Walden University
ScholarWorks

Walden Dissertations and Doctoral Studies

Walden Dissertations and Doctoral Studies Collection

5-13-2024

A correlational study of technical and social factors that affect agile software project success

Christopher Joel Fry
Walden University

Follow this and additional works at: https://scholarworks.waldenu.edu/dissertations

This Dissertation is brought to you for free and open access by the Walden Dissertations and Doctoral Studies Collection at ScholarWorks. It has been accepted for inclusion in Walden Dissertations and Doctoral Studies by an authorized administrator of ScholarWorks. For more information, please contact ScholarWorks@waldenu.edu.

Walden University

College of Management and Human Potential

This is to certify that the doctoral study by

Christopher Fry

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. Gary Griffith, Committee Chairperson, Information Technology Faculty
Dr. Osama Morad, Committee Member, Information Technology Faculty

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

Walden University
2024

Abstract

A correlational study of technical and social factors that affect agile software project success

by

Christopher Joel Fry

MS, Southern New Hampshire University, 2019

BS, Indiana Wesleyan University, 2001

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Information Technology

Walden University

June 2024

Abstract

Agile software development team composition can lead to budgetary and timeline overruns, ultimately affecting software development project success. IT leaders and software development project managers must identify the critical success factors that affect agile software development project success, as unsuccessful project outcomes can make an organization seem inefficient and incapable by stakeholders. Grounded in the critical success factors theory, the purpose of this quantitative, correlational study was to examine the relationship between software development team size, experience, the level of effort estimation accuracy, and agile software development project success. Survey data (N = 61) were collected from software development project managers in Texas. The regression model was not significant, $F(3, 57) = 2.093$, $p = 0.111$, $R^2 = 0.099$. The R^2 (0.099) value indicated that approximately 9% of the variance in agile software development project success is accounted for by the linear combination of the predictor variables. All three predictors were insignificant: team experience (t= .956, p = .343, ß = .537), team size (t= .1.17, p = .247, ß = .683), and effort estimation accuracy (t= -1.90, p = .061, ß = -1.563). One recommendation is for IT leaders and software development project managers to consider the number of members when comprising a development team. The implications for positive social change include the potential for IT leaders and software development project managers to increase stakeholder satisfaction and project success rates.

A correlational study of technical and social factors that affect agile software project success

by

Christopher Joel Fry

MS, Southern New Hampshire University, 2019

BS, Indiana Wesleyan Unversity, 2001

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Information Technology

Walden University

June 2024

Dedication

I dedicate this study to my three children, Alanis, Joshua, and Olivia, and my wife, Lauren. I also want to dedicate this study to my parents, who instilled in me a great work ethic and have been a source of constant encouragement.

Acknowledgments

First, I want to thank God, who gave me the courage and strength to take on this opportunity to improve myself by advancing my education so I can positively impact the lives of others. I also want to thank my family, who sacrificed greatly so that I could continue my education. A special thank you to my wife, who took on more responsibility so I had time to dedicate to my studies. Lastly, I want to thank my committee members (Dr. Gary Griffith and Dr. Osama Morad) and all the faculty at Walden University who helped me through every stage of this dissertation and provided me with constructive feedback.

Table of Contents

List of Tables v

List of Figures vi

Section 1: Foundation of the Study 1

Background of the Problem 1

Problem Statement 2

Purpose Statement 2

Nature of the Study 3

Research Question 5

Hypotheses 5

Theoretical Framework 5

Assumptions, Limitations, and Delimitations 6

Assumptions 6

Limitations 7

Delimitations 7

Significance of the Study 7

Contribution to Information Technology Practice 7

Implications for Social Change 8

A Review of the Professional and Academic Literature 9

Agile Software Development 10

Theoretical Framework –Critical Success Factors 12

Critical Success Factors in Software Development. 13

Measurement 21
Development Team Experience 24
Development Team Size 26
Level Of Effort Estimation Accuracy 28
Agile Estimation Methods 30
Software Development Project Success 43
Success Frameworks 44
Methodologies 46
Qualitative 47
Quantitative 47
Mixed-Method 48
Transition and Summary 49
Section 2: The Project 51
Purpose Statement 51
Role of the Researcher 52
Participants 54
Research Method and Design 55
Method 55
Research Design 56
Population and Sampling 57
Ethical Research 59
Data Collection 61

Instruments 61
Data Collection Technique 67
Data Analysis Technique 68
Data Integrity 68
Statistical Analysis 69
Reliability and Validity 72
Reliability 72
Validity 74
Transition and Summary 75
Section 3: Application to Professional Practice and Implications for Change 77
Overview of Study 77
Presentation of the Findings 77
Variables 78
Descriptive Statistics 79
Agile Software Development Project Success Factors 82
Regression Analysis 86
Tests of Assumptions 87
Independence 89
Outliers 89
Linearity 89
Multicollinearity 90
Confidence Intervals 90

Inferential Results 91

Analysis Summary 94

Applications to Professional Practice 95

Implications for Social Change 96

Recommendations for Action 97

Recommendations for Further Study 98

Reflections 99

Summary and Study Conclusions 100

References 101

Appendix A: Research Participation Request 131

Appendix B: Structure of the Survey Instrument 132

Appendix C: Permission to Reuse Survey Instrument 134

Appendix D: SPSS Output 135

List of Tables

Table 1. Survey Response Likert Values 64

Table 2. Software Development Team and Project Characteristics 79

Table 3. Software Development Team Size 80

Table 4. Software Development Team Experience 81

Table 5. Software Development Team Level of Effort estimation accuracy 82

Table 6. Software Development Project Success – Product Quality 83

Table 7. Software Development Project Success – Requirements Scope 84

Table 8. Software Development Project Success – Project Timeline 85

Table 9. Software Development Project Success – Project Budget 86

Table 10. Correlation Coefficients Among Study Variables 88

Table 11. Bootstrapped Confidence Intervals for Study Variables 91

Table 12. Regression Model Summary 92

Table 13. Regression ANOVA output 92

Table 14. Regression Coefficients Output 93

List of Figures

Figure 1. Power as a Function of Total Sample Size 59

Figure 2. Historgram of Sample Distribution 78

Figure 3. Normal P-P of Regression Standardized Residuals..... 88

Figure 4. Scatterplot of the Standardized Residuals 89

Figure 5. Scatter matrix of variables 90

Section 1: Foundation of the Study

Many factors contribute to the success of a software development project. The accuracy of the level of effort estimation is incredibly challenging in the software industry. Challenges present themselves due to the complexity of software development and various social and technical factors (Kula et al., 2022). A common approach to software development and maintenance is the agile methodology. Software industries widely use the agile software methodology due to its flexibility and dynamic nature, and it is successful with small teams (Butt et al., 2022). As part of the agile methodology, each development team member has input regarding the effort required to accomplish a task. The determination of an agile software development project's success relies on estimation accuracy, relative team experience, and team size (Butt et al., 2022).

Background of the Problem

Software development teams have difficulty providing stakeholders with accurate development effort estimates. In the context of agile projects, this sentiment is especially true. The agile methodology is widely used in the software industry to develop and produce software rapidly (Venkatesh et al., 2016). There are six primary agile methods: Extreme Programming (XP), Scrum, Dynamic System Development Method (DSDM), Feature Driven Development (FDD), Adaptive Software Development (ASD), and Crystal. However, Scrum is the preferred method software development teams use because of its lightweight process with fixed-time iterations (Butt et al., 2022). The effectiveness of Scrum is demonstrated primarily in small-scale and co-located development teams. The team members play a significant role in project success,

including on-time delivery, a quality product, and cost estimation (Chiang & Lin, 2020). Project teams often contain developers with varying development levels and small-team experience (Gren et al., 2020). The project team size also varies from project to project and is determined by several factors. In the agile methodology, team members evaluate and estimate the effort required to perform a task. Team member experience and subject knowledge are essential factors that may influence estimation accuracy (Kula et al., 2022). In this study, I evaluated the relationships between project team size, experience level, level of effort estimation accuracy, and project success rate.

Problem Statement

Software projects are often delivered late and have overrun budgets due to software development being complex and affected by various technical and social factors (Kula et al., 2022). It is not uncommon for software development project costs to exceed budgetary limitations by 35% (Saeeda et al., 2020). The general IT problem was software development project managers lack an understanding of the factors that drive costs and delay agile software development projects. The specific IT problem was that some IT leaders did not understand the relationship between software development team size, team experience, level of effort estimation accuracy, and agile software development project success rates.

Purpose Statement

The purpose of this quantitative correlational study was to examine the relationship between software development team size, team experience, level of effort estimation accuracy, and project success rate of agile software development projects. The

independent variables were team size, team experience, and level of effort estimation accuracy, and the dependent variable was the project success rate. The targeted population was agile software development project managers located in Texas. I evaluated technical factors that affect development and production costs and how they correlate so that project managers can compose a development team leading to project success, thus impacting social change positively. By conducting this study, I developed insight into how project managers can mitigate production costs and provide a quality product that meets or exceeds timeline and budgetary constraints. Project management success criteria are often focused on the delivery and implementation phases to determine if project elements were performed correctly. Correctly performing elements may result in an on-time completion and within cost; however, it does not necessarily improve team efficiency and effectiveness. Identifying team components that improve efficiency and effectiveness may reduce development costs, allowing enhancements such as consumer data protection.

Nature of the Study

For this study, I used a quantitative correlational design to evaluate the relationships between software development team size, team experience, level of effort estimation accuracy, and agile software development project success rate. Three prominent research design models exist: quantitative, qualitative, and mixed methods. Developing and testing hypotheses are elements of a quantitative design (McCusker & Gunaydin, 2015). These elements align with this study's goal, making quantitative design appropriate because of its hypothesis testing support and capabilities. Researchers use a

qualitative method to explore a phenomenon, including interpreting and understanding social interactions (Jonsen et al., 2018; Moser & Korstjens, 2018). I did not seek to explore strategies or social experiences; therefore, a qualitative design was not the appropriate method to analyze and identify causal relationships. A mixed-method design views qualitative and quantitative from a single study to identify inferences (Alavi & Hąbek, 2016). Because the mixed-methods method implements elements of a qualitative design, it was inappropriate for establishing relationships between variables. Additionally, I did not seek motives influencing a phenomenon; therefore, a qualitative and mixed-methods design was not ideal.

For this quantitative study, I used a correlational design. Correlational research determines the prevalence and relationships among two or more variables (Curtis et al., 2016). I evaluated the relationships among software development team size, team experience, level of effort estimation accuracy, and software development project success rates; therefore, a correlative design is the most appropriate. Quantitative designs include descriptive, experimental, and quasi-experimental (Norris et al., 2015). The purpose of a descriptive study is to observe the behavior of the variables in their natural environment so that new phenomenological characteristics can be developed (Loeb et al., 2017). I did not use a descriptive design because it observes the current state of variables; however, relationships among variables are not assessed. Establishing causal relationships by manipulating independent variables and assigning participants randomly to control groups are elements of experimental design (McCarthy et al., 2017). A quasi-experimental design does not include randomly selected participants, making it different

from an experimental design (Haegele & Hodge, 2015). Experimental and quasi-experimental designs were inappropriate for this study because my goal was not to investigate and understand the cause and effect of variables versus examining relationships between them.

Research Question

Research Question (RQ): What is the relationship between software development team size, team experience, the accuracy of the level of effort estimation accuracy, and agile software development project success rates?

Hypotheses

Null Hypothesis (H_0): There is no significant relationship between software development team size, team experience, level of effort estimation accuracy, and agile software development project success rates.

Alternative Hypothesis (H_1): There is a significant relationship between software development team size, team experience, level of effort estimation accuracy, and agile software development project success rates.

Theoretical Framework

In this study, I viewed project success within the framework of critical success factors of agile software development projects defined by Chow and Cao (2008). Methods of measuring project success have evolved, starting with the iron triangle developed by Atkinson in 1999 (Atkinson, 1999). Then, lists of critical success factors were generated, followed by success frameworks used today. The framework created by Chow and Cao (2008) identified four dimensions in which project failures can be

categorized: (a) organizational, (b) people, (c) process, and (d) technical. Many social and technical factors contribute to software development team inefficiency and ineffectiveness (Stankovic et al., 2013).

Furthermore, Chow and Cao (2008) identified three critical success factors: (a) delivery strategy, (b) agile software engineering techniques, and (c) team capability. According to Misra et al. (2009), competent teams provide fast deliveries of quality software that meets or exceeds stakeholder requirements. The capabilities of team members are influenced by understanding and knowledge of agile development techniques and other factors (Tam et al., 2020).

Assumptions, Limitations, and Delimitations

Research activity is almost always affected by exceptional circumstances and situations. The following assumptions, limitations, and delimitations identify conditions and events that influenced this quantitative study. According to Levitt et al. (2017), acknowledging influences and circumstances increases the integrity of the study.

Assumptions

Accepted beliefs or statements believed to be accurate are known as assumptions (Yang et al., 2018). My primary assumption in this study was that participants would answer the survey questions as truthfully as possible and that they understood the questions before answering. My second assumption was that the participants have experience working in teams on agile software development projects.

Limitations

Limitations are potential weaknesses of a study. A limitation is an imperfection in theory or methodology that does not impair the validity of the study's findings (Busse et al., 2017). The number of respondents who do not provide accurate or truthful answers can skew the results, producing inaccuracies in the statistical computations. If respondents did not understand the questions correctly, their answers could be nonsensical or inaccurate. This concern is mitigated using a vetted and proven survey instrument used in previous studies such as Chow and Cao (2008) and Stankovic et al. (2013). Additionally, for this study I targeted individuals with inconsistent levels of understanding of agile methods. The lack of understanding could have affected how the respondent answers the survey.

Delimitations

Limitations set by the researcher to explicitly define the research parameters are known as delimitations. This study was limited to software development project managers that use the agile methodology. This study was also limited to project managers in Texas with teams of three or more members.

Significance of the Study

Contribution to Information Technology Practice

The results of this quantitative correlational study may provide agile software development professionals with an understanding of the impact of development team experience, team size, and level of effort estimation accuracy on project success. Software development teams are often comprised of developers with varying levels of

development experience. The findings may provide project managers insight into what mixture of development experience levels offers the best possibility for a successful project. The results may also show the impact of development team size on project success. Software development project team size varies based on resource availability, developer skills, and the size of the solution being developed. The findings of this research may also improve knowledge for project managers on the importance of level of effort estimation accuracy and its correlation to project success. Enhanced understanding of how attributes of software development project teams may add value to the software development project and product managers, IT organizations, and stakeholders of the software product.

Implications for Social Change

If the research successfully identifies the relationships between software development team size, team experience, level of effort estimation accuracy, and the impact on project success, it can improve the lives of those responsible for assembling project teams. A well-assembled project team could reduce the stress on IT project managers, software development professionals, product owners, and stakeholders. Project team composition that reflects successful software development projects can also benefit development teams by improving morale and introducing low or non-experienced developers to the thought processes of experienced developers. In addition, accurate effort estimation can aid in providing accurate and realistic time frames for the software solution delivery. Therefore, supplying IT project managers with knowledge of attributes

that produce successful development projects may incite positive social change for various stakeholders.

A Review of the Professional and Academic Literature

The literature review for this study contains an analysis and synthesis of journal articles about agile software development projects' accuracy of software development effort estimation, team experience, team size, and related topics. Additional information regarding agile project management practices that affect software development project success is included. The themes addressed in the literature review are (a) agile software development, (b) critical success factors of agile software development projects, (c) current research social and technical factors that affect project success, and (d) supporting and contradictory theories. The chosen themes provide agile software development background information, social and technical factors, development effort estimation factors, and an analysis of the theoretical framework of critical success factors by Chow and Cao (2008). The academic literature review on characteristics and factors that affect software development project success supports a correlation between software development team size, team experience, the accuracy of development effort estimation, and project success.

In this quantitative correlational study, I examined the relationship between software development team size, team experience, level of effort estimation accuracy, and project success for agile software development projects. In the literature review, I identified current factors of agile software development projects and their effect on software development project success. The literature review contains articles from SAGE

Knowledge Journals, ACM Digital Library, Science Direct, Thoreau Multi Database Search, IEEE Xplore Digital Library, Project Management Journals, and Google Scholar. Ulrich was used to verify that the references in this review were peer-reviewed. The literature review contains 92 articles, 25 (27%) related to my theoretical framework, and 74 (80%) were published within five years of my expected graduation.

Agile Software Development

In 2001, a summit of 17 supporters of lightweight development processes met in Snowbird, Utah (Krstić et al., 2018), where the term agile was adopted. The supporters wanted to create an alternative to plan-driven development; thus, the *Agile Manifesto* was introduced (Abdalhamid & Mishra, 2017). The Manifesto was designed to provide 12 core values related to agile software development principles (Stoica et al., 2016). The *Agile Manifesto* ideology includes four primary standards (a) interactions and individuals are valued above processes and tools, (b) functioning software is valued over complete and comprehensive documentation, (c) contract negotiation is not as important as collaboration with customers, and (d) responding to change versus a defined project plan (Coleman, 2016; Drury-Grogan et al., 2017). Collaboration and communication of self-organized teams are the primary focus of the agile approach to software development (Vallon et al., 2018).

Typical waterfall software development methodologies do not provide the flexibility found in agile. Agile promotes the incorporation of user involvement in the development process with minimal management (Taylor, 2016). Short development cycles, continuous deployments, and evolving requirements are significant benefits of the

agile approach to development (Drury-Grogan et al., 2017). Agile methods improve quality by promoting the breakdown of tasks into small and manageable developmental units (Kandengwa & Khoza, 2021). Development teams have entirely accepted the agile process for the management and construction of software.

Agile Software Development Methodologies

Legacy development methodologies, including waterfall, spiral, and stagewise, introduced challenges leading to the evolution of agile software management. The early stages of software development can be summarized as a cycle of coding and bug fixing. The fundamental concept was to write code first without planning and designing (Oyong & Ekong, 2019). Traditional software development methodologies (TDSMs) are considered heavyweight because they require extensive planning and work best when developers know all requirements up-front (Kula et al., 2022).

The worldwide popularity of agile software development methodologies has increased (Haines et al., 2017). The foundations of agility are that critical processes are iterative and bound by timed cycles that accommodate changing requirements (Boby et al., 2017). Over the past two decades, agile development methods have evolved. Some of the most popular include adaptive software development, scrum, crystal, and feature-driven development (Kuutila et al., 2020). The agile approach to development provides solutions to issues associated with legacy methodologies that prohibit flexible iterations.

The agile development methodology embraces software requirements that must be clarified or defined. The procedure also invites customers, known as stakeholders, to be part of the development process resulting in changes to software requirements and can

adversely affect the software product quality (Baruah, 2015). Reduced costs, project flexibility, and fast delivery are some reasons why organizations embrace the agile methodology. (Ebert & Paasivaara, 2017; Stoica et al., 2016). The agile approach to software development provides reduced time to market, a higher level of customer satisfaction, and an increased competitive margin (Mirzaei & Mabin, 2017). The agile methodology offers increased knowledge learning, employee satisfaction, and scalability (Haines et al., 2017).

Theoretical Framework –Critical Success Factors

Software is a critical component of all facets of the modern world; however, software development is imperfect. Rockart and Crescenzi (1984) developed an approach to identifying and measuring an organization's performance based on critical success factors. The process was later refined and became a well-established method of performance benchmarking (Bullen & Rockart, 1981; Chow & Cao, 2008; Rockart & Crescenzi, 1984). Bullen and Rockart (1981) defined a critical success factor as a limited number of areas where satisfactory results ensure successful and competitive organizational performance. According to Chow and Cao (2008), critical success factors are critical areas where everything must go right for the business to flourish and attain goals. Reel (1999) stated that many influences, such as project management, affect project outcomes. In software development, critical success factors are often related to fundamental project management techniques or the combination of business strategy and software engineering (Chow & Cao, 2008; Reel, 1999).

Chow and Cao (2008) produced a list of 19 failure and 36 success factors from their literature review. Their findings are significant to this study in that there are more factors affecting project success than failure. Chow and Cao (2008) identified four dimensions in which project successes can be categorized: organizational, people, process, and technical. Each category contained three to four success factors aggregated from the 36 originally defined success factors resulting in the identification of 12 critical success factors that contributed to the success of agile software development projects. These factors are (a) management commitment, (b) organizational environment, (c) team environment, (d) team capability, (e) customer involvement, (f) project management process, (g) project definition process, (h) agile software engineering techniques, (i) delivery strategy, (j) project nature, (k) project type, and (l) project schedule. Alignment of these factors is necessary for successful agile software development; if the critical success factors are unaligned with project objectives, the potential for budgetary, schedule, and project overruns increase (Chiyangwa & Mnkandla, 2018; Chow & Cao, 2008).

Critical Success Factors in Software Development.

Critical success factors in software development include several aspects, including the development life cycle, level of effort estimation, project management, and resource planning (Chow & Cao, 2008). Arcos-Medina and Mauricio (2020) asserted that critical factors are aspects that must be present for an agile project to be successful. In software projects, these factors can exist in several dimensions including organizational,

people, and technical that cover a wide range of factors regarding software development success factors (Arcos-Medina & Mauricio, 2020).

The software development lifecycle, the executive management, strategic level planning, and resource and project management are all potential factors (Arcos-Medina & Mauricio, 2020). Reel (1999) suggested that regardless of the development methodology, implementation language, or application domain people are one of the prominent success factors. Planning a software project requires having the resources to get the job done, the right people with the appropriate skillset to accomplish the task. Naslund and Kale (2020) stated a software development project should comprise developers who have similar work habits and comparable personalities.

Iqbal (2019) considered cost management a critical success factor of software development projects. A project manager is responsible for delivering the software within the scheduled time and cost scope while satisfying the stakeholder's needs (Iqbal, 2019). Ciric-Lalic et al. (2022) suggested that an agile or hybrid management approach is more successful than a traditional one concerning impact on project success dimensions. The primary focus of software development project managers is team efficiency (Oztaysi et al., 2019). Efficiency affects a project's budgetary and timeline scope (Reel, 1999). Software development is multifaceted and complex (Ciric-Lalic et al., 2022). Determining what a successful project is and what factors make it successful is equally complex. The one common resource throughout every study in literature is people. Executive and project management, software developers, and stakeholders all affect software development project success.

Alignment with the Agile Manifesto

The framework Chow and Cao (2008) developed aligns with the 12 principles of agile software development laid down in the *Agile Manifesto* (Martin, 2003). The framework's top three critical success factors (delivery strategy, agile software engineering practices, and team capability) are consistent with many attributes in the Manifesto (Tam et al., 2020). Delivery strategy correlates to the first and third principles about continuously delivering quality working software in short periods. Agile software engineering practices, parallel with the ninth and 10th Manifesto attributes: continuous attention to technical excellence with an emphasis on simple design (Tsoy & Staples, 2021). Lastly, team capability parallels the fifth practice: building projects around motivated individuals (Tam et al., 2020; Tsoy & Staples, 2021). The remaining nine critical success factors identified in the framework also correlate to several Manifesto practices. Project management success relates to the sixth and eighth practices of face-to-face conversation with the team and maintaining a consistent cadence. Team environment correlates with the 11th practice: a self-organizing team. The first and fourth practices regarding customer satisfaction and stakeholders working closely with developers align with the customer involvement critical success factor (Chow and Cao, 2008).

The *Agile Manifesto* principles are the foundation of the agile software development methodology. The correlation between the critical success factors framework (Chow & Cao, 2008) and the *Agile Manifesto* is relevant to this study by providing the history of principles and attributes of agile software development. Relevance demonstrates alignment with core agile principles and definitions while

providing a baseline for comparison of the correlation between project success, development team size, team experience, and accuracy of effort estimation.

Nonfactors

While Chow and Cao (2008) defined 12 critical success factors for agile software development projects, they also defined factors that do not appear to affect outcomes. Contrary to historical belief, executive support and sponsorship commitment are critical elements to project success, but the framework does not support this theory (Chow & Cao, 2008). In addition, agile-style workplace accommodations, including paired programming stations and communal areas, highly recommended in agile literature, are unsupported (Chow & Cao, 2008). The results suggest that the project team may adapt the environment to best suit their needs without following the rigid guidelines promoted in agile literature. Another nonfactor Chow and Cao (2008) identified was the organizational environment's role in agile software development. The framework defined that agile projects can be completed on time and within cost, regardless of the acceptance (or lack thereof) of agile-friendly factors, including collaborative culture, oral culture, and appropriate rewarding systems (Chow & Cao, 2008).

Use in Studies

Stankovic et al. (2013) used the 12 critical success factors Chow and Cao (2008) defined to study agile software. Stankovic et al. set out to replicate the research, except their population and response rates were vastly different from those of Chow and Cao (2008) though they did use the same instrumentation and data analysis techniques. Their population included developers and technical managers from software development

companies in the former Yugoslavia. They received only 23 responses to their inquiry. The study produced three potential additions to the list of critical success factors relating to timeliness and cost (Stankovic et al., 2013). Additionally, the study results could not confirm that the 12 critical success factors Chow and Cao (2008) defined affected software development project success rates. Although Stankovic et al. (2013) were unable to confirm the 12 critical success factors outlined in the Chow and Cao framework, other studies including Dikert et al. (2016), Tsoy and Staples (2021), and Chiyangwa and Mnkandla (2018) were able to confirm and validate them.

Dikert et al. (2016) conducted a systematic literature review regarding scaled agile transformations. They started with an initial population of over 2,000 papers and filtered them down to 52. Ninety percent of the 52 were experience reports, and six were empirical research. Through the process of analyzing, theming, and coding, Dikert et al. (2016) were able to identify 29 critical success factors relating to scaling agile. The 29 critical success factors were then grouped into 11 categories, and Dikert et al. (2016) also identified 35 challenges into nine categories. Many of the critical success factors identified by Chow and Cao (2008) were present in the study results. The most notable differences were requirements engineering, communications/coordination, and quality control. Dikert et al. (2016) focused on large-scale software development organizations comprised of a minimum of 50 people or at least six development teams. The study relates to this study in scope to the problem statement by understanding the correlation between development team size and agile software development project success. This study is limited to agile software development projects comprised of a minimum of three

members with no defined upper bound. The findings of the study conducted by Dikert et al. (2016) give credence to the critical success factors of agile software development projects identified by Chow and Cao.

Tsoy and Staples (2021) used the framework Chow and Cao (2008) developed to study critical success factors for agile analysis projects. They performed four case studies, identifying potentially 43 new attributes that affect the existing list of 12 critical success factors. In addition, Tsoy and Staples conducted interviews with key project stakeholders, team members, and clients. They measured success through the business implementation/deployment as the primary indicator. Tsoy and Staples focused on data analysis, big data, and business intelligence. They found that substantial customer involvement and a systematic project definition process are critical success factors for successful agile analysis projects.

The study conducted by Tsoy and Staples (2021) complements Chow and Cao (2008) in two ways: (a) they performed an objective assessment of project performance, and (b) they studied a smaller number of studies, albeit in much more depth. Their findings are consistent with Chow and Cao (2008) regarding agile software development projects, adding to the framework's validity. The variance between results is evident in the team capability factor due to the high capability of the development teams studied (Tsoy & Staples, 2021). This study relates to the problem statement through the correlation between project success and team experience attributes of agile software development projects.

Chiyangwa and Mnkandla (2018) performed a similar study to Stankovic et al. (2013), studying critical success factors in agile software development projects in South Africa. They used the framework of critical success factors Chow and Cao (2008) developed to compare with their study results. Specifically, they focused on the impact of organizational factors on the success rate of agile software development projects. This study aims to understand the correlation between agile software development team factors including experience, size, and accuracy of level of effort estimation. Organizational factors, including staff availability and experience levels, are indirectly related to development team size and team experience attributes. Therefore, the Chiyangwa and Mnkandla (2018) study validates the organizational dimension factors identified in the Chow and Cao (2008) framework.

Misra et al. (2009) conducted a large-scale empirical study to identify agile success factors for organizations of significant size. Nine statistically significant factors were identified: (a) customer satisfaction, (b) customer collaboration, (c) customer commitment, (d) decision time, (e) corporate culture, (f) control, (g) personal characteristics, (h) societal culture, and (i) training and learning (Henriksen & Pedersen, 2017). The studies reviewed by Misra et al. (2009) indicate many success factors; however, there is significant overlap between them. The critical success factors can be narrowed to those likely to affect an agile software development project. The critical success factors identified can be aligned with those of the critical success factors of the agile software development framework identified in the Chow and Cao (2008) study. The evident alignment of elements adds validity to the framework. Additionally, it relates to

the problem statement of understanding the correlation between agile software development team size, experience, level of effort estimation accuracy, and project success rates.

Contrasting Theories. Khan et al. (2021) developed a framework similar to Chow and Cao (2008) that focused primarily on critical success factors for Global Software Development (GSD) companies based in China. Their literature review produced 23 success factors deduced into four categories organization, team, technical, and process. Similar to Chow and Cao (2008), that used organization, people, technical, and processes. Khan et al. (2021) theorize that communication and control are the highest-rated critical success factors for GSD companies. Chow and Cao (2008) suggest that agile software engineering techniques' delivery strategy and proper practice significantly contribute to software development project success. Khan et al. used a Fuzzy AHP approach with pairwise comparisons to reveal the most prioritized category of critical success factors for GSD projects.

In contrast, Chow and Cao (2008) used multiple regression to determine the most prioritized category of critical success factors. Khan et al. (2021) suggest that agile software development projects will be successful if the team is considered high-caliber and practices rigorous agile software practices. In contrast, Chow and Cao (2008) identified delivery strategy as the most critical success factor for successful agile software development projects.

Arcos-Medina and Mauricio (2020) conducted research to identify relationships between agile practices and critical success factors. Arcos-Medina and Mauricio (2020)

recognized that many previous studies had analyzed associations between the influence of agile practices on quality and the impact of critical success factors on quality. Additionally, Arcos-Medina and Mauricio (2020) recognized that many studies have helped identify factors and patterns that influence agile development. Arcos-Medina and Mauricio (2020) theorized that restrictions placed on agile practices by organizational structure directly affect the success of software development projects. Software quality can be improved through critical success factors identified in using agile techniques. Arcos-Medina and Mauricio (2020) chose six critical success factors from human and organizational behavior theories and categorized them into two dimensions: the personnel dimension and the organization dimension. They found that 10 of their 13 proposed hypotheses significantly influence agile practices. The primary factors are confidence, perceived self-efficacy, and the integrity of the information, and experiences learned positively impact agile practices. The results of the Arcos-Medina and Mauricio (2020) study contrast with Chow and Cao's (2008) study, where agile practices were studied versus agile development.

Measurement

Rockart and Crescenzi (1984) presented a three-phase process that successfully engages top management in information technology. The first phase involved an introductory workshop where five management team members identified the company's information needs. Three benefits emerged from the workshop: a managerial perspective for systems development, establishing business priorities, and active involvement (Rockart & Crescenzi, 1984). The second phase involved conducting interviews to

identify critical success factors. Rockart and Crescenzi (1984) found that critical success factors are key areas in which successful performance contributes to the individual manager's objectives. A corporation views critical success factors as key areas to focus on to achieve its objectives (Rockart & Crescenzi, 1984). The third and final phase is a focused workshop. During the workshop, Rockart and Crescenzi solidified critical success factors identified in the initial workshop and interviews. Although the study refers to critical success factors relating to IT management and business, Rockart and Crescenzi developed the concept of identifying factors crucial for success.

Chow and Cao (2008) utilized the critical success factor approach developed by Rockart and Crescenzi in 1984. The approach they used was to identify any attributes that affect agile software development project outcomes and categorize them. Rockart and Crescenzi (1984) defined this approach during their study of how engagement of top management affects project outcomes. Chow and Cao employed a web survey to gather data. The web survey included Likert scale questions and demographic information broken into four sections. The first section is related to demographic information and agile project information. The second section comprises questions that measure the importance of success factors. A 7-point Likert scale was used to reflect the respondent's perception of the question. The third section related to the perception of software development project success and used a 7-point Likert scale to reflect the respondent's perception level. The last section of the web survey provided the respondent with a free-form text area, which could be used for additional comments or clarification. In order to prevent ambiguity in the perception of success on the part of the respondent, the

questions focused on a single project of the respondent's choice. In addition, Chow and Cao (2008) tested the web survey content validity by executing a pilot process. They sent the web survey to five members of the Agile Alliance and requested feedback for improving the survey. After a six-week survey period, 408 people responded, and 109 distinct projects were submitted. Multiple studies have used Chow and Cao's (2008) study to identify critical success factors relating to software development, thus enhancing its validity and reliability.

Misra et al. (2009) administered a web-based questionnaire to many agile software development practices. The questionnaire was developed from combined elements of the hypothetical framework constructed and hypotheses. Respondents were asked to fill out a structured questionnaire comprised of close-ended, multiple-choice, and open-ended questions. The open-ended questions were not used in the statistical analysis but were a means for the respondent to provide additional information. The multiple-choice questions used a 5-point Likert scale. Misra et al. (2009) set an initial target sample size of 150 respondents that practiced at least six of the twelve agile software development principles and had been a part of a team that used to practice traditional development practices. A total of 241 responses were received, of which 174 were eligible based on the above criteria. Misra et al. (2009) deemed it necessary to pre-test the questionnaire before submitting it publicly. They submitted the survey to 10 qualified software development professionals, of which only five responded. The five respondents worked in private and public sectors with varying experience levels and

work functions. The pre-test helped clarify any ambiguities, the ordering of questions, and its readability.

Tam et al. (2020) used an online survey to collect data to determine how people factors identified by Chow and Cao (2008) and Misra et al. (2009) explain the success of agile software development projects. The target population was individuals involved in at least one agile software development project as team members or stakeholders. Tam et al. limited the survey to Portugal; however, the industry was not restricted. The survey comprised multiple-choice questions and used a 7-point Likert scale to understand the respondents' agreement level toward each item except the perceived level of success questions. Similar to Chow and Cao (2008) and Misra et al. (2009), Tam et al. (2020) conducted a pre-test of their survey by asking five agile-knowledgeable professionals to provide feedback. The survey was modified to account for the feedback and re-released to an additional validity pre-test group of 30 agile professionals. A total of 600 invitations were sent, resulting in 242 respondents. In addition, Tam et al. (2020) used a standard method bias test using the variable marker technique (Lindell & Whitney, 2001; Malhotra et al., 2006) to ensure no systematic bias was influencing the data.

Development Team Experience

Many social and technical factors contribute to software development team inefficiency and ineffectiveness (Stankovic et al., 2013). According to Misra et al. (2009), competent teams deliver quality software fast that meets or exceeds stakeholder requirements. The capabilities of team members are influenced by understanding and knowledge of agile development techniques and other factors (Tam et al., 2020). The

composition of a software development team plays a significant role in the outcome of a software development project (Kasauli et al., 2021). One must analyze several attributes of a team member when assembling a software development team. Characteristics, including but not limited to the level of development experience and familiarity with the methodology, affect the contribution of a given member (Bogopa & Marnewick, 2022). A project development team may include but is not limited to project managers, business analysts, scrum masters, and software developers (Almadhoun & Hamdan, 2018). The expertise needed in a development team includes technical skills (programming languages and methodologies) and teamwork, process, product, and domain knowledge (Šmite et al., 2017). In addition, the team must be familiar with the architecture, source code structure, and allocation of concepts contained therein.

Studies with varying methodologies have shown that the length of experience directly correlates to project outcomes. Bogopa and Marnewick (2022) conducted a study in South Africa to identify critical success factors in software development projects. Their study asked respondents to provide the number of years of experience they have involved in software development. Of 212 study participants, 78 (36%) had more than ten years of experience (Bogopa & Marnewick, 2022). The largest category of respondents fell into the 1-5 year experience range at 29.7% (Bogopa & Marnewick, 2022). Bogopa and Marnewick (2022) found that 7% of respondents strongly disagreed that team experience affected project outcomes, whereas 48% strongly agreed that team experience affected project outcomes. Butler et al. (2020) conducted an international study that asked respondents to identify their experience level. Of 121 respondents, 76

(62%) reported more than ten years of development experience. Like Bogopa and Marnewick, Butler et al. (2020) found that the level of developer experience directly affects project outcomes.

Garousi et al. (2019) found similar results, with 54% of respondents identifying developer experience with the task and domain affecting project outcomes. In addition, they found that the team's experience level with the development methodology also contributes to project outcomes. The experience level of the respondents leaned more toward a less experienced pool, with only 30% having more than ten years of experience. Compared to previous studies, Garousi et al. (2019) identified the project team's experience with the software development methodology as the third highest critical success factor that affects project outcomes. Misra et al. (2009) found that technical competency and experience level have no statistically significant correlation with project outcomes. However, they determined their results counterintuitive because inexperienced team members should impact project outcomes. In addition, the results contradict Boehm's principle of using fewer and better people (Boehm, 1981; Misra et al., 2009).

Development Team Size

Team size relates to the number of members or teams that may be co-located with a shared understanding regarding the project (Butt et al., 2022). The team size may fluctuate in the agile scrum methodology based on the requirements and the sprint's deadline. Research has shown that team size correlates to project outcomes. Butt et al. (2022) found that some developers perform better with larger teams than smaller ones. In addition, the range of experience and skill levels can affect the team size requirements. A

combination of highly and less experienced developers in every project promotes a positive project outcome (Gren et al., 2017).

Tsoy and Staples (2021) found an adequate small-scale agile software development team size of ten or fewer members. Other studies have measured team size by the number of collaborating and coordinating teams. Examples of large-scale sizes include 40 people and seven teams, a project cost of over 10 million, and a group of around 50 people (Berger & Beynon-Davies, 2009; Dikert et al., 2016; Paasivaara et al., 2008). It is noteworthy that early agile studies identified size to be variant depending on the methodology. According to Fowler, the Crystal methodology suits up to 50 people and is considered small-scale (Dikert et al., 2016; Fowler, 2000). Studies are ongoing regarding the definition of small and large-scale development team sizes. Bermejo et al. (2014) suggest that the perfect team size is subjective to the requirements defined in the early stages of product development.

Research has shown that development team size significantly impacts software projects' schedules and budgetary deviations. Kula et al. (2022) found that team size significantly affects schedule deviations in epics, thus affecting project outcomes. Almadhoun and Hamdan (2018) used a genetic algorithm to determine the optimal development team size. They found that self-organizing teams of sizes ranging from five to nine members undercut timeline and budgetary constraints compared to larger-sized teams by 40%. The size of the development team correlates with several factors of a development project. Chow and Cao (2008) found that the size of the development team affects the overall cost, timeline, and quality of work in a software project.

Level Of Effort Estimation Accuracy

The level of effort and cost estimation is an essential process during agile software development. Multiple studies have researched factors that need consideration for practical estimation. The process of estimating, under the agile framework, is the practice of estimating development cost and effort required to complete a task (Govil & Sharma, 2022). Development cost is monetary, while effort is measured regarding the time required to complete the task (Freire et al., 2018). An estimate can be interpreted from the requirement of resources, such as tools and people (Bourque & Fairley, 2014; Sudarmaningtyas & Mohamed, 2021). In software development, effort estimation is significant in establishing delivery schedules and cost assessments (Bilgaiyan et al., 2017). According to Bilgaiyan et al. (2016), a software project's success relies on estimations that accurately predict effort. Software development projects within budget, on time, and delivering expected results are the exception, not the rule (Rahikkala et al., 2015). Project scheduling, budgeting, and planning are all affected by the accuracy of effort estimations. Effort estimations can provide forecasting data to aid in assigning resources and planning.

Multiple factors contribute to inaccurate effort estimations. Uncertainty about the effort required to complete a task, inconsistent and incomplete data, environmental dependency, software complexity, estimation experience, and frequent changes to requirements are all factors that harm software development level of effort estimation accuracy (Sehra et al., 2016). According to Tanveer (2017), the accuracy of effort estimation depends on the developer's experience, complexity, and the impact of changes

requested to the system. In addition, several estimation models are available, and each performs differently depending on the environment and project type (Sehra et al., 2017). Effort estimation in software development can be difficult considering multiple factors, and underestimation is commonplace in software development projects.

Several factors are influential during the process of software development estimation. The complexity level of the tasks, the team's overall experience, the skill of the team as a whole, the number of nonfunctional requirements, the level of involvement from the stakeholder or product owner, and the geographic distribution of the team (Usman et al., 2017). Software development level of effort estimation accuracy can also be affected by changes in requirements, data completion, uncertainty and clarity of requirements, and estimator experience (Sehra et al., 2016). According to Jørgensen et al. (2021), using local context, historical estimation error intervals, a checklist, and avoidance of estimating effort based on incomplete information are all methods of improving estimation accuracy. Effort estimation is adversely affected by multiple factors, including individual and collective.

Providing accurate effort estimates in software development is problematic. According to Sehra et al. (2016), the cone of uncertainty makes estimating effort at the beginning of the development lifecycle challenging. During the initial feasibility stage, actual cost and effort can exceed more than the initial estimate by over 250%

(Dragicevic et al., 2017). Accuracy can improve if estimation is delayed until the requirement specification phase (Sehra et al., 2016). Estimations are often requested before the requirements are elaborated upon. Uncertainty decreases as knowledge increases as the project progresses (Arifin et al., 2017). Each project is unique, and no two projects have the exact requirements, so estimates are predictions with some uncertainty.

Agile software development projects pose more of a challenge in estimating effort due to the changing requirements and the uncertainty that may be experienced in the development process. According to Rahikkala et al. (2015), two factors contributed to project success regarding estimates identified by senior management. The first factor was that estimates are critical to organizational success (Rahikkala et al., 2015). The second factor was that software estimations rely on facts rather than opinions or guessing. (Rahikkala et al., 2015). Estimation accuracy increases the probability of project success; however, the most effective method still needs to be discovered.

Agile Estimation Methods

Algorithmic and non-algorithmic are the two primary categories of software estimation methodologies. All estimation methods are either parametric, nonparametric

or a combination of the two (Idri et al., 2015; Osmanbegović et al., 2017; Soni & Kohli, 2017). Parametric or algorithmic approaches use mathematical equations or models, whereas nonparametric do not (Khuat & Le, 2016).

Development teams use multiple strategies to provide estimates of effort. Shekhar and Kumar (2016) suggested that combining estimation techniques is considerably better than using one method. Projects with extensive known requirements benefit from non-algorithmic approaches using expert judgment (Shekhar & Kumar, 2016). The algorithmic approach suits projects where there are many unknown requirements. However, a mix-method or combination approach produces more accurate estimates (Shekhar & Kumar, 2016).

The agile effort estimation philosophy is that the team doing the work also performs the estimation to obtain a realistic assessment (Taylor, 2016). Some estimation models are better suited for different development models (Prakash & Viswanathan, 2017). Successful estimation models have distinct characteristics, including estimations created by team members, collaboration with product owners, and usage of story points for relative measurement (Prakash & Viswanathan, 2017). The early stages of agile software development projects often need more refined requirements or details. Therefore, obtaining accurate effort estimations early in software development becomes problematic.

Story Points

A story point relatively represents a user story or development effort's overall size. The most popular estimation method is to use story points to estimate the effort

needed to implement a user story (Choetkiertikul et al., 2018; Dragicevic et al., 2017). In estimation, the complexity of the work is calculated from the bottom up (Jadhav et al., 2017). Many standard formulas define the size or effort in story point utilization (Osman & Musa, 2016). One standard method uses the Fibonacci sequence to express relative size (Alostad et al., 2017; Jadhav et al., 2017; Raslan et al., 2015). The gaps between the sequences represent a higher level of uncertainty in the level of effort. The larger the action, or the more significant the size, the estimation error is more likely (Raslan et al., 2015). When the Fibonacci sequence is used as an estimation metric, it is considered relatively unbiased (Fox, 2016). Another comprehensive approach, as suggested by Usman et al. (2017), is to provide estimates by averaging the fastest, the maximum, and most practical values to generate a final assessment.

Story points are relative values and numerical representations of complexity (and size), and they can differ from development team to development team. Each time, it uses a specific estimating approach based on the cumulative knowledge of its members (Choetkiertikul et al., 2018). In addition, the value assigned to a story point can change from team to team depending on the relative baseline story (Choetkiertikul et al., 2018; Soni & Kohli, 2017). Story points are used in varying scales to determine velocity over time (Ahmed et al., 2017). Story points provide relative measurements rather than quantitative ones (Soni & Kohli, 2017). The estimation accuracy of story points is subjective to the person or team conducting the estimation; it is often derived from experience (Arifin et al., 2017).

One of the critical success factors of story point estimation is that the development team estimates user stories using the same scale. The story point estimates are translated into the team's velocity to forecast future iterations, efforts, and sprints (Brad et al., 2016). The team's velocity represents the total story points the development team can deliver in a given iteration (Soni & Kohli, 2017). The velocity is also a valuable predictor of the team's development capabilities (Ahmed et al., 2017). Independent of time units, story points are a common and successful approach in software development effort estimation (Zahraoui & Idrissi, 2015). Harzl (2017) suggested that using a team's shared experience as a critical success factor can be disadvantageous. In the initial stage of a software development project, it is challenging to establish velocity because team members are inexperienced in estimating and providing accurate estimates collaboratively (Harzl, 2017). Although the story point approach is commonly used, it is subject to error.

T-Shirt Sizing

T-shirt sizing is an alternative estimation approach using relative valuations. The naming of the approach lends to its use of t-shirt sizes ranging from extra-small, small, medium, large, and extra-large (Alostad et al., 2017; Raslan et al., 2015). The approach requires a common understanding of the estimated value, and, similar to story points, it can vary from team to team (Alostad et al., 2017). According to Alostad et al. (2017), t-shirt sizing works best when the development team has prior experience estimating tasks and when a significant task needs estimation. The t-shirt sizing approach aids early assessments that give organizations a complexity metric to determine the level

of effort (Harzl, 2017). Stakeholders often use early estimates to determine if the level of effort is worth the value generated from the action (Conoscenti et al., 2019). The effort estimation process can be complex, and t-shirt sizing is a simple and efficient alternative.

One advantage of t-shirt sizing is fewer relative size values, resulting in an expeditious voting process. The t-shirt sizing approach does not suggest precision due to its abstract nature (Harzl, 2017). T-shirt sizing is non-numerical, making the method simple and easy to understand (Harzl, 2017). Another advantage of t-shirt sizing is that it provides an initial non-technical estimation that can be used to support effective project control (Conoscenti et al., 2019). Some disadvantages of the t-shirt sizing approach are that the scale needs more detail, mathematical size correlation, and difficulty measuring velocity (Harzl, 2017). Conoscenti et al. (2019) assert that establishing velocity or team performance over time is problematic due to the need for numerical values.

Expert Judgment

Expert judgment uses an expert who understands the task and has prior experience estimating effort. According to Shekhar and Kumar (2016), expert judgment utilizes an expert's knowledge and is a commonplace strategy for estimating effort. Expert judgment relies on the estimator and their previous experiences with similar projects; however, the estimator can exhibit bias (Khuat & Le, 2016). Arifin et al. (2017) assert that expert judgment comprises two approaches: effort size and effort time. Effort size is a relative effort measurement, such as t-shirt sizing or story points, whereas effort time is an absolute value, such as person-hours or person-days (Arifin et al., 2017). Decomposing

tasks, through a top-down approach, into an effort time of less than two days enhances the accuracy and effectiveness of expert judgment (McConnell, 2006).

Expert judgment is a common technique used in software development effort estimation. Commercial estimation tools are highly-available; however, expert estimation remains the most widely accepted estimation methodology (Ivan & Despa, 2016; Shekhar & Kumar, 2016; Usman et al., 2017). Expert-based techniques are beneficial when restrictions exist for retrieving quantified empirical data (Conoscenti et al., 2019). Expert judgment is sometimes inconsistent, overly dependent on human memory, and lacks repeatability due to its non-algorithmic nature (Sehra et al., 2017). Experts may be over-optimistic and over-confident leading to estimation inaccuracies.

Delphi

The Delphi technique combines the experiences of several experts to come to a consensus-based estimation approach. The Delphi approach requires the experts to have specific application knowledge and domain experience (Adnan & Afzal, 2017; Strasser, 2017). Structured group discussions are conducted to produce a consensus (Osman & Musa, 2016; Perkusich et al., 2017). According to Rai et al. (2017), the Delphi estimation process begins with a group of experts identifying a task to estimate, then providing and discussing the application of the estimation method, and arriving at a consensus regarding the level of effort. The experts employ a method that utilizes multiple rounds of voting to yield results that can be assessed and summarized (Lima et al., 2016). There are

variations of the Delphi process where the experts may repeat the voting process until they reach a consensus or until the answer is satisfactory (Nguyen & Nguyen, 2018). The Delphi estimation technique provides an estimate based on the consensus of experts.

The Delphi method uses expert opinions, calls for team cooperation, and specifies the conditions under which team members must confidently do their estimations. Prakash and Viswanathan (2017) state that varying estimations require further discussions and re-evaluations. After each round, the results are shared with the group for further discussion to reach a consensus and assess the consensus for appropriateness (Bilgaiyan et al., 2016). The Delphi technique provides a defined practice for assessing effort estimation by capturing factors from multiple experts (Rai et al., 2017). The Delphi method utilizes the collective assessments of experts to determine an agreed-upon estimation.

COCOMO II

Software effort estimation can be achieved through an algorithmic approach, such as the COCOMO (Constructive Cost Model) II model. COCOMO II uses size and numerical input measures regarding application points multiplied by empirically determined constants to provide estimations (Pereira dos Santos et al., 2019). Organization-specific calibration of the constants through historical data increases accuracy (Moharreri et al., 2016). One advantage of the COCOMO II model is the calibration of the model with previous projects and experiences, and development factors (Osmanbegović et al., 2017). Model calibration to previous experience and development factors is imperative in keeping the model efficient.

The algorithmic nature of the COCOMO II model relies heavily on historical data and other developmental factors to remain efficient. Developmental factors include risk, platform experience, the volatility of the platform, team cohesion, development team flexibility, architecture, reuse, continuity of personnel, complexity, team capability, and complexity (Boehm et al., 2000). According to Prakash and Viswanathan (2017), another advantage of COCOMO II is that the customization and modification capabilities of the model are straightforward. The method becomes less effective if the historical is unavailable (Prakash & Viswanathan, 2017). In addition, the COCOMO II model aligns better with a procedural development model than the agile development model (Kukreja & Garg, 2017; Rath et al., 2016).

Bayesian Network

As part of the probabilistic graph model category, Bayesian networks represent knowledge about uncertain domains (Perkusich et al., 2017). Bayesian networks represent a joint probability distribution over variables (Freire et al., 2018). The Bayesian network model is suitable for effort estimation in the agile software development methodology because it is applicable in early planning phases and does not directly impact agility (Dragicevic et al., 2017). The model generates diagnostics, predictions, and probabilities with ambiguous data (Dragicevic et al., 2017). Effort estimations use Bayesian networks to incorporate various factors to calculate conditional probability in estimations.

The Bayesian network model calculates relationships between casually related variables that have the potential to be probabilistic. The model has advantages for small

projects and projects where incomplete data sets are provided (Zare et al., 2016). Perkusich et al. (2017) assert that the model adds an advantage through the support of decision analysis and the explicit treatment of uncertainty. Bayesian networks are helpful in effort estimation because probability distributions can be updated as new information and causal influences become available (Perkusich et al., 2017). Bayesian networks account for the combination of expert judgment and historical data.

Planning Poker

Planning poker is used widely in agile software development teams to estimate effort (Prakash & Viswanathan, 2017; Soni & Kohli, 2017; Usman et al., 2017). The estimation method minimizes peer pressure using a consensus approach (Taylor, 2016) and can be used if historical data is unavailable (Anooja & Rajawat, 2017). Planning poker begins with a domain expert explaining the user story and providing any requested information or clarification (Lopez-Martinez et al., 2017; Torrecilla-Salinas et al., 2015). Next, the team members create a private preliminary estimate using cards representing a value, followed by disclosure to the entire team (Torrecilla-Salinas et al., 2015). The team members must explain their reasoning behind the estimation value while all team members reflect on said explanations (Miranda, 2017). If estimations differ significantly, additional estimation rounds may be required (Miranda, 2017; Torrecilla-Salinas et al., 2015). Each estimation round consists of estimators with the lowest and highest values providing reasoning. Subsequently, rounds are conducted as a team until a consensus on the estimation value is made (Torrecilla-Salinas et al., 2015; Vyas et al., 2018).

Developers use a collective forum with open discussions, similar to the Delphi method, to reach a group-based estimation agreement.

Planning poker is a team-based estimation methodology that assigns relative estimate values to software development requirements to produce a specific feature. The method uses a numerical sequence such as Fibonacci (Ramirez-Noriega et al., 2016). Additionally, planning poker combines the requirement of expert judgment and relative numerical sequences to provide accurate effort estimates (Osman & Musa, 2016; Usman et al., 2017). Estimations created using the planning poker technique result in a value estimation of effort and are based on the consensus of the development team. Dönmez and Grote (2018) state that planning poker is a team-based method that is incremental and collectively analyzes requirements resulting in an estimation. The method aims to derive an estimation that will persevere future scrutinization (Osman & Musa, 2016). Introduced in 2002 by Grenning, planning poker combines expert opinion, disaggregation, and analogy into a reliable and efficient estimation method (Rai et al., 2017). The most effective use of planning poker is when an expert is engaged with team members with previous experience with similar tasks.

Artificial Neural Networks

Artificial neural networks (ANN) are known for their generalization and learning abilities and are used as data analysis tools (Rashid et al., 2023). Inspired by biological neural networks, ANN is an algorithmic model where each connection between neurons is associated with a weighted value (Prakash & Viswanathan, 2017). The associated weights have attributes regarding the input signals and are multiplied by corresponding

inputs to solve a specific problem (Rankovic et al., 2021). Neural networks use historical data to adjust algorithmic parameters to reduce estimate prediction errors automatically (Rashid et al., 2023). Common types of neural networks used are polynomial, probabilistic neural networks, and general regression networks (Prakash & Viswanathan, 2017). According to Ali et al. (2023), ANN models applied in effort estimation are the function link artificial neural network (FLANN) and the radial basis function network (RBFN). Rankovic et al. (2021) state that the RBFN model has a strong tolerance for noise, significant learning ability, and offers a straightforward design. The FLANN method is preferred when using nonlinear data (Ali et al., 2023). An ANN is a framework of learning algorithms and is not an algorithm itself, even though it is considered an algorithmic process.

The design of ANNs is inspired by the biological nervous system, which processes information using nodes to operate through weighted inputs to provide accurate effort estimates (Bilgaiyan et al., 2017; Mittas et al., 2015). Kaur (2017) states that ANNs utilize pattern matching through machine learning to discover relationships between dependent and independent variables. ANNs have increased in popularity as an effort estimation prediction technique due to the ability to capture complex data and filter out noise (Pospieszny et al., 2018). ANN uses historical data to infer outputs through learned data (Ali et al., 2023). In addition, the ANN is most effective when the amount of historical data is considerable, thus providing a more accurate estimation (Naik & Nayak, 2017).

Function Points

The function point method calculates using inputs, outputs, files, and inquiries to estimate effort. Introduced in 1979 by Albrecht (Hans & Gahlot, 2016), the method aims to form an end user's perspective and measure the size of data processing systems to determine an effort estimation (Abualkishik et al., 2017). This method allows estimators to calculate effort for a specific use case or when system analysis is unavailable (Dewi & Subriadi, 2017a). Function points are calculated by summating external interface files, inquiries, external outputs, and internal logical files (Hans & Gahlot, 2016; Yoshigami et al., 2017). Function points are numerical that determine the size of the software from a data processing viewpoint rather than from developmental complexity.

The function point method is best suited to estimate size and cost rather than estimate effort (Prakash & Viswanathan, 2017). Abualkishik et al. (2017) state that function points represent functionality available to the end user through data transaction analysis and determining operations that involve application interaction. One advantage of the function points method is that estimations of size of the software can be projected early in the project life cycle (Qi et al., 2017). Another advantage is that the method is development technology agnostic and can be applied to any software development project using any methodology (Hans & Gahlot, 2016; Yoshigami et al., 2017).

COSMIC

Seasoned software measurement experts created the Common Software Measurement International Consortium (COSMIC) to overcome issues experienced with function point measurements. The consortium defined standards, rules, and principles for

measuring the functional size of software (Almakadmeh et al., 2018). The COSMIC approach focuses on data storage, input, and output characterized by software development projects which is the opposite of the traditional function point method (Prakash & Viswanathan, 2017).

COSMIC is a second-generation method of assigning function points to address the areas where the function method fell short. Input and output movements, including the reading and writing of data to storage, are used in COSMIC calculations and classified into entry and exit points (Almakadmeh et al., 2018). Each entry and exit point represents one COSMIC function point size that is summed to represent the total size of the functional processes (Abualkishik & Lavazza, 2018). The COSMIC method calculates the size of the software application through the related number of movements; therefore, the more movements, the more significant size of the software.

Use Case Points

Use case points (UCP) utilizes the unified modeling language to diagram use cases to estimate the size of the software. Inspired by the function point method, the UCP method, developed by Karner in 1993, is suited for use in the early stages of software development (Effendi et al., 2019). Karner developed the technique for object-oriented software development to coincide with effort estimation (Nhung et al., 2022). According to Mehta and Kumari (2016), UCP estimation applies better to object-oriented development compared to COCOMO and function points. The UCP method calculates complexity from use case definitions while function points count data movements, and COCOMO uses historical data (Effendi et al., 2019).

When using the UCP method, the first step is to calculate the complexity of the interaction and the unadjusted actor's weight (Mahmood et al., 2020). The next step is categorizing the transactions as simple, average, or complex (Mahmood et al., 2020). The response between an actor and the system is represented as transactions. The final step is to adjust the complexity values based on an environmental adjustment factor and the technical complexity of the transaction (Nhung et al., 2022).

The UCP technique has been utilized broadly for quite some time (Azzeh et al., 2021). In addition, studies have shown that the method is reliable (Dewi & Subriadi, 2017b). One disadvantage of the UCP method is that the assigned values are arbitrary; therefore, providing time-based estimates is challenging (Mahmood et al., 2020). According to Dewi and Subriadi (2017b), size is not directly proportional to time-based effort.

Software Development Project Success

Software development project success can be defined in multiple ways. The definition can vary depending on the connotation. A project manager's definition of success may be based on if the project meets budget, timeline, and quality constraints (Bogopa & Marnewick, 2022). The stakeholder may define project success if the completed project benefits the organization through maintaining a competitive advantage, increasing sales and profits, or enhancing a brand image (Bogopa & Marnewick, 2022). According to Burke (2011), there are three traditional constraints of software success, cost, time, and quality. Other authors identify additional conditions like risks. Gandomani et al. (2013) assert that regardless of the approach to the project, projects require

planning, budgeting, and scheduling to be successful. Butler et al. (2020) defined project success as the extent a project achieves project goals within the allotted schedule, budget, productivity, and quality. The methods used to measure project success have evolved. First, the iron triangle for project assessment and critical success factors was developed, leading to today's success frameworks (Tam et al., 2020).

Success Frameworks

Morris and Hough's (1987) pioneer framework defined project success elements by analyzing the functionality, management, contractor performance, and termination. Functionality refers to the financial and technical requirements, while management assesses whether the project meets schedule and budget restrictions (Tam et al., 2020). Contractor performance determines if the contractors have obtained a benefit from the project. Finally, project performance and termination concerns making a reasonable decision to cancel or terminate a project (Tam et al., 2020). Pinto and Slevin (1988) developed a framework for implementation success containing three elements: technical validity, organizational validity, and organizational effectiveness (Gandomani et al., 2013). Technical validity assesses whether the project is practical and technically sound. Organizational validity considers client requirements, satisfaction level, and usability (Tam et al., 2020). Finally, organizational effectiveness considers the project's positive contributions to the organization (Pinto & Slevin, 1988).

Munns and Bjeirmi's (1996) framework requires progression throughout the implementation phase, increased customer satisfaction, and the end-user's perception of the product. Belassi and Tukel (1996) developed a holistic framework of firm and

industry-related factors. The authors identified a need to classify success factors for categorization associations (Tam et al., 2020). Classification of success factors provides a method for correlational analysis between success factors. Shenhar et al. (1997) proposed a multidimensional framework to assess project success. The framework focused on project efficiency, customer impact, organizational success, and future preparation (Shenhar et al., (1997). The authors made a clear distinction between resource-related and performance-related factors. Resource-related factors refer to time and budget, while performance-related factors involve customer satisfaction (Tam et al., 2020). Lester (1998) asserted that project success depends on constant interaction between stakeholders and the organization.

Project Success Factors

Studies have shown that complexity and dynamism influence project success (Butler et al., 2020). Kandengwa and Khoza (2021) identified additional project success factors beyond the triple constraint of time, scope, and cost. They recognized business value, project visibility, process scope, and process improvement as success factors. Customer satisfaction is often used as a metric to measure success because it is essential to satisfy customers (Bergmann & Karwowski, 2018; Kandengwa & Khoza, 2021). Chiyangwa and Mnkandla (2018) assert that software development project success depends on factors such as project leadership and the associated management style. In addition, they suggest that agile project leadership lacks experience in dealing with people and cultural factors, process and project characteristics, and technical failure

factors (Chow & Cao, 2008; Cohn & Ford, 2003; Dybå & Dingsøyr, 2008; Koch, 2005; Lindvall et al., 2004; Misra et al., 2009).

Tam et al. (2020) found that the relationship between societal cultures, personal characteristics, and team capability can significantly impact project outcomes. They emphasize the importance of understanding personal factors and societal cultures when considering customer involvement. Their findings confirm those of Misra et al. (2009) that results of individual characteristics and societal culture are used to explain customer involvement. Bermejo et al. (2014) also identified that organizations with high success rates have higher capacity rates regarding culture, team, communication with the client, and environmental configuration (Bergmann, 2018). Tam et al. (2020) confirm the theory that team capability and customer involvement lead to agile software development project success; however, results also indicate that team capability significantly impacts project outcomes. Garousi et al. (2019) state that team experience with the software development methodologies used, the team's experience with the task, and project monitoring and controlling are all essential factors contributing to an agile software development project's success.

Methodologies

There are three primary methodologies used in research studies. They are quantitative, qualitative, and mixed-method. Each methodology has strengths, weaknesses, and a specific purpose. The research study's purpose determines the methodology used. The quantitative method tests a hypothesis and measures something (Creswell & Poth, 2018; Matović & Ovesni, 2023). Quantitative studies generally deal

with numerical analysis to determine correlational factors amongst datasets. Qualitative studies explore ideas, thoughts, and meanings (Creswell & Poth, 2018). Qualitative studies include any data that is nonnumerical. Mixed-method studies combine qualitative and quantitative methods to determine a correlational and phenomenological understanding of a topic (Creswell & Poth, 2018; Matović & Ovesni, 2023). Software development project success is a multi-faceted and complex topic primarily because there is no absolute definition of a successful project. Therefore, researchers have used all three study methodologies to try and understand the topic.

Qualitative

The literature has shown that several studies used a qualitative research approach. Kasauli et al. (2021) conducted a multiple case study with seven large-scale systems companies. They focused on identifying challenges with two popular large-scale agile frameworks and their impact on project success. The study involved 20 qualitative interviews, five focus groups, and eight cross-company workshops. Tsoy and Staples (2021) also used a multiple case study of four projects throughout their life cycle. The purpose of using the qualitative method was to fit their research question and allow them to examine contemporary events with complex contextual conditions, focusing on software development project success. The authors collected data for each project by interviewing 29 key stakeholders, project team members, and project leadership.

Quantitative

Chow and Cao (2008) used a quantitative approach for their study. Their goal was to identify critical success factors of agile software development projects. They obtained

data via survey resulting in 109 participants from all over the world. Using multiple regression analysis, the authors differentiated success and failure factors in agile software development projects. Stankovic et al. (2013) conducted an extension to the exploratory study performed by Chow and Cao (2008); therefore, they also used a quantitative approach. Like Chow and Cao (2008), the authors used multiple regression analysis to identify critical success factors of agile software development projects. Kandengwa and Khoza (2021) used a questionnaire typical of a quantitative research approach focusing on management support and agile experience relating to project success. Ciric-Lalic et al. (2022) conducted a quantitative study primarily focused on understanding project management's role in software development project success.

Mixed-Method

Misra et al. (2009) used a mixed-method research methodology. They used quantitative (statistical) analysis to determine critical success factors that affect project success. In addition, they asked open-ended questions in a questionnaire requiring a qualitative approach. Gren et al. (2017) also used a mixed-method approach. They used quantitative methods to correlate critical success factors with project success. In addition, they conducted interviews using open-ended questions for qualitative data. Like Gren et al. (2017), Tam et al. (2020) utilized open-ended questions to obtain qualitative data, then used thematic analysis to group results. In addition, quantitative analysis was used to identify critical success factors contributing to software development project success. Butler et al. (2020) used quantitative analysis to determine the correlations between

project complexity and dynamism. They used qualitative data to understand themes from open-ended questions in interviews conducted with agile project managers.

Transition and Summary

Section 1 introduces the problem of understanding technical and social factors affecting the success of the agile software development project. In the literature review, I have discussed estimation methodologies, the definitions of agile software development project success, team experience, and team size. In addition, I have discussed the concept of critical success factors within the context of agile software development projects. This study evaluates the correlation between software development team experience, team size, accuracy of effort estimation, and agile software development project success. I have chosen a quantitative approach for this study to address hypotheses and answer research questions. The twelve critical success factors Chow and Cao (2008) identified serve as this study's theoretical framework. The literature discusses the issues of development team experience, team size, and complications of effort estimation.

In Section 2, I discuss the role of the researcher in the data collection process. I also define who the target participants are for the study. In addition, I discuss the population, sample size, and the research method and design used in the study by justifying using the quantitative approach and how it aligns with the information technology problem identified in the problem statement. In section 2, I also discuss research ethics, data collection procedures, data organization techniques, and analysis techniques. Lastly, I discuss the reliability and validity of the study.

In Section 3, I provide an overview of the study and present the findings. The presentation will include graphs and informative statements regarding quantitative correlation analysis of software development team experience, team size, level of effort estimation accuracy, and project success. Additionally, I will discuss applications of the findings to professional practice and the implications for social change. I will also discuss recommendations for further study and recommendations for action. Lastly, I will present a reflection on the research and draw any conclusions from it.

Section 2: The Project

Section 2 includes a presentation of the reasons for selecting a quantitative correlational study approach for this research. This section includes an explanation of the purpose of the research, my role in the process, and the sample population. The criteria for selecting the sample population and the ethical considerations for this study are explained and justified. Finally, the data collection technique, the organization, the analysis, validity, and reliability of data are described.

Purpose Statement

The purpose of this quantitative correlational study was to examine the relationship between software development team size, team experience, level of effort estimation accuracy, and project success rate for agile software development projects. The independent variables were team size, team experience, and level of effort estimation accuracy, and the dependent variable was the project success rate. The targeted population was agile software development project managers located in Texas. The implication for positive social change was to gain an understanding of technical factors that affect development and production costs and how they correlate.

In this study, my goal was to develop insight into how project managers can mitigate production costs and provide a quality product that meets or exceeds timeline and budgetary constraints. Project management success criteria are often focused on the delivery and implementation phases to determine if project elements were performed correctly. Correctly performing elements may result in an on-time completion and within cost; however, it does not necessarily improve team efficiency and effectiveness.

Identifying team components that improve efficiency and effectiveness may reduce development costs, allowing enhancements such as consumer data protection.

Role of the Researcher

The role of the researcher varies due to circumstances, such as the traditional methods used to conduct research and the research method used. As the researcher, I identified the research problem, formulated objectives, and hypothesized potential outcomes. In addition, I recruited participants for the study, collected data, examined, analyzed, and presented the findings. The role of the researcher in a quantitative study is to play a neutral role in the research process and collect data (Yilmaz, 2013). The quantitative method requires the researcher to use a standardized instrument or predetermined response categories into which the participant's experiences and perspectives are expected to align (Yilmaz, 2013). According to Starcher et al. (2018), the researcher is responsible for gathering data and developing a phenomenological understanding of the study. Köhler et al. (2017) assert that the researcher plays a crucial role in quantitative research as it formulates hypotheses and identifies independent and dependent variables. It is essential when researching to recognize the potential for bias and mitigate any prejudice where possible (Yilmaz, 2013). I used a published survey that reflected issues identified in the research problem and elicits participant responses without affecting the results.

I am an experienced software developer with experience in software project management. I have managed teams of two to 10 members with varying experience levels. The experience levels range from recent university graduates with little to no

practical, real-world experience to senior-level developers with five years or more practical experience. I am familiar with the agile software development methodology and have used it in several projects I have worked on. As an experienced developer familiar with the agile methodology, I have provided effort estimations for small to large-size development projects. My professional experience aided me during this research study.

The purpose of the survey questions was to gain knowledge by obtaining the attitudes and opinions of participants so testing of associations between variables can commence. Surveys are useful to answer three types of questions: descriptive questions, questions about relationships between variables, and questions about predictive relationships between variables (Couper & Miller, 2008). A successful survey design requires several elements, such as identifying its purpose and indicating why a survey is preferred. Additionally, specify the rationale for the procedure, using arguments based on its strengths and weaknesses, costs, data availability, and convenience (Creswell & Creswell, 2017). I incorporated these elements into my survey to ensure the participants were informed of the purpose and rationale for the procedure.

I followed the three fundamental principles outlined in *The Belmont Report*. The principles are ensuring justice, respecting the person, and maximizing beneficence (National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research, 1979). The participants were fully informed about the intent of this research and ensured that participant information would be kept confidential. Individual participants received informed consent forms to establish a working relationship. Full disclosure of the purpose of the research must be presented in an easily comprehended

format so participants can make competent decisions regarding involvement (Geier et al., 2021; Kaewkungwal & Adams, 2019).

Participants

In this section, I define the eligibility requirements for participant selection and how I obtained access to the participants. The targeted population was agile software development project managers in Texas who have managed development teams with three or more members. Participants were current or former IT agile software development project managers who have experienced both successful and unsuccessful project outcomes. The participants were involved in agile team composition regarding the number of team members and their development experience. The participants also worked closely with stakeholders to define requirements and obtain level of effort estimates from the development team. Additionally, the participants were involved in project management, budgeting and control, conflict resolution, knowledge management, and reporting. The specific roles of the participants were senior and junior project managers, senior software developers, and technical leads.

Participants were identified and personally selected on LinkedIn and member listings from the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I sent an inquiry to the potential participant asking for their participation in taking an online survey regarding their personal experience with agile software development project outcomes. In the inquiry, I explained the purpose of the research, eligibility criteria, and expected contributions to practitioner literature and business practice (see Appendix A).

Research Method and Design

The research method includes the approach and underlying philosophy used in the planning and conducting of research. Additionally, the research method describes the design strategy and methodology when and how data is collected during the research process. According to Abutabenjeh and Jaradat (2018), the research problem, question, and purpose must align with the chosen research methodology. The research method and design depend on whether the researcher seeks to explore and interpret a phenomenon or a causal explanation of a phenomenon (Basias & Pollalis, 2018). In this study, I used a quantitative research method with a correlational design to answer the research question.

Method

I used a quantitative research method in this study. Quantitative and qualitative are the two primary methods of social science research (Basias & Pollalis, 2018; Farghaly, 2018; Rutberg & Bouikidis, 2018). The quantitative methodology aligns with the positivist worldview and is informed by objectivist epistemology (Basias & Pollalis, 2018; Farghaly, 2018). To infer and predict behavior in the broader population, the researcher uses statistical techniques to measure causal relationships between variables (Basias & Pollalis, 2018; Taguchi, 2018). In a quantitative study, the researcher uses pre-selected variables representing measurable constructs (Taguchi, 2018). According to Farghaly (2018), the quantitative approach is suitable for testing statistical hypotheses, summarizing statistical data, and supporting the generalization of results. I used a quantitative method for this study. The quantitative method was appropriate because I

measured the effect of software development team size, team experience, and the level of effort estimation accuracy on the success rate of agile software development projects.

Researchers who explore a phenomenon through a flexible, descriptive, flexible, and context-driven framework align their study with a constructivist epistemology by employing a qualitative method (Basias & Pollalis, 2018; Taguchi, 2018). In a qualitative approach, the researcher is committed to understanding the reasoning behind a phenomenon and searches for evidence patterns in data using many elements (Taguchi, 2018). Since I did not conduct exploratory research, the qualitative method was inappropriate for this study.

A third approach, a mixed method, combines qualitative and quantitative methods concurrently or sequentially in a single study, potentially combining the best of both methods (Rutberg & Bouikidis, 2018; Taguchi, 2018). The mixed method approach was also inappropriate for this study since I only used measurable constructs and did not perform an exploratory component.

Research Design

I used a correlation design in this quantitative study. The quantitative methodology has four primary research designs: experimental, descriptive, correlational, and quasi-experimental (Bloomfield & Fisher, 2019; Duckett, 2021). In a correlational design, the researcher measures relationships between a dependent variable and one or more independent variables (Kaeedi et al., 2023). In this study, I employed a correlational design to measure the distribution of agile software development projects' outcomes as a function of social and technical variables. Researchers who employ an experimental or

quasi-experimental design seek to determine cause and effect rather than a measurable relationship afforded by a correlational design (Chen et al., 2023). Experimental and quasi-experimental designs require control groups and manipulation of the variables (Duckett, 2021; Miller et al., 2020). In addition, randomized assignment of participants to groups would be required in an actual experiment (Duckett, 2021; Miller et al., 2020). In this study, I did not seek to determine cause and effect and did not require control groups or the ability to manipulate variables; I did not choose an experiment or quasi-experimental design. In a descriptive design, the researcher seeks to quantify the characteristics of a significantly sized sample (Bloomfield & Fisher, 2019; Duckett, 2021). In addition, a descriptive design is intended to understand the current state of a situation, population, or phenomenon (Chen et al., 2023). I measured the relationship between variables; therefore, a descriptive design was inappropriate for this study.

Population and Sampling

The targeted population was software development project managers in Texas who use or have used the agile software development methodology with a team size of between three and 20 members. The sample of primary participants for primary data collection was software development project managers directly involved in the decision-making, management, and building of agile software development teams throughout the software development lifecycle.

A sample represents the population and unit of analysis in a quantitative research study (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). This study's sample frame was the list of software development project managers who use or have used the agile software

development methodology with a team of three or more members. There are two types of sampling in academic research: random and non-random (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). Random sampling is also called probability, while non-random sampling is called non-probability (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). Probability sampling is the process where each participant is chosen randomly from the population; therefore, each participant has an equal chance of being selected (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). In non-probability sampling, the chance of selection is unknown for each participant, and the researcher would employ other strategies to perform unit selection (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). For this study, I chose the judgmental (or purposive) non-random sampling strategy because the population was specific to software project managers in Texas with a specific skill set. When a researcher needs to select units from a population that satisfies a particular purpose, purposive or judgment sampling is used (Mweshi & Sakyi, 2020; Sarstedt et al., 2018). In this study, the potential participants were software development project managers with specific knowledge of the agile software development methodology.

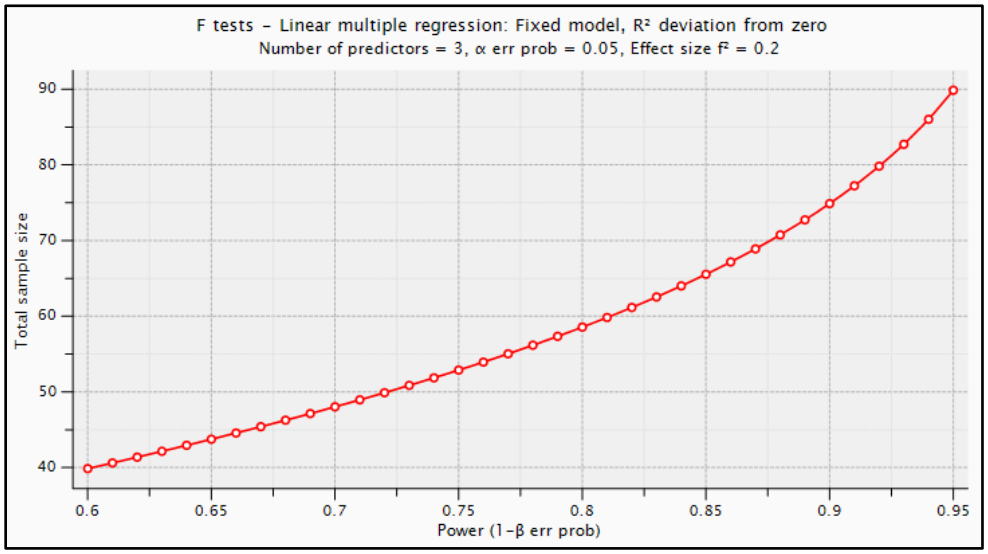
According to Norouzian (2020), in quantitative research, the sample size is critical to the validity of statistical analysis. I used a power analysis to determine the appropriate sample size. There are three critical parameters used in power analysis. They are (a) the alpha level (the probability of falsely rejecting the null hypothesis denoted as $\boldsymbol{\alpha}$), (b) the power (the probability of correctly rejecting the null hypothesis denoted as (1-$\boldsymbol{\beta}$), and (c) the effect size (the attributable amount of variance to the predictor variables denoted as $\boldsymbol{f^2}$) (Hickey et al., 2018). In quantitative research, typical values for $\boldsymbol{a}$ range

between 0.05 and 0.01; values for ***b*** range between 0.2 and 0.1; and the effect size is typically applied from a priori research study (Hickey et al., 2018).

I used G*Power 3.1 software to perform the power analysis for this study. The parameters used were $\boldsymbol{\alpha}$ of 0.05, $\boldsymbol{\beta}$ of 0.2, and a medium effect size $\boldsymbol{f^2}$ of 0.2 taken from a priori research study with similar predictor variables by Misra et al. (2009). The sample size for a linear regression statistical procedure with three predictor variables was 56. Using G*Power, I extended the analysis by increasing the power to 0.95, increasing the sample size to 90. The analysis results provided a targeted sample size of 56 and 90 for this study (Figure 1).

Figure 1

Power as a Function of Total Sample Size



Ethical Research

I followed Walden University's research protocol, which ensures that studies involving human subjects undergo an independent ethical review. Strict ethical protocols

must be followed to protect participants from potential informational, physiological, and physical harm (Geier et al., 2021). Following the research protocol ensures that this research adheres to the three fundamental principles outlined in the *Belmont Report*. In 1979, the National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research defined three basic principles of ethical research: respecting the person, ensuring justice, and maximizing beneficence.

Respecting the person is the first principle of the *Belmont Report* and is closely associated with the informed consent process. Participants were fully informed that their participation would be kept strictly confidential and informed of the aim of the research. The names of the participants were not collected. The IRB reviewed the data use agreement and informed consent form used in this study. Survey participation was voluntary, and participants could withdraw by not completing the online survey.

The second principle of the *Belmont Report*, ensuring justice, ensures that participants will be selected fairly and without targeting vulnerable groups (National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research, 1979). I obtained certification and approval from Walden University's IRB before collecting any primary data (approval # 11-14-23-1135428). To protect the confidentiality of the participants, the data collected was stored on a secured drive, and will be saved for five years, then permanently deleted.

Beneficence is the third principle of the *Belmont Report* regarding incentivizing participants to participate in a study. I did not offer any incentives for participation in my research. The third principle ensures that the minimal risk of harm to individuals

participating in the study would outweigh the benefits of contributions to business and social change (National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research, 1979).

I collected primary data from software development project managers who use the agile methodology within Texas via a survey method. The surveys were administered anonymously, and I did not link participant's identities with individual responses. The persistence of anonymity aligns with the principle of the Belmont Report: *respect the person* (National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research, 1979).

Data Collection

Instruments

The data collection instruments for this study were the demographics dimension, the organizational dimension, the people dimension, the process dimension, the technical dimension, the project dimension, and the perception of success of the agile project. These dimensions are adapted from prior research by Chow and Cao (2008) in their study of critical success factors of agile software development projects. The entire survey instrument focuses on concepts of agile software development and elements that affect the project outcome. These dimensions have been used in extant research on critical success factors of agile software development projects (Dikert et al., 2016; Stankovic et al., 2013; Tsoy & Staples, 2021).

The demographics dimension, influenced by Dikert et al. (2016) and Stankovic et al. (2013), provides values for general information regarding the participant's agile

software development project to base their answers. This section was purely informational and not used in any quantitative statistical analysis. This dimension has eight questions: four regarding basic information regarding the agile project and four questions requesting basic information regarding the participants and their respective organizations.

The organizational dimension, influenced by research by Chow and Cao (2008), is measured by three underlying factors: management commitment, organizational environment, and team environment. Each underlying factor in the survey questionnaire has two to four items. The people dimension, also influenced by research by Chow and Cao (2008), has two underlying factors: team capability and customer involvement. Each underlying factor has one to two items in the survey questionnaire. The people dimension, influenced by research by Chow and Cao (2008) and Stankovic et al. (2013), provided values for the software development team experience independent variable. The process dimension also has two underlying factors: the project management process and the project definition process. Each of these factors has two to three items in the survey questionnaire. The project dimension, influenced by research by Chow and Cao (2008) and Tsoy and Staples (2021), provides the software development team size independent variable values. The technical dimension has two underlying factors: agile software techniques and delivery strategies. Each of these factors has three to four items in the survey questionnaire. The technical dimension provides, also influenced by research by Chow and Cao (2008), values for the software development team's level of effort estimation accuracy independent variable.

The perception of the success of an agile project, influenced by research by Chow and Cao (2008), is measured with four underlying factors: the project was successful in terms of software quality, the project was successful in terms of scope and requirements, the project was successful in terms of timeliness, and the project was successful in terms of costs and efforts being under budget or within estimates. The perception of the project success dimension provides the values for the software development project outcome-dependent variable.

The survey instrument was appropriate for this study because it had been used in extant literature. Additionally, it allowed for collecting information from a sample of individuals through their responses to questions. Historically, survey research has included extensive population-based data collection to obtain information reflecting demographic and personal characteristics (Ovan & Saputra, 2020). The survey instrument provided an appropriate method for collecting data regarding this study's problem statement. An online survey is cost-effective and convenient for participants while eliminating geographic restrictions (Jordan, 2018). A formal request to reuse the survey instrument was sent to the authors (Appendix C).

The survey was administered online via a secured internet service using SurveyMonkey. The survey was expected to take less than 20 minutes to complete. Participants responded to these items/questions using a 7-point Likert scale (see Table 1). The independent variables of software development team size, team experience, and the level of effort estimation accuracy are numeric and have an interval scale, with values ranging from 1 to 7. Agile software development project outcome is numeric and has an

interval scale, with values ranging from 1 to 7. The complete structure of the survey instrument is shown in Appendix B.

Table 1

Survey Response Likert Values

Likert Value	Response Description
1	Strongly disagree
2	Disagree
3	Somewhat disagree
4	Neither disagree nor agree
5	Somewhat agree
6	Agree
7	Strongly Agree

The original survey instrument designed by Chow and Cao (2008) did not include questions regarding the accuracy of the level of effort estimation. Thus, I added two additional questions to the technical dimension section of the survey. In the demographics section, I removed the questions about the project's location and the participant’s role because I limited this study to software development project managers in Texas. I also removed the questions about the company name and contact information, which is irrelevant to my study. Lastly, I changed the definition of small team size to 3 to 20 members.

The survey questionnaire Chow and Cao (2008) developed has been used in other studies regarding the critical success factors of agile software development projects. Chiyangwa and Mnkandla (2018) used the instrument to collect data on critical success factors of agile software development projects in South Africa. Dikert et al. (2016) used the instrument to obtain data from European large-scale software development companies. Stankovic et al. (2013) applied Chow and Cao's (2008) instrument while focusing on critical success factors in agile software development projects in former Yugoslavia.

In academic research, the degree to which a data collection technique or study successfully captures its intended measure can be described as its validity (Bahariniya et al., 2021; Jordan, 2018; Ovan & Saputra, 2020). Reliability can be described as the degree to which consistent results would be obtained if the study was replicated by other researchers using the same data collection method (Bahariniya et al., 2021; Jordan, 2018; Ovan & Saputra, 2020). The validity and reliability of the instruments were assessed by three tests: content, discriminant, and convergent validity (Evans & Mathur, 2018). Content validity ensures the comprehensiveness of the elements used to create each measure.

The elements for agile software development project success have been verified by extant literature and interviews with software development project managers (Chow & Cao, 2008). Convergent validity examines the average variance of the measurement constructs for reliability. The measures yielded an average variance between 0.6 and 0.10 for samples in the prior study by Chow and Cao (2008). The discriminant validity of the

measures derives from the square root of the average variance for each construct. In addition to the abovementioned measures, the survey instrument has several general questions, like Chow and Cao's (2008) approach. These questions included information regarding agile projects, the participant organization, and personal experiences.

Chow and Cao (2008), as part of the pilot process and to test content validity and readability, elicited feedback from five members of the Agile Alliance on how to improve the survey. This ensured the questions were relevant, comprehendible, and detailed enough to elicit accurate participant responses. Tsoy and Staples (2021) and Tam et al. (2020) used a modified version of Chow and Cao's (2008) instrument for their study. Their findings show that the instrument produced similar results, increasing its test-retest reliability. I used Cronbach's alpha to test the reliability of the survey. Cronbach α is a coefficient alpha, a function of the number of items and their significance of inter-correlation (Chow & Cao, 2008; Cronbach, 1951).

Additionally, I totaled all the scores for each participant and performed a Pearson bivariate correlation analysis. The correlation coefficient was then compared to the critical value of the degrees of freedom. The survey question was considered valid if the correlation was significant and the correlation coefficient was larger than the critical value for the degrees of freedom (Cramer et al., 2006).

The raw data from the survey was downloaded from SurveyMonkey after the survey was closed. The data itself was made available in appendices after the data collection stage. The data may also be provided in digital form via request from the researcher.

Data Collection Technique

This quantitative correlational study evaluated data to determine if a significant relationship exists between software development team size, team experience, level of effort estimation accuracy, and agile software development project outcomes. To accomplish this, I used an online survey instrument to collect primary data for software development team size, team experience, level of effort estimation accuracy, and agile software development project outcome. I posted the link to my survey on ACM and IEEE member boards that target agile software project management. In addition, I adopted a supplemental data collection technique that involved using the LinkedIn professional network to publish my consent form and survey participation requests. I sent the consent form and survey participation requests (Appendix A) to targeted groups of agile software development project managers in Texas.

The survey was administered online using SurveyMonkey. The online survey method is cost-effective, easy to administer, and convenient for participants. In addition, online questionnaires allow for the quick collection of large volumes of data from geographically dispersed participants (Ball, 2019). Although online surveys are cost-efficient and convenient, they are impersonal and have a high chance of response bias and fraud (Ball, 2019). Additionally, due to the online factor and lack of personal communication, it is difficult to interpret the sentiments behind the participant's answers.

SurveyMonkey provides export capabilities in a variety of formats. I utilized the XLS file format to export results. To conform to the ethical principles outlined earlier, I ensured the data was encrypted and confirmed that the third-party provider,

SurveyMonkey, removed the data from their server at the study's conclusion. The data file(s) have been password-protected and stored on a secured drive.

Data Analysis Technique

The research question for this study: What is the relationship between software development team size, experience, level of effort estimation accuracy, and agile software development project success rates? The null hypothesis, H_0: There is no significant relationship between software development team size, experience, level of effort estimation accuracy, and agile software development project success rates. The alternate hypothesis: H_1: There is a significant relationship between software development team size, experience, level of effort estimation accuracy, and agile software development project success rates. An analysis of how the three predictor or independent variables affected the dependent variable, the outcome of agile software development projects, was required to answer the research question and test hypotheses.

Data Integrity

This study used an online survey administered by SurveyMonkey. There were some questions regarding primary demographic data that allow freeform text. These questions were for context relating to the participant and were not used in statistical analyses. The need for data cleansing is minimal, if nonexistent. The questions used for statistical analyses were defined so that they are required and are not freeform. These questions had possible answers tied to a Likert scale of 1-7. The data has been stored on an online server administered by SurveyMonkey; therefore, no external access to the data is permitted. After the study, I downloaded the data to a Microsoft Excel spreadsheet and

reviewed it for completeness. If, for any reason, invalid answers were given, the records were notated and removed from the dataset before any analysis.

Statistical Analysis

In this study, I used multiple linear regression for statistical analysis. Several statistical tests can be used to analyze the relationship or correlation between independent and dependent variables. These include simple bivariate linear regression, the t-test or comparison of means, multiple linear regression, ANOVA or analysis of variance, and MANOVA or multiple analysis of variance (Abu-Bader, 2021). Two or more independent variables can be analyzed simultaneously with multiple linear regression analysis, resulting in an equation to predict the score of the dependent variable (Abu-Bader, 2021). Simple bivariate linear regression, ANOVA, MANOVA, and t-tests are helpful when only one predictor variable is used; therefore, they are inappropriate for this study. The multiple linear regression technique enhances the simple bivariate linear regression technique and can be used when multiple predictor-independent variables are present (Mari et al., 2019). In this study, I used multiple linear regression analysis because the research question involves three independent variables and how they predict the outcome of the agile software development project, the dependent variable.

All three independent variables, software development team size, experience, and level of effort estimation accuracy, are numeric. Additionally, all three independent variables have an interval scale with values ranging from 1 to 7. The dependent variable, agile software development project outcome, is also numeric and has an interval scale with values ranging from 1 to 7. I used the multiple linear regression function in the

SPSS statistical software to compute the regression analysis. SPSS is a statistical software package used in research that is powerful and user-friendly and can perform all the statistical tests mentioned above (Abu-Bader, 2021). I also used SPSS to perform the correlations to determine the significance of intercorrelation between variables.

Data assumptions include linearity, homoscedasticity, independence of residual outliers, and normal data distribution. Homoscedasticity is primarily related to dependence relationships between variables and refers to the assumption that variables do not exhibit equal levels of variance across the predictor variables (Johns et al., 2022). When the homoscedasticity assumption is violated, there is an increased probability of Type I errors or a decrease in statistical power (Rosopa et al., 2013). The normalized data distribution assumption in multivariate analysis refers to the shape of the data distribution for an individual metric variable and its correspondence to the normal distribution (Johns et al., 2022). Linearity is an implicit assumption of all multivariate techniques based on correlation measures of association (Rosopa et al., 2013). Correlations represent only linear associations between variables; therefore, the correlational value will not represent only nonlinear effects (Johns et al., 2022). Independence concerns longitudinal data, where observations are collected from the same entity over time. There are two types of data: cross-sectional and longitudinal. Cross-sectional data is collected from entities only once, whereas longitudinal data is collected from the same source more than once (Johns et al., 2022). The independence assumption is assumed to be met with cross-sectional datasets (Rosopa et al., 2013).

Probability and scatter plots of dependent variable residuals can be used to test for independence of outliers, homoscedasticity, linearity, outliers, and normality (Chantarangsi et al., 2018). I used SPSS to generate normal probability plots and scatterplots and analyze the results for outliers, linearity, normality, homoscedasticity, and independence of residual outliers. The statistical validity of research findings increases as the sample size increases (Norouzian, 2020). I used G*Power analysis to determine the sample size and ensure statistical validity. I aimed for a sample size larger than the minimum of 56 required by power analysis to improve the overall statistical validity of this study.

In the data analysis, I examined the effect of each independent variable separately to determine its impact on the dependent variable, agile software development project outcome. The evaluation will require evaluating the coefficient for each independent variable from the SPSS linear multiple regression output. I also used the SPSS output to construct a regression equation that includes all three independent variables. The presumptive predictive equation is:

Agile Software Development Project Outcome = constant value + coefficient(team size) + coefficient(team experience) + coefficient(level of effort estimation accuracy)

I also performed a data transformation on the four project dimension question values that identified the project's success. The four questions were assigned a numeric value ranging from 1 to 7 (Likert scale), measuring Very Unsuccessful to Very Successful. The transformation added the Likert values of the four questions related to

project success together, forming a composite value. I then used the composite value as the dependent variable for regression analysis.

Reliability and Validity

In quantitative research, rigor is achieved by reliability and validity measures to ensure that the measures properly represent the theoretical constructs. In research, validity and reliability measures reinforce trustworthiness in the study and position the study to be used by software development project managers when making decisions (Hall & Van Ryzin, 2019).

Reliability

Research reliability may be defined as the extent to which the data collection method will yield similar findings and similar conclusions if conducted by another researcher (Surucu & Maslakci, 2020). In quantitative research, obtaining similar results is straightforward because of the numerical nature of the data. According to Dobakhti (2020), any researcher can analyze numerical data, and the results would not have a significant variance from one context to another. However, in qualitative research, the same data from interview or observation may be inconsistent from one context to another (Dobakhti, 2020). Engel and Schutt (2017) suggest that it is better to agree that the results are dependable and consistent based on the data collection processes.

External reliability is concerned with the replication of the study. If another researcher were to reproduce a study and obtain similar results, it would be considered externally reliable (Surucu & Maslakci, 2020). Dobakhti (2020) suggests that the researcher follows five procedures to augment the external reliability of the research. The

five procedures include the choice of informants, the status of the researcher, the social situations and conditions, the analytic constructs and premises, and the methods of data analysis and collection (Dobakhti, 2020; Engel & Schutt, 2017; LeCompte & Goetz, 1982; Nunan, 1992). The procedures mentioned above require the researcher to disclose their social position regarding the participants and describe the participants as fully as possible, leaving no ambiguity. The procedures also encourage the researcher to ensure that the participant's social situations and conditions are consistent. Lastly, the different data collection procedures and phases must be explicitly explained. I used a modified version of the survey instrument created by Chow and Cao (2008). It has achieved external reliability through use in studies including Stankovic et al. (2013), Dikert et al. (2016), and Tsoy and Staples (2021).

Internal reliability is concerned with the consistency of collecting, analyzing, and interpreting data. Internal reliability is obtained when an independent researcher reanalyzes the study and produces findings similar to those of the original researcher (Dobakhti, 2020). In general, there are four strategies a researcher can use to guard against threats to internal reliability. The strategies are low inference descriptors, multiple researchers, peer examination, and mechanically recorded data (Dobakhti, 2020; LeCompte & Goetz, 1982; Nunan, 1992).

Low inference descriptors are easily observable and readily qualified (counted or measured). The instrument used in this study uses questions on a Likert scale that can be counted and measured. Multiple researchers or participant researchers are not feasible in this study. However, there are instances where experienced participants may be used to

aid a researcher in data collection and validation (Surucu & Maslakci, 2020). Peer examination can be implemented by applying and utilizing findings from other researchers, resulting in enhanced internal reliability (Dobakhti, 2020). Lastly, mechanically recorded data enhances internal reliability by providing data that can be analyzed and reanalyzed. In qualitative research, mechanically recorded data could be a recorded interview. In quantitative research, mechanically recorded data could be electronic results of an online questionnaire or survey. In this quantitative study, I utilized an online survey through SurveyMonkey that will mechanically record the data.

Validity

Validity can broadly be defined as the extent to which the data collection method will result in similar findings if research is conducted by other researchers (Surucu & Maslakci, 2020). According to Dobakhti (2020), validity is a matter of utility, dependability, and trustworthiness that the researcher instills in it. Additionally, validity concerns the research being accurate and believable and evaluating what it purports to evaluate (Dobakhti, 2020).

Internal validity is the extent to which the instruments accurately demonstrate a causal relationship in a quasi-experimental or experimental research study (Flannelly et al., 2018). This study is a correlational design and not an experimental or quasi-experimental design; internal validity threats such as mortality, statistical regression, selection, selection by maturation, instrumentation, history, and testing are not applicable. In correlational analysis, such as this study, the researcher is concerned with threats to statistical conclusion validity.

The instrument's reliability is measured by how well the instrument avoids a Type I error. A Type I error is the probability of incorrectly rejecting the null hypothesis when the null hypothesis is true (Gohary, 2019). To mitigate this type of error, a robust level of significance (denoted by alpha level or α-value) is applied to the study. The significance value of 0.05 used in this study indicates that there is only a 5% chance of committing a Type I error. In quantitative research, the most commonly used level of significance is 5% (Bajpai & Chaturvedi, 2021; Indrayan, 2020).

The extent to which causal relationships can be generalized to other contexts is external validity (Fabrigar et al., 2020). Research that uses probability or random sampling would be externally valid and have the greatest freedom from bias (McEwan, 2020). I used a non-probabilistic sampling technique to target a specific knowledge base in this study. The population for this study was restricted to Texas. Indications of a moderating variable specific to another context could be present when the findings of a study cannot be replicated (Fabrigar et al., 2020). A generalization of the findings of this study should be limited to other agile software development project managers within Texas.

Transition and Summary

In Section 2, I outlined the research methodology and design for conducting this quantitative correlational study. I also described my role as the researcher and the steps that will be taken to reduce bias and ensure objectivity. Additionally, I outlined the steps I took to ensure ethical research practices are followed. The rationale for choosing the study participants, population, and sample size was also outlined. In this section, I

explained in detail the *priori* survey questionnaire instruments for software development team experience, team size, level of effort estimation accuracy, and agile software development projects' outcomes.

Additionally, I outlined the technique I used for data collection: an online web-administered survey questionnaire. I also explained the steps I took to ensure the validity and reliability of the study findings. Additionally, I demonstrated the benefits of using SPSS statistical software as an analytical tool. Lastly, I calculated an appropriate sample size (between 56 and 90) using G*Power to ensure the study validity and significance of the results.

In Section 3, I present the data analysis results and findings. I outlined the results of the study validity and reliability testing. Additionally, I explain the implications for agile software development project managers, contributions to academic and professional literature, and contributions to social change. I explain this study's limitations and future research opportunities. Lastly, I present my conclusions and reflections.

Section 3: Application to Professional Practice and Implications for Change

The purpose of this study was to examine the relationship between software development team size, team experience, level of effort estimation accuracy, and project success rate for agile software development projects. This section includes (a) an overview of the study, (b) presentation of the findings, discussion of the study's (c) application to professional practice and (d) implications for social change, (e) recommendations for actions, (f) suggestions for further research, and (g) personal reflections and conclusion.

Overview of Study

The purpose of this quantitative correlational study was to examine the relationship between software development team size, team experience, level of effort estimation accuracy, and project success rate for agile software development projects, as moderated by software development team size (SDTS), software development team experience (SDTE), level of effort estimation accuracy (LEEA), and project success rate (PSR). SPSS was used for initial data validation and exploratory factor analysis.

Presentation of the Findings

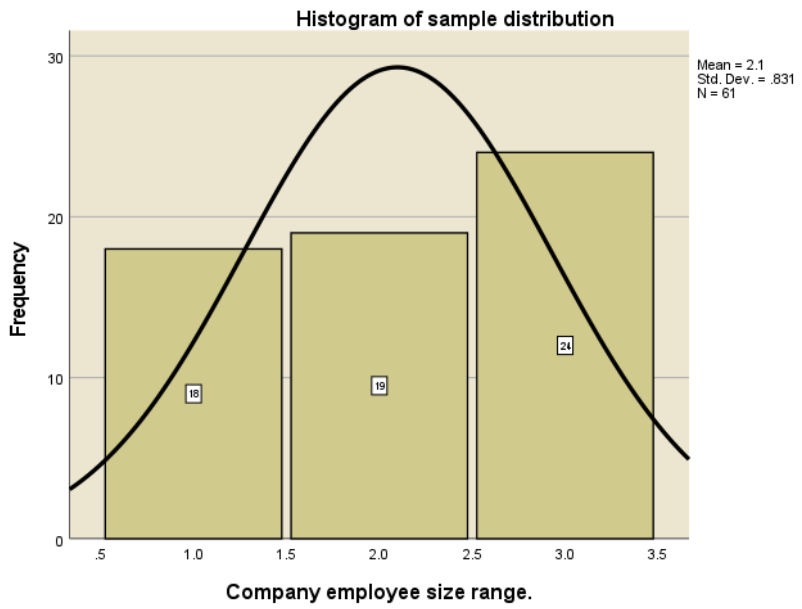
The IRB approved the data collection process on the 9th of November, 2023 (approval # 11-14-23-1135428). The survey was deployed on the 18th of November, 2023, and closed on the 12th of March, 2024. The survey collected a total of 82 respondents. The respondents who qualified for the survey were filtered by Questions 5, 6, and 44. These questions ensured that the participants were IT project managers with agile project experience who worked in an agile software development team between

three and 20 members. Upon filtering on these qualifications, the total number of respondents was reduced to 73, thus resulting in an 89% completion rate.

After reviewing the data in Microsoft Excel, respondents who did not fully answer the survey were excluded. This resulted in a final sample of 61 respondents (*n = 61*). The distribution of the company size of final sample respondents is shown in Figure 2, hereunder.

Figure 2

Historgram of Sample Distribution



Variables

The SDTS variable was collected using a Likert scale of 1-7, where respondents provided a level of agreement regarding team size (see Table 3). The SDTE variable was collected using a Likert scale of 1-7, where respondents provided a level of agreement regarding team experience (see Table 4). The LEEA variable was collected using a Likert

scale of 1-7, where respondents provided a level of agreement regarding the level of effort estimation accuracy (see Table 5).

The PSR variable was not collected directly but is a composite value of the four agile project success (APS) variables (PSR = APS1 + APS2 + APS3 + APS4). The project success rate variables were based on a 7-point Likert scale and collected directly via the survey data collection instrument (see Tables 6, 7, 8, and 9).

Descriptive Statistics

After the initial data cleansing in Microsoft Excel, I used SPSS (version 28) to (a) understand the sample and (b) conduct testing related to normality and reliability. I evaluated the data uploaded to SPSS for missing data relating to the constructs for software development team size, team experience, level of effort estimation accuracy, and project success as per Table 2.

Table 2

Software Development Team and Project Characteristics

No.	Category	Team Size	Team Experience	Level of effort estimation accuracy	Project Success
N	Valid	61	61	61	61
	Missing	0	0	0	0

The software development team size analysis revealed that the majority (52.5%) of the participant's team size was five members or fewer, as per Table 3. This supports prior research conducted by Tsoy and Staples (2021), Dikert et al. (2016), and Kula et al. (2022), where a significant percentage of team sizes were five members or fewer. Notably, 10 (16.4%) of all participants reported team sizes of 11-15 members, and only

two (3.3%) reported a team size of 16-20. The findings suggest that most agile software development team sizes fall within one to 10 members at 83.6%.

Table 3

Software Development Team Size

	Frequency	Percent	Valid Percent	Cumulative Percent
5 members or less	32	52.5	52.5	52.5
6-10 members	19	31.1	31.1	83.6
11-15 members	10	16.4	16.4	96.7
16-20 members	2	3.3	3.3	100.0
Total	61	100.0	100.0	

The software development team experience analysis revealed that 8 (13.1%) project managers disagreed that the team comprised a good mixture of non-experienced and experienced members, as per Table 4. The analysis also showed that five (8.2%) neither disagreed nor agreed that the team included a good mixture of non-experienced and experienced members. Conversely, 48 (78.8%) participants agreed that the development team included a good mixture of non-experienced and experienced members. A prior study by Bogopa and Marnewick (2022) found that of 212 respondents, 78 (36.7%) stated that members had more than 10 years of experience, while 62 (29.7%) had less than five years of experience. Butler et al. (2020) found similar results, finding that of 121 respondents, 76 (62%) reported team members having 10 or more years of experience. Their findings support the analysis by suggesting that having a mixture of non-experienced and experienced team members is important.

Table 4

Software Development Team Experience

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly Disagree	1	1.6	1.6	1.6
Disagree	2	3.3	3.3	4.9
Somewhat Disagree	5	8.2	8.2	13.1
Neither Disagree nor Agree	5	8.2	8.2	21.3
Somewhat Agree	14	23.0	23.0	44.3
Agree	25	41.0	41.0	85.2
Strongly Agree	9	14.8	14.8	100.0
Total	61	100.0	100.0	

Analysis of the software development team's level of effort estimation accuracy revealed that 9 (14.8%) disagreed that team member's estimations were accurate, as per Table 5. Additionally, 4 (6.6%) disagreed nor agreed that member estimations were accurate. Lastly, 48 (78.8%) agreed that the effort estimations provided by team members were accurate. The vast disparity between the number of respondents reporting accurate and inaccurate estimates is significant. The analysis suggests that more than 75% of all estimates were considered accurate by the respondents. These findings show a considerable increase in accuracy compared to prior study results. Prakash and Viswanathan (2017) asserted that 46.3% of respondents reported inaccurate effort estimations. Similarly, Govil and Sharma (2022) found that 38.7% of respondents reported team member estimation accuracy to be inaccurate.

Table 5

Software Development Team Level of Effort estimation accuracy

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly Disagree	2	3.3	3.3	3.3
Disagree	0	-	-	-
Somewhat Disagree	7	11.5	11.5	14.8
Neither Disagree nor Agree	4	6.6	6.6	21.3
Somewhat Agree	14	23.0	23.0	44.3
Agree	30	49.2	49.2	93.4
Strongly Agree	4	6.6	6.6	100.0
Total	61	100.0	100.0	

Agile Software Development Project Success Factors

Descriptive statistics regarding product quality regarding the software development project's success are provided in Table 6. Analysis of the data revealed that 12 (19.7%) participants reported that the quality of the product did not meet the desired standards. Furthermore, 9 (14.8%) participants responded that they somewhat agreed that the desired standards were met. The most significant number of participants, 40 (65.6%), answered that they agreed or strongly agreed that their project was successful regarding product quality. The many positive outcomes regarding product quality are significant, considering that only 19.7% reported unsuccessful outcomes. Conoscenti et al. (2019) found that 56% of respondents reported that product quality was achieved and the software development project was successful.

Table 6

Software Development Project Success – Product Quality

	Frequency	Percent	Valid Percent	Cumulative Percent
Very Unsuccessful	0	-	-	-
Unsuccessful	7	11.5	11.5	11.5
Somewhat Unsuccessful	5	8.2	8.2	19.7
Neither Successful nor Unsuccessful	0	-	-	-
Somewhat Successful	9	14.8	14.8	34.4
Successful	28	45.9	45.9	80.3
Very Successful	12	19.7	19.7	100.0
Total	61	100.0	100.0	

The analysis of project success concerning the product meeting the scope of requirements revealed that 12 (19.7%) disagreed on some level that the project was unsuccessful, as per Table 7. Conversely, 49 (80.3%) agreed that the project was determined to be successful in meeting or exceeding the project's requirements and scope. The analysis suggests that most agile software development projects successfully meet or exceed project requirements.

Table 7

Software Development Project Success – Requirements Scope

	Frequency	Percent	Valid Percent	Cumulative Percent
Very Unsuccessful	1	1.6	1.6	1.6
Unsuccessful	9	14.8	14.8	16.4
Somewhat Unsuccessful	2	3.3	3.3	19.7
Neither Successful nor Unsuccessful	0	-	-	-
Somewhat Successful	11	18.0	18.0	37.7
Successful	28	45.9	45.9	83.6
Very Successful	10	16.4	16.4	100.0
Total	61	100.0	100.0	

Descriptive statistics for the software development project success regarding the project timeline are provided in Table 8. The analysis revealed that 15 (24.6%) disagreed that the software development project was unsuccessful in meeting the project timeline constraints. The results also suggest that most (73.8%) projects successfully met timeline constraints. In addition, there is a disparity between prior research and the results. Garousi et al. (2019) found that 45% of their respondents met or exceeded timeline constraints in successful projects. Similarly, Govil and Sharma (2022) found that 42% met or exceeded timeline constraints in successful agile software development projects.

Table 8

Software Development Project Success – Project Timeline

	Frequency	Percent	Valid Percent	Cumulative Percent
Very Unsuccessful	4	6.6	6.6	6.6
Unsuccessful	6	9.8	9.8	16.4
Somewhat Unsuccessful	5	8.2	8.2	24.6
Neither Successful or Unsuccessful	1	1.6	1.6	26.2
Somewhat Successful	12	19.7	19.7	45.9
Successful	26	42.6	42.6	88.5
Very Successful	7	11.5	11.5	100.0
Total	61	100.0	100.0	

Descriptive statistics for the success of the software development project relating to the project budget are provided in Table 9. The analysis revealed that a small number of 12 (19.7%) disagreed that the agile software development project successfully adhered to budget constraints. Conversely, 47 (77.1%) respondents agreed that their software development project successfully met or exceeded budgetary constraints.

Table 9

Software Development Project Success – Project Budget

	Frequency	Percent	Valid Percent	Cumulative Percent
Very Unsuccessful	2	3.3	3.3	3.3
Unsuccessful	5	8.2	8.2	11.5
Somewhat Unsuccessful	5	8.2	8.2	19.7
Neither Successful or Unsuccessful	2	3.3	3.3	23.0
Somewhat Successful	12	19.7	19.7	42.6
Successful	26	42.6	42.6	85.2
Very Successful	9	14.8	14.8	100.0
Total	61	100.0	100.0	

Regression Analysis

This section details the relationships between software development team size, team experience, level of effort estimation accuracy, and agile software project success. The multiple regression analysis considered linearity, normality, multi-collinearity, and homoscedasticity. I performed a multiple linear regression analysis to produce an equation that can be used to predict the PSR from SDTS, SDTE, and LEEA. The independent variables X_1, X_2, and X_3 (predictors) were used to predict the outcome of the dependent variable Y: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + e$. Where β_0, β_1, β_2, and β_3 are the respective independent variable coefficients. The analysis also provided insight into the impact the three independent variables have on agile software development project success regarding product quality, timeline, budget, and requirement scope.

Multiple regression can be used to analyze and assess the effects of a variable. In total, four variables are considered. Three variables are independent (Critical Success

Factors) and one dependent variable which is a composite of four measures of success (Agile Software Project Success). I conducted a statistical test to test the hypothesis that there is no significant relationship between Software Development Team Size, Software Development Team Experience, Level of Effort Estimation Accuracy, and Project Success.

Tests of Assumptions

Assumptions of multicollinearity, outliers, normality, linearity, homoscedasticity, and independence of residuals were evaluated. I used statistical tests to evaluate assumptions of multicollinearity, outliers, normality, linearity, homoscedasticity, and independence of residuals. Tests include an analysis of bivariate correlations, a scatterplot of standardized residuals, and a normal probability plot. A careful evaluation of these tests shows no major violation of these assumptions.

I reviewed Pearson correlation coefficients among the predictor variables to evaluate multicollinearity. Bivariate correlations were not significant. Thus, there are no violations of the assumption of multicollinearity (see Table 10).

Table 10

Correlation Coefficients Among Study Variables

Variable	PSR	SDTE	LEEA	SDTS
PSR	1.000	.084	.180	-.255
SDTE	.084	1.000	-.068	.110
LEEA	.180	-.068	1.000	-.159
SDTS	-.255	.110	-.159	1.000

I used a normal probability plot (P-P) of regression standardized residuals and a scatterplot of standardized residuals to determine if assumptions involving normality, linearity, outliers, independence of residuals, and homoscedasticity have been violated. Assumptions of normality and linearity were not significantly violated (see Figure 3). A review and examination of standardized residuals (see Figure 4) shows no specific pattem regarding data points. Thus, data assumptions were not significantly violated.

Figure 3

Normal P-P of Regression Standardized Residuals

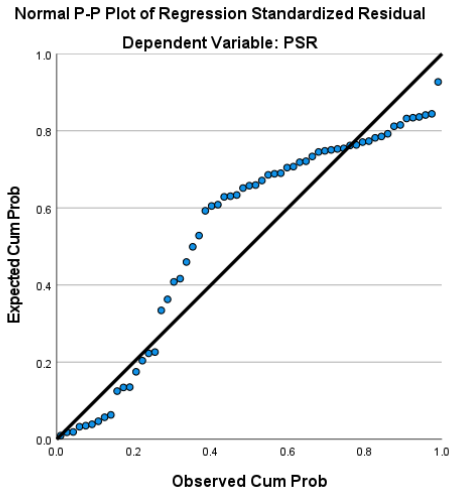
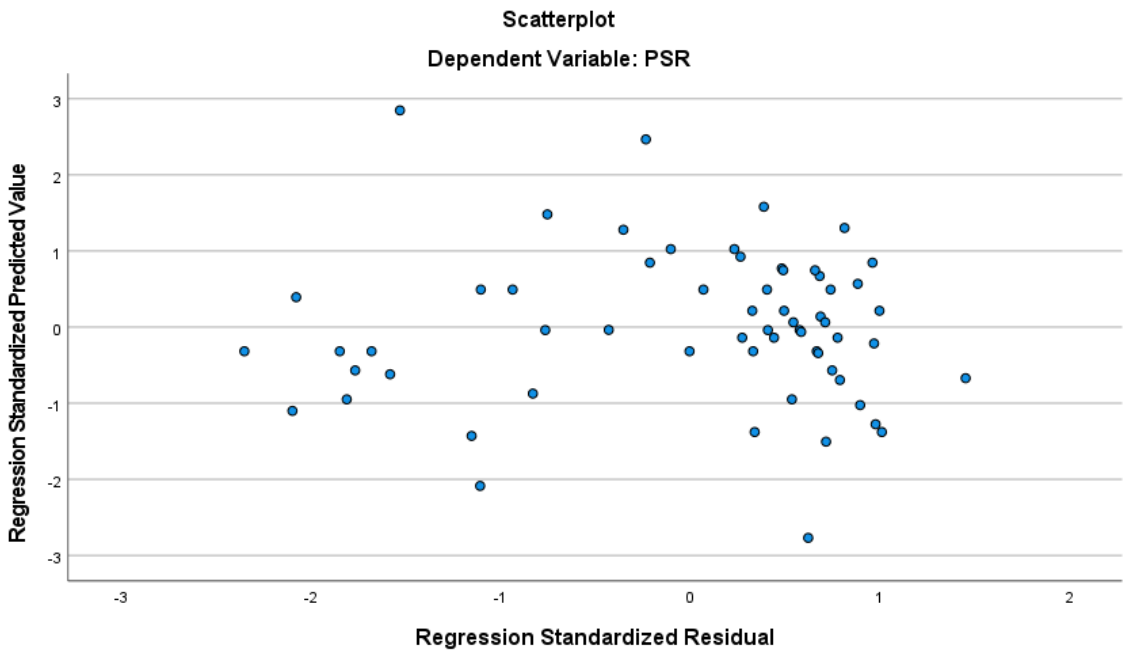


Figure 4

Scatterplot of the Standardized Residuals



Independence

The sample used in the analysis is obtained by utilizing a random sampling technique, and independence of observations can be assumed.

Outliers

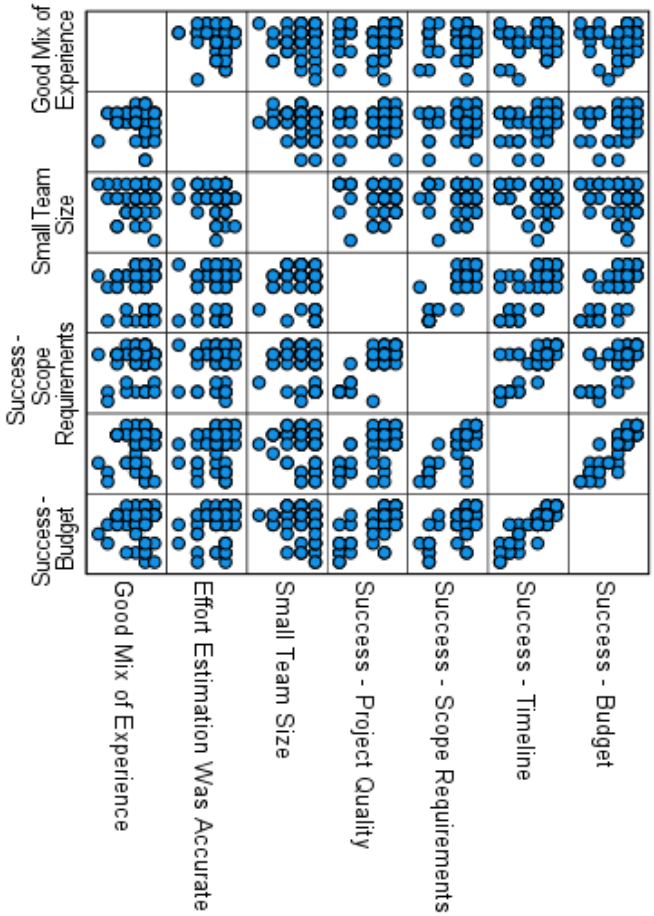
Casewise diagnostics did not return any case with a standardized residual higher than ±3, which indicates an absence of extreme values.

Linearity

A scatter matrix, as seen below in Figure 5, reveals there is not a strong linear relationship between the three independent variables, and the dependent variable. Therefore, the assumption is not adequately met.

Figure 5

Scatter matrix of variables



Multicollinearity

The variance inflation factor for all three variables is within the acceptable range of less than five (SDTE = 1.015, SDTS = 1.036, LEEA = 1.029).

Confidence Intervals

The lack of perfect alignment in the P-P of regression standardized residuals indicates the residual variance of homoscedasticity. I used a bootstrapping technique to mitigate this violation. The technique I used involved 2,000 resampling and a 95% confidence interval. The bootstrap results for the coefficients were used to confirm the

regression model (see Appendix D). Bootstrapped confidence intervals for the mean (M) of variables are shown in Table 11.

Table 11

Bootstrapped Confidence Intervals for Study Variables

Variable	*M*	*SD*	N	Bootstrapped 95% CI (*M*)
PSR	20.62	6.126	61	[10.03, 38.47]
SDTE	5.30	1.383	61	[-0.586, 1.73]
LEEA	5.20	1.339	61	[-0.593, 1.90]
SDTS	6.02	.957	61	[-3.76, -.004]

Inferential Results

A standard multiple linear regression with $\alpha = .05$ (two-tailed) was used to examine the relationship between software development team size, experience, level of effort estimation accuracy, and project success. The independent variables were SDTS, SDTE, and LEEA. The dependent variable was PSR, measuring the outcome of agile software development projects. The null hypothesis was that there is no significant relationship between software development team size, experience, level of effort estimation accuracy, and agile software development project outcome. The alternative hypothesis is that there is a significant relationship between software development team size, experience, level of effort estimation accuracy, and agile software development projects' outcomes. Based on the analysis of data assumptions, no serious violations of the assumptions of outliers, normality, multicollinearity, homoscedasticity, and independence of residuals were present (see *Tests of Assumptions*).

The regression model produced the equation representing agile software development project success: $F(3, 57) = 2.093$, $p = 0.111$, $R^2 = 0.099$. The R^2 (0.099) value indicated that approximately 9% of variations in agile software development project success are accounted for by the linear combination of the predictor variables (SDTE, SDTS, LEEA), see Tables 12 and 13. The coefficient analysis results show that all three predictor variables have a p-value greater than 0.05 (see Table 14). This indicates that all three independent variables (SDTS, SDTE, and LEEA) have no significance in predicting the success of agile software development projects. The final predictive equation was PSR = 23.63 + 0.537(SDTE) + 0.683(LEEA) – 1.563(SDTS).

Table 12

Regression Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.315[a]	.099	.052	5.965

Table 13

Regression ANOVA output

Model	Sum of Squares	Df	Mean Square	F	Sig.
Regression	223.534	3	74.511	2.093	.111[b]
Residual	2028.794	57	35.593	35.593	
Total	2252.328	60			

SDTE

The positive B value (.537) for SDTE indicates that software development team experience was a significant predictor of PSR. This means that for every 1-point increase

in SDTE, there was an approximate increase of .537 in PSR. This also indicates that PSR increases as SDTE increases.

LEEA

The positive B value (.683) for LEEA indicates that the software development team's level of effort estimation accuracy was a significant predictor of PSR. This means that for every 1-point increase in LEEA, there was an approximate increase of .683 in PSR. This also indicates that PSR increases as LEEA increases.

SDTS

The negative B value (-1.563) for SDTS indicates that software development team size was a significant predictor of PSR. This means that for every 1-point increase in SDTS, there was an approximate decrease of 1.536 in PSR. This also indicates that PSR decreases as SDTS increases.

Table 14

Regression Coefficients output

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
	B	Std. Error	Beta			Lower Bound	Upper Bound
(Constant)	23.639	6.739		3.508	<.001	10.144	37.134
SDTE	.537	.561	.121	.956	.343	-.587	1.660
LEEA	.683	.583	.149	1.171	.247	-.485	1.851
SDTS	-1.563	.819	-.244	-1.909	.061	-3.204	0.77

Analysis Summary

Data collection procedures were discussed, and the results obtained from the survey were presented using descriptive statistics and multiple regression analysis. The data was a representation of agile software development project managers from IT organizations located in Texas. The survey data included factors related to the failure and success of agile software development projects. The collected data was analyzed based on answers provided by the respondents to answer the research question.

The research question (RQ) was to identify if a relationship exists between agile software development project success rates and software development team experience, team size, and level of effort estimation accuracy. The answers provided to questions 21, 41, 44, and 48-51 were considered. The answers provided by the respondents confirmed that characteristics of the agile software development team contribute to the success of development projects. The results supported the assertion that software development team composition directly impacts the success of a software development project.

Multiple linear regression analysis was conducted to test the possibility of predicting project success. The independent variable variables are the Critical Success Factors of agile software development projects. The F-statistics indicated that the linear model fits well. Using this model, I demonstrated that the linear combination of the three predictor variables does not have a significant predictive power over the dependent variable. Therefore, the p-value was greater than 0.05% and we fail to reject the null hypothesis: there is no significant relationship between software development team size,

experience, level of effort estimation accuracy, and software development project success rates.

Applications to Professional Practice

IT professionals and software development project managers can leverage the findings of this study to understand how complementarity and other factors can drive project success rates. Each software development project is different; therefore, each team may differ. The findings of this study can be used to aid project managers in comprising the right team members to produce the best results. Software development teams often comprise members with varying experience levels and expertise. Specifically, software developers have varying degrees of understanding regarding the effort required to perform a given task.

In addition, the findings of this survey can help identify the correct composition of team members by accounting for team member's level of experience and the number of available developers to give agile software development projects a greater chance of success. The findings suggest that software development team size is a critical factor in the success of a software development project. This statistic can benefit IT professionals and software development project managers by suggesting that smaller teams are more successful. Ultimately, the project's stakeholders benefit from a well-formed and balanced software development team. The right combination of team members and accurate effort estimations can produce successful projects that meet or exceed budgetary and timeline constraints while delivering a quality product.

Implications for Social Change

Many factors can affect the outcome of an agile software development project. In addition, there are several implications for social change within software development and project management. Leveraging this study, project managers could learn the optimal team composition. Organizations may consider restructuring teams to be smaller and more agile, leading to more efficient resource allocation and better collaboration among team members.

In addition, project managers can leverage this study to understand the correlation between team experience and agile software development project success. Organizations may prioritize investing in training and development programs to enhance the skills and expertise of their software development teams. This could lead to higher-quality work, improved productivity, and excellent project success rates. Project managers may also use the study to identify a correlation between level of effort estimation accuracy and software development project success. Organizations could focus on improving estimation techniques and investing in tools or training to support more accurate estimation practices, thus leading to better project planning, budgeting, and resource allocation.

After viewing this study, organizations may encourage more collaboration and knowledge sharing among team members. This could involve creating opportunities for mentorship, fostering a culture of continuous learning, and promoting interdisciplinary collaboration within software development teams. By leveraging the collective expertise of team members, organizations can improve project outcomes and foster innovation.

This study may also enhance project management practices by accommodating smaller, more experienced teams and prioritizing activities that improve the level of effort estimation accuracy. This could lead to more effective project management, reduced risks, and increased stakeholder satisfaction.

Recommendations for Action

For organizations to have higher agile software development project success rates, project managers should pay close attention to the fitness of software development team members and the team's overall composition. Project managers can measure an individual's development experience, experience working in a team environment, and estimation accuracy. Then, project managers can evaluate the team member's capabilities and project requirements to ensure the best fit possible. A best-fit scenario is essential to form a balance so the team has the resources and expertise to accomplish the project's goals effectively.

Organizations and project managers can offer opportunities for continuous learning, skill enhancement, and professional development to improve the expertise and effectiveness of software development teams. This investment can lead to higher-quality work, increased productivity, increased stakeholder satisfaction, and greater project success. In addition, organizations should promote training on improving effort estimation techniques. This may include training and using historical data on past projects to inform estimates and implementation of tools or software to support more accurate estimates.

Although positive, the effect of team expertise on agile software development success was not statistically significant in this study. However, it would benefit organizations to foster a culture of collaboration, communication, and interdisciplinary teamwork within software development teams. This will provide mentorship, peer learning, and cross-functional collaboration opportunities to enhance problem-solving, innovation, and team performance.

Recommendations for Further Study

In this study, I considered only the project managers' perspective since the purpose was to aid IT leaders and project management in making more informed decisions regarding software development team composition. Given the lack of confirmatory statistical significance for two of the predictor variables, a research opportunity for further research is capturing software developer's viewpoints regarding their assessment of project management involvement. Additional research capturing the software developer's perspective would help to corroborate or challenge the results presented in this study.

In this study, I focused on software development team factors such as team experience and size and their effects on software development project success. I also considered the accuracy of effort estimation and its impact on software development project success. However, several other factors, software development team-related, and otherwise, affect agile software development project outcomes. Some of these include the impact of stakeholder involvement and influence in decision-making and the overall satisfaction of the stakeholder compared to budgetary and timeline constraints. The

impact of team member turnover within a development cycle may also be considered. Additional research can be done to quantify some of these additional factors and build a complete model for optimal software development team composition.

The results of this study suggest that out of the three independent variables, more experienced team members are more likely to have successful project outcomes. Additional research could be done to quantify what the optimal team experience threshold would be for the maximum agile software development project success rate.

This study used a relatively small population due to time and resource limitations. This sample satisfied the requirements for minimal statistical power; however, a larger sample and greater participation by more software development project managers would lead to more substantial statistical results and greater result confidence.

Reflections

Over the past 3 years, I have studied various factors affecting the success of an agile software development project. Having spent the better part of my career on the software developer side of development projects, I have witnessed many project managers experience less than satisfactory results from development teams. I believed project managers would reap better results if they understood and could quantify the synergy between team members' capabilities and project requirements. I sought to further the research by Chow and Cao (2008) by quantifying specific characteristics of agile software development projects and applying them through the lens of project management. Additionally, I sought to quantify the effect of other software development factors, such as the organization's acceptance of the agile methodology. I hoped for

stronger correlations and statistical significance for the agile software development project success predictors. However, my main hypothesis was supported by the findings of this study. I realize that many factors must be considered when quantifying and predicting the outcome of an agile software development project. Ultimately, I am satisfied that I can contribute to the industry and body of knowledge.

Summary and Study Conclusions

Software development projects are often delivered late, and costs exceed budgetary limitations by 35% due to complexities of various technical and social factors (Kula et al., 2022; Saeeda et al., 2020). Considering the significant resources organizations commit to software development projects, it would benefit IT and organizational leadership to strongly consider the relationship between software development team composition and project requirements to improve the return on investment and increase project success rates.

References

Abdalhamid, S., & Mishra, A. (2017). Adopting of agile methods in software development organizations: Systematic mapping. *TEM Journal*, *6*(4), 817-825. https://doi.org/10.18421/TEM64-22

Abualkishik, A. Z., Ferrucci, F., Gravino, C., Lavazza, L., Liu, G., Meli, R., & Robiolo, G. (2017). A study on the statistical convertibility of IFPUG Function Point, COSMIC Function Point and Simple Function Point. *Information and Software Technology*, *86*, 1–19. https://doi.org/10.1016/j.infsof.2017.02.005

Abualkishik, A. Z., & Lavazza, L. (2018). IFPUG function points to COSMIC function points convertibility: A fine-grained statistical approach. *Information and Software Technology*, *97*(July 2017), 179–191. https://doi.org/10.1016/j.infsof.2018.01.012

Abu-Bader, S. H. (2021). Using statistical methods in social science research: With a complete SPSS guide (3rd ed.). Oxford University Press.

Abutabenjeh, S., & Jaradat, R. (2018). Clarification of research design, research methods, and research methodology. *Teaching Public Administration*, *36*(3), 237–258. https://doi.org/10.1177/0144739418775787

Adnan, M., & Afzal, M. (2017). Ontology based multiagent effort estimation system for scrum agile method. *IEEE Access*, *5*, 25993–26005. https://doi.org/10.1109/ACCESS.2017.2771257

Ahmed, A. R., Tayyab, M., Bhatti, S. N., Alzahrani, A. J., & Babar, M. I. (2017). Impact of story point estimation on product using metrics in scrum development process.

International Journal of Advanced Computer Science and Applications, 8(4), 385–391. https://doi.org/10.14569/IJACSA.2017.080452

Alavi, H., & Hąbek, P. (2016). Addressing research design problem in mixed methods research. *Management Systems in Production Engineering, 21*(1), 62–66. https://doi.org/10.12914/MSPE-10-01-2016

Ali, S. S., Ren, J., Zhang, K., Wu, J., & Liu, C. (2023). Heterogeneous Ensemble Model to Optimize Software Level of effort estimation accuracy. *IEEE Access, Access, IEEE, 11*, 27759–27792. https://doi.org/10.1109/ACCESS.2023.3256533

Almadhoun, W., & Hamdan, M. (2018). Optimizing the Self-Organizing Team Size Using a Genetic Algorithm in Agile Practices. *Journal of Intelligent Systems, 29*(1), 1151–1165. https://doi.org/10.1515/jisys-2018-0085

Almakadmeh, K., Al-Sarayreh, K. T., & Meridji, K. (2018). A measurement model of the functional size of software maintainability requirements. *Journal of Theoretical and Applied Information Technology, 96*(12), 3829–3845. https://www.jatit.org/

Alostad, J. M., Abdullah, L. R. A., & Aali, L. S. (2017). A Fuzzy based model for effort estimation in scrum projects. *International Journal of Advanced Computer Science and Applications, 8*(9), 270–277. www.ijacsa.thesai.org

Anooja, A., & Rajawat, S. (2017). Comparative analysis of software cost-effort estimation and agile in perspective of software development. *International Journal of Advanced Research in Computer Science, 8*(8), 121–125. https://doi.org/10.26483/ijarcs.v818.4666

Arcos-Medina, G., & Mauricio, D. (2020). Identifying Factors Influencing on Agile

Practices for Software Development. *Journal of Information and Organizational Sciences, 44*(1), 1–31. https://doi.org/10.31341/jios.44.1.1

Arifin, H. H., Daengdej, J., & Khanh, N. T. (2017). An empirical study of effort-size and effort-time in expert-based estimations. *Proceedings - 8th IEEE International Workshop on Empirical Software Engineering in Practice, IWESEP 2017*, 35–40. https://doi.org/10.1109/IWESEP.2017.21

Atkinson, R. (1999). Project management: cost, time, and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International Journal of Project Management, 17*(6), 337–342. https://doi.org/10.1016/S0263-7863(98)00069-6

Azzeh, M., Bou Nassif, A., & Attili, I. B. (2021). Predicting software effort from use case points: A systematic review. *Science of Computer Programming, 204*. https://doi.org/10.1016/j.scico.2020.102596

Bahariniya, S., Ezatiasar, M., & Madadizadeh, F. (2021). A brief review of the types of validity and reliability of scales in medical research. *Journal of Community Health Research, 10*(2), 100–102. http://dx.doi.org/10.18502/jchr.v10i2.6582

Bajpai, R., & Chaturvedi, H. K. (2021). Toward a more nuanced interpretation of statistical significance in biomedical research. *Asian Journal of Oncology, 7*(2), 049–051. https://doi.org/10.1055/s-0041-1727066

Ball, H. L. (2019). Conducting online surveys. *Journal of Human Lactation, 35*(3), 413–417. https://doi.org/10.1177/0890334419848734

Baruah, N. (2015). Requirement management in agile software environment. *Procedia*

Computer Science, 62(Scse), 81–83. https://doi.org/10.1016/j.procs.2015.08.414

Basias, N., & Pollalis, Y. (2018). Quantitative and qualitative research in business & technology: Justifying a suitable research methodology. *Review of Integrative Business and Economics Research, 7*(1), 91–105. http://buscompress.com/journalhome.html

Belassi, W., & Tukel, O. I. (1996). A new framework for determining critical success/failure factors in projects. *International Journal of Project Management, 14*(3), 141–151.

Berger, H., & Beynon-Davies, P. (2009). The utility of rapid application development in large-scale, complex projects. *Information Systems Journal. 19*(6), 549-570. https://doi.org/10.1111/j.1365-2575.2009.00329.x

Bergmann, T. (2018). *The Relationship between Agile Project Management and Project Success Outcomes*.

Bergmann, T., & Karwowski, W. (2018), Agile project management and project success: A literature review, presented at the international conference on applied human factors and ergonomics, Springer, pp. 405–414.

Bermejo, S., Zambalde, A. L., Tonelli, A. O., Souza, S. A., Zuppo, L. A., & Rosa, P. L. (2014). Agile principles and achievement of success in software development: A quantitative study in Brazilian organizations. *Procedia Technology, 16*(2014), 718-727. https://doi.org/10.1016/j.protcy.2014.10.021

Bilgaiyan, S., Mishra, S., & Das, M. (2016). A review of software cost estimation in agile software development using soft computing techniques. *2016 2nd International*

Conference on Computational Intelligence and Networks (CINE), 112–117. https://doi.org/10.1109/CINE.2016.27

Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2017). A systematic review on software cost estimation in agile software development. *Journal of Engineering Science and Technology Review*, *10*(4), 51–64. https://doi.org/10.25103/jestr.104.08

Bloomfield, J., & Fisher, M. J. (2019). Quantitative research design. *Journal of the Australasian Rehabilitation Nurses' Association*, *22*(2), 27–30. https://doi.org/10.33235/jarna.22.2.27-30

Boby, J. B., Kadadevaramath, R. S., & Edinbarough, I. A. (2017). Designing software development processes to optimize multiple output performance characteristics. *Software Quality Professional*, *19*(4), 16–24. http://www.asq.org

Boehm, B. (1981). *Software Engineering Economics*. Prentice Hall.

Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B., Hoirowitz, E., & Steece, B. (2000). *Software cost estimation with Cocomo II*. Prentice Hall PTR.

Bogopa, M., & Marnewick, C. (2022). Critical success factors in Software Development Projects. *South African Computer Journal*, *34*(1). https://doi.org/10.18489/sacj.v34i1.820

Bourque, P., & Fairley, R. E. (2014). Guide to the software engineering body of knowledge. *IEEE Computer Society Press*. https://doi.org/10.1234/12345678

Brad, M. C., Birloi, F., Bratulescu, A., & Blaga, I. B. (2016). A comparative study of agile project management software tools. *Economy Informatics*, *16*(1), 27–39.

http://www.economyinformatics.ase.ro

Bullen, C. V., & Rockart, J. F. (1981). A primer on critical success factors (Working Paper No. 69), Massachusetts Institute of Technology, *Sloan School of Management, Center for Information Systems Research*, Cambridge, Massachusetts. https://www.scirp.org/(S(czeh2tfqw2orz553k1w0r45))/reference/referencespapers.aspx?referenceid=1530263

Burke, R. (2011). *Advanced Project Management: Fusion Method Xyz: A Project Methodology Systems Approach for the Project Sponsor to Implement Corporate Strategy*. Burke Publishing.

Busse, C., Kach, A. P., & Wagner, S. M. (2017). Boundary conditions: What they are, how to explore them, why we need them, and when to consider them. *Organizational Research Methods, 20*(4), 574–609. https://doi.org/10.1177/1094428116641191

Butler, C. W., Vijayasarathy, L. R., & Roberts, N. (2020). Managing Software Development Projects for Success: Aligning Plan- and Agility-Based Approaches to Project Complexity and Project Dynamism. *Project Management Journal, 51*(3), 262–277. https://doi.org/10.1177/8756972819848251

Butt, S. A., Khalid, A., Ercan, T., Ariza-Colpas, P. P., Melisa, A.-C., Piñeres-Espitia, G., De-La-Hoz-Franco, E., Melo, M. A. P., & Ortega, R. M. (2022). A software-based cost estimation technique in Scrum using a developer's expertise. *Advances in Engineering Software, 171*, N.PAG.

https://doi.org/10.1016/j.advengsoft.2022.103159

Chantarangsi, W., Liu, W., Bretz, F., Kiatsupaibul, S., & Hayter, A. J. (2018). Normal probability plots with confidence for the residuals in linear regression. *Communications in Statistics – Simulation and Computation*, *47*(2), 367–379. https://doi.org/10.1080/03610918.2016.1165840

Chen, M., Pustejovsky, J. E., Klingbeil, D. A., & Van Norman, E. R. (2023). Between-case standardized mean differences: Flexible methods for single-case designs. *Journal of School Psychology*, *98*, 16–38. https://doi.org/10.1016/j.jsp.2023.02.002

Chiang, H. Y., & Lin, B. M. T. (2020). A Decision Model for Human Resource Allocation in Project Management of Software Development. *IEEE Access, Access, IEEE*, *8*, 38073–38081. https://doi.org/10.1109/ACCESS.2020.2975829

Chiyangwa, T., & Mnkandla, E. (2018). Agile methodology perceived success and its use: The moderating effect of perceived compatibility. *South African Computer Journal*, *30*(2). https://doi.org/10.18489/sacj.v30i2.554

Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T. T. M., Ghose, A., & Menzies, T. (2018). A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*, *14*(8). https://doi.org/10.1109/TSE.2018.2792473

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems & Software*, *81*(6), 961–971. https://doi.org/10.1016/j.jss.2007.08.020

Ciric-Lalic, D., Lalic, B., Delić, M., Gracanin, D., & Stefanovic, D. (2022). How project

management approach impact project success? From traditional to agile. *International Journal of Managing Projects in Business*, *15*(3), 494–521. https://doi.org/10.1108/IJMPB-04-2021-0108

Cohn, M., & Ford, D. (2003). Introducing an agile process to an organization [software development]. *Computer*, *36*(6), 74–78. https://doi.org/10.1109/MC.2003.1204378

Coleman, G. (2016). Agile software development. *Software Quality Professional*, *19*(1), 23–29. http://asq.org

Conoscenti, M., Besner, V., Vetrò, A., & Fernández, D. M. (2019). Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. *The Journal of Systems & Software*, *156*, 126–135. https://doi.org/10.1016/j.jss.2019.06.075

Couper, M., & Miller, P. (2008). Web Survey Methods: Introduction, *Public Opinion Quarterly*, *72*(5), Pages 831–835, https://doi.org/10.1093/poq/nfn066

Cramer, M., Atwood, J., & Stoner, J. (2006). Measuring community coalition effectiveness using the ICE instrument. *Public Health Nursing*, *23*(1), 74–87. https://doi.org/10.1111/j.0737-1209.2006.230111.x

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage Publications.

Creswell, J. W., & Poth, C. N. (2018). Qualitative inquiry & research design: *Choosing among five approaches (4th ed.)*. Sage Publications, Inc.

Cronbach, L. J. (1951) Coefficient alpha and the internal structure of

tests. *Psychometrika 16*, 297–334. https://doi.org/10.1007/BF02310555

Curtis, E. A., Comiskey, C., & Dempsey, O. (2016). Importance and use of correlational research. *Nurse Researcher, 23*(6), 20–25. https://doi.org/10.7748/nr.2016.e1382

Dewi, R. S., & Subriadi, A. P. (2017a). A comparative study of software development size estimation method: UCPabc vs function points. *Procedia Computer Science, 124*, 470–477. https://doi.org/10.1016/j.procs.2017.12.179

Dewi, R. S., & Subriadi, A. P. (2017b). A modification complexity factor in function points method for software cost estimation towards public service application. *Procedia Computer Science, 124*, 415–422. https://doi.org/10.1016/j.procs.2017.12.172

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large scale agile transformations: A systematic literature review. *The Journal of Systems & Software, 119*, 87-108. https://doi.org/10.1016/j.jss.2016.06.013

Dobakhti, L. (2020). The Process of Enhancing Validity, Reliability, and Ethics in Research. *Iranian Journal of Applied Language Studies, 12*(2), 59–88. https://doi.org/10.22111/ijals.2020.5978

Dönmez, D., & Grote, G. (2018). Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities. *Information and Software Technology, 93*, 94–111. https://doi.org/10.1016/j.infsof.2017.08.015

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software, 127*,

109–119. https://doi.org/10.1016/j.jss.2017.01.027

Drury-Grogan, M. L., Conboy, K., & Acton, T. (2017). Examining decision characteristics & challenges for agile software development. *Journal of Systems and Software*, *131*, 248–265. https://doi.org/10.1016/j.jss.2017.06.003

Duckett, L. J. (2021). Quantitative research excellence: Study design and reliable and valid measurement of variables. *Journal of Human Lactation*, *37*(3), 456–463. https://doi.org/10.1177/08903344211019285

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Ebert, C., & Paasivaara, M. (2017). Scaling agile. *IEEE Software*, *34*(6), 98–103. https://doi.org/10.1109/MS.2017.4121226

Effendi, A., Setiawan, R., & Rasjid, Z. E. (2019). Adjustment Factor for Use Case Point Software Effort Estimation (Study Case: Student Desk Portal). *Procedia Computer Science*, *157*, 691–698. https://doi.org/10.1016/j.procs.2019.08.215

Engel, R. J., & Schutt, R. K. (2017). *The practice of research in Social Work* (4th ed.). SAGE.

Evans, J. R., & Mathur, A. (2018). The value of online surveys: a look back and a look ahead. *Internet Research*, *28*(4), 854–887. https://doi.org/10.1108/IntR-03-2018-0089

Fabrigar, L. R., Wegener, D. T., & Petty, R. E. (2020). A validity-based framework for understanding replication in psychology. *Personality and Social Psychology*

Review, *24*(4), 316–344. https://doi.org/10.1177/1088868320931366

Farghaly, A. (2018). Comparing and contrasting quantitative and qualitative research approaches in education: The peculiar situation of medical education. *Education in Medicine Journal*, *10*(1), 3–11. https://doi.org/10.21315/eimj2018.10.1.2

Flannelly, K. J., Flannelly, L. T., & Jankowski, K. R. B. (2018). Threats to the internal validity of experimental and quasi-experimental research in healthcare. *Journal of Health Care Chaplaincy*, *24*(3), 107–130. https://doi.org/10.1080/08854726.2017.1421019

Fowler, M. (2000). Put Your Process on a Diet, *Software Development*, *8*(12). http://martinfowler.com/articles/newMethodology.html

Fox, R. (2016). The true cost. *Digital Library Perspectives*, *32*(4), 232–238. https://doi.org/10.1108/DLP-07-2016-0018

Freire, A., Perkusich, M., Saraiva, R., Almeida, H., & Perkusich, A. (2018). A Bayesian networks-based approach to assess and improve the teamwork quality of agile teams. *Information and Software Technology*, *100*, 119–132. https://doi.org/10.1016/j.infsof.2018.04.004

Gandomani, T. J., Ghani, A., Zulzalil, H., Azim, A., Bakar, A., Sultan, M., & Nafchi, M. Z. (2013). Obstacles in moving to agile software development methods: At a glance. *Journal of Computer Science*, *9*(5), 620–625. https://doi.org/10.3844/jcssp.2013.620.625

Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., & Demirörs, O. (2019). Correlation of critical success factors with success of software projects: an empirical

investigation. *Software Quality Journal*, *27*(1), 429–493. https://doi.org/10.1007/s11219-018-9419-5

Geier, C., Adams, R. B., Mitchell, K. M., & Holtz, B. E. (2021). Informed consent for online research-is anybody reading?: Assessing comprehension and individual differences in readings of digital consent forms. *Journal of Empirical Research on Human Research Ethics: JERHRE*, *16*(3), 154–164. https://doi.org/10.1177/15562646211020160

Gohary, T. (2019). Hypothesis testing, type I and type II errors: Expert discussion with didactic clinical scenarios. *International Journal of Health and Rehabilitation Sciences (IJHRS)*, *8*(3), 132. https://doi.org/10.5455/ijhrs.0000000180

Govil, N., & Sharma, A. (2022). Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors. *Advances in Engineering Software*, p. *172*, N.PAG. https://doi.org/10.1016/j.advengsoft.2022.103209

Gren, L., Goldman, A., & Jacobsson, C. (2020). Agile ways of working: A team maturity perspective. *Journal of Software: Evolution & Process*, *32*(6), 1–13. https://doi.org/10.1002/smr.2244

Gren, L., Torkar, R., & Feldt, R. (2017). Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *The Journal of Systems & Software*, *124*, 104–119. https://doi.org/10.1016/j.jss.2016.11.024

Haegele, J. A., & Hodge, S. R. (2015). Quantitative methodology: A guide for emerging

physical education and adapted physical education researchers. *Physical Educator, 72*(5), 59–75. https://doi.org/10.18666/TPE-2015-V72-I5-6133

Haines, T., Idenudia, E., & Raisinghani, M. S. (2017). The conceptual model for agile tools and techniques. *American Journal of Management, 17*(2007), 77–88. http://www.na-businesspress.com/ajmopen.html

Hall, J. L., & Van Ryzin, G. G. (2019). A norm of evidence and research in decisionmaking (nerd): Scale development, reliability, and validity. *Public Administration Review, 79*(3), 321–329. https://doi.org/10.1111/puar.12995

Hans, A., & Gahlot, S. (2016). A review: Software metrics and effort estimation in aspect software development program. *International Journal of Advanced Research in Computer Science, 7*(1). http://www.ijarcs.info

Harzl, A. (2017). Can FOSS projects benefit from integrating Kanban: a case study. *Journal of Internet Services and Applications, 8*(1). https://doi.org/10.1186/s13174-017-0058-z

Henriksen, A., & Pedersen, S. A. R. (2017). A qualitative case study on agile practices and project success in agile software projects. *Journal of Modern Project Management*, 62–73. https://doi.org/10.19255/JMPM01306

Hickey, G. L., Grant, S. W., Dunning, J., & Siepe, M. (2018). Statistical primer: Sample size and power calculations—why, when and how? *European Journal of CardioThoracic Surgery, 54*(1), 4–9. https://doi.org/10.1093/ejcts/ezy169

Idri, A., Amazal, F. A., & Abran, A. (2015). Analogy-based software development effort estimation: A systematic mapping and review. *Information and Software*

Technology, 58, 206–230. https://doi.org/10.1016/j.infsof.2014.07.013

Indrayan, A. (2020). Attack on statistical significance: A balanced approach for medical research. *Indian Journal of Medical Research, 151*(4), 275–278. https://doi.org/10.4103/ijmr.IJMR_980_19

Iqbal, J. (2019). The Impact of Agile Methodologies and Cost Management Success Factors: An Empirical Study. *Baghdad Science Journal, 16*(2). https://doi.org/10.21123/bsj.2019.16.2(SI).0496

Ivan, I., & Despa, M. L. (2016). Estimating maintenance cost for web applications. *Informatica Economica, 20*(4/2016), 62–75. https://doi.org/10.12948/issn14531305/20.4.2016.06

Jadhav, S. K., Shaga, V., & Thorat, S. D. (2017). User stories for proposed web application using agile. *International Journal of Advanced Research in Computer Science, 8*(9), 570–574. https://doi.org//10.26483/ijarcs.v8i9.5148

Johns, D., Whittaker, C., & Cox-Ganser, J. (2022). Impacts of risk assessment data, assumptions, and methods: Considering the evidence for diacetyl and 2,3-pentanedione. *Frontiers in Public Health, 10*. https://doi.org/10.3389/fpubh.2022.972136

Jonsen, K., Fendt, J., & Point, S. (2018). Convincing qualitative research: What constitutes persuasive writing?. *Organizational Research Methods, 21*(1), 30–67. https://doi.org/10.1177/1094428117706533

Jordan, K. (2018). Validity, reliability, and the case for participant-centered research: Reflections on a multi-platform social media study. *International Journal of*

Human-Computer Interaction, 34(10), 913–921. https://doi.org/10.1080/10447318.2018.1471570

Jørgensen, M., Bergersen, G. R., & Liestol, K. (2021). Relations Between Effort Estimates, Skill Indicators, and Measured Programming Skill. *IEEE Transactions on Software Engineering, Software Engineering, IEEE Transactions on, IIEEE Trans. Software Eng*, *47*(12), 2892–2906. https://doi.org/10.1109/TSE.2020.2973638

Kaeedi, A., Esfahani, A. R. N., Sharifian, F., & Moosavipour, S. (2023). The Quantitative and Qualitative Study of the Effectiveness of the Problem-based Learning Approach in Teaching Research Methods. *Journal of University Teaching & Learning Practice*, *20*(5), 1–27. https://doi.org/10.53761/1.20.5.06

Kaewkungwal, J., & Adams, P. (2019). Ethical consideration of the research proposal and the informed-consent process: An online survey of researchers and ethics committee members in Thailand. *Accountability in Research: Policies & Quality Assurance*, *26*(3), 176–197. https://doi.org/10.1080/08989621.2019.1608190

Kandengwa, E., & Khoza, L. T. (2021). Measuring Agile software project success beyond the triple constraint. *South African Journal of Information Management*, *23*(1), 1–8. https://doi.org/10.4102/sajim.v23i1.1375

Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *The Journal of Systems & Software*, *172*(2021), 1-26. https://doi.org/10.1016/j.jss.2020.110851

Kaur, A. (2017). Comparison of maintenance activity for effort estimation in open source software projects. *International Journal of Advanced Research in Computer Science*, *8*(5), 1535–1539. http://www.ijarcs.info/

Khan, A. A., Shameem, M., Nadeem, M., & Akbar, M. A. (2021). Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping. *Applied Soft Computing Journal*, *102*. https://doi.org/10.1016/j.asoc.2021.107090

Khuat, T. T., & Le, H. (2016). Optimizing parameters of software effort estimation models using directed artificial bee colony algorithm. *Informatica*, *40*, 427–436. http://www.informatica.si/index.php/informatica

Koch, A. S. (2005). *Agile software development. [electronic resource] : evaluating the methods for your organization*. Artech House.

Köhler, T., Landis, R. S., & Cortina, J. M. (2017). From the Editors: Establishing Methodological Rigor in Quantitative Management Learning and Education Research: The Role of Design, Statistical Methods, and Reporting Standards. *Academy of Management Learning & Education*, *16*(2), 173–192. https://doi.org/10.5465/amle.2017.0079

Krstić, M., Skorup, A., & Lapčević, G. (2018). Trends in agile innovation management. *International Review*, (3–4), 58–70. https://doi.org/10.5937/intrev1804058k

Kukreja, N., & Garg, U. (2017). Effort estimation of object oriented system using stochastic tree boosting technique. *International Journal of Advanced Research in Computer Science*, *8*(7), 91–96. https://doi.org/10.26483/ijarcs.v8i7.4246

Kula, E., Greuter, E., van Deursen, A., & Gousios, G. (2022). Factors Affecting On-Time Delivery in Large-Scale Agile Software Development. *IEEE Transactions on Software Engineering, Software Engineering, IEEE Transactions on, IIEEE Trans. Software Eng*, *48*(9), 3573–3592. https://doi.org/10.1109/TSE.2021.3101192

Kuutila, M., Mäntylä, M., Farooq, U., & Claes, M. (2020). Time pressure in software engineering: A systematic review. *Information and Software Technology*, *121*. https://doi.org/10.1016/j.infsof.2020.106257

LeCompte, M., & Goetz, J. (1982). Problems of reliability and validity in ethnographic research. *Review of Educational Research*, *52*(1), 31-60.

Lester, D. H. (1998). Critical Success Factors for New Product Development. *Research-Technology Management*, *41*(1), 36–43.

Levitt, H. M., Motulsky, S. L., Wertz, F. J., Morrow, S. L., & Ponterotto, J. G. (2017). Recommendations for Designing and Reviewing Qualitative Research in Psychology: Promoting Methodological Integrity. *Qualitative Psychology*, *4*(1), 2–22. https://doi.org/10.1037/qup0000082

Lima, J. E., West, G. B., Winston, B. E., & Wood, J. A. (2016). Measuring organizational cultural intelligence. *International Journal of Cross Cultural Management*, *16*(1), 9–31. https://doi.org/10.1177/1470595815615625

Lindell, M. K., & Whitney, D. J. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology*, *86*(1), 114–121. https://doi.org/10.1037/0021-9010.86.1.114

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., & Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, *37*(12), 26–34. https://doi.org/10.1109/MC.2004.231

Loeb, S., Dynarski, S., McFarland, D., Morris, P., Reardon, S., & Reber, S. (2017). Descriptive analysis in education: A guide for researchers (NCEE 2017-4023). National Center for Education Evaluation and Regional Assistance, i–B5.

Lopez-Martinez, J., Ramirez-Noriega, A., Juarez-Ramirez, R., Licea, G., & Martinez-Ramirez, Y. (2017). Analysis of Planning Poker Factors between University and Enterprise. *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 54–60. https://doi.org/10.1109/CONISOFT.2017.00014

Mahmood, Y., Kama, N., & Azmi, A. (2020). A systematic review of studies on use case points and expert-based estimation of software development effort. *Journal of Software: Evolution & Process*, *32*(7), 1–20. https://doi.org/10.1002/smr.2245

Malhotra, N. K., Kim, S. S., & Patil, A. (2006). Common Method Variance in IS Research: A Comparison of Alternative Approaches and a Reanalysis of Past Research. *Management Science*, *52*(12), 1865–1883. https://doi.org/10.1287/mnsc.1060.0597

Mari, F., Pratiwi, U., Oktanisa, I., & Utaminingrum, F. (2019). Comparative study of numerical methods in multiple linear regression for stock prediction Jakarta Islamic Index (JII). *2019 International Conference on Sustainable Information Engineering and Technology (SIET), Sustainable Information Engineering and*

Technology (SIET), 2019 International Conference On, 110–115. https://doi.org/10.1109/SIET48054.2019.8985999

Martin, R. C. (2003). Agile Software Development: Principles, patterns, and practices. Prentice Hall.

Matović, N., & Ovesni, K. (2023). Interaction of quantitative and qualitative methodology in mixed methods research: Integration and/or combination. *International Journal of Social Research Methodology: Theory & Practice*, *26*(1), 51–65. https://doi.org/10.1080/13645579.2021.1964857

McCarthy, C. J., Whittaker, T. A., Boyle, L. H., & Eyal, M. (2017). Quantitative approaches to group research: Suggestions for best practices. *Journal for Specialists in Group Work*, *42*(1), 3–16. https://doi.org/10.1080/01933922.2016.1264520

McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press.

McCusker, K., & Gunaydin, S. (2015). Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion*. *30*(7), 537-542. https://doi.org/10.1177/0267659114559116

McEwan, B. (2020). Sampling and validity. *Annals of the International Communication Association*, *44*(3), 235–247. https://doi.org/10.1080/23808985.2020.1792793

Mehta, S., & Kumari, S. (2016). A tool to evaluate the performance of UCP. *International Journal of Engineering Science*, *6*(7), 8193–8198. https://doi.org/10.4010/2016.1913

Miller, C. J., Smith, S. N., & Pugatch, M. (2020). Experimental and quasi-experimental

designs in implementation research. *Psychiatry Research, 283*. 1–7. https://doi.org/10.1016/j.psychres.2019.06.027

Miranda, E. (2017). Documentless assessments using nominal group interviews. *Software Quality Professional, 19*(2). http://www.asq.org/pub/sqp/

Mirzaei, M., & Mabin, V. (2017). Agile project management and public policy development projects: A case study from New Zealand. *New Zealand Journal of Applied Business Research, 15*(1), 59–76. https://www.manukau.ac.nz/

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *The Journal of Systems & Software, 82*(11), 1869–1890. https://doi.org/10.1016/j.jss.2009.05.052

Mittas, N., Papatheocharous, E., Angelis, L., & Andreou, A. S. (2015). Integrating non-parametric models with linear components for producing software cost estimations. *Journal of Systems and Software, 99*, 120–134. https://doi.org/10.1016/j.jss.2014.09.025

Moharreri, K., Sapre, A. V., Ramanathan, J., & Ramnath, R. (2016). Cost-effective supervised learning models for software effort estimation in agile environments. *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, 135–140. https://doi.org/10.1109/COMPSAC.2016.85

Morris, P. W. G., & Hough, G. H. (1987). *The anatomy of major projects: A study of the reality of project management*. John Wiley & Sons Ltd.

Moser, A., & Korstjens, I. (2018). Series: Practical guidance to qualitative research. Part 3: Sampling, data collection and analysis. *European Journal of General Practice,*

*24(*1), 9–18. https://doi.org/10.1080/13814788.2017.1375091

Munns, A. K., & Bjeirmi, B. F. (1996). The role of project management in achieving project success. *International Journal of Project Management, 14*(2), 81–87.

Mweshi, G. K., & Sakyi, K. (2020). Application of sampling methods for the research design. *Archives of Business Research, 8*(11), 180–193. https://doi.org/10.14738/abr.811.9042

Naik, P., & Nayak, S. (2017). Insights on research techniques towards cost estimation in software design. *International Journal of Electrical and Computer Engineering (IJECE), 7*(5), 2883. https://doi.org/10.11591/ijece.v7i5.pp2883-2894

Naslund, D., & Kale, R. (2020). Is agile the latest management fad? A review of success factors of agile transformations. *International Journal of Quality and Service Sciences, 12*(4), 489–504. https://doi.org/10.1108/IJQSS-12-2019-0142

National Commission for the Protection of Human Subjects and Biomedical and Behavioral Research. (1979). The Belmont Report: Ethical principles and guidelines for the protection of human subjects of research. U.S. Department of Health and Human Services. https://www.hhs.gov/ohrp/humansubjects/guidance/Belmont.html

Nguyen, V. D., & Nguyen, N. T. (2018). Intelligent collectives: Theory, applications, and research challenges. *Cybernetics and Systems, 0*(0), 1–19. https://doi.org/10.1080/01969722.2017.1418254

Nhung, H. L. T. K., Van Hai, V., Silhavy, R., Prokopova, Z., & Silhavy, P. (2022). Parametric Software Effort Estimation Based on Optimizing Correction Factors

and Multiple Linear Regression. *IEEE Access, Access, IEEE*, *10*, 2963–2986. https://doi.org/10.1109/ACCESS.2021.3139183

Norouzian, R. (2020). Sample size planning in quantitative l2 research: A pragmatic approach. *Studies in Second Language Acquisition*, *42*(4), 849–870. https://doi.org/10.1017/S0272263120000017

Norris, J. M., Plonsky, L., Ross, S. J., & Schoonen, R. (2015). Guidelines for reporting quantitative methods and results in primary research. *Language Learning*, *65*, 470–476. https://doi.org/10.1111/lang.12104

Nunan, D. (1992). Research methods in language learning. Cambridge University Press.

Osman, H., & Musa, M. (2016). A survey of agile software estimation methods. *International Journal of Computer Science and Telecommunications*, *7*(3), 38–42. http://www.ijcst.org/Volume7/Issue3/p6_7_3.pdf

Osmanbegović, E., Suljić, M., & Agić, H. (2017). A review of estimation of software products development cost. *Ekonomski Vjesnik/Econviews-Review of Contemporary Business, Entrepreneurship and Economic Issues*, *3*, 195–208. https://hrcak.srce.hr/ojs/index.php/ekonomski-vjesnik/article/view/4108

Ovan, O., & Saputra, A. (2020). CAMI: Web-based application to test validity and reliability of research instruments. *Journal of Education, Teaching and Learning*, *5*(2), 244–248. https://doi.org/10.26737/jetl.v5i2.2032

Oyong, S. B., & Ekong, V. E. (2019). An Explorative Survey of Formal and Agile Software Development Methods. *Global Journal of Pure & Applied Sciences*, *25*(1), 71–79. https://doi.org/10.4314/gjpas.v25i1.10

Oztaysi, B., Onar, S. C., & Kahraman, C. (2019). Performance Measurement Model for Software Development Teams Using Interval-valued Intuitionistic Fuzzy Analytic Hierarchy Process. *Journal of Multiple-Valued Logic & Soft Computing*, *33*(4/5), 321–329.

Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Using scrum in a globally distributed project: a case study. *Software Process: Improvement & Practice*, *13*(6), 527–544. https://doi.org/10.1002/spip.402

Pereira dos Santos, L., & Ferreira, M. (2019). Applying COCOMO II for a DO-178C Safety-Critical Software Effort Estimation. *Journal of Aerospace Technology & Management*, *11*, 1–16. https://doi.org/10.5028/jatm.v11.1031

Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A. (2017). Assisting the continuous improvement of scrum projects using metrics and bayesian networks. *Journal of Software: Evolution and Process*, *29*(6), 1–18. https://doi.org/10.1002/smr.1835

Pinto, J. K., & Slevin, D. P. (1988). Project success: Definitions and measurement techniques. *Project Management Journal*, *19*(1), 67–72.

Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, 184–196. https://doi.org/10.1016/j.jss.2017.11.066

Prakash, B., & Viswanathan, V. (2017). A survey on software estimation techniques in traditional and agile development models. Indonesian Journal of Electrical

Engineering and Computer Science, *7*(3), 867–876. https://doi.org/10.11591/ijeecs.v7.i3.pp867-876

Qi, F., Jing, X. Y., Zhu, X., Xie, X., Xu, B., & Ying, S. (2017). Software effort estimation based on open source projects: Case study of Github. *Information and Software Technology*, *92*, 145–157. https://doi.org/10.1016/j.infsof.2017.07.015

Rahikkala, J., Leppänen, V., Ruohonen, J., & Holvitie, J. (2015). Top management support in software cost estimation: A study of attitudes and practice in Finland. *International Journal of Managing Projects in Business*, *8*(3), 513–532. https://doi.org/10.1108/IJMPB-11-2014-0076

Rai, A., Gupta, G. P., & Kumar, P. (2017). Estimation of software development efforts using improved delphi technique: A novel approach. *International Journal of Applied Engineering Research*, *12*(12), 3228–3236. http://www.ripublication.com/ijaer.htm

Ramirez-Noriega, A., Juarez-Ramirez, R., Navarro, R., & Lopez-Martinez, J. (2016). Using Bayesian networks to obtain the task's parameters for schedule planning in scrum. *Proceedings - 2016 4th International Conference in Software Engineering Research and Innovation, CONISOFT 2016*, (1), 167–174.

Rankovic, N., Rankovic, D., Ivanovic, M., & Lazic, L. (2021). A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays. *IEEE Access, Access, IEEE*, *9*, 26926–26936. https://doi.org/10.1109/ACCESS.2021.3057807

Rashid, J., Kanwal, S., Nisar, M. W., Kim, J., & Hussain, A. (2023). An Artificial Neural

Network-Based Model for Effective Software Development Effort Estimation. *Computer Systems Science & Engineering*, *44*(2), 1309–1324. https://doi.org/10.32604/csse.2023.026018

Raslan, A. T., Darwish, N. R., & Hefny, H. A. (2015). Towards a fuzzy based framework for effort estimation in agile software development. *International Journal of Computer Science and Information Security*, *13*(1), 37. http://search.proquest.com/openview/0ffd04c753ca702dbd632daf9f7ffb81/1?pq-origsite=gscholar&cbl=616671

Rath, S. K., Acharya, B. P., & Satapathy, S. M. (2016). Early stage software effort estimation using random forest technique based on use case points. *IET Software*, *10*(1), 10–17. https://doi.org/10.1049/iet-sen.2014.0122

Reel, J. S. (1999). Critical success factors in software projects. *IEEE Software*, *16*(3), 18–23. https://doi.org/10.1109/52.765782

Rockart, J. F., & Crescenzi, A. D. (1984). Engaging Top Management in Information Technology. *Sloan Management Review (Pre-1986)*, *25*(4), 3. https://www.proquest.com/scholarly-journals/engaging-top-management-information-technology/docview/206805019/se-2

Rosopa, P. J., Schaffer, M. M., & Schroeder, A. N. (2013). Managing Heteroscedasticity in General Linear Models. *Psychological Methods*, *18*(3), 335–351. https://doi.org/10.1037/a0032553

Rutberg, S., & Bouikidis, C. D. (2018). Focusing on the fundamentals: A simplistic differentiation between qualitative and quantitative research. *Nephrology Nursing*

Journal, 45(2), 209–213. https://library.annanurse.org/anna/articles/1898/view

Saeeda, H., Dong, J., Wang, Y., & Abid, M. A. (2020). A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project. *Journal of Software: Evolution & Process, 32*(7), 1–24. https://doi.org/10.1002/smr.2247

Sarstedt, M., Bengart, P., Shaltoni, A. M., & Lehmann, S. (2018). The use of sampling methods in advertising research: A gap between theory and practice. *International Journal of Advertising, 37*(4), 650–663. https://doi.org/10.1080/02650487.2017.1348329

Sehra, S. K., Brar, B. Y., & Kaur, N. (2016). Predominant factors influencing software effort estimation. *International Journal of Computer Science and Information Security, 14*(7), 107–110. https://sites.google.com/site/ijcsis/

Sehra, S. K., Brar, Y. S., Kaur, N., & Sehra, S. S. (2017). Research patterns and trends in software effort estimation. *Information and Software Technology, 0*, 1–21. https://doi.org/10.1016/j.infsof.2017.06.002

Shekhar, S., & Kumar, U. (2016). Review of various software cost estimation techniques. *International Journal of Computer Applications, 141*(11), 31–34. http://www.ijcaonline.org/

Shenhar, A. J., Levy, O., & Dvir, D. (1997). Mapping the dimensions of project success. *Project Management Journal, 28*(2), 5–13.

Šmite, D., Moe, N. B., Šāblis, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and*

Software Technology, *86*, 71–86. https://doi.org/10.1016/j.infsof.2017.01.003

Soni, D., & Kohli, P. J. (2017). Cost estimation model for web applications using agile software development methodology. *Pertanika Journal of Science and Technology*, *25*(3), 931–938. http://www.pertanika.upm.edu.my/index%20-%20JST.htm

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems & Software*, *86*(6), 1663–1678. https://doi.org/10.1016/j.jss.2013.02.027

Starcher, R. L., Dzubinski, L. M., & Sanchez, J. N. (2018). Rigorous missiological research using qualitative inquiry. *Missiology: An International Review*, *46*(1), 50–66. https://doi.org/10.1177/0091829617741911

Stoica, M., Ghilic-Micu, B., Mircea, M., & Uscatu, C. (2016). Analyzing agile development - from waterfall style to scrumban. *Informatica Economica*, *20*(4/2016), 15–25. https://doi.org/10.12948/issn14531305/20.4.2016.02

Strasser, A. (2017). Delphi method variants in information systems research: Taxonomy development and application. *Electronic Journal of Business Research Methods*, *15*(2), 120–133. http://www.ejbrm.com/index.htm

Sudarmaningtyas, P., & Mohamed, R. (2021). A Review Article on Software Effort Estimation in Agile Methodology. *Pertanika Journal of Science & Technology*, *29*(2), 837–861. https://doi.org/10.47836/pjst.29.2.08

Surucu, L., & Maslakci, A. (2020). Validity and reliability in quantitative research.

Business & Management Studies: An International Journal (BMIJ), 8(3), 2694–2726. http://dx.doi.org/10.15295/bmij.v8i3.1540

Taguchi, N. (2018). Description and explanation of pragmatic development: Quantitative, qualitative, and mixed methods research. *System*, *75*, 23–32. https://doi.org/10.1016/j.system.2018.03.010

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

Tanveer, B. (2017). Guidelines for utilizing change impact analysis when estimating effort in agile software development. *In ACM (pp. 252–257).* https://doi.org/10.1145/3084226.3084284

Taylor, K. J. (2016). Adopting agile software development: the project manager experience. *Information Technology and People*, *29*(4), 670–687. https://doi.org/10.1108/ITP-02-2014-0031

Torrecilla-Salinas, C. J., Sedeño, J., Escalona, M. J., & Mejías, M. (2015). Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology*, *61*, 124–144. https://doi.org/10.1016/j.infsof.2015.01.006

Tsoy, M., & Staples, D. S. (2021). What Are the Critical Success Factors for Agile Analytics Projects? *Information Systems Management*, *38*(4), 324–341. https://doi.org/10.1080/10580530.2020.1818899

Usman, M., Börstler, J., & Petersen, K. (2017). An effort estimation taxonomy for agile software development. *International Journal of Software Engineering and Knowledge Engineering*, *27*(04), 641–674. https://doi.org/10.1142/S0218194017500243

Vallon, R., José, B., Prikladnicki, R., & Grechenig, T. (2018). Systematic literature review on agile practices in global software development. *Information and Software Technology*, *96*(April 2017), 161–180. https://doi.org/10.1016/j.infsof.2017.12.004

Venkatesh, V., Brown, S. A., & Sullivan, Y. W. (2016). Guidelines for conducting mixed-methods research: An extension and illustration. *Journal of the Association for Information Systems*, *17*(7), 435–494. https://aisel.aisnet.org/jais/

Vyas, M., Bohra, A., Lamba, C. S., & Vyas, A. (2018). A Review on Software Cost and Effort Estimation Techniques for Agile Development Process. *International Journal of Recent Research Aspects*, *5*(1), 1–5. https://www.ijrra.net/

Yang, C., Liang, P., & Avgeriou, P. (2018). Assumptions and their management in software development: A systematic mapping study. *Information and Software Technology*, *94*, 82–110. https://doi.org/10.1016/j.infsof.2017.10.003

Yilmaz, K. (2013). Comparison of quantitative and qualitative research traditions: epistemological, theoretical and methodological differences. *European Journal of Education*, *48*(2), 311–325. https://doi.org/doi:10.1111/ejed.12014

Yoshigami, K., Tsunoda, M., Yamada, Y., & Kusumoto, S. (2017). Should function point

elements be used to build prediction models? *Proceedings - 8th IEEE International Workshop on Empirical Software Engineering in Practice, IWESEP 2017*, 41–46. https://doi.org/10.1109/IWESEP.2017.18

Zahraoui, H., & Idrissi, M. A. J. (2015). Adjusting story points calculation in scrum effort & time estimation. *2015 10th International Conference on Intelligent Systems: Theories and Applications*, SITA 2015. https://doi.org/10.1109/SITA.2015.7358400

Zare, F., Zare, H., & Fallahnezhad, M. S. (2016). Software effort estimation based on the optimal Bayesian belief network. *Applied Soft Computing Journal*, *49*, 968–980. https://doi.org/10.1016/j.asoc.2016.08.004

Appendix A: Research Participation Request

A study on how software development team size, team experience, and level of effort estimation accuracy affects agile software development project outcomes.

Hello John Doe,

I am a doctoral student in Information Technology at Walden University, and I am requesting your participation in an **anonymous** online survey to complete my study. If you had a role in software development projects using the agile methodology, I ask that you participate. The survey has 40 questions and will take about 20 minutes to complete.

I am seeking about 100 participants, and will share my findings in the LinkedIn group Program Manager, Senior Project Manager IT - Worldwide once my study is complete. I have attached the consent form for your reference. Please click on the SurveyMonkey link below to participate:

Survey link:

Thank you for your consideration!

Appendix B: Structure of the Survey Instrument

Variable	Item	Question
Software Development Team Experience		
	SDTE1	The selected project team members had high technical competence and expertise (problem-solving, programming, subject matter).
	SDTE2	Project team members had great motivation and were committed to the project's success.
	SDTE3	Project team was knowledgeable in agile principles and processes.
	SDTE5	The project team worked in a coherent, self-organizing teamwork manner, i.e., relying on the collective ability of an autonomous team to solve problems and adapt to changing conditions
	SDTE6	The project team had a good mixture of non-experienced (less than 5 years) developers and more senior (5 years or more) developers.
Software Development Team Size		
	SDTS1	The development team size was adequate for the project.
	SDTS2	Increasing the development team size would have improved output quality.
	SDTS3	Increasing the development team size would improve the level of effort estimation accuracy.
	SDTS4	The project had a small team size (3 to 20 members).
	SDTS5	The project had no multiple independent teams working together.
Level of effort estimation accuracy		
	LEEA1	The project imposed a well-defined coding standard up front.
	LEEA2	The project pursued simple design, e.g., programmers used the most straightforward possible design for each module to avoid waste and to facilitate cooperative work.
	LEEA3	The project pursued vigorous refactoring activities to ensure the results are optimal

		and to accommodate well all changes in requirements.
	LEEA4	The project delivered working software regularly within short periods.
	LEEA5	The project team's level of effort estimation per iteration was appropriate.
	LEEA6	The project team's level of effort estimation was accurate.
Perception of Success of the Agile project		
	PSAP1	The project was successful in terms of the quality of the project outcome or of the resulting software product.
	PSAP2	The project was successful in terms of scope and requirements of the project being met.
	PSAP3	The project was successful in terms of timeliness of project completion.
	PSAP4	The project was successful in terms of costs and efforts being under budget or within estimates.

Appendix C: Permission to Reuse Survey Instrument

From: Cao, Dac
Date: Friday, November 11, 2022 at 5:39 PM
To: Christopher Fry
Cc: Cao, Dac
Subject: RE: Survey Instrument

Dear Mr. Fry,

Attached is the requested questionnaire for your reference.
Best wishes for your educational endeavors.

Sincerely,

Dac-Buu Cao, PhD

Director, Software Product Support

www.sw.siemens.com

SIEMENS
Ingenuity for life

From: Christopher Fry
Sent: Friday, November 11, 2022 12:33 PM
To: Cao, Dac (DI IT SW SE PS)
Subject: Survey Instrument

Dr. Cao,

My name is Christopher J. Fry, and I am a doctoral student at Walden University. I am proposing to perform a quantitative correlational study between software development team size, experience, accuracy of effort estimation, and project success rates. I would like to use your survey instrument to obtain data from my sample population. I will be surveying software project managers in the South Texas, USA region.

I appreciate your consideration and time.

Respectfully,
Chris Fry

Appendix D: SPSS Output

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.315[a]	.099	.052	5.96598

a. Predictors: (Constant), SDTS, SDTS, LEEA
b. Dependent Variable: PSR

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	223.534	3	74.511	2.093	.111[b]
	Residual	2028.794	57	35.593		
	Total	2252.328	60			

a. Dependent Variable: PSR
b. Predictors: (Constant), SDTS, SDTE, LEEA

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
		B	Std. Error	Beta	t	Sig.	Lower Bound	Upper Bound
1	(Constant)	23.639	6.739		3.508	<.001	10.144	37.134
	SDTE	.537	.561	.121	.956	.343	-.587	1.660
	LEEA	.683	.583	.149	1.171	.247	-.485	1.851
	SDTS	-1.563	.819	-.244	-1.909	.061	-3.204	.077

a. Dependent Variable: PSR

Bootstrap for Coefficients

Model		B	Bootstrap[a] Bias	Std. Error	Sig. (2-tailed)	BCa 95% Confidence Interval Lower	Upper
1	(Constant)	23.639	.426	7.316	.002	10.751	39.624
	SDTE	.537	.057	.581	.364	-.673	1.918
	LEEA	.683	-.013	.624	.286	-.646	1.934
	SDTS	-1.563	-.101	.939	.093	-3.537	-.156

a. Unless otherwise noted, bootstrap results are based on 2000 bootstrap samples