2020

Organization Global Software Development Challenges of Software Product Quality

Patrick Enabudoso
Walden University

Walden University

College of Management and Technology

This is to certify that the doctoral dissertation by

Patrick Oaiye Enabudoso

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. Richard Schuttler, Committee Chairperson, Management Faculty
Dr. Nikunja Swain, Committee Member, Management Faculty
Dr. Anton Camarota, University Reviewer, Management Faculty

Chief Academic Officer and Provost
Sue Subocz, Ph.D.

Walden University
2020

Abstract

Organization Global Software Development Challenges of Software Product Quality

by

Patrick Oaiye Enabudoso

MPhil, Walden University, 2019

MTech, Rivers State University, 2006

PGD, Rivers State University, 2001

Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Management of Information Systems

Walden University

October 2020

Abstract

Leaders of global software development (GSD) processes in organizations have been confronting low software product quality. Managers of these processes have faced challenges that have been affecting customer satisfaction and that have resulted in negative social impacts on public safety, business financial performance, and global economic stability. The purpose of this qualitative exploratory multiple case study was to discover a common understanding shared by managers in Canadian GSD organizations of how to meet software product quality goals and enhance customer satisfaction. The conceptual framework for the study was based on Deming's 14 principles of quality management. The purposeful sample included 30 knowledgeable participants who worked in Canada as GSD managers. Semistructured interviews conducted through telephone and audioconference tools, along with the review of related documents, were used to gather data. Eight themes emerged from the data analysis: developing a clear purpose and work principles, improving processes and employee skills, developing adequate personnel management strategies, promoting autonomy and personal worker development, formulating life cycle and development techniques, identifying challenges, formulating solutions, and focusing on product quality. The research findings have implications for positively influencing social change through the provision of methods and process knowledge to GSD organizational leaders. This information consists of best management and industry practices that can be applied to achieve software product quality and customer satisfaction, create management systems, maintain a competitive advantage, and prevent global software development project failures.

Organization Global Software Development Challenges of Software Product Quality

by

Patrick Oaiye Enabudoso

MPhil, Walden University, 2019

MTech, Rivers State University, 2006

PGD, Rivers State University, 2001

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Management of Information Systems

Walden University

October 2020

Dedication

I dedicated this research work to my lovely wife, Ifeoma Enabudoso, and my children: Sharonrose, Abigail, Michael, and Naomi. I know how much you made sacrifices for me to continue and achieve this great venture.

Acknowledgments

I thank God Almighty for His sustaining grace and blessings that are readily available for us all. He made it possible for me to complete this rigorous PhD academic journey.

I am indebted to all those who offered words of encouragement whenever I felt frustrated. I want to especially thank my wife, Ifeoma Enabudoso, for being there for me when I needed words of encouragement and supports. I thank my children, Sharonrose, Abigail, Michael, and Naomi, for their patience in those moments they needed my attention.

I thank the government of Canada for their Ontario Student Assistance Program (OSAP). My appreciation goes to all the participants who volunteered their time, shared knowledge, and experiences. I thank my school colleagues, Dr. Adewole Omogunsoye, and Mr. Darren Obazu, for their encouragements and shared ideas. I appreciate Dr. Sam Mgbatogu and Mr. Tony Oji, for their support and encouragements throughout this study.

I thank my academic dissertation committee members who impacted my research work positively: Professor Richard Schuttler (chair), Professor Nikunja Swain (committee member), and Professor Anton Camarota (University Research Reviewer). They ensured that I always did the right thing and made reasonable progress in my academic PhD journey.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction to the Study

Global software development (GSD) is a fast-emerging area in software engineering due to a shift from centralized to distributed software development because of the accruing benefits, such as cost and time savings, improved quality, faster development, and availability of cheap skilled manpower (Alzaidi & Qureshi, 2014). The use of software products is increasing in almost every human endeavor, such as with health care, transportation, and household devices as additional tasks are directed by the software structure (Miguel, Maurico, & Rodriguez, 2014). Software product quality is a leading issue in the software industry, due to poor software quality measurement and evaluation (Hossain, 2018). Higher demand is placed on quality of software products to meet the standard requirement in some sensitive departments where common mistakes have great consequences, such as banking, healthcare, aviation, and security (Rodriguez, Oviedo, & Piattini, 2016). Software developers spent most of their development costs on identifying errors, correcting, and debugging (Hossain, 2018). Hence, the pursuit of high authenticity and quality assurance helps prevent defective software arrangement (Shen, Ding, Ren, &Yang, 2018).

The research findings from this study included the reports of how GSD organizational leaders in Canada effectively manage the challenges of low software product quality in organizations to enhance customer satisfaction. In Chapter 1, the gap in the literature on GSD software product quality leads to the problem statement, purpose statement, and significance of study sections. In this study, the research questions were qualitative in nature, and the conceptual framework involved concepts and ideas of the management methods that guided the study. Completing this chapter are research assumptions, study scope, delimitations, summary, and the transition into the next chapter.

Background of the Study

The desire to achieve a higher level of software product quality brought about the increase in the number of software development life cycle (SDLC) certifications, such as the capability maturity model (CMM) for software, capability maturity model for integration (CMMI), International Organization for Standardization (ISO)15504, ISO 12207, ISO 9001, the new combined International Organization for Standardization and International Electrotechnical Commission (IEC) 25000 series that defines the framework for software quality product evaluation (Rodriguez et al., 2016). The software quality model evolution and researchers concentrated only on the improvement of software quality processes. The 21st century philosophy shifted from improving processes to software product quality. The frequent and most challenging software failure in GSD is attributable to a poor-quality culture in software engineering (Usman & Khan, 2018). The most effective software development approaches will still encounter challenges due to a complex environment. Alfaro and Chandrasekaran (2015) investigated how the national blend create impediments to information flow among GSD multisite teams. The findings showed that there are significant positive software quality effects due to national concerns on the team with a high measure of past working ties and negative impacts are projected for a team with low measure future working ties (Alfaro & Chandrasekaran, 2015).

Software quality management early inclusion in every stages of software development is imperative in global software development. Bergmane, Grabis, and Zeiris (2017) conducted the software quality defect root causes using the Isikawa technique, finding that human errors, complex code, and new technologies are possible root causes of software defects. Empirical evaluation of software quality assurance practice by Sowunmi, Misra, Fernandez-Sanz, Crawford, and Soto (2016) in developing countries such as Nigeria, and Turkey led to conclusions that software quality assurance is achievable through implementation of proper documentation and continuous monitoring of processes throughout the software project life cycle. Akbar, Shafiq, Kamal and Hamza (2019) emphasized change management that requires periodic and continuous updates of change in product life cycle to meet customer expectations. The need for software quality inclusion consideration at every stage of planning design and execution is imperative, so that defects and cost effects are minimized (Alebebisat, Alhalhouli, Alrawashdeh, & Alshabatat, 2018). The new software outsourcing partnership team is a gathering of successful outsourcing team members that equally face the same barriers that affect GSD, such as software quality, communication, and coordination challenges (Ali et al., 2018).

The advancing globalization Is a problem to software development distributed environment and human resource allocations. Bhatti and Ahsan (2016) confirmed the challenges of globalization as an indicator of the process in a distributed team’s setting and the effects of human resources management implementation of GSD. The authors deduced that the findings of their study inferred that the problems of GSD impact process improvement. Vrhovec, Kumer, Trkman, and Krisper (2015) surveyed three sites on the transition of economics and GSD approach to compensate for a lack of skilled information and communication technology (ICT) professionals by outsourcing to remote developers. The existing research studies on software quality can be categorized as total quality management (TQM), ISO 9000 quality management system, capability maturity models, defects prediction models, and evaluation techniques of specific software development process (John, Kadadevaramath, & Immanuel, 2016).

The software quality is the main challenge of GSD along with ensuring that every vendor organization satisfies customer requirement (Usman & Khan, 2018). The fear of losing reputation and trust due to software defects and software project failures is on the rise in contemporary GSD companies. Rehman and Khan (2014) asserted that modern organizations often compromise software quality practice for a cost reduction, meeting schedule, high productivity, and other benefits. The empirical research in GSD that could proffer a solution to the problem of low software quality is immature and evolving (Wickramarachchi & Lai, 2016). The empirical research reporting on the implication of quality is scarce (Chadli et al. 2016). The strong organizational leadership in GSD requires communication and coordination among teams in different locations and the influence of organizational structure affects the quality of software products (Brooks, 1995). Nagappan, Murphy, and Basili (2008) reported that GSD projects fail mainly because of poor communication, coordination, and lack of trust with the key stakeholders; other success and failure factors of GSD are cultures, geographical distance, and time zones difference. GSD leaders confront dynamic situations that negatively impact software product quality.

The GSD setting is characterized by periods of distribution between geographically dispersed sites and handover of project responsibilities between sites. Jabangwe, Borstle, and Petersen (2014) noted that few empirical studies have reported on GSD challenges on an aspect of the severity of the effect on quality. In their research, Jabangwe et al. applied the case study exploratory approach to the examination of change in quality over time on multiple releases of products in the GSD environment. Cost and quality are of high priority in the software industry. Cost is addressed by searching for locations with low cost of labour while the research work on evaluation and prediction of quality in a global environment is still immature (K. Khan, Khan, Aamir, Zulfikar, & Khan, 2013).

Customer satisfaction is the main concern of software developers, as companies strive to build economical software that is within budget and timely. Meeting customer requirements would prevent low software product quality that could cause project delays, over budgets, and reliability issues. A quality requirement, either functional or nonfunctional requirements, could constrain software development. The lack of software quality implementation would increase cost and lead to; loss of market share, extensive reconditioning, product rejection, and customer dissatisfaction. Shen et al. (2018) concluded that guaranteed software quality is achievable by software quality level, quality evaluation, and check procedure management.

The use of software has become almost compulsory, because of the high demand for software used in every facet of human life (Saleem, Mathrani, & Taskin, 2019). Most business organizations use information technology (IT) to gain competitive advantages, operational efficiency, productivity, and quality of service (John et al., 2016). Despite the rapid increase in software demand and software firms, software engineering is relatively new with respect to the design, development, and maintenance stages (Gheorghe, 2015). The adoption of GSD at different distributed site locations within a country, different countries, and different time zones comes with numerous benefits and challenges, such as quality. In this study, common understanding stemmed from what leaders in GSD organizations in Canada experienced to mitigate low software product quality, meeting customer satisfaction goals. GSD standards on quality provide valuable inputs to help practitioners make better decisions about the management of future projects (Gheorghe, 2015).

Problem Statement

The general problem addressed in this research was the organizational leaders' management of GSD software product quality challenges, that adversely impact customer satisfaction. The rapid growth of software development complexities constitutes problems of identification, evaluation, and measurement of software quality (Miguel et al., 2014). The software defects and failures have immense negative impacts on the cost of realizing business objectives in reputable organizations and some major world economies. The National Institute of Standards and Technology (as cited by Hossain, 2018) reported that the costs of software failure in the U.S. economy is approximately $60 billion per annum or 0.6% of the gross domestic product. The lack of customer satisfaction, reliability and maintainability affects both the software development organizations and software end-users such as; governments, institutions, companies, and individuals (Hussain, Farid, & Mumtaz, 2019).

The specific management problem addressed in this research was a lack of common understanding among GSD organizational leaders that, according to Yaseen, Ali, and Ullah (2016) have led to low software product quality and not achieving customer satisfaction. Software developers used about 80% of developing costs on identifying defects and performing debugging (Rustambekovicha, Gulyamovb, Usmanovac, & Mirzaevd, 2017). Research studies on GSD challenges and how they influence the software product quality are few, narrowly viewed, and under-researched with a limited empirical approach (Jabangwe et al., 2014; Niazi, Mahmood, Alshayeb, & Hroub, 2016; Wickramaarachchi & Lai, 2016). Chadli et al. (2016) noted the absence of mature research and an all-inclusive quality model for evaluating and measuring the impact of defects on software quality products. Prior studies did not lead to an overall view of the method of practice of systematic software development but entailed single development facets that lacked continuity (Kuhrmann & Fernández, 2015). GSD organizations encounter challenges, such as communication, socio-cultural factors, time zones, requirements and regulations, and technical skills that affect software product quality (Ali & Lai, 2017).

Purpose of the Study

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding, shared by managers in Canadian GSD organizations, of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction. The focus of this research study was on managers and supervisors in Canadian GSD organizations, with data collection from interviews. The total population was leaders from approximately 200 top Canadian GSD companies. The research sample was 30 managers and supervisors from six GSD organizations, located in six different Canadian provinces. The purposeful sampling selection criteria for the participants included a minimum of 6 years of experience in GSD with relevant competency standard certifications. Detailed participant information reflected a true representation of the population. The collected data represented the multiple views of how GSD organizational leaders managed software product quality in Canada. Additional information sources included documents from government reports, software institutions, associations, and journals on GSD. The findings from these sources provided valuable information to help bridge the gap in the literature on software quality of GSD and help GSD organizational leaders make informed decisions.

Research Question

RQ: What are the common understandings of what GSD organizational leaders need to meet software product quality in Canadian GSD organizations?

Conceptual Framework

The conceptual framework for this case study encompassed the quality management principles originally developed by Deming (1986). According to Deming, GSD organizational leaders would create the constancy of purpose for strong leadership, new philosophy, building quality over inspection, training, continuous improvement of processes, and products that meet customer satisfaction. The use of Deming’s theory of quality management (QM) was appropriate for this study, because it aligned with the view that GSD leaders make the most logical decisions based on knowledge of quality management concepts that could improve the software quality processes meeting customer expectations. Anderson, Rungtusanatham and Schroeder (1994) noted that the 14 principles of Deming's quality management method guide organizational behaviour and the practice of quality management.

The impact of Deming's theory and practice of quality management in organizations worldwide existed for a half century (Marchewka, 2007). There is empirical support for the effectiveness of the Deming's management method (Anderson, Rungtusanathan, Schroeder, & Devaraj, 1995). Anderson et al. (1994) developed the first underlying quality management model to explain the effectiveness of Deming's management theory, based on Deming's ideas and practice. Douglas and Fredenall (2004) analyzed Deming's management model in the organization's total quality in services. Marchewka (2007) verified the application of Deming's management method in IT projects.

Miles and Huberman (1994) deduced that a conceptual framework could be either graphical or in a narrative form, encompassing things one would investigate. The structure of ideas can be a depiction of the researcher's picture of the working theory and the current version of the researcher's domain (Maxwell, 1996; Miles & Huberman 1984). Figure 1 shows my self-developed conceptual framework for this study and the linked relationship of concepts. This structure of ideas is a constructed paradigm that helped shape the study, clarify research problems, align processes, derive new insights, and provide answers to the research questions.

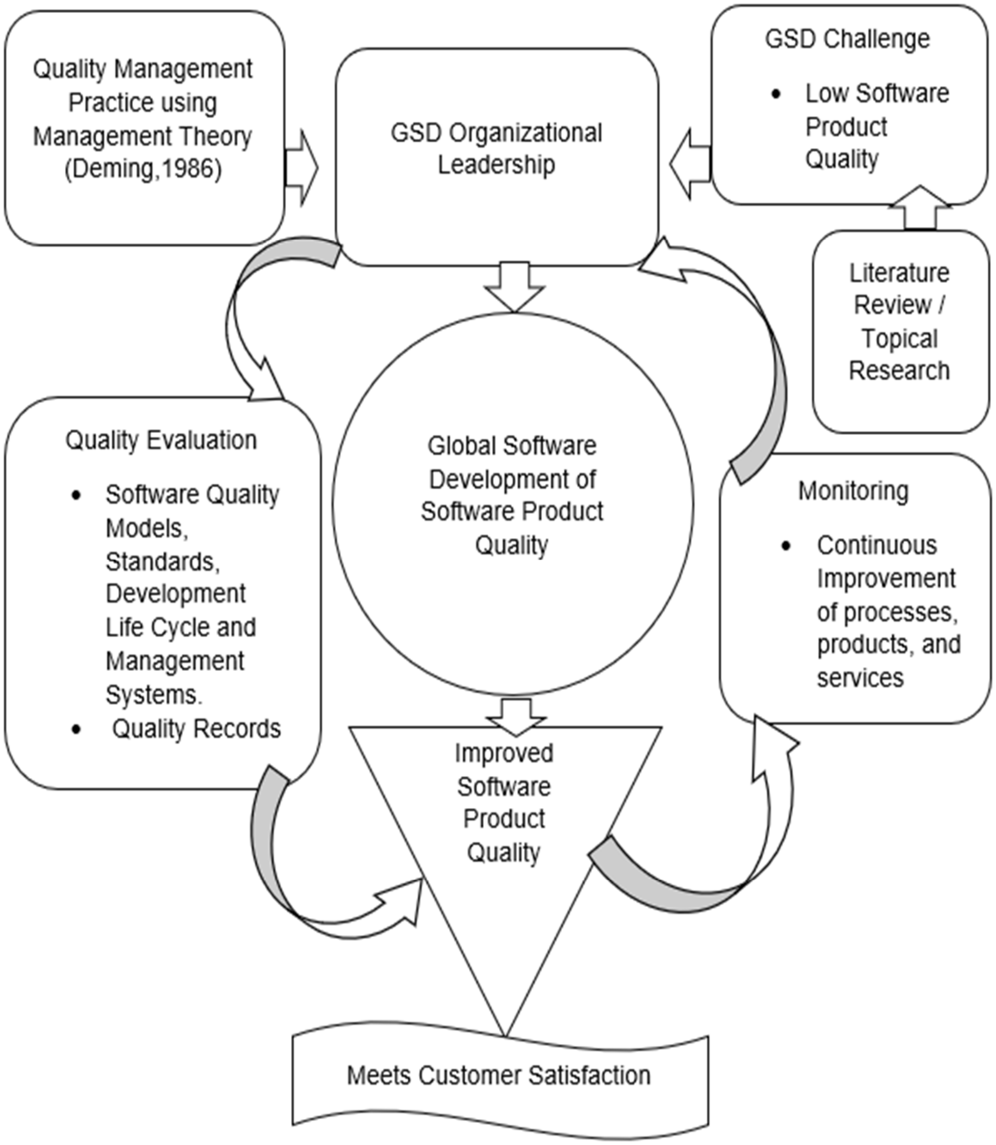
Quality Management Practice using Management Theory (Deming,1986)
GSD Organizational Leadership
GSD Challenge
• Low Software Product Quality
Literature Review / Topical Research
Global Software Development of Software Product Quality
Quality Evaluation
• Software Quality Models, Standards, Development Life Cycle and Management Systems.
• Quality Records
Monitoring
• Continuous Improvement of processes, products, and services
Improved Software Product Quality
Meets Customer Satisfaction

Figure 1. Conceptual framework flow chart for this study, developed by self; the GSD concepts, ideas, and linked relationships include GSD organizational leadership, low software product quality, the challenge in literature and the use of Deming’s (1986) quality management principles.

The intellectual goal of the conceptual framework is to identify GSD challenge of low software product quality in the organization. This peculiar challenge, confirmed through existing studies in the literature and topical research, is empirical in nature. The GSD organizational leadership derived the process of software development using Deming's management principles. Incorporated in this framework were processes of quality evaluations, continuous monitoring that may improve software quality, and meeting customer satisfaction. Bergmane et al. (2017) asserted that attaining high software quality requires continuous process monitoring and development. The conceptual framework may be subjected to further review, possibly modified after research findings which assures the trustworthiness of the research work (Goetz & LeCompte, 1984; Guba & Lincoln, 1989). A conceptual framework is integrative and evolving (Ravitch & Riggan, 2017).

Deming's quality management theory and Deming's explanation quality management model by Anderson et al. (1994) aligns with this study. Through the lens of Deming's quality management theory, I unveiled the quality management strategies that the GSD leaders in the organizations used to identify GSD management problems of low software quality products and to effectively evaluate quality to predict software product quality. Additionally, I explored how organizational leaders implement Deming's continuous process improvements to enhance customer satisfaction.

Nature of the Study

Researchers used the qualitative method when they are engaged in an in-depth interview that will reveal answers to questions, experience, and meaning mainly from the participant perspective (Hammarberg, Kirkman, & Lacey, 2016). The design of this study was an exploratory case study, involving analysis of data, such as from in-depth interviews, which led to an understanding of the details and richness of information about a phenomenon. Stake (1995) described the characteristics of qualitative research as empirical, naturalistic, holistic, open-ended, interpretative, neutral, contextual, and inductive. Use of the qualitative research method helps to answer questions about experience, meaning, context, and participants' perspective (Maxwell, 1996). Hammarberg et al. (2016) asserted that nonnumerical data collection in qualitative research is characterized by a proper organization, systematic process, textual, and visualization. The data for analyses were from semistructured, flexible, in-depth interviews, audioconferencing, telephone, online, and document review. The qualitative method was more appropriate because in-depth interviews led to the collection of valuable data suitable for analysis pertaining to the design and improvement of a product that meets customers' needs. In this case study, qualitative data represented rich, in-depth insights into participants' experiences and behaviors, which would not have been possible in a quantitative method and design.

Researchers have implemented both basic and applied research using qualitative, quantitative, or mixed methods (Bentley, Gulbrandsen, & Kyvik, 2015). The quantitative and mixed methods were not suitable for this study because they involve hypotheses testing, numerical analysis, or comparing relationships among variables, which were not necessary in this study. The quantitative research method involves statistical analysis, counting, and measurement (Gillham, 2010). The qualitative method was preferable for this study because of the advantage of an in-depth evaluation of data, flexibility, the economy of the design, and the accessibility of participants and data collection. The quantitative method and mixed method were not appropriate for this study because quantitative data are typically narrow without rich detail, with datasets that could be ostensible or sparse.

A case study design is an in-depth research strategy used to investigate contemporary patterns with real-world contexts when boundaries between situations are not clear (Yin, 2017). Researchers often adopt a case study design approach because of valuable analytical rigor leading to a holistic understanding of the situations (Merriam, 1988). The case study design was appropriate for this research because the use of the exploratory study design helped to expose meaningful patterns derived from data. According to N. Chan, Walker, and Gleaves (2015), the phenomenological approach is not just about individuals but the essence and insights from a phenomenon. Phenomenological design was not a choice for this study because I could not answer the research questions by exploring the essence and meanings of unique experiences. The grounded theory is a qualitative research design, deriving theory from data to further explain a phenomenon (Chong & Yeo, 2015). This approach was not appropriate for this study because the aim of this research was not to discover a theory from collected and analyzed data. A better understanding of a specific phenomenon and a direct strategy can stem from uncovering rich and vivid explanations derived from the qualitative case study design (Cronin, 2014).

The incorporation of multiple sources of data collection such as interviews and reviews of the literature helps to ensure data quality and credibility. Organizing, describing, classifying, and interpreting data involves codes and themes. The research data were from 30 participants who were GSD leaders in different companies located in six Canadian provinces. Using the purposeful sampling technique to identify and recruit prospective participants led to a sample of GSD leaders from five different companies. Analysis of data from semistructured interviews with open-ended questions answered by leaders in GSD organizations led to an understanding of a situation from different perspectives of interest. Answers to the research questions were possible to find through the qualitative exploratory multiple case study design.

Definitions

Distributed software development in knowledge sharing: The act of understanding the knowledge required in software development leads to knowledge sharing among team members, reducing the problem of knowledge vaporization and documentation (Waheed, Hamid, Jhanjhi, Humayun, & Malik, 2019).

Empirical evaluation of software quality assurance practices: Software quality evaluation deduced as ensuring proper documentation, monitoring, and initiating implementation of quality control of the software development life cycle (Sowunmi et al., 2018).

Regional and geographically located software development: This is a distributed software development technique known as outsourcing method, where people from different cultural backgrounds and different locations remotely contribute their skills and knowledge as a team to the development of software (Haq, Raza, Zia, & Khan, 2011).

Requirements management: The process of managing the dynamic change requirement, occurring throughout the software developmental life, is necessary for the high quality of software projects and meeting customer expectations (Akbar et al., 2019).

Software defects root causes: Software quality defects stem from the root causes of software failure, such as human errors, complex codes, time pressures, and the complexity of infrastructures and advancing new technologies (Bergmane et al., 2017).

Software outsourcing partnership: An emerging strategy with a successfully outsourced team that may be convertible to outsourcing partnerships, with peculiar challenges or barriers, like the traditional GSD method (Ali et al., 2018).

Software quality: Quality is the degree of the in-built set of characteristics that meet the standard requirements of a product (Project Management Institute [PMI], 2018), which stems from the process of designing software to meet delivery, cost, and quality requirements (Alebebisat et al., 2018).

Software quality for system design: The acts of designing software products with quality in mind to meet specific business processes and standards requirements (Stefani, 2018).

Tailoring of software development stages in Agile processes: Standard practice technique in software development to customized or adapt a process that meets the requirement, eliminating unwanted processes in order to improve quality and enhance reusability (Akbar, 2019).

Usability evaluation method: A method to measure the success of the software characteristics as required by ISO, other than the heuristic evaluation and questionnaire method, encompasses usability evaluation methods (Wahyuningrum & Akhari, 2017).

Assumptions

Researchers identify assumptions in a research study to understand possible restrictions that exist in a research process (Madsen, 2013). These restrictions could be resources, policies, theory, phenomenon, and human behavioural effects. The definition of assumptions affects data evaluation and research findings (Foss & Hallberg, 2014). The first assumption for this study was that the participants were experienced and knowledgeable enough to provide appropriate answers to research questions. My second assumption was that the organizations selected for the study represented the generality of the research population in Canada. The third assumption was that accessed organizational software quality records depicted the organizations' quality management system deliverables.

Scope and Delimitations

Purposeful sampling culminated in a sample of 30 participants from six different Canadian GSD organizations. I asked open-ended interview questions in a semistructured interview format to generate data and thereby gain an understanding of how the GSD leaders manage the quality of software products in their organizations. Related data were from multiple sources of evidence such as the review of quality records, related documents, and GSD literature. I collected, combined, and analyzed data to derive a better understanding of the different contexts of the quality of GSD. The findings added to the knowledge that GSD managers can use to make informed decisions that could mitigate GSD evolving challenges, improve quality processes, and ensure high software product quality that leads to customer satisfaction.

Delimitation is the boundary or definition of the limits of investigation (Denzin & Lincoln, 2011). The perspectives of software product quality managers were paramount in this study to establish a common understanding of how to mitigate low software quality, as understood by GSD organizational leaders. The research scope was delimitated to participants with a minimum of 6 years of experience in GSD in Canada. The expectation was that the 30 GSD organizational leaders demonstrated integrity, responsibility commitment, accountability, creativity, competence, and experience. The managers who did not self-report enough experience, knowledge, and competency were excluded. The precondition of the study was to exclude the participants who were not involved with GSD, though they may have had knowledge of in-house software development.

Limitations

The limitations in a research process are shortcomings beyond the control of the researcher which may impact the outcome of the study (Silverman, 2013). The potential barriers for this study were possible participants' reluctance to accept interview offers because of the current trend of privacy and security concerns in the software development industry. Another limitation was the difficulty in accessing quality records for review, due to organizational document restriction policies. The approach to overcome these challenges was to follow research ethical guidelines and adopt the use of multiple sources of information. The multiple sources of data collection not only validate but also provide stronger research evidence (Patton, 2015; Stake, 1995).

Significance of the Study

The findings from this research contributed to the understanding of quality management in GSD, interpreted in the light of a results-oriented conceptual framework. This study was unique because the research method was an in-depth, broad, holistic investigative approach of a contemporary phenomenon in a real context. According to Yin (2017), the quest for an understanding of a real-world context would impel a case study with contextual conditions that are unique to a case being examined. In this research study, the results from the research added valuable information that could be useful to leaders who aim to solve the problem of GSD challenges on software product quality. Minimizing the adverse effects of GSD quality failures directly and indirectly improves software public safety, economic stability, cost-effectiveness, and competitive advantage.

Significance to Practice

The revelation of a common understanding of the needs pertaining to best management practices helps to improve software quality, effectiveness, performance, process, training, staff empowerment, and new technology adoption. The common understanding that emerged as a result of this study can enable managers to act on insights that could foster organizational continuous improvement of the GSD software product quality production process. The results from this study included more knowledge and a better understanding of GSD software quality challenges and how to prevent them.

Changes to practices, based on insights from this study, could enhance economic growth in GSD organizations and the international business software trade, because of the possible benefits of new ideas from evolving software quality management in GSD. The prevailing challenges of software defects in public and government facilities could lessen from the application of recommendations derived from the findings from this study, enabling software development organizational leaders to make informed management decisions. The high cost of fixing defects in the software resulted in the termination of many software projects and the closure of many companies (Hossain, 2018). According to Bergmane et al. (2017), studies show that 80% of web site functionality failures are due to software defects and human errors. This study culminated in insights into software quality and recommendations about ways to mitigate challenges and enhance customer satisfaction.

Significance to Theory

This study resulted in valuable empirical insights and solution models to enable researchers to continue further research on outsourced software quality product, thereby contributing to the GSD body of knowledge. I applied Deming's management theory to investigate management principles applied by organization leaders in the GSD environment. Simultaneously, I explored the common understandings to help solve the problems of low software quality. The inclusion of Deming's management principles and related software quality models used by GSD organizational leaders expanded theoretical applications and led to suggestions for future GSD research. The use of a meticulously documented multiple case study design led to trustworthy insights into the management of software product quality in organizations, thus offering reasonable reliable information about maximizing software development time, reducing defects, and realizing cost saving. This study added to the body of knowledge for the evolving GSD approach over the traditional method of software development.

Significance to Social Change

The world currently depends on software systems that enhance cars, electronics, smartphones, and intelligent appliances in homes and organizations. The industry stakeholders in software functionality are increasingly higher in number today. The aviation, automotive, and financial sectors use software code for their operations. Poor software quality would result in disastrous consequences such as threats to public safety, business loss, and huge cost implications. This study led to the discovery of ways to improve software product quality in GSD, enhancing public safety and global economic stability. Software failures can have huge negative cost impacts on the U.S. economy. Over $23 billion annual savings could result from using effective testing procedures, early identification of defects, and quick removal of defects (Hossain, 2018). This study culminated in a common understanding among Canadian GSD leaders applicable to improve the reliability of software, predict software quality, and increase software life expectancy. The complexities of software require a common understanding among GSD organizational leaders about how to manage GSD.

Summary and Transition

This chapter is an overview of a quality management problem study on low software product quality in the GSD environment. It contains how GSD organizational leadership may adopt Deming's management theory principles to solve the challenges of software quality in organizations in Canada. There are few existing studies of GSD software product quality. Despite the fast-evolving nature of GSD and adoptions of various QM techniques by organizations, there have been unanswered questions about what GSD organizational leaders need to meet software product quality in Canadian GSD organizations. This research was a qualitative exploratory multiple case study to derive a common understanding of what GSD organizational leaders need to meet software product quality in Canadian GSD organizations.

Chapter 2 contains the explanation of the evolution of GSD from the traditional methods of software development to the GSD approach, the perceived benefits, and the prevailing challenges in GSD organizations. The discussion of the state-of-the-art of GSD and software product quality in organizations follow, leading to information about the organizational methodology of quality management techniques and software quality models that could predict software product quality. The chapter includes details about the relationship between this study and the conceptual framework and the similarity and dissimilarity of this research to previous studies, including the uniqueness of the work.

Chapter 2: Literature Review

Introduction

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding, shared by managers in Canadian GSD organizations, of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction. This thorough investigation of what GSD organizational leaders need to mitigate the management problem of low software product quality led to findings to help enhance customer satisfaction. The incessant and most challenging software failure in GSD is attributable to a poor-quality culture in software engineering (Usman & Khan, 2018). Quality is the degree of the in-built set of characteristics that meet the standard requirements of a product (PMI, 2018). GSD process is classified as offshore software development with peculiar critical characteristics in a distributed environment that are different from a collocated or in-house software development (Chadli et al. 2016; Rehman & Khan, 2014). The dynamic and steadily advancing GSD trend comes with flurries of benefits, including time savings, cost-effectiveness, and competitive advantages over the traditional method of software development.

All other identified GSD challenges are believed to negatively impact software quality product if not properly managed. Ryan (2016) noted that the most effective software development approaches are characterized by challenges. The outsourced method is referred to as breaking barriers of cultural boundaries and geographical distances but confronted with numerous challenges (Usman & Khan, 2018). The software quality is the main challenge of GSD that also qualify software products for customer acceptance (Rehman & Khan, 2014; Usman & Khan, 2018). This chapter provides an overview of previous and current scholarly research work in fields pertaining to organizational challenges of GSD of software product quality. There are six major headings in the literature review: relationship to the conceptual framework of the study, origin of GSD, the evolution of GSD, benefit of GSD, the state of the art of GSD, and the gap in the literature.

Literature Search Strategy

The adopted literature review approach is an intensive investigation of research published in peer-reviewed articles and scholarly journals, and by institutions and organizations in industry and government reports, conference proceedings, and books. To find relevant information, I used Internet database search engines, such as Google scholar, Walden University Library database, EBSCOhost, ProQuest, ACM Digital Library, Journals IEEE Xplore, Science Direct, Wiley Inter-Science, Elsevier Science Direct, AIS e-library, and Springer Link. These information sources led to the comprehensive evaluation of the challenges of GSD in organizations and impacts on software product quality. More insights came from the study of past GSD approaches, present practices, and suggested future studies. The initial search produced about 3,000 results from scholarly sources that related to GSD and software development topics.

Some of the searched terms used to locate articles about GSD challenges and software product quality included *global software development and low software product quality, distributed software development and software product quality, global software development challenges and software quality, organization and global software* *development quality challenges, global software development organizational leaders,* and *software quality management.* I primarily searched peer-reviewed scholarly literature, using Google Scholar and Walden Library databases, with a focus on literature within 5 years of publication. I also examined research articles published from 2000 to 2014, because of the need for GSD scholarly foundational knowledge and support. Older scholarly literature reviewed was minimal for the purpose of tracking GSD history and evolution of the subject. I included 145 sources for literature review, of which 130 are from peer-reviewed journal articles, with 125 literature sources published within 5 years of the expected date of the publication.

Conceptual Framework

The conceptual framework encompasses certain characteristics for a study, including a series of sequences of ideas and logical propositions, with the purpose to ground the study, and demonstrate appropriate technique, importance, and rigour (Ravitch & Riggan, 2017). The expression of the main ideas of the study and their relationships could be in a narrative or graphical form (Miles & Huberman, 1994). This study aligned with the conceptual framework, as it was a portrayal of an argument of relevance to answer the research questions. The adoption of a qualitative case study research design was consistent with the context and goal of the study. The research data collected through interviews with experienced GSD organizational leaders were enough for the exploration of the phenomenon in the depth required to answer the research questions. The effective data analysis and evaluation of results led to answers to the research questions.

The research management argument and significance are from existing literature. A critical aspect was that grounding of the case study was in the theory of QM developed by Deming (1986). GSD organizational leaders confront many challenges, but the research was on the specific problem of low software product quality in a distributed software development environment. The study revolved around GSD organizational managers and their common understanding to mitigate the management problem of low software product quality to enhance customer satisfaction. The main concepts of the framework included GSD leaders managing organizational systems using Deming's 14 management principles and the challenge of low software product quality, with the evaluation of information on quality prediction through the organizations' quality models, standards, and records. The other perspective was to discover how GSD organizational leaders manage software quality by monitoring and engaging in improvement processes. The final concept was to explore how GSD organizational leaders improve software products and meet customer satisfaction expectations.

Literature Related to the Different Methodologies

Theory of quality management was derived to describe the effect of the Deming management method in practice. The Anderson et al. (1994) used the synthesis of Deming's literature and industry practices, gathering expert information using the Delphi technique. The finding included formulated processes and an explanation underlying Deming's theory of QM. The path analytic model of quality management was useful to explore the empirical strength of relationships in the theory and reasonable supports were observable in a manufacturing plant (Anderson et al., 1995). Douglas and Fredenall

(2004) adopted a model developed by Anderson et al. (1994) to analyze quality in service using Deming's management model. Douglas and Fredenall confirmed Deming's management philosophy on organizational leadership as ubiquitous to the success in the quality improvement process in any organization. Marchewka (2007) investigated the Deming management method in IT Projects with 63 IT professionals to provide empirical support for using the Deming management method (Marchewka, 2007). The author reported cogent evidence with strong support for using Deming management concepts in IT project quality management (Marchewka, 2007).

The true concept of QM is often misunderstood in organizations because people have different meanings for QM (Barouch &Ponsignon, 2016). This misconception gave rise to the Deming philosophy of organizational transformation, interrelated as profound knowledge, including the system, variation, theory, and psychology (Moen & Norman, 2016). An epistemological framework using three paradigms includes systematic, pragmatic and constructivist to demonstrate the grounding of QM theory in ideas (Barouch & Ponsignon, 2016). The Barouch &Ponsignon (2016) inferred that the framework will help understand QM, both in theory and practice. The methodological approach was a synthesis of theoretical frameworks for QM using six core areas of QM. The Deming based theory (Anderson et al., 1994), compares with other methods: ISO 9000 requirements (2015), a theoretical model for TQM (Dahlgaard-Park & Dahlgaard, 2015) and generic conceptual framework for QM, and TQM (Barouch & Ponsignon, 2016). The six popular QM concepts in the literature are customer satisfaction, visionary leadership, employee fulfillment, process management, continuous improvement, and internal and external cooperation (Barouch & Ponsignon, 2016). Barouch and Ponsignon (2016) concluded that the combination of systematic, pragmatic, and constructivist strategies may provide managers with a proper understanding of the epistemological foundation of QM and direction for implementation.

Researchers formulated narrative concepts and techniques on how to implement TQM, which supports Deming's quality management. Alhassan, Alzahrani, and AbdulAziz (2017) explained the similarities between the software product and process development, quality measure in SDLC, and Deming's 14 principles in software development. Product development life cycle is a requirement involved in analysis, design, building prototypes, testing prototypes, piloting release, full release, and follow up (Alamri & Azziz, 2016; Iyer, 2018).The early software development approach was a code-and-fix-approach with a common characteristic of the repeated cycle until project completion (Yilmaz & Chatterjee,1997).

SDLC is typically a waterfall model for software development (Alhassan et al., 2017). This process includes requirement planning, definition, design, coding, testing, and maintenance (Alamri & Azziz, 2016). The limitations of the waterfall model are the absence of advance planning, insufficient interaction, and collaboration (Yilmaz & Chatterjee, 1997). Another software development approach is the use of an iterative model where the entire process repeats until a product emerges (Yilmaz & Chatterjee, 1997). The use of Deming's management method PDCA cycle (Plan, Do, Check, and Act) is useful in software development to achieve software quality (Alhassan et al., 2017).

The Deming's 14 principles were explored to create a relationship between the different concepts. Agrawal (2019) developed the strategic interpretative structural modelling (ISM) and MICMAC analysis helped to build an understanding of the relationships for both dependencies and driving power. The inputs from academia and industry were useful to interpret the interactions between concepts, and ideas. The finding shows that the Deming 14 principles are akin to transformation, organizational systems, and the operational requirement for TQM (Agrawal, 2019). The use of the ISM and MICMAC could be helpful to improve customer satisfaction, efficiency, and market shares for the service and manufacturing industry (Agrawal, 2019). The new Deming's lens is versatile and may be applicable in industry, government, and education (Moen & Norman, 2016). The traditional Deming's cycle PDCA may help to solve the problem of low quality and continuous improvement of processes in the agricultural sector (Dudin, Smirnova, Vysotskaya, Frolova, &Vilkova, 2017). Liao (2019) noted applications of the Deming management principles in heavy industry enterprise in China to verify the feasibility transformation to address the issue of cost, low efficiency, and quality.

The similarities and dissimilarities of existing research work and current research. There were similarities between my research and previous research studies. The synthesis of literature and ideas from practitioners led to valuable findings (Douglas & Fredenall, 2004). The reliance on literature and industry perspectives was consistent with the current research where GSD organizational leaders were the participants and research problems emerged from peer-reviewed published literature. Douglas and Fredenall (2004) noted that organizational leadership ensures the success of process improvement according to Deming’s principles. In this research work, GSD organizational leaders were integral to the development of software because of their involvement with all aspects of developmental stages. Marchewka (2007) and Anderson et al. (1995) showed the empirical strength of relationships and theory to confirm the reasonableness of the Deming management method. In this study, there was a comparison of the empirical outcome with the Deming management approach. Most of the researchers used models and framework to further understand the phenomenon, which is like this study. Agrawal (2019) explored the ISM and MICMAC models to understand the relationship between concepts and learned how customer satisfaction, costs, and quality could be improved.

The dissimilarities of this study with other studies was the premise that organizational leadership could institutionalize quality, without which there might be no meaningful progress (Moen & Norman, 2016; Yilmaz & Chatterjee, 1997). Exploring the common understanding of GSD organizational leaders could help ensure software product quality and enhance customer satisfaction. In the conceptual framework for this study GSD organizational leadership was depicted as having interrelationships with all components of distributed software development. From the inception of the discovery of the management problem of low software quality in the literature, the use of Deming’s 14 principles of management became top management responsibilities. Deming’s 14 points are characteristics of organizational transformation for quality assurance (Yilmaz & Chatterjee, 1997). This transformation, as a system based on profound knowledge, may help GSD organizational leaders share and adopt meaningful ideas.

The uniqueness of this research study to previous studies. The organization systems encompass the interrelationship between components that enable communication, coordination and cooperation among teams (Yilmaz & Chatterjee, 1997). Managing a system impacts the quality of software and stakeholders. In this study, there was an investigation of the application of Deming's 14 management principles, evaluation of quality, and processes improvement by GSD organizational leaders. Anderson et al. (1994) compared models with QM main components. Alhassan et al. (2017) described the implementation of Deming's quality management and TQM process in a software development setting. This study was on system view, which most software development technique does not consider. Agile cannot be a stand-alone quality management system, because it does not have a comprehensive system view (Krehbiel & Miller, 2018). Deming was a strong advocate of system management through the profound knowledge theory (Krehbiel & Miller, 2018). This theory encompasses Deming's 14 principles of quality management (Yilmaz & Chatterjee, 1997). In contemporary software development, the agile technique sometimes combines with Scrum, Lean, or Kanban to ensure flexible and rapid response to change (Krehbiel & Miller, 2018). This study was unique because of the view that the application of Deming profound knowledge, including system thinking, variation evaluation, epistemology, and psychology was the most holistic way to improve quality.

There is no empirical research that has tested Deming's concepts on quality performance or investigates their interrelationship, probably due to the lack of existing reliable and validated measurement of Deming's 14 concepts (Tamimi, Gershon, &

Curall, 1995). Tamimi et al. (1995) developed operational measures using data from 173 business managers as a measurement instrument to evaluate a model of Deming's management method initially developed and tested by Anderson et al. (1994). Fisher, Elrod, and Mehta (2011) replicated the instrument for Deming's 14 points, using data from 100 manufacturing and service companies in United States of America and Canada. In their investigation for validation, improvement, and operationalization of the scales by Tamimi et al. for management performance, there was evidence that the Deming's 14 points were strongly consistent with current data, being reliable and valid (Fisher et al., 2011). Business managers could use a guided method for improvement of implementations, market share, efficiency and competitive edge by employing Deming's 14 points of quality management (Fisher et al., 2011). Tamimi et al. affirmed that psychometric properties of the measures are reliable and valid and can be adopted as valuable tools for practitioners and researchers. Managers can use Deming's framework as a benchmark for self-assessment and evaluation of their TQM culture (Fisher, et al 2011; Tamimi, et al 1995).

In this study, the data collection protocol included open-ended interview questions based on the Tamimi et al. (1995) validated instrument. Deming's 14 points were the basis of an in-depth exploratory study of the common knowledge among managers in the GSD organizations. The interview protocol questions also pertained to the implementation of SDLC and international standard on software product quality ISO/IEC 25000 series to examine how GSD managers predict software product quality. The open-ended qualitative questions did not involve the use of numerical numbers or calculations.

Denzin and Lincoln (2011) asserted that in a qualitative study, individual perceptions cannot be generalized. Trustworthiness was enhanced by a comparison of research findings with existing theories, literature, and related reports. The use of the interview protocol question based on Tamimi's validated instrument for Deming's 14 principles helped to elicit perceptions of experiences with quality in the participants' organizations. The focus was on areas of priorities for quality improvement, effective allocation of resources, top management commitment, and various aspects of management behavioral practices.

Literature Review

Origin and Success of Global Software Development

Software engineering is traced back to the 1950s when the art of computer programming was evolving in the industries and institutions. According to Cusick, Prasad, and Tepfenlart (2008), the act of computing started as a global industry beginning with the United States and United Kingdom International multi-sites software development in the 1960s (Carmel, 1999). The development challenges and shortcomings brought about the aspect of software engineering in the NATO-sponsored conference of 1968 in Garmisch-Partenkirchen, Germany (Ebert, 2018). Software development became a global affair after the ICT revolution of the 1990s (Carmel, 1999). The modern aspect of GSD started in the 1990s (A.A.Khan, Keung, Amin, &Al-Wadud, 2017). As at the year 2000, the National Association of Software Service Companies (NASSCOM) in India reported that 200 out of the Fortune 500 companies depended on GSD firms (Denny, Mani, Nadella, Swaminathan, & Samadal 2008). Akbar, Khan, and Adnan

(2020) confirmed that GSD started in the late 1990s and mid-2000s with a popular acceptance from software companies. The increasing need for productivity, schedule, and more profit-making at the expense of quality caused a drastic deterioration of software quality in organizations.

The complexity of software requirements became more recognized as the market size grew bigger and many facets of life enhanced with software products. The adoption of cheap software design and programs produces poor quality software requiring the need for global efforts to produce reliable, economical and safe software for human benefits. The contemporary approach to software development is In-house and GSD. The software product classification is custom-made or commercial off-the-shelf that is programmed based (Musa & Alkhteeb, 2013). Tamimi, Alghamdi and Yaseen (2019) noted that custom-made software is developed by in-house companies with peculiar requirement characteristics. The software global development easily manifests software quality problems requiring alignment with all stakeholders (Barney et al., 2013).

Rapid Growth and Adoption of Global Software Development

The GSD trend is now mainstream in the software industry because many companies are increasingly adopting the development of software in other distributed geographical locations, irrespective of the associated barriers to software product quality. The evolving transformation of GSD is moving very fast from just a phenomenon to a paradigm due to business needs, organizational demand, and software community acceptance (Babar & Lescher, 2014). According to Denhere, Horne, and Van der Poll (2015), software companies set up subsidiaries overseas or get contracted third-party companies in regions with low economies like the Middle East to be destinations for software development projects and maintenance. Denhere et al. acknowledged other names of distributed multi-sites software development; distributed software development, GSD or collaborative software development. Babar and Lescher (2014) applied the term *Global Software Engineering* in their study on how to identify GSD challenges and provide solutions. S.U. Khan and Azeem (2014) used the term *Offshore Software Development Outsourcing* to represent GSD in the study of inter-cultural challenges in offshore outsourcing locations in India, China, Ireland, and Russia. The client organizations were identified from different countries like the United States, the United Kingdom and Japan.

The advent of the high-speed and reliable internet system, ICT, improved telecommunication systems and a low-cost distributed environment enables easy set-up and management of a global virtual team. The collocated team or face-to-face team of traditional software development is easier to set-up and managed compared to the complex outsourced GSD. At the inception of GSD, the practitioners identified certain challenges and potential risks; culture, language, religion, quality issues, different work ethics, tool compatibility, time zones, temporal distance, coordination and communication (Denhere et al., 2015). The challenges of GSD continue to unfold due to the dynamics and complexities of the international business environment. Emerging new barrier factors are the advancing new technologies, modern business implementation techniques, government policies, politics, cyber-security and regional trade interests, regulations and standardization. Empirical research results can help GSD organizational leaders to properly understand the challenges and improve the performance of the virtual team (Iftikhar, Alam, Musa, & Suud, 2018). Goltz et al. (2015) deduced that present innovations, methodologies, concepts and tools enable the current evolutional trend in software engineering. The combination of efforts and integration of approaches from different software engineering disciplines will help to meet the demand of changing functions, and quality requirements (Goltz et al., 2015).

The Benefits and Acceptance of the Global Software Development

The rapid drive for globalization makes modern organizations search for ways that are cheaper and faster to develop software that meets the quality requirements of the customer, investment stakeholders, and government (Marinho, Luna, & Beecham, 2018). GSD approach satisfies the needs of the software development investors but requiring teams of different cultures to work together with a potential barrier of communication, and coordination (Babar, & Lescher, 2014; Yue, Ahmed, Wang, & Redmiles, 2019). The spread of GSD across national and geographical boundaries increased due to available skilled resources, emerging new markets, the round-the-clock development, merger capitalization, proximity to market, and global acquisition opportunities (Yassen & Ali, 2019). Huisman and Rubin (2018) stated that software development provides innovation, learning and knowledge transfer. The common perception is that GSD multi-skilled teams use different managerial strategies, advanced technologies, and globally distributed settings to gain a high-quality level of software, reducing development costs and delivery time (C. Kahlil & Kahlil, 2019). GSD offers numerous benefits, along with challenges that impact software product quality negatively

(Hidayati, Purwandari, Budiardio, & Solichah, 2018). Researchers intensified efforts to study how to address the many challenges of GSD in cross-site modular development projects with easy access to customers and highly skilled expertise at low prices. The evolution of GSD procedure is both profitable to organizations and business communities and beneficial to the stakeholders (Babar & Lescher, 2014).

Contemporary Issues in Software Development and Software Product Quality

GSD organizational leaders in Canadian organizations can act on a common understanding of what is needed to mitigate the management problem of low software product quality. Globalization is surging, high with impacts on the economy, advancing the software development industry. In 2018, the Canadian ICT sector recorded strong growth, according to the 2018 economy assessment reports by the Innovation, Science and Economic Development (ISED) Canada. The industry structure evaluation of ICT companies in sub-sectors showed that software and computer services are 90% out of 100% in terms of size (ISED, 2018). According to ISED Canada (2018), GDP by ICT sub-sector amounted to a total of $87 billion in Canadian currency, of which software and computer services are 48%. The software and computer service also achieved a positive and fastest annual growth rate of 6%, compared to other sectors (ISED, 2018). Software quality is the act to fulfill expectations or certain characteristics that meet customer satisfaction. These quality attributes can be measured or evaluated to predict quality. The common attributes of a quality product from a predicting model are reliability, usability, efficiency, maintainability, security, scalability, availability, efficiency, consistency, recoverability, and customizability portability (Djouab & Bari, 2016).

Software and Information Industry Association teams identified software product quality as a major reason decision-makers preferred the GSD environment (Fauzi & Sobri, 2018). GSD managers aspire to meet software quality and customer satisfaction in a complex environment with technical and socio-cultural issues. Ghanbari, Vartiainen and Siponen (2018) investigated how software organizational managers sometimes ignore recommended software development quality assurance practices that resulted in deficiencies in software quality. The quality practice omission related to organizational leaders' decisions because of constraints and market pressure demands that often lead to trade-off. Few studies were about virtual software development, representing a gap in GSD body of knowledge. The concept of quality practice is vital to sustaining software quality and minimizing the high failure rate of software development projects (Hossain, 2018. The use of the snowball technique of systematic literature review enabled easy detections of gaps. Practical interventions can emerge from future research by an in-depth exploration of the social, organizational, and economic consequences of neglecting quality practices (Ghanbari et al., 2018).

Use of Models and Frameworks in GSD

Global software users' concern for quality increased with the rise of different techniques for the development of software. One of the biggest tasks for software organizational leaders is how to measure and evaluate quality to meet the needs of the customers. After 2000, the use of models became imperative to predict quality. The use of the model is an acceptable technique to enhance quality management software products (Miguel et al., 2014). The software quality model has characteristics and relationships that help defined quality requirements and evaluation (ISO/IEC is 9126-1, 2001). In each quality characteristic, there are measurable attributes that enable evaluation of software quality (Tiwari, & Chakraborty, 2015). The quality model allowed building quality into software, defining standard coding and detection of a quality defect in the software development process (Kumar, 2017). Using models to evaluate software product quality gained popularity in contemporary times, because the quality model could help evaluate defined organizational set goals for software product quality. The act of mapping of characteristics of the intended study can lead to adding them to the software model to obtain a valuable prediction of software quality. Djouab and Bari (2016) proposed a model using e-learning characteristics based on the ISO 9126 standard. Their research outcome enabled the evaluation of existing systems. They recommended future research that is evaluation and validation of the proposed model for organizations or university environments. In the evolution of quality models, models are classified as basic models and tailored models.

The basic models are those models derived before the year 2000, whereas tailored component quality models are those developed after the year 2000 (Miguel et al., 2014). The adoption of the ISO/IEC 25010 for software product quality evaluation means the addition of more quality characteristics. Calabrese et al. (2018) identified characteristics of software security as a case study with a defined matric and results obtained from three studies where software security was the impact of software quality. In their study, they evaluated software quality using GQM (Goal, Question, and Metric), alongside ISO/IEC 25010 standards. Established was that organization can suffer dire consequences of differentiating elements of competitive advantage, corporate image, monetary losses, and high failure rates because of poor software quality (Calabrese et al., 2018).

Software development leaders strived to combine several models in order to ensure software quality. Alrawashdeh, Muhairat, and Althunibat (2013) adopted a quality model of Enterprise Resource Planning (ERP) using the ISO 9126 standard. Indicated from their research evaluation were three minimum characteristics required for creating a quality model that includes 27 sub-characteristics. The developed model was useful to determine if the ERP model would fail or succeed in higher education institutions. Polillo (2011) examined the methodology of QM in web 2.0 site environment compared with ISO/IEC 9126 and ISO 25010 for software products. The QM may be valuable for requirement gathering, defining quality, design, process development, operation, and maintenance.

Modern software business managers engaged in strategies that could enhance requirement gathering, process improvement, cost-effectiveness, and software product quality. Bourgroun, Saber, and Bouchentouf (2017) evaluated IT assessment challenges to improve software product quality and maintenance cost using Object-Oriented (OO) program model properties. Their model compared OO models with the ISO 25000 model series to reconcile internal and external software quality attributes. There is evidence that in any model, matrices, or framework to achieve software product quality, there are requirements to meet to satisfy customer expectations. According to Baquero et al. (2018), the concept of quality refers to a set of characteristics like artifacts, processes, and resources with the capacity to meet requirements or needs. Software quality model could improve by the review of literature, interviews, questionnaire data processing, observations, and weight factor (Sugiyanto, Rochimah, & Sarwosri, 2016).

Process Improvement Management

The adoption of GSD by organizations means involves the current globalization phenomenon of cultural barriers and geographical distance (A.Khan et al., 2017). The main reasons for virtual software development are cost-effectiveness, improved software product quality, and return on investment. There are Software Process Improvement (SPI) challenges encountered by software development practitioners. Researchers are working on how to improve GSD processes because of poor software quality due to distributed software development. It is imperative to investigate the effectiveness of Software Process Improvement Implementation and Management Model (SPIIMM) due to the problem of software process improvement in software development (A.A.Khan et al., 2017). A systematic literature review occurred to determine critical success factors, critical barriers, and practices of SPI. The methodology was survey questionnaires from 111 participants who were experts in SPI. Inputs from both critical success factors and critical barriers fed into the capacity maturity integrated model to ascertain organizational SPI maturity level readiness and the effectiveness of (SPIIMM). A.A.Khan, Keung, Niazi, Hussain, and Ahmad (2017) noted many challenges with SPI initiatives in GSD organizations. The authors examined the barriers that can negatively affect SPI in GSD organizations through both vendor and client perspectives using literature review and survey questionnaires to identify and validate barriers. They concluded that findings from the studies showed that identified SPI barriers can assist GSD organizations to succeed in projects.

Software quality is one of the most important factors of GSD, because it is the main determinant of customer satisfaction. The modern trend of GSD includes the implementation software process model because of software quality, task setting, establishing project duration, and ensuring continuous process improvement (Akbar et al., 2018). A new software development life cycle A-Z model introduced by Akbar et al. (2018), may help overcome the limitation of traditional methods and improve the quality of software in GSD. Further study on the improvement of SPI led to the identification of similar success factor using Fuzzy AHP method to collect responses and comparison factors (A.A.Khan, Shameem, Kumar, Hussain, &Yan, 2019). The approach depends on the effective prioritization of success factors that enable the removal of uncertainties during decision making.

The effectiveness of the existing techniques for process improvement in software development distributed setting is a concern. Bhatti and Ahsan (2017) confirmed that one of the major challenges of GSD is the process of monitoring and controlling in a software distributed environment. The authors developed a process improvement framework using grounded theory methodology with a reliable technique for validation. The results showed that the framework is valid for the GSD setting (Bhatti & Ahsan, 2017). According to Bhatti and Ahsan, GSD organizational leaders who implement process improvement may likely achieve software quality. The recommendation was the implementation of the framework distributed environment using both failure and success factors (Bhatti & Ahsan (2017). Other process improvement frameworks and models in the study included QMS, TQM, CMMI, ISO, SCRUM, Six Sigma, Lean, Control Objective for Information Technologies, People Capability Maturity Model, Organizational Project Management Maturity Model, and Portfolio Programme Project Management Maturity Model (Bhatti & Ahsan, 2017).

Requirement Management

The prioritization requirements in GSD organization are serious considerations in SDLC for the achievement of software product quality (Jayatilleke, Lai, & Reed, 2018). Engineering requirement prioritization is about grade ranking or in order of value up to the release stage (Chimagus, Selamat, Ibrahim, & Mahin, 2014). Software requirement changes frequently, and wrong choices can lead to rework, rejection, cost, and waste of time. Software errors occur from improper changes, wrong program fixing, and incorrect updates (Yi, Qi, Tan, & Roychoudhury, 2015). Change in requirement is part of every stage of the software development life cycle and can originate from customer requests, change in business, or operating environments (Jayatilleke, Lai, & Reed, 2018). Most organizations use changes analysis techniques to prioritize requirements, while others develop a framework to allocate requirements according to the project schedule. An incremental approach is preferable sometimes, to avoid pitfalls by team members. Awais (2016) used the ETVZ prioritization model that is applicable to both in-house and GSD requirement selections. Awais identified the challenges of the proposed model and areas of improvement, so that it can be more effective for GSD requirement prioritization. The methodology of the study was a literature review on various requirement prioritization techniques and their challenges; types of requirements modularized into subsections that helped in the decision-making process, as values were assigned to the main requirement hierarchy (Awais, 2016).

Software development is becoming more complex because of continuous change and a need for a requirement prioritization analysis (Jayatilleke et al., 2018). An analysis method that consisted of change by functions, identified change barriers, and dependency using a matrix was the focus in a study on requirement analysis approach. This process applied to the University course change management program, limited to the University system, not a complex environment. Future work can extend to a large database where the outcome will help decision-makers with the requirement that may avert a waste of time in the software development process (Jayatilleke et al., 2018). GSD is not a risk-free venture despite the high capability maturity level of the distributed software development industry.

The success of GSD projects depends on proper requirement engineering prioritization and management of the prevailing challenges, including socio-cultural, language difference, communication, change management, knowledge sharing, and lack of trust. There are advantages of the advanced technologies in virtual communication, computer systems, and skilled software engineers in GSD (Usmani, Hassan, & Mahmood, 2017)**.** The removal of impediments during constant requirement updates in the complete phases of the software development circle is key to having quality product quality. Usmani et al. (2017) implemented the requirement improvement model using the methodology of SLR and questionnaires to gather data from stakeholders on what challenges they faced while in the requirement process in a GSD environment. The focus was on how the GSD practitioners manage the challenges of requirement gathering for software development. The research findings were that organizational leaders are reluctant to bring in requirement engineering for fear of project failure. Requirement engineering is necessary at the early stage, using available tools to make the best out of the GSD setting (Usmani et al., 2017).

The aspect of Requirement Change Management (RCM) is a critical part of SDLC in GSD approach because change is inevitable (Anwer, Wen, Wang, & Mahmood, 2019). The stages of requirement considered in studies include elicitation, specification, management change, validation, and documentation (Ali & Lai, 2017; Anwer at al.2019; Awais, 2016). The study by Anwer et al. (2019) on requirement change management was like that of Usmani et al. (2017), which showed the previous method of validation and documentation of a requirement for in-house software development may not be effective for the GSD environment. Developed requirement graphs moved through all virtual teams for updates and validation.

Communication, Collaboration, and Integration Management

The software systems became an essential part of business transactions in the world. Communication is vital during requirement gathering at the time of coordination and control of the virtual teams in different locations. The communication risk is more glaring during requirement change management, because change occurs frequently during the phases of the SDLC. The main causes of software failure in GSD are attributable to communication, collaboration and control (A.A.Khan, Basri, Dominic, &

Amin, 2013). Software product quality issues reports are mostly when there is the engagement of outsourcing relationships, which can lead to low quality products (Agburu, Anza, &Iyortsuun, 2017). The perspective of communication risks associates with GSD, as either internal or external threats due to interaction that is beyond the boundaries of socio-culture, geographical, and temporal distance (Rehman & Khan, 2014). The quest to gain trust among GSD stakeholders in different environments is important to initiating proper integration (Arumugam & Kaliamourthy, 2016).

Collaboration and communication issues among GSD team members is a prevailing challenge for software development. Sharma, Kaur, and Kaur (2015) asserted that agile software development may be of benefit if combined with GSD because agile enables collaboration, interaction, continuous requirement gathering, cost-saving, and prompt software project delivery. The empirical research by Sharma et al. (2015) was on investigating GSD challenges in the literature and industry to develop a framework to help alleviate the GSD team communication issues. After a systematic literature review and industry study, conclusions were that the implementation of the agile method in a distributed environment will help mitigate challenges with appropriate tool support (Sharma et al., 2015). Both benefits and challenges emerged from previous research and current work. Development of a Natural Language Processing model applied to help solve the language-translation problem in GSD (Sharma et al., 2015). According to Horta et al. (2019), customer demand for a high quality of software increased the search for experienced software managers who can collaborate with their group and solve complex problems. The Stack Overflow was useful to model network and NetSCAN algorithms to locate aggregated groups (Horta et al., 2019). The final evaluation revealed high-performance experts.

A good understanding of integration failure may help with the development of an appreciation of most integration challenges at the GSD setting (Zafar et al., 2018). Researchers revealed that, at the development stage, GSD organizational leaders may overlook controlling integration which can suddenly become a problem during the system integration phase due to incompatibilities. Zafar et al. (2018) examined success factors of integration by conducting an industrial survey, review of existing literature, concept mapping, and identification of 40 taxonomy integration failure factors. The finding was that the evaluation of various failure factors taxonomy enabled the determination of the problem domain of integration at the early stage of the projects (Zafar et al., 2018). Proper integration in the distributed environment would enhance communication, coordination, and control in GSD virtual locations.

Knowledge Transfer Management

Expectations are that GSD organizational leaders comply with appropriate knowledge management practices that define how to understand and share knowledge among virtual team members. Software development is a knowledge-intensive area (Razzak, Bhuiyan, & Ahmed, 2014). Effective knowledge management will ensure software product quality and social usefulness (Ouriques, Wnuk, Gorschek, & Svensson, 2019). Knowledge consists of data and information needed by the teams to perform their tasks. The prevalent problems of collaboration and communication in a distributed software development setting hindered knowledge sharing, and documentation, and led to knowledge vaporization (Waheed et al., 2019). Additional knowledge management efforts, tools, and techniques to share information and knowledge among team members are needs (Huzita et al., 2012). The backbone of software development is effective knowledge management (Ahmed, Mansor, & Ahmad, 2017). The lack of team knowledge could lead to inconsistency, low productivity, poor documentation, and low quality of software development (Niazi et al., 2016).

GSD organizational leaders are focusing on how to effectively manage and share knowledge among team members because of the perceived accruing benefits. Knowledge resides within the team members, and affects software processes, methodologies, environment, and organizational assets (Waheed et al., 2019). Knowledge management makes easy the acts of sharing, distributing, creating, and capturing needed features that meet desired software product quality and client requirements (Ahmed et al., 2017).

Knowledge sharing enables mutual understanding and improved team effectiveness. Knowledge sharing involves all stages of the software development life cycle and release process of the finished product to the customer. Researchers and practitioners are investigating various aspects of knowledge management, such as purposes, types of knowledge, and applicable technologies to achieve high software quality for the end-users. The reuse of knowledge among team members is now gaining more attention in a distributed software development environment (Indumini & Vasanthapriyan, 2018).

Knowledge management effectiveness in the software development environment is being researched using different methods. Ahmed et al. (2017) researched the challenges of agile software development informal tacit knowledge orientation and GSD explicit knowledge formal modes of communication in an environment dispersed by time and space. Challenges in the literature based on the frequency of occurrence pertained to the agile development technique when combined with GSD. According to Ahmed et al. (2017), the most critical challenges are the ability to manage knowledge storing, knowledge sharing, knowledge transfer, and tacit knowledge. Added to further finding were knowledge creation, lack of common universal language, and documentation as minimal effects (Ahmed et al., 2017). The future knowledge process deserved verification, validation, use of specialized methodology, and matching of newly discovered challenges against identified existing challenges (Ahmed et al., 2017).

The problem of collaboration among team members is a major issue of knowledge sharing in a distributed space. Waheed et al. (2019) proposed a framework with documentation and knowledge vaporization issues as inputs then performed an evaluation on an industrial case study. Waheed et al. examined the framework with real-life scenarios; data were from interviews, analyzed using thematic analysis, and System Usability Scale (SUS) questionnaire. The research outcome included team members confirming that the knowledge sharing process improved by the implementation of the research proposed solution (Waheed et al., 2019).

The aspect of contextual information management required more attention because extra efforts in coordination and communication are needs of GSD virtual teams (Huzita et al., 2012). Areas of interest included challenges and opportunities for conceptual information management. Considered in the research was how to manage knowledge, performance, social-cultural, information storage, and data sharing in GSD (Huzita et al., 2012). The research plan included how GSD organizational leaders would understand and use a mechanism to support capturing contextual information using the paradigm of cloud computing. Physical distance in a distributed environment is a challenge to effective collaboration among team members, with an impact mostly on cost and quality (Huzita et al., 2012). Contextual information is the idea where team members know how the information came about and how to use them. GSD organizational leaders expect to collect information, store, process, and distribute to the team. A distributed software environment needs the right infrastructures and mechanisms to capture contextual information for knowledge sharing (Huzita et al., 2012). Research findings could guide the academic and industrial interests and increase GSD interest groups in Brazil.

Tasks Allocation Management

GSD leaders are constantly trying to devise new strategies to overcome challenges and risks. Planning and managing task allocation in a GSD structure is critical and challenging (Mahmood al.2017). Further compounding the complexities of task allocation is the use of inappropriate criteria for task allocation decisions at the planning stage (Filho, Pinheiro, & Albuquerque, 2015). Organizational leaders used prioritization and classification of tasks during the planning and at the developmental process of software. Task assignment is a critical management activity throughout the development life cycle of software that requires informed decisions (Filho et al., 2015). The improper allocation of tasks has great potential to impact the project schedule, cost, and quality.

Task allocation and prioritization is an important aspect of software development. Filho, Pinheiro and Albuquerque (2018) conducted a survey to evaluate task allocation in GSD projects with an agile approach. The adopted methodology was a systematic literature review on task allocation in a distributed environment and task allocation with the agile method. The research finding was that there is agility in GSD and few studies included prioritization and classification of task allocation. The authors' recommendations were the use of a multi-criteria model to support the process of prioritization and classification of task allocation to achieve team performance. GSD researchers and practitioners proposed several models and frameworks to address the complexities of task allocations for distributed software development. The lack of proper understanding of the criteria influenced task allocation in a socio-cultural impacted and geographically dispersed workforce. Filho et al. (2018) advanced the study on task allocation by using multicriteria models to assign tasks in a GSD project. The qualitative exploratory method involved extensive interviews and a set of questions. Finding were that the approach would be a guide to researchers who want to venture into future study on task allocation in distributed software development settings (Filho et al., 2018). A further recommendation was that future research may address and identify the impacts of influencing factors on task allocation. A proposal was an approach to support the perspective of task allocation in distributed software development space.

Managers of global software development are expected to be conversant with the criteria for task allocation at every stage of SDLC. Mahmood et al. (2017) applied a similar method of systematic literature review and a questionnaire survey as did Filho et al. (2018). Feedback was from 62 industry practitioners; results were no significant difference between SLR findings and questionnaires (Mahmood et al., 2017). Task size and cost were important in a decentralized GSD structure. GSD organizational leaders should strive to understand the factors influencing the criteria for task allocation management at the planning stage, so that appropriate task decision is possible (Mahmood et al., 2017). According to Mahmood et al. (2017), several techniques of task allocation proposed by practitioners include the sun-development model, cost mode, and risk-based multi-criteria optimization model.

Project Management

GSD is inclusive of client and vendor organizations, virtual teams, enabling technologies, and other resources. The quality of software could be the strongest or weakest link in the chain of the software development processes (Birgun & Altan, 2019). Hence, there is a vital need for effective project management, for meaningful transformation in GSD. Software development and use is now part of the digital business environment of every community, either through a mobile device, web, or computer system (Rajagopalan, 2014). This digital evolution also impacts Industry 4.0 structures such as robots, smart factories, cybersecurity, cloud computing and integrated system that function with software (Birgun & Altan, 2019). Contemporary software managers in organizations are increasingly changing to GSD because of the benefits of high quality and low developmental costs (Niazi et al., 2016). Management of software development processes, organizational interfaces, and interaction among geographically distributed team members requires leadership.

GSD organizational leaders must ensure effective communication, coordination, and control of the virtual team to achieve success. The main stages of the Software development life cycle in the software development process includes planning, creating, testing, and deployment (Magableh & AlSobe, 2018). A good and realistic management plan helps to effectively manage distributed software development, because of the perceived hidden costs, coordination and communication, and lack of GSD organization project management readiness (Niazi et al., 2016). Khraiwesh (2013) noted the fact that software development planning entails analysis of resources, work breakdown, timing, clear direction, established project deliverables, defined responsibilities, and identified project constraints. Proper planning and documentation in project management is a success factor criterion in the quality of software development (Denhere et al., 2015; Rajagopalan, 2014).

The project life cycle is a series of phases of a project from start to completion (PMI, 2018). Project life cycles are either predictive or adaptive. Project management is key to software development (Rajagopalan, 2014). Apart from project planning, there are other essential projects management process groups, including initiating, planning, executing, monitoring, and closing (PMI, 2018). In the PMI guide on project management, other processes are knowledge areas; these knowledge areas are 10 elements of management: integration, scope, risk, schedule, flexibility, cost, quality, resource, communication, procurement, and stakeholder (PMI, 2018). GSD organizational leaders could use these management group processes and categorized knowledge areas, depending on the project needs to initiate and complete a timely software project.

The GSD organizational leaders continually explore alternative strategies to remotely develop software because of many advantages, including nearness to market, skilled workers, all-round development because of different time zones, availability advance technologies that enhance communication, and coordination (Saleem et al., 2019). Implementation of project management processes is integral to the success of software development in the GSD environment. Some of the benefits of remote software development may be a favorable government tax policy, lower cost of development, quality, and regulation. Conversely, destination challenges include infrastructure, communication connectivity problems, and interdependency of virtual team members (M. Khan & Khan, 2018). GSD organizational leaders need an understanding of how to manage a virtual team effectively despite the popularity of available communication technological tools. Change in the version of the technology, and test labs could pose a problem to the team members (M.Khan & Khan, 2018).

Development of the team management challenge mitigation model by M.Khan and Khan (2018) was to assist in virtual team management. The authors used the methodology of SLR to identify challenges encountered by outsourced vendors and questionnaire surveys in the industry to validate SLR findings (M.Khan & Khan, 2018). They searched for project management success factors in GSD by SLR, then compared them with a string of derived questions from software development participants in the industry. M. Khan and Khan (2018) identified project management success factors as the project managers, organization structure, culture, and trust. The research outcome was that the model could be useful to help predict and measure organizational readiness for project management in GSD settings (M.Khan & Khan, 2018).

Effort and Cost Estimations

GSD organizational leaders build a trustworthy effort for the estimation of a software project, which is an integral part of project management. Cost estimation is predictive of the effort required by software managers from the start to finish time of a project (Keshta, 2017). According to Ramacharan and Rao (2016), GSD is a preference of software managers because of the benefit of shorter software developmental cycle. Effort estimation is the preliminary phase of a software development process where there are project definitions with a certain level of estimation accuracy. Authorization to start a project occurs at project charter before planning, executing, monitoring, and closing phases (PMI, 2018). Effort cost estimation becomes useful in software development project processes, acting as inputs in iterative project planning, analysis of contract biddings, resource allocation, procurement, and evaluation of financial budgets (Keshta, 2017; Ramacharan & Rao, 2016). There were two empirical search-based models (COCOMO II and SLIM Models) in the study (Ramacharan & Rao, 2016).

Schedule-based estimation model with calibration parameters provided evidence of the accuracy of effort estimation (Ramacharan & Rao, 2016). In the methodology, the authors used COSEEKMO, a software tool to make selections. The findings were that both models provided the same accuracy for in-house software development and GSD projects. The schedule-based model accuracy was higher for GSD projects. The recommendation was that future researchers should be able to compare model-based and GSD expert-based ideas (Ramacharan & Rao, 2016). Keshta (2017) used similar software estimation models (COCOMO II and SLIM Models) in an investigation then compared with the other four estimation models: Cocomo, Hybrid, Putman, Function point, and Wavelet Neutral Network. The research findings indicated that it is in the best interest of the project manager to use more than one estimation model so that results can be properly evaluated before final decisions are made. Boehm (2017) argued that the practice of one-size-fits-all software cost-schedule estimation models are no longer realistic due to the GSD evolutional pattern of wide variations and characteristics. Software estimation is a continuous process from the initiating phase of the project through the developmental life cycle and is a key success factor to project management (Keshta, 2017).

Organizational Structure

GSD management organizational leadership is a need to ensure effective coordination, communication, and control among the team members. Managerial difficulties comprise 77% of GSD communication problems (Ammad, Janjua, Madni, Cheema, & Shahid, 2017). Despite the emergence of new telecommunication technologies, collaborative environment, and group support systems, the management of a virtual team is a great challenge (Shameem, Kumar, & Chandra, 2017). Enforcing the applicable GSD countermeasures to the prevailing challenges would be of no effect without addressing the managerial aspect of GSD in distributed locations (Ammad et el., 2017; Birgun & Altan, 2019). Project performance of in-house software development is higher than that of GSD, because of a lack of communication and coordination. Galin

(2018) identified procedures that ensure timely performance in software projects as the implementation of project control components, including monitoring of schedules, budgets, and quality. The effects of globalization on the economy affect the software industry. Project managers should analyze all aspects of project performance with a complete feedback evaluation.

Quality is made in the boardroom by top management, vested with the power to make decisions and changes to design or systems (Deming, 2013). GSD challenges on management negatively impacted software product quality and stakeholders (Ammad et al., 2017).The GSD methods and strategies have inherent issues, such as management, technical, organizational, team, and customer issues that affect communication among the team members (Alzoubi, Gill, & Al-Ani, 2016; A.Khan, Keung, Hussain, & Bennin, 2015). The increasing demand for software products brings changes to the organizational structure of companies in the software development industry. The reliable organizational structure will enhance quality of software, minimize cost of production, enable timely delivery, and help meet users' expectations by adopting a strategic arrangement that manages and fosters interaction among team members (Massago, Leal, & Balancieri, 2018).

The evolving nature of GSD was from the popular tradition of the small collocated team to widely spread geographically teams that contribute to the development of software products. The distributed teams are involvement at all stages, starting from conceptualization, development, and maintenance (Yue et al., 2019). One of the major challenges of GSD is low software product quality, due to many problems associated with distributed teams working in different locations to develop a software product. Evidence in the literature is that early researchers and practitioners used various software development techniques and quality management theories to develop a software product. Software procedures, ISO standards, frameworks, and software models helped to predict software product quality. Organizational leaders used Deming's theory, Stewart's Cycle, Crosby's theory, TQM, Agile, Lean, Scrum, and Kanban to develop software. The other specific software quality models used by practitioners for quality examination and prediction is the Boehm, McCall, FURPS, Dormey, and ISO 9126 (Al-Obthani & Ameen, 2018; Musa & Alkhateeb, 2013). Jabangwe, Wohlin, Petersen, Smite, and Borstler (2016) developed a method for investigating quality using defects data as a proxy in a GSD developing environment at different industrial sites using lessons learned to improve the process.

GSD has different terminology because of the emerging new paradigm in the research area. The evolution of GSD practice brought about more questions, conducted research, and literature. Prikladnicki, Damian, and Audy (2008) noted that the diversity in terminology shows that the area is still immature, lacking proper definition and standardization. In their study of the pattern of evolution in the practice of distributed software development, Prikladnicki et al. identified research papers and describe existing models, suggesting the need for distributed software development models. The GSD researchers ventured into combining GSD process and CMMI to achieve advanced software process quality and good product. Earlier studies indicated how project planning is measurable in CMMI, analyzing projects in terms of resource management and task breakdown. Khraiwesh (2013) used GQM procedure to derive three specific goals and fourteen defined project planning areas in CMMI. The CMMI model is for process improvement, reducing cost, eliminating inconsistencies, and product quality (Hidayati et al., 2018). The CMMI is a process model for assessment and SPI. GSD organizational leaders often deploy the SPI model to increase efficiency, productivity, and software quality. The CMMI model has 22 processes grouped into four sections: process management, project management, engineering, and support (Hidayati et al., 2018).

The Use of Agile: Scrum, Lean, and Kanban with GSD

The efforts to overcome GSD challenges necessitated the use of the agile method in GSD setting. Alotaibi and Qureshi (2014) examined scrum practices on temporal distance-based on GSD team coordination challenges. Alsahil, Khan, and Alyahya (2017) reviewed the literature on GSD challenges and searched several agile practices on how to mitigate software development problems with a special focus on Scrum. Organizational leaders of GSD, and modern researchers constantly experimented with various remedies to reduce GSD challenges. The agile practices of regular communication and self-organization at various remote locations could help minimize GSD challenges (Vallon, Estacio, Prikladnicki, & Grechenig, 2018). Agile and GSD combination is classified as a maturing field but with potentials that may contribute to the improvement of software quality. Alvis, Gonecalves, and Bax (2017) noted that the scrum method enables process innovation and improvement. The combination of the scrum, and extreme Programming is a most useful agile method but there is a need to use defined empirical results in the development of a better framework that is compatible with GSD (Alvis et al., 2017; Vallon et al., 2018).

There were attempts by practitioners to combine lean thinking and Kanban principles with the GSD approach to overcome certain software development challenges. According to Tanner and Dauane (2017), lean thinking starts with value creation for the customer, while Kanban principles pertain to the concept of value creation using visuals and work in progress management. Software companies choose methodologies for software development, with resulting impacts on the quality of the product and future markets. In the application of scrum principles, the practitioners could not add lean principles of measure and practice (Shahzeydi, Gandomani, & Sadeghi, 2018). Krehbiel and Miller (2018) concluded that Lean and Six Sigma are effective in IT-based business processes but not in IT organization, when comparing Deming's system of profound knowledge, strength of relationship with Agile, Lean, and Six Sigma. Nataraja and Ramesh (2016) stated that the software development evolving trend is continuous with innovations of new methodologies, technologies, design tools, and emerging new ideas.

Change Management

Complexities and dynamics of the GSD environment are further compounded by challenges of organizational creativity. The frequency of organizational innovations and creativities is evident in a constant changing GSD setting by the unfolding new technologies with peculiar requirements. Software development change and evolution may have a significant influence on software quality (Jabangwe et al., 2016). Managing creativity to sustain innovations is one of the challenges of GSD organization, due to the continuously changing market environment (Frank & Mengiste, 2014). The study of creativity in a software organization is traceable back to the 1990s. Frank and Mengiste (2014) applied a multiadaptive approach to collect data from participants in Information Systems Development organizations to compare management perspective and perceived challenges to the findings in the literature. The main problem of software requirements engineering is change. RCM improves the requirement process. Alsanad, Chikh, and Mirza (2019) investigated the use of ontology methodology in a GSD environment to improve the requirement process by eliminating inconsistencies and incompleteness.

Global Software Development Challenges

GSD challenges impact the quality of software (Shanyour & Qusef, 2018). Despite the benefits of GSD, there are challenges and issues with the continuous progression of technologies, due to an evolving trend that is heterogeneous in nature (Shanyour & Qusef, 2018). The GSD challenges documented in the existing literature grow, but as GSD evolves, new challenges are fast emerging. These challenges could be in control, coordination, and communication, due to socio-cultural, geographical, and temporary distance (Asiri & Qureshi, 2014; S.U.Khan & Niazi, 2012). Or they may be human (Shameem, Chandra, Kumar, & Kumar, 2018), poor quality (Jaffar, Hussain, & Chiad, 2019), technical, political, and social (Suman & Jain, 2015), as well as related to distance and speed (Carrillo de Gea, Nicolas, Fernandez□Aleman, Toval, & Idris, 2016). Other challenges surround standardization (Jain & Suman, 2015), stakeholder management (Colomo-Palacios, Soto-Acosta, García-Peñalvo & García-Crespo, 2012), trust (Trainer & Remiles, 2018), risk (Aslam, Ijaz, Lali, & Mehmood, 2017), and replacing legacy systems (Matthiesen & Bjorn, 2015).

Global Software Development and Cloud Computing

The combination of GSD and cloud computing is an emerging strategy to overcome the challenge of collaboration, with an impact on quality. The advantages are the common characteristics of virtualization, reduce cost, scalability, infrastructures, and performance (Shanyour & Qusef, 2018). Service-Oriented Architecture (SOA) enable access to data applications. According to Shanyour and Qusef (2018), SOA alleviates the problem of GSD coordination and knowledge transfer. The ideas to use the cloud to facilitate GSD challenges, such as collaboration, are expanding because of growing software markets that are beyond national boundaries. Harshmi et al. (2011) noted that cloud computing characteristics can improve the GSD process and product because of knowledge gained from GSD and SOA. Cloud computing is an enabler of agile global software development, despite agile, social, and technical challenges because of the distributed environment (Haig-Smith & Tanner, 2016).

GSD Education and Training

The contemporary organizational GSD leaders are proactive by creating awareness and training of IT engineers on the complexities and socio-cultural challenges of GSD in a dispersed and diverse team. Chevers (2018) concluded that organization climate, user training, education, requirement management and managerial technical support influence quality and success. Aslam et al. (2017) explored how GSD competent workforces can expedite skills acquisition for entrepreneurs and IT professionals in less developed countries. Deming (1986) emphasized the place of employee capacity building among the defined 14 principles of quality management theory. The global software market is reshaping tremendously with innovations, as businesses strive for best practice. Countries like India and China have become major players in software global markets within a short space of time because of the IT revolution. GSD is sustained by IT organizational leadership through the performance of system reviews, research, education, and competitive advantage (Talib, Saeed, Awais, & Hanif, 2107). As GSD evolves, the training and engagement of the workforce depend on leadership strategies to enhance the outcomes of employees and investment gains for the organization.

Software Reuse Management

GSD is rapidly growing because of the available pool of resources and innovations scattered all over the world. The increasing change and high demand from stakeholders must be met by software organizations, despite project constraints. There is a need to build in quality and consider the accessibility of every user in every stage of the software development life cycle, because the greater audiences are becoming users of the software systems (Baquero, Gil & Hernánde, 2018). The software reuse process is complex because of software legacy, notwithstanding the benefits of improving productivity and the quality of software products (Paschali, Ampatzogloua, Bibi, Chatzigeorgiouc, & Stamelos, 2017). GSD organizational leaders are developing better techniques to mitigate challenges and ensure high quality, productivity, and accessibility, and to reduce time and cost, simplify software testing, and enable reusability of software. Kalia and Sood (2014) affirmed that standardization and organizational issues, like infrastructure, legalities, and motivations are some of the challenges of component-based software reuse. In order to implement the reusability process during software development software, practitioners make use of already developed software components. The quest to satisfying developmental needs brought about software reuse. The components-based software reuse process is about the building of systems from the approved design and pre-tested components that can save the cost of redesign and writing of new codes (Kalia & Sood, 2014).

The reuse frameworks help to expedite the developmental process by preventing reliance on system components or existing markets, providing a quick response to the new requirement. A. Khan, Khan, Amir, and Khan (2014) initiated a component-based framework using the strategy of software architecture language identification, object-oriented design, and a software architecture based on a domain-specific and generic software product. The certain characteristics are common with component-based technologies that enable programmers with guidelines, prototypes, and principles (A. Khan et al., 2014). The demand for minimum developmental time and cost, with greater quality, encourages the use of the component-based framework for software reusability. Existing software components must possess some attributes on software architecture, including functionality, behavior, and interfaces. A. Khan et al. used the method of SLR to identify several component-based frameworks, tools, techniques, merit, and demerit. The already derived component artifacts enter a repository for future use, while the second part of the framework applied to perform analysis on the stored component for reuse (A. Khan et al., 2014). A different aspect of the component-based framework includes software development, component development, reusable domain analysis, reusable tagging, reusable component repository, search reusable artifacts, component integration, and system validation (A.Khan et al., 2014). A.Khan et al. evaluated studies, compared different reusable techniques, and then determined how to reduce cohesion between two independent components.

The software development component-based research work of Paschali et al. (2017) has many similarities with that of A. Khan et al. (2014) on component-based software reusability. Paschali et al. identified existing literature and models on component-based software reuse and proposed a new model for improvement of software quality and cost reduction. To solve the problem of complexity in the reuse process, the proposed model for the software life cycle of component-based has three phases: design, integration, and run time (Paschali et al., 2017). Some basic questions on component selections pertain to the software component life cycle model definition. The questions of the component needs, sequence, and composition require answers (Paschali et al., 2017). The different levels of reuse were specification, design, and code reuse; Paschali et al. (2017) concluded that the study provided more understanding to practitioners and researchers on the reusability of software.

Software Security Management

Software security is a vital aspect of the SDLC from the requirement to final disposal in GSD (R.A. Khan & Khan, 2018). Software use is expanding and becoming more user-friendly in all facets of life. At first, software applications were common in the business environment; but now they are useful for more personal transactions and interactions, enhancing daily social networking among people. The internet-enabled software application drives many processes, such as email, text messages, video, telephone calls, online banking, and ATM machines. Software is extensive in many fields of endeavours, such as medical, education, telecommunication, agriculture, climate, finance, transportation, operation, industry, administration, military, research, and entertainment (Kalia & Sood, 2014). This high demand posed a lot of challenges to meets requirements and software product quality. The issues of security, personal privacy, safety, data protection, and regulation became key requirement factors to address from the viewpoints of customers and end-users. Software security systems are an internet-enabled computer software application subject to frequent malicious attacks (R.A.Khan & Khan, 2018). Software must function and perform satisfactorily under any form of attack. Software assurance assessment is ensuring the system design and building is free from defects and any kind of vulnerabilities (Rashidi & Hemanyati, 2018).

Software security is not an aftermath process, considered at every stage of the software development process. M.U.A. Khan and Zulkernine (2009) noted that in traditional software development software, security starts at the beginning of the life cycle. R.A. Khan and Khan (2018) identified the perspective of software security in the literature, empirical studies, and their implementation research by GSD organization vendors. The authors examined state-of-art software security practices in SDLC and developed the Software Security Assurance Model (SSAM) to determine the readiness of GSD vendors' adoption of software security in their environment (R.A. Khan & Khan, 2018). The methodology approach was a systematic mapping study, comparing findings from the literature review to the outcomes, leading to suggestions for future empirical study of GSD vendors to determine their level of practice in software security (R.A. Khan & Khan, 2018).

Risk and Trust Mitigations

The GSD Organizational leaders are conscious of the presence of internal and external risks in the software development process. Risk management is a continuous process that captures and controls potential risks that may affect the objectives of the software development information system, especially in a complex environment with uncertainties (Khraiwesh, 2013). Risk impacts software development, organization reputation, software quality, and industry competitiveness (Tavares, Sanches da Silva, & Diniz de Souza, 2019).

Software developmental risks are bound to increase, because GSD involves additional steps and new decisions on requirements that threaten the risk management process (Galli, 2018).The web enabled platform enhances GSD collaboration with new innovations, ideas, and lines of product for customers (D.C.Chou & Chou, 2011). GSD managers should consider the disperse nature of the team, geographical distance, cultural difference, and organizational structure levels in risk management (Galli, 2018). Risk identification and categorization helped to reveal the impacts and how to address it in a software development life cycle (Aslam et al., 2017). Poor product quality and rework are preventable by using effective, flexible, and robust risk management strategies in GSD (Husin et al., 2019).

The issue of trust in a GSD virtual team is a challenge in collaborating efforts among team members. A team with a high level of trust could easily achieve high productivity and quality of software. A team without trust may still have high productivity, but with extra-effort of team members monitoring each other for compliance (Trainer & Remiles, 2018). Trainer and Remiles (2018) examined how software tool support can create trust awareness among globally distributed and dispersed team members. The qualitative research involved literature review about how developers form reliance on one another, and visualization strategies used by 28 students and 12 software professionals (Trainer & Remiles, 2018). The findings reported by Trainer and Remiles were that team members that used visualization form accurate positive attribution with a certain level of trustworthiness for their teammates who worked remotely; the authors further noted that the limitation of the study was the measure of trustworthiness, which was not possible to completely ascertain.

The Gap in the Literature

GSD offers numerous benefits to GSD organizations. However, there are several challenges that affect software product quality (Hidayati et al., 2018). GSD organizational leaders may have a common understanding of how to mitigate these challenges to software quality, to meet customer expectations. There is a need for key stakeholders to have a common understanding of quality (Barney et al., 2013). Through the reviewed literature, it is evident that software quality is one of the major factors that impact GSD, because of coordination and communication issues. Conversely, only a few studies showed that all GSD challenges negatively impact quality. Shanyour and Qusef

(2018) emphasized that several GSD issues affect software quality: teams, technical issues, knowledge transfer, security, privacy, language requirements, and process management. The use of a qualitative exploratory approach to evaluate the status of GSD and software quality led to the suggestion that GSD challenges affect overall software product quality.

As the GSD method gains wider acceptance and evolved to maturity in a fast-dynamic business world, new software requirements emerge, due to high demand for software. The research community contributed to state-of-the-art processes but left fundamental gaps in research (Schneider, Torkar, & Gorschek, 2013). Ghanbari et al. (2018) revealed that only a few studies were about software engineering and information system literature omitted much of software quality practices. The GSD practitioners and researchers are aware of the effects of GSD challenges, such as geographical distance, temporal distance, socio-cultural, and quality concerns. Several organizations' managers that moved GSD offshore to generate more revenue suddenly realized that projects are failing because of unforeseen threats. The GSD field is still at an early stage, because best practices and models in GSD are immature in the literature (Mishra et al., 2013).

There is a challenge of scarce empirical studies about software product quality in software development distributed environment. Wickramarachchi and Lai (2016) explained that there is a limited understanding of the GSD modality success or failure because research on software quality and cost-benefit investment is scarce. Empirical studies on the extent of the negative effects of the GSD challenges of communication and coordination on software quality are few (Jabangwe et al., 2014). Researchers revealed

GSD is still evolving, lacking mature research and an all-inclusive quality model that can evaluate software defects and predict software quality (Chadli et al., 2016).

A holistic approach is required to solve the problem of GSD and quality considering the advantages of modern technology. Kuhrmann and Fernández (2015) confirmed that existing studies lack the overall representation approach of GSD, and instead, concentrate on single items that mostly end without further evaluation and validation. GSD organizational leaders aim for high-quality and cost-savings in a distributed development environment, with a lack of awareness on how to adopt and use GSD existing tools (Niazi et al., 2014). Marandi and Khan (2015) recognized that improved software tools may reduce testing supports required to achieve excellent quality software products. Few works of literature pertain to the recent development in the field of GSD issues and its impact on software quality (Shanyour & Qusef, 2018). According to Shanyour and Qusef (2018), the advancement of GSD is due to internet-enabled smart technologies and new techniques with features of flexibility, scalability, usage tracking, and independence. The authors used the exploratory method and empirical resources to arrive at a consolidated viewpoint about GSD challenges and impacts on software quality (Shanyour & Qusef, 2018).

GSD organization leaders are adopting GSD approaches because of its popularity and production of high-quality software. Jain and Suman (2015) reported that research works of the GSD cycle are in harmony, but require integration, consolidation, and understanding for GSD organizational leaders and researchers. The absence of consensus on the clear definition of software quality exists in the research community (Furtado, Vignando, Franca, & Oliveira, 2019). GSD grew from practice to influencing the research environment, requiring many aspects of GSD method development for maturity (Aranda, 2010). There is utmost need to build a body of knowledge on quality management for GSD projects, so that defined experience and practice may provide a better understanding (Mishra et al., 2013).

Summary and Conclusions

GSD in organizations is fast growing in different parts of the world, due to the great benefits and characterized challenges that negatively impact software product quality (Hidayati et al., 2018; Shanyour & Qusef, 2018). The common understanding of GSD organizational leaders on how to mitigate the management problem of low software quality and meet customer satisfaction remained unexplored. GSD is still evolving, termed immature with scanty empirical research on software product quality (Chadli et al., 2016; Wickramarachchi &Lai, 2016). This study was important because of the high level of software project failures in GSD settings. Additionally, exploring the common understanding of GSD organizational leaders was critical because of the role of management in a distributed development environment to ensure effective communication, coordination, and control among the teams at multiple sites.

Previously, the practices stemmed from varied expert opinions, unstandardized developmental approaches, and socio-cultural factors (Jain & Suman, 2015; Vignando et al., 2019). It is important to investigate context, opinions, culture, lived experiences, challenges, used models, and quality records of GSD organizational leaders (Marandi & Khan, 2015; Shanyour & Qusef, 2018). As the GSD low software product quality persists globally, the understanding from a specific population such as the GSD organizational leaders in Canadian organizations adds to the knowledge that guides the practitioners and researchers on how to improve software product quality. The qualitative method with interview questions was useful to investigate the perceptions of GSD organizational leaders about what they need to mitigate low software product quality in Canadian organizations.

The qualitative case study niche tends to answer questions about *what*, *how,* and *why* of a contemporary phenomenon (Yin, 2017). Qualitative approaches with interviews for data collection and analysis may provide more interaction between the participants and the researcher that will foster a deeper understanding (Akbar et al., 2020). The framework of this study was a multiple case study with an exploratory and in-depth nature that enabled a better understanding of the experiences of GSD organizational leaders, considered in the light of Deming's management principles. GSD organizations continue to strive for better techniques to improve the quality of software products in a distributed environment. As software demand continues to increase, more requirements and challenges emerge. This study was a response to the utmost need to address the GSD management problem of low software product quality and customer satisfaction.

Chapter 3: Research Method

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding, shared by managers in Canadian GSD organizations, of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction. The knowledge and common understanding derived from this study represents insights and solutions for the management of low software product quality. In this study, 30 participants who were GSD organizational managers from six Canadian organizations with at least 6 years of experience in GSD practice completed interviews. The analysis of the qualitative data led to the identification of a common understanding of how best to manage low software quality in the GSD environment. The research questions pertained to the common understanding of GSD held by organizational leaders and what these leaders need to mitigate low software product quality challenges, improve the software development process, and achieve customer satisfaction. The criteria for the selection of GSD participants who were managers helped ensure that these leaders were knowledgeable and experienced.

In this chapter, I present the research methodology for this study and how it conformed to study objectives, the three research questions, and the research design. I discuss the rationale for adopting a qualitative study, as well as the research environment, population, sampling method, justification for the sampling method, and scope of the study. Additionally, I explain the criteria for the selection of 30 GSD participants from the population, the instruments, sources of data collection, data transcription technique, data verification, storage, and analysis. Further discussion involves protocols, trustworthiness of the study, and ethics of the research study.

Research Design and Rationale

Research Questions

RQ1: What are the common understandings of what GSD organizational leaders need to meet software product quality in Canadian GSD organizations?

RQ2: How can common understanding be gained from GSD organizational leaders?

RQ3: How do GSD organizational leaders predict the outcome of software product quality?

A case study design was useful to explore the phenomenon of interest through in-depth interviews of GSD organizational leaders concerning their lived experiences, their common understanding, and what they need to mitigate low software product quality. Exploration of the perceptions of organizational managers led to a common understanding of how they manage software product quality in GSD Canadian organizations. Concepts that emerged from data-coding processes coalesced into the themes. Analysis of data through identifying patterns and notating features culminated in themes that, as Saldana (2016) recommended, represented an extensive contextual description of the phenomenon. I was open to emerging meanings drawn from the data without restriction.

I chose the qualitative research method over the quantitative research method for this study because the former enables the researchers to gain a meaningful understanding of underlying reasons and impulses. Qualitative nonnumerical data do not stem from randomly selected samples but do form into patterns of less generalizable results (Ridder, 2017; Simons, 2009; Stake, 1995; Yin, 2017). The qualitative method produced rich details from open-ended responses, and data analysis led to the identification of patterns and features that led to themes. In contrast, the quantitative approach involves data and statistical analysis including relationships and cause and effect, from data derived from precise measurements and randomly selected samples, leading to more generalizable results (Yilmaz, 2014). The most suitable qualitative research design for this study was case study that was an exploratory and descriptive address of a phenomenon through an in-depth analysis of data, including details about the context. The case study involves illustrative, heuristic, evaluative, and descriptive patterns (Marriam, 1998).

Role of the Researcher

I personally conducted the process of data collection for this multiple case study. My focus was on understanding the cases by collecting, organizing, and inductively analyzing data, ensuring reliability and validity of data, and reporting research findings. As a researcher, I adopted a descriptive and interpretive approach to provide meaning from the experiences of GSD organizational managers and supervisors in Canadian organizations. As a quality manager for 10 years in the oil and gas industry and a trained lead auditor on quality management systems, I employed qualitative methodology when I interviewed managers and supervisors in companies during the periodic quality management audit process for major projects. Quality management is a trusted profession because managers and supervisors are comfortable sharing their experiences with quality managers for process and product improvement purposes. Quality managers are conversant with the art of listening to what organizational leaders are saying and deriving meanings from these discussions. To be a better and more effective quality manager, I learned to understand what motivates, or influences, the decision making and actions of organizational managers. I applied my training in problem solving, listening strategy, and decision-making techniques to confront complex leadership management situations.

In quality management professional practice, I learned to ask probing questions, listen, and interpret statements. As a former quality manager, I understood some of the needs and challenges of leaders in organizations. This personal background increased my risk of bias during interviews with the participants. My vested interest was discovering a solution to the problem of GSD low software product quality, based on the research findings. The act of becoming aware of personal values, perceptions, or interest in a research study limits chance that prejudgment will occur (Z.C.Y.Chan, Fung, & Chien, 2013). Researcher bias is not acceptable during participant interviews and analyses of data (Yin, 2017). To mitigate biases, I avoided participant selection bias, use of jargon, irrelevant terminology, and misleading questions. I utilized clear language, framed questions in a neutral tone with clear intention, kept my style transparent, built trust, and sustained a high level of honesty and integrity.

Using previous data collection techniques instead of allowing personal emotions and experiences to direct research will mitigate personal bias in the research process (Berger, 2015). I tailored questions to target the desired population and meet set research goals. The achievement of the research objective was through following a defined set of goals and requirements for the participants (Roulston & Shelton, 2015). The use of clear language and avoidance of misleading questions helped to build an honest, nonjudgmental approach and minimize misinterpretation of questions. By adopting a conventional research approach with an emphasis on a good communication style, I followed an acceptable process with a focus on research goals and outcomes.

In the sections that follow, I explain the research purpose, background, planned interview process, ethics governing the research study, and procedures for obtaining informed consent from the participants. An integral aspect of the data collection process is participants' willingness to sign and return the research consent form before the conduct of interviews (Cummings, Zagrodney, & Day, 2015). During the interviews, I also took notes and after interviews, I transcribed recordings into verbatim textual data. Participants were able to review the data and initial interpretations of data; allowing participants to review their own data and the initial interpretations of data in a member checking process helps to ensure accuracy and build trustworthiness in the findings through validation (Benes, Mazerolle, & Bowman, 2014). I ensured that the participants completed interviews according to approved Walden University Institutional Review Board (IRB) protocols. My position was objective, given that I was in another country and remained inactive with a quality group in the oil and gas industry for 4 years.

Methodology

Three different types of research methods predominate academic writings (Creswell, 2014). Qualitative, quantitative, and mixed methods are methods of contemporary research practices (Habib, Pathik, & Maryam). In this section, I explain and justify the selection of the qualitative method and case study research design as the exploratory approach that was most suitable for this study.

Research Method

The qualitative method was most appropriate for this study because of its alignment with the research questions and purpose. In the qualitative method, the researcher seeks to understand the *what*, *how* or *why* of a phenomenon (Yin, 2017). I selected the qualitative method because of its flexibility, the involvement of human interactions, the use of nonrandom sampling, and the method's in-depth exploratory capability to elucidate rich descriptions of a phenomenon, as described by Denzin and Lincoln (2011). The open-ended interview questions, review of documents, and contextualized data in the qualitative research method may help a researcher derive opinions, views, and ideas from participants (Patton, 2015). A qualitative method was most practical for the study of a common understanding of GSD organizational leaders to mitigate low software quality and meet customer expectations.

Quantitative and mixed methods were not appropriate for this study because, as Creswell (2014) explained, phenomena like human behavior may be difficult to measure in a natural setting. I did not test hypotheses or theory, generalize the findings, or validate data statistically in this study. The quantitative method involves the testing of theories and hypotheses, numerical quantification, and comparisons of dependent and independent variables (Wisdom, Cavaleri, Onwuegbuzie, & Green, 2012). The survey instrument and sampling techniques in a quantitative survey may be susceptible to errors (Onwuegbuzie, Bustamante, & Nelson, 2010). The mixed method involves the combination of qualitative and quantitative methods to improve evaluation, understanding, and flexibility while mitigating limitations (Birchall, Murphy, & Milne, 2016). The mixed method can result in complications due to faulty design or harmonization of research data (Creswell, 2014). I did not use the mixed method approach in this research because it was not appropriate, in that the use of quantitative elements, including a closed-ended questionnaire and a random sample, were unlikely to result in the in-depth answers to the research questions. The qualitative method was most feasible for the collection of interview and document data that could be most useful in answering the research questions.

Research Design

The qualitative case study research design is unique from other qualitative strategies of inquiry. The case study design involves specific circumstances and contexts that are not clear (Lunnay, Borlagdan, McNaughton, & Ward, 2015). Case study is an empirical form of inquiry through which a researcher examines contemporary phenomenon in-depth in real-life situations (Yin, 2017). Exploratory case study enables researchers to explain complex situations (Creswell, 2014). These attributes of a case study aligned with the contemporary GSD organizational management problem of low software product quality as presented in the exiting literature. I answered the research questions by interviewing participants and reviewing documents, including quality records, procedures, and policies.

Qualitative case study designs enhance practical research problem solving (Stake, 1985). This research occurred to gain a common understanding of what GSD organizational leaders need to mitigate low software product quality. Other qualitative designs would not be suitable for this study. Phenomenological design, for instance, pertains to the meaning and essence of lived experiences of participants within a setting (Bawa & Watson, 2017), which was not the purpose of this study. The grounded theory design is primarily useful to help a researcher develop a new theory (Marshall & Rossman, 2014), which was not of interest in this study. Ethnography revolves around community and cultures of people (Sarmento, Gysels, Higginson, & Gomes, 2017), which would not have led to in-depth business insights about processes and procedures. In the narrative design, storytelling is a means to provide information about people or individuals' lived experiences (Bell, 2017), which would have failed to result in answers to the research questions that pertained to needs in a business-oriented concepts. The case study design was most appropriate for this study because encompassed a comprehensive in-depth analysis of the perspectives of GSD organizational managers in a distributed setting. I obtained relevant information from participants during the interviews. Rich and thick explanations that may lead to an answer to the research question can originate from informed research participants (Cornelissen, 2016). I continued to collect research data from participants until the point of redundancy when no new information was apparent, leading to the rich descriptions of context, the pertinent demographics of the sample, the and the answers to the research questions, which were appropriate steps in case study research designs.

Participant Selection Logic

The research population included leaders from 200 Canadian GSD organizations. The sample included 30 participants from six companies across six Canadian provinces. In a qualitative case study, the criteria for sample size is the sample size that will provide enough information to meet the objectives of the research study (Yin, 2017). The analysis for this in-depth qualitative case study revolved around a common understanding of what GSD organizational leaders need to mitigate low software product quality management problems in order to enhance customer satisfaction. These units of analyses involved GSD organizational managers and performances of the organization pertaining to the management of software development in a distributed environment. The case study sample size of 30 participants, including GSD organizational managers, was justifiable for the research objectives. The sampling technique was purposeful sampling, to obtain insights from knowledgeable experienced leaders who represent their organizations.

In a case study design, multiple sources of data collection (such as interview data from multiple informants and the review of documents) culminates in strong evidence, data quality, and credibility (Yin, 2017). The use of criteria-based selection is desirable for sample determination (Merriam & Tisdell, 2016). The criteria for selection of GSD managers included involvement with GSD for a minimum of 6 years, with demonstrated competency, responsibility, commitment, honesty, and integrity. Leaders needed relevant experience, certifications, and knowledge in software development. Five organizational leaders were from each of six GSD Canadian organizations, for a total of 30 participants in the sample. This strategy of purposeful sampling can enhance the trustworthiness of the study (Patton, 2015), and was likely to reveal the skills of each organizational manager in their own individual organizations and enable comparison. The approach to sampling was representative of the population of GSD leaders across Canadian GSD organizations.

Instrumentation

I was the primary data collection instrument for this exploratory and multiple case studies. I collected data from participants to better understand the meaning of their lived experiences in GSD organizations. Semistructured interviews occurred through audioconferencing and the telephone, and the collection of quality record documents was via email. In this study, the protocol with open-ended interview questions was based on the Tamimi et al. (1995) validated instrument for Deming's 14 points. This enabled an in-depth exploratory study of the common knowledge among managers in GSD organizations. The interview protocol questions included questions on how GSD organizational leaders predicted software product quality using the SDLC, existing software models, and standards. Semistructured interviews allow for flexibility and consistency (Dikko, 2016; Dunn, Margaritis, & Anderson, 2017). For example, Padgett, Gossett, Mayer, Chien, and Turner (2017) used semistructured interviews in their qualitative research to garner a better understanding of a phenomenon in a high reliability organization. The use of multiple sources of data, recommended by Stake (1995) and Yin (2017) for case study research, ensured strong evidence and enough data to provide the answers to the research questions. The understanding of the researcher on the topic increases by multiple data collection (Wang, 2016). The goal of the data collection was to gather enough data from GSD managers to reveal a common understanding needed to mitigate the management problem of low software product quality can emerge. Interviews with GSD organizational leaders occurred with the use of an interview protocol with open-ended questions based on a validated instrument on Deming's 14 points developed by Tamimi et al. (1995). The open-ended interview protocol included questions about the prediction of software product quality using modern software models, SDLC, and international standards (see Appendix). Participants received the questions in advance of the interviews. Hanley, Fileborn, Larcombe, Henry, and Powell (2015) similarly sent interview questions in advance to their research participants.

Interview protocols include ground rules for the interview process to ensure the validation of content and reduction of the duplication of information (Brubacher, Poole, & Dickinson, 2015). I requested an interview time schedule of 60 minutes from each of the GSD organizational leaders as an anticipated duration for each interview. I recorded the interview process after obtaining participants' signatures on the consent forms. I provided all participants informed consent forms with vital information about the purpose of the study, interview procedure, and the right to participate or withdraw from the study by email notification. The research participants had a chance to confirm review the transcripts of their responses after the interviews for verification and accuracy. Member checking in a research process allows participants to review and comment on the initial interpretations of data (Harvey, 2015). The research data validity can be achieved through member checking exercise (Zhao, Zuo, & Deng, 2015). I asked participants to review the initial interpretations of data in a member checking process.

I conducted the interviews with GSD organizational leaders by asking the participants about their understanding and experience in software development in a distributed environment. A protocol for the interviews with open-ended questions, based on the Tamimi et al. (1995) validated instrument for Deming’s 14 points, included questions about how GSD organizational managers predicted software product quality using existing SDLC, models, and standards. The interview approach helped put participants' peculiar experiences in context. I strived to conduct interviews in a quiet and confidential space where the participants could feel comfortable with a sense of trust, privacy protection, and good rapport. Research data were in files, labeled alphabetically with dates for easy traceability, and data analysis. In order to achieve effective data organization and management, I utilized the Computer-Assisted Qualitative Data Analysis Software (CAQDAS). The CAQDAS was helpful in indexing, data comparison, coding, searches, and related steps in data organization and analysis. I stored data in accordance with Walden University IRB regulations, ensuring the safety and security of information. The use of NVivo software in the research process likely enhanced the legitimacy and credibility of the research findings. The use of NVivo typically improves the accuracy of data analysis and coding. I coded data to help unveil themes, appreciate the convergence and divergence across the dataset, and identify categories, concepts, and groups to form themes about the phenomenon.

I was the primary research instrument that collected data and performed the data coding. The software enhances the coding process, which stores, organizes, manages, and configures the data for improvement, and better interpretation (Strauss & Corbin, 1998).

CAQDAS is widely useful for coding, and data analysis. There are limitations to CAQDAS, just like the popular traditional manual coding technique. The justification for a data coding method in a research study depends on the nature of the research projects in terms of size and availability of resources. Edhlund and McDougall (2016) asserted that there are limitations of software application for data management because researchers often miss components and there can be formatting errors and poor duplication of images. Other disadvantages are incorrect coding and misplacement of relevant pattern and categories due to inaccurate recognition of data by the computer systems.

The CAQDAS is readily available, effective, and easy to use (Zamawe, 2015). I employed CAQDAS such as NVivo for coding and data analysis of this research work. The manual approach known as the Traditional Manual Method was not be useful for this study, though it has some advantages that require less expertise and suitability for small research projects. The manual method is susceptible to researchers' bias and is time-consuming (Saldana, 2016). In addition, CAQDAS had advantages, such as the ability to handle large data, rigor, and speed, the expectation was that it would save time, facilitate the exploration of ideas, and foster learning.

Procedures for Recruitment, Participation, and Data Collection

Population Sampling

I relied on the nonrandom purposive sampling approach to select the participants. Often, researchers use certain criteria consistent with qualitative research features to select and engage the participants (Jones et al., 2016). The Canadian GSD organizational leaders who managed quality software projects successfully for a period not less than 6 years were eligible to participate in this study. The inclusion of participants from each of six selected GSD organizations in five provinces of Canada led to a sample of 30 participants for this study. Yin (2017) posited that a minimum of three and a maximum of eight participants could be adequate for a case study qualitative research. The preferred research interview arrangement and location were mutually agreeable to the researcher and the interviewee. In this study, the participants received advanced explanations about the content of the interview. Inform consent was a requirement before commencing data collection. The participants received information about the interviews, requests for consent, and locations before the scheduling of any interview appointments. This act helped to provide the interviewee with a sense of trust, belonging, and empowerment. This gesture enabled interviewees to be free to confidently answer research questions without pressure.

Data Collection Instruments

I collected information from participants, with the goal to understand a phenomenon in rich details that could allow me to observe and document patterns and themes. During the semistructured interviews, I interacted with GSD leaders to gain rich understanding and focus on meaning from their phenomenon. A protocol with open-ended interview questions was based on the Tamimi et al. (1995) validated instrument for Deming’s 14 points. Questions pertained to current quality management practices, models, and standards to predict software product quality.

Data Collection Technique

Different authors described data collection techniques in diverse ways. Yin (2017) classified data collection techniques as in the forms of interviews, observations, documentation, archival records, and artifacts. I used audioconferencing, email, online format, telephone interviews, and document review. The semi-structured interview approach was a predefined consistent way to improve data quality while remaining open to pertinent related experiences and ideas that emerged during a natural process. The semistructured technique of data collection allowed the participant's experience, ideas, and opinions to emerge through the kind of in-depth approach typical in qualitative research. The initial open-ended question helped relieve unease, create cohesion, and develop rapport. The debriefing and follow-up questions during the interviews were probes for further information from participants' responses. Notations of the possible follow up questions were in the procedure. The use of semistructured interview techniques imply that approved protocols should guide the data collection process (Creswell, 2014).

Informed consent procedures with prospective participants adhered to the approved Walden University IRB procedures and best practices. The informed consent form conformed to protocols of the IRB to ensure consistency and validity. The informed consent guides may contain an introductory part, interview ground rules, and confidentiality statement (Creswell, 2014; Patton, 2015; Ranney et al. 2015).

Member checking is a data collection validation process to achieve dependability and reliability (Morse, 2015). Member checking may have shortcomings in a research process (Harvey, 2015). Member checking is sometimes recommended at the initial stage after data collection or introduced at the end of the data analysis (Varpio et al., 2017). The interview participants could review their data and the initial data interpretations and offered their feedback in a member checking process in this study.

The document review is a vital data collection technique in qualitative method because it has the benefit of being stable and specific (Yilmaz, 2014). I collected documents as research data from standard documents relevant to this study, such as software quality management records, policies, objectives, key performance indicator reports, non-conformance reports, quality evaluation reports, and risk registers. I requested that participants give me these documents at the end of the interviews. Document data may have some shortcomings or limitations because of data retrieval, bias, and access problem (Yin, 2017).

Research data will remain in secured electronic storage, cataloged to maintain originality and accuracy. I ensured that all collected data were in a locked system. The electronic database repository device, an external hard drive, was a means to store data for preservation, data protection, data security, and for easy retrieval. The collected data and information storage shall continue for 5 years before proper disposal. The confidentiality of the participants' information shall remain without disclosure.

Data Organization Strategy

In this research study, the data collection instruments included the semi-structured interviews and documentation. Qualitative data is multifaceted, unstructured, and rich in pattern and providing meanings in findings (Ishak & Baker 2012). I organized and tracked data using notes, recorder devices, and qualitative data analysis software. Qualitative software reduces the burden of data management and analyses (Fielding, Fielding, & Hughes, 2013). The use of computer assisted qualitative data analysis helps to manage large volumes of data, improve the quality of data, and creates trustworthiness (Houghton et al., 2017). I strategically organized data by using journaling to record notes and employed computer software, NVivo 11 software, to manage files, label, store, organize, and analyze data. The NVivo qualitative data analysis software was helpful with the effective data storage, easy access, and retrieval of information. The electronic database repository device, an external hard drive, was a means to preserve, protect, and secure data for easy retrieval.

The purpose of coding was to eventually be able to recognize themes in text. The researcher develops codes by examination, assigning names, grouping codes, and analyzing the categories of texts or graphical representation (Zamawe, 2015).

The qualitative data analysis is complex when large volume research evidence is involved using interpretivism paradigm (Carcary, 2011).

The research data and related materials will remain preserved for a period of 5 years before destruction, in accordance with the University research rules and regulations. These related materials consist of copies of participants' consent forms, scanned and uploaded electronic documents, recorded audio data, transcriptions, and notes, on an external hard drive, stored securely for a minimum of 5 years. The data deletion and destruction will help to protect and ensure confidentiality and privacy of the participants.

Data Analysis Plan

Qualitative data analysis includes the use of thematic analysis techniques to extract meanings from data to make informed decisions (Marriam, 1998; Stake, 1995). Castleberry and Nolen (2018) described the qualitative model for data analysis as compiling, disassembling, reassembling, interpreting, and concluding. In the qualitative data analysis, the flexible and methodical approach applies to the compilation of data (Yin, 2015). I used the thematic analysis approach for data analysis because of the methodical attributes. Data source triangulation for this study included data derived from interviews and document reviews. Multiple sources of data enable researchers to have rich evidence that help answer research questions (Joslin & Muller, 2015).

Compiling of Data

I compiled and organized interview and document data collected from GSD organizational leaders. Classifications of similar data included categories forming themes. To be familiar with the data, I listened to recorded interviews and transcribed files, generating codes for the development of themes. A researcher should be acquainted with research data by reviewing data multiple times to gain more insight (Acharya & Gupta, 2016). I used the qualitative data analysis software NVivo 11 because of multiple functions leading to the identification of themes, such as querying, data coding, and effective data organization and management.

Disassembling

The disassembling of data is an iterative process of filtering the data with notetaking to achieve broader grouping (Yin, 2015). Broader meanings in the patterns during data disassembling helps to answer the research question and provides more insights into the research topic (Yin, 2015). Qualitative coding included not only on interviews but also the field notes, journals, and results of the document reviews. I used the NVivo software to perform data coding, identification of patterns, and organization of themes from categories. The reporting of findings from analysis were through the presentation of texts and tables.

Reassembling

The reassembling is an iterative process of considering different approaches to themes and categories until the emergence of valuable ideas (Yin, 2015). Accordingly, the larger data groupings formed at code levels in hierarchical arrays. Elimination of the redundancies and misfits occurred with data that were a challenge to the robustness of labels and coding. I used the NVivo software to accomplish the reassembling aspect of data analysis for comparison and rival thinking. Inherent in CAQDAS was the ability to build hierarchies in graphical forms, matrices, and concept maps.

Interpreting

I interpreted the information by providing meanings to data arranged in themes. There are attributes that portray interpretation as comprehensive and good; the acts of completeness, fairness, empirical accuracy, added value, and credibility are features of an appropriate interpretation (Yin, 2015). I demonstrated creative thinking to evaluate new categories and produce quality data analysis, interpretations, and reports.

Concluding

I analyzed research data and drew conclusions based on a thematic analysis of data in relation to the significance of the study and the research question. I engaged in comparing research findings with the conceptual framework and existing theories, to provide answers the research question. I explored a common understanding of what GSD organizational leaders expressed they needed to mitigate the management problem of low software product quality to enhance customer satisfaction. Yin (2015) asserted that conclusions should be overarching statements that reflect a higher hierarchy of broader ideas.

Issue of Trustworthiness

In qualitative research, certain characteristics of trustworthiness form the evaluation criteria for scientific rigor, method justification, and unbiased transparency. There is no common acceptable standard by which to judge a qualitative research endeavor (Noble & Smith, 2015). The most accepted established criteria for quality evaluation of a qualitative the research was developed by Lincoln and Guba (1985) with the concepts of credibility, transferability, dependability, and confirmability (Susan & Rasulova, 2017). Moon, Brewer, Januchowski-Hartley, Adams, and Blackman (2016) identified the fifth criteria as authenticity in a quality research. These criteria improved rigor and ethical standards in this qualitative research process. The quest for new technologies for the different basis for qualitative characteristics of rigor are termed as desirability, consistency, impartiality, relevancy and application (Noble & Smith, 2015).

Credibility

Researchers use methodological strategies to ensure the trustworthiness of the research findings. These strategies are helpful to mitigate personal bias that may have influence on the findings (Noble & Smith, 2015). I triangulated data the qualitative data sources by comparing different perspectives of 30 GSD organizational leaders from interview responses with review documents and current reports. The research participants had opportunities to review the transcribed data and initial interpretations to confirm accuracy. Member checking and feedback in a qualitative study enhance research findings through a process of respondent validation (Marshall & Rossman, 2016). To ensure the impartial outcome of the research, I engaged in reflexivity during the systematic construction of ideas.

Transferability

Transferability depicts the cultural context and meaning of phenomenon that are easily transferable to other settings or groups who can derive meanings. A transferable context should be clear and representative of the framed process (Houghton et al., 2013). I provided a detailed description of the participants to provide context for the research findings. The account included the description of the variation in participant selection. Assumption of generalization is not typical for a qualitative researcher (Marshall & Rossman, 2016). The qualitative researcher's responsibility is not to prove the applicability of the research findings (Lincoln & Guba, 1985). I did develop descriptions of contextual elements to form a rich and thick description, for others to judge the possible application of findings to other contexts, situations, and populations.

Dependability

Dependability is the process of describing and taking into consideration the changes that occur during the duration of the research study. The use of thick descriptions ensured rich accounts of participants' perspectives. I applied data triangulation to improve the quality of findings from research data. Using different data source methods will lead to comprehensive findings (Noble & Smith, 2015). I transcribed all interviews and asked participants to review their transcript and the initial interpretations of data. The use of a dialogue strategy between the researcher and the participants is valuable in a research process (Harvey, 2015). Transcribing all interviews improve the quality of research data (Rosenthal, 2016).

Confirmability

Confirmability is the level of acceptance by other researchers of the research findings. Researchers should remain neutral, unbiased, and transparent to enhance the confirmability and validity of findings (Patton, 2015). I heightened confirmability by engaging in the iterative process of rechecking data, taking notes of contradictions, making observations, and drawing comparisons with existing knowledge. I encouraged member checking so that the findings represented participants' intended meanings. I used assessment criteria of qualitative methodology, including addressing credibility, transferability, dependability, and confirmability to buttress research quality and validity. Confirmability pertains largely to the participants' responses (Lincoln & Guba, 1985). I included verbatim quotes of the participants' accounts in the report of findings to enhance confirmability.

Ethical Procedures

I received IRB approval number (06-16-20-0461277) before the commencement of data collection and recruitment of participants. I was available to GSD organizational managers to discuss the study and obtained signed informed consent agreements. Informed consent forms included consent for recoding interviews, and the review of organization quality documents. Yin (2009) attested that obtaining inform consent from the participants represents a formal request to volunteer and to participate in the study. I participated in the ethics course to meet the Walden University research requirement. I ensured that this study complied with ethical principles and guidelines for the protection of human subjects of research, as stipulated by the National Institutes of Health Human Research Protections. My interview setting was in a secure environment for safety, privacy, and confidentiality. All collected research information from the interviews and documents were confidential. I did not use the personal research information for any reason other than for the purpose of this study. The research findings do not include names or personal identifiers in the reports. The data shall remain secure, locked and only accessible by me alone. All electronic data files will remain secure and protected with secret codes on my computer for 5 years, as stipulated by the University before destroyed.

Summary

In Chapter 3, I presented the methodology and design for the study. Discussion included the data collection and analysis that led to the answers to the research questions. I used a qualitative research method to gather data about the lived experiences of GSD organizational leaders in a distributed setting. The 30 participants in the sample were knowledgeable about the subject investigated for this study and met the inclusion criteria. The justification for the selection of the sample, ethical concerns, privacy, safety, and confidentiality were parts of the chapter. I used a purposive sampling technique to recruit 30 participants from six GSD Canadian organizations located across six provinces. I discussed the collection of data and defined my role as a primary instrument of the research. Chapter 4 includes the results derived from the analysis of the research data. I describe the sample, the data collection results, the analysis applied to the data, and the findings, explaining and illustrating the thematic results that emerged from the study.

Chapter 4: Results

Introduction

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding, shared by managers in Canadian GSD organizations, of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction. The overarching research question for the study was: What are the common understandings of what GSD organizational leaders need to meet software product quality in Canadian GSD organizations? The search for an answer occurred to discover what GSD organizational leaders need to mitigate the challenges of low software product quality and enhance customer satisfaction. The second question was: How can common understanding be gained from GSD organizational leaders? And the third question was: How do GSD organizational leaders predict the outcome of software product quality? The answers to the second and third questions addressed a common understanding shared by GSD managers about how to predict the outcome of software product quality.

Canadian global software development organizational leaders in six organizations across the different provinces completed interviews, during which they answered questions about their professional experiences in software developments. The interviews took place through Zoom meetings and the telephone. By answering interview questions, the GSD managers expressed their perceived management problems in software development and shared how those problems impacted software product quality and customer satisfaction. They answered interview questions related to the mitigations needed to continuously improve the global software development life cycle and to achieve a high quality of software product that meets customer expectations.

Chapter 4 includes the description of the research setting and sample. The process of data collection from the participants in the study lead to an added discussion of the steps to enhance the trustworthiness of the study. The results of the data analysis process and findings from the analysis complete the chapter.

Research Setting

The research sample included 30 knowledgeable IT professional managers involved with global software development in Canadian organizations. I recruited participants using the publicly available contact information on a professional social media platform. As a member of this professional body, I expanded my search capability for potential members who appeared to be eligible to participate in the research study. After identifying potentially eligible members to recruit for the study, I sent each prospective participant an invitation email to the designated inbox on the social media platform. I sent the invitation emails through my Walden University student email address, which I used for subsequent communications. Prospective participants who were interested in the study replied to the message in their inbox, thereby sending me additional personal email addresses for ongoing communications. After I confirmed their eligibility and sent the informed consent form, prospective participants indicated interest in opting into the study and becoming part of the purposeful sample. Thirty eligible members formally consented to the informed consent form, indicating their agreement with a reply email stating “I consent.” A follow-up email to each of the members of the sample included the request to schedule the date and time for the interview at each of the participant's convenience.

The recruitment process and interviews occurred according to the data collection procedures proposed for the study. The personal challenge of a few participants was that during interviews, interruptions by family members occurred, such as hearing voices of children in the background. On three occasions, participants asked to discontinue the interview sessions due to the interruptions of children. We continued the interview after about an hour for two participants, and for another participant, we postponed the interview to the next day. Most of the participants were reportedly working from home due to the impacts of the COVID-19 pandemic. These few personal interferences did not appear to influence the quality or interpretations of data that led to the study results.

Demographics

This qualitative exploratory multiple case study occurred to reveal a common understanding among GSD organizational leaders. Interview questions pertained to their experiences with the challenges of low software product quality and achieving customer satisfaction. The recruitment technique led to a heterogeneous sample (see Table 1). Alphabetical pseudonyms for organizations are the letters A to F. Representation of the participants is with alphanumeric pseudonyms, P1 through P5. Included in the demographic profile are the participants' educational achievements in the fields of study and current job titles. The array of 30 participants included software development consultants for government and institutions, software company executives, software managers in organizations, software developers, software architects, product managers, software quality assurance analysts, and software business analysts. The number of years participants engaged in global software development ranged from 6 to 16 years.

Table 1. *Participant Demographic Outline*

Participants and organizations	Level of education	Years of experience	Job titles
AP1	MSc	15	Project manager
AP2	PhD	12	Consultant
AP3	BSc	10	Business analyst
AP4	BA	9	Software developer
AP5	MSc	10	Software vendor
BP1	PhD	14	Project manager
BP2	BSc	16	Product manager
BP3	BSc	8	Consultant
BP4	Diploma	11	Project manager
BP5	MSc	13	Software developer
CP1	BA	14	Project coordinator
CP2	MSc	15	Project manager
CP3	BSc	10	Consultant
CP4	PhD	12	Consultant
CP5	BSc	6	Software manager
DP1	BSc	12	Project manager
DP2	BA	9	Business analyst
DP3	PhD	13	Cloud engineer
DP4	Diploma	14	Software engineer
DP5	BSc	11	Project manager
EP1	PhD	15	Consultant
EP2	MSc	10	Software developer
EP3	BA	6	Business analyst
EP4	Diploma	7	Software vendor
EP5	MSc	13	Project manager
FP1	BA	16	Project manager
FP2	BSc	8	Software engineer
FP3	Diploma	7	Project coordinator
FP4	BSc	10	Cloud engineer
FP5	MSc	13	Consultant

Additional demographic information collected included ages, gender, and the province in which the participants' companies operated. The ages of participants were between 29 and 57 years. Seven female participants and 23 male participants were in the sample. The participants were from different provinces in Canada, including Ontario, Alberta, Manitoba, Quebec, Saskatchewan, and British Columbia. In addition to formal college or university educations, all participants had various certifications from accredited software development bodies.

Data Collection

I used a case study design to explore the in-depth understanding of the participants through semistructured interviews with open-ended questions. The questions were based on Deming's 14 points (1986), SDLC, and ISO/IEC 25000 Series standards (see Appendix). I recruited a purposeful sample of 30 participants through publicly available contact information and a professional network social media platform.

A subscription to the media platform increased access to more professional organizations and members. I identified the six software development organizations and reviewed the experiences and qualifications of the employees to recruit an appropriate purposeful sample. I posted the approved research invitation email to the personal inbox of the intended professional members that prompted them to either accept or reject the invitation to participate in a research study. Those who indicated an interest in the research provided an email address in my inbox for a formal invitation. I used my Walden University student email address to officially send them an invitation email and an approved consent form email. Interested members consented to participate in the study by replying to my email with the words “I consent.” I sent approved follow-up emails to schedule dates and times for the interviews at their convenience. The members indicated their preferred mode of communication for the interview, either through a telephone call or audio conference tools like the Zoom meeting. I asked for a telephone number based on participant preference or scheduled a Zoom meeting for the interview.

I collected data based on semistructured interviews with 30 participants, which included five participants from each of the six Canadian organizations. An open-ended question format was useful to collect in-depth narrative data during the interviews. I used a Sony ICD-UX570 recorder to record the interviews. The average interview duration was 45 minutes. The longest interview lasted 75 minutes and the shortest interview was 35 minutes. I stored the recorded interview audio files in my personal password-protected computer.

The problem of time zones affected two scheduled interviews. Two participants forgot the interview time, so I rescheduled and conducted the interviews on other dates. Another challenge was family member interferences during some interviews, especially by children. The participants attributed these interruptions to working from home and children not being in school due to the COVID -19 pandemic. All the participants provided answers to all the interview questions. Revisiting those questions before the end of the interviews helped participants provide complete answers to all the interview questions. During the interviews, journaling and active listening skills were useful to document vital information. Some of the participants volunteered documents on global software development, cloud platform, and SDLC frameworks to support their views on the subject.

I transcribed recorded interviews into verbatim records of textual data. I sent the interview transcripts and initial interpretations of data by email to all participants for member checking, and validation. One participant responded with a corrected typographic error in the transcript. This data collection process conformed to the initially proposed data collection procedures described in Chapter 3 without any deviations. I transferred the transcripts to NVivo data analysis software for more organization and data coding.

Data Analysis

The analysis of the data occurred to identify the common understanding of GSD organizational leaders about software product quality and customer satisfaction. Data were from the expressed perspectives and experiences of managers in Canadian GSD organizations and related documents. Results from analysis of the data collected from 30 participants from six organizations across six provinces include the aforementioned pseudonyms for the 30 interviewees (AP1-AP5, BP1-BP5, CP1-CP5, DP1-DP5, EP1-EP5, and FP1-FP5). Eight themes emerged from the data analysis process. The next section includes the discussion of the data analysis process used in this study to identify the main themes from emerged from the data.

The six-step thematic analysis process by Braun and Clarke (2019) was useful in this study for the qualitative data analysis process. I audio-recorded data from the 30 interviewees and transcribed the verbatim data using Microsoft Word. The transcribed data then transferred to NVivo software for qualitative analysis. The first step of the data analysis process was familiarizing myself with the collected information. Specifically, I examined and took notes on the obtained interview transcripts. According to Braun and Clarke (2019), the first-step entails knowing the data before commencing on analyzing individual items. In the second step, I started coding the collected data by highlighting sentences and phrases in the interview texts and creating “codes” or shorthand labels to describe the shared content expressed by the 30 interviewees. Based on the two interview sections used in this study, different codes emerged, as summarized in Table 2. Codes stemmed from the inclusion of words and phrases that all 30 participants similarly expressed when they answered the interview questions.

Table 2. *Codes Identified from the Interview Data*

Codes	Contributing participants	Interview questions
Creating constancy of purpose, adopting the new philosophy, ceasing inspection and requiring evidence, improving supply quality supplies, continuously improving production, training and educating employees, supervisors helping employees, driving out fear, eliminating boundaries, eliminating the use of slogans, eliminating numerical standards, allowing employees to be proud of their work, encouraging self-improvement, and committing to the ever-improving quality	30	Interview section A: Deming's 14 principles of quality management
Software development lifecycle, software development method, potential hurdles, new practices, the solution to problems, Deming's Quality management used in GSD, quality management method, predictive software product quality, software quality management, internal and external software quality, product quality, covering quality aspects	30	Interview section B: SDLC or ISO/IEC 25000 quality series

The codes became nodes using NVivo software, incorporating all similar information from participants into the identified nodes. Every node identified represented the main feeling or idea expressed by the participants regarding the Deming's 14 principles of quality management and the SDLC or ISO/IEC 25000 quality series. I continued to examine each transcript of every interview and document, noting relevant codes similar to the identified ones highlighted, using common phrases or similar sentence information. After the examination of all the interview transcripts, the information collated into groups identified by codes. The codes were important in assisting with developing a condensed overview of the common meanings and main points that recurred throughout the dataset of 30 interview responses.

The third step involved generating themes from created coded categories. The process of generating themes entailed examining the created codes and identifying common patterns among them, and in the process formulating important themes from the obtained interview responses. Several codes combined to form a single theme as presented in Table 3. Following this iterative process, eight themes emerged from the data, as follows: (a) develop clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate life cycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product quality. The identification of all the eight themes was contingent upon the contributions from all the 30 participants to the coded categories, such that every participant contributed to the data that led to each emergent theme.

Table 3. *Emergent Themes from Coded Interview Data*

Coded categories	Themes	Interview questions
Create constancy of purpose, adopt new philosophy, cease inspection, require evidence.	Develop clear purpose and work principles	Interview section A: Deming's 14 principles of quality management
Improve the quality of supplies, continuously improve production, train-educate workers.	Improve processes and employee skills	30 participants identified each theme
Supervisors must help people, drive out fear, eliminate boundaries, eliminate the use of slogans, eliminate numerical standards.	Develop personnel Management Strategies	
Let people be proud of their work, encourage self-improvement, and commit to the ever-improving quality.	Promote autonomy and personal worker development	
Software development lifecycle, Software development method.	Formulate lifecycle and development techniques	Interview section B: SDLC or ISO/IEC 25000 quality series
Potential problems and hurdles, limitations to effective processes.	Identify challenges	30 participants identified each theme
Develop new practices, identify the solution to problems.	Formulate solutions	
Demin's quality management in GSD, quality management method; predictive software product quality, software quality management, internal and external software quality, product quality, covering quality aspects.	Focus on product quality	

The fourth step entailed reviewing themes from the codes and confirming whether they were accurate and useful representations of the data, and that major themes emerged from the contributions of all the participants' interview data. To achieve this, I examined the data and compared data to the identified themes to confirm that important or relevant items were not missing. The examination ensured that the themes adequately reflected the collected interview data. Focus was on any need for changes. However, the identified themes appeared to represent the main insights shared by all the participants regarding the common understandings of what GSD organizational leaders need for software product quality in Canadian GSD organizations.

The fifth step was defining and naming themes. The process of defining themes occurred to formulate the exact meaning of each theme and to determine how they reflected the comprehension of collected data. The process of naming themes involved creating an easily understandable and succinct name for each theme. Reflecting on the presence of the themes across the dataset entailed consideration of the possible ranking of themes, based on prominence in the data. Given that the purpose of the study was to reveal a common understanding, the emergent themes were those ideas that were present across the dataset in all cases.

Any potential discrepant cases, such as missing information on some questions, were considerations throughout the coding and thematic analysis process. The discrepant cases emerged from variations in software development processes and quality management techniques that were unique to each of the interviewed participants. Also, due to variation in personal experience and understanding of each of the interview questions, participants might have attached a different meaning to quality management and life cycle development. These concerns were addressed during the naming of themes, based on the identified themes and the semistructured interview questions. In the process, I included discrepant findings pertaining to life cycle development and software management under development methods and software development life cycle unique to specific companies. Finally, the last step involved writing up the obtained results and discussing each theme in light of the formulated research questions.

Evidence of Trustworthiness

Credibility

Credibility in a qualitative research process is the extent of consistency, correctness, and plausibility in an agreement between the researcher and the participants. Following Walden University's IRB approved methodology for data collection helped to enhance credibility in the research process. The adoption of recognized research methods and the use of an appropriate strategy to select participants ensures credibility (Chowdhury, 2015). The purposeful sampling of 30 knowledgeable GSD participants who experienced the phenomenon of study enhanced confidence in the study findings. The use of the member checking process enhanced the credibility of findings. Triangulation of data involved my review of the documents provided by participants and the comparison of research data with existing literature and reports.

Transferability

Transferability is the culture of transferring meaningful contexts of a phenomenon that is comprehensible, reliable, dependable and useful to another group for appropriate judgment and applicability. The measure of usefulness by the group in another environment pertains to transferability (Kihn & Ihantola, 2015). The inclusion of the rich accounts of the participants helped to showcase originality and trustworthiness in the research data and findings. The act of providing rich descriptions of elements and context helps other judge appropriate transferability of findings. Details about the sample, research steps, and findings represent information for other researchers to use to evaluate and possibly apply to future situations or populations.

Dependability

The dependability is the stability of data over time that would create reliability, consistency, and confidence that allow repeatable study results (Chowdhury, 2015). All the participants had equal opportunities to answer the same questions from the research protocol of interview questions. Some participants answered all the questions while others skipped questions. Dependability stemmed from alignment of the data collection process with a consistent protocol and firmly established research methodology. Every step and criterion for the selection of participants was clear, implementable, traceable, and repeatable, making it very easy for replication by future researchers. The inclusion of dates in all audio files, transcripts, field journals, notes, and other research documents will enable audit trail in the future.

Confirmability

The confirmability is the process of confirming research data with the participants to establish truth and value. Each participant received a follow-up email to engage in the member checking process. The data accuracy check of all the participants in the research study occurred after transcribing recorded interview audio files into Word documents. All participants had time to review their interview transcript and the initial interpretations of data and to provide feedback for validation purposes. The participants provided feedback by responding to the email request. I documented a comprehensive research process to established audit trails that could enhance the confirmability of the research process. I included the participants' accounts through verbatim quotes in the findings to enhance confirmability. Research reports with in-depth methodical detailed descriptions and documentation enhance confirmability (Bratich, 2018; Chowdhury, 2015).

Study Results

The current section includes the in-depth description of the results, which are the eight themes identified in the data: (a) develop clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate life cycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product quality. The themes include important attributes that GSD organizational leaders need to meet software product quality in Canadian GSD organizations, leading to a common understanding gained from GSD organizational leaders. The results include aspects of how GSD organizational leaders predict the outcome of software product quality.

Research Question 1

The first central research question was: What are the common understandings of what GSD organizational leaders need to meet software product quality in Canadian GSD organizations? The subsequent subsections include the discussion of the first four main themes identified in the data from the interview responses, summarized in Table 4. These first four themes include the need by organizational leaders to develop clear purpose and working principles, improve processes and employee skills, develop adequate personnel management strategies, and promote autonomy and personal development.

Table 4. *Themes: Software Product Quality*

Themes	Issues
Develop clear organizational purpose and work principles	Communicate aim and purpose, regular training, constant inform sharing, publish new products, include quality into the process, improve quality of incoming materials, do not rely on prices alone, minimize costs, work with single suppliers
Improve internal processes and develop employees' skills	Promote skills development, budget allocation for training, on-the-job training, external training, online training, continually improve quality and services, reduce costs and errors
Develop appropriate personnel management strategies	Eliminate fault-finding, promote a positive environment, checks and controls, eliminate fear, create trust, formulate support systems, provide feedback
Promote autonomy and personal worker development	Address unhealthy competition, promote autonomy in individual goals, facilitate education and knowledge, management commitment to improvement and learning

Theme 1: Develop clear organizational purpose and work principles. To meet software product quality in Canadian GSD organizations, the interviewees shared that leaders need to develop clear purpose and working principles. In meeting this goal, GSD organizational leaders expressed the need to focus on three factors: (a) creating constancy

of purpose in product and service improvement to stay in business, become competitive, and provide jobs; (b) adapting new philosophy; (c) ceasing dependence on inspection as a way of achieving quality; and (d) improving the quality of incoming material instead of awarding operations based on price. First, regarding creating constancy of purpose, the interviewees share that organizations need to inform all employees about the purpose and aim of the organization.

These observations were evident during the interview responses. For example, AP5 shared that “Top management documents the short and long-term plans in the company strategic plan and organization quality plan. Trainings are organized for employees through inductions, seminars, workshops and virtual online courses.” BP3 said, “When the top management wants to introduce a new software product, so there's always a campaign through emails. Before they select the product, everyone is informed that there is a new product to be introduced.” AP2 shared that constancy of purpose is achieved through information sharing and “employee training where managers recommend the people within the team or inform everyone there is a new development or new application.” CP1 further noted that, “awareness and training are put in place, in addition to dialogue with the external consultants to see how we can provide awareness training for our developer.” Similar observations on the need to create and publish the aim and purpose emerged from the data offered by other interviewees, including DP 2, DP4, EP1, EP3, EP5, FP1, and FP4, where constancy of purpose was central to organizational success in product quality development.

Second, interviewees noted that organizational leaders need to adopt a new philosophy about technology. Specific focus was the need to ensure everyone learns the new philosophy, focusing on defect prevention, as opposed to defect detection. Everyone should be involved in the quality journey. These insights were shared by interviewees, such as AP1, BP3, CP 1, DP 4, AP2, EP1, EP3, FP2, and FP5. For example, AP1 said, “One of the goals is to make sure that once new technology comes in, over 80 per cent of our employees are trained. We motivate them to be able to achieve this.” BP3 claimed, “There's always a set of individuals that always review the process, and always examine how things could be better improved.” CP1 explained that, “commitment for quality and regular learning is very important because events have shown us that when we don't address quality especially at the initial phase for application development.”

Third, participants, such as AP1, AP2, BP1, CP1, DP3, EP1, DP1, EP4, FP3, and FP5, shared that organizational managers need to cease depending on inspection as a way of achieving quality. Instead, participants indicated that leaders should ensure quality is built into the product planning process. Most participants considered the mass inspection unreliable and time-consuming. Instead, efforts need to reduce failures and subsequently eliminate acceptance sampling. For example, AP2 shared that, “If you don't include the quality at the beginning that means you are going to be patching the software as you go along. So, ensure that when they are doing the design, everything is about quality.” CP5 said, “Our top management ensure the building of quality at every developmental phase. They do not wait until the end before implementing the quality of the product. This will be a disaster should we wait.” DP3 explained, “In this regard, you don't want to put quality as adding special things, it must be built-in. What I mean by that is that quality must be built right from the beginning." According to EP4, "quality has to be built into the planning process to start with and throughout the software development life cycle." Similar observations were also made by EP 5 who noted that "mass inspection costs more because you have to start inspecting this product one after the other, quality need to be included from the study to avoid mass inspection." FP3 also shared that, "in terms of quality for procurement perspective, you ensure that you have the necessary SLAs in place and you are also ensuring that you have factored in quality testing as part of the engagement." These observations revealed that GSD leaders need to ensure that quality is built-in at every stage of software development to achieve product quality.

Fourth, interviewees also agreed that GSD leaders need to focus on the quality of incoming materials, instead of basing their awards on prices only. That is, there needs to be meaningful measures of quality, along with the prices. Also, there is a need to minimize total costs by working with a single supplier. For example, CP1 noted that "we usually go by quality because of the better quality sometimes, the higher the price and so, we use an evaluative method to address those two variables in terms of selecting our suppliers." FP2 shared that "quality is rated higher than the price in my present and previous organization because of the problem of poor quality of software." AP4 shared that, "Every company must have its method to examine quality. They look at the price, too. Every company has the criteria that they use to select vendors." AP5 said, ""Both quality and price are used to determine potential suppliers." BP1 explained, "Price is sometimes, it's not the primary factor in the selection of suppliers. In a long time, relationship, quality of their services, reputation, integrity and all these things are a very big factor in selecting those suppliers." BP5 similarly stated, "Quality is taken of high important before price, though the price is equally important in decision and selection of suppliers."

Theme 2: Improve processes and develop employees' skills. Interviewees also agreed that organizations are more likely to be successful if they improve their internal processes and enhance employee skills. The GSD leaders may achieve this goal through two main approaches: (a) institute on-the-job training and (b) constantly improve the system of services and products. First, participants agreed that GSD leaders need to train and educate workers to ensure they are to perform their job correctly. For example, AP5 shared that, "Management establish quality management awareness training for all employees, defined and organize as needed skill training for employees. Recommend online specific periodic training for the upgrade of knowledge." CP3 said, "On an annual basis, the top management allows the team to go lookout for certain training, relevant training that has reviewed and approved. And they can go for the training be it online, be it physical training."

Interviewees also emphasized that GSD leaders need to ensure constant improvement processes to achieve product quality. According to FP2, "there is a need to identify problems and constantly work towards better systems of service delivery and production." BP2 noted the need to ensure "continuous reduction of waste and facilitation of quality improvement in every process." Also, EP4 indicated the need by an organization to work towards "decreasing costs of productivity and regularly improve each planning process." CP3 also claimed, "We constantly check each of the functionalities to see if it meets the requirements. If it doesn't, we go back to the vendor. So, a vendor keeps doing that until he gets it right." DP1 said, "When you put software out on the market, people subscribe to it. Then you open what they call a feedback loop. … you want the response of the users that are using your software for continuous improvement."

FP5 shared, "We perform frequent engagement by in-person meetings or virtual meetings or conferencing to gather requirements. We also get feedback, in the same way, to keep improving."

Theme 3: Develop appropriate personnel management strategies. Ability to effectively manage workers is key to long-term product quality success. Findings from interview responses included the need by GSD leaders in the organization to have adequate leadership, create trust and confidence among workers, eliminate fear, and optimize the productivity of their teams. Interviewees shared that adequate personnel management strategies may be realized through (a) teaching and instituting proper leadership; (b) driving out fear and creating trust; (c) optimizing the efforts of staff, groups, and teams; (d) eliminating targets, exhortations, and slogans based on zero defects; and (e) eliminating work standards and numerical quotas on employees based on quantity and focusing on quality.

Insights by DP2 indicated that GSD organizational leaders need to improve their supervisory skills instead of focusing on "fault-finding and negative attitude." Participants AP4, BP1, CP3, and EP4 also emphasized that supervisors need to ensure a supportive and positive atmosphere. Thus, these interviewees felt that the GSD leaders should teach and institute appropriate leadership when leading their employees. For example, BP3 shared that, “It's the manager's job to ensure that the employees in a project are adequately supported and led towards achieving organizational and product quality goals.” CP3 said, “Usually supervisors need to have certain skills, management skills, people skills.” CP3 added, “And it's one basic requirement for recruiting supervisors and leaders and team leads because they need to be able to bring out the best in the team and manage the team.”

The GSD leaders further need to drive out fear, create a conducive climate for innovations, and create trust. According to AP3, DP1, EP5, and FP3, working under fear drives out creativity among employees. Thus, productive outcomes are met when there is effective two-way communication. CP3 shared that “when employees have issues in certain fields, leaders need to be open enough to assist.” Such an approach may relieve fear of job loss, fear of change to individuals, and ensure that performance appraisals do not hinder employee productivity. According to CP1, organizations can create trust by creating “a knowledge management share point site, where employee express their ideas and other work to make work easier.” Further, BP3 noted that there should be checks and balances where “effective controls must ensure that at any point in time the employee is heard, or employee is not working under fear.”

Moreover, GSD organizational leaders need to ensure they optimize the teams and groups they manage. Participants CP1 and DP4 noted the need to break down barriers between staff and departments. CP1 explained “we usually hold quarterly town-hall meetings where share experiences and break down the barriers between departments." Moreover, DP5 shared that they regularly hold "inter-departmental meetings to share ideas and information on a project is encouraged." FP2 elaborated that "the company breaks barriers through long-term strategies of collaborative working to facilitate teamwork and positive group engagement." Eliminating slogans was essential as DP2 noted that "using posters and exhortations on workers create adversarial relationships." Also, EP3 indicated that "leaders need to avoid slogans since nice-sounding phrases should not replace effective organizational leadership and people management." For people in management, FP4 shared the need to eliminate numerical quotas because they focus on quantity which "results in poor product quality in product and service development." Thus, these findings emphasize the need by GSD leaders to ensure they have appropriate personnel management strategies to achieve desire software product quality.

Theme 4: Promote autonomy and personal worker development. The need to ensure employee development and autonomy was another key theme that GSD leaders considered helpful for achieving software product quality. CP3 and FP1 noted the need by organizational leaders to remove barriers that rob employees of the pride of workmanship, such as doing away with the annual merit system and rating. AP5 shared that employee autonomy is facilitated by ensuring "work plans are developed by every employee" meeting "the intended goals for the year." FP4 noted that "every individual interest and every individual goal is aligned to the broad objectives to avoid unhealthy competition."

Besides working to promote individual autonomy and pride, GSD leaders also need to focus on encouraging self-improvement and education for all employees. BP4 indicated that leaders “facilitate a vigorous plan of education meant to improve skills for everyone.” DP2 also elaborated that “organizations do not only need people but the constant improvement in education to meet the changing competitive environment.” Finally, management needs to take responsibility for transformation, quality improvement and productivity. According to EP2, “management needs to provide quality improvement process towards active transformation.” FP1 shared that transformation is achieved when managers “use external consultants to perform quality improvements.” These insights indicate the need of GSD organizational leaders to promote worker autonomy, personnel development, and management responsibility in organizational transformation, as approaches towards long-term product quality.

Research Question 2

The second main research question was: How can common understanding be gained from GSD organizational leaders? Findings from interview responses revealed the need to focus on three aspects related to (a) formulating effective life cycle and development techniques, (b) identifying problems early, and (c) formulating solutions to identified problems. Table 5 includes a summary of the themes pertaining to a common understanding from GSD leaders about issues associated with software quality development.

Table 5. *Themes: A Common Understanding*

Themes	Issues
Formulate lifecycle and development techniques	Identify an appropriate lifecycle, have suitable development methods, consider Agile, Waterfall, or Activate processes
Identify challenges	Collaboration issues, coordination of multiple groups, variation in time zones, quality monitoring problem, managing diverse workers and ensuring reliable communication.
Create solutions	Improve communication, break cultural barrier, automatic testing tools, education, self-improvement, standardization of processes

Theme 5: Formulate lifecycle and development techniques. The interviews emphasized the importance of formulating appropriate lifecycle and development techniques with an emphasis on the use of agile technology. AP1 shared that “Agile enables us to do continuous incremental deliveries to the customer and it's effective because the customer is the center of whatever we do.” However, AP2 noted that in their organization they prefer Activates to Agile because “it improves the mission groundwork where you find out what, prepare the project, and you interview the client to get the requirement - you know what they want. That gives you an idea of what you're going to do.” CP3 also discussed the importance of having suitable development technology such as Agile that is “very operationalized since you develop, someone tests, and another remediates the issues known, then it goes back into the building.” According to FP1,

every company uses unique quality management methods, such as Six Sigma and Waterfall in place of an agile development approach.

Theme 6: Identify challenges. In-depth understanding of software product quality may also be achieved by identifying potential challenges in the organization. For example, AP1 and EP3 noted that GSD leaders need to take note of potential challenges and problems such as communication and cultural differences. AP1 noted that potential challenges include “knowledge and coordinating projects in different time zones [which] could be very complex because most of our meetings, we don't usually do it in a particular time zone.” According to BP2, “one of the challenges we encounter is too many tools that don't always work well together. In most of the companies, there are also security challenges to effective software development.” DP4 shared that “the challenge for distributed work is that you don't see if someone's goofing off as they are unsupervised.” FP1 noted that in their company “one of the biggest challenges is its quality assurance,” while CP5 noted that major challenges are “communication and coordination.” Considering the different challenges unique to every organization, interview participants noted that one key aspect in achieving software product quality is identifying potential challenges before they emerge and formulating suitable solutions.

Theme 7: Create solutions. Solutions to problems is another important consideration for GSD leaders, such as developing new practices and identifying solutions to emerging or potential problems. According to AP1, potential solutions for GSD leaders may include “constant collaboration within the team and with the customer, and then using automation like Devox, REP in solving most of those problems. Improve our communication, break cultural barrier, improve coordination and then automation will help." CP3 added that "the most important thing is standardization because standards will ensure a lot of things. If you have a standard, then it covers requirements, it covers quality, it covers security, it just covers a lot of things." AP 3 noted that "what is most needed is a very organized structure, in terms of the codebase. In terms of the source code, an organized structure that supports multiple people working on the same thing." CP5 noted a potential solution as "ensuring full implementation of quality management in software development." BP2 further shared that "what is needed is more of the automated tools for testing," while FP3 emphasized the need to institute education and self-improvement" to address emerging challenged in software product development and quality. Thus, GSD leaders need to consider potential solutions for each scenario emerging from each process of product and service development.

Research Question 3

The third research question was: How do GSD organizational leaders predict the outcome of software product quality? Findings indicated that the outcome of software product quality stems from examining it by considering seven quality considerations. These seven quality considerations include (a) Deming's quality management, (b) quality management method, (c) predictive software product quality, (d) software quality management, (e) internal and external software quality, (f) product quality, and (g) covering quality aspects. Together these ideas coalesced into the eighth major theme.

Theme 8. Focus on product quality. The seven quality considerations emerged from the data as the means for focusing on product quality and predicting the outcomes of software product quality. For example, CP3 agreed that “planning is very key, designing is key and coding, testing, and deploying [is] relevant to achieve [quality].” AP1 noted that product quality may be achieved through “requirement analysis in addition to system design, system development, deployment and maintenance. But because we use Agile methodologies …we do it as the project progresses.” Therefore, a combination of approaches may be appropriate, as explained below.

Deming’s quality management was an important approach expressed by participants for predicting the outcome of software quality development among GSD leaders. According to AP1, the product development is initiated and then “checks made and if there is any problem, the designs are rechecked.” However, CP3 noted that in their organization, they do not consider such quality outcome measures as Deming, “but they are part of the entire system. So, we have established policies, procedures and processes.” AP3 noted the need to develop “milestones to make sure that at every stage, what is expected of the design is complete before we go to the next one. Some of them will call it agile.” Effective quality management methods such as Activates, Lean, and Six-Sigma are useful to enhance quality. FP1 also indicated that their company uses “Kanban because we have what we call Kanban Dashboards that are monitored at regular interval.” EP3 shared that “Deming's principle is one quality management method, we also consider TQM by Kaizen and Crosby, which are under the ISO 9000 and 9001.”

The GSD leaders also need to consider and apply new things to ensure software product quality and customer satisfaction in the GSD environment. CP4 noted that they use “Deming's tool in addition to automated testing, continuous delivery, and continuous deployment to ensure that the quality of the software is increased." DP2 also discussed the importance of new approaches to facilitate "training of the developers, going into the market to see what improvement, getting feedback from users, from customers." The interview participants also explained that they predict software product quality using the ISO/IEC 25000 series. For example, AP1 shared that this is "the newest standard and we use this model to predict software product quality such as maintainability, reliability, portability, and even maintenance." BP4 also shared "We use ISO/IEC 25000 series to predict and ensure maintainability of the software." CP5 said, "We use the capability maturity model CMM/ISO standards. I think is effective and covers all vital quality attributes." DP3 noted that software quality is evaluated using "software development lifecycle which is more projected, and it's more structured since it easily flows from one stage to another when ensuring product quality"

Additional considerations by AP4 in software quality management include focusing on using "teams that are in different locations to reduce costs and to get the best of quality." Further, CP3 shared that "the common understanding is having an effective solution that solves the problem at less cost using less resources. At the management level in terms of quality, they want to be sure that whatever solution is developed is quality." Additional focus was on external and internal software quality where DP3 indicated that the process is "pretty technical … a CTO, CIO, or CIS may only be interested in the business side of global software development leaving the middle-level management and the team to understand the technical part or technical components of global software development." AP1 also indicated that the intention of the focus on product quality is to promote customer needs where BP2 noted that they "give it a higher weight because potentially that will tell us how effective or cost-effective the software is. Then the functionality of the software too, we give it a weight that is higher to achieve desired quality." Therefore, GSD leaders need to ensure their models effectively predict product quality based on the set quality standards such as ISO/IEC 25000, CMM/ISO standards, and other internal software development life cycle standards that align with company goals.

Summary

The research sample included 30 knowledgeable IT professional managers involved with global software development in Canadian organizations, who provided the data for this qualitative multiple case study. Represented in this sample were leaders from six companies across six different Canadian provinces. Eight themes emerged from the process of data analysis. These eight themes emerged from the data to find answers to the main research questions of this study. There is a common understanding among the Canadian GSD organizational leaders in the sample that, to meet software product quality goals and enhance customer satisfaction, there is a need to: (a) develop clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate lifecycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product quality.

Chapter 4 began with the reiteration of the purpose statement and research questions foundational to this study. Central to this chapter were the descriptions of the research setting, sample demographics, data collection, and data analysis, with a discussion of the evidence of trustworthiness. The focus in the chapter was on the report of the results and findings from the data. In this chapter, I explained in detail the data collection and analysis process, the results, and how the themes that emerged from the data represented answers to the research questions. The collection of data representing the lived experiences of GSD organizational leaders in a distributed environment and the related explanations of the lived experiences of GSD managers in organizations led to the generation of themes and patterns discussed in Chapter 5. The analysis of the textual data, derived from the verbatim transcriptions of the interviews, using NVivo data analysis software that helped to organize and code data, led to the identification of eight major themes.

Chapter 5 includes the discussion and interpretations of the findings. Addressed in the chapter are the limitations of the research and recommendations based on the results of the study. Study implications complete the chapter, leading to the final research conclusions.

Chapter 5: Discussion, Conclusions, and Recommendations

Introduction

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding, shared by managers in Canadian GSD organizations, of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction. The qualitative method was appropriate because the goal of the study was to reveal answers to research questions based on experiences and meanings from the participants' real-life in-depth perspectives. The case study strategy enabled an in-depth exploration of experiences through interviews, which led to a deeper understanding from good descriptions and richness of data. The participants, through their interviews, shared their experiences, provided insights into the common understanding of GSD managers, and expressed their beliefs about what they needed to effectively manage software product quality in a distributed setting. They discussed alternative ways being implemented to alleviate the challenge of low software product quality in a global software development organization. The participants proposed new concepts and ideas in GSD organizations on how to improve management practices such as commitment and focus on team effort, performance, process, training, empowering staff, adopting new technology, leadership, motivation, engagement, cultural cohesiveness, and communication.

Interpretation of Findings

This qualitative exploratory multiple case study led to the discovery of a common understanding of what GSD organizational leaders need to meet software product quality and customer satisfaction, from the perspectives of managers in Canadian GSD organizations. Eight themes emerged from the data across the three main areas of research inquiry. Results from the analysis of data derived from 30 interviews revealed that, to achieve quality and customer satisfaction, GSD leaders need to develop clear organizational purpose and working principles. Internal processes, such as on-the-job training and skills, should be developed for all workers. These findings echo observations by Agrawal (2019), who indicated the need of organizational leaders to model product development processes based on Deming's quality principles. The specific focus for GSD leaders included creating constancy of purposes to ensure company competitiveness and adopting new philosophy based on-time product delivery, reduced errors, and avoidance of defective products and services (Alauddin & Shu. 2020).

These findings align with Deming's (1986) principles of creating a constant purpose to improve product quality and adapting new a philosophy, such as creating a vision, preparing for change, and promoting equality in the organization. Results also revealed that GSD managers need to develop appropriate personnel management strategies by eliminating fault-finding and promoting a positive environment. O'Regan (2019) examined the fundamentals of software quality and observed the importance of a positive working environment on employee creativity. Interviewees in this study shared similar insights that GSD managers need to address unhealthy competition, promote autonomy, and facilitate continuous skills and knowledge development among their personnel. The focus on employee training should be ensuring that quality during product development is in-built and incorporated early during the project creation to avoid depending on mass inspection for quality assessment (S.Kim & Kim 2018; Rais, 2016).

Quality processes will also be achievable during product development if GSD leaders adapt lifecycle and development methods that meet their organizational vision and customer needs. Janicijevic, Krsmanovic, Zivkovic, and Lazarevic (2016) observed that there is no single standard lifecycle model; however, the focus should be on the effective breakdown of project activities into sequential phases depending on deliverables to meet customer specifications. Even so, most interviewees noted that they mostly use the Agile model, while others use the Waterfall model, Kanban, Scrum, and Activate, depending on the project concept and goals. Six-sigma methodology is used to eliminate defects and enhance quality efficiency. Hynninen, Kasurinen, and Taipale (2018) emphasized the need for every manager to facilitate frameworks for observing software maintenance, runtime metrics, and overall quality, with the Agile model considered suitable in ensuring continuous quality assessment based on customer feedback.

Despite the substantial positive impact associated with using appropriate quality management models, Ghareb and Gary (2019) noted that GSD process entails potential challenges that need to be considered. As applied to this study, interviewees shared that the commonly encountered problems include collaboration issues, coordinating multiple groups, time-zone differences, quality monitoring problem, and managing diverse workers. The literature also reflected that the process of software development is marked by different challenges, such as rapid technological advancement, increasing customer demands, and time limitations (Castleberry & Nolen, 2018; Krehbiel & Miller, 2018).

Aryanny and Iriani (2020) cautioned that failure to explore potential challenges contributes to poor TQM and underachievement of the desired customer expectations. Deming (1986) advised the need for product developers to find problems and constantly work to improve the systems of production and service delivery, with a specific focus on reducing waste and decreasing costs.

Identifying potential challenges creates an avenue for GSD learners to formulate solutions to ensure continuous improvement of product quality. Interview responses revealed the need to improve communication, develop automatic testing tools, educate employees on quality, encourage self-improvement, standardize processes, and break cultural barriers, as some of the solutions to potential challenges during software development. A systematic review by Arcos-Medina and Mauricio (2019) confirmed the need to improve product quality, as opposed to focusing on prices and costs, in line with Deming's principles. Argotti, Baron, and Esteban (2019) pointed out the importance of solution formulation based on modern approaches to education and employee training. Further, effective measures need to focus on improved communication and information sharing to facilitate the mitigation of potential product quality problems that might emerge during product development.

Predicting the outcome of software product quality needs to be at the base of problem identification and solution formulation. Interviewees shared that GSD leaders may predict the outcomes of software product quality using established standards, such as ISO/IEC 25000 series, or using SDLC and CMM/ISO standards. The use of these standards aligns with past literature findings where their application has been a noted key component of the checklist for the development, supply, and maintenance of computer software (Janicijevic et al., 2016).

The interviewed participants shared that using established standards helps predict software product quality, such as maintainability, reliability, portability, and continuous maintenance based on customer needs. Using ISO/IEC 25000 standards was a noted key to eliminating potential defects and enhancing software product quality. These observations also align with past literature, where SDLC and ISO/IEC 2500 have been central to achieving software quality based on continuous user and developer interaction (Lee et al., 2020; Malleswari, Rakesh, Subrahmanyam, & Vadlamudi, 2019). Faqihuddin, Wahyuddin, and Nathasia (2020) indicated that organizational leaders are well-positioned to optimize SDLC, where costs and risks are potentially reduced with increased product quality and customer satisfaction. Using these quality standards, therefore, may help GSD organizational leaders in Canada when auditing and predicting whether the developed software quality addressed potential challenges, created relevant solutions, and met customer expectations on software product quality. Application of these standards may be anchored in Deming's 14 principles to ensure quality processes factor in organizational goals, personnel working principles, and leadership strategies intended to align with customer expectations and software product quality.

Thematic findings pertaining to identify challenges, formulate solutions, and focus on product quality were consistent with the existing knowledge in literature. The interview participants spoke about the need for top management commit to quality priority over project constraint, market demand, schedule and price. Ghanbari et al.

(2018) asserted that the neglect of quality practices in software development by managers has serious social, organizational, and economic consequences. The absence of standardization or a consensus on software quality development procedure among organizations is a prevailing challenge in a geographically distributed setting (Furtado et al., 2019). The interviewees collectively emphasized the need for a comprehension of business challenges and adequate feedback from the customer. The manager must identify challenges, formulate solutions, and address product quality. The common challenges are team supervision, collaboration, risk, tasks allocation, communication, privacy, technical issues, language requirement, process management, and knowledge management that negatively impact software quality (Shanyour & Qusef, 2018). Management is about prediction, evaluating a framework, producing information, and making decisions (Deming, 2013).

Thematic findings pertaining to formulate life cycle and development techniques also aligned well with the previous research findings in the peer-reviewed literature. The participants stressed the importance of organizational structure and project management in software development in similar ways expressed in the literature review. They claimed that management must build in quality and software security at all phases of software developmental life cycles, including requirement, planning or defining, designing, coding, testing, deployment, and maintenance. Organizational structure is a requirement for adequate performance of SDLC. Managers ideally perform evaluation, provide leadership, manage change, and provide performance feedback for the wellness of the organization (Jabangwe et al., 2016; Niazi et al., 2016). Reliable organizational structure will ensure software quality and team collaboration (Massago et al., 2018).

The findings from this study aligned with the conceptual framework, which together led to answers to the research questions, reflected in the eight themes: (a) develop a clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate lifecycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product quality. The use of research protocol questions based on Deming's 14 points quality management principles and SDLC, ISO/IEC 25000 series unveiled the common understanding of what GSD managers need to manage software product quality properly in a distributed environment. Bergmane et al. (2017) stated that continuous process improvement enhances software development. These research findings are consistent with literature review about continuous process improvement and customer satisfaction. The conceptual framework based on Deming's quality management aligned with the research findings regarding strategies for managers on management behavioral practices and best standard practice in software quality evaluation and predictions. The Deming's quality management framework reflects quality practices well in organizations requiring top management commitment (Anderson et al., 1994; Tamimi et al.1995).

Limitations of the Study

The scope of the study included 30 participants from six Canadian organizations. Data were primarily from semistructured interviews due to the gained advantage of flexibility in the interview process. There were some barriers encountered with the option of engaging potential participants online and accessing their publicly available contacts through a social media professional platform. Some of the professional members refused to respond to the study’s email invitation and others were reluctant to accept research invitation offers because of privacy and security concerns. These barriers prolonged the time to recruit members as participants and delayed the research study plan. Social media as a means for recruitment could introduce some level of sampling bias, which was unlikely to impact the results of a study, because it involved purposeful sampling of technology-oriented leaders.

Another limitation was that most of the interview participants provided few documents to back up their explanations. I used existing knowledge in literature and additional online reports for comparisons with participants' views, to ensure validation and stronger research evidence. In addition, surrounding the individual participants’ responses to the GSD strategy questions were unique interpretations, knowledge, personal bias, concepts, ideas, and verbal acumen that could have influenced the outcome of the study. The answers to the research questions involved a common understanding derived from participants’ answers; thus, the results of the study stemmed from that common understanding derived from expressed experiences and not from unique interpretations.

The interviews occurred through electronic conferencing tools, including the use of Zoom, and the telephone. During the interviews, my setting was private and quiet, conducive to a confidential and professional research environment. I could not control each participant's environment during the interview sessions. During the interview sessions, two of the participants had their family members interrupt the interviews. The interviews stopped and restarted after about an hour respectively for each of the participants. The problem of time zones affected the scheduled interview sessions. Two participants missed their interview sessions because of misunderstandings about the different time zones. The interview occurred on another day. These events were potential limitations to the thoroughness and depth of the data collected. It is unlikely that the interruptions that occurred jeopardized the trustworthiness of the findings.

Recommendations

Recommendations for Action

Findings from this study included a common understandings of what GSD organizational leaders need to achieve software product quality and customer satisfaction in Canadian GSD organizations. The data from participants in this study represented insights on what knowledge, skills, and strategies are needed to solve some of the management problems of global software development in a distributed setting. The recommendations documented in the literature review were consistent with the ones discussed by the research participants. The first recommendation was that GSD organizational leaders should commit to good management behaviors and best practices, according to Deming’s 14 principles of quality management. Deming (1986) asserted that organizational managers should create constancy of purpose, adopt new philosophies, cease reliance on mass inspection, ensure continuous improvement, engage in employee training, demonstrate strong leadership, drive out fear, and break down departmental barriers. Additionally, they should eliminate slogans and numerical goals, remove metric systems, end the practice of awarding business base on price tag alone, apply statistical principles in management, encourage self-development, and involve all workers in the transformation. A manager must manage a system which is an interrelated functionality of associated components, departments, teams, and platforms with a defined purpose for the system (Deming, 2013). The use of automation, complex networks, specifications, and predictions are better understood by the application of statistical techniques (Deming, 2013).

The second recommendation pertained to how GSD managers could mitigate the management problems of low software product quality and achieving customer satisfaction. The recommended actions in the literature review that were consistent with participants' discussions include the following:

- Foster team collaboration, communication, and ensuring adequate control.
- Use Subject Matter Experts when necessary for software developmental processes.
- Adopt reliable and effective tools for software development.
- Schedule regular quality audits to identify gray areas.
- Engage stakeholders early for effective requirement gathering.
- Build in quality at every SDLC developmental phases and ensure compliance to standards.
- Explore automated testing technology.
- Engage in server upgrade and continuous recovery.

- Ensure effective development and release management.
- Involve third-party independent testers when necessary.
- Ensure adequate supervision at multisite destinations.
- Adopt statistical thinking techniques in software quality.
- Comply with regulation and support standardization.
- Establish organized management structure.
- Promote effective knowledge sharing or transfer.
- Use tested models to predict software quality.
- Create quality management awareness and continuous training in specific skill areas.
- Build in security from the beginning of software development

To promote the use of the above-identified strategies for effective. software development in organizations, I will provide a summary of the research findings to the participants. I will share and disseminate the study findings to academic and IT professional journals, institutions, and companies by visitation. I will make a presentation in public enterprise conferences, academic conferences, GSD workshops, training sessions and seminars, online platforms, websites, and electronic communication.

Recommendations for Further Research

This qualitative exploratory multiple case study led to the discovery of a common understanding of what GSD organizational leaders need to meet software product quality goals and enhance customer satisfaction, from managers' perspectives in Canadian GSD organizations. Based on the results of the study, managers can commit to management behavioral practices consistent with the thematic findings: (a) develop clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, and (d) promote autonomy and personal worker development consistent with the earlier existing literature. Future research could revolve around each of these practices individually (to reveal further in-depth findings through qualitative research) or collectively (to reveal potential quantitative findings from comparative or correlation research involving one or more quality and customer satisfaction measures).

Further understanding can originate from ongoing study of the effective management of global software development, which focus on facets of software product quality and customer satisfaction. In this study, an in-depth exploration of the lived experiences of GSD managers in Canadian organizations produced a rich description of data and findings that pertained to a broad concept of software quality and customer satisfaction. The participants in this study included both vendors and client organizations, without separating them, but a common understanding among them pertained to the management of software product quality. Ongoing research could help to expand knowledge about how this common understanding is implemented across organizational levels. Future research could focus on vendor and client organizations separately to examine how each manage software product quality and customer expectations.

The limitation of the study could be that data collected from the participants may not be an adequate representation of the population since there were no participants from some provinces in Canada. The future researcher may expand the study beyond its scope to compare, confirm, or disconfirm findings from other regions and geographical locations worthy of study. For example, a further understanding might result from the application of the qualitative case study methodology to a broader investigation of GSD in more Canadian organizations. Future quantitative research could involve Deming's quality management method in the comparison of organizational management practices with a quantitative method. A future study may provide more insights into how GSD organizations predict software product quality using models and standards.

Implications

Relationship to Conceptual Framework and Methodology

The conceptual framework revolved around Deming's 14 points quality management principles (Deming, 1986). The GSD organizational leaders should create the constancy of purpose, establish strong leadership, adopt a new philosophy, build in quality at every software developmental phase, engage in employee training and development, and continuously improve the process and product to achieve customer satisfaction. The findings from this study reflected Deming's management principles in managerial commitments and employee performance. The organizational leaders can apply Deming's quality management principles to increase employee performance, engage in continuous process improvement, and enhance customer satisfaction. Managers can use Deming's framework for self-assessment of the quality management system and evaluation of improvement implementation, system efficiency, identify priorities, resource allocation, market share, and competitive advantage. This empirical framework is advantageous because of the inclusion of system view perspectives by Deming quality management principles (1986). Deming's system strategy advocated profound knowledge, system thinking, variation evaluation, epistemology and psychology that is holistic in nature.

Findings from the analysis of the data collected in this study included top management commitment to quality as an imperative to the successful implementation of the quality management system in a GSD environment. Apart from top management commitment to the company's quality policies, objective, and plan, there should be a deliberate demonstration of management behavioral best practices by managers. All the participants expressed a common understanding that is consistent with the Deming's quality management principles. The understanding is that the work environment should be without fear or undue pressure on the employees and that managers should build trust, coaching the employees on the job, which may help improve the process and product. The participants' responses revealed that management used incentives, bonuses, organized team events, sponsored online courses and tuition fees reimbursement for self-development as motivational strategies for employee development and performance. Managers can use Deming's quality management principles to improve employee-manager engagement.

Findings Related to Existing Literature

The research findings included eight themes from this study which were consistent with the existing literature on how common understanding among GSD organizational leaders can help improve quality software products and achieve customer satisfaction: (a) develop a clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate lifecycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product. All the participants described how their organizational managers are involved with management practices. The common knowledge is embedded in management behavior practices and using the best industry standards to develop quality. The analysis of the participants' data that culminated in answers to the research questions led to findings consistent with Deming's management principles that emphasized leadership through the performance of system reviews, research, education, and competitive advantage. These management principles become management behaviors that may help GSD leaders mitigate the challenges of software quality management.

The literature review in this study aligned with the data from participants and resultant answers to the research questions about the different approaches on how to improve the global software development processes. Emphasis was on gathering customer feedback, employee training, knowledge transfer, collaboration, communication, integration, control, project management, organizational structure, software security management, change management, task allocation, process improvement, a combination of software development methodology, and the use of models to predict software quality. Most of the participants discussed the aspects of global software development challenges pertaining to culture, locations, temporary distance, time-zones, humans, standardization, trust, replacing legacy systems, government regulation, risk, quality, stakeholder engagement, technical issues, and politics. These encountered GSD challenges discussed by participants were also consistent with the documented findings in the literature review for this study.

Applications to Professional Practice

The GSD managers' abilities to identify and implement the management behavioral practices in a global software development distributed setting will help mitigate the challenges of software product quality management and meet customer expectations. The use of best industry standards in software development will enable continuous process improvement, high employee performance, and increased software product quality. The GSD organizational leaders can make use of the findings from this study to improve employee performance and sustainability, prevent software defects, save cost, make informed decisions, and increase productivity. According to Deming (1986), only top management can make a decision that can assure quality with workers expected to follow defined instructions on how to do the job.

Implications for Social Change

The social change relationship is a transformation of human cultural aspects and social institutions through positive or negative impacts on society. The social change implications affect citizens, communities, societies, institutions, organizations, countries, and the entire world. The GSD organizational leaders can use the findings from this study to bring about a positive impact on social change in their organizations and beyond. The research findings, encompassing the eight themes revealed in this study, contribute to the closure of the gap in the literature due to scare available research studies on software quality product quality in GSD. The managers can use the strategies that emerged from this research to improve management behaviors and implement best industry standards among the geographically distributed teams. The sustained management best practices through common understanding will improve employee performance and positively impact their families, communities, and the respective countries. The findings from this study can be useful to proffer improvement strategies for the reliability of software products, achievement of high software product quality, prevention of software project failures, improvement of economic stability, and enhancement of software public safety.

Conclusion

The purpose of this qualitative exploratory multiple case study was to reveal the common understanding of what GSD organizational leaders need to meet software product quality and customer satisfaction from managers in Canadian GSD organizations. The data were from participants who responded to open-ended questions in semistructured interviews and from the review of related documents. The GSD organizational managers responded to interview questions, with journaling and note-taking adding reflexivity to the data collection process. The data analysis indicated eight themes: (a) develop clear purpose and work principles, (b) improve processes and employee skills, (c) develop adequate personnel management strategies, (d) promote autonomy and personal worker development, (e) formulate life cycle and development techniques, (f) identify challenges, (g) formulate solutions, and (h) focus on product quality. The conceptual framework for the study encompassed the 14 points of quality management principles, developed by Deming (1986). The research findings included the common understanding among GSD organizational leaders that may help with managing software product quality and achieving customer satisfaction. Among these findings are that the top management should commit to quality, continuously adopt management behavioral practices, and use best industry standards to evaluate and predict software product quality.

References

Abu Bakar, A., & Ishak, N. M. (2012). Qualitative data management and analysis using NVivo: An approach used to examine leadership qualities among student leaders. *Education Research Journal, 2*(3), 94-103. Retrieved from http://www.resjournals.com/ERJ

Acharya, A., & Gupta, M. (2016). An application of brand personality to green consumers: A thematic analysis. *The Qualitative Report*, *21*, 1531-1545. Retrieved from http://nsuworks.nova.edu/tqr/

Agburu, J., Anza, N., & Iyortsuun, A.S. (2017). Effect of outsourcing strategies on the performance of small and medium scale enterprises (SMEs). *Journal of Global Entrepreneur Research, 7,* 26- 27. doi:10.1186/s40497-017-0084-0

Agrawal, N. M. (2019). Modeling Deming's quality principles to improve performance using interpretive structural modeling and MICMAC analysis. *International Journal of Quality & Reliability Management, 26,* 1159-1180. doi:10.1108/IJQRM-07-2018-0204

Ahmed, Z., Mansor, Z., & Ahmad, K. (2017). An analysis of knowledge management challenges in agile global software development. *Journal of Telecommunication, Electronic and Computer Engineering*, *9*(4), 63-66. Retrieved from https://journal.utem.edu.my/index.php

Akbar, M., Shafig, M., Kamal, T., & Hamza, M. (2019). Towards the successful requirements change management in the domain of offshore software development outsourcing: Preliminary results. *International Journal of*

Computing and Digital Systems, 8(3)*,* 205-215. doi:10.12785/ijcds/080301

Akbar, M. A., Sang, J., Khan, A. A., Amin, E. F., Nasrullahi, S., Shafiq, M., … Xiang, H. (2018). Improving the quality of software development process by introducing a new methodology—AZ model. *IEEE Access, 8,* 4811- 4823. doi:10.1109/ACCESS.2017.278798

Akbar, R. (2019). Tailoring agile-based software development processes. *IEEE Access, 7,* 139852-139869. doi:10.1109/access.2019.2944122

Akbar, R., Khan, A. R., & Adnan, K. (2020). Software development process evolution in Malaysian companies. *Advances in Intelligence System and Computing,* 139-150. doi:10.1007/978-981-32-9949-8_10

Alamri, S. A., & Aziz, A. A. (2016). The effectiveness of implementing total quality management in software development*. International Advanced Research Journal in Science, Engineering and Technology, 3(6),* 206-211. doi:10.17148/IARJSET.2016.3640

Alebebisat, F. M., Alhalhouli, Z. T., Alrawashdeh, T. I., & Alshabatat, T. E. (2018). Review of literature on software quality. *World of Computer Science and Information Technology Journal, 8*(5), 32-42. Retrieved from http://wcsit.org

Alfaro, I., & Chandrasekaran, R. (2015). Software quality and development speed in global software development teams. *Springer Link, 57*(2), 91–102. doi:10.1007/s12599-015-0372-6

Alhassan, A., Alzahrani, W., & AbdulAziz, A. (2017). Total quality management for software development. *International Journal of Computer Applications*, *158*(5), 38-44. doi:10.5120/ijca2017912850

Ali, N., & Lai, R. (2017). A method of software requirements specification and validation for global software development. *Requirements Engineering*, *22*, 191–214, doi:10.1007/s00766-015-0240-4

Ali, A., Ullah, N., Abrar, M. F., Majeed, M. F., Umar, M. A., & Huang, J. (2019). Barriers to software outsourcing partnership formation: An exploratory analysis. *IEEE Access, 7,* 164556–164594. doi:10.1109/ACCESS.2019.2949919

Al-Obthani, F. S., & Ameen, A. A. (2018). Towards customized smart government quality model. *International Journal of Software Engineering & Applications, 9*(2)*,* 41-50. doi:10.5121/ijsea.2018.9204

Alotaibi, A. F., & Qureshi, M. (2014). Scrum and temporal distance-based global software development. *International Journal of Computer Network and Information Security, 6,* 48-53. doi:10.5815/ijcnis.2014.06.07

Alrawashdeh, T. A., Muhairat, M., & Althunibat, A. (2013). Evaluating the quality of software in ERP systems using the ISO 9126 model. *International Journal of Ambient Systems and Applications, 1*(1), 1-9. Retrieved from https://www.zuj.edu

Alsahli, A., Khan, H., & Alyahya, S. (2017). Agile development overcomes GSD challenges: A systematic literature review. *International Journal of Computer Science and Software Engineering, 6*(1), 7-18. Retrieved from www.IJCSSE.org

Alsanad, A. A., Chikh, A., & Mirza, A. (2019). A domain ontology for software requirements change management in global software development environment. *IEEE Access, 7,* 49352- 49362. doi:10.1109/ACCESS.2019.2909839

Alauddin, N. & Yamada, S. (2020). Overview of Deming criteria for total quality management conceptual framework design in education services, *Journal of Engineering and Science Research, 5*(3), 12-20. doi:10.26666/rmp.jesr.2019.5.3.

Alvis, E., Goncalves, C. A., & Bax, M. (2017). Agile methods from the viewpoint of information. *Informação & Informação,* 178-210. doi:10.5433/1981-8920.2017v22n3p178

Alzaidi, A. S., & Qureshi, M. (2014). Scrum practices and global software development. *Information Engineering and Electronic Business, 5,* 22-28. doi:10.5815/ijieeb.2014.05.04

Alzoubi, Y., Gill, A., & Al-Ani, A.(2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information and Management, 53*(1), 22-37. doi:10.1016/j.im.2015.08.003

Ammad, G., Janjua, U., Madni, M., Cheema, M., & Shahid, A. R. (2017). An empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry. *IEEE Access,* 1-28. doi:10.1109/ACCESS.2019.2953008,

Anderson, J. C., Rungtusanatham, M., & Schroeder, R. G. (1994). A theory of quality management underlying the Deming management method. *Academy of Management Review, 19*, 472-509. doi:10.2307/258936

Anderson, J., Rungtusanathan, M., Schroeder, R., & Devaraj, S. (1995). A path analytic model of a theory of quality management underlying the Deming management method: Preliminary empirical findings. *Decision Sciences*, *26*, 673-658. doi:10.1111/j.1540-5915.1995.tb01444.x

Anwer, S., Wen, L., Wang, Z., & Mahmood, S. (2019). Comparative analysis of requirement change management challenges between in-house and global software development: Findings of literature and industry survey. *IEEE Access, 4,* 1-39. doi:10.1109/ACCESS.2017

Aranda, G. (2010). Special issue on “global software development and its challenges”. *Open Access, 4*(1). doi:10.2174/1874107X010040100001

Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: A systematic literature review. *International Journal of System Assurance Engineering and Management*, *10*, 867-897. doi:10.1007/s13198-019-00840-7.

Argotti, Y., Baron, C., & Esteban, P. (2019). Quality quantification in systems engineering from the qualimetry eye. *2019 IEEE International Systems Conference,* 1-8. doi:10.1109/syscon.2019.8836756

Arumugam, A., & Kaliamourthy, B. (2016). Global software development: A design to measure risk of global practitioners. *International Conference on Computing, Analytics and Security Trends*, 1-4. doi:10.18293/SEKE2016-077

Aryanny, E., & Iriani.J. (2020). Analysis of quality management by implementing total quality management based on Deming prize. *Journal of Physics: Conference Series*, *1569*. doi:10.1088/1742-6596/1569/3/032015.

Asiri, S., & Qureshi, M. (2014). Proposed GSD model to solve coordination and communication problems. *International Journal of Modern Education and Computer Science, 11,* 25-30. *doi:*10.5815/ijmecs.2014.11.03

Aslam, W., Ijazi, F., Lali, M., & Mehmood, W. (2017). Risk aware and quality enriched effort estimation for mobile applications in distributed agile software development. *Journal of Information Science and Engineering, 33,* 1481-1500. doi:10.6688/JISE.2017.33.6.6

Awais, M. A. (2016). Requirements prioritization: Challenges and techniques for quality software development. *Advances in Computer Science: An International Journal, 5*(2). Retrieved from http/:www.ACSIJ.org

Babar, M., & Lescher, C. (2014). Global software engineering: Identifying challenges is important and providing solutions is even better. *Information and software Technology, 56*(1), 1-5. doi:10.1016/j.infsof.2013.10.002

Baquero, L., Gil, C., & Hernánde, M. (2018). The usability and accessibility as quality factors of a secure web product. *International Journal of Applied Engineering Research, 13*, 16288-16294. Retrieved from http://www.ripublication.com

Barney, S., Muhankumar, V., Chatzipetrou, P., Aurum, A., Wolin, C., & Angelis, S. (2013). Software quality across borders: Three case study on company internal alignment. *Information and Software Technology, 56,* 20-38. doi:10.1016/j.infsof.2013.06.004

Barouch, G., & Ponsignon, F. (2016). The epistemological basis for quality management. *Routledge, 27*, 944-962. doi:10.1080/14783363.2016.1188659

Bawa, P., & Watson, S. L. (2017). A phenomenological study of graduate Chinese students' English writing challenges. *Qualitative Report*, *22*, 779-796. Retrieved from http://nsuworks.nova.edu/tqr/

Bell, E. E. (2017). A narrative inquiry: A black male looking to teach. *Qualitative Report, 22*, 1137-1150. Retrieved from http://nsuworks.nova.edu/tqr/

Benes, S. S., Mazerolle, S. M., & Bowman, T. G. (2014). The impact of clinical experiences from athletic training student and preceptor perspectives. *Athletic Training Education Journal*, *9*, 156-165. doi:10.4085/0904156

Bentley, P., Gulbrandsen, M., & Kyvik, S. (2015). The relationship between basic and applied research in universities. *High Education*, *70*, 689 -709. doi:10.1007/s10734-015-9861-2

Berger, R. (2015). Now I see it, now I don't: Researcher's position and reflexivity in qualitative research. *Qualitative Research, 15*, 219-234. doi:10.1177/1468794112468475

Bergmane, L., Grabis, J., & Zeiris, E. (2017). A case study: Software defect root causes. *Information Technology and Management Science, 20,* 54-57. doi:10.1515/itms-2017-0009

Bhatti, M. W., & Ahsan, A. (2016). Global software development: An exploratory study of challenges of globalization. *HRM Practices, and Process Improvement*, *10*, 649-682. doi:10.1007/s11846-015-0171-y

Bhatti, M. W., & Ahsan, A. (2017). Global monitoring and control: A process Improvement framework for globally distributed software development teams, *Journal of Global Information Technology Management, 20*(1), 43-63.

doi:10.1080/1097198X.2017.1280303

Birchall, S. J., Murphy, M., & Milne, M. (2016). Mixed methods research: A comprehensive approach for study into the New Zealand voluntary carbon market. *Qualitative Report*, *21*, 1351-1365. Retrieved from http://nsuworks.nova.edu/tqr

Birgun, S., & Altan, Z. (2019). A managerial perspective for the software development process: Achieving software product quality by the theory of constraints. *IGI Global,* 243-250. doi:10.4018/978-1-5225-7865-9.ch013

Boehm, B. W. (2017). Software cost estimation meets diversity. *IEEE/ACM 39th International Conference on Software Engineering Companion,* 1-15. doi:10.1109/ICSE-C.2017.159

Bourgroun, Z., Saber, M., & Bouchentouf, T. (2017). The reconciliation of external and internal quality attributes of ISO using the OO quality concepts. *4th International Conference on Automation, Control Engineering and Computer Science, 19,* 52-55. Retrieved from http://ipco-co.com/PET_Journal

Brooks, F. P. (1995). *The mythical man-month*. Boston, MA: Addison-Wesley Longman

Brubacher, S. P., Poole, D. A., & Dickinson, J. J. (2015). The use of ground rules in investigative interviews with children: A synthesis and call for research. *Developmental Review*, *36*, 15-33. doi:10.1016/j.dr.2015.01.001

Braun, V., & Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, *11*, 589-597. *doi:*10.1080/2159676x.2019.1628806

Calabrese, J., Munoz, R., Pasini, A., Espnda, S., Boracchia, M., & Pesado, P. (2018). Assistant for the evaluation of software product quality characteristics proposed by ISO/IEC 25010 based on GQM-defined metrics. *Argentine Congress of Computer Science,* 164-175. doi:10.1007/978-3-319-75214-3_16

Carcary, M. (2011). Evidence analysis using CAQDAS: Insights from a qualitative researcher. *Electronic Journal of Business Research Methods*, *9*(1), 10-24. Retrieved from http://www.ejbrm.com

Carmel, E. (1999). *Global software teams: Collaborating across borders and time zones*. Englewood Cliffs, NJ: Prentice Hall.

Carrillo de Gea, J.M., Nicolas, J., Fernandez□Aleman, J.L., Toval, A., & Idris, A. (2016). Are the expected benefits of requirements reuse hampered by distance? an experiment. *Springer Plus, 5,* 20-27. doi:10.1186/s40064-016-

Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Current in Pharmacy Teaching and Learning*, *10*, 807-815. doi:10.1016/j.cptl.2018.03.019

Chadli, S. Y., Idris, A., Ros, J. N., Fernadez-Aleman, J. L., Carillo, J. M., & Toval, A. (2016). Software project management tools in global software development: A systematic mapping study. *Springer Plus, 5*, 1-3. doi:10.1186/s40064-016-3670-7

Chan, N., Walker, C., & Gleaves, A. (2015). An exploration of students lived experiences of using smartphones in diverse learning contexts using a hermeneutic phenomenological approach. *Computers & Education, 82,* 96-106. doi:10.1016/j.compedu.2014.11.001

Chan, Z. C. Y., Fung, Y., & Chien, W. (2013). Bracketing in phenomenology: Only undertaken in the data collection and analysis process? *The Qualitative Report, 18*, 1-9. Retrieved from http://www.nova.edu

Chever, D. A. (2018). A mixed method approach to investigate the antecedents of software quality and information systems success in Canadian software development firms. *Electronic Journal of Information Systems Evaluation, 21*(2), 109-130. Retrieved from http//: www.ejise.com

Chimagus, P., Selamat, A., Ibrahim, R., & Mahin, M. N. (2014). Systematic literature review of software requirement prioritization. *Information and Software Technology, 56,* 568–585. doi:10.1016/j.infsof.2014.02.001

Chong, C., & Yeo, K. (2015). An overview of grounded theory design in educational research. *Canadian Center of Science and Education, 11*, 1911-2025. doi:10.5539/ass.v11n12p258

Chou, D. C., & Chou, A. (2011). Innovation outsourcing: Risks and quality issues. *Computer Standards & Interfaces*, *33*, 350-356. doi:10.1016/j.csi.2010.10.001

Clandinin, D. J., & Connelly, E. M. (2000). *Narrative inquiry: Experience and story in qualitative research*. San Francisco, CA: Jossey-Bass.

Colomo-Palacios, R., Soto-Acosta, P., García-Peñalvo, F., & García-Crespo, A. (2012). A study on the impact of global software development in packaged software release planning. *Journal of Universal Computer Science, 18*, 2646-2668.

doi:10.3217/jucs-018-19-2646

Corbin, J., & Strauss, A. (2014). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (4th ed.). Thousand Oaks, CA: Sage.

Cornelissen, J. P. (2016). Preserving theoretical divergence in management research: Why the explanatory potential of qualitative research should be harnessed rather than suppressed. *Journal of Management Studies, 54*, 368-383. doi:10.1111/joms.12210

Creswell, J. W. (2014). *Qualitative inquiry and research design: Choosing among five approaches* (3rd ed.). Thousand Oaks, CA: Sage

Cronin, C. (2014). Using case study research as a rigorous form of inquiry. *Nurse Researcher*, *21*(5), 19-27. doi:10.7748/nr.21.5.19.e1240

Cummings, J. A., Zagrodney, J. M., & Day, T. E. (2015). Impact of open data policies on consent to participate in human subject research: Discrepancies between participant action and reported concerns. *PLOS ONE*, *10*, e0125208. doi:10.1371/journal.pone.0125208

Cusick, J. J., Prasad, A., & Tepfenlart, W. M. (2008).Chapter 6 global software development: Origins, practices, and directions. *Advances in Computers, 74,* 201-269. doi:10.1016/S0065-2458(08)00606-2

Dahlgaard-Park, S. M., & Dahlgaard, J. (2015). Total quality management and business excellence. *Routledge, 26*, 933-937. doi:10.1080/14783363.2015.1091578

Deming, W. E. (1986). *Out of the crisis*. Cambridge, MA: Center for Advanced Engineering Study, Massachusetts Institute of Technology.

Deming, W. E. (2013). A system must be managed. In J.N. Orsini (Eds.), *The essential Deming: Leadership principles from the father of quality* (1st. ed., p151). New York, NY: McGraw-Hill

Denhere, N., Horne, T., & Van der Poll, A. (2015). Managing globally distributed software development projects using virtual teams: A middle east case study. *ACM,* 1-10. doi:10.1145/2815782.2815786

Denny, N., Mani, S., Nadella, S., Swaminatha, M., & Samadal, J. (2008). Hybrid offshoring: Composite personae and evolving collaboration technologies. *Information Resource Management Journal, 21*(1), 89-104. Retrieved from http://www.igi-global.com

Denzin, N. K., & Lincoln, Y. S. (2011). *The sage handbook of qualitative research* (4th ed.). Thousand Oaks, CA: Sage.

Dikko, M. (2016). Establishing construct validity and reliability: Pilot testing of a qualitative interview for research in Takaful (Islamic insurance). *Qualitative Report*, *21*, 521-528. Retrieved from http://nsuworks.nova.edu/tqr/

Djouab, R., & Bari, M. (2016). An ISO 9126 based quality model for the e-learning systems. *International Journal of Information and Education Technology, 6*, 370-375. doi:10.7763/IJIET.2016.V6.716

Douglas, T. J., & Fredenall, L. D. (2004). Evaluating the Deming management model of total quality in services. *Decision Sciences*, *35*, 393-422. doi:10.1111/j.0011-7315.2004.02569.x

Dudin, M, Smirnova, O., Vysotskaya, N., Frolova, E., & Vilkova, N. (2017). The

Deming cycle (PDCA) concept as a tool for the transition to the innovative path of the continuous quality improvement in production processes of the agro-industrial sector. *European Research Studies Journal, 20(*2), 283-293.

Dunn, P. J., Margaritis, V., & Anderson, C. L. (2017). Understanding health literacy skills in patients with cardiovascular disease and diabetes. *Qualitative Report*, *22*, 33-46. Retrieved from http://nsuworks.nova.edu/tqr/

Ebert, C. (2018). 50 years of software engineering: Progress and perils. *IEEE Software, 35*(5), 94-101. doi:10.1109/MS.2018.3571228

Edhlund, B., & McDougall, A. (2016). *NVivo 11 essentials*. Retrieved from Lulu.com.

Fauzi, S., Suali, A., & Sobri, W. (2018). A state of the art: Software configuration management tools for global software development. *Journal of Physics Conference Series, 1049*(1), 1-9. doi:10.1088/1742-6596/1049/1/012006

Faqihuddin, A., Wahyuddin, I., & Nathasia, N. D. (2020). Mysql database processing information system using the system development life cycle (SDLC) method at quality guarantee agency working unit at national university. *Jurnal Mantik*, *4*, 399-406. Retrieved from http://iocscience.org/ejournal/index.php

Filho, M. S., Pinheiro, P. R., & Albuquerque, A. (2015). Task allocation approaches in distributed agile software development: A systematic review. *Advances in Intelligent Systems and Computing,* 1-10. doi:10.1007/978-3-319-18473-9_24

Filho, M. S., Pinheiro, P. R., Albuquerque, A., & Rodrigues, J. (2018). Task allocation in distributed software development: A systematic literature review. *Wiley Hindawi Complexity,* 1-13. doi:10.1155/2018/6071718

Fielding, J., Fielding, N., & Hughes, G. (2013). Opening up open-ended survey data using qualitative software. *Quality & Quantity*, *47*, 3261-3276. doi:10.1007/s11135-012-9716-1

Fisher, C., Elrod, C., & Mehta, R. (2011). A replication to validate and improve a measurement instrument for Deming's 14 Points. *International Journal of Quality & Reliability Management*, *28*, 328-358. doi:10.1108/02656711111109928

Foss, N. J., & Hallberg, N. L. (2014). How symmetrical assumptions advance strategic management research. *Strategic Management Journal, 35*, 903-913. doi:10.1002/smj.2130

Frank, U., & Mengiste, S. A. (2014). The challenges of creativity in software organizations. *Advances in Information and Communication Technology, 429*, 16-34. *doi:*10.1007/978-3-662-43459-8_2

Furtado, V. R., Vignando, H., Franca, V., & Oliveira, E. (2019). Comparing approaches for evaluation of software engineering experiments: An empirical study on software product line experiments. *Journal of Computer Science, 15*, 1396-1429. *doi:*10.3844/jcssp.2019.1396.1429

Galin, D. (2018). Project progress control. *Wiley-IEEE Press,* 617-634. doi:10.1002/9781119134527.app3

Ghanbari, H., Vartiainen, T., & Siponen, M. (2018). Omission of quality software development practices: A systematic literature review. *ACM Computing Surveys, 51*(2), 1-27. doi:10.1145/3177746

Ghareb, M., & Gary, A. (2019). An empirical evaluation of metrics on aspect-oriented programs. *UHD Journal of Science and Technology*, *3*(2), 74-86. doi:10.21928/uhdjst.v3n2y2019.

Gheorghe, M. (2015). Software quality validation for web applications developed using geographically distributed human resources. *Informatica Economica, 16*(2), 101-111. doi:10.12948/issn14531305/19.2.2015.10

Gillham, B. (2010). *Case study research methods*. London, Oxford: Bloomsbury

Goetz, J.P., & LeCompte, M.D. (1984). *Ethnography and qualitative design in educational research*. San Diego, CA: Harcourt Brace Jovanovich.

Goltz, U., Reussner, R.H., Goedicke, M., Hasselbring, W., Martin, L., & Vogel-Heuser, B. (2015). Design for future: Managed software evolution, *Computer Science Research Developments, 30,* 321–331. doi:10.1007/s00450-014-0273-9

Guba, E. G., & Lincoln, Y. S. (1989). *Fourth generation evaluation*. Newbury Park, CA: Sage.

Habib, M. M., Pathik, B. B., & Maryam, H. (2014). *Research methodology contemporary practices: Guidelines for academic researchers*. Newcastle, UK: Cambridge Scholars.

Haig-Smith, T., & Tanner, M. (2016). Cloud computing as an enabler of agile global software development. *Issues in Informing Science and Information Technology, 13*, 121-144. Retrieved from http://www.informingscience.org

Hammerberg, K., Kirkman, M., & Lacey, S. (2016). Qualitative research methods: When to use them and how to judge them. *Advanced Access Publication, 31*(3), 498–501, doi:10.1093/humrep/dev334

Hanley, N., Fileborn, B., Larcombe, W., Henry, N., & Powell, A. (2015). Improving the law reform process: Opportunities for empirical qualitative research? *Australian & New Zealand Journal of Criminology*, 1-18. doi:10.1177/0004865815604195

Haq, S., Raza, M., Zia, A., & Khan, M.N.A. (2011). Issues in global software development: Acritical review. *Journal of Software Engineering and Applications, 4,* 590-595. doi:10.4236/jsea.2011.410069

Harshmi, S., Clerc, V., Razavian, M., Manteli, C., Tamburri, D.A., Lago, P., … Richardson, I. (2011).Using the cloud to facilitate global software development challenges. *International Conference on Global Software Engineering Workshops, IEEE Computer Society,* 70-77. doi:10.1109/ICGSE-W.2011.19

Harvey, L. (2015). Beyond member-checking: A dialogic approach to the research interview. *International Journal of Research & Method in Education*, *38*(1), 23-38. doi:10.1080/1743727x.2014.914487

Hidayati, A., Purwandari, B., Burdiardjo, E. K., & Solichah, I. (2018). Global software development and capability maturity model integration: A systematic literature review. *Third International Conference on Informatics and Computing,* 1-7. doi:10.1109/IAC.2018.8780489

Horta, V., Stroele, V., Schettimo, V., Oliveira, J., David, J. M., & Araujo, A. P. (2019). Collaboration analysis in global software development. *IEEE 23rd International Conference on Computer Supported Cooperative Work in Design,* 28-32. doi:10.1109/CSCWD.2019.8791857

Hossain, M. (2018). Challenge of software quality assurance and testing. *International Journal of Software Engineering and Computer Systems, 4*(1), 133-144. doi:10.15282/ijsecs.4.2018.11.0044

Houghton, C., Murphy, K., Meehan, B., Thomas, J., Brooker, D., & Casey, D. (2017). From screening to synthesis: Using NVivo to enhance transparency in qualitative evidence synthesis. *Journal of Clinical Nursing*, *26*, 873-881. doi:10.1111/jocn.13443

Huisman,M., & Rubin, J. (2018).Software quality tools and techniques presented in fase'17, *International Journal on Software Tools for Technology Transfer*, *20*, 611–61. doi:10.1007/s10009-018-0504-7

Hussain, S., Farid, S., & Mumtaz, I. (2019). Is customer satisfaction enough for software quality? *International Journal of Computer Science and Software Engineering, (8)*2, 40-47. Retrieved from www.IJCSSE.org

Hynninen, T., Kasurinen, J., & Taipale, O. (2018). Framework for observing the maintenance needs, runtime metrics and the overall quality-in-use. *Journal of Software Engineering and Applications*, *11*(04), 139-152. doi:10.4236/jsea.2018.114009

Iftikhar, A., Alam, M., Musa, S., & Suud, M.M. (2018). Trust development in virtual teams to implement global software development (GSD): A structured approach to overcome communication barriers, *IEEE Explore*, 1-5. doi:10.1109/ICETSS.2017.8324169

Indumini, U., & Vasanthapriyan, S. (2018). Knowledge management initiatives in agile software development: A literature review. *National Information Technology Conference,* 1-10. doi:10.1109/NITC.2018.8550066

Innovation, Science, and Economic Development of Canada. (2018). *Canadian ICT sector profile: Automotive, transportation and digital technologies branch*. Retrieved from https://www.ic.gc.ca

Ishak, N. M., & Baker, A.Y.A. (2012). Qualitative data management and analysis using NVivo: An approach used to examine leadership qualities among student leaders. *Education Research Journal, 2*(3), 94-103. Retrieved from http://www.resjournals.com/ERJ

ISO/IEC is 9126-1. (2001). *Software engineering - product quality – part 1: Quality model.* Geneva, Switzerland: International Organization for Standardization.

Iyer, V.G. (2018). Total quality management or continuous improvement system in education sector and its implementation framework towards sustainable international development. *Ideal Journal of Education and Policy Studies, 5*(1), 22-34. doi:10.2991/csece-18.2018.119

Jabangwe, R., Borstler, J., & Peterson, K. (2014). Handover of managerial responsibilities in global software development: A case study of source code evolution and quality. *Software Quality Journal, 23,* 539-566. doi:10.1007/s11219-014-9247-1

Jabangwe, R., Wohlin, C., Petersen, K., Smite, D., & Borstler, J. (2016). A method for investigating the quality of evolving object-oriented software using defects in global software development projects. *Journal of Software Evolution and Process, 28,* 22-64. doi:10.1002/smr.1788

Jaffar, R., Hussain, A., & Chiad, W. (2019). A new model for study of quality attributes to components-based development approach. *Periodicals of Engineering and Natural Sciences*, *7*, 1177-1185. Retrieved from http://pen.ius.edu.ba

Jain, R., & Suman, U. (2015). A systematic literature review on global software development life cycle. *ACM SIGSOFT Software Engineering Notes, 40*(2), 1-14, doi:10.1145/2735399.2735408

Janicijevic, I., Krsmanovic, M., Zivkovic, N., & Lazarevic, S. (2016). Software quality improvement: A model based on managing factors impacting software quality. *Software Quality Journal, 24*(2), 247-270. doi:10.1007/s11219-014-9257-z

Jayatilleke, S., Lai, R., & Reed, K. (2018). A method of requirements change analysis. *Requirements Engineering*, *23*, 493–508. doi:10.1007/s00766-017-0277-7

John, B., Kadadevaramath, R. S., & Immanuel, E. A (2016). Recent advances in software quality management: A review. *Merit Research Journal of Business and Management, 4*(3), 18-26. Retrieved from http://www.meritresearchjournals.org/bm/index.htm

Kalia, A., & Sood, S. (2014). Component based software reusability and framing standards for components to be reused. *International Journal of Computer Science and Information Technologies, 5*, 1459-1462. Retrieved from httpwww.ijcsit.com

Keshta, I. M. (2017). Software cost estimation approaches: A survey. *Journal of Software Engineering and Applications, 10,* 824-842. doi:10.4236/jsea.2017.1010046

Khalil, C., & Khalil, S. (2019). Exploring knowledge management in agile software development organizations, *International Entrepreneurship and Management Journal,* 1-16. doi:10.1007/s11365-019-00582-9

Khan, A., Khan, K., Amir, M., & Khan, M.N.A. (2014). A component-based framework for software reusability. *International Journal of Software Engineering and Its Applications, 8*(10), 13-24. doi:10.14257/ijseia.2014.8.10.02

Khan, A. A., Basri, S., Dominic, P., & Amin, F. (2013). Communication risks and best practices in global software development during requirements change management: A systematic literature review protocol. *Research Journal of Applied Sciences, Engineering and Technology, 5*(9). doi:10.19026/rjaset.6.3554

Khan, A. A., Keung, J., Naizi, M., Hussain, S., & Ahmad, A. (2017). Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective. *Information and Software Technology, 87,* 180-205. doi:10.1016/j.infsof.2017.03.006

Khan, A. A., Keung, J.W., Amin, F. E., & Al-Wadud, M. A. (2017). Spiimm: Toward a model for software process improvement implementation and management in global software development. *IEEE Access, 5,* 2169-3536.

doi:10.1109/ACCESS.2017.2728603

Khan, A., Keung, J., Hussain, S., & Bennin, K. (2015). Effects of geographical, socio-cultural and temporal distances on communication in global software development during requirements change management: A pilot study. *The Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering,* 159-168. doi:10.5220/0005331401590168

Khan, A. A., Shameem, M., Kumar, R. R., Hussain, S., & Yan, X. (2019). Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development. *Applied Software Computing, 83.* doi:10.1016/j.asoc.2019.105648

Khan, K., Khan, A., Aamir, M., Zulfikar, S., & Khan, M.N.A. (2013). Quality assurance assessment in global software development. *World Applied Sciences Journal, 24*, 1449-1454. doi:10.5829/idosi.wasj.2013.24.11.13286

Khan, M., & Khan, A. W. (2018). Virtual team management challenges mitigation model (VTMCMM). *International Conference on Computing, Mathematics and Engineering Technologies, 2*(2), 88-93. doi:10.1109/ICOMET.2018.8346328

Khan, M.U.A., & Zulkernine, M. (2009). *A survey on requirements and design methods for secure software development*. Technical Report No. 2009-562. Queen's University School of Computing website. Retrieved from https://research.cs.queensu.ca/TechReports/Reports/2009-562

Khan, R. A., & Khan, S. U. (2018). A preliminary structure of software security assurance model. *Proceedings of 13th International Conference on Global*

*Software Engineering,*132-135. doi:10.1145/3196369.3196385

Khan, S.U., & Azeem, M. I. (2014). Intercultural challenges in offshore software development outsourcing relationships: An exploratory study using a systematic literature review. *IET Software, 8*(4), 161-173 .doi:10.1049/iet-sen.2013.0012

Khan, S.U., & Niazi, M. (2012).Critical challenges in offshore software development outsourcing: An empirical study. *International Conference on Software Engineering*, 1-9. doi:10.2316/P.2012.780-019

Khraiwesh, M. (2013). Project planning measures in CMMI. *International Journal of Software Engineering & Applications, 4*(2), 103-121. doi:10.5121/ijsea.2013.4208

Kihn, L., & Ihantola, E. (2015). Approaches to validation and evaluation in qualitative studies of management accounting. *Qualitative Research in Accounting & Management, 12*(3), 230-255. doi:10.1108/QRAM-03-2013-0012

Kim, S., & Kim, W. (2018). Evaluation of software quality-in-use attributes based on analysis network process. *Cluster Computing*, *22*(1), 2101-2114. doi:10.1007/s10586-018-2309-6

Krehbiel, T. C., & Miller, D. P. (2018). Should agile be part of your quality management system? *Journal of International Technology and Information Management, 27*(1), 1-28. Retrieved from https://scholarworks.lib.csusb.edu/jitim

Kuhrmann, M., & Fernandez, D.M. (2015). Systematic software development: A state of the practice report from Germany. *International Conference on Global Software Engineering,* 1-20. doi:10.1109/ICGSE.2015.11

Kumar, D. (2017). Requirement paradigms to implement the software projects in agile development using analytical hierarchy process. *International Journal of Decision Support System Technology, 9*(3), 28-29. doi:10.4018/IJDSST.201707010

Lee, S.J., Koo, R., Varuttamaseni, A., Yue, M., Li, M., Cho, J., & Gookkang, H (2020). Optimization of software development life cycle quality for NPP safety software based on a risk-cost model, *Annal of Nuclear Energy*,*135,* doi:10.1016/j.anucene.2019.106961

Liao, L. (2019). Feasibility research on quality management technique with Deming's management principles to heavy industries in mainland China. *Advances in Economics, Business and Management Research, 80,* 630- 640. doi:10.2991/bems-19.2019.111

Lincoln, Y. S. & Guba, E.G. (1985). *Naturalistic inquiry.* Newbury Park, CA: Sage.

Lunnay, B., Borlagdan, J., McNaughton, D., & Ward, P. (2015). Ethical use of social media to facilitate qualitative research. *Qualitative Health Research, 25*(1), 99-109. doi:10.1177/1049732314549031

Madsen, A. K. (2013). Virtual acts of balance: Virtual technologies of knowledge management as co-produced by social intentions and technical limitations. *Electronic Journal of E-Government, 11*, 183-197. Retrieved from http://www.ejeg.com

Magableh, A. A., & AlSobe, M. R. (2018). Securing software development stages using Aspect-orientation concepts. *International Journal of Software Engineering Applications, 9*(6), 57-71, doi:10.5121/ijsea.2018.9605

Mahmood, S., Anwer, S., Ziazi, M., Alshayeb, M., & Richardson, I. (2017). Key factors that influence task allocation in global software development. *Information and Software Technology,* 1-15. doi:10.1016/j.infsof.2017.06.009

Malik, B. H., Faroom, S., Ali, M. N., Shehzad, N., Yousaf, S., & Saleem, H. (2018). Geographical distance and communication challenges in global software development: A review. *International Journal of Advanced Computer Science and Applications, 9*, 406-414. doi:10.14569/IJACSA.2018.090553

Malleswari, N., Rakesh, D., Subrahmanyam, K., & Vadlamudi, D. (2019). An efficient model for software quality analysis based on user and developer interaction. *International Journal of Innovative Technology and Exploring Engineering, 9*(2), 715-732. doi:5940/ijitee.B6822.129219

Marandi, A. K., & Khan, D. A. (2015). An impact of linear regression models for improving the software quality with estimated cost. *International Multi-Conference on Information Processing, 54,* 335-342. doi:10.1016/j.procs.2015.06.039

Marchewka, J. T. (2007). An application of the Deming management method for information technology projects. *Journal of International Technology and Information Management, 16*(2), 57-72. Retrieved from https://scholarworks.lib.csusb.edu/jitim

Marinho, M., Luna, A., & Beecham, S. (2018). Global software development: Practices for cultural differences, *Engineering and Computer Science*, 1-6. doi:10.1007/978-3-030-03673-2_22

Marshall, C., & Rossman, G. B. (2014). *Designing qualitative research* (6th ed.). Thousand Oaks, CA: Sage.

Massago, Y., Leal, G., Balancieri, R., & Galdamez, E. (2018). Towards classifying sociocultural aspect in global software development. *Journal of Information Systems and Technology Management, 15,* 1-5. doi:10.4301/S1807-1775201815007

Matthiesen, S., & Bjorn, P. (2015). Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. *Collaboration in a Globalized World,* 876-890. doi:10.1145/2675133.2675232

Maxwell, J.A. (1996). *Qualitative research design: An interactive approach.* Thousand Oaks, CA: Sage

McPhie, N. A. (2005). *COR: Managing the government's technical experts*. Retrieved from http://www.mspb.gov

Merriam, S. (1988). *Case study research in education: A qualitative approach*. San Francisco, CA: Jossey-Bass.

Merriam, S., & Tisdell, E. (2016). *Qualitative research: A guide to design and implementation* (4th ed.). San Francisco, CA: Jossey-Bass.

Miguel, A., Maurico, D., & Rodriguez, A. (2014). A review of software quality models for the evaluation of software products. *International Journal of Software Engineering & Applications, 5*(6), 31-53. doi:10.5121/ijsea.2014.5603

Miles, M., & Huberman, A. (1984). *Qualitative data analysis*. Newbury Park, CA: Sage.

Miles, M., & Huberman, A. (1994). *An expanded sourcebook: Qualitative data analysis* (2nd ed.). Thousand Oaks, CA: Sage.

Mishra, D., Mishra, A., Colomo-Palacios, R., & Casado-Lumbreras, C. (2013). Global software development and quality management: A systematic review. *OTM Confederated International Conferences on the Move to Meaningful Internet Systems*, *8186*, 302-311. doi:10.1007/978-3-642-41033-8_39

Moen, R. D., & Norman, C. (2016). Always applicable. *Quality History,* 43-53. Retrieved from http/:www.quality progress.com

Moon, K., Brewer, T., Januchowski-Hartley, S., Adams, V., & Blackman, D. (2016). A guideline to improve qualitative social science publishing in ecology and conservation journals, *Ecology and Society, 21*(3), 1-14. doi:10.5751/ ES-08663-210317

Morse, J. M. (2015). Critical analysis of strategies for determining rigor in qualitative inquiry. *Qualitative Health Research, 25,* 1212-1222. doi:10.1177/1049732315588501

Musa, K., & Alkhateeb, J. (2013). Quality model based on cots quality attributes. *International Journal of Software Engineering and Application, 4*(1), 1-8. doi:10.5121/ijsea.2013.4101

Nagappan, N., Murphy, B., & Basili, V. (2008). The influence of organizational structure on software quality: An empirical case study. *Proceedings of the 30th International Conference on Software Eng*ineering, 521-530. doi:10.1145/1368088.1368160

Nataraja, P., & Ramesh, B. (2016).A review on lean thinking for software process improvement. *International Journal of Advanced Research in Computer and Communication Engineering, 5*(11), 131-136. doi:10.17148/IJARCCE.2016.51128

Niazi, M., Mahmood, S., Alshayeb, M., & Hroub, A. (2014). Challenges of the existing tools used in global software development projects. *The Ninth International Conference on Software Engineering Advances,* 385-389. doi:10.1049/iet-sen.2014.0130

Niazi, M., Mahmood, S., Alshayeb, M., Riaz, M. R., Faisal, K., Cerpa, N., & Khan, S. U., & Richardson, I. (2016). Challenges of project management in global software development: A client-vendor analysis. *Information and Software Technology, 80,* 1-19. doi :10.1016/j.infsof.2016.08.002

Noble, H., & Smith, J. (2015). Issues of validity and reliability in qualitative research*,* *Evidence Based Nursing, 18*(2), 1-35. doi:10.1136/eb-2015-102054

Onwuegbuzie, A. J., Bustamante, R. M., & Nelson, J. A. (2010). Mixed research as a tool for developing quantitative instruments. *Journal of Mixed Methods Research, 4*(1)*,* 56-78. doi:10.1177/1558689809355805

O'Regan, G. (2019). Fundamentals of software quality. *Concise Guide to Software Testing*, 1-31. doi:10.1007/978-3-030-28494-7_1

Ouriques, R.A.D.B., Wnuk, K., Gorschek, T., Svensson, R.B. (2019). Knowledge management strategies and processes in agile software development: A systematic literature review. *International Journal of Software Engineering and*

Knowledge Engineering, 1-24. doi:10.1142/S0218194019500153

Padgett, J., Gossett, K., Mayer, R., Chien, W., & Turner, F. (2017). Improving patient safety through high reliability organizations. *Qualitative Report*, *22*, 410-425. Retrieved from http://nsuworks.nova.edu/tqr/

Patton, M. (2015). *Qualitative research and evaluation methods* (4th ed.). Thousand Oaks, CA: Sage.

Paschali, M., Ampatzogloua, A., Bibi, S., Chatzigeorgiouc, A., & Stamelos, I. (2017). Reusability of open source software across domains: A case study. *Journal of Systems and Software, 134,* 211-227. doi:10.1016/j.jss.2017.09.009

Polillo, R. (2011). Quality models for web 2.0 sites: A methodological approach and a proposal. *Computer Science Bibliography*, 251-261. doi:10.1007/978-3-642-27997-3

Prikladnicki, R., Damian, D., & Audy, J. (2008). Pattern of evolution in the practice of distributed software development: Quantitative results from a systematic review. *International Conference on Evaluation and Assessment in Software Engineering,* 1-10. doi:10.14236/ewic/EASE2008.11

Project Management Institute. (2018). *A guide to the project management body of knowledge* (6th ed.). Newton Square, PA: Author.

Rais, A. (2016). Interface based software testing. *Journal of Systems Integration*, *7*(4), 46-54. doi:10.20470/jsi.v7i4.277

Rajagopalan, S. (2014). Review of the myths on original software development model.

International Journal of Software Engineering & Applications, 5(6), 103-111. doi:10.5121/ijsea.2014.5607

Rashidi, H., & Hemanyati, M. (2018). Software quality models: A comprehensive review and analysis. *Journal of Electrical and Computer Engineering Innovations, 6*(1), 59-76. doi:10.22061/JECEI.2019.1076

Ramacharan, S., & Rao, K. (2016). Scheduling based cost estimation model: An effective empirical approach for GSD project. *Thirteenth International Conference on Wireless and Optical Communications Networks,* 1-11. doi:10.1109/WOCN.2016.7759881

Ravitch, S. M., & Riggan, M. (2017). *Reason and rigor: How conceptual frameworks guide research* (2nd ed.). Thousand Oaks, CA: Sage.

Razzak, M. A., Bhuiyan, T. & Ahmed, R. (2014). Knowledge management in distributed agile software development projects. *IFIP International Workshop on Artificial Intelligence for Knowledge Management,* 1-27. doi:10.1007/978-3-319-28868-0_7

Rehman, S., & Khan, S. U. (2014). Software quality attributes in global software development: Initial results. *Scientific International (Lahore)*, *26*, 1415-1419. Retrieved from http://www.sci-int.com

Ridder, H. (2017). The theory contribution of case study research designs. *Business Research,* 1-25. doi:10.1007/s40685-017-0045-z

Rodriguez, M., Oviedo, J.R., & Piattini, M. (2016). Evaluation of software functional suitability: A case study. *Software Quality Professionals, 18*(3), 18-29. Retrieved from https://asq.org/quality-resources/pub/software-quality-professional

Roulston, K., & Shelton, S. A. (2015). Reconceptualizing bias in teaching qualitative research methods. *Qualitative Inquiry*, *21*, 332-342. doi:10.1177/1077800414563803

Rustambekovicha, Y. N., Gulyamovb, S. M., Usmanovac, N. B., & Mirzaevd, D. A. (2017). Challenging the ways to determine the faults in software: Technique based on associative interconnections. *Procedia Computer Science, 120,* 641-648. doi:10.1016/j.procs.2017.11.290

Ryan, K. T. (2016). Software processes for a changing world. *Journal of Software Evolution and Process, 28,* 236-240. doi:10.1002/smr.1756

Saldana, J. (2016). *The coding manual for qualitative researchers* (3rd ed.). Thousand Oaks, CA: Sage.

Saleem, N., Mathrani, S., & Taskin, N. (2019). Understanding the different levels of challenges global software development. *14th ACM/IEEE International Conference on Global Software Engineering,* 76-77. doi:10.1109/ICGSE.2019.00027

Sarmento, V. P., Gysels, M., Higginson, I. J., & Gomes, B. (2017). Home palliative care works: But how? A meta-ethnography of the experiences of patients and family caregivers. *BMJ Supportive & Palliative Care,* 1-25. doi:10.1136/bmjspcare-2016-001141.

Schneider, S., Torkar, R.., & Gorschek, T. (2013).Solution in global software engineering: A systematic literature review. *International Journal of Information Management*, *33*(1), 119-132 doi.org/10.1016/j.ijinfomgt.2012.06.002

Shahzeydi, M., & Gandomani, T. J., & Sadeghi, R. (2018). Quality aspects of continuous delivery in practice. *International Journal of Advanced Computer Science and Applications, 9*, 2010-212. doi:10.14569/ijacsa.2018.090527

Shameem, B., Chandra, B., Kumar, R. R., & Kumar, C. (2018). A systematic literature review to identify human related challenges in globally distributed agile software development: Towards a hypothetical model for scaling agile methodologies. *International Conference on Computing Communication and Automation,* 1-7. doi:10.1109/CCAA.2018.877753

Shameem, M., Kumar, C., & Chandra, B. (2017). Challenges of management in the operation of virtual software development teams: A systematic literature review. *International Conference on Advanced Computing and Communication Systems,* 6-7. doi:10.1109/ICACCS.2017.8014695

Shanyour, B., & Qusef, A. (2018). Global software development and its impact on software quality. *IEEE Xplore,* 1- 6. doi:10.1109/ISIICT.2018.8613294

Sharma, S., Kaur, P., & Kaur, U. (2015). Communication understandability enhancement in GSD. International conference on futuristic trend in computational analysis and knowledge management (ABLAZE). *Springer Berlin Heidelberg,* 172-177. doi:10.1109/ABLAZE.2015.7154962

Shen, P., Ding, X., Ren, W., & Yang, C. (2018). Research on software quality assurance based on software quality standards and technology management. *IEEE Explore,* 27-29. doi:10.1109/SNPD.2018.8441142

Silverman, D. (2013). *Doing qualitative research: A practical handbook* (4th ed.). Thousand Oaks, CA: Sage.

Simon, M. K., & Goes, J. (2013). *Dissertation and scholarly research: Recipes for success.* Lexington, KY: Dissertation Success

Simons, H. (2009). *Case study research in practice*. Los Angeles, CA: Sage

Stake, R. E. (1995). *The art of case study research.* Thomas Oaks, CA: Sage

Stefani, A. (2018). On the design of effective e-commerce applications: An iso-based lifecycle model. *Journal of Computer and Communications*, *6*, 15-30. doi:10.4236/jcc.2018.65002.

Stewart, H., Gapp, R., & Harwood, I. (2017). Exploring the alchemy of qualitative management research: Seeking trustworthiness, credibility and rigor through crystallization. *Qualitative Report*, *22*(1), 1-19. Retrieved from http://nsuworks.nova.edu/tqr/

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Newbury Park, CA: Sage

Sowunmi, O., Misra, S., Fernandez-Sanz, L., Crawford, B., & Soto, F. (2016). An empirical evaluation of software quality assurance practices and challenges in a developing country: A comparison of Nigeria and Turkey. *Springer Plus, 5*, 1-13. doi:10.1186/s40064-016-3575-

Sugiyanto, S., Rochimah, S., & Sarworsri, S. (2016). The improvement of software quality model for academic websites based on multi-perspective approach, *Journal of Theoretical and Applied Information Technology, 86*, 464-471. Retrieved from https://www.jatit.org

Suman, U., & Jain, R. (2015). A systematic literature review on global software development life cycle. *ACM SIGSOFT Software Engineering, 40*(2), 1-14. doi:10.1145/2735399.2735408

Susan,J., & Rasulova,S. (2017). Qualitative research and the evaluation of development impact: Incorporating authenticity into the assessment of rigour. *Journal of Development Effectiveness, 9*(2), 263-276. doi :10.1080/19439342.2017.1306577

Talib, R., Saeed, Y. M., Awais, M., & Hanif, K. (2017). Interactive systems regarding global software development and offshoring. *Current Science, 112,* 2134-2138. doi:10.18520/cs/v112/i10/2129-2134

Tamimi, M., Alghamdi, F., & Yaseen, A. (2019). A systematic snapshot review of custom-made software enterprises from the development perspectives, *International Journal of Information Systems, 9*(1), 1-23. doi:10.24247/ijismrdjun20191

Tamimi, N., Gershon, M., & Currall, S.C. (1995) Assessing the psychometric properties of Deming's 14 principles, *Quality Management Journal*, *2*(3), 38- 52, doi:10.1080/10686967.1995.11918690

Tanner, M., & Dauane, M. (2017). The use of Kanban to alleviate collaboration and communication challenges of global software development. *Issues in Informing*

Science and Information Technology Education, 14, 177- 197. Retrieved from http://www.informingscience.org/Publications/3716

Tavares, B., Sanches da Silva, C., & Diniz de Souza, A. (2019). Project to improve risk management in agile projects. *International Journal of Software Engineering and Knowledge Engineering*, *29*, 381-399. doi:10.1142/S0218194019500165

Tovar, E. (2014). Project managers in global software development teams: A study of the effects on productivity and performance. *Software Quality Journal, 22,* 3-19. doi 10.1007/s11219-012-9191-x

Tiwari, A., & Chakraborty, P. (2015). Software component quality characteristics model for component-based software engineering. *IEEE Xplore,* 47-51. doi:10.1109/CICT.2015.40

Usman, A., & Khan, A.W. (2018). Software outsourcing quality challenges model systematic literature review (SLR) protocol. *University of Sindh Journal of Information and Communication Technology, 2*(4), 196-201. Retrieved from http://sujo2.usindh.edu.pk/index.php/USJICT/

Usmani, N., Hassan, R., & Mahmood, W. (2017). Impediments to requirement engineering in distributed team. *International Journal of Information Engineering and Electronic Business, 6,* 10-18. doi:10.5815/ijieeb.2017.06.02

Vallon, B., Estacio, B., Prikladnicki, R., & Grechenig, T. (2018). Systematic literature review on agile practice in global software development. *Information and Software Technology, 96,* 161-180. doi:10.1016/j.infsof.2017.12.004

Varpio, L., Ajjawi, R., Monrouxe, L., O'Brien, B., & Rees, C. (2017). Shedding the cobra effect: Problematising thematic emergence, triangulation, saturation and member checking. *Medical Education*, *51*(1), 40-50. doi:10.1111/medu.13124

Vignando, M., Aiello, M., Rinaldi, A., Cattaruzza, T., Mazzon, G., Manganotti, P., Eleopra, R., & Rumiati, R. (2019). Food knowledge depends upon the integrity of both sensory and functional properties: A VBM, TBSS and DTI tractography study. *Scientific Reports, 9*, 1-11. doi:10.1038/s41598-019-43919-8

Vrhovec, S., Kumer, A., Trkman, M., & Krisper, M. (2015). Outsourcing as an economic development tool in transition economies: Scattered global software development *Information Technology for Development, 21*, 445–459. doi:10.1080/02681102.2013.874316

Waheed, S., Hamid, B., Jhanjh, N. Z., Humayun, M., & Malik, N. A. (2019). Improving knowledge sharing in distributed software development. *(IJACSA) International Journal of Advanced Computer Science and Applications, 10*(6). doi:10.14569/IJACSA.2019.0100656

Wahyuningrum, T., & Akhari, A. (2017).Usability evaluation method based on iso/iec square: A systematic mapping study. *International Journal of Computer Science Issues, 14*(2), 17-23. *doi:*10.20943/01201702.1723

Wang, M. (2016). The impact of cultural values on Chinese students in American higher education. *Qualitative Report*, *21*, 611-628. Retrieved from http://nsuworks.nova.edu/tqr/

Wickramaarachchi, W., & Lai, R. (2016). Effort estimation in global software development: A systematic review. *Computer Science and Information Systems, 14*, 393–421. doi:10.2298/CSIS160229007W

Wisdom, J. P., Cavaleri, M. A., Onwuegbuzie, A. J., & Green, C. A. (2012). Methodological reporting in qualitative, quantitative, and mixed methods health services research articles. *Health Services Research, 47*, 721-745. doi:10.1111/j.1475-6773.2011.01344.x

Yaseen, M., Ali, S., & Ullah, N. (2016). An improved framework for requirement implementation in the context of global software development: A systematic literature review protocol. *International Journal of Database Theory and Application, 9*(6), 161-170. doi:10.14257/ijdta.2016.9.6.16

Yaseen, M., & Ali, Z. (2019). Success factors during requirements implementation in global software development: A systematic literature review*, International Journal of Computer Applications, 182*(48), 17-23. doi:10.5120/ijca2019918728

Yaseen, M., Baseer, S., & Sherin, S. (2015). Critical challenges for requirement implementation in context of global software development: A systematic literature review. *International Conference on Open Source Systems and Technologies,* 120-125. doi:10.1109/ICOSST.2015.7396413

Yi, J., Qi, D., Tan, S. H., & Roychoudhury, A. (2015). Software change contracts. *ACM Transactions on Software Engineering and Methodology, 24*(3), 1-43. doi:10.1145/2729973

Yilmaz, K. (2014). Comparison of quantitative and qualitative research traditions: Epistemological, theoretical, and methodological differences. *European Journal* *of Education*, *48*, 311-325. doi:10.1111/ejed.12014

Yilmaz, M. R., & Chatterjee, S. (1997). Deming and the quality of software development. *Business Horizon, 40*(6), 51-55. doi:10.1016/S0007-6813(97)90068-9

Yin, R. (2015). *Qualitative research from start to finish* (2nd ed.). New York, NY: Guilford Press.

Yin, R. (2017). *Case study research and applications: Design methods* (4th ed.). Thousand Oaks, CA: Sage.

Yue, Y., Ahmed, I., Wang, Y., & Redmiles, D. (2019). Collaboration in global software development: An investigation on research trends and evolution. *International Conference on Global Software Engineering,* 1-30. doi:10.1109/ICGSE.2019.00028

Zafar, A., Saif, S., Khan, M., Iqbi, J., Ahkunzada, A., Wdood, A., ----Alamri, A. (2018). Taxonomy of factors causing integration failure during global software development. *IEEE Access, 6,* 22228- 22239. doi:10.1109/ACCESS.2017.2782843

Zamawe, F. C. (2015). The implication of using NVivo software in qualitative data analysis: Evidence-based reflections. *Malawi Medical Journal*, *27*(1), 13-15. doi:10.4314/mmj.v27i1.4

Zhao, D., Zuo, M., & Deng, X. N. (2015). Examining the factors influencing cross project knowledge transfer: An empirical study of IT services firms in China. *International Journal of Project Management, 33*, 325-340. doi:10.1016/j.ijproman.2014.05.003

Appendix

Interview Protocol Questions

Self-modified Qualitative Protocol Questions Based on Deming's 14 points, SDLC and ISO/IEC 25000 Series standards.

Respondents answer the questions using quality management practices in their companies and their lived experience.

Part A: Modified validated Instrument open-ended qualitative questions based on Deming's 14 points (Tamimi et al., 1995)

Point 1: Creating constancy of purpose

- How do top management provide for long-term plan, new technology, research and development, and employee training/education?

Point2: Adopting New technology

- What ways are the top management committed to quality improvement, setting objective for quality, and continuous quality enhancement as the primary goal?

Point 3: Ceasing reliance on mass inspection

- How do the top management support the belief that quality must be "built into" the product and not "inspected into" it?
- How do your suppliers use statistical quality control technique?
- How is the top management ensuring using statistical control techniques are used to minimize reliance on mass inspection?

Point 4: Ending the practice of awarding business based on price tag alone

- What ways do top management relates with suppliers in terms of developing long-term relationship and reliance on a few dependable suppliers?
- How are suppliers selected based on quality and price?
- What aspect do the suppliers play in product/service development process?

Point 5: Improving constantly the system of production or service

- How do you analyze the customer's requirement and how do you get feedback in the process of developing software product/service?
- What are the ways top management assess its competitors in order to improve the product/service?

Point 6: Instituting Training

- How are supervisors and employees trained in quality management, specific work-related skills, and statistical improvement techniques?

Point 7: Institution leadership

- How does supervisor help, build trust and coach their employees on the job?

What are the ways the supervisor ensure that their actions are consistent with the aim of the organization?

Point 8: Driving out fear

- How are employees expressing new ideas related to improving work condition or seek assistance when unsure of their task?
- What are the ways employees report working conditions that interfere with quality, without fear of being punished or feeling that they have no job security?

Point 9: Breaking down barriers among departments

- How do you ensure that different departments have compatible goals, good communication, and teamwork in the product /service design process?

Point 10: Eliminating slogan and targets

- How do top management provide its workers with method/procedures to meet the goals?
- What ways do top management remove obstacles that can cause defects/errors and that is not responsibility of hourly workers?
- How is it evidence that top management do not use vague slogans (i.e. “Do It Right the First Time”) in communication with its employees?

Point 11: Eliminating numerical quotas

- How are work standard assessed in terms of quality and quantity and process capability studies?
- How are do you ensure numerical quotas are not given higher priority than that quality of workmanship?

Point 12: Removing barriers to pride in workmanship

- How do top management set realistic goals for its employees and ensuring that there is no employee pressure for short-term results?
- How do you ensure adequate documentation on how to do the job and that quality of work environment is good?
- What is the evidence that employees are not ranked by performance appraisals and what are the method used to rank employees?

Point 13: Instituting education and self-improvement

- What are the available programs to develop teamwork , ensure effective communication , employees' conflict resolution skills, and to broaden employees' skills for future organizational needs?

Point 14: Taking action to accomplish the transformation

- What are the efforts of top management to towards executing its quality improvement policies?
- How are the top management making quality improvement polices visible to all employees?
- How do top management relies on internal and external consultants to implement its quality improvement policies?

Part B. Research Protocol Interview Questions based on SDLC and ISO/IEC 25000 Series standard

- What is the software development life cycle/framework in GSD recommended by top management?
- Why and how effective is this framework in your organization?
- What are the key elements of your used SDLC for GSD (example; requirement analysis, planning/defining, designing, coding, testing, deployment and maintenance)?
- How are the elements of Deming's quality management method used in GSD software development process in your organization?
- What quality management method is used in your organization and why Is it a preferred method? How effective is this quality management method?

- What software development methods are you using? Are they combined? And how effective is it?
- What are the new things your organization is doing to ensure software product quality and customer satisfaction in the GSD environment?
- What are the GSD challenges you normally encounter? And how do you mitigate these challenges?
- What do you think is most needed to solve the problem of low software product quality in a distributed setting like GSD?
- How do you predict software product quality? What software model or standards do you use to predict quality and how effective are they?
- What are the common understand of software quality management among GSD organizational leaders/manager?
- How do you implement Quality Standard ISO/IEC 25000 series and its three main area such as; definitive model, predictive model and assessment model?
- How do you measure external (for customer) and internal (users- developers) software quality characteristics?
- How do you determine software product quality weight for customer required characteristics?
- How does your model cover all aspect of quality software development processes (such as static and dynamic requirements analysis, cost, budget and time, customer satisfaction, the role of participants, all developmental phases, product and process factors, set of quality metrics and source code)?

- What documentary or software quality evaluation reports would you like to share to back up your explanations?
- Do you have any questions? Or ideas you wish to share?