/2351676.2351705

Giorgi, A. (2012). The descriptive phenomenological psychological method. *Journal of Phenomenological Psychology*, *43*(1), 3–12. doi:10.1163/156916212X632934

Harter, D. E., Kemerer, C. F., & Slaughter, S. A. (2012). Does software process improvement reduce the severity of defects? A longitudinal field study. *IEEE Transactions on Software Engineering*, *38*, 810–827. doi:10.1109/TSE.2011.63

Heger, C., Happe, J., & Farahbod, R. (2013). Automated root cause isolation of performance regressions during software development. *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, 27-38. doi:10.1145/2479871.2479879

Higdon, B. (2016). *Strategies independent coffee shop owners require to survive beyond five years* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3746740)

Hinojo, F. J. L. -L. (2014). Agile, CMMI, RUP, ISO / IEC 12207: Is there a method in this madness? *SIGSOFT Software Engineering Notes*, *39*(2), 1–5. doi:10.1145/2579281.2579299

Huang, Y. C., Peng, K. L., & Huang, C. Y. (2012). A history-based cost-cognizant test case prioritization technique in regression testing. *Journal of Systems & Software*, *85*, 626–637. doi:10.1016/j.jss.2011.09.063

Ilkay, M. S., & Aslan, E. (2012). The effect of the ISO 9001 quality management system on the performance of SMEs. *International Journal of Quality & Reliability Management, 29*, 753–778. doi:10.1108/02656711211258517

Jeffery, K. G. (2012). Quality aspects in CLOUD computing. *2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC),* 4-9. doi:10.1109/QUATIC.2012.47

Kapoulas, A., & Mitic, M. (2012). Understanding challenges of qualitative research: Rhetorical issues and reality traps. *Qualitative Market Research: An International Journal, 15*, 354–368. doi:10.1108/13522751211257051

Kastelic, M., & Peer, P. (2012). Managing IT services: Aligning best practice with a quality method. *Organizacija, 45*(1), 31. doi:10.2478/v10051-012-0004-6

Khalane, T., & Tanner, M. (2013). Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. *2013 International Conference on Adaptive Science and Technology (ICAST),* 1-6. doi:10.1109/ICASTech.2013.6707499

Lami, G., Fabbrini, F., & Fusani, M. (2013). A methodology to derive sustainability indicators for software development projects. *Proceedings of the 2013 International Conference on Software and System Process*, 70–77. doi:10.1145/2486046.2486060

Lee, Y. -A. (2014), Insight for writing a qualitative research paper. *Family and Consumer Sciences Research Journal, 43*(1), 94–97. doi:10.1111/fcsr.12084

Leopoulos, V., & Chatzistelios, G. (2014). Quality management systems development based on a production systems taxonomy. *TQM Journal, 26*(2), 215–229. doi:10.1108/TQM-03-2012-0024

Leveson, N. G. (2004). The role of software in spacecraft accidents. *Journal of Spacecraft and Rockets, 41,* 564–575. doi:10.2514/1.11950

Lin, C., Chuang, H., & Shih, D. (2012). Development stage and relationship of MIS and TQM in the e-business era. *International Journal of Electronic Business Management, 10*(1), 50–60. Retrieved from http://ijebm-ojs.ie.nthu.edu.tw/IJEBM_OJS/index.php/IJEBM

Lindgren, M., & Packendorff, J. (2009). Social constructionism and entrepreneurship: Basic assumptions and consequences for theory and research. *International Journal of Entrepreneurial Behavior & Research, 15*(1), 25–47. doi:10.1108/13552550910934440

Machado, I. do C., McGregor, J. D., Cerqueira Cavalcantic, Y., & Santana de Almeida, E. (2014). On strategies for testing software product lines: A systematic literature review. *Information & Software Technology*, *56,* 1183–1199. doi:10.1016/j.infsof.2014.04.002

Majstorovic, V., & Sibalija, T. (2015). From IMS and Six Sigma toward TQM: An empirical study from Serbia. *TQM Journal, 27*(3), 1–16. doi:10.1108/TQM-12-2013-0130

Malik, A., & Blumenfeld, S. (2012). Six Sigma, quality management systems and the development of organizational learning capability. *International Journal of*

Quality & Reliability Management, *29*(1), 71–91. doi:10.1108/02656711211190882

Masa'deh, R. (2012). The impact of management information systems (MIS) on quality assurance (QA): A case study in Jordan. *International Journal of Information, Business & Management*, *4*(2), 93-110. Retrieved from http://ijibm.elitehall.com/index4.htm

Maskara, A. (2014). *A process framework for managing quality of service in private cloud* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3636194)

McCann, D. E. (2013). *Managing changes in project scope: The role of the project constraints* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3568940)

McKnight, W. C. (2013). *Process and product quality assurance influence on compliance to the capability maturity model integration* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3564561)

Mora, M., O'Connor, R. V., Raisinghani, M., & Gelman, O. (2013). Design, build and evaluation of an ontology-based KMS for supporting CMMI-DEV understanding: Benefits and limitations. *International Journal of Software Engineering & Knowledge Engineering*, *23*, 999–1032. doi:10.1142/S0218194013500290

Morris, P. D. (2012). The perfect process storm: Integration of CMMI, Agile, and Lean Six Sigma. *Journal of Defense Software Engineering*, *25*(6), 39–45. Retrieved

from http://www.crosstalkonline.org/storage/issue-archives/2012/201211/201211-0-Issue.pdf

Mosadeghrad, A. M. (2014). Why TQM programmes fail? A pathology approach. *TQM Journal*, *26*(2), 160–187. doi:10.1108/TQM-12-2010-0041

Müller, S. D., & Nielsen, P. A. (2013). Competing values in software process improvement: A study of cultural profiles. *Information Technology & People*, *26*(2), 146–171. doi:10.1108/ITP-03-2013-0052

Murthy, D. (2013). Ethnographic research 2.0: The potentialities of emergent digital technologies for qualitative organizational research. *Journal of Organizational Ethnography, 2*(1), 23–36. doi:10.1108/JOE-01-2012-0008

Naciri, S., Idrissi, M. A. J., & Kerzazi, N. (2015). A strategic release planning model from TPM point of view. *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 1–9. doi:10.1109/SITA.2015.7358396

Nenty, H. J. (2009). Writing a quantitative research thesis. *International Journal of Educational Sciences, 1*(1), 19–32. Retrieved from http://krepublishers.com/02-Journals/IJES/IJES-00-0-000-000-2009-Web/IJES-00-0-000-000-2009-1-Cover.htm

Nicholas, J. (2016). Hoshin kanri and critical success factors in quality management and lean production. *Total Quality Management & Business Excellence*, *27*, 250–264. doi:10.1080/14783363.2014.976938

Nieminen, M., & Räty, T. (2015). Adaptable design for root cause analysis of a model-based software testing process. *2015 12th International Conference on Information Technology - New Generations (ITNG)*, 379–384. doi:10.1109/ITNG.2015.67

Nord, R. L., Ozkaya, I., Kruchten, P., & Gonzalez-Rojas, M. (2012). In search of a metric for managing architectural technical debt. *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference*. doi:10.1109/WICSA-ECSA.212.17

O'Mahony, K., & Garavan, T. N. (2012). Implementing a quality management framework in a higher education organization. *Quality Assurance in Education, 20*(2), 184–200. doi:10.1108/09684881211219767

Omar, M. K., & Murgan, S. (2014). An improved model for the cost of quality. *The International Journal of Quality & Reliability Management, 31*(4), 395–418. doi:10.1108/ijqrm-05-2012-0066

Pedersen, M. (2013). *A quantitative examination of critical success factors comparing agile and waterfall project management methodologies* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3602588)

Peltzer, J. N., & Teel, C. S. (2012). The development of a comprehensive community health center in a rural community. *Leadership in Health Services, 25*(1), 52–61. doi:10.1108/17511871211198070

Petersen, P. (1999). Total quality management and the Deming approach to quality management. *Journal of Management History, 5*, 468–474. doi:10.1108/13552529910290520

Pillai, A., Pundir, A., & Ganapathy, L. (2012). Implementing integrated Lean Six Sigma for software development: A flexibility framework for managing the continuity: Change dichotomy. *Global Journal of Flexible Systems Management, 13*(2), 107–116. doi:10.1007/s40171-012-0009-2

Rafique, Y., & Mišić, V. B. (2013). The effects of test-driven development on external quality and productivity: A meta-analysis. *IEEE Transactions on Software Engineering, 39*, 835–856. doi:10.1109/TSE.2012.28

Raza, M., & Faria, J. P. (2014). A model for analyzing estimation, productivity, and quality performance in the personal software process. *Proceedings of the 2014 International Conference on Software and System Process*, 10–19. doi:10.1145/2600821.2600828

Rennie, D. L. (2012). Qualitative research as methodical hermeneutics. *Psychological Methods, 17*, 385–398. doi:10.1037/a0029250

Robillard, P. N., & Lavallée, M. (2012). Software team processes: A taxonomy. *Proceedings of the International Conference on Software and System Process,* 101–109. doi:10.1109/icssp.2012.6225952

Roche, J. (2013). Adopting DevOps practices in quality assurance. *Queue*, *11*(9), 20–27. doi:10.1145/2538031.2540984

Rowley, J. (2012). Conducting research interviews. *Management Research Review*, *35*(3/4), 260–271. doi:10.1108/01409171211210154

Sahin, E., Kaynak, I. K., & Sencan, M. U. (2013). A pilot study: Opportunities for improving software quality via application of CMMI measurement and analysis. *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 243–246. doi:10.1109/IWSM-Mensura.2013.42

Schmidt, M. T., Elezi, F., Tommelein, I. D., & Lindemann, U. (2014). Towards recursive plan-do-check-act cycles for continuous improvement. *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 1486–1490. doi:10.1109/IEEM.2014.7058886

Scott, M. E. (2015). *Strategies for retaining employees in the hospitality industry* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3738026)

Seaman, C., Guo, Y., Izurieta, C., Cai, Y., Zazworka, N., Shull, F., & Vetrò, A. (2012). Using technical debt data in decision-making: Potential decision approaches. In *2012 Third International Workshop on Managing Technical Debt (MTD)*. doi:10.1109/mtd.2012.6225999

Sedevich-Fons, L. (2014). Financial indicators in healthcare quality management systems. *TQM Journal*, *26*, 312–328. doi:10.1108/TQM-01-2014-0009

Shah, S. M. A., Morisio, M., & Torchiano, M. (2012). The impact of process maturity on defect density. *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 315–318. doi:10.1145/2372251.2372308

Shapiro, R. J. (2014). The U.S. software industry: An engine for economic growth and employment. *Software and Information Industry Association*. Retrieved from http://www.siia.net/Admin/FileManagement.aspx/LinkClick.aspx?fileticket=ffCbUo5PyEM%3D&portalid=0

Sharma, M., & Vishwakarma, R. G. (2012). CMMI based software metrics to evaluate OOAD. *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*. doi:10.1145/2393216.2393220

Shaw, B. T. (2012). *Exploring the factors of an enterprise resource planning system in a local government organization* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3510463)

Shih, S. -P., Shaw, R. S., Fu, T. Y., & Cheng, C. P. (2013). A systematic study of change management during CMMI implementation: A modified activity theory perspective. *Project Management Journal*, *44*(4), 84–100. doi:10.1002/pmj.21358

Shokri, A., Waring, T. S., & Nabhani, F. (2016). Investigating the readiness of people in manufacturing SMEs to embark on Lean Six Sigma projects: An empirical study in the German manufacturing sector. *International Journal of Operations & Production Management*, *36*, 850–878. doi:10.1108/IJOPM-11-2014-0530

Siebra, C. S. A., Tonin, G. S., Silva, F. Q. B., Oliveira, R. G., Junior, A. L. O. C., Miranda, R. C. G., & Santos, A. L. M. (2012). Managing technical debt in practice: An Industrial report. *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. doi:10.1145/2372251.2372297

Simon, M. K. (2011). *Dissertation and scholarly research: Recipes for success* (3rd ed.). Seattle, WA: Dissertation Success, LLC.

Singh, B., & Kannojia, S. P. (2013). A review on software quality models. *2013 International Conference on Communication Systems and Network Technologies (CSNT)*, 801–806. doi:10.1109/CSNT.2013.171

Software Engineering Institute. (2010). CMMI for Development, Version 1.3. Retrieved from http://www.sei.cmu.edu/reports/10tr033.pdf

Song, K. -W., & Kim, J. -S. (2012). Measurement and management of the level of quality control process in SoC (System on Chip) embedded software development. *International Journal of Advanced Robotic Systems, 9*, 1–7. doi:10.5772/45642

Spengler, S. S. (2015). *Educators' perceptions of a 21st century digital literacy framework* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3701210)

Subramony, M., & Holtom, B. C. (2012). The long-term influence of service employee attrition on customer outcomes and profits. *Journal of Service Research, 15*, 460–473. doi:10.1177/1094670512452792

Swinarski, M., Parente, D. H., & Kishore, R. (2012). Do small IT firms benefit from higher process capability? *Communications of the ACM, 55*(7), 129–134. doi:10.1145/2209249.2209276

Talib, F., Rahman, Z., & Qureshi, M. N. (2011). A study of total quality management and supply chain management practices. *International Journal of Productivity and Performance Management, 60*(3), 268–288. doi:10.1108/17410401111111998

Taneja, S., Pryor, M. G., & Hayek, M. (2016). Leaping innovation barriers to small business longevity. *Journal of Business Strategy, 37*(3), 44–51. doi:10.1108/JBS-12-2014-0145

Thomas, S. J. (2015). *Exploring strategies for retaining information technology professionals: A case study* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3681815)

Tlapa, D., Limon, J., García-Alcaraz, J. L., Baez, Y., & Sánchez, C. (2016). Six Sigma enablers in Mexican manufacturing companies: A proposed model. *Industrial Management & Data Systems, 116*, 926–959. doi:10.1108/IMDS-06-2015-0265

Tuan, N. N., & Thang, H. Q. (2013). Combining maturity with agility: Lessons learnt from a case study. *Proceedings of the Fourth Symposium on Information and Communication Technology.* doi:10.1145/2542050.2542072

Urquhart, C., & Fernández, W. (2013). Using grounded theory method in information systems: The researcher as blank slate and other myths. *Journal of Information Technology, 28*, 224–236. doi:10.1057/jit.2012.34

U.S. Department of Health and Human Services. (1979). *The Belmont Report*. Retrieved from http://www.hhs.gov/ohrp/humansubjects/guidance/belmont.html

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly, 37*(1), 21–54. Retrieved from http://www.misq.org/

Vierhauser, M., Rabiser, R., & Grünbacher, P. (2014). A case study on testing, commissioning, and operation of very-large-scale software systems. *Companion Proceedings of the 36th International Conference on Software Engineering,* 125–134. doi:10.1145/2591062.2591179

Wang, C. H. (2014). A longitudinal study of innovation competence and quality management on firm performance. *Innovation: Management, Policy & Practice, 16*(3), 392–403. doi:10.5172/impp.2013.2877

Williams, C. (2013). *A quantitative comparative and correlational study of critical success factors for information technology projects* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3564328)

Wilkinson, R. T. (2012). *Perceptions of supervisory relationship influences on cognitive complexity development during practicum supervision: A qualitative study* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3520492)

Wolfe, L. (2013). *Choosing the right software development plan for the project: Linear, iterative, agile, or a blend.* (Master's thesis). Retrieved from ProQuest Dissertations and Theses. (UMI No. 1545973)

Wright, G. P. (2013). *Success rates by software development methodology in information technology project management: A quantitative analysis* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3590342)

Yin, R. K. (2014). *Case study research: Design and methods*. Thousand Oaks, CA: SAGE Publications, Inc.

Yoshikawa, H., Weisner, T. S., Kalil, A., & Way, N. (2013). Mixing qualitative and quantitative research in developmental science: Uses and methodological choices. *Qualitative Psychology, 1*(S), 3–18. doi:10.1037/2326-3598.1.S.3

Young, D. K. (2013). *Project-method fit: Exploring factors that influence agile method use* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses. (UMI No. 3607635)

Zein, S., Salleh, N., & Grundy, J. (2016). A systematic mapping study of mobile application testing techniques. *Journal of Systems & Software, 117,* 334–356. doi:10.1016/j.jss.2016.03.065

Zilber, T. B. (2014). Beyond a single organization: Challenges and opportunities in doing field level ethnography. *Journal of Organizational Ethnography, 3*(1), 96–113. doi:10.1108/JOE-11-2012-0043