ENHANCING DELPHI METHOD WITH ALGORITHMIC ESTIMATES FOR SOFTWARE EFFORT ESTIMATION: AN EXPERIMENTAL STUDY

Tharwon Arnuphaptrairong

Department of Statistics, Chulalongkorn University, Bangkok, Thailand

ABSTRACT

Literature review shows that more accurate software effort and cost estimation methods are needed for software project management success. Expert judgment and algorithmic model estimation are two predominant methods discussed in the literature. Both are reported almost at the comparable level of accuracy performance. The combination of the two methods is suggested to increase the estimation accuracy. Delphi method is an encouraging structured expert judgment method for software effort group estimation but surprisingly little was reported in the literature. The objective of this study is to test if the Delphi estimates will be more accurate if the participants in the Delphi process are exposed to the algorithmic estimates. A Delphi experiment where the participants in the Delphi process were exposed to three algorithmic estimates –Function Points, COCOMO estimates, and Use Case Points, was therefore conducted. The findings show that the Delphi estimates are slightly more accurate than the statistical combination of individual expert estimates, but they are not statistically significant. However, the Delphi estimates are statistically significant more accurate than the individual estimates. The results also show that the Delphi estimates are slightly less optimistic than the statistical combination of individual expert estimates but they are not statistically significant either. The adapted Delphi experiment shows a promising technique for improving the software cost estimation accuracy.

KEYWORDS

Software Effort Estimation, Delphi Method, Algorithmic Estimates, Experimental Study.

1. INTRODUCTION

For software project management success, more accurate software effort and cost estimation methods have long been searching for. Expert judgment technique is the most preferred method found in the software industry [1-8]. The findings from the literature surveys show that despite a great deal of research for better algorithmic models for software effort estimation, the expert judgment method remains the dominant approach. Consequently, expert judgment has been encouraged for the future research direction for a better estimation method [9]. Delphi and Group process expert judgment techniques for software estimation can be found in the literature [10, 11-13]. Moløkken-Østvold et al., [10] presented the findings of an experiment that investigated the effects of planning poker --an expert judgment combination technique, in software cost estimation. Passing and Shepperd [11] evaluated the effect of checklists and the effect of group discussions on a software project size and effort estimation. Moløkken-Østvold and Jørgensen [12] did a controlled experiment on estimation groups in a web-development company. Gandomani et al., [13] reported the software costs estimation accuracy using the Wideband Delphi method for one company and planning poker for another company, both on the historical data of the completed software.

DOI: 10.5121/ijsea.2021.12401

Surprisingly Delphi method, an encouraging structured group method, was not found reported for software effort group estimation and little was reported in the literature [10]. Besides expert judgment, algorithmic model estimation is another predominant method discussed in the literature. Jørgensen [14] did an extensive review of 15 accuracy comparison studies between expert judgment and formal models of software development effort. The results indicate that there is no substantial evidence in preference the use of formal estimation models. The combination of the two methods is suggested in the literature to increase the estimation accuracy. [9, 15-18].

The objective of this study is, therefore, to empirically investigate the performance of the Delphi method where the participants in the Delphi experiment will be exposed to three algorithmic estimates –Function Points, COCOMO estimates, and Use Case Points, for software project estimation as suggested.

This report is organized as follows. Section II gives the literature review of the expert judgment technique and the 3 algorithmic estimation models proposed. Section III discusses the research questions. Section IV describes the research method and the findings are given in section V. Section VI discusses and concludes the research.

2. Related Work

This section presents the literature related to the expert judgment technique and Delphi method, and the 3 algorithmic estimation models proposed in the experiment –Function Points, COCOMO estimates, and Use Case Points.

2.1. Expert Judgement Technique

This section gives the review on expert judgement technique, Delphi technique, group process and works related to Delphi study, and group process in software estimation.

2.1.1. Definition of Expert Judgment Technique

The expert judgment technique is generally applied in forecasting domains. In the “software estimation” context, several definitions can be found [14,19-22]. In all, expert judgment can generally be seen as the solicitation of estimates from the experts in the domains. The judgment strategy may vary from pure intuition to judgment with the aid of guidelines, work breakdown checklists or historical data.

The issue regarding the expert judgment method is the term "expert". To be successful with the expert judgment method, the careful expert selection is needed. The expert needs to have experience in the application area including experience with estimation, management, and practical work [16, 20]. The good points of the expert judgment method are: take less time to generate; need little time and cost, and can be as accurate as other methods that are expensive [21]. Expert judgment estimation is also suitable when pure model-based estimation methods are not practical or possible because the needed historical or technical data are not available or when estimating complex, ill-defined or poorly understood problems [23]. To be successful, Rush and Roy [21] asserted that the expert requires many years of experience. The main weakness is concerned with the subjective nature, inconsistency and uncertainty, and prone to the bias of the outcomes.
‘

2.1.2. Combining Experts' Judgment, statistical groups, and Delphi method

Various techniques can be used to combine expert judgment estimates, for example, statistical groups, unstructured groups (unstructured face-to-face meetings or FTF), Delphi, Wideband Delphi, planning poker, decision markets/ prediction markets [10], and Nominal Groups Technique (NGT or estimate-talk-estimate) [24].

In statistical groups, individual estimates are just combined statistically. The mean or median of the different individual estimates will usually be calculated as a combined estimate. A simple average of the estimates often works best for combining estimates [10, 16].

2.1.3. Delphi Method

Delphi technique is perhaps the most formal and rigorous method for capturing expert opinions [21]. There are serval reviews on the Delphi method and the history of Delphi, including the extensive work of [25] and [26]. Three conclusions were drawn [25]: 1) Statistical group estimate of several individual judgments is more accurate than individual judgments. 2) Unstructured groups (unstructured direct interaction) are more accurate than statistical groups. 3) Weak points of the unstructured groups (unstructured direct interaction) may induce suboptimal accuracy of judgments. Based on these three pieces of evidence, researchers were trying to develop other judgment techniques that hold the advantages but eliminate the disadvantages of the unstructured groups discussed above. Many techniques have been proposed. One of the most well-known methods is the Delphi method. Delphi method is a structured and indirect interaction (no direct interaction allowed).

"Delphi" was the name invented by Kaplan, when he was working for the Rand Corporation. Kaplan et al., [27] reported that unstructured groups (unstructured direct interaction) was not more accurate than statistical groups. To anticipate the problem, Gordon, Helmer, and Dalkey developed the Delphi method in the 1950s, when they were also working at the Rand Corporation. Not until 1964, Gordon and Helmer had published their Delphi work [25, 28].

Several definitions of Delphi techniques can be found [10, 24, 26, 29]. In essence, the Delphi technique is a group process used to reach the group consensus estimates. The Delphi method aims to avoid biases and drawbacks of the unstructured groups or unstructured face-to-face meetings, i.e. group pressures or group politics by using rounds of anonymous polling. In each round of polling for estimates, the other experts' opinions on the estimates of the previous round were summarized and reported as feedback. Delphi method is characterized by 3 main features: anonymity, iteration, and feedback [25]. The statistical aggregation of group response was added as the fourth feature [24]. Anonymity portrays whether the owner of the estimates is anonymous to each other. Questionnaires and other computer-mediated media may be used to achieve anonymity. The experts can convey their opinions freely without social pressures. Iteration portrays whether the estimators interact with each other. This part is the polling of the experts' opinions in several rounds. The number of rounds is usually fixed or depending on a criterion of consensus or the stability in individual judgments. It is usually set up in advance prior to the process. From round to round, the participants can refine their estimates in light of the development of the group work. Feedback and statistical aggregation of group responses portray whether the information about the estimates or participant's perspectives of the whole group is provided to all participants/experts. After each round of polling, the statistical format of the estimates of the whole group on the previous round is fed back to all participants/experts. Usually, it is presented in a simple statistical summary of the group estimates --a mean or median of the whole group on the previous round. It provides the opportunity for participants/experts to clarify or modify their views on the next round.

At the end of the last round, individual estimates are combined statistically (mean/median) and taken as group consensus estimate.

In sum, the experts/participants do not meet and discuss in person. To avoid direct interaction between group members, the written interaction is utilized instead. The aim is to avoid biases and drawbacks of the unstructured groups or unstructured face-to-face meetings, i.e. group pressures or group politics by using rounds of anonymous polling.

2.1.4. Works Related to Delphi Study and Group Process in Software Estimation

This section presents the studies related to the Delphi experimental study in software estimation found in the literature. Since the interest of this study is in experimental evaluations of Delphi in software estimation, the search used the words "Delphi, expert judgment, software estimation, experiment, and accuracy". Four related studies were found [10-13]. None of them is a Delphi study in software estimation. Nevertheless, the work of Graefe and Armstrong [24] is also added in the literature review since it is one of the latest Delphi studies found, even though it is not directly related to the software estimation.

Moløkken-Østvold et al., [10] experimented to explore the effects of the planning poker technique. 4 questions of which 3 main related research questions are: "Q1: Are group consensus estimates less optimistic than the statistical combination of individual expert estimates? Q2: Are group consensus estimates more accurate than the statistical combination of individual expert estimates? Q3: Are group consensus estimates more accurate than the existing individual estimation method?" The experiment was performed with 15 to 20 tasks of a software project using the Scrum development method of a medium-size Norwegian software company. The experts who participated in the process were the project development team members. The finding shows that the first question was supported but the second and the third were rejected. However, group estimates showed slightly more accuracy than a statistical combination and similar accuracy for group estimates and the existing individual estimation method.

Passing and Shepperd [11] evaluated the effect of checklists and the effect of group discussions on a project size and effort estimation. The study was an experiment of thirteen masters of software engineering degree students --11 males and two females. The students formed three groups of three and one group of four students. The experiment was organized to explore how checklists and group discussions affect their estimates. The investigation was based on three rounds of estimation. For the first round, the participants used any estimation effort method they wished to estimate the size and effort for their project. The students were then, asked to reassess their first round's estimates. They were again asked to estimate the size and the effort estimates in the form of a checklist. After a break, group discussion and the Delphi technique were introduced. Each group then discussed in separate rooms. After the discussion, the students were asked to estimate the software size and effort for the third and final estimation round. It is found that checklists increased software estimates in size and effort, but group discussions did not add any effects. Both checklists and group discussions improved the project size estimation accuracy but the data is inadequate for the effort estimation analysis. The reason for the better results is that checklists and discussions lead to the acquisition of new knowledge and different views.

Moløkken-Østvold and Jørgensen [12] did a controlled experiment in a web-development company on estimation groups. Two hypotheses were raised. "H1: Group effort estimates are, on average, less optimistic than the average of the experts' individual effort estimates. H2: Individual effort estimates are, on average, less optimistic after a group discussion with other experts than before group discussion". Five participants were selected from each of the four company roles

(Engagement Manager/Sales and Client Responsibility, Project Manager, User Analyst/Designer, and Technical programmer. The required task for the participants was to estimate the effort needed to complete a software project. In the experiment, first, the participants were asked to individually estimate the software effort needed. Then, the participants were divided into groups. There were not participants in each group that had the same company role. The findings supported both hypotheses. It is found that the group estimate was less optimistic than the average expert estimate in four out of five groups because forgotten tasks were identified by the group process. It indicates that the group estimates are more accurate than a statistical group method. The authors also believe that the quality of the group estimates improved because the participants were from different roles.

Gandomani et al., [13] reported two case studies (companies). The software costs were estimated using the Wideband Delphi method for one company and planning poker for another. Both Wideband Delphi and planning poker techniques were applied to the history data of the completed software. The results show that both methods increased cost estimation accuracy. However, planning poker reveals more accuracy than the Wideband Delphi method.

Graefe and Armstrong [24] performed several laboratory experiments to evaluate the accuracy of unstructured face-to-face meetings (FTF) with the three other structured methods --nominal groups, Delphi, and prediction markets. Participants were 227 students at the University of Pennsylvania. The recruited students were randomly assigned to 44 heterogeneous groups – 11 per method. They were to solve ten factual questions, which required percentage estimates. The results did not show any statistically significant in accuracy differences of the four methods. Delphi was reported as accurate as face-to-face meetings (FTF) for eight questions and outperformed face-to-face meetings (FTF) for the rest two questions. Delphi was also more accurate than statistical groups. The findings confirm the prior research reviewed. There were no accuracy differences between participants' priors in Nominal Group Techniques (NGT), Delphi method, and prediction markets. Delphi method was most accurate because it had a lower Mean Absolute Error (MAE) than the individual priors for the ten questions. Delphi method was significantly more accurate than the statistical groups for three out of the ten questions.

2.2. The Proposed Algorithmic Estimates –Function Points estimates, COCOMO estimates, and Use Case Points estimates

Function Points estimates, COCOMO estimates, and Use Case Points estimates are the three selected algorithmic estimates to include in the Delphi process. They are well accepted non-proprietary techniques in the literature [30]. The following section provides a short review on these techniques.

2.2.1. Function Points

Albrecht originated Function Points (FP) method in 1979 [31-32]. The Function Points method is widely acknowledged with a lot of variations. The main concept of the Function Points technique is that the more the functions of a software, the more the Line of Codes. Function Points presumes five unique types of function -- External Output, External Input, Internal Logical File, External Interface File, and External Query, as shown in Figure 1.

Each of these five function types is categorized individually as low, average, or high complexity with the weight varies from 3 (for low complex external inputs) to 15 (for high complex internal files) as shown in Table 1.

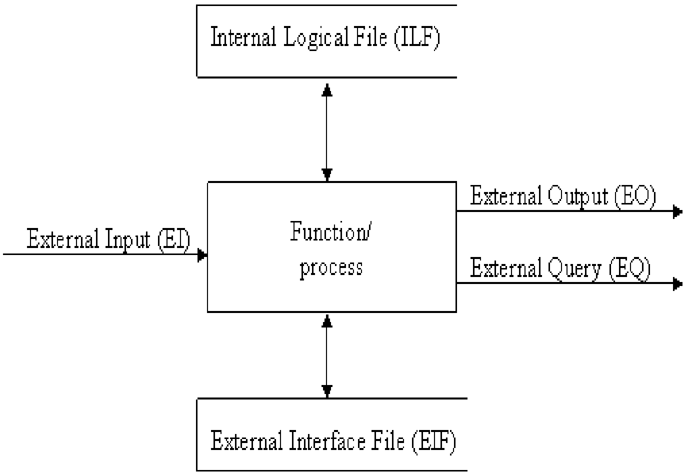

Figure 1. The five unique function types.

Table 1. The Functions Points Weights

	Complexity		
Function Type	**Low**	**Average**	**High**
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6
Internal Logical File	7	10	15
External Interface File	5	7	10

The software size is measured in terms of Function Points. The Unadjusted Function Points (UFP) is computed as follows:

$$\text{UFP} = \sum_{i=1}^{5}\sum_{j=1}^{3} N_{ij}W_{ij}$$

Where Nij is the counts of each function type i of the five types and Wij is the corresponding complexity weights value j of the 3 levels –low, average, and high.

The calculated Function Points may need to be adjusted with the technical complexity factors (TCF) which can be obtained by the formula:

$$\text{TCF} = 0.65 + (\text{sum of factors}) / 100$$

Each of the 14 technical factors is rated based on its level of effect from no influence (0) to very influential (5). The adjusted Function Points (FP) is then calculated as follows:

$$\text{FP} = \text{UFP x TCF}$$

At present, the standards for the Function Point Size Measurement is maintained by the International Function Point User Group (IFPUG) to ensure that function points counting is the same and comparable. The standard manual can be obtained at http://www.ifpug.otg.

2.2.2. COCOMO Model

The Constructive Cost Model (COCOMO) [33] was developed by Barry W. Boehm in the late 1970s. The model was named COCOMO I or COCOMO 81. The first level, Basic COCOMO is as follows:

$$\textbf{Effort} = a \times (\textbf{Size})^{b}$$

Where **Effort** is the estimated effort in person-months.

Size is the estimated size of the software to be developed in Kilo Delivered Source Instructions (KDSI).

a is a constant.
b is another constant.

Later, COCOMO II was updated in 1995 and published in 2000 [34]. The model for Nominal Schedule (NS) is as follows:

$$\mathbf{PM_{NS} = A \times (Size)^{E} \times \Pi\, EM_i}$$

where

PM_{NS} is the estimated effort in person months.
A is a constant.

Size is the estimated size of the in software to be developed in Kilo Source Line of Code (KSLOC).

E is the Scale factors (calculated from 5 factors).
EM_i is the Effort Multipliers or Cost drivers (altogether 17 Multipliers).

2.2.3. Use Case Points

Use Case Points (UCP) method [35-36] was introduced in Karner's 1993 M.Sc. thesis. UCP technique uses the information available from use case diagrams to estimate the software size. UCP employs 2 drivers –actors and use cases. The UCP steps are as follows:

1. Actors are categorized as simple, average or complex with the weight of 1, 2, and 3 respectively. The total Unadjusted Actor Weight (UAW) is computed as:

$$UAW = \sum \text{actor } i \ \times \ \text{weight } i$$

2. Use cases are categorized into ≤ 3, 4-7, and > 7 transactions with the weight of 5, 10, and 15 respectively. The Unadjusted Use Case Weights (UUCW) is calculated as:

$$UUCW = \sum UC\ i \ \times \ \text{weight } i$$

The Unadjusted Use Case Points (UUPC) is computed as:

$$UUCP = UAW + UUCW$$

3. The Technical Complexity Factor (TCF) and Environmental Factor (EF) are computed from the following equations:

$$TCF = 0.6 + (0.1 * TFactor)$$
$$EF = 1.4 + (-0.03 * EFactor)$$

Where each of the 13 TFactor and 8 EFactor are rated based on its level of effect from 0 to 5 (0 for no influence to 5 for very influential) and then multiply by its weight.

4. Finally, the Use Case Points (UCP) is then computed as follows:

$$UCP = UUCP * TCP * EF$$

3. RESEARCH QUESTIONS

For the last 25 years, Delphi or Wideband Delphi method has not been exposed to extensive empirical research in a software engineering environment [12]. Kitchenham et al., [37] – a study of a company that employed the Delphi approach, was the only one. It is also reported that there is no empirical study on the accuracy of Delphi and Wideband Delphi in a software engineering context either [10]. The search for later evidence on empirical studies on Delphi and Wideband Delphi method also does not found reported except the study of Gandomani et al., [13].

The objective of this study is, therefore, to empirically investigate the performance of the Delphi method where the participants in the Delphi experiment will be exposed to three algorithmic estimates –Function Points, COCOMO estimates, and Use Case Points, for software project estimation as recommended in the literature [9, 15-18].

First, the research aims to further explore whether optimism could be reduced using the adapted Delphi technique, as found experimented on group estimation of both [10] and [12]. The study of group process such as unstructured groups [12] and planning poker [10], defended that experts share different project experiences and knowledge easily through face-to-face interaction. However, on the other hand, it can be argued that the Delphi technique, even though without face-to-face interaction but still share the feedback and statistical aggregation of group response, may motivate the participants more committed to their decisions. The following research question was then, raised:

RQ1: Are group consensus estimates using the Delphi method less optimistic than the statistical combination of individual expert estimates?

The hypothesis that "Group effort estimates are, on average, less optimistic than the average of the experts' individual effort estimates" was found supported in [10]. However, the opposite was found in [12]. For this research question, the following hypothesis is, therefore, raised:

H1: Group consensus estimates using the Delphi method are less optimistic than the statistical combination of individual expert estimates.

An estimate is more optimistic than another if and only if it takes a lesser amount of time to finish a task than the other estimate [10]. For example, an estimate of 5 hours is more optimistic than an estimate of 6 hours. Moløkken-Østvold et al. [10] explained this observation of reduced optimism as the process of a choice shift. Zuber et al., [38] defined choice shift as the difference between the arithmetic mean of individual estimates and the group consensus estimate. The choice shift is generally described as increased risk willingness and optimism [39] or risky shift. In this case of reduced optimism is the opposite direction of choice shift.

However, when optimism is found, it does not guarantee that Delphi estimates would be more accurate. Thus, this study also intends to investigate whether Delphi estimates are more accurate than statistical groups. Therefore, the following research question is proposed:

RQ2: Are group consensus estimates using the Delphi method more accurate than the statistical combination of individual expert estimates?

Woudenberg [25] reviewed 102 Delphi application papers published. He concluded that "there is no evidence to support that the Delphi is more accurate than other judgment methods". However, it is reported in [14] that a group-based process led to the highest accuracy in the area of human judgment and forecasting. Moløkken-Østvold et al., [10] have not found the support for the hypothesis --"Group estimates were more accurate than statistical groups". Group estimates showed slightly more accurate than statistical groups. According to Graefe and Armstrong [24], the Delphi method was found significantly more accurate than the statistical groups for three out of the ten questions. From the mixed findings, the following hypothesis is therefore proposed:

H2: Group consensus estimates using the Delphi method are more accurate than the statistical combination of individual expert estimates.

There are three related studies in comparing the accuracy of group estimates with individual estimates for software effort estimation [10-12]. Passing and Shepperd [11] evaluated the effect of checklists and the effect of group discussions on a project size and effort estimation. Moløkken-Østvold and Jørgensen [12], did a study of the unstructured group process in software effort estimation while Moløkken-Østvold et al., [10] did an empirical study of planning poker application in software cost estimation. Similarly, this study aims to compare the estimates that were derived using the Delphi method to a set of estimates that were derived by individual experts that are subjected to a subsequent group Delphi method. The following research question is then, raised:

RQ3: Are group consensus estimates using the Delphi method more accurate than the individual estimation method?

Similar to the second hypothesis, Moløkken-Østvold et al., [10] have not found the support for the hypothesis -- "Group estimates were more accurate than the individual estimation method". However, group estimates showed slightly more accuracy than the existing individual estimation method. According to Graefe and Armstrong [24], the Delphi method showed a lower Mean Absolute Error (MAE) than the individual priors for the ten questions. The following hypothesis is therefore proposed:

H3: Group consensus estimates using the Delphi method are more accurate than the individual estimation method.

4. RESEARCH METHODOLOGY

To answer the proposed research questions, a modified Delphi experiment was conducted to test the hypothesis. The Delphi method was adapted to incorporate the exposition to the experts of the three estimates from algorithmic models –Function Points, COCOMO, and Use Case Points. To investigate the effects and validate the proposition, at least 5 subjects (experienced estimators) were planned to be recruited to participate in the Delphi process to estimate the efforts needed for the completed software projects provided. The details of the experimental setting are given in the following sessions.

4.1. Subjects / The Experts

The subject/expert in this study is defined as a software project manager, developer or system analyst with experience related to software effort estimation. Solicitation for participants in the experiment was made through software companies and on Facebook. The main criteria are that the participants had a minimum of ten years of experience and were familiar with software development and cost-estimation responsibility.

In total, there were 5 participants -- 4 males and 1 female, agreed to participate in the experiment. This is a sufficient number of experts as indicated in [14]. Table 2 presents the profile data for each expert. Their age ranges from 33 to 42 with an average age of 38.4 years. Their experience in software development ranges from 10 to 17 years with an average of 13.40 years of experience. Three of them are project managers. One is an assistant manager and another one is a system analyst. Each subject's experience reflected a different language and development environment.

Table 2. The Subject /Expert Profile

Expert	**Sex**	**Age**	**Year of Exp.**	**Present Position**	**Language skill**
1	M	39	17	Project manager	C#, Java, Java script
2	M	38	13	Assistant manager	Vb.net, C#, VBA
3	F	42	11	Project manager	C#, Java, Java script
4	M	40	16	Project manager	PHP, Java, Java script
5	M	33	10	System analyst	C#, Java, Java script

4.2. Material--the software projects

Table 3. The Software Project Profile

No.	**Project**	**Line of codes**	**Type Of Application**	**No. of Functional req't**	**No. of Non-functional req't**	**Actual Effort (man-months)**
1	HHTP	8,371	Cross-platform mobile application	13	3	2.5
2	PTT	11,131	Android Application	8	0	7.5
3	KAPP	9,258	Cross-platform mobile application	11	1	8.5
4	EATD	6,532	Android Application	10	3	5
5	ST	35,974	Web application	29	0	14
6	CINOUT	1,596	Cross-platform mobile application	7	2	3
7	MINSH	10,115	Web application	11	2	3
8	LMS	4,210	Cross-platform mobile application	14	2	3
9	LEAP	3,409	Web Application	10	2	5.5
10	TM	2,440	Web Application	12	4	4

The literature reviewed shows that the number of software projects, for this type of experiment, can vary from 1 to 10 [14]. Therefore, it is planned to look for at least 10 software projects. The Solicitation for projects was made through both software companies and software freelancers. It was concluded with 10 software projects. The profile of the 10 projects is presented in Table 3. In total, there are 4 android applications, 2 web applications, and 4 cross-platform applications. The size of the applications ranges from 1,596 to 39,574 lines of code. The number of functional requirements ranges from 7 to 29 requirements while the non-functional requirements is from 0 to 4 requirements. The actual effort varies from 2.5 to 14 man-months.

4.3. Procedure –the Delphi steps

The research method consists of the following Delphi steps.

1) In the experiment, the moderator explained, in a meeting, to the experts the objectives of the experiment.
2) For each project, the moderator then described the software projects to be estimated. The topics included were the objectives of the project, the project scope, functional requirements and non-functional requirements, and related issues.
3) The experts generated the estimates individually for the software in the questionnaire provided.
4) The experts were then, presented with three estimates from the algorithmic model i.e. Function Points, COCOMO model, and Use Case points, for the experts to compare with their original estimate, and then, the experts reviewed and re-estimated if they required to.
5) The moderator then collected the estimates and their opinions from the experts.
6) The collected estimates were then, analysed in the statistical format i.e. mean, mode, maximum and minimum values, and fed back to all expert participants.
7) With the feedbacks obtained, the experts reviewed their estimates and revised their original estimates if necessary.
8) Steps 5, 6 and 7 were repeated until a consensus or the predefined criterion was met for all 10 software projects. In this experiment, the criterion is that the experiment will stop when the changes of the Delphi estimate from the previous round are less than 10%.

To encourage participants to put serious intention and effort into accurate estimates, there was a reward of 10,000, 5,000, and 2,500 Thai baht, for the most, second, and third accurate estimates.

4.4. The Measurement of Accuracy

Two measures --Magnitude Relative Error (MRE) and Balanced Relative Error (BRE), were used for measuring estimation accuracy. Magnitude Relative Error (MRE) is defined as follows [40]:

$$\text{MRE} = \left| \frac{MMest - MMact}{MMact} \right|$$

Where **MMact** is the actual software effort, and **MMest** is the estimated software effort.

MRE is extensively used for measuring accuracy. However, Moløkken-Østvold et al., [10] adopted Balanced Relative Error (BRE) for the reason that it is a more balanced measure than MRE, referencing the study of Miyazaki et al., [41]. Balanced Relative Error (BRE) is defined as follows:

$$\text{BRE} = \frac{|\,MMact - MMest\,|}{Min\ (MMact,\ MMest)}$$

Where **MMact** is the actual software effort and **MMest** is the estimated software effort.

In addition to the Magnitude of Relative Error (MRE) and Balanced Relative Error (BRE), the measure of prediction level or PRED (p) is also employed to incorporate with the accuracy performance analysis. PRED (p) is defined as follows [42-43]:

$$\text{PRED}(p) = k/n$$

Where **n** is the total number of estimates, and **k** is the number of estimates that have an accuracy less than or equal to the value **p**.

For example, PRED (0.25) = 0.50 means that half of the estimates have accuracy within 0.25 or 25%. The level of accuracy usually accepted is PRED (0.25) = 0.75 or meaning it should be within 25% accuracy for 75 % of the estimates [42, 43].

5. FINDINGS

Three research questions are posed in this study. The first question compares estimates using the Delphi method with the statistical combination of individual expert estimates. The second question compares the accuracy of the Delphi method with the statistical combination of individual expert estimates. The last question compares the accuracy of Delphi estimates with the individual estimates. The findings of the three research questions are presented as follows:

RQ1: Are group consensus estimates using the Delphi method less optimistic than the statistical combination of individual expert estimates? The hypothesis for this research question:

H1: Group consensus estimates using the Delphi method are less optimistic than the statistical combination of individual expert estimates.

Table 4 shows the results of the statistical group estimates (X1) and the Delphi estimates (X3) of the 10 software projects from the Delphi experiment. The average of the statistical group estimates (X1) is 2.924 man-months and of the Delphi estimates (X3) is 3.030 man-months. 5 projects (4, 5, 6, 8, and 9) that shows Delphi estimates are less optimistic than the statistical group estimates while only 3 projects (2,3,7) shows that Delphi estimates are more optimistic than the statistical group estimates.

Shapiro-Wilk test for normality gave a p-value of 0.295, and 0.016 for the statistical group estimates (X1) and the Delphi estimates (X3) respectively. The results indicate that X1 is normally distributed while X3 is not. The Wilcoxon Matched-Pairs Signed-Ranks Test for hypothesis 1 is thus, employed. The test gave a p-value of 0.311. Hypothesis 1 is, therefore, rejected.

Table 4. The statistical group estimates (X1) and the Delphi estimates (X3) of the 10 software projects from the Delphi experiment.

Project	**Statistical group estimates (X1)**	**Delphi estimates (X3)**	**X3 : X1**
1.HHTP	2.3	2.3	=
2.PTT	3.7	3.4	<
3.KAPP	3.9	3.7	<
4.EATD	3.0	3.1	>
5.ST	6.0	7.1	>
6.CINOUT	1.34	1.5	>
7.MINSH	2.6	2.5	<
8.LMS	3.1	3.3	>
9.LEAP	1.6	1.7	>

10.TM	1.7	1.7	=
Mean	2.924	3.030	

RQ2: Are group consensus estimates using the Delphi method more accurate than the statistical combination of individual expert estimates? The hypothesis for this research question:

H2: Group consensus estimates using the Delphi method are more accurate than the statistical combination of individual expert estimates.

Table 5a and Table 5b show the accuracy of the estimates (MRE and BRE) of both statistical group estimates (X1) and the Delphi estimates (X3) of the 10 software projects from the Delphi experiment. The average MRE of the statistical group estimates (X1) is 0.4103 (41.03%) and of the Delphi estimates (X3) is 0.4097 (40.97%). The average BRE of the statistical group estimates (X1) is 0.9510 (95.10%) and of the Delphi estimates (X3) is 0.9063 (90.63%). 4 projects (4, 5, 6, and 9) that shows Delphi estimates are more accurate than the statistical group estimates and 4 projects (2, 3, 7, and 8) that shows Delphi estimates are less accurate than the statistical group estimates in term of both MRE and BRE.

Shapiro-Wilk test for normality gave a p-value of 0.065 and 0.117 for the MRE of statistical group estimates (X1) and the Delphi estimates (X3) respectively. The results indicate that both the MRE of X1 and X3 are normally distributed. The one-sided Paired-Sample T-test for hypothesis 2 is thus, employed. The test gave a p-value of 0.4815. Hypothesis 2 is, therefore, rejected for MRE.
Shapiro-Wilk test for normality gave a p-value of 0.280 and 0.399 for the BRE of statistical group estimates (X1) and the Delphi estimates (X3) respectively. The results also indicate that both BRE of X1 and X3 are normally distributed. The one-sided Paired-Sample T-test for hypothesis 2 is thus, employed. The test gave a p-value of 0.2155. Hypothesis 2 is, therefore, also rejected for BRE.

Table 5a. The accuracy of the estimates (MRE and BRE) of both statistical group estimates (X1) and the Delphi estimates (X3) of the 10 software projects.

Project	MRE X1	%MRE X1	MRE X3	%MRE X3
1.HHTP	0.0800	8.00	0.0800	8.00
2.PTT	0.5067	50.67	0.5467	54.67
3.KAPP	0.5412	54.12	0.5647	56.47
4.EATD	0.4000	40.00	0.3800	38.00
5.ST	0.5714	57.14	0.4929	49.29
6.CINOUT	0.5533	55.33	0.5000	50.00
7.MINSH	0.1333	13.33	0.1667	16.67
8.LMS	0.0333	3.33	0.1000	10.00
9.LEAP	0.7091	70.91	0.6909	69.09
10.TM	0.5750	57.50	0.5750	57.50
Mean	0.4103	41.03	0.4097	40.97

Table 5b. The accuracy of the estimates (MRE and BRE) of both statistical group estimates (X1) and the Delphi estimates (X3) of the 10 software projects

Project	BRE X1	%BRE X1	BRE X3	%BRE X3
1.HHTP	0.0870	8.70	0.0870	8.70
2.PTT	1.0270	102.70	1.2059	120.59
3.KAPP	1.1795	117.95	1.2973	129.73

4.EATD	0.6667	66.67	0.6129	61.29
5.ST	1.3333	133.33	0.9718	97.18
6.CINOUT	1.2388	123.88	1.0000	100.00
7.MINSH	0.1538	15.38	0.2000	20.00
8.LMS	0.0333	3.33	0.1000	10.00
9.LEAP	2.4375	243.75	2.2353	223.53
10.TM	1.3529	135.29	1.3529	135.29
Mean	0.9510	95.10	0.9063	90.63

RQ 3: Are group consensus estimates using the Delphi method more accurate than the individual estimation method? The hypothesis for this research question:

H3: Group consensus estimates using the Delphi method are more accurate than the individual estimation method.

Table 6. The mean MRE and BRE of the Delphi estimates and of the individual estimates

N	**50**	**N**	**50**
Mean of MRE of Delphi estimates	0.4097	Mean of BRE of Delphi estimates	0.9063
Mean of MRE of individual estimates	0.4863	Mean of BRE of individual estimates	1.2875
Differences	0.0766	Differences	0.3812
p-Value	0.007	p-Value	0.003

Table 6 shows that the average MRE of the Delphi estimates is 0.4097 (40.97%) and of the individual estimates is 0.4863 (48.63%). The average BRE of the Delphi estimates is 0.9063 (90.63%) and of the individual estimates is 1.2875 (128.75%).

Shapiro-Wilk test for normality gave a p-value of 0.000 and 0.085 for the MRE of Delphi estimates and the individual estimates respectively. The results indicate that the MRE of individual estimates is normally distributed while of the Delphi estimates is not. The Wilcoxon Matched-Pairs Signed-Ranks Test for hypothesis 3 is thus, employed. The test gave a p-value of 0.007. Hypothesis 3 is, therefore, accepted for MRE.

Shapiro-Wilk test for normality gave a p-value of 0.399 and 0.280 for the BRE of Delphi estimates and the individual estimates respectively. The results indicate that BRE both of the Delphi estimates and the individual estimates are normally distributed. The one-sided Paired-Sample T-test for hypothesis 3 is thus, employed. The test gave a p-value of 0.003. Hypothesis 3 is, therefore, accepted for BRE as well.

The test is also performed project-wise. Table 7 shows the mean differences of MRE and BRE between Delphi and individuals estimates for the 10 projects.

Although, the negative mean differences in table 7 indicate that Delphi estimates are more accurate but statistical test in table 8a table 8b shows that only 2 out of ten projects for MRE (project 7 and 8) and 3 out of ten for BRE (project 1,7 and 8) are statistically supported for hypothesis 3.

Table 7. The mean differences of MRE and BRE between Delphi and individuals estimates for the 10 software projects.

Project	The mean differences of MRE between Delphi and individuals estimates	The mean differences of **MRE Delphi** < individuals estimates	The mean differences of BRE between Delphi and individuals estimates	The mean differences of **BRE Delphi** < individuals estimates
1.HHTP	-0.16	<	-0.34	<
2.PTT	0.04		-0.31	<
3.KAPP	0.02		-0.17	<
4.EATD	-0.02	<	-0.10	<
5.ST	-0.08	<	-0.67	<
6.CINOUT	-0.05	<	-0.66	<
7.MINSH	-0.23	<	-0.43	<
8.LMS	-0.27	<	-0.34	<
9.LEAP	-0.16	<	-0.69	<
10.TM	-0.16	<	-0.10	<

Table 8a. The test results by projects: the p-value of Shapiro-Wilk, the Statistic used, and the p-value for the one-tailed test for MRE.

Project	P-value of Shapiro-Wilk	Statistic used	P-value for one- tailed	MRE Mean X3< MRE X1
1.HHTP	0.135	t-test	0.089	×
2.PTT	0.994	t-test	0.355	×
3.KAPP	0.332	t-test	0.394	×
4.EATD	0.146	t-test	0.367	×
5.ST	0.207	t-test	0.195	×
6.CINOUT	0.094	t-test	0.296	×
7.MINSH	0.046	Wilcoxon	0.0295	/
8.LMS	0.054	t-test	0.0475	/
9.LEAP	0.421	t-test	0.3745	×
10.TM	0.000	Wilcoxon	0.240	×

Table 8b. The test results by projects: the p-value of Shapiro-Wilk, the Statistic used, and the p-value for one-tailed test for BRE.

	P-value of Shapiro-Wilk	Statistic used	P-value for one- tailed	BRE Mean X3< BRE X1
1.HHTP	0.008	Wil-coxon	0.039	/
2.PTT	0.166	t-test	0.333	×
3.KAPP	0.267	t-test	0.350	×
4.EATD	0.284	t-test	0.248	×
5.ST	0.383	t-test	0.092	×
6.CINOUT	0.311	t-test	0.141	×
7.MINSH	0.272	t-test	0.028	/
8.LMS	0.428	t-test	0.025	/
9.LEAP	0.266	t-test	0.186	×
10.TM	0.000	Wil-coxon	0.240	×

6. Conclusion and Discussions

In summary, the findings for the three research questions posed in this research are as follows:

The first research question: Are group consensus estimates using the Delphi method less optimistic than the statistical combination of individual expert estimates?

The average of the statistical group estimates of the 10 software projects is 2.924 man-months and of the Delphi estimates is 3.030 man-months. This indicates a slight average decrease in optimism. However, the findings are not statistically significant. 5 projects (4, 5, 6, 8, and 9) show that the Delphi method is less optimistic than the statistical groups while only 3 projects (2,3,7) show that Delphi estimates are more optimistic than the statistical group estimates.

For the second research question: Are group consensus estimates using the Delphi method more accurate than the statistical combination of individual expert estimates?

The average MRE of the statistical group estimates is 0.4103 (41.03%) and of the Delphi estimates is 0.4097 (40.97%). The average BRE of the statistical group estimates is 0.9510 (95.10%) and of the Delphi estimates is 0.9063 (90.63%). It shows that the Delphi estimates are slightly more accurate than the statistical groups both in terms of MRE and BRE. Nevertheless, the numbers are not statistically significant either. 4 projects (4, 5, 6, and 9) show that Delphi estimates are more accurate than the statistical group estimates and 4 projects (2, 3, 7, and 8) show that Delphi estimates are less accurate than the statistical group estimates in term of both MRE and BRE.

For the third research question: Are group consensus estimates using the Delphi method more accurate than the individual estimation method?

The average MRE of the individual estimates is 0.4863 (48.63%) and of the Delphi estimates is 0.4097 (40.97%). The average BRE of the individual estimates is 1.2875 (128.75%) and of the Delphi estimates is 0.9063 (90.63%). It suggests that the Delphi estimates are more accurate than individual estimates both in terms of MRE and BRE. The numbers are also statistically significant (p-Value of 0.007 for MRE and 0.003 for BRE).

In conclusion, this may be the effect of the Delphi method and the exposing to the experts the three algorithmic model estimates in the Delphi experiment. However, in the accuracy performance point of view, the average MRE of 0.4097 (40.97%) and BRE of 0.9063 (90.63%) are considered far from satisfactory. Only 3 projects (1, 7, and 8) show both MRE or BRE of less than 25% or PRED (0.25) = 0.30. The concluded results seem to conform to the postulation of the research. The adapted Delphi experiment shows a promising technique for software cost estimation.

Another major threat to the validity of the results is the experimental design. This includes the subjects (the experts), the material (the software projects), the process (the Delphi steps), and the experimental setting. The number of 5 experts is generally acceptable for the expert judgment experiment. However, only one group of 5 experts and 10 software projects were tested. The experts in this experiment were solicited voluntarily. The characteristic of these experts may differ from the environment other than the projects to develop especially for the "team motivation". The experts in real life are often allocated from inside the organization who may possess better knowledge about the environmental factors affecting the development effort. This may contribute to the external validity for generalization. Many drawbacks of the expert judgment method, for example, being biased or inconsistent of the expert judgment, time-consuming, and dependent on the expert experience, may also hinder the Delphi performance and need further investigation.

REFERENCES

[1] Arnuphaptrairong T. (2018) "The State of Practice of Software Cost Estimation: Evidence From Thai Software Firms", in IMECS 2018 Proceedings of the International Multi-Conference of Engineers and Computer Scientists 2018, Vol II, March 14-16, Hong Kong.

[2] Hihn J. and Habib-agahi H. (1991) “Cost Estimation of Software Intensive Projects: A Survey of Current Practices”, in ICSE '91 Proceedings of the 13th international conference on Software engineering, 1991, pp.276-287.

[3] Verner J. M. and Evanco W. M. (2000) ‘State of the Practice of Effort Estimation in Business Environment’, in Proceedings of the ESCOM-SCOPE, 2000, pp.229-238.

[4] Moløkken-Østvold K. and Jørgensen M. (2003) “A Review of Surveys on Software Effort Estimation”, In IEEE International Symposium on Empirical Software Engineering (ISESE 2003), September 30 - October 1, 2003, Rome, Italy.

[5] Moløkken-Østvold K. and Jørgensen M. and Tanilkan S.S. (2004) “A Survey on Software Estimation in the Norwegian Industry”, in METRICS'04 Proceeding of the 10th International Symposium on Software Metrics, 2004.

[6] Yang D., Wang Q., Li M., Yang Y., Ye K. and Du J. (2008) “A Survey on Software Cost Estimation in the Chinese Software Industry”, in ESEM '08 Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, 2008, pp.253-262.

[7] Trendowicz A., Munch J. and Jeffery R. (2011) 'State of the Practice in Software Effort Estimation: A Survey and Literature Review’, in IFIP International Federation for Information Processing, 2011, pp.232-245.

[8] Mansor Z., Kasirun Z. M., Yahya S. and Arshad N. H. (2011) “Current Practices of Software Cost Estimation Technique in Malaysia Context”, in ICIEIS 2011: Informatics Engineering and Information Science, pp.566-574, 2011.

[9] Boehm B. W., Rifkin S. and Jørgensen M. (2009) “Software Development Effort Estimation: Formal Models or Expert Judgment?” IEEE Software, Vol. 26, pp.14-19.

[10] Moløkken-Østvold K., Haugen N. C. and Benestad H. C. (2008) “Using Planning Poker for Combining Estimates in Software Projects”, Journal of Systems and Software, Vol.81, No.12, pp.2106-2117.

[11] Passing U. and Shepperd M. (2003) ‘An experiment on software project size and effort estimation’, in ISESE 2003 International Symposium on Empirical Software Engineering, 2003.

[12] Moløkken-Østvold K. and Jørgensen M. (2004) “Group Process in Software Effort Estimation”, Empirical Software Engineering, Vol.9, pp.315-334.

[13] Gandomani T. J., Wei K. T. and Binhamid A. K. (2014) “A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique”, International Journal of Software Engineering and its Applications, Vol. 8, No.11, pp.173-182.

[14] Jørgensen M. (2004) “A review of studies on expert estimation of software development effort”, Journal of Systems and Software archive, Vol. 70, No.1-2, pp.37-60.

[15] Kusters R. J., van Genuchten M. J. I. and Heemstra F. J. (1990) ‘Are software cost-estimation models accurate?’ Information and Software Technology, Vol. 32, No.3, pp.187-190.

[16] Jørgensen M. (2005) “Practical Guidelines for Expert-Judgment-Based Software Effort Estimation”, IEEE Software, pp.57-63, May-June.

[17] Jørgensen M. (2007) ‘Forecasting of software development work effort: Evidence on expert judgement and formal models’, International Journal of Forecasting, Vol.23, No.3, pp.449-462.

[18] Sehra S. K., Brar Y. S., Kaur N., and Sehra S. S. (2017) “Research patterns and trends in software effort estimation”, Information and Software Technology, Vol.91, pp.1–21.

[19] Boehm B. W. (1984) “Software engineering economics”. IEEE Transactions on Software Engineering, Vol. 10, No.1, pp.7-19.

[20] Hughes R. T. (1996) “Expert judgement as an estimating method”, Information and Software Technology, Vol.38, No.2, pp.67-75.

[21] Rush C. and Roy R. (2001) “Expert Judgement in Cost Estimating: Modelling the Reasoning Process”, Concurrent Engineering: Research and Applications, Vol.9, No.4, pp.271-284.

[22] Faria P. and Miranda E. (2012) “Expert Judgment in Software Estimation During the Bid Phase of a Project -- An Exploratory Survey”, in Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), Assisi, Italy, 2012, pp.126-131.

[23] Lin S. W. and Bier V. M. (2008) “A study of expert overconfidence”, Reliability Engineering & System Safety, vol. 93, No. 5, pp. 711-721.

[24] Graefe A. and Armstrong J. S. (2011) “Comparing Face-to-Face Meetings, Nominal Groups, Delphi and Prediction Markets on an Estimation Task”, International Journal of Forecasting, Vol. 27, No.1, pp.183-195.

[25] Woudenberg F. (1991) "An evaluation of Delphi", Technological Forecasting and Social Change, Vol.40, No.2, p.131-150.

[26] Rowe G. and Wright G. (1999) "The Delphi technique as a forecasting tool: issues and analysis", International Journal of Forecasting, Vol.15, No.4, pp.353-375.

[27] Kaplan A., Skogstad A. and Cirshick M.A. (1949) The Prediction of Social and Technological Events, Rand Corporation. April.

[28] Gordon T. J. and Helmer O. (1964) Report on a Long-range Forecasting Study, Rand Corporation, P-2982, September.

[29] Dalkey N. C. and Helmer O. (1963) "An experimental application of the Delphi method to the use of experts", Management Science, Vol. 9, pp.58–467.

[30] Arnuphaptrairong T. (2016) "A Literature Survey on the Accuracy of Software Effort Estimation Model"', in Proceedings of the International MultiConference of Engineers and Computer Scientists 2016 Vol II, IMECS 2016, March 16 - 18, 2016, Hong Kong.

[31] Albrecht A. J. (1979) "Measuring application development productivity", in Proceeding of the IBM Applications Development Symposium, California, October 14-17, 1979, pp. 83-92.

[32] Gencel C. and Demirors O. (2008) "Functional size measurement revisited", ACM Transaction on Software Engineering and methodology, vol.17 No. 3, pp.15.1-15.36.

[33] Boehm B. W. (1981) Software Engineering Economics, Prentice-Hall, New Jersey.

[34] Centre for Software Engineering, USC, COCOMO II Model Definition Manual, Version 2.1, 1995-2000.

[35] Karner G. (1993), Metrics for objector, Diploma Thesis, University of Linkoping, Sweden, No. LiTH-IDA-Ex- 9344:21.

[36] Anda B., Dreiem H., Sjoberg Dag I. K. and Jørgensen M. (2001), "Estimating software development effort based on Use Cases --experiences from Industry", In G. Martin, and C. Kobryn (ed) UML 2001: the unified modeling language: modeling languages, concepts, and tools: 4th international conference, Toronto, Canada, December 1993.

[37] Kitchenham B. S., Pfleeger L., McColl B. and Eagan S. (2002) "An empirical study of maintenance and development estimation accuracy", Journal of systems and software, Vol. 64, pp.55–77.

[38] Zuber J. A., Crott H. W. and Werner J. (1992) "Choice shift and group polarization: an analysis of the status of arguments and social decision schemes", Journal of Personality and Social Psychology, Vol. 62, pp.50–61.

[39] Buehler R., Messervey D. and Griffin D. (2005) "Collaborative planning and prediction: does group discussion affect optimistic biases in time estimation?" Organizational Behavior and Human Decision Processes, Vol. 97, pp.47–63.

[40] Fenton N. E. and Pfleeger S. L. (1997) "Software Metrics: A rigorous and Practical Approach", Paper presented at the International Thomson Computer Press.

[41] Miyazaki Y., Takanou A., Nozaki H., Nakagawa N. and Okada K. (1991) "Method to estimate parameter values in software prediction models", Information and Software Technology, Vol.33, pp.239–243.

[42] Basha S. and Dhavachelvan P. (2010) "Analysis of Empirical Software Effort Estimation Models", International Journal of Computer Science and Information Security, Vol.7, No.3, pp. 68-77.

[43] Conte S., Dunsmore H. and Shen V. (1986) Software Engineering Metrics and Models, Menlo Park, Benjamin/Cummings.

Authors

Tharwon Arnuphaptrairong is an Associate Professor in Business Information Technology at the Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Thailand. He received a B.Sc. Degree in Statistics from Chulalongkorn University, a M.Sc. in Computer Applications from Asian Institute of Technology, Bangkok, Thailand, and a Ph.D. Degree in Management Sciences from University of Waterloo, Canada. His research interests include Software Project Management, Software Risk Management, Software Cost Estimation and Empirical Software Engineering.