Framework for Developing a Software Cost Estimation Model for Software Modification Based on a Relational Matrix of Project Profile and Software Cost Using an Analogy Estimation Method

Hathaichanok Suwanjang and Nakornthip Prompoon

***Abstract*—Software cost estimation is one of the most challenging works due to the fact that it requires detailed information about each stage of the project, including the project scope, project requirements, software development methodology and the required resources. Often, this information is not known until the final stage of project while the cost of project needs to be estimated at the initial stage. There are many different ways to estimate software costs and each way has its own characteristics and purpose according to each project profile. The purpose of this research is to propose a framework for developing a software cost estimation model for software modification project based on a relational matrix of a project profile and software cost using an analogy estimation method and multiple regression analysis to estimate the amount of software modification list and the amount of effort need for the new project from the data collection of the previous project.**

***Index Terms*—Software cost estimation, cost factor, relation matrix, analogy method**

I. Introduction

A software development project consists of various activities which may be different according to each project profile and environment. When existing software is modified to add new functions to fulfill new requirements, the new functions are listed, called a software modification list. Usually, the software development life cycle process for an existing software will be conducted in five stages: project initiation, requirements gathering and software configuration, software modification, system preparation and deployment. At each stage, there is software cost including effort, schedule and resources. For project management efficiency, it is necessary for the project manager to estimate software cost at the initial stage of the project. It helps formulate the project plan, manage the schedule and resources. However, according to the nature of software development, there is usually little information at the initial stage. Consequently expert judgment is often used for the software cost estimation. Lacking standards for estimation, the estimated result, which depend on an expert's knowledge, experience and opinion, is often inaccurate. It may cause deviations and problems between the project execution and the project planning. Moreover, it is a main reason for budget deficits, late submissions and project failures. At present, various cost estimation models have been proposed for developing software. Before applying a software estimation model to a particular software development project, the model customization will be needed according to each project profile.

Software cost estimation methods are divided into two major types: algorithmic methods and non-algorithmic methods. Algorithmic methods use a formula for estimating software costs for example COCOMO (COnstructive COst MOdel) and Function Points. On the other hand, non-algorithmic methods do not use a formula for estimation for example an analogy method. Analogy methods estimate by comparing the basic characteristic of the new project to previous completed one has a similar project. This method is suitable for projects that modify existing software to fulfill new requirements. For software cost estimation model construction for a software modification projects, choosing the appropriate method and factor are important for the accuracy of the software cost estimation. We have therefore studied and analyzed the software cost and project profile from each main factor including customer factors, project factors, software factors and team factors.

The purpose of this research is to propose a framework for developing a software cost estimation model by using an analogy method from the exits software that have been developed and stored in the software collection. The previous software project characteristic is analyzed to select the major factors may have influence on the cost of software development and to find a relational matrix between a project profile and its software cost, so that we are able to estimate the amount of software modification items and the amount of effort needed for the new project requirements. For our study, we have chosen a human resource management related company which has many software modification projects as our case study to illustrate the application of our proposed framework. The existing software collections are composed of a variety of modules: a core module, a workflow module, a time-management module, a leave-management module, a payroll module, an employee self service module and a report module.

The research is organized as follows: Section 2 describes the underlying concepts of software cost estimation and project cost. Section 3 discusses related researches. Section 4 explains the research methodology and how to create the framework for developing a software cost estimation model.

Manuscript received January 15, 2012; revised March 15, 2012.

The authors are with Software Engineering Laboratory, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand (Tel.: +66 2 2186996; fax: +66 2 2186955; e-mail: nakornthip.s@chula.ac.th , hathaichanok.suw@student.chula.ac.th).

Section 5 shows the result of applying framework. Section 6 discusses the limitations of this research. Finally, section 7 concludes our research and outlines the plans for further work.

II. The Underlying Concepts

A. Software Cost Estimation Method

Software cost estimation means estimating the cost or expenditure of software development. The cost includes effort, schedule and resources used for the software development. Cost estimation plays an important role in defining project scope, schedule and resources. Fig. 1. shows a software cost estimation process adapted from [1]. High level cost estimation is done at step 3 Project characteristic analysis and lower level estimation are done at steps 5-8, include estimating the effort and identifying the risks of each activity, allocating resources and reviewing and publicizing plans. Steps 5-8 are repeated until the estimation is stable or has sufficiently small deviation between iterations. Software cost estimation methods are divided into two major types[5]:

- Software cost estimation by non- algorithmic methods

Non-algorithmic methods do not use a formula to calculate the software cost estimate.

- Software cost estimation by algorithmic methods

Algorithmic methods use a formula to calculate the software cost estimate. The formula is developed from models which are created by combining related cost factors. In addition, the statistical method is used for model construction.

The comparison among cost estimation methods is presented in Table 1. in terms of strengths and weaknesses[1]-[3], [5], [10].

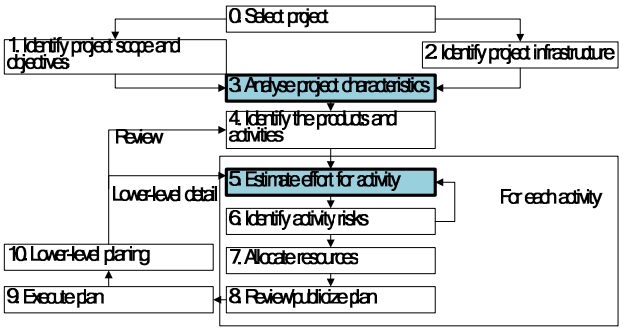

Fig. 1. Software cost estimation process [1]

TABLE 1: Strengths and Weaknesses of Software Cost Estimation Methods

Method	Details	Strengths	Weaknesses
Non-algorithmic methods			
expert judgment methods	Estimate the cost of the software from the experience of experts or experts-group to calibrate their estimates with those of other experts.	Easy way to estimate during initial stage.	Inconsistent and subjective; is not repeatable.
analogy methods	Estimate by comparing the new project to previous completed similar project.	Estimate based on historical data of similar projects and past experience. Estimation can be done during initial stage.	Lack of similar projects may render the method unusable. Historical data may not be accurate.
Algorithmic methods			
COCOMO 2.0	・Basic COCOMO is a simple model for rough estimation during the initial stage. The size of the software is calculated by "Object Points". ・Intermediate COCOMO is a model for detailed estimation after the requirements definition has been developed. The size of the software is calculated by "Function Points". ・Advanced COCOMO is an advanced model for estimation after the detailed design has been finished. The size of the software is calculated by lines of code.	Objective, repeatable and analyzable formula. Flexibility to use at each project phase. Many adjustment factors.	COCOMO is based on the size of the software but it may be uncertainty value in the early phase. The size of the software may not be an accurate measure for development using a code generator.
function points	Estimate based on a combination of program characteristics: external inputs and outputs, user interactions, external interfaces, files used by the system and then modified by complexity of the project	FPs can be used to estimate LOC depending on the average number of LOC per FP for a given programming language	FPs is very subjective depend on the estimator and cannot count FPs automatically.

B. Software Project Cost

A software development project can be broken into various stages and each stage has its own activities. The calculations for software project cost also include expenses for each activity. Software project management can be divided into four phases: project initiation, project planning, project execution and project closing. Fig. 2. shows the implementation stage of a software modification project from the company used as our case- study. The case study software collection provides software modules implemented the system functionalities in human resource management (HRM) domain. However, it must be customized (modified and adjusted) according to each customer's specific requirements. Each customization is considered as a software modification project. The cost calculation for software modification projects is based on five separated stage as follows: cost at the project initiation stage, cost at the requirements gathering and software configuration stage, cost at the software modification stage, cost at the system preparation stage, and cost at deployment stage

Software cost estimation of this research is limited to the software modification stage, because at this stage, the software development cost varies depending on the amount of modification to fulfill customer requirements. Fig. 3.

shows the project implementation plan of a software modification project from our case study company. One important characteristic of the software modification projects from this company is that each stage of each project except for the software modification stage takes approximately the same amount of time, resources and effort. The software modification stage spends quite variability of time, resources and effort for each project.

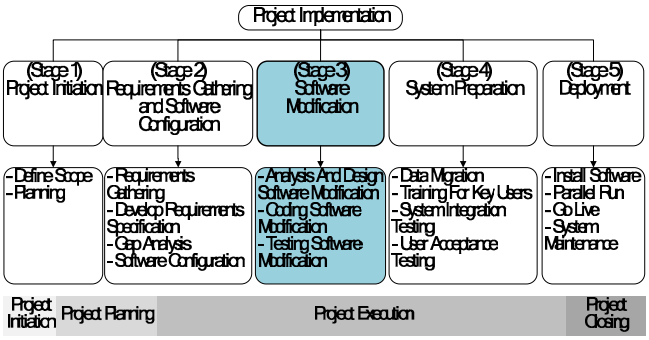

Fig. 2. Project Implementation Stages

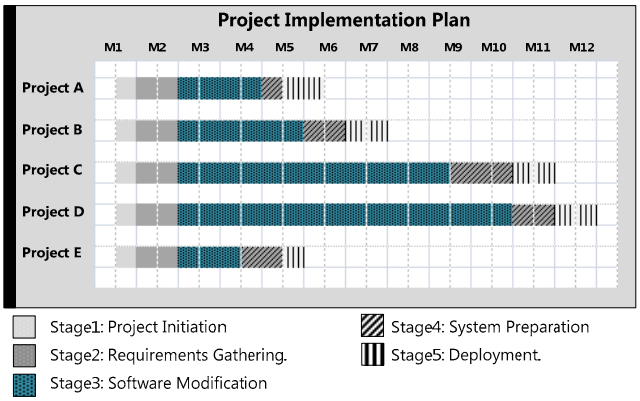

Fig. 3. Project Implementation Plan

III. Related Researches

According to our study of software cost estimation researches, we found that each research work has adopted different techniques and factors depending on the characteristics of the projects studied by that work. In the research "Estimating Software Based on Use Case Point" [6], a use case method is described. Requirements are analyzed and use cases created from those requirements. Software size is estimated by Use Case Points (UCP). Each use case is categorized and assigned a value based on point factors. The assigned value will be later adjusted basing on technical factors and environment factor. The UCP method can be used to estimate software costs after requirements have been gathered and the detailed design completed. Another research work entitled "Effort Estimation of Use Case for incremental Large Scale Software Development" [7] also used the UCP method for estimation. The companies studied in this research were in the banking business. The UCP method was adapted for incremental software development. Incremental development is the enhancement of software efficiency and upgrading software to the new version. Since the software cost estimate for each project depends on the project characteristics, the research analyzed the project components to find factors that affected software cost estimate. The research work entitled "Component-Based project Estimation Issues for Recursive Development" [8] studied and analyzed the project components by dividing each component into two categories: objective features and subjective features. Objective features of a project component consist of the development team, environment and tools. All of the components effects on software cost estimates at the analysis, design, implement and testing stages.

However, no research work was found to develop a model to estimate the cost of software development based on the modification of existing software by using an analogy method.

IV. Research Methodology

The research methodology of this research is shown in Fig. 4.

The step of our research methodology is as follows:

Step 1. Review related theories and literatures. Study methods of software cost estimation in terms of strengths and weaknesses for developing software cost estimation model.

Step 2. Study and analyze project implementation stages and project profiles which affect the cost of software development.

Step 3. Analyze software module structure and types of modifications to the existing software.

Step 4. Study statistical methods suitable for creating the software cost estimation model and identify the data needed to be collected from cost factors have influences on estimating two items:

- The amount of software modification list (ML). It is classified by the modification types.
- The amount of effort (E). It is categorized by the modification types.

Step 5. Apply statistical method to develop the software cost estimation model**.** It is composed of project profiles collecting for each factor and sub-factors data and statistical analysis. The project profile; customer factors, project factors, software factors and team factors, are collected from each previous project profile of the company used as the case study.

Step 6. Evaluate the accuracy of the software cost estimation model [4][9]. The accuracy of the cost estimation could be evaluated by calculating the Mean Relative Error (MRE) between the actual value and the estimated value as in Eq. (1) and Mean Magnitude of the Relative Error (MMRE) for each project as in Eq. (2)

$$MRE = \left| \frac{\text{Actual Value} - \text{Estimated Value}}{\text{Actual Value}} \right| \quad (1)$$

$$MMRE = \frac{1}{n} \sum_{i=1}^{n} MREi \quad (2)$$

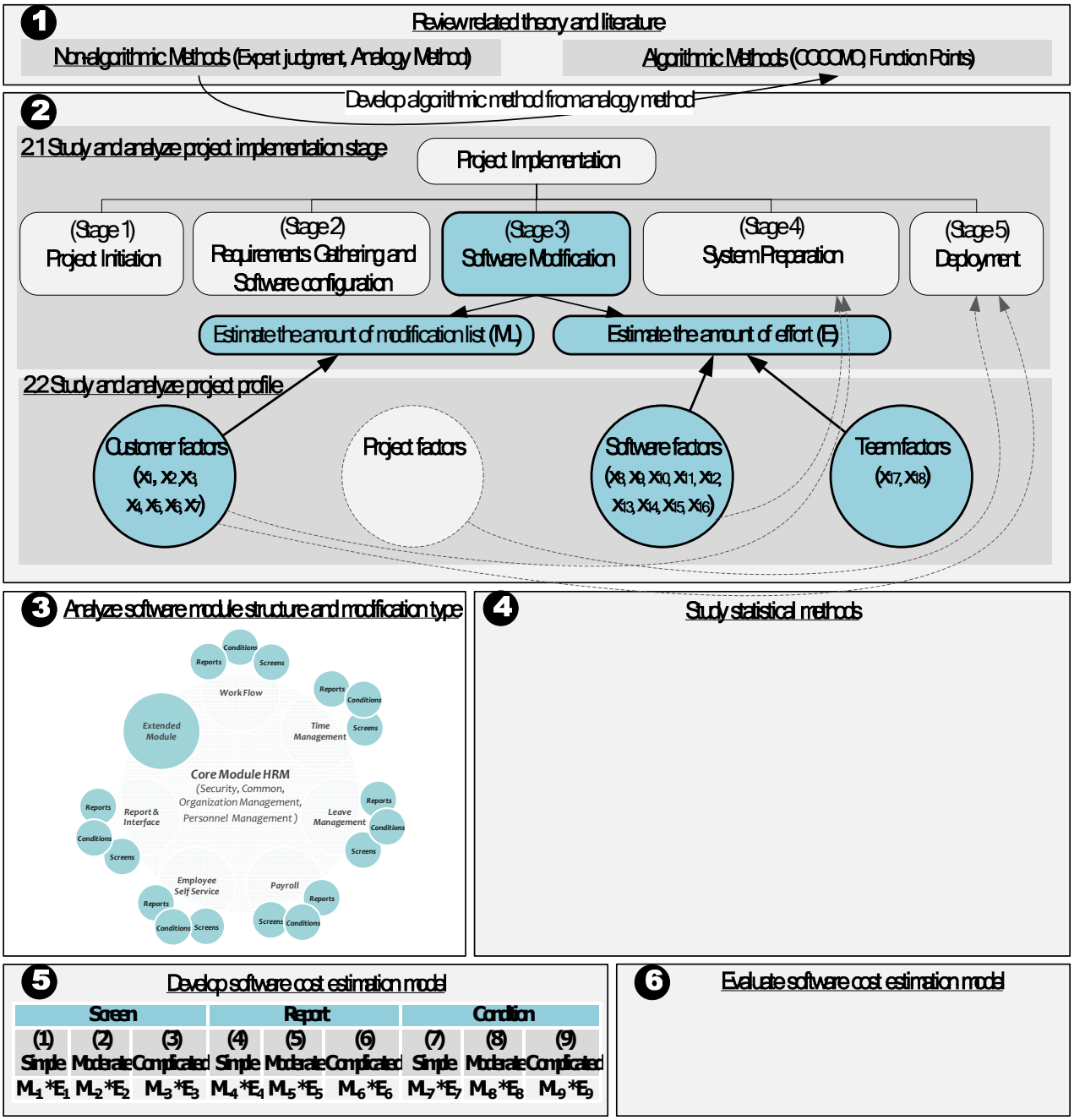

Fig. 4. Research methodology

TABLE 2: ANALYSIS OF PROJECT PROFILE CLASSIFIED INTO FACTORS AND SUB-FACTORS AND SOFTWARE COST OF EACH PROJECT IMPLEMENTATION STAGES.

Variable	Sub-factors	Sub-factors detail	Project Implementation Stages							
			Project Initiation	Req. Gathering	Software Modification				Prepare System	Deployment
					Modification List	Effort				
						Screen	Report	Condition		
Customer factors										
X_1	Number of employees	Number of current employees in customer company			✓				✓	✓
X_2	Number of policies	Number of calculation policies each of employee categories		✓	✓				✓	
X_3	Number of pay cycles	Number of pay cycles.			✓				✓	✓
X_4	Number of roles	Ex. HR Admin, Supervisor, Employee		✓	✓				✓	
X_5	Service models	Software as a Service **(1)** | Outsourcing **(0)**		✓	✓				✓	✓
X_6	Customer organization types	Customer organization types are divided into two types.			✓				✓	

Variable	Sub-factors	Sub-factors detail	Project Implementation Stages							
			Project Initiation	Req. Gathering	**Software Modification**				Prepare System	Deployment
					Modification List	**Effort**				
						Screen	**Report**	**Condition**		
		Special: Factory, Bank, Hotel **(1)** | General **(0)**								
X_7	Number of integrations	Number of integrations to third party system		✓	✓				✓	
	Number of data migrations	Number of month for backdate data migrations							✓	✓
Project factors										
	Number of go live phases	The number of iterative phase that separate to go live.								✓
	Number of parallel runs	The number of month need to be compare between existing system and new system								✓
Software factors										
X_8	Screen types	Multi-transaction per page **(1)** | One transaction per page **(0)**				✓			✓	
X_9	Screen behaviors	Need approval by workflow process **(1)** | General **(0)**				✓			✓	
X_{10}	Table accesses	Number of table accesses				✓	✓	✓	✓	
X_{11}	Number of input validations	Number of input validations exception of the common validation by centralized framework				✓	✓	✓	✓	
X_{12}	Number of buttons	Number of buttons exception of the standard button				✓			✓	
X_{13}	Number of popup screens	Number of popup screens				✓			✓	
X_{14}	Data depends on intervals	Depend on intervals **(1)** | Not depend on intervals **(0)**				✓	✓	✓	✓	
X_{15}	Integration types	Real Time **(1)** | Batch **(0)**				✓			✓	
X_{16}	Number of output fields	Number of output fields for display				✓	✓	✓		
Team factors										
X_{17}	Experience in business domain	Beginner **(1)** | Expert **(0)**				✓	✓	✓		
X_{18}	Experience in programming language and logical skill	Beginner **(1)** | Expert **(0)**				✓	✓	✓		

TABLE 3: THE RESULT FORMAT DERIVED FROM THE ESTIMATING CATEGORIZED BY MODIFICATION TYPES AND COMPLEXITY

Screen			**Report**			**Condition**		
(1) Simple	(2) Moderate	(3) Complicated	(4) Simple	(5) Moderate	(6) Complicated	(7) Simple	(8) Moderate	(9) Complicated
$ML_1 * E_1$	$ML_2 * E_2$	$ML_3 * E_3$	$ML_4 * E_4$	$ML_5 * E_5$	$ML_6 * E_6$	$ML_7 * E_7$	$ML_8 * E_8$	$ML_9 * E_9$

V. RESULTS

According to the research methodology, we found that there are four main factors of a project profile that may effect in the cost at each project stage as shown in Table 2. In the scope of our research which mainly concerns about software modification cost estimate, there are only three factors may effects to software modification stage. After analyzing the module structure and types of modifications, software modification in each module can be divided into three types: screens, conditions and reports; each type can be further classified according to complexity into three levels: simple, moderate and complicated as shown in Table 3. The results of our research are presented only the results from step 1- 4 since step 5-6 deals with the data analysis which will be covered in our future work.

After the study of statistical methods, the most suitable statistical method which is Multiple Regression Analysis is chosen to construct the cost estimation model. It is used to identify the relation between the factors and estimated items (an amount of modification list and the amount of effort).

There are five steps of applying the multiple regression analysis method to estimate the amount of software which needs to be modified (ML) and the amount of effort (E) for each modification types as follows

- **Step of choosing factors**. It is the step of choosing which factors have relation with the software modification. We found that the factors affect to the amount of modification list is $X_1, X_2, X_3, X_4, X_5, X_6, X_7$ and the factors affect to amount of effort is $X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}$. The full name of each variable is presented in Table 2.

- **Step of creating equation.**

The equation used for estimating the amount of software which needs to be modified was developed as follows.

$$ML_1 = a + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7$$

$$\vdots$$

$$ML_9 = a + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 \quad (3)$$

The equation used for estimating the amount of effort was divided into screen development effort and report and condition development effort.

Effort required for develop the screen is developed as follows.

$$E_1 = a + b_8X_8 + b_9X_9 + b_{10}X_{10} + b_{11}X_{11} + b_{12}X_{12} + b_{13}X_{13} + b_{14}X_{14} + b_{15}X_{15} + b_{16}X_{16} + b_{17}X_{17} + b_{18}X_{18}$$

$$E_3 = a + b_8X_8 + b_9X_9 + b_{10}X_{10} + b_{11}X_{11} + b_{12}X_{12} + b_{13}X_{13} + b_{14}X_{14} + b_{15}X_{15} + b_{16}X_{16} + b_{17}X_{17} + b_{18}X_{18} \quad (4)$$

Effort required for develop the report and condition are developed as follows.

$$E_4 = a + b_{10}X_{10} + b_{11}X_{11} + b_{14}X_{14} + b_{16}X_{16} + b_{17}X_{17} + b_{18}X_{18}$$

$$E_9 = a + b_{10}X_{10} + b_{11}X_{11} + b_{14}X_{14} + b_{16}X_{16} + b_{17}X_{17} + b_{18}X_{18} \quad (5)$$

The main equation used for estimating the total effort is shown as follows.

$$TE = \sum_{i=1}^{n} (ML_i \times E_i) \quad (6)$$

where TE is total effort, ML is amount of modification list, and E is amount of effort

- **Step of analysis of one-way ANOVA**. We use one-way ANOVA for checking the relation between dependent variable and all factors (independent variables) is conducted to make sure that at least 1 factor has relation with dependent variable.
- **Step of diagnosing the relation between dependent variable and each factor (independent variable)**. This is the step that diagnoses the relation between dependent variable and all factors. Consequently, only factors that affect dependent variable will be selected.
- **Step of studying the level of relation**. After getting the result from the previous step, this step is to study the degree of relation between factors and dependent variable in order to find out how much the relation influence dependent variable.

VI. Research Limitation

The research presents a framework for developing a software cost estimation model only for the case of software modifications; specifically this research only considered the software modifications done by a single company on human resource management application software. However, it is an enterprise application level, consists of one core module and six specific modules. The approximate number of modules for screen, report and condition is 400, 300, and 200 respectively. It has been used for a hundred projects. Cost estimates at the other stages of a software development project and other costs during a project are not included in the research.

VII. Conclusion and Future Work

This research proposed a framework for developing software cost estimation model for the existing software module collection from the previous project at the software modification stage. It is based on an analogy method and multiple regression analysis. A case study for the application of our proposed framework is human resource management domain. The framework can be adapted to any other project software cost estimation. In addition, it may be applied in other business domains as well. Our future work will fulfill our research goal to develop the cost estimation model and develop an automatic supporting software tool for the proposed framework and model.

References

[1] B. Hughes and M. Cotterell, *Software Project Management*. 5th Edition. McGraw-Hill, 2009.

[2] B. W. Boehm, *Software Engineering Economics*. ACM SIGSOFT Software Engineering Notes. New York: ACM, 1983, pp. 44-60.

[3] P. Panichkul and K. Pakdeewattanakul, *Software Engineering*, KTP, 2009.

[4] P. Jodpimai, P. Sophatsathit, and C. Lursinsap, "Analysis of Effort Estimation based on Software Project Models," in *Proceedings of the 9th International Symposium on Communication and Information Technologies (ISCIT 2009)*, Korea, 2009, pp. 715-720.

[5] H. Leung and Z. Fan, "Software Cost Estimation. Handbook of Software Engineering and Knowledge Engineering," World Scientific Pub. Co, River Edge. NJ, 2002.

[6] E. R. Carroll, "Estimating Software Based on Use Case Points," *OOPSLA '05 ACM SIGPLAN Object Oriented Programming Systems and Applications Conference*, USA, 2005, pp. 257-265.

[7] P. Mohagheghi, B. Anda, and R. Conradi, "Effort Estimation of Use Cases for Incremental Large-Scale Software Development," *ICSE '05 Proceedings of the 27th international conference on Software engineering*. USA. 2005, pp. 303-311.

[8] Y. Altunel and M. R Tolun, Component-Based Project Estimation Issues for Recursive Development, *Advances in Computer and Information Sciences and Engineering (2008).* Springer Netherlands. 2008, pp. 577-581.

[9] K. Vanichbuncha, *Statistics for research.* 5th Edition, Thammasan, 2010.

[10] J. E. Matson, B.E. Barrett, and J. M. Mellichamp, Software Development Cost Estimation Using Function Points, *IEEE Trans.Software Eng*., vol. 20, no. 4. 1994, pp. 275-287.