

December 1998

An Activity Based Costing Approach to Systems Development and Implementation

Ginny Ooi

Nanyang Technological University

Christina Soh

Nanyang Technological University

Pui-Mun Lee

Nanyang Technological University

Follow this and additional works at: <http://aisel.aisnet.org/icis1998>

Recommended Citation

Ooi, Ginny; Soh, Christina; and Lee, Pui-Mun, "An Activity Based Costing Approach to Systems Development and Implementation" (1998). *ICIS 1998 Proceedings*. 35.

<http://aisel.aisnet.org/icis1998/35>

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in ICIS 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

AN ACTIVITY BASED COSTING APPROACH TO SYSTEMS DEVELOPMENT AND IMPLEMENTATION

**Ginny Ooi
Christina Soh
Pui Mun Lee**

Nanyang Business School
Nanyang Technological University
Singapore

Abstract

Many organizations are in the process of replacing legacy systems with large, integrated systems using new technological platforms. Accurate estimation of project resources required and appropriate allocation of actual development and implementation costs to users are two critical issues faced. This study proposes an activity based costing (ABC) approach to estimating and recovering system development and implementation costs.

We develop the model in the context of a bank, using a two-stage approach. A prototype is first developed using two systems projects. Development of the prototype requires us to identify the resource pools and attendant costs, the major development activities, the consumption of resource costs by activities, the primary cost driver for each activity, and the driver rates for allocating activity costs to the system being implemented. The prototype model is then refined and validated using additional data collected from another 20 projects.

Keywords: IS project planning, effort estimation, time and cost estimation, chargeback.

1. INTRODUCTION

This study presents an activity based costing (ABC) approach to estimating and recovering software development and implementation costs. We address two problems currently faced by many organizations: (1) inaccurate estimation of project resources and (2) incorrect allocation of actual resource costs to projects.

Inaccurate estimation occurs because there is often no systematic organizational procedure for learning from previous estimation errors. The ABC approach enables organizations to track actual resource consumption by development activity for each project and provides a basis for variance analysis of estimated and actual costs. This facilitates learning that is specific to the organizational context.

Many organizations apply a single chargeout rate to the number of man-days consumed by the project. This does not recognize the vastly different unit costs of the different types of IT resources and penalizes the simple project that may be using relatively less skilled and hence less expensive development resources. The ABC approach explicitly recognizes different resource pools and allocates the different costs to various development and implementation activities.

In the following sections, we present our conceptual model, drawing on relevant prior research. We then describe our research methodology and the findings that we plan to present at the Conference.

2. PRIOR RESEARCH

In developing the ABC model, we looked first at different approaches to project estimation, such as function point analysis (Albrecht 1979), constructive cost model (Boehm 1981), artificial neural networks (Albus 1981), and case-based reasoning (Watson and Marir 1994). Of these, function point analysis is widely used in the industry and in the research field (e.g., Dolardo 1997; Finnie, Wittig and Desharnais 1997), we therefore compare our model primarily with function point analysis.

Function point analysis (FPA) is used to estimate the effort required in developing software systems (Albrecht 1979; Albrecht and Gaffney 1983; Banker and Kemerer 1989; Kemerer 1987), software development productivity evaluation (Behrens 1983; Gaffney 1986; Rudolph 1983), and software quality evaluation (Behrens 1983).

Although FPA is a popular estimation model for software development among practitioners and academics, it has its weaknesses. First, FPA is a generic approach to estimating software development effort across technologies, languages, and environments (Albrecht and Gaffney 1983). The complexity factors were determined by “trial and debate,” and there is no customization to cater to the unique characteristics of individual organizations. Second, the approach does not address the problem of the single charge-out rate for IS departments as it only estimates the man-days required to develop a software system. Third, FPA counts are subject to users’ interpretation and judgement as definition and counting guidelines are not strictly standardized (Low and Jeffery 1990; Kemerer and Porter 1992). Inter-rater difference in estimation is likely to be more than 30% (Rudolph 1983; Low and Jeffery 1990). Further, FPA is a “craft”-type application where skills can only be accumulated through experience, and such experience is not easily transferable across projects and estimators (Kemerer 1993). There is also lack of a software system to support the function point computation and interpretation (Kemerer 1993).

A recent study by Dolado (1997) showed that there is *no relationship* between effort and the corresponding FP. It suggests that the measure of “functionality,” while useful for estimating SLOC, cannot adequately represent the effort required for the whole set of development activities. The model developed in this paper seeks to address some of these weaknesses.

3. THEORETICAL FRAMEWORK

Activity based costing (ABC) is a non-traditional cost allocation system for allocating overhead costs introduced by Cooper and Kaplan (1988). It traces the overhead costs to the activities that consume the costs, and then to the cost objects which consume the activities. ABC has been widely used for product pricing (Brimson 1991; Kroll 1996), identification of value added activities (Reiman and Kaplan 1990), outsourcing decisions (Roodhooft and Konings 1997), and total quality management (Reiman and Kaplan 1990). ABC is widely used in manufacturing (Greeson and Kocakulah 1997; Pemberton, Arumugam and Hassan 1996) and service industries (Helmi and Hindi 1996).

The first step of the model groups the organization resources into various resource pools (see Figure 1) and determines the cost of each resource pool. Some examples of resource pools include rental expense, salaries, and depreciation. Various tasks performed in the organization are then grouped into major, functionally homogeneous activity categories, such as R&D, receiving, and quality management. Each activity consumes different amounts of one or more resource pools. For example, R&D uses 5% of rental and 40% of salaries. The costs of performing the activities are determined by allocating the costs from the resource pools to each activity according to the percentage of the resource pool consumed.

Once the activity costs are determined, they are then allocated to the cost objects, such as the product or service being produced, using cost drivers. Cost drivers measure the frequency and intensity of the demands placed on activities by cost objects. For example, the cost driver for delivery activity is the number of shipments. The organization sums up the total number of shipments made and divides the delivery activity cost by the total number of shipments to determine the rate per shipment. If a project (cost object) requires 10 shipments, it would be allocated 10 times the rate per shipment.

The major benefit of ABC is that it focuses on each activity in the software development process, (i.e., requirement analysis, detailed design, programming, testing, etc.), whereas the FPA approach estimates only the total effort. This allows management

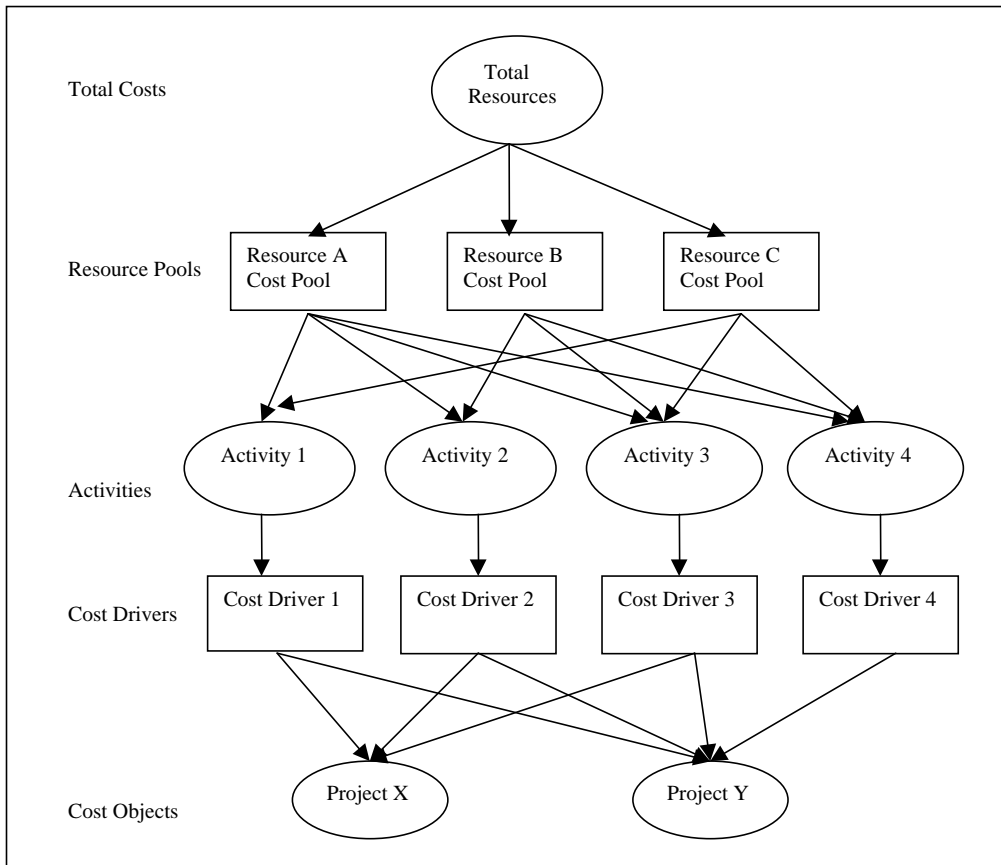


Figure 1. Generic ABC Model

to monitor and evaluate the performance by activity. Second, ABC is more than an effort estimation tool as it takes into account the different types of resources (and hence costs) consumed by each activity.

Variability in counting the number of drivers by different raters can be minimized through clear driver definition and counting procedures in the context of organization platforms. The automated estimation model using ABC software eliminates computation mistakes.

Learning and customization are incorporated into the ABC approach. First, variance analysis between estimated and actual time required for each activity serves as a learning tool for project managers as they obtain a better understanding of the sources of variance. The total variance can be parti-

tioned into three causes: (1) productivity variance, (2) changes in requirements by the users, and (3) changes in resource mix. Second, data collected for all of the projects can be stored in a database and used to further refine the ABC model over time. For example, the organization may develop different drivers and driver rates when it adopts a new technology platform.

4. RESEARCH METHODOLOGY

This is an exploratory study as we have not found any documented prior attempts to apply ABC concepts to systems development and implementation. We adapted the methodology proposed by various professional bodies such as the American Management Association, Institute of Management Accountants, and ABC Technologies, Inc.

The methodology required that we first select an organization. We chose a bank as the banking industry is IT intensive, and hence IT costs are significant, resulting in senior management interest and support for this study. The next step was to conduct interviews with the manager of an IT quality management department and to review development documentation to obtain an overview of the development process and to identify major activities and resource pools. Five resource pools and 10 activities were identified (see Figure 2). Resource cost data was obtained from the IT department's accountant.

Next, we developed an ABC prototype, which is a simple model showing how costs would be allocated to activities and from activities to two specific projects. The prototype would allow early feedback from the study's participants and enable early correction of incorrect model assumptions and parameters. To do this, we selected two projects of moderate size (approximately

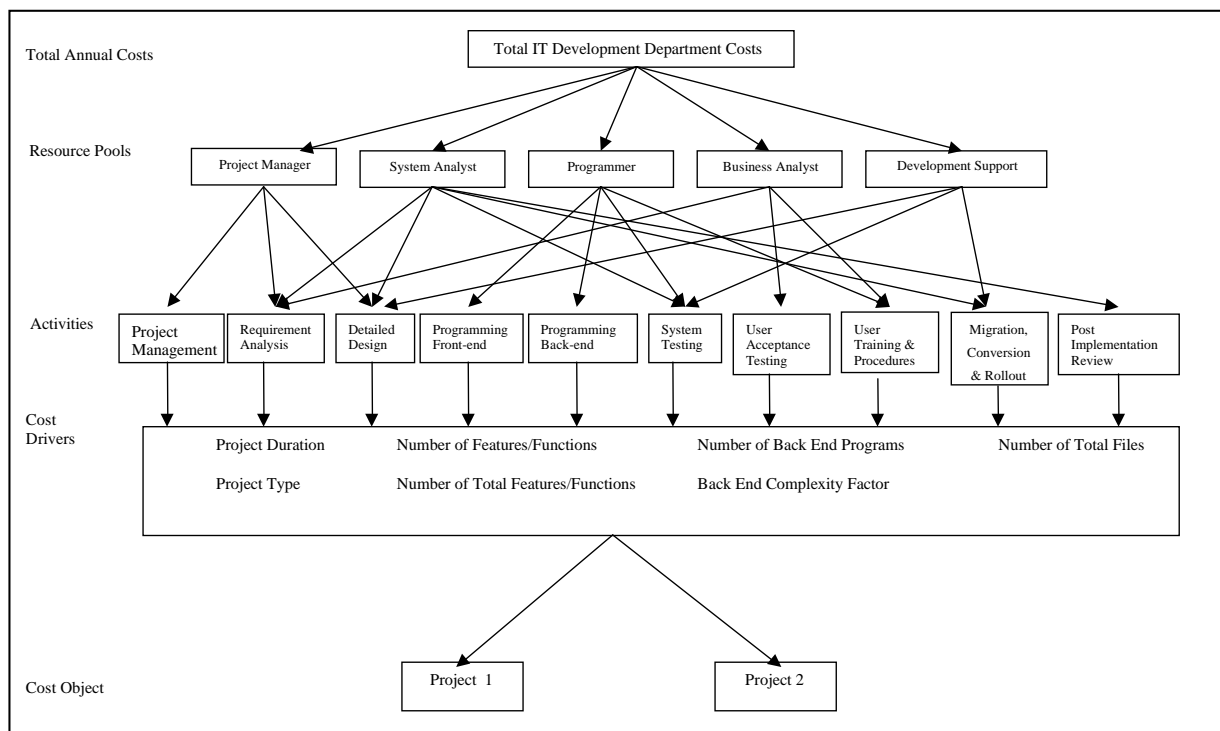


Figure 2. Prototype ABC Systems Design and Implementation Model

400 and 1,000 man-days each) using newer technologies such as data warehousing and Internet-based tools. We determined how resources were allocated to activities by analyzing the records of time spent by each member of the two project teams. This gave us initial estimates for allocating resource costs to the activities. We also interviewed all members of the project teams to determine the drivers for each activity (see Figure 2).

The next phase of the study is to refine and validate the prototype. The in-depth study of the first two projects enabled us to determine with confidence the criteria for the subsequent projects. Our criteria included project size, percentage of project completion, the type of technology platform, and adequate activity information in project work plans. In all 20 additional projects have been selected. The sample consisted of five enhancements, nine development and eight package projects. Five were host-based systems, and 17 were client-server. Projects ranged from 49 man-days to 2640 man-days, with average man-days being 651.

We interviewed the team leader of each project to validate and refine the activities and drivers. The drivers obtained from the 22 project managers were further validated by a focused group discussion with 10 assistant vice presidents and senior managers in the IT department. The drivers identified for the organization include number of features/functions, programs, files, and back end complexity and project type. The number of features/functions is similar to the “external inputs” of function point analysis, while the number of files are similar to “external interface files.” These FPA functions are also used in artificial neural networks and case-base reasoning. The back end complexity of the ABC model is similar to Intermediate COCOMO’s “product complexity.”

We also collected the time spent by each team member on each activity. For each activity, we entered the actual time as the dependent variable and all of the drivers as independent variables in a stepwise regression. This enabled us to identify the key drivers for each activity.

We are currently in the process of analyzing the data obtained from the 22 projects. The analysis will be completed shortly and we will have the validated model, with data illustrating the pattern of resource allocation to development activities, and the drivers and related driver rates that were found to significantly predict activity consumption. We will also provide a project example of using the model for estimation, costing, and variance analysis.

References

- Albrecht, A. J. "Measuring Application Development Productivity," in *GUIDE/SHARE Proceedings of the IBM Application Development Symposium*, October 1979, pp. 83-92.
- Albrecht, A. J., and Gaffney, J. E. "Software Function, Source Line of Codes, and Development Effort Prediction: A Software Science Validation," *IEEE Transaction on Software Engineering* (9:6), 1983, pp. 639-648.
- Albus, J. S. *Brain, Behavior, Robotics*, BYTE Books, Peterborough, NH, 1981.
- Banker, R. D., and Kemerer, C. F. "Scale Economies in New Software Development," *IEEE Transaction of Software Engineering* (15:10), 1989, pp. 416-429.
- Behrens, C. A. "Measuring the Productivity of Computer System Development Activities with Function Points," *IEEE Transaction of Software Engineering* (9:6), 1983, pp.648-652.
- Boehm, B. W. *Software Engineering Economics*, Prentice-Hall, Englewood Cliff, NJ, 1981.
- Brimson, J. A. *Activity Accounting: An Activity-Based Costing Approach*, John Wiley and Sons, New York, 1991.
- Cooper, R., and Kaplan, R. S. "Measure Costs Right: Make the Right Decisions," *Harvard Business Review*, September/October 1988, pp. 96-103.
- Dolado, J. J. "A Study of the Relationship among Albrecht and MK II Function Points, Lines of Codes, 4GL and Effort," *The Journal of System and Software* (37:2), 1997, pp. 161-173.
- Finnie, G. R.; Wittig, G. E.; and Desharnais, J-M. "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-based Reasoning and Regression Models," *Journal of Systems Software* (39), 1997, pp. 281-289.
- Gaffney, J. E. "The Impact on Software Development Costs of Using HOL's," *IEEE Transaction of Software Engineering* (12:3), March 1986.
- Greeson, C. B., and Kocakulah, M. C. "Implementing an ABC Pilot at Whirlpool," *Journal of Cost Management*, March/April, 1997, pp. 16-21.
- Helmi, M. A., and Hindi, N. "Activity Based Costing in Banking: A Big Challenge," *The Journal of Bank Cost and Management Accounting* (9:2), 1996, pp. 5-19.
- Kemerer, C. F. "An Empirical Validation of Software Cost Estimation Models," *Communications of the ACM* (30:5), May 1987, pp. 416-429.
- Kemerer, C. F. "Reliability of Function Points Measurement—A Field Experiment," *Communications of the ACM* (36:2), February 1993, pp. 85-97.
- Kemerer C. F., and Porter, B. S. "Improving the Reliability of Function Point Measurement: An Empirical Study," *IEEE Transactions on Software Engineering* (18:11), 1992, pp.1011-1024.
- Kroll, K. M. "The ABCs Revisited," *Industry Week* (254:22), December 1996, pp. 19-21.
- Low, G. C., and Jeffery, D. R. "Function Points in Estimation and Evaluation of the Software Process," *IEEE Transaction on Software Engineering* (16:1), 1990, pp. 64-71.
- Pemberton, N. L.; Arumugam, L.; and Hassan, N. "ABM at Dayton Technologies—From Obstacles to Opportunities," *Management Accounting*, March 1996, pp. 20-27.
- Reimain, B. C., and Kaplan, R. S. "The ABCs of Accounting for Value Creation," *Planning Review* (18:4), 1990, pp. 33-34.
- Roodhooft, F., and Konings, J. "Vendor Selection and Evaluation - An Activity Based Costing Approach," *European Journal of Operational Research*, January 1997, pp. 97-102.
- Rudolph, E. E. "Productivity in Computer Application Development," Working Paper 9, University of Auckland, Department of Management Studies, Auckland, New Zealand, 1983.
- Watson, I., and Marir, F. "Case-based Reasoning: A Review," *The Knowledge Engineering Review* (9:4), 1994, pp. 327-354.