Functional Decomposition Based Effort Estimation Model for Software-Intensive Systems

Nermin Sökmen

***Abstract*—**An effort estimation model is needed for software-intensive projects that consist of hardware, embedded software or some combination of the two, as well as high level software solutions. This paper first focuses on functional decomposition techniques to measure functional complexity of a computer system and investigates its impact on system development effort. Later, it examines effects of technical difficulty and design team capability factors in order to construct the best effort estimation model. With using traditional regression analysis technique, the study develops a system development effort estimation model which takes functional complexity, technical difficulty and design team capability factors as input parameters. Finally, the assumptions of the model are tested.

***Keywords*—**Functional complexity, functional decomposition, development effort, technical difficulty, design team capability, regression analysis.

I. Introduction

SOFTWARE-INTENSIVE system projects face great challenges when they attempt to measure complexity of system design to estimate design effort. The literature studies have shown that effort estimations in software-intensive projects are made through the software size. A software-intensive system is a computer-based system which is ranging over software applications, information systems, embedded systems, and systems-of-systems [1]. Although software plays a critical role in the development of a system, it is important to mention that a software-intensive system requires hardware not only to run on but also perform specific tasks. Therefore, the hardware part of the whole system should be taken into consideration when making estimates for project effort.

This paper introduces a different approach to estimating system development effort in software-intensive projects. The aim of this paper is twofold: to define a system design complexity metric and to construct a parametric effort estimation model for embedded and real time systems. The remainder of the paper is structured as follows. The paper first examines software and hardware size and complexity metrics and effort estimation models in the literature. It then describes the research method used in the construction of a system effort estimation model. The next section presents the analysis results and the constructed model. Finally, the paper ends with conclusion.

Dr. Nermin Sökmen is chief senior researcher at Informatics and Information Security Research Center (BİLGEM) of Scientific and Technological Research Council of Turkey (TÜBİTAK), PK 74 Gebze Kocaeli, 41470, Turkey (phone: +90-262-765-3109; fax: +90-262-648-1100; e-mail: nermin.sokmen@tubitak.gov.tr).

II. Literature Research

A number of software size and effort metrics have been identified in the literature. Putman's SLIM (Software Life Cycle Management) model incorporating software size and development time parameters computes software effort estimation based on the Rayleigh function [2]. Albrecht [3] first introduces function points methodology to calculate software size. Albrecht and Gaffney [4] then show the relationship between function points and development effort. Kemerer [5] evaluates four cost estimation models (SLIM, COCOMO, Function Points and ESTIMACS) with using a data set that covers 15 large completed data-processing projects. Matson et al. [6] develop effort estimation equations with using function points data taking from 104 projects. Zheng et al. [7] propose a linear equation for software effort estimation based on Albrecht's function point. The System Evaluation and Estimation of Resources-Software Estimation Model (SEER-SEM) estimates development effort as a function of three parameters: effective software size, effective technology and staffing complexity [8].

Boehm [9] introduces the first COCOMO model for software development effort estimation. The model estimates effort based on size of software and pre-determined constants. Boehm's the intermediate COCOMO model computes software development effort as a function of estimated software size and a set of cost drivers that consists of product, hardware and personnel characteristics [10]. The formula uses different sets of coefficients when calculating program effort for organic, semi-detached and embedded software projects.

Further, Nassif et al. [11] present a log-linear regression model based on the use case point model (UCP) to calculate the software effort based on use case diagrams. Sharma and Kushwaha [12] propose a measure for the estimation of software development effort on the basis of requirement based complexity.

The literature rarely addresses the problem of modeling hardware design complexity. Salchak and Chawla [13] propose a hardware design complexity measure for avionics systems. The measure has been derived from an avionics software design complexity measure constructed from six components, namely reuse, internal cohesion, external cohesion, interface complexity, data coupling and real-time coupling [14]. Even in a different domain, Bashir and Thomson [15] first propose a product complexity measure, and then develop number of parametric models to estimate design effort. Number of parametric models was developed to estimate design effort with using product complexity metric. Bashir and Thomson [16] develop an analogy-based model for hardware design effort estimation. Using traditional regression analysis, Bashir and Thomson [17] construct two types of parametric models: a single variable model based on product complexity, and a multivariable model based on product complexity and severity of requirements. Finally, Bashir and Thomson [18] use product complexity, difficulty to expertise ratio, type of drawings submitted to the customer and involvement of design partners as input parameters and develop a parametric model to estimate the effort needed to execute designs for hydro-electric generators.

III. Data Set and Methodology

Historical data from completed 13 software-intensive projects were obtained from a research institute and an Information Technology (IT) company. System development effort, the dependent variable, was calculated with using the sum of hardware and embedded software development efforts spent in all phases of product development lifecycle, including requirement analysis, design, implementation and test.

Traditional regression analysis was used to develop a system development effort estimation model for embedded and real time projects. This study focuses on three factors: functional complexity, technical difficulty and design team capability.

A. Functional Complexity (FC)

Product complexity has the most significant impact on development time [19], [20] and effort [16]-[18]. Grifinn [19] defines complexity as the number of functions of a product. Hobday [21] emphasizes that the quantity of components and sub-systems, the hierarchical manner in which they are integrated together and the degree of technological novelty are the important indicators of product complexity. El-Haik and Yang [22] identify three components of design complexity: coupling, variability, and correlation.

A hierarchical structure is needed for managing complexity [23]. Bashir and Thomson [15] define hardware product complexity as a function of the number of functions and the depth of their functional trees. They propose the formula in (1):

$$\text{Product Complexity (PC)} = \sum_{j=1}^{l} F_j x\, j \qquad (1)$$

where F_j is the number of functions at level j and l is the number of levels.

Hardware aspect of a system consisting of electronics sub-systems and components can be self-contained or embedded. In this study, system is defined as a hardware system alone or together with its embedded software. This paper focuses on Bashir and Thomson's product complexity measure to calculate functional complexity of an electronic system. Fig. 1 shows the functional tree of an embedded system with its corresponding product complexity.

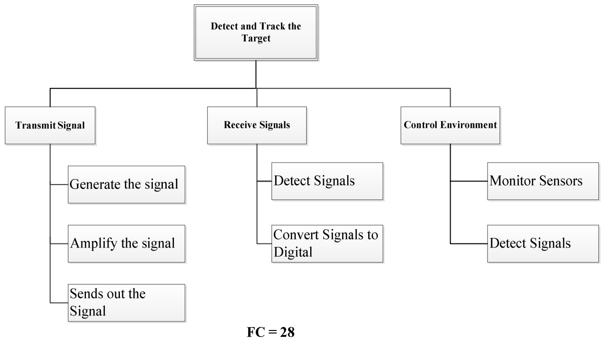


Fig. 1 The functional tree of an embedded system with its complexity

B. Technical Difficulty (TD)

Technical complexity, technical difficulty and technological newness are analyzed in various studies [24]-[27]. Technologically easy solutions can be used in the design of very complex products and radical technology changes can be required by less complicated products [28]. In addition to product complexity, this study focuses on technical difficulty. Griffin [28] identifies technical difficulty as the difficulty of developing the scientific solution to the problem. Technical difficulty indicates the degree of difficulty of technical goals and product specifications in a project [29]. In this study, technical difficulty is measured on seven-point scale that ranges from 1 (implemented existing/reuse technologies) to 7 (designed and implemented very complex and emerging technologies).

C. Design Team Capability (DTC)

Lack of required knowledge and skills in the project personnel was identified as one of the important risk items in software development projects [30]-[33]. This study also considers impact of design team capability that consists of knowledge, skill and experience variables. Similar to technical difficulty factor, it is measured on seven-point scale that ranges from 1 (knowledge, skill and experience do not exist) to 7 (highly qualified experienced team composition).

IV. System Development Effort (SDE) Estimation Model

A. Data Analysis Results

The descriptive statistics show that system development effort and functional complexity variables are normally distributed. Mean, median, minimum and maximum values of functional complexity variable are 67.3, 54, 19 and 153, respectively.

Table I presents correlations among the basic and derived variables. The relationship between functional complexity and system development effort was supported at 0.01 level with a coefficient of 0.826. On the other hand, the regression model constructed with this variable explains 71 percent of the total variation in the system development effort variable. Durbin-Watson (DW) was found as 1.71. Since the DW value is less than 2.0, there may be some indication of serial correlation.

TABLE I
DESCRIPTIVE ANALYSIS AND CORRELATIONS

Variables	Mean	Pearson Correlations		
		FC	TD/DTC	MPC
SDE (Person-month)[a]	39.9	0.826[b]	0.268[c]	0.957[b]
FC[a]	67.3	[a] Normally distributed		
TD/DTC	1.07	[b] $p<0.01$		
MPC (Modified PC)	66.3	[c] $p>0.05$		

The Pearson correlation test results showed that technical difficulty, design team capability and technical difficulties to design team capability variables were not associated with system development effort.

Further, to make the development effort estimation more precise and accurate, it is necessary to consider functional complexity factor together with other factors. After several tries, the relationship between development effort and functional complexity is increased to 0.957 with the help of formula in (2).

$$\text{MPC (Modified PC)} = FC\ x\ (\frac{TD}{DTC})^{0.5} \quad (2)$$

where MPC is modified product complexity, FC is functional complexity of embedded or real time systems, TD is the degree of technical difficulty and DTC is the degree of design team capability.

Technical difficulty to team expertise ratio was also used by Bashir and Thompson [18]. On the other hand, their final effort equation is quite different than the model constructed in their study.

B. Model Generation

Linear regression analysis was used to develop a model for estimating system development effort from the degree of technical difficulty of the system and the degree of design team capability. The results of the regression analysis are shown in Table II.

TABLE II
MODEL COEFFICIENTS

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
MPC	.589	.025	.989	23.141	.000

The regression coefficient was found to be statistically significant. The generated system development effort estimation model is given in (3).

$$\text{SDE} = 0.589x\ FC\ x\ (\frac{TD}{DTC})^{0.5} \quad (3)$$

C. Model Verification

The mean magnitude of relative error (MMRE) and prediction quality indicator (Pred(m)) are the two most important indicators used in the performance assessment of software effort estimation models [34], [35]. This study used MMRE and Pred(0.25) indicators to test the accuracy of the regression model. MMRE formula is given in (4).

$$\text{MMRE} = \frac{1}{n} \times \sum_{i=1}^{n} \text{MRE}_\text{i} \quad (4)$$

where MRE_i is the difference between the actual and the estimated effort relative to the actual effort, n is the number of systems in the dataset. MRE_i is given in (5).

$$\text{MRE}_\text{i} = \frac{|SDE_i - \widehat{SDE}_i|}{SDE_i} \quad (5)$$

where $\widehat{SDE}_i$ is the predicted effort of system i and SDE_i is the actual effort of system i.

Table III shows the actual efforts, the estimated efforts and MRE values calculated for each system in the dataset. The table also gives the MMRE and Pred(0.25) values. MMRE should be equal to 0.25 or less [16], [17], [35]. The computed MMRE for the dataset is 0.157. Since MMRE is less than 0.25, the model is considered to be acceptable.

TABLE III
ESTIMATED ACCURACY TABLE

SDE (person-month)	$\widehat{SDE}$ (person-month)	MRE_i
24.0	21.6	0.10
30.0	27.4	0.09
39.0	35.6	0.09
32.0	28.3	0.12
9.0	9.6	0.06
22.0	26.5	0.20
82.0	79.1	0.03
87.0	79.6	0.08
24.5	32.5	**0.33**
40.5	44.2	0.09
25.0	15.8	**0.37**
42.0	31.2	**0.26**
62.0	76.2	0.23
MMRE: 0.157		
Pred.(0:25): 0.77		

Pred(0.25) is a measure of the percentage of observations whose MRE is less than or equal to 0.25. Pred(0.25) is given in (6).

$$\text{Pred.}(0.25) = \frac{k}{n} \quad (6)$$

where k is the number of observations whose MRE is less than or equal to 0.25, n is the total number of systems.

The model is considered to be acceptable if (Pred(0.25)) ≥ 0.75 [17], [35]. Pred(0.25) is 0.77. The model can be acceptable.

The study also verifies the regression assumptions. Table IV gives ANOVA test results. The F test in the ANOVA table implies that the model can fit for predicting system development effort estimation (Sig. < 0.01).

Table V gives the model summary. As is shown in Table V, the results of the regression analysis indicate that modified product complexity variable is significantly related to system development effort. The regression model explains 97.6 percent of the total variation in system development effort estimation. Since the DW statistic is close to 2.0, there is no autocorrelation problem.

TABLE IV
ANOVA TABLE

Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	26752.987	1	26752.987	535.495	.000
Residual	599.513	12	49.959		
Total	27352.500d	13			

TABLE V
MODEL SUMMARY

R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
.989	.978	.976	7.06820	1.925

The plot of residuals versus the predicted values is shown in Fig. 2. The residuals fall within a generally random pattern.

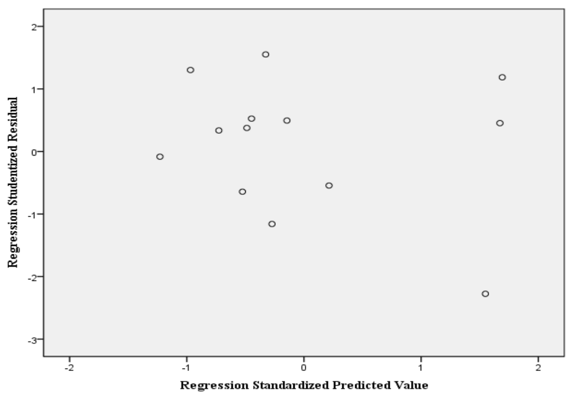
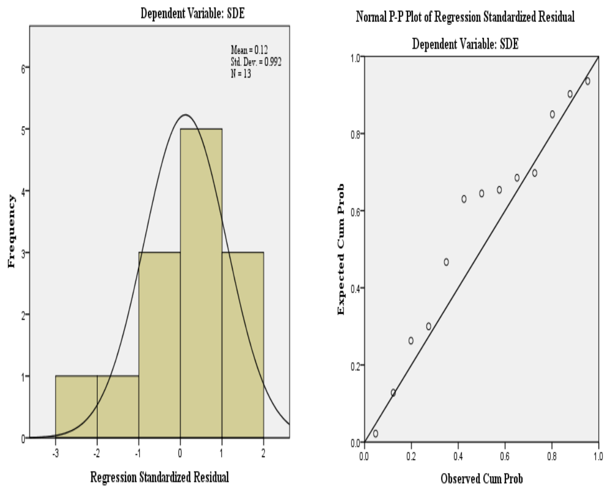


Fig. 2 Analysis of Residuals

The patterns shown in Fig. 3 indicate that the residuals are normally distributed.



(a) (b)

Fig. 3 The residuals diagrams: (a) The histogram of the residuals (b) Normal P-P plot

V. CONCLUSION

Effort estimations in software-intensive projects are mostly made through the software size. In the literature, there are limited number of studies that address hardware complexity and effort estimation. On the other hand, an effort estimation model is needed for software-intensive systems that consist of hardware and embedded software parts.

This study focused on embedded and real time systems. It first investigated a suitable indicator to measure functional complexity of a computer system. Due to its systematic approach and its language-independence, the functional decomposition technique was selected.

The study then examined the relationships among system development effort, functional complexity, technical difficulty and design team capability. Test results showed the strong relation between development effort and functional complexity.

Finally, the paper constructed a parametric model to estimate the development effort for software-intensive projects. The constructed regression model takes functional complexity, technical difficulty and design team capability factors as input variable. Model verification results show that the constructed model meets all regression assumptions and the criteria of MMRE and Pred(0.25) even though sample size is small.

REFERENCES

[1] M. W. Maier, D. Emery, and R. Hillard, "ANSI/IEEE and Systems Engineering," *System Engineering*, vol. 7, no. 3.pp. 257-270, 2004.

[2] L. H. Putman, " General empirical solution to the macro software sizing and estimating problem," *IEEE Transactions on Software Engineering*, vol. SE 4, no. 4, pp. 345-361, July 1978.

[3] A. J. Albrecht, "Measuring Application Development Productivity," *in Proceedings of the IBM Applications Development Symposium*, GUIDE/SHARE (Goiterey, Calic,'Oct. 14-17), IBM, 1979,' pp. 83-92.

[4] A. J Albrecht and J. E Gaffiney, "Software functions, source lines of code and development effort prediction: A software science validation," *IEEE Transactions on Software Engineering*, vol. SE-9, no.6, pp. 639-648, November 1983.

[5] C.F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," *Communication of the ACM*, vol. 5, no. 5, pp. 416-429, May 1987.

[6] J. E. Matson, E. Barrett, and J. M. Mellichamp, "Software Development Cost Estimation Using Function Points," *IEEE Transactions on Software Engineering*, vol. 20, no. 4, pp. 275-287, April 1994.

[7] Y. Zheng, B. Wang, Y. Zheng, and L. Shi, "Estimation of software projects effort based on function point," *ICCSE'09 4th International Conference on Computer Science & Education*, Nanning, China, 2009, pp. 941-943.

[8] L. Fischman, K. McRitchie, and D. D. Galorath, "Inside SEER-SEM," *CrossTalk-The Journal of Defense Software Engineering*, pp. 26-28, April 2005.

[9] B.W. Boehm, "*Software Engineering Economics*," Englewood Cliffs, NJ: Prentice Hall, 1981.

[10] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece, "*Software Cost Estimation with COCOMO II*," Englewood Cliffs, NJ: Prentice-Hall, 2000.

[11] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *The Journal of Systems and Software*, vol. 86, pp. 144– 160, 2013.

[12] A. Sharma, and D. S. Kushwaha, "Estimation of Software Development Effort from Requirements Based Complexity," *in 2nd Int. Conf. on Computer, Communication, Control and Information Technology (C3IT-2012), Procedia Technology*, vol. 4, pp. 716 – 722, February 2012.

[13] P. W. Salchak and P. Chawla, "Supporting Hardware Trade Analysis and Cost Estimation Using Design Complexity," *in Proceedings of the 1997 VHDL International User's Forum (VIUF'97)*, pp. 126, Washington, DC, USA, 1997.
[14] K. Littlejohn, M. Olis, R. Lentz, and M. Barnett, "Developing a design complexity measure, " *in 15th AIAA/IEEE Digital Avionics Systems Conference*, Atlanta, GA, USA, 27-31 Oct. 1996, pp. 31-36.
[15] H. A. Bashir and V.Thomson, "Estimating Design Complexity," *Journal of Engineering Design*, vol. 10, no. 3, pp. 247-257, 1999.
[16] H. A. Bashir and V.Thomson, "An Analogy Based Model for Estimating Design Effort," *Design Studies*, vol. 22, no. 2, pp. 157–167, March 2001.
[17] H. A. Bashir and V.Thomson, "Models for estimating design effort and time," *Design Studies*, vol. 22, no. 2, pp. 141-156, March 2001.
[18] H. A. Bashir and V.Thomson, "Estimating design effort for GE hydro projects," *Computers & Industrial Engineering*, vol. 46, no. 2, pp. 195–204, 2004.
[19] A. Griffin, "Modeling and measuring product development cycle time across industries," *J. Eng. Technol. Manage*, vol. 14, no. 1, pp. 1-24. 1997.
[20] H. A. Bashir, "Modeling of development time for hydroelectric generators using factor and multiple regression analyses," International Journal of Project Management, vol. 26, pp. 457–464, 2008.
[21] M. Hobday, "Product complexity, innovation and industrial organization," *Research Policy*, vol. 26, pp. 689-710, 1998.
[22] B. El-Haik and K. Yang, "Measures of complexity in design," *in Proceedings of the 1999 ASME Design Engineering Technical Conference*, September 12-15, Las Vegas, Nevada, 1999.
[23] M. Keating, "Measuring Design Quality by Measuring Design Complexity, " *in Proceedings of IEEE 2000 First International Symposium on Quality Electronic Design*, San Jose, CA, USA, 20-22 March 2000, pp. 103-108.
[24] M. H. Meyer and J. M. Utterback, "Product Development Cycle Time and Commercial Success," *IEEE Transactions on Engineering Management*, vol. 42, no. 4, pp. 297-304, 1995.
[25] M. V. Tatikonda and S. R. Rosenthal, " Technology Novelty, Project Complexity, and Product Development Project Execution Success: A Deeper Look at Task Uncertainty in Product Innovation," *IEEE Transaction on Engineering Management*, vol. 47, no. 1, pp. 74-87, 2000.
[26] M. Swink, " Completing projects on-time: how project acceleration affects new product development," *J. Engineering and Technology Management*, vol. 20, no. 4, pp. 319–344, 2003.
[27] L. Wallace and M. Keil, "Software Project Risks and Their Effect on Outcomes," *Communications of the ACM*, vol. 47, no. 4, pp. 68-73, April 2004.
[28] A. Griffin, " Metrics for Measuring Product Development Cycle Time," *Journal of Product Innovation Management*, vol. 10, no. 2, pp. 112-125, March 1993.
[29] R. E. Purser, W. A. Pasmore, and R. V. Tenkasi, "The influence of deliberations on learning in new product development teams," *Journal of Engineering and Technology Management,* vol. 9, pp. 1-28, 1992.
[30] T. Moynihan. "How experienced project managers assess risk," *IEEE Software*, vol. 14, no. 3, pp. 35-41, May-June 1997.
[31] M. Keil, P. E. Cule, K. Lyytinen, and R. C. Schmidt, "A Framework for Identifying Software Project Risks," *Communications of the ACM*, vol. 41, no. 11, pp. 76-83 November 1998.
[32] P. Cule, R. Schmidt, K. Lyytinen, K. Mark, " Strategies for Heading Off is Project Failure," *Information Systems Management*, vol. 17, no. 2, pp. 1-9, 2000.
[33] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: an international Delphi study," *Journal of Management Information Systems,* vol. 17, no. 4, pp. 5–36, 2001.
[34] B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, and M. J. Shepperd,"What accuracy statistics really measure," in *IEE Proceedings of Software*, vol. 148, no. 3, pp. 81-85, Jun 2001.
[35] M. O. Elish, "Improved estimation of software project effort using multiple additive regression trees," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10774–10778 , September 2009.