Deanship of Graduate Studies

Al-Quds University

Effort Estimation Techniques in Software Development - A Case Study for Palestinian Software Companies

Oday Abed Mohammad Al-Wahsh

M.Sc. Thesis

Jerusalem - Palestine

1444 / 2022

Effort Estimation Techniques in Software Development - A Case Study for Palestinian Software Companies

Prepared By:

Oday Abed Mohammad Al-Wahsh

B.Sc. Information Technology from Palestine Ahliya University - Palestine.

Supervisor: Dr. Raid Zaghal

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science / Department of Computer Science Faculty of Science & Technology / Deanship of Graduate Studies /Al-Quds University.

1444 / 2022

Al-Quds University

Deanship of Graduate Studies

Department of Computer Science

Master in Computer Science

Thesis Approval

Effort Estimation Techniques in Software Development - A Case Study for Palestinian Software Companies

Prepared By: Oday Abed Mohammad Al-Wahsh

Registration No: 21911047

Supervisor: Dr. Raid Zaghal

Master thesis submitted and accepted, Date: 24/12/2022.

The names and signatures of the examining committee are as follow:

1. Head of Committee: Dr. Raid Zaghal signature .....
2. Internal Examiner: Dr. Radwan Qasrawi signature .....
3. External Examiner: Dr. Jamil Itmazi signature .....

Jerusalem - Palestine

1444 / 2022

Dedication

I dedicate my thesis to my family. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears, and who never left my side and are very special. I also dedicate it to the soul of my dear grandfather Musa Abu Dayyeh. I also dedicate this dissertation to my brothers, my sisters, and friends who have supported me throughout the process.

Thank you all

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signature: 

Oday Abed Mohammad Al-Wahsh

Date: 24/12/2022

Acknowledgment

All praise is to Allah,

I would like to express my sincere gratitude to my supervisor, Dr. Raid Zaghal. He has offered me great freedom on choosing my favorite research topic and developing my research interests, and continuously provided me help and encouragement with extensive knowledge. also I would like to thank Mr. Asem Thaher, who provided me with valuable information in the field of effort assessment and was always available to answer any question. I also thank the examining committee, all my colleagues and relatives for their support. Finally, i would like to thank the managers, team leaders and developers in the Palestinian companies who have participated in our case study.

Many thanks to my mother and my wife for them praying and providing support and to my father thank you very much.

Abstract

Effort estimation is one way to evaluate a software project and understand its schedule and budget. It is one of the most important aspects of the software development life cycle. The failure of not recognizing the accurate effort estimation may lead to increase the financial costs of the companies and their clients which will cause negative impact on their job duties and their future marketing plans besides the client disappointment and dissatisfaction.

The Palestinian IT sector is considered to be one of the most developing and promising sectors. However, the studies that investigate the methods and techniques of effort estimation are most likely missing. For that reason, we were motivated to study the status of the software development companies in Palestine in order to better understand how the technical teams estimate the needed effort of their software projects.

The purpose of this study is (i) to survey the existing effort estimation techniques used by prominent Palestinian software development companies and analyze their practices, and (ii) to suggest an appropriate effort estimation technique that can suite the nature and needs of these companies and to validate this technique via real application within actual software projects in a selected subset of these companies.

Based on our survey and analysis, we have selected an existing effort estimation technique called (Triangulation) as the most appropriate EE method for the small companies and for the Palestinian software development industry, then we have designed an extension of this technique in order to (fine-tune) for the company's exact needs and provide it with some flexibility to make some changes and adjustments (e.g., decrease project delivery time by increasing resources). Moreover, this technique was applied on the Palestinian software

projects to validate its results.

We believe this study can be a valuable resource for Palestinian software development companies; and they can use it as a guideline to help them get better and more accurate effort estimates, which in return can reduce costs and provide better and more accurate scheduling and staffing needs.

Keywords

Software development, effort estimation technique, ensemble effort estimation, Software engineering, Agile.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Abbreviations Full Name
EE	Effort Estimation
RFP	Request for Proposal
IT	Information Technology
QA	Quality Assurance
PM	Project Manager
COCOMO	Cost Constructive Model
ANN	Artificial Neural Network
NN	Neural Network
RN	Recurrent Neural Networks
CNN	Convolution Neural Networks
GA	Genetic Algorithm

Chapter 1

Introduction

This chapter introduces the thesis. It describes the problem statement, purpose, research questions, limitations, contributions, methodology and organization of the thesis.

1.1 Introduction

The importance of effort estimation for software projects is as important as any other stage in project management. It is defined as the process of predicting the number of working hours or the time needed to develop software. It is usually expressed in units of working hours or unit months of work. Nowadays, software development is costly and difficult. Any failure to estimate the cost or time of a software project can increase financial burdens on the project and waste resources. It has also been found that inaccurate effort estimation can lead to a complete failure of the project [1].

Providing a model for estimating effort with high quality and accuracy is an integral part of the success of the software project. Comparing the estimated efforts with actual efforts can be used as an indicator of the accuracy of the estimation model used.

It has been shown in many studies [2] [3] [4] [5] [6] [7] [8] [9] that effort estimation can be dealt with in different ways. Various methods of effort estimation are used for software projects. Also, each method or model has its strengths and weaknesses regarding the quality of the estimates it produces.

Software estimation models can be classified into three main categories: Algorithmic Methods, Non-Algorithmic Methods, and Learning Oriented Models [10]. All models have many challenges and in some cases some models are combined to improve the quality of the estimate.

Algorithmic Estimation like COCOMO (Cost Constructive Model), Putnam's Model, Parametric Based Models and Analogy Based Models, these methods are based on mathematical models which produce function of a number of variables Puntametc.

Non-Algorithmic Methods or are expressed in Expertise Estimation such as poker planning, Top-Down or Bottom-Up Estimation, and Price-to-Win Estimation. Domain experts, historical data, and team skills are the main factors affecting the quality of the estimate [11] According to studies conducted by [8] [9], expert-based models are the most widely used models in effort estimation. These Methods are used when there is complexity to collecting data and requirements.

The IT (Information Technology) sector in Palestine is one of the promising and ever-growing sectors. However, studies dealing with effort estimation approaches and techniques in Palestinian IT sector are still largely missing. It has been noted that software development companies in Palestine usually have small to medium-sized development teams (e.g., up to 20 programmers). Therefore, we were motivated to conduct a case study of Palestinian software development companies to increase our understanding of how technical teams approach estimating software efforts and to explore the challenges they face. We aim to provide Palestinian Software Development companies with a feasible set of recommendations and practical ideas that would help them improve their effort estimation and thus save them valuable resources.

This study suggests using an existing technique to improve the accuracy of estimating the efforts of software development companies and help them calculate the estimation of effort, which is **Triangulation**, which gives better results and higher accuracy especially to small and medium-size projects. We have also designed and tested an extension of the triangulation technique, which in turn enables Palestinian companies to adjust (and fine-tune) the project delivery time by making some adjustments on the project's resources and parameters, and therefore predicting the cost more accurately.

1.2 Problem Statement

Effort Estimation and Estimation Accuracy are one of the major challenges for software development projects [12] [13] [14]. The Palestinian IT sector is a promising sector and has a respectable number of mature and new companies working in the software development industry [15]. This sector is one of the most fast-growing in Palestine which employs thousands of professionals.

Based on previous studies dealing with an understanding of effort estimation in software companies in Palestine, there was a 25% overrun in project costs due to inaccuracy in estimating effort for software projects [16], and there is also a lack of research in the field of effort estimation in Palestine. Also, Unfortunately, there is a lack of knowledge among the Palestinian companies on how to estimate effort using published tools and techniques, which in turn usually affects their project planning (e.g., scheduling, human resources, budgeting, etc); Therefore, these companies usually suffer from continuous delays in delivering their products due to the failure to accurately estimate the effort needed to get the job done from the beginning. This usually leads to increased financial costs for the companies and for their

customers, which will have adverse consequences on their businesses and marketing plans as well. On the part of customers, there is a problem of not knowing the pricing of the software in order to clarify the prices of the software industry in Palestine. On the other hand, Palestinian engineers are assigned tasks in any given Sprint (within SCRUM to Agile development methodology), and they are asked to set a time to accomplish this task, and the engineer has to deliver the task on time. So, engineers need a way to calculate time to get this task done on time.

Based on a survey we conducted in 2021, it was found that a large percentage of software engineers in Palestinian companies do not use any formal techniques for estimating effort, but instead, they resort to common knowledge and personal experience. Accordingly, it was found that there is a clear lack of knowledge of methods for calculating effort estimation among engineers and/or project managers in Palestinian companies, and therefore there is a real need for this research, to be able sort out this issue, and to find a reference EE (Effort Estimation) model for these companies, not only in Palestine, but also for all countries which have similar status and Software industry requirements.

1.3 Research Purpose

Due to the currently very few numbers of published studies that have dealt with methods and techniques for effort estimation in Palestine (and similar countries); We also noted the high percentage of inaccurate / improvised estimations that had led to cost overruns. We believe no such model had been proposed to improve the accuracy of effort estimations for our software industry; and therefore, we have conducted this study to improve the accuracy of effort estimation in software projects, by suggest an appropriate effort estimation

model/technique based on well-known model with our own extension/modification; in order to get accurate results relevant to the needs of Palestinian companies. and then to validate it and connect these findings to our proposal and make relevant recommendations that may positively affect these companies and the quality of their products and the accuracy of their estimations.

1.4 Research Questions

We have formulated four research questions based on research purpose stated above:

- What are the common software EE techniques used in Palestine?
- How can we improve the software EE in Palestine?
- What is the appropriate technique for Palestinian companies to calculate the EE?

1.5 Research Limitations

In the research we did the survey, the experiments, and validation in Palestinian software development sector. It was a challenging task to get the full collaboration of a large number of these companies (they are usually very busy and won't give-up even a small fraction of their valuable time for us); therefore, we had to rely on a small number of companies that had agreed to cooperate including the researcher's own company (ASAL Technologies).

1.6 Research Contributions

The main contribution of this research is to propose a reference model for the Palestinian

companies and software engineers that will help/guide them to accurately estimate the effort needed to complete their software development projects – before they begin. By employing the appropriate EE technique and verifying its validity and accuracy level, they can improve their work-schedule and the overall quality of their services/products. Also, an improvement was added to the Triangulation technique, which will enable Palestinian companies to adjust the project delivery time and calculate the project cost more accurately.

1.7 Research Methodology

In this study, based on previous studies that had dealt with exploring and knowing what are the methods and techniques used in estimating effort in Palestinian companies, we will do the following:

- Conduct a survey of Palestinian companies to learn about the methods and techniques used in EE.
- Perform data collection and analysis of previous software projects from Palestinian companies.
- Select / Develop an appropriate EE technique that suits the Palestinian market needs.
- Validate our work by applying our technique to software projects that have already been completed – in order to find the accuracy level by comparing the difference between what-is (the actual effort) and what-is-expected according to our estimation.

1.8 Thesis Outline

This thesis is structured as follows:

Chapter 1: Introduction: Gives an overview of the research and declares the problem statement, research purpose, questions, scope, limitations, methodology, contribution.

Chapter 2: Background: Provides a general background of the concepts needed to understand the rest of the thesis.

Chapter 3: Literature Reviews: Reviews relevant previous works and literature as a framework for our intervention, we will explain the gap that we are trying to fill in, especially that is related to the SD industry in Palestinian.

Chapter 4: Research Methodology: Here, we present the research methodology and process; starting with a survey of Palestinian companies, data collection and data analysis of previous software projects from Palestinian companies, programing and experimentation of well-known EE technique, and ending with the proposed EE technique and the relevant recommendations.

Chapter 5: Experiment and Verification: The selection modification of the appropriate technique. This includes collecting information about a software project developed in the Palestinian companies, applying the selected (modified/ if necessary modification) technique to that project, and obtaining the accuracy of this technology by comparing the results of this technology with the actual results.

Chapter 6: Conclusion and future work: Summary, conclusion, future work and recommendations.

Chapter 2

Background

This chapter provides a general background of the concepts needed to understand the rest of this research and covers the most important techniques used in estimating effort. It covers basic concepts of effort estimation in software engineering and concepts related to the chosen technique which is Triangulation.

2.1 Effort Estimation Methods

Summarizing the concept of software effort estimation is a method of predicting the most accurate amount of time for a person's work or required to develop a software project. The process of Effort estimation is a set of steps to know the estimates of any software. Absence of process the result must be inaccurate [36].

Estimation = Duration of task completion + task completion Cost

Effort estimation is a way in which inputs which it takes and output which it produces. During this process some resources are adopted with estimation methods. Quantitative and expertise both data are used for estimation. Quantity and Quality of data are two major factors for estimation [37].

Normally estimation is done by previous or past projects but if quality data in perspective of current scenario is not available then estimators have to do new effort estimates instead of taking past historical data.

According to the literature review, three main effort estimation methods were found:

2.1.1 Algorithmic Methods

These models rely mainly on mathematical formulas and expressions to give project prediction, an example of these models is:

- **COCOMO**

It is one of the most popular models used for estimating effort, developed by Boehm[38,39]. Cocomo is a regression model based on LOC, i.e., number of Lines of Code. This model was built on the basis of 63 software projects.

Depending on the level of accuracy and correctness needed, many Cocomo models have been put out to anticipate cost estimation at various levels. These models can all be utilized for a variety of projects, depending on the features of those projects on which the value of the constant to be used in further calculations will be based. The following lists these traits as they relate to various system kinds.

Embedded, organic, and semi-detached systems according to Boehm, Software projects are considered to be of the "**organic**" kind if the necessary team size is small enough, the problem is well known and has already been handled, and the team members have only rudimentary familiarity with the issue.

Software projects are classified as **semi-detached** if key factors including team size, expertise, and familiarity with different programming environments fall somewhere between those of organic and embedded projects. Semi-detached projects demand more expertise, better supervision, and creative thinking than organic projects because they are less well-known and more challenging to create. Examples of semi-detached types include compilers

and various embedded systems.

Software projects that demand the most complexity, creativity, and experience fall into the category of **embedded**. In comparison to the other two models, such software necessitates a bigger team size, and the programmers must be sufficiently skilled and original to create such intricate models.

The constants used in effort calculations are used in various ways by each of the system types mentioned above.

Model Types: COCOMO is a hierarchy of three progressively more increasingly detailed and accurate forms. Any of the three formats can be used if it meets our criteria. These COCOMO model varieties include:

1) Basic COCOMO Model
2) Intermediate COCOMO Model
3) Detailed COCOMO Model

Calculating the cost of software can be done quickly and roughly using the Basic COCOMO level. Due to insufficient factor considerations, its precision is somewhat constrained. These Cost Drivers are taken into account by Intermediate COCOMO, and Detailed COCOMO also takes into account the impact of individual project phases, meaning that Detailed COCOMO takes into account both of these Cost Drivers and also performs calculations phase-by-phase, yielding a more accurate result.

- ## Function Point Analysis

Proper metric systems continue to be a fundamental topic in software engineering since software systems include a variety of functional features. Software engineers concentrate on

quantifying the quantity of functionality in software development projects as a result. Functional point analysis approach was created to simplify the challenging work of decreasing software metrics in terms of functional size [40].

Functional point analysis is a term used to describe standardized techniques for estimating software sizes using functional constraints that identify the most important features to be built [41]. Because it may be used to a variety of programming languages and technologies, this strategy is really ubiquitous. Measurements are made of two main components in function point analysis. The data functionality and transaction functionality qualities are the most important components of the software application measure in function point analysis [42].

In particular, the system performance, productivity, quality indicators, and scope of the system product are assessed to define the metrics of the two fundamental aspects. Function point analysis analyzes the metrics of the system from a functional standpoint, resolving problems brought on by technology dependence in the development lifecycle [40]. Three steps of in-depth application analysis are used to maximize FPA's efficiency in software engineering [40]. Identifying the kind of transactions that will be made in the software applications is the first step in the function point analysis process. Second, the engineers assess and evaluate the software system's constituent parts. Evaluations of the overall system characteristics are a final step in the process. Fundamentally, there are 14 primary features used to classify general system characteristics [40]. Data processing, system performance, hardware configurations, transaction rates, data entry, end-user productivity, online updates, reusability, ease of use, support for numerous sites, and change facilitations are some of these [40]. Incorporating non-functional needs into function point analysis has also been the subject of extensive research.

2.1.2 Non-Algorithmic Methods

- **Expert Judgement**

  One of the traditional methods used in the early stages of software development for software cost estimation is expert judgment [42]. The method is such that it heavily relies on the knowledge and experience of a cost estimation specialist. Rather than historical data, it rely on the expert's domain expertise [42]. Because they have the necessary knowledge to ensure cost estimation is as accurate as possible, experienced estimators are given the task of estimating software costs [42]. Similar to the previous point, the estimator's professional judgment is frequently constrained to a certain area of competence.

  An expert with in-depth knowledge and expertise in a given field is more likely to have majored in the project at hand, which puts them in a better position to estimate the cost. When there are restrictions that prevent effective and efficient data collecting, expert judgment is extremely helpful [43]. In order to make choices based on the constraints and strict factors, professional judgment is needed. One example of an expert judgment approach is the Delphi technique.

  When it comes to comprehending the effects of incorporating a system, an expert may provide an open and knowledgeable view on the best course of action. Another drawback is the time-consuming procedures involved in recording the details that an expert need in order to render a decision [42].

  The majority of the elements in many software projects are many, making it tiresome and challenging to record them. This method has no specific validation because the estimation is completely based on the expert's prior experiences and domain knowledge. The drawback of receiving skewed and optimistic expense estimates is

another one. It is quite possible that the expert's emotions will affect the decision-making process because professionals make expert judgments and have human emotions [42] [43]. As a result, bias, pessimism, and optimism could all have an impact on how something is judged.

- **Top-Down Estimation**

  The algorithmic or non-algorithmic methods can be used in the Top-Down cost estimation methodology, which focuses on calculating the cost of a project from the global properties of the overall project [44]. The estimation is then proportionally divided into the various components [44].

  When there is little historical information about a similar project available, this approach can be used. When the project is still in its early phases, this strategy is more advantageous. This is such that no further information regarding the project is required at this time [44].

  High-level judgments are made via top-down estimating when the planning horizon is quite long. When there is only a general idea of the project and we need an estimate, this method is used. This methodology focuses on tasks that are ignored by other cost estimation methods, such as management and integration, in contrast to other methods. The ballpark estimate's biggest drawback is how wrong it is [44].

- **Bottom-Up Estimation**

  In comparison to top-down estimating, this is the complete opposite. According to this method, the price of each software component is determined, and the entire cost of the project is determined by adding these components together [44]. The goal of this technique is to accumulate all of the smaller software component estimations to get a

proper estimate. Both techniques can be helpful depending upon the variety of the projects [44].

The estimation process works best to estimate costs for minor projects. In the bottom-up estimation method, we assign a monetary value to each work package or activity in a schedule before adding them all up to get the project's total cost [44]. This can result in a highly precise estimate that is very thorough. The bottom-up approximation does have certain disadvantages, though.

1) If we add up each activity, there might not be any coordination between them, such as resource overlap or dependencies where one activity must be completed before the next can begin.
2) This level of estimate can be expensive to produce because it can take a long time to produce and has a high degree of accuracy. It is therefore advisable to perform a high-level estimate, or a top-down estimate, for the entire project in order to overcome these drawbacks, and then to perform some specific estimates for the next short-term tasks.

2.1.3 Learning Oriented Models

One may define machine learning as an approach to data analysis used to automate analytical models. It is also a subset of Artificial Intelligence, which can be considered to be founded on the idea that computerized equipment and robots can gain knowledge from experience [45]. These methods are capable of learning from the past and making predictions about the future based on the past. In order to estimate software costs, a variety of machine learning techniques have been used, including neural networks, fuzzy logic, evolutionary algorithms, Bayesian networks, support vector regression, and analogy-based techniques. In order to

increase the estimation accuracy, researchers have presented a number of machine learning software cost estimating methods [46]. Additionally, a number of studies have noted that the projected accuracy varies when the same model is built using various historical project datasets or other experimental designs [46] [47] [48] [49] [50] [51].

According to several research, machine learning models perform better than non-machine learning models when it comes to software cost estimation [52] [46] [53] [10]. Due to the large margin of error in the traditional estimation models, the employment of these machine learning models in the software cost estimation process has become increasingly popular [10]. Machine learning algorithms have undergone significant and ongoing advances, which help to produce predictions that are more accurate when used [46] [10].

These learning models consistently predict accurate results because of its learning nature from the previously completed projects. The analysis by Monika et al. [10] came to the conclusion that ANN (Artificial Neural Network) was the most popular strategy for creating estimation models. In the context of software cost estimate, a brief review of these models and a discussion of their advantages and disadvantages have been provided. This will help the researchers determine which methodology-appropriate option to take. Different machine learning models favor various estimating situations because they have various advantages and disadvantages [46].

- ## Artificial Neural Networks

  One of the most often used methods in the field of machine learning models is ANN. As the name implies, it is frequently motivated by the neural portion of the brain system with the aim of mimicking an intelligent, living being [54]. It has two layers: the input layer and the output layer. Within these layers, there is a hidden layer made up of units whose primary function is to apply weights to the data sent in from the

input. These weights are given to the data at random [55].

Due to their great efficacy, they are among the instruments that are most frequently used in cost estimating [52]. Although neural networks have been there for a while, they can be traced back to the year 1940 the missing link was the best way to take advantage of them, and it has now been found. There are various varieties of neural networks, and each one has a specific level of complexity and application. The feed forward neural network (NN) is the most prevalent and all-purpose type of NN because, as its name implies, data only flows from input to output [54].

Recurrent Neural Networks (RN), Convolution Neural Networks (CNN), LTSM Recurrent Neural Networks, and others are other forms. The ANN technique is utilized in cost estimation when it is necessary to comprehend the pattern of specific data before predicting the project cost. It is used to cluster data into specified classifications, forecast outcomes, and categorize data in the field of cost estimation [55]. For instance, it is used to forecast the market through the utilization of historical data in initiatives like the stock exchange market. An important feature of the model is how well artificial neural networks typically capture non-linear interactions.

Deep neural net is used in the model, which requires fewer features [54]. When there are few features in the dataset for model training, the neural network's capacity to create its own features can be useful. It is versatile in that a wide range of features, such as CNNs, RNNs, LSTM RNNs, etc., are available for one to select from (Figure 2.1). Additionally, it has some drawbacks, such as the tendency to sometimes over-fit the data during the software cost estimation process [46].

Additionally, the model needs a tremendous amount of computing power, which might not always be available. Since putting up a high level of processing power is expensive for the collaborating firm, it is preferable if the model can work well even

in areas where there is less power consumption.

According to the inquiry [53], a number of Artificial Neural Networks models have been applied to the processes for estimating the cost of software:

a) Radial basis function (RBF) networks, b) recurrent neural networks, c) feed-forward neural networks, and d) neuro-fuzzy networks.

Choosing the appropriate artificial neural network model is crucial to obtaining accurate estimates, according to Hamza et al. [53]. In their study [53], they also came to the conclusion that feed-forward neural networks perform better than other ANN models, but that noise must be reduced by data filtering. The radial basis function is better in cases of noisy data.

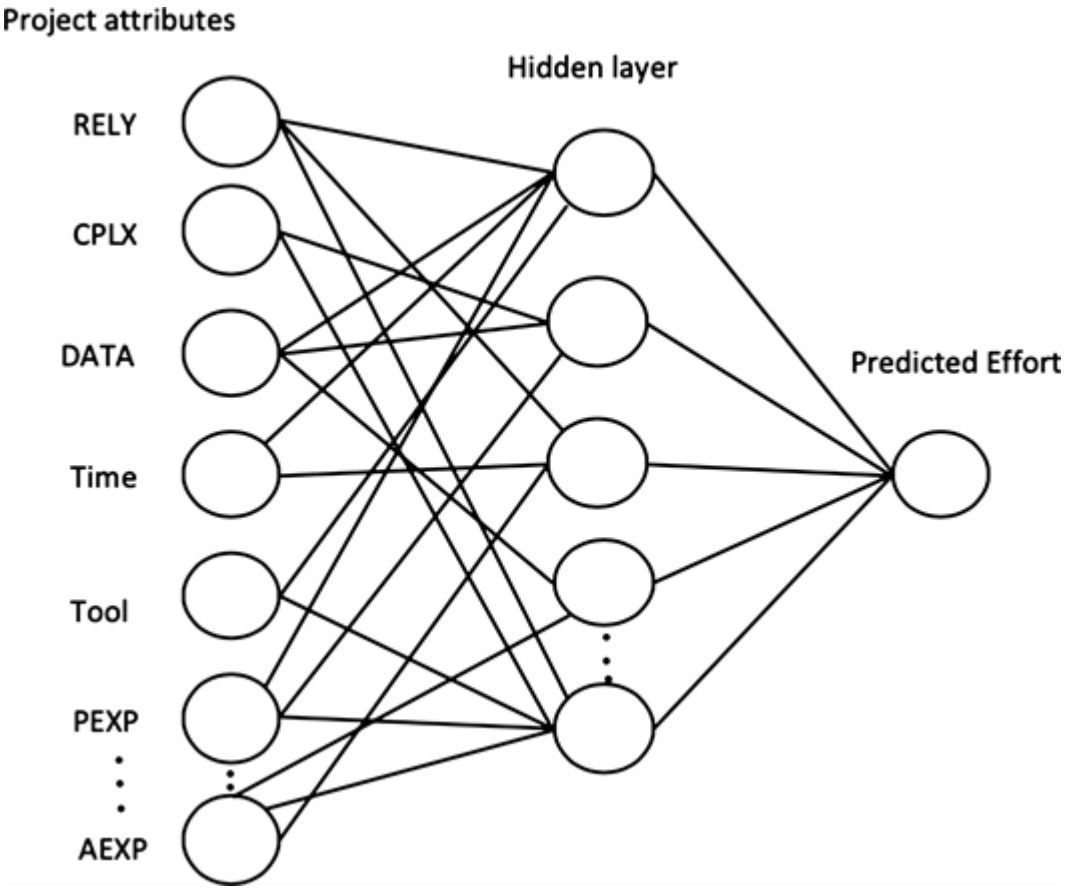

Figure 2.1: A neural network estimation model [42]

- **Genetic Algorithms**

They are described as adaptive and heuristic search algorithms that fall within Darwin's theory of natural selection. They are included as one of the most active topics of research in a recent study that used metaheuristics inspired by nature [56]. One of the soft computing techniques used in the software cost estimating process is the genetic algorithm (GA), whose major function is to modify the parameters of traditional approaches like the COCOMO approach in order to anticipate software costs more precisely [56].

GA has been extensively used in a variety of cost estimation applications, such as rectifying identification systems and solving path-searching issues inside projects [56]. GA has also been utilized to resolve a number of NP-hard computing issues [56]. This model draws its inspiration from nature, such as the design of a fish-inspired bullet train. To name a few, the cuckoo search, particle swarm optimization, and the firefly method are used for optimization issues (Np-problems). The optimization challenge is typically exploited by GA models employing an evolutionary process.

The model's first advantage is that, despite being less flexible than neural networks, it is simpler to set up. The algorithm is on its own once it starts. Since it picks up new features on its own, we don't need to watch over the process. Its drawbacks, on the other hand, include less flexibility, a lot of hyperparameters, such as function preferences, reproduction rates, the amount of elitism and cross-over, handling out-of-bounds conditions, developing a plan, and establishing the necessary tree sizes and depths within the model [30].

The model contains a variety of parameters, sometimes the outcome is very reliant on the parameter settings, and there is no guarantee that it will discover an optimal

solution in infinite time due to asymptotic convergence. It also features settings that are self-adaptive. The approach is computationally intensive and also makes use of meta models for functions. The approach was initially used to calculate the amount of labor needed to finish a specific project [56].

It is also employed in cost estimation scheduling, indicating that it has the capacity to identify and assess a variety of activities inside a single project. It has also been utilized in the same context for data mining, which is regarded as a challenging task when using conventional techniques. On the far end, the software cost estimation process has employed GA to optimize distributed jobs. Distributed inquiries, which indicate that all pertinent queries regarding a particular project can be obtained during the early stages of its development, have been solved using it. Using this strategy, it is very simple to make assumptions and forecasts about the software that is being produced. It is rather time-efficient, especially when used by knowledgeable staff.

The conclusion of the literature review is that there are factors for choosing estimation techniques and the factors are summarized by several points [31]:

1. Is the model having understandable structure and Process?
2. Is the method less expensive, easier to use and clear?
3. Does the model keep missing or past data?
4. Does the model consider uncertainty?
5. How do models consider uncertainty?
6. Is the estimation result accurate or not?
7. Is the method using agile methods or not?
8. Is the method dynamic or not?
9. Does the method produce accurate output if requirement changes during

development?

10. Is the model less time consuming?
11. Does the model fit in the current situation?
12. Does the model produce trustworthy results?

2.2 Document Analysis

Document analysis is used to elicit business analysis information including understanding the context and requirements. Upon receipt of any software project, the requirements are in several forms such as RFP (Request for proposal), Tender, Business requirements and technical specifications. Document analysis may also be used to validate findings from other elicitation efforts such as interviews and observations. Data mining is one approach to document analysis that is used to analyze data in order to determine patterns, group the data into categories, and determine opportunities for change [17].

2.3 Benchmarking

Benchmarking is the process of considering previous projects or studies with the same goal to improve the current study. Once Benchmarking is carried out, the researcher is able to build on the previous study and enhance the project with much more data, effort, and resources. This process approaches the risk analysis that has already been conducted in order to estimate the effort driven for the current study [18].

2.4 Functional Decomposition

Functional decomposition is a technique used to break down the process into smaller parts in order to make the idea of the project a way easier for the receiver to track what will be discussed later. This process takes into consideration that each part will be analyzed separately in detail. This technique aims to encapsulate components; that is, to merge two or more components into one specific category. The process of decomposing high or large functions into sub-functions helps the addressee to get the general order of each sub-element starting from a higher rank to lower and to know that those sub-elements refer to this large element. The way the functional decomposition is diagrammed depends on the functional hierarchy of each sub-components that's supposed to have only one parent component [17].

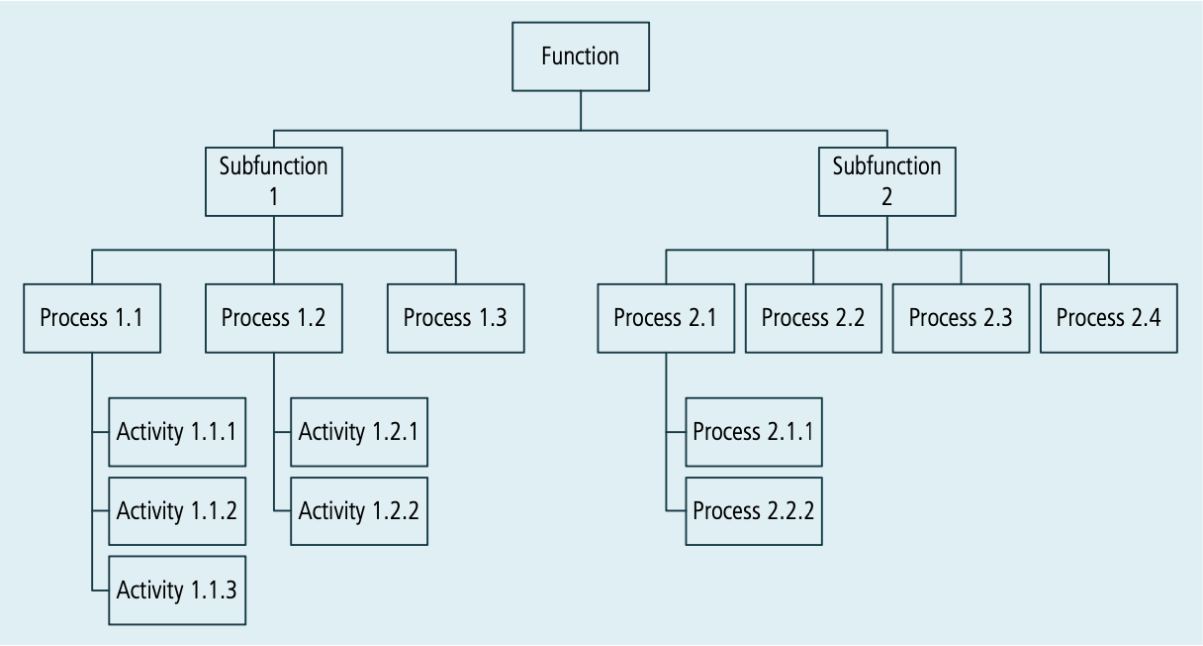

Figure 2.2: Functional Decomposition Diagram [17]

2.5 Brainstorming

Brainstorming is an excellent way to foster creative thinking about a problem. The aim of brainstorming is to produce numerous new ideas, and to derive from them themes for further analysis. Also, it is a technique that aims to produce a wide range of ideas and solutions about a problem. This technique is best applied in a group as it draws on the experience and creativity of all members of the group. In the absence of a group, one could brainstorm on one's own to spark new ideas to heighten creativity [17].

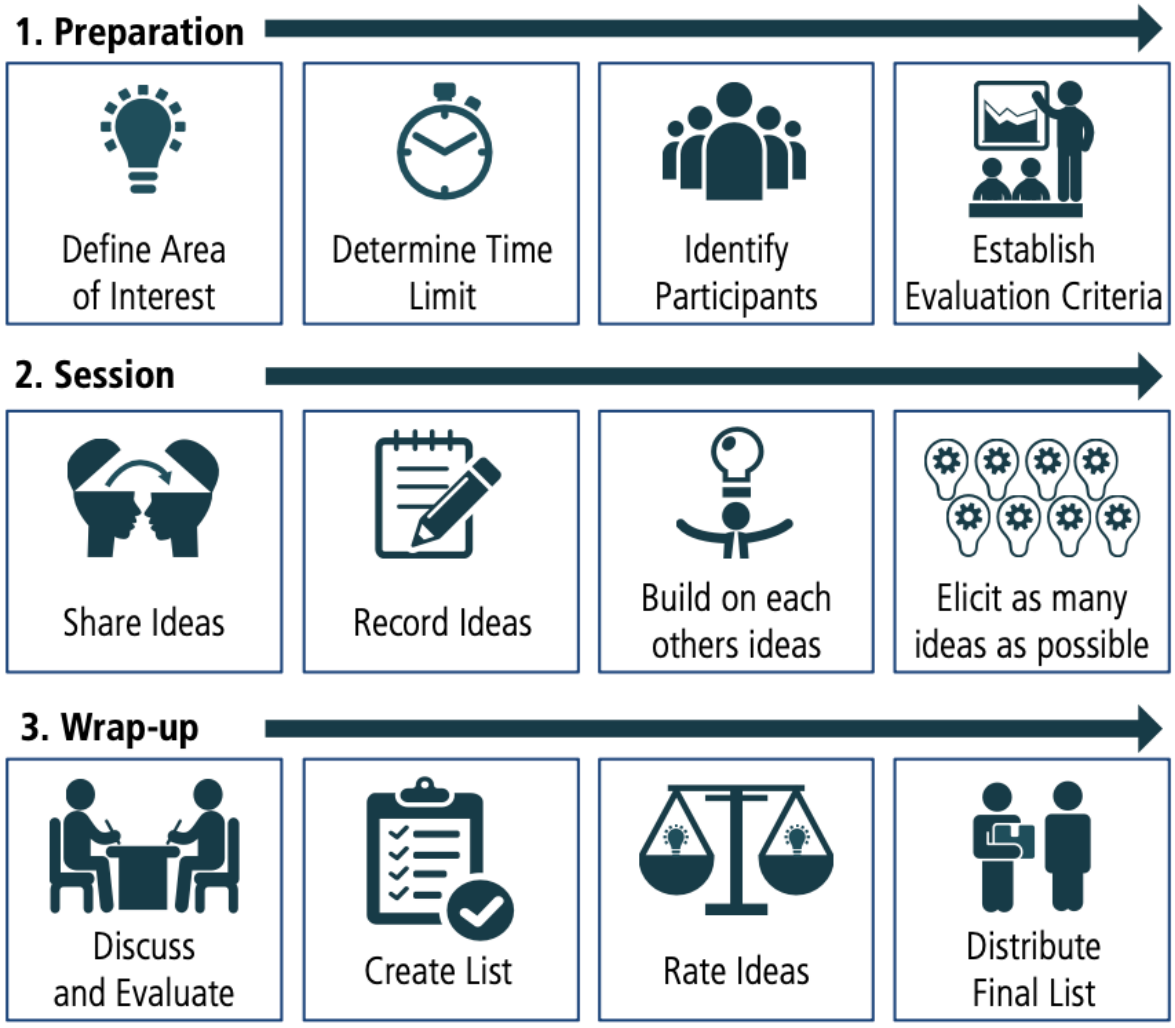

Figure 2.3: Brainstorming [17]

2.6 Calendar Day

It means the total number of working days according to the calendar, including holidays, weekends and working days, for example when saying 12 working days starting from the date 5 June 2022, so the end of the first 12 working days according to the calendar day will be a day 16 June 2022 with weekend days included.

2.7 Business Day

It means the total number of working days only, excluding vacation days and weekends, and one working day is only working hours, that is approximately six working hours, for example when saying 12 working days starting from the date 5 June 2022, so the end of the first 12 working days according to the Business Day will be a day 20 June 2022 without weekend days included.

Chapter 3

Literature Reviews

In this chapter, different related works are studied. The chapter is divided into two sections, in section 3.1 we will review some related work about Effort Estimation Techniques studies in Software Development, in section 3.2 we will give some conclusions about this chapter.

3.1 Effort Estimation Techniques studies in Software Development

Previous studies will be displayed based on the time of their publication from the oldest to the most recent.

In their paper [20], researchers Rashmi Bubley and Naresh Chauhan proposed a method for accurately estimating the cost and effort in developing agile software and focusing on the problems found in agile practices. The paper also focused on understanding the causes of inaccurate estimations in agile software development and discussing several agile estimation techniques. Their proposed method enables the calculation of cost, effort and duration for small and medium enterprises only. The method did not include more factors that make the estimate more valid and efficient, the usefulness of the proposed method is only to reduce the risk of project failure.

Researchers TAILOR, SAINI and RIJWANI presented in their paper [25] a comparative analysis and taxonomy of program cost, effort estimation methods, they categorized the effort estimation methods into four categories: algorithmic model, non – algorithmic model, parametric model and machine learning model They also discussed each technique and

classified it into one of the four models and developed an explanation of each technique and its mechanism of action. Also, they mentioned that there is no single technique that is best for all situations because the choice of technique depends on several criteria such as the complexity of the project, the duration of the project, the experience of employees and the method of development, and finally they added a classification table for all techniques that contain the advantages and disadvantages of each technique.

In his master's thesis, the researcher KAFLE presented an empirical study on software test estimating [24], Which was a case study and collected empirical evidence from software development companies in Nepal, the minimum company size was 30 while the maximum company size was 200, and he performed the case study by conducting interviews with a set of structured questionnaires. The most important results that he reached were test effort estimation error seems to be closely correlated with development effort estimation error. A company that had estimated a total of 3500 man-months had actually spent 4200 man-months implying 700 man-months of effort/cost overruns to complete the project. Another company that projected testing effort of 100 man-hour actually ended up in 120 man-hours at the end of project causing 20 man-hour effort/cost overruns. He was also recommended that there is a need for more studies in this area.

Researchers Gumaei, Almaslukh and Tagoug presented an empirical study for software cost estimation in the Saudi software industry [23], Which focused on software companies in Saudi Arabia and They prepared a questionnaire to collect data with goal of exploring the software cost estimation models and analyzing the reasons which effect on the selection of software cost estimation models or methods in Saudi Arabia software industry. The research methodology they followed which is Research Objectives Establishment and then Design of the Survey and then Analysis and Discussion of Survey Results, the results of the study were that most of the companies use Expert judgment and price-to-win methods in software cost

estimation. They also found that the algorithmic model and estimation by analogy were less commonly used in software cost estimation. They also summarized and discussed several reasons for the inaccuracy in cost estimation methods.

Researchers and specialists seek to find the best technique for estimating effort that gives accurate results on the given datasets and the other applicable attributes. The Project Management Institute (PMI) conducted a survey in 2017, investigating that 69% of software successfully achieved the project's original goals and business priorities, 43% were not finished within their initial budgets, 48% were delivered late and 32% failed due to budget loss [27].

Researchers B. Prakash and V. Viswanathan [21] presented a survey of the techniques used in traditional and agile models. All of these methods used in their paper are mentioned. The estimation model fits based on the development process model used to build the software. There are no single estimation techniques that fit all types of projects. Exceptional estimation method is difficult to determine as it largely depends on project size and various other factors but COCOMO-II and Function point techniques are used in the traditional approach and poker planning is most widely accepted in Agile-based projects.

The Zain and Zarour [22] study was a unique study in the Palestinian market, which was a qualitative study that approached the Palestinian market to increase understanding of how Palestinian companies deal with estimating efforts and exploring the challenges they face. The most prominent results of this study are that about 25% of the cost overrun in software projects is due to Inaccurate estimations, and that the most used model in Palestine is expert-based estimation models, and the accuracy of effort estimation is greatly affected by the team's experience, knowledge of the field, and clarity of requirements.

Accurate and forecasting of the cost and effort of software projects is an essential task of a

successful software project. Researchers Reza and Chirra [29] presented a systematic review paper on models used for software cost estimation that are summarized in algorithmic methods, non-algorithmic methods and learning-oriented methods. The paper aims to provide an overview of models for estimating software costs Effort, summarizing strengths, weaknesses, accuracy, amount of data required, and validation techniques used, and their findings showed that neural network-based models showed higher accuracy over other techniques, and the survey also shows that no single technology is perfect and all have their advantages and disadvantages.

Researchers Mahmood, Kama, Azmi and Ali presented a proposed model [26] to improve the estimation of the prediction accuracy of software development efforts, Proposed model based on use case points (UCP), expert judgment and case-based inference (CBR) techniques. They conducted this research through a primary study (a multi-case involving software companies) study to make an ensemble model, Future work is to evaluate the prediction accuracy of the proposed model by selecting projects from preliminary studies as case choices in the application of a quantitative approach through industry experts, archival data on estimates and evaluation metrics.

IBM-PMO Consultants conducted a survey of 1,500 change management executives. IBM survey in the project success/failure levels shows that only 40% of projects meet schedule, budget and quality goals, underestimating the complexity of the project is also identified as a challenging factor in 35% of the projects.

3.2 Conclusions

In summary, the literature indicates that there is a 25% overrun in project costs due to

inaccurate estimation of efforts for software projects based on research conducted to study the Palestinian market. And there is also a lack of research in the field of effort estimation in Palestinian market. also Unfortunately, there is a lack of knowledge among in Palestinian companies on how to estimate effort using published tools and techniques, usually affects their project planning (e.g., scheduling, human resources, budgeting, etc).

Given the importance of estimating effort in developing software projects, there was a strong impetus to conduct a study to help Palestinian companies by suggesting a technique for estimating effort, which is triangulation.

The main reason behind picking this technique over any other techniques is that the data which has been collected from the IT Palestinian companies is useful for the Triangulation technique. On the other hand, this data didn't go with other techniques like cocomo technique.

For instance, the web project takes into consideration three main parts like Front-end, Back-end and QA (Quality Assurance). Most other techniques don't take those parts into account.

Moreover, the Triangulation technique goes perfectly with the Agile methodology; that is, some techniques the time for the whole project for one time only. However, Triangulation allows the user to set time estimation for each sub task for example.

Undoubtedly, the Triangulation technique supports effort and a time-consuming approach. In other words, some project managers and Software engineers face the issue of the complexity of using some techniques especially when they cost them much time and effort to set time estimation.

All above mentioned reasons pushed me to select this technique among all other techniques to apply it on the Palestinian projects.

Chapter 4

Research Methodology

This chapter proposes an effort estimation technique and the extension of selection Technique that is suitable for Palestinian companies to help them calculate the effort and cost estimation for their software projects. As shown in figure 4.1 and to implement this research. Thus, this chapter is split into five sections, in section 4.1 We studied the state of the Palestinian market by knowing what are the techniques and methods it uses in estimating the effort of software projects through a questionnaire, in section 4.2 Data collection from previously implemented software projects in Palestinian companies to pilot the selected technique, in section 4.3 Analysis of data collected from Palestinian companies, in section 4.4 We searched for the best technique suitable for software projects in Palestinian companies, in section 4.5 We have added modifications to the selected technique to help us obtain additional information about the effort estimation.

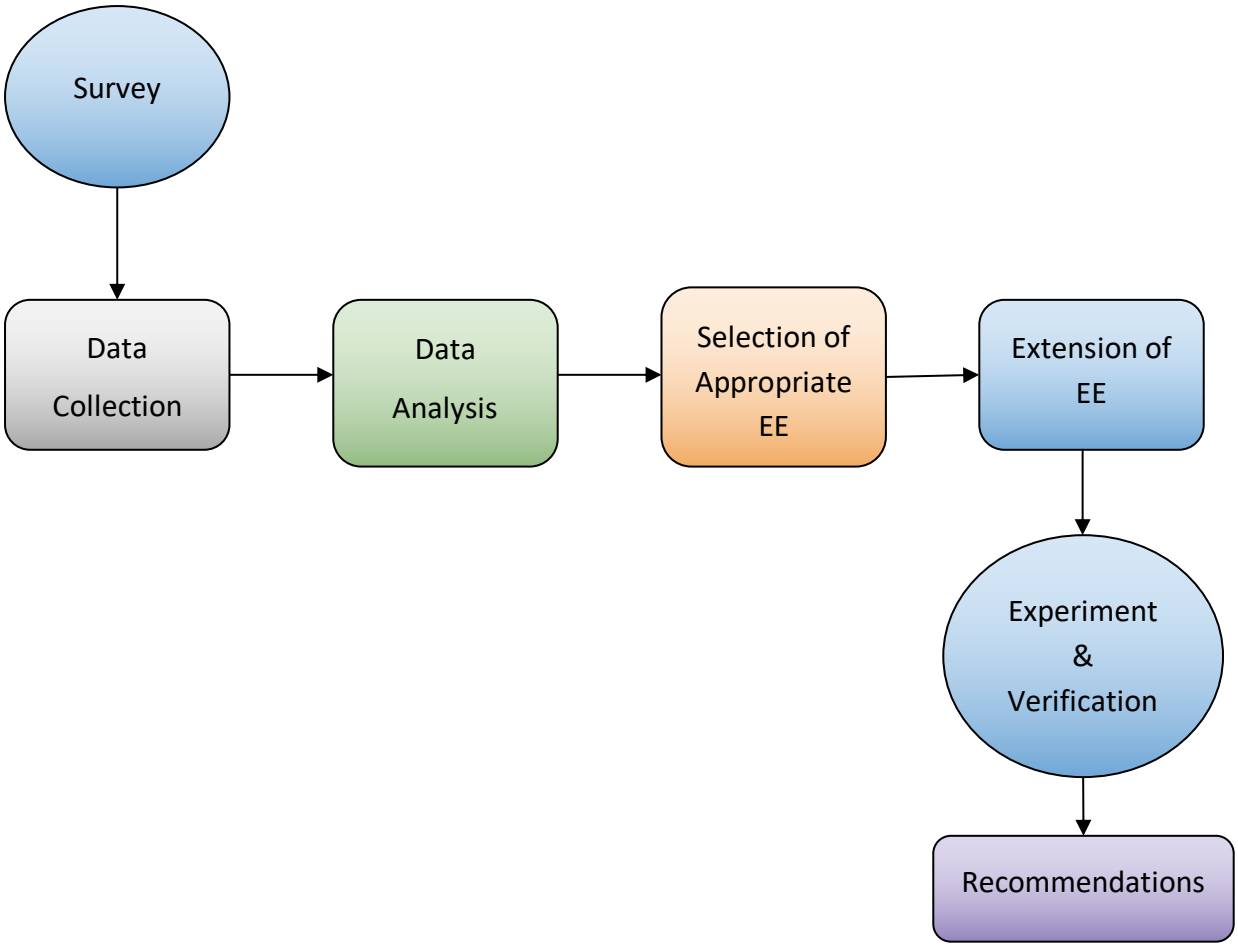

Figure 4.1: The steps of implement the methodology model

4.1 Survey Design

The stage of knowing what techniques and methods is used in Palestinian companies is important and the goal is to improve these methods or suggest new ways to help Palestinian companies improve the effort and cost estimate for software projects.

Knowing the current situation in this field and the current methods used by Palestinian companies to estimate effort was through a survey on Google Forms Published in 2021 consisting of a set of questions whose aim is to collect information about the current situation in Palestinian companies about effort estimation. Survey questions were designed to be

closed and open-ended, in which, respondent can add more information to the answers. We also took into account that the questionnaire should not be long and not take more than ten minutes. The idea of writing text also avoided most of the answers in the form of multiple choice or adding another answer, all in order to obtain accurate and correct data. The questionnaire was published on social networking sites for the largest gathering of the IT sector in Palestine (Peeks and PalGeeks) and also sent to some companies such as Asal and Crypton. Table 4.1 below shows the list of survey questions and explain the objective of each question.

Table 4.1A: Survey questions and objectives.

Id	Question	Objective
1	What is your current job title?	To have a general look on the software development jobs titles to which we can know if there are any effort estimation related titles.
2	How many years of experience do you have in software development?	To find out the effect of the number of years of experience in estimating effort.
3	How many employees work in your company?	To know the size of the company
4	Have you ever participated in/run level of effort estimation or you participate in/run level of effort	To understand his experience and share in estimating effort in projects.

Table 4.1B: Survey questions and objectives.

	estimation as part of your current job?	
5	If I suggested you a technique in calculating the estimation of effort, would you use it?	To understand the need for a technique to help him calculate the effort estimate.
6	What techniques do you use in effort estimation?	To understand the technique used in estimating effort in the company
7	What do you think about your level of Effort Estimation?	To understand and know if the responded have previous knowledge in software effort estimation.
8	How do you get the system requirements? Either in the form of a Request for Proposal (RFP), Tender, Business Requirements or Technical Specifications.	To understand the system requirements that are obtained in companies in order to start calculating the system effort estimation and know how clear the system is.
9	Have you previously encountered an inaccurate effort estimation in one of the projects? If yes, what is the percentage of deviation from the actual time?	To know the accuracy of the effort estimation technique
10	What is the unit of measurement for the effort rating output of the	To know the unit of measurement for estimating effort and cost either hour /

Table 4.1C: Survey questions and objectives.

	system?	days / months / weeks
11	Do you calculate risk factors when calculating the effort estimate? If the answer is yes, what is the percentage?	To find out whether risk factors are taken into account because they affect the accuracy of the results of the effort estimation
12	How many hours does your effort estimation technique take?	To understand whether the technique used takes time and effort

4.2 Data collection

In this section, we collected data for previously implemented software projects in Palestinian companies in order to apply the effort estimation technique and verify that it is suitable for software projects in Palestine. We sent a request to various Palestinian companies to obtain data for previous projects, some of which responded and sent to us, and some of them reserved because of privacy.

The collected software projects are diverse in their size, ideas, programming languages that were implemented using them, and the size of the companies, whether they are start-up companies or out-sourcing companies such as Asal company, Crpton company and other companies.

The data collected was about the form of requirements for the project, how many developers have worked on these projects, the techniques used, the time taken to complete the project and other information of interest to us for EE Technique.

4.3 Data Analysis

In this section, we analyzed the data collected from Palestinian companies and extracted from them the data that are of interest to us for the application of energy efficiency technique, and we summarized the analysis of this collected data in the form of a table.

Table 4.2 shows information on the projects collected from Palestinian companies. The number of projects is 14.

Note, we did not obtain permission from the Palestinian companies to display all the data in more detail than this

Table 4.2 A: Dataset for software projects in Palestinian companies.

Brief Description of the project	**Basis of Estimation**	**Technology Stack**	**Level of Certainty**	**Items Added (During Implementation)**	**Risk Factor Used**	**Actual Effort /unit**
Web application that is using analytical hierarchical process to help people who want to buy mobile phones to compare different options	Mockups, requirements gathering interviews	PHP, MySQL	High	Yes	10%	180 / Hours

Table 4.2 B: Dataset for software projects in Palestinian companies.

Mobile app for university students - Phase I	Design, Technical Specification	Flutter cross platform	High	Yes	15%	140/ Hours
Integration solution with facebook conversion APIs and Google ads	Business requirements, research	Netcore, MongoDB	Low	Yes	15%	150/ Hours
Mini CRM solution for small businesses	Mockups, database design, requirements gathering interviews	ASP.NET MVC5, SQL Server	High	Yes	10%	60/Days
Online learning platform for short videos - Phase I	Requirements gathering interviews	VueJS, Netcore	Medium	Yes	15%	40/Days
Online learning platform for short videos - Phase II	Requirements gathering interviews	VueJS, Netcore	Medium	Yes	10%	50/Days

Table 4.2 C: Dataset for software projects in Palestinian companies.

Web application for submitting business ideas for evaluation based on Lean canvas	Requirements gathering interviews	PHP, MySQL	Medium	No	10%	30/Days
Web application for submitting internship project ideas to industry professionals	Workshop	ASP.NET MVC5, SQL Server	High	No	10%	55/Days
Web application for students to submit mini research papers for review	Requirements gathering interviews	PHP, MySQL	Medium	No	15%	40/Days
Website for election campaign	Content, iterative feedback on design	WordPress	High	No	10%	45/ Hours
Archiving system for insurance agency	Requirements gathering interviews	ASP.NET MVC5, SQL Server	High	No	10%	220/ Hours
Automated process to generate study groups	Technical specification	C#	High	No	20%	25/ Hours

Table 4.2 D: Dataset for software projects in Palestinian companies.

Crpton Reception, Application to receive patients in dental clinics to organize appointments	Requirements gathering interviews	React with Ionic, ASP.NET MVC5, PostgreSQL	Medium	Yes	15%	34/ Days
Crpton Mobile App, an application for managing dental clinics, managing appointments, treatment plans and financial costs for patients	Requirements gathering interviews	React with Ionic, ASP.NET MVC5, PostgreSQL	Medium	No	15%	32/ Days

4.4 Selection of Appropriate EE Techniques

In this section, After the data of software projects has been collected from Palestinian companies and analyze it, an EE Technique that fits this data will now be identified and selected. The technique that we will apply in the experiment is Triangulation technique.

There are several reasons and motives for choosing this technique over others, which is that the data which has been collected from the IT Palestinian companies is useful for the Triangulation technique. On the other hand, this data didn't go with other techniques like cocomo technique.

For instance, the web project takes into consideration three main parts like Front-end, Back-

end and QA. Most other techniques don't take those parts into account.

Moreover, the Triangulation technique goes perfectly with the Agile methodology; that is, some techniques the time for the whole project for one time only. However, Triangulation allows the user to set time estimation for each sub task for example.

Undoubtedly, the Triangulation technique supports effort and a time-consuming approach. In other words, some project managers and Software engineers face the issue of the complexity of using some techniques especially when they cost them much time and effort to set time estimation.

Factors for choosing estimation techniques have also been mentioned in a survey to estimate the cost effect of different programmers, The researchers stated that no single method is available for accurate estimates than others. Each method affects different factors [31]:

1) Is model having an understandable structure and Process?
2) Is method less expensive, easier to use and clear?
3) Does the model keep missing or past data?
4) Does model consider uncertainty?
5) How model consider uncertainty?
6) Is estimation result being accurate or not?
7) Is method using agile methods or not?
8) Is method being dynamic or not?
9) Is method produce accurate output if the requirement changes during development?
10) Is the model less time consuming?
11) Is model fit in the current situation?
12) Are model produce trustworthy results?

This is a set of reasons that made me choose this technique and start applying it to the data set obtained from Palestinian companies.

4.5 Extension of Triangulation Technique

In this section, we will explain the framework that we have added to the triangulation technique, whose input depends on the results of the technique. This framework was developed by me based on my experience and the experiences of project managers in Palestinian companies.

This framework will help Palestinian companies in adjusting the delivery time of the project and adjusting the cost of the project in terms of knowing more resources such as the number of developers, the number of quality engineers and project managers, also in short to get the total working days and calendar days.

This framework is used in the phase of calculating the estimation of effort for the project, which depends on the technique of calculating the estimation of effort, which is triangulation, after the time required to complete the project is calculated from the triangulation technique, this time if the client is requested to complete the project with a delivery time less than the time that was completed It is calculated from the technique of estimating the effort, here comes the role of this framework, which is calculated for you with certain variables. You can change the delivery time by, for example, adding more software engineers, these inputs are entered on this framework and do mathematical operations, and then you get a new time to finish the project and accordingly it is determined the price you want based on the sources you added, by calculating developer hours to calculate developer hourly costs.

To understand the variables and calculations more for this framework, I added a table that

expresses simulations of how to deal with this framework and get the new project delivery time and cost. Table 4.3 shows the simulations of this proposed framework.

Table 4.3 A: Simulation of the proposed framework.

Estimated Time	60 / Hours	
No. of Junior Developers	1	
No. of Mid-Level Developers	0	
No. of Senior Developers	0	
No. of Junior QA	1	
No. of Mid-Level QA	0	
No. of Senior QA	0	
Actual working hours per day	6،5	
Management overhead	0،2	
Acceptance test factor	0،2	
Risk factor	0،4	
Reduction Factor	0	
Total number of resources	**2**	
Acceptance test period	**12**	
Estimated time including acceptance test	**72**	
Risk hours	**28،8**	

Table 4.3 B: Simulation of the proposed framework.

Estimated time including risk factor	**100،8**	
Junior Developer man hours	**100،8**	Price
Mid-Level Developer man hours	**0**	
Senior Developer man hours	**0**	
Junior QA man hours	**100،8**	
Mid-Level QA man hours	**0**	
Senior QA man hours	**0**	
PM man hours	**20،16**	
Calendar days	**23،26153846**	Deadline
Calendar months	**0،7753846154**	

Here you explain all the variables in the table and the output:

- **Estimated Time**

  This value is obtained from the triangulation technique and is the value of the estimated time of the project for which the effort estimate is calculated Without multiplying the value of the risk factor in it, because this value will be used in the framework, value is measured either in hours, days, or Estimation Units from triangulation output value.

- **No. of Developers**

  It means the number of developers proposed to be added to the project in order to reduce the delivery time and this also affects the increase in the cost of the project, which includes developers with different experiences and they are Junior, Mid-Level and Senior Developer. The idea of separating each experience is that developers' costs vary based on their experience.

- **No. of QA**

  It means the number of QA engineers for the project, and of course, this increase or decrease in the number affects the cost of the project and the time of delivery of the project as well. Also, the idea of separating each experience is that QA engineers costs vary based on their experience.

- **Actual Working Hours per Day**

  It means the number of working hours for the team, which is usually 6.5 hours, based on the Working Hours in business day.

- **Management Overhead**

  It means the number of working hours per day for the project manager, and a project manager can be added to the text of a working day or a third of a working day for the project.

- **Acceptance Test Factor**

  Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use. Also, It is a period

after the completion of the project development, during which the client checks the system to ensure that it has met all its requirements, and this period often affects the calendar day.

Therefore, it must be taken into account in calculating the estimation of effort and often affects 10% of the total Estimated Time.

- **Risk Factor**

  This value is obtained from the triangulation technique and the risk factor is an input from the triangulation technique inputs.

- **Reduction Factor**

  It means Percentage to reduction in project delivery time, in the case of adding two developers to the project, it does not mean that you reduced the time in half. In the case of adding one developer, you reduced the time by only 35% and not half, and this value also depends on whether the system is modular or not, for example, there are projects that are not affected by adding another developer because it is not modular. There are also tasks that can be worked on asynchronously and tasks that depend on each other all are taken into consideration. also, the value of Reduction Factor Decreases as the number of resources increases.

- **Total Number of Resources**

  This value is the sum of two values, which is No. of Developers and No. of QA, it is explained by the following equation:

Total Number of Resources = No. of Developers (Junior, Mid-Level and Senior) +

No. of QA (Junior, Mid-Level and Senior)

- **Acceptance Test Period**

  This value is the multiplication operation of the estimated time value and the value of the acceptance test factor, it is explained by the following equation:

  Acceptance Test Period = Estimated Time * Acceptance Test Factor

- **Estimated Time Including Acceptance Test**

  This value is the sum of two values, which is estimated time and Acceptance Test Period, it is explained by the following equation:

  Estimated Time Including Acceptance Test = Estimated Time + Acceptance Test Period

- **Risk Hours**

  This value is the multiplication operation of the Estimated Time Including Acceptance Test value and the value of the Risk Factor, it is explained by the following equation:

  Risk Hours = Risk Factor * Estimated Time Including Acceptance Test

- **Estimated Time Including Risk Factor**

  This value is the sum of two values, which is Estimated Time Including Acceptance Test and Risk Hours, it is explained by the following equation:

Estimated Time Including Risk Factor = Risk Hours + Estimated Time Including Acceptance Test

- **Total developers Man Hours**

  Here, the number of developers working hours is calculated according to each experience through the following equation:

  (Junior / Mid-Level / Senior) Developer man hours = Estimated Time Including Risk Factor * No. of *(Junior / Mid-Level / Senior)* Developers * (1 - Reduction Factor);

- **Total QA Man Hours**

  Here, the number of QA engineers working hours is calculated according to each experience through the following equation:

  (Junior / Mid-Level / Senior) QA man hours = Estimated Time Including Risk Factor * No. of *(Junior / Mid-Level / Senior)* QA * (1 - Reduction Factor);

- **PM Man Hours**

  This value is obtained from the following mathematical equation:

  PM Man Hours = Estimated Time Including Risk Factor * Management Overhead * (1 - Reduction Factor);

- **Calendar Days**

  This value is obtained from the following mathematical equation:

  Calendar Days = Estimated Time Including Risk Factor * 1,5 * (1 - Reduction Factor) / Actual Working Hours per Day;

- **Calendar Months**

  This value is divided by the value of the calendar days by 30 days

  Calendar Months = Calendar Days / 30;

Chapter 5

Results Experiment and Verification

In this chapter, we present the experiments and results of the experiments and also present a method to validate these results. In Section 5.1 We will display the results of our survey, In Section 5.2 we will explain the details of the application of triangulation technique and the projects that have been applied and the results of this technique, In Section 5.2 we will explain the details of the application of Extension of Triangulation Technique and the projects that have been applied and the results of this extension, finally in Section 5.3 we will show him the verification of triangulation technology and Extension of Triangulation Technique.

5.1 Survey Results

We have reached nearly 112 people who have filled out the survey on Google Forms. Here are results for those who filled out the survey:

- Regarding a question, what is your current job title?

  The results were as follows:

112 ردًا

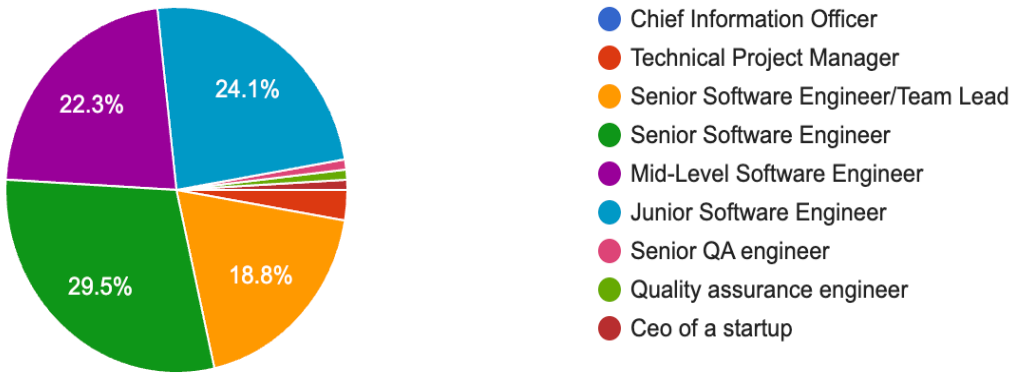

Figure 5.1: what is your current job title

- Regarding a question, how many years of experience do you have in your Field?

  The results were as follows:

112 ردًا

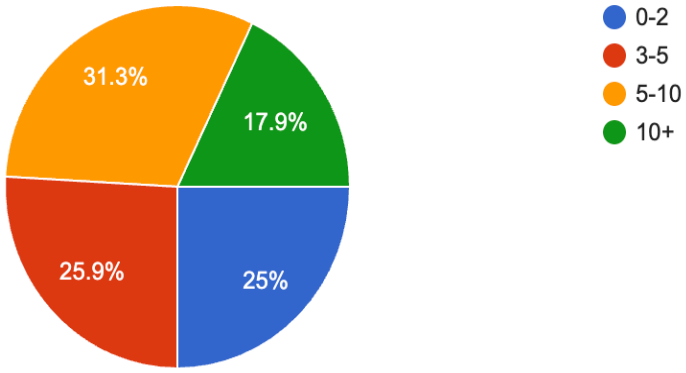

Figure 5.2: how many years of experience do you have in your Field

- Regarding a question, how many employees work in your company?

  The results were as follows:

112 ردًا

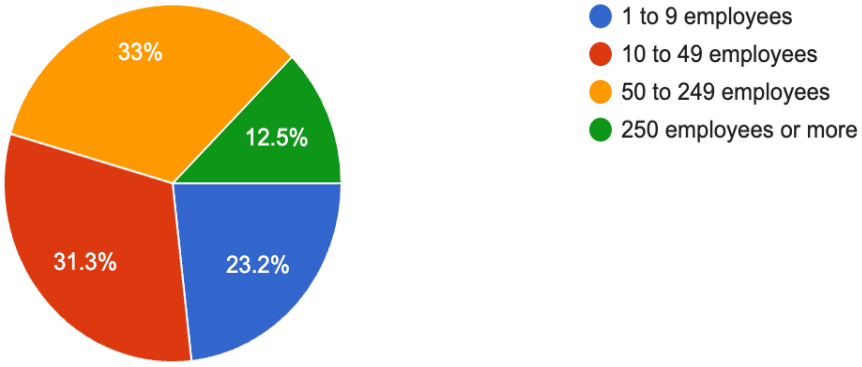

Figure 5.3: how many employees work in your company

- Regarding a question, have you ever participated in/run level of effort estimation or you participate in/run level of effort estimation as part of your current job?

  The results were as follows:

112 ردًا

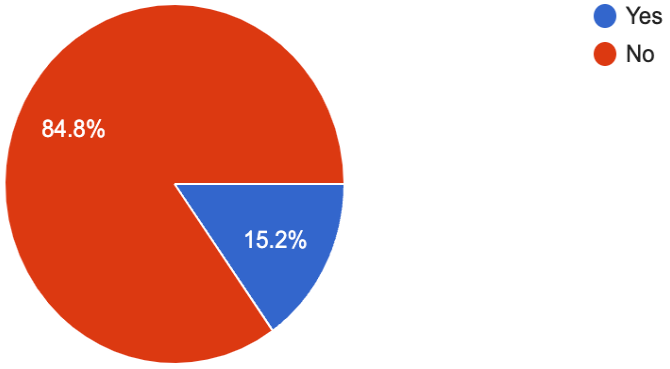

Figure 5.4: have you ever participated in/run level of effort estimation or you participate in/run level of effort estimation as part of your current job

- Regarding a question, If I suggested you a technique in calculating the estimation of effort, would you use it?

The results were as follows:

95 ردًا

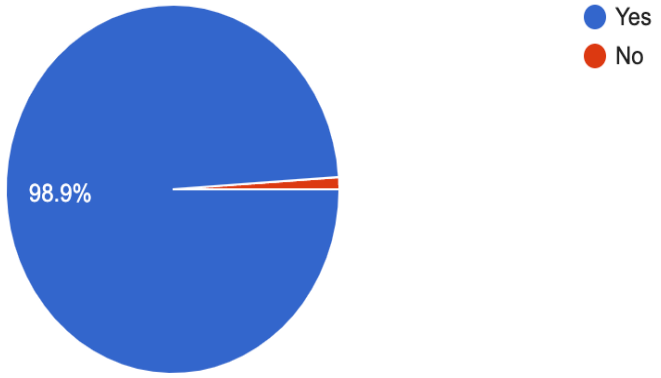

Figure 5.5: If I suggested you a technique in calculating the estimation of effort, would you use it

- Regarding a question, what techniques do you use in effort estimation?

The results were as follows:

17 ردًا

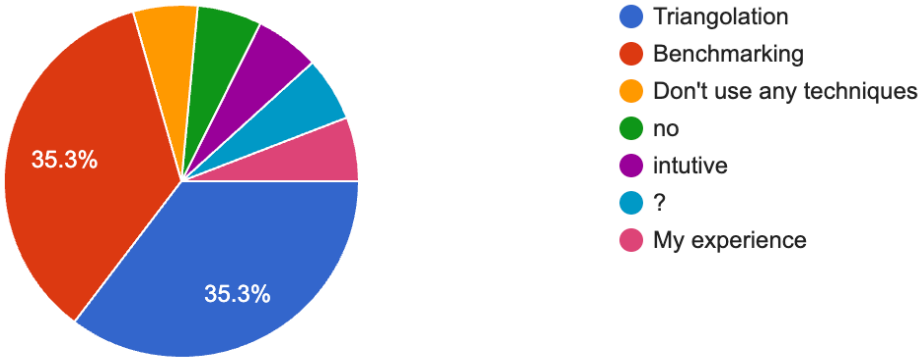

Figure 5.6: what techniques do you use in effort estimation

- Regarding a question, what do you think about your level of Effort Estimation?

The results were as follows:

17 ردًا

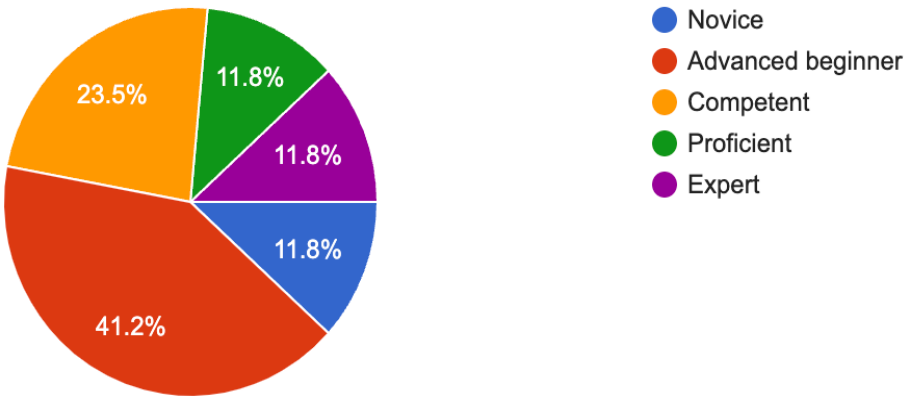

Figure 5.7: what do you think about your level of Effort Estimation

- Regarding a question, how do you get the system requirements?

The results were as follows:

17 ردًا

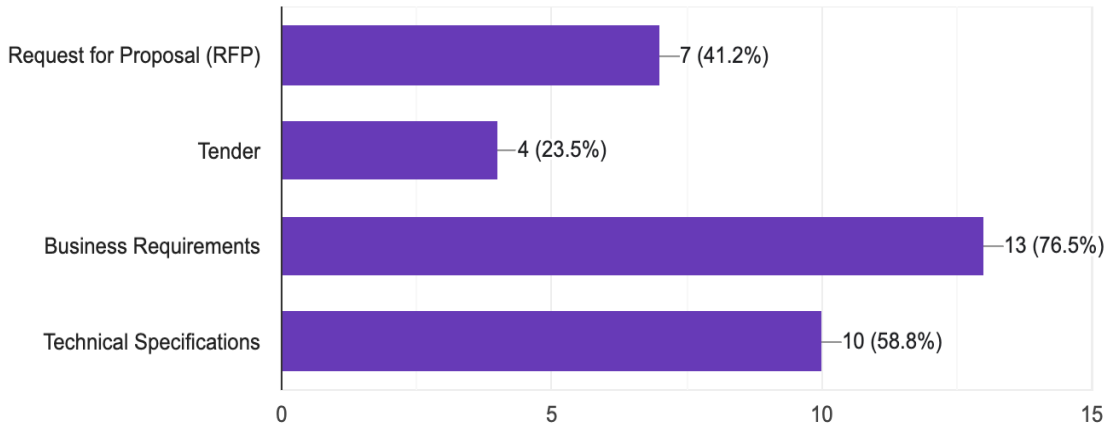

Figure 5.8: how do you get the system requirements

- Regarding a question, have you previously encountered an inaccurate effort

estimation in one of the projects? If yes, what is the percentage of deviation from the actual time?

The results were as follows:

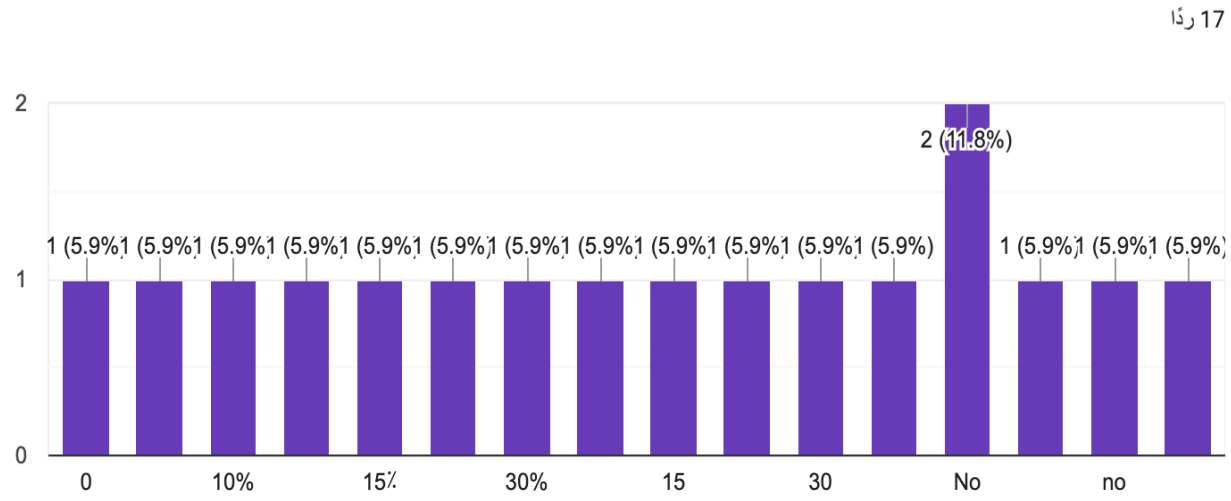

Figure 5.9: have you previously encountered an inaccurate effort estimation in one of the projects? If yes, what is the percentage of deviation from the actual time

- Regarding a question, what is the unit of measurement for the effort rating output of the system?

  The results were as follows:

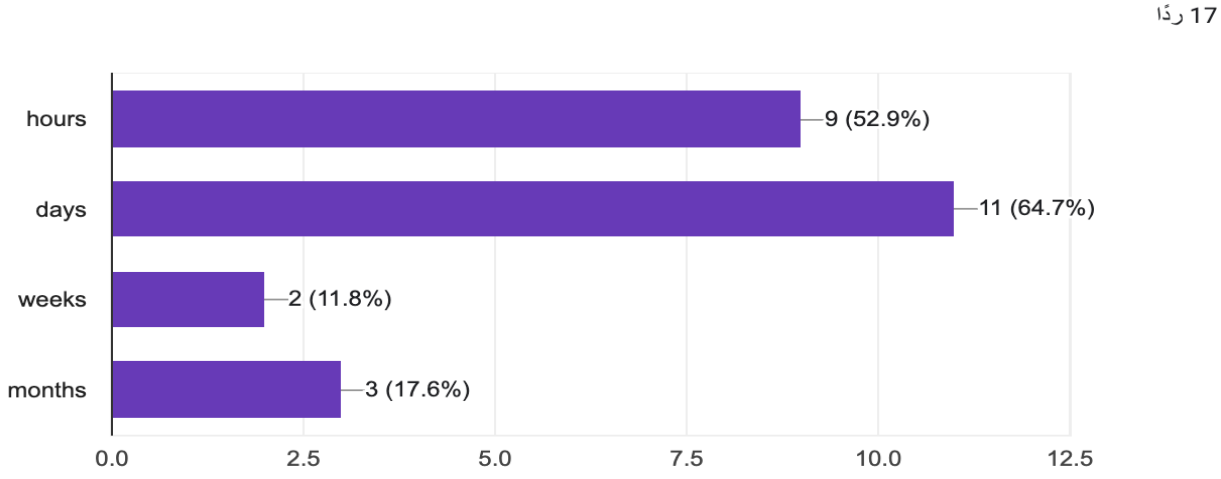

Figure 5.10: what is the unit of measurement for the effort rating output of the system

- Regarding a question, do you calculate risk factors when calculating the effort estimate? If the answer is yes, what is the percentage?

The results were as follows:

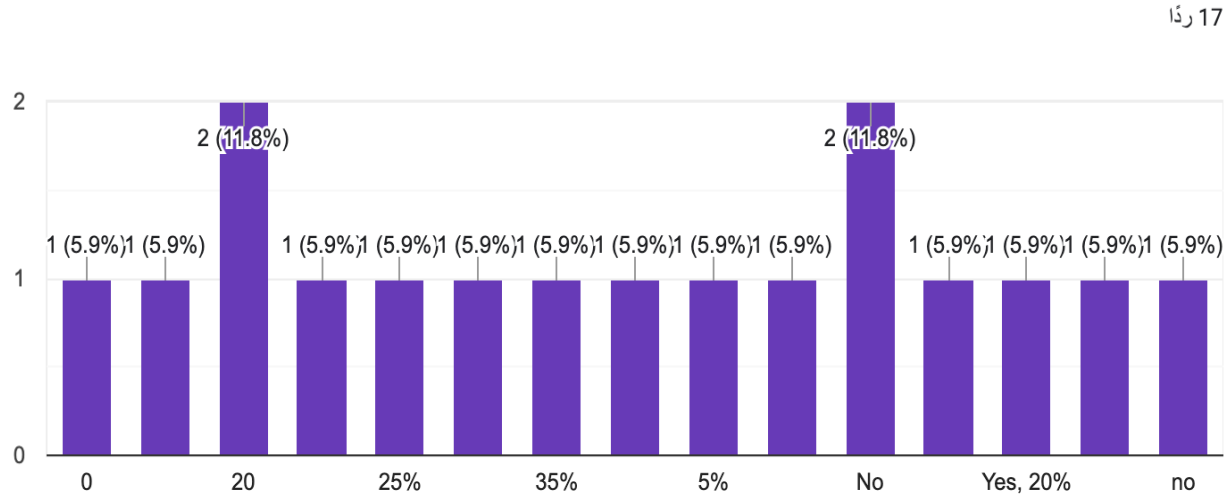

Figure 5.11: do you calculate risk factors when calculating the effort estimate? If the answer is yes, what is the percentage

- Regarding a question, how many hours does your effort estimation technique take?

The results were as follows:

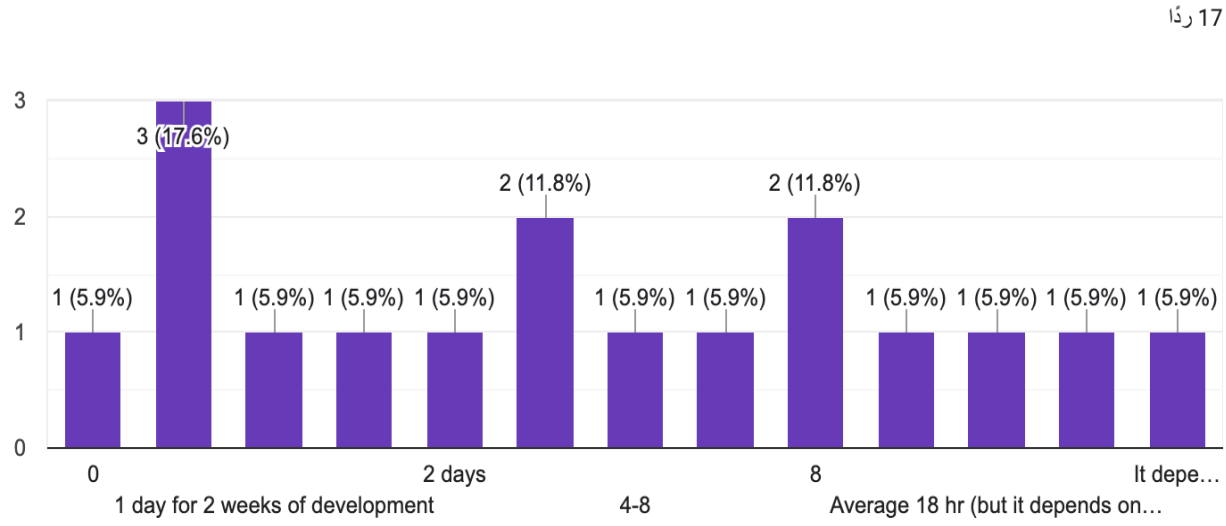

Figure 5.12: how many hours does your effort estimation technique take

Based on our survey, we show that a sample of 95 between a software engineer and a project manager out of 112 do not use any effort estimating technique in Palestinian companies. When asked if there is a motivation to use a technique to help them calculate the effort estimate, most of the answers were yes 94 out of 95 and one answer was no. and I also got the look of those who answered that they use a technique for estimating effort about what technique they used.

5.2 Experiment EE Technique

In this section, I will explain the details of the application of triangulation technique. I did not get approval from the companies to mention data details about the software projects that I obtained, so I will explain the details of triangulation technique in this section.

5.2.1 Inputs

In the beginning, I will explain the inputs for any project that takes several forms. That is, the initial description of the project before starting it, which gives a description of what the project is, which is delivered to any software company to know what the project wants, what its goal is and what it described. All of this falls under the inputs.

The input is either in the form of a Request for Proposal (RFP), a Tender, a Business Requirements, or a Technical Specification.

Table 5.1 shows the inputs for projects as well as the level of details, which is meant by the clarity of this input, and the higher the degree of clarity, the more clear its details, and

therefore the estimate of the effort will be more accurate because of the clarity of these details of the project

Table 5.1: Input Forms of a Software Project

Input	Level of Details
Request for Proposal (RFP)	Low
Tender	High
Business Requirements	Moderate
Technical Specification	High

I will briefly explain these concepts:

- **Request for Proposal (RFP)**

  A request for proposal (RFP) describes a project in detail so potential vendors can submit proposals for the development process, technology and so on. Complex software development projects demand a lot of detailed information, so with an RFP, companies are able to address many products development challenges by collecting ideas from several vendors [32].

  A detailed RFP for software development allows issuers to compare vendor proposals, evaluate different ways to solve a single problem. It also helps save valuable time since one company can send project information to more than one vendor and later compare their approaches [32].

  Companies use an RFP to exchange details about projects and set up partnerships and collaborations. Once a company releases information about a new project, competitive

tendering begins, as potential vendors start proposing their solutions. An RFP is beneficial for a variety of reasons, for both issuers and vendors. Since it is often used for large, complex projects, it allows companies to get a number of different solutions, examine them in detail, and decide which would best solve their problem. Higher accuracy of project estimates. With a standardized RFP for software development, it becomes easier to explain a specific project and receive realistic project estimates. Understanding how many customers a business has, in which stage their product is, or whether the company has come up with an idea and is looking for someone to build it. Overall, the RFP document provides a context that vendors can use to provide a more accurate timeframe, project scope, price range, and use—to generate tailor-made offers [32].

- **Business Requirements**

  Business requirements is a phase in Software development life cycle which felicitates the requirements of the end users as the very first task in order to guide the design of the future system. Business requirements are usually captured by business analysts or product owners who analyze business activities who in turn act as subject matter expertise (SME's) [33].

  Requirements are essential to ensuring that all stakeholders and other team members are on the same wavelength as the software development group. They act as a starting point for a project's development process as they keep all team members aligned to a single, clearly defined goal. High quality, detailed business requirements documentation also helps projects within budget and ensures it is complete within the desired time-frame or schedule. According to the Business Analysis Body of

Knowledge. The process of defining a company's business requirements involves considering the tasks and goals of the business and customers' needs. Unsurprisingly this can be a complex task and requires input and questions from various people involved throughout the organization a company's business requirements to be useful and attainable, there are some tools and steps that can be taken during the requirement gathering process to achieve the best results [34].

- **Technical Specification**

  A technical specification is a detailed and comprehensive document that describes all technical procedures related to product development. It covers all the vital, nitty-gritty information about the process of product development. A technical specification, especially written using a good template, is like your product development bible [35].

5.2.2 Steps to Get the Estimated Time by Triangulation

In this section, we are going to present the main steps we can follow to know the estimated time by following the Triangulation technique. It's important to know that these steps depend on each other. That is, every single output of a step is considered to be the main input of the following step. The following table 5.2 shows the steps.

Table 5.2: The Triangulation Main Steps for Estimating the Time.

Steps	Note
Functional Decomposition	Missing items are a big risk
Running Triangulation Technique Benchmarking (optional)	
Getting The Total	
Multiply by a Risk Factor	10% - 35%
Record Assumptions	

- **Functional Decomposition**

We consider the Functional Decomposition concept to divide the input into items. In other words, in case the input is a technical specification, we divide the system parts into items, so the result is a divided output of the input and that output will be used for the following step. Moreover, the divided items can help in recognizing the level of effort of each item and the deliverables.

The person who is in charge of dividing the item should be an expert in the domain like a team lead, senior and so on. However, the functional decomposition has a risk which can be basically related to forgetting any detail between the lines.

- **Running Triangulation Technique**

The triangulation discusses the rule of thirds. The principle of this technique invites three software engineers with three different seniority levels. We distribute playing

cards based on the Fibonacci rule while paying attention to the different seniority of the three engineers. The Fibonacci starts from 1 and ends at 8. It's impossible to exceed the 8 number because the items in this case must be decomposed (to get in the functional decomposition process).

By way of illustration, we create a scenario for a specific item

- Scenario 2,2,2: Three engineers select the same card. For example, the three persons choose 2 for this item, so in this case the selected estimated time for this item is 2.
- Scenario 2,3,3: One of the three engineers selects 2 while the other two engineers select 3. The result of the selected estimated time for this item is 3.
- Scenario 2,2,3: The two engineers agree on selecting 2 while the other engineer decides to go for 3. In this scenario, the estimator of 3 should justify the reason behind selecting this card and then replay.
- Scenario 2,3,5: If one engineer selects 2 and the other one decides to choose 3 and the last engineer goes for 5, then the estimators of 2 and 5 need to justify why they have estimated this time and then replay. If the results are still the same, then we select 5. The reason behind considering card 5 not the 2 is mainly related to decreasing the risk of time estimation. The item might really need this time according to the estimator vision.

- **Benchmarking**

This step can't be considered as a mandatory step but can be categorized as optional to assure the results. Since Benchmarking is mainly based on comparing the previous projects results with the current one, we can use this step in some projects that have similar results. we can use this technique just to confirm the results we already

worked on. However, some projects are new and we don't have previous similar projects, so we need to start from scratch. In that case, there is no way to use Benchmarking and that's why it's optional.

- **Getting The Total**

  Here, after completing the calculation of all the items by the triangulation technique in the previous step, we perform the process of summing all these results and obtaining a number. This number is the sum total of the estimates for each item.

- **Risk Factor**

  The risk factor is defined as a condition that may constitute an inaccuracy of the effort estimation when calculating the EE using the triangulation technique , the proportion of the risk factor is determined by an expert in estimating effort and is often between 10% - 35% , often this percentage is determined based on several criteria or reasons such as: Unclear inputs , during the process of applying the triangulation technique Senior people are giving optimistic estimates or Some people are giving very high estimations , when the technology (such as the programming language) to be implemented in the project, the members of the programming team do not have any previous experience with it and in case the system to be worked on is complex.

- **Record Assumptions**

  When calculating the Effort estimate, there is a hypothesis that has been relied upon in the calculation. For example, assuming the project is a web application, the premise is that we will develop this project in terms of the back-end being C-Sharp language, the front-end being React.js, and the database being MongoDB. Based on this

assumption, we have made an estimate of the project effort, because often in the inputs for software projects, the client did not specify what technology stack, and therefore we estimate the effort of the program project based on these assumptions because if the technology is changed, the value of the estimate effort will change the simplest reason is that the development team may not have any experience with the technology required.

5.2.3 Triangulation Technique Results

In this section, we will present the results of applying the triangulation technique to 14 software projects from Palestinian companies in Table 5.3.

Table 5.3 A: Dataset for Results of the application of the triangulation technique.

Brief Description of the project	**Basis of Estimation**	**Technology Stack**	**Level of Certainty**	**Items Added (During Implementation)**	**Risk Factor Used**	**Estimated Efforts**
Web application that is using analytical hierarchical process to help people who want to buy mobile phones to	Mockups, requirements gathering interviews	PHP, MySQL	High	Yes	10%	167 / Hours

Table 5.3 B: Dataset for Results of the application of the triangulation technique.

compare different options						
Mobile app for university students - Phase I	Design, Technical Specification	Flutter cross platform	High	Yes	15%	120 / Hours
Integration solution with Facebook conversion APIs and Google ads	Business requirements, research	Netcore, MongoDB	Low	Yes	15%	119 / Hours
Mini CRM solution for small businesses	Mockups, database design, requirements gathering interviews	ASP.NET MVC5, SQL Server	High	Yes	10%	61 / Days
Online learning platform for short videos - Phase I	Requirements gathering interviews	VueJS, Netcore	Medium	Yes	15%	45 / Day

Table 5.3 C: Dataset for Results of the application of the triangulation technique.

Online learning platform for short videos - Phase II	Requirements gathering interviews	VueJS, Netcore	Medium	Yes	10%	48 / Day
Web application for submitting business ideas for evaluation based on Lean canvas	Requirements gathering interviews	PHP, MySQL	Medium	No	10%	32 / Day
Web application for submitting internship project ideas to industry professionals	Workshop	ASP.NET MVC5, SQL Server	High	No	10%	50 / Day
Web application for students to submit mini research papers for review	Requirements gathering interviews	PHP, MySQL	Medium	No	15%	35 / Day
Website for election campaign	Content, iterative feedback on design	WordPress	High	No	10%	40 / Hours

Table 5.3 D: Dataset for Results of the application of the triangulation technique.

Archiving system for insurance agency	Requirements gathering interviews	ASP.NET MVC5, SQL Server	High	No	10%	241 / Hours
Automated process to generate study groups	Technical specification	C#	High	No	20%	16.5 / Hours
Crpton Reception, Application to receive patients in dental clinics to organize appointments	Requirements gathering interviews	React with Ionic, ASP.NET MVC5, postgresql	Medium	Yes	15%	36 / Day
Crpton Mobile App, an application for managing dental clinics, managing appointments, treatment plans and financial costs for patients	Requirements gathering interviews	React with Ionic, ASP.NET MVC5, postgresql	Medium	No	15%	33 / Day

5.3 Experiment the Extension of Triangulation Technique

Here, I applied this Extension on 3 projects from Palestinian companies, the experiment was over a certain period of the project, the results of applying this Extension are in Table 5.4.

The Table 5.4 shows the details of each project separately in terms of the number of developers and the number of QA engineers and the rest of the details the inputs shown in the table and the results of calculating the project completion time and cost details for all sources separately in the number of hours.

Table 5.4 A: Dataset for Results of application for Extension of Triangulation Technique.

	Mobile app for E-School	**Web Application - Online store for a Palestinian company**	**Web App for insurance company, can automatically analyze, and process vehicle damage at high precision without skilled personnel**	
Estimated Time (Hours)	65	130	195	
No. of Junior Developers	2	1	1	
No. of Mid-Level Developers	0	0	1	
No. of Senior Developers	0	1	1	
No. of Junior QA	1	0	1	
No. of Mid-Level QA	0	1	0	

Table 5.4 B: Dataset for Results of application for Extension of Triangulation Technique.

No. of Senior QA	0	0	0	
Actual working hours per day	6،5	6،5	6،5	
Management overhead	0،2	0،3	0،3	
Acceptance test factor	0،1	0،1	0،2	
Risk factor	0،3	0،2	0،15	
Reduction Factor	0،35	0،35	0،35	
Total number of resources	**3**	**3**	**4**	
Acceptance test period	**6،5**	**13**	**39**	
Estimated time including acceptance test	**71،5**	**143**	**234**	
Risk hours	**21،45**	**28،6**	**35،1**	
Estimated time including risk factor	**92،95**	**171،6**	**269،1**	
Junior Developer man hours	**120،835**	**111،54**	**174،915**	Price
Mid-Level Developer man hours	**0**	**0**	**174،915**	
Senior Developer man hours	**0**	**111،54**	**174،915**	
Junior QA man hours	**60،4175**	**0**	**174،915**	
Mid-Level QA man hours	**0**	**111،54**	**0**	
Senior QA man hours	**0**	**0**	**0**	

Table 5.4 C: Dataset for Results of application for Extension of Triangulation Technique.

PM man hours	12،0835	33،462	52،4745	
Calendar days	13،9425	25،74	40،365	Deadline
Calendar months	0،46475	0،858	1،3455	

5.4 Verification

The experiments aimed at applying the triangulation technique and knowing its suitability with the Palestinian market projects, and then establishing an extension of this technique in order to improve and assist Palestinian companies in estimating the effort and knowing more accurately the resources and costs of the projects, in order to evaluate the accuracy and validity of the triangulation technique and to evaluate the accuracy of the extension of this technique. The method I followed for the triangulation technique is the following. Firstly, I gathered the previous software projects that were implemented, and then I applied the mentioned technique on those projects. Secondly, I made a comparison between the actual results and the triangulation technique's results (Estimated Efforts). By doing that, it was possible to check whether the technique's results were accurate or not.

As for the method of verifying the accuracy of the triangulation technique extension, I had applied the triangulation technique extension to three projects that have software engineers and QA engineers then I applied the simulation of this extension of the triangulation technique and compared the results of this technique with the actual result of the projects, which focuses on the number of sources and working hours for each engineer in order to

make sure It is the accuracy of the extension of the triangulation technique in the event that developers are added to the project, the accuracy of the time reduction of project delivery it is accurate or not.

5.4.1 Verification of Triangulation Technique

Here I will present the results of verifying the accuracy of the triangulation technique for the software projects that were used in the experiment phase shown in Table 4.2, and they are 14 projects, Table 5.5 shows the results of checking the accuracy of the triangulation technique, the second column (Estimated Effort) shows the value of the time value of the triangulation technique and the third column shows the actual time for project delivery.

Table 5.5 A: Dataset of Triangulation Technique Verification Results.

Brief Description of the project	**Estimated Effort**	**Actual Effort**	**Estimation Unit**
Web application that is using analytical hierarchical process to help people who want to buy mobile phones to compare different options	167	180	Hour
Mobile app for university students - Phase I	120	140	Hour
Integration solution with Facebook conversion APIs and Google ads	119	150	Hour
Mini CRM solution for small businesses	61	60	Days

Table 5.5 B: Dataset of Triangulation Technique Verification Results.

Online learning platform for short videos - Phase I	45	40	Days
Online learning platform for short videos - Phase II	48	50	Days
Web application for submitting business ideas for evaluation based on Lean canvas	32	30	Days
Web application for submitting internship project ideas to industry professionals	50	55	Days
Web application for students to submit mini research papers for review	35	40	Days
Website for election campaign	40	45	Hour
Archiving system for insurance agency	241	220	Hour
Automated process to generate study groups	16.5	25	Hour
Crpton Reception, Application to receive patients in dental clinics to organize appointments	36	34	Days
Crpton Mobile App, an application for managing dental clinics, managing appointments, treatment plans and financial costs for patients	33	32	Days

Factors that depend on the estimated time not being close to the actual time is the risk factor and the Level of Certainty , These two factors mainly affect the accuracy of the estimated

time compared to the actual time, as is the case in Project No. 3, there is a high risk factor and a decrease in Level of Certainty, which is the clarity of the project from the requirements of the client ,Thus, I see that this technique has a high accuracy rate for Palestinian projects based on input factors that control the accuracy of this technique.

5.4.2 Verification for Extension of Triangulation Technique

Here I will explain the results of evaluating the accuracy of Extension triangulation Technique, I applied the Extension to three software projects in a certain period of time, their details are shown in the table 5.4, adding the same number of developers and QA engineers, and calculating the estimated time of the Extension and comparing it with the real time. Knowing the accuracy of time also gives us knowledge of the accuracy of the number of resources like developers and QA engineers.

Table 5.6: Verification Results of Extension of Triangulation Technique.

	Mobile app for E-School	**Web Application - Online store for a Palestinian company**	**Web App for insurance company, can automatically analyze, and process vehicle damage at high precision without skilled personnel**
Estimated Effort / Calendar days	**13,9425**	**25,74**	**40,365**
Actual Effort / Calendar days	15 days	26 days	40 days

We see here that the accuracy of the Extension of Triangulation Technique is high and close to the actual result. There are factors that affect the accuracy of this result, such as the risk factor as in the first project (Mobile app for E-School) and we also see that there is a result close to the actual result in the second two projects because the risk factor was less.

Chapter 6

Conclusion and Future work

6.1 Conclusion

We believe that this study is an important reference model and addition for Palestinian companies that can help them calculate the effort needed to perform their software projects in terms of cost, delivery time and number of resources. This model can be used as a guideline to get better and more accurate effort estimates. We chose the Triangulation technique, did intensive experimentation and validation using actual data and the results were very promising.

We have chosen the triangulation technique for several reasons, the most important of which is its suitability for the Palestinian IT companies as small-team / small-business projects. On the other hand, we have checked this data against other formal techniques (like COCOMO technique), and it found that they did not serve the needs of Palestinian IT sector.

We would like to add that we have designed the extension of the Triangulation method to be able to control and adjust the project delivery time by varying the allocated resources (e.g., the number of engineers). There were several factors taken into consideration such as the acceptance test factor, risk factor, reduction factor, and their relevant combinations

6.2 Future work

Future studies can adopt the experiment of the Extension of Triangulation Technique to other

software projects with different specifications and evaluate the accuracy of the technique, and there may also be other parameters that might affect the effort estimation accuracy (like the programming language, the work environment, the type of administration support, the skills levels of the engineers, etc.). These can be the basis for new studies in an attempt to broaden the technique for more cases and better accuracy and flexibility.

Bibliography

1. Lenarduzzi, V., Morasca, S., & Taibi, D. (2014, August). Estimating software development effort based on phases. In 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications (pp. 305-308). IEEE.
2. Nguyen-Cong, D., & Tran-Cao, D. (2013, November). A review of effort estimation studies in agile, iterative and incremental software development. In The 2013 RIVF International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for Future (RIVF) (pp. 27-30). IEEE.
3. Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. Journal of Engineering Science & Technology Review, 10(4).
4. Tailor, O., Saini, J., & Rijwani, M. P. (2014). Comparative analysis of software cost and effort estimation methods: a review. Interfaces, 5(7), 10.
5. Usman, M., Mendes, E., Weidt, F., & Britto, R. (2014, September). Effort estimation in agile software development: a systematic literature review. In Proceedings of the 10th international conference on predictive models in software engineering (pp. 82-91).
6. Rijwani, P., Jain, S., & Santani, D. (2014). Software effort estimation: a comparison based perspective. International Journal of Application or Innovation in Engineering and Management (IJAIEM), 3(12), 18-29.
7. Borade, J. G., & Khalkar, V. R. (2013). Software project effort and cost estimation techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 3(8).
8. Usman, M., Mendes, E., & Börstler, J. (2015, April). Effort estimation in agile

software development: a survey on the state of the practice. In Proceedings of the 19th international conference on Evaluation and Assessment in Software Engineering (pp. 1-10).

9. Usman, M., & Britto, R. (2016, October). Effort estimation in co-located and globally distributed agile software development: A comparative study. In 2016 joint conference of the international workshop on software measurement and the international conference on software process and product measurement (IWSM-MENSURA) (pp. 219-224). IEEE.
10. Chirra, S. M. R., & Reza, H. (2019). A survey on software cost estimation techniques. Journal of Software Engineering and Applications, 12(06), 226.
11. Jørgensen, M., Boehm, B., & Rifkin, S. (2009). Software development effort estimation: Formal models or expert judgment?. IEEE software, 26(2), 14-19.
12. Lenarduzzi, V. (2015, September). Could social factors influence the effort software estimation?. In *Proceedings of the 7th International Workshop on Social Software Engineering* (pp. 21-24).
13. Dagnino, A. (2013, May). Estimating software-intensive projects in the absence of historical data. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 941-950). IEEE.
14. Tanveer, B. (2017, June). Guidelines for utilizing change impact analysis when estimating effort in agile software development. In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (pp. 252-257).
15. Zein, S., Salleh, N., & Grundy, J. (2015, September). Mobile application testing in industrial contexts: an exploratory multiple case-study. In *International conference on intelligent software methodologies, tools, and techniques* (pp. 30-41). Springer,

Cham.

16. Zarour, A., & Zein, S. (2019). Software development estimation techniques in industrial contexts: An exploratory multiple case-study. *International Journal of Technology in Education and Science*, *3*(2), 72-84.
17. IIBA, A. (2015). Guide to the Business Analysis Body of Knowledge (BABOK Guide) Ver 3.0.
18. Trendowicz, A. (2013). *Software Cost Estimation, Benchmarking, and Risk Assessment: The Software Decision-Makers' Guide to Predictable Software Development*. Heidelberg: Springer.
19. Chirra, S. M. R., & Reza, H. (2019). A survey on software cost estimation techniques. *Journal of Software Engineering and Applications*, *12*(06), 226.
20. Popli, R., & Chauhan, N. (2014, February). Cost and effort estimation in agile software development. In 2014 international conference on reliability optimization and information technology (ICROIT) (pp. 57-61). IEEE.
21. Prakash, B., & Viswanathan, V. (2017). A survey on software estimation techniques in traditional and agile development models. Indonesian Journal of Electrical Engineering and Computer Science, 7(3), 867-876.
22. Zarour, A., & Zein, S. (2019). Software development estimation techniques in industrial contexts: An exploratory multiple case-study. International Journal of Technology in Education and Science, 3(2), 72-84.
23. Gumaei, A., Almaslukh, B., & Tagoug, N. (2015). An Empirical Study of Software Cost Estimation in Saudi Arabia Software Industry. International Journal of Soft Computing and Engineering (IJSCE), ISSN, 2231-2307.
24. Kafle, L. P. (2014). An empirical study on software test effort estimation. Int. J. Soft Comput. Artif. Intell, 2(2).

25. Tailor, O., Saini, J., & Rijwani, M. P. (2014). Comparative analysis of software cost and effort estimation methods: a review. Interfaces, 5(7), 10.
26. Mahmood, Y., Kama, N., Azmi, A., & Ali, M. (2020, June). Improving estimation accuracy prediction of software development effort: A proposed ensemble model. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) (pp. 1-6). IEEE.
27. Langley, M. A. (2017). Success Rates Rise: Transforming the high cost of low performance. *PMI, editor. Pulse of the Profession2017*, 32.
28. Popli, R., & Chauhan, N. (2012). Research challenges of agile estimation. *Journal of Intelligent Computing and Applications*.
29. Munialo, S. W., & Muketha, G. M. (2016). A review ofagile software effort estimation methods.
30. Trendowicz, A., & Jeffery, R. (2014). Software project effort estimation. *Foundations and Best Practice Guidelines for Success, Constructive Cost Model–COCOMO pags*, *12*, 277-293.
31. A, N., Qurashi, J.A., & Kumar, S. (2019). A Survey Study of Various Software Cost Effort Estimation in Perspective of India. *International Journal of Computer Sciences and Engineering*.
32. Vidjikant, S. (2023, January 4). How to Write a Request for Proposal for Software Development in 2023. Softjourn Inc.
33. *Business Requirement*. (n.d.-b).
34. Software Development UK: #1 Leading Software Professionals. (2021, July 13). Software Development UK.
35. All of us at monday.com. (2022, October 24). How to write a technical specification [with examples]. monday.com Blog.
36. Popli, R., & Chauhan, N. (2012). Research challenges of agile estimation. *Journal of*

Intelligent Computing and Applications.

37. Munialo, S. W., & Muketha, G. M. (2016). A review ofagile software effort estimation methods.
38. Boehm, B. W. (1981). Software engineering economics (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA).
39. Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. (1995). Cost models for future software life cycle processes: COCOMO 2.0. Annals of software engineering, 1(1), 57-94.
40. IFPUG, F. (2000). International Function Point Users Group (IFPUG) Function Point Counting Practices Manual.
41. Garg, K., Kaur, P., Kapoor, S., & Narula, S. (2014). Enhancement in COCOMO Model Using Function Point Analysis to Increase Effort Estimation. International Journal of Computer Science and Mobile Computing, 3(6), 265-572.
42. Boehm, B., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches—A survey. Annals of software engineering, 10(1), 177-205.
43. Rush, C., & Roy, R. (2001). Expert judgement in cost estimating: Modelling the reasoning process. Concurrent Engineering, 9(4), 271-284.
44. Leung, H., & Fan, Z. (2002). Software cost estimation. In Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies (pp. 307-324).
45. Sharma, S. (2017). Applications of genetic algorithm in software engineering, distributed computing and machine learning. International Journal of Computer Applications and Information Technology (IJCAIT), 9, 208-212.
46. Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. Information

and Software Technology, 54(1), 41-59.

47. Gray, A. R., & MacDonell, S. G. (1999). Software metrics data analysis—exploring the relative performance of some commonly used modeling techniques. Empirical Software Engineering, 4(4), 297-316.
48. Jeffery, R., Ruhe, M., & Wieczorek, I. (2000). A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data. Information and software technology, 42(14), 1009-1016.
49. Heiat, A. (2002). Comparison of artificial neural network and regression models for estimating software development effort. Information and software Technology, 44(15), 911-922.
50. Dolado, J. J. (2001). On the problem of the software cost function. Information and Software Technology, 43(1), 61-72.
51. Myrtveit, I., & Stensrud, E. (1999). A controlled experiment to assess the benefits of estimating with analogy and regression models. IEEE transactions on software engineering, 25(4), 510-525.
52. Sharma, P., & Singh, J. (2017, December). Systematic literature review on software effort estimation using machine learning approaches. In 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS) (pp. 43-47). IEEE.
53. Hamza, H., Kamel, A., & Shams, K. (2013, April). Software effort estimation using artificial neural networks: A survey of the current practices. In 2013 10th International Conference on Information Technology: New Generations (pp. 731-733). IEEE.
54. Rijwani, P., & Jain, S. (2016). Enhanced software effort estimation using multi layered feed forward artificial neural network technique. Procedia Computer Science,

89, 307-312.

55. Idri, A., Khoshgoftaar, T. M., & Abran, A. (2002, May). Can neural networks be easily interpreted in software cost estimation?. In 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291) (Vol. 2, pp. 1162-1167). IEEE.
56. Singh, B. K., & Misra, A. K. (2012). Software effort estimation by genetic algorithm tuned parameters of modified constructive cost model for nasa software projects. International Journal of Computer Applications, 59(9).

الملخص

تقنيات تقدير الجهد في تطوير البرمجيات - دراسة حالة لشركات البرمجيات الفلسطينية

تقدير الجهد هو إحدى طرق تقييم مشروع برمجي وفهم جدوله الزمني وميزانيته. إنه أحد أهم جوانب دورة حياة تطوير البرمجيات. قد يؤدي عدم التعرف على تقدير الجهد الدقيق إلى زيادة التكاليف المالية للشركات وعملائها مما يؤدي إلى تأثير سلبي على واجباتهم الوظيفية وخططهم التسويقية المستقبلية بالإضافة إلى خيبة أمل العميل وعدم رضاه.

يعتبر قطاع تكنولوجيا المعلومات الفلسطيني من أكثر القطاعات الواعدة والنامية. ومع ذلك، فإن الدراسات التي تبحث في أساليب وتقنيات تقدير الجهد هي على الأرجح مفقودة. لهذا السبب، كنا متحمسين لدراسة حالة شركات تطوير البرمجيات في فلسطين من أجل فهم أفضل لكيفية تقدير الفرق الفنية للجهود المطلوبة لمشاريعهم البرمجية.

الغرض من هذه الدراسة هو (1) مسح تقنيات تقدير الجهد الحالية المستخدمة من قبل شركات تطوير البرمجيات الفلسطينية البارزة وتحليل ممارساتها، و (2) لاقتراح أسلوب تقدير الجهد المناسب الذي يمكن أن يتناسب مع طبيعة واحتياجات هذه الشركات وللتحقق من صحة هذه التقنية عبر تطبيق حقيقي ضمن مشاريع برمجية فعلية في مجموعة فرعية مختارة من هذه الشركات.

بناءً على المسح والتحليل الذي أجريناه، اخترنا تقنية تقدير الجهد الحالية تسمى (Triangulation) كأسلوب تقدير الجهد والتكلفة الأنسب للشركات الصغيرة وصناعة تطوير البرمجيات الفلسطينية، ثم قمنا بتصميم امتداد لهذه التقنية من أجل (صقل) للاحتياجات الدقيقة للشركة وتزويدها ببعض المرونة لإجراء بعض التغييرات والتعديلات (على سبيل المثال، تقليل وقت تسليم المشروع عن طريق زيادة الموارد). علاوة على ذلك، تم تطبيق هذه التقنية على مشاريع البرمجيات الفلسطينية للتحقق من نتائجها.

نعتقد أن هذه الدراسة يمكن أن تكون موردا قيما لشركات تطوير البرمجيات الفلسطينية. ويمكنهم استخدامه كدليل إرشادي لمساعدتهم في الحصول على تقديرات أفضل وأكثر دقة للجهود، والتي بدورها يمكن أن تقلل التكاليف وتوفر جدولة أفضل وأكثر دقة واحتياجات التوظيف.