Pertanika J. Sci. & Technol. 30 (4): 2851 - 2878 (2022)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Significant Factors in Agile Software Development of Effort Estimation

Pantjawati Sudarmaningtyas[1,2] and Rozlina Mohamed[1]*

[1]Faculty of Computing, Universiti Malaysia Pahang, 26300 UMP, Gambang, Kuantan, Pahang, Malaysia
[2]Department of Information System, Universitas Dinamika, 60298 Surabaya, Jawa Timur, Indonesia

ABSTRACT

The Agile effort estimation involves project-related and people-related factors. This research objective is to find the factors that influence Agile effort estimation significantly through path analysis using a structural equation model. This research built an agile effort estimation path coefficient model from six constructs from theories and previous studies. Project-related factors represent by requirement and design implementation constructs. People-related factors are measured by the construct of experience, knowledge, and technical ability. The last construct is the effort itself. SmartPLS is employed for the confirmatory composite analysis and the structural model assessment. The confirmatory composite analysis indicated that all constructs are reliable and valid. Furthermore, the structural model assessment found that all factors of project-related constructs have a positive relationship and significant influence, showing a coefficient path value of 59.1% between requirement and design implementation constructs. All constructs represent people-related factors indicated by the coefficient path value of 67% between experience and knowledge, 42.6% between experience and technical ability, and 54.4% between knowledge and technical ability. In addition, all constructs proved influential simultaneously to effort by 31.1%. Positively contribute provided by requirement, experience, and technology's ability. Significantly influenced provided by constructs of the developer's knowledge and technical ability. The largest effect is given by technical ability, knowledge, and experience on medium and small scales. Contrarily, both constructs from project-related effects can be negligible because there was no influence. Based on the result,

ARTICLE INFO

Article history:
Received: 26 November 2021
Accepted: 14 March 2022
Published: 19 September 2022

DOI: https://doi.org/10.47836/pjst.30.4.30

E-mail addresess:
pantja@dinamika.ac.id (Pantjawati Sudarmaningtyas)
rozlina@ump.edu.my (Rozlina Mohamed)
*Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

this study concludes that the significant factors in Agile effort estimation are technical ability, knowledge, and experience.

Keywords: Agile methodology, effort estimation, path analysis, people-related factors, project-related factors, structural equation model

INTRODUCTION

One of the crucial parts of a software project is effort estimation. The activity estimates the effort necessary for developing the software product in either man-days or man-hours. In addition, the effort is a key factor that serves as a basis for calculating the cost and schedule needed for completing the project (Bloch et al., 2012). Generally, an effort is formed by a combination of people and time, which calculates the number of productive working hours needed to complete a job. Man-hours, man-days, man-months, or man-years are typically used to express the units of effort (Trendowicz & Jeffery, 2014).

In contrast, agile software development methodology is more emphasis on coding, shorter delivery cycles, and several iterations to complete the software development. Therefore, the effort estimation in this methodology is crucial because delivery time and project velocity are calculated based on the estimation results. Another characteristic of Agile methodology is the collaborative and cooperative approach between all stakeholders, actively involving users and empowering the team to make decisions (Project-Management.com, 2019). Agile's estimation process is divided into two phases: early estimation and iterative estimation. Early estimation was used to get the initial scope just enough to describe the entire software project. Meanwhile, iterative estimation conducted at the start of an iteration is to anticipate new or change requirements.

The top three methods currently used in agile are machine learning, expert judgment, and algorithmic. The Planning Poker technique is a part of the expert judgment method most implemented in Agile. Although this technique is in accordance with Agile characteristics, the result has shown biased value (Sudarmaningtyas & Mohamed, 2021). In addition, these methods heavily rely on the estimator experience and rarely involve other factors in the process of estimation.

The Agile estimation identified factors that are critical in determining software effort. In addition, people factors are also important because the estimation process involves a team of developers from different disciplines (Munialo & Muketha, 2016). Therefore, this study is conducted to find the significant factors that influence effort estimation in Agile.

Considering the importance of estimation efforts in Agile, expected the significant factors produced in this research can contribute to improving the existing Agile effort estimation method. In addition, implementation of those factors expected creates a better estimate of effort and increases accuracy while reducing the technicality and time required.

This paper organizes into five sections. Section 1 contains the research background, followed by describing related works. Section 3 discusses the research methodology that involves the development of the path coefficient model, a questionnaire based on our research questions and hypothesis, and examining and assessing the structural model. Section 4 is the result and discussion of our findings, and the last section is wrapped up our conclusions.

RELATED WORKS

Agile software development (ASD) methods are lightweight methods that focus on simplicity, speed, self-organizing teams, and involving the customer as part of the team. Simplicity and speed can be achieved through more straightforward design and iterative development cycles. Each development cycle emphasizes delivering a demonstrable working product that focuses on the main functions. The customer involved in each development cycle is like a team member so that the requirement customer, who often changes, can be anticipated quickly by the team (Abrahamsson et al., 2002; Bourque & Fairley, 2014).

ASD is based on an iterative and incremental development model that promotes rapid response to changes and focuses on customer satisfaction, timely and continuous delivery, informal methods and minimal planning (Fernandez-Diego et al., 2020). ASD methods have several advantages, such as delivering working software faster, dealing with changing user requirements, and promoting better working relationships among all stakeholders (Zhang et al., 2010). Those advantages can significantly reduce the apparent overhead associated with heavyweight, plan-based methods used in large-scale software development projects and resolve issues of slow execution on concurrent engineering (Boehm, 2006; Bourque & Fairley, 2014).

Scrum, Xtreme Programming (XP), Test Driven Development (TDD), Agile Unified Process (AUP), Kanban, and Distributed Agile Software Development (DASD) are ASD dynamic methods available for developing a software product. Each method has unique attributes and qualities, so choosing the proper method is critical and should be based on the project requirements (Fernandez-Diego et al., 2020; Prakash & Viswanathan, 2017).

Effort estimation is a crucial part of ASD for three reasons: 1) maximize project velocity; 2) optimize individual developer effort across multiple projects; 3) optimal scheduling to achieve global efficiencies (Malgonde & Chari, 2019). However, most of the effort estimation method used in ASD is the expert judgment method, especially the Planning Poker technique (Sudarmaningtyas & Mohamed, 2021).

Following the characteristics of ASD, effort estimation is usually conducted by structured group consensus and lightweight approaches. In addition, it is also based on the user story that delivers in one or two sentences and contains pieces of functionality or feature software worth for the user (Trendowicz & Jeffery, 2014). Commonly, estimates are conducted periodically, internally generated, and must be involved stakeholders to compromise, review, and reach an agreement concerning the number of resources and time to finish the projects (Bourque & Fairley, 2014). Providing an approximation of the resources needed to complete a project, especially the delivery of products or services in accordance with the specified characteristics of functional and non-functional, is the objective of the estimation effort (Project Management Institute, 2017).

Instead of giving a single value, the estimation value is better expressed in intervals because the estimator has confidence in the possibility that the actual efforts will be within range (Jørgensen, 2016). In addition, it could also provide three estimates: best-case (optimistic), normal-case (most likely), and worst-case (pessimistic) scenarios. To obtain best-case and worst-case scenarios, multiply percentages according to the organization's norms and confidence level in the opportunity in question with normal-case (Chemuturi, 2009).

The Agile effort estimation is influenced by factors related to the project and factors related to people. The factors related to the project include the type of project, quality requirements, hardware and software requirements, ease of operation, complexity, data transaction, and multiple sites. People-related factors consist of communication skills, familiarity with a team, managerial Skills, security, working time, experience with the previous project, and technical ability (Popli & Chauhan, 2014). Accuracy in the agile effort estimation relied heavily on the expertise of professionals in software development. Thereby expert groups' estimates could diminish the optimism bias arising from group discussion (Lenarduzzi et al., 2015; López-Martínez et al., 2018; Mahnič & Hovelja, 2012).

METHODOLOGY

Development of Path Coefficient Model

In the Agile effort estimation, the project factors are essential to determine. In addition, people factors are also important because the estimation process involves a team of developers from different disciplines (Munialo & Muketha, 2016; Popli & Chauhan, 2014). People-related factors represented each team's expertise, experience, domain knowledge, and technical ability (Popli & Chauhan, 2014). For better results, estimation effort should include observance of various factors derived from software development methodology. For that reason, project-related factors acquired from an Agile development methodology consist of two activities which are: requirement engineering and design implementation (Chemuturi, 2009; Sommerville, 2011). The mapping factors are based on the theoretical aspect of the previous research described in Table 1.

Table 1
Mapping factors based on the theoretical aspect

	Agile Development Methodology	
People Factors	Requirement Engineering	Design Implementation
Experience	The quality requirements	Ease of operation
Domain knowledge		Complexity
Technical ability		Data transaction

This study constructs a relation of factors grounded in previous literature studies, and the graph consists of three parts, as depicted in Figure 1. The first part, indicated with the orange dash line, represents project-related factors derived from the Agile development methodology column in Table 1, comprising two variables. The second part, the area surrounded by the blue dash line, corresponds with the people factors column, containing three variables. Finally, the last part reflects the effort itself. Survey results will confirm the relation of factors in this graph to achieve this research objective.

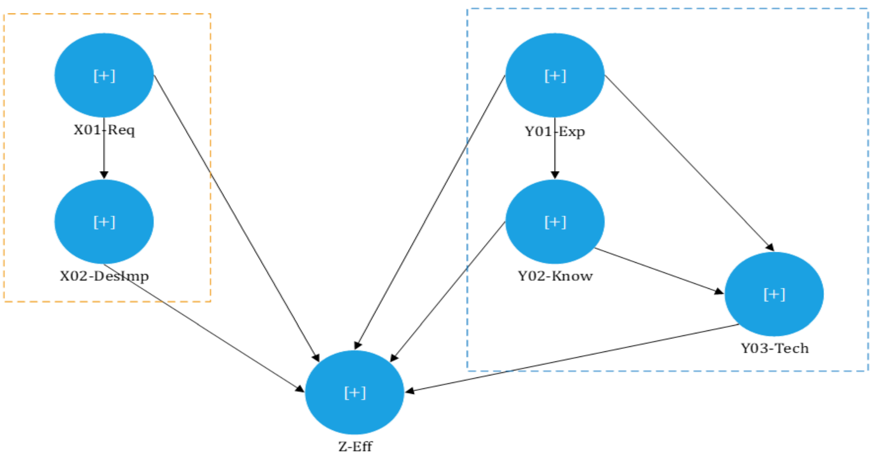

Figure 1. The factors relation grounded in previous literature studies

Project-related factors comprise the requirement and design implementation variables, symbolized by X01-Req and X02-DesImp, respectively. People-related factors, indicated by Y01-Exp, Y02-Know, and Y03-Tech, are represented by experience, knowledge domain, and technical ability. In addition, the effort constructed is denoted by Z-Eff. Based on theory, variable in the project-related factors has a correlation, where the requirement influences design implementation. On the other hand, variable in the people-related factors has a relation that assumes that developers' experience would leverage their knowledge and technical ability. In addition, technical ability is also affected by knowledge. Therefore, the effort is influenced by the requirement, design implementation, experience, knowledge, and technical ability.

This study proves assumptions on this model through nine hypotheses constructed based on research questions emerge. Table 2 reveals the research questions derived from this model and the hypotheses related to the research questions. The first column contains research questions about the relationship of any constructs, while the next column explains hypotheses related to those research questions.

Table 2
Research question lead hypotheses

Research Question	Hypothesis
RQ1: Do requirements influence the design implementation?	H1: The requirement construct is positively related to the design implementation construct.
RQ2: Do requirements affect effort estimation?	H2: The requirement construct is positively related to the effort construct.
RQ3: Does design implementation affect the effort?	H3: The construct of design implementation is positively related to the effort construct.
RQ4: Does experience have an impact on a developer's knowledge domain?	H4: The experience construct is positively related to the knowledge domain construct.
RQ5: Does experience influence a developer's technical ability?	H5: The experience construct is positively related to the technical ability construct.
RQ6: Do experience influence effort estimation?	H6: The experience construct is positively related to the effort construct.
RQ7: Does the knowledge affect the developer's technical ability?	H7: The knowledge domain construct has positive relationships with the technical ability construct.
RQ8: Does knowledge have an impact on effort estimation?	H8: The knowledge domain construct has positive relationships with the effort construct.
RQ9: Does the technical ability influence effort estimation?	H9: The construct of technical ability has a positive relationship with effort.

The relationship and influence between constructs are captured in Table 3. The relation notation between two constructs is written in the first column, the relation direction in the second column, and the associated hypotheses number in the last column.

Table 3
Structural relationships and hypotheses

Structural Relationships	Direction	Hypothesis
X01-Req -> X02-DesImp	+	H1
X01-Req -> Z-Eff	+	H2
X02-DesImp -> Z-Eff	+	H3
Y01-Ex -> Y02-Know	+	H4
Y01-Ex -> Y03-Tech	+	H5
Y01-Ex -> Z-Eff	+	H6
Y02-Know -> Y03-Tech	+	H7
Y02-Know -> Z-Eff	+	H8
Y03-Tech -> Z-Eff	+	H9

The first column in Table 3 contains a notation of relations between two constructs, and the second column contains the relation direction. For example, the structural relationship denoted X01-Req -> X02-DesImp indicates that the requirement construct influences the design implementation construct. When X01-Req positively influences X02-DesImp, the direction column fills with a plus sign (+). In contrast, the negative direction relationship signaled the negative impact between the two constructs.

Questionnaire

To confirm the relations of those factors conducted by the survey through distributed questionnaire was developed based on six constructs in Figure 1, and several indicators measure each construct. The association between factor, construct, and indicator is revealed in Table 4.

Table 4
Factor, construct, and indicator relationship

Factors Derived from Previous Literature Studies	Construct	Indicator	Description
Requirement Engineering	Requirement	Functional requirement Non-functional requirement	Measure the quality requirements

Table 4 *(Continue)*

Factors Derived from Previous Literature Studies	Construct	Indicator	Description
Design Implementation	Design Implementation	User interface	Assess the ease of operation
		Software and hardware interfaces	
		Diversity of technology	Represent the complexity
		The sophisticated or novelty technology	
		Coding complexity	
		Database size	Element of the data transaction
		Database complexity	

Construct is a representative of a conceptual definition built from a set of indicators. An indicator is in the form of a single variable used in conjunction with one or more variables to form a composite measure, where each indicator has attributes. Attributes embedded in indicators are complexity and size because these attributes mostly used agile effort estimation in the last three years (Fernandez-Diego et al., 2020; Sudarmaningtyas & Mohamed, 2021). All constructs, indicators, and attributes used in the questionnaire are briefly described in Table 5, and the questionnaire used to collect data is entirely served in Appendix 1.

Table 5
Questionnaire constructs and indicators

Construct	Scales Type	Indicator	Description
X01-Req (Requirement)	1:Never (0%) 2: Occasionally (30%) 3:Sometimes (50%) 4:Normally (80%) 5:Always (100%)	X1: Functional requirement Complexity	Frequency of the complexity of *functional requirements* considered by developers in estimating efforts.
		X2: Non-functional requirement Complexity	Frequency of the complexity of *non-functional requirements* considered by developers in estimating efforts.

Table 5 *(Continue)*

Construct	Scales Type	Indicator	Description
X02-DesImp (Design Implementation)	1:Never (0%) 2: Occasionally (30%) 3:Sometimes (50%) 4:Normally (80%) 5:Always (100%)	X3: User interface Complexity	In effort estimation, directly or indirectly, consider the complexity of the software user interface
		X4: Software and hardware interfaces Complexity	In effort estimation, directly or indirectly, consider the complexity of interfaces between software and hardware
		X5: Diversity of technology	In effort estimation, directly or indirectly, consider the diversity technology that uses
		X6: The sophisticated or novelty technology	In effort estimation, directly or indirectly, consider the sophisticated or novelty technology that uses
		X7: Coding Complexity	In effort estimation, directly or indirectly, consider the difficulty level of coding
		X8: Database Size	In effort estimation, directly or indirectly, consider the database size
		X9: Database Complexity	In effort estimation, directly or indirectly, consider the database complexity
Y01-Exp (Experience)	1: None; 2: Low; 3: Fair; 4: High; 5: Very High.	Y1: Job experience	Adequate job experience can affect a person produce more accurate estimates
		Y2: Effort estimation experience	Experience in estimating effort may affect one individual to produce a more accurate estimate.
Y02-Know (Domain Knowledge)	1: None; 2: Low; 3: Fair; 4: High; 5: Very High.	Y3: Similar project	Frequently being involved in similar projects can affect a person to produce more accurate estimates.
		Y4: Good track record	A good track record in previous projects can affect a person to produce more accurate estimates.
Y03-Tech (Technical Ability)	1: None; 2: Low; 3: Fair; 4: High; 5: Very High.	Y5: Technical ability	Developers who have technical expertise can produce more precise estimates.
		Y6: Involve in many projects	Involving in many projects can affect a person producing more accurate estimates.

Table 5 *(Continue)*

Construct	Scales Type	Indicator	Description
Z-Eff (Effort)	0: No; 1: Yes.	Z1: Guess/intuition	Mechanism of estimating effort just by guessing or based on intuition.
		Z2: Experience	Mechanism of estimating effort based on experience.
		Z3: Track record	Mechanism of estimating effort based on previous projects' track record.
		Z4: Attributes	Mechanism of estimating effort using specific attributes that are related to the project.

Requirement constructs have two indicators because software system requirements are generally classified as functional or non-functional. The indicators are used to measure design implementation, classified by the interface, technology, developing programs, and database. The chosen indicator represents an important design implementation process because most agile methods users do not require detailed design documentation. Interface indicator denoted by the user interface and hardware-software interface. Technology is represented by indicators of diversity and sophistication or novelty. Finally, coding is an indicator for developing a program, while the size and complexity indicators represent the database (Hamouda, 2014; Khatri et al., 2016; Pasuksmit et al., 2021; Rosa et al., 2021; Sommerville, 2011; Yuliansyah et al., 2018).

The developer's experience, domain knowledge about the project, and technical ability are important aspects of software project development experience measured by job experience and effort estimation experience. In addition, domain knowledge is measured by being frequently involved in similar projects and having a good track record in previous projects. Finally, technical ability is measured by the developer's technical ability and involvement in many projects (Adnan & Afzal, 2017; Fernandez-Diego et al., 2020; Popli & Chauhan, 2014).

How to perform the effort estimation is an indicator of effort. Performing effort estimation through guess or intuition is the first indicator. The second indicator performs effort estimation based on experience. The third estimation effort is based on the track record in previous projects, and the last uses attributes-related projects to estimate the effort (Popli & Chauhan, 2013).

Respondents

Agile effort estimation is usually done through a group discussion that empowers the developers' team to decide the estimated effort (Sudarmaningtyas & Mohamed, 2020). Therefore, this study is targeting developers as respondents. In addition, estimation efforts conducted by the developers have higher accuracy than beginner's estimates (Lenarduzzi et al., 2015; López-Martínez et al., 2018; López-Martínez et al., 2017). The expertise of professionals in software development is the primary aspect that influences estimation accuracy through Planning Poker. Therefore, the estimation conducted by expert groups could be diminished the optimism bias arising from group discussion (Lenarduzzi et al., 2015; López-Martínez et al., 2018; Mahnič & Hovelja, 2012).

The sample size was determined by G*Power software because the previous studies did not specifically mention the number of respondents. Our sample size is calculated with criteria t-test linear multiple regression with a significant error of 5%, effect size 0.35, and 5 predictors. With those criteria, the minimal sample size that G*Power suggests is 40. The source data in this study were collected from 41 developers as respondents, fulfilling the minimal sample size. Respondent demographics are classified by gender and three aspects, as revealed in Table 6.

Table 6
Respondent demographics

Aspect		Male	Female	Total
Job Experience	> 10 years	5	3	8
	6–10 years	6	1	7
	3–5 years	10	4	14
	< 3 years	12	0	12
Total		33	8	41
Project Experience	> 10 projects	16	4	20
	6–10 projects	7	0	7
	3–5 projects	7	3	10
	< 3 projects	3	1	4
Total		33	8	41
Conduct Effort Estimation	Always (100%)	11	3	14
	Normally (80%)	17	3	20
	Sometimes (50%)	5	2	7
	Occasionally (30%)	0	0	0
	Never (0%)	0	0	0
Total		33	8	41

Table 6 shows that as much as 80% of respondents are dominated by males, while 20% are female. In addition, 71% of respondents have job experience of more than three years, and 66% have been involved in more than six projects. Most respondents (83%) have experience in effort estimation, and 95% of respondents believe that effort estimation result is more accurate when attributes associated with the project are considered during estimation—raw data from the survey is contained in Appendix 2.

Examine and Confirm the Structural Model

The structural model was examined and confirmed by executing a confirmatory composite analysis (CCA) by evaluating path coefficients, construct reliability, and construct validity. The indicator loadings measure Path Coefficients. Construct Reliability can be assessed using the traditional Cronbach's alpha and composite reliability approach. However, composite reliability is the preferred reliability metric for structural equation modeling (SEM) statistical techniques. Finally, construct Validity is evaluated by examining convergent and discriminant validity (Hair et al., 2020).

The path coefficient examined by the values of indicator loadings should be at least 0.50 and ideally 0.708 or higher. The reliability of constructs is suggested to be above 0.70 for both Cronbach's alpha and composite reliability. Convergent validity is measured by the average variance extracted (AVE) with a value of at least 0.5. Discriminant validity is evidenced when a reflective construct has the strongest relationships with its indicators compared with any other construct in the path model. Discriminant validity is evaluated through the result of the Fornell-Larcker criterion. On this criterion, the shared variance within the constructs should be larger than the shared variance between the constructs (Bourque & Fairley, 2014).

Assess the Structural Model

The structural model, known as the inner model, was assessed by evaluating the size and significance of the structural path relationships, asses R^2, examining the f^2 effect size, and evaluating the predictive relevance based on Q^2. Evaluating the size and significance of the structural path relationships is required to examine whether the determined hypotheses gain empirical support that, indicated by t values, exceed +/- 1.645. The asses R^2 measures the variance of endogenous variables exhibited X02-DesImp, Y02-Know, Y03-Tech, and Z-Eff (Hair et al., 2020).

The f^2 statistics indicate the relative strength of each independent variable in predicting the dependent variable. The f^2 was evaluated by observing the change in R^2 when each independent variable was excluded (Equation 1). The f^2 value of 0.02, 0.15 and 0.35 represents small, medium, and large effects (Hair et al., 2020).

$$f^2 = \frac{R^2_{included} - R^2_{excluded}}{1 - R^2_{included}} \qquad (1)$$

By evaluating the Q^2, the positive value indicates that the model can accurately predict the data points as reflective indicators of endogenous variables (Hair et al., 2020).

RESULTS AND DISCUSSION

Confirmatory Composite Analysis

The preliminary path coefficient model to find significant factors in Agile effort estimation (AEE) has six constructs and 19 indicators, as revealed in Figure 2. Two constructs in project-related factors are measured by 11 indicators, where requirement constructs have two indicators, while nine support the design implementation construct. Each construction represents a people-related factor measured by two indicators, and the rest contributes to the efforts' construct. We implement the preliminary path coefficient model in SmartPLS.

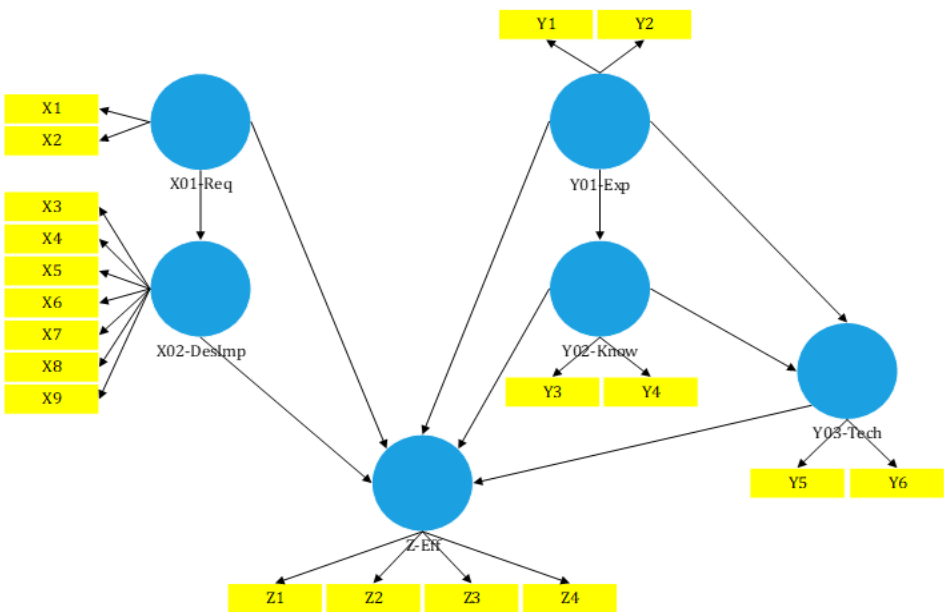

Figure 2. Preliminary path coefficient model

The result of outer loading in the first running found that indicators X7 and Z1 in X02-DesImp and Z-Eff constructs have insignificant *t* Statistic. Consequently, those indicators were omitted from the preliminary model. Furthermore, the X02-DesImp construct AVE value is lower than requisite in the second execution. Therefore, X8 as the indicator with the lowest outer loading is deleted. Thus, the X02-DesImp construct is expected to reach the value of the required AVE. Finally, all constructs have achieved the requisite AVE value in the third running, and the confirmed path coefficient model is depicted in Figure 3.

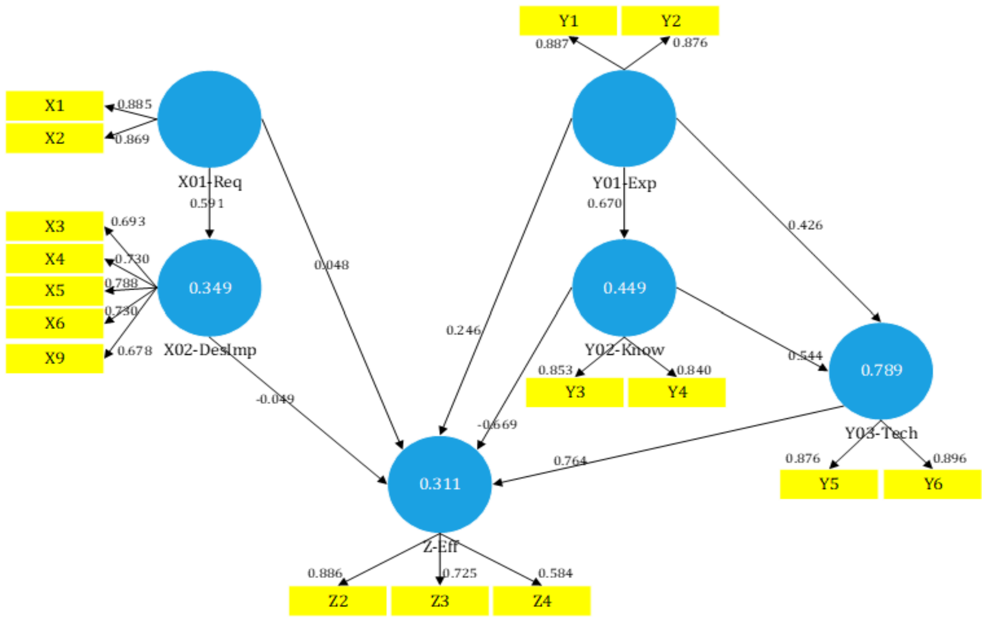

Figure 3. Confirmed path coefficient model

In the confirmed path coefficient model, almost all indicators have ideal factor loadings values, except X3, X9, and Z4. Although the factors are not ideal, the third indicator complied with the rule as the load factor value is above 0.5. Therefore, the third indicator has fulfilled the rule because the loading factor value is above 0.5.

Constructs that represent project-related factors, both indicators in the constructs of Requirement (X01-Req) can be measured significantly with the loadings of functional requirement (X1) is 88.5% and non-functional requirement (X2) is 86.9%. In addition, the construct of Design and Implementation (X02-DesImp) influenced significantly worth 69.3% of the user-interface complexity (X3), 73% of the complexity of the software-hardware interface (X4), 78.8% by a variety of technologies (X5), 73% of sophisticated/novelty of the technology (X6), and 67.8% of complexity database (X9). However, the Design and Implementation constructs are insignificantly influenced by indicators of difficulty of coding (X7) and database size (X8).

In contrast to project-related factors, all indicators on each construct representing the people-related factors can significantly influence because they have loadings factor values above the ideal value. In addition, the Effort (Z-Eff) construct has the same situation as the X02-DesImp construct, where not all indicators are significantly influential. The indicators that significantly influence Z-Eff are Z2, Z3, and Z4, with loadings factor values worth 88.6%, 72.5%, and 58.4%, while indicator Z1 (guessing/intuition) proved no effect. A conventional reliability assessment with Cronbach's alpha indicates that four constructs have good reliability, while the others are classified as acceptable. Likewise, composite reliability, the preferred reliability metric in sem, reflects good reliability because the value is above 0.70 for all constructs. Therefore, all constructs in this model are reliable and valid, as shown in Table 7.

Convergent validity assessed by AVE demonstrates good validity because the AVE value is above 0.5 for all constructs. In addition, discriminant validity assessed by Fornell-Larcker also indicates good validity. It is supported that the value of the reflective construct with its indicators is higher than any other construct in the path model.

Table 7
Results of reliability assessment

Construct	Reliability Assessment		Validity Assessment	
	Cronbach's Alpha	Composite Reliability	AVE	Fornell–Larcker
X01-Req	0.700**	0.869**	0.769	0.877***
X02-DesImp	0.776**	0.847**	0.525	0.725***
Y01-Exp	0.713**	0.874**	0.777	0.881***
Y02-Know	0.604*	0.835**	0.716	0.846***
Y03-Tech	0.726**	0.879**	0.784	0.886***
Z-Eff	0.576*	0.781**	0.550	0.742***

Note: Acceptable; **Good; ***The value of the reflective construct with its indicators is higher than any other construct.

The measurement quality of the CCA result shows that all constructs in the path coefficient model can be confirmed. This result also assures that the confirmed path coefficient model's six reflectively measured composite constructs are reliable and valid.

The Structural Model Assessment

Estimates of path coefficients and significance of the structural path relationships are conducted to examine whether there is empirical support for the pre-determined hypotheses. Five hypotheses (H1, H4, H5, H7, H9) are empirically supported because they have consistent and significant values with hypothecated ones. Although H2 and H6 have a positive direction but are insignificant; in contrast, H8 has a significant influence although a negative tendency. In addition, H3 has inconsistent signs and is insignificant. Path coefficients and significance testing as the result of SmartPLS revealed in Figure 3 are presented in Table 8.

The path coefficient values and *t* Statistics indicate that the Requirement construct is positively related to and significantly influences the Design and implementation construct. Thereby, the H1 hypothesis is accepted. Thus, an increase of 55.9% in the construct of

Design and implementation for each increase occurred in the Requirement construct. On the other hand, the Requirement construct is positively related to the Effort construct, although the influence is insignificant. Therefore, it means the H2 hypothesis is also acceptable.

Table 8
Path coefficients and significance testing

Hypothesis	Structural Relationships	Path Coefficients	*t* Statistic
H1	X01-Req -> X02-DesImp	0.591	4.912**
H2	X01-Req -> Z-Eff	0.048	0.184
H3	X02-DesImp -> Z-Eff	-0.049	0.203
H4	Y01-Exp -> Y02-Know	0.670	4.926**
H5	Y01-Exp -> Y03-Tech	0.426	3.686**
H6	Y01-Exp -> Z-Eff	0.246	0.774
H7	Y02-Know -> Y03-Tech	0.544	5.034**
H8	Y02-Know -> Z-Eff	-0.669	2.173*
H9	Y03-Tech -> Z-Eff	0.764	2.511*

Note: $^{*}p<0.05$; $^{**}p<0.01$

Design and Implementation construct are not positively related, and influences are insignificant to the Effort construct. In other words, every increase in Design and Implementation construct will decrease the Effort construct by 4.8%. Due to that reason, the H3 hypothesis is rejected.

The Experience construct has positively related to the constructs of Knowledge, Technical Ability, and Effort. This construct significantly influences the Knowledge construct at 67%, while the Technical Ability construct is influenced by 42.6%. Nevertheless, this construct is insignificant to the Effort construct. Therefore, based on the statement, the hypotheses of H4, H5, and H6 are acceptable.

Knowledge constructs significantly influence the technical ability and Effort constructs, respectively, reaching 54.4% and 66.9%. However, the Knowledge construct contrasts with the other constructs because it has negatively related to the Effort construct. Therefore, hypothesis H7 is acceptable nevertheless rejects hypothesis H8.

Hypothesis H9 is acceptable because the Technical Ability construct has positively associated and significantly influenced the Effort construct. Every improvement in the Technical Ability constructs increases the Effort construct by 76.4%.

Observing R^2 values shows the ability of exogenous constructs to clarify endogenous constructs. The exogenous construct X01-Req gives moderate influences of 34.9% to endogenous construct X02-DesImp, while exogenous construct Y01-Exp gives strong influences of 44.9% to endogenous construct Y02-Know. Endogenous variable Y03-Tech got strong influence simultaneously from exogenous constructs Y01-Exp and endogenous construct Y02-Know as 78.9%. The last endogenous construct is Z-Eff that simultaneously influenced by constructs X01-Req, X02-DesImp, Y01-Exp, Y02-Know, and Y03-Tech by 31.1%. It indicates that the Effort construct was also influenced by another construct not mentioned in this study.

The interest of this study is in the effect size of endogenous construct Z-Eff. Construct Y01-Exp contributes a small size effect to Z-Eff by 0.028. On the other hand, the constructs of Y02-Know and Y03-Tech give a medium effect size to the Z-Eff. In addition, the effect of X01-Req and X02-DesImp is considered negligible because it did not affect the Z-Eff construct. Constructs that account for most of the variance in Z-Eff are Y03-Tech followed by Y02-Know with effect sizes f2 of 0.164 and 0.155. The results of examining the f2 effect size are completely served in Table 9.

The evaluation revealed that all endogenous constructs have Q^2 values: X02-DesImp of 0.148, Y02-Know has 0.279, Y03-Tech by 0.576, and Z-Eff is worth 0.102. This evaluation indicated that all Q^2 values are above 0.0, which means the confirmed path coefficient model can provide relevant predicting for all endogenous constructs.

Table 9
Examine the f^2 effect size

	R^2 for Z-Eff	f^2 Effect Size
R2 Includes all variable	0.311	
R2 Excludes X01-Req	0.309	0.003
R2 Excludes X02-DesImp	0.310	0.001
R2 Excludes Y01-Exp	0.292	0.028
R2 Excludes Y02-Know	0.204	0.155
R2 Excludes Y03-Tech	0.198	0.164

CONCLUSION

This study found that the project-related factors, represented by requirement and design implementation, and people-related factors, which consist of experience, knowledge, and technical ability, proved influential simultaneously to effort by 31.1%. This outcome supports and aligns with Popli and Chauhan's (2014) research result and Munialo and Muketha's (2016). Although all factors simultaneously influence effort, not all factors have a positive relationship and significant influence. According to our acceptable hypotheses, requirement, experience, and technology's ability positively contribute to Agile effort estimation. In addition, Agile effort estimation is significantly influenced by the developer's knowledge and technical ability.

The assessment of the people-related factors provides effect size to effort on a small scale and medium scale, where technical ability (Y03-Tech) gives the largest effect size, followed by knowledge (Y02-know) and experience (Y03-Exp). Contrarily, the effects of project-related can be negligible because there was no influence from both constructs of their representatives. It is consistent with the previous results, stating that requirements, designs and implementation do not significantly influence the effort. Therefore, this study concludes that people-related factors, especially technical ability, knowledge, and experience, are significant in Agile effort estimation.

Our conclusion is supported by the research finding of Asnawi et al. (2012) that software developers' involvement is the top factor in Agile methods. In addition, our acceptable hypotheses strengthened the conclusions research of Ramessur and Nagowah (2020) that stated the two most impacting factors in Agile projects are staff experience and technical ability and complexity of requirements.

Further research is built more accurate Agile effort estimation by implementing those significant factors. In addition, exploring the other factors that influence agile effort estimation is also important because, based on our findings, the simultaneous influence on project-related and people-related factors is 31.1%. This value convinces us that there are still many other indicators that future researchers can explore.

ACKNOWLEDGMENT

The work is supported in part by UMP-UIAM-UITM under Grant Nos. RDU200755 titled "Design and Event Management Prioritization Framework Exploiting Fuzzy Ahp and Fuzzy Topsis Approach."

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis*. arXiv Preprint. http://arxiv.org/abs/1709.08439

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

Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2012). Factor analysis: Investigating important aspects for agile adoption in Malaysia. In *2012 Agile India* (pp. 60-63). IEEE Publishing. https://doi.org/10.1109/AgileIndia.2012.13

Bloch, M., Blumberg, S., & Laartz, J. (2012). *Delivering Large Scale IT.pdf*. https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

Boehm, B. (2006). A view of 20th and 21st century software engineering. In *Proceeding of the 28th International Conference on Software Engineering - ICSE '06* (pp. 12-29). ACM Publishing. https://doi.org/10.1145/1134285.1134288

Bourque, P., & Fairley, R. E. (2014). *Guide to the software engineering - Body of knowledge.* IEEE Computer Society.

Chemuturi, M. (2009). *Software estimation best practices, tools & techniques: A complete guide for software project estimators*. J. Ross Publishing.

Fernandez-Diego, M., Mendez, E. R., Gonzalez-Ladron-De-Guevara, F., Abrahao, S., & Insfran, E. (2020). An update on effort estimation in agile software development: A systematic literature review. *IEEE Access*, *8*, 166768-166800. https://doi.org/10.1109/ACCESS.2020.3021664

Hair, J. F., Page, M., & Brunsveld, N. (2020). *Essentials of business research methods* (4th Ed.). Routledge.

Hamouda, A. E. D. (2014). Using agile story points as an estimation technique in CMMI organizations. In *2014 Agile Conference* (pp. 16-23). IEEE Publishing. https://doi.org/10.1109/AGILE.2014.11

Jørgensen, M. (2016). Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *Journal of Systems and Software*, *117*, 274-281. https://doi.org/10.1016/j.jss.2016.03.048

Khatri, S. K., Malhotra, S., & Johri, P. (2016). Use case point estimation technique in software development. In *2016 5th international conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)* (pp. 123-128). IEEE Publishing. https://doi.org/10.1109/ICRITO.2016.7784938

Lenarduzzi, V., Lunesu, I., Matta, M., & Taibi, D. (2015). Functional size measures and effort estimation in agile development: A replicated study. In *International Conference on Agile Software Development* (pp. 105-116). Springer. https://doi.org/10.1007/978-3-319-18612-2_9

López-Martínez, J., Ram´ırez-Noriega, A., Ju´arez-Ram´ırez, R., Licea, G., & Mart´ınez-Ram´ırez, Y. (2017). Analysis of Planning poker factors between university and enterprise. In *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 54-60). IEEE Publishing. https://doi.org/10.1109/CONISOFT.2017.00014

López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., & Jiménez, S. (2018). User stories complexity estimation using Bayesian networks for inexperienced developers. *Cluster Computing*, *21*(1), 715-728. https://doi.org/10.1007/s10586-017-0996-z

Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, *85*(9), 2086-2095. https://doi.org/10.1016/j.jss.2012.04.005

Malgonde, O., & Chari, K. (2019). An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering, 24*, 1017-1055. https://doi.org/10.1007/s10664-018-9647-0

Munialo, S. W., & Muketha, G. M. (2016). A review of agile software effort estimation methods. *International Journal of Computer Applications Technology and Research*, *5*(9), 612-618.

Pasuksmit, J., Thongtanunam, P., & Karunasekera, S. (2021). Towards just-enough documentation for agile effort estimation: What information should be documented? IN *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 114-125). IEEE Publishing. https://doi.org/10.1109/ICSME52107.2021.00017

Popli, R., & Chauhan, N. (2013). A sprint-point based estimation technique in Scrum. In *2013 International Conference on Information Systems and Computer Networks* (pp. 98-103). IEEE Publishing. https://doi.org/10.1109/ICISCON.2013.6524182

Popli, R., & Chauhan, N. (2014). Agile estimation using people and project related factors. In *2014 International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 564-569). IEEE Publishing. https://doi.org/10.1109/IndiaCom.2014.6828023

Prakash, B., & Viswanathan, V. (2017). A survey on software estimation techniques in traditional and agile development models. *Indonesian Journal of Electrical Engineering and Computer Science*, *7*(3), 867-876. https://doi.org/10.11591/ijeecs.v7.i3.pp867-876

Project-Management.com. (2019). *10 key principles of agile software development*. Project Management.Com. https://project-management.com/10-key-principles-of-agile-software-development/

Project Management Institute. (2017). *A guide to the project management body of knowledge (PMBOK® Guide)* (Sixth Edit). Project Management Institute, Inc. https://www.pmi.org/

Ramessur, M. A., & Nagowah, S. D. (2020). Factors affecting sprint effort estimation. In *Advances in Intelligent Systems and Computing* (Vol. 1089, pp. 507-518). Springer. https://doi.org/10.1007/978-981-15-1483-8_43

Rosa, W., Clark, B. K., Madachy, R., & Boehm, B. (2021). Empirical effort and schedule estimation models for agile processes in the US DoD. In *IEEE Transactions on Software Engineering* (pp. 1-1). IEEE Publishing. https://doi.org/10.1109/TSE.2021.3080666

Sommerville, I. (2011). *Software engineering* (9th Ed.). Addison-Wesley.

Sudarmaningtyas, P., & Mohamed, R. (2021). A review article on software effort estimation in agile methodology. *Pertanika Journal of Science and Technology*, *29*(2), 837-861. https://doi.org/10.47836/pjst.29.2.08

Sudarmaningtyas, P., & Mohamed, R. B. (2020). Extended planning poker: A proposed model. In *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (pp. 179-184). IEEE Publishing. https://doi.org/10.1109/ICITACEE50144.2020.9239165

Trendowicz, A., & Jeffery, R. (2014). *Software project effort estimation: foundation and best practice guidelines for success*. Springer International Publishing. https://doi.org/10.1007/978-3-319-03629-8

Yuliansyah, H., Qudsiah, S. N., Zahrotun, L., & Arfiani, I. (2018). Implementation of use case point as software effort estimation in Scrum Framework. *IOP Conference Series: Materials Science and Engineering*, *403*, Article 012085. https://doi.org/10.1088/1757-899X/403/1/012085

Zhang, X., Hu, T., Dai, H., & Li, X. (2010). Software development methodologies, trends, and implications. *Information Technology Journal*, *9*(8), 173-178.

Appendix 1

Questionnaire

EFFORT ESTIMATION QUESTIONNAIRE

Effort estimation is a crucial part of the software development process because effort estimation accuracy can help determine the software delivery time to customers. Besides that, the estimation effort can be used to calculate the cost of developing software. Trendowich (2014) defines effort as a combination of person and time, representing the amount of fully productive work time one person would need to complete a certain work. This questionnaire aims to determine the developer's point of view about the factors contributing to making an effort estimation and the attributes that influence effort estimation accuracy.
This questionnaire consists of three (3) parts. The first part contains questions about personal data. The second part contains factors that influence the estimation value, and the third part contains parameters that make the estimates more accurate.
To achieve these objectives, please fill out this questionnaire based on your experiences. Researchers would like to thank you for your participation.

* Required

1. Email *

PART 1. PERSONAL DATA

Your personal data is protected, not published, and used only in this research.

2. *Name* *

3. *Gender* *

Mark only one oval.

- ◯ *Male*
- ◯ *Female*

4. *Period of employment (year)* *

Mark only one oval.

- ◯ < 3
- ◯ 3 - 5
- ◯ 6 - 10
- ◯ > 10

5. *Number of involvement in software development projects.* *

Mark only one oval.

- ◯ < 3 projects
- ◯ 3 - 5 projects
- ◯ 6 - 10 projects
- ◯ > 10 projects

PART 2. EFFORT ESTIMATION ATTRIBUTES	*Commonly, software development projects are done by a team. Any members of the team have a specific role, task, and responsibility. The team's estimation determines the completion time of a project to the tasks under their responsibility.* *Fill in the following statements based on your perspective and experience while conducting effort estimation for each task.*

6. *Every involved in a project, I ...* *

Mark only one oval.

	1	2	3	4	5	
Never done effort estimation	◯	◯	◯	◯	◯	*Always doing the effort estimation*

7. *I conduct effort estimation through ...* *

Check all that apply.

- ☐ *Just guessing or based on intuition*
- ☐ *Based on experience*
- ☐ *Based on the track record of the previous project*
- ☐ *Based on specific attributes that related to the project*

8. *Effort estimation more accurate when based on attributes associated with the project.* *

Mark only one oval.

	1	2	3	4	5	
Disagree	◯	◯	◯	◯	◯	*Strongly Agree*

9. *Directly or indirectly, I consider the following attributes to create an effort estimation.* *

Mark only one oval per row.

	Never (0%)	Occasionally (30%)	Sometimes (50%)	Normally (80%)	Always (100%)
The complexity of software user-interface	◯	◯	◯	◯	◯
The complexity of interfaces between software and hardware	◯	◯	◯	◯	◯
Variety of technologies that use	◯	◯	◯	◯	◯
The use of sophisticated/ novelty technologies	◯	◯	◯	◯	◯
Functional requirements complexity	◯	◯	◯	◯	◯
Non-functional requirements complexity	◯	◯	◯	◯	◯
The difficulty level of coding	◯	◯	◯	◯	◯
Database size	◯	◯	◯	◯	◯
Database complexity	◯	◯	◯	◯	◯

10. *Write three (3) attributes that you normally or always use in effort estimation but not yet mentioned in the previous point.*

11. *I generally expressed the value of effort estimation in the form of ... **

Mark only one oval.

- ◯ *Story points that stated the effort thru the Fibonacci numbers series* (1, 2, 3, 5, 8, ...).
- ◯ *T-shirt sizing, the effort size stated small, medium, or large labels that indicate level complexity.*
- ◯ *Time buckets, the effort estimation expressed in days and hours.*

PART 3.
ACCURACY
ATTRIBUTES

Several factors can influence the accuracy of a person in estimates. This section used to know factors affecting someone in making accurate estimates.

12. *In my experience, the following attributes can affect a person to produce more accurate estimates. **

Mark only one oval per row.

	No Impact	*Low Impact*	*Fair Impact*	*High Impact*	*Very High Impact*
Application experience and familiarity.	◯	◯	◯	◯	◯
The technical ability of the developer.	◯	◯	◯	◯	◯
Effort estimation experience.	◯	◯	◯	◯	◯
Involving in many projects.	◯	◯	◯	◯	◯
Adequate job experience.	◯	◯	◯	◯	◯
A good track record on previous projects.	◯	◯	◯	◯	◯

13. *Write three* (3) *attributes that influence a person to produce a more accurate estimation but not yet mentioned in the previous point.*

____

Appendix 2

Survey result

ID RESPONDEN	Personal Data			Requirements		Design Implementation							Experience		Domain Knowledge		Technical Ability		Effort			
	Gender	Job Experience (years)	Project Experience (projects)	Functional Requirements Complexity	Non-functional Requirements Complexity	User Interface Complexity	Software and Hardwarel interfaces Complexity	Diversity of Technology	The Sophisticated or Novelty Technology	Coding Complexity	Database Size	Database Complexity	Job Experience	Effort Estimation Experience	Similar Project	Good Track Record	Technical Ability	Involve in Many Projects	Guess/Intuition	Experience	Track Record	Attributes
				X1	X2	X3	X4	X5	X6	X7	X8	X9	Y1	Y2	Y3	Y4	Y5	Y6	Z1	Z2	Z3	Z4
R_01	F	4	2	4	3	3	2	2	3	5	3	2	4	2	4	4	3	3	1	0	0	0
R_02	M	1	2	3	3	4	2	3	3	2	1	1	4	4	5	3	5	4	0	1	1	1
R_03	F	4	4	5	3	5	5	3	4	5	4	4	5	5	5	5	5	5	0	1	1	1
R_04	M	2	4	5	4	5	5	5	5	5	3	4	5	5	5	5	5	5	0	1	1	1
R_05	M	1	4	4	3	4	3	4	4	4	3	4	4	3	4	3	4	4	0	1	0	1
R_06	M	1	2	4	2	5	2	3	4	5	1	1	3	5	4	4	5	4	0	1	1	1
R_07	M	4	4	5	4	5	5	5	5	3	1	5	5	5	5	3	5	5	0	1	1	1
R_08	M	4	4	3	2	2	3	3	3	3	2	3	3	3	3	3	3	3	0	1	1	1
R_09	F	3	1	3	3	5	5	5	4	3	4	4	5	4	5	5	5	5	0	1	1	1
R_10	F	2	2	5	4	5	5	5	5	4	4	5	5	5	5	5	5	4	0	1	1	1
R_11	M	3	4	5	5	5	4	4	4	5	3	4	3	5	5	3	5	3	0	1	1	1

ID RESPONDEN	Personal Data			Requirements		Design Implementation							Experience		Domain Knowledge		Technical Ability		Effort			
	Gender	Job Experience (years)	Project Experience (projects)	Functional Requirements Complexity	Non-functional Requirements Complexity	User interface Complexity	Software and Hardware Interfaces Complexity	Diversity of Technology	The Sophisticated or Novelty Technology	Coding Complexity	Database Size	Database Complexity	Job Experience	Effort Estimation Experience	Similar Project	Good Track Record	Technical Ability	Involve in Many Projects	Guess/Intuition	Experience	Track Record	Attributes
				X1	X2	X3	X4	X5	X6	X7	X8	X9	Y1	Y2	Y3	Y4	Y5	Y6	Z1	Z2	Z3	Z4
R_12	M	2	4	4	2	3	4	4	3	5	2	4	3	5	4	3	4	2	1	1	0	1
R_13	M	1	2	4	3	4	4	4	3	3	3	3	5	5	5	5	5	5	0	0	0	1
R_14	M	2	3	5	4	5	5	5	5	4	3	3	4	4	4	4	5	4	0	1	1	1
R_15	F	2	4	4	4	4	3	4	3	5	4	4	5	4	4	5	5	4	0	1	1	1
R_16	M	2	4	5	3	5	5	3	3	3	3	3	4	4	4	4	5	3	0	1	1	0
R_17	M	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	1	1	1
R_18	F	4	4	5	5	5	5	4	4	4	3	4	4	5	5	4	5	4	0	1	0	1
R_19	M	1	3	4	4	4	4	3	5	3	3	4	4	4	3	4	4	3	0	1	1	1
R_20	M	2	4	5	4	5	3	4	4	5	4	4	5	5	5	4	4	5	0	1	1	1
R_21	M	2	3	4	3	4	2	3	3	4	2	3	4	3	5	4	5	5	0	1	1	1
R_22	M	3	4	5	4	4	2	5	4	5	5	5	5	5	5	5	5	5	0	1	1	1

ID RESPONDEN	Personal Data			Requirements		Design Implementation									Experience		Domain Knowledge		Technical Ability		Effort			
	Gender	Job Experience (years)	Project Experience (projects)	Functional Requirements Complexity	Non-functional Requirements Complexity	User interface Complexity	Software and Hardware Interfaces Complexity	Diversity of Technology	The Sophisticated or Novelty Technology	Coding Complexity	Database Size	Database Complexity	Job Experience	Effort Estimation Experience	Similar Project	Good Track Record	Technical Ability	Involve in Many Projects	Guess/Intuition	Experience	Track Record	Attributes
				X1	X2	X3	X4	X5	X6	X7	X8	X9	Y1	Y2	Y3	Y4	Y5	Y6	Z1	Z2	Z3	Z4
R_23	M	1	3	5	4	5	4	4	3	5	3	3	4	3	5	4	5	3	1	0	0	0
R_24	M	1	2	4	4	4	1	4	4	5	3	4	4	4	4	4	4	5	0	1	1	1
R_25	M	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	1	1	0
R_26	M	4	4	5	5	4	4	4	4	4	3	3	5	5	5	5	5	5	0	1	1	0
R_27	M	4	4	5	5	2	3	4	4	3	3	3	3	4	5	4	5	4	0	0	0	1
R_28	M	3	4	5	5	5	5	5	4	5	5	5	5	5	5	4	5	5	1	1	0	1
R_29	M	3	4	5	4	5	3	5	5	5	4	4	4	3	4	3	5	3	0	1	1	0
R_30	M	3	4	5	3	5	2	4	4	5	5	5	4	4	5	4	4	4	0	0	1	0
R_31	M	1	3	4	4	4	4	4	4	4	4	4	3	4	4	4	4	3	0	1	0	0
R_32	M	1	1	5	5	5	5	5	5	5	5	5	2	2	2	2	2	2	0	1	0	0
R_33	M	2	2	4	3	3	3	4	4	5	4	4	4	4	4	4	4	4	0	0	1	0
R_34	M	3	4	4	3	4	4	5	5	5	4	4	5	5	5	4	5	5	1	1	1	0

ID RESPONDEN	Personal Data			Requirements		Design Implementation							Experience		Domain Knowledge		Technical Ability		Effort			
	Gender	Job Experience (years)	Project Experience (projects)	Functional Requirements Complexity	Non-functional Requirements Complexity	User interface Complexity	Software and Hardware Interfaces Complexity	Diversity of Technology	The Sophisticated or Novelty Technology	Coding Complexity	Database Size	Database Complexity	Job Experience	Effort Estimation Experience	Similar Project	Good Track Record	Technical Ability	Involve in Many Projects	Guess/Intuition	Experience	Track Record	Attributes
				X1	X2	X3	X4	X5	X6	X7	X8	X9	Y1	Y2	Y3	Y4	Y5	Y6	Z1	Z2	Z3	Z4
R_35	M	1	3	3	3	4	3	5	4	5	5	5	5	5	4	4	5	4	1	0	0	0
R_36	M	2	2	5	3	3	3	5	5	5	3	5	4	4	4	5	5	4	0	0	0	1
R_37	F	2	2	5	3	5	4	5	5	3	1	3	2	3	4	2	3	2	0	0	0	1
R_38	F	2	4	5	2	4	1	4	5	4	2	1	2	3	5	5	5	3	0	0	1	0
R_39	M	2	2	4	3	4	4	5	4	4	3	5	3	3	5	3	4	4	0	0	0	1
R_40	M	2	3	5	4	5	5	4	4	3	3	3	3	5	5	3	5	4	1	1	1	1
R_41	M	1	1	5	4	5	4	4	5	5	4	4	4	4	4	4	4	3	1	0	0	0