11-7-2025

U.S. Government Software Development Entity Implements a Hybrid of Waterfall and Agile to Address Failures

Dwight Jerome Peters Jr.
Walden University

Walden University

College of Management and Human Potential

This is to certify that the doctoral dissertation by

Dwight J. Peters, Jr.

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. Robert Haussmann, Committee Chairperson, Management Faculty
Dr. Christina Anastasia, Committee Member, Management Faculty

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

Walden University
2025

Abstract

U.S. Government Software Development Entity Implements a Hybrid of Waterfall and Agile to Address Failures

by

Dwight J. Peters, Jr.

MS, Walden University, 2023

MS, Troy State University, 2002

BS, United States Military Academy, 1993

Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Management

Walden University

November 2025

Abstract

Although organizations allocate substantial resources to software development process initiatives, a significant portion of these projects fail. The U.S. government software development leaders and managers need to understand the disadvantages of both waterfall and agile methodologies as leading indicators of project failure. Grounded in the diffusion of innovation theory, the purpose of this qualitative case study was to explore the consequences in government software development fostered by a hybrid approach of waterfall and agile methodologies pertaining to the inherent shortcomings of each methodology. The participants consisted of four current and four former team members from the U.S. government software development entity. Data were collected using face-to-face interviews and U.S. Department of Defense open-source references. Four themes emerged from the thematic analysis: (a) a hybrid approach is the best of both worlds, (b) agile is better than waterfall, (c) resistance to change, and (d) lack of knowledge of methodologies. U.S. government software development leaders and managers can use these identified strategies to mitigate the weaknesses of both the waterfall and agile methodologies, leveraging their strengths to foster improvements in the government software development industry. The implications for positive social change include the potential for leaders and managers of U.S. government software development entities to implement software development best practices, thereby improving software that benefits service members, government agencies, and allies.

U.S. Government Software Development Entity Implements a Hybrid of Waterfall and Agile to Address Failures

by

Dwight J. Peters, Jr.

MS, Walden University, 2023

MS, Troy State University, 2002

BS, United State Military Academy, 1993

Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Management

Walden University

November 2025

Dedication

First, I want to dedicate my dissertation to my God; through Him, all things are possible. Second, I want to dedicate my dissertation to my wonderful children, LaKeisha, Dwight (III), Royce, Sarah, Cayla, and Camryn. You are all far better children than I deserve. Your presence has sincerely made me a better man. Lastly, I want to dedicate my dissertation to my mother, as well as my father and grandfather, who have left this earth. Thank you so much for your unconditional love and guidance, which have been instrumental in making me the man that I am.

Acknowledgments

First, I want to thank and acknowledge my committee chair and other committee members for their knowledge, expertise, diligence, and support through this process. Additionally, I want to acknowledge the current leader of the studied government software development entity, who gave me his permission to conduct this study. Lastly, I want to acknowledge my contractor organization, which supported me financially, professionally, and emotionally through this process.

Table of Contents

List of Tables

List of Figure

Chapter 1: Introduction to the Study

Software development has made major positive impacts on our society. However, success in software development is difficult (Tam et al., 2020). In fact, researchers revealed that the software development industry considered only 39% of projects, worldwide to be "successfully concluded" (Tam et al., 2020, p. 166). Software developers created methodologies to increase the success rate of software development. Over the years, software developers have established, adapted, and changed these methodologies.

Initially, software developers used traditional methodologies for all information technology (IT) projects. Traditional software methodologies consist of (a) a waterfall approach, (b) an iterative and incremental approach, (c) a spiral approach, and (d) an evolutionary approach (Al-Saqqa et al., 2020). The most common traditional methodology is waterfall (Gaborov et al., 2021). However, despite its prevalence, many development leaders/managers who used the waterfall methodology for their projects faced failure due to limitations such as frequent changes in user requirements, lack of communication between stakeholders and developers, and organizational morale issues (Gaborov et al., 2021). To overcome these challenges and address these failures, software developers introduced a new methodology–agile.

In contrast to the process-oriented waterfall methodology, the agile methodology prioritizes the software itself and emphasizes adaptability. The agile methodology contrasts with waterfall methodology by focusing on collaboration with users, continuous testing, refactoring, and incremental development, rather than strict upfront planning and

sequential states (Gaborov et al., 2021). However, like the waterfall, the agile methodology also has identifiable weaknesses.

The agile methodology has difficulty with standardization (Baham & Hirschheim, 2021). When using the agile methodology, software developers focus on the final product rather than software design and documentation. This focus on the final product leads them to neglect project documentation because of the amount of time it takes and its lower priority (Reiff & Schlegel, 2022). Because the customer continually re-prioritizes their tasks, software development leaders/managers have difficulty accurately planning and budgeting their projects (Reiff & Schlegel, 2022). Leaders/managers of large-scale projects also have difficulty implementing the basic principles of the agile methodology, such as team autonomy (Reiff & Schlegel, 2022). To emphasize the advantages and mitigate the disadvantages of both methodologies, the hybrid project management methodology highlights the best of both methodologies and mitigates their weaknesses (Reiff & Schlegel, 2022).

Background of the Study

Tam et al. (2020) provided the "hook" for this dissertation. Despite its many successes, software development has an astonishing failure rate. The article reads that success in software development comes with "great difficulty" (Tam et al., 2020, p. 166). I articulated how a hybrid approach of the waterfall and agile methodologies can address this difficulty in this dissertation.

Digital Village provided the "anchor" that justifies the "hook" with a key data point (70% of software development projects fail. Yours doesn't need to be one of them,

n.d.). This article explained why 70% of software development projects failed. Additionally, this article found that the current agile methodology needed four adjustments. The four adjustments are as follows: (a) start by defining the challenge or problem to fix, (b) define the outcome using a quantifiable measure, (c) describe the six to twelve features or functions the solution must have, and (d) identify the first half a dozen users who are engaged at the outset and throughout.

Bianchi et al. (2021) grounded my dissertation in a gap in the literature. This article reads of the prevalent assumptions with waterfall and agile methodologies. This article studied the success rates of waterfall, agile, and hybrid software development projects. Additionally, this article found that, as it pertains to software development, "one size does not fit all. This (reality) justified the growing exploration of hybrid models." (Bianchi et al., 2021, p.11).

Gaborova et al. (2021) explained how combining aspects of waterfall and agile provided overall success. This article studied the inherent weaknesses of the two most prevalent software development methodologies – waterfall and agile. It also emphasized the importance of constant communication between developers and their customers. IT companies now use the agile methodology in combination with of other methodologies. Projects use this hybrid approach to eliminate the shortcomings of each methodology.

Otero et al. (2020) provided the strengths and weaknesses of agile software development methodologies. This article studied the usage of agile methodologies in educational environments. It found that agile methodology is successful. However, leaders/managers need to use it in adaptive environments. Once I decided on the topic, I

had to decide on the research approach.

My doctoral coursework covered three research approaches: (a) qualitative, (b) quantitative, and (c) mixed methods. The difference between these approaches is the information the researcher seeks (Taherdoost, 2022). I wanted to learn of the attitudes and experiences of the current and former team members, which lends itself to a qualitative approach. Therefore, I decided to use the qualitative approach.

Patton (2015) provided guidelines and proposed a method for my qualitative study. This book studied why and how to conduct qualitative research and the different inquiry frameworks of qualitative research. It found that qualitative research is a viable way to understand a phenomenon's "meaning making" process (Patton, 2015, p. 3). The "meaning making" process supports my decision to use qualitative research for this dissertation. Once I decided on the approach, I had to decide on the research method.

The desired information is social research. One of the most commonly used methodologies of social research is a case study (Priya, 2020). Belcher (2019) explained the appropriateness of using a case study for research that collects data from participants' experiences, beliefs, perceptions, and intentions. This dissertation studied teachers' limited understanding and perceptions of the value of situated learning and authentic assessment. Additionally, it found that teachers have an accurate, yet limited, understanding of situated learning and authentic assessment as well as the benefits of incorporating these approaches. Teachers thoroughly understand the three challenges of using situated learning and authentic assessment but need help understanding mitigation strategies.

Case studies provide other advantages. For example, case studies can provide insight into complex phenomena. In this dissertation, I wanted to gain in-depth insights into the phenomenon of the studied government software development entity that used a hybrid approach of the waterfall and agile methodologies.

Rogers (2003) provided the basic concepts of diffusion of innovations (DOI) theory. This book studied how new ideas spread via communication channels over time. It also explains that the diffusion process entails a few individuals who are the first to adopt an innovation and then spread the word among their acquaintances. This diffusion typically takes months or years.

Bibhunandini (2022) provided the theoretical framework for my dissertation. This article studied the literature on the DOI theory and analyzed its outcome in different sectors. Additionally, this article found that all sectors benefited from DOI. I wanted to elaborate on how DOI influenced how the studied government software development entity was able to convince its parent organization and its customer to use the hybrid approach of the waterfall and agile methodologies.

Dearing and Cox (2018) defined “diffusion” and provided background for the theoretical framework – DOI theory. This article studied the question: “What is Diffusion?” and clarified how DOI relates to processes of dissemination, implementation, and sustainability of innovation. Additionally, this article found that healthcare organizations can use DOI theory to operationalize and accelerate the rate of adoption of health innovations.

Grant and Osanloo (2014) showed why I used a theoretical framework for my

dissertation study. This article studied the definition and differences in theoretical and conceptual frameworks using the analogy of a house. This article found that the theoretical framework resembled a house blueprint. The conceptual framework resembled a house floor plan.

Guest et al. (2016) provided standards for reaching data saturation in studies. This article studied the degree of data saturation and variability throughout thematic analysis. Additionally, the article found that saturation occurred within the first 12 interviews, but essential elements for themes were present as early as six interviews.

The software developed by the studied government software development entity in this case study trained the weapons system operators on the weapons system's hardware, software, and integration into a "system of systems" architecture. This training software provides realistic institutional training for system operators/crewmembers, staff, and commanders. The software simulates system hardware interfaces and provides institutional training in operational functions. Additionally, the software that generates digitized scenarios on video display terminals. The studied government software development entity falls under the purview of an overarching capability maturity model integration (CMMI) certified government software development organization that provides collaborative and innovative life cycle systems engineering, software engineering, modeling and simulation, systems development, prototyping, and technical systems management products and solutions.

Before taking over this task, a local civilian software development company provided this training software for the weapons system. However, the weapons system's

leadership decided to give the studied government software development organization the task because the leadership was dissatisfied with the results from the local civilian software development company. The weapons system's leadership wanted more responsiveness and incremental input in the software's development. Additionally, the weapon system's leadership wanted more flexibility with requirements and prioritization. When the leader of the government software development entity became aware of the customer's dissatisfaction, he created a proposal for the studied government software development entity to take over, build, and sustain the software development. Then, he presented the proposal to the weapon system's leadership.

The government software development entity leader proposed leveraging the overarching CMMI-certified government software development organization's established software development practices aligned with the waterfall methodology. However, when the government software development organization leader gave his proposal, he briefed the system's leadership on incorporating values and principles aligned with the agile methodology to provide the weapon system's leadership the capabilities that it desired but did not receive from the local civilian software development company. Many organizations need help implementing a hybrid approach of the waterfall and agile methodologies (Prenner et al., 2021). However, the studied government software development entity executed this hybrid approach and provided increased customer satisfaction to the weapon system's leadership.

Problem Statement

Over the years, software methodologies have undergone many improvements to

keep up with new technological advancements and modern business requirements. The overarching goal of software methodology is to continually develop practical approaches to reach the desired software product as accurately and efficiently as possible (Alsaqqa et al., 2020). Software development leaders/managers and their teams must know each methodology's benefits because the correct choice of methodology dramatically contributes to their project's success (Gaborova et al., 2021). The choice of methodology should focus on the process that best fits the project. A popular process improvement method is CMMI.

CMMI is a mainstay in multiple industries. In the software development industry, CMMI is a software process improvement model that enhances productivity while reducing project time and cost (Ayyagari & Atoum, 2019). There are five levels of the CMMI model: (a) Maturity Level 1 (ML1): Initial, (b) Maturity Level 2 (ML2): Managed, (c) Maturity Level 3 (ML3): Defined, (d) Maturity Level 4 (ML4): Quantitatively Managed, and (e) Maturity Level 5 (ML5): Optimizing (Thomas et al., 2022). Each level corresponds to a different activity, which I will elaborate on later in this study

Many Departments of Defense (DoD) and U.S. Government contracts use CMMI for their software development projects (Thomas et al., 2022). DoD software developers created CMMI to decrease production costs and increase quality. In time, CMMI became a mainstream software engineering study and a benchmark for software development capability (Hou et al., 2021). The two most prevalent methodologies in software development are waterfall and agile (Andrei et al., 2019). Waterfall is the methodology

that most closely resembles CMMI.

The waterfall methodology is an established software development approach. Software developers named this methodology “waterfall” from the analogy to the “finish-to-start” relationship that exists between the project phases in this approach (Fagarasan et al., 2021). Each phase in the waterfall methodology has different skills for completion and a formal acceptance and approval process at the end of each phase (Fagarasan et al., 2021, p. 2). There are different variations of the waterfall approach; however, the most common version has five phases in its process: (a) requirements, (b) design, (c) implementation, (d) verification, and (e) maintenance (Fagarasan et al., 2021). This vertical process encourages developers to move on to the next phase only after the current one is complete. One of CMMI/waterfall’s most significant deficiencies is its inflexibility; software developers follow a sequential flow according to the agreed-upon plan. When requirement changes occur, the changes cause problems in terms of delivery time (Otero et al., 2020). Additionally, customers rarely know all their needs for the software at the beginning of the project, which can lead to changes during software development. To address these challenges, software industry members established agile software development.

In the agile methodology, the software development process occurs in small incremental iterations (Otero, 2020). Additionally, software developers use the agile methodology because it enables them to deal with the increasing complexity of software development projects and handle the inevitable changes in requirements throughout the software lifecycle (Aldave et al., 2019). By using the agile methodology, customers can

prioritize what developers will deliver in the next interaction or software release by assigning priority to the requirements based on the costs of the requirements and their importance to the overall project (Otero, 2020). In short, customers choose which requirements take priority, and the software development team works to deliver the most recently agreed-upon requirements in the next iteration (Otero, 2020). However, software developers using the agile methodology must be highly skilled because there needs to be more guidance on requirement gathering, analysis, and risk management. (Al-Saqqa, et al., 2020).

Waterfall and agile are the two most common software development methodologies. Each of these methodologies has inherent disadvantages. In this study, I addressed the specific research problem that many software development projects are unsuccessful because of the inherent disadvantages of these two methodologies. To address these disadvantages that lead to a lack of success, the research problem is how software developers can best address the inherent disadvantages of the waterfall and agile methodologies. To that end, I explored what consequences a hybrid approach of the waterfall and agile methodologies approach would foster in the government software development industry.

Purpose of the Study

The purpose of this qualitative case study was to explore the consequences in government software development fostered by a hybrid approach of waterfall and agile methodologies pertaining to the inherent shortcomings of each methodology. I conducted interviews with eight current and former team members to explore this approach. I

conducted eight interviews to reach theoretical saturation and understand common perceptions and experiences among relatively homogeneous individuals (see Guest et al., 2006). Additionally, I used document analysis for data triangulation.

Despite their numerous successes, software development projects have an astonishing failure rate (Gaborova et al., 2021). An analysis of 50,000 software projects conducted by The Standish Group found that in 2015, an average of 29% of projects were successful (meaning the developers delivered the product on time, on budget, and to a satisfactory standard); the remaining 71% were either failures or 'challenged' to meet expectations (70% of Software Development Projects Fail. Yours Doesn't Need to Be One of Them., n.d.). The two most prevalent software development methodologies – waterfall and agile – have inherent weaknesses detrimental to software development (Gaborova et al., 2021). Research supports the idea that neither the waterfall nor agile methodologies fit all software development projects, which justifies the growing exploration of hybrid models between waterfall and agile (Bianchi et al., 2021). Therefore, the social problem is that the inherent shortcomings of the two most prevalent software development methodologies – waterfall and agile- lead to many software development project failures (Gaborova et al., 2021).

Research Question

RQ: What consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology?

Theoretical Foundation

The theory that grounds this study is the DOI. DOI provides a framework for studying a social system's adoption of an innovation through the lens of change (Çakıroğlu et al., 2022). As it pertains to organizations, adopting an innovation requires a change in organizational culture (Call & Herber, 2022). Dearing and Cox (2018, p. 183) defined "diffusion" as "a social process that occurs among people in response to learning about an innovation." Diffusion of Innovation occurs in five stages: (a) knowledge, (b) persuasion, (c) decision, (d) implementation, and (e) confirmation (Jierasup & Leelasantitham, 2024). Knowledge is the upfront understanding of the innovation; persuasion is a favorable disposition; the decision is a commitment to the innovation; implementation is the execution of the innovation; confirmation is the positive outcomes that reinforce use (Grover et al., 2019). As it pertains to various modes of technology diffusion, researchers have concluded that factors like information on availability, nature of technology, strategic behavior, adopters' characteristics, cost/benefits from the technology, growth of complementary technology, improvement in technology, the role of diffusing agencies, and government policies influence the diffusion process (Bibhunandini, 2022).

Nature of the Study

The specific research design included a rich story about the program to address the research questions in this qualitative study. This case study explored how a government software development program under a CMMI-certified organization implemented a hybrid approach of the waterfall and agile methodologies. To that end, I

conducted a descriptive case study that used eight face-to-face interviews as the primary means of data collection. I performed eight face-to-face interviews. According to Guest et al. (2006), data saturation has become the gold standard by which researchers determine purposive sample sizes (Guest et al., 2006). Later in the article, Guest et al. concluded that six to eight interviews are sufficient to reach data saturation.

Additionally, I used document analysis in this study. I analyzed documents created while a team member of the studied government software development entity. These documents "capture(d) context, the unfolding of events over time, and critical interactions, (including talks) with those involved in the activities observed" (Patton, 2015, p. 27). The combination of interviews and document analysis illuminated the inquiry question and provided data triangulation (Patton, 2015).

A qualitative case study was appropriate for this research because it collects rich, textured data from the participants' experiences, beliefs, perceptions, motivations, and intentions (Belcher, 2019). A case study allows the researcher to study the phenomena without affecting the settings and measuring instruments, which encourages detailed insights into the situation (Priya, 2020). An explanatory case study that uses causal reasoning creates a coherent view of how the program under a CMMI-certified organization combined waterfall and agile methodologies to develop the software for training a weapons system.

I conducted 30-60-minute, face-to-face interviews with eight current or former team members of the studied government organization. I chose the interviewees through purposeful sampling. I selected the participants because their respective positions in the

team gave them unique perspectives on the program's overall mission success. Specifically, I chose people from different levels and activities within the team.

The current leader of the studied government software development entity gave me permission to interview current and former team members. After receiving permission from the current leader of the studied software development entity, the Army Human Resource Protection Office (AHRPO), and the Walden University Institutional Review Board (IRB), I coordinated for a current team member to email former and current team members to solicit interest in giving interviews. After receiving responses from eight possible participants, I sent the possible participants a consent form that described the interview details. When a possible participant agreed to the interview, the email asked the participant to reply to the email with: "I consent.". Once I received consent, I coordinated interview dates and times. I based the research design of a case study on Patton (2015), who defined a case study as a product that "stands on its own as a detailed and rich story about a person, organization, event, campaign, or program – whatever the focus of study (unit of analysis)" (p. 259).

Definitions

Capability maturity model integration (CMMI): A software development process that improves productivity and reduces the time and cost of projects (Ayyagari & Atoum, 2019).

Case study: A research strategy that studies a phenomenon in its natural setting (Priya, 2020)

Coding: The process of labeling and organizing data for the identification of

different themes and the relationships between those themes (Ovienmhada et al., 2021).

Confirmability: The degree to which others confirm the researcher's interpretations and conclusions (Nassaji, 2020, p. 428).

Consequences: Changes that occur to an individual or a social system as a result of the adoption or rejection of an innovation (Rogers, 2003, p. 436)

Credibility: The degree to which the research findings and conclusions can be viewed to be believable (Nassaji, 2020, p.428).

Data saturation: The point in data collection at which no additional themes and categories arise (Daher, 2023).

Dependability: The way a study should be reported so that others could arrive at similar interpretations if they review the data (Nassaji, 2020, p. 428).

Diffusion effect: The cumulatively increasing degree of influence upon an individual to adopt or reject an innovation resulting from the activation of peer networks about the innovation in the social system (García-Avilés, 2020, p. 3).

Diffusion of innovation: A form of social process to communicate innovation through various media within a specified period between members in a social system (Sipahutar et al., 2020).

Failure: The moment in time where a software project no longer meets its expectations (Tamburri et al., 2020, p.600).

Hybrid approach: A software project model that combines a plan-driven development model with agile approaches (Yahya & Maidin, 2023).

Methodology: A general strategy that outlines how someone performs a task and

identifies the methods used (Alharahsheh & Pius, 2020).

Mindset: A person's definitions, values, and principles (Aghajani et al., 2023).

Rapport: Developing and maintaining a working relationship with a human source by managing their motivations and welfare while ensuring they understand the purpose of the relationship in order to secure reliable intelligence (Nunan et al., 2022).

Reflexivity: A set of continuous, collaborative, and multifaceted practices through which researchers self-consciously critique, appraise, and evaluate how their subjectivity and context influence the research processes (Olmos-Vega et al., 2023, p. 241).

Software development: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (Krishna & Sreekanth, 2016).

Transferability: The extent to which the researchers' interpretation or conclusions are transferable to other similar contexts (Nassaji, 2020, p.428).

Assumptions

There were five assumptions in the execution of this study.

1. My primary assumption was that the case study participants were honest and trustworthy in their interview responses. Otherwise, the information from their interviews would not provide accurate information for this case study.
2. Government software development organizations must use best practices in their respective software development processes to provide the best software. Otherwise, there would be no need to seek out these practices.
3. Interviewees were knowledgeable about the subject matter. Otherwise, their

responses would not be germane to the study.

4. Readers of this research can take the results of this research and apply them to other government software development entities. Otherwise, the results only pertain to the studied government software development entity.
5. The final assumption is that the case participants had enough experience regarding software development and the researched methodologies to understand the interview questions. Otherwise, their responses would be uninformed and impractical outside this study.

Scope and Delimitations

I focused on how a government software development organization under a CMMI-certified organization implemented a hybrid approach of the waterfall and agile methodologies. Similar software development organizations throughout the government develop, build, test, improve, and sustain software in support of United States service members, other government organizations, and United States allies. The work these organizations do is critical to the country's success. As a result, leaders/managers of these government software development entities must be aware of best practices within the software development industry and use these best practices when doing so will create better software for United States service members, other government agencies, and United States allies.

Limitations

Researchers should anticipate limitations, challenges, and barriers when dealing with government agencies. Specifically, much of the documentation for the weapons

system is designated “For Official Use Only” (FOUO). As a result, I coordinated with the local Public Affairs Office (PAO) to ensure that the information that appears in this dissertation is entirely open-source and accessible to non-government personnel. To that end, I used a non-government personal computer while writing this dissertation because the absence of a Common Access Card (CAC) kept me from accessing websites that are not open source and require a CAC to access. Additionally, I ensured that the information gathered from the participants is entirely anonymous so that their responses do not threaten their employment by violating nondisclosure agreements (NDA). To that end, I only collected positional information for each participant so that no one could connect individual responses to individual participants.

I received informal approval from the current program leader through a meeting with him and his immediate subordinate on the program. To ensure that the dissertation is in line with the guidance provided by the current program leader, I sent approved prospectuses and chapters to the program’s subordinate leader for review. Additionally, being a former member of this team could cause bias. As a former team member, my bias is to be optimistic about the past and current aspects of the program. In order to mitigate this bias, I focused on the objective data gathered through approved data collection.

Significance of the Study

This study is significant because it improves the theoretical and practical understanding of the government software development industry. Leaders/managers in the software development industry are under tremendous pressure to produce new products and features in less time, with less money, and with higher quality (Prenner et

al., 2021). As a result, Software development leaders/managers continually work to develop the best methodologies and practices to meet the criteria. The established best practices for software development have been the waterfall and agile methodologies. However, both of these methodologies have inherent weaknesses.

Regarding the waterfall methodology, software developers expressed significant areas for improvement, such as the inability to deal with changing requirements and late feedback from customers and other stakeholders (Prenner et al., 2021). To address these issues, software developers developed the agile methodology, which was more amenable to changing requirements and facilitated continual feedback from customers and stakeholders. However, the agile methodology also had significant deficiencies. The agile methodology has difficulty managing large-scale or distributed software development. Additionally, the agile methodology also has difficulty complying with security standards. Initially, software developers created the waterfall methodology to address these issues.

To address the inherent shortcomings of waterfall and agile methodologies, Prenner et al. (2021) wrote that studies showed that software development organizations only used partial waterfall and partial agile methodologies. Many software development organizations use a combination or hybrid approach of the waterfall and agile methodologies to combine the benefits of both methodologies while mitigating their respective weaknesses (Prenner et al., 2021). Research supports the idea that neither waterfall nor agile methodologies fit all software development projects, which justifies the growing exploration of hybrid methodologies between waterfall and agile (Bianchi et

al., 2021).

Significance to Practice

By exploring the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies, the research outcome will have practical significance. Specifically, the research outcome will have a meaningful impact on the real-life development of government software and the entities that use these software applications. Using the research findings, leaders/managers of government software entities can make informed decisions on the best approach for their software development programs. Due to its tremendous advantage, software developers in various software development sectors have made the hybrid approach of the waterfall and agile methodologies a "state-of-the-practice" in the software development industry (Prenner et al., 2021, p. 1). Consequently, leaders/managers of government software development entities can use this document as a credible source to convince their leaders and higher decision-makers to move away from their possible familiarity and tendencies and embrace the social change that may come from better government software development.

Significance to Theory

When the study was complete, I explored the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies and provided readers with a new understanding of the hybrid approach. In doing so, I created an awareness of how leaders/managers of government software development entities can use this hybrid approach to improve software development in their respective programs by providing them with knowledge about the benefits of the

hybrid approach. Using this theory of a hybrid approach, leaders/managers of government software development entities can tailor their software development methodology to the needs of the project (Yahya & Maidin, 2023). As a result, leaders/managers of government software development entities know how to properly assess and decide on the best software development approach and address the social problem, which is the inherent shortcomings of the two most prevalent software development methodologies – waterfall and agile -- lead to a large percentage of software development project failures. (Gaborova et al., 2021). I provided theoretical knowledge on how government software development leaders/managers can solve this root social problem by executing this study. Additionally, I provided the theoretical significance of filling the gap in the literature as it pertains to the exploration of hybrid models (Bianchi et al., 2021).

Significance to Social Change

By exploring the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology, I showed how the consequences fostered by a hybrid approach of the waterfall and agile methodologies can bring about positive social change. Using the research gathered in this document, leaders/managers can provide their leaders and decision-makers with information they may need to institute a paradigm shift in government software development from the institutionally familiar and often mandated focus on waterfall software development methodology to a more beneficial approach. By using more beneficial software development methodologies and leveraging principles of

DOI to implement the development methodologies, it is possible for leaders of software development entities to positively influence the social system of government software development and directly bring about individual or social change through better software for the entities that use their software products such as United States service members, other government entities, and allies to the United States.

Summary and Transition

In Chapter 1, I provided the objective for this case study, which was to explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology. In this chapter's problem of the study, I wrote about the research question, interview questions, theoretical framework, assumptions, limitations, delimitations, and the research approach for the case study. By deciding on the software development methodology for their respective programs, leaders/managers of government software entities play an integral role in the direction of their respective programs and their programs' overall success. In this study, I addressed the shortcomings of the two most prevalent software development methodologies – waterfall and agile – the benefits of leaders/managers of government software development entities using a hybrid approach and how these leaders can leverage the principles of DOI to implement the hybrid approach. The research question was as follows: What consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodologies regarding the inherent shortcomings of each methodology?

I presented a logical connection between the theoretical framework using DOI

and the nature of the study. I used DOI to elaborate on how leaders/managers of government could implement a hybrid approach of the waterfall and agile methodologies in their social system, government software development, to improve software development and the software it creates for United States service members, other government agencies, and United States allies (Rogers, 2003). DOI provides a framework for leaders/managers of government software development entities to implement a hybrid approach to software development, especially if decision-makers and stakeholders of their respective programs are more familiar and empathetic to other software development methodologies such as a waterfall.

Regarding the case study, the overarching government entity of the program was a CMMI-certified organization whose practices closely resembled the waterfall methodology. Additionally, the customers of software developed by the studied government software development entity were familiar with and even preferred the waterfall methodology. To explore how a government software development entity that fell under a CMMI-certified organization implemented a hybrid approach of the waterfall and agile methodologies, I conducted a descriptive case study that used eight face-to-face interviews as the primary means of data collection; I performed eight face-to-face interviews. Guest et al. (2006) wrote that data saturation has become the gold standard by which researchers determine purposive sample sizes. Later in the article, Guest et al. concluded that six to eight interviews are sufficient to reach data saturation.

Additionally, I used document analysis. For the analysis, I used documents created while I was in the program to “capture context, the unfolding of events over time,

and critical interactions, (including talks) with those involved in the activities observed" (Patton, 2015, p. 27). The methodological combinations of face-to-face interviews and document analysis illuminated the inquiry question and provided data triangulation (Patton, 2015). My primary assumption was that the participants in the case study were honest and trustworthy. The research is significant because it improved the practical and theoretical understanding of how a hybrid approach of the waterfall and agile methodologies mitigated the inherent shortcomings of each methodology and fostered consequences in studied government software development entity.

Chapter 2 is a literature review of how a hybrid approach of the waterfall and agile methodologies mitigates the inherent shortcomings of each methodology and fosters consequences in government software development. In Chapter 2, I reviewed the literature on DOI theories related to how the studied government software development organization convinced its parent organization and its customers to agree to an innovative software development methodology of a hybrid approach. In Chapter 3, I wrote about the research design and rationale, role of the researcher, participant selection, instrumentation, recruitment, data analysis and trustworthiness. In Chapter 4, I wrote of the study's findings. In Chapter 5, I presented conclusions, implications, and recommendations for future research.

Chapter 2: Literature Review

Technology projects have an astonishing failure rate. Despite extraordinary efforts and allocation of resources by leadership, only 23% of large-scale technology projects were successful, with "successful" meaning developers delivered their respective projects on time, on budget, and to a satisfactory standard (The Standish Group, 2018). The two most prevalent software development methodologies – waterfall and agile – have inherent weaknesses that are detrimental to software development (Gaborova et al., 2021). Research supports the idea that neither the waterfall nor agile methodologies fit all software development projects, which justifies the growing exploration of a hybrid methodology between waterfall and Agile (Bianchi et al., 2021). Therefore, the social problem is that the inherent shortcomings of the two most prevalent software development methodologies – waterfall and agile - lead to many software development project failures (Gaborova et al., 2021). The purpose of combining the two methodologies is to leverage the advantages of each methodology (Yahya & Maidin, 2023). This qualitative study explored the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies. To discover what consequences a hybrid waterfall and agile methodologies approach can foster, I included the research and synopsis of current literature about software development and implementation.

Software developers created software development methodologies out of necessity. In the early years of software development, developers did not have formal development methods (Kakar & Kakar, 2020). Instead, they used a "code and fix"

approach that had developers write code and fix bugs in the code until the customer was satisfied (Kakar & Kakar, 2020). This approach worked because software development was in its infancy, and developers did not know any other way to create software. However, as software became more prevalent in day-to-day life, software developers had to derive a more deliberate way of developing software that would improve the efficiency and quality of their software (Kakar & Kakar, 2020). Software developers needed a better way to create software.

Software developers needed a formal methodology. To that end, software developers created the initial foundation of software development -- the waterfall methodology (Saeed et al., 2019). Software developers recognize the waterfall methodology as the first applied software development methodology, which resembled the development methodologies of other industries at the time (Andrei et al., 2019; Saeed et al., 2019). The waterfall methodology provided much-needed structure to software development. Early on, the waterfall methodology was very successful.

Despite its early success, the waterfall methodology had some significant points for improvement. The biggest flaw of the waterfall methodology is its difficulty in making changes in a previous stage once the software development has moved beyond that stage (Hui et al., 2020). Software developers who use the waterfall methodology assume that their customers can clearly and accurately articulate almost all the requirements at the beginning of the development process. Software developers can make this assumption when their respective projects are relatively simple.

However, as software developers took on more complex projects, they could no

longer assume that their customers could clearly and accurately articulate almost all the requirements at the beginning of the development process. Software developers who use the waterfall methodology cannot react quickly to changing requirements and trends, regardless of where developers are in the process (Jangra et al., 2023). As software developers faced more challenging software development projects, they needed the ability to adapt to changes throughout the development process. However, software developers who use the agile methodology face other challenges. As a result, software developers needed an approach that accentuated the benefits of waterfall and agile methodologies and mitigated their respective weaknesses.

When faced with the challenge of using the established software development practices of its parent organization, which closely resembled the waterfall methodology, and satisfying the needs of the customer, which closely resembled the agile methodology, the studied government software development entity convinced its parent organization and customer to agree upon a hybrid approach of the waterfall and agile methodologies. For requirements establishment and tracking, the government software development organization utilized its parent organization's established best practice of using Dynamic Object Oriented Requirements System (DOORS) as its requirements management tool. DOORS is an application that optimizes requirements communication, verification, and collaboration in software development organizations by allowing the developer to select requirements from a list of attributes (Gazzawe, 2019). DOORS provides requirements traceability coverage for source code-based development (Jadoon et al., 2019). Using DOORS, subordinate leaders/managers of the parent organization can standardize all the

process activities because CMMI provides specific practices to follow at each developmental stage (Jadoon et al., 2019). As a result, the DOORS software innately provides the framework for standardization, technique adherence, and process evaluation. This framework filters down to its subordinate programs. At the beginning of the process, software developers work closely with their customers to put the requirements in the DOORS application and track requirement accomplishments throughout the software's lifecycle. However, leaders/managers of the studied government software development organization needed to incorporate the ability to adequately handle changes in requirements and environment.

To adequately handle changes in the software development lifecycle, the initial government lead of the studied government software development entity established weekly integrated planning team (IPT) meetings with key leaders of the studied government software development organization and a designated representative of the customer organization. In these IPT meetings, the leaders of the studied government software development entity gave progress reports on the software's development. The leaders of the studied government software development organization used this meeting to articulate challenges and present possible solutions for the designated customer representative's awareness and decision-making. The designated customer representative also used this IPT to disseminate changing requirements throughout the software's lifecycle. In short, this IPT served as a continuous collaboration with the customer, which is a staple within the agile methodology. When changes occurred, the leaders of the studied government software development organization used the minutes from the IPT as

the documentation needed to create change requests (CR) to the requirements, in DOORS, and other aspects of the program. As a result, the leaders/managers of the studied government software organization could incorporate aspects of the agile methodology into its parent organization's structured, accepted waterfall methodology. This study explored the consequences in government software development that a hybrid approach of the waterfall and agile methodologies could foster.

Literature Search Strategy

Since the topic of software development is such a broad genre, I performed an extensive search and review of relevant peer-reviewed articles and literature reviews. While the extensive search provided the necessary foundation for this dissertation, I used the study to sculpt the research questions that facilitated data collection, interpretation, and presentation of the findings. I used the following keywords and phrases in the search process: *hybrid, capability maturity model integration (CMMI), waterfall, agile, software development*, and *diffusion of innovation (DOI)*. I performed a literature search covering topics on software development and the current processes used by the software development industry. I used scholarly peer-reviewed articles for research studies published between 2018 and 2025 using the following databases: the Walden Library, Ebsco host, ProQuest, and Wiley. I also used Boolean/Phrase, expanders, full-text articles, limiters like peer-reviewed journals only, and university library articles.

I used the literature review to evaluate the articles and other sources about how combining waterfall and agile methodologies could improve government software development. From the literature review, I discovered pertinent concepts and findings to

support a case study on the topic. I used the literature review to discern what information and methods he should use throughout the dissertation. I answered the research questions in the subsequent chapters using the discerned information and methods.

Theoretical Foundation

To explore a government software development program that fell under a CMMI-certified organization and implemented a hybrid approach of the waterfall and agile methodologies, I used basic principles of the DOI theory to describe the needed factors to understand decisions toward adopting innovations. Scholars credit Rogers as the originator of DOI theory (Çakıroğlu et al., 2022). In his book, Rogers defined "diffusion" as how stakeholders communicate innovations to other stakeholders throughout the social system and communication as the ways stakeholders share information so that everyone in the social system comes to a mutual understanding of concepts (Rogers, 2003, p. 19). The diffusion effect is the accumulated influence on an individual or group to either adopt or reject an innovation based on the peers of the social system (García-Avilés, 2020). Rogers defined "innovation" as "an idea, practice, or object that is perceived as new by an individual or other unit of adoption" (Rogers, 2003, p. 12). DOI combines the characteristics of innovation, the innovation-decision process, and its potential adopters (Çakıroğlu et al., 2022). For example, when challenged with innovation, individuals (i.e., potential adopters) and groups go through a decision-making process to determine whether to adopt or reject the innovation; this process includes gathering information, testing, and deciding on whether to add the innovation to their respective skillsets (Rogers, 1995). Many would find this concept intuitive.

While this process may appear intuitive, researchers need to have a firm grasp of the basics of DOI to understand where failures in diffusion occur regarding innovation and how to address them adequately for the betterment of society. The four main elements in DOI are communication channels, social systems, time, and innovation (Fuah & Ganggi, 2022). Researchers can find these four elements in every diffusion research study, campaign, or program (Roger, 2003). Communication channels are the avenues that individuals use to get messages from one individual to another (Roger, 2003). There can be multiple avenues by which messages get from one individual to another, such as face-to-face communications, phone communications, and emails (Ball et al., 2014). While mass media can also serve as a communications channel, Rogers (2003) contended that interpersonal communication channels are the most effective form of communication because the direct experiences of close peers are the most effective concerning potential adoption. An examination of Nova Southeastern University's early implementation of distance education supports this concept.

Costa and Walsh (2018) used the four elements of DOI to examine the success of Nova Southeastern University's (NSU) early implementation of distance education. Interpersonal communication channels such as peer-to-peer and student-to-student encouraged NSU's adoption of distance education programs (Costa & Walsh, 2018). As a result, enrollment rose. In contrast, mass media communication channels disseminated disparaging information about NSU's educational practices (Costa & Walsh, 2018). As a result, NSU spent over a decade trying to dismiss the poor information about its program (Costa & Walsh, 2018). In the end, the peer-to-peer and student-to-student

communication prevailed. NSU is the 10th largest independent university in the country, and the Southern Regional Education Board's Electronic Campus has certified 83 of its online programs (Costa & Walsh). Costa and Walsh also wrote about DOI and social systems in their article.

Social systems are the environments for innovation. Rogers defined "social systems" as "a set of interrelated units that are engaged in joint problem solving to accomplish a common goal" (Rogers, 2003, p. 23). Because DOI occurs within social systems, the structure of social systems influences the diffusion process (Costa & Walsh, 2018). Additionally, opinion leadership and the type of innovation decisions are essential variables within the social system (Costa & Walsh, 2018). However, opinion leaders (and their authority decisions) generate the fastest DOI adoption rate (Costa & Walsh, 2018). As it pertains to NSU and social systems, serving as NSU's president, Abraham S. Fischler led the diffusion of distance education within NSU's social system (Costa & Walsh, 2018). Initially, NSU professors resisted the change in thinking to distance learning due to the perceived high cost of implementation. However, because of the NSU President's authority to implement distance learning, NSU began its transition and diffusion throughout the social system (Costa & Walsh, 2018). As more and more programs transitioned, the value to students, educators, and the social system became more and more apparent (Costa & Walsh, 2018). However, this transition did not occur overnight; it took time to implement and take hold.

In most other behavioral science studies, researchers ignore the aspect of time. However, Rogers's (2003) inclusion of time as a variable in DOI theory is one of DOI's

strengths. Rogers included the aspect of time in three instances:

1. the innovation-decision process – the time it takes for individuals to progress from their initial knowledge of the innovation to either their adoption or rejection of the innovation
2. the relative earliness or lateness that individuals adopt innovation
3. the individuals' rate of adopting innovation is measured by the number of members of the social system who adopt the innovation within a given period. In their article, Costa and Walsh started their examination of NSU regarding time with the innovation-decision process.

In 1971, NSU's Abraham S. Fischler College of Education initiated the distance education innovation-decision process. After the first year, NSU offered online master's degrees in Information Science and Computer Education and an educational specialist degree in Computer Education (Costa & Walsh, 2018). In 1983, NSU introduced online doctoral programs in Information Science and Computer Education (Costa & Walsh, 2018). By the early 1990s, NSU offered residential graduate programs to introduce online distance education alternative programs. By 2000, NSU's president, Ray Ferrero, began facilitating distance education initiatives throughout NSU (Pleasants, 2013). However, the entire movement of NSU's distance learning initiative started with innovation.

Many researchers use the terms "technology" and "innovation" interchangeably. As written earlier, Rogers defined "innovation" as "an idea, practice, or object that is perceived as new by an individual or other unit of adoption" (Rogers, 2003, p. 12). Individuals' perceived newness determines their reaction to an idea; DOI does not

consider the objective time-lapse from the idea's first use or discovery (Roger, 2003). Additionally, for individuals to adopt innovation and diffuse that innovation throughout the social system, individuals must believe that there is a relative advantage of the innovation over current technologies (Rogers, 2003). NSU's distance learning initiative was a huge success compared to traditional post-secondary instruction.

From the beginning of NSU's distance education program, students and faculty perceived the cluster and field-based teaching practices as an improvement over residential, cohort-centered practices (Costa & Walsh, 2018). The field-based distance education program involved instructors flying countrywide to conduct classes; it had a relative advantage over traditional teaching practices because it allowed students and instructors greater flexibility in the learning experience (Goldstein, 1989).

There are a few reasons why I used DOI for the present study. One of these reasons is socialization. While innovation is important, it is only helpful with socialization (Rizkiansyah et al., 2022). Also, in terms of this case study, DOI explained how the leadership of the government software development entity was able to convince its leadership (a CMMI-certified organization) and its customer (who was most comfortable with the waterfall software development model) to accept a new, hybrid approach of software development that combined aspects of waterfall and agile software development methodologies. The adoption rate is another reason I used DOI for the present study.

As written earlier, researchers measure the adoption rate of innovation by the number of members of the social system who adopt the innovation within a given period.

Rogers defines “innovativeness” as the extent to which a person or unit of adoption adopts new ideas earlier than other persons or units of adoption in the social system (Rogers, 2003, p. 22). To delineate these individuals or units of adoption, Rogers created five “adopter categories”: (a) innovators, (b) early adopters, (c) early majority, (d) late majority, and (e) laggards; these categories delineate themselves from one another based on the extent the members of that category adopt new ideas (Rogers, 2023, p. 22).

Of the five adopter categories, innovators are the quickest to adopt innovation in DOI theory. One of the critical challenges for innovators is financial constraints (Ngongoni et al., 2022). In concert, Rogers (2003) wrote that access to substantial financial resources enables innovators to withstand the negative financial impact of a failed innovation. The fewer challenges an actor has towards innovation, the quicker the actor will adopt the innovation—perceived challenges directly impact where actors fall within the adopter categories.

The next, quickest adopters of innovation in DOI theory are early adopters. While early adopters do not have the resources that innovators have, they do have the respect and trust of the social system (Çakıroğlu, 2022). Researchers identified five characteristics of early adopters: (a) innovativeness, (b) tech-savvy, (c) needing travel information, (d) having a multimodal mindset, and (e) wanting freedom of choice (Zijlstra et al., 2020). Additionally, early adopters have the highest degree of opinion leadership (Rogers, 2003). When early adopters adopt an innovation, these five characteristics help them initiate critical mass because possible adopters reference early adopters for advice and information concerning an innovation (Rogers, 2003). Routinely,

the members of the subsequent adopter categories check with early adopters before adopting a new idea (Rogers, 2003).

The early majority is the next quickest adopter in DOI theory. Even though the innovation-decision process is much longer with the early majority than innovators or early adopters, the early majority often plays the role of leaders in the diffusion of innovation (Cirus & Simonova, 2020). Rogers (2003) wrote that the early majority adopts new ideas just before the average number of people in the system. The early majority's position between the early and late adopters makes them essential in the diffusion process because they connect the different persons, units of adoption, and categories (Rogers, 2003, p. 254). In other words, the early majority serves as the "glue" in the innovation-decision process that "ties" the first two adopter categories to the last two adopter categories.

The next adopter category in DOI theory is the late majority. Rogers wrote that members of the late majority approach innovations with a "skeptical and cautious air" (Rogers, 2003, p. 284). Consequently, members of the late majority only adopt an innovation once other system members have tried and tested it (Iqbal & Zahidie, 2022). Members of the late majority have limited access to resources (Rogers, 2003). As written earlier, their limited access to resources makes members of the late majority less likely to embrace innovation unless the need for innovation is clear and the innovation's success is proven, leading to the last adopter category–laggards.

Laggards are the last adopter category in DOI theory. Also, laggards are the last in a social system to adopt innovation (Rogers, 2003). Rogers (2003) wrote that laggards

have limited resources, which implies they must be sure that a new idea will continue to be viable through adoption. However, some researchers dismiss Rogers' negative connotation of laggards. Stewart and her fellow authors (2019) wrote that nearly none of the characteristics that Rogers associated with laggards were accurate.

In contrast to Rogers' perspective, Stewart et al. (2019) wrote that laggards are slow to adopt innovation because of their personal experiences and the personal experiences of others in this category. They resist the idea of data and other empirical evidence over their own experience (Stewart et al., 2019). Lastly, Stewart et al. wrote that it is a mistake to assume that laggards eventually succumb to innovation. Instead, others in the social system should work to include other viewpoints for the sake of the system.

Many individuals assume that faster rates of adoption are optimal. However, Andrews (2023) wrote that adopters may prefer slower adoption rates. This paradox stems from the different perspectives of actors – the members of the social system and the implementers of the innovation. Social system members want innovations as soon as possible because of the possible positive impacts on their lives. In contrast, other innovation stakeholders must carefully weigh factors such as cost, delays, and impact on professional reputation and longevity if the innovation is unsuccessful. As a result, a slower adoption rate may allow other stakeholders to become more comfortable with the innovation and even become advocates for it. Iqbal and Zahidie (2022) used a bell curve that represents the percentage of members in each of the adopter categories.

In the bell curve, Iqbal and Zahidie (2022) conveyed that innovators comprise 2.5% of the social system. In turn, early adopters make up 13.5%, the early majority

make up 34%, the late majority make up 34%, and laggards make up 16%. Readers can infer much from this information that aligns with I wrote in this paper. For example, innovators make up the smallest percentage of the social system because they tend to possess financial resources that are not available to members of the other adopter categories in the social system. The early and late majorities are also more significant than other adopter categories. By sheer numbers, these categories would provide the most force in the adoption of innovations, with members of the early majority serving as the "glue" for innovation because they provide interconnectedness throughout the social system.

When researchers study the adaptor categories of DOI, they seamlessly move through the consequences of innovation. Despite their importance, researchers have given little study to consequences, which are the "ultimate end" to the means of innovation and diffusion (Rogers, 2003, p. 436). Rogers (2003) wrote of three dimensions of consequences: (a) desirable versus undesirable, (b) direct versus indirect, and (c) anticipated versus unanticipated. Desirable consequences are the functional consequences of an innovation; undesirable consequences are the dysfunctional consequences of an innovation (Roger, 2003). Direct consequences are consequences that occur in "immediate response" to innovation; indirect consequences are consequences that occur due to the direct consequences of the innovation (Rogers, 2003, p. 445). Anticipated consequences are consequences that stakeholders foresaw due to the innovation; unanticipated consequences are consequences that stakeholders did not foresee (Roger, 2003). Since innovation comes with a variety of consequences, change agents and

stakeholders must be aware of these consequences for the sake of the overall objective. In terms of software development, the overall objective would be a better product for the customer.

The research question relates to and builds upon existing theory. As it pertains to methodologies for software development, Bianchi and his fellow authors wrote: “one size does not fit all,” which grounds and justifies “the growing exploration of hybrid models” (Bianchi et al., 2021, p.31). To that end, this case study explored what consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodology as it pertains to the inherent shortcomings of each methodology. However, I wrote about best practices before writing on methodologies.

Literature Review

Organizations aggressively seek increased productivity and efficiency. Sreenivasan and Kothandaraman (2019) wrote that CMMI is a collection of best practices for process improvement that organizations can use in various industries to improve productivity, reduce defects, optimize processes, and ensure the predictability of operations (Sreenivasan Kothandaraman, 2019). Sreenivasan and Kothandaraman (2019) also wrote that CMMI is one of the most popular models for improving products and services (Sreenivasan & Kothandaraman, 2019). CMMI has five levels: (a) initial, (b) managed, (c) defined, (d) predictable, and (e) optimization (Hao, 2022). Brahmantara et al. depict each level. Each Maturity Level (ML) includes basic principles in multiple human resource management fields (Hao, 2022). The CMMI model’s first Maturity Level (ML-1) is “Initial.”

ML-1 (initial) is the foundation for the CMMI model; the remaining four levels build on this maturity level. ML-1 represents a process that is “unmanaged, undefined, and chaotic” (Brahmantara et al., 2021, p. 89). This description implies that the organization has just begun the development of its processes. Consequently, the organization’s leaders still need to assign roles in the process and define those roles. As a result, the process could be more coherent. They are neither easily duplicated nor documented. From ML-1 (initial), organizations hope to progress to ML-2 of the CMMI model – managed.

ML-2 (managed) is the next intuitive step when organizations attempt to solidify their processes. At ML-2, organizations have an established project management system (Thomas et al., 2022). Establishing a project management system begins with addressing the absence of assigned roles and defining those roles as they pertain to project management. When project management/leadership defines the roles below their levels, the project management/leadership provides greater fidelity to the roles within the project and the team members in their purview. This excellent fidelity appears in subsequent levels of the CMMI model.

ML-3 (defined) formally defines processes. At ML-3, organizational leadership/management defines processes and integrates those processes into the development life cycle (Machado et al., 2022). Organizations formally document processes at this level if they need to be replicated or improved. Once organizational leadership/management defines their processes, they must marry them to their organizational goals.

ML-4 (qualitatively managed / predictable) is where the processes are married to goals. At ML-4, organizational leaders assign all processes quantitative goals based on quality and process performance; organizational leaders refer to these goals for process management (Luo & Chen, 2022). ML-4 consists of four components: (a) resource management capability, (b) multiple assessment capability, (c) evaluation capability, and (d) group learning processes (Luo & Chen, 2022). Leaders integrate processes and evaluations at a group level and marry processes to goals. As organizations move to the final level of the CMMI model, leaders try to get the most from their respective organizations, processes, and team members.

ML-5 (optimizing / optimization) is the final and highest level of the CMMI model. At ML-5, organizations are mature with highly effective processes that ensure predictability in results (Dutta & Omolayole, 2021). While no process nor model is perfect, ML-5 organizations have enough documentation and process evaluation to give organizational leaders/managers confidence in the process and the products derived from the program. While researchers consider ML-1 organizations immature, employing ad hoc processes with unpredictable results, researchers consider the subsequent four levels of the CMMI process a systematic progression to ensure product development success.

Most researchers group CMMI Level 2 with CMMI Level 3 and CMMI Level 4 with CMMI Level 5. Organizational leaders/managers for most CMMI organizations stop at Level 3 or lower (Grossi et al., 2014). For DoD Acquisition Category I programs, the Undersecretary of Defense set CMMI Maturity Level 3 as a screening criterion for software development contractors, and CMMI Maturity Level 5 became a discriminator

(Grossi et al., 2014). Organizational leaders perceived that the most significant obstacle to moving their organizations from CMMI Level 3 to CMMI Level 5 was the return on their investment (Grossi et al., 2014). Consequently, as government contracts for software development became more competitive, organizational leaders became more incentivized to move their organizations from CMMI Level 3 to CMMI Level 5. However, competitive advantage was not the only reason many organizational leaders used CMMI.

Along with competitive advantage, organizational leaders/managers want to improve their organizations. CMMI is one of the software industry's most successful tools to influence process assessment effectively and accomplish the required process improvements of organizations (Thomas et al., 2022). CMMI's ability to influence practical improvements is another reason CMMI is prevalent within the government software development community. Since its development in 1999 by Carnegie Mellon, CMMI has been the industry standard for process development and evaluation (Khan et al., 2021). As a result, current government agency leaders have developed a familiarity and comfort level with software development projects that follow the CMMI model. For this reason, most government software development agencies use CMMI as a foundation for their processes and evaluation. To this end, these government software development agencies are more than willing to allocate the resources necessary to attain the highest CMMI designation.

The entity in the case study fell under a CMMI-certified organization. DoD initially sponsored CMMI, and large companies that used traditional software

development quickly adopted CMMI (Henriquez et al., 2022). This well-founded practice of using CMMI within government agencies makes its decision-makers comfortable with large software development projects that use CMMI. The software development process most aligned with CMMI is the Software Development Lifecycle (SDLC) methodology.

The complexity of software development consists of many processes that developers must execute. Developers call this collection of processes the software development life cycle (Garg et al., 2022). Developers use SDLC to design, build, and validate high-quality software (Garg et al., 2022). To that end, SDLC consists of six phases.

Garg et al. (2022) conveyed that SDLC offers in a structured way to develop and deliver customer software products that meet their needs. While all SDLC methodologies share basic principles and characteristics, developers' choice of which SDLC methodology to use for their respective projects is critical to the project's overall success (Garg et al., 2022). The first phase of SDLC is Analysis.

In the Analysis phase of SDLC, customers and software developers initiate the SDLC process. Customers and software developers discuss the requirements and exchange primary data (Khan & Kumari, 2021). Additionally, Khan and Kumari (2021) wrote that the Analysis phase is the most critical phase of the SDLC methodology because it is the foundation for any quality software. Software developers use the Analysis phase for the initial direction of the entire project. Once software developers have this direction, they can move to the next phase of SDLC – Design.

In the Design phase of SDLC, software developers plan and execute the initial

software and its delivery. Software developers check to see if the initially delivered software fits all the requirements that customers presented in the Analysis phase (Dwivedi et al., 2022). It is rare that software developers successfully address all the requirements in the initial delivery, especially for large-scale projects. As a result, customers must implement the software in the actual environment (or a close replica of the actual environment) to see how it performs. Implementing the software in the actual environment is the next phase of SLDC – Implementation.

In the Implementation phase of SDLC, customers and software developers see and document how the initially delivered software performs. Software developers begin their precise development and build the programming of the software. In the Implementation phase, software developers select and train personnel, install the software, and test it (Rahmani & Hikmawati, 2020). In turn, customers must articulate and document detailed and honest feedback on the performance of the software so that the software developers can refine the software. Because of the detail and collaboration required between the customer and software developers during his phase, the Implementation phase is the most prolonged in SDLC (Hoti et al., 2023). The Implementation phase is also critical to the overall success of the software project. Once software developers have implemented the software, they must test it. Testing the software is the next phase of SLDC–Testing.

In the Testing phase of SDLC, software developers add rigor to the software development phase. Software developers use the Testing phase to determine the outcome of utilization and validate expected results of the software (Gupta & Gayathri, 2022). In

concert with their customers, software developers create the environment and criteria for testing and the thresholds determining the software's success or failure. Once software developers complete the testing phase, customers and software developers determine how they will disseminate the software throughout the intended organization. Disseminating the software throughout the organization is the next phase of SLDC–Deployment.

In the Deployment phase, software developers turn over the software to their customers. The software developers release the software to customers and lay out the schedule for the deployment process throughout the customers' organizations (Pothukuchi et al., 2023). Software developers use configuration management to formally deliver the software to the customer (Freitas et al., 2023). Chatziamanetoglou and Rantos (2023) defined "configuration management" as a disciplined and structured approach to monitoring and controlling changes to maintain the integrity and consistency of a supported system. In practical terms, configuration management tracks and documents all facets of a project, including the formal delivery of software to the customer. Once delivered, software developers still have a vital role to play in the lifecycle of their developed software. Software developers must care for the delivered software. Caring for the delivered software is the next phase of SLDC–Maintenance.

In the Maintenance phase, software developers can improve the delivered software after formally delivering it to the customer. Software developers can make changes such as upgrades, repairs, and "bug" fixes that neither the software developers nor the customer caught prior to formal delivery (Jalaja & Adilakshmi, 2023). Customers request changes to the software developer as a change request (CR) (Jalaja &

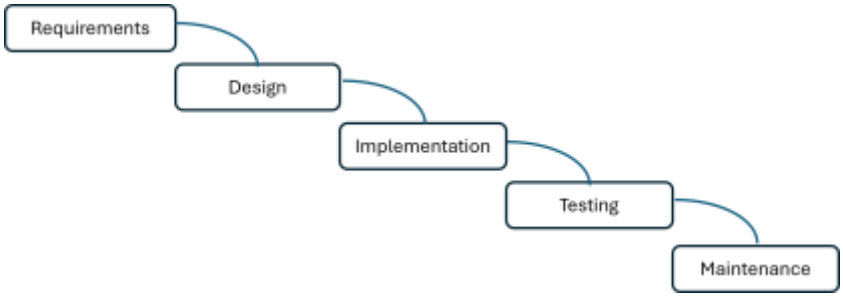
Adilakshmi, 2023). Software developers can use SDLC to create multiple methodologies for software development.

There are several methodologies within SDLC. The most popular SDLC methodologies are waterfall and agile (Ismail & Dawoud, 2022). Between these two methodologies, the waterfall methodology has been around the longest. However, the agile methodology has been around for decades and is well established. Since software developers established waterfall first, I began my comparison with the waterfall methodology.

The waterfall methodology has been around for several decades. Winston Royce introduced the waterfall methodology in 1970, and software developers further developed the methodology throughout the years (Aroral, 2021). The waterfall methodology follows a sequential progression. The waterfall methodology has five stages: (a) requirement analysis, (b) system design, (c) implementation, (d) testing, and (e) operation and maintenance (Henoch & Yerik, 2023). Figure 1 depicts the logical progression or waterfall of development and how this methodology received its name.

Figure 1

Figure of the Waterfall Methodology



Many software development projects use the waterfall methodology. Because of

its longstanding popularity in the software development industry, experienced software development people are remarkably familiar with its components and foundations (Horton, 2020). The foundation of the waterfall methodology is its five stages. The waterfall methodology begins with the initial stage–Requirement Analysis.

Requirement Analysis provides the foundation for the waterfall methodology. Software developers use several information collection techniques to grasp their projects' requirements at this stage. These collection techniques could include discussions with the customer, surveys within and outside the projects, interviews with team members/colleagues, and observations of the projects' environments (Ramadani et al., 2022). These collection techniques enable software developers to understand the purposes and objectives of software projects and how they support the organization's overall mission. From this foundation, software developers can move from Requirements Analysis to the next stage–System Design.

System Design is the second stage in the waterfall methodology. System Design incorporates the requirements for the project and how software developers intend to create a product that addresses those requirements to the customer's satisfaction (Chintami & Astriratma, 2021). Software developers addressing the requirements to the satisfaction of the customer is critical. Developers must ensure that the design efficiently supports the customer's needs within the constraints the customer provides. While this concept may appear intuitive, the failure of software developers to provide the customer with what is needed is a key reason many projects fail (Chintami & Astriratma, 2021). However, suppose developers do an excellent job in this project step. In that case, their

actions create the environment for success in the subsequent steps of the waterfall methodology and the overall successful project. To that end, developers can move to the next stage–Implementation.

Implementation is the third stage in the waterfall methodology. In this step, software developers write the original code and compile an operational application (Heriyanti & Ishak, 2020). Software developers have created the software and introduced it in the actual intended environment or an environment that closely resembles the intended environment (Hasibuan & Zakir, 2021). As the name of this step suggests, software developers have a product to satisfy the customer and an environment in which to see if the product meets the requirements. With that in mind, developers move to the next stage – Testing.

Testing is the fourth stage in the waterfall methodology. The purpose of testing (or verification) is to determine whether there are errors (or "bugs") that exist in the software when it is running (Santonanda et al., 2022). Additionally, testing determines whether or not the features in the software run according to what the customer desires (Santonanda et al., 2022). Due to the complexity of most software development projects, all software has errors when developers first test it. Testing aims to isolate and fix software errors that keep it from performing its desired task(s). To that end, testing is a crucial step in all software development projects. The customer must be involved in the testing process to ensure that the software satisfies the requirements that customers and developers agreed upon at the beginning of the waterfall process. Once software developers complete testing to the extent that the customer is satisfied with the product,

continued oversight of the software is necessary. Continued oversight of the software is the next and final stage–Operation and Maintenance.

Operation and Maintenance is the fifth and final stage in the waterfall methodology. In this stage, developers perform regular maintenance on the software to continue to run according to its function (Saleh & Ipantri, 2023). Additionally, maintenance can consist of further developing the software to satisfy the customer's growing needs (Saleh & Ipantri, 2023). As end users work with the software, they may have ideas for making slight improvements to it without having to make drastic changes. The Operation and Maintenance stage allows developers to implement these minor, unintrusive changes to the software to make the experience better for the users. Additionally, interaction with the software may cause errors to occur. The Operation and Maintenance stage allows developers to correct these problems, as well.

The sequential approach of the waterfall is excellent when the requirements are well-known at the beginning of the process. However, software developers have found that this sequential approach to software development is not optimal for projects with frequently changing requirements and priorities (Hema et al., 2020). As software development projects became more complex and requirements often changed during the SDLC, the sequential nature of the waterfall methodology became less conducive to successful project development. As a result, the need for another option for software development became apparent.

Another SDLC software development methodology is the agile methodology. The agile software development methodology is the other most popular software development

methodology (Horton, 2020). In 2001, 17 participants collaborated to create the Agile Manifesto (Kakar, 2023). In contrast to the waterfall methodology, the agile methodology is far less rigid and focuses the software development process on the customer and change versus the overall process. To navigate this far less rigid approach to software development, creators of the agile methodology established values and principles that explain the essence of agile and serve as guidelines as software developers and their customers work through the software development process.

The agile methodology has four fundamental values. These values are as follows: (a) individuals and interactions over processes and tools, (b) software over documentation, (c) customer collaboration over initial agreements, and (d) flexibility over following a plan (Lalmi et al., 2022). These values lend themselves to a more iterative process versus the sequential process of the waterfall methodology. Additionally, these values lend themselves to far more customer involvement throughout the software development process (Lalmi et al., 2022). Customers who are more involved throughout the development process may feel more ownership of the software and become more invested in its overall success. Along with values, the agile methodology has principles to facilitate the overall success of the process.

The agile methodology has 12 overarching principles. These 12 principles form a concept of what the agile methodology is. These 12 principles are as follows:

1. Software developers' highest priority is to satisfy the customer through early and continuous software delivery.
2. Customers and developers must work together daily throughout the project.

3. Face-to-face conversation is the best method for conveying information.
4. Software development leaders/managers should build projects around motivated individuals, give them the environment/support they need, and trust them to do the job.
5. Self-organizing teams produce the best architectures, requirements, and designs.
6. Teams should try to become more effective at regular intervals and adjust their behavior accordingly.
7. Software developers should deliver software with a preference towards shorter timescales.
8. Simplicity–the art of maximizing the work not done–is essential.
9. Stakeholders should welcome changing requirements, even late in development, because leveraging change can provide a competitive advantage.
10. The primary measure of progress is working software.
11. Stakeholders' continuous attention to technical excellence and good design enhances flexibility.
12. Stakeholders should maintain a constant pace indefinitely because the agile processes promote sustainable development

(Sharp et al., 2020). Researchers understand the agile methodology better by analyzing these twelve principles individually.

The first principle of the agile methodology is to satisfy the customer by

delivering software early and continuously. When software developers provide early and continuous software delivery, they establish trust and flexibility between the customer and the developers (Gheorghe et al., 2020). By providing early and continuous delivery of software, developers receive valuable feedback from the customer regarding their understanding of the requirements and how to best meet them. Customers gain confidence in the process and product because they see or use the software's primary, partially functioning iterations (Gheorghe et al., 2020). Through this interaction with the software, customers can make changes in requirements and prioritization for the product's overall success (Gheorghe et al., 2020). However, to meet this principle of continuous software delivery, customers and developers must have enormous communication throughout the development process.

The following principle of the agile methodology is that customers and developers must work together daily throughout the process. These frequent interactions between the customer and the developers allow the software developers to continually ask their customers questions on topics that directly affect the project (Gheorghe et al., 2020). These interactions allow developers to remain in tune with the customer as the environment of the project changes. Effective communication between the customer and software developers enables stakeholders to navigate changes within and outside the project successfully.

The following principle of the agile methodology is face-to-face communication, which is the most efficient and effective means of communication. While technology provides many avenues for communication, this principle stresses direct, in-person

communication between customers and software developers versus written specifications, plans, or formal agreements. (Gheorghe et al., 2020). When requirements or priorities change, direct, in-person communications allow software developers to ask the customer questions immediately to understand intent. Additionally, direct, in-person communication can allow the software developer to take informal cues from the customer and allow software developers to anticipate future needs and changes. When software developers anticipate change versus react to it, software development leaders/managers can empower their team members to take the initiative and be creative problem anticipators and solvers.

The following principle of the agile methodology is building teams with motivated individuals and creating an environment that supports their needs. Software development leaders/managers must realize that their team members are the most important factor in success (Gheorghe et al., 2020). To that end, software leaders/managers must trust and empower their team members to make the appropriate decisions to do their respective jobs at the appropriate levels. Team members should have the power to change process steps if those steps are an obstacle to the team and project success (Gheorghe et al., 2020). Software leaders/managers must effectively organize their teams for this principle to have the most significant effect.

The following principle of the agile methodology is that self-organizing teams get the best effect. When software development leaders/managers self-organize their teams, they share the project responsibilities and determine the best way to handle them through the knowledge gained throughout the development process. This knowledge and

assertiveness achieve the optimal team structure (Gheorghe et al., 2020). Teams change and adapt for the primary purpose of overall project success. However, this optimal team structure comes from internal reflection within the self-organized teams.

The following principle of the agile methodology is regularly reflecting on how to become more effective. Teams should regularly and intentionally reflect on their practices and encourage initiatives that make the team more effective, and it is just as crucial for teams to incorporate these initiatives (Gheorghe et al., 2020). When software development leaders/managers incorporate good initiatives, they do more than create better processes; they cultivate an environment where team members feel their leadership appreciates their thoughts and ideas. By doing so, leaders/managers encourage more initiatives for improvement and increase overall team morale (Kiker et al., 2019). Also, leaders/managers can improve team morale through frequent, successful deliveries to the customer that demonstrate overall project success.

The following principle of the agile methodology is for software developers to deliver software frequently with shorter time scales. Of course, “frequently” and “shorter” are relative words. This principle attempts to restrict the first principle–delivering software early and continuously–by introducing relative terms that focus on the needs and desires of the customer (Gheorghe et al., 2020). Customers determine the frequency and duration of time scales and the actual values for these terms. Not only does this principle focus on the customer’s needs, but it also enables software developers to focus their efforts on what customers deem essential.

The following principle of the agile methodology is simplicity. Software

developers should focus on delivering a simple product that can handle sudden changes while fulfilling customer requirements (Gheorghe et al., 2020). To reach this goal, software developers must create efficient and effective software that allows for changes even after they make the final software delivery. By keeping the software focused on the customers' needs, software developers produce a product that can easily interact and change as the customers' needs change to adapt to an ever-changing environment. In other words, customers and developers must encourage and welcome change.

The following principle of the agile methodology is welcoming changing requirements to give the customer a competitive advantage. As the focus of an environment changes, customers must adapt; the ability to adapt and change quicker than competitors provides a competitive advantage in the marketplace. Consequently, software developers should not view changing requirements as a hassle but as a way to sustain their customers' competitive advantage in their respective marketplaces (Salgado et al., 2022). To do this, developers must focus on the software, not the process of making the software.

The following principle of the agile methodology is that working software is the primary measure of progress. The University of Alabama head football coach, Nick Saban, is famous for his belief in focusing on the process rather than the outcome (Johnson, 2022). However, this approach is tailored to a set of steps that have proven successful in the past and a landscape that rarely changes. In contrast, software developers work in an ever-changing landscape with many internal and external factors influencing overall success. To navigate this ever-changing landscape, software

developers who use the agile methodology break down the final product into smaller pieces, leading to early and frequent software delivery. Using this approach, software developers have a better way to measure progress and have more impact through the running codes (Gheorghe et al., 2020). However, software developers must pay continuous attention to technical excellence and good design to execute this principle.

The following principle of the agile methodology is continual attention to technical excellence and sound design. To provide the highest quality software, developers should strive to be flexible and responsive to change (Gheorghe et al., 2020). Software developers should be eager to make such changes to the software while preserving the overall external behavior as a response to changes in requirements (AlOmar et al., 2021). While software developers may find this continual change taxing, leaders/managers should seek a sustainable pace of development so that their subordinates do not feel burnt out by the activity and frequent changes.

The following principle of the agile methodology is sustainable development. Stakeholders should be able to maintain this pace of constant change indefinitely (Gheorghe et al., 2020). To that end, leaders and managers must strike a good balance between the rate of change and their team members' ability to maintain that rate of change. Otherwise, teams may have significant turnover rates, which could harm overall team success (Zhang et al., 2019).

In contrast to the waterfall methodology's sequential progression, agile has a more cyclical, circular approach to software development. In its presentation for an agile scrum workshop, the creators of the workshop depicted the agile methodology as a

cyclical, iterative approach to software development (Agile and Scrum Workshops Start Jan. 31, 2019, n.d.). This cyclical, iterative approach makes the agile methodology more amenable to changing requirements throughout the software development process. While the waterfall and agile methodologies are graphically portrayed differently, both methodologies adhere to the overarching SDLC process.

As written earlier, the waterfall methodology follows a linear progression of software development. A developer can only move to the next step in the methodology after completing the prior step, which makes the methodology simplistic and easy to follow (Aroral, 2021). This logical progression and simple structure made the complex software development process much more manageable. Making the software development process more manageable has led to other advantages.

Since the waterfall methodology includes a sequence of steps, software developers find faults in one phase before moving to the next. Software developers sometimes identify intricate details and anticipate errors before writing code (Aroral, 2021). By doing so, software developers save immense development time (Aroral, 2021). However, the waterfall methodology does more than save development time.

By using the waterfall methodology, software developers are more amenable to developing technical documents properly. Properly developed technical documents make it easier for customers to know what to expect from the software (Aroral, 2021). Because software developers properly developed technical documents, their efforts helped the software development team. When new team members join the software development team, they have a much easier time getting up to speed on the project due to the

adequately developed technical documentation (Aroral, 2021). Along with advantages for the team's customers and fellow software developers, the waterfall methodology also provides advantages for the team's leadership/management.

Accurate time and cost estimates are always crucial to the leadership of software development teams. When software developers follow the waterfall methodology, they can provide accurate time and cost estimates to their leadership and customers (Aroral, 2021). When software development leaders/managers accurately communicate time and cost estimates to the customer, they keep projects on time and within budgetary constraints, which is critical to overall project success. However, team leadership and management must know and recognize the best waterfall methodology environments.

Determining the best courses of action for their project is essential for software development leaders/managers. Software development leaders/managers should use the waterfall methodology for small projects requiring fewer resources than larger ones (Aroral, 2021). Additionally, the waterfall methodology enables software development leaders/managers to compartmentalize the processes and allocate time and other resources accordingly (Aroral, 2021). By compartmentalizing the team regarding the process, software development leaders/managers can create a detailed timeline for project completion based on the designated time given to each department of the process. Lastly, software development leaders/managers should use the waterfall methodology for service-oriented projects and non-tangible deliverables such as code, copywriting, and designing projects (Aroral, 2021). However, there are instances where software leaders/managers of projects should not use the waterfall methodology.

Despite its advantages, there are disadvantages to the waterfall methodology that software development leaders/managers should be aware of. Under the software development methodology, software developers will face difficulties if there are changes in the requirements throughout software development (Evelin et al., 2021). Additionally, software developers may encounter problems later in the software development process due to the sequential nature of the waterfall methodology (Evelin et al., 2021). Lastly, software developers risk long release cycles and a high risk of manual deployment when they decide to use the waterfall methodology (Chen et al., 2022). However, challenges like the ones the researcher described in this document can result in software developers failing to meet the customers' requirements (Evelin et al., 2021). As the software development process matured, software developers realized that they needed a different methodology that addressed the shortcomings of the waterfall methodology.

In response to the shortcomings of the waterfall methodology, software developers created the agile methodology. Software developers created the agile methodology to adequately handle changes in project requirements at any stage (Al-Saqqa et al., 2020). In the agile methodology, software developers embark on continual feedback with the customer more collaboratively to create an end product. This collaborative process differs significantly from the formal agreement and contracts at the heart of the waterfall methodology (Al-Saqqa et al., 2020). However, even in the agile methodology, software developers still need some agreement between themselves and the customer to articulate how the parties will negotiate and document changing requirements (Al-Saqqa et al., 2020). In the agile methodology, software development

leaders and managers have a better process for handling changing requirements (Beerbaum, 2019). Handling change is just one advantage of the agile methodology.

Along with adequately handling changing requirements, the agile methodology has advantages regarding the end product. Using the agile methodology, software development leaders/managers have increased flexibility because of the customer's focus on the end product versus the software development process (Beerbaum, 2019). Additionally, when teams and team members have daily interactive sessions, the teams and their respective members become more cohesive due to the collaborative culture of the project (Beerbaum, 2019). The collaborative culture facilitates immediate feedback between teams and among team members. This interaction enables teams and team members to address issues earlier in the process (Beerbaum, 2019). Lastly, project leaders and managers can celebrate success with all team members (Beerbaum, 2019). While it has significant advantages for software development, like waterfall, the agile methodology also has disadvantages.

One of the disadvantages of agile is its immaturity. As written earlier, software developers created the waterfall methodology in the 1970s. At the same time, the 17 participants at The Lodge at Snowbird Ski Resort in the Wasatch Mountains of Utah did not publish the Manifesto for Agile Software Development until 2001 (Kakar, 2023). As a result, most senior software developers–the leaders/managers of software development projects–are more familiar and comfortable with the waterfall methodology than with the agile methodology. This familiarity and comfort with the waterfall methodology versus the agile methodology make decision-makers of many software development projects

more inclined to use the waterfall methodology. Along with a lack of familiarity, the agile methodology has other disadvantages compared to the waterfall methodology.

In addition to the relative maturity of the agile methodology, software developers must start planning the project before starting any work. Some software developers may see this as a disadvantage of the agile methodology because they cannot negotiate the software development process with the product in mind, which causes the planning of the process to be less in-depth (Kakar, 2023). Additionally, while traversing projects that use the agile methodology, software developers may have to change their project's course, priority, and direction several times throughout the software development process; these multiple "course corrections" can cause delays in scheduling and deliverables. Many software developers prefer established and agreed-upon deadlines and timetables, which are not the focus of the agile method (Kakar, 2023). The lack of established and agreed-upon deadlines and timetables can make the project leadership/management job far more challenging. Lastly, project leadership/management could find the continual iterations and improvements extremely challenging, and the end product may look much different than its initial goal (Kakar, 2023). The disadvantages of the agile methodology go beyond a project's leadership/management levels.

The nature of the agile methodology and the inherent smaller sizes of software development teams pose unique challenges to their team members. Below the leadership/management levels, team members must be highly skilled in software development to seamlessly adjust to agile projects' ever-changing directions and focus (Trivedi, 2021). With the ever-changing directions and focuses of agile projects, project

leadership/management may find keeping team members dedicated to specific tasks and components within the project too cumbersome (Trivedi, 2021). Team members may need to be "jacks-of-all-trades," which puts a higher value on the more extensive, more varied skill set of team members (Trivedi, 2021, p. 93). If team members do not have the required, varied skill sets, they may feel inadequate, leading to poor individual and team morale (Al-Suraihi et al., 2021). The inherent shortcomings of both the waterfall and Agile methodologies lend themselves to a hybrid approach of the waterfall and agile methodologies that enhances the strengths of both methodologies while mitigating their respective weaknesses.

As was written earlier in Chapter 1 of this dissertation, a hybrid combines two or more entities, with the end product accentuating their strengths while mitigating their weaknesses. The two principal methodologies in software development are waterfall and agile (Popa, 2021). By combining the two principal methodologies in software development -- waterfall and agile -- the result would be a hybrid software development methodology that accentuates the strengths of waterfall and agile but mitigates their weaknesses. Software developers have coined this combination of waterfall and agile as a hybrid approach.

The hybrid approach combines software development, which can be plan-driven development (waterfall) or flexibility (agile). Additionally, the hybrid approach combines software delivery, which can be incremental (waterfall) or iterative (agile). Yahya, et al. 2021 conveyed the degrees of change within the project also tie directly to the frequency of deliverables and the relationship of all these facets of software development. In turn,

the more frequent the delivery, the more desired methodology tends to be plan-driven (waterfall). In order to achieve a balance between the two methodologies, a hybrid approach is more amenable than either waterfall or agile. Therefore, many researchers have done extensive work on combining the waterfall and agile methodologies.

The research on the hybrid approach of the waterfall and agile methodologies has shown many successes. In their paper, Razali and fellow authors (2021) wrote a paper on how the Legal Advisor Office (LAO) for Universiti Teknologi MARA (UiTM) developed an integrated web-based management system for themselves and other UiTM entities to generate, update, manage, track, report, and monitor legal documents more efficiently. Prior to the development of the web-based management system, the LAO faced many challenges, such as the inability to coordinate the drafting, vetting, editing, endorsement, and approval of legal documents (Razali et al., 2021). Because they relied so heavily on email and physical mail, the LAO needed help monitoring and tracking the progress of legal documents adequately (Razali et al., 2021). This antiquated business method led to communication breakdowns between the LAO and their clients (Razali et al., 2021). To address these and other issues, the LAO needed a collaborative tool that enabled it to have real-time communication with its clients (Razali et al., 2021).

Additionally, the tool needed to act as a virtual workspace with centralized resources and tools (Razali et al., 2021). The solution was a customized tool named "Networked Integrated Legal Affairs Management System" or (NILAMs). The NILAMs had to have three distinctive features, which made them different from most commercial systems. First, the developers had to customize NILAMs for the LAO of public

universities, government agencies, and statutory bodies (Razali et al., 2021). Second, the developers had to provide four main legal activities: (a) drafting of Memorandums of Understanding (MoU)/ Memorandums of Agreement (MoA), (b) vetting legal documents, (c) writing legal opinions, and (d) supplying written answers to parliamentary questions about UiTM (Razali et al., 2021). Third, the developers had to ensure that NILAMs complied with the Ministry of Higher Education (MOHE) and UiTM policies on managing legal documents (Razali et al., 2021). The developers used a hybrid approach of the waterfall and agile methodologies to provide these three distinctive features. Specifically, the developers combined well-structured, comprehensive procedures (waterfall) with flexible practices (agile). However, this was different from the developers' original plan.

Originally, the developers planned to use the waterfall methodology. However, as the developers progressed in the early stage of the project, they decided on a more balanced software development approach (Razali et al., 2021). The developers wanted a more traditional configuration management and documentation approach–waterfall. However, the developers wanted a more flexible approach for requirements development, implementation, integration, and testing–agile. As a result, the developers and stakeholders for NILAMs decided on a hybrid strategy of waterfall and agile that would balance the interests of the project (Razali et al., 2021). The concept of balancing conflicting interests is a focal point of the hybrid approach and essential to the next example of the hybrid approach.

In a case study in Macau, China, the authors investigate project management

methodology for the Macau construction industry. Like software development, the construction industry wants to increase customer feedback, communications, and planning while decreasing the waste of resources (Lobo Marques et al., 2023). As written earlier in this dissertation, Lobo Marques and fellow authors list the advantages and disadvantages of the waterfall and agile methodologies. Then, the authors wrote of how combining the two methodologies–waterfall and agile–enhanced their strengths while mitigating their weaknesses.

Lobo Marques et al. (2023) listed the advantages of the waterfall methodology. In the waterfall methodology, the customer and developers agree upon the project's requirements early in the process. This action forces the customer and developer to focus on the requirements and ask vital questions to drive the latter steps. In the waterfall methodology, developers can show more quantifiable execution (Lobo Marques et al., 2023). Developers can use quantifiable execution to show their projects' objective, steady progress. In the waterfall methodology, project teams can focus on their respective part of the task versus the entire task (Lobo Marques et al., 2023). When teams can focus on their tasks, they become experts and increase their efficiency and effectiveness on those specific tasks. Lastly, in the waterfall methodology, customers only need a little after they have agreed upon the requirements (Lobo Marques et al., 2023). When customers are not in the same vicinity as the developers or their schedules conflict, all stakeholders value this aspect of the waterfall methodology. However, along with the advantages, Lobo Marques and fellow authors also list the disadvantages of the waterfall methodology.

Lobo Marques et al. (2023) wrote of two significant disadvantages of the

waterfall methodology. First, in the waterfall methodology, customers have difficulty defining and gathering documentation for the requirements (Lobo Marques et al., 2023). Specifically, in highly complex projects with long development times in ever-changing environments, like construction and software development, it is challenging for customers to gather enough information to define and document requirements that will not change multiple times before the completion of the project. Second, in the waterfall methodology, customers are often dissatisfied with the end product because developers used requirements agreed upon at the beginning of the project but are not viable after the project (Lobo Marques et al., 2023). Developers provide customers with an end product that satisfies the customer's past needs, not the customer's present or future needs. The agile methodology attempts to address these shortcomings of the waterfall methodology.

Lobo Marques et al. (2023) listed the advantages of the agile methodology. In the agile methodology, customers are frequently involved throughout the development process, especially at decision and change points (Lobo Marques et al., 2023). Since there is customer involvement throughout the development process, they can communicate changing requirements due to changes that occur after the initial requirements development phase. In the agile methodology, customers have a sense of ownership because of their involvement throughout the development process (Lobo Marques et al., 2023). When customers have ownership of the project, they become advocates for it versus solely adjudicators of its success or failure (Peck, 2021). In the agile methodology, developers can develop a base version of the end product for further improvement (Lobo Marques et al., 2023). By developing a base version, developers can provide their

customers with a foundation from which the customer can adjust as their requirements change. Developers can use this iterative approach to show tangible progress to the customer while allowing them to adjust requirements as their environments change. Lastly, the Execution phase of agile methodology is more user-focused (Lobo Marques et al., 2023).

Compared to the waterfall methodology, which puts the execution of the project almost solely in the purview of the developers, the agile methodology takes a collaborative approach to the Execution phase. With the continued involvement of customers throughout the Development process, customers can impact and tailor the execution of the Development phase. As a result, customers receive end products that are more conducive to their current and future needs than those they had at the beginning of the project. However, like the waterfall methodology, Lobo Marques et al. (2023) also listed the disadvantages of the agile methodology.

Lobo Marques et al. (2023) wrote of several significant disadvantages of the agile methodology. In the agile methodology, some customers may want to avoid involvement in the process throughout its development (Lobo Marques et al., 2023). Instead, some customers prefer to develop the requirements at the beginning of the process and only engage directly at critical milestones such as delivery and execution. In the agile methodology, the customer may need more time and expertise for the frequent involvement of the customer throughout the development process (Lobo Marques et al., 2023). Many customers may have conflicting time interests or need more technical expertise to provide developers with what they need throughout development. In the agile

methodology, project leaders/managers need to allocate their team members solely to the project (Lobo Marques et al., 2023). As a result, project leaders/managers lose the flexibility to surge human resource assets and expertise to other projects within the organization, as needed. In the agile methodology, due to the frequent reprioritization of tasks throughout the development process, developers may need help to complete some items for delivery within the desired timeframe (Lobo Marques et al., 2023). As a result, developers may need more time, money, and other resources than the developers and customers originally agreed upon to complete the project. Lastly, due to the iterative nature of the agile methodology, customers may have to redefine the project's scope as developers create the end product (Lobo Marques et al., 2023). As a result, developers may be unable to deliver a product to the quality that the developers and customers originally agreed to. Customers and developers routinely encounter this disadvantage when using the agile methodology for large-scale and integration-intensive projects (Lobo Marques et al., 2023). Like the software development industry, construction favors one of these methodologies over the other due to its nature. However, like the software industry, a hybrid approach of the waterfall and agile methodologies can accentuate the strengths of the preferred methodology while mitigating its weaknesses.

Due to the nature of the overall task, the construction industry follows a sequential framework. As a result, construction leaders/managers traditionally use sequential project management methodologies such as waterfall (Lobo Marques et al., 2023). Additionally, construction leaders/managers refrain from using the agile methodology because of the high costs of changing requirements throughout the

construction process (Lobo Marques et al., 2023). However, Lobo Marques et al. (2023) asserted that construction leaders/managers could benefit from incorporating certain aspects of the agile methodology within the traditional, waterfall-like construction process.

Specifically, construction leaders/managers could incorporate agile principles such as prioritizing customer satisfaction and continuous daily collaboration with customers, face-to-face communications with customers, self-organizing teams, and regular meetings to discuss team improvement (Olshevska et al., 2023). Lobo Marques et al. (2023) hypothesized and concluded that construction leaders/managers could incorporate agile principles in their development process while retaining the fundamental methodology of waterfall. By doing so, construction leaders/managers increased their level of success and their customers' satisfaction (Lobo Marques et al., 2023). Even though they were not formally aware of the agile principles, many construction leaders/managers discovered they had incorporated agile principles in the traditional waterfall methodology out of necessity (Lobo Marques et al., 2023). Lobo Marques et al. concluded that the hybrid approach gave customers and developers "the best of both worlds" regarding the respective waterfall and agile methodologies (p. 37). Members of the Innovation and Emerging Services Department used this "best of both worlds" concept when they wrote their paper on how to build and deploy Artificial Intelligence (AI) and Machine Learning (ML) solutions using a hybrid approach of the waterfall and agile methodologies.

Many organizations have formally researched how a hybrid approach of the

waterfall and agile methodologies can improve their internal processes. The Innovation and Emerging Services Department is a Global Information Systems group of over 50 professionals that helps the ABB business improve its internal processes using advanced IT technologies such as AI and ML solutions (Dagnino et al., 2022). In their paper, Dagnino et al. (2022) explored how ABB businesses could use a hybrid approach of the waterfall and agile methodologies to build and deploy AI and ML solutions more efficiently. The authors divided this hybrid approach into three phases: (a) definition, (b) development, and (c) deployment. Each phase has a solid correlation to either the waterfall or agile methodology.

In the Definition phase, developers follow a waterfall methodology. In the Definition phase, customers and developers define a "blueprint" of the solution, identify roles, develop requirements, and create initial effort and cost estimates (Dagnino et al., 2022). In the Definition phase, customers and developers establish the foundation for the project. The second phase is the Development phase.

The Development phase has a slightly different focus than the definition phase. In the Development phase, customers and developers need a certain amount of flexibility to input revisions and add new requirements to the project for a better final product (Dagnino et al., 2022). To do this, developers move from the waterfall methodology and incorporate more agile principles (Dagnino et al., 2022). The third and final phase is the Deployment phase.

In the Deployment phase, developers give the solution to customers for use. There are two stages in the Deployment phase; the first stage is Hypercare. In Hypercare,

developers provide extra support to users to help them understand how the solution works (Dagnino et al., 2022). At this stage, developers receive the users' feedback on the updates and improvements to make the solution optimal in the users' environment (Dagnino et al., 2022). The second stage is Maintenance and Support. At this stage, developers ensure that users can report any issues arising from fixes and revisions to the software and address any of the users' concerns so that users can have the best experience with the new solution (Dagnino et al., 2022). To best facilitate the back and forth between developers and the users, developers incorporate more agile principles such as "Working software is the primary measure of progress" and "Most efficient and effective method of conveying information to and within a development team is face-to-face conversation" (Isaacs, 2022, p. 8).

Summary and Conclusions

Technology projects have an astonishing failure rate. Despite extraordinary efforts and allocation of resources by leadership, only 23% of large-scale technology projects were successful, meaning developers delivered their respective projects on time, on budget, and to a satisfactory standard (The Standish Group, 2018). The two most prevalent software development methodologies–waterfall and agile–have inherent weaknesses detrimental to software development (Gaborova et al., 2021). Research supports the idea that neither the waterfall nor agile methodologies fit all software development projects, which justifies the growing exploration of hybrid methodologies between waterfall and agile (Bianchi et al., 2021). Therefore, the social problem is that the inherent shortcomings of the two most prevalent software development

methodologies–waterfall and agile–lead to many software development project failures. (Gaborova et al., 2021).

This qualitative study explored the consequences in government software development fostered by a hybrid waterfall and agile methodologies approach. To discover what consequences a hybrid waterfall and agile methodologies approach can foster. After describing CMMI, SLDC, waterfall, and agile, I listed and compared the advantages and disadvantages of the waterfall and agile methodologies. Then, I cited examples of organizations using the hybrid approach of the waterfall and agile methodologies in software and other industries.

Chapter 3: Research Method

I attempted to research the social problem of facing software development. Specifically, the inherent shortcomings of the two most prevalent software development methodologies–waterfall and agile–lead to a large percentage of software development project failures (Gaborova et al., 2021). The purpose of this qualitative study was to explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology. I conducted a case study to explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology.

I contemplated multiple research approaches for this study–qualitative, quantitative, and mixed. Researchers use the qualitative method when they want to understand the meaning people place on things; researchers use the quantitative method when they want to quantify the change of things based on certain factors (Kandel, 2020). I aimed to understand the meaning people attribute to the hybrid approach of the combining the waterfall and agile methodologies, which lends itself to a qualitative research approach. Additionally, I wanted to generalize his findings to other group settings and populations, which lends itself towards the qualitative research method. Therefore, I chose the qualitative research approach for this study. After choosing the research method, I contemplated multiple research designs for his study.

I studied the advantages and disadvantages of multiple qualitative research designs to choose the best one for his study. Researchers use research design to provide

specific direction for their research (Kandel, 2020). After researching multiple qualitative research designs, I chose a case study because it explores the answers of "what", "why", and "how" questions more completely than other qualitative research designs (Alam, 2021). Once I chose the research design, I moved to the specifics of the research.

Research Design and Rationale

This case study examined the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies. Previous researchers have concluded that they will reach saturation after eight to ten interviews (Breen, 2007). Therefore, I used a study population of eight former and present members of the studied government software development organization and possibly the designated customer representative for the weapons system. I selected individuals through purposive sampling because purposive sampling is the best sampling method to match the study sample to the purpose of the research, improve the rigor of the study, and increase the trustworthiness of the data (Campbell et al., 2020). By conducting a case study on how this organization implemented a hybrid waterfall and agile methodology, future government software development entities may foster consequences in government software development and directly contribute to positive social change through a more efficient and effective government for its citizens. However, software developers can face specific challenges with implementing aspects of the agile software development methodology in government entities because the agile software methodology is best suited for organizations that are not hierarchical and bureaucratic–which most government entities are (Fontana & Marczak, 2020).

On November 7th, 2024, the IRB gave me conditional approval to collect data for this dissertation. The IRB only gave conditional approval because I intended to interview indirect employees (i.e., government contractors) of the United States Army for data collection. As a result, I also needed to get approval from the AHRPO to interview government contractors.

The AHRPO has multiple responsibilities and played a critical role in this study's process. One of the responsibilities of the AHRPO is to oversee research institutions with Army-issued DoD Assurances and Human Research Protections Programs (HRPP) (Responsibilities, 2025). Since I planned to use government contractors as the interviewees for this study, I needed to gain the approval of the AHRPO. Once the AHRPO gave its approval, I returned to the IRB for unconditional approval for this study.

After receiving the conditional approval from the IRB, I submitted much of the paperwork provided to the Walden University IRB and some paperwork unique to the AHRPO approval process to the AHRPO. The AHRPO asked a few questions about how I would invite potential interviewees to the study. This question caused a slight change in how I invited potential interviewees to the study.

I always intended to send my invitations for the study to nongovernment email addresses. Instead, I planned to use the potential invitee pool's contractor email addresses. I believed that using the contractor email addresses would provide enough separation from the project so that the potential interviewees would not feel pressured to participate in the study. However, the AHRPO disagreed.

The AHRPO believed that using any kind of official email address would give the

impression that participation in the study was not voluntary. After some research with the current team members of the studied government software development entity, I discovered that the organization had a social distribution list. Once I discovered what this list was, he believed that using this list to solicit potential interviewees would satisfy the requirement by the AHRPO.

The social distribution list is a distribution list of email addresses that use non-government, non-contractor email addresses of former and current team members of the studied government software development entity. Since none of the addresses on the social distribution list were governmental nor contractor emails, I thought using this distribution list would satisfy the requirement for the AHRPO. As a result, I revised the paperwork unique to the AHRPO to reflect the change in distribution lists used to solicit potential interviewees.

After some back and forth, clarification of data collection, and revision of the submitted documentation, I received AHRPO approval to collect data on December 2nd, 2024. On December 18, 2024, I received unconditional approval from the IRB to collect data for this dissertation. After gaining unconditional approval from the Walden University IRB, I began the process for data collection through face-to-face interviews.

With written permission from the current leadership of the studied government software development entity, the research used a social distribution list maintained by the current team members of the studied government software development entity. The distribution list consists of both current and former team members. Since the list is for social purposes, current and former team members can provide their “non-government”,

"non-government contractor" email addresses to receive updates, social events, and other non-program-related information on the program. I used this list to solicit potential interviewees so that there would not be any perception of professional impact.

The maintainer of the social list sent a recruitment email to current and former team members. The recruitment letter consisted of my name, the topic of my dissertation, the inclusion of interviews as a means of data collection for my dissertation, and an invitation to participate in the study as an interviewee. The recruitment email ended with my contact information for my committee chair and the IRB. When a potential interviewee responded to the recruitment email, I sent each respondent a consent email.

The consent email had several components. First, the consent email invited the respondent to participate in the study as an interviewee and restated the purpose of the study. Next, the consent form explained "informed consent" and outlined the procedure for the face-to-face interviews to include duration of the interviews, audio recording of the interviews, and the opportunity for each interviewee to member check the transcription of their interview for accuracy. Lastly, the consent form listed the questions I asked during the face-to-face interview and the voluntary nature of participating in the study. Once I received consent from an interviewee, I scheduled and conducted the interviews.

In total, I interviewed eight current and former team members of the studied government software development entity. I conducted the interviews away from the interviewees' work location. I decided to conduct interviews away from the interviewees' work area to facilitate comfort and open dialogue between himself and the interviewee by

promoting anonymity of each interviewee. Each interview lasted between 30 and 60 minutes. I used an audio recording device to record each interview and download each interview audio recording to a computer. I did not use any visual recording devices during the interviews. Additionally, before I did any formal analysis of the interview, he allowed each interviewee to member check their respective interview transcriptions for accuracy. I did not encounter any unusual circumstances while scheduling, conducting, recording, transcribing, or giving the opportunity for interviewees to member check their respective interviews. After interviewing a combination of eight former and current team members of the studied government software development entity, I wanted to evaluate if he had reached data saturation for this part of his study.

In qualitative research, data saturation is an important concept to understand and accomplish. Data saturation ensures credibility and quality in qualitative research (Rahimi, 2024). While there are defined definitions of data saturation, I found it difficult to reach a numerical standard for when a qualitative study has reached data saturation. As a result, I researched a more functional definition of the concept of data saturation.

After extensive research, I found a functional definition of data saturation. According to Squire et al. (2024), data saturation is reached when the collection of new data does not produce new information. After interviewing four former and four current team members of the studied government software development entity, I recognized that the information provided by the interviewees was repetitive. As a result, I determined that I had reached data saturation from the interviews. While most of the concepts stated during the interviews did not come as a surprise, one concept was very surprising to me.

When I developed the interview questions for the interviews, he assumed that the studied government software development entity continued to use a hybrid approach with an even mixture of waterfall and agile methodologies. However, after interviewing the four current team members of the studied government development entity, I realized that was not a valid assumption.

Once I had each of the interviewee's member check the transcriptions of their interviews, I looked for common themes within the transcripts for coding of the interview transcripts. After analyzing the interview transcripts, I realized that the eight transcripts yielded consistent common themes. Furthermore, after analyzing the eight, face-to-face interviews, I realized that I had reached data saturation because there were no new information, codes, nor themes from the interviews (see Hossain et al., 2024).

I did not encounter any unusual circumstances while collecting nor analyzing the data. The largest impediment to the data collection and analysis of the interviews was scheduling the interviews. Since interviewees and I were working during the time of the interviews, it was difficult to find a time and place to conduct the interviews that was amenable to both parties. On multiple occasions, I met interviewees at coffee shops in the morning while the respective interviewees were on their way to work. In other instances, I met interviewees at restaurants after work at locations between the interviewees' work and home locations. Overall, I was able to effectively conduct the interviews. However, the execution of those interviews was more difficult than I had originally anticipated. I found scheduling and executing the interviews much harder than he had anticipated due to finding times and places that worked for both he and the interviewees. In addition to

the face-to-face interviews, I analyzed the minutes from the first 52 IPT meetings between the studied government software development entity and the customer using the MAXQDA software to answer the research question for this study. The difficulty with coding the 52 IPT meetings using the MAXQDA software came from learning how to use the software. I had to learn how to import the IPT meeting transcripts into the MAXQDA software. Additionally, I had to learn how to use the functions within the MAXQDA software to upload the words associated with waterfall and agile into a dictionary and perform the word frequency analysis of the imported IPT meetings using that dictionary. The spreadsheet generated by the MAXQDA software provided "Word Frequency" and "% (Percentage)". The "Word Frequency" column depicts the number of times the word in that row appeared in the 52 IPT meeting transcripts; the "% (Percentage)" column depicts the percentage of the total verbiage the word in that row appears.

With written permission from the current leadership of the studied government software development entity, I was able to gather and analyze the first 52 IPT meetings between the studied government software development entity and the customer. As written earlier, I used the MAXQDA software to analyze the meeting notes from these IPT meetings. MAXQDA has many tools that qualitative researchers can use for their analysis (Rizzo et al., 2024). For this study, I used MAXQDA for coding of the verbiage transcribed during the IPT meetings, specifically word frequency. To determine which words the MAXQDA software should identify for waterfall and agile methodologies, I identified words in the software industry for each methodology.

As I wrote in this study, waterfall and agile methodologies accomplish software development in different ways. As a result, waterfall and agile have different verbiage associated with each methodology. The article by Pargaonkar (2023) listed the advantages and disadvantages for each methodology. In turn, the article lists certain verbiage associated with each methodology. I used that verbiage to create a dictionary in MAXQDA. The following are the terms in the Pargaonkar article that I used to depict the waterfall methodology: documentation, limited flexibility, linear, little revisions, planned, rigidity, predictive, structure, easy management, well-defined. In turn, the following are the terms in the article that I used to depict the agile methodology: flexibility, collaboration, customer needs, adapt, iterative, change, and uncertainties. Using that dictionary, I used the "word frequency" feature in MAXQDA to create a spreadsheet that listed the words in the dictionary in the far-left column. In the top row, the MAXQDA software depicted the percentage and frequency of a word in the dictionary that appeared in the 52 IPT meeting transcripts, along with the date of each IPT meeting. The subsequent rows and cells provide additional information.

The subsequent rows of cells depict the percentage and frequency each word in the dictionary appears in all 52 IPT meeting transcripts as well as how often each word appears in each of the 52 IPT meeting transcripts. The spreadsheet also color-codes each row (word in the dictionary) as either pertaining to waterfall (yellow) or agile (green). This visible representation of the data yields quick and detailed analysis of the data. Taking a quick look at the spreadsheet in Appendix B, an observer can compare how often terms associated with each methodology appear in the verbiage of the IPT

meetings.

By taking a closer look at the percentages and total number of times each word appeared in the transcriptions, an observer can get a more in-depth understanding of how much the participants in the IPT meetings referred to the waterfall and agile methodologies while discussing the overall objectives, goals, and tasks provided by the customer and accomplished by the studied government software development entity. In the Data Analysis section of this chapter, I wrote on this analysis. However, like the face-to-face interviews, I did not encounter any unusual circumstances while collecting nor analyzing the data. Again, the largest impediment to the data collection and analysis was learning how to use the MAXQDA software. Specifically, I had to learn how to import the transcripts of the 52 IPT meetings and use the functions within the MAXQDA software to upload the words associated with waterfall and agile into a dictionary and perform the word frequency analysis of the imported IPT meeting transcripts using that dictionary. In addition to the face-to-face interviews and analysis of the first 52 IPT meetings between the studied government software development entity, I analyzed the organizational charts of the studied government software development entity at the beginning of each of the first four quarters.

I wanted to know how using the combined methodology of waterfall and agile impacted the turnover rate of the studied government software development entity. To that end, I retrieved and analyzed organizational charts of the studied government software entity at the beginning of each of the first four quarters of the entity's existence. By comparing those who were in the same positions at the beginning of each quarter, I

calculated a percentage of turnover for each quarter. I used the following equation to calculate the percentage of turnover for each quarter: number of positions that have a new person in that position from the previous quarter divided by (/) the total number of positions (30) times (x) 100 (Li et al., 2024). Using this percentage, the research will be able to quantify how much turnover occurred within the studied government software development entity over the first year of its inception.

Using that quantifiable percentage of turnover, I established if the studied government software development entity has a relatively high turnover rate. Research suggests that 10% annual turnover tends to be ideal (Nababan, et al., 2024). In addition to comparing the turnover rate to the ideal percentage, I compared the previous turnover rate of the studied government software development entity with its current turnover rate by doing a similar comparison of the last four quarters of the studied government software development entity. Like the face-to-face interviews and collection of data from the first 52 IPT meetings minutes, I did not encounter any unusual circumstances while collecting nor analyzing the data provided for the charts.

Role of the Researcher

In qualitative research, the researcher has multiple roles. Researchers in qualitative research must use purposeful methodology and documented data to increase rigor and transparency, which reduces potential bias (Mackieson et al., 2019). Researchers must be deliberate with their methodology and use documented data to increase objectivity and reduce the opportunity for bias in their research. When researchers adhere to these criteria, it gives credibility to the research. Another role for

the researcher is their role with the study participants.

In qualitative research, the participants play a vital role. Consequently, researchers should move from making the participants the objects of the research to making the participants the representatives of the research (Moriña, 2020). By doing so, researchers can provide the participants' perspectives in their research and reduce their personal biases. Along with controlling bias, researchers should approach research with the correct mindset.

Before articulating how researchers' mindsets affect their research, defining the term "mindset" for this dissertation is essential. (Aghajani et al., 2023) define "mindset" as a person's definitions, values, and principles. These definitions, values, and principles provide the framework for how people see the environment around them. In terms of research, how researchers see the environment around them greatly influences the research they conduct. Consequently, I moved to how researchers' mindsets affect their research and how he used that knowledge for the sake of his dissertation.

The literature reads of two types of mindsets: (a) fixed and (b) growth. (Zigler et al., 2021) wrote that those with a fixed mindset believe their capabilities are "fixed" and no one can change their thinking meaningfully. In contrast, Zigler et al. (2021) wrote that those with a growth mindset believe that challenges and failure serve as a means to increase their understanding and capabilities. In terms of research, researchers with a fixed mindset may be more likely to have their biases negatively affect their research, while researchers with a growth mindset may be more likely to approach the research as a means of learning. As a result, researchers with a growth mindset are less likely to let

their biases negatively affect their research. As a result, I maintained a growth mindset throughout the research to minimize personal biases.

Another role of researchers was to establish trust with their research participants. Nunan (2020) wrote of two aspects of trust: (a) trust with study participants to facilitate data collection and (b) trust with the search's consumers. While many researchers believe that informed consent is the "central feature" in establishing trust with research participants, research participants can also see informed consent as a means of increased surveillance, which can reduce trust (Nunan, 2020, p. 393). To reduce this sentiment of surveillance, researchers can make their research participants believe that they will maintain the participants' anonymity and use the data for its stated and intended research. Researcher can foster this belief by building rapport with their research participants.

The term "rapport" is another term many people use, but it tends not to have a standard meaning. For the sake of this dissertation, "rapport" is developing and maintaining a working relationship by managing motivations and welfare while understanding the purpose of the relationship (Nunan et al., 2022). Researchers can build rapport and trust by agreeing with their research participants towards a common goal, giving them a voice in the research process, and maintaining an understanding of what research participants feel throughout the research process (Pashak & Heron, 2022). Researchers define their research process through their methodology.

Methodology

Participant Selection Logic

I put much thought into participant selection. I wanted to make sense of the

observed phenomenon with selected individuals, which is the focus of qualitative research (see Johnson et al., 2020). To that end, I used purposive sampling to optimize data sources for answering the research question (see Johnson et al., 2020). Specifically, I selected an equal number of interviewees from lower levels of the organization, the leaders of the organization, and members of the customer that the organization supported with the development of the software. By using this purposive sampling, I selected participants that optimize data sources for answering the research question. I used this objective to create the research design of the case study.

Routinely, the studied government software development organization maintains a staff of around 40 people. I conducted face-to-face interviews with a desired length of 30-60 minutes. The case study participants were former and current team members of the studied government software development organization. I selected the case study participants based on their positions within the studied government software development organization. To that end, participants identified and selected themselves by responding to an email sent for me (see Herodotou et al., 2020). Via the recruitment email, I requested former and current team members of the studied government software development organization to volunteer for the case study. Once I had a viable pool of participants, he used purposeful sampling to select the final eight participants in the case study.

I made a deliberate decision to use purposive sampling. When researchers use a trustworthy selection of participants, they receive trustworthy data (Gabarre & Gabarre, 2020). I used purposeful sampling to receive trustworthy data in the case study. I also

made a deliberate decision on his choice of research design.

The research question determined the use of a case study for the research design. Case studies are research designs of inquiry, and the research question inquires to the "what", "why", and "how" to analyze a phenomenon in-depth (Asenahabi, 2019). For this dissertation, I wanted to analyze in-depth how a government software development organization used a hybrid waterfall and agile methodologies approach to create software for its customers. However, I did not decide on using a case study solely for in-depth analysis.

I used a case study for the research design for other strengths. I used data triangulation, a significant strength of case studies, to ensure that the participants' descriptions are accurate and that I precisely captured their descriptions (see White-Lewis, 2020). Along with the in-depth analysis, I wanted to ensure he accurately captured the participants' perspective on the government software development organization's use of the hybrid waterfall and agile approach. Then, I focused on collecting the data for the research.

Instrumentation

I conducted 30-60-minute, face-to-face interviews as the primary means for collecting data and an interview guide as the instrument for the interviews. I considered using online applications and platforms for his interviews. However, researchers have written that online approaches produce lower levels of relationship satisfaction between interviewee and interviewer and consensus development (Davies et al., 2020). Interviewers must develop a rapport with the interviewees to get the most authentic

information. Researchers perceive in-person interviews as the “gold standard” because of the rapport they build between the interviewer and interviewee (Irani, 2019, p. 3). Other reasons support my choice of face-to-face interviews.

The accuracy of data is crucial for the success of this dissertation. Studies showed that videoconference interviews received lower performance ratings and more negative perceptions than face-to-face interviews (Basch et al., 2021). Face-to-face interviews increase the trustworthiness of the interview data. Additionally, face-to-face interviews allowed the interviewer to gain more information than solely what the interviewees said.

Interviewers can gain additional information from their interviewees beyond their verbal answers. Interviewers have difficulty reading visual cues from the interviewee when interviewers use online approaches (Thunberg, 2022). I wanted to give this study the best opportunity to capture and accurately reflect the interviewees’ perspectives. As a result, I decided to use face-to-face interviews versus online approaches to my interviews. However, I wanted to use some facets of the online approach.

While conducting the face-to-face interviews, I wanted to leverage laptops and cameras to record the interviews via the Zoom application. Zoom would allow me to record the interviews and securely store the data from the interviews (see Gray et al., 2020). I would have been able to rewatch the interviews and possibly glean additional information he may have missed during the initial interviews. However, I may have had to address legal and ethical issues with visually recording the interviews.

In today’s environment, society is leery of recording other people. Many states have formal laws regarding recording people without their knowledge and/or consent

(Self, 2021). Additionally, there are ethical challenges to recording other people, even for research purposes (Self, 2021). I had the Walden University IRB review the verbiage, processes, and procedures for disseminating, recording, storing, and deleting the data and information I derives during the study. I believed the IRB would be an excellent asset for navigating any legal or ethical issues with recording, storing, and deleting the interviews for this study. I also leveraged the IRB with the verbiage in its emails to potential and eventual participants in the study. While agreeing with the other aspects of the study, the IRB denied my request to visually record the interviews. Therefore, I only audio-recorded the face-to-face interviews.

I wanted to ensure the consistency and uniformity of the general information he distributed to the potential interviewers. Therefore, I used an email with multiple potential interviewees in the "To" line so that potential interviewees received the same information (see Ingley et al., 2020). I used email to send and receive consent forms from the study participants, which may increase their confidence in the legitimacy and professionalism of the study (see Ingley et al., 2020). Another attempt to increase the credibility of the study is data triangulation.

I used a combination of face-to-face interviews and document analysis. Researchers accomplish data triangulation by incorporating multiple data collection methods such as face-to-face interviews and document analysis (Natow, 2020). I wrote about face-to-face interviews. I went into further detail on document analysis.

I used document analysis and face-to-face interviews in this case study. Researchers can use documents relevant to the research questions to complement face-to-

face interviews (Pfister & Pozas, 2023). In the early stages of the studied government software development organization taking over the work, the organization conducted weekly meetings with a customer representative. The participants in these weekly meetings discussed observations during the previous week and possible priority changes. The studied government software development organization documented critical aspects of these weekly meetings and, in some cases, used this documentation as the foundation for change requests to the software. I used document analysis in concert with face-to-face interviews for data collection in this case study.

I used document analysis as a tertiary data collection form. Researchers can use document analysis to provide insight into the "what, how, and why" of studies (Kayesa et al., 2021). The technical documents for the weapons system named the weapons system, which the current leader of the studied government software development organization did not allow for this dissertation. However, I did use open-source references that do not name the weapons system for his document analysis. Additionally, I used DoD open-source references to provide the best data source for DoD regulations and guidance regarding their weapons systems.

Procedures for Recruitment, Participation, and Data Collection

I began the recruitment process by having a current team member of the studied government software development entity email close to twenty former and current team members of the studied government software development organization. Since I was a former member of this organization, I had knowledge of more than twenty former and current team members of the studied government software development organization to

solicit for this study. Researchers should be aware of factors that facilitate the successful recruitment of participants, such as making themselves personally known to potential participants, identifying potential benefits of participating, and communicating possible positive outcomes from their participation (Stuber et al., 2020).

As a former team member of this entity and a current team member of another program with shared email and site access, I received permission to email current and former team members and site access to conduct interviews. I excluded all other participants because he wanted to understand the team members' experiences of the setting (see Patton, 2015). Other exclusionary criteria were as follows:

1. I only interviewed former and current team members, no strangers.
2. I did not seek out protected populations such as children, prisoners, residents of any facility, or mentally/emotionally disabled individuals for this project.
3. I stored project data in electronic format (e.g., Word, Excel, MAXQDA, etc.) for the duration of the study and disposed of the data at the end of the study.
4. I de-identified the data as soon as is realistically possible to minimize the risk of inappropriate disclosure of personal information.
5. I did not give payments, compensation, reimbursement, free services, extra credit, or other gifts to the project participants, which ensures voluntary participation.

I received verbal permission from the current leader of the studied government software development organization to conduct interviews with the study participants when he began data collection. I issued a written inquiry defining the study's extent and purpose. I

contacted study participants once he had completed the process. I used the screening process to ensure that potential participants met the recruitment criteria and positively influenced the participants' interest and willingness to participate in the study. I interviewed eight former and current team members face-to-face to reach data saturation. Once I received positive responses to the initial interview request, I emailed the participants the consent form.

Once he received consent from eight possible interviewees, I coordinated dates and times to conduct the interviews. I conducted the interviews with the interview questions in Appendix A. I asked Questions #1 and #2 to get an idea of participants' perceptions of the strengths and limitations of the CMMI software process improvement model (Question #1), waterfall (Question #2), and agile (Question #3) methods. These questions laid the foundation for how a hybrid approach accentuates their respective strengths and mitigates weaknesses.

During the interview, I audio-recorded the interview and took notes on critical responses, non-verbal cues, and feedback. Additionally, I kept in mind that he was asking people to share their stories and time (see Petersén & Carlsson, 2021). Consequently, I did everything possible to keep the interviews within the 30–60-minute timeframe depicted by the recruitment and consent emails. Along with the interviews, I collected data for this study through data analysis. I used these secondary and tertiary means of data collection to provide data triangulation for this study, which will provide trustworthiness.

In the earliest time of the studied government software development organization

taking over the responsibility for the training software, the organization's leader coordinated a weekly meeting with the customer representative. During this meeting, the leader had a scribe to take verbatim transcriptions of the meetings. Researchers can use documented document analysis as another factor for data triangulation (Sana et al., 2021). I used the documented notes from the weekly meetings for document analysis. This data collection was the secondary means of data collection for this study.

I used data analysis of the quarterly organization charts as the tertiary means for data collection for this study. Researchers can use data analysis as a viable means of data collection for data triangulation (Collins & O'Riordan, 2022). Regarding turnover in the studied government software development organization, I highlighted how the hybrid approach of the waterfall and agile methodologies can impact turnover in the organization. Researchers can use this information to study future research on the waterfall and agile hybrid approach.

Data Analysis Plan

I used face-to-face interviews as the primary means of data collection for this dissertation. Researchers use face-to-face interviews because they provide the best personal connection between themselves and the research participants (Jain, 2021). In terms of the research question, I explored what consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodologies. I leveraged my connections with the interviewees to gain greater insight into how the hybrid approach of the waterfall and agile methodologies enabled the studied government software development organization to satisfy its customers' requirements. After

conducting the interviews, I transcribed them to enable me to analyze the data through coding.

After gaining permission from the participants to audio record the interviews, I audio recorded each interview. I received written consent from each interviewee before conducting and audio recording the interview (see Kang & Hwang, 2021). I received written consent prior to conducting each interview to address the legal and ethical issues I raised earlier in this dissertation. Additionally, I audio recorded the interviews to facilitate verbatim transcriptions.

I transcribed the recordings using the Microsoft (MS) Word application. Researchers have used MS Word to transcribe recordings of interviews even though this process can be time-consuming (Adeoye-Olatunde & Olenik, 2021). However, I used this method to transcribe the recording to force him to become intimate with the verbatim data in the interviews. Additionally, I seamlessly used the MS Word transcriptions for coding. Lastly, I forwarded the respective transcriptions to each interviewee for their review before analysis.

I used the verbatim member checked transcripts of the face-to-face interviews to code the data. I analyzed the data from the face-to-face interviews manually. Manually analyzing the transcripts was time-consuming. However, this process was more informative because the MAXQDA software did not provide consistencies throughout the interviews that would lead to data saturation for the study. While I did not use the MAXQDA software for the face-to-face interview transcripts, he did use the MAXQDA software for the analysis of the first 52 IPT meetings because the software did provide

valuable data analysis for this information.

While several coding software applications are available, I used the MAXQDA software for this task in this study. MAXQDA has built-in statistical tools for simple and correlational analysis (Guetterman & James, 2023). Since the manual coding of the verbatim transcriptions of the IPT meeting using MS Word may be time-consuming, I used the MAXQDA software because it was not as time-consuming.

I used MAXQDA to code the verbatim transcription of the IPT meetings. I used the MAXQDA software to code the verbatim transcriptions of IPT meetings because using software to code verbatim transcripts tends to provide more information than manually coding scripts (see Rutakumwa et al., 2020). I planned to use any discrepant cases to strengthen his research.

Government software development is a complex landscape. When researching complex landscapes, researchers provide a more realistic account of the studied phenomenon by acknowledging theme variations (Rose & Johnson, 2020). To that end, I elaborated on any discrepant cases during his research to increase its validity. I used this perspective in all facets of the data collection.

As written earlier in this dissertation, the leader of the studied government software development organization coordinated for a customer representative to attend weekly meetings where key participants discussed observations, and an assigned person documented those observations. Researchers can use document analysis as a viable means of data collection for data triangulation (Vu, 2023). I used these minutes from these weekly IPT meetings as document analysis of the studied government software

development organization and how the organization used the hybrid approach of the waterfall and agile methodologies. The scribe for the weekly meetings typed verbatim comments from the meetings. As a result, I input the minutes from the meeting directly into the MAXQDA software for coding. In conjunction with the face-to-face interviews and analysis, I used a third means of data collection for data triangulation.

I used data analysis of organizational charts as a third means of data collection. I analyzed the quarterly turnover rate of the studied government software development organization and the levels of the organization where the turnover occurred. High turnover can lead to adverse effects on an organization, such as knowledge drain, reduction in productivity, and increases in defects (see Mirsaeedi & Rigby, 2020). I described the effects of using the hybrid approach concerning turnover rate.

Issues of Trustworthiness

Researchers have a challenge of trustworthiness in qualitative research. Researchers must use transparency to achieve trustworthiness in qualitative research (Adler, 2022). Researchers achieve transparency in qualitative research by clearly articulating their research techniques and the theoretical basis of the research (Adler, 2020). I established the trustworthiness of his qualitative research by clearly laying out his research techniques and using DOI as a foundation for his research. Additionally, I used multiple criteria for establishing the soundness of the research.

Qualitative researchers have developed standards to provide rigor to their research. I used the four established trustworthiness principles to provide rigor for my research: credibility, transferability, dependability, and confirmability (see Nassaji,

2020). Qualitative researchers use these principles to provide rigor for quantitative research (Nassaji, 2020). I used credibility as an integral part of establishing the trustworthiness of my research.

Credibility

Researchers can use various strategies, such as triangulation, prolonged contact, member checks, saturation, etc., to establish credibility. I used triangulation and saturation in his research. I used established means of triangulation, such as face-to-face interviews and data analysis (see Santos et al., 2020). I used eight interviewees to reach data saturation because no new data, themes, and coding occurred after this number of interviews (see Aguboshim, 2021). Informally, I used prolonged contact in my research. I was a member of the studied government software development organization. As a result, I understood the culture and setting of the studied government software development organization and has a good relationship with many interviewees (see Sanip, 2020). I used these means to establish credibility for the research.

Transferability

Another criterion of trustworthiness in qualitative research is transferability. Researchers define "transferability" as the extent to which the researcher can come to similar results in a different setting (Munthe-Kaas et al., 2020). I wanted readers to take this case study and transfer the findings. Specifically, I want other government software development entities to use a hybrid of the waterfall and agile methodologies to address failures.

Researchers have argued that transferability requires thick description, which is

necessary to yield the necessary data and transfer the original finding to other contexts or individuals (Lavee & Itzchakov, 2023). I provided a detailed account of the studied government software development organization and a detailed account of how he gathered and documented the data for his research. Additionally, I gathered data from team members of multiple levels within the studied government software development organization.

Dependability

Dependability is another means to establish trustworthiness in qualitative research. Dependability is the equivalent term for reliability in quantitative research and depicts qualitative researchers' responsibility to document their procedures for generating and analyzing the results (Riazi et al., 2023). I increased the trustworthiness of his research by incorporating dependability in his dissertation. I used documentation and coding to support the dependability of his research.

I used multiple means to increase the dependability of his research. I provided detailed documentation and explanations of coding and data analysis so that readers of the research can arrive at the same conclusions (see Riazi et al., 2023). I made the conclusions of the research dependable and increased the trustworthiness of the research. I used confirmability to increase the trustworthiness of the research.

Confirmability

Confirmability is the last factor I used to increase the trustworthiness of his research. Researchers use confirmability to show that the research and its findings come from the data, not the biases of the researchers, and provide objectivity to the research

(Enworo, 2023). All researchers have biases that may influence their findings and research. Since I was a former member of the studied government software development organization, the opportunity for bias is increased. Consequently, I traced the data from the source to the research findings to ensure objectivity. While objectivity confirms research and gives it trustworthiness, there is a place for the researcher's subjectivity in qualitative research.

Researchers can incorporate their subjectivity in their research through reflexivity. Reflexivity is a set of practices researchers can use to critique, appraise, and evaluate how their subjectivity influences the research process (Olmos-Vega et al., 2023). In this dissertation, I used my familiarity with the studied government software development organization and its team members to gather in-depth data that provided more detailed research than from someone outside the organization. However, I was careful to ensure my familiarity with the studied government software development organization's team members did not provide ethical challenges by adhering to established ethical procedures.

Ethical Procedures

I began collecting data only after he had secured approval from the Walden University IRB. I gained documented permission from the current leader of the studied government software development organization to conduct the research and interview current team members. While gaining permission from the current leader of the studied government software development organization, the leader directed me to inform the interviewees that they cannot charge the interview time to the government or the contract on which they are working. Additionally, the current leader directed me not to use any

data from the studied government software development organization nor refer to the organization by name in his dissertation.

Interviewees reviewed the research's purpose, had assurances of the confidentiality of their identities, and signed an informed consent form before participating in their face-to-face interviews. The informed consent form read of the purpose of the research, the intent to audio record the interviews, the voluntary nature of their participation, and no personal benefits from participation. Lastly, I secured the data in a locked safe in my office for 5 years and will destroy the data after that period has passed.

I coordinated the date and time of the interviews with the interviewee so that I conducted the interviews on dates and times that were the most convenient for the interviewee. I conducted the interviews away from the location of the studied government software development organization. I conducted interviews away from the studied government software development organization, so the interviewees felt comfortable sharing their thoughts while keeping their anonymity.

Lastly, I coordinated with the local Office of Public Affairs Office throughout data collection. The mission of the local Public Affairs Office is to serve as the lead public information office for the local command (Office of Public Affairs | Homeland Security, n.d.). I leveraged the local Public Affairs Office to ensure that the content of the research and dissertation comes from open sources. Additionally, I used a computer that did not require a Common Access Card (CAC) to ensure the data he collects does not come from sites that require a CAC.

Summary

In this qualitative case study, I explored how a hybrid approach of the waterfall and agile methodologies can address failures in government software development. I was a team member of a government software development organization that took over the development of training software for a weapons system. The customer expressed frustration with the previous developer because they did not satisfy the customer's needs. The leader of the studied government software development organization used a hybrid waterfall and agile methodologies approach to satisfy the customer's needs. I used this phenomenon for my case study because it provided a unique perspective of a hybrid approach to software development for a government entity.

This study addressed how the studied government software development organization used a hybrid approach to address the failures in government software development. Researchers classified 31.1% of IT projects as failed, meaning leaders/managers abandoned or canceled projects, and discovered that 52.7 of completed IT projects are over cost, over time, and/or lacking agreed-upon functionality (Iriarte & Bayona, 2020). I realized that the failure of the local civilian software development company was common and wanted to study why the studied government software development organization was successful with the same task. I concluded that the methodology was the critical difference between the local civilian company and the government software development entity. The local civilian software development company used a waterfall methodology, while the government software development organization used a hybrid approach of the waterfall and agile methodologies. However,

the leader of the studied software development organization had to convince the organization's parent organization and customer to agree to the hybrid approach. I attributed the parent organization and customer's adaptation of the hybrid approach to DOI.

I used DOI as a theoretical lens to better understand how the parent organization and customer agreed to the studied software development organization's hybrid approach of the waterfall and agile methodologies. DOI is one of the most commonly used theories to study the adoption of innovations that focus on the traits that influence whether or not users accept innovation (Amini & Jahanbakhsh, 2023; Menzli et al., 2022). Once the parent organization and customer agreed to the innovation of the hybrid approach, the studied organization implemented the new approach to address the failures of the local civilian software development company.

I used face-to-face interviews as the primary means of data collection. I interviewed eight former and current team members of the studied government software development organization to reach data saturation. There were no new data, themes, or coding after that number of interviews (see Aguboshim, 2021). I used analysis of documented observations as the secondary means for data collection. The former leader of the studied government software development organization conducted weekly meetings with a customer representative and key leaders of the studied government software development organization and had a scribe create verbatim minutes of these meetings. Lastly, I analyzed quarterly organization charts to see how the hybrid approach affected the studied organization's turnover rate. I used these three means of data

collection to provide data triangulation for his research.

I collected the data to determine the success or failure of the hybrid approach. Researchers use a qualitative approach to collect group members' experiences to generate general rules from which researchers can generate hypotheses (Bazen et al., 2021). I collected and documented the experiences provided by former and current team members of the studied organization to generate a general rule and develop a hypothesis about how a hybrid approach of the waterfall and agile methodologies addressed the failures of the local civilian software development company. I took measures to facilitate the rigor of his research to support the generated rule and developed hypothesis.

There are multiple methods for researchers to provide rigor to their research. Researchers commonly use data triangulation to provide rigor to their research (Johnson et al., 2020). I used data triangulation through face-to-face interviews and data analysis (see Amelia & Hartanti, 2023). Additionally, researchers can use trustworthiness to establish the rigor of their research (Dyar, 2022). The four trustworthiness factors are credibility, transferability, dependability, and confirmability (Amin et al., 2020). I incorporated these factors in the research to provide rigor and support the findings.

Chapter 4: Results

To explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies, I researched the inherent advantages and disadvantages of both methodologies. With that foundation of knowledge, I conducted interviews with eight current and former members of the studied government software development entity. Additionally, I analyzed the first 52 IPT meetings between the studied government software development entity and its customer for the developed software. Lastly, I analyzed the initial and subsequent organizational charts of the studied government software development entity to provide insight into the turnover rate associated with a hybrid approach of the waterfall and agile methodologies. The use of multiple avenues of data collection helped me answer the research question:

RQ: What consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology?

When responding to the research question, I wanted to ensure trustworthiness in the study. The use of these multiple avenues of data collection also helps provide trustworthiness to the research through triangulation (Morgan, 2024). Along with triangulation, data saturation is a key component of the research. Therefore, I looked for the criteria for data saturation in research.

While conducting the study, I wanted to ensure he collected enough data to provide a valid conclusion. To that end, I wanted to reach data saturation. Data saturation is the point where increased data collection does not provide new codes of themes for the

research (Mannan, 2024). Using relatively homogeneous individuals, Guest (2006) wrote that saturation should be reached from eight to 12 interviews.

Pilot Study

I conducted a pilot study in support of this dissertation. Researchers use pilot studies as small-scale studies at the beginning of the research to identify challenges that they may face during their research (Mondal e al., 2023). To that end, I conducted a face-to-face interview with a former and current team member of the studied government software development entity. During the pilot study, I was able to identify possible challenges to the research. One of the challenges that I identified during the pilot study was the coordination of dates and times to conduct the face-to-face interviews.

When I decided to conduct the pilot study, he anticipated some challenges that the pilot study would identify. However, I did not anticipate a challenge with the coordination of dates and times to conduct the face-to-face interviews. Since I and pilot study interviewees worked, it was a challenge to coordinate a good date and time for me to conduct the interviews with the interviewees. Along with coordinating a date and time to conduct the interview, I found challenges in where to conduct the face-to-face interviews for the pilot study.

Once I was able to coordinate the date and times for the pilot study interviews, he faced challenges with finding an appropriate place to conduct the interview. I wanted to find a location for the face-to-face that yielded a convenient environment for the interviewees that also limited distractions and provided privacy for the interviewees (see Demirci, 2023). When interviewing the current team member for the pilot study, I

conducted the interview within the footprint of his workspace, and the interviewee was visibly concerned about those who may overhear his comments. As a result, I knew I had to conduct the interviews away from the footprint of the studied government software development entity. During the pilot study, I discovered the importance of the vehicle used to record the interviews.

During the interviews for the pilot study, I realized that the term “consequences” had a negative connotation. In DOI theory, Rogers used the term “consequences” to identify the outcome of innovation (as cited in Hari et al., 2021). Since I used DOI theory to ground this study, he wanted to stay consistent with the terminology for outcomes. However, when the interviewees heard the term “consequences,” both of them searched to find negatives about the hybrid approach. As a result, in the interviews for the study, I stressed to the interviewees that they could interpret the term “consequences” for both positive and negative outcomes of combining waterfall and agile methodologies.

Lastly, I used my cellphone to record the audio of the face-to-face interviews for the pilot study. However, I discovered that the format my cellphone used to record was not a compatible format to use as input for the Microsoft Word application to transcribe. If I had continued to use my cellphone to record the face-to-face interviews, it would have been very time consuming to manually transcribe the interviews. Researchers have concluded that it can take a skilled executive assistant 6 to 7 hours to transcribe a 1-hour interview (Rosenberg & Mojadeddi, 2024). If I had failed to realize this challenge until after I had conducted all of the interviews, it would have cost me a large amount of time that could have been avoided. Instead, I procured an audio recorder that enabled me to

record the interviews in a format that was compatible for input into MS Word application.

Research Setting

The initial leader for the studied government software development entity first pitched the idea of creating the entity to the customer in 2015. The initial leader pitched the idea of creating the government software development entity to the customer because he had heard that the customer was not satisfied with the commercial software development company that was developing and providing the software at the time. Specifically, the customer was dissatisfied with the lack of flexibility and periodic input in the software development process. Also, the customer believed that the cost of the project was far more expensive than the quality of the product. With that information in hand, the initial leader of the studied government software development entity began to create a proposal for the customer to take over the work from the commercial software development entity.

The initial leader for the studied government software development entity already had a relationship with the customer. At the time of the proposal to take over the work from the commercial software development entity, the initial leader of the studied government software development entity was currently serving as the lead for another program for the customer. This relationship gave the initial leader an insight into the key decision makers within the customer's organization. Additionally, the initial leader had space in the facility for a seamless transition.

After giving the initial proposal, the customer gave the initial leader $250,00.00 to

procure equipment for the new lab. After 3 months, the newly created government software development entity had made enough progress in equipment procurements to foster an additional investment in the government software development entity for new employees. At the 6-month mark, the government software development entity was able to start the transition of the work from the commercial software development entity. By 2016, the government software development entity had an organizational chart of 45 team members. Of those 45 original team members, only a few are still with the program.

Demographics

Reference to the demographics of the interviewees, I used multiple viewpoints. Specifically, I used age, gender, job position, and current place on the studied government software development entity (i.e. current or former). I chose these specific viewpoints to differentiate and categorize responses as it pertained to the research question and experience with performing their individual tasks. I also chose these viewpoints to identify and correlate trends of responses based on these perspectives of the interviewees.

Age

Because of contractual requirements for the studied government software development entity and to maintain research quality, none of the interviewees were younger than 18. Additionally, none of the participants were older than 70. At the time of being a team member of the studied entity, all interviewees were legally employed on a government contract. Consequently, the interviewees were between 30 and 60 and active in the day-to-day activity of the community.

Gender

The studied government software development entity had male and female team members. This has been true from the beginning of the studied software development entity. I wanted to gauge if there were any differences in options based on gender.

Job Position

The studied government software development entity had an internal hierarchy. Directly under the initial government leader of the studied software development entity, there were people in leadership positions in the program. Directly under these leaders were functional leads (FL), who oversaw specific functions that make up the overall performance of the software (i.e. Systems Engineering, Software Engineering, Test Engineering, Lab Architecture, Courseware, etc.). Directly under the functional leads are program participants (PP) who provide a focused contribution to the team under the function their position resides.

Current Place on Studied Entity

I interviewed former and current team members of the studied government software development entity. I decided to use former and current team members to analyze how the program has matured over the several years of its existence. Additionally, I planned to solicit interviews from multiple job positions of former and current team members.

Data Collection

I had to make one change to this original data collection plan. Initially, I intended to use video and audio recordings of the interviews. I believed the video recordings of the

interviews would allow me to go back and gauge body language and other non-verbal cues from the interviewees that would gain more insight into the data he gathered from the interviews (see De Villiers et al., 2022). However, both the Walden University IRB and AHRPO told me that I could not video tape the interviews. Instead, I could only use audio recordings of the interviews. To compensate for the inability to video record the interviewees and record non-verbal cues, I intended to take note during the interviews on non-verbal cues that he observed during the interviews. However, I did not see any major non-verbal cues from the interviewees that warranted notetaking.

Data Analysis

Table 1 is a participation table of the interviewees.

Table 1

Participation Table

Participant #	Age	Gender	Position	Place
P01	65	Male	Leadership	Current
P02	45	Female	Program participant	Former
P03	64	Female	Functional lead	Former
P04	51	Male	Leadership	Former
P05	49	Female	Functional lead	Current
P06	59	Male	Program participant	Current
P07	43	Female	Functional lead	Former
P08	45	Male	Functional lead	Current

Referring to the Demographics section of this chapter, I interviewed two people who served in leadership positions within the studied government software development entity – one former and one current. Additionally, I interviewed four people who served in functional lead positions within the studied government software development entity–

two former and two current. Lastly, I interviewed two people who served in program participant positions within the studied government software development entity–one former and one current. With regard to their answers to the 10 interview questions, the responses from all eight interviewees were consistent throughout the face-to-face interviews.

In Question 1, I asked the interviewees to describe the waterfall software development methodology in their own words. Each interviewee described a standardized, linear process of software development where developers could not move to the next step of software development until they had completed the preceding step in the process. The sequential definition of software development is consistent with research for the waterfall methodology (Saravanos & Curinga, 2023). I used Question 1 to set up Question 2.

In Question 2, I asked the interviewees how they felt about the waterfall software development methodology. The interviewees' responses were mixed. Some like it, some do not. However, the interviewees who like waterfall conceded that it is not good for larger software development projects because of its inherent shortcomings. The interviewees mentioned that the sequential nature of the waterfall methodology led to challenges such as difficulty with adapting to changing requirements as well as limited customer feedback and participation of all members of the software development team throughout the process (see Kirpitsas & Pachidis, 2022). In response to these challenges, software developers created the agile software development methodology.

In Question 3, I asked the interviewees to describe the agile software development

methodology in their own words. Each interviewee described an iterative approach of software development where developers are working on smaller amounts of time called "sprints," which allows for customer feedback early in the process. Interviewees describe the agile software methodology as more collaborative than in the waterfall software development methodology. The interviewees' iterative and collaborative description of software development is consistent with research for the agile methodology (Sarkar et al., 2024). I used Question 3 to set up Question 4.

In Question 4, I asked the interviewees how they felt about the agile software development methodology. Every interviewee responded that they liked the agile methodology and preferred it over the waterfall methodology. The interviewees mentioned that the iterative and collaborative nature of the agile methodology was better for changing requirements and provided more flexibility than the waterfall methodology. The research supports this concept of greater flexibility (Rasheed et al, 2021). However, there are still disadvantages with the agile software development methodology such as team management challenges and limited documentation (Singh, 2021). In response to these challenges of both software development methodologies, software developers created the hybrid approach of the waterfall and agile software development methodologies.

In Question 5, I asked the interviewees to describe a hybrid approach of the waterfall and agile software development methodologies. A hybrid approach of the waterfall and agile software development methodologies is a combination of a plan-driven software development methodology (waterfall) and agile (Yahya et al., 2021). The

interviewees described a hybrid approach of the waterfall and agile software development methodologies that took the benefits of the waterfall software development methodology – requirements planning and predictability–and combined them with the benefits of the agile software methodology–flexibility and customer involvement, which is in line with the research.

In Question 6, I asked the interviewees how they felt about the hybrid approach of the waterfall and agile software development methodologies. Every interviewee except one responded that they felt good about the hybrid approach. Multiple interviewees described the hybrid approach as "the best of both worlds," meaning that the hybrid approach would accentuate the strengths of the respective software development methodologies while mitigating their respective weaknesses. The lone exception interviewee, P06, had a discrepant response to this question:

> A disaster. And, it's stupid, and it doesn't work. I think you are fooling yourself if you think that it is going to work because I've seen that. I saw that when we were trying to make a transition. People were saying that we're going to go hybrid, first. And, it doesn't work. You're either all in with one or the other. Because I think it has to do with the mindset. A hybrid approach is pretending to drive on both sides of the road at the same time to pretend that you can go both ways. You cannot. You cannot do that.

However, P06 conceded that certain parts of the studied government software development entity used aspects of the waterfall methodology while other parts of the

studied software development entity used aspects of the agile methodology, making it a hybrid approach. Specifically, during his interview, P06 said, “On the development side, we are using agile … Now, let’s flip over to the sustainment side. Absolutely not agile.” While transcribing the interview with P06, I determined that P06 interpreted a hybrid approach as moving back and forth between waterfall and agile methodologies, which was not the intent of the question.

In Question 7, I asked the interviewees about the consequences that combining waterfall and agile methodologies might foster. After the pilot study, I realized that the term “consequences” had a negative connotation. However, I used the term “consequences” due to its use in DOI theory, which I used to ground this study. To offset this misconception, I stressed that the term “consequences” could be used for both positive and negative outcomes of combining waterfall and agile methodologies. In response, all but one of the interviewees described that combining waterfall and agile methodologies would provide a better product because it would stress the strengths of the respective methodologies while mitigating their weaknesses. Additionally, P08 stressed the importance of communicating with the customer in a hybrid approach.

In Question 8, I asked the interviewees if they felt the studied government software development entity combined the waterfall and agile methodologies and why. All but one of the interviewees felt that the studied government software development entity combined both methodologies. Their respective answers led to the follow-up question.

In Question 9, I asked the interviewees about the impacts of their answer to

Question 8 on the studied program. All interviewees expressed that combining the waterfall and agile methodologies positively impacted the program. Their respective answers led to the final question.

In Question 10, I asked the interviewees how they felt about the overall success of the program. All of the interviewees expressed that they felt the program was a success. Using all of this data from the interviews of former and current team members of the studied government software development entity, I delineated themes that permeated throughout the interviews.

Table 2 shows the themes that permeated throughout the interviews of former and current team members of the studied government software development entity.

Table 2

Themes From Interviews

Theme	Description	Frequency
1	Hybrid approach: "Best of Both Worlds"	39
2	Agile is better than waterfall	30
3	Resistance to change	19
4	Lack of knowledge of methodologies	8

Theme 1: Hybrid Approach is "the Best of Both Worlds"

The literature review and face-to-face interviews of former and current team members established that a hybrid approach of the waterfall and agile methodologies can emphasize the advantages of both methodologies while mitigating their respective weaknesses. Yahya et al. (2021) wrote that the hybrid approach is "testable, reliable and maintainable." My findings in this dissertation support this statement.

Multiple interviewees expressed that the hybrid approach of the waterfall and

agile methodologies accentuated the strengths of the methodologies while mitigating the weaknesses. Interviewees P04, P05, and P07 used the exact phrase “best of both worlds” in their respective interviews to describe how the hybrid approach of the waterfall and agile methodologies accentuated the strengths and mitigated the weaknesses of the respective methodologies. In Question 7, all but one of the interviewees, P06, expressed that the hybrid approach provided positive outcomes due to its combination of strengths from the waterfall and agile methodologies. Additionally, the interviewees provided several extended quotes to support the theme of a hybrid approach of the waterfall and agile methodologies accentuating strengths while mitigating weakness of the respective software development methodologies.

While there were several extended quotes to choose from for the hybrid approach of the waterfall and agile methodologies, I decided to highlight a few of them. P02 stated, “The hybrid (leveraged) the strengths of each of them (waterfall and agile). Using the waterfall for the well-defined, predictable tasks and agile for the uncertain, iterative components.” P04 stated, “(The hybrid methodology) start(s) with defined types of requirements, like waterfall, then transition(s) to more of an agile process as you get to the final product.” P07 stated that the hybrid approach provided the following: “More efficiency. More collaboration. Like I really cannot think of any negatives.” Lastly, P08 stated that the hybrid approach allows software development teams to

> Plan in advance. And, have your scope of what you are going to do and when you are going to deliver it (waterfall). And, in the Development phase, you break that down into sprints and test as you go (agile). And then, from that point, you do

more of a phased approach at the end where you test it all together (waterfall).

All of these extended quotes support the theme that the hybrid approach of the waterfall and agile methodologies accentuates the strengths of waterfall and agile while mitigating their weaknesses. However, while each of the methodologies has its inherent weaknesses, the interviewees had a clear preference between the two methodologies.

Theme 2: Agile is Better Than Waterfall

When discussing the two software methodologies, the interviewees clearly favored the agile methodology over the waterfall methodology. Again, the waterfall methodology has its inherent strengths. For example, P02 stated that the waterfall methodology provided a,

> Fixed timeline and budget (for projects). Project leaders can use the waterfall methodology when requirements are well understood in the beginning and not expected to change over the life of the project. (Additionally, with waterfall) Project leaders can sets schedules with deadlines for each phase of development. And milestones can be easily identified.

Additionally, P02 stated, "Waterfall emphasizes predictability and sequential phases ensuring thorough documentation and controls." Lastly, P04 stated that the waterfall methodology provided "clearly defined requirements … clear process." However, the interviewees provided far more negative comments about the waterfall methodology than positive comments.

Almost all of the interviewees had negative comments about the waterfall methodology. P03 stated that waterfall "does not allow for any type of reaction or

appropriate changes for problems" and "can take a long time. And, maybe at the end of it, I may end up with something that is different than what my intent was because things change." P05 stated that waterfall was "not one of the best choices (for software development methodology" and he "(didn't) really appreciate the serial-ness of (waterfall). Waiting for one event to end before another one starts." P01 stated that team members using the waterfall method try to

> pretend to be busy when they are not … you are wasting a lot of manpower and things like that … they don't want to have people see them not being busy. You know their jobs are on the line.

In addition to the negative comments about the waterfall methodology, the interviewees were also clear about their preference for the agile methodology over the waterfall methodology.

The interviewees stated that they preferred agile over waterfall in an overwhelming fashion. P05 stated, "I love it (agile) over waterfall." P06 stated that developers "find out really quickly that (waterfall) doesn't work so well." Additionally, P06 stated that agile was "Definitely superior to waterfall." P07 stated, "Agile is more collaborative across those different functions within the software development lifecycle. And I think it works a lot better with quick response, rapid response with feature derivation and production." P07 stated that waterfall was "good for what it is good for. But it's not the best idea for software development, especially if you want to do things quickly." P08 stated that if he had to choose between predictability and flexibility, he "would lean more to having more flexibility" and agile provides "more flexible. More

engagement with the customer." Even though there was overwhelming support for agile over waterfall, this paradigm shift can come with some resistance to the change.

Theme 3: Resistance to Change

Resistance to a change is just as much of a constant as change itself. P01 made several statements about the resistance to change. For example. P01 stated, "(There are) guys who really would love to implement just waterfall." Additionally, P01 stated, "(There is) always going have some level of waterfall, at least in DoD stuff." In the instance, P06 was working for a popular software application, the application's leadership had to decide to make a drastic change in how it executed a critical task in its software. The organization needed to make this change to ensure the overall success of the company. The value that this change brought eventually outweighed the challenges the adoption of the innovation faced. However, P06 clearly stated how ego and job protection motivated many challengers to the proposed change instead of the viability of the product. For example, P06 stated, "You can't have that conversation (of change) sometime because people will say: 'Do you have any idea how much money we have spent on this.'" Additionally, P06 stated, "(Organizations are) stuck in the habit of doing bad things, usually because of people's ego." Many times, resistance to change comes from a solid understanding of the important facts of situations. Other times, decision makers' lack of knowledge could lead to resistance and poor decisions.

Theme 4: Lack of Knowledge of Methodologies

When interviewing a current leader in the studied government software development entity, some of the current leader's responses shocked me. During the face-

to-face interview, P01 stated, “You still have to go through design. Before you can even really start doing the agile process.” However, the Literature Review of this dissertation depicts that design is part of the agile methodology. The agile methodology requires design for software developers to communicate with other software developers and other stakeholders (Moyano et al., 2022). Therefore, P01’s assertion that design is not part of the agile methodology is inaccurate.

When P01 made the comment, it implies that there is little to no planning in the agile software development methodology. However, the agile software development methodology does include planning. Moyano et al. (2022) wrote that leaders in agile software development craft plans to provide an overview of what their teams should do and help their teams mitigate potential risk. P01 has worked in the DoD software development industry for decades. Additionally, P01 had a Master of Science degree, was knowledgeable about software development, and was a productive and essential member of the studied government software development entity. As a result, I was surprised that P01 would make such a comment about the agile process. However, when I did more research on why people think certain things, a reason for P01’s misconception arose.

Since P01 has been in the DoD government software development industry for decades, he has been around those in the DoD software who exclusively learned the waterfall methodology in their undergraduate education. Over the years, P01 began to believe that their comments about the agile software development methodology were true because he believed their comments were accurate, even without clear evidence to support their claims; P01 did not have a reason to suspect otherwise. Researchers call this

phenomenon truth-default theory or TDT (Levine, 2022). When P01 heard others make comments about the agile methodology, he took their comments to be true, even though they are not. And, P01 did not have any personal experiences to contradict what he heard.

Further in the face-to-face interview, P01 acknowledged his lack of experience with the agile software development methodology. P01 stated, “I have never been part of a company that has done true agile from start to finish. So, I do not know how well it works. This is me working out how it would look if we tried it.” P01 had never been part of an organization that did true agile from start to finish. As a result, he did not know how well it worked. In his words, he was “working out how it would look if he tried it.” This comment also begs the question of where is he getting his opinion on the agile methodology if he has never been part of an organization that did true agile from start to finish? As I continued to conduct the face-to-face interview, it was apparent that P01 was gleaning his opinion about the agile methodology from other people when he made another comment about the agile software development methodology and government software development.

P01 made another assertion about where software developers should use the agile software development methodology that was not accurate. P01 stated that once you have a product, you could use the agile methodology, but not if you were starting a new product. P01 further asserted that once the organization had a software, the sustainment and updating of that product would lend itself to the agile software development methodology; however, the agile software methodology would not work in the DoD. This was another inaccurate statement about agile.

In her face-to-face interview, P02 alluded to programs that were in P01's division, which are DoD projects and use purely agile software development methodology. The realization that a qualified, successful leader in the studied government software development entity would make such assertions surprised me. However, it did enlighten him on the challenges the agile software development methodology faces within the DoD and possibly other software development entities. Another challenge to the correct selection of software development methodology is the tendency to attempt to be consistent and how that can adversely affect decision making.

If a decision maker in DoD software development has used the waterfall methodology for years or even decades, they have developed comfort and familiarity with it. In turn, this comfort and familiarity could cause the decision maker to have perceptions on the methodology that are contradictory to the possible success of another methodology (i.e., agile). DoD decision makers need to be aware of this unconscious tendency to not contradict what they perceive as a methodology that has produced success. Instead, they need to evaluate the requirements for the software and the conditions surrounding the development of that software and have those factors determine the software methodology.

As written earlier in this study, many DoD software development programs had a mandate to use the waterfall methodology. As a result, software developers within the DoD became very knowledgeable with the waterfall methodology to be successful. This extensive knowledge led to intense familiarity with the waterfall methodology. This familiarity has led to comfort with the waterfall methodology and an uncertainty with

other software development methodologies such as agile. This familiarity bias may be keeping DoD software development decision makers from making the correct choice when it comes to deciding on the software methodology for their respective programs. In the same way I was allowing his comfort and familiarity with my personal experiences to guide the study versus theory, many DoD software development decision makers may be allowing their comfort and familiarity with the waterfall methodology to guide their decisions on the software development methodology versus the requirements and conditions of the project.

During his interview with P06, I recalled how the initial government leader of the studied government software development entity attempted to start a previous program that used the agile software development methodology. However, this program failed. When the program failed, I said that the DoD stakeholders "tied agile around its (the program's) neck and watched it drown." When this project was unsuccessful, I believed that the DoD stakeholders either unfairly and/or unwittingly blamed the use of the agile software development methodology for the failure of the program. As written earlier in this dissertation, this phenomenon is called confirmation bias. Then, I recalled discussions with the initial leader of the studied government software development leader that puzzled me at the time, but made more sense as I wrote this dissertation.

While learning about multiple software development methodologies during his education and training, my personal work experience and familiarity has been in the agile software methodology. As a result, I recalled conversations early on in the studied government software development entity's existence with the initial government leader

where I voiced how the entity was using aspects of the agile software development methodology. I recalled how the initial government leader of the studied government software development entity would quickly and forcefully push back on this assertion. In hindsight, I realized that the leader's perceived experience with the agile software development methodology and the familiarity bias of the customer may have solicited this strong response to the suggestion of using agile principles in the studied government software development entity (see Tan, 2021). However, familiarity and perceived past experiences should not influence the decision on a program's software development methodology. Other factors should determine that decision.

In addition to using face-to-face interviews, I also used the first 52 IPT meetings between the studied government software development entity for analysis in this study. The IPT meetings served as the consistent feedback from the customer to the studied government software development entity. I used the minutes from these IPT meetings to determine how much the participants referred to waterfall and agile methodologies.

I used the MAXQDA software to analyze the IPT meetings. When I analyzed the minutes from the first 52 IPT meetings of the studied software development entity using the MAXQDA software, the analysis yielded a spreadsheet of the terms assigned to either the waterfall or agile methodology. The spreadsheet depicted an even distribution of the terms for waterfall and agile–17 waterfall terms, 16 agile terms. However, when the reader looks at the frequency of terms, there is a strong focus towards the agile methodology.

The first four rows of the MAXQDA spreadsheet of the analysis of the first 52

IPT meetings are all associated with the agile methodology. Furthermore, from the first ten rows of the MAXQDA spreadsheet of the analysis of the first 52 IPT meetings, seven of them are associated with the agile methodology. From these observations, readers could infer that aspects of the agile methodology were far more prevalent in the first 52 IPT meetings than the waterfall methodology. In association with the discovery that the studied government software development entity has shifted to a largely agile methodology for the development of the software, the reader can infer that the studied government software development entity could have used the agile methodology from the very beginning of the program's inception. The literature review, past discussions with the first government civilian leader of the studied software development entity, and subsequent face-to-face interviews support the concept that decision makers in the customer organization may not have been comfortable with using a largely agile software methodology for this program. As a result, the studied government software development entity slowly integrated more and more aspects of the agile software development methodology in the software development process over time. As the decision makers became more and more comfortable with the aspects of the agile methodology and satisfied with the products the agile-focused methodology provided, the customer and studied government software development entity agreed to move to a mostly agile methodology for the program. According to Nwachukwu and Boatengu (2024), organizational resistance is one of the most significant challenges organizations face when adopting agile methodologies. However, according to Rogers (2003), desirable consequences are the functional consequences of an innovation. Therefore, the desirable

outcome of the hybrid, and eventually largely agile, software development methodology overcame the organizational resistance towards it.

Lastly, I used the organizational charts from the studied government software development entity to assess turnover in the program. The organizational charts are from the beginning of each of the first four quarters of studied government software development entity. Each organizational chart has a date at the bottom left of the slide to tell the reader when the organizational chart was created. The charts serve as "skeletons" of the organization with solely the position names that constitute the organization. I created a color scheme to delineate positions in the studied government software development organization as well as when there was a change in who held that position.

In the first organizational chart, I colored all of the positions in "Blue" except for the government lead position, which is colored in "Purple." In the studied government software development entity, the program consisted of one government civilian; the rest of the program were government contractors. In the three subsequent organizational charts, I colored positions that had a new person in that position from the previous quarter in "Orange." The contrast in color while using the same organization structure makes it easy for the reader to quickly discern the amount of turnover from one quarter to the next.

Over the years, the studied government software development organization has changed its structure from its inception. Therefore, I took an organization chart of the studied entity from 1 year ago. For the study, this chart served as the initial baseline for the current studied software development entity. Like I did for the initial set of organizational charts, I colored all the positions in "Blue" except for the government lead

position, which is colored in “Purple.” In the three subsequent organizational charts, I colored positions that had a new person in that position from the previous quarter in “Orange.” The contrast in color while using the same organization structure makes it easy for the reader to quickly discern the amount of turnover from one quarter to the next.

Looking at organizational charts, the reader can quickly infer whether there is significant turnover in the studied government software development entity from one quarter to the next. Specifically, large numbers of “Orange” boxes in subsequent organizational charts from the baseline chart denote a large turnover rate for that quarter. Earlier in this study, I wrote that an annual turnover rate of 10% is healthy. In an organization with 45 people, that would equate to four or five people leaving the program in a year’s time. From a quarterly perspective, that would be between one and two people leaving the program each quarter. In terms of a quick view of subsequent organizational charts, a healthy turnover rate would be one or two orange-colored boxes for each quarter. I analyzed the baseline organizational charts from the beginning and a year from this study and compare them to the subsequent three organizational charts.

Evidence of Trustworthiness

Trustworthiness has an important place in qualitative research. Trustworthiness is essential to establishing credibility and reliability of qualitative findings through the rigor of the collected data (Peels & Bouter, 2023). Four pillars provide the rigor of the data (Ahmed, 2024). The first of these pillars is credibility.

Credibility

Qualitative research has struggled with credibility. To that end, the qualitative

research community has derived standards to achieve credibility in its community such as member checking, meaning the collected data is returned to the study participants to allow them to check for accuracy and tone of the collected data (Vella, 2024). In the study, I attained credibility through multiple avenues.

Researchers can use various strategies, such as triangulation, member checks, saturation, prolonged contact, etc., to establish credibility. I used triangulation and saturation in his research. I used established means of triangulation, such as face-to-face interviews and data analysis (see Santos et al., 2020). I had the interviewees member check their respective interviews before he analyzed them. After eight interviews, I reached data saturation because no new data, themes, and coding occurred after this number of interviews (see Aguboshim, 2021). Informally, I used prolonged contact in his research. I was a member of the studied government software development organization. As a result, I understood the culture and setting of the studied government software development organization and had a good relationship with many interviewees (see Sanip, 2020). I used these means to establish credibility for the research. Lastly,

Transferability

Transferability also has an important place in qualitative research. Transferability is the ability to provide enough information so that readers can determine if they can apply the findings from one study or setting to another study or setting (Stalmeijer et al., 2024). As it relates to this study, I used the information gathered during this study to apply it to other similar settings to the studied government software development entity. Specifically, I wanted to generalize the studied government software development

entity's use of a hybrid approach of the waterfall and agile methodologies to other government software development entities.

Most government software development entities share similar challenges. Routinely, changes in how the government performs its tasks are faced with bias and unfairness (Alhosani & Alhashmi, 2024). Many organizations within the government have a well-established way of performing tasks such as software development. I wanted to use this study to help government software development entities transition to using the best software development methodology for the task versus using the software development methodology most familiar to decision makers.

Dependability

Like credibility and transferability, dependability also has an important place in qualitative research. Dependability refers to the level of appropriateness between researchers' findings and the supporting data. Dependability means how closely researchers' findings are supported by the data gathered in their research. Researchers have to be careful that their biases do not adversely affect the dependability of their studies.

Bias can have an adverse effect on research. As a result, researchers must consistently manage personal biases because they can negatively influence data collection, analysis, and interpretation of a study (Ahmed, 2024). By recognizing their biases, researchers can lessen the possible negative effects their biases have on their research. I took steps to manage his bias.

Since I was a former team member of the studied government software

development entity, I realized that my bias would be to provide a favorable outcome for the entity. To address this bias, I used a random pool of former and current team members. By doing so, I did not unconsciously select interviewees for the study who may share his favorable view of the entity.

Confirmability

Confirmability is the fourth and last pillar of trustworthiness. Confirmability stresses that the results of the research derive from the data collected during the study and not from the subjectivity of the researcher (Lim, 2024). As written earlier in this dissertation, researchers' biases can negatively affect the data collection, analysis, and interpretation of a study. Additionally, preconceived notions that are not based on facts can negatively affect a study. I took steps to ensure the confirmability of the data used for this study.

When I performed my presentation for proposal approval, I caught a bias and asked if I could change my research question. In the original research question, I asked what improvements in government software development are fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology. While giving my presentation, I realized that the word "improvements" suggests a positive effect of the hybrid approach before he had collected, analyzed, and interpreted the data. Once I realized my bias, I asked my committee members if I could change the word "improvements" to "consequences". I used the word "consequences" because it refers to an outcome of an event or action without giving that outcome a positive nor negative connotation.

Also, I had a current member of the studied government software development entity send the recruitment email. Once four current and four former team members responded, I used those eight interviewees for his study. Additionally, I had the interview questions reviewed by the members of his committee and the Walden University IRB. An impartial review of the interview questions helped ensure that the interview questions were neither biased nor slanted towards a specific outcome.

Study Results

The research question for this study was as follows: What consequences in government software development are fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology? As was written earlier in this study, I changed the research question to alleviate some bias in his research. Most of the data from the face-to-face interviews, 52 IPTs, and four organizational charts was not a large surprise. As a former team member of the studied government software development entity, I assumed that the program continued to use a hybrid approach of the waterfall and agile methodologies. However, after interviewing the four current team members, I realized that was no longer the case.

When I interviewed the current team members of the studied government software development entity he was surprised by their answers. In short, the project had transitioned to mostly agile ever since they were on the project. The current team members had the program knowledge to provide further fidelity on how the program transitioned from a hybrid approach of the waterfall and agile methodologies to mostly agile. From their interviews, I gathered that the customer for the program wanted

consistent input throughout the software development process. Specifically, the customer wanted the ability to make changes to the software throughout the development process based on factors such as budget and shifting priorities based on feedback from the warfighter. The agile methodology's traits of responsiveness and adaptability made it a natural fit for the demands of the program (Amajuoyi et al.,2024). Along with traits of the agile methodology, a changing leadership construct in the customer organization influenced the change methodology.

A couple of years prior to the program's transition to a mostly agile methodology, the customer's organization underwent a significant change. In response to the COVID-19 pandemic, employment among older workers had a rapid decline due to the risk of infection to that population (Goda, 2023). This decline in older workers brought about a significant change in the hierarchy of the customer organization. Specifically, seasoned decision makers who were at the top of the organization had a familiarity with the waterfall methodology.

These decision makers' familiarity with the waterfall methodology may have made them bias towards the waterfall methodology and against the agile methodology. As written earlier in this study, changes in how the government performs its tasks often face bias and unfairness (Alhosani & Alhashmi, 2024). Additionally, seasoned developers valued the dependability and structure of the waterfall methodology. However, as these seasoned decision makers left the customer organization, younger decision makers took their places. These younger decision makers valued the attributes of the agile methodology more than their predecessors. As a result, the program gradually

moved towards a mostly agile methodology.

While the transition of the program from a hybrid approach of the waterfall to agile methodologies to mostly agile was a surprise to me, the other data collected during the study was not. The face-to-face interviews yielded that the hybrid approach of the waterfall and agile methodologies tend to accentuate the strengths of each of the methodologies while mitigating their weaknesses. Peddisetty (2024) echoed this concept in her research. The face-to-face interviews also yielded why the studied government software development entity adopted the hybrid approach of the waterfall and agile methodologies.

As written earlier, I was a former team member of the studied government software development entity. I was also one of the two people the original leader of the studied software development entity worked with to get this work from the customer, who had originally given this work to a local civilian software development company. The other person whom the original leader of the studied software development entity worked with to get this work from the customer was one of the interviewees in this study and a former team member of the studied government software development entity. Her interview provided why the studied government entity adopted a hybrid approach of the waterfall and agile methodologies.

The original leader of the studied software development entity learned that the customer was dissatisfied with the work the local civilian software development company provided. As a result, the original leader of the studied software development entity worked with me and another person to build a proposal for the customer to receive the

work. The presentation focused on satisfying the needs that the local civilian software development company was not satisfying.

The customer had multiple issues with the local civilian software development company. The customer wanted the program to have the ability to adapt to changing requirements, greater visibility and control, and more collaboration between the customer and software developers during the software lifecycle. While these attributes are neither in the waterfall nor agile methodology, they are attributes in the hybrid approach of the waterfall and agile methodologies (Hua et al., 2023). The studied government software development entity was a program under a CMMI-certified organization, whose processes and procedures closely resembled the waterfall methodology. Consequently, the original leader of the studied software development entity worked with the parent organization to leverage the weekly IPTs as formal trouble tickets and change requests that the studied software development entity could use in DOORS to make changes and revisions to the delivered software. The IPTs also served as a good source of data to compare waterfall and agile concepts within the program.

I used MAXQDA software to analyze the first 52 IPTs between the studied government software development organization and its customer. The MAXQA analysis yielded a spreadsheet that compared how often terms related to the waterfall and agile methodologies appear in the word-for-word transcriptions of the IPTs. The initial review of the spreadsheet reflects an even distribution of waterfall and agile terms. However, a closer look reveals a slightly different interpretation.

Of the 33 words used in the dictionary for the analysis of the first 52 IPTs, I

assigned 17 of those words to the waterfall methodology and 16 of those words to the agile methodology. While the number of words appearing in the IPT meetings were evenly distributed, their frequency was not evenly distributed.

The four most frequent words/phrases of the words in the dictionary – CR (change report), update, rev (revision), and build–are associated with the agile methodology. Furthermore, of the ten words that appear most in the IPTs, seven of those words are associated with the agile methodology. Using this information, I surmised that while the studied government software development entity used a hybrid approach of the waterfall and agile methodologies, the software development leaned heavily toward the agile methodology. However, the customer wanted to retain positive attributes of the waterfall methodology.

Some of the waterfall attributes that the customer wanted to maintain in the software's development were a focus on the process, baselining, and testing. Research supports that those three concepts are waterfall attributes (Thesing et al, 2021). The MAXQDA analysis of the first 52 IPTs supports the idea that waterfall concepts were present within the program. According to the MAXQDA spreadsheet, these three words appear within the first ten most frequently used words in the first 52 IPTs. These words and the frequency of their appearances within the first 52 IPTs also support the importance of waterfall concepts within the program. With the confirmation of a hybrid approach, I wanted to analyze the effect the hybrid approach had on the turnover rate of the program.

Turnover is a reality for every organization. Work environment and stress greatly

impact turnover (Nurimansjah et al., 2023). Looking at the first four and last four organizational charts for the studied government software development entity, the reader can quickly see a drastic difference between the two sets of organizational charts. Comparing the first four organizational charts to one another, the reader can see approximately two positions within the studied government software development entity has a different person in that position each quarter. In contrast, when comparing the last four organizational charts to one another, the reader can see approximately two positions within the studied government software development entity has a different person in that position over the entire year. The two main differences between the first year of the studied government entity and the last year of the government entity were leadership and the methodology for developing the software. I took a closer look at the methodology for developing the software.

During the first year of the studied software development entity, the program used a hybrid approach of the waterfall and agile methodologies. A hybrid approach of the waterfall and agile methodologies requires a comprehensive knowledge of methods and software development on the parts of the team's management and team members (Reiff & Schlegel, 2022). If team members do not feel they have the skill set to accomplish their assigned task, their work stress could rise and lead to a desire to leave the program. If a high percentage team members feel this stress, it could lead to a high turnover rate in the program.

To compare the turnover rates between the first year and last year of the studied government software development entity, I wanted to identify a standard turnover rate for

organizations. Between 2001 and 2021, the United States' overall turnover rate for organizations ranged between 20% and 30%, annually (Dogru et al., 2023). Furthermore, turnover in government agencies have a higher transactional cost than public agencies because of the government's hiring procedures and security concerns (An & Meier, 2022). During the first four quarters of the studied government software development entity, the organization averaged between 10% and 20% turnover, quarterly. In comparison, during the last four quarters of the studied government software development entity, the organization averaged between 10% and 20% turnover, annually. Therefore, the turnover rate in the beginning of the studied government software development entity, when using the hybrid approach of the waterfall and agile methodologies, was much higher than the turnover rate over the last year, when using solely the agile methodology.

Summary

I attempted to answer what consequences a government software development entity fostered by using a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology. To that end, I used multiple forms of data collection. First, I conducted eight face-to-face interviews of former and current team members of the studied government software development entity. Next, I analyzed the first 52 IPTs between the studied government software development entity and its customer. Lastly, I analyzed the organizational charts of the first four quarters of the studied government software development entity and the last four quarters of the studied software development entity. While all of the means of data collection were

important, the primary means of data collection for this study was face-to-face interviews.

Qualitative research has multiple means of collecting data. While qualitative research has several acceptable means of data collection, interviews build nuances and variations across data (Knott et al., 2022). To that end, I conducted face-to-face interviews and learned that the program had transitioned from a hybrid approach of the waterfall and agile methodologies to solely agile. Additionally, the face-to-face interviews yielded that the hybrid approach of the waterfall and agile methodologies accentuated the strengths of each methodology while mitigating their respective weaknesses. Lastly, the face-to-face interviews yielded how the original leader of the studied government software development entity combined the methodologies to conform to a CMMI-certified parent organization. Along with face-to-face interviews, I used document analysis for data collection.

I analyzed the first 52 IPTs between the studied government software development entity and its customer. I used MAXQDA to analyze the word frequency of words and phrases in the documents. Specifically, I used words associated with either the waterfall and agile methodology and placed those words and phrases in a dictionary that the MAXQDA used for analyzing. The result was that while there were many references for waterfall and agile, the most references were for the agile methodology, meaning the agile methodology was more prevalent in the program.

Lastly, I analyzed organizational charts for the first four quarters and last four quarters of the studied software development entity. The reader could quickly access that

there was much more turnover in the first four quarters of the studied government organization than in the last four quarters. Along with changes in leadership, the main change from the first four quarters and last four quarters is the software development methodology used. During the first four quarters, the organization used a hybrid approach of the waterfall and agile methodologies. During the last four quarters, the organization used a mostly agile methodology.

Chapter 5: Discussion, Conclusions, and Recommendations

The purpose of this qualitative study was to explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies. To that end, I decided to use a case study because he wanted to use a research design that would facilitate a rich story about the program to address the research questions in this qualitative study. I studied these consequences as they pertain to the inherent shortcomings of each methodology. Through eight face-to-face interviews, data analysis of the first 52 IPT meetings, and data analysis of the first four quarterly and last four quarterly organizational charts of the studied government software development entity, I explored the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology.

I used Rogers's DOI theory as the theoretical lens to understand how the parent organization of the studied software development and customer agreed to the hybrid approach of the waterfall and agile methodologies. DOI is one of the most commonly used theories to study the adoption of innovations that focus on the traits that influence whether users accept innovation (Amini & Jahanbakhsh, 2023; Menzli et al., 2022). To enable future government software development entities to take advantage of the positive consequences a hybrid approach of the waterfall and agile methodologies may foster, I conducted a case study on how the studied government software development entity implemented a hybrid waterfall and agile methodology. This change in government software development could directly contribute to positive social change through a more

efficient and effective government for its citizens.

With the need for social change through more efficient and effective government for its citizens, I performed a review and synthesis of the strengths and weaknesses of the waterfall and agile methodologies. From this study, the reader can infer that the hybrid approach does emphasize the advantages and mitigate the disadvantages of both methodologies. However, the more important choice is to decide on the correct methodology based on the requirements of the customer and the conditions of the project.

Interpretation of Findings

After conducting the eight face-to-face interviews, I arrived at a finding that I did not expect. At the beginning of the study, I believed that I was conducting a study on the shortcomings of the waterfall and agile methodologies, and those respective shortcomings in the methodologies were the reason why most software development projects fail (see Mishra & Alzoubi, 2023). However, as I conducted and transcribed the face-to-face interviews, he realized that the flaws of people, mainly people in decision-making positions, were just as much a factor in the failure of projects as the methodologies. In short, I told a mentor of his who has earned his doctoral degree, “I started my dissertation believing it would be a technical endeavor on the waterfall, agile, and hybrid methodologies and a social sciences study broke out.” Before elaborating on the social sciences knowledge obtained through this dissertation, I needed to establish the definition of the “social sciences.”

Like many phrases and terms often used by people who are not formally trained on the subject, people have used the term “social sciences” with various meanings. For

the sake of this dissertation, “social sciences” is the study of how humans impact the things around them (Zalasiewicz et al., 2024). With regard to software development, how humans perceive software methodologies often determines which methodology they decide to use for their software development projects.

Like social sciences, the term “hybrid” is a familiar term. However, many may not know the formal definition. A hybrid is the mixing or combining of different characteristics to obtain a better version (Jamous et al., 2021). That definition provides the basis for the hybrid approach of the waterfall and agile software development methodologies. Both of these software development methodologies have inherent strengths and weaknesses. However, the hybrid of the two methods combines the characteristics of each methodology to obtain a better software development methodology.

Originally, software development did not have a formal methodology. Instead, there was “code and fix” process where software developers simply wrote code and fixed bugs in the code until the customer was satisfied (Kakar & Kakar, 2020). As software development grew and software development projects became larger endeavors, software developers needed a more formal way to develop software. As a result, software developers created the waterfall methodology that provided more rigor to the software development process. Some advantages of the waterfall software development methodology are excellent controls for projects and easier management of large projects excellent control lines for the project. It makes the management of large projects much easier, as the project organization is easily understood and planned (Ebule, 2025).

However, software developers became frustrated with the waterfall methodology's limited flexibility with requirement changes and other unforeseen conditions (Ghena & Ghiculescu, 2023). Due to this frustration with the lack of flexibility, software developers created another software development methodology–agile.

The agile software development methodology differs from the waterfall software development methodology in multiple ways. One of the biggest differences between the methodologies is that the agile software development methodology emphasizes flexibility and collaboration, whereas the waterfall software development methodology focuses on organizational and development structure (Diansyah et al., 2023). This shift in focus for the software developers reflects a shift in focus for software development industry. As a result, while both methodologies have their inherent strengths and weaknesses, the preference for flexibility has driven the preference methodology.

Neither waterfall nor agile is a "good" nor "bad" software development methodology. Instead, the circumstances should drive the choice or preference for methodology. Customers often change their minds and may not have a firm grasp of their requirements (Shrivastava et al., 2021). Consequently, customers and software developers will prefer a software development methodology that is flexible enough to deal with changes in requirements or circumstances. These criteria lend themselves to the agile software development methodology. However, resistance to change could keep decision makers from making the best decisions for software development methodologies for their programs.

There are many theories that articulate challenges to change. I used the DOI

theory by Rogers to articulate the resistance to change for government software development entities. Over the decades, DoD leadership has encouraged software development program decision makers to choose the waterfall methodology for their programs. (Grossi et al., 2014). Therefore, it will take time for these decision makers to overcome their resistance to changing from the waterfall to agile methodology. Unlike many other theories that articulate resistance to change, Rogers's (2003) inclusion of time as a variable in DOI theory is one of DOI's strengths. Rogers included the aspect of time in three instances:

1. The innovation-decision process – the time it takes for individuals to progress from their initial knowledge of the innovation to either their adoption or rejection of the innovation
2. The relative earliness or lateness that individuals adopt innovation
3. The rate of adopting innovation is measured by the number of members of the social system who adopt the innovation within a given period.

The first instance depicts the time it takes for government software development decision makers to move from waterfall to agile methodologies. Along with time, another element of DOI is social systems. Time and social systems depict decision makers' initial (or lack of) knowledge about waterfall and agile software development methodologies and the slow acceptance of agile in government software development.

A group of software developers created and proposed The Agile Manifesto in 2001 (Magistretti & Trabucchi, 2025). When the studied government software development entity began in 2015, many of the decision makers for government software

development may have had limited knowledge of agile software development because they were not exposed to the methodology in their undergraduate studies. Additionally, Costa and Walsh (2018) wrote that opinion leaders (and their authority decisions) generate the fastest DOI adoption rate. Since the opinion leaders in the government software development community had limited knowledge and exposure to the agile methodology, the community did not quickly accept agile software development. Thus, decision makers made decisions on software development based on familiarity and not a deliberate process. This indiscriminate manner of deciding on which software development methodology to use for projects greatly impacts the failure rates of those projects.

When deciding on the software development methodology for a project, decision makers need to analyze the requirements and conditions of the project. In their article, Mishra and Alzoubi (2023) created a simplistic decision tree to help program decision makers decide whether to use the waterfall or agile software development methodology. While this is a good start and emphasized the need for analysis to determine software development methodology, after gathering data for this study, I contended that decision makers must consider more factors when determining which software development methodology to use for their respective programs. I will provide more in-depth analysis of this concept later in this chapter.

In addition to using the minutes from the first 52 IPT meetings, I also used organizational charts to research the studied government software development entity. I used the first four and last four organizational charts of the studied government software

development entity. The reader can see a distinct contrast in turnover rate between the first four organizational charts and the last four organizational charts. However, there needs to be an understanding of turnover rates in order to make a determination on the consequences of the turnover rates produced.

As written earlier in this study, turnover is a reality in any organization. Additionally, some employee turnover is positive to an organization (Chung et al., 2021). An and Meier (2022) wrote that moderate levels of employee turnover can improve organizations by infusing new ideas from new employees and preventing rigidity in the organization; however, the cost of turnover can be significant and hurt organizations. Additionally, turnover in government agencies has a higher transactional cost than public agencies because of the government's hiring procedures and security concerns (An & Meier, 2022). So, turnover has to have a balance to be positive. Otherwise, employee turnover is a negative to an organization.

Between 2001 and 2021, the United States' overall turnover rate for organizations ranged between 20% and 30%, annually (Dogru et al., 2023). For the purpose of turnover rate, this percentage will serve as the optimal turnover rate for an organization. Analyzing the first four organizational charts, the reader can quickly determine a large turnover rate after each quarter. Specifically, there was a 10-20% turnover rate in the studied government software development entity each quarter, which is much higher than the optimal turnover rate of 20-30% each year. In contrast, the last four organizational charts depict an annual turnover rate of 10-20%, which is much more in line with the optimal turnover rate. The literature review can provide a reason as to why there was such a large

turnover rate at the beginning of the studied government software development entity than at its current state.

Limitations of the Study

My primary challenge to this study was bias. As a former team member of the studied government software development entity, I had a bias towards seeing the entity in a favorable manner. I may have wanted to confirm what I wanted to be true. Zhou and Shen (2021) wrote that this kind of confirmation bias can be a key factor in misinformation. To keep misinformation out of the study, I took steps to limit the confirmation bias in this study.

To avoid confirmation bias, it is important to understand what it is. Piksa and fellow authors (2024) defined confirmation bias as "the tendency to favor information that aligns with pre-existing beliefs or attitudes" (p. 1). As a former team member of the studied government software development entity, I would have a tendency to give more weight (i.e., favor) to information that supported any notions he had of the entity and the consequences of the hybrid approach this dissertation studied. After acknowledging this confirmation bias, I took steps within this study to mitigate his confirmation bias on the studied government software development entity and the hybrid approach.

Acknowledging the possible presence of confirmation bias is good. However, according to Li et al. (2025), acknowledging the possible presence of confirmation bias is not enough. To overcome confirmation bias, I focused the study on the data (see Blay et al., 2024). Additionally, I used multiple means of data collection in his study to avoid confirmation bias. While confirmation bias was a challenge to me, one challenge that I

anticipated was not as problematic as he thought it would be.

The limitations, challenges, and barriers did not come from gaining permission to research a DoD/Department of the Army (DA) program. I thought that obtaining permission to research such a program would involve much "red tape," which is a euphemism for rules and obligations that can hinder progress within government (see Madsen et al., 2022). However, that was not the case. According to emails, phone calls, and video chats with AHRPO, the process to receive permission to conduct the research on the studied government software development entity was as simple as getting permission from the person who had command authority of the program. Before starting this study, I gained permission from the first-line supervisor of the person who had command authority of the studied government software development entity. This person gave stipulations for the study that later gave AHROP and the person who had command authority over the program and AHRO comfort for me to conduct the study.

When I originally asked the first-line supervisor of the person who ultimately had command authority over the studied government software development entity, the first-line supervisor gave three guidelines for the study:

1. The interviewees could charge the time of the interviews to neither the government nor the contracts the interviewees worked on.
2. I could not use any output data directly from the training program the studied government software development entity produced.
3. I could not refer to any organizations I wrote of in the study by name.

After obtaining that person's permission, the person who had command authority

of the studied government software development entity readily gave me his permission to study the program. However, I was surprised by a stipulation from the IRB and AHRPO that made him make a slight change to the data gathering originally proposed by me.

Originally, I intended to use video and audio recordings of the interviews. I believed the video recordings of the interviews would allow him to go back and gauge body language and other non-verbal cues from the interviewees that would gain more insight into the data he gathered from the interviews (see De Villiers et al., 2022). However, both the Walden University IRB and AHRPO told me that I could not video tape the interviews. Instead, I could only use audio recordings of the interviews. To address this change in the face-to-face interviews, I intended to take notes during the interviews on non-verbal cues that I observed during the interviews. However, this was not the biggest limitation, challenge, or barrier to the study. The biggest limitation, challenge, or barrier to the study was something I thought would be a very easy task, but it was not.

It took me a bit longer than I anticipated to get Walden University and AHRPO concurrence to move forward with data collection. Along with not allowing the video recording of face-to-face interviews, the back and forth between me and the approval authority took multiple months instead of the few weeks I anticipated. However, when it was time to do the face-to-face interviews, scheduling, conducting, transcribing, and getting the interviewees to member check their respective interviews took multiple months instead of the few weeks I anticipated. The challenges and barriers to scheduling, conducting, transcribing and getting the interviewees to member check their respective

interviews stemmed from conflicts with time amongst the stakeholders (i.e., me and the interviewees).

The first challenge to the face-to-face interviews was scheduling a date, time, and location to conduct the interviews. This challenge stemmed from a lack of available time between me and interviewees (see Small & Cook, 2023). Specifically, all stakeholders in the interviews had competing requirements to their time to conduct the interviews. I had time availabilities that were not consistent. In kind, the current team members of the studied government software development entity had sprints, short development periods where software developers prioritize adding/creating features in the software with the highest customer impact, in their software development process (see Ogundipe et al., 2024). Additionally, finding areas away from the interviewees' work environment that were not too out of the way was another challenge. Due to all of these challenges, it took the research multiple months, instead of a couple of weeks to conduct, transcribe, and have the interviewees member check their respective interviews.

To address this challenge, in between interviews, I worked on analyzing and writing the other data collection for the study. When there was a lull in conducting the face-to-face interviews, I moved my focus to the data analysis of the first 52 IPT meetings and organizational charts. Along with saving time, this approach was advantageous because I was able to learn more about the MAXQDA analysis software before using it to analyze the transcripts of the face-to-face interviews.

Recommendations

Reference the research question as to what consequences in government software

development are fostered by a hybrid approach of the waterfall and agile methodologies, the answer from the research is clear. Each interviewee stated that the studied government software development entity used a hybrid approach of the waterfall and agile methodologies. Additionally, each interviewee stated that the studied government software development entity was successful. Therefore, the study suggests that using a hybrid approach of the waterfall and agile methodologies fosters positive consequences in government software development. DoD software development decision makers can refer to this case study to champion the use of the hybrid approach of the waterfall and agile methodologies in other government software development programs. However, it is up to the readers of this study to decide for themselves whether the results of this study can be applied to other settings (see Drisko, 2025). Another recommendation that arose from this study is how DoD decision makers should come to their decisions on which software development methodologies they should use for their respective programs.

Earlier in this chapter, I wrote of a researcher who used a decision tree that decision makers could use to determine whether to use the waterfall or agile software development methodology for a program. However, I concluded that this decision tree was too simplistic. Instead, I recommend that DoD software development decision makers ask a series of questions and let the answers to the following questions determine which software development methodology to use if choosing between the waterfall and agile methodologies.

1. What are the requirements for the program?
2. Will the requirements change? [If so, how often?]

3. Do the stakeholders desire customer feedback throughout the process?
4. 4)Is the program autonomous or does it fall within a “system of systems” architecture?
5. How knowledgeable, experienced, and mature is the team doing software development?

While the list of questions, above, is not all-inclusive, like the decision-tree created by Mishra and Alzoubi (2023), it does encourage DoD decision makers to take an analytical approach to deciding between waterfall and agile methodologies. Readers of this study can infer that leaders of programs taking a non-analytical approach to deciding on their programs’ software methodology could be one reason why most information technology projects fail.

Both the literature review and data collection support the concept that combining the waterfall and agile methodologies fosters positive impacts in software development. Many interviewees said that he hybrid approach is “the best of both worlds.” In other words, the hybrid approach highlights the best aspects of each methodology while mitigating their respective weaknesses.

People within the software development industry developed the waterfall and agile methodologies because they needed a better way to develop software than the standard practice for developing software at the time. Before the waterfall methodology, software developers used a “code and fix” method of software development. In this method, developers did not use much planning. Instead, they wrote code and fixed “bugs” in the software as they arose, which was not effective as software development projects

grew in complexity (Najihi et al., 2022). As a result, software developers created the waterfall methodology, which provided more structure. However, decades later, the waterfall methodology gave way to another industry standard in software development.

While the waterfall methodology improved software development from the "code and fix method," it had its detractors as well. The projects that used the waterfall methodology struggled to meet deadlines and deliver software that met their customers' needs because members of the team were not engaged much of the time, and the waterfall methodology did not allow for easy flexibility with requirements as conditions changed (Natarajan & Pichai, 2024). As a result, software developers created the agile methodology which kept all team members engaged and provided more flexibility with requirements.

The agile software development methodology was an improvement over the waterfall methodology. From the face-to-face interviews, all of the interviewees preferred the agile methodology to the waterfall methodology. Additionally, research shows that projects that use the agile software development methodology face fewer challenges than those that use the waterfall software development methodology (Mishra & Alzoubi, 2023). However, the agile software methodology has challenges of its own. Specifically, the agile methodology lacks adequate requirement engineering and requires an excessive amount of communication and collaboration (Gupta et al., 2022). To address these challenges with the agile software methodology, developers are gradually moving to a hybrid approach of the waterfall and agile methodologies.

A hybrid approach of the waterfall and agile software methodologies can increase

the success rate of projects of using either a waterfall or agile software development methodology. The hybrid approach incorporates the predictability of waterfall through extensive requirements development, upfront planning, compliance of documentation with the flexibility of agile through iterative development and continuous feedback from the customer, which yields higher project success rates through better risk management and improved collaboration amongst stakeholders (Luca, 2022). The success of the studied government software development entity supports this assertion.

Implications

Improving the success rate of software development entities provides a direct positive impact on social change. Many firms lose several million dollars a year due to the failures of their projects (Gupta et al., 2019). These financial losses can have grave impacts on these firms and the people that make up these firms. For example, such losses could lead to the failure of these firms. When firms fail, large numbers of people become unemployed. Their unemployment would directly affect these individuals and their respective families. Additionally, a large number of unemployed workers would stress the government through their need for government assistance in areas such as unemployment benefits as well as other entitlements that the government would provide. More stress on these government programs would also affect others in these programs and negatively impact their lives. However, the loss of several millions of dollars from these could have other impacts on society.

To better understand the impact of project failures, the reader needs to understand what a project is. Like other terms in this study, the term project has many meanings. For

the sake of this study, projects are temporary endeavors initiated by firms to create a product or service (Gupta et al, 2019). Firms initiate projects to address certain needs. When a project fails, for any reason, the need that the organization initiated for the project to address still remains unresolved. Consequently, project failures not only negatively affect the firms that initiated them. Project failures fail to properly address their intended challenge.

While such failures have an impact on the private sector, failures in the government can have even more far-reaching impacts on society. Like other terms I used in this study, government failure can have multiple meanings. For the sake of this study, government failure occurs when government actions do not achieve their intended outcomes or worsen the circumstances they were meant to improve and can negatively impact the economy, society, and the ability to govern (Government Failure: Causes and Implications in Public Choice, 2023). Along with those negative effects, government failure can erode its citizens' confidence in its government's ability to properly lead the people it governs as well as widen the social inequalities within society (Government Failure: Causes and Implications in Public, 2023). If most IT projects fail, one can assume that most government IT projects fail. For fiscal year 2020, the DoD reported that of the $37 billion it spent on IT projects, only 35% of the projects were within budget (Flyvbjerg et al., 2022).

In contrast, improving the success rate of government software development could increase public trust, lessen social inequalities, enable better decision-making on the part of the government, and improve the allocation of the government's resources. If

government software development had better requirements engineering and design, the government would be more equipped to engage and deal with societal issues in sectors like education.

For example, I developed a Microsoft (MS) Access database for the School Psychology sectors of several District of Columbia and Cincinnati, Ohio's public schools. This database tracked individual data and critical milestones for at-risk students and aided the school districts in remaining compliant with local and national guidelines. Not only did this database improve the quality of service to these at-risk students, but it also reduced lawsuits against the District of Columbia public school system, which saved money and allowed the schools to use that money more effectively. Additionally, the MS database increased the confidence of the team members within the District of Columbia public system and the students it served. Having the proper tools to effectively and efficiently perform the assigned tasks of the School Psychologist and other teachers and faculty increased morale with the District of Columbia schools that use the MS Access database (Khumalo, 2022). The MS Access database also garnered informal recognition from the leadership of the District of Columbia public school system.

Software development is gaining more importance and visibility with the government. Governments are increasing their use of information systems to deliver government services and improve their ability to govern (Nyansiro et al., 2021). As a result, the government's ability to properly engineer, design, implement, and sustain its information systems will gain more importance. With that in mind, DoD software development decision makers must make sound decisions on the software development

methodologies they choose for their respective programs. These sound decisions not only affect the success of the project; they affect how people view their government.

Conclusions

In response to the research question, a hybrid approach of the waterfall and agile methodologies fostered positive consequences in the product the studied government software development entity produced. Each person interviewed conveyed their perceived success of the project. Additionally, key members of the studied government software development entity received a certificate of appreciation from the customer for distinguishing themselves by the successful development and fielding of the software. This same customer was dissatisfied with the software they previously received from a civilian software development entity.

The purpose of this qualitative study was to explore the consequences in government software development fostered by a hybrid approach of the waterfall and agile methodologies as it pertains to the inherent shortcomings of each methodology. The success of the studied government software development entity depicts that a hybrid approach of the waterfall and agile methodologies could foster positive consequences for other government software development entities because the hybrid approach provides the ”best of both worlds [methodologies].” Specifically, the hybrid approach maintains the rigor of requirements engineering and design that the waterfall methodology provides while incorporating the flexibility and customer input of the agile methodology. However, another finding arose during the study.

The study revealed that DoD many software development decisions makers have

a bias towards the waterfall methodology, even though the interviewees and research reflect that the agile methodology is the better of the two. More importantly, the study shows the importance of having the requirements and conditions of project direct the choice of software development methodology, not the biases and inaccurate understandings of the decision maker. To help with this endeavor, I developed a series of questions to ask about the project to help decisions makers make the correct decision on software methodology.

References

Adeoye-Olatunde, O. A., & Olenik, N. L. (2021). Research and scholarly methods: Semi-structured interviews. *Journal of the American College of Clinical Pharmacy, 4*(10), 1358–1367. https://doi.org/10.1002/jac5.1441

Adler, R. H. (2022). Trustworthiness in qualitative research. *Journal of Human Lactation, 38*(4), 598–602. https://doi.org/10.1177/08903344221116620

Aghajani, M., Ruge, G., & Jugdev, K. (2023). An integrative review of project portfolio management literature: Thematic findings on sustainability mindset, assessment, and integration. *Project Management Journal, 54*(6), 629–650. https://doi.org/10.1177/87569728231172668

Agile and Scrum workshops start Jan. 31, 2019. (2019, January 16). https://www.lib.ncsu.edu/news/main-news/agile-and-scrum-workshops-start-jan-31-2019

Aguboshim, F. C. (2021). Adequacy of sample size in a qualitative case study and the dilemma of data saturation: A narrative review. *World Journal of Advanced Research and Reviews, 10*(3), 180–187. https://doi.org/10.30574/wjarr.2021.10.3.0277

Ahmed, S. K. (2024). The pillars of trustworthiness in qualitative research. *Journal of Medicine, Surgery, and Public Health, 2*, 100051. https://doi.org/10.1016/j.glmedi.2024.100051

Alam, M. K. (2020). A systematic qualitative case study: Questions, data collection, NVivo analysis and saturation. *Qualitative Research in Organizations and*

Management: An International Journal, 16(1), 1–31. https://doi.org/10.1108/qrom-09-2019-1825

Aldave, A., Vara, J. M., Granada, D., & Marcos, E. (2019). Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software, 157*, 110396. https://doi.org/10.1016/j.jss.2019.110396

Alharahsheh, H. H., & Pius, A. (2020). A review of key paradigms: Positivism vs interpretivism. *Global Academic Journal of Humanities and Social Sciences, 2*(3), 39–43.

Alhosani, K., & Alhashmi, S. M. (2024). Opportunities, challenges, and benefits of AI innovation in government services: a review. *Discover Artificial Intelligence, 4*(1), 18. https://doi.org/10.1007/s44163-024-00111-w

AlOmar, E. A., Mkaouer, M. W., & Ouni, A. (2021). Mining and managing big data refactoring for design improvement: Are we there yet? In *Knowledge management in the development of data-intensive systems*. 127–140. https://doi.org/10.1201/9781003001188

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies, 14*(11), 246–270. https://doi.org/10.3991/ijim.v14i11.13269

Al-Suraihi, W. A., Samikon, S. A., & Ibrahim, I. (2021). Employee turnover causes, importance and retention strategies. *European Journal of Business and Management Research, 6*(3). 1–10. https://doi.org/10.24018/ejbmr.2021.6.3.893

Amajuoyi, P., Benjamin, L. B., & Adeusi, K. B. (2024). Agile methodologies: Adapting product management to rapidly changing market conditions. *GSC Advanced Research and Reviews, 19*(2), 249–267. https://doi.org/10.30574/gscarr.2024.19.2.0181

Amelia, L., & Hartanti, D. (2023). Examining effectiveness of management control system within Indonesian anti-corruption agency: A levers of control perspective. *Scientific Journal of Management, Economics, and Accounting, 7*(2), 821–845. https://doi.org/10.31955/mea.v7i2.3068

Amini, M., & Jahanbakhsh Javid, N. (2023). A multi-perspective framework established on diffusion of innovation (DOI) theory and technology, organization and environment (TOE) framework toward supply chain management system based on cloud computing technology for small and medium enterprises. *International Journal of Information Technology and Innovation Adoption, 11*, 1217–1234.

Amin, M. E. K., Nørgaard, L. S., Cavaco, A. M., Witry, M. J., Hillman, L., Cernasev, A., & Desselle, S. P. (2020). Establishing trustworthiness and authenticity in qualitative pharmacy research. *Research in Social and Administrative Pharmacy, 16*(10), 1472–1482. https://doi.org/10.1016/j.sapharm.2020.02.005

An, S. H., & Meier, K. J. (2022). Optimal turnover rates and performance in public organizations: Theoretical expectations. *Public Performance & Management Review, 45*(3), 582-604. https://doi.org/10.1080/15309576.2022.2059769

Andrei, B., Casu-Pop, A., Gheorghe, S., & Boiangiu, C. (2019). A study on using waterfall and agile methods in software project management. *Journal of*

Information Systems & Operations Management, 125–135. https://www.proquest.com/scholarly-journals/study-on-using-waterfall-agile-methods-software/docview/2237828314/se-2

Andrews, C. J. (2023). The greening of America's building codes: Promises and paradoxes. *Journal of the American Planning Association, 89*(3), 406–407. https://doi.org/10.1080/01944363.2023.2174751

Aroral, H. K. (2021). Waterfall process operations in the fast-paced world: Project management exploratory analysis. *International Journal of Applied Business and Management Studies*, *6*(1), 91–99.

Asenahabi, B. M. (2019). Basics of research design: A guide to selecting appropriate research design. *International Journal of Contemporary Applied Researches*. *6*(5), 76–89.

Ayyagari, M. R., & Atoum, I. (2019). CMMI-DEV implementation simplified - A spiral software model. *International Journal of Advanced Computer Science and Applications*, *10*(4), 445–450.

Baham C, & Hirschheim R. (2022). Issues, challenges, and a proposed theoretical core of agile software development research. *Information Systems Journal, 32*(1), 103–129. https://doi.org/10.1111/isj.12336

Ball, J., Ogletree, R., Asunda, P., Miller, K., & Jurkowski, E. (2014). Diffusion of innovation elements that influence the adoption and diffusion of distance education in health. *American Journal of Health Studies*, 29(3), 240–246. https://doi.org/10.47779/ajhs.2014.221

Basch, J. M., Melchers, K. G., Kurz, A., Krieger, M., & Miller, L. (2021). It takes more than a good camera: Which factors contribute to differences between face-to-face interviews and videoconference interviews regarding performance ratings and interviewee perceptions? *Journal of Business and Psychology*, *36*, 921–940.

Bazen, A., Barg, F. K., & Takeshita, J. (2021). Research techniques made simple: An introduction to qualitative research. *Journal of Investigative Dermatology*, *141*(2), 241–247. https://doi.org/10.1016/j.jid.2020.11.029

Belcher, A. M. (2019). Perceptions of High School Mathematics Teachers on Situated Learning and Authentic Assessment: A Case Study (Doctoral dissertation, Northcentral University).

Beerbaum, D. (2020). Applying agile methodology to regulatory compliance projects in the financial industry: A case study research. *Journal of Applied Research Review*, 1–11. https://doi.org/10.2139/ssrn.3834205

Bianchi, M. J., Conforto, E. C., Rebentisch, E., Amaral, D. C., Rezende, S. O., & de Padua, R. (2021). Recommendation of project management practices: A contribution to hybrid models. *IEEE Transactions on Engineering Management*, *69*(6), 3558–3571. https://doi.org/10.1109/TEM.2021.3101179

Bibhunandini, D. (2022). Diffusion of innovations: Theoretical perspectives and empirical evidence. *African Journal of Science, Technology, Innovation and Development*, *14*(1), 94–103. https://doi.org/10.1080/20421338.2020.1814517

Blay, M., Gauld, C., Espi, P., & Falissard, B. (2024). Confirmation bias and quantitative approach in psychiatry: should ideological competing interests be declared?

Frontiers in Psychiatry, *15*, 1–5. 1365733.

Brahmantara, R. P., Hutapea, D. Y., & Ruldeviyani, Y. (2021, September 22–23). Evaluation on data operations management using CMMI and DMBOK: BDPJN case study [Paper presentation]. *2021 9th International Conference on Cyber and IT Service Management (CITSM)*, Bengkulu, Indonesia, pp. 1–6. https://doi.org/10.1109/CITSM52892.2021.9588888

Breen, R.L. (2007). A Practical Guide to Focus-Group Research. *Journal of Geography in Higher Education*, *30*(3), 463–475. https://doi.org/10.1080/03098260600927575

Çakıroğlu, Ü., Saylan, E., Çevik, İ., Mollamehmetoğlu, M. Z., & Timuçin, E. (2022). Faculty adoption of online teaching during the Covid-19 pandemic: A lens of diffusion of innovation theory. *Australasian Journal of Educational Technology*. *38*(3), 87–103. https://doi.org/10.14742/ajet.7307

Call, D. R., & Herber, D. R. (2022). Applicability of the diffusion of innovation theory to accelerate model-based systems engineering adoption. *Systems Engineering*, *25*(6), 574–583. https://doi.org/10.1002/sys.21638

Campbell, S., Greenwood, M., Prior, S., Shearer, T., Walkem, K., Young, S., Bywaters, D. & Walker, K. (2020). Purposive sampling: Complex or simple? Research case examples. *Journal of Research in Nursing*, *25*(8), 652–661. https://doi.org/10.1177/1744987120927

Chatziamanetoglou, D., & Rantos, K. (2023). Blockchain-based security configuration management for ICT systems. *Electronics*, *12*(8), 1–18.

https://doi.org/10.3390/electronics12081879

Chen, M., Yao, W., Chen, J., Liang, H., Chen, Y., Qiao, H., Yang, C., Li, M., & Tong, J. (2022, April 22-24). Critical challenges and solutions for an ultra-large-scale enterprise DevOps platform [Paper presentation]. *2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, Chengdu, China, 167–171. https://doi.org/10.1109/ICCCBDA55098.2022.9778937

Chintami, A. S., & Astriratma, R. (2021, October 28-29). An information system design of web-based student learning outcomes assessment at Pekayon Jaya VI Elementary School [Paper presentation]. *2021 International Conference on Informatics, Multimedia, Cyber and Information Systems (ICIMCIS)*, Jakarta, Indonesia, pp. 375–380. https://doi.org/10.1109/ICIMCIS53775.2021.9699237

Chung, H., Quan, W., Koo, B., Ariza-Montes, A., Vega-Muñoz, A., Giorgi, G., & Han, H. (2021). A threat of customer incivility and job stress to hotel employee retention: do supervisor and co-worker supports reduce turnover rates?. *International Journal of Environmental Research and Public Health*, *18*(12), 1–18.

Cirus, L., & Simonova, I. (2020). Rogers' diffusion of innovation theory applied on primary education: A case study of Czech teachers. *2020 International Symposium on Educational Technology (ISET)*. 33–37. https://doi.org/10.1109/ISET49818.2020.00017

Collins, C., & O'Riordan, R. (2022). Data triangulation confirms learning in the zoo environment. *Environmental Education Research*, *28*(2), 295–317.

https://doi.org/10.1080/13504622.2021.1974351

Costa, L., & Walsh, G. (2018). Nova Southeastern University: A diffusion of innovation analysis of distance education. *Distance Learning*, *15*(4), 57–63. https://eric.ed.gov/?id=EJ1298289

Dagnino, A., Kolomycki, M., & Kucheria, A. (2022, August 15-18). MAP: Design, development, deployment, and maintenance of Industrie 4.0 AI applications. [Paper presentation]. *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService)*, Newark, CA, USA. p. 108–113. https://doi.org/10.1109/BigDataService55688.2022.00024

Daher, W. (2023). Saturation in qualitative educational technology research. *Education Sciences*, *13*(2), 98. https://doi.org/10.3390/educsci13020098

Davies, L., LeClair, K. L., Bagley, P., Blunt, H., Hinton, L., Ryan, S., & Ziebland, S. (2020). Face-to-face compared with online collected accounts of health and illness experiences: a scoping review. *Qualitative Health Research*, *30*(13), 2092–2102. https://doi.org/10.1177/1049732320935835

Dearing, J. W., & Cox, J. G. (2018). Diffusion of innovations theory, principles, and practice. *Health Affairs*, *37*(2), 183–190. https://doi.org/10.1377/hlthaff.2017.1104

Demirci, J. (2023). About Research: Conducting Better Qualitative Interviews. Journal of *Human Lactation*, *40*(1), 21–24. https://doi.org/10.1177/08903344231213651

De Villiers, C., Farooq, M. B., & Molinari, M. (2022). Qualitative research interviews using online video technology–challenges and opportunities. *Meditari*

Accountancy Research, *30*(6), 1764–1782.

Diansyah, A. F., Rahman, M. R., Handayani, R., Cahyo, N., Diffran, D., & Utami, E. (2023). Comparative analysis of software development lifecycle methods in software development: A systematic literature review. *International Journal of Advances in Data and Information Systems*, *4*(2), 97–106.

Digital Village. (2021). *70% of software development projects fail. Yours doesn't need to be one of them*. https://www.linkedin.com/pulse/70-software-development-projects-fail-yours-doesnt-/

Dogru, T., McGinley, S., Sharma, A., Isık, C., & Hanks, L. (2023). Employee turnover dynamics in the hospitality industry vs. the overall economy. *Tourism Management*, *99*, 1–7.

Drisko, J. W. (2025). Transferability and generalization in qualitative research. *Research on Social Work Practice*, *35*(1), 102–110.

Dutta, S., & Omolayole, O. (2021). Impact of vendor CMMI capability on user system success perception and its interaction with trust in an IT outsourcing project. *American Journal of Management*, 21(3), 102–116.

Dwivedi, N., Katiyar, D., & Goel, G. (2022). A comparative study of various software development life cycle (SDLC) models. *International Journal of Research in Engineering, Science and Management*. *5*(3), 141–144.

Dyar, K. L. (2022, January). Qualitative inquiry in nursing: Creating rigor. *Nursing Forum*, *57*(1), 187–200. https://doi.org/10.1111/nuf.12661

Ebule, A. (2025). Agile vs. waterfall: A Comprehensive analysis of methodologies for

effective project management. *International Journal of Novel Research and Development*, *10*(1), 273.

Enworo, O. C. (2023). Application of guba and lincoln's parallel criteria to assess trustworthiness of qualitative research on indigenous social protection systems. *Qualitative Research Journal*, 372–284. https://doi.org/10.1108/qrj-08-2022-0116

Evelin, Suclinton, Kwa, R. V., & Anggreainy, M. S. (2021, August 02-04). Developing an application to hire a private tutor using the scrum method [Paper presentation]. *2021 International Conference on ICT for Smart Society (ICISS)*, Bandung, Indonesia, pp. 1–6. https://doi.org/10.1109/ICISS53185.2021.9532506

Fagarasan, C., Popa, O., Pisla, A., & Cristea, C. (2021, August). Agile, waterfall, and iterative approach in information technology projects [Paper presentation]. *In IOP Conference Series: Materials Science and Engineering*, 1169 (1), p. 012025. IOP Publishing. https://doi.org/10.1088/1757-899X/1169/1/012025

Flyvbjerg, B., Budzier, A., Lee, J. S., Keil, M., Lunn, D., & Bester, D. W. (2022). The empirical reality of IT project cost overruns: discovering a power-law distribution. *Journal of Management Information Systems*, *39*(3), 607–639.

Fontana, R.M & Marczak, S. (2020). Characteristics and challenges of agile software development adoption in Brazilian government. *Journal of Technology Management & Innovation*, *15*(2), 3–10. https://doi.org/10.4067/S0718-27242020000200003

Freitas, M. B., Araújo, V. M., & Magalhães, J. P. (2023, January 23-25). Process SDLC-GDPR: Towards the development of secure and compliant applications [Paper

presentation]. *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, Jeddah, Saudi Arabia, pp. 1–6. https://doi.org/10.1109/ICAISC56366.2023.10085308

Fuah Siti Mas, & Ganggi Roro Isyawati Permata. (2022, October 31). The diffusion of innovations elements in library science journals [Paper presentation]. *E3S Web of Conferences, Online*, pp. 359. https://doi.org/10.1051/e3sconf/202235903019

Gabarre, C., & Gabarre, S. (2020). Trustworthiness in sampling selection: Remedies against introspective chaos. *The Qualitative Report*, *25*(12), 4352–4375. https://doi.org/10.46743/2160-3715/2020.3670

Gaborova, M., Karuovicb, D., Kavalicc, M., Radosavc, D., Milosavljevc, D., Stanisaljevd, S., & Bushatidž J. (2021). Comparative analysis of agile and traditional methodologies in IT project management. *Journal of Applied Technical and Educational Sciences*, *11*(4), 1–24. https://doi.org/10.24368/jates.v11i4.279

García-Avilés, J. A. (2020). Diffusion of innovation. *The International Encyclopedia of Media Psychology*, 1–8. https://doi.org/10.1002/9781119011071.iemp0137

Garg, A., Kaliyar, R. K., & Goswami, A. (2022, March 25-26). PDRSD-A systematic review on plan-driven SDLC models for software development [Paper presentation]. *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, p. 739–744. https://doi.org/10.1109/ICACCS54159.2022.9785261

Gazzawe, F. H. (2019). Requirement artefacts: Finding the missing link [Unpublished

doctoral dissertation]. Loughborough University

Ghena, M. F., & Ghiculescu, L. D. (2023). Applicability of waterfall and agile methodologies. *FAIMA Business & Management Journal*, *11*(4), 55–65.

Gheorghe, A. M., Gheorghe, I. D., & Iatan, I. L. (2020). Agile software development. *Informatica Economica*, *24*(2/2020), 90–100. https://doi.org/10.24818/issn14531305/24.2.2020.08

Goda, G. S., Jackson, E., Nicholas, L. H., & Stith, S. S. (2023). The impact of Covid-19 on older workers' employment and Social Security spillovers. *Journal of population economics*, *36*(2), 813–846.

Goldstein, S. L. (1989). The search for Nova University: An essay on its first twenty-five years 1964–1989. Fort Lauderdale, FL: Nova Southeastern University

Government Failure: Causes and Implications in Public Choice • Public Administration Notes by PubAdmin.Institute. (2023, December 24). Public Administration Institute. https://pubadmin.institute/perspectives-on-public-administration/government-failure-causes-implications

Grant, C. & Osanloo, A. (2014). Understanding, selecting, and integrating a theoretical framework in dissertation research: Creating the blueprint for your house. *Administrative Issues Journal: Connecting Education, Practice, and Research*. *4*(2), 12–26.

Gray, L. M., Wong-Wylie, G., Rempel, G. R., & Cook, K. (2020). Expanding qualitative research interviewing strategies: Zoom video communications. *The Qualitative Report*, *25*(5), 1292–1301. https://doi.org/10.46743/2160-3715/2020.4212

Grossi, L., Calvo, M. J. A., & San Feliu, T. (2014). High-maturity levels: Achieving CMMI ML-5 in a consultancy company. *Journal of Software: Evolution & Process*, *26*(9), 808–817. https://doi.org/10.1002/smr.1666

Grover, P., Kar, A. K., & Janssen, M. (2019). Diffusion of blockchain technology: Insights from academic literature and social media analytics. *Journal of Enterprise Information Management*, *32*(5), 735–757. https://doi.org/10.1108/JEIM-06-2018-0132

Guest, G., Bunce, A., & Johnson, L. (2006, February). How many interviews are enough? An experiment with data saturation and variability. *Field Methods*, *18*(1), 59–82. https://doi.org/10.1177/1525822X05279903

Guetterman, T. C., & James, T. G. (2023). A software feature for mixed methods analysis: The MAXQDA Interactive Quote Matrix. *Methods in Psychology*, *8*, 1–9. 100116.

Gupta, A., Poels, G., & Bera, P. (2022). Using conceptual models in agile software development: A possible solution to requirements engineering challenges in agile projects. *IEEE Access*, *10*, 119745–119766.

Gupta, S., & Gayathri, N. (2022, November 18-19). Study of the software development life cycle and the function of testing [Paper presentation]. *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*, Bengaluru, India, pp. 1270–1275. https://ieeexplore.ieee.org/document/10060231

Gupta, S. K., Gunasekaran, A., Antony, J., Gupta, S., Bag, S., & Roubaud, D. (2019). Systematic literature review of project failures: Current trends and scope for

future research. *Computers & Industrial Engineering*, *127*, 274–285.

Hari, H., Iyer, R., & Sampat, B. (2021). Customer brand engagement through chatbots on bank websites–Examining the antecedents and consequences. *International Journal of Human–Computer Interaction*, *38*(13), 1–16. https://doi.org/10.1080/10447318.2021.1988487

Hasibuan, R.T. & Zakir, S. (2021). Implementation of the brute force algorithm in the design of android-based thesis catalogs. *International Journal of Knowledge*, *1*(2), 148–161. https://doi.org/10.30983/ijokid.v1i2.5042

Henriquez, V., Calvo-Manzano, J. A., Moreno, A. M., & San Feliu, T. (2022). Agile-CMMI V2. 0 alignment: Bringing to light the agile artifacts pointed out by CMMI. *Computer Standards & Interfaces*, *82*, 1–12. https://doi.org/10.48550/arXiv.2110.00399

Hema, V., Thota, S., Kumar, S. N., Padmaja, C., Krishna, C. B. R., & Mahender, K. (2020, October 9-10). Scrum: An effective software development agile tool [Paper presentation]. *IOP Conference Series: Materials Science and Engineering*, Warangal, India, *981*(2), 022060. https://doi.org/10.1088/1757-899X/981/2/022060

Henoch J. C., & Yerik A. S. (2023). Analysis and design of student guidance information system through software development life cycle (SDLC) and waterfall model. *Journal of Information Systems and Informatics*, *5*(1), 259–270. https://doi.org/10.51519/journalisi.v5i1.443

Heriyanti, F., & Ishak, A. (2020, October 17). Design of logistics information system in

the finished product warehouse with the waterfall method: Review literature [Paper presentation]. *IOP Conference Series: Materials Science and Engineering*, Medan, Indonesia, *801*(1), p. 012100. https://doi.org/10.1088/1757-899X/801/1/012100

Herodotou, C., Rienties, B., Hlosta, M., Boroowa, A., Mangafa, C., & Zdrahal, Z. (2020). The scalable implementation of predictive learning analytics at a distance learning university: Insights from a longitudinal case study. *The Internet and Higher Education*, *45*, 1–13. 100725. https://doi.org/10.1016/j.iheduc.2020.100725

Hao, L. (2022). Optimal allocation of human resource structure based on capability maturity model integration. *Security and Communication Networks*, 1–11. https://doi.org/10.1155/2022/3162328

Horton, S. (2020). Are software security issues a result of flaws in software development methodologies? (Publication No. 27962083) [Masters dissertation, Utica College. ProQuest Dissertations.

Hoti, D., Maloku, M., & Gashi, K. (2023). SDLC Phases of a Mobile Application. *In Designing and Developing Innovative Mobile Applications. IGI Global*, 232–249. https:/doi.org/10.4018/978-1-6684-8582-8.ch013

Hou, L., Liu, Q., Saeed, K., Ali Haidery, S., Uddin, M. I., & Khattak, H. (2021). Enhancement of the capability maturity model for improving the quality of software projects in developing countries. *Scientific Programming*, *2021*, 1–10. https://doi.org/10.1155/2021/9982227

Hua, L., Zem, L., & Neto, R. D. A. P. (2023). Choosing the best project management

methodology for research and development projects: Agile, waterfall, or hybrid?. *Revista Foco*, *16*(11), e3336-e3336.

Hui, L. K., Machap, K., & Arun, K. C. (2020). Bandwidth management system to monitor the internet connection. *Journal of Applied Technology and Innovation*, *4*(2), 1–4

Ingley, C., Singh, S., & Malkani, A. (2020). Teasing out fieldwork challenges with e-mail methods in cross-border research. *Qualitative Research in Organizations and Management: An International Journal*, *15*(3), 402–418. https://doi.org/10.1108/QROM-02-2017-1490

Iqbal, M., & Zahidie, A. (2022). Diffusion of innovations: a guiding framework for public health. *Scandinavian Journal of Public Health*, *50*(5), 533–537. https://doi.org/10.1177/14034948211014104

Irani, E. (2019). The use of videoconferencing for qualitative interviewing: Opportunities, challenges, and considerations. *Clinical Nursing Research*, *28*(1), 3–8. https://doi.org/10.1177/1054773818803170

Iriarte, C., & Bayona, S. (2020). IT projects success factors: a literature review. *International Journal of Information Systems and Project Management*, *8*(2), 49–78. https://doi.org/10.12821/ijispm080203

Isaacs, T. (2022). Agile Teaching and the Agile Manifesto [Paper presentation]. *Pedagogicon Conference Proceedings*, p. 1–11. https://encompass.eku.edu/pedagogicon/2021/buckleup/2

Ismail, S., & Dawoud, D. W. (2022, January 26-29). Software development models for

IoT [Paper presentation]. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, Nevada, USA, pp. 524–530. https://doi.org/10.1109/CCWC54503.2022.9720750

Jain, N. (2021). Survey versus interviews: Comparing data collection tools for exploratory research. *The Qualitative Report*, *26*(2), 541–554. https://doi.org/10.46743/2160-3715/2021.4492

Jalaja, P., & Adilakshmi, T. (2023). Automated change impact analysis tool for software maintenance phase. *Journal of Theoretical and Applied Information Technology*, *101*(7), 2794–2805.

Jadoon, G., Shafi, M., & Jan, S. (2019, December 03-05). A model-oriented requirements traceability framework for small and medium software industries. *2019 International Arab Conference on Information Technology (ACIT)*, Al Ain, United Arab Emirates, pp. 91–96. https://doi.org/10.1109/ACIT47987.2019.8991116

Jamous, N., Garttan, G., Staegemann, D., & Volk, M. (2021, August). Hybrid Project Management Methods Efficiency in IT Projects. *In Americas Conference on Information Systems (AMCIS)*, 1–10.

Jangra, K., Yadav, S., & Malhotra, V. (2023). Optimizing the product development process for speed to market, quality, and customer satisfaction: A comparative study of agile and waterfall methodologies. *International Research Journal of Modernization in Engineering Technology and Science*, *5*(4), 4542–4548. https://doi.org/10.56726/IRJMETS36854

Johnson, M. (2022). Turning teams of champions into championship teams: A thematic analysis of identity entrepreneurship in college football. *Rollin College. Psychology Department*. https://scholarship.rollins.edu/cgi/viewcontent.cgi?article=1176&context=honors

Johnson, J. L., Adkins, D., & Chauvin, S. (2020). A review of the quality indicators of rigor in qualitative research. *American Journal of Pharmaceutical Education*, *84*(1), 138–146. https://doi.org/10.5688/ajpe7120

Jierasup, S., & Leelasanthitham, A. (2024). A change from negative to positive of later adoption using the innovation decision process to imply sustainability for HR chatbots of private companies in Thailand. Sustainability, *16*(13), 1–29. https://doi.org/10.3390/su16135641

Kandel, B. (2020). Qualitative versus quantitative research. *Journal of Product Innovation Management*, *32*(5), 1–5.

Kayesa, N. K., & Shung-King, M. (2021). The role of document analysis in health policy analysis studies in low and middle-income countries: Lessons for HPA researchers from a qualitative systematic review. *Health Policy*, *2*, 1–3. https://doi.org/10.1016/j.hpopen.2020.100024

Khan, I. A., & Kumari, D. (2021). The role of the analysis phase of SDLC for small scale business application-A review. *International Journal Of Humanities, Engineering, Science And Management*, *2*(1), 63–75. https://journal.rkdfuniversity.org/index.php/ijhesm/article/view/82

Khan, M., Ali, I., Mehmood, W., Nisar, W., Aslam, W., Shafiq, M., & Choi, J. G. (2021).

CMMI-compliant modernization framework to transform legacy systems. *Intelligent Automation & Soft Computing*, *27*(2), 311–331. https://doi.org/10.32604/iasc.2021.014280

Khan, S., & Mahadik, S. (2022). A comparative study of agile and waterfall software development methodologies. *International Journal of Advanced Research in Science, Communication and Technology*, *2*(1), 399–402.

Knott, E., Rao, A. H., Summers, K., & Teeger, C. (2022). Interviews in the social sciences. *Nature Reviews Methods Primers*, *2*(73), 1–15. https://doi.org/10.1038/s43586-022-00150-6

Kakar, A. & and Kakar, A. (2020). A brief history of software development and manufacturing. *Association for Information Systems Electronic Library (AISeL)*, 1–7. https://aisel.aisnet.org/sais2020/4

Kakar, A. K. (2023). A rhetorical analysis of the agile manifesto on its 20th anniversary. *The Journal of the Southern Association for Information Systems*, *10*(1), 20–29. https://doi.org/10.17705/3JSIS.00030

Kang, E., & Hwang, H. J. (2021). Ethical conducts in qualitative research methodology: Participant observation and interview process. *Journal of Research and Publication Ethics*, *2*(2), 5–10.

Khumalo, N. (2022). Factors That Affect the Morale of Employees in the Institution of Higher Learning in South Africa. *International Journal of Higher Education*, *11*(3), 58–66.

Kiker, D. S., Callahan, J. S., & Kiker, M. B. (2019). Exploring the boundaries of servant

leadership: A meta-analysis of the main and moderating effects of servant leadership on behavioral and affective outcomes. *Journal of Managerial Issues*, *31*(2), 172–197. https://www.proquest.com/scholarly-journals/exploring-boundaries-servant-leadership-meta/docview/2319666235/se-2

Krishna, S., & Sreekanth, S. (2016). Performance of ten software development process models with principles. *International Journal of Science, Engineering and Computer Technology*, *6*(5), 161–171.

Lalmi, A., Fernandes, G., & Boudemagh, S.S. (2022). Synergy between traditional, agile and lean management approaches in construction projects: Bibliometric analysis. *Procedia Computer Science*, *196*, 732–739. https://doi.org/10.1016/j.procs.2021.12.070

Lavee, E., & Itzchakov, G. (2023). Good listening: A key element in establishing quality in qualitative research. *Qualitative Research*, *23*(3), 614–631. https://doi.org/10.1177/14687941211039402

Levine, T. R. (2022). Truth-default theory and the psychology of lying and deception detection. *Current Opinion in Psychology*, *47*, 1–4.

Lim, W. M. (2025). What is qualitative research? An overview and guidelines. *Australasian Marketing Journal*, *33*(2), 199–229.

Li, S., Davidson, T. J., Xiong Bearfield, C., & Wall, E. (2025, April). Confirmation Bias: The Double-Edged Sword of Data Facts in Visual Data Communication. *In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3706598.3713831

Li, Y., Huang, Z., Dineen, B., Wang, M., & van Jaarsveld, D. (2025). Voluntary turnover rate fluctuations, human resource practices, and innovation: A within-organization investigation. *Personnel Psychology*, *78*(1), 103–122. https://doi.org/10.1111/peps.12661

Lobo Marques, J. A., Bragança dos Reis Morais, J. J., Alves, J., & Gonçalves, M. (2023). Effectiveness analysis of waterfall and agile project management methodologies - A case study from Macau's construction industry. *Journal of Management Analysis, 12*(1), 23–38. https://doi.org/10.12662/2359-618xregea.v12i1.p23-38.2023

Luca, C. (2022). Hybrid Methodologies: Integrating waterfall and agile approaches. *Journal of Project Management Research*, *12*(6), 1–19.

Luo, W. H., & Chen, Y. C. (2022). Constructing a teaching capability maturity model for content and language-integrated learning teachers in Taiwan. *Humanities and Social Sciences Communications*, *9*(1), 1–10. https://doi.org/10.1057/s41599-021-00928-1

Machado, E. M. M., Mexas, M. P., Meza, L. A., & de Oliveira, I. A. M. (2022). Implementation processes of quality multi-models: ISO 9001, CMMI-DEV ML3, and MPT. BR ML3 in small and medium-sized enterprises (SMEs) of information technology. *Economic and Social Development: Book of Proceedings*, 29–38.

Mackieson, P., Shlonsky, A., & Connolly, M. (2019). Increasing rigor and reducing bias in qualitative research: A document analysis of parliamentary debates using applied thematic analysis. *Qualitative Social Work*, *18*(6), 965–980.

https://doi.org/10.1177/1473325018786996

Magistretti, S., & Trabucchi, D. (2025). Agile-as-a-tool and agile-as-a-culture: a comprehensive review of agile approaches adopting contingency and configuration theories. *Review of Managerial Science*, *19*(1), 223–253.

Mannan, M.A. (2024). Data collection and saturation in exploring the public sector accounting in Bangladesh. *International Center for Research and Resource Development (ICRRD) Journal*, *5*(1), 179–197.

Madsen, J. K., Mikkelsen, K. S., & Moynihan, D. P. (2022). Burdens, sludge, ordeals, red tape, oh my!: a user's guide to the study of frictions. *Public Administration*, *100*(2), 375–393.

Menzli, L. J., Smirani, L. K., Boulahia, J. A., & Hadjouni, M. (2022). Investigation of open educational resources adoption in higher education using Rogers' diffusion of innovation theory. *Heliyon*, *8*(7), 1–12. https://doaj.org/article/e3bde8b3aeb443db991e95c870f4dd2c

Mirsaeedi, E., & Rigby, P. C. (2020). Mitigating turnover with code review recommendation. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 1183–1195. https://doi.org/10.1145/3377811.3380335

Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, *14*(4), 1504–1522.

Mondal, H., Mondal, S., & Podder, I. (2023). Using ChatGPT for writing articles for patients' education for dermatological diseases: A pilot study. *Indian*

Dermatology Online Journal, *14*(4), 482–486.

Morgan, H. (2024). Using triangulation and crystallization to make qualitative studies trustworthy and rigorous. *The Qualitative Report*, *29*(7), 1844–1856.

Moriña, A. (2020). Using life history with students with disabilities: researching with, rather than research. *Educational Review*, *72*(6), 770–784. https://doi.org/10.1080/00131911.2018.1550054

Moyano, C. G., Pufahl, L., Weber, I., & Mendling, J. (2022). Uses of business process modeling in agile software development projects. *Information and Software Technology*, *152*, 1–15. 107028. https://doi.org/10.1016/j.infsof.2022.107028

Munthe-Kaas, H., Nøkleby, H., Lewin, S., & Glenton, C. (2020). The TRANSFER Approach for assessing the transferability of systematic review findings. *BMC Medical Research Methodology*, *20*(1), 1–22. https://doi.org/10.1186/s12874-019-0834-5

Nababan, D., Manurung, T., & Sianipar, E. V. (2024). Work engagement relationship with nurse intention turnover at Portsea hospital. *International Journal of Health Engineering and Technology (IJHET)*, *2*(6), 209–216. https://doi.org/10.55227/ijhet.v2i6.190

Najihi, S., Elhadi, S., Ait Abdelouahid, R., & Marzak, A. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, *203*, 775–782.

Nassaji, H. (2020). Good qualitative research. *Language Teaching Research*, *24*(4), 427–431. https://doi.org/10.1177/1362168820941288

Natarajan, T., & Pichai, S. (2024). Transition from waterfall to agile methodology: An

action research study. *IEEE access*, *12*, 49341–49362.

Natow, R. S. (2020). The use of triangulation in qualitative studies employing elite interviews. *Qualitative Research*, *20*(2), 160–173. https://doi.org/10.1177/1468794119830077

Ngongoni, C. N., Wasswa, W., Makubalo, L., Moeti, M., & Chibi, M. (2022). Towards a healthcare innovation scaling framework-The voice of the innovator. *International Journal of Environmental Research and Public Health*, *19*(23), 1–16. https://doi.org/10.3390/ijerph192315515

Nunan, D. (2020). Building and maintaining trust in research. *International Journal of Market Research*, *62*(4), 393–395. https://doi.org/10.1177/147078532094

Nunan, J., Stanier, I., Milne, R., Shawyer, A., Walsh, D., & May, B. (2022). The impact of rapport on intelligence yield: Police source handler telephone interactions with covert human intelligence sources. *Psychiatry, Psychology and Law*, *29*(1), 1–19. https://doi.org/10.1080/13218719.2020.1784807

Nurimansjah, R. A., Kardini, N. L., & Idrus, S. (2023). The role of work stress and working environment on employee turnover intention in mining company. *JEMSI (Jurnal Ekonomi, Manajemen, dan Akuntansi)*, *9*(2), 339–343.

Nwachukwu, A.S. & Boatengu, K.E. (2024). The role of agile methodologies in modern software development: A comprehensive analysis of impact, challenges, and best practices. *Journal of Business & IT*, 41–49. https://doi.org/10.14311/bit.2024.02.05

Nyansiro, J. B., Mtebe, J. S., & Kissaka, M. M. (2021). E-government information

systems (IS) project failure in developing countries: Lessons from the literature. *The African Journal of Information and Communication*, *28*, 1–29.

Office of Public Affairs Homeland Security. (n.d.). www.dhs.gov. https://www.dhs.gov/office-public-affairs

Ogundipe, D. O., Odejide, O. A., & Edunjobi, T. E. (2024). Agile methodologies in digital banking: Theoretical underpinnings and implications for customer satisfaction. *Open Access Research Journal of Science and Technology*, *10*(02), 021–030.

Olmos-Vega, F. M., Stalmeijer, R. E., Varpio, L., & Kahlke, R. (2023). A practical guide to reflexivity in qualitative research: AMEE Guide No. 149. *Medical Teacher*, *45*(3), 241–251. https://doi.org/10.1080/0142159X.2022.2057287

Olshevska, O., Zinchenko, I., Sakaliuk, O., Kozub, O., & Kolesnyk, V. (2023). Application of agile principles in the organization of service units of higher education institutions. Part 1. *Automation of Technological & Business Processes*, *15*(3), 4–9. https://doi.org/10.15673/atbp.v15i3.2619

Otero, T. F., Barwaldt, R., Topin, L. O., Vieira Menezes, S., Ramos Torres, M. J., & de Castro Freitas, A. L. (2020, October 21-24). Agile methodologies in an educational context: a systematic review [Paper presentation]. *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 1–5. https://doi.org/10.1109/FIE44824.2020.9273997

Ovienmhada, U., Mouftaou, F., & Wood, D. (2021). Inclusive design of earth observation decision support systems for environmental governance: a case study of Lake

Nokoué. *Frontiers in Climate*, 3, 1–18. https://doi.org/10.3389/fclim.2021.717418

Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) agile & waterfall model advantages, disadvantages, and application suitability in software quality engineering. *International Journal of Scientific and Research Publications*, *13*(8), 120–124. http://dx.doi.org/10.29322/IJSRP.13.08.2023.p14015

Pashak, T. J., & Heron, M. R. (2022). Build rapport and collect data: A teaching resource on the clinical interviewing intake. *Discover Psychology*, *2*(20), 1–11. https://doi.org/10.1007/s44202-022-00019-5

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). SAGE.

Peck, J., Kirk, C. P., Luangrath, A. W., & Shu, S. B. (2021). Caring for the commons: Using psychological ownership to enhance stewardship behavior for public goods. *Journal of Marketing*, *85*(2), 33–49. https://doi.org/10.1177/0022242920952084

Peddisetty, N. (2024). Integrating waterfall and agile: A hybrid approach to project Management. *International Journal of Business of Quantitative Economics*, *8*(1), 52–58.

Peels, R., & Bouter, L. (2023). Replication and trustworthiness. *Accountability in Research*, *30*(2), 77–87.

Petersén, C., & Carlsson, E. (2021). Life with a stoma—coping with daily life: Experiences from focus group interviews. *Journal of Clinical Nursing*, *30*(15–16), 2309–2319. https://doi.org/10.1111/jocn.15769

Pfister, F., & Pozas, C. (2023). The influence of Chile's food labeling and advertising law and other factors on dietary and physical activity behavior of elementary students in a peripheral region: a qualitative study. *BMC Nutrition*, *9*(11), 1–13. https://doi.org/10.1186/s40795-023-00671-7

Piksa, M., Noworyta, K., Gundersen, A., Kunst, J., Morzy, M., Piasecki, J., & Rygula, R. (2024). The impact of confirmation bias awareness on mitigating susceptibility to misinformation. *Frontiers in Public Health*, *12*, 1–8. https://doi.10.3389/fpubh.2024.1414864

Pleasants, J. M. (2013). The making of Nova Southeastern University: A tradition of innovation, pp.1964–2014. Fort Lauderdale, FL: Nova Southeastern University.

Popa, O., Mihele, C., Făgărăşan, C., & Pîslă, A. (2021, May 27-28). Leadership approach towards Agile, Waterfall, and Iterative implementation of the software development products [Paper presentation]. *In IOP Conference Series: Materials Science and Engineering*. Oradea, Romania, *1169*(1), 1–11. https://doi.org/10.1088/1757-899X/1169/1/012017

Pothukuchi, A. S., Kota, L. V., & Mallikarjunaradhya, V. (2023). Impact of generative AI on the software development lifecycle (SDLC). *International Journal of Creative Research Thoughts*, *11*(8), b287–b291.

Prenner, N., Unger-Windeler, C., & Schneider, K. (2021). Goals and challenges in hybrid software development approaches. *Journal of Software: Evolution and Process*, *33*(11), 1–25. e2382. https://doi.org/10.1002/smr.2382

Priya, A. (2020). Case study methodology of qualitative research: Key attributes and

navigating the conundrums in its application. *Sociological Bulletin*, *70*(1), 94–110. https://doi.org/10.1177/0038022920970318

Rahimi, S. (2024). Saturation in qualitative research: An evolutionary concept analysis. *International Journal of Nursing Studies Advances*, *6*, 1–11. https://doi.org/10.1016/j.ijnsa.2024.100174

Rahmani, H., & Hikmawati, E. (2020). Combining SDLC method And ITIL framework by involving auditors. *Jurnal AKSI (Akuntansi dan Sistem Informasi)*, *5*(1), 6–12. https://doi.org/10.32486/aksi.v5i1.431

Ramadani, S.S., Kurniawan, H., & Wijaya, R.F. (2022). Online attendance system website-based at the Village Hall Office Paya Bakung using the waterfall method. *Journal of Applied Engineering and Technological Science*, *4*(1), 505–511. https://doi.org/10.37385/jaets.v4i1.1178

Razali, S, Kamal Bashah N. S., Abdul-Rahman, S., Mohamad Yusop, N. S., Mohamad Hashim, H. N., &Dziyauddin, H. (2021, November 17-19). Collaborative platform of networked integrated legal affairs management system (NILAMs) using hybrid agile-waterfall approach [Paper presentation]. *2021 IEEE International Conference on Computing (ICOCO)*, Kuala Lumpur, Malaysia, 19–24. https://doi.org/10.1109/ICOCO53166.2021.9673571

Reiff, J., & Schlegel, D. (2022). Hybrid project management–a systematic literature review. *International Journal of Information Systems and Project Management*, *10*(2), 45–63. https://doi.org/10.12821/ijispm100203

Responsibilities. (2025). Army.mil. https://ahrpo.army.mil/About-

AHRPO/Responsibilities/

Riazi, A. M., Rezvani, R., & Ghanbar, H. (2023). Trustworthiness in L2 writing research: A review and analysis of qualitative articles in the journal of second language writing. *Research Methods in Applied Linguistics*, *2*(3), 1–14. https://doi.org/10.1016/j.rmal.2023.100065

Rizzo, A., Aquilina, R., & Falzon, R. (2024). Applying MAXQDA in a grounded theory study: Evaluation. In *Computer Supported Qualitative Research: New Trends in Qualitative Research*. *World Conference on Qualitative Research 2024 (WCQR2024)* (Vol. 1061, 175–189).

Rizkiansyah, M., Mastur, & Yunus, U. (2022, August 11-12). The dissemination of electric control installation based on microcontroller through digital application in the vocational school environment [Paper presentation]. *2022 International Conference on Information Management and Technology (ICIMTech)*, Semarang, Indonesia, 312–317. https://doi.org/10.1109/ICIMTech55957.2022.9915124

Rogers, E. M. (1995). Lessons for guidelines from the diffusion of innovations. *The Joint Commission Journal on Quality Improvement, 21*(7), 324–328. https://doi.org/10.1016/S1070-3241(16)30155-9

Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). Free Press. (Original work published 1962).

Rose, J., & Johnson, C. W. (2020). Contextualizing reliability and validity in qualitative research: Toward more rigorous and trustworthy qualitative social science in leisure research. *Journal of Leisure Research*, *51*(4), 432–451.

https://doi.org/10.1080/00222216.2020.1722042

Rosenberg, J., & Mojadeddi, Z. M. (2024). Automated transcription of interviews in qualitative research using artificial intelligence A simple guide. *Journal of Surgery Research and Practice*, *5*(2), 1–6. https://doi.org/10.46889/jsrp.2024/5204

Rutakumwa, R., Mugisha, J. O., Bernays, S., Kabunga, E., Tumwekwase, G., Mbonye, M., & Seeley, J. (2020). Conducting in-depth interviews with and without voice recorders: A comparative analysis. *Qualitative Research*, *20*(5), 565–581. https://doi.org/10.1177/1468794119884806

Saeed, S., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2019). Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, *8*(5), 446–460. https://doi.org/10.12785/ijcds/080502

Saleh, M. P. A. Q. & Ipantri, I. (2023). Design of Android-based school information media application case study of Qomariah educational institution. *JISA (Journal of Informatics and Science)*, *6*(1), 62–68. https://doi.org/10.31326/jisa.v6i1.1647

Salgado, C.C.R., Aires, RFF & de Araújo, A.G. Transient competitive advantage model (TCAM) to analyze companies in the context of transience. *Global Journal of Flexible Systems Management*, *23*, 185–199. https://doi.org/10.1007/s40171-021-00296-4

Sana, T., Yasmeen, S., DureSamin, A., Lubna, B., & Faraz, K. (2021). Strengthening supportive supervision: A case study of the Expanded Programme on Immunization in Sindh, Pakistan. *Journal of Global Health*, *11*, 1–10.

https://doi.org/10.7189/jogh.11.06004

Sanip, S. (2020). Research methodological challenges and recommendations for conducting a comparative qualitative longitudinal study across two countries on different continents. *International Journal of Qualitative Methods*, *19*, https://doi.org/10.1177/1609406920917493

Santonanda, A. L., Nathaniel, R., Putra, W. L., Anggreainy, M. S., Danaparamita, M., & Elok Amalia, A. (2022, September 13-14). Online craftsman ordering application development using waterfall methodology [Paper presentation]. *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, Jakarta, Indonesia, 150–155. https://doi.org/10.1109/IC2IE56416.2022.9970021

Santos, K. D. S., Ribeiro, M. C., Queiroga, D. E. U. D., Silva, I. A. P. D., & Ferreira, S. M. S. (2020). The use of multiple triangulations as a validation strategy in a qualitative study. *Collective Health Science*, *25*, 655–664. https://doi.org/10.1590/1413-81232020252.12302018

Self, B. (2021) Conducting interviews during the covid-19 pandemic and beyond. *Forum Qualitative Social Research*, *22*(3), 1–18. https://dx.doi.org/10.17169/fqs-22.3.3741

Sharp, J. H., Mitchell, A., & Lang, G. (2020). Agile teaching and learning in information systems education: An analysis and categorization of literature. *Journal of Information Systems Education*, *31*(4), 269–281.

Shrivastava, A., Jaggi, I., Katoch, N., Gupta, D., & Gupta, S. (2021, July). A systematic review on extreme programming. In *Journal of Physics: Conference Series* 1969

(1), 1–11. https://doi.org/10.1088/1742-6596/1969/1/012046

Sipahutar, R. J., Hidayanto, A. N., Rahardja, U., & Phusavat, K. (2020, November). Drivers and barriers to IT service management adoption in Indonesian start-ups based on the diffusion of innovation theory [Paper presentation]. *Fifth International Conference on Informatics and Computing (ICIC)*, Gorontalo, Indonesia, 1–8. https://doi.org/10.1109/ICIC50835.2020.9288556

Small, M. L., & Cook, J. M. (2023). Using interviews to understand why: Challenges and strategies in the study of motivated action. *Sociological Methods & Research*, *52*(4), 1591–1631. https://doi.org/10.1177/0049124121995552

Squire, C. M., Giombi, K. C., Rupert, D. J., Amoozegar, J., & Williams, P. (2024). Determining an appropriate sample size for qualitative interviews to achieve true and near code saturation: Secondary analysis of data. *Journal of Medical Internet Research*, *26*, e52998.

Sreenivasan, S., & Kothandaraman, K. (2019). Improving processes by aligning capability maturity model integration and the scaled agile framework. *Global Business & Organizational Excellence*, *38*(6), 42–51. https://doi.org/10.1002/joe.21966

Stalmeijer, R. E., Brown, M. E., & O'Brien, B. C. (2024). How to discuss transferability of qualitative research in health professions education. *The Clinical Teacher*, *21*(6), 1–7. https://doi.org/10.1111/tct.13762

Stewart, R. E., Beidas, R. S., & Mandell, D. S. (2019). Stop calling them laggards: Strategies for encouraging nonadopters to incorporate evidence-based practices.

Psychiatric Services, *70*(10), 958–960. https://doi.org/10.1176/appi.ps.201900031

Stuber, J. M., Middel, C. N., Mackenbach, J. D., Beulens, J. W., & Lakerveld, J. (2020). Successfully recruiting adults with a low socioeconomic position into community-based lifestyle programs: A qualitative study on expert opinions. *International Journal of Environmental Research and Public Health*, *17*(8), 1–15. https://doi.org/10.3390/ijerph17082764

Taherdoost, H. (2022). What are different research approaches? Comprehensive review of qualitative, quantitative, and mixed method research, their applications, types, and limitations. *Journal of Management Science & Engineering Research*, *5*(1), 53–63. https://doi.org/10.30564/jmser.v5i1.4538

Tam, C., da Costa Moura, E. J., Oliveira, T., & Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, *38*(3), 165–176. https://doi.org/10.1016/j.ijproman.2020.02.001

Tamburri, D. A., Palomba, F., & Kazman, R. (2020). Success and failure in software engineering: A follow up systematic literature review. *IEEE Transactions on Engineering Management*, *68*(2), 599–611. https://doi.org/10.1109/TEM.2020.2976642

Tan, L. S., Audrey, L. C. L., & Cheah, Y. Y. (2021). The moderating effect of mental health in familiarity bias and self-attribution of financial planning behavior among Malaysians. *Studies of Applied Economics*, *39*(4), 1–12. http://dx.doi.org/10.25115/eea.v39i4.4567

The Standish Group. (2018). *Project resolution benchmark for IBEX Financial Corp*. https://www.standishgroup.com/sample_research_files/DemoPRBR.pdf

Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus waterfall project management: Decision model for selecting the appropriate approach to a project. *Procedia Computer Science*, *181*, 746–756.

Thomas, T., Saleeshya, P.G., & Suresh, M. (2022). Assessment of CMMI level of manufacturing industry using fuzzy logic approach: A case study. *Journal of Modelling in Management*, *17*(4), 1098–1125. https://doi.org/10.1108/JM2-09-2020-0229

Thunberg, S., & Arnell, L. (2022). Pioneering the use of technologies in qualitative research–A research review of the use of digital interviews. *International Journal of Social Research Methodology*, *25*(6), 757–768. https://doi.org/10.1080/13645579.2021.1935565

Trivedi, D. (2021). Agile methodologies. *International Journal of Computer Science & Communication*, *12*(2), 91–100. https://doi.org/10.13140/RG.2.2.33950.25928

Vella, J. (2024). In pursuit of credibility: evaluating the divergence between member-checking and hermeneutic phenomenology. *Research in Social and Administrative Pharmacy*, *20*(7), 665–669. https://doi.org/10.1016/j.sapharm.2024.04.001

Vu, P. (2023). Promoting fourth grade students' learning engagement and math performances through using ability grouping rotations technique. *The Journal of Teacher Action Research*, *10*(1).

White-Lewis, D. K. (2020). The facade of fit in faculty search processes. *The Journal of Higher Education*, *91*(6), 833–857. https://doi.org/10.1080/00221546.2020.1775058

Yahya, N., Maidin, S.S., & Safari, M.S. (2021). The development of interview protocol to explore hybrid agile software development phases. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(3), 1639–1645. https://doi.org/10.30534/ijatcse/2021/221032021

Yahya, N., & Maidin, S. S. (2023). Hybrid agile development phases: The practice in software projects as performed by the software engineering team. *Indonesian Journal of Electrical Engineering and Computer Science*, *29*(3), 1738–1749. https://doi.org/10.11591/ijeecs.v29.i3.pp1738-1749

Zalasiewicz, J., Adeney Thomas, J., Waters, C. N., Turner, S., & Head, M. J. (2024). The meaning of the Anthropocene: Why it matters even without a formal geological definition. *Nature*, *632*(8027), 980–984.

Zhang, X., Ma, L., Xu, B., & Xu, F. (2019). How social media usage affects employees' job satisfaction and turnover intention: An empirical study in China. *Information & Management*, *56*(6), 1–12. https://doi.org/10.1016/j.im.2018.12.004

Zhou, Y., & Shen, L. (2021). Confirmation bias and the persistence of misinformation on climate change. *Communication Research*, *49*(4), 500–523. https://doi.org/10.1177/00936502211028049

Zigler, E. D., Grelson, S. L., & Hoepner, J. K. (2021). Exploring the role of mindsets in a sophomore level undergraduate research course. *Teaching and Learning in*

Communication Sciences & Disorders, *5*(1), 1–18. https://doi.org/10.30707/TLCSD5.1.1624982519.527792

Zijlstra, T., Durand, A., Hoogendoorn-Lanser, S., & Harms, L. (2020). Early adopters of mobility-as-a-service in the Netherlands. *Transport Policy*, *97*, 197–209. https://doi.org/10.1016/j.tranpol.2020.07.019

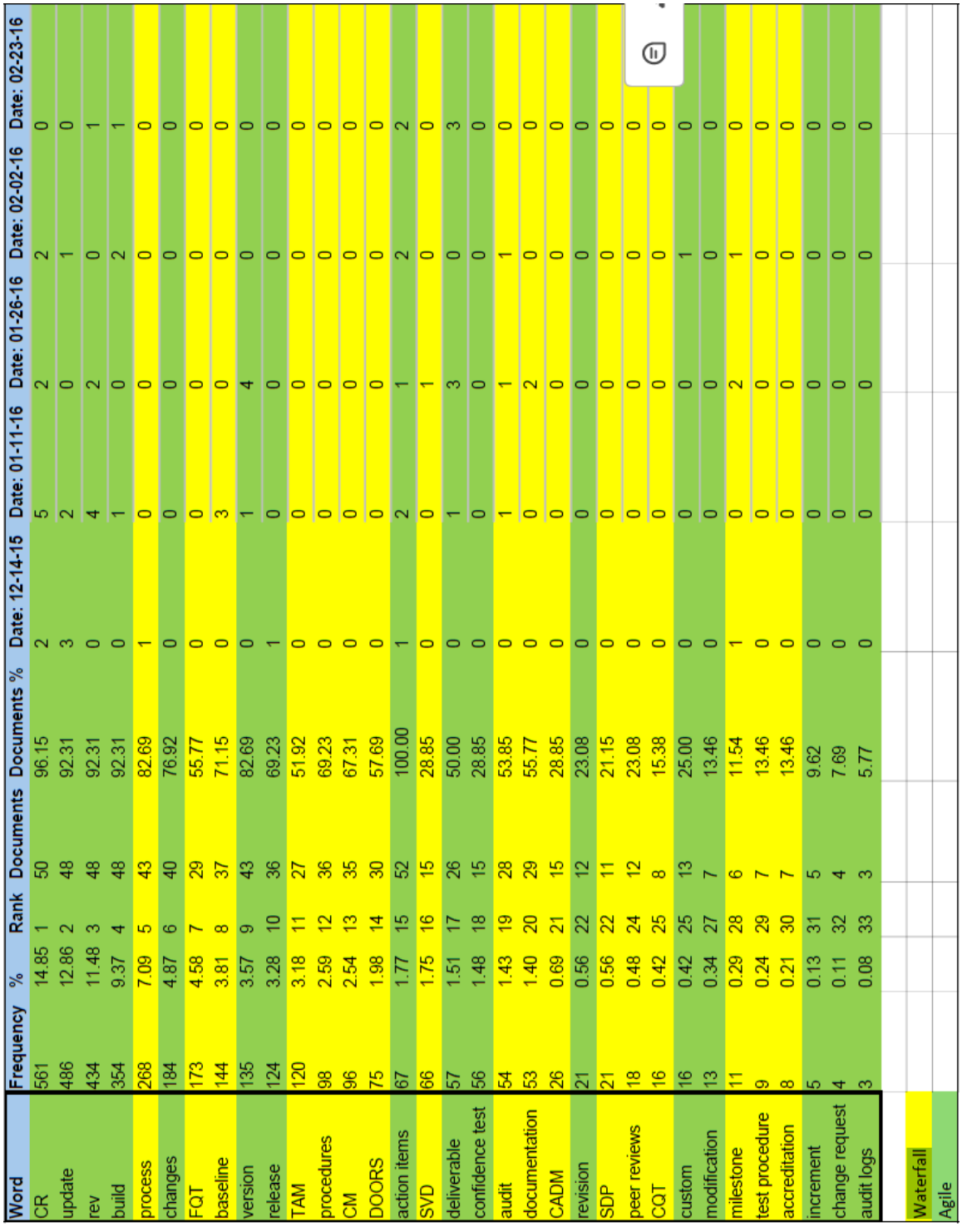
Appendix A: Interview Questions

1. Describe the waterfall methodology.
2. How do you feel about the waterfall methodology?
3. Describe the agile methodology.
4. How do you feel about the agile methodology?
5. Describe a hybrid approach of the waterfall and agile methodologies.
6. How do you feel about a hybrid approach of the waterfall and agile methodologies?
7. What consequences might combining waterfall and agile methodologies (i.e. hybrid approach) foster?
8. Do you feel that this project combines waterfall and agile methodologies, and why?
9. What is/are the impact(s) of your answer to Question #8 on the project?
10. How do you feel about the overall success of this project?

Closing Statement:

"Thank you so much for your time. It has truly been a privilege for me to hear, chronicle, and understand your point of view. I genuinely need your perspective and story to help me explain this phenomenon and possibly lead to improvements in our profession. Would you mind if I contacted you, again, for further clarification, if needed?"

Appendix B: MAXQDA Spreadsheet for the First 52 IPT Meetings



Word	Frequency	%	Rank	Documents	Documents %	Date: 12-14-15	Date: 01-11-16	Date: 01-26-16	Date: 02-02-16	Date: 02-23-16
CR	561	14.85	1	50	96.15	2	5	2	2	0
update	486	12.86	2	48	92.31	3	2	0	1	0
rev	434	11.48	3	48	92.31	0	4	2	0	1
build	354	9.37	4	48	92.31	0	1	0	2	1
process	268	7.09	5	43	82.69	1	0	0	0	0
changes	184	4.87	6	40	76.92	0	0	0	0	0
FQT	173	4.58	7	29	55.77	0	0	0	0	0
baseline	144	3.81	8	37	71.15	0	3	0	0	0
version	135	3.57	9	43	82.69	0	1	4	0	0
release	124	3.28	10	36	69.23	1	0	0	0	0
TAM	120	3.18	11	27	51.92	0	0	0	0	0
procedures	98	2.59	12	36	69.23	0	0	0	0	0
CM	96	2.54	13	35	67.31	0	0	0	0	0
DOORS	75	1.98	14	30	57.69	0	0	0	0	0
action items	67	1.77	15	52	100.00	1	2	1	2	2
SVD	66	1.75	16	15	28.85	0	0	1	0	0
deliverable	57	1.51	17	26	50.00	0	1	3	0	3
confidence test	56	1.48	18	15	28.85	0	0	0	0	0
audit	54	1.43	19	28	53.85	0	1	1	1	0
documentation	53	1.40	20	29	55.77	0	0	2	0	0
CADM	26	0.69	21	15	28.85	0	0	0	0	0
revision	21	0.56	22	12	23.08	0	0	0	0	0
SDP	21	0.56	22	11	21.15	0	0	0	0	0
peer reviews	18	0.48	24	12	23.08	0	0	0	0	0
CQT	16	0.42	25	8	15.38	0	0	0	0	0
custom	16	0.42	25	13	25.00	0	0	0	1	0
modification	13	0.34	27	7	13.46	0	0	0	0	0
milestone	11	0.29	28	6	11.54	1	0	2	1	0
test procedure	9	0.24	29	7	13.46	0	0	0	0	0
accreditation	8	0.21	30	7	13.46	0	0	0	0	0
increment	5	0.13	31	5	9.62	0	0	0	0	0
change request	4	0.11	32	4	7.69	0	0	0	0	0
audit logs	3	0.08	33	3	5.77	0	0	0	0	0
Waterfall										
Agile										

Word	Date: 02-29-16	Date: 03-07-16	Date: 03-08-16	Date: 03-14-16	Date: 03-15-16	Date: 03-28-16	Date: 04-04-16	Date: 04-11-16
CR	8	0	10	9	1	4	11	3
update	1	3	1	6	0	0	6	3
rev	1	10	6	7	0	4	5	7
build	2	6	4	1	0	2	2	2
process	3	1	5	0	0	0	5	0
changes	5	0	1	0	0	1	3	0
FQT	0	1	0	0	0	0	0	1
baseline	13	0	0	2	0	2	5	0
version	1	0	0	2	0	1	1	4
release	1	0	4	0	0	0	0	1
TAM	0	0	0	0	0	0	0	2
procedures	0	2	2	0	0	2	1	1
CM	1	2	0	1	0	2	2	1
DOORS	1	0	0	1	0	0	1	1
action items	1	1	1	2	1	1	1	1
SVD	0	0	0	0	0	0	0	0
deliverable	3	2	2	1	0	0	0	0
confidence test	0	0	0	0	0	0	0	0
audit	0	1	0	2	0	3	0	2
documentation	0	0	0	1	0	2	0	0
CADM	0	0	0	0	0	1	1	0
revision	0	0	0	0	0	0	0	0
SDP	0	0	0	2	0	5	0	3
peer reviews	0	0	0	0	0	0	1	0
CQT	0	0	0	0	0	0	0	0
custom	0	1	0	0	0	0	0	0
modification	0	0	0	0	0	1	0	0
milestone	0	0	0	0	0	0	0	0
test procedure	0	0	0	0	0	0	1	0
accreditation	0	0	0	0	0	0	1	0
increment	0	0	0	0	0	0	0	0
change request	0	0	0	0	0	0	0	0
audit logs	0	0	0	0	0	0	0	0
Waterfall								
Agile								

Word	Date: 04-25-16	Date: 05-02-16	Date: 05-09-16	Date: 05-16-16	Date: 05-31-16	Date: 06-06-16	Date: 06-13-16	Date: 06-20-16
CR	21	16	12	5	12	8	7	10
update	1	2	5	5	3	12	6	3
rev	9	6	0	2	4	4	13	8
build	0	1	4	1	6	6	9	2
process	4	4	1	8	5	4	4	4
changes	2	2	2	1	0	1	0	2
FQT	0	0	1	0	0	0	0	0
baseline	0	1	3	0	0	1	3	1
version	0	1	12	1	1	2	2	0
release	1	0	1	0	1	1	1	1
TAM	1	2	1	2	1	2	1	1
procedures	1	1	0	1	2	1	3	3
CM	4	0	0	1	0	0	1	0
DOORS	9	0	1	0	0	2	1	2
action items	1	1	1	1	1	2	2	2
SVD	0	1	0	0	0	0	0	0
deliverable	4	3	0	0	0	0	0	0
confidence test	0	0	0	0	0	0	0	0
audit	0	0	0	1	0	0	2	3
documentation	1	0	1	1	0	0	0	0
CADM	3	0	0	0	1	0	4	2
revision	0	0	0	0	0	0	0	0
SDP	1	0	0	0	3	1	1	1
peer reviews	1	0	0	0	0	0	1	1
CQT	0	0	0	0	0	0	0	0
custom	0	0	0	0	0	0	1	0
modification	0	1	3	0	0	1	0	0
milestone	0	0	0	0	0	0	0	0
test procedure	0	0	1	0	1	0	0	0
accreditation	0	0	0	0	0	0	0	0
increment	0	0	0	0	0	1	0	0
change request	1	0	0	0	0	0	0	0
audit logs	0	0	0	0	0	0	0	0
Waterfall								
Agile								

Word	Date: 07-11-16	Date: 07-18-16	Date: 07-25-16	Date: 08-08-16	Date: 08-15-16	Date: 08-22-16	Date: 09-06-16	Date: 09-12-16
CR	16	16	9	7	14	23	14	14
update	4	7	12	11	11	6	7	19
rev	8	8	1	9	8	6	8	3
build	6	10	6	7	13	15	32	3
process	2	4	2	2	3	1	6	9
changes	1	7	1	3	1	2	1	3
FQT	0	1	1	6	6	2	1	4
baseline	1	7	0	4	6	2	6	3
version	4	1	3	1	1	2	7	6
release	0	2	0	2	1	7	12	3
TAM	3	3	0	1	3	4	3	0
procedures	1	2	1	6	4	4	1	3
CM	1	2	2	0	1	3	0	4
DOORS	2	9	2	2	3	2	2	4
action items	1	2	1	1	2	1	1	2
SVD	0	1	0	0	1	2	0	0
deliverable	2	2	1	0	1	0	1	2
confidence test	0	0	0	0	0	0	0	0
audit	1	0	1	4	0	0	0	0
documentation	1	1	2	0	1	1	1	4
CADM	1	2	0	3	1	0	0	0
revision	0	0	0	0	0	0	1	2
SDP	0	0	2	0	0	0	0	0
peer reviews	0	1	0	0	0	0	0	0
CQT	0	0	0	0	0	0	0	0
custom	0	0	0	0	1	0	1	0
modification	0	0	0	0	1	0	0	0
milestone	0	0	0	0	0	0	0	0
test procedure	0	0	0	0	0	0	0	1
accreditation	0	1	0	0	0	0	0	0
increment	0	0	0	0	0	0	0	0
change request	0	0	0	0	0	0	1	0
audit logs	0	0	0	0	0	0	0	0
Waterfall								
Agile								

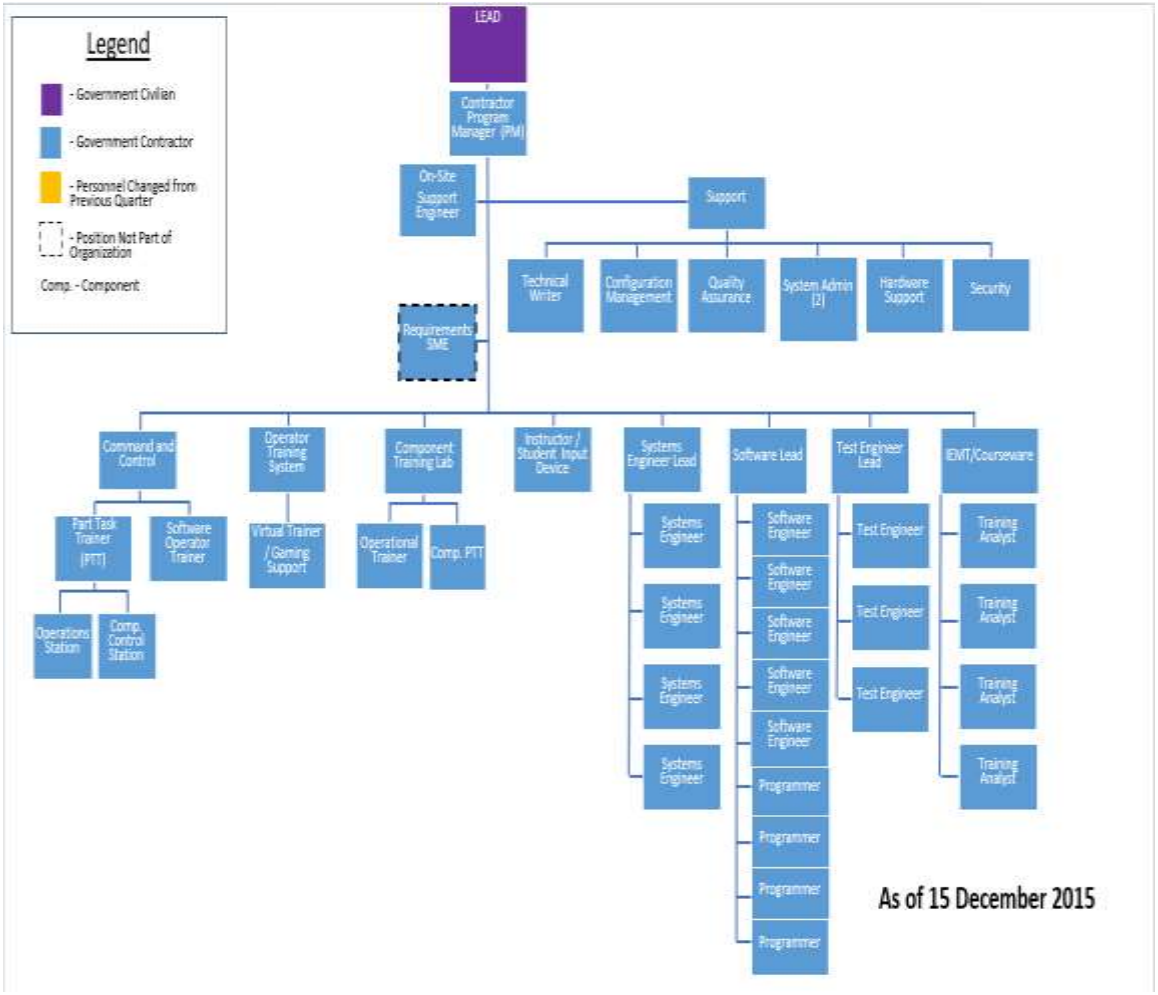
Word	Date: 09-19-16	Date: 09-26-16	Date: 10-03-16	Date: 10-11-16	Date: 10-17-16	Date: 10-25-16	Date: 11-01-16	Date: 11-08-16
CR	21	41	37	3	13	4	15	16
update	15	19	28	20	16	10	28	32
rev	15	29	22	12	11	7	27	23
build	12	8	17	7	15	10	10	9
process	7	13	10	6	11	0	21	9
changes	9	18	6	2	3	3	17	9
FQT	11	20	8	11	15	8	13	11
baseline	6	3	1	9	1	5	15	4
version	1	3	3	1	3	6	2	2
release	7	11	5	4	3	1	5	12
TAM	1	0	14	0	0	0	0	0
procedures	1	7	6	14	6	0	6	1
CM	4	2	4	1	4	0	2	2
DOORS	5	4	0	0	1	0	3	2
action items	1	1	1	1	1	1	1	1
SVD	0	2	3	12	0	0	1	0
deliverable	1	0	1	0	0	0	0	0
confidence test	0	0	0	0	9	8	9	3
audit	3	3	4	0	0	0	1	0
documentation	1	0	0	0	3	1	0	3
CADM	0	0	0	0	0	0	1	2
revision	0	1	2	1	0	0	0	0
SDP	0	0	0	0	0	0	0	0
peer reviews	2	0	1	2	0	0	2	0
CQT	0	0	0	0	0	0	0	0
custom	0	1	1	3	0	0	0	1
modification	0	0	0	0	0	0	0	0
milestone	0	0	0	0	0	0	0	0
test procedure	0	0	0	0	0	0	1	0
accreditation	0	0	0	0	0	0	0	0
increment	0	0	1	0	1	0	0	0
change request	0	0	0	0	1	0	0	0
audit logs	0	0	0	0	0	0	0	0
Waterfall								
Agile								

Word	Date: 11-15-16	Date: 01-10-17	Date: 01-17-17	Date: 01-24-17	Date: 01-31-17	Date: 02-28-17	Date: 03-07-17	Date: 03-14-17
CR	6	7	6	14	6	14	2	3
update	26	12	8	24	6	18	4	8
rev	21	13	21	11	3	15	7	8
build	14	9	1	15	25	14	20	5
process	23	7	14	10	2	5	2	11
changes	8	13	11	9	3	9	1	4
FQT	19	4	1	4	1	3	0	0
baseline	2	8	4	2	1	4	1	0
version	10	4	2	2	2	7	9	4
release	9	1	1	3	0	4	3	4
TAM	0	0	33	6	0	1	0	8
procedures	0	1	0	0	1	1	0	1
CM	2	0	0	6	1	15	1	8
DOORS	2	2	0	1	1	0	3	0
action items	2	1	1	1	1	1	1	1
SVD	20	0	0	0	0	0	4	4
deliverable	0	2	3	5	0	0	0	0
confidence test	3	6	1	3	1	3	2	1
audit	2	0	3	1	0	1	0	1
documentation	3	2	0	1	3	5	0	2
CADM	0	0	1	2	0	0	0	0
revision	1	1	5	1	0	0	0	3
SDP	0	1	0	1	0	0	0	0
peer reviews	4	0	0	0	0	0	1	0
CQT	0	0	0	1	1	0	0	0
custom	0	1	0	0	1	0	0	0
modification	0	0	0	0	0	0	0	0
milestone	0	0	0	2	0	2	0	0
test procedure	0	0	0	0	0	1	0	0
accreditation	0	1	1	2	1	1	0	0
increment	0	1	0	0	0	0	0	0
change request	0	0	1	0	0	0	0	0
audit logs	0	0	0	0	0	0	0	0
Waterfall								
Agile								

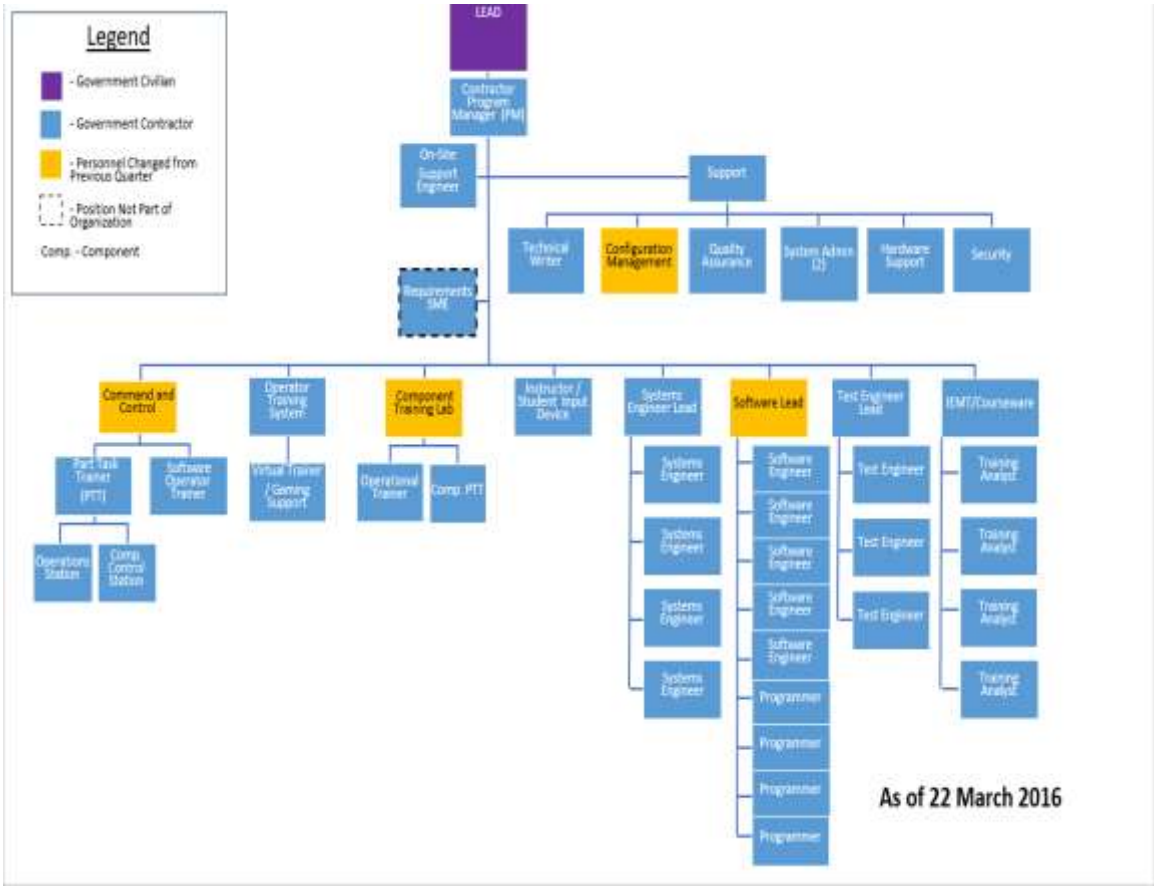
Word	Date: 03-21-17	Date: 03-28-17	Date: 04-18-17	Date: 04-25-17	Date: 05-02-17	Date: 05-09-17	Date: 05-16-17
CR	28	6	18	9	10	10	11
update	12	9	20	6	9	3	13
rev	16	4	3	6	3	3	10
build	2	1	4	4	1	6	1
process	8	4	5	7	3	3	9
changes	4	4	3	2	5	0	2
FQT	10	1	0	1	0	5	3
baseline	1	2	3	0	3	0	6
version	5	3	1	2	2	3	0
release	4	0	1	1	4	1	0
TAM	0	3	3	6	5	4	6
procedures	0	1	3	1	0	1	5
CM	6	1	3	2	1	1	2
DOORS	0	0	3	0	2	1	0
action items	1	2	2	1	1	2	2
SVD	11	0	2	0	0	0	1
deliverable	1	0	4	1	2	3	3
confidence test	1	0	4	0	0	0	2
audit	0	3	3	2	1	2	1
documentation	1	2	2	1	0	3	1
CADM	0	0	1	0	0	0	0
revision	1	0	0	0	0	0	2
SDP	0	0	0	0	0	0	0
peer reviews	0	0	0	0	0	1	0
CQT	2	6	1	0	1	3	1
custom	2	0	0	0	0	0	1
modification	0	0	0	2	0	0	4
milestone	3	0	0	0	0	0	0
test procedure	0	0	0	0	0	0	3
accreditation	0	0	0	0	0	0	0
increment	0	0	0	0	0	0	1
change request	0	0	0	0	0	0	0
audit logs	0	1	0	0	0	1	1
Waterfall							
Agile							

Appendix C: Organizational Charts

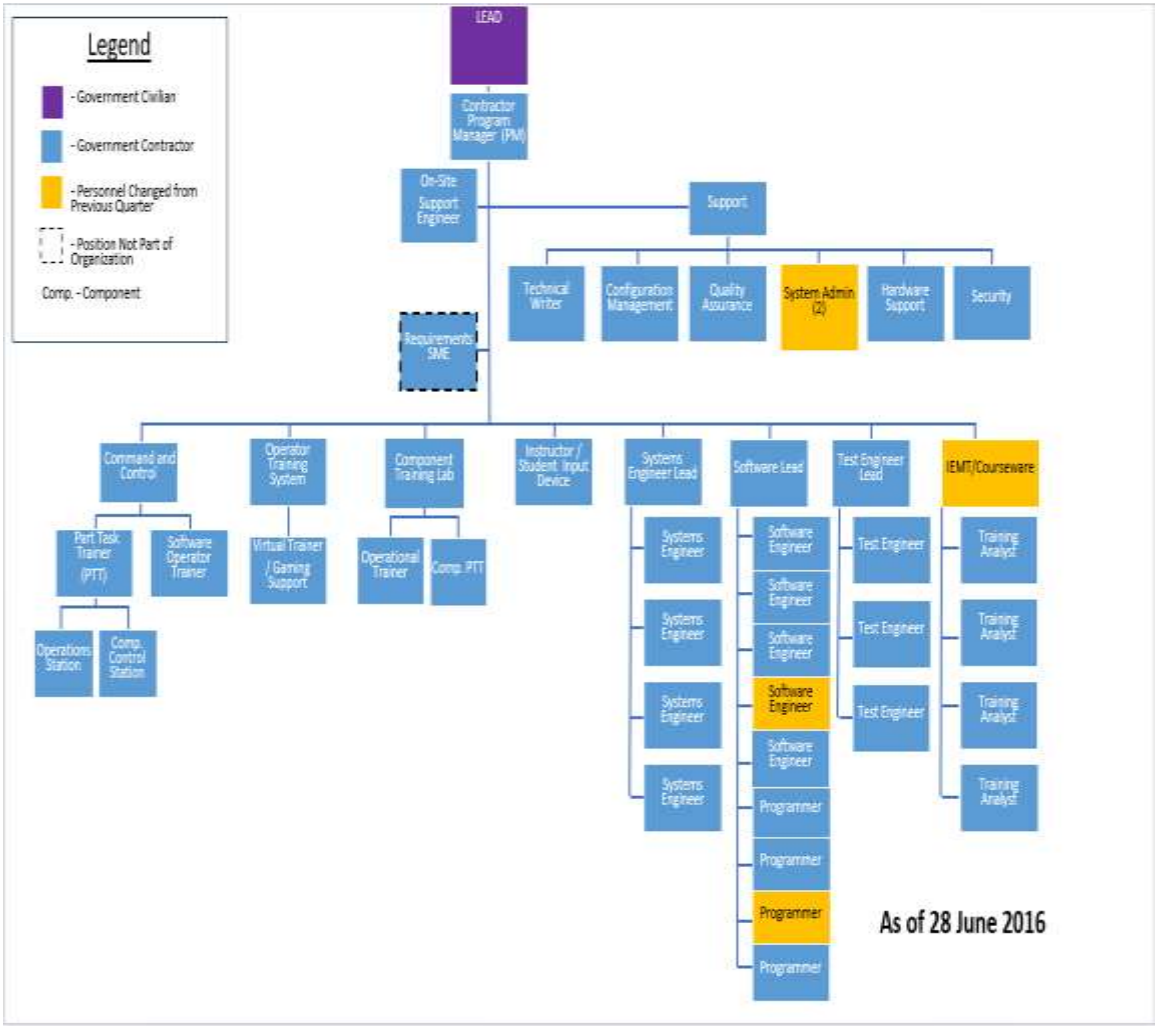
Organizational Chart: As of 15 December 2015



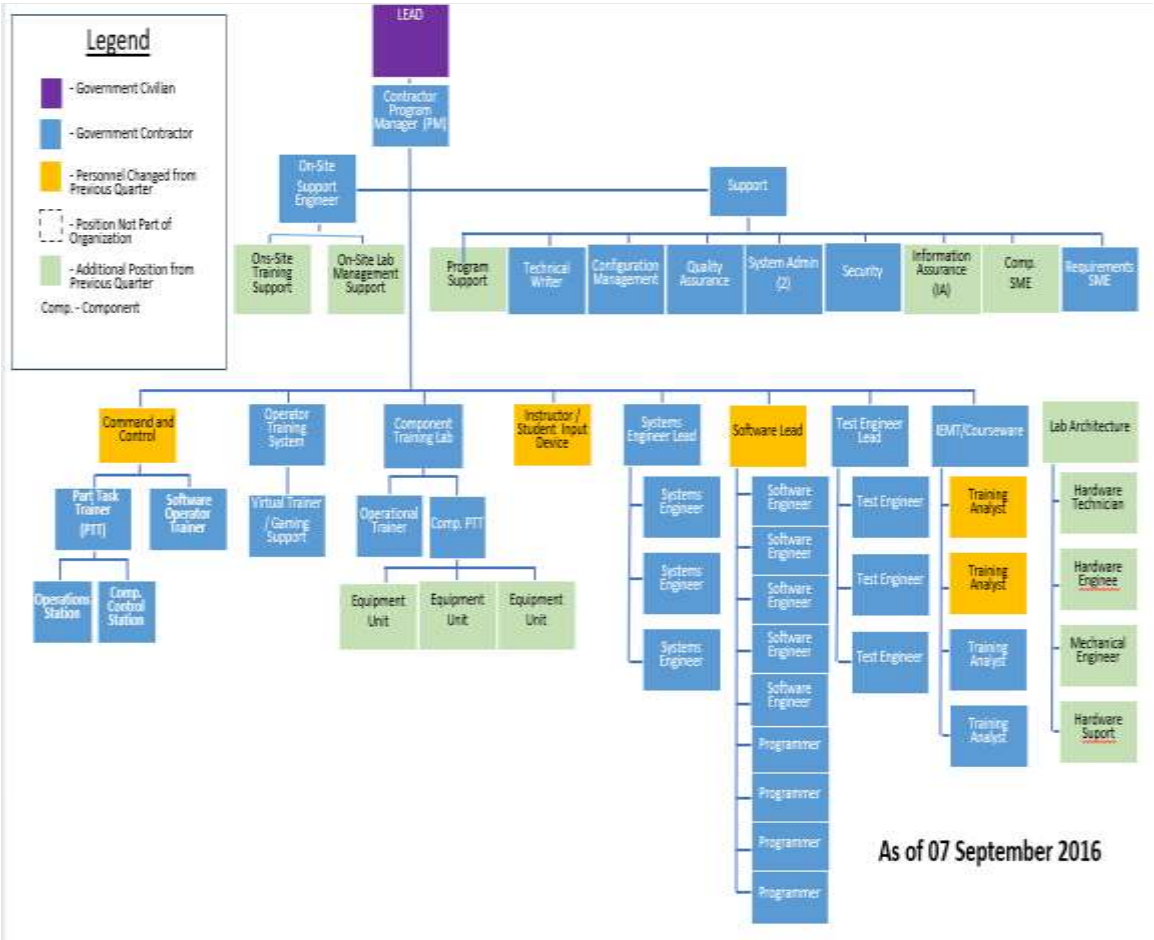
Organizational Chart: As of 22 March 2016



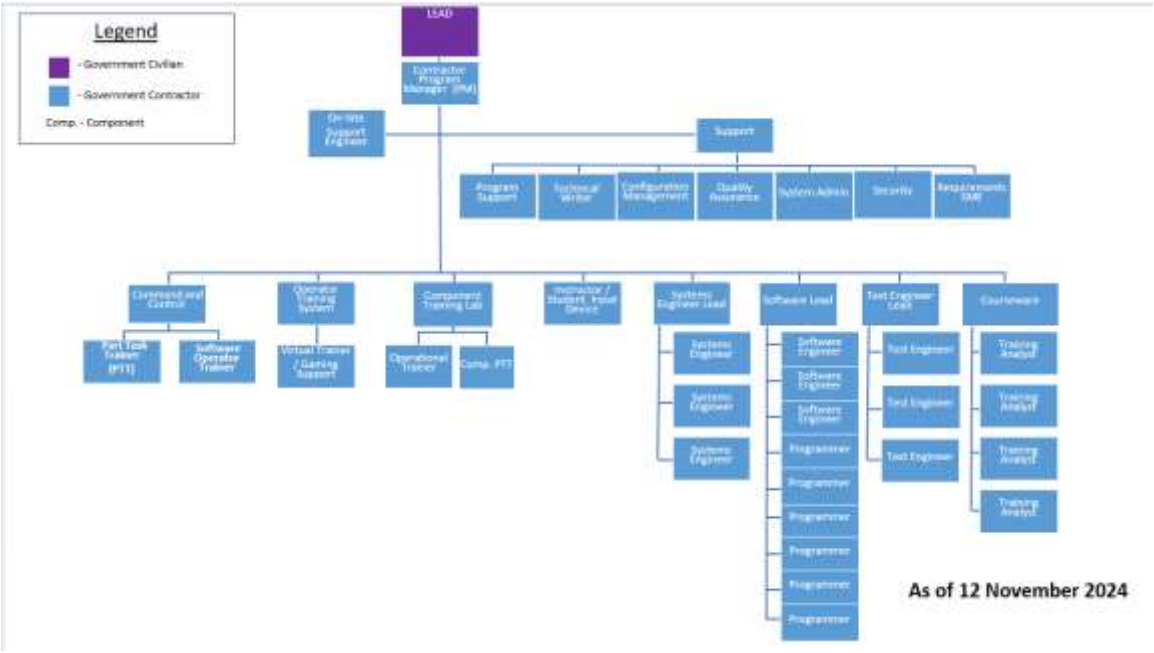
Organizational Chart: As of 28 June 2016



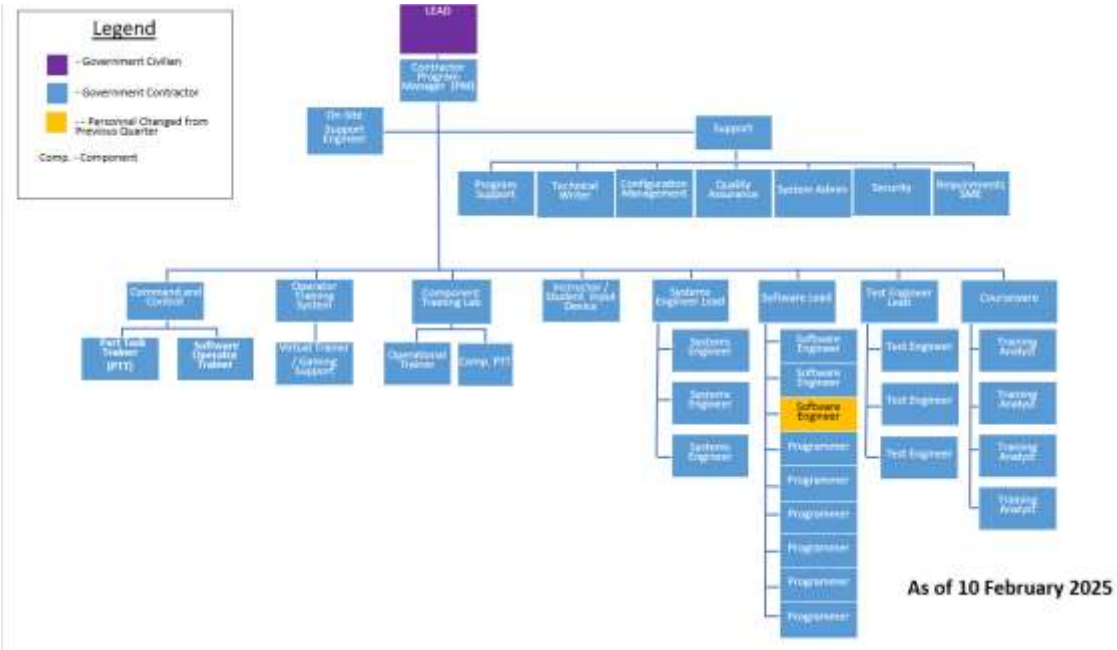
Organizational Chart: As of 07 September 2016



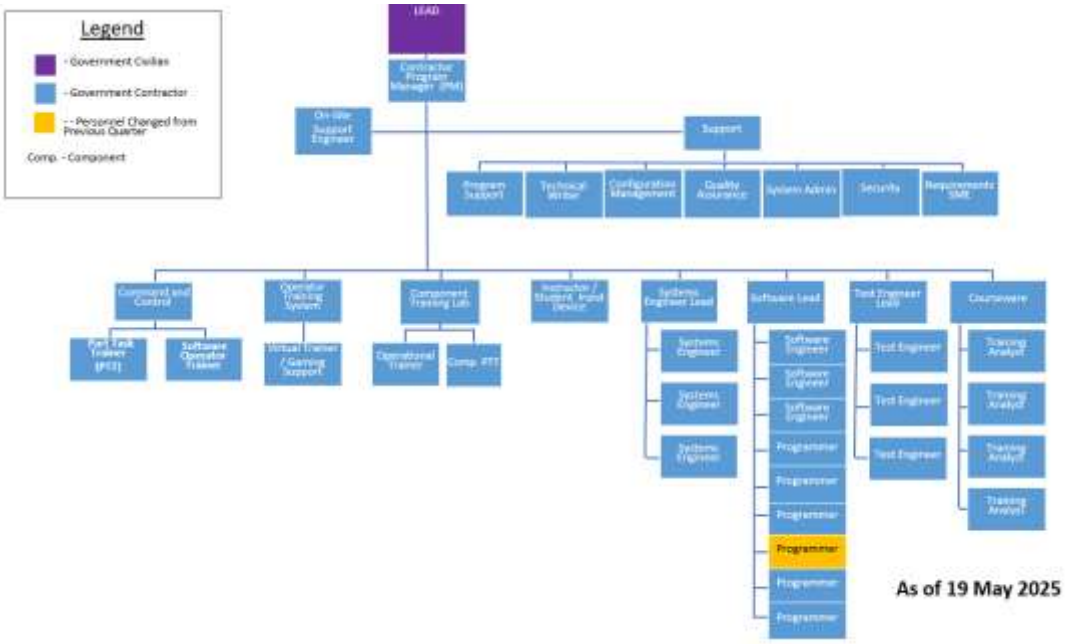
Organizational Chart: As of 12 November 2024



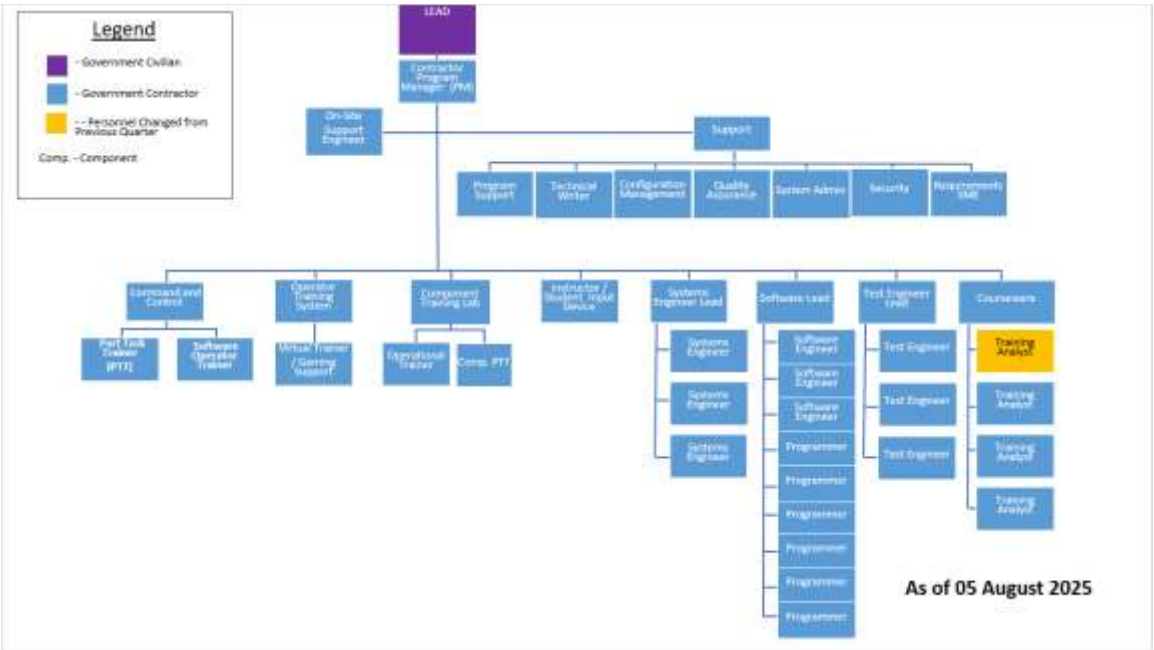
Organizational Chart: As of 10 February 2025



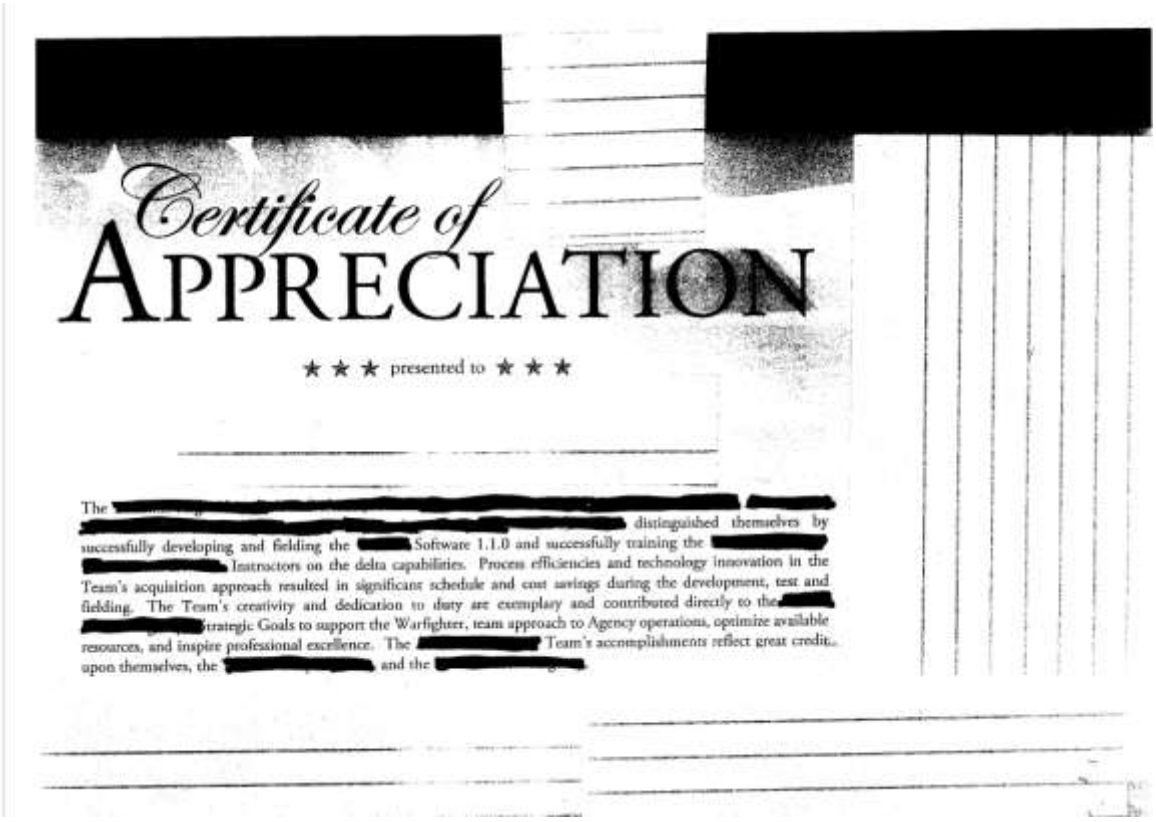
Organizational Chart: As of 19 May 2025



Organizational Chart: As of 05 August 2025



Appendix D: Certificate of Appreciation



[redacted]

Certificate of

APPRECIATION

★ ★ ★ presented to ★ ★ ★

The [redacted] distinguished themselves by successfully developing and fielding the [redacted] Software 1.1.0 and successfully training the [redacted] Instructors on the delta capabilities. Process efficiencies and technology innovation in the Team's acquisition approach resulted in significant schedule and cost savings during the development, test and fielding. The Team's creativity and dedication to duty are exemplary and contributed directly to the [redacted] Strategic Goals to support the Warfighter, team approach to Agency operations, optimize available resources, and inspire professional excellence. The [redacted] Team's accomplishments reflect great credit upon themselves, the [redacted] and the [redacted]