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 (BILGEM) 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


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.


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
Product complexity has the most significant impact on development time [19], [20] and effort [16]-[18]. Griffin [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):

$$\text{Product Complexity (PC)} = \sum_{j=1}^{l} F_j x_j$$  \hspace{1cm} (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.

**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)**

Functional complexity is the most significant impact on development time [19], [20] and effort [16]-[18]. Griffin [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):

$$\text{Product Complexity (PC)} = \sum_{j=1}^{l} F_j x_j$$  \hspace{1cm} (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.

**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

<table>
<thead>
<tr>
<th>Variables</th>
<th>Mean Value</th>
<th>Pearson Correlations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FC</td>
</tr>
<tr>
<td>SDE (Person-month)**</td>
<td>39.9</td>
<td>0.826</td>
</tr>
<tr>
<td>FC**</td>
<td>67.3</td>
<td>Normally distributed</td>
</tr>
<tr>
<td>TD/DTC</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>MPC (Modified PC)</td>
<td>66.3</td>
<td></td>
</tr>
</tbody>
</table>

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 trials, the relationship between development effort and functional complexity is increased to 0.957 with the help of formula in (2).

\[
MPC = FC \times \left(\frac{TD}{DTC}\right)^{0.5}
\]

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>
<thead>
<tr>
<th>Variables</th>
<th>Unstandardized Coefficients</th>
<th>Standardized Coefficients</th>
<th>t</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>Std. Error</td>
<td>Beta</td>
<td></td>
</tr>
<tr>
<td>MPC</td>
<td>589</td>
<td>0.25</td>
<td>0.989</td>
<td>23.141</td>
</tr>
</tbody>
</table>

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

\[
SDE = 0.589 \times FC \times \left(\frac{TD}{DTC}\right)^{0.5}
\]

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).

\[
MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i
\]

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).

\[
MRE_i = \frac{|SDE_i - \overline{SDE}_i|}{SDE_i}
\]

where \(\overline{SDE}_i\) 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 \(MRE_i\) 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>
<thead>
<tr>
<th>SDE (person-month)</th>
<th>SDE (person-month)</th>
<th>MRE_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.0</td>
<td>21.6</td>
<td>0.10</td>
</tr>
<tr>
<td>30.0</td>
<td>27.4</td>
<td>0.09</td>
</tr>
<tr>
<td>39.0</td>
<td>35.6</td>
<td>0.09</td>
</tr>
<tr>
<td>32.0</td>
<td>28.3</td>
<td>0.12</td>
</tr>
<tr>
<td>9.0</td>
<td>9.6</td>
<td>0.06</td>
</tr>
<tr>
<td>22.0</td>
<td>26.5</td>
<td>0.20</td>
</tr>
<tr>
<td>82.0</td>
<td>79.1</td>
<td>0.03</td>
</tr>
<tr>
<td>87.0</td>
<td>79.6</td>
<td>0.08</td>
</tr>
<tr>
<td>24.5</td>
<td>32.5</td>
<td>0.33</td>
</tr>
<tr>
<td>40.5</td>
<td>44.2</td>
<td>0.09</td>
</tr>
<tr>
<td>25.0</td>
<td>15.8</td>
<td>0.37</td>
</tr>
<tr>
<td>42.0</td>
<td>31.2</td>
<td>0.26</td>
</tr>
<tr>
<td>62.0</td>
<td>76.2</td>
<td>0.23</td>
</tr>
</tbody>
</table>

MMRE: 0.157
Pred(0.25): 0.77

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

\[
Pred. (0.25) = \frac{k}{n}
\]

where \(k\) is the number of observations whose MMRE is less than or equal to 0.25, \(n\) is the total number of systems.

The model is considered to be acceptable if \((Pred(0.25)) \geq 0.75\) [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

<table>
<thead>
<tr>
<th>Model</th>
<th>Sum of Squares</th>
<th>df</th>
<th>Mean Square</th>
<th>F</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>26752.987</td>
<td>1</td>
<td>26752.987</td>
<td>535.495</td>
<td>.000</td>
</tr>
<tr>
<td>Residual</td>
<td>599.513</td>
<td>12</td>
<td>49.959</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>27352.500</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TABLE V

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>R Square</th>
<th>Adjusted R Square</th>
<th>Std. Error of the Estimate</th>
<th>Durbin-Watson</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.989</td>
<td>.978</td>
<td>.976</td>
<td>7.06820</td>
<td>1.925</td>
</tr>
</tbody>
</table>

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](image)

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

![Fig. 3 The residuals diagrams](image)

V. CONCLUSION

Effort estimations in software-intensive projects are mostly made through the software size. In the literature, there are limited numbers 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


