SUCCESS OF EFFORT ESTIMATION STRATEGIES AND PRACTICES OF DEVOPS BASED SOFTWARE DEVELOPMENT IN SRI LANKA

Thennakoon Mudhiyanselage Harshanika Thennakoon

189126L

Degree of Master of Business Administration in Information Technology

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

August 2021

SUCCESS OF EFFORT ESTIMATION STRATEGIES AND PRACTICES OF DEVOPS BASED SOFTWARE DEVELOPMENT IN SRI LANKA

Thennakoon Mudhiyanselage Harshanika Thennakoon

189126L

The dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfilment of the requirement for the Degree of Master of Business Administration in Information Technology.

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

August 2021

DECLARATION

I declare that this is my work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

……………………………….

T.M Harshanika Thennakoon Date:

(Signature of the candidate)

The above candidate has researched for the Master's thesis under my supervision.

……………………………….

Dr. Dulani Meedeniya Date:

Signature of the Supervisor

COPYRIGHT STATEMENT

ABSTRACT

Effort estimation of software development is one of the most crucial things in Software Engineering. The effort estimations are conducted in the initial stages of the project management. These estimations help the customers, investors, managers, and software developers to recognize the total investment, financial plan, project schedule, and resources necessities. The process used to estimate the efforts differs from organization to organization. Several aspects need to be thought by the software developers when estimating the efforts. Although there were many studies related to this area, this study explores factors that help to succeed in effort estimation in DevOps-based software development in the Sri Lankan context.

According to the literature, the software development process would be easy in the DevOps context. Hence, the effort estimation would be easy when works on DevOps-based software development also. This study has chosen nineteen factors, by thoroughly reviewing the past literature and studies. The preliminary interview was done for identifying the most important factors. Poor communication, measurements, monitoring, technology stack, skills & experience, knowledge sharing, and deployment process were identified as the most important factors. The followed research methodology was the quantitative approach for this research study. Hence the online survey was conveyed among the 450+ software experts who have work experience in DevOps-based software development in the Sri Lanka IT industry. There were 41 questions with one open-ended question on the questioner. The reliability analysis was led to check the stability and validity of the questions before distributing the questionnaire. Descriptive analysis was used to elaborate on the elementary characteristic of the research data. The Pearson coefficient correlation was used for statistical data analysis and hypothesis testing. Moreover, the regression analysis was done to test the robustness of the connection between the predictor (independent) and predicted (dependent) variables in this research study.

As the results of the exploration, communication, and technology stack are highly impacted to the success of effort estimation in DevOps-based software development while knowledge sharing, skills & experience, and deployment process moderately impact the dependent variable. Measurement and monitoring were low impacts on the success of effort estimation in DevOps-based software development. Additionally, the set of practices and guidelines was suggested to follow when doing the effort estimation in the DevOps context. By following those practices and guidelines, software professionals can easily estimate the efforts and those efforts may be more confidently. It would be highly advantageous when achieving project deadlines and customer requirements easily.

Keywords: Effort Estimation practices, DevOps

ACKNOWLEDGEMENT

Premier, I might want to offer my genuine thanks to my supervisor Dr D. Meedeniya for eternal assist me in completing my research study and the thesis on " Success of Effort Estimation Strategies and Practices of DevOps based Software Development in Sri Lanka ", for her understanding, inspiration, excitement, and enormous information. Her direction encouraged me all the time in the exploration and composing of this thesis.

Other than my supervisor, I might want to thank our Project coordinator Dr Kutila Gunasekera, for their encouragement, insightful comments, immense support, and guidance over the research to make this become success.

Further, I wish to pass on my unprecedented appreciation to the software experts who caused me by rounding out the online review. Without their information, the exploration couldn't have been finished. Their participation by noting a protracted survey is valued.

I wish to appreciate the help and extraordinary love of my family including the husband’s family. They always head up to me on completing this research study. This work would not have been possible without their input. Then I hope to offer my appreciation to all who encouraged me by one single word in the success of this research study.

Last yet not least, I might want to thank my parents, for bringing forth me in any case and supporting me profoundly for the duration of my life.

TABLE OF CONTENTS

1. INTRODUCTION 1
1.1. Background 1
1.1.1. Motivation 3
1.1.2. Research Scope 4
1.2. Problem Statement 4
1.2.1. Research Objectives 5
1.2.2. Research Significance 5
1.2.3. Outline 6
2. LITERATURE REVIEW 7
2.1. Chapter Introduction 7
2.2. Overview to DevOps based Software Development 7
2.2.1. DevOps Process 8
2.2.2. DevOps Practice in Sri Lankan Software Industry 11
2.3. Effort Estimation in DevOps practice 12
2.3.1. Effort Estimation Strategies 12
2.3.2. Effort Estimation Practices 13
2.4. Factors affecting the success of effort estimation in DevOps practice 15
2.4.1. Collaboration Aspects 15
2.4.2. Monitoring and Automation 15
2.4.3. Measurements in DevOps 16
2.4.4. Organization Culture 17
2.4.5. Socio-Technical Aspects 17
2.4.6. External Pressure 18
2.5. Related Works 18
2.6. Testing and Analysis Methods 19
2.6.1. Cronbach Alpha Calculation 19
2.6.2. Descriptive Analysis 20
2.6.3. Inferential Analysis 21
2.6.4. Hypothesis Testing Methods 21
2.6.5. Pearson Coefficient Correlation 22

2.6.6. Linear Regression 23
2.6.6.1. Model Summary Table 24
2.6.6.2. ANOVA Table 24
2.6.6.3. Coefficient Table 24
2.7. Discussion 25
2.7.1. Summary of the comparison of related work 25
2.7.2. Literature related to the factors 26
2.8. Conclusion 28
3. RESEARCH METHODOLOGY 29
3.1. Chapter Introduction 29
3.2. Research Approach 29
3.3. Data Collection Method 32
3.3.1. Primary Data 32
3.3.2. Secondary Data 32
3.4. Sample Design 33
3.4.1. Population 33
3.4.2. Sample Size 34
3.4.3. Research Instruments 36
3.5. Conceptual Model/Design 37
3.6. Hypothesis 45
3.7. Pilot Survey 47
3.7.1. Reliability Analysis 47
3.8. Questionnaire 48
3.9. Conclusion 51
4. DATA ANALYSIS 52
4.1. Chapter Introduction 52
4.2. Data Collection 52
4.3. Reliability Analysis 53
4.3.1. Poor Communication 54
4.3.2. Measurements 54
4.3.3. Monitoring 55
4.3.4. Tech Stack 55
4.3.5. Knowledge Sharing 56

4.3.6. Skills and Experience 56
4.3.7. Deployment Process 57
4.3.8. Effort Estimation Strategies and Practices in DevOps based software development 58
4.4. Descriptive Statistic of Demographic Data 59
4.4.1. Analysis of Demographic Data using Charts 59
4.4.2. Statistical Analysis of Demographic Data Using Pearson Correlation 63
4.4.2.1. Analyzing the relationship between Gender and Success of EE Strategies and Practices in a DevOps environment. 63
4.4.2.2. Analyzing the relationship between Working Track and Success of EE Strategies and Practices in a DevOps environment. 64
4.4.2.3. Analyzing the relationship between Work Experience and the Success of EE Strategies and Practices in a DevOps environment. 65
4.4.2.4. Analyzing the relationship between Education Level and the Success of EE Strategies and Practices in a DevOps environment. 65
4.4.2.5. Analyzing the relationship between No. of Employees and Success of EE Strategies and Practices in a DevOps environment. 66
4.4.2.6. Analyzing the relationship between Following EE Strategies and Success of EE Strategies and Practices in a DevOps environment. 67
4.4.2.7. Analyzing the relationship between Category of Organization and Success of EE Strategies and Practices in a DevOps environment. 68
4.4.2.8. Analyzing the relationship between Opinion on EE Strategies and the Success of EE Strategies and Practices in a DevOps environment. 68
4.5. Inferential Analysis 69
4.5.1. Correlation Analysis of Variables for entire sample Data 69
4.5.2. Hypothesis Testing 70
4.5.2.1. Poor Communication and the Success of EE strategies and Practices in DevOps 71
4.5.2.2. Measurements and the Success of EE strategies and Practices in DevOps 72
4.5.2.3. Monitoring and the Success of EE strategies and Practices in DevOps 73
4.5.2.4. Technology Stack and the Success of EE strategies and Practices in DevOps 74
4.5.2.5. Knowledge Sharing and the Success of EE strategies and Practices in DevOps 75

4.5.2.6. Skills & Experience and the Success of EE strategies and Practices in DevOps 76

4.5.2.7. Deployment Process and Success of EE strategies and Practices in DevOps 77

4.6. Regression Analysis 79

4.6.1. Independent Variable – Poor communication 79

4.6.2. Independent Variable – Measurements 81

4.6.3. Independent Variable – Monitoring 83

4.6.4. Independent Variable – Technology Stack 84

4.6.5. Independent Variable – Knowledge Sharing 86

4.6.6. Independent Variable – Skills and Experience 88

4.6.7. Independent Variable – Deployment Process 90

4.7. Conclusion 91

5. RECOMMENDATIONS AND CONCLUSION 93

5.1. Chapter Introduction 93

5.1.1. Research End Results based on Demographic analysis 93

5.2. Consequences of the research 95

5.2.1. Hypothesis Testing: Poor Communication 96

5.2.2. Hypothesis Testing: Measurements 96

5.2.3. Hypothesis Testing: Monitoring 97

5.2.4. Hypothesis Testing: Technology Stack 97

5.2.5. Hypothesis Testing: Knowledge Sharing 98

5.2.6. Hypothesis Testing: Skills and Experience 99

5.2.7. Hypothesis Testing: Deployment Process 99

5.3. Factor Analysis Output 100

5.4. Recommendations 101

5.4.1. Establishing sufficient time on communicating effectively at the effort estimation stage 101

5.4.2. Encouraging to maintain the measurements and matrices at the Effort Estimation. 103

5.4.3. Introducing the monitoring tools and guidelines. 104

5.4.4. Enabling tech-supportive environment. 105

5.4.5. Establish Knowledge Sharing and a fully skilled platform. 106

5.4.6. Automating the repetitive activities in the application 107

5.5. Research Limitations 108
5.6. Future Directions in the research 109
5.7. Conclusion 110
REFERENCES 111
APPENDIX 121

LIST OF FIGURES

Figure 1: The DevOps Process 8
Figure 2: Coefficient for Poor Communication and EE Strategies 24
Figure 3: Research Approach 30
Figure 4: Summarization of the Preliminary Interview result 37
Figure 5: Conceptual Framework 43
Figure 6: Responses count over time 53
Figure 7: The Reliability Analysis of Poor Communication 54
Figure 8: The Reliability Analysis of Measurements 54
Figure 9: The Reliability Analysis of Monitoring 55
Figure 10: The Reliability Analysis of Tech Stack 55
Figure 11: The Reliability Analysis of Knowledge Sharing 56
Figure 12:The Reliability Analysis of Skills and Experience 56
Figure 13: The Reliability Analysis of Deployment Process 57
Figure 14: Reliability Analysis of Effort Estimation Strategies and Practices in a DevOps environment. 58
Figure 15: Reliability statistics using Cronbach's alpha 59
Figure 16: Gender Distribution 60
Figure 17: Distribution of Working Track 60
Figure 18: Distribution of Work Experience 60
Figure 19: Distribution of Education Level 60
Figure 20: No. of Employees Distribution 61
Figure 21: Distribution of Org. Category 61
Figure 22: Following EE Strategies & Practices 62
Figure 23: Opinion on EE Strategies & Practices 62
Figure 24: Pearson correlation between gender and Dependent variable 64
Figure 25: Pearson correlation between working track and Dependent variable 64
Figure 26: Pearson correlation between work experience and Dependent variable 65
Figure 27: Pearson correlation between education level and Dependent variable 66
Figure 28: Pearson correlation between no. of employees in EE strategies and Dependent variable 66
Figure 29: Pearson correlation between following in EE strategies and Dependent variable 67
Figure 30: Pearson correlation between following in EE strategies and Dependent variable 68
Figure 31: Pearson correlation between following in EE strategies and Dependent variable 69
Figure 32: Pearson correlation between all the Independent variables and Dependent variable 70
Figure 33: Correlation analysis of Poor communication and EE Strategies and Practices 71
Figure 34: Correlation analysis of Measurement and EE Strategies and Practices 72
Figure 35: Correlation analysis of Monitoring and EE Strategies and Practices 73

Figure 36: Correlation analysis of Technology Stack and EE Strategies and Practices 74
Figure 37: Correlation analysis of Knowledge Sharing and EE Strategies and Practices 75
Figure 38: Correlation analysis of Skills and Experience and EE Strategies and Practices 76
Figure 39: Correlation analysis of Deployment Process and EE Strategies and Practices 77
Figure 40: Model Summary of Poor Communication........ 79
Figure 41: ANOVA Table of Poor Communication........ 80
Figure 42: Coefficient Table for Poor Communication........ 81
Figure 43: Model Summary Table for Measurement........ 81
Figure 44: ANOVA Table for Measurement........ 82
Figure 45: Coefficient Table for Measurement........ 82
Figure 46: Model Summary Table for Monitoring........ 83
Figure 47: ANOVA Table of Monitoring........ 83
Figure 48: Coefficient Table for Monitoring........ 84
Figure 49: Model Summary Table for Technology Stack........ 85
Figure 50: ANOVA Table of Technology Stack........ 85
Figure 51: Coefficient Table for Technology Stack........ 86
Figure 52: Model Summary for Knowledge Sharing........ 86
Figure 53: ANOVA Table for Knowledge Sharing........ 87
Figure 54: Coefficient Table for Knowledge Sharing........ 88
Figure 55: Model Summary for Skills and Experience........ 88
Figure 56: ANOVA Table for Skills and Experience........ 89
Figure 57: Coefficient Table for Skills and Experience........ 90
Figure 58: Model Summary Table for Deployment Process........ 90
Figure 59: ANOVA Table for Deployment Process........ 91
Figure 60: Coefficients Table for Deployment Process........ 91
Figure 61: Summary of Factor Analysis........ 101

LIST OF TABLES

Table 1: Rule of Thumb for Results 20
Table 2: Standards of Pearson correlation coefficient 23
Table 3: Summary of the comparison of related works 25
Table 4: Grouped table of the factor table in related works 26
Table 5: Literature Related to Factors 27
Table 6: Main factors and subfactors with the literature 42
Table 7: Detailed structure of the factor related questions 49
Table 8: Summary of Reliability Analysis in the survey questions 58
Table 9: Summary of the hypothesis testing results 78

LIST OF ABBREVIATIONS

SVR - Support Vector Regression

ANOVA - Analysis of Variance

ANCOVA - Analysis of Covariance,

GLM - General Linear Model

CAMS - Culture, Automation, Measurements, and Sharing

CALMS - Culture, Automation, Measurements, Sharing and Lean

CAMM - Culture, Automation, Measurement, Monitoring

SCM - Software Configuration Management

RS - Requirement Specification

DOKS – DevOps Knowledge Sharing

SPSS – Statistical Package for Social Science

EE - Effort Estimation

1. INTRODUCTION

1.1. Background

The software industry can be considered one of the fastest-growing industries in the present world. The growing urge of the society to ease the routine work-life by using novel software has majorly resulted in this revolution, simultaneously creating a rapid increase in demand for software development projects. However, the vital consideration revolves around ‘delivering software development projects on time’, to ensure client satisfaction. Prompt delivery might also be considered the root cause for either survival or cancellation of the many projects. This is the situation where "effort estimation" comes into discussion.

Development effort is viewed as one of the significant segments of software costs, especially for worldwide development, and it is typically hard to anticipate (El Bajta, 2015). It is said that effort estimation can be considered the most important part of the software development life cycle. Because the estimated effort could highly affect the increase in the project cost. Therefore, when estimating the timeline, we need to follow a proper process. Effort Estimation in programming is one of the most significant programming forms. Unfortunately, low evaluation precision may give terrible outcomes (Phannachitta, 2018).

Development and Operations are the two main components of DevOps. Operation is a new circumstance that facilitates continuous deployment in the software development process. It has brought new opportunities to software companies (Lwakatare, Kuvaja, et al., 2016). Hence, DevOps-based software development has become new normal to the IT industry, we have to consider several factors from both perspectives (development and operation) when doing the effort estimation.

Several estimation methods that are supportive for project managers when estimating the software development timeline are being proposed during recent years. A new approach has been stepped by (El Bajta, 2015) which enhances the performance and costs. This approach portrays an improved relationship-based model. According to (El Bajta, 2015), that model could be helpful for primitive phases of the stage of effort estimation for distributed projects. But in practice, there are several challenges in effort estimation in DevOps-based environments due to the abrupt changes requested by the clients on initial requirements or the enhancements amid the development phase. Also, there are several workloads involved with deployment on the server in DevOps-based software development. Contrarily, many server-related issues could occur as well. Not only that, the technology gaps, unplanned tasks, and identified issues may also affect the success of effort estimation. When unplanned tasks and issues come into the ongoing development cycle, it may affect the project delivery time. In such an instance, we need to hold the development of a particular task that has less priority and less risk. Upon that, the developer will be assigned to the unplanned task or issue. This existing process may cause to reduce the performance of the employees as well.

Most of the models require project size estimation for their calculations, which is mainly difficult (Sabrjoo et al., 2015). According to the author, the size of the project could be considered as the line numbers in the source code. Even without starting the project, the number of code lines could not be determined at the initial stage. As per the author, this may be a reason for the failure of certain effort estimation models.

Some other techniques can be taken into consideration as well. A new technique based on Support Vector Regression (SVR) for effort estimation of analogy-based software development is a fine example of such techniques. This has been proposed by (Ali Idri et al., 2018). Also, data sets from different observations are being used for proposing this technique.

In this context, there should be a proper guideline to follow when effort estimating. This research proposes and evaluates best practices that can be followed at the phase of estimating the software development timeline. Moreover, this study indicates a new effort estimation process by analyzing the factors which can be affected to the success of the effort estimation.

1.1.1. **Motivation**

On-time delivery of the software project, promising ideas and strictly adhering to the budget are vital components to attain a competitive advantage over the software development project. Because software projects are difficult endeavors (Rahikkala et al., 2015). However, there may be several issues in a practical scenario. The first being, some unplanned tasks, and issues may arise in the development cycle. Sometimes these may depend on existing ongoing tasks. Then, it should be included in repairing any problem that arose during the development when estimating the effort. At the same time, we have to consider the accuracy of effort estimation of the project time. The accuracy in the effort estimation is basic for compelling administration and control in software development projects (Ali Idri et al., 2018). However, software projects still suffer from inaccurate estimates. At the same time, the need to consider the success of effort estimation of the project time is important.

According to (Phannachitta, 2018), it is likely that software application cases are quite similar incidentally (e.g., size of the software). However, they may unintentionally diverge in terms of diverse development endeavors. This results in the theoretical question of; whether the software project, which has the same characteristics should have the same effort estimation? There may be some other reasons for this as well. Such as, different technology, and frameworks that may have been used. Using different frameworks, implementations, and project architecture may not be the same as previous. Then it may lead to consumption of time. Therefore, the estimators cannot predict the exact project timeline at the initial stage itself. However, if a guideline or procedure for estimating the project timeline is in their possession, it could be considered a value-addition.

Without having a proper guideline for estimators, they may give a rough estimation for a particular task. This could ultimately lead to the propensity of missing out on certain affecting factors to the overall timeline. The procedure of "Best Practices", will be proposed in this study with the intent that implementing such a procedure could lead to a reduction in wastage of resources, avoid schedule/budget overruns, and/or quality compromises. This may also result in on-time delivery of the project, protecting client confidentiality, helping to increase the performance of task developers, and reduce wasting time on low-priority tasks as well.

1.1.2. **Research Scope**

This study will concentrate on distinguishing the components influencing the success of effort estimation in DevOps-based software development projects.

Since the expected population and the examples are chosen from the Sri Lankan IT industry, the extent of this exploration is constrained to the Sri Lankan context. This questionnaire was disseminated distinctly among the IT specialists of Sri Lanka.

1.2. Problem Statement

(El Bajta, 2015) has mentioned the challenges and aspects that influence the effort estimation approach. Further, the effort estimation techniques estimate a comparatively less amount of time to finish a given project. Therefore, the effort estimation process should be improved. Moreover, two categories of effort estimation methods have been mentioned in this research study. Firstly, the mathematically-based category, and secondly, the experience-based category. Experience-based categories provide the direction needed to play out the effort estimation process.

In this context, there are several issues identified in the existing effort estimation methods. Overestimation and underestimation may negatively affect the software performance. The underestimation potentially brings about enormous bad results to the schedule or budget. This could ultimately lead to the worst outcome of project cancellation (Phannachitta, 2018). It is identified that there is an emerging problem of not having a proper estimation process in DevOps-based software development. Hence, this study tries to introduce a proper guideline (Best Practices) that could be used when conducting effort estimation.

1.2.1. **Research Objectives**

The objectives of this research are to;

- Explore the factors which need to be considered for the success of the effort estimation phase of a project in a DevOps environment.
- Analyze the existing techniques which are currently used in the effort estimation phase in DevOps practices.
- Explore the drawbacks of existing effort estimation techniques, and;
- Recommend a useful effort estimation guideline ("Best Practices") for a more reliable forecast of the software project development timeline in DevOps based environment.

1.2.2. **Research Significance**

This study is focused on identifying the factors that need to be considered when estimating effort in DevOps-based software development and explore the drawbacks of existing effort estimation techniques. Due to the poor effort estimation, a software project will not be able to deliver with high quality and within the scheduled time. There is also a tendency for budget overruns to occur. Besides, customer satisfaction is a key component in this industry. Therefore, the driving need to protect the level of satisfaction of the customers is essential. Hence, the outcome of this research will serve as assistance to ensure customer satisfaction.

We will be able to recognize the factors that need to be considered in the effort estimation phase on DevOps-based software development, in the latter part of this research. Upon recognizing those factors, the necessity to identify which factors can mostly stimulate the effort estimation arises. Finally, new estimation strategies and techniques can be introduced.

1.2.3. **Outline**

The background of the study, the motive of conducting this research, the research problem, objectives and significance of the study are elaborated at the beginning of this section, later on, this research will elaborate on the identification of the factors which need to be considered at the effort estimation phase in DevOps software projects.

The first chapter portrays the aims that are wished to accomplish in the research and the extension that is covered.

2. LITERATURE REVIEW

2.1. Chapter Introduction

In this chapter, the researcher describes the background of the study, and what other researchers have discovered on the effort estimation practices and strategies in a DevOps-based environment. Journal articles, research papers, books, and web sources have been thoroughly evaluated and reviewed by the researcher for this chapter. The factors which are affected the success of effort estimation strategies and practices have been analyzed by the researcher in this chapter. Additionally, the analysis methods which were used to statistically analyze the survey results are included here.

2.2. Overview to DevOps based Software Development

As (Rubasinghe et al., 2018b) mentioned, different software development practices are used widely. Among them, DevOps is the most reliable software development practice as it ensures quality deliveries, adaptability to frequent changes, and staying within the budget**.** There is a scientific definition that is defined by (Penners & Dyck, 2015). It is considered as a mindset that encourages working together between the teams' dev and IT. Also, it runs flexible systems and speeding up continuous deliveries. It allows working collaboratively among Dev and Ops teams. DevOps gives fast and reliable deployment in superior quality (Nicolau de França et al., 2016; Rubasinghe et al., 2017; Rubasinghe et al., 2018a).

The primary objective of DevOps is to recognize and eradicate the space between Dev and Ops teams (Bucena & Kirikova, 2017). The identified main goals in DevOps are: Protecting business esteem, Breakdown obstructions among development and operations by empowering trust and shared possession and Supporting the development (Perera, Silva, et al., 2017).

2.2.1. **DevOps Process**

It is an approach used in the software development process. This is about speedy, and adaptable development (Hemon, Fitzgerald, et al., 2020). DevOps process is inclusive of several phases. Those phases are; Development, Integration, Testing, Deployment, and Monitoring (Palihawadana et al., 2017; Meedeniya et al., 2019a; Rubasinghe et al., 2020). Carrying out the above phases continuously is vital in DevOps Process. Figure 1 depicts the process, which helps a team to diminish issues and solve problems (Ebert et al., 2016).

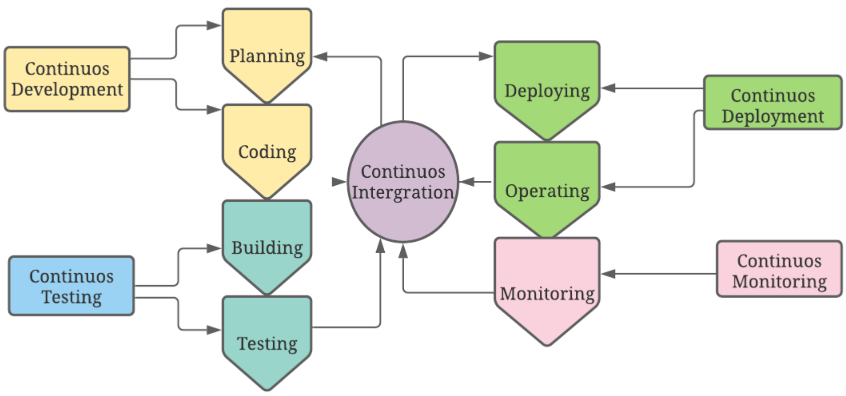

Figure 1: The DevOps Process

- **Continuous Development**

In this phase, the team is involved in two main actions. Those are; 'planning the project', and 'coding the application'. The project vision is declared at the planning stage. Then the software engineers (developers) start coding on the application (Meedeniya et al., 2020). Any language can be used to code and the coding maintains are handle by the Version Control Tools. It is referred to as "Source Code Management" and there are various tools such as Git, SVN, CVS, and JIRA which are used for it. Also, there are several tools such as Gradle, Maven, and Ant for building and packaging the code. Those are referred to as "Build Automation Tools". In this phase, the code is converted to executable files by the Build Automation Tools (Erich et al., 2017).

There are several benefits of continuous development, Delivering new features, quality products, risk avoidance and increased productivity are some of them (Laukkarinen et al., 2017).

- **Continuous Testing**

This is the stage where Quality Assurance engineers continuously test the application. Automating all the testing processes and configuring them to run after the deployment is referred to as Test Automation (Riungu-Kalliosaari et al., 2016; Rubasinghe et al., 2018a). Selenium and Junit are used as Test Automation Tools. There are several parts included under the continuous testing phase. They are:

a. Unit testing and Integration testing
b. Static and Security code analysis
c. Load and Performance testing

The major benefits derived from test automation are, it saves time, effort, and labour. Hence, it executes test cases automatically instead of running them manually. Furthermore, test automation generates reports, which could be considered another major benefit. The execution time can be scheduled as preferred at a predefined time (Erich et al., 2017).

- **Continuous Integration**

This phase can be considered the heart of the DevOps process. All developers are mandated to commit their changes to source code frequently (Palihawadana et al., 2017). The existing codebase will be updated with new changes when they commit and push their changes (Rubasinghe et al., 2017). This can be done on daily basis. This helps in the early detection of the functionality breakdown, conflicts, and merging issues if they are present. This incorporates several important actions like; Code reviewing, component testing (unit testing), integration testing, and building packages. Once the code is built, it will be compiled (Hussain et al., 2017). As the code base continuously develops, it needs to be incessantly integrated with the system (Meedeniya et al., 2020; Rubasinghe et al., 2020). Then updated code and relevant changes will be reflected on the end-users. This should be done smoothly. Jenkin is one of the popular tools used in this phase. According to (Lwakatare, Karvonen, et al., 2016) study, he emphasizes the importance of continuous integration practices as primary action to the progress of the continuous deployment.

- **Continuous Deployment**

In this stage, the code is placed in the production environment. It is important to assure that the built code is deployed successfully into the production. Continuous Delivery is referred to as Continuous deployment by (Lwakatare, Kuvaja, et al., 2016). As mentioned (Meedeniya et al., 2019a), Continuous Deployment is a Continuous Delivery pipeline. Human involvement is unnecessary in DevOps-based software development until the code reaches the production environment (Artač et al., 2016). Once the code is deployed into the production environment, it will not be executed until the right time. This choice additionally permits turning on the component to a specific arrangement of end-users (Chung, 2017).

As (Lwakatare, Karvonen, et al., 2016) state, having continuous releases helps to identify the risks associated with the deployment and steer the quicker criticism concerning software application changes. Hence, continuous deployment gives companies occasion to quickly verify their features and study their usefulness to the customer.

- **Continuous Monitoring**

This is an extremely essential phase in the DevOps process where you persistently screen the health of the software application. The most important task is to keep the application running healthy (Pérez et al., 2015). This refers to the database connection performing well and being stable throughout the time. In this instance, the team ensures the performance of the entire application. Therefore, performance monitoring is most essential in DevOps (Lwakatare, Karvonen, et al., 2016). In practice, infrastructure teams blame the dev team and vice versa, when the issue is present due to the stability and performance. Hence, in the DevOps culture, issues are collaboratively resolved by sharing responsibilities among the teammates (Chung, 2017).

2.2.2. DevOps Practice in Sri Lankan Software Industry

Even as a lower-middle country, Sri Lanka transpires IT-BPM (Business Process Management) globally in many key focus domain areas. As A.T. Kearney mentioned, Sri Lanka is ranked among the Top 50s in the category of Global Outsourcing Destinations. Sri Lanka is also ranked among the Top 20s in Emerging Cities by Global Services Magazine (SLASSCOM, 2014). Currently, 300 over IT and BPM companies are operating in Sri Lanka. Most of them are small, medium and a few giant scale organizations in the global industry. The ICT and BPM sectors have become the fifth largest source of foreign income in Sri Lanka, despite few challenges in the industry (SLASSCOM, 2014). Due to the competitive growth in the IT industry, identifying the capacities and experiences are required for software development companies. Also, guiding them to recognize the suitable process that can be used for the software development, is necessary. If not, the software companies could not expect survival in the business.

The concept of DevOps emerged in the software development industry moderately late. Accordingly, Sri Lankan software companies tend to follow it as a practice (Meedeniya et al., 2020). In the Sri Lankan context, the developers and the operations team, work collaboratively to provide quality software, or services to the customers. They share the resources and obligations among the team members and increase the process performance (Perera, Bandara, et al., 2017).

According to (Meedeniya et al., 2019a), the mentioned five stages in DevOps practice are followed by the software professionals in the Sri Lankan IT Industry. Developers or development teams are responsible for planning, coding, building, and finally testing the software project, while the operations team takes the responsibility of releasing, deploying, operating, and monitoring the system. In Sri Lanka, the "DevOps Engineer" plays an important role in coordinating human resources in a DevOps environment. He can be a single person on a set of groups that manage the involving automation, version control, configuration, and system maintenance (Meedeniya et al., 2019b). The DevOps roles can differ from the organization's environment and its requirements. The “Release Manager” generally releases the features and check the post-release product stability in the system, and “Automation Expert” is responsible for automation, “Software Engineer” or “Tester” develops the code and tests it.

2.3. Effort Estimation in DevOps practice

Effort estimation is evaluating the most realistic effort that consumes to develop or maintain a particular task or part of the software application. This would be happening any time in the project. It is a vital factor for software development to be successful (Britto et al., 2014). When it comes to the DevOps context, the Dev and Operation team working efficiently for the continuous delivery of the software releases in the DevOps environment (Bucena & Kirikova, 2017). The main characteristic of estimating techniques in the DevOps practice is that it is not estimated the exact hours or time but in points (Hussaini, 2014). This gives the relative size of the activity. Instead of estimating the working hours, it gives that how many quantities as a previously identified user story.

It is essential to estimate the relative scale of the whole team for better estimation (Popli & Chauhan, 2014). Open communication between Dev and Ops team is needed to achieve this. Then learning curve is automatically generated by such an approach. The team members can learn from the previous estimation when making a new one (Owais & Ramakishore, 2017). Estimating in DevOps is completely different from estimating in other software development methodologies. Other methodologies try to make estimates detailed estimations of entire projects.

2.3.1. Effort Estimation Strategies

The existing methods, models, or techniques that are used to estimate the effort are defined as the “Effort Estimation Strategies” that are extensively discussed in this study by the researcher.

Effort estimation methods are divided into two, such as “Model-based” and “Expert-based” (Menzies et al., 2006). Summarizing old data and using some algorithms, the “Model-based” methods are made for a prediction about the new projects. “Expert-based” methods use human expertise to generate predictions.

For example, various estimation techniques are used to estimate the software development timeline in DevOps. Planing poker, T-shirt sizing, Swim laze sizing and Bucket sizing are a few of them (Britto et al., 2014). In Planning Poker, team members assign relative story points for the items. In that, all team members will be participating in a meeting, and use cards to represent the number of story points.

When discussing the T-shirt sizing, the items are grouped into T-shirt sizes instead of story points. Extra Small, Small, Medium, Large and Extra Large are used as the T-shirt sizes. The range of sizes used up to the team to decide. There are several regulations and limitations in those estimation techniques when used as it is, at the effort estimation phase (Fávero et al., 2018). That means when using those estimation techniques, we are stuck into its parameters such as scope, characteristics, initial velocity, and individuals expert opinion. But in practice, we generally adjust those techniques according to the usage, based on our organization and project requirements. Hence these theoretical aspects are not matched with our practical scenarios (Khan & Araghinejad, 2010). At that time, we have to do some modifications by following our resources, requirements, and capacity. In this study, the researcher discussed those adjustments or modifications (what to do) when practising estimation methods.

2.3.2. **Effort Estimation Practices**

There is a deviation between the theoretical aspects of effort estimation strategies and what practitioners do daily at the effort estimation phase (Britto et al., 2016). Hence, the adjustment practitioners have done as practices for those day-to-day activities are defined as the “Effort Estimation Practices” by the researcher in this study. The researcher expects to propose an effort estimation guideline after identifying those practices and exploring them.

According to (Menzies et al., 2006), there is a list of best practices under the expert-based method such as, evaluate the estimation accuracy and avoid pushing people on to evaluation, use documented data of the previous tasks, ask estimators to justify their estimation and find estimation experts from the relevant background to validate given estimation. Also, the researcher has defined best practices when using model-based estimation methods. Some of those methods could be considered as, Reuse of the regression parameters, local calibration, and variable subset selection. As stated by (Phannachitta, 2018), use estimation checklists, estimate top-down and base up the other way around autonomously of one another. As few of the best practices, 'top-down and base up the other way around autonomously of one another' combines estimates from different experts and separate strategies, and provide feedback for given estimation.

- Effort Estimation metrics

The researcher has used the term 'Effort estimation metrics" to explain some sort of technique, graph or diagram that is used to calculate the effort. These metrics are eventual and effective for evaluating estimation value, and they have a perfect effect on the results of estimation. We would reduce project risks, initial and primary works, system necessities, customer satisfaction by the usage of the metrics (Yavari et al., 2011). Metrics can be used to compare the values as well. It could be useful to determine the development strategy that is going to take place.

2.4. Factors affecting the success of effort estimation in DevOps practice

2.4.1. Collaboration Aspects

According to the authors (Perera, Silva, et al., 2017), DevOps is encouraged to work in collaboration and cooperation between Dev and Ops teams. It is an extended version of agile. In a collaboration environment, the operations staff and developers communicate to deliver high-quality applications and services continuously, reliably. by sharing the assigned tasks and responsibilities within the. Also, each of them is fully accountable for their services from development to deployment. (Bucena & Kirikova, 2017) has proven that the collaboration practices are one of the highest levels of priority practices (e.g. active participation of stakeholders, shared responsibilities, knowledge and information sharing practices, etc.).

As (Hussaini, 2014) says, the 4Cs (Communication, Co-operation, Culture, and Collaboration) need to be addressed when implementing DevOps. Because, existing gaps are leading to competition, team conflict, inordinate delays, and inefficiency of software systems. Collaboration is intended through sharing knowledge, widening of skill sets, and moving obligations among the team members just as imparting a feeling of collaboratively sharing their duties and obligations (Lwakatare, Kuvaja, et al., 2016). (Jones et al., 2016) has stated that management visualizes a DevOps approach, where both the software developers and system administration work in the close coordinated effort as a component of an incorporated entirety. This incorporation is substantive in the DevOps, which brings attention to a culture of joint effort through the harmonization of the product development and IT operations tasks.

2.4.2. Monitoring and Automation

Generally, screening is used during the functioning use of the application system (Asiamah et al., 2017). It serves as feedback to developers as it helps evaluate the standards of the product plan, and to recognize the areas which need to be refactoring (Debbiche, 2019). It is mandatory to log and trace each action when running the software system (Chen, 2019).

As (Ebert et al., 2016) mentioned, automation needs to be conducted to a certain extent, if we require high-quality software deliveries frequently. Two common practices are included in automation. Those are the deployment process automation and infrastructure-as-code (IaC) (Lwakatare, Kuvaja, et al., 2016). Automating the deployment process can be achieved automatically by equipment that assists with overseeing application and foundation setups. As (Hemon, Lyonnet, et al., 2020) state, automating all turn of events and task exercises however much as could reasonably be expected, is a major principle of DevOps. There, the highest attention will be paid to automating all processes related to the operational aspects such as deployment-related activities, dev testing, and paying attention to managing the configuration of applications (Jabbari et al., 2016).

2.4.3. Measurements in DevOps

Different metrics are used to monitor and assess the performance of processes in development and deployment. As an example, to convey the system quality, code quality, and system stability. According to (Lwakatare, Kuvaja, et al., 2016) DevOps encourage the use of common metrics that are often business-focused and give motivation to both Dev and Ops teams. According to (Nicolau de França et al., 2016) the measurement is a significant rule frequently started up by gathering proficient metrics. It helps decision-making in software engineering and its' operations.

According to (Perera, Silva, et al., 2017) the measurements are very important in DevOps environments. Hence, feedback for the improvements should be conducted frequently, across multiple dimensions and angles.

2.4.4. **Organization Culture**

As (Forsgren & Kersten, 2018) says, research shows that software delivery and organizational performance are driven by organizational cultures whilst company revenues are driven by job satisfaction. There needs to be a change in people's mindsets when working as a team. (Artač et al., 2016) has stated about the efficiency of DevOps. As they elaborate it, the series of software engineering strategies and tools promise to deliver software products with speed and quality simultaneously, without additional expenses. Therefore, it is a mix of organizational software culture which supports the DevOps strategies with proper tooling. As stated by (Perera, Bandara, et al., 2017), the culture is prepared as the background for interaction, accepting the level of risk, and ability to adapt.

(Leite et al., 2019) have defined the concept of "Culture of collaboration". It is based on several components such as knowledge sharing, appliances and tools, and procedures. As they have mentioned, the impediments to the continuous delivery acquisition are not the extent of expertise of each team member, but the unsuccessfulness of the top executives and at the administration level. High-performing organizations try to execute the tasks as efficiently as possible, regardless of the workers being fungible "assets" (Rong et al., 2017). Management is responsible for making the failures be accepted and employees seek to continuously improve (Jones et al., 2016).

2.4.5. **Socio-Technical Aspects**

Socio-Technical issues which arise from the social and technical aspects may lead to building lack of confidence among the team. Cultural differences, the necessity of teams filled with new skill sets, existing problems due to the geographic distribution of teams are a few of the issues (Nicolau de França et al., 2016). As (Hussaini, 2014) said, one of the fundamental issues rousing at the adoption of DevOps is, ineffective communication among stakeholders in the Dev and Ops teams.

2.4.6. External Pressure

As (Nicolau de França et al., 2016) state, in practical scenarios, it is required to respond quickly to the market conditions, hence the customers highly expect the quality and rapid business change, and also the demand for new features is relatively high.

2.5. Related Works

As per (El Bajta, 2015), there are two categories of effort estimation methods: “Mathematical based-category” and “Experience based-category”. The required amount of effort is constructed and represented on a mathematical basis in the “Mathematical based-category”. The information which needs to be performed for the effort estimation process is supplied by the “Experience based-category”, mainly based on experience. Effort estimate falls into 3 categories according to (Fávero et al., 2018): 1. human-centric; 2. model-based: and 3. produced by prediction techniques.

According to (El Bajta, 2015), they have developed the 5 step model of estimation framework. The proposed estimation model is conclusive of the below steps. Firstly, the effort estimations are obtained and saved into the database because they could be used later to improve the accuracy of future estimations. The accuracy of the estimation is needed for distributed projects, and the results can be used for future estimations. Secondly, the modifications should be conducted or remove a particular record for stored records. Finally, the mechanism should be updated by adding similarity measures. However, this approach helps estimators to achieve the estimated values in global software development. Although, the researcher has mentioned the inputs of the model are not exhaustive.

As (Menzies et al., 2006) state, they have found the effort modelling workbench called “COSEEKMO” and it assists in tracking down the best assessment technique for a specific area. The advantage of “COSEEKMO” is that all the analysis is done automatically and it investigates the different exertion assessment techniques. Contrastively, there are certain restrictions in “COSEEKMO”. Input data should be formatted as specified models (COCOMO 81 and COCOMO-II) is one of the restrictions.

According to (Sabrjoo et al., 2015), they have proposed a new effort estimation methodology called “Improvement of COCOMO || Model” by enhancing the general COCOMO model with the use of different cost drivers. The advantage of this model is that it provides better results than the COCOMO model. However, there are some limitations to this methodology as well. Structure of the model, project size, and changing the hypothesis over time are a few of them.

(A. Idri et al., 2002) have presented a new procedure called a fuzzy analogy approach that is grounded from analogies, fuzzy logic, and linguistic quantifiers (nominal data). Identifying the causes, retrieving similar cases, and adapting to the cases are the three steps of this procedure. Generally, the linguistic values are not considered in classical analogy procedures. However, this approach is developed by improving the classical analogy procedure. The main advantage of this approach is handling imprecision and uncertainties correctly.

2.6. Testing and Analysis Methods

Several testing and analysis methods are used by the researcher for the data analysis in this study. Those methods are described below in this section.

2.6.1. Cronbach Alpha Calculation

Lee Cronbach was the founder of Cronbach alpha in the year 1951. The internal consistency (reliability) of a set of test items is assessed by Cronbach alpha. “Reliability” refers to how well a test estimates what it should do. The internal consistency should be determined prior to a test. It ensures the validity of the research or examination. (Tavakol & Dennick, 2011). Cronbach's alpha formula is shown in Equation 1. Table 1 shows the strength of the internal consistency levels of alpha.

$$\alpha = \frac{N \cdot \bar{c}}{\bar{v} + (N-1) \cdot \bar{c}}$$

Where:

N = the number of items

$\bar{c}$ = average covariance between item-pairs

$\bar{v}$ = average variance

Equation 1: Cronbach Alpha Calculation

Table 1: Rule of Thumb for Results

Cronbach's Alpha	Internal Consistency
$\alpha \geq 0.9$	Excellent (High-Stakes testing)
$0.7 \leq \alpha < 0.9$	Good (Low-Stakes testing)
$0.6 \leq \alpha < 0.7$	Acceptable
$0.5 \leq \alpha < 0.6$	Poor
$\alpha < 0.5$	Unacceptable

The ‘α’ is responsive to the quantity in the test. If ‘α’ is high, the questions are highly correlated and if it is low, it means there are not enough questions in the test (Tavakol & Dennick, 2011).

2.6.2. **Descriptive Analysis**

Descriptive statistics summarize and organize principal characteristics of data in a meaningful way in the Descriptive analysis. For instance, the pattern might emerge from the data (Laerd, 2018). Those patterns are easy to visualize and understand even if there are large numbers of data. Descriptive statistics enables the interpretation of data through statistics such as the mode, median, mean, range, quartiles, absolute deviation, variance, and **standard deviation**. Tables, Graphs, and charts are used to represent the picturesque analysis of Descriptive statistics (Laerd, 2018).

2.6.3. **Inferential Analysis**

Inferential statistics help reach conclusions, make inferences, and judgments from data. Inferential statistics are useful for the experiment and evaluation of the outcome. ANOVA, ANCOVA, regression analysis, factor analysis, multidimensional scaling, cluster analysis, discriminant function analysis have come from the GLM. (Trochim, 2020).

2.6.4. **Hypothesis Testing Methods**

There are three main steps in the general hypothesis testing procedure in every research study. The researcher stated the hypothesis to be tested. As the first step, brief an analysis plan. In this step, briefly explain how the plan works. Then inspect the sample data. In this phase, studying the sample data has occurred. Finally, interpret the results. The results of the hypothesis testing are translated to meaningful implications which match with the research. Every hypothesis test is required to be stated in both ways: the null hypothesis (H_0) and the alternative hypothesis (H_1). (stattrek.com, 2019).

- T-test

In the statistical methods, the T-test is developed by William S. Gosset (Jankowski et al., 2018), also known as "Student's t-test" and "two-sample t-test", which is generally used to compare groups' mean (Al-achi, 2019). Inference about the sample can be generated by the t-test: the difference between the samples (Jankowski et al., 2018). The distribution of t is bell-shaped.

- ANOVA Test

R.A. Fisher developed Analysis Of Variance (ANOVA) in 1920 (Kaur, 2015). An ANOVA is a way to detect the differences between the groups of the experiment results. It helps to figure out the acceptance or rejection of the null hypothesis (Sawyer, 2009).

2.6.5. **Pearson Coefficient Correlation**

In statistical analysis, there are various names used to identify the Pearson correlation. Namely, the Pearson correlation coefficient, and Bivariate distribution of Pearson's Product Moment Correlation (Pearson's r). Equation 2 depicts the formula for the Pearson correlation coefficient.

When the values of the variables represent either the interval or ratios scales in measurements, this distribution can be considered appropriate for use. That can be compared to find the association between them (Obilor & Amadi, 2018). Core relation is involved when measuring the negative, positive, or non-relationship of the association of two variables. In other terms, it measures the strength of the relationship between two variables.

The coefficient of correlation has two extents: high and low. Also, It has two directions: positive and negative. It varies from minus one (-1) to positive one (+1). Whereas a negative end (-1) denotes a completely negative correlation, while a positive end (+1) denotes a completely positive correlation. 0 denotes there's no correlation. If the two variables have positive relationships, when one variable is increased positively, another one also should increase positively in a fixed amount. If there's a negative association, when one variable increases positively, another one decreases negatively in a fixed amount. Similarly, a 0 relationship indicates there's neither a negative nor positive relationship between two variables (Obilor & Amadi, 2018). Table 2 shows the Standards of the Pearson correlation coefficient. The relationship level of variables is decided with the use of Table 2.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2]\,[n(\sum y^2) - (\sum y)^2]}}$$

Where,

r = Pearson Correlation Coefficient

n = number of pairs of the stock

∑xy = sum of products of the paired stocks

∑x = sum of the x scorzes

∑y = sum of the y scores

∑x2 = sum of the squared x scores

∑y2 = sum of the squared y scores

Equation 2: Pearson correlation coefficient formula

Table 2: Standards of Pearson correlation coefficient

Size of correlation	Interpretation
0.90-1.00 (−0.90 to −1.00)	Very high positive (negative) correlation
0.70-0.90 (−0.70 to −0.90)	High positive (negative) correlation
0.50-0.70 (−0.50 to −0.70)	Moderate positive (negative) correlation
0.30-0.50 (−0.30 to −0.50)	Low positive (negative) correlation
0.00-0.30 (0.00 to −0.30)	Negligible correlation

2.6.6. **Linear Regression**

Regression analysis is a strategy for assessing the relationship among gatherings of factors which are the explanation and result in a connection (Uyanık & Güler, 2013). Analyze the relationship between the dependent & independent variables, and formulate the linear equation as the main focus of univariate regression analysis. Multilinear regression is called the regression model with one dependent variable and more than one independent variable.

2.6.6.1. Model Summary Table

R and R^2 values are provided by the Model Summary Table. Correspondingly, the correlation is represented by R. The amount of the total variation in the dependent variable can be clarified by the independent variable which is indicated by the R^2 value.

2.6.6.2. ANOVA Table

ANOVA table displays to which extent the regression equation fits the data. In other words, it means that the regression model predicts the dependent variable fundamentally.

2.6.6.3. Coefficient Table

The Coefficient table is yet another significant table. The sample coefficient table is shown in Figure 2. It helps predict the Dependent Variable from a particular Independent variable, by providing the necessary information. By scrutinizing the "Sig." column, we can determine whether the Independent variable statistically contributes to the model. Furthermore, the values in column "B" under the "Unstandardized Coefficients" section can be used to present the regression equation.

For Example: -

Success of EE Strategies & Practices = 0.717 (Poor Communication) +0.518

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	.518	.065		8.000	<.001	.390	.645
	PoorCommunication	.717	.038	.700	19.033	<.001	.643	.791

a. Dependent Variable: EEStrategiesPractices

Figure 2: Coefficient for Poor Communication and EE Strategies

2.7. Discussion

2.7.1. Summary of the comparison of related work

The Factors which are tended by previous researches, advantages, and disadvantages of the denoted literature resources are depicted in Table 3.

Table 3: Summary of the comparison of related works

Related work	Method	Factors Considered	Advantages	Disadvantages
(Lwakatare, Kuvaja, et al., 2016)	In phase one, three interviews were conducted in one company. MLR was conducted in phase two.	How professionals classify the DevOps experience? What are the practices of DevOps as indicated by software professionals?	The researchers have improved the scientific definition of DevOps and identified some patterns of DevOps practices.	Researchers have used only a few number of interviewees for their investigations.
(Leite et al., 2019)	Followed qualitative procedure. They have set analysis procedures based on Grounded Theory strategies.	Formulating a model to supervise the software professionals in the investigation of the DevOps process. Examining and studying DevOps ideas and difficulties from the viewpoints of: engineers, managers, and analysts.	This survey investigates a more extensive scope of sources and is exceptional in comparison to the past examinations. It includes the feasible exposure, and ramification of DevOps.	The researchers have not considered academic workshops, and all the books regarding DevOps in the field. Selected papers are subjectively biased.
(Perera, Bandara, et al., 2017)	The research indicators are measured qualitatively by questionnaire and quantitatively by interviews.	Check whether the components such as quality, responsiveness to business needs, and agility (adapting to technology) are behaving when installing DevOps?.	The researchers have found a positive core relation between DevOps and successful software development in Sri Lanka.	The main recommendations are developed only based on qualitative data.
(Perera, Silva, et al., 2017)	The linear model was obtained from the linear regression analysis. Online questionnaires and interviews have been used for qualitative data.	How have DevOps practices impacted the quality of the software? How to enhance quality successfully?	CAMS framework affects the quality of the software products. Hence practising DevOps will help in revamping the software quality.	The information has been gathered only from the DevOps experts in Sri Lanka.

2.7.2. **Literature related to the factors**

The summarization of identified 19 factors was denoted in Table 4. Simultaneously, they were grouped into 7, according to behaviour. Some groups showed multiple factors and those are indicated within the brackets saying which factor is found under which group. Each factor is depicted with a relevant group. The literature related to factors is elaborated in Table 5.

Table 4: Grouped table of the factor table in related works

Grouped Factors	Factors found from the literature
Collaboration Aspects	1. Poor communication
	2. Natural communication
Organization Culture	1. Organization silos
	2. Empathy
	3. Unsatisfactory test environment
	4. Information flow
Socio-Technical Aspects	1. Technology stack
	2. Skills and experience
	3. Knowledge sharing
	4. Programming language
	5. Critical importance of other tasks
	6. IT Operation not involved in RS
External Pressure	1. Customer expectation
	2. Responsiveness to customer
	3. Customer usage of the software
Automation	1. Deployment process
	2. Infrastructure as Code

Table 5: Literature Related to Factors

Research article	Collaboration Aspects	Monitoring	Measurements	Organization Culture	Socio-Technical Aspects	External Pressure	Automation
(Artač et al., 2016)		X	X	X (1)	X (1,2,3)	X(1)	X(1)
(Artac et al., 2017)	X(2)		X	X(1)	X(1,2,3,5)		X(1,2)
(Chung, 2017)	X(2)	X		X(1)	X(2)		X (1)
(Jones et al., 2016)	X(1,2)	X		X(1,3,4)	X(1,2,,3,5)	X (1,2)	X (1)
(Kerzazi & Adams, 2016)	X(1)	X	X	X(1)	X (1,2,3)		
(Lwakatare et al., 2015)	X(1,2)	X	X				X(1,2)
(Leite et al., 2019)	X(2)	X	X	X(1,4)	X(1,2,,3,6)		X(1,2)
(Lwakatare, Karvonen, et al., 2016)	X(1,2)	X	X	X(1)	X(1,2,,3)	X (1)	X (1)
(Lwakatare, Kuvaja, et al., 2016)	X(1)	X	X	X(1,2,3,4)	X(1,2,,3,6)	X(2,3)	X(1,2)
(Nicolau de França et al., 2016)	X(1,2)	X	X	X(1,2)	X(1,2,,3,5,6)	X(1,2,3)	X(1,2)
(Perera, Bandara, et al., 2017)	X(1,2)		X	X(1,2,4)	X(1,2,,3)	X(1,2,3)	X(1,2)
(Bucena & Kirikova, 2017)	X(2)	X	X	X(1,3)	X(1,2,,3,5)	X (1)	X (1)
(Debbiche, 2019)	X(2)	X	X	X(1)	X(2,,3)	X(1,2)	X(1,2)
(Díaz et al., 2018)	X(2)			X(1,3,4)	X(1,5,6)	X(1,2,3)	X (1)
(Ebert et al., 2016)	X(1)	X		X(1,2)	X(1,2,3)		X(1,2)
(Erich et al., 2017)	X(1)		X	X(2)	X(1,2,3)	X (3)	X (1)
(Forsgren & Kersten, 2018)	X(1)		X	X(1,3)	X(1)		X (1)

2.8. Conclusion

The researcher has done the exploration based on the past literature about what the factors are affecting the success of the effort estimation in the DevOps environment. Starting from the overviewing of the DevOps-based software development, DevOps process, how the DevOps practice evolves in the Sri Lankan context, effort estimation strategies, and practices. Subsequently, the researcher has described and thoroughly analyzed the outcome when effort estimation is conducted in a DevOps background and the final result when several factors affect the success of the effort estimation phase in DevOps-based software development. Finally, the researcher could identify the factors affecting the success of effort estimation in the DevOps context.

3. RESEARCH METHODOLOGY

3.1. Chapter Introduction

Exploring the factors which need to be considered the success of effort estimation, and recommend a useful guideline for a more reliable forecast about the software development timeline in a project are leading objectives of this research. The research approach is described in the first part of this chapter. Then, this chapter flows through the data collection process, an illustrative view of the design, conceptual diagram, and hypothesis. Subsequently, the pilot survey and reliability analysis are explained. Finally, data analysis techniques used for data analysis are elaborated in this chapter.

3.2. Research Approach

The plan and the process of the research that explains the stages from broad assumptions to detailed methods of data collection, analysis, and interpretation is defined as a Research Approach. There are three types of research. Namely; “Exploratory Research”, “Descriptive Research” and “Explanatory “or “Casual Research”. Descriptive researches are focused on expanding the knowledge on existing issues through the data collection process, and addresses “what” in the research subject than the question of “why”. Also, describing, explaining, and validating the findings are the three main purposes of the descriptive studies. Hence, based on the above explanation, we could state that this research study is defined as “Descriptive” research.

The research design intends to give a suitable framework for a study. Choosing the research design approach is the most important decision of the research design process since it determines how necessary information and facts are obtained in the study (Jilcha Sileyew, 2019). Qualitative, quantitative, and mixed methods are the main three research approaches. This study is employed as quantitative research because it includes collecting and changing over information into mathematical form. Therefore, statistical calculations can bring on those numeric values and ultimately helps to conclude. The survey inquiry was distributed among the employees who work in a

DevOps-based environment in software companies in Sri Lanka. To identify the thoughts of the participants, the numeric scale was used to quantify the opinions, attitudes, and behaviours in the survey questionnaire. The researcher has done the data analysis after the data collection. The SPSS tool which was provided by IBM was used for the data analysis in this study. The mentioned analytic tools were used by the researcher for generating the facts and hidden patterns in this study. After that, the researcher was able to present the set of guidelines that can be used when doing the effort estimation in DevOps-based software development in the Sri Lankan context. Figure 3 depicts the stages of the research that have been passed on by the researcher.

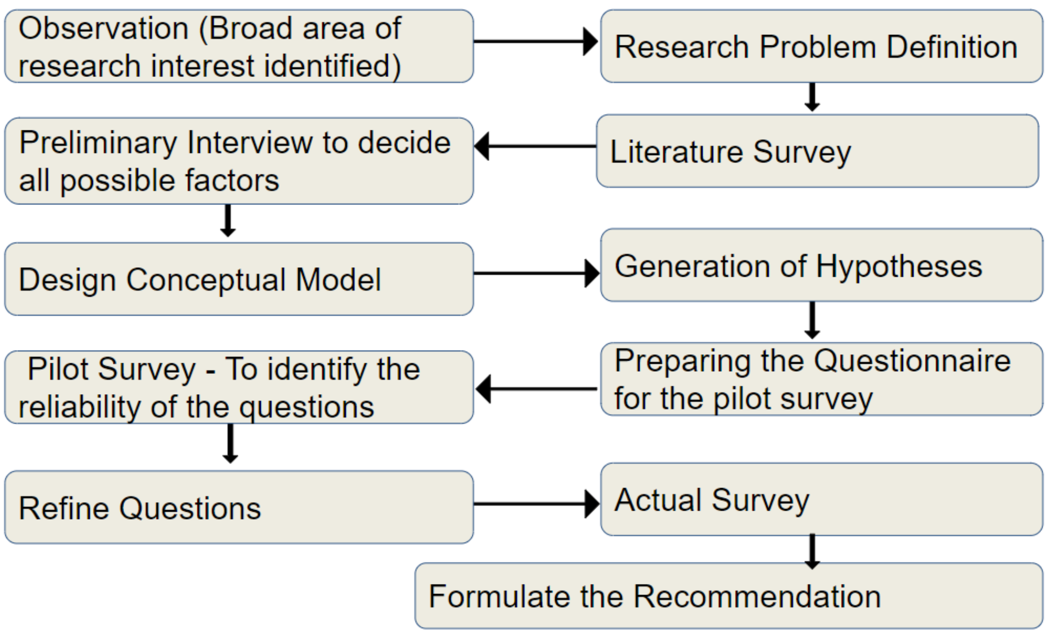

Figure 3: Research Approach

In the beginning, the research problem which is going to focus on this study is identified by the researcher. Respectively, the research scope and the research objectives were defined for this study. Later, the previous studies and related works were reviewed by the researcher to recognize the findings, drawbacks, or issues of past effort estimation models or techniques on literature, and the gap between the current research problem which is going to be addressed in this investigation. Then 19 independent variables were identified by the researcher that needs to be considered the success of effort estimation in the DevOps practice. After that, the preliminary interview was conducted. Recognizing the key factors in the actual practice of software development activities in the Sri Lankan context was the main purpose of conducting the preliminary interview. Finally, as the results of the preliminary interview, the 7 variables were filtered as the most important factors for the success of effort estimation in the DevOps context. It is presented in Figure 5.

The conceptual framework model provides a graphical illustration of the relationships between the independent and the dependent variables of the study. Figure 5 shows the conceptual framework for this research study. After selecting the independent factors in opposition to the one dependent variable, which is the success of effort estimation practices in the DevOps context, the conceptual model was created by the researcher. Upon that, the hypothesis was described to cover all the selected independent variables. Ultimately, the research questionnaire was built by the researcher as the next step, to suppress all the relationships and areas between the independent and dependent variables. This questionnaire was shared among over 350 software professionals who are working or have worked in the DevOps context in the IT Industry of Sri Lanka. Then the pilot survey was conducted by the researcher to compute the reliability and validity of the questions. It was done by distributing the questionnaire to a few software professionals who have worked in a DevOps environment. Then the researcher was able to make the final questionnaire after refining all questions numerous times. The final questionnaire was distributed among software professionals through online sources by analyzing the collected questionnaires' responses. This was concluded with the finding that which hypotheses have been accepted or rejected. As a result, a set of instructions, practices were proposed to follow at the effort estimation phase in the DevOps environment IT industry of Sri Lanka.

3.3. Data Collection Method

The correct information collection strategy refers to the contrast between valuable insights into knowledge and time-squandering confusion (Ainsworth, 2019). It is clear evidence that the data collection method can be considered a very important aspect of the research. There are numerous methods of data collection such as surveys, interviews (individual or in-depth), focus groups, and observations. The data collection method is defined as a process of collecting details from relevant origins with the determination of answering the research questions, hypothesis testing, and evaluating the outcomes. There are two categories in the data collection method. Those are Primary Data Collection methods and Secondary Data Collection methods.

3.3.1. Primary Data

If the data is directly collected from the main sources, it is defined as Primary data. The interviews, surveys, and experiments are a few of the main sources. The principal foundations of primary data are typically chosen to fulfill the needs or necessities of specific research. Hence, aspects like research aim and target population need to be recognized before selecting primary data collection sources (Dudovskiy, 2018). Since this research is quantitative, data must be numerical and quantifiable. The questionnaire is with a close-ended question that covers all selected factors for this research study. Besides that, the preliminary interview was conducted with a few stakeholders.

3.3.2. Secondary Data

Secondary data is a type of data that has already been published in books, journals, newspapers, magazines, reports, etc. A large amount of data is available in the above sources which match with the research area. Despite that, the background of the research area is not considered in those data. Hence, it is very important to pay attention to the Secondary Data selection criteria in the research study as it plays a major role with regards to rising the validity and the reliability of the research. However, these criteria are not restricted to date of publication, author credentials, reliability of the source, quality of conversations, depth of analysis, the degree of commitment of the text to the improvement of the research area, etc. (Dudovskiy, 2018).

3.4. Sample Design

The plans and techniques to be continued in choosing the sample from the given population are defined as the sample design (Kabir, 2016). It is an estimation strategy function for processing the sample statistics. These insights are the appraisals used to construe the population boundaries. The sample design is decided before the data collection of the research study. Probability Sampling and Non-Probability Sampling are the main two types of sample designs.

3.4.1. Population

The population is referred to as a set of individuals or objects with the same characteristics that the researcher is interested in (Jilcha Sileyew, 2020). The researcher should carefully define and identify the population in the research study, hence credibility is essential in research, and data integrity is influenced by the credibility of the research outcome (Asiamah et al., 2017).

At the moment, the Sri Lankan ICT workforce is experiencing several changes in its structure while the global ICT sector is in the transition period. According to (ICTA, 2019), the overall strength of the workforce is **146,089** in 2019. As per their survey which was conducted in 2019, ICT companies have become the dominant employer with a share of **65.5%** of the workforce. The ICT workforce has been made with the composite of several job categories like “Software Engineers”, “Quality Assurance Engineers”, “Business Analytics”, “Project Managers”, “System Administrators/Engineers”, and “Database Administrators”. As per their survey, those job categories are shared with **39%, 15%,6%, 4%, 5%, and 4%**. of the workforce respectively.

In this research, the researcher has decided to consider specified job roles such as "Software Engineers", "Quality Assurance Engineers", "Business Analytics", "Project Managers", and "System Administrators/Engineers" and "Database Administrators" as the survey users who are from the DevOps based software companies in Sri Lanka.

(Perera, Silva, et al., 2017) have mentioned in their research that **66%** of respondents are from the DevOps-based software companies in Sri Lanka. Hence, the population size can be derived as: -

The ICT workforce in 2019 = **149,089* 65.5%**

Total workforce with respect to considered job roles in ICT workforce

= **149,089 * 65.5% (39%+15%+6%+4%+5%+4%)**

Considering back, the considered job roles from DevOps based software companies

= **149,089 * 65.5%*66% (39%+15%+6%+4%+5%+4%)**

Population Size = **47,049**

Hence, the researcher decided **47,049** would be the identified population size for this research study based on the assumptions and resources. The researcher is supposed to select respondents based on their work experience in DevOps-based software development.

3.4.2. **Sample Size**

Sampling refers to a selected group of objects from the general population. Also, the selected group represents the actual population of that particular research. The random sample should be a satisfactory size, to avoid sampling errors or biases (Taherdoost, 2017). In terms of the sample size required in the research study, the qualitative and quantitative research methods may differ. The pioneer of sampling is defining the acceptability of the population. If the researcher was unable to identify and understand the accessible population properly and correctly, the sample may lead to worthless outcomes. Any attempt to sample may lead to unnecessary outcomes until the accessible population is identified and understood (Asiamah et al., 2017). There is a formula that was introduced by (Krejcie & Morgan, 1970) for determining the sample size of a research study. Equation 3 shows the formula used to calculate the sample size. There is a table that consists of the population (N) and sample size (S). That applies to any defined population.

$$S = \frac{x^2 NP(1-P)}{d^2(N-1) + x^2 P(1-P)}$$

Equation 3: Sample size calculation formula (Krejcie & Morgan, 1970)

Where:

S = Required sample size.

X^2 = The table value of chi-square for 1 degree of freedom at the desired confidence level (3.841).

N = The population size.

P = The population proportion (assumed to be .50 since this would provide the maximum sample size).

d = The degree of accuracy expressed as a proportion (.05).

$$\text{Sample Size} = \frac{3.841 * \mathbf{47{,}049} * (0.5)^2}{(0.05)^2 * \mathbf{47{,}048} + 3.841 * 0.5 * 0.5}$$

$$= \mathbf{380}$$

Hence, the population size is **47,049** for this study, **380** would be the representative sample size for this research study with a confidence level (3.841) and an error margin of 0.5.

3.4.3. Research Instruments

Research instruments are simply devices or tools for obtaining data relevant to the research project such as surveys, interviews, checklists, and tests, and there are many alternatives from which to choose (Wilkinson & Peter, 2003).

The close-ended questionnaire was the main instrument used for the data collection in this research study, and it was shared between the software professionals who have work experience in a DevOps-based environment in the Sri Lankan context.

The questionnaire consists of 41 questions and it is published in a google form that is easily accessible for everyone. The rating scale was the “Five-point Likert” scale. This questionnaire is comprised of two main segments and one open-ended question. The first segment consists of demographic questions to get a better understanding of certain background characteristics of an audience. The questions in the second segment are more focused on gathering information, and for the intention of testing the hypothesis. The open-ended question was included to collect the general idea of the respondents, about the effort estimation strategies and practices in the DevOps environment.

The preliminary interview can be considered as the second main research instrument in this study which was done by the researcher. It was conducted to verify the most important factors (variable) out of all the 19 factors which are found in the literature review. As the preliminary interview, the researcher prepared a questionnaire inclusive of 12 questions. It contained a few questions to identify the factors affecting the success of effort estimation in DevOps-based software development in the Sri Lankan IT Industry. The specific job roles such as Software Developers, QA Engineers, Business Analysis, Project Managers, and System Engineers who have work experience in the DevOps context were selected for the interview by the researcher. Also, the researcher did not forget to select an equal number of interviewees from each job role, hence not biasing the responses. The interview was conducted informally and some were interviewed over the phone and through social media platforms. After gathering 25 responses, the researcher was able to select the 5 most important factors (variables) with the highest number of responses Those responses were selected to address in this research study. The summary of the preliminary interview result is shown in Figure 4.

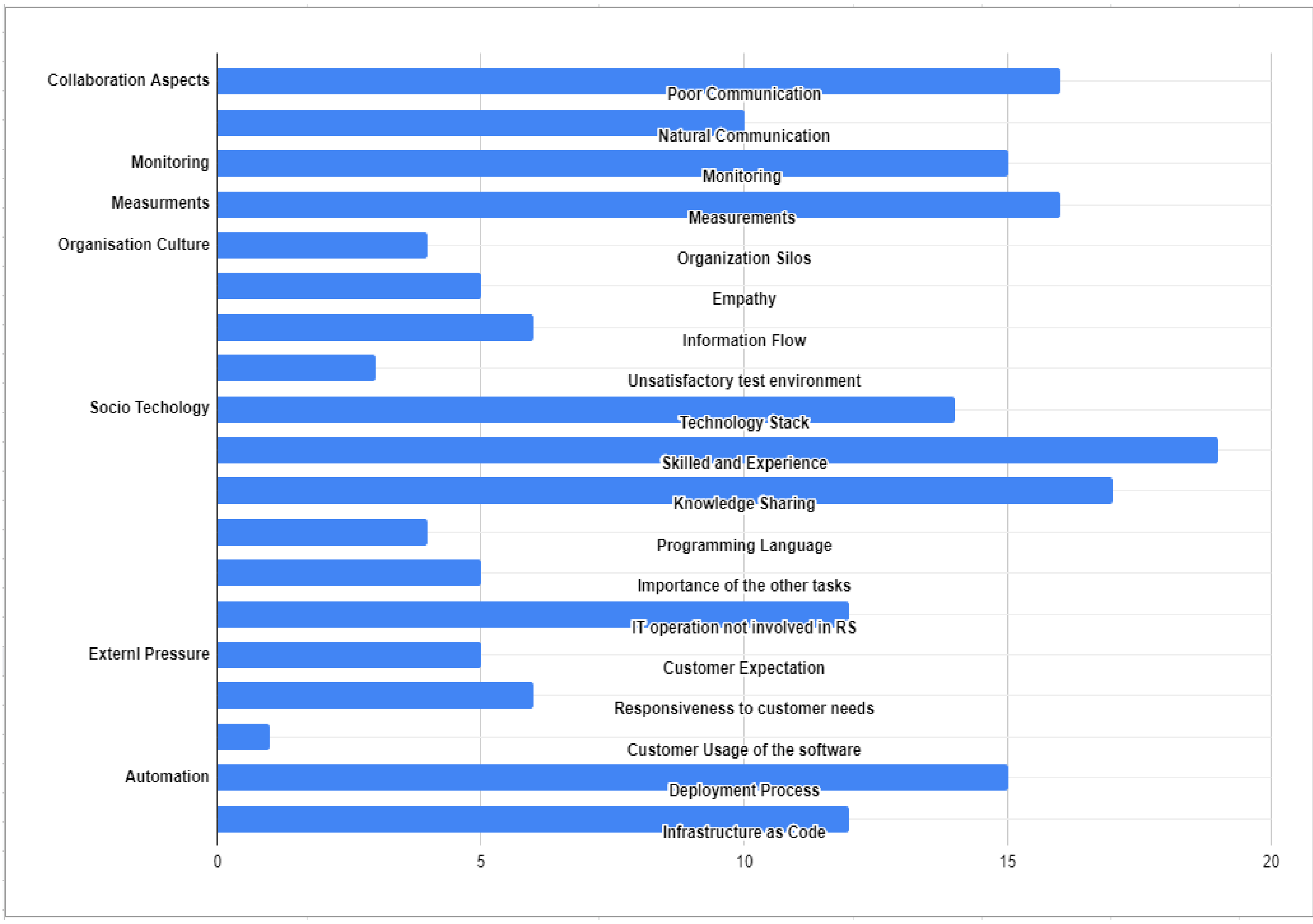

Figure 4: Summarization of the Preliminary Interview result

3.5. Conceptual Model/Design

A Conceptual Model is a design that clarifies the progressing sequence of the components to be examined by the researcher. (Adom & Hussain, Emad. Kamil. and Joe, 2018). Contrarily, it is how the researcher explored the research problem through the study. As (Grant & Osanloo, 2014) stated, the relationships between the main component and other sub-components are described in the conceptual framework from a statistical point of view. It is coordinated intelligently and its' structure gives an image or visual picture of how ideas are connected in a study. It is not a static graph of variables and could be subject to change. However, it displays a relationship between the variables, relating to knowledge, the system of methods, and assumptions. Each entity of the conceptual framework displays the study of what it is (Mugizi, 2019).

Depending on the literature review, 19 independent DevOps-based effort estimation factors were identified by the researcher, listed as,

Poor communication,

Natural communication,

Measurements,

Monitoring,

Organization silos,

Information flow in the organization,

Unsatisfactory test environment,

Technology stack,

Skills and experience

Empathy (ability to understand and share the feelings of another),

Knowledge sharing,

Programming language,

Importance of other tasks,

Customer expectation,

Customer Usage of the software,

Responsiveness to customer needs,

Deployment process and

Infrastructure as Code.

IT operation not involved in RS (Requirement Specification) tasks,

Then the preliminary interview was conducted to identify the most important effort estimation factors in the DevOps context. After analyzing the preliminary interview responses and previous literature, the researcher was able to choose the seven important factors as independent variables to continue the research. The researcher also identified those seven factors as the sub factors and used the group name as its main factor. It is depicted in Table 6 with relevant literature.

Table 6: Main factors and subfactors with the literature

Main Factor	**Sub Factors**	Literature
Collaboration aspects	**Poor communication**	(Forsgren & Kersten, 2018), (Lwakatare, Kuvaja, et al., 2016) (Ebert et al., 2016), (Erich et al., 2017),(Kerzazi & Adams, 2016), (Lwakatare et al., 2015)
Measurements		(Artač et al., 2016), (Kerzazi & Adams, 2016), (Lwakatare et al., 2015), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Perera, Bandara, et al., 2017), (Bucena & Kirikova, 2017), (Debbiche, 2019), (Erich et al., 2017), (Forsgren & Kersten, 2018)
Monitoring		(Artač et al., 2016), (Chung, 2017), (Jones et al., 2016), (Kerzazi & Adams, 2016), (Lwakatare et al., 2015), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Bucena & Kirikova, 2017), (Debbiche, 2019), (Ebert et al., 2016)
Socio-Technology	**Technology stack**	(Artač et al., 2016), (Artac et al., 2017), (Jones et al., 2016), (Kerzazi & Adams, 2016), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Perera, Bandara, et al., 2017), (Bucena & Kirikova, 2017), (Díaz et al., 2018), (Ebert et al., 2016), (Erich et al., 2017), (Forsgren & Kersten, 2018)
	Knowledge sharing	(Artač et al., 2016), (Artac et al., 2017), (Jones et al., 2016), (Kerzazi & Adams, 2016), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Perera, Bandara, et al., 2017), (Bucena & Kirikova, 2017), (Debbiche, 2019), (Ebert et al., 2016), (Erich et al., 2017),

	Skills and experience	(Artač et al., 2016), (Artac et al., 2017), (Chung, 2017), (Jones et al., 2016), (Kerzazi & Adams, 2016), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Perera, Bandara, et al., 2017), (Bucena & Kirikova, 2017), (Debbiche, 2019), (Ebert et al., 2016), (Erich et al., 2017)
Automation	**Deployment process**	(Artač et al., 2016), (Artac et al., 2017), (Chung, 2017), (Jones et al., 2016), (Lwakatare et al., 2015), (Leite et al., 2019), (Lwakatare, Karvonen, et al., 2016), (Lwakatare, Kuvaja, et al., 2016), (Nicolau de França et al., 2016), (Perera, Bandara, et al., 2017), (Bucena & Kirikova, 2017), (Debbiche, 2019), (Ebert et al., 2016), (Erich et al., 2017), (Díaz et al., 2018), (Forsgren & Kersten, 2018)

The effort estimation practices in DevOps-based software development are recognized as the dependent variable in this research study. Hence, the conceptual model is shown as below: -

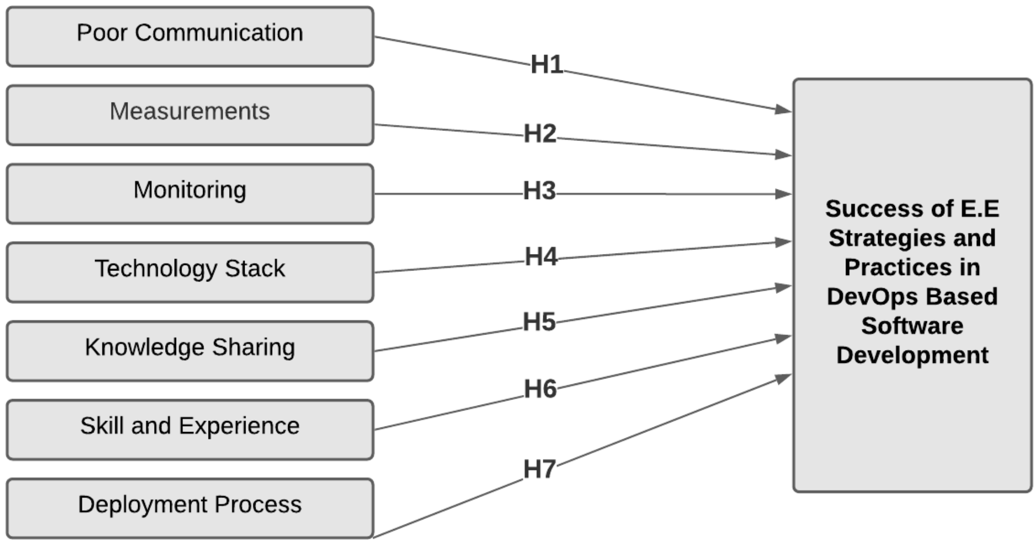

Figure 5: Conceptual Framework

“**Poor Communication**” was selected as one of the independent variables, hence the respondents were selected as one of the most important factors, and (Riungu-

Kalliosaari et al., 2016) has mentioned that uncoordinated activities may result due to poor communication between developers and operations person in the team.

"**Measurements**" are the best match for representing the success of the software development (Perera, Bandara, et al., 2017), and "Measurements" is considered as a concept of DevOps (Jabbari et al., 2016). It was mentioned as the third-highest important factor by the respondents in Figure 4. Hence, it is selected as another independent variable out of the 19 factors.

The interviewees also mentioned how "**Monitoring**" was an important factor out of the 19 factors and (Erich et al., 2017) have mentioned how practitioners had defined DevOps in various mnemonics. The model of CAMS, CALMS (CAMS +Lean), and CAMM are examples of them. In every technique, Monitoring was a key component. Hence this was also selected as one of the independent variables in this research study.

According to Figure 4, "**Technology Stack** "has obtained 13 responses from over 25 respondents. Also (Senapathi et al., 2018) have mentioned it as a key challenge related to DevOps adoption. As per (Perera, Silva, et al., 2017), developers and the operational teamwork collaboratively to give quality releases by being accountable for their service and underlying technology stack in the DevOps-based software development. Further, service-level agreements play an important role (Garusinghe et al., 2017). Hence, this factor was also selected by the researcher as an independent variable.

The interviewees mentioned **"Skills & Experience**" as the most important factor and (Hemon, Lyonnet, et al., 2020) highlighted the importance of skills & experience and the mindset of Ops people in DevOps based software development context. The skills and experience expected from DevOps roles are varying from country to country and project to project (Hussain et al., 2017). Hence, this was selected as an independent variable in the research design.

Knowledge Sharing is also obtained in 17 responses, which means it is considered as one of the most important factors by interviewees. (Hemon, Fitzgerald, et al., 2020) has mentioned that Knowledge Sharing is confined within the team, thereby limiting the chances for Knowledge Sharing in accomplishing ongoing improvement. On the other hand, it is not constantly encouraging the team.

Therefore, “Knowledge Sharing” is selected as the main factor in this research study.

Two-third of respondents have mentioned that the “**Deployment Process**” is an important factor and (Díaz et al., 2018) highlighted how frequently production release should be given with the Deployment pipeline. This means putting changes into production through automation. It may result in many deployments every day. Hence, it was considered as one of the important factors. Therefore, the researcher selected it as another independent variable in this research study.

3.6. Hypothesis

Poor Communication

H$_{1-0}$: There is no relationship between Poor Communication and success of effort estimation **Practices** in the DevOps Environment.

H$_{1-1}$: There is a relationship between Poor Communication and the success of effort estimation **Practices** in the DevOps Environment.

Measurements

H$_{2-0}$: There is no relationship between Measurements and the success of effort estimation **Practices** in the DevOps Environment.

H$_{2-1}$: There is a relationship between Measurements and the success of effort estimation **Practices** in the DevOps Environment.

Monitoring

H$_{3-0}$: There is no relationship between the Monitoring function and the success of effort estimation **Practices** in the DevOps Environment.

H$_{3-1}$: There is a relationship between the Monitoring function and the success of effort estimation **Practices** in the DevOps Environment.

Technology Stack

H_{4-0}: There is no relationship between the Technology Stack function and the success of effort estimation **Practices** in the DevOps Environment.

H_{4-1}: There is a relationship between the Technology Stack function and the success of effort estimation **Practices** in the DevOps Environment.

Knowledge Sharing

H_{5-0}: There is no relationship between the Knowledge Sharing function and the success of effort estimation **Practices** in the DevOps Environment.

H_{5-1}: There is a relationship between the Knowledge Sharing function and the success of effort estimation **Practices** in the DevOps Environment.

Skills and Experience

H_{6-0}: There is no relationship between the Skills and Experience function and the success of effort estimation **Practices** in the DevOps Environment.

H_{6-1}: There is a relationship between the Skills and Experience function and the success of effort estimation **Practices** in the DevOps Environment.

Deployment Process

H_{7-0}: There is no relationship between the Deployment Process function and the success of effort estimation **Practices** in the DevOps Environment.

H_{7-1}: There is a relationship between the Deployment Process function and the success of effort estimation **Practices** in the DevOps Environment.

3.7. Pilot Survey

As per (In, 2017), a good research study requires the relevant experimental design and accurate performance to obtain a high-quality research outcome. Hence, analyzing its feasibility and improve the efficiency and quality before performing the main study, can be beneficial. This prior event can be identified as a "Pilot survey". Inclusive of 8 demographic questions and all selected variables, the questionnaire was distributed among 25 software professionals such as "Software Engineers", "Quality Assurance Engineers", "Business Analytics", "Project Managers", "System Administrators/Engineers" and "Database Administrators" who have work experience in a DevOps environment. After collecting the responses of the pilot study, the researcher was able to calculate the Cronbach alpha value for each question to check its reliability.

3.7.1. Reliability Analysis

Research is a unique and systematic investigation attempted to increase existing knowledge and comprehension of the obscure to establish facts and principles.

Therefore, the quality of the research is the most vital fact. The reliability of the analysis should be completed to confirm the quality of the research instruments. In simple terms, the reliability of the research is how much the research method produces steady and reliable consistent output under the same conditions. Reliability analysis provides the relationship between every item in the instrument. Internal consistency is an approach to determine how well a survey or test is estimating what you need it to gauge. The degree of similarity between two items is measured by computing the correlation in the reliability analysis. If the correlation values are high, the questionnaire is a more reliable one.

Internal consistency can be measured by Cronbach's alpha. It measures how closely related a set of items in a group are. It is viewed as an evaluation of scale dependability

(Nicolau de França et al., 2016). Cronbach’s alpha is used to calculate the reliability of this research study. Table 1 shows the robustness of the alpha values and its’ levels.

The reliability analysis was executed after conducting the pilot survey. The analysis was done for all variables (dependent and independent) in the research study. The IBM SPSS Statistics tool was used for reliability analysis. As per Table 1 in Section 2.6.1, the researcher was able to arrive at the below decisions related to the questions in the questionnaire.

$\alpha > 0.7$ - The set of questions of the particular variable possess reliability. Additionally, those questions have good internal consistency for measuring the suitable variable.

$\alpha < 0.6$ - The set of questions are insufficient to quantify the variable. Hence, the researcher happened to remove and improve certain questions.

The summary of the reliability analysis result is shown in Table 8.

3.8. Questionnaire

The Questionnaire is one of the most generally utilized methods to gather information in a particular sociology research. The principal target of the questionnaire in an examination is to acquire important data most dependably and legitimately (Taherdoost, 2018). The questionnaire of this research study consists of forty-one questions. It was created to get the data on the success of effort estimation strategies and practices in the DevOps environment. The first section comprises of 4 demographic questions that help to recognise certain foundation attributes of the spectators, regardless of whether there was gender, job role, work experience, educational level, work situation, etc. The next 4 general questions are related to the organization's background, DevOps experience, and effort estimation strategies and practices.

The second section consists of thirty-two questions which are designed to collect data on the variables identified at the preliminary assessment. Four to Five questions were designed to gather information about each independent variable of this research study.

All questions are designed to a five-point Likert scale. This is otherwise referred to as a satisfaction scale. That ranges from one extreme (1. Strongly Agree) attitude to another (5. Strongly Disagree).

The detailed structure of these thirty-two questions is demonstrated in Table 7. The last question was designed to collect the data related to the opinion of the selected sample with regards to the success of effort estimation and strategies, and practices in DevOps-based software development and to get ideas from the respondents. The researcher hopes to use that question when declaring the recommendation in this study.

This questionnaire was distributed through Google Forms which can be easily accessed by the public, and it was published among over 380 software professionals to get the maximum number of responses.

Table 7: Detailed structure of the factor related questions

#	Question	Poor Communication	Measurements	Monitoring	Technology Stack	Knowledge Sharing	Skills & Experience	Deployment Process	E. E Strategies Practices in DevOps
9	The use of communication tools and well-defined communication guidelines make it easy to estimate the required effort in the DevOps practice.	×							×
10	The scope of the assigned tasks can be easily understood if we improve the communication skills.	×							
11	Proper communication helps to improve the motivation to fulfil a task in DevOps practice.	×							
12	The accuracy of the effort estimation in DevOps practice can be improved by allocating sufficient time to communicate with the collaborators.	×							
13	Proper communication at the stage of effort estimation defines a lightweight process to accomplish a task.	×							
14	The use of Measuring tools and well-defined measuring guidelines make it easy to estimate the required effort in the DevOps practice.		×						
15	Daily Effort Measurements will increase effectiveness.		×						

16	The structure of effort estimation metrics make estimation easy in DevOps practice.		×						
17	The effort of similar tasks can be predicted since the matrices are given numeric outputs.		×						
18	Monitoring the server's health ensures that configurations and scanned features are working as intended.			×					
19	Monitoring development milestones helps in determining the effectiveness of the Dev team.			×					
20	The potential threats can be recognized in advance by the Ops team through monitoring the system continuously.			×					
21	Monitoring the Full-Stack and Faster Troubleshooting are the main characteristics of proper Monitoring tools.			×					
22	The impacted areas of development can be identified by Continuous Monitoring of the system.			×					
23	The project Requirements and Enhancements should be compatible with the Tech-Stack (Set of tools, frameworks, and libraries) of the application.				×				
24	The involvement of Technical Expertise will be an added advantage in the effort estimation.				×				×
25	Having a good understanding of the Tech-Stack (Structure Wise-Frontend/Backend) of the application to the team will make it easy to estimate the effort.				×				
26	Selecting Ready-Made (Ideally suited) solutions of Tech-Stack (Combination of technologies) will make the effort estimation more convenient while doing a workaround on the application.				×				
27	Sharing the technical knowledge of Senior Engineers within the team helps to ease the effort estimation.					×			
28	Having an idea about the existing reusable implementation makes it easy to estimate the effort.					×			
29	Conducting Technical sessions about the architecture of the application helps effort estimation in DevOps practice.					×			
30	Maintaining Best Practices in development and testing with the team make it easy to estimate the effort.					×			
31	The use of prior Skills and Experiences would help to measure the effort in DevOps practice.						×		×
32	Sharing Research and Development (R&D) experiences in the application, make it easy to estimate the effort in the DevOps practice.						×		

33	Having skills & Experience in a specific area in development personally helps me in effort estimation.						×		
34	The use of prior Skills and Experiences in the specific programming language would help to measure the effort easily in the DevOps context.						×		
35	An Automated deployment process helps to reduce the effort and risks in the production environments.							×	
36	The Automated deployment process reduces the integration time in the DevOps environment.							×	
37	Both debugging and fixing efforts are considered as a total effort of the effort estimation in the DevOps environment.							×	
38	The Automated deployment process allows continuous delivery and makes effort estimation easy in a DevOps environment.							×	
39	Following effort estimation practices helps in continuous delivery of software with high quality in the DevOps environment.								×
40	I feel that we can be more productive at work if we follow effort estimation practices as a team.								×

3.9. Conclusion

Research methodology refers to specific techniques used to identify, select and analyze, collected information about the topic. It also refers to the interrelationship between independent and dependent variables. In this chapter, the researcher explains how the researcher approached the research, the data collection methods used for this research study, how the researcher arrived at selecting the target population, how he derived it, how the calculation of the sample size was done and, how the research data was collected and analyzed, etc. Nineteen factors related to the success of effort estimation strategies were identified in the literature review phase by the researcher. Then the final survey was created based on the selected most important factors from the preliminary interview and the literature. Additionally, this chapter explains how the researcher calculated Cronbach’s alpha value, how the stability of the questions was checked, and how the quality of the questionnaire was improved.

4. DATA ANALYSIS

4.1. Chapter Introduction

The findings and results of data analysis are demonstrated in this chapter. The researcher numerically represented variables by calculating the mean value of sub-elements under each independent variable. This research mainly focuses on the relationship between numerically appointed variables and dependent variables, hence this study is a quantitative one.

As the first analysis in this research study, reliability analysis was conducted for the final questionnaire to check the consistency of the questions. After that, the descriptive analysis was conducted for the demographic data. Then the correlation analysis to check the relationship between variables, inferential analysis, and regression analysis to check the relationship between the dependent variable with multiple variables were performed respectively, with the use of the IBM statistic subscription tool. Finally, the defined hypotheses were tested by the researcher with the aid of the above analysis.

4.2. Data Collection

Quantitative examination is characterized as a deliberate examination of phenomena by collecting quantifiable information and performing measurable, numerical, or computational strategies. The reliability of quantitative information was measured with the use of statistical methods in this research study.

The data collection lasted for one month for this research study. The researcher has collected quantitative data by publishing an online survey. This online survey questionnaire was created using the google form (online tool) and it was dispensed among the software professionals who have work experience in the DevOps environment in the Sri Lankan IT Industry. The researcher used Social Media, Gmail, and LinkedIn as online sources to distribute this survey.

The researcher was able to collect 380 data sets. It took one month to complete the data collection. There are no missing values in the data set. The data collection pattern over time is demonstrated by Figure 6.

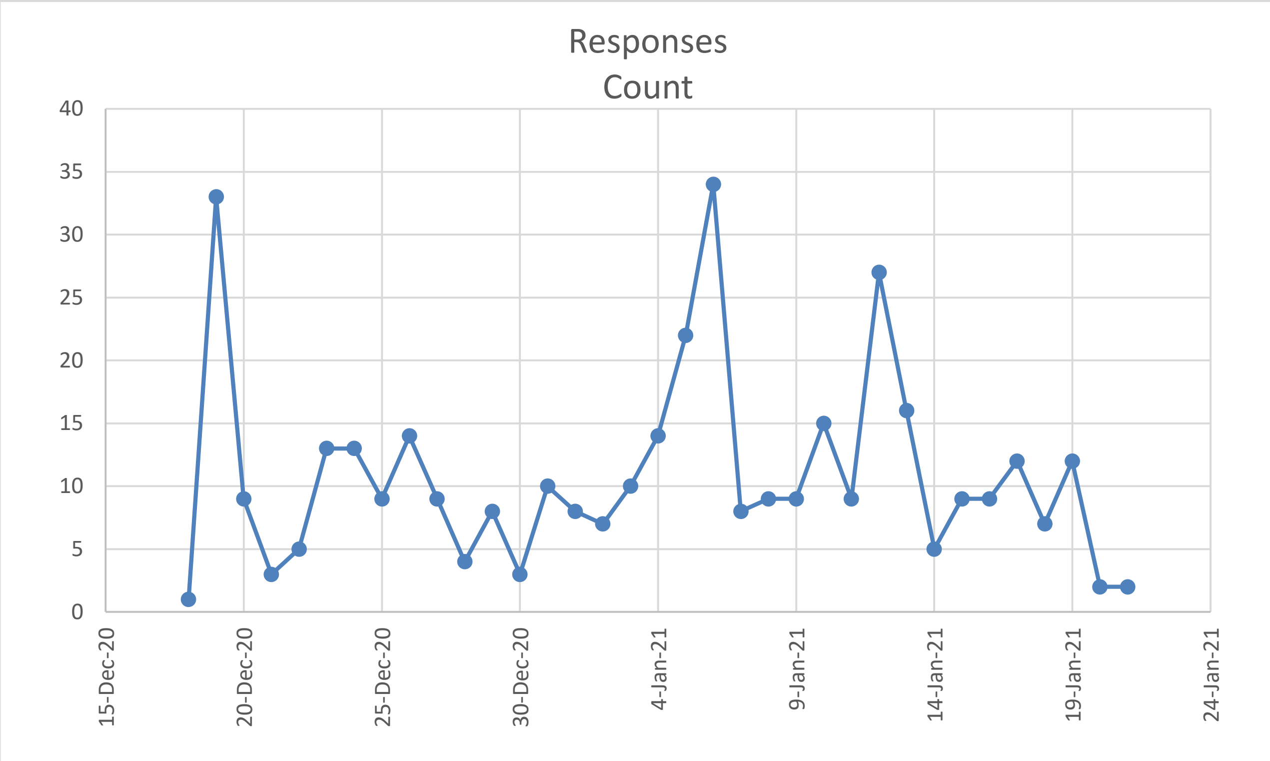

Figure 6: Responses count over time

4.3. Reliability Analysis

Checking the quality of the questionnaire is the most essential part of the research study. Hence, the researcher has carried out a reliability analysis for the survey questions. It was the very first analysis executed to collect data for this study. Internal Consistency is the measure of reliability. It was calculated for the set of items in the reliability analysis. If the questionnaire has a high positive core relation value, then it is deliberated as a reliable questionnaire.

For measuring the reliability of the questions in the final survey, Cronbach’s alpha was used by the researcher. Table 1 shows the strength level of the alpha values. The alpha values for each independent variable are calculated with the usage of the SPSS Statistic

Subscription tool. The detailed view of the reliability analysis for this study is shown in Table 8 and Figure 15.

4.3.1. **Poor Communication**

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.724	.726	5

Figure 7: The Reliability Analysis of Poor Communication

Figure 7 depicts the reliability statistic of "Poor Communication". As per Figure 7, Poor Communication has 0.724 of Cronbach's alpha. According to Table 1 in Section 2.6.1, Poor Communication has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate Poor Communication. In another way, "Poor Communication" has good internal consistency. There is an adequate amount of questions to capture the particular factor.

4.3.2. **Measurements**

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.839	.846	4

Figure 8: The Reliability Analysis of Measurements

Figure 8 reflects the reliability statistic of "Measurements". As per Figure 8, Measurements has 0.839 of Cronbach's alpha. According to Table 1 in Section 2.6.1, Measurement has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate Measurement. In other terms, "Measurement" has good internal consistency, and there are enough questions to capture the particular factor.

4.3.3. **Monitoring**

Reliability Statistics		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.709	.709	5

Figure 9: The Reliability Analysis of Monitoring

Figure 9 shows the reliability statistic of "Monitoring". As per Figure 9, Monitoring has 0.709 of Cronbach's alpha. According to Table 1 in Section 2.6.1, Monitoring has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate Monitoring. In another way, "Monitoring" has good internal consistency, and there is an adequate number of items to catch the particular factor.

4.3.4. **Tech Stack**

Reliability Statistics		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.707	.708	4

Figure 10: The Reliability Analysis of Tech Stack

The reliability analysis of "Tech Stack" is shown in Figure 10. According to Figure 10, Tech Stack has 0.707 of Cronbach's alpha. As per Table 1 in Section 2.6.1, Tech

Stack has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate Tech Stack. Hence, "Tech Stack" has good internal consistency, and there are enough questions to capture the particular factor.

4.3.5. **Knowledge Sharing**

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.796	.800	4

Figure 11: The Reliability Analysis of Knowledge Sharing

The reliability analysis of "Knowledge Sharing" is shown in Figure 11. According to Figure 11, Knowledge Sharing has 0.796 Cronbach's alpha value. As per Table 1, the questions which were created to measure Knowledge Sharing have good reliability. In other terms, "Knowledge Sharing" has good internal consistency. the researcher is fully ensured about the questions, and the questionnaire included enough questions to capture the particular factor.

4.3.6. **Skills and Experience**

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.801	.812	4

Figure 12:The Reliability Analysis of Skills and Experience

The reliability analysis of "Skills & Experience" is shown in Figure 12. According to Figure 12, Skills & Experience have 0.801 of Cronbach's alpha. As per Table 1 in Section 2.6.1, Skills & Experience has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate Skills & Experience Hence, "Skills & Experience" has good internal consistency, there are enough questions to capture the particular factor.

4.3.7. **Deployment Process**

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.812	.816	4

Figure 13: The Reliability Analysis of Deployment Process

The reliability analysis of the "Deployment Process" is shown in Figure 13. According to Figure 13, the Deployment Process has 0.812 of Cronbach's alpha. As per Table 1 in Section 2.6.1, the Deployment Process has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate the Deployment Process. Therefore, the "Deployment Process" has good internal consistency, there are enough questions to capture the particular factor.

4.3.8. Effort Estimation Strategies and Practices in DevOps based software development

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.705	.712	5

Figure 14: Reliability Analysis of Effort Estimation Strategies and Practices in a DevOps environment.

The reliability analysis of "Effort Estimation Strategies and Practices in DevOps-based software development" is shown in Figure 14. According to Figure 14, Effort Estimation Strategies and Practices in DevOps-based software development has 0.705 of Cronbach's alpha. As per Table 1 in Section 2.6.1, Effort Estimation Strategies and Practices in DevOps-based software development has good reliability. It reveals that the researcher is fully ensured about the questions created to evaluate the Effort Estimation Strategies and Practices in DevOps-based software development. Hence, the "Effort Estimation Strategies and Practices in DevOps-based software development" has good internal consistency, and there are enough questions to capture the particular factor.

Table 8: Summary of Reliability Analysis in the survey questions

Variables	**Number of Items**	**Cronbach's Alpha**
Independent		
Poor Communication	5	0.724
Measurements	4	0.839
Monitoring	5	0.709
Technology Stack	4	0.707
Knowledge Sharing	4	0.796
Skills and Experience	4	0.801
Deployment Process	4	0.812
Dependent		
The Success of Effort Estimation Strategies and Practices in DevOps based software development environment	5	0.705

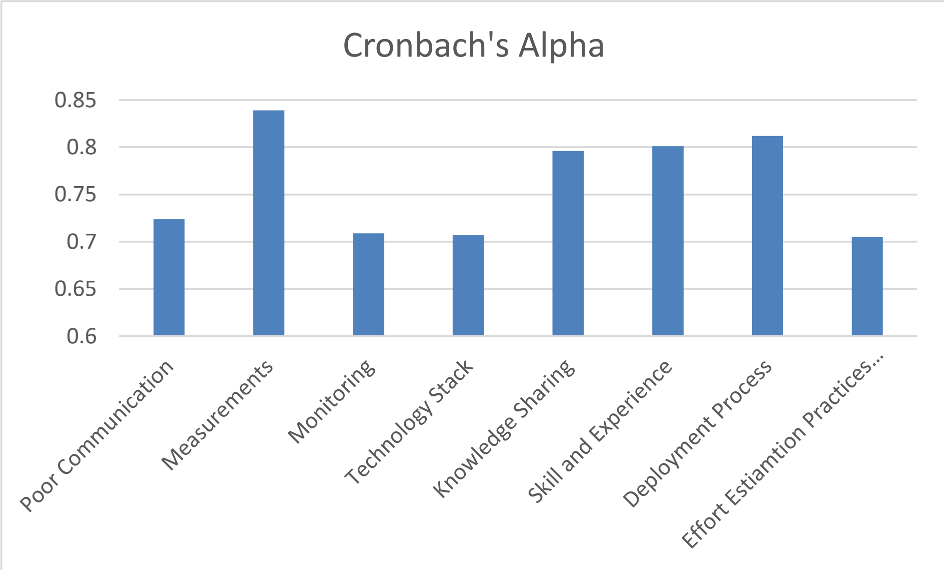

Figure 15: Reliability statistics using Cronbach’s alpha

4.4. Descriptive Statistic of Demographic Data

Descriptive statistics describe the basic features of the data and provides the main summary and measures in the research study. It gives an idea about the distribution of the data and enables the identification of the relationships among the variables.

4.4.1. Analysis of Demographic Data using Charts

Gender and Current Working Track Classification of Sample

The Gender distribution of Figure 16, clearly shows that most of the respondents are Male. Out of the 380 samples of data, 57% of the respondents are presented as Male software professionals, and the rest of the sample was represented as Female software professionals. It is 43%, out of the 380 samples of the data.

The Distribution of the Current Working Track of the sample is presented in Figure 17. It clearly shows that most of the respondents were attributed to Development. Out of the 380 samples of data, 27% of the software professionals belong to Development Track. 26 % of the 380 sample data represents the Quality Assurance Track. The rest of the sample is represented in Operation Track and Delivery. Those are 24% and 23% out of the 380 sample data respectively. Both Project Managers and Business Analysts are attributed to the Delivery Track. These statistics show an equal number of Software Professionals in the IT Industry to a moderate extent.

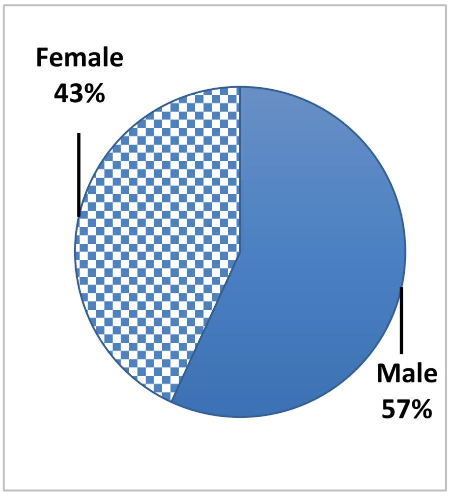

Figure 16: Gender Distribution

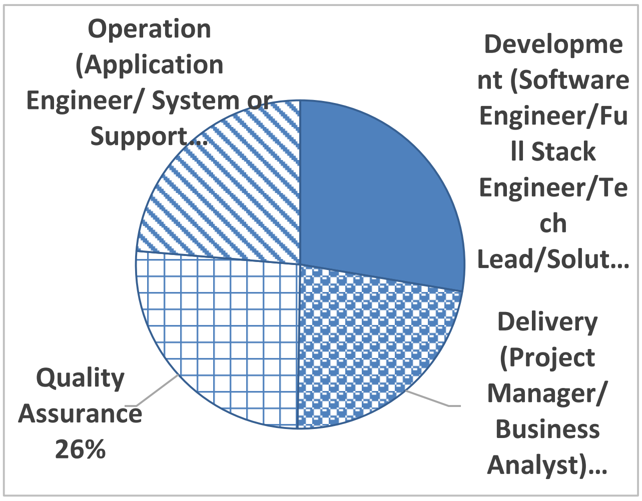

Figure 17: Distribution of Working Track

Distribution of Work Experience and Education Level Classification of Sample

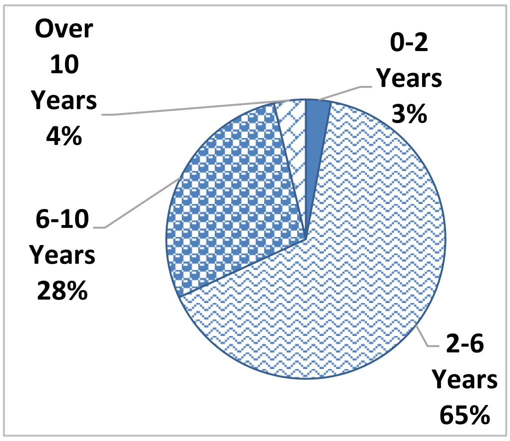

Figure 18: Distribution of Work Experience

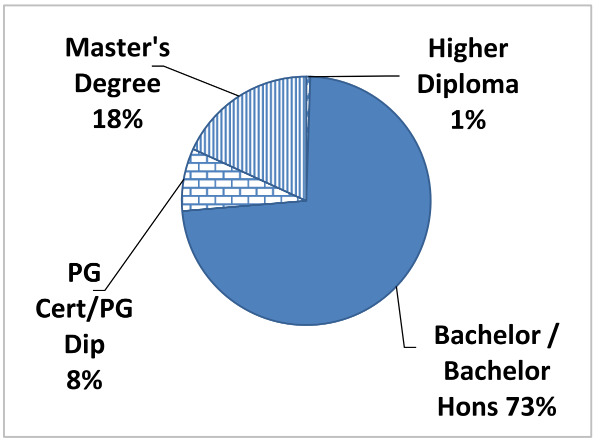

Figure 19: Distribution of Education Level

The distribution of Work Experience in Figure 18, clearly shows that most of the respondents have 2-6 years of work experience. Out of the 380 samples of data, 28% of the respondents were attributed as 6-10 years of experience as software professionals. The rest of the 3% and 4% of the sample data belongs to 0-2 years and over 10 years experienced software professionals respectively.

The distribution of the Highest Education level of the sample is shown in Figure 19, and it clearly shows that most of the respondents are Bachelor's / Bachelor Hons' degree holders. The second most respondents are Master's Degree holders. 8% of the sample is attributed as PG Cert/PG Dip. This statistic is strong evidence that well-educated software professionals are in the industry.

No. of Employees and Organization Category Classification of Sample

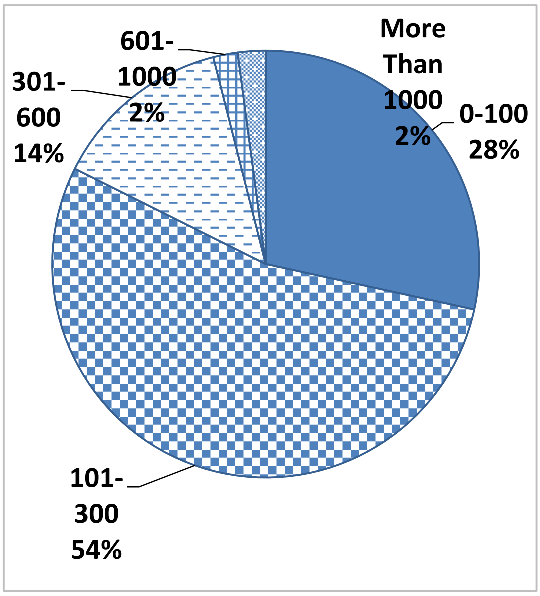

Figure 20: No. of Employees Distribution

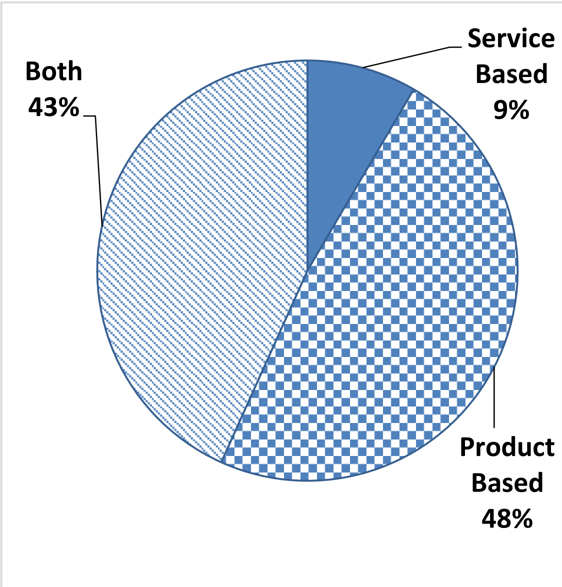

Figure 21: Distribution of Org. Category

Figure 20 portrays the distribution of No. of employees in the organization of Respondents. It clearly shows that most of the respondents are working in a mid-level organization. Out of the 380 sample data, 54% have attributed to 101-300 groups of the organization, and the rest of the sample has attributed to the 0-100 category of organization.

The distribution of the Organization category of Respondents is shown in Figure 21. It demonstrates that most of the respondents are working in product-based software organizations. Out of the 380 of the sample data, 48% have attributed to Product based group and the rest of the sample mentioned that their organizations are functioned by Service-based and Product-based.

Following EE Strategies in Organization and Opinion about EE Practices Classification in Sample

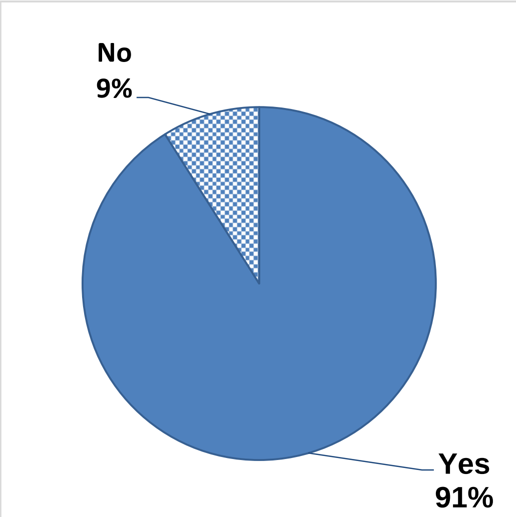

Figure 22: Following the EE Strategies & Practices

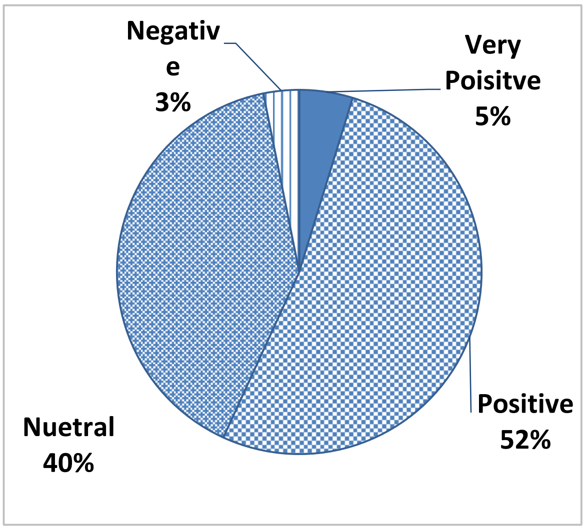

Figure 23: Opinion on EE Strategies & Practices

The distribution of the following EE Strategies in the Organization of the sample is shown in Figure 22. It signifies that most of the respondents attributed that their organization follows the effort estimation strategies. 51% of the sample have attributed it as "Yes" and the rest of the 49% of the sample elaborates that their organization doesn't follow the effort estimation strategies.

Figure 23 exhibits the classification of opinion on DevOps practices of Respondents. It clearly shows that most of the respondents have a positive mindset about effort estimation practices in DevOps-based software development. Out of the 380 of sample data, 52% have attributed to Positive, and the rest of the sample of 40% have attributed to Neutral.

4.4.2. Statistical Analysis of Demographic Data Using Pearson Correlation

Pearson product-moment coefficient correlation or bivariate correlation is used to measure the statistical relationship or association between two continuous variables. Hence, the researcher assumed that the Likert scale data is interval or ratio data. The strength of the relationship is denoted by Pearson's r or coefficient correlation value (r) while P-value represents the statistical significance of the experiment. The Pearson's R, or (r) value is between -1 and 1, and the significance value (P-value) is between 0 and 1.

$P > 0.05$ - There is a significant relationship between the two variables.

$P < 0.01$ or $P < 0$ - There is no significant relationship between the variables.

$R > 0$ - It has a positive relationship.

$R < 0$ - It has a negative relationship.

Here, the researcher has measured the correlation between demographic variables and dependent variables.

4.4.2.1. Analyzing the relationship between Gender and Success of EE Strategies and Practices in a DevOps environment.

The Pearson correlation was calculated to determine that there is a relationship between poor communication and Success of Effort Estimation strategies and practices in the DevOps environment. As per Figure 24, Pearson's r is -0.386. The results of the two-tailed test are less than 0.001.

$r(378) = -0.386$, $p < 0.001$ and $p < 0.01$

Hence, there is a significant negative relationship between gender and the success of effort estimation strategies in the DevOps environment.

Correlations			
		Gender	EEStrategies Practices
Gender	Pearson Correlation	1	-.386**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.386**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 24: Pearson correlation between gender and Dependent variable

4.4.2.2. Analyzing the relationship between Working Track and Success of EE Strategies and Practices in a DevOps environment.

As per Figure 25, the coefficient correlation was computed to check that there is a relationship between the current working track and the Success of Effort Estimation strategies and practices in the DevOps environment. The Pearson's correlation coefficient (r) was -0.209 and the two-tailed significant value is less than 0.001 in between working track and the success of effort estimation strategies in the DevOps environment.

$r(378) = -0.209$, $p < 0.001$ and $p < 0.01$

There is a significant negative relationship between the working track and the success of effort estimation strategies in the DevOps environment.

Correlations			
		Current track	EEStrategies Practices
Current track	Pearson Correlation	1	-.209**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.209**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 25: Pearson correlation between working track and Dependent variable

4.4.2.3. Analyzing the relationship between Work Experience and the Success of EE Strategies and Practices in a DevOps environment.

As appeared in Figure 26, the Pearson correlation value was figured out to verify that there is a relationship between work experience and the Success of Effort Estimation strategies and practices in the DevOps environment. Then, the Pearson's correlation coefficient(r) was -0.023 and the two-tailed significant value was 0.660.

$r(378) = -0.023$, $p = 0.660$ and $p > 0.05$

After the calculation, it was determined that there is no relationship between work experience and the Success of Effort Estimation strategies and practices in the DevOps

Correlations

		Working experience	EEStrategies Practices
Working experience	Pearson Correlation	1	-.023
	Sig. (2-tailed)		.660
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.023	1
	Sig. (2-tailed)	.660	
	N	380	380

environment.

Figure 26: Pearson correlation between work experience and Dependent variable

4.4.2.4. Analyzing the relationship between Education Level and the Success of EE Strategies and Practices in a DevOps environment.

The Pearson correlation was calculated to determine that there is a relationship between the education level and the Success of Effort Estimation strategies and practices in the DevOps environment. As per Figure 27, Pearson's r is -0.095. The result of the two-tailed test is 0.063.

$r(378) = -0.095$, $p = 0.063$ and $p > 0.05$

After the calculation, it was decided that there is no relationship between work experience and the Success of Effort Estimation strategies and practices in a DevOps environment.

Correlations

		Level of Education	EEStrategies Practices
Level of Education	Pearson Correlation	1	-.095
	Sig. (2-tailed)		.063
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.095	1
	Sig. (2-tailed)	.063	
	N	380	380

Figure 27: Pearson correlation between education level and Dependent variable

4.4.2.5. Analyzing the relationship between No. of Employees and Success of EE Strategies and Practices in a DevOps environment.

Correlations

		Employees	EEStrategies Practices
Employees	Pearson Correlation	1	-.116*
	Sig. (2-tailed)		.024
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.116*	1
	Sig. (2-tailed)	.024	
	N	380	380

*. Correlation is significant at the 0.05 level (2-tailed).

Figure 28: Pearson correlation between no. of employees in EE strategies and Dependent variable

As per Figure 28, the coefficient correlation was computed to check that there is a relationship between No. of Employees in the organization and the Success of Effort Estimation strategies and practices in the DevOps environment. Then, the Pearson's correlation coefficient (r) was -0.116, and the two-tailed significant value was 0.024

in between working track and the success of effort estimation strategies in the DevOps environment.

$r(378) = -0.116$, $p = 0.024$ and $p < 0.05$

No. of Employees in the organization and the success of effort estimation strategies in the DevOps environment has a significant negative relationship.

4.4.2.6. Analyzing the relationship between Following EE Strategies and Success of EE Strategies and Practices in a DevOps environment.

The Pearson correlation was calculated to determine that there is a relationship between the Following EE Strategies and the Success of Effort Estimation strategies and practices in the DevOps environment. As per Figure 29, Pearson's r is -0.094. The result of the two-tailed test showed a 0.069 significance value.

$r(378) = -0.094$, $p = 0.069$ and $p > 0.05$

After the calculation, it was decided that there is no relationship between the Following EE Strategies and the Success of Effort Estimation strategies and practices in the DevOps environment.

Correlations

		Organization follow the effort estimation strategies	EEStrategies Practices
Organization follow the effort estimation strategies	Pearson Correlation	1	-.094
	Sig. (2-tailed)		.069
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.094	1
	Sig. (2-tailed)	.069	
	N	380	380

Figure 29: Pearson correlation between following in EE strategies and Dependent variable

4.4.2.7. Analyzing the relationship between Category of Organization and Success of EE Strategies and Practices in a DevOps environment.

As appeared in Figure 30, the Pearson correlation value was figured out to verify that there is a relationship between Organization Category and the Success of Effort Estimation strategies and practices in the DevOps environment. Then, the Pearson's correlation coefficient (r) was -0.063 and the two-tailed significant value was 0.223

$r(378) = -0.063$, $p = 0.223$ and $p > 0.05$

After the calculation, it was determined that there is no relationship between the Organization Category and the Success of Effort Estimation strategies and practices in the DevOps environment.

Correlations

		Category of organization	EEStrategies Practices
Category of organization	Pearson Correlation	1	-.063
	Sig. (2-tailed)		.223
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.063	1
	Sig. (2-tailed)	.223	
	N	380	380

Figure 30: Pearson correlation between following in EE strategies and Dependent variable

4.4.2.8. Analyzing the relationship between Opinion on EE Strategies and the Success of EE Strategies and Practices in a DevOps environment.

As appeared in Figure 31, the Pearson correlation value was figured out to verify that there is a relationship between the opinion on E.E Strategies and the Success of Effort Estimation strategies and practices in the DevOps environment. Then, Pearson's correlation coefficient (r) was -0.083, and the two-tailed significant value was 0.104.

$r(378) = -0.083$, $p = 0.104$ and $p > 0.05$

After the calculation, it was determined that there is no relationship between opinion on E.E Strategies and the Success of Effort Estimation strategies and practices in the DevOps environment.

Correlations

		Opinion on effort estimation practices in DevOps based software development	EEStrategies Practices
Opinion on effort estimation practices in DevOps based software development	Pearson Correlation	1	-.083
	Sig. (2-tailed)		.104
	N	380	380
EEStrategiesPractices	Pearson Correlation	-.083	1
	Sig. (2-tailed)	.104	
	N	380	380

Figure 31: Pearson correlation between following in EE strategies and Dependent variable

4.5. Inferential Analysis

4.5.1. Correlation Analysis of Variables for entire sample Data

The statistical relationship between all the independent variables and dependent variables was measured by the Pearson product-moment coefficient correlation (PPMCC) or bivariate correlation. As viewed in Figure 32, all the independent variables (poor communication, measurement, monitoring and technology stack, knowledge sharing, skills and experience, and deployment process), have positive Pearson's R(r) values with the dependent variable the success of effort estimation strategies and practices in DevOps based software development. Perusing Section 4.2.2, the dependent variable has a positive correlation with all the independent variables. The positive relationship between the dependent and independent variables is also shown in Figure 32.

Correlations

		EEStrategies Practices	PoorCommunication	Measurements	Monitoring	Technology Stack	Knowledge Sharing	SkillAndExperience	DeploymentProcess
EEStrategiesPractices	Pearson Correlation	1	.700**	.496**	.469**	.760**	.639**	.682**	.598**
	Sig. (2-tailed)		<.001	<.001	<.001	<.001	<.001	<.001	<.001
	N	380	380	380	380	380	380	380	380
PoorCommunication	Pearson Correlation	.700**	1	.618**	.568**	.666**	.577**	.560**	.556**
	Sig. (2-tailed)	<.001		<.001	<.001	<.001	<.001	<.001	<.001
	N	380	380	380	380	380	380	380	380
Measurements	Pearson Correlation	.496**	.618**	1	.513**	.596**	.515**	.640**	.597**
	Sig. (2-tailed)	<.001	<.001		<.001	<.001	<.001	<.001	<.001
	N	380	380	380	380	380	380	380	380
Monitoring	Pearson Correlation	.469**	.568**	.513**	1	.519**	.310**	.548**	.393**
	Sig. (2-tailed)	<.001	<.001	<.001		<.001	<.001	<.001	<.001
	N	380	380	380	380	380	380	380	380
TechnologyStack	Pearson Correlation	.760**	.666**	.596**	.519**	1	.593**	.718**	.683**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001		<.001	<.001	<.001
	N	380	380	380	380	380	380	380	380
KnowledgeSharing	Pearson Correlation	.639**	.577**	.515**	.310**	.593**	1	.660**	.643**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001		<.001	<.001
	N	380	380	380	380	380	380	380	380
SkillAndExperience	Pearson Correlation	.682**	.560**	.640**	.548**	.718**	.660**	1	.721**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001	<.001		<.001
	N	380	380	380	380	380	380	380	380
DeploymentProcess	Pearson Correlation	.598**	.556**	.597**	.393**	.683**	.643**	.721**	1
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001	<.001	<.001	
	N	380	380	380	380	380	380	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 32: Pearson correlation between all the Independent variables and Dependent variable

4.5.2. **Hypothesis Testing**

Hypothesis testing is also called significance testing and is a statistical method that is used to decide the entire population. Ronald Fisher, Jerzy Neyman, Karl Pearson, and Pearson's son, Egon Pearson are the founders of Hypothesis testing. There is a value called P-value (level of significance or criterion) which is used to make the decision. If the level of significance (Ho & Ho, 2019) is lesser than 0.05, the null hypothesis becomes true. Then the researcher happens to reject the value that is stated in the null hypothesis.

4.5.2.1. Poor Communication and the Success of EE strategies and Practices in DevOps

H_{1-0}: There is **no relationship** between Poor Communication and the success of effort estimation **Practices** in the DevOps Environment.

H_{1-1}: There is **a relationship** between Poor Communication and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between Poor communication and EE Strategies and Practices in the DevOps environment. It is shown in Figure 33. The significance value (p) was < 0.001 at the 0.01 level.

$r (378) = 0.700$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between poor communication and EE Strategies and Practices in the DevOps environment according to its correlation analysis.

Its Pearson's correlation coefficient (r) is= 0.700 and P-value < 0.01

Hence, the researcher was able to reject the **H_{1-0}** and accept the alternative hypothesis **H_{1-1}**.

Correlations

		PoorCommunication	EEStrategiesPractices
PoorCommunication	Pearson Correlation	1	.700**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.700**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 33: Correlation analysis of Poor communication and EE Strategies and Practices

4.5.2.2. Measurements and the Success of EE strategies and Practices in DevOps

H2-0: There is **no relationship** between Measurements and the success of effort estimation **Practices** in the DevOps Environment.

H2-1: There is **a relationship** between Measurements and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between the Measurement and EE Strategies and Practices in the DevOps environment. It is shown in Figure 34. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.496$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between measurement and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

And its Pearson's correlation coefficient (r) = 0.496 and P-value < 0.01

Hence, the researcher was able to reject the **H2-0** and accept the alternative hypothesis **H2-1**

Correlations

		Measurements	EEStrategies Practices
Measurements	Pearson Correlation	1	.496**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.496**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 34: Correlation analysis of Measurement and EE Strategies and Practices

4.5.2.3. Monitoring and the Success of EE strategies and Practices in DevOps

$H_{3\text{-}0}$: There is **no relationship** between the Monitoring function and the success of effort estimation **Practices** in the DevOps Environment.

$H_{3\text{-}1}$: There is **a relationship** between the Monitoring function and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between the Monitoring and EE Strategies and Practices in the DevOps environment. It is shown in Figure 35. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.469$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between monitoring and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

Its Pearson's correlation coefficient (r) is= 0.469 and P-value < 0.01

Hence, the researcher was able to reject the **$H_{3\text{-}0}$** and accept the alternative hypothesis **$H_{3\text{-}1}$**.

Correlations

		Monitoring	EEStrategies Practices
Monitoring	Pearson Correlation	1	.469**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.469**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 35: Correlation analysis of Monitoring and EE Strategies and Practices

4.5.2.4. Technology Stack and the Success of EE strategies and Practices in DevOps

H_{4-0}: There is **no relationship** between the Technology Stack function and the success of effort estimation **Practices** in the DevOps Environment.

H_{4-1}: There is **a relationship** between the Technology Stack function and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between the Technology Stack and EE Strategies and Practices in the DevOps environment. It is shown in Figure 36. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.760$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between technology stack and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

Its Pearson's correlation coefficient (r) is = 0.760 and P-value < 0.01

Hence, the researcher was able to reject the **H_{4-0}** and accept the alternative hypothesis **H_{4-1}**.

Correlations

		TechnologyStack	EEStrategiesPractices
TechnologyStack	Pearson Correlation	1	.760**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.760**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 36: Correlation analysis of Technology Stack and EE Strategies and Practices

4.5.2.5. Knowledge Sharing and the Success of EE strategies and Practices in DevOps

$H_{5\text{-}0}$: There is **no relationship** between the Knowledge Sharing function and the success of effort estimation **Practices** in the DevOps Environment.

$H_{5\text{-}1}$: There is **a relationship** between the Knowledge Sharing function and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between Knowledge Sharing and EE Strategies and Practices in the DevOps environment. It is shown in Figure 37. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.639$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between knowledge sharing and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

Its Pearson's correlation coefficient (r) is= 0.639 and P-value < 0.01

Hence, the researcher was able to reject the **$H_{5\text{-}0}$** and accept the alternative hypothesis **$H_{5\text{-}1}$**.

Correlations

		KnowledgeSharing	EEStrategiesPractices
KnowledgeSharing	Pearson Correlation	1	.639**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.639**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 37: Correlation analysis of Knowledge Sharing and EE Strategies and Practices

4.5.2.6. Skills & Experience and the Success of EE strategies and Practices in DevOps

H6-0: There is **no relationship** between the Skills and Experience function and the success of effort estimation **Practices** in the DevOps Environment.

H6-1: There is **a relationship** between the Skills and Experience function and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between Skills & Experience and EE Strategies and Practices in the DevOps environment. It is shown in Figure 38. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.682$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between skills & experience and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

Its Pearson's correlation coefficient (r) is= 0.682 and P-value < 0.01

Hence, the researcher was able to reject the **H1-0** and accept the alternative hypothesis **H1-1**.

Correlations

		SkillAndExperience	EEStrategiesPractices
SkillAndExperience	Pearson Correlation	1	.682**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.682**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 38: Correlation analysis of Skills and Experience and EE Strategies and Practices

4.5.2.7. Deployment Process and Success of EE strategies and Practices in DevOps

$H_{7\text{-}0}$: There is **no relationship** between the Deployment Process function and the success of effort estimation **Practices** in the DevOps Environment.

$H_{7\text{-}1}$: There is **a relationship** between the Deployment Process function and the success of effort estimation **Practices** in the DevOps Environment.

Pearson's r was calculated to verify the relationship between Deployment Process and EE Strategies and Practices in the DevOps environment. It is shown in Figure 39. The significance value (p) was < 0.001 at the 0.01 level.

$r(378) = 0.598$, $p < 0.001$ and $p < 0.01$

There is a positive relationship between the deployment process and EE Strategies and Practices in the DevOps environment based on its correlation analysis.

Its Pearson's correlation coefficient (r) is= 0.598 and P-value < 0.01

Hence, the researcher was able to reject the **$H_{1\text{-}0}$** and accept the alternative hypothesis **$H_{1\text{-}1}$**.

Correlations

		DeploymentProcess	EEStrategiesPractices
DeploymentProcess	Pearson Correlation	1	.598**
	Sig. (2-tailed)		<.001
	N	380	380
EEStrategiesPractices	Pearson Correlation	.598**	1
	Sig. (2-tailed)	<.001	
	N	380	380

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 39: Correlation analysis of Deployment Process and EE Strategies and Practices

Summary of the Hypothesis Testing

Table 9 shows the summary of the hypothesis testing of this research study. The strength of the relationship is denoted according to Table 2 in Section 2.6.5.

Table 9: Summary of the hypothesis testing results

Hypothesis	H0	HA	Coefficient Correlation value	Strength of the relationship	Direction
There is a relationship between Poor Communication and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.700	High	Positive
There is a relationship between Measurements and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.496	Low	Positive
There is a relationship between the Monitoring function and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.469	Low	Positive
There is a relationship between the Technology Stack function and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.760	High	Positive
There is a relationship between the Knowledge Sharing function and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.639	Moderate	Positive
There is a relationship between the Skills and Experience function and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.682	Moderate	Positive
There is a relationship between the Deployment Process function and the success of effort estimation **Practices** in the DevOps Environment.	Rejected	Accepted	0.598	Moderate	Positive

4.6. Regression Analysis

Regression analysis is used to check the relationship between a dependent variable and multiple independent variables. It consists of several statistical methods and can be utilized to measure the strength of the relationships and to prototype the future relationships of the variables. There are three variances of regression analysis, such as linear, multiple linear, and nonlinear. The researcher has conducted a simple linear regression analysis to find out the relationship between the variables in this study. A detailed description of the regression analysis will be denoted under this section by using three main tables mentioned in Section 2.6.6.

4.6.1. Independent Variable – Poor communication

The summary of the regression analysis between Poor Communication and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 40. R and R^2 values are 0.700 and 0.489 respectively. According to the explanation in Section 2.6.6.1, 48.9% of the variance in the Success of Effort Estimation strategies and practices can be explained by Poor Communication in this case.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.700[a]	.489	.488	.26367

a. Predictors: (Constant), PoorCommunication

b. Dependent Variable: EEStrategiesPractices

Figure 40: Model Summary of Poor Communication

ANOVA table for Poor Communication and Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 41. Here, the F value is 362.240 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Poor Communication) and dependent variable (Success of EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in a DevOps environment by Poor Communication.

$F(1,378) = 362.240$ and $P < 0.01$

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	25.184	1	25.184	362.240	<.001[b]
	Residual	26.279	378	.070		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), PoorCommunication

Figure 41: ANOVA Table of Poor Communication

The coefficient table for Poor Communication and the Success of Effort Estimation Strategies and Practices in DevOps-based software development appears in Figure 42. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.717(Poor Communication) +0.518

It reveals that 0.717 units increase in the Success of Effort Estimation Strategies and Practices when one unit increases in Poor Communication. The beta value is 0.700 and it represents the positive relationship between the variables. Considering this, poor communication at the effort estimation will be affected by the Success of effort estimation strategies and practices in the DevOps environment. Hence, there’s a need to enable communication methods when conducting the effort estimation in the DevOps environment.

Coefficients[a]								
		Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model		B	Std. Error	Beta	t	Sig.	Lower Bound	Upper Bound
1	(Constant)	.518	.065		8.000	<.001	.390	.645
	PoorCommunication	.717	.038	.700	19.033	<.001	.643	.791

a. Dependent Variable: EEStrategiesPractices

Figure 42: Coefficient Table for Poor Communication

4.6.2. **Independent Variable – Measurements**

The summary of the regression analysis between Measurement and Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 43. R and R^2 values are 0.496 and 0.246 respectively. According to the explanation in Section 2.6.6.1, 24.6% of the variance in the Success of Effort Estimation strategies and practices can be explained by Measurement in this case.

Model Summary[b]				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.496[a]	.246	.244	.32043

a. Predictors: (Constant), Measurements

b. Dependent Variable: EEStrategiesPractices

Figure 43: Model Summary Table for Measurement

ANOVA table for Measurement and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 44. Here, the F value is 123.228 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Measurement) and dependent variable (Success of EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by Measurement.

$F(1,378) = 123.228$ and $P < 0.01$

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	12.652	1	12.652	123.228	<.001[b]
	Residual	38.811	378	.103		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), Measurements

Figure 44: ANOVA Table for Measurement

The coefficient table for Measurement and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is depicted in Figure 45. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.403(Measurement) +0.979

It reveals that there is a 0.403 unit increase in the Success of Effort Estimation Strategies and Practices when one unit increases in Measurement. The beta value is 0.496 and this indicates the positive relationship between the variables. Considering this, measurement at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, we need to break down the task into small chunks and measure the effort of each of them when conducting effort estimation in the DevOps environment.

Coefficients[a]								
		Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model		B	Std. Error	Beta	t	Sig.	Lower Bound	Upper Bound
1	(Constant)	.979	.069		14.219	<.001	.844	1.115
	Measurements	.403	.036	.496	11.101	<.001	.332	.474

a. Dependent Variable: EEStrategiesPractices

Figure 45: Coefficient Table for Measurement

4.6.3. **Independent Variable – Monitoring**

The summary of the regression analysis between Monitoring and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 46. R and R^2 values are 0.469 and 0.220 respectively. According to the explanation in Section 2.6.6.1, 22.0% of the variance in the Success of Effort Estimation strategies and practices can be explained by Monitoring in this case.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.469[a]	.220	.217	.32598

a. Predictors: (Constant), Monitoring

b. Dependent Variable: EEStrategiesPractices

Figure 46: Model Summary Table for Monitoring

ANOVA table for Monitoring and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 47. Here, the F value is 106.311 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Monitoring) and dependent variable (EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by Monitoring.

$F(1,378) = 106.311$ and $P < 0.01$

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	11.297	1	11.297	106.311	<.001[b]
	Residual	40.166	378	.106		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), Monitoring

Figure 47: ANOVA Table of Monitoring

The coefficient table for Monitoring and the Success of Effort Estimation Strategies and Practices in DevOps-based software development appears in Figure 48. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.445(Monitoring) +0.833

It reveals that 0.445 units increase in the Success of Effort Estimation Strategies and Practices when there's one unit increase in Monitoring. The beta value is 0.469 and it indicates the positive relationship between the variables. Considering this, monitoring at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, there's a need to enable monitoring methods at the effort estimation in the DevOps environment.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	.833	.088		9.479	<.001	.660	1.005
	Monitoring	.445	.043	.469	10.311	<.001	.360	.530

a. Dependent Variable: EEStrategiesPractices

Figure 48: Coefficient Table for Monitoring

4.6.4. **Independent Variable – Technology Stack**

The summary of the regression analysis between Technology Stack and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 49. R and R^2 values are 0.760 and 0.577 respectively. According to the explanation in Section 2.6.6.1, 57.7% of the variance in the Success of Effort Estimation strategies and practices can be explained by Technology Stack in this case.

Model Summary[b]				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.760[a]	.577	.576	.23992

a. Predictors: (Constant), TechnologyStack

b. Dependent Variable: EEStrategiesPractices

Figure 49: Model Summary Table for Technology Stack

ANOVA table for Technology Stack and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 50. Here, the F value is 516.050 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Technology Stack) and dependent variable (EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by Technology Stack.

$F(1,378) = 516.050$ and $P < 0.01$

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	29.705	1	29.705	516.050	<.001[b]
	Residual	21.758	378	.058		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), TechnologyStack

Figure 50: ANOVA Table of Technology Stack

The coefficient table for Technology Stack and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is indicated in Figure 51. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.749(Technology Stack) +0.340

It reveals that there's a 0.445 units increase in the Success of Effort Estimation Strategies and Practices when there's a one-unit increase in Technology Stack. The beta value is 0.760 and it represents the positive relationship between the variables. Considering this, the technology stack at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, the team needs to think about the technology stack of the software project at the effort estimation in the DevOps environment.

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.	95.0% Confidence Interval for B Lower Bound	95.0% Confidence Interval for B Upper Bound
1	(Constant)	.340	.062		5.480	<.001	.218	.462
	TechnologyStack	.749	.033	.760	22.717	<.001	.684	.813

a. Dependent Variable: EEStrategiesPractices

Figure 51: Coefficient Table for Technology Stack

4.6.5. **Independent Variable – Knowledge Sharing**

The summary of the regression analysis between Knowledge Sharing and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 52. R and R^2 values are 0.639 and 0.408 respectively. According to the explanation in Section 2.6.6.1, 40.8% of the variance in the Success of Effort Estimation strategies and practices can be explained by Knowledge Sharing in this case.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.639[a]	.408	.407	.28388

a. Predictors: (Constant), KnowledgeSharing

b. Dependent Variable: EEStrategiesPractices

Figure 52: Model Summary for Knowledge Sharing

ANOVA table for Knowledge Sharing and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 53. Here, the F value is 260.617 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Knowledge Sharing) and dependent variable (EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by Knowledge Sharing.

$F(1,378) = 260.617$ and $P < 0.01$

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	21.002	1	21.002	260.617	<.001[b]
	Residual	30.461	378	.081		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), KnowledgeSharing

Figure 53: ANOVA Table for Knowledge Sharing

The coefficient table for Knowledge Sharing and Effort Estimation Strategies and Practices in DevOps-based software development appears in Figure 54. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.550(Knowledge Sharing) +0.721

It reveals that there's a 0.445 unit increase in the Success of Effort Estimation Strategies and Practices when there's a one-unit increase in Knowledge Sharing. The beta value is 0.639 and this represents the positive relationship between the variables. Considering this, knowledge sharing at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, an organization needs to encourage and enable Knowledge Sharing methods at the effort estimation in the DevOps environment.

Coefficients[a]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.	95.0% Confidence Interval for B Lower Bound	Upper Bound
1	(Constant)	.721	.064		11.312	<.001	.595	.846
	KnowledgeSharing	.550	.034	.639	16.144	<.001	.483	.617

a. Dependent Variable: EEStrategiesPractices

Figure 54: Coefficient Table for Knowledge Sharing

4.6.6. **Independent Variable – Skills and Experience**

The summary of the regression analysis between Skills and Experience and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 55. R and R^2 values are 0.682 and 0.465 respectively. According to the explanation in Section 2.6.6.1, 46.5% of the variance in the Success of Effort Estimation strategies and practices can be explained by Skills and Experience in this case.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.682[a]	.465	.463	.26998

a. Predictors: (Constant), SkillAndExperience
b. Dependent Variable: EEStrategiesPractices

Figure 55: Model Summary for Skills and Experience

ANOVA table for Skills & Experience and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 56. Here, the F value is 328.047 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Skills and Experience) and dependent variable (EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by Skills and Experience.

$F(1,378) = 328.047$ and $P < 0.01$

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	23.911	1	23.911	328.047	<.001[b]
	Residual	27.552	378	.073		
	Total	51.463	379			

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), SkillAndExperience

Figure 56: ANOVA Table for Skills and Experience

A coefficient table for Skills & Experience and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is depicted in Figure 57. As per the explanation in section 2.6.6.3, the regression equation is: -

EE Strategies & Practices = 0.629(Skills and Experience) +0.539

It reveals that there's a 0.629 units increase in the Success of Effort Estimation Strategies and Practices when there's a one-unit increase in Skills and Experience. The beta value is 0.682, and this represents the positive relationship between the variables. Considering this, knowledge sharing at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, the project team needs to give priority to the Skills and Experience at the effort estimation in the DevOps environment. That means expertise judgments on the estimation are acceptable.

Coefficients[a]								
		Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model		B	Std. Error	Beta	t	Sig.	Lower Bound	Upper Bound
1	(Constant)	.539	.067		8.081	<.001	.408	.670
	SkillAndExperience	.629	.035	.682	18.112	<.001	.561	.697

a. Dependent Variable: EEStrategiesPractices

Figure 57: Coefficient Table for Skills and Experience

4.6.7. **Independent Variable – Deployment Process**

The summary of the regression analysis between the Deployment Process and the Success of Effort Estimation strategies and practices in the DevOps environment is shown in Figure 58. R and R^2 values are 0.598 and 0.357 respectively. According to the explanation in Section 2.6.6.1, 35.7% of the variance in the Success of Effort Estimation strategies and practices can be explained by the Deployment Process in this case.

Model Summary[b]				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.598[a]	.357	.355	.29586

a. Predictors: (Constant), DeploymentProcess

b. Dependent Variable: EEStrategiesPractices

Figure 58: Model Summary Table for Deployment Process

ANOVA table for Deployment Process and the Success of Effort Estimation Strategies and Practices in DevOps-based software development is represented in Figure 59. Here, the F value is 209.938 when the significance level is < 0.01. Section 2.6.6.2 reveals a positive relationship between predictor (Deployment Process) and dependent variable (EE Strategies and Practices), and there is a significant model. Therefore, predictions can be made on the Success of Effort Estimation strategies and practices in the DevOps environment by the Deployment Process.

.F (1,378) = 209.938 and P < 0.01

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	18.376	1	18.376	209.938	<.001[b]
	Residual	33.087	378	.088		
	Total	51.463	379			

ANOVA[a]

a. Dependent Variable: EEStrategiesPractices

b. Predictors: (Constant), DeploymentProcess

Figure 59: ANOVA Table for Deployment Process

The coefficient table for Deployment Process and the Success of Effort Estimation Strategies and Practices in DevOps-based software development appears in Figure 60. As per the explanation in section 2.6.6.3, the regression equation is: -

Success of EE Strategies & Practices = 0.436(Deployment Process) +0.907

It reveals that there's a 0.717 units increase in the Success of Effort Estimation Strategies and Practices when there's a one-unit increase in the Deployment Process. The beta value is 0.598 and this represents the positive relationship between the variables. Considering this, the deployment process at the effort estimation will be affected by the success of effort estimation strategies and practices in the DevOps environment. Hence, the team should pay attention to its project deployment process at the effort estimation in the DevOps environment.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	.907	.058		15.565	<.001	.792	1.021
	DeploymentProcess	.436	.030	.598	14.489	<.001	.377	.495

a. Dependent Variable: EEStrategiesPractices

Figure 60: Coefficients Table for Deployment Process

4.7. Conclusion

The researcher has explained how data analysis occurred for the collected data in this chapter. Data was collected by preparing a questionnaire with the use of Google Forms.

Then the questioner was published in certain Social Media applications like Gmail Groups, and LinkedIn. Reliability analysis was conducted before publishing the questionnaire. The purpose was to verify the validity of the questions in the questionnaire. After collecting the 380 data, the researcher started to analyze the data with the IBM SPSS subscription tool. As the first data analysis, the researcher has done the descriptive analysis for the demographic data. Then the researcher was able to get an idea about the population in the study. Then, the researcher analyzed the relationship between demographic data and the dependent variable in the research study. By doing that, the researcher could get an idea about how the population move with effort estimation strategies and practices in DevOps. After that, the researcher has performed the inferential and correlation analysis. By doing the correlation analysis researcher could identify the relationship between the variables. The Inferential analysis helped to make inferences about the population with the use of sample data. Finally, the regression analysis was carried out for the collected data. It assisted in examining the impact on the dependent variable by each selected independent variable. Then the researcher could find all the independent variables which impacted the dependent variable in this research study.

5. RECOMMENDATIONS AND CONCLUSION

5.1. Chapter Introduction

The data analysis is elaborated in chapter 4, for the sake of exploring the defined objectives of the research study. The hypotheses were tested successfully and the defined objectives were analyzed subsequently. In this chapter, the conclusion and the inferences are briefed with the aid of the results in the data analysis in chapter 4. Later on, the recommendations are presented for continuing the effort estimation phase successfully. Future research plans are described to expand this research study. Finally, the events which could not be controlled by the researcher are included.

5.1.1. Research End Results based on Demographic analysis

The detailed explanation was elaborated in Section 4.4.1. It is about the demographic data analysis and understanding of the basic characteristics or features of the respondents. They were involved in this research study by filling the questionnaire. The Pearson correlation was used to identifying the relationship between the predictor and demographic variables. The correlation analysis for the demographic variables is shown in Section 4.4.2.

According to Figure 16, in Section 4.4.1, the majority of software professionals in the IT industry are males, and it was 57% of the 380 sample data. The rest of the 43% of the sample data was mentioned as female software professionals. Pearson's r was -0.386 and it is attributed in Section 4.4.2.1. The result of the two-tailed test is <0.001. Hence, the significant value is less than 0.01. Then, there is a relationship between gender (demographic variable), and the effort estimation strategies and practices in the DevOps environment (dependent variable).

The researcher tried to identify the respondents by working track-wise without sticking to the particular job roles. Hence, the researcher included the “Dev” and “Ops” into two answers. The answer Dev, included all the software engineers, Full-stack engineers, Tech Lead or Technical Architecture. The Ops included Application Engineer, System Support Engineer, DevOps Engineers, and Database administrators. When examining the current working track of the respondents in Figure 17 in Section 4.4.1, the majority of the software professionals belongs to the "Dev" track. That is, 28% out of 380. Secondly, most of the respondents belong to the "Quality Assurance" track and it was 26% of the sample data. It shows that most of the software professionals belonged to Dev in the Sri Lankan IT industry. According to the correlation analysis for the current working track in Section 4.4.2.2, Pearson's correlation coefficient (r) is -0.209 with <0.001 two-tailed significant value. Therefore, the significant value is <0.01, and there is a relationship between the current working track and the Success of Effort estimation strategies and practices in DevOps-based software development.

When comparing the work experience of the respondents, the majority of the sample had 2- 6 Years of work experience in the software industry, and it was 65% of the 380. Secondly, large respondents have attributed as 6-10 Years and it was 28% of the sample data. It is shown in Figure 18 in Section 4.4.1. This reveals that most of the software professionals belonged to the mid-experience level in DevOps-based software development in the Sri Lankan IT industry. Software professionals have some extent of experience in their carrier. Based on the correlation analysis with work experience and EE strategies and practices in the DevOps environment in Section 4.4.2.3, Pearson's correlation coefficient (r) was -0.023 with a two-tailed significant value of 0.660. Hence, the significant value > 0.05, and there is no relationship between the work experience and EE strategies and practices in the DevOps environment.

As shown in Figure 19 of Section 4.4.1, the majority of the respondents were Bachelor's degree holders. It is 73% of the 380. The second-highest respondents were 18% of the 380 sample data. They have attributed to Master's degree holders in this research study. Hence, 91% of the sample were degree holders. It indicates that there are well-educated software professionals in the Sri Lanka IT industry. When considering the correlation analysis in Section 4.4.2.4, there is no relationship between the highest education level and EE strategies and practices of the particular respondents in this industry, because the calculated Pearson's correlation coefficient (r) was -0.095 with a two-tailed significant value of 0. 063. Hence, with the significant value being > 0.05, there is no relationship between the dependent variable and the independent variable.

As per Figure 22 in Section 4.4.1, 91% of the respondents have mentioned that they are following effort estimation strategies and practices in their organization, and the rest of the 9% of the sample data have attributed as no. That means, most of the respondents in organizations are following the effort estimation strategies and practices.

According to Figure 21 in Section 4.4.1, most of the respondents have mentioned that their organizations are Product-based companies. It is 46% of the sample data. The second highest respondents have attributed it as both product and service-based. This means nowadays the software companies do not stick to the product or service based while they provide their services to customers. Most organizations follow effort estimation strategies regardless of whether the software organization is a product-based or a service-based company. When considering their opinion on effort estimation strategies and practices in the DevOps environment in Figure 23, in Section 4.4.1, 52% of the sample data of the respondents have indicated it as positive. It means that most software professionals have a positive mindset on the effort estimation strategies in the DevOps environment. Most of the software professionals are aware of nothing except the DevOps job role regarding the term called DevOps.

5.2. Consequences of the research

The main purpose of this research is to explore the factors which need to be considered at the effort estimation phase in DevOps-based software development. Then, analyze the existing techniques which are currently used at the effort estimation phase and explore the drawbacks of existing effort estimation techniques. Finally, the researcher expects to introduce an effort estimation guideline ("Best Practices") which could be used at the effort estimation phase, and it will help in predicting the reliable software project timeline. The researcher identified the seven possible factors as Poor

Communication, Measurement, Monitoring, Technology Stack, Knowledge Sharing, Skills and Experience, and Deployment Process. In this section, the researcher is going to find out the most considered factors in the success of effort estimation in the DevOps environment.

5.2.1. Hypothesis Testing: Poor Communication

$H_{1\text{-}1}$: There is **a relationship** between Poor Communication and the Success of Effort Estimation **Practices** in the DevOps Environment.

The calculated Pearson's r (r) was 0.700. The positive correlation is denoted between Poor communication and EE strategies and practices in the DevOps environment in the correlation analysis of Section 4.5.2.1. As there is a high positive relationship, the efforts can be easily measured when enabling the high communication opportunities at the estimation. The efforts generated after spending sufficient time on communication will be more accurate. Communication can be enabled in several ways such as encouraging to have team discussions, clearing all the tasks by questioning back and forth, and by using visual communication to explain difficult scenarios. When comparing other factors' correlation values, this has the second-highest correlation value. 48.9% of the variance in EE strategies is explained by Poor communication in Section 4.6.1. It means that enabling communication or in other words, avoiding poor communication hindrance will help on effort estimation.

5.2.2. Hypothesis Testing: Measurements

$H_{2\text{-}1}$: There is **a relationship** between Measurements and the Success of Effort Estimation **Practices** in the DevOps Environment.

The calculated Pearson's r (r) was 0.496. The positive correlation is denoted between Measurements and EE strategies and practices in the DevOps environment of the correlation analysis in Section 4.5.2.2. As there is a low positive relationship, the effort consumed for a particular task can be easily measured by carrying on measurement for the subtasks at the estimation. In other words, the total effort can be easily measured for a particular task, having a small chunk of measurement for subtasks. That means the main tasks are divided into small pieces of subtasks. Then, the efforts are estimated for those subtasks. Finally, the total effort of a particular task is calculated together. According to Section 4.6. 2. the variance of the measurements and effort estimation strategies is 24.6%. This means the 24.6% variance of the Success of Effort estimation strategies can be explained by measurements. The effort estimation can be successful by managing small measurements for each subtask in the main task.

5.2.3. **Hypothesis Testing: Monitoring**

H_{3-1}: There is **a relationship** between the Monitoring function and the Success of Effort Estimation **Practices** in the DevOps Environment.

The calculated Pearson's r (r) was 0.469. The positive correlation is denoted between Monitoring and EE strategies and practices in the DevOps environment of the correlation analysis in Section 4.5.2.3. As there is a positive relationship, the efforts can be easily achieved when enabling the monitoring opportunities at the estimation. 22.0% of the variance in EE strategies is explained by Monitoring in Section 4.6.3. It means that enabling monitoring or, in other words, tracking each task and subtasks will help with effort estimation in the DevOps environment.

5.2.4. **Hypothesis Testing: Technology Stack**

H_{4-1}: There is **a relationship** between the Technology Stack function and the Success of Effort Estimation **Practices** in the DevOps Environment.

The calculated Pearson's r (r) was 0.760. The positive correlation is denoted between Technology Stack and EE strategies and practices in the DevOps environment of the correlation analysis in Section 4.5.2.4. As there is a high positive relationship, the efforts can be easily measured when considering the tech-stack of the project at the estimation. The efforts which are generated after considering the tech stack in the system will be more accurate. Tech Stack can be enabled in several ways such as, by conducting technical sessions, and having tech talks to share the technical experience of the system with others. This has the highest correlation value when comparing with other correlation values of factors. 57.7% of the variance in EE strategies is explained by the Tech stack of the project in Section 4.6.4. It means that the effort estimation will be affected by the technology stack of the project. Hence, we should consider that the change is matching with the tech stack or is compatible when estimating the effort.

5.2.5. Hypothesis Testing: Knowledge Sharing

H_{5-1}: There is **a relationship** between the Knowledge Sharing function and the Success of Effort Estimation **Practices** in the DevOps Environment.

The calculated Pearson's r (r) was 0.682. The positive correlation is denoted between Knowledge Sharing and EE strategies and practices in the DevOps environment of the correlation analysis in Section 4.5.2.5. As there is a moderate positive relationship, the effort consumed for a particular task can be easily measured by carrying on measurement for the subtasks at the estimation. In other words, all the team members can measure the effort of a particular task, if they possess good knowledge about that particular area of the project. That means, all the developers and operations should know about what others are doing in the system. Then, all have the same amount of knowledge and can predict the effort time for a particular task. According to section 4.6.5. the variance of the measurements and effort estimation strategies is 40.8%. This means that the 40.8% variance of the Success of Effort estimation strategies can be explained by knowledge sharing. The entire team doesn't have an idea of what others do when the project becomes large and the number of developers increases. Hence, effort estimation can be successfully measured if everyone can share their experience once a modification is done to the project.

5.2.6. Hypothesis Testing: Skills and Experience

H6-1: There is **a relationship** between the Skills and Experience function and the Success of effort estimation **Practices** in the DevOps Environment.

The positive correlation is denoted between Skills & Experience and EE strategies and practices in the DevOps environment in the correlation analysis in Section 4.5.2.6, and the calculated Pearson's r (r) was 0.682. As there is a moderate positive relationship, the effort consumed for a particular task can be easily measured at the estimation by improving the skill. In other words, all team members have different skills when coding. Hence, those skills are useful when estimating the effort. This means, all the developers and operations should have the skill of working on any kind of situation and any part of the project in the system. Then, all have that same knowledge and can predict and measure the effort time for a particular task. According to section 4.6.6, the variance of the skills and experience, and effort estimation strategies is 46.5%. This means that the 46.5% variance of the Success of Effort estimation strategies can be explained by skills and experience. The entire team doesn't have an idea of what others do when the project becomes large and the number of developers increases. Hence, effort estimation can be successfully measured if everyone can share their experience once a modification is done to the project.

5.2.7. Hypothesis Testing: Deployment Process

H7-1: There is **a relationship** between the Deployment Process function and the Success of Effort Estimation **Practices** in the DevOps Environment.

The positive correlation is denoted between Deployment Process and EE strategies and practices in the DevOps environment in the correlation analysis in Section 4.5.2.7, and the calculated Pearson's r (r) was 0.598. As there is a moderate positive relationship, the effort consumed for a particular task can be easily measured at the estimation by enabling the deployment process in the system. In other words, if the project has conveyed a deployment process, then time and effort consumed for the releases could not be considered. Because there's no requirement for the occupation of human resources for deployment. This means, the deployment process saves the time of continuous deploying and delivering effort and time. According to section 4.6.7, the variance of the deployment process and effort estimation strategies is 35.7%. This means that the 35.7% variance of the Success of Effort estimation strategies can be explained by the deployment process.

5.3. Factor Analysis Output

As shown in the correlation analysis in Section 4.5.2, two factors are most influenced by the dependent variable, the Success of Effort Estimation strategies, and practices in DevOps. Those factors are, poor communication and Tech stack. Knowledge sharing, skills & experience, and deployment process are moderately impacted on effort estimation strategy and practices in DevOps-based software development. Measurements and monitoring are a low impact on the success of effort estimation strategies and practices in DevOps-based software development. Figure 61 shows the strength of the relationship of each factor with the dependent variable. This means, the strength of the correlation between factors and dependent variable. The colours which are used for independent variables shows the strength of the relationship. High correlation is represented by dark blue. The blue colour for moderate correlation. The low correlation is denoted by light blue.

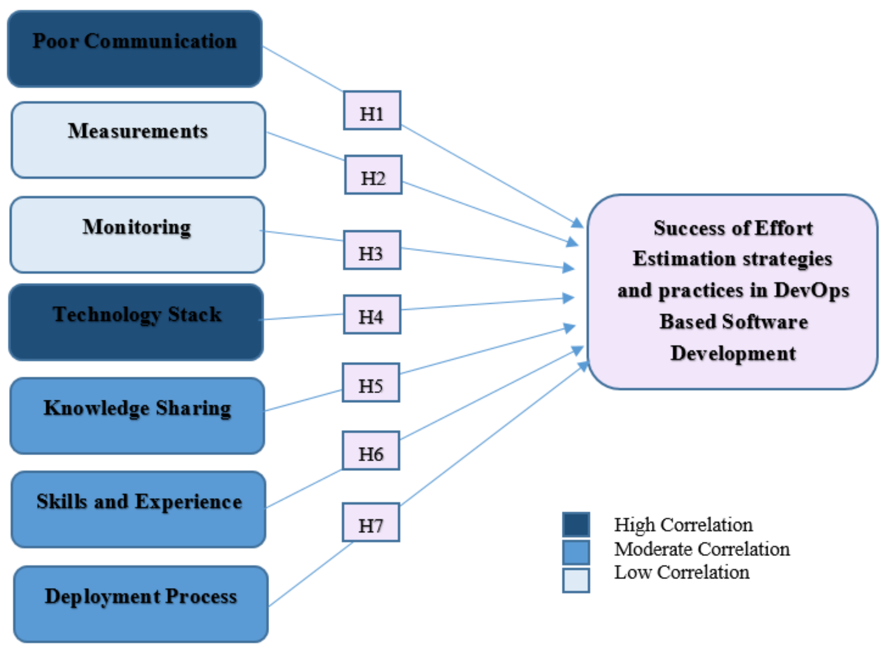

Figure 61: Summary of Factor Analysis

5.4. Recommendations

5.4.1. Establishing sufficient time on communicating effectively at the effort estimation stage.

Effective Communication is one of the most important factors affecting effort estimation in the DevOps environment. As (Lwakatare, Karvonen, et al., 2016) stated, effective communication is required when working on new features. Because, when people work in several modules they don't have an idea about what takes place outside the project. It means people work silos in their modules and don't know what others are doing. Sometimes, several developers do the same thing in different modules. They are unaware of it because of poor communication. Hence, people need to communicate effectively in the estimation stage. There is a high positive correlation between poor communication and the success of effort estimation strategies and practices in the DevOps environment in Section 4.5.2.1. It is stated that 48.9% of the variance in the success of effort estimation strategies and practices can be explained by poor communication in Section 4.6.1. The factor called poor communication has been identified as a subfactor under the collaboration aspects by the researcher. Hence, the developers can easily estimate the effort by enhancing communication and allowing sufficient time for communication at the effort estimation stage in an effective way. It will help to deliver projects on time and win over the customers. When comparing the remaining seven factors, poor communication plays a vital role in effort estimation. In other words; collaboration aspects should be improved in a team at the effort estimation. According to the works of literature, the researchers have proved that the collaboration should be improved within and between the teams to accelerate the deliveries in the DevOps environment.

Allowing proper communication and improving the communication at the estimation were the answers of 95% of the respondents in DevOps culture. It is shown clearly when considering the responses of the questions like "The scope of the assigned tasks can be easily understood if we improve the communication skills", "Proper communication helps to improve the motivation to fulfill a task in DevOps practice" in the questionnaire.

Especially, the team should be encouraged to communicate the related areas such as "What are the impacted areas", "What sort of development they are going to do", "Similar types of development have done before" of the tasks. Then it would help to increase others' knowledge regarding the development process. Improving communication can be done at the teams' level and organization level. For that, the team or organization can follow below practices:

- Convincing the employees at the interviews that the communication at the estimation is highly considered.
- Reminding team members, the importance of communication at estimation before every estimation cycle.
- Let all team members be involved in effort estimation and communicating.
- Expertise should give their honest feedback on communication to convey the estimation stage effectively.

5.4.2. Encouraging to maintain the measurements and matrices at the Effort Estimation.

Another important aspect is measuring the important objects which are related to effort estimation at the estimation stage. Here, the researcher considered the numerical values or relative sizes for the events or objects as the measurement. It simply states out the maintaining the matrices at the estimation. According to (Kerzazi & Adams, 2016) and (Nicolau de França et al., 2016), measurement is a key component and it is started from collecting systematic metrics. It helps to make decisions in software development. Measurement is understanding the code and environment when considering the importance of effort estimation. For example, if the team is discussing the effort of a task, all the team members should be participating in the estimation by break downing those tasks into small pieces such as identifying the frontend /back end development separately, database changes, deciding the library usage, etc.

Then, estimate the effort for those small pieces with the use of metrics. Finally, calculate the total effort of a particular task, by getting the summation of the total efforts of small pieces. If everyone uses their matrices to estimate the effort, it will be easy to identify the easiest way to develop that particular task. By comparing measured estimations, the team can decide the proper way to develop that task. There is a positive correlation between measurement and the success of effort estimation strategies and practices in the DevOps culture of the correlation analysis in Section 4.5.2.2. As per Section 4.6.2, there is 24.6% of the variance in the success of effort estimation strategies and practices can be explained by measurement. The factor called measurement is another important aspect of effort estimation. Hence, the developers can easily estimate the effort by breaking down the main task into small tasks. If the team can introduce the proper guideline for keeping the measurements, then the team can follow the same protocol when doing estimation. If the team can identify the common structure for the estimation matrices, they can follow that structure to estimate. This would eventually make estimation easier for all.

Introducing the Measuring Tools and well-defined Measuring Guidelines at the estimation, and giving the structure of effort estimation metrics were the answers of

92% of the respondents in DevOps culture. It is shown clearly by the responses of the questions like "The use of Measuring Tools and well-defined Measuring Guidelines, make it easy to estimate the required effort in the DevOps practice.", "Given the structure of effort estimation metrics make it easy estimation in DevOps practice" in the questionnaire.

The following practices can be followed by the Dev and Ops in DevOps culture at the estimation stage:

- Advise team members to follow estimations guidelines.
- Convincing them to follow effort estimation matrices at the effort estimation stage.
- Identify the estimation gap between each other by comparing each ones' estimation for a particular task.

5.4.3. **Introducing the monitoring tools and guidelines.**

Monitoring is another important aspect that affects effort estimation in DevOps culture. As (Ebert et al., n.d.) mentioned, the monitoring tools let the identified infrastructure problems be resolved before they affect the working business process. Hence, monitoring is playing a big role in DevOps culture. Many researchers have proved that continuous monitoring is more helpful to fast project deliveries since developments are happening over time in DevOps culture. There is a positive correlation between monitoring and the success of effort estimation strategies and practices in the DevOps environment in Section 4.5.2.3. It is stated, 22.0% of the variance in the success of effort estimation strategies and practices can be explained by the monitoring in Section 4.6.3.

Also, the responses of the questions such as "Monitoring development milestones helps to determine the effectiveness of the Dev team", "Involvement of Technical Expertise will be an added advantage in the effort estimation.", "The impacted areas of development can be identified by Continuous Monitoring of the system", and over 85% of the respondents believe that if the team monitor the application continuously, they can find the issues before it occurs in the production environment, and it is helped to resolve those issues on board.

5.4.4. Enabling tech-supportive environment.

Most of the software and IT industries strive to be fast and efficient. DevOps has come into the industry as a model which brings innovative products and new features faster to the market. There are several technologies available for the transition between development and operations in the market. As mentioned in (Perera, Bandara, et al., 2017), all members should be accountable for their duties and their undelaying technical aspects from the start to the end in the production model. Hence, everyone should be conscious of the end to end technical backgrounds of the project. Therefore, technical expertise should have participated in the effort estimation phase. The developers ponder on how to conduct the development part once the new requirements come in. At that instant, they should think about the compatibility of the particular change with the existing technology stack. If that change is not compatible with the tech stack, developers need to find a substitute solution for that. Hence, technical expertise should participate in effort estimation.

There is a high positive correlation between technology-stack and the success of effort estimation strategies and practices in the DevOps environment in Section 4.5.2.4, and it is stated that 57.7% of the variance in the success of effort estimation strategies and practices can be explained by the technology stack in Section 4.6.4. The factor called technology-stack has been identified as a subfactor under the socio-technical aspects by the researcher. Hence, the developers can easily estimate the effort if they have an idea about the technology stack on the project. If all team members possess a good understanding of the structure-wise frontend-backend of the application, the team can easily estimate the efforts. When comparing with the other seven factors, the technical aspects play a vital role in effort estimation. If stated in other terms; the socio-technical aspects should be improved in the team at the effort estimation. Knowledge sharing and Skills and Experience are the other two aspects identified by the researcher, under the socio-technical aspects. According to the works of literature, the researchers have proven that the DevOps team requires members with multiple skills. Hence, the DevOps team should comprise all available techs in the projects.

Also, the responses of the questions such as “The project Requirements and Enhancements should be compatible with the Tech-Stack (Set of tools, frameworks, and libraries) of the application.”, “Involvement of Technical Expertise will be an added advantage in the effort estimation.”, “Having a good understanding of the Tech-Stack (Structure Wise-Frontend/Backend) of the application to the team will make it easier to estimate the effort.”, 90% of the respondents believe that if the team fully understands the tech stack in the project, they can easily estimate effort.

Therefore, the team can follow the below practices to enable a tech-supportive environment at the estimation.

- Try to find the ready-made (Ideally suited) solution for new requirements and enhancements.
- Create a group of knowledgeable persons who possess a good understanding on the Tech-Stack (Structure Wise-Frontend/Backend).
- Always validate new changes that are going to be implemented in the project by the tech expertise.
- Tech expertise should guide the estimation phase.

5.4.5. Establish Knowledge Sharing and a fully skilled platform.

The positive correlation between knowledge sharing and effort estimation strategies and practices in the DevOps environment is shown in Section 4.5.2.5. Section 4.5.2.6 elaborates a positive correlation between skills and experience and the success of effort estimation strategies and practices in the DevOps environment. It is stated that 40.8% of the variance in the success of effort estimation strategies and practices can be explained by knowledge sharing in Section 4.6.5. Also, 46.5% of the variance in the success of effort estimation strategies and practices can be explained by skills and experience in Section 4.6.6. The factors called knowledge sharing and skills and experience are identified as subfactors under the socio-technical aspects by the researcher. As mentioned in (Hemon, Fitzgerald, et al., 2020), knowledge sharing is considered as one of the four pillars of DevOps culture. Hence, it is an important aspect in the effort estimation stage as well. Knowledge Sharing occurs when someone shares what he knows with others. If all team members are well aware of the application, the effort can be easily estimated. (Nielsen et al., 2017) has identified the DOKS framework to distribute the knowledge among the teammates.

It is shown by the responses of the questions such as, "The use of prior Skills and Experiences would help to measure the effort in DevOps practice.", "Sharing Research and Development (R&D) experiences in the application makes it easy to estimate the effort in the DevOps practice.". 90% of respondents believe that sharing the knowledge of the project activities helps in effort estimation and continuous deliveries. The below practices can be followed when sharing the knowledge:

- Sharing the tactic knowledge such as experience and feelings with the team.
- Allowing to share the knowledge through meetings that can be held with the participation of customers or IT Development and IT Operation teams (Scrum meeting and Sprint Retrospective meeting).
- Helping to conduct the Technical sessions, Tech Talks within the team and with the participating industry experts.

The responses of the questions such as "Having Skills and Experience in a specific area in development helps me in effort estimation personally", "The use of prior Skills and Experiences in the specific programming language would help to measure the effort easily in the DevOps context" reveal that 90% of the respondents believe if they have skills and experience in different technology backgrounds, they can easily estimate efforts.

5.4.6. **Automating the repetitive activities in the application**

The positive correlation between the deployment process and the success of effort estimation strategies and practices in the DevOps environment is shown in Section

4.5.2.7. It is stated that 57.7% of the variance in the success of effort estimation strategies and practices can be explained by the technology stack in Section 4.6.7. The factor called deployment process has been identified as a subfactor under the automation aspects by the researcher. As stated in (Nicolau de França et al., 2016) and (Bucena & Kirikova, 2017), the DevOps culture aims to automate all repetitive activities such as deployment process, scripting, and all operational tasks, delivering process and quality assurance process, etc. This means that arranging the flexible and lean process in the team is highly considered in DevOps culture. By automating every possible activity, software projects can easily continue without technical issues. It is shown by the responses of the questions such as "An automated deployment process helps to reduce the effort and risks in the production environments.", "The Automated deployment process reduces the integration time in the DevOps environment.". "The Automated deployment process allows continuous delivery and makes effort estimation easy in a DevOps environment.". 90% of respondents believe that automating the project activities help in effort estimation and continuous deliveries.

Hence, the team should follow the below practices:

- Automating all analysis of the application. For example: continuous monitoring during the testing and continuous integration and perform it on an automated dashboard.
- Enabling automated feedbacks for performance prediction and performance models.
- Integrating the deployment plans and automating continuous deployment and continuous delivery.
- Managing the changes and integrating the change management mechanism.

5.5. Research Limitations

The researcher had to select the most important seven factors out of nineteen factors to continue this research study. Many other factors need to be considered when doing effort estimation in DevOps culture.

When doing the pilot survey, the researcher understood that the majority of the respondents are not aware of the DevOps culture even though they are working in a DevOps environment. However, every respondent is well aware of the role of "DevOps engineer". This may be a reason for some independent variables not showing the desired output with the dependent variable at the analysis. The majority of the respondents have a positive idea about the success of effort estimation strategies and practices in DevOps-based software development. It means that they are aware that DevOps-based software development provides a more flexible and easy way for the completion of the projects. The sample size was 380 for this research study, and it would be better if the researcher could find more than 450 of the sample size because the sample size represents all job categories in the Sri Lankan IT Industry. Since there are 41 questions in the questionnaire, the researcher tried to reduce them without creating an effect on the Cronbach's Alpha value. But, it seems the respondents showed disinterest in the middle of the questionnaire. The researcher has considered only the perceived value for the success of effort estimation strategies and practices in DevOps-based software development.

5.6. Future Directions in the research

There are multiple ways to expand this research study as future work. Since this research represents all personnel in the job categories who have work experience in DevOps-based software development, the same research can be performed from different perspectives such as productivity in DevOps culture. For example:

- Challenges and barriers when adopting the DevOps culture in the Sri Lankan context.
- How the company scale (Small, Medium, and Large) is impacted by the adoption of the DevOps culture?
- DevOps advances in the release management process.
- How to enable DevOps in the startups over the Sri Lankan context?

Since the sample size is the most important aspect for the credibility of the results of a quantitative research study, it would be better to increase the sample size in future work. There is a problem with the measuring of the strategies in this research study since the definition of that term is not declared at the beginning of this research study. Another important aspect that the researcher has identified during the data analysis period was, the respondents not having a clear idea about the term called “DevOps”. The only term that they are familiar with is DevOps Engineers. Even though some of the respondents’ organizations are doing DevOps-based software development, they don’t have a clear picture of the DevOps culture.

5.7. Conclusion

According to the comprehensive analysis and the discussion in this research study, poor communication, measurement, monitoring and tech stack, knowledge sharing, skills and experience impact to the effort estimation in DevOps based software development in the Sri Lankan context. Even though there were several studies related to the effort estimation techniques and methods, there was no proper guideline for the practices which should be followed at the effort estimation stage in the DevOps context. Hence, this research focuses on identifying the factors which need to be considered at the stage of effort estimation. Communication within the team and applications’ technology stack are highly considered at the effort estimation phase. Thus, knowledge sharing, skills and experience, and deployment process are moderately impacted to the effort estimation in DevOps based software development. Also, according to this research study, the measurements and monitoring show a low impact on effort estimation. This research study answers the questions of how to communicate, how to improve the skills and experience by knowledge sharing and continuing that practices in the effort estimation. Later, the researcher has proposed some guidance based on the final analysis results and data. Those suggestions are for improving the communication skills, sharing the knowledge, enhancing the experiences, and enabling the automation process for every repetitive activity in the process. If each person in the team is well aware of the technology stack in the application, they can easily estimate the effort. Also, monitoring and measurements are play a big role in the effort estimation step. Besides that, the research limitation and future research directions are also expressed in this chapter.

REFERENCES

Adom, D., & Hussain, E. K. and Joe, A. (2018). Theoretical and Conceptual Framework: Mandatory Ingredients of a Quality Research. *International Journal of Scientific Research*, *7*(1), 93–98.

Ainsworth, Q. (2019, August 27). *Data Collection Methods*. https://www.jotform.com/data-collection-methods/

Al-achi, A. (2019). The Student's t-Test : A Brief Description. *Journal of Hospital and Clinical Pharmacy*, *5*(1), 4–6.

Artač, M., Borovšak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2016). Model-Driven continuous deployment for quality DevOps. *2nd International Workshop on Quality-Aware DevOps, pp.* 40–41. DOI: https://doi.org/10.1145/2945408.2945417

Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2017). DevOps: Introducing infrastructure-as-code. *IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 497–498. DOI: https://doi.org/10.1109/ICSE-C.2017.162

Asiamah, N., Mensah, H. K., & Oteng-Abayie, E. F. (2017). General, target, and accessible population: Demystifying the concepts for effective sampling. *Qualitative Report*, *22*(6), 1607–1621.

Bordin S., De Angeli A. (2016). Focal Points for a More User-Centred Agile Development. In: Sharp H., Hall T. (eds) Agile Processes, in Software Engineering, and Extreme Programming. *17th international conference on International Conference on Agile Software Development (XP 2016),* Lecture Notes in Business Information Processing, vol 251. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-33515-5_1

Britto, R., Freitas, V., Mendes, E., & Usman, M. (2014). Effort estimation in global software development: A systematic literature review. *9th International Conference on Global Software Engineering (ICGSE),* pp. 135–144. DOI:

https://doi.org/10.1109/ICGSE.2014.11

Britto, R., Mendes, E., & Wohlin, C. (2016). A specialized global software engineering taxonomy for effort estimation. *11th IEEE International Conference on Global Software Engineering (ICGSE)*, PP. 154–163. DOI: https://doi.org/10.1109/ICGSE.2016.11

Bucena, I., & Kirikova, M. (2017). *Simplifying the DevOps Adoption Process Challenges of DevOps Adoption*. BIR Workshops, 1-15.

Chen, B. (2019). Improving the software logging practices in DevOps. *IEEE/ACM 41st International Conference on Software Engineering: Companion*, pp. 194–197. DOI: https://doi.org/10.1109/ICSE-Companion.2019.00080

Chung, S. (2017). Object-oriented programming with DevOps. *18th Annual Conference on Information Technology Education (SIGITE)*, 65. https://doi.org/10.1145/3125659.3125670

Debbiche, F. Wrang, M., & Sinkala, K. (2019). *Accelerating Software Delivery in the context of Requirements Analysis and Breakdown for DevOps: A multiple-case study Bachelor*. Thesis, University of Gothenburg.

Díaz, J., Almaraz, R., Pérez, J., & Garbajosa, J. (2018). DevOps in practice - An exploratory case study. *ACM International Conference Proceeding Series*, *Part F1477*, 18–20. DOI: https://doi.org/10.1145/3234152.3234199

Dudovskiy, J. (2018). *Data Collection Methods - Research-Methodology*. Business Research Methodology, https://research-methodology.net/research-methods/data-collection/

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*. 33(3), 94-100, DOI: https://doi.org/10.1109/MS.2016.68

Edureka. (2020). *DevOps Roles*, https://www.edureka.co/blog/devops-roles-which-is-your-dream/

El Bajta, M. (2015). Analogy-based software development effort estimation in global software development. *10th International Conference on Global Software*

Engineering Workshops (ICGSEW), pp. 51–54. DOI: https://doi.org/10.1109/ICGSEW.2015.19

Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, *29*(6), 1–20. DOI: https://doi.org/10.1002/smr.1885

Fávero, E. M. D. B., Pereira, R., Pimentel, A. R., & Casanova, D. (2018). Analogy-based Effort Estimation: A Systematic Mapping of Literature. *Infocomp*, *17*(2), 7–22.

Forsgren, N., & Kersten, M. (2018). DevOps metrics. *Communications of the ACM*, *61*(4), 44–48. DOI: https://doi.org/10.1145/3159169

Grant, C., & Osanloo, A. (2014). Understanding, Selecting, and Integrating a Theoretical Framework in Dissertation Research: Creating the Blueprint for Your "House." *Administrative Issues Journal Education Practice and Research*, *4*(2), 12–26. DOI: https://doi.org/10.5929/2014.4.2.9

Garusinghe, A., Perera, I., & Meedeniya, D. (2017). Service oriented product lines - managed service level agreements for better quality of service. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 10(2), 1-11. DOI: http://doi.org/10.4038/icter.v10i2.7184

Hemon, A., Fitzgerald, B., Lyonnet, B., & Rowe, F. (2020). Innovative Practices for Knowledge Sharing in Large-Scale DevOps. *IEEE Software*, *37*(3), 30–37. DOI: https://doi.org/10.1109/MS.2019.2958900

Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*, *22*(4), 927–945.DOI: https://doi.org/10.1007/s10796-019-09905-1

Ho, R. (2019). Introduction to Hypothesis Testing. *Understanding Statistics for the Social Sciences with IBM SPSS*, Chapman and Hall/CRC, 203–214. DOI: https://doi.org/10.4324/9781315182452-12

Hussain, W., Clear, T., & MacDonell, S. (2017). Emerging trends for global

DevOps: A New Zealand perspective. *12th International Conference on Global Software Engineering (ICGSE)*, PP. 21–30. DOI: https://doi.org/10.1109/ICGSE.2017.16

Hussaini, S. W. (2014). Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through a systems approach. *17th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 178–183. DOI: https://doi.org/10.1109/ITSC.2014.6957687

ICTA. (2019). National IT-BPM Workforce Survey 2019. *National IT - Bpm Workforce Survey*, https://nvq.gov.lk/LMI_Bulletin/2019_Vol_I/files/basic-html/page1.html.

Idri, A., Abran, A., & Khoshgoftaar, T. M. (2002). Estimating software project effort by analogy based on linguistic values. *International Software Metrics Symposium*, *pp.* 21–30. DOI: https://doi.org/10.1109/METRIC.2002.1011322

Idri, Ali, Abnane, I., & Abran, A. (2018). Support vector regression-based imputation in analogy-based software development effort estimation. *Journal of Software: Evolution and Process*, *30*(12), 1–23. DOI: https://doi.org/10.1002/smr.2114

Jabbari, R., Ali, N. Bin, Petersen, K., & Tanveer, B. (2016). What is DevOps? A systematic mapping study on definitions and practices. *ACM International Conference Proceeding Series*, *24-May-201*, 1–11. DOI: https://doi.org/10.1145/2962695.2962707

Jankowski, K. R. B., Flannelly, K. J., & Flannelly, L. T. (2018). The t-test: An Influential Inferential Tool in Chaplaincy and Other Healthcare Research. *Journal of Health Care Chaplaincy*, *24*(1), 30–39. DOI: https://doi.org/10.1080/08854726.2017.1335050

Jilcha Sileyew, K. (2020). Research Design and Methodology. In *Cyberspace*, pp. 1–12. DOI: https://doi.org/10.5772/intechopen.85731

Jones, S., Noppen, J., & Lettice, F. (2016). Management challenges for DevOps adoption within UK SMEs. *2nd International Workshop on Quality-Aware*

DevOps (QUDOS), pp. 7–11. DOI: https://doi.org/10.1145/2945408.2945410

Kaur, J. (2015). Techniques Used in Hypothesis Testing in Research Methodology – A Review. *International Journal of Science and Research (IJSR), 4*(5), 362–365.

Kerzazi, N., & Adams, B. (2016). Who needs release and DevOps engineers, and why?. *International Workshop on Continuous Software Evolution and Delivery, CSED, 1*, pp. 77–83. DOI: https://doi.org/10.1145/2896941.2896957

Khan, K., & Araghinejad, S. (2010). The Evaluation of Well-known Effort Estimation Models based on Predictive Accuracy Indicators. *Measurement*, 213–251. DOI: https://doi.org/10.1007/978-94-007-7506-0_7

Krejcie, R. V, & Morgan, D. (1970). *Determining Sample Size for Research Activities*. *Educational and Psychological Mesurement*, 30, 607–610.

Laerd. (2018). *Understanding Descriptive and Inferential Statistics | Laerd Statistics*. Laerd Statistics. https://statistics.laerd.com/statistical-guides/descriptive-inferential-statistics.php

Laukkarinen, T., Kuusinen, K., & Mikkonen, T. (2017). DevOps in regulated software development: Case medical devices. *IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track (ICSE-NIER)*, pp. 15–18. DOI: https://doi.org/10.1109/ICSE-NIER.2017.20

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys, 52*(6). DOI: https://doi.org/10.1145/3359981

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of DevOps, in: Lassenius, C., Dingsøyr, T., & Paasivaara, M. (eds) Agile Processes, in Software Engineering, and Extreme Programming. *16th international conference on International Conference on Agile Software Development (XP 2015),* Lecture Notes in Business Information Processing, vol 250. Springer, Cham. Helsinki, Finland, pp. 212–217. DOI: https://doi.org/10.1007/978-3-319-18612-2

Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., & Oivo, M. (2016). Towards DevOps in the embedded systems domain: Why is it so hard?. *Annual Hawaii International Conference on System Sciences*, 5437–5446. DOI: https://doi.org/10.1109/HICSS.2016.671

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices. 11th International Conference on Software Engineering Advances (*ICSEA)*, pp. 91–99.

Meedeniya, D., Rubasinghe, I., & Perera, I. (2019a). Software Artefacts Consistency Management towards Continuous Integration: A Roadmap. International Journal of Advanced Computer Science and Applications, 10(4), 100-110. https://doi.org/10.14569/IJACSA.2019.0100411

Meedeniya, D., Rubasinghe, I., & Perera, I. (2019b). Traceability Establishment and Visualization of Software Artefacts in DevOps Practice: A Survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *10*(7), 66 - 76. DOI: http://dx.doi.org/10.14569/IJACSA.2019.0100711

Meedeniya, D., Rubasinghe, I., & Perera, I. (2020). Artefact Consistency Management in DevOps Practice: A Survey. Pendyala, V., (Eds)., in *Tools and Techniques for Software Development in Large Organizations*, ch. 4, 98-129. Hershey, PA: IGI Global. DOI: http://dx.doi.org/10.4018/978-1-7998-1863-2.ch004

Menzies, T., Chen, Z., Hihn, J., & Lum, K. (2006). Selecting best practices for effort estimation. *IEEE Transactions on Software Engineering*, *32*(11), 883–895. DOI: https://doi.org/10.1109/TSE.2006.114

Mugizi, W. (2019). Constructing a Conceptual Framework for Quantitative Data Analysis in Social Science Research. *Interdisciplinary Journal of Education*, 2(1), 74 - 88. DOI: https://journals.iuiu.ac.ug/index.php/ije/article/view/77

Nicolau de França, B. B., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by hearing multiple voices. *ACM International Conference Proceeding Series*, 53–62. DOI: https://doi.org/10.1145/2973839.2973845

Nielsen, P. A., Winkler, T. J., & Nørbjerg, J. (2017). Closing the IT Development-Operations Gap : The DevOps Knowledge Sharing Framework. *BIR Workshop,* 1-15.

Obilor, E. I., & Amadi, E. C. (2018). Test for significance of Pearson's correlation coefficient (r). *International Journal of Innovative Mathematics, Statistics & Energy Policies*, *6*(1), 11–23.

Owais, M., & Ramakishore, R. (2017). The effort, duration, and cost estimation in agile software development. *9th International Conference on Contemporary Computing (IC3)*. Noida, India, 2016, pp. 1-5, DOI: https://doi.org/10.1109/IC3.2016.7880216

Palihawadana, S., Wijeweera, C. H., Sanjitha, M. G. T. N., Liyanage, V. K., , Perera, I., & Meedeniya, D. (2017). Tool support for traceability management of software artefacts with DevOps practices. *Moratuwa Engineering Research Conference (MERCon),* Colombo, Sri Lanka, pp. 129-134. DOI: 10.1109/MERCon.2017.7980469

Penners, R., & Dyck, A. (2015). Release Engineering vs. DevOps - An Approach to Define Both Terms. *Full-Scale Software Engineering (FsSE)*, 49–54.

Perera, P., Bandara, M., & Perera, I. (2017). Evaluating the impact of DevOps practice in Sri Lankan software development organizations. *16th International Conference on Advances in ICT for Emerging Regions (ICTer),* pp. 281–287. DOI: https://doi.org/10.1109/ICTER.2016.7829932

Perera, P., Silva, R., & Perera, I. (2017). Improve software quality through practising DevOps. *17th International Conference on Advances in ICT for Emerging Regions (ICTer),* pp. 13–18. DOI: https://doi.org/10.1109/ICTER.2017.8257807

Pérez, J. F., Wang, W., & Casale, G. (2015). Towards a DevOps approach for software quality engineering. *ACM/SPEC Workshop on Challenges in*

Performance Methods for Software Development (WOSP-C), pp.5–10. DOI: https://doi.org/10.1145/2693561.2693564

Phannachitta, P. (2018). Robust comparison of similarity measures in analogy-based software effort estimation. *International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA)*. Malabe, Sri Lanka,, pp. 1-7. DOI: https://doi.org/10.1109/SKIMA.2017.8294126

Popli, R., & Chauhan, N. (2014). Cost and effort estimation in agile software development. *International Conference on Reliability, Optimization and Information Technology (ICROIT),* pp. 57–61. DOI: https://doi.org/10.1109/ICROIT.2014.6798284

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

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. *International Conference on Product-Focused Software Process Improvement (PROFES).* LNCS 10027, 590–597. DOI: https://doi.org/10.1007/978-3-319-49094-6_44

Roche, J. (2013). Adopting DevOps Practices in Quality Assurance - Merging the art and science of software development. *Communications of the ACM*, 56(11), 1–8. DOI: https://doi.org/10.1145/2524713.2524721

Rong, G., Gu, S., Zhang, H., & Shao, D. (2017). DevOpsEnvy: An Education Support System for DevOps. 3*0th IEEE Conference on Software Engineering Education and Training (CSEE and T),* Savannah, GA, USA, pp. 37-46. DOI: https://doi.org/10.1109/CSEET.2017.17

Rubasinghe, I., Meedeniya, D., & Perera, I. (2017). Towards Traceability Management in Continuous Integration with SAT-Analyzer. *3rd International Conference on Communication and Information Processing (ICCIP 2017),* Tokyo, Japan, pp. 77-81. DOI: https://doi.org/10.1145/3162957.3162985

Rubasinghe, I., Meedeniya, D., & Perera, I. (2018a). Automated Inter-artefact Traceability Establishment for DevOps Practice. *17th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2018),* Singapore, pp. 211-216. DOI 10.1109/ICIS.2018.8466414

Rubasinghe, I., Meedeniya, D., & Perera, I. (2018b). Traceability Management with Impact Analysis in DevOps based Software Development. *International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, pp. 1956–1962. DOI: https://doi.org/10.1109/ICACCI.2018.8554399

Rubasinghe, I., Meedeniya, D., & Perera, I. (2020). Tool Support for Software Artefact Traceability in DevOps Practice: SAT-Analyser. Pendyala, V., (Eds)., in *Tools and Techniques for Software Development in Large Organizations*, ch. 5, 130-167. Hershey, PA: IGI Global. DOI: http://dx.doi.org/10.4018/978-1-7998-1863-2.ch005

Sabrjoo, S., Khalili, M., & Nazari, M. (2015). Comparison of the accuracy of effort estimation methods," *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 724-728, DOI: https://doi.org/10.1109/KBEI.2015.7436134.

Sawyer, S. F. (2009). Analysis of Variance: The Fundamental Concepts. *Journal of Manual & Manipulative Therapy*, *17*(2), 27E-38E. DOI: https://doi.org/10.1179/jmt.2009.17.2.27e

Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps capabilities, practices, and challenges: Insights from a case study. *22nd International Conference on Evaluation and Assessment in Software Engineering*, pp. 57–67, DOI: https://doi.org/10.1145/3210459.3210465

SLASSCOM. (2014). *Sri Lankan IT/BPM Industry 2014 Review*. 15. DOI: http://www.slasscom.lk/content/sri-lankan-it-bpm-industry-review-2014

stattrek.com. (2019). *How to Test Hypotheses*. *February*. DOI: https://stattrek.com/hypothesis-test/how-to-test-hypothesis.aspx

Taherdoost, H. (2017). Determining sample size; How to calculate the survey sample size. *International Journal of Economics and Management Systems*, *2*(2), 237–239. DOI: http://www.iaras.org/iaras/journals/ijems

Taherdoost, H. (2018). Validity and Reliability of the Research Instrument; How to Test the Validation of a Questionnaire/Survey in a Research. *SSRN Electronic Journal*, *5*(3), 28–36. DOI: https://doi.org/10.2139/ssrn.3205040

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International Journal of Medical Education*, *2*, 53–55. DOI: https://doi.org/10.5116/ijme.4dfb.8dfd

Trochim, W. (2020). Research Methods Knowledge Base - Inferential Statistics. *The SAGE Glossary of the Social and Behavioral Sciences*, pp. 4–5. DOI: https://conjointly.com/kb/inferential-statistics/

Uyanık, G. K., & Güler, N. (2013). A Study on Multiple Linear Regression Analysis. *Procedia - Social and Behavioral Sciences*, *106*, 234–240. DOI: https://doi.org/10.1016/j.sbspro.2013.12.027

Wilkinson, D., & Peter, B. (2003). *Using Research Instruments - A Guide for Researchers*. Routledge; 1st edition.

APPENDIX

Appendix A: Pre Survey

Questions asked at the preliminary interview

Job Role: - ………………….

1. What is your perception regarding the “DevOps”?
2. Have you had work experience in the DevOps environment? (Yes/No)
3. Please indicate what types of effort estimation is practiced in your organization in the DevOps environment?
    a. (e.g.: - Expert Based
        i. Bottom-Up
        ii. Top Down
        iii. Analogy Based
        iv. SLIM- Software Life Cycle Model
        v. COCOMO 1
        vi. COCOMO 2)
4. Are you aware of any of the effort estimation strategies and practices in the DevOps context?
5. What are the tools/techniques that are used to estimate the effort in a DevOps environment?
6. What kind of efforts that your team used for estimating?
    a. (e.g.: - Duration, Size)
7. What will you do from the obtained effort estimation results?
8. How do you use the results to estimate the effort of a particular task?
9. What are the practices (what actually do) that you followed in the effort estimation phase in the DevOps context?
10. Do you see any issues or drawbacks of existing effort estimation strategies? If “yes”, mention what they are (each of them)?
11. What do you suggest as aspects that we need to consider in the effort estimation phase in the DevOps environment?

12. What do you think about the following factors? How far do they affect the effort estimation in the DevOps environment? Can you state the importance of each one of them?

- Collaboration aspects
    - Poor Communication
    - Natural Communication
- Monitoring the functions in DevOps
- Measurements (Useful metrics) in DevOps
- Organization Culture
    - Organization silos
    - Information Flow in the organization
    - Empathy (ability to understand and share the feelings of another)
    - Unsatisfactory environment
- Socio-Technical aspects
    - Technology Stack
    - Skills and Experience
    - Knowledge Sharing
    - Programming Language
    - Importance of the other tasks
    - IT operation not involved in Other tasks
- External Pressure
    - Customer Expectation
    - Customer Usage of the software
    - Responsiveness to customer needs
- Automation technology
    - Deployment Process
    - Infrastructure as Code

APPENDIX B: Pilot Survey

#	Question	Poor Communication	Measurements	Monitoring	Technology Stack	Knowledge Sharing	Skills & Experience	Deployment Process	E. E Strategies Practices in DevOps
9	The use of communication tools and well-defined communication guidelines make it easy to estimate the required effort in the DevOps practice.	×							×
10	The scope of the assigned tasks can easily understand improving communication skills.	×							
11	Proper communication helps to improve the motivation to fulfil a task in DevOps practice.	×							
12	The accuracy of the effort estimation in DevOps practice can be improved by allocating sufficient time to communicate with the collaborators.	×							
13	Proper communication at the stage of effort estimation defines a lightweight process to accomplish a task.	×							
14	The use of Measuring tools and well-defined Measuring guidelines, make it easy to estimate the required effort in the DevOps practice.		×						
15	Daily Effort Measurements will increase effectiveness.		×						
16	The given structure of effort estimation metrics make it easy estimation in DevOps practice.		×						
17	The effort of similar tasks can be predicted since the matrices are given numeric outputs.		×						

18	Monitoring the server’s health ensures that configurations and scanned features are working as intended.			×					
19	Monitoring development milestones helps determine the effectiveness of the Dev team.			×					
20	The potential threats can be recognized in advance by the Ops team through monitoring the system continuously.			×					
21	Monitoring the Full-Stack and Faster Troubleshooting are the main characteristics of the proper Monitoring tools.			×					
22	The impacted areas of development can be identified by Continuous Monitoring of the system.			×					
23	The project Requirements and Enhancements should be compatible with the Tech-Stack (Set of tools, frameworks, and libraries) of the application.				×				
24	The involvement of Technical Expertise will be an added advantage in the effort estimation.				×				×
25	Having a good understanding of the Tech-Stack (Structure Wise-Frontend/Backend) of the application to the team will make it easy to estimate the effort.				×				
26	Selecting Ready-Made (Ideally suited) solutions of Tech-Stack (Combination of technologies) will make it more convenient for the effort estimation while doing a workaround on the application.				×				
27	Sharing the technical knowledge of Senior Engineers within the team helps ease the effort estimation.					×			
28	Having an idea about the existing reusable implementation makes it easy to estimate the effort.					×			

29	Conducting Technical sessions about the architecture of the application helps effort estimation in DevOps practice.					×			
30	Maintaining Best Practices in development and testing with the team makes it easy to estimate the effort.					×			
31	The use of prior Skills and Experiences would help to measure the effort in DevOps practice.						×		×
32	Sharing Research and Development (R&D) experiences in the application, make it easy to estimate the effort in the DevOps practice.						×		
33	Having skills & Experience in a specific area in development helps me in effort estimation personally.						×		
34	The use of prior Skills and Experiences in the specific programming language would help to measure the effort easily in the DevOps context.						×		
35	An Automated deployment process helps to reduce the effort and risks in the production environments.							×	
36	The Automated deployment process reduces the integration time in the DevOps environment.							×	
37	Both debugging and fixing efforts are considered as a total effort at the effort estimation in the DevOps environment.							×	
38	The Automated deployment process allows continuous delivery and makes effort estimation easy in a DevOps environment.							×	
39	Following effort estimation practices helps in continuous delivery of software with high quality in the DevOps environment.								×
40	I feel that we can be more productive at work if we follow effort estimation practices as a team.								×

Appendix C: Final Survey

Please find the below questions made to identify the factors affecting effort estimation in DevOps-based software development for the final survey.

PART A – Demographic

1. What is your gender?
    - ✓ Male
    - ✓ Female

2. Which best describes your current track?
    - ✓ Operation (Application Engineer/ System or Support Engineer/DevOps Engineer/Database Administrator)
    - ✓ Quality Assurance
    - ✓ Development (Software Engineer/Full Stack Engineer/Tech Lead/Solution or Technical Architect)
    - ✓ Delivery (Project Manager/ Business Analyst)

3. Which category below includes your work experience in the IT industry?
    - ✓ 0- 2 years
    - ✓ 2-6 years
    - ✓ 6-10 years
    - ✓ Over 10 years

4. What is the highest level of education you have completed?
    - ✓ Advanced Certificate
    - ✓ Diploma
    - ✓ Higher Diploma
    - ✓ Bachelor/ Bachelor Hons
    - ✓ PG cert /PG Dip
    - ✓ Master's Degree
    - ✓ MPhil/ PhD

5. How many employees are there in your organization?
    - ✓ 0-100
    - ✓ 101-300
    - ✓ 301-600
    - ✓ 601-1000
    - ✓ More than 1000

6. Does your organization follow the effort estimation strategies?
    - ✓ Yes
    - ✓ No

7. Which category does your organization belong to?

- ✓ Product-based
- ✓ Service-based
- ✓ Both
- ✓ Other

8. What is your opinion on effort estimation practices in DevOps-based software development.? I have .... mind regarding effort estimation in DevOps based software development.
    - ✓ Very negative
    - ✓ Negative
    - ✓ Neutral
    - ✓ Positive
    - ✓ Very Positive

PART B –

The objective of these questions is to measure the factors which are most affected by effort estimation strategies and practices of DevOps-based software development in Sri Lanka.

9. The use of communication tools and well-defined communication guidelines make it easy to estimate the required effort in DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

10. The scope of the assigned tasks can be easily understood if we improve the communication skills.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

11. Proper communication helps to improve the motivation to fulfil a task in DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

12. The accuracy of the effort estimation in DevOps practice can be improved by allocating sufficient time to communicate with the collaborators.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

13. Proper communication at the stage of effort estimation defines a lightweight process to accomplish a task.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

14. The use of Measuring Tools and well-defined Measuring Guidelines, make it easy to estimate the required effort in the DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

15. Daily Effort Measurements will increase effectiveness.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

16. Given the structure of effort estimation metrics make it easy estimation in DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

17. The effort of similar tasks can be predicted since the matrices are given numeric outputs.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

18. Monitoring the server’s health makes sure configurations and scans features are working as intended.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

19. Monitoring development milestones helps to determine the effectiveness of the Dev team.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

20. The potential threats can be recognized in advance by the Ops team through monitoring the system continuously.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

21. Monitoring the Full-Stack and Faster Troubleshooting are the main characteristics of the proper Monitoring tools.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

22. The impacted areas of development can be identified by Continuous Monitoring of the system.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

23. The project Requirements and Enhancements should be compatible with the Tech-Stack (Set of tools, frameworks, and libraries) of the application.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

24. The involvement of Technical Expertise will be an added advantage in the effort estimation.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

25. Having a good understanding of the Tech-Stack (Structure Wise-Frontend/Backend) of the application to the team will make it easy to estimate the effort.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

26. Selecting Ready-Made (Ideally suited) solutions of Tech-Stack (Combination of technologies) will make more convenient the effort estimation while doing a workaround on the application.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

27. Sharing the technical knowledge of Senior Engineers within the team helps to ease the effort estimation.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

28. Having an idea about the existing reusable implementation makes it easy to estimate the effort.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

29. Conducting Technical sessions about the architecture of the application helps to effort estimation in DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree

- ✓ Neither Agree or Disagree
- ✓ Disagree
- ✓ Strongly Disagree

30. Maintaining Best Practices in development and testing with the team makes it easy to estimate the effort.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

31. The use of prior Skills and Experiences would help to measure the effort in DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

32. Sharing Research and Development (R&D) experiences in the application, make it easy to estimate the effort in the DevOps practice.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

33. Having Skills and Experience in a specific area in development helps me in effort estimation personally.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

34. The use of prior Skills and Experiences in the specific programming language would help to measure the effort easily in the DevOps context.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

35. An automated deployment process helps to reduce the effort and risks in the production environments.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

36. The Automated deployment process reduces the integration time in the DevOps environment.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

37. Both debugging and fixing efforts should be considered for the total effort at the effort estimation in the DevOps environment.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

38. The Automated deployment process allows continuous delivery and makes effort estimation easy in a DevOps environment.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

39. Following effort estimation practices helps to continuous delivery of software with high quality in the DevOps environment.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree
    - ✓ Strongly Disagree

40. I feel that we can be more productive at work if we follow effort estimation practices as a team.
    - ✓ Strongly Agree
    - ✓ Agree
    - ✓ Neither Agree or Disagree
    - ✓ Disagree

- ✓ Strongly Disagree

PART C –

The Open-Ended question was included getting an opinion about the experience or suggestion on effort estimation strategies and practices in DevOps-based software development.

41. Please indicate your experience or suggestions on effort estimation strategies and practices in DevOps-based software development.

Appendix D: Summary of Final Survey Results

Variable	Question	Number of Responses				
		Strongly Agree	**Agree**	**Neither Agree or Disagree**	**Disagree**	**Strongly Disagree**
Poor Communication	9. The use of communication tools and well-defined communication guidelines make it easy to estimate the required effort in the DevOps practice.	**19.2%**	**78.5%**	**2.4%**	**0.00%**	**0.00%**
	10. The scope of the assigned tasks can easily understand improving communication skills.	**33.6%**	**62.2%**	**3.4%**	**0.8%**	**0.00%**
	11. Proper communication helps to improve the motivation to fulfil a task in DevOps practice.	**22.3%**	**74.8%**	**2.6%**	**0.3%**	**0.00%**
	12. The accuracy of the effort estimation in DevOps practice can be improved by allocating sufficient time to communicate with the collaborators.	**29.9%**	**65.1%**	**4.5%**	**0.5%**	**0.00%**
	13. Proper communication at the stage of effort estimation defines a lightweight process to accomplish a task.	**24.9%**	**61.4%**	**12.1%**	**1.6%**	**0.00%**
Measurements	14. The use of Measuring tools and well-defined Measuring guidelines, make it easy to estimate the required effort in the DevOps practice.	**15.5%**	**78.7%**	**5.5%**	**0.00%**	**0.00%**
	15. Daily Effort Measurements will increase effectiveness.	**10.2%**	**75.3%**	**8.9%**	**5.5%**	**0.00%**
	16. Given the structure of effort estimation metrics make it is an easy estimation in DevOps practice.	**15.7%**	**72.4%**	**11.5%**	**0.3%**	**0.00%**
	17. The effort of similar tasks can be predicted since the matrices are given numeric outputs.	**16.3%**	**54.9%**	**26.2%**	**2.4%**	**0.00%**
Monitoring	18. Monitoring the server's health makes sure configurations and scans features are working as intended.	**14.7%**	**70.6%**	**9.2%**	**5.2%**	**0.00%**
	19. Monitoring development milestones helps to determine the effectiveness of the Dev team.	**27%**	**68.2%**	**2.1%**	**2.4%**	**0.00%**

	20. The potential threats can be recognized in advance by the Ops team through monitoring the system continuously.	**17.1%**	**76.6%**	**3.9%**	**2.4%**	**0.00%**
	21. Monitoring the Full-Stack and Faster Troubleshooting are the main characteristics of the proper Monitoring tools.	**9.7%**	**71.7%**	**18.4%**	**0.00%**	**0.00%**
	22. The impacted areas of development can be identified by Continuous Monitoring of the system.	**12.3%**	**74%**	**2.9%**	**10.8%**	**0.00%**
Tech Stack	23. The project Requirements and Enhancements should be compatible with the Tech-Stack (Set of tools, frameworks, and libraries) of the application.	**24.7%**	**59.1%**	**15.2%**	**1.00%**	**0.00%**
	24. Involvement of Technical Expertise will be an added advantage in the effort estimation.	**37.3%**	**61.7%**	**1.00%**	**0.00%**	**0.00%**
	25. Having a good understanding of the Tech-Stack (Structure Wise-Frontend/Backend) of the application to the team will make it easy to estimate the effort.	**31%**	**65.1%**	**3.9%**	**0.00%**	**0.00%**
	26. Selecting Ready-Made (Ideally suited) solutions of Tech-Stack (Combination of technologies) will make more convenient the effort estimation while doing a workaround on the application.	**11.8%**	**68.8%**	**18.4%**	**0.8%**	**0.00%**
Knowledge Sharing	27. Sharing the technical knowledge of Senior Engineers within the team helps to ease the effort estimation.	**27%**	**67.2%**	**5.8%**	**0.00%**	**0.00%**
	28. Having an idea about the existing reusable implementation makes it easy to estimate the effort.	**23.9%**	**70.9%**	**5%**	**0.3%**	**0.00%**
	29. Conducting Technical sessions about the architecture of the application helps to effort estimation in DevOps practice.	**20.7%**	**73.8%**	**5.5%**	**0.00%**	**0.00%**
	30. Maintaining Best Practices in development and testing with the team makes it easy to estimate the effort.	**22.3%**	**73.8%**	**2.6%**	**1.3%**	**0.00%**
Skills and Experience	31. The use of prior Skills and Experiences would help to measure the effort in DevOps practice.	**34.1%**	**64.6%**	**1.3%**	**0.00%**	**0.00%**
	32. Sharing Research and Development (R&D) experiences in the application, make it easy to estimate the effort in the DevOps practice.	**17.1%**	**74%**	**8.7%**	**0.00%**	**0.00%**

	33. Having Skills and Experience in a specific area in development helps me in effort estimation personally.	**21.3%**	**66.4%**	**12.1%**	**0.3%**	**0.00%**
	34. The use of prior Skills and Experiences in the specific programming language would help to measure the effort easily in the DevOps context.	**15.2%**	**69.8%**	**13.6%**	**1.00%**	**0.30%**
Deployment Process	35. An automated deployment process helps to reduce the effort and risks in the production environments.	**25.7%**	**64%**	**9.4%**	**0.8%**	**0.00%**
	36. The Automated deployment process reduces the integration time in the DevOps environment.	**25.2%**	**63.8%**	**10%**	**1.00%**	**0.00%**
	37. Both debugging and fixing efforts should be considered for the total effort at the effort estimation in the DevOps environment.	**20.5%**	**69%**	**9.7%**	**0.8%**	**0.00%**
	38. The Automated deployment process allows continuous delivery and makes effort estimation easy in a DevOps environment.	**24.7%**	**66.7%**	**7.9%**	**0.8%**	**0.00%**
The success of EE strategies and practices in the DevOps environment.	39. Following effort estimation practices helps to continuous delivery of software with high quality in the DevOps environment.	**37%**	**62%**	**1%**	**0.00%**	**0.00%**
	40.I feel that we can more productive at work if we follow effort estimation practices as a team.	**19%**	**78%**	**37%**	**0.00%**	**0.00%**

Appendix E: Company List

List of software companies utilized for this study.

1. Efuturesworld Pvt Ltd
2. Virtusa (Pvt) Ltd
3. Ideahub Pvt Ltd
4. Inova IT Systems (Pvt) Ltd
5. eBuilder Technology Centre (Pvt) Ltd
6. OrangeHRM (Pvt) Ltd
7. IFS R&D International
8. 99x Technology
9. Wso2
10. Creative Solution Pvt Ltd
11. Geveo Australasia Pvt Ltd
12. Codegen International Pvt Ltd
13. Dialog Axiata PLC
14. Softcodeit Solutions (Pvt) Ltd
15. Mitra IT Solutions
16. Embla Software Innovation
17. Eyepax IT Consulting (Pvt) Ltd
18. DMS Software Engineering (Pvt) Ltd
19. Wiley Global Technology Pvt. Ltd.
20. hSenid Software International
21. Cake Labs