Software Development Effort Estimation using Fuzzy Logic Framework- An Implementation

1Rshma Chawla, 2Deepak Ahlawat
1,2MMICTBM, MMU Mullana
Email: 1rshma.chawla@mmumullana.org, 2deepakahlawat1983@gmail.com

Abstract: Software development effort estimation is very important and critical task when we have to develop new or reusable software. The main objective of this research paper is to provide a technique for software cost estimation technique that performs better than other techniques on the accuracy of effort estimation. This study proposed to extend the Constructive Cost Model (COCOMO) by incorporating the concept of fuzziness into the measurements of size, mode of development for projects and the cost drivers contributing to the overall development effort. This study explores four fuzzy logic membership functions: Fuzzy Triangular Membership Function, Gbell Membership Function, Gauss2 Membership Function and Trapezoidal Membership Function for Software development effort estimation. Implementation is done by taking estimated effort of Intermediate COCOMO.

I. INTRODUCTION

1.1. Software Development

Software development is the computer programming, documenting, testing, and bug fixing involved in creating and maintaining applications and frameworks involved in a software release life cycle and resulting in a software product [32].

Various Phases in Software Development

a. Requirements specification.
b. Planning.
c. Design.
d. Construction.
e. Testing.
f. Debugging.
g. Deployment.
h. Maintenance.

1.2. Software Effort Estimation

Software effort estimation is the prediction of the likely amount of cost, time, and staffing levels required to build a software system at an early stage during a project. It is difficult to obtain the effort estimate during the preliminary stages, because of the limited resources available at that time [2].

Need for effort estimation

• It would facilitate increased control of time and overall cost benefit in software development life cycle.

• Software development effort estimates are the basis for project bidding and planning.

• Software effort estimation has even been identified as one of the three most demanding challenges in software application areas. During the development process, the cost and time estimates are useful for the initial rough validation and monitoring of the project’s completion process. And in addition, these estimates may be useful for project productivity assessment phases.

1.3. Size Estimation

The estimation of size is very critical and difficult area of project planning. It has been recognized a crucial step from the very beginning. Many Effort estimation models take size as basic unit for calculating effort.

Methods of Size Estimation:

a. Lines of Code: It is software metric used to measure the size of a software program by counting the number of lines in the text of the program’s source code. A line of code is any line of program text that is not a comment or a blank line, regardless of the number of statements or fragment of statements on the line. This specially includes all lines containing program header, declarations, and executable and non-executable statements. Unit of measurement is KLOC/ KSLOC/ KDLOC (1000 lines of code).

There are several cost, schedule, and effort estimation models which use LOC as an input parameter, including the widely-used Constructive Cost Model (COCOMO) invented by Barry Boehm [2].

It can be calculated by:

a1. Define the statement of scope.

a2. Historical data. •

a3. Prepare some rough coding(it may be misleading but have to be done when there is no previous experience).
a4. Automated tools are available.
b. Function Points: A function point is a unit of measurement to express the amount of business functionality an information system (as a product) provides to a user. Function points measure software size. The cost of a single unit is calculated from past projects [1].

It can be calculated as:
b1. Internal Logical files (ILF).
b2. External Interface files (EIF).
b3. External Input (EI).
b4. External Output (EO).
b5. External Inquiry (EQ).

II. COCOMO MODEL

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. COCOMO was first published in Boehm’s 1981 book Software Engineering Economics as a model for estimating effort, cost, and schedule for software projects [2]. COCOMO is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity. This model estimates the total effort in terms of “person – months” of the technical project staff. COCOMO is a hierarchy of software cost estimation models, which include basic, intermediate and detailed sub models.

Boehm proposed 3 modes of projects:
1. Organic mode – simple projects that engage small teams working in known and stable environments.
2. Semi-detached mode– projects that engage teams with a mixture of experience. It is in between organic and embedded modes.
3. Embedded mode– complex projects that are developed under tight constraints with changing requirements.

BASIC MODEL:

Equation for effort calculation

\[ DE = a \times (SIZE)^{b} \]

Where, DE is development effort.

Constants a and b are dependent upon the mode of the project.

INTERMEDIATE MODEL:

\[ Ei = a \times (SIZE)^{b} \]

Where, Ei = Estimated effort

Table 2.1 gives the value of a and b for different modes [29].

Table 2.1: Intermediate COCOMO nominal effort estimating equations

<table>
<thead>
<tr>
<th>Development Mode</th>
<th>Nominal Effort Equation ( E_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organic</td>
<td>((MM)_{NOM} = 3.2(KDLOC)^{1.05})</td>
</tr>
<tr>
<td>Semi-Detached</td>
<td>((MM)_{NOM} = 3.0(KDLOC)^{1.12})</td>
</tr>
<tr>
<td>Embedded</td>
<td>((MM)_{NOM} = 2.8(KDLOC)^{1.20})</td>
</tr>
</tbody>
</table>

Development Effort = EAF * Ei

EAF is effort adjustment factor, which can be calculated from 15 different factors or cost drivers.

III. FUZZY LOGIC

Four basic concepts [17], [18], [21]:
a) Fuzzy Sets – sets with smooth boundaries. “Jane is old” [0,1]

b) Linguistic variables – consider the sentence “The amount of trading is heavy” uses a fuzzy set “Heavy” to describe the quantity of the stock market trading in one day. More formally, this can be expressed as Trading Quantity is Heavy.

Here Trading Quantity is linguistic variable.
c) Possibility distribution – constraints on the value of a linguistic variable imposed by assigning it a fuzzy set.

Age (suspect) = [20, 30].

d) Fuzzy if-then rules – a knowledge representation scheme for describing a functional mapping or a logic formula that generalizes an implication in two-valued logic.

Figure 3.1: Fuzzy System

3.1. Fuzzy Inference Process

i) Fuzzification:

Membership Functions

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept.

ii) Logical Operators and if-then Rules:

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic. A single fuzzy if-then rule assumes the form

if x is A then y is B

Where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule “x is A” is called the antecedent or premise, while the then-part of the rule “y is B” is called the consequent or conclusion.

iii) Defuzzification: We can implement two types of fuzzy inference systems in the toolbox: Mamdani-type and Sugeno-type.

Mamdani’s fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani-type inference, as defined for the toolbox, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. Mamdani method finds the centroid of a two-dimensional function [25], [26], [27].

We use Mamdani-Type inference system.

IV. FUZZY LOGIC FOR SOFTWARE DEVELOPMENT EFFORT ESTIMATION

Available data in the initial stages of project is often incomplete, inconsistent, uncertain and unclear [8], [34]. A fuzzy model is more apt when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague [35], [36], [37]. The major difference between our work and previous works is that four fuzzy logic functions will be used for software development effort estimation on COCOMO model and then it’s validated with gathered data. The advantages of fuzzy logic are combined and learning ability and good generalization are obtained. The main benefit of this approach is it has good interpretability by using the fuzzy rules. The effort predicted using four fuzzy logic functions will be compared with Intermediate COCOMO.

V. DEVELOPMENT TOOLS AND TECHNIQUES

MATLAB 7.5

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

• Math and computation
• Algorithm development
• Data acquisition
• Modeling, simulation, and prototyping
• Data analysis, exploration, and visualization
• Scientific and engineering graphics
• Application development, including graphical user interface building.

The Matlab System:

→ Development Environment.
→ The MATLAB Mathematical Function Library.
→ The MATLAB Language.
→ Graphics.
→ The MATLAB Application Program Interface (API).

Fuzzy Logic Toolbox

Figure 5.1: GUI Tools of Fuzzy LogicToolbox
VI. IMPLEMENTATION

1. This research will implement Constructive Cost Model (COCOMO) using Mamdani Fuzzy Inference System (FIS).

2. Fuzzy Triangular membership (trimf) function, Trapezoidal membership function (trapmf), generalized bell (Gaussian Bell) membership function (gbellmf) and two-sided gaussian curve built in membership function (gauss2mf) are used with NASA63 dataset for software effort estimation.

3. The results were analyzed using the criterion – MMRE (Mean Magnitude of Relative Error).

4. Less MMRE means the result is more accurate [5, 6].

6.1. Research Methodology Used

1. Select a particular type of Fuzzy Inference System (Mamdani).

2. Define the input variables and output variable.

3. Set the type of the membership functions for input variables.

4. Set the type of the membership function for output variable.

5. Write the rules in Rule Editor.

6. The data is now translated into a set of if–then rules written in Rule editor.

7. A certain model structure is created, and parameters of input and output variables can be tuned to get the desired output.


- If MODE is Organic and SIZE is S1 then EFFORT is EF1
- If MODE is Semidetached and SIZE is S1 then EFFORT is EF2
- If MODE is Embedded and SIZE is S1 then EFFORT is EF3
- If MODE is Organic and SIZE is S2 then EFFORT is EF4
- If MODE is Semidetached and SIZE is S2 then EFFORT is EF5
- If MODE is Embedded and SIZE is S3 then EFFORT is EF5
- If MODE is Organic and SIZE is S3 then EFFORT is EF4
- If MODE is Embedded and SIZE is S4 then EFFORT is EF3
- If MODE is Organic and SIZE is S4 then EFFORT is EF4
- If MODE is Embedded and SIZE is S5 then EFFORT is EF6

6.3. Fuzzy Framework Used

Figure 6.1: Fuzzy Framework

- If MODE is Organic and SIZE is S4 then EFFORT is EF4

6.4. Experimental Results

1. Trapezoidal Membership Function

   Figure 6.2: Output 1

   Calculated Fuzzy Effort for Trap MF = Fuzzy Nominal Effort * EAF
   
   \[
   \text{Calculated Fuzzy Effort for Trap MF} = 768 \times 2.72 = 2088.96 = 2089
   \]

2. Triangular Membership Function

   Figure 6.3: Output 2

   Calculated Fuzzy effort for TMF = 2.72 * 763 = 2075

3. GBell Membership Function

   Figure 6.4: Output 3

   Calculated Fuzzy Effort for Gbell MF = 715 \times 2.72 = 1945

4. Gauss2 Membership Function

   Figure 6.5: Output 4

   Calculated Fuzzy Effort for Gauss2 MF = 718 \times 2.72 = 1956
Table 6.1: Developmental Effort Estimate using various Techniques

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2 E</td>
<td>113</td>
<td>2.72</td>
<td>2040</td>
<td>814.4</td>
<td>2215.24</td>
<td>2089</td>
<td>2075</td>
<td>1945</td>
<td>1956</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1.2 E</td>
<td>18</td>
<td>2.38</td>
<td>321</td>
<td>89.9</td>
<td>213.83</td>
<td>201.6</td>
<td>200.4</td>
<td>187.8</td>
<td>188.73</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>1.2 E</td>
<td>24</td>
<td>0.85</td>
<td>79</td>
<td>127</td>
<td>107.85</td>
<td>102</td>
<td>101.5</td>
<td>94.35</td>
<td>95.2</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>1.2 E</td>
<td>966</td>
<td>0.73</td>
<td>6600</td>
<td>10694</td>
<td>7806.45</td>
<td>7373</td>
<td>7329.2</td>
<td>6854.7</td>
<td>6891.2</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>1.2 E</td>
<td>109</td>
<td>0.94</td>
<td>724</td>
<td>780</td>
<td>733.16</td>
<td>690.9</td>
<td>687.14</td>
<td>643.9</td>
<td>646.72</td>
</tr>
<tr>
<td>6</td>
<td>43</td>
<td>1.05 O</td>
<td>28</td>
<td>0.96</td>
<td>83</td>
<td>105.8</td>
<td>101.61</td>
<td>95.71</td>
<td>95.14</td>
<td>89.2</td>
<td>89.6</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>1.12 SD</td>
<td>91</td>
<td>0.36</td>
<td>156</td>
<td>469</td>
<td>168.87</td>
<td>159.12</td>
<td>158.04</td>
<td>148.32</td>
<td>149</td>
</tr>
<tr>
<td>8</td>
<td>63</td>
<td>1.2 E</td>
<td>10</td>
<td>0.39</td>
<td>15</td>
<td>44.36</td>
<td>17.3</td>
<td>16.3</td>
<td>16.22</td>
<td>15.17</td>
<td>15.25</td>
</tr>
</tbody>
</table>

Table 6.1 shows Estimated Effort for fuzzy logic functions, four membership functions namely Trapezoidal Membership Function (Trapmf), Triangular Membership Function (TMF), Guass Bell Membership Function (GBellMF), Gauss2mf - Gaussian curve built-in Membership Function. These are obtained in Fuzzy logic Toolbox by selecting the appropriate MF. Inputs to the FIS are MODE and SIZE. Range of MODE should be selected among 1.05, 1.12, 1.2. Range of SIZE can be noted from examining the project id. Fuzzy rules are applied and then select the Nominal Effort of COCOMO as the range of Fuzzy Nominal Effort.

Estimated Fuzzy Effort = Fuzzy Nominal Effort * EAF.

6.5. Comparison

The main parameter for the evaluation of cost estimation models is the Magnitude of Relative Error (MRE) which is defined as follows:

\[
\text{MRE} = \frac{\text{Actual Effort} - \text{Estimated Effort}}{\text{Actual Effort}} \times 100 \quad \text{(1)}
\]

The MRE value is calculated for each observation i whose effort is predicted.

Table 6.2: Comparison of MRE

<table>
<thead>
<tr>
<th>S. No.</th>
<th>MRE using Inter. COCOMO Model</th>
<th>MRE using TRAP MF</th>
<th>MRE using TMF</th>
<th>MRE using GBell MF</th>
<th>MRE using Gauss2 MF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0859</td>
<td>0.0240</td>
<td>0.0171</td>
<td>0.0465</td>
<td>0.0411</td>
</tr>
<tr>
<td>2</td>
<td>0.3339</td>
<td>0.3719</td>
<td>0.3757</td>
<td>0.4149</td>
<td>0.4120</td>
</tr>
<tr>
<td>3</td>
<td>0.3651</td>
<td>0.2911</td>
<td>0.2803</td>
<td>0.1943</td>
<td>0.2050</td>
</tr>
<tr>
<td>4</td>
<td>0.1828</td>
<td>0.1171</td>
<td>0.1105</td>
<td>0.0385</td>
<td>0.0386</td>
</tr>
<tr>
<td>5</td>
<td>0.0126</td>
<td>0.0457</td>
<td>0.0509</td>
<td>0.1106</td>
<td>0.1067</td>
</tr>
<tr>
<td>6</td>
<td>0.2242</td>
<td>0.1531</td>
<td>0.1462</td>
<td>0.0746</td>
<td>0.0795</td>
</tr>
<tr>
<td>7</td>
<td>0.0825</td>
<td>0.0200</td>
<td>0.0130</td>
<td>0.0492</td>
<td>0.0448</td>
</tr>
<tr>
<td>8</td>
<td>0.1533</td>
<td>0.0866</td>
<td>0.0813</td>
<td>0.0113</td>
<td>0.0166</td>
</tr>
<tr>
<td>Σ</td>
<td>1.4403</td>
<td>1.0638</td>
<td>1.0750</td>
<td>0.9399</td>
<td>0.9444</td>
</tr>
</tbody>
</table>

Table 6.3: Computation of MMRE

<table>
<thead>
<tr>
<th>Estimation Technique</th>
<th>MMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate COCOMO</td>
<td>0.1800</td>
</tr>
<tr>
<td>FL Trapezoidal MF</td>
<td>0.1329</td>
</tr>
<tr>
<td>FL Triangular MF</td>
<td>0.1344</td>
</tr>
<tr>
<td>FL GBell MF</td>
<td>0.1175</td>
</tr>
<tr>
<td>FL Gauss2 MF</td>
<td>0.1180</td>
</tr>
</tbody>
</table>

Figure 6.6: Comparison of MMRE

![Comparison of MMRE](image)

VII. CONCLUSION

We can conclude that the introduction of fuzzy logic in the process of Software Development Effort Estimation yields more accurate results than the previous Empirical Model Approach. Trapezoidal Membership function gives more accurate estimate of Effort as compare to Triangular Membership Function. Gbell MF gives more accurate result than Gauss2 MF. Gaussian Curve MF’s are more accurate than the simple straight line Membership functions. Between five techniques that we used in our research Gbell MF has the lowest value of MMRE. So, it has highest accuracy.


VIII. FUTURE WORK

The above research work can be easily employed in the software industries. By applying some more effort there is the possibility to develop other customized membership functions, which represents inputs more closely to tolerate uncertainty and imprecision in inputs, and hence restricting the same to be propagated to the outputs. Today various softwares are available in the market to make use of COCOMO model; the above research can be deployed on COCOMO II, with experts on the subject providing required information for developing fuzzy sets and appropriate rules. This work can be extended by integrating with neural networks to take the advantage of its features, such as learning ability and good interpretability. Thus, a promising line of future work is to extend to the neuro-fuzzy approach that allows the integration of numerical data and expert knowledge.

REFERENCES


