

# DETERMINATION OF PROJECT VARIABLES USING FUZZY DECISION TREE FOR EFFORT ESTIMATIONS

<sup>1</sup>DR G LALITHA KUMARI <sup>2</sup>MRS. Y SUREKHA <sup>3</sup>DR M SITARAM <sup>4</sup>MR. N RAMESH BABU  
<sup>5</sup>DR K KOTESWARA RAO\*

<sup>1</sup> Sr. Assistant Professor, <sup>24</sup>Assistant Professor <sup>53</sup>Associate Professor,

<sup>125</sup>Department of CSE, Prasad V Potluri Siddhartha Institute Of Technology Andhra Pradesh, INDIA

<sup>3</sup>Department of CSE, Lakireddy Bali Reddy College Of Engineering Mylavaram Andhra Pradesh, INDIA

Department of ECE, RGKUT, Srikakulam, Andhra Pradesh, INDIA

E-mail: lalithajoy.nuthakki@gmail.com , yalamanchili.surekha@gmail.com, ramcs8010@gmail.com  
koteswara2003@yahoo.co.in , ramesh.nuthakki@gmail.com,

## ABSTRACT

The success of a project depends on accurate effort estimations, managers are always under pressure to prepare accurate effort estimations, in COCOMO model to estimate the effort it requires project parameters. Identification of the type of the project and choosing the project parameters are very important aspect, accurate project parameters generation and estimations are coming from mature organizations others owing to lack of history databases. Estimations are based on lines of code (size of the project), functionality of the project. If the project estimations are based on size of the project the Constructive Cost Model plays vital role. This work explains an expert system that integrates conventional and Soft Computing techniques for dealing uncertainty i.e. virtual nodes generated in the decision paths at leaf node level and we will create an additional node, the main objective of the paper is how to generate suitable values for these nodes. For this purpose we propose the method to generate project parameters using fuzzy logic. In this work solid line indicates the project already done; dotted line indicates the project with uncertainties. After the project parameters are generated using Fuzzy Logic then effort estimations can be prepared.

**Keywords:** Fuzzy Systems, Estimations, Virtual node, Expert systems.

## 1. INTRODUCTION

Project is a collection of interrelated activities that are executed in a certain order. Software engineering is the establishment and use of the sound engineering principles in order to achieve the software that is reliable and economically efficient. Software engineering is a framework that encompasses process, set of methods and array of tools. Project Management is the discipline of defining and achieving targets while optimizing (or just allocating) the use of resources (time, money, people, materials, energy, space, provisions, communication, etc.) over the course of a project (a set of activities of finite duration)[5,7]

### 1.1. Project Management Activities:

Project management is composed of several different types of activities such as[14,15,16]

1. Planning the work
2. Assessing and mitigating risk

3. Estimating resources
4. Allocating of resources
5. Organizing the work
6. Acquiring human and material resources
7. Assigning tasks
8. Directing activities
9. Controlling project execution
10. Reporting progress
11. Analyzing the results based in the facts achieved
12. Analysis & Design

#### 1.1.1. Project control variables:

The three important project control variables are Duration (Time), Cost (Budget), Man Power (people) indirect control variables are Quality, Scope and Risk of the project. [5, 7]

## 1.2 Conventional Software Project

### Management:

Conventional software management practice is mostly sound in theory, but practice is still tied to

archaic technology and techniques. Conventional software management performance [5, 7]

1. Finding and fixing a software problem after delivery costs 100 times more than finding and fixing the problem in early design phases.
2. You can compress software development schedules 25% of nominal, but no more.
3. For every \$1 you spend on development, you will spend \$2 on maintenance.
4. Software maintenance and development costs are primarily a function of the number of source lines of code.
5. Variation among people account for biggest differences in software productivity.
6. The overall ratio of software to hardware is still growing. Only about 15% of software development effort is devoted to programming.
7. Software system and products typically cost 3 times as much per SLOC as individual software programs.
8. Walkthroughs catch 60% of the errors.
9. 80% of the contribution comes from 20% of the contributions[5,7]

### 1.3 Modern Software Project Management:

Although the current software management principles are improved on conventional techniques, they still do not emphasize the modern principles on which this project is based. Building on Davis's format, there are top ten principles of modern software management[5,7,14,15,16]

1. Architecture first approach
2. Iterative life cycle process
3. Component based development
4. Change management environment
5. Round trip engineering
6. Model based notation
7. Objective Quality control
8. Demonstration based approach
9. Evolving levels of details
10. Configurable process

## 2. PROPOSED SYSTEM:

This paper presents an expert system for estimations that combines conventional methods and also soft computing techniques for handling imprecision and uncertainty. [1,2,3,4,6]

### 2.1. Soft Computing Techniques

Soft computing refers to a collection of new computational techniques in AI, machine learning, computer science and may be applied in engineering areas where one tries to study, model and analyze very complex phenomena with low cost and complete solution. Sometimes soft

computing is consortium of new computational techniques; the components are Soft Computing Techniques are[8,9,13,14] [17,18]

1. Fuzzy logic
2. Neural networks
3. Genetic algorithms
4. Adaptive Resonance Theory
5. Associative Memory.

**2.1.1. Fuzzy logic:** Logic is the science of the reasoning, symbolic or mathematical logic turnout to be a powerful computational paradigm. Mathematical sets have been classified into crisp sets and fuzzy sets, logic can also be broadly viewed as crisp logic and fuzzy logic. It (Fuzzy Logic) explains how the Human Brain is going to categorize the objects. This Part is entirely deals with fuzzy logic; this part is entirely deals with fuzzy logic. FL is a logic used to represent the Fuzzy sets to respective logical values of any hedge. A Fuzzy set  $a$  can be defined by its membership  $M_a(x)$ . [8,9][17,18]

To represent FL are use a list of pairs each pair represents value and Fuzzy membership.

$$\text{Ex : } a = \{ (x_1, M_a(x_1)) \dots (x_n, M_a(x_n)) \} \text{-----}(1)$$

Example representing the heights of three members, Joseph, John, James by using Fuzzy logic. A concept such as height which can have values from a range of Fuzzy values including "tall", "medium", "short". Joseph is of 7 feet, John is of 4 feet, and James is of 5.10 feet.

The definition of height of James falls under category of "Tall" for some and "Medium" for some people. By using Fuzzy James can be put in the list of "tall" people by giving rate of "tall" So the Fuzzy values can be defined as  $\{1,0,0.5\}$  for Joseph, John, James. [13,14][17,18]

### 2.1.2 Why we chosen fuzzy logic

1. Because of rule base operation any reasonable number of inputs can be processed and numerous Outputs can be generated.
2. FL is an inherently robust and don't require Precise.
3. FL can control nonlinear systems that would be difficult are impossible to model mathematical.
4. FL can give accurate result.
5. Since FL control processes user defined rules governing the target control system. It may be modified easily to improve performance. [8,9,13,14][17,18]

## 2.2. Fuzzy Expert System

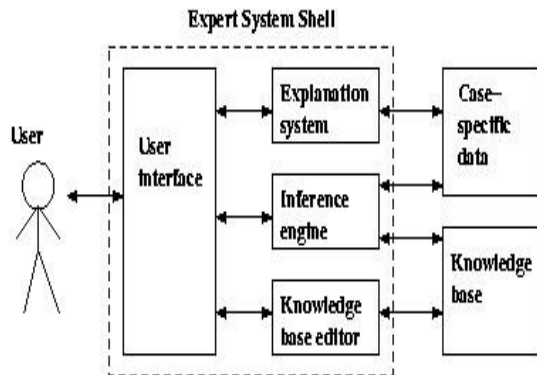


Figure 2.1: Fuzzy Expert System

Fuzzy logic is a form of logic that applies to fuzzy variables. Fuzzy logic is non-monotonic, in the sense that if a new fuzzy fact is added to a database, this fact may contradict conclusions that were previously derived from the database.

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. Expert systems are meant to solve real problems which normally would require specialized human expert. (Doctor or Mineralogist) Building an expert system requires extracting knowledge from expert (human)[17,18]

**Knowledge Engineer:** - Knowledge Engineer can extract knowledge and can build expert system knowledge database.

**Rule and Expert System:** Expert system is based on IF - THEN rule work. Rule based systems can be 1. Goal driven using back ward chaining, 2. Data driven using forward chaining. Expert system can use both. The user interacts with the system through UI which may use menus, Natural language or other style of information. Inference engine used to reason with expert knowledge and data specific of problem. Case specific data includes both data provided by the user and partial conclusions, explanation system which allows the program to explain its reasoning to the user. Knowledge database editor helps their expert or knowledge engineer. [5, 7, 8, 9, 13, 14]

### 2.2.1. Steps in Fuzzy Expert System

The important steps in the development of the Fuzzy Expert System are

1. Obtain information from one or more experts
2. Define the Fuzzy sets
3. Defining the Fuzzy rules.

In the other way fuzzy systems development can be represented in 7 steps as

1. Identify the problem
2. Define the input and output variables
3. Define the membership functions
4. Define Fuzzy Rules
5. Select inference or composition methods
6. Select Defuzzification method
7. Validate the system. [17, 18]

### 2.2.2. Directed steps to use Fuzzy Expert System

The steps to use the Fuzzy Expert System are

1. Relate the observations to the Fuzzy sets
2. Evaluate case for all Fuzzy rules
3. Combine information from the rules.
4. Deffuzify the results. [8, 9]

## 3. DECISION TREE

In decision theory, a **decision tree** is a graph of decisions and their possible consequences, (including resource costs and risks) used to create a plan to reach a goal. Decision trees are constructed in order to help with making decisions. A decision tree is a special form of tree structure.

**Decision tree** is a predictive model; that is, a mapping of observations about an item to conclusions about the item's target value. Each interior node corresponds to a variable; an arc to a child represents a possible value of that variable.

A leaf represents the predicted value of target variable given the values of the variables represented by the path from the root [8, 9].

The machine learning technique for inducing a decision tree from data is called **decision tree learning**, or (colloquially) **decision trees**. Decision tree learning is also a common method used in data mining. Here, a decision tree describes a tree structure wherein leaves represent classifications and branches represent conjunctions of features that lead to those classifications [1, 6]. A decision tree can be learned by splitting the source set into subsets based on an attribute value test [1,9]. This process is repeated on each derived subset in a recursive manner. The recursion is completed when splitting is either non-feasible, or a singular classification can be applied to each element of the derived subset. A random forest classifier uses a number of decision trees, in order to improve the classification rate. Decision trees are also a descriptive means for calculating conditional probabilities.

Decision tree can be described also as the synergy of mathematical and computing techniques that aids on the description, categorization and generalization of a given set of data. Data comes in records of the form:

$$(x, y) = (x_1, x_2, x_3, \dots, x_k, y) \dots \dots \dots (2)$$

The dependent variable, Y, is the variable that we are trying to understand, classify or generalize. The other variables  $x_1, x_2, x_3$  etc. are the variables that will help us on that job

Decision tree has three other names like **Classification tree** analysis is a term used when the predicted outcome is the class to which the data belongs. **Regression tree** analysis is a term used when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital). **CART** analysis is a term used to refer to both of the above procedures. The name CART is an acronym from the words Classification and Regression Trees, and was first introduced by Breiman et al. [BFOS84].[3,4,6]

**3.1. Decision Tree Advantages:**  
Amongst other data mining methods, decision trees are the method that has several advantages:  
Decision trees:

- **Are simple to understand and interpret.** People are able to understand decision tree models after a brief explanation.
- **Require little data preparation.** Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- **Able to handle both nominal and categorical data.** Other techniques are usually specialized in analyzing datasets that have only one type of variable. Ex: relation rules can be only used with nominal variables while neural networks can be used only with numerical variables.
- **Use a white box model.** If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic. An example of a black box model is an artificial neural network since the explanation for the results is excessively complex to be comprehended.
- **Make it possible to validate a model using statistical tests.** That makes it possible to account for the reliability of the model.
- **Are robust, perform well with large data in a short time.** Large amounts of

data can be analyzed using personal computers in a time short enough to enable stakeholders to take decisions based on its analysis.[3,4,6]

#### 4. FUZZY BASED SOLUTION FOR IMPRECISION AND UNCERTAINTY

This paper presents an expert system for estimations that combines conventional methods and also soft computing techniques for handling imprecision and uncertainty. This chapter deals with fuzzy expert system designing steps, and the directed steps to use fuzzy expert system, decision tree traversal mechanism, and determination of constants of effort estimation function. After the determination of constants **a** and **b**, using COCOMO estimations for project can be completed. The in-detailed procedure can be represented as [17,18]

##### 4.1. Defining Fuzzy Set

Having obtained the suitable information from the experts, the fuzzy set is defined.

TMC- Total minimum cost.

MP- Man Power.

LOC-Lines of code

As well as defining the linguistic variables, there is need to give each one a range of possible values. Linguistic variables of TMC, MP, and LOC are high (h), low (l).

To represent the fuzzy membership functions, the notation used is  $W_{AB}$  where A is the variable (tmc, mp, loc) and B is the variable (h, l). For example,  $W_{tmc}$  represents the function of total minimum cost low. The membership functions we use here give the weights for each of the fuzzy sets. The representation of weights may be of the form  $W_{ab}$  – where 'a' gives the fuzzy set and 'b' gives the linguistic variable of the fuzzy set.

##### 4.2. Defining fuzzy rules:

The second step in creating a fuzzy expert system is to define a set of fuzzy rules. These rules, unlike those used by traditional expert systems, are expressed in vague English terms and do not define cut-off or thresholds, but rather use subjective terms such as "high" and "low". This map more naturally to the way an expert would express his or her knowledge and makes the process of converting that knowledge into rules far simpler and less prone to error.

We relate the observations to the fuzzy sets. We will examine three sets of data, for the three

PERTs. The obtained related values are defuzzified by using defuzzication.

**4.3. Defuzzication:** We now need to Defuzzify the outputs to obtain a crisp QOA and QOB to get the required result. The crisp values are obtained by calculating the Centroid.

#### 4.4. Decision Tree Traversal Mechanism

The below figure represents the Decision tree representation that encompasses root node, leaf node, virtual nodes ,path with the solid line and dotted line.

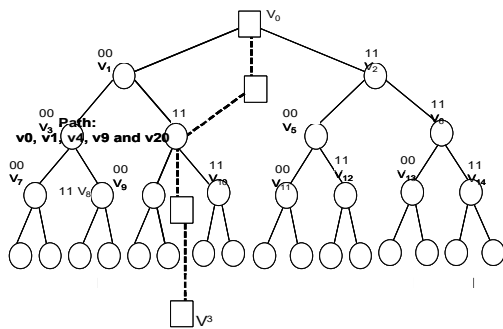


Figure 4.1: Decision tree and paths

Solid Line: No Uncertainty

Dotted Line: Uncertainty Path

Path: V0, V1, V4, V2 and V3

The decision nodes in a decision tree can be labeled using attributes of their children. The label refers to the complete list of attributes their child nodes that help us in making a decision regarding which node to be included in the decision path in next step. In figure we have shown each node having two attributes **00 & 11** which can be used to order the child nodes of a decision node from left to right. The decision tree contains 16 goal nodes and 15 decision nodes. We have also shown the decision path comprising of nodes v0, v1, v4 and v9. The corresponding path identifier is **00110011** the path is shown as a solid line. A matured organization will have access to history database that would indicate Governing Function based on Loc, and FP metrics for effort estimations.

Given X , we access the look up table (Table & Fig) to ascertain corresponding a & b. For example pid of 00110011 produces a=.9589 and b= 1.36, the constants required for determining effort for a particular company, for projects with no uncertainty. Sometimes it may not be possible for the user to select all the attributes that determine a

child node, thus giving rise to virtual nodes (shown with dotted line). [17,18]

#### 4.5. Fuzzy Threshold Mechanism And Methodology: Here The Values That Represents The Data

0 : Negative

1 : Affirmative

- : Uncertain, no response or don't know

Node	Bit String	A (n)	B (n)
1	00 00 00 00	.456	.75
2	00 00 00 11	.492	.86
3	00 00 11 00	.645	.98
4	00 00 11 11	.767	1.05
5	00 11 00 00	.8858	1.29
6	00 11 00 11	.9589	1.36
7	00 11 11 00	.987	1.57
8	00 11 11 11	1.154	1.78
9	11 00 00 00	1.458	1.97
10	11 00 00 11	1.784	2.25
11	11 00 11 00	1.858	2.354
12	11 00 11 11	1.9789	2.457
13	11 11 00 00	2.254	2.574
14	11 11 00 11	2.3914	2.722
15	11 11 11 00	2.4157	3.091
16	11 11 11 11	2.5784	3.541

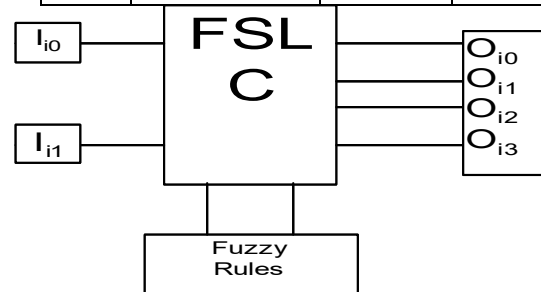


Figure 4.2 Fuzzy System Logic Controllers

UB/LB responses at each stage  
Model for Fuzzy system Logic Mapping from 2  
input variable to 1 out of 9 output states.

In our model input number functions  $I_i$ ,  $o$ ,  $I_i$ , are fed into fuzzy system logic controller. Set of fuzzy rules, specified at table to any one of nine out comes indicated.[17,18]

Table 4.2: Set Of Fuzzy Rules

IF	Response is	0 0	THEN	Travesseto	$V_i + 1, 1$
IF	Response is	0 -	THEN	Travesseto	$V_i + 1, 2$
IF	Response is	- 0	THEN	Travesseto	$V_i + 1, 3$
IF	Response is	0 1	THEN	Travesseto	$V_i + 1, 4$
IF	Response is	--	THEN	Travesseto	$V_i + 1, 5$
IF	Response is	1 0	THEN	Travesseto	$V_i + 1, 6$
IF	Response is	- 1	THEN	Travesseto	$V_i + 1, 7$
IF	Response is	1 -	THEN	Travesseto	$V_i + 1, 8$
IF	Response is	1 1	THEN	Travesseto	$V_i + 1, 9$

#### 4.6 Encoding decision paths as Binary Strings

The user may supply only part of attribute information, allowing more nodes to be included at each step. In particular, the user may either (a) select a specific attribute of a child, or (b) leave the attribute unspecified, or (c) omit the attribute for inclusion. For allowing such specifications, we must refine our representation of the child nodes using binary representation to include virtual nodes, those that are not in the initial decision tree, to be specified. The virtual nodes are those that share some attributes with one child node and the rest with other child nodes in the

initial decision tree. (We exclude the case in which all the attributes are omitted or left unspecified.) In fact, for a decision node with a list of  $k$  decision attributes, we can allow as many as the  $2^k$  child nodes (including the virtual nodes and those in the initial tree). Thus, at each step, we take the nodes in the next step to be the union of all the strings, whose attributes match the specification by the user. However, we now have to construct decision attributes for each virtual node for subsequent steps. In Figure we have shown the virtual nodes that are traversed by dotted line.

Figure and Table illustrate traversal based on responses and boundaries, and information stored at each stage/level.

Table 4.1 Look Up Table

For example to reach the virtual node  $V^1$ , the table indicates, the response required from the user is **01** and  $V^1$  has lower bound of  $V_1$  and an upper bound of  $V_2$ . [17,18]

#### 5. DETERMINATION OF VARIABLES IN A EFFORT ESTIMATION FUNCTION:

The virtual nodes generated in the decision paths result in nodes at leaf-node level that do not coincide with the goal nodes in the initial decision tree. For each such node at the leaf-node level, we create an additional goal node. The lexicographic ordering on the extended binary string encoded paths render the goal nodes with a natural ordering that is compatible with the ordering they have as child nodes of their parents. It now remains to find a way to choose suitable values of the parameters for the additional goal nodes that have been introduced. For this purpose, we suggest that the parameters be interpolated such that between any two consecutive goal nodes in the initial decision tree, the interpolation curve is the monotonic. That is, for two consecutive real goal nodes,  $n_1$  and  $n_2$ , if  $a(n_1)$  and  $a(n_2)$  are the values of the parameter  $a$  then, for all virtual nodes  $n$  and  $n'$  with  $n_1 = n = n' = n_2$ ,  $a(n_1) = a(n) = a(n') = a(n_2)$ . Similarly, for the other parameter,  $b$ . This method allows us to define the values of the parameters for all goal nodes including those generated newly by allowing virtual nodes. Let the decision path under uncertainty and imprecision be represented (Figure 1) by **dotted path**  $v_0, v^1, v_4, v^2$  and  $v^3$ . The corresponding path identifier (pid) is : **01111001**. From Table 3, it can be detected that virtual node ( $V^3$ ) boundaries are  $V_{20}$  and  $V_{21}$  which are goal nodes with numbers 6 and 7. The values of  $a(v^3)$  and  $b(v^3)$  can be generated using suitable mapping function. By using these project parameters estimations can be done by using **Constructive cost model (COCOMO)** which is designed by Barry Boehm.[5,7,14]

#### 6. STAGES OF COCOMO:

1. SIZE (in terms of KDLOC- Thousands Delivered Lines of Code) is being calculated using the input table of Size Estimates of different modules.[14,15,16]
2. INITIAL EFFORT is calculated using the formula:  $E_i = a * (SIZE)^b$  ..... (2)

Where a and b are constants depending on the project type.

3. EAF- Effort Adjustment Factor is calculated using the table of cost drivers.

EAF = multiplication of selected cost drivers.

4. EFFORT for the project is calculated using the INITIAL EFFORT AND EAF.

$$\text{Effort} = EI * EAF \dots\dots\dots (3)$$

5. Phase wise effort can be calculated by using the following formula.

Percentage of Effort of that phase =  $(a + (b/c) * \text{total effort}) \dots\dots\dots (4)$

Where a = Intermediate % of effort for that phase

b= (medium KDLOC % of that phase - intermediate KDLOC % of that phase)

c= medium KDLOC size - Intermediate KDLOC size of that phase Effort of that phase = % effort of that phase /100\* total effort

6. DURATION which gives the total number of months needed for development of a project is calculated using the formula:

$$D = 2.5 * (\text{Total effort})^{0.38} \dots\dots\dots (6)$$

7. Duration for each phase is calculated using the following formula

Percentage Duration of that phase =  $(a + (b/c) * \text{total duration}) \dots\dots\dots (7)$

Where

a = Intermediate % of duration for that phase

b= (medium KDLOC % of that phase - intermediate KDLOC % of that phase)

c= medium KDLOC size - Intermediate KDLOC size of that phase Duration of that phase = % duration of that phase /100\* total duration

8. MAN-POWER calculation is done using effort for each phase and duration for each phase. Staff for Each Phase = (Effort for Each Phase) / (Duration for Each Phase) \dots\dots\dots (8)

## 7. RESULTS

1. Here the project manager can login into system, by entering the user name and password.



2. After the successful login of the Project manager, the welcome and successful logged screen will be displayed as follows.



3. Here the project manager can enter the details of the project i.e. project ID, goal identifier, KDLOC of project.



4. This result screen contains details of the project i.e. Pid, KDLOC, and name of the project manager.



5. This result screen shows, how EAF can be calculated, by selecting 15 cost drivers.



6. This result screen shows the decision tree, Goal node of the project, Bit string, constants a and b. EAF, initial effort estimate, and total effort.



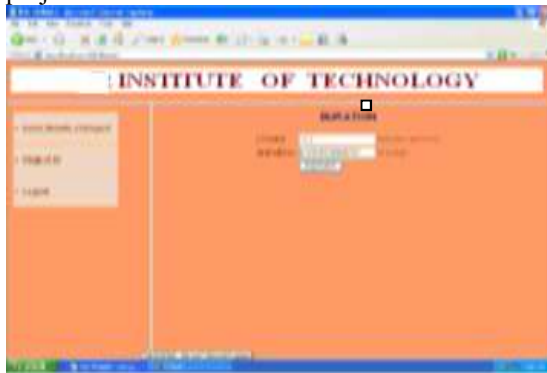
7. This result screen shows the links to the project reports, links are as listed in figure.



8. This result screen shows phase wise effort distribution of the project i.e. percentage of effort for that phase and person months for every phase.



9. This Result screen shows the total duration of the project.



10. This Result screen shows the phase wise distribution of the project



11. This Result screen shows, how nominal productivity of the project can be calculated.



12. This Result screen shows the total number of persons required for every phase of the project.



## 8. CONCLUSION

Complex projects need firm and well defined plans and to present cost and time overruns need effective monitoring and control. This aspect calls for a reliable and accurate forecasts for project estimations like effort Duration, complexity and cost etc. Thus we need an expert system that provides accurate forecasts under imprecision and partial truths. The prototype presented is a fuzzy based expert system for forecasting effort estimation under imprecision. If software experts

are efficient to claim the size of the project then it is better to go for COCOMO model to complete the estimation.

Engineering and Its Applications Vol.9, No. 10(2015). pp. 107-116.

## REFERENCES

- [1] Pankaj Jalote ,” An Integrated Approach to Soft ware Engineering” 2 nd Edition , Narosa Publishing House,2004,Chapter-4 (Planning a Software Project),Pg no. 166-170. ISBN – 81-7319-271-5.
- [2] Roger S Pressmen, “Software Engineering - Practitioner’s Approach” 6<sup>th</sup> Edition Mc Graw Hill international Edition, Pearson education, ISBN 007 - 124083 - 7 .
- [3] Waman S Jawdekar, “Software Engineering Principles and Practices” Tata Mc graw hill ISBN 0 -07 - 058371 – 4.
- [4] Walker Royce “Software Project Management - A unified frame work” 2nd Edition, low price Edition, ISBN 81 - 7758 – 378 - 6, Pearson education.
- [5] Ian Somerville, “Software Engineering” 5th Edition low price Edition, International Computer Science Series.
- [6] Carlo ghezzi, Mehdi Jazayeri, Dino Mandrioli, “Fundamentals of Software Engineering”, PHI, ISBN 81 - 203 - 0865 –4.
- [7] “Software Test Case Generation and its curtailing using G-Genetic Algorithm” ‘International Journal of Recent Technology and Engineering (IJRTE)’, ISSN: 2277-3878, Volume-8 Issue-2, July 2019.
- [8] Reducing interactive fault proneness in software application using genetic algorithm based optimal directed random testing” International Journal of Computers and Applications TJCA, Taylor Francis, 1417769  
I:10.1080/ 1206212X.2017.1417769.
- [9] “Particle swarm optimization based optimal directed random testing for reducing interactive faults” International Journal of Control and Automation Vol. 12, No. 2 (2018).
- [10] “Random Testing: The Best Coverage Technique.....An Empirical proof” International Journal of Software Engineering and Its Applications, Vol. 9, No. 12 (2015), pp. 115-122, I. Vol.9, No.12,
- [11] “Empirical Investigations to Find Illegal and its Equivalent Test Cases using RANDOM-DELPHI.” International Journal of Software Engineering and Its Applications Vol.9, No. 10(2015). pp. 107-116.
- [12] “Theoretical Investigations to Random Testing Variants and its Implications.” International Journal of Software Engineering and Its Applications Vol.9, No. 5(2015). pp. 165-172.
- [13] Developing Optimal Directed Random Testing Technique to Reduce Interactive Faults-Systematic Literature and Design Methodology.” Indian Journal of Science and Technology, Volume 8(8), 715-719, April 2015, ISSN 0974-6846.
- [14] Optimizing the Software testing efficiency by Using a Genetic Algorithm; A Design Methodology: ACM SIGSOFT Vol 38, Issue3, May 2013
- [15] “Effort Estimation based on lines of code and function points in Software Project Management”, International Journal of Computer Science & Network Security, Vol.no.8, No.6 .
- [16] “Optimal Selection of PERT for the large complex and distributed projects”, International Journal of Computer Science & Network Security, Vol.no.8, No.6 .
- [17] S.Rajasekaran, G. A. Vijayalakshmi Pai(2007) ‘Neural Networks, Fuzzy Logic and Genetic Algorithms “PHI 2003 ISBN 971-81-203-2186-1.
- [18] S.N. Sivanandam, S. N. Deepa (2012). "Principles of Soft Computing"2nd Edn, 978-81-265-2741-0, Wiley- India.