A Model for Cost Estimation of Component-Based Software in Object-Oriented Environment

Warda Khanam[1], Huma Tauseef[1], Saima Farhan[1] and Muhammad Abuzar Fahiem[1]

[1]Department of Computer Science, Lahore College for Women University, Lahore, Pakistan

***Abstract*—Software cost estimation of a project is an essential key to acceptance or rejection of any software project. Various software cost estimation techniques have been in practice with their own advantages and limitations. Accurate cost estimation helps us complete the project within time and budget. For this work we have knowledge of all available techniques methods and tools. This research work provides comparison of various software cost estimation methods and models that are frequently used for the SCE projects. Currently object-oriented approach for software cost estimation is based on classes function point method and Line of Code etc. Comparatively less attention has been paid to the software cost estimation in component-based software engineering. The main aim of this research is to provide a model for cost estimation in component based software in object oriented environment and comparative analysis of various techniques and tools.**

***Keywords*— Software Cost Estimation, Object Oriented Environment and Cost Estimation Model**

I. INTRODUCTION

In software industry the estimation of the cost is one of the major issues during development of any software. Software cost estimation (SCE) is a vital pre-development activity before starting actual development of software. In software project the estimation of the cost are based on a forecast of the size of the future system. Unluckily the software professionals miscalculate when estimating cost and schedule. Preliminary estimation of the cost always includes several elements of insecurity. Reliable early estimation is hard to obtain because of the lack of detailed information about the future system at an early stage. On the other hand early estimation is required at the time of a contract or to determine whether a project is feasible in terms of cost benefits. The process prediction guides decision making. A prediction is useful only if it is reasonably accurate [1].

Measurements are essential to assess the type of the project, the product, the process and resources. By means of measurement, the project can be controlled. Determining appropriate productivity values for the local measurement environment is called calibration; it is likely to make early predictions using models, technique, methods and tools. But several cost estimation methods and tools are too difficult to use and understand to be of much help in the estimation process. Various studies have attempted to evaluate cost models. Research has shown that estimation accuracy is enhanced if models are calibrated to a specific organization [2]. Estimation often depends on their past experience when predicting effort for software projects. Cost estimation models can support expert estimation. That's why it is crucial interest to the software industry to develop estimation methods that are easy to use, understand and calibrate.

Recently object oriented environment has emerged as a main practice. The increase of object oriented practices has required software developer and their managers to reorganize the way they have been estimating the cost, effort and size of their development software. To enhance the understanding and controlling of these costs the software houses frequently uses parametric cost model for software development cost and time estimation. On the other hand the cost accuracy of these models is pitiable when the default values embedded in the models are used [3]. The accuracy can be improved by calibrating the parameters against the data. Accurate estimation of the size is very important at pre development stages of software. At these stages it is necessary to identify a size measure by Line of code which is used for estimation of the cost [4].

II. LITERATURE REVIEW

There are two major types of cost estimation methods. Algorithmic Methods and Non-Algorithmic Methods [5].

- ***Algorithmic Methods***

Cost estimation model was initially generated using algorithmic methods. Algorithmic software cost estimation generally involves use of equations [5]. Using mathematical relations usually large amount of data is needed at first and then the results are calculated. Nowadays for cost estimation methods these models are commonly used. Algorithmic methods are further divide into different models.

The Constructive Cost Model (COCOMO) is a commonly used algorithmic cost model. Putman model was proposed by Putnam. According to Putman: "Manpower distribution and

examination of many software projects" [6]. Agile COCOMO model is a combination of COCOMO model plus an estimate by analogy method. While easy to learn and use, it provides the facility to estimate the cost of the project. In Agile COCOMO model we compare the similarity between the old project and new project and drive the cost [5]. To measure the functionality of the project in 1983 Albrecht presented function point metric. The estimation is done by this indicator using user inputs, user outputs, logic files, inquiries and interface.

- ***Non-Algorithmic Methods***

Non Algorithmic methods are different from algorithmic methods. Non-algorithmic methods are generally based on analytical comparisons and inferences. The estimation process is done using the previous datasets analysis or some information about the previous similar software in the non-algorithmic methods [7].

Expert judgment methods have capability to predict or avoid the problem which may occur in the domain. For software cost estimation it is more usable method. In expert judgment methods first of all select the experts or group of experts which have relevant experiences and understanding of specific domain. For satisfying result of cost estimation it is important for experts to exchange large volume of data among each other. There is coordinator which produces a specification and estimation form and distributes the form among experts. The experts fill out the form. The coordinator calls a group meeting in which all the experts' disuse the issue which is related to cost estimation of the software as a result of it the coordinator generates a summary [5].

For software cost estimation analogy cost estimation method is a very useful method. On the basis of analogy method a number of cost estimation models have been developed. In analogy method firstly characterize the software then select the most similar completed software. Derive the cost by similar software [5]. The top-down method of estimation is generally based on software uniqueness. In top-down estimating method the project is divided into lower-level components and higher level components. In early cost estimation this method is very important. Macro model is also called a top-down estimating method. This method is more appropriate when global properties are known. It is very useful in early phase when no detailed information is available [5].

In cost estimation process like top-down estimation method bottom-up estimation model is also an important method. In Bottom-up estimation method, the individual component cost is estimated then the cost of all components is combined to estimate the cost of software. In early life cycle process it is difficult to use the bottom-up estimating method [5].

A detailed comparison of cost estimation models is given in Table 1.

TABLE 1
COMPARISON OF COST ESTIMATION MODELS

	Requirements	COCOMO	PRICE	EJM	PUTNAM	FPA	EBA	BYL	ESTIMACS	TDE	BUE	SPQR	BIS
Model Requirements	Linked to software control method	✖	✖	✖	✖	✖	✓	✖	✓	✓	✖	✖	✖
	Applicable at an early stage	✖	✖	✓	✖	±	±	±	✓	✓	✖	±	≠
	Using available data	±	✖	±	✖	✖	±	✖	✖	✓	✓	✖	✓
	Adjustment to objectives	±	±	±	±	✖	✓	✖	±	±	±	✓	✖
	Definition of scope/domain	±	≠	✓	≠	✓	✓	≠	≠	✓	±	≠	✓
Application Requirements	Calibration	≠	✖	✖	✖	≠	✖	±	±	≠	±	≠	≠
	Accuracy	**NT**	**NT**	**NT**	**NT**	**NT**	**NT**	**T**	**T**	**NT**	**NT**	**T**	**NT**
Implementation Requirements	User friendliness	✓	≠	✓	±	±	✓	✓	±	±	✓	±	±
	Sensitivity analysis	✖	±	✖	✖	✖	✓	✓	✓	✓	✓	≠	≠
	Risk analysis	✖	✖	✓	✖	✖	±	✖	✓	✓	✖	±	✖
	Open model/traceability	✓	✖	✓	✓	✓	✖	✓	≠	≠	≠	≠	±
	Definition input	✓	≠	±	✓	≠	✓	±	±	✖	✖	±	±
	Completeness and detail output	±	✓	✖	≠	≠	✖	✓	✓	✖	✖	✓	✓

✖= the model does not satisfy the requirement ✓= satisfies the requirement ±= sufficient ≠=insufficient NT= the model was not tested T== the models were tested

III. METHODOLOGY

There is no generalized model or technique to estimate the cost of component based software. A model is presented which estimates the cost of the software by combination of different techniques and formulas. The model can be divided into following phases:

A) Requirement Validation

In requirement validation phases the requirements divided according to two types; software requirements and software domain requirements. "A detailed software description which can serve as a basis for a design or implementation, these are the requirements written for developers". The well-defined the software requirements, more accurate is the estimate of the cost of the software.

"Software Domain requirements reflect the environment in which the system operates so when we talk about an application domain we mean environments such as train operation, medical records, e-commerce etc." [8]. For each software, there are many domain requirements for example some softwares are academic based, some are industrial, some for business sector etc.

B) Steps of Experts Judgment Methods

In expert judgment method first select an expert or a group of experts which have a relevant experience and understanding of required software to estimate its cost.

The coordinator produces the specification (in which detailed description of all requirements is present) and estimation form. The specification and estimation form can be distributed among experts. The expert fills out the form. To achieve the satisfactory result they need to exchange the volume of the information among experts. The coordinator calls a group meeting in which all the experts discuss issues with coordinator and give their opinion to each other.

Characterize the required software according to domain requirements. After characterizing required software selecting the most similar completed software whose characteristics are similar to required software. From similar complete software the expert derive the cost of the software. Are there are any new requirements that can be added by the customer? If there is any new requirement firstly it will be validated. If the requirement is valid the second iteration of cost estimation can be repeated. If the requirement is not valid then coordinator generates a summary of cost estimation.

If the summary is valid then the process will be stopped. If the summary is not valid then again third iteration of cost estimation procedure will be repeated.

C) Domain Analysis Modeling

"Components that result from domain analysis are better suited for reusability because they capture the essential functionality required in the domain" [9]. In domain analysis modeling, the expert first analyzes to reuse domain component. If the software is related to specific domain, then first analyze which of the already built component can be used in the required software .This research is related to object oriented environment so the expert searches for classes or methods that can be used into new required software. The productivity can be derived by following formula:

$$RL = \frac{\text{Productivity with Reuse}}{\text{Productivity without Reuse}} * 100$$

The third phase is component based design methodology the code can be divided into four types.

- Number of classes
- Number of simple and complex data in each class
- Number of inherited classes.
- Reference parameters in the method

Calculate Line of code by LOC. The code reused component in an application plus the total Line of code delivered in the application can be calculated by this formula:

LOC=Reuse(C)/Size(C)*100%

where,

Reuse: The lines of code component reused in an application.

Size: The total lines of code in the application.

By COCOMO effort value, estimate the cost of the project by following formula:

Estimated cost of the project = Estimated Effort * Labor Rate of the organization

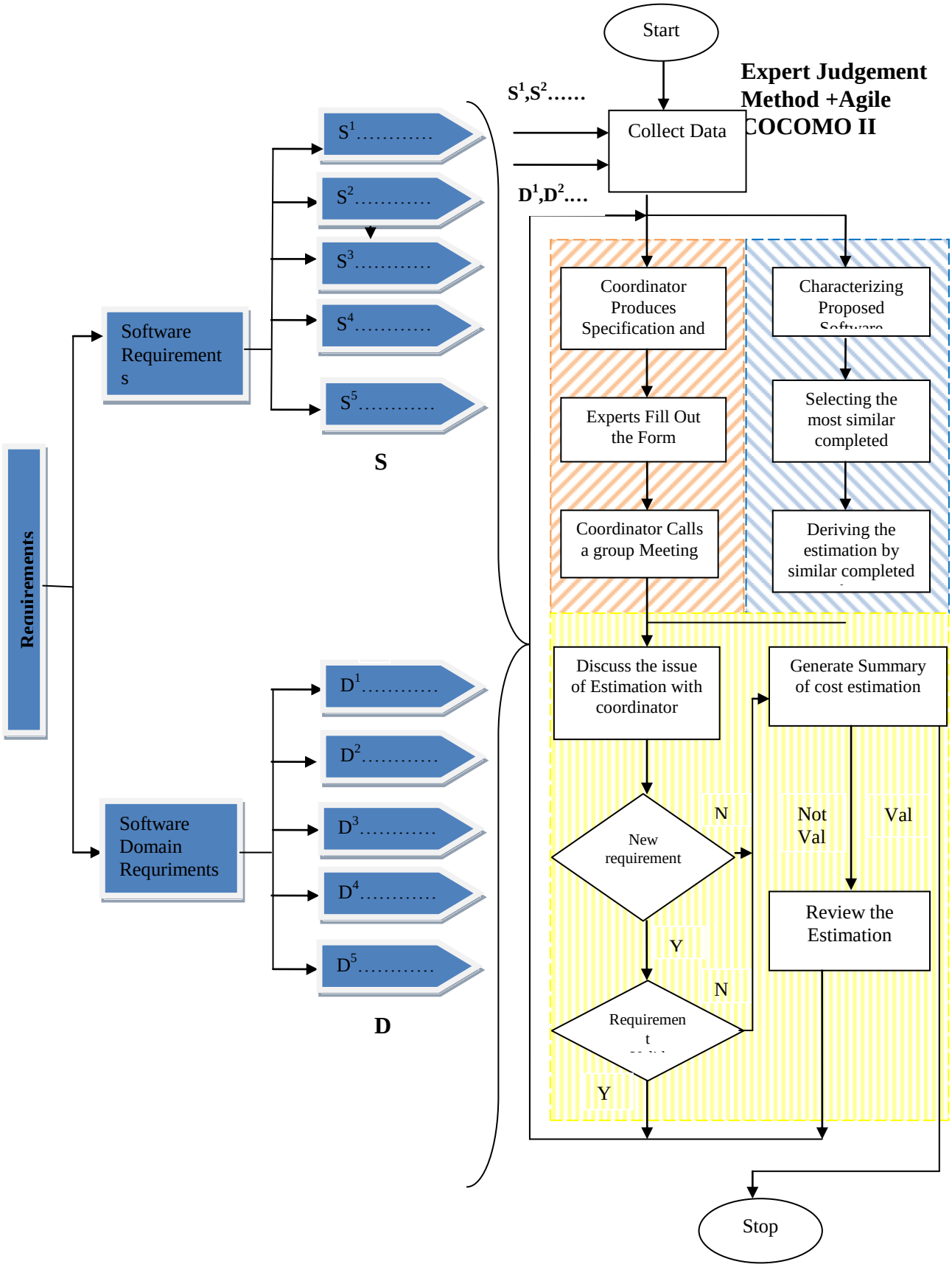

Fig.1: Proposed Cost Estimation Model

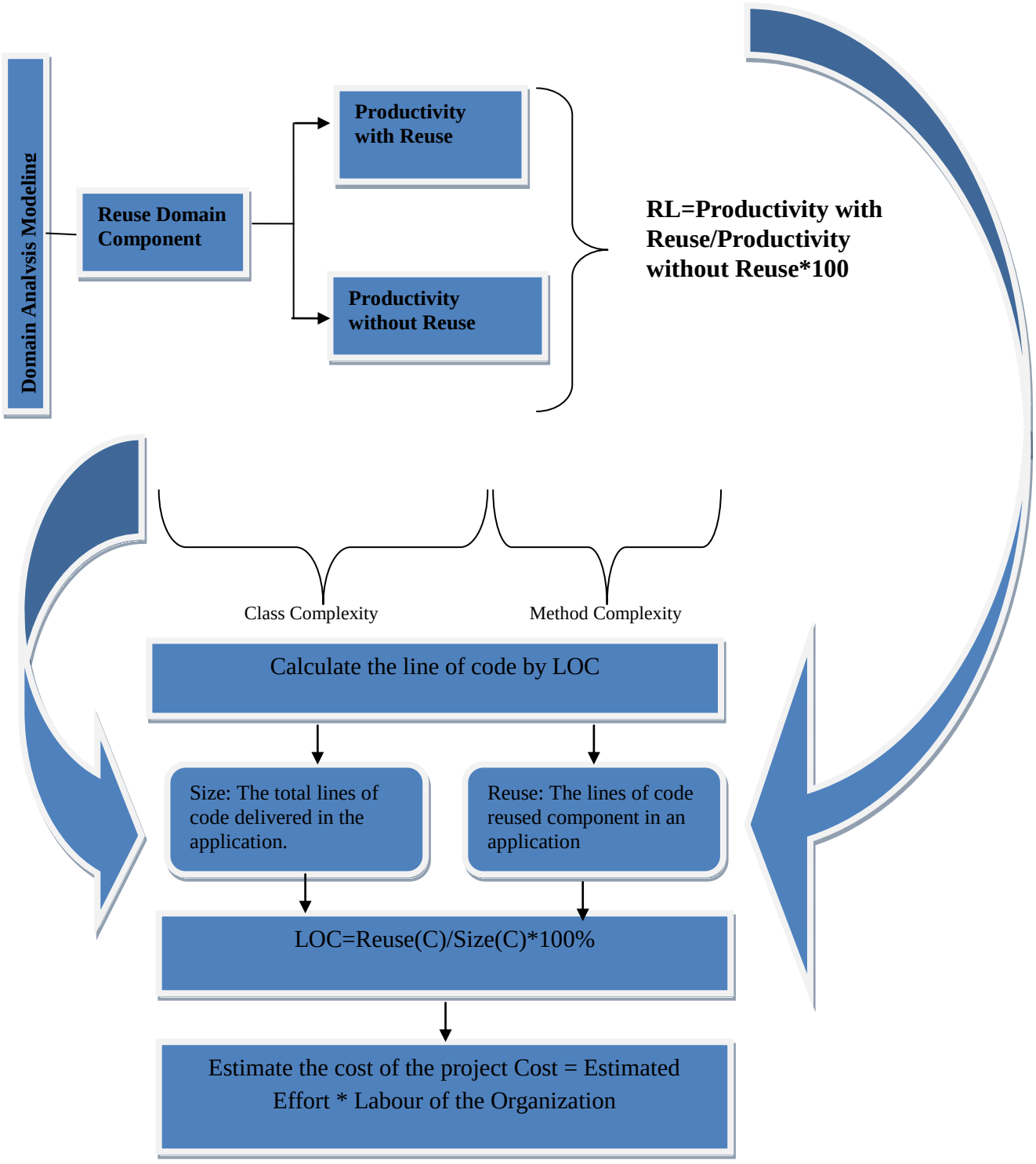

Fig. 2: Cost Estimation Process

IV. RESULTS AND DISCUSSION

Proposed model is verified and tested using a case study. The case study is presented as under:

The XYZ University wants to design the "Online test system" the system is developed to support university student to take test online and allow the student to enter fee information online. The university categorizes the fields into 3 parts. The user of the system is limited to only one batch of students. The system provides the user name and password to every student in this batch. The system provides different modes to enter edit and view. The system allows facility to

take test online. The administrator easily get the fee information about the student and provide quick view of the list of the students that are not pay the fee and are not eligible for test. The system will be providing the information to all viewers but limited access to view the test. Once the student click the button start test the time will be start and session will be establish and after session over the student will not allow to entering any data any more. The result will be generated automatically. Both system level security and data level security are important. The student will not be able to view the test of different category and before session establishes. The system only allows the authorized user to enter the fee information. The student should not be able to view the fee information of other students. The system shows different interfaces for different categories. The administrator can retrieve the data by batch wise the system must prevent access to unauthorized users. The system only allows authentic users to enter the information regarding fee slip. The system provides the privileges to access the test only by assigned category. The administrator has permission to view the data of the student. The system should be able to recover within few minutes if it is down. The system shall generate error messages when the user attempt to enter invalid data. The interface will be linked with the category in order to distinguish the category for the student.

The University already has an online system to fill out the performas that are properly working and has the complete information about university students.

In the case study there are different software requirements and software domain requirements. First of all the requirements are classified into software requirements and software domain requirements. All data is collected from customer. In first phase according to the model, COCOMO agile method is used to characterize the required software. According to given case study the system is record keeping system. Similarly, expert judgment method selects an expert or a group of experts who have record keeping software development experience and have understanding of that kind of softwares to estimate it's cost.

To achieve the satisfactory result the experts exchange the volume of the information among each other. After characterizing the required software the expert select most similar completed software whose characteristic are similar to required software select the software automatic fill the form by student on same university. The coordinator produces the specification (in which the detailed description of all the requirements define above are present) and estimation form. The specification and estimation form will be distributed among experts. The experts fill out the form. Complete software (online filling of the performas) its cost is Rs. 35000/-. The expert deriving the cost of the new requires software cost. The coordinator calls a group meeting in which all the experts discuss with the coordinator and each other that the most similar completed software has a cost of Rs. 35000/-. Further they discuss what issues and problem can arise to alter its cost. There are no new requirements so expert generates a summary of cost estimation. On domain analysis phase experts suggest that some classes and methods which are used in data form software can be used in required software.

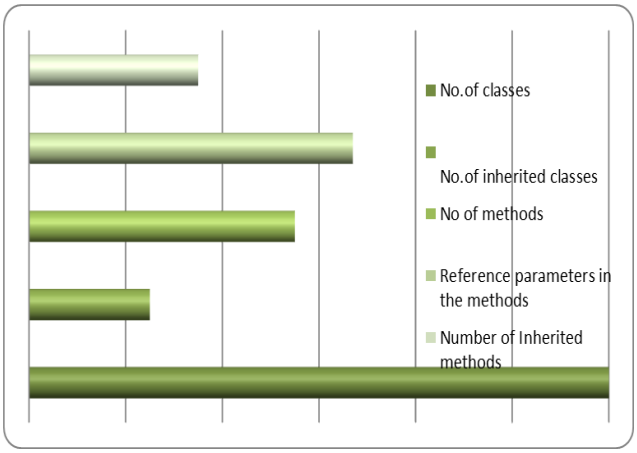

Fig. 3: Factors Calculated for Cost Estimation Using Proposed Model

For the given case study there are 120 Number of classes 25 Numbers of inherited classes 55 No of methods 67 reference parameters in the methods 35 number of inherited methods. In Component based design methodology we calculate number of the classes, number of inherited classes and number of methods. Afterwards, class complexity and method complexity are calculated.

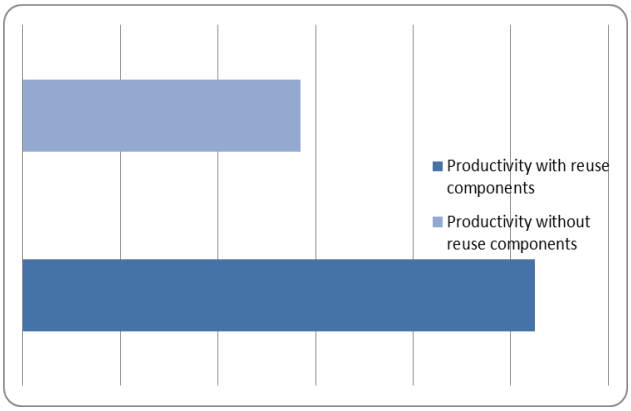

Fig. 4: Productivity Calculated by the Proposed Modell

In Domain analysis modeling, productivity with reuse components is 105. On the other hand productivity without reuse components is 57. So this is a proof that productivity increases by the reuse of components as compared to without reuse of component.

$$\text{Size} = \frac{\text{Productivity with Reuse} * 100}{\text{Productivity without Reuse}}$$

$$\text{Size} = 105/57*100=184.21$$

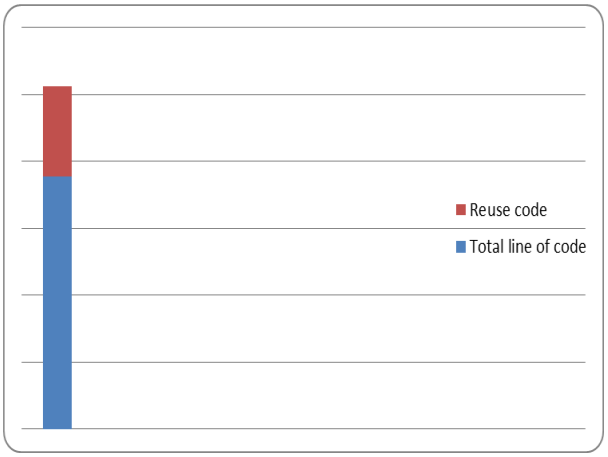

Fig. 5: LOC Calculated Using the Proposed Model

There are 256 total line of code. According to case study for the required new software firstly identify which ready built in code will be reused in the new required software. As a result 57 line of code is reused code in development of required software.

LOC=Reuse(C)/Size(C)*100%
LOC=184.21/57*100% = 256

Estimated cost of the project= Estimated Effort * Labor Rate of the organization
Estimated cost of the project= 240*200 per hours
Estimated cost of the project=48000

V. CONCLUSION

For the last decade many researches are done on the subject i.e., why the software projects fail? According to many researches the main reason of software project fails is inaccurate estimation of the cost in early stages, more over one technique is not appropriate to estimate the cost of the software. By the combination of different cost estimation technique it is possible to get an accurate result .For improving the cost estimation process many different models and methods are proposed. The main aim of this research is to present a model for component based software in object oriented environment. Here a brief but comprehensive comparison among many cost estimation techniques and model is also included.

The proposed research has provided a foundation for cost estimation for component-based software. It can be improved by the combination of several other software cost estimation technique. Proposed model can be extended to deal with other types of software like service oriented software, aspect oriented software plan driven software and agent based software etc.

REFERENCES

[1] Fenton, N.: 'Software Metrics: A Rigorous & Practical Approach 2nd Ed; PWS Pub', in Editor (Ed.)^(Eds.): 'Book Software Metrics: A Rigorous & Practical Approach 2nd Ed; PWS Pub' (Company, 1997, edn.), pp.

[2] Kitchenham, B.A.: 'Software metrics: measurement for software process improvement' (Blackwell Publishers, Inc., 1996. 1996)

[3] Ram, D.J., and Raju, S.: 'Object oriented design function points', in Editor (Ed.)^(Eds.): 'Book Object oriented design function points' (IEEE, 2000, edn.), pp. 121-126

[4] Assassa, G., Aboalsamh, H., and Al Hussan, A.: 'Software cost estimation: A tool for object oriented console applications', 2006

[5] Sharma, N., Bajpai, A., and Litoriya, M.R.: 'A comparison of software cost estimation methods: A Survey', The International Journal of Computer Science and Applications (TIJCSA), 2012, 1, (3)

[6] Kemerer, C.F.: 'An empirical validation of software cost estimation models', Communications of the ACM, 1987, 30, (5), pp. 416-429

[7] Khatibi, V., and Jawawi, D.N.: 'Software Cost Estimation Methods: A Review 1', 2011

[8] Sommerville, I.: 'Construction by configuration: Challenges for software engineering research and practice', in Editor (Ed.)^(Eds.): 'Book Construction by configuration: Challenges for software engineering research and practice' (IEEE, 2008, edn.), pp. 3-12

[9] Prieto-Diaz, R.: 'Domain analysis for reusability', in Editor (Ed.)^(Eds.): 'Book Domain analysis for reusability' (1987, edn.), pp. 23-29