



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

The Software Cost Estimation and Cost-Efficient Fees Structure Format for Educational Institutions

G. Pavithra (MBA) Department of Management studies

Anna University Regional Campus, Coimbatore

ABSTRACT:

This paper aims to provide improvement in educational software by using cost estimation techniques. Many software analyzes the project performance by measuring cost estimation accuracy. High estimation error in software leads to poor estimation skills. As we enter the new technological era, students and teachers have both changed their methods of learning and acquiring information, making the implementation of ICT technologies in the teaching process is necessary. They had two major Cost techniques: Algorithm and Non-Algorithm. Each has its own strength and weakness. In the present study, our main objective is to analyze the application of Learning Management System and making to implement the software and also compare the cost and suggest them whether to buy or make the software and also efficient fees format for Educational Institutions.

Keywords:

Software cost estimation, cost-efficient, Fees Format, Algorithm model, Non-Algorithm model.

INTRODUCTION:

In Today's environment of the Software Industry, the developed organization will have the capacity to develop and deliver the software product to the customers or end-users within the promised period of time while staying within financial budgetary boundaries. In other words, it may be said it is quite necessary to understand and control the cost by proper estimation for the proper management, enhanced quality, and better understanding of software project. The overall process of developing a software cost estimation is not different from the process for estimating any other element of cost. There is a process that is peculiar to software estimating. Some of the unique aspects of the software estimation are driven by the nature of software as a product. Continual cost estimation is ensuring that the spending is in line with the budget. Accurate software cost estimation is critical to both developers and customers. They used it for generating requests for proposals, contract negotiations, scheduling, monitoring, and control. Software cost estimation involves one or more of the following estimates:

- effort (usually in person-months)
- project duration (in calendar time)
- cost

Most cost estimation models generate the effort which is shown as a person-months that is converted in project duration and cost. This effort estimate can be converted into a cost figure by calculating an average salary per unit time of the staff involved and then multiplying this by the estimated effort required. Practitioners have struggled with three fundamental issues:

- Which software cost estimation model is to use?
- Which software size measurement to use – lines of code (LOC), function points (FP), or feature point?
- How to estimate without error?

Here I too have the same doubt but the later widely practiced cost estimation method is expert judgment. For many years, project managers have relied on experience and the existing industry norms as a basis to develop cost estimates. And also, we are going to study the fees format process we are going to analyze the college which is offering online courses and we need to compare the fees and prepare the format for educational institutions.

RELATED WORK:

In the current scenario of software industries, successful project completion within time is most vital for any industry. From the management point of view effort prediction may be a complicated process. The report says that 65-80 percent of the project faces an overrun of the delivery date. Effort overrun directly proportional to cost overrun, so accurate effort prediction is vital. There are many models available for the prediction of software development effort and cost. COCOMO (Constructive cost model) is the most commonly used, model. But machine learning methods for software prediction are more appropriate because they are more adaptable. While talking about software development effort prediction problems, the output (effort value) of the system is very complex dependent on input parameters, such as the size of the problem, experience, and lots of others.

Boetticher conducted over and above 33,000 different experiments using neural networks on empirical data collected from separate corporatedomains. The experiments assessed the contribution of various internal product metrics (size, vocabulary, complexity, and object) to programming effort, using neural networks.

ESTIMATION TECHNIQUES:**ALGORITHM MODEL:****1. COCOMO 81*****1(a) Basic COCOMO:*****COCOMO (Cost constructive model):****Formula:**

$$E = a(KLOC)^b$$

$$D = c(E)^d \quad P = E/D$$

E=Effort D=Development time P=Number of people

Here the a,b,c,d denotes software project.

Now here minimum 200 and maximum 2000. so we are falling in category of semi-detached and Embedded $Effort = a_1 * (KLOC)^{a_2}$
PM

$$Effort = 3.0 * (200)^{1.12} \Rightarrow 1133.12 \text{ persons months} \quad T_{dev}(D) = b_1 * (Effort)^{b_2}$$

$$\text{Months } T_{dev}(D) = 2.5 * (1133.12)^{0.35} \Rightarrow 29.3 \text{ Months}$$

$$\text{Average Staff Size} = \text{Effort} / \text{Development}$$

$$\text{Average Staff Size} = 1133.12 / 29.3 \Rightarrow 38.67 \text{ Persons}$$

$$\text{Productivity} = \text{KLOC} / \text{Effort} \quad \text{Productivity} = 200 / 1133.12 \Rightarrow 0.1765 \text{ kloc /pm}$$

$$P = 176 \text{ Loc/pm}$$

1(b) Intermediate model:

<u>Project Size:</u>	<u>Nature of Project:</u>	<u>Innovation:</u>	<u>Deadlines of Project:</u>
Organic: 2-50KLOC	Small size/Experience develop e.g. Payroll, Inventory Project	Little	Not tight
Semi-Detached: 50-300KLOC	Medium size project/Medium size team e.g. Database System	Medium	Medium
Embedded: Over 300KLOC	Large projects real-time system. e.g. ATMS, Air Traffic Controls	Significant	Tight

The Intermediate model evaluates the software development effort as a function of program size and set of cost drivers that include subjective examination of the products, hardware, personnel and project attributes.

It is used for medium sized projects. The cost drivers are used for intermediate to basic and advanced COCOMO. Reliability of product, size of database, execution and storage are function of cost drivers.

The intermediate COCOMO model takes the form:

$$\text{EFFORT} = a * (\text{KLOC})^b * \text{EAF}$$

Here effort is calculated as a person-months and KLOC is the lines of codes which is used in the project.

Product attribute:

-
- Required Software reliability Extent (RELY)
- Size of application database (DATA)
 - Complexity of the project (CPLX)

Hardware Attributes:

- Run time performance constraints (TIME)
- Memory Constraints (STOR)
- Volatility of the Virtual machine environment (VIRT)
 - Required Turnaround

time (TURN) Personal

Attributes:

- Analyst capability (ACAP)
- Software engineering capabilities (PCAP)
- Application Experience (AEXP)
- Virtual Machine Experience (VEXP)
 - Programming language

experience (LCAP) Project Attributes:

- Use of software tools (TOOL)
- Application software engineering methods (MODP)
- Requirement development schedule (SCED)

Illustration:

Effort= $a_i \cdot (KLOC)^{b_i} \cdot EAF$
 (Economic Adjustment Factor 0.9-1.4)
 Effort= $1133.12 \cdot (0.9)$
 Effort=1019.808 PM

Effort= $1133.12 \cdot (1.4)$
 Effort=1586.368
 PM

Development Time= $c_i \cdot (Effort)^{d_i}$
 Development
 Time= $2.5 \cdot (1019.808)^{0.35}$
 Development Time=15.569 Month

Development Time= $2.5 \cdot (1586.368)^{0.35}$
 Development Time=18.173 Month

1(c) The detailed COCOMO:

Detailed COCOMO includes all the characteristics of intermediate version with an assessment of the cost driver’s impact on each step of the software engineering process. The detailed COCOMO model uses different effort multipliers for each cost driver attribute. The detailed COCOMO is divided into different modules and then we apply COCOMO in different modules to estimate effort and then add the effort. The Six phases of detailed COCOMO are:

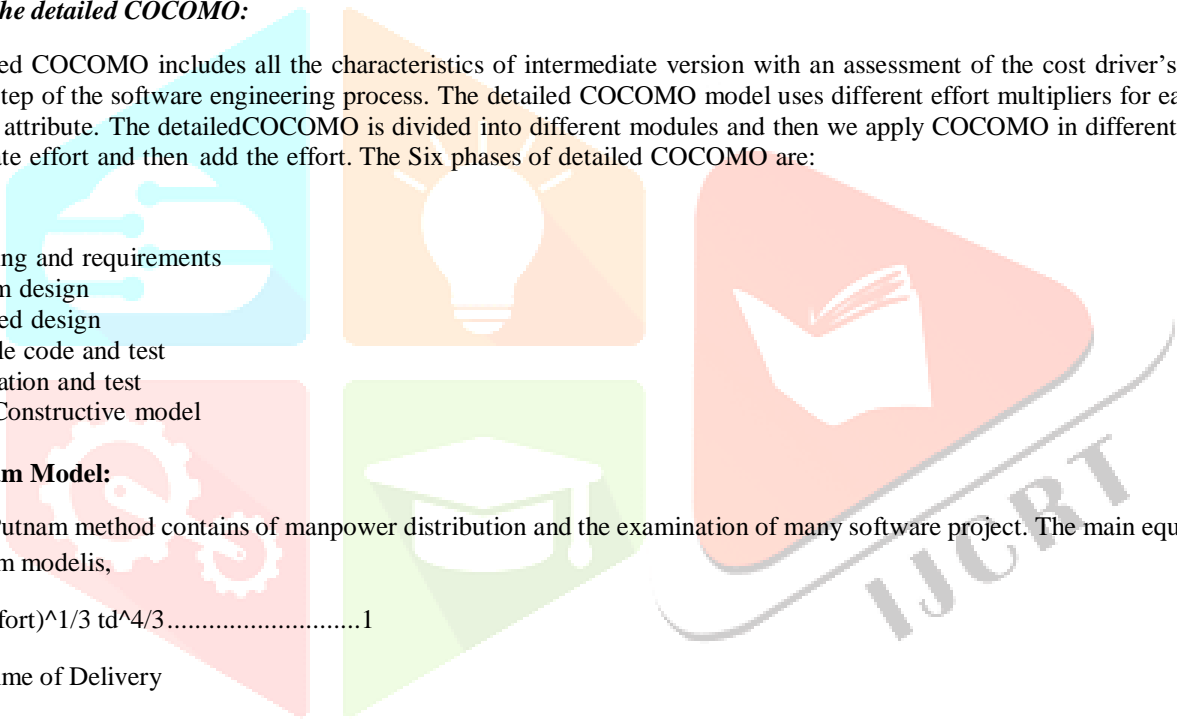
1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

2. Putnam Model:

This Putnam method contains of manpower distribution and the examination of many software project. The main equation of Putnam modelis,

$$S = (Effort)^{1/3} \cdot t_d^{4/3} \dots\dots\dots 1$$

Td=Time of Delivery



S=Person year and time of code

Putnam presented another formula for effort as follows:

Effort= $D_0 * t^3$ 2

D_0 =Manpower builds up factor varies from 8(new software) to

27(rebuild software)By equation 1&2

Effort= $(D_0^{4/7} * E^{-9/7}) * s^{9/7}$ 3

Td= $(D_0^{-1/7} * E^{-3/7}) * s^{3/7}$ 4

SLIM is tool that acts according to the Putnam model.

3. COCOMO II

COCOMO II differs from COCOMO

- Precedentedness
- Development flexibility
- Risk resolution
- Team cohesion
- Process maturity

The COCOMO II research effort was started in 1994 at USC. The model consists of three variants Application composition model, Earlydesign model, and Post architecture model

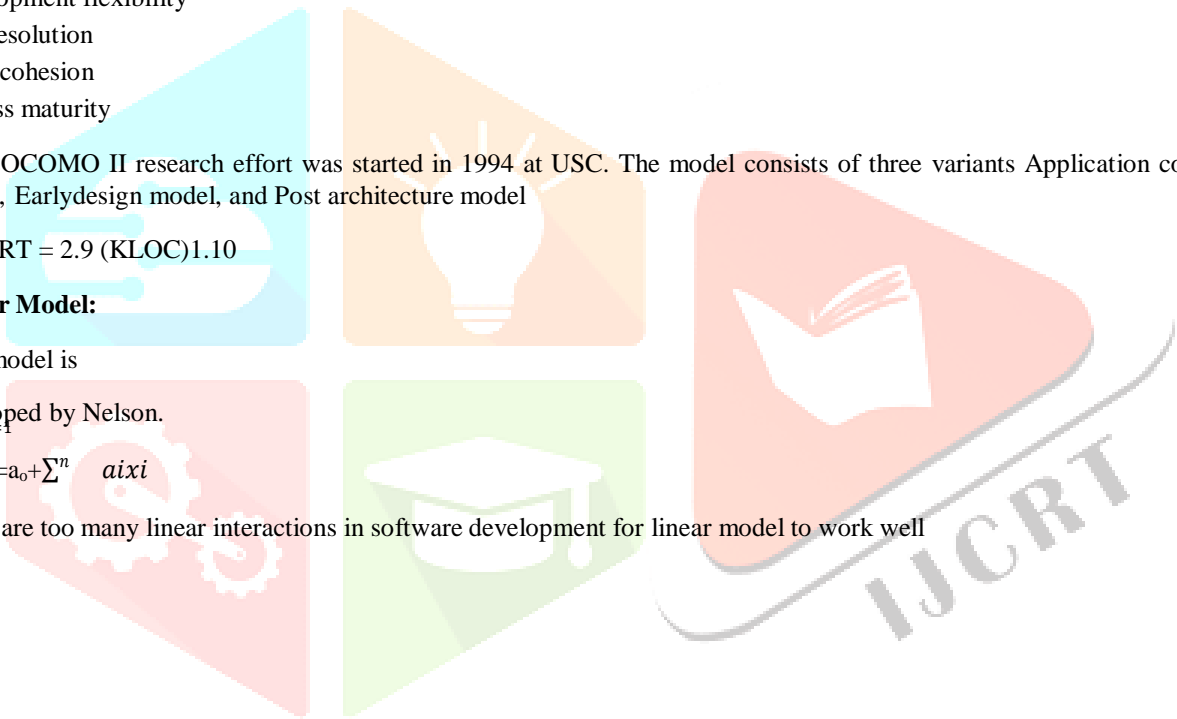
EFFORT = 2.9 (KLOC)^{1.10}

4. Linear Model:

This model is developed by Nelson.

Effort= $a_0 + \sum^n a_i x_i$

There are too many linear interactions in software development for linear model to work well



There are advantages and disadvantages of Algorithm model:

Method	Advantages	Disadvantages
Basic COCOMO	Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, commonly used in small projects, compatible for assemble language to PL/I.	It is not used in large projects where size is greater than 10000. Accuracy is limited. Its prediction is .25 which is quite poor
COCOMO II	It provides more support for modern software development processes and an updated project database. Provide support to mainframe, code reusability and batch processing	It cannot estimate the effort at all the different phase of SDLC. Its prediction is .68 which is quite good.
Detailed COCOMO	Phase sensitive effort multipliers are each to determine the amount of effort required to complete each phase	Lots of parameters involved in estimation time complexity is high. Its prediction is .70 which is good.
Linear Model	It is a best method of prediction using linear regression technique.	Little difference between actual and predicted result and error is needed to calculate.
Putnam	A Probabilistic model, Used in every large project	For only large projects

NON-ALGORITHM MODEL:**1. Analogy Costing:**

The term analogy costing is estimate from previous projects which similar to new project. It is subsystem level where it has providing the advantage of providing more detailed assessment of similarities and the difference between new project and completed project. The actual project experience and the similarity gives us more advantage.

2. Expert Judgement Method:

This method consulting with one or more experts. The experts provide using their own methods and experience. Expert consensus also called Delphi technique or PERT. The present coordinator will record. Each expert does not discuss the form individually and allowed to ask questions with coordinator. So, the coordinator collects all the summary of the experts.

3. Parkinson:

Work expand to fill available volume:-12 month and 5 people are available, the effort is estimated to be 60 person months. This method provide unrealistic estimates and it does not promote good software engineering practices.

4. Price to win:

It is prepared on the basis of customer budget instead of software functionality. Suppose project needs 100-person month but customer only afford 60-person month effort in order to with the project. Suppose again it is delay of delivery or force development team to work overtime.

5. Bottom-Up:

Each component separately estimated and result aggregated to produce and estimate overall system. This is how the system is development into different components.

6. Top-Down:

It is opposite to bottom-up method. This is used in algorithmic and non-algorithmic methods; the total costs are split in to various components. This approach suitable for cost estimation at early stages.

There are advantages and disadvantages of Non-Algorithm model:

Method	Advantages	Disadvantages
Expert judgment	Fast prediction, adapt to special projects.	Its success depends on expert usually is done incomplete
Analogy	Works based on actual experience, having an expert is not important	A lot of information about past projects is required, in some situation, there are no similar project
Parkinson Price-to-win	Oftenly win the contract	Poor practice; May have large over runs
Top-down	System-level focus; Faster and easier than a bottom-up method; Require minimum project detail	Provide little detail for justifying estimates; Less accurate than other methods
Bottom-up	Based on detailed analysis; Support project tracking better than another method, as its estimates low-level tasks	May overlook system-level cost factors; Require more estimation effort compared to Top-down; Difficult to perform the estimate early in the lifecycle

SOFTWARE OUTSOURCING:

- **Moodle**

Name: Moodle **Vendor:** Moodle **Founded:** 2001

Headquarters: Perth, Australia

Ownership: Private

Deployment Model: Cloud, On-Premise

Free Trial: Yes

Customer: Unknown

Cost Calculations:

- Starter-50 user/250 MB Storage \$150 AUD/Year.
- Mini-100 user/500MB Storage \$275 AUD/Year.
- Small-200 user/1GB Storage \$500 AUD/Year.
- Medium-500 user/2.5 GB Storage \$1125 AUD/Year.
- Large-1000 user/5 GB Storage \$2000 AUD/Year.

For example, here we are choosing large plan because we

had 1000 users. Today 1 AUD=Rs.56.20

\$2000 AUD-

Rs.112400/Year

1000

user=112400

For 1000 user they cost

112400 per year.

$112400/1000=112.4$

• **Paradiso:**

Name: Paradiso

LMS Vendor:

Paradiso Solutions

Founded: 2011

Headquarters: Boca Ratan, FL

Ownership: Private

Deployment Model: Cloud, On-Premise

Free Trial: Yes

Customer: Unknown

It is software which is created in United States which contains next level practical

features. Cost: \$2.95/month/user

Today dollar rate: Rs.72.86

$\$2.95 * 72.86 = 214.937$

per user/month

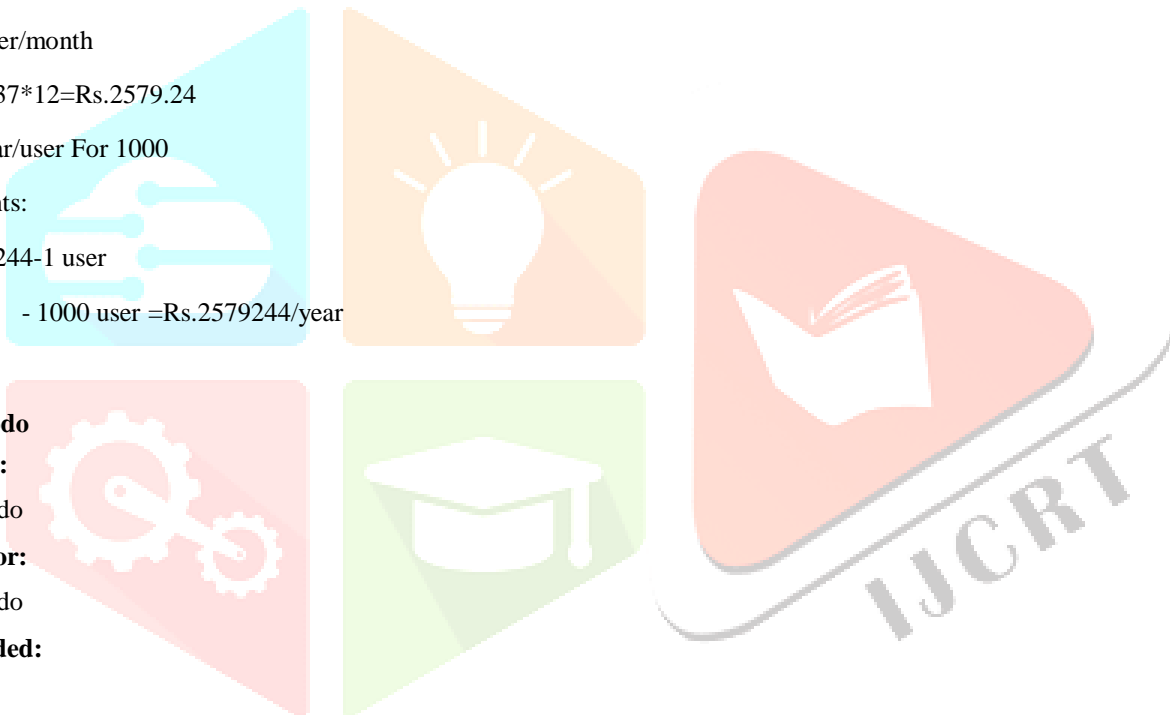
$214.937 * 12 = \text{Rs.}2579.24$

4 /Year/user For 1000

students:

2579.244-1 user

? - 1000 user =Rs.2579244/year



• **Edmodo**

Name:

Edmodo

Vendor:

Edmodo

Founded:

2008

Headquarters: San

Mateo, CA **Ownership:**

Private **Deployment**

Model: Cloud **Free**

Trial: Yes

Customer: 72 Million plus User

Cost:

Edmodo has “freemium” accounts – meaning parents, teachers, students and even school districts can create their own accounts for free. It does offer training options, such as a one-day onsite training that costs \$2,500 for 25 users.

25 users- \$2500

1000 user- \$100000 (Rs.72,86,000)

SUGGESTION:

Here, my suggestion is to Make or Buy

Decision. Reason to Outsource the software:

1. Cost Saving
2. Flexibility
3. Time Saving
4. Less Risk
5. Revision and Maintenance protocols
6. Insufficient of Resources
7. Access of latest Technology
8. Get Expertise on Board
9. Faster turnaround times and High-quality content
10. Lower your Training Development cost
 1. Leverage Existing Resources: By utilizing technology you already have; this not only reduces costs dramatically but increases efficiency. Even when it comes to maintenance, your staff already know the system and can quickly and easily make updates when needed.
 2. Integration: Systems that are not integrated result in significant increases in cost and resource consumption. It's wise to choose an LMS that has the ability to easily 'talk' to all your other business systems.
 3. Reporting Analytics: A good LMS will have top notch reporting capabilities so you can drill down to specific information that can connect training performance to essential business results.
 4. Compliance: A breach in compliance can result in huge financial penalties. An LMS should have thorough compliance tracking capabilities in case of an audit.
 5. To select the software, we need to see the hidden cost which is included in that software Implementation set up fees, licensing fee, Upgrade fee, Licensing Fee, customization fees, Support fees, Maintenance Fees.
 6. One factor in favor of eLearning in most industries is cost, namely that an eLearning solution costs less to deploy and run compared to traditional learning.
 7. Enhances Collaboration to get more Revenue by making more effective software
 8. Each and every person can access so teachers can easily access. So, we can save money for recruiting separate person for access.
 9. Need additional cost for technical proficiency
10. controlled costs - cost-savings achieved by outsourcing can help you release capital for investment in other areas of your business.
11. Increased efficiency - choosing an outsourcing company that specializes in the process or service you want them to carry out for you can help you achieve a more productive, efficient service, often of greater quality. COST INCLUDED TO CREATE SOFTWARE:

- Learning Management
- Content development
- Content Library
- Employee Training
- SCORM Compliance
- Asynchronous learning
- Synchronous learning
- Mobile learning
- Gamification
- Testing and assessments
- Certification and compliance management

- Performance tracking
- Infrastructure
- IT support

The cost classified in to:

- Per Learner, Per use
- Per Learner, per Month
- Per course
- Licensing Fee

Additional Cost:

- Implementation
- Training
- Support
- Maintenance
- Content Creation
- Configuration
- Upgrades
- Technical Support for students and Teachers

FEES STRUCTURE FORMAT:

Here, my suggestion about Fees Structure Format for (Government Institutions):

So normally enrollment fees contain of Rs.200, Subscription fees for Rs.112.4 (Maximum Rs.500), E-Learning Material contains for Rs. 2500 – 3500 (include of Video,PPT,PDF), Other Fees contains for Rs.1000 – 2000.Total =Rs. 4200 – Rs.6200 per semester (Minimum andMaximum).we can give as much as low fees structure and also, we can educate each and every student.

First Semester	Every Subsequent Semester
Enrollment Fees- XXX(Application fee)	Subscription Fees-XXX
Subscription Fees- XXX(Tuition Fees)	E-learning Materials- XXX(Books)
E-learning Materials- XXX(Books)	Other Fees-XXX (Course Development Fees/Learning Aids)
Other Fees -XXX (Course Development Fees/Learning Aids)	

CONCLUSION:

Findings are most important reason for the software project failures has been the subject of many researches in last decade. According to the results of several researches, the root cause for software project failures is inaccurate estimation in early stages of the project. So, introducing estimation we get accurate and reliable calculations.

It is necessary to understand that principals of each estimation method to choose the best. Because performance of each estimation method depends on each parameter such as complexity of the project, duration of the project, expertise of the staff, development method and so on. Some evaluation metrics and an actual estimation example have been presented in this paper just for describing the performance of an estimation method. And also, we concluded with efficient fees format for educational institution.

REFERENCES:

- [1] A Study on Software Cost Estimation Josephin.M. Baby Anusha M.E , Rajeswari Mukesh M.E, Ph.D. Hindustan Institute of Technology and Science
- [2] Boehm, B. W., & Valerdi, R. "Achievements and challenges in cocomo-based software resource estimation", IEEE Software, ,2008.
- [3] Jorgensen, M., B. Boehm, et al. "Software Development Effort Estimation: Formal Models or Expert Judgment?" Software, IEEE,2009.
- [4] A survey paper on e-learning based learning management Systems (LMS) Chirag Patel, Mahesh Gadhavi, Dr. Atul Pate (Research Gate)
- [5] Stein Grimstad and Magne Jorgensen, "A Framework for the Analysis of Software Cost Estimation Accuracy "
- [6] Attarzadeh, I. Siew Hock Ow, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks", IEEE International Conference on Computer Engineering and Technology (ICCET).
- [7] Kemerer, C. "An empirical validation of software cost estimation models", Communications of the ACM, ,1987.
- [8] Albrecht.A.J. and J. E. Gaffney, "Software function, source lines of codes, and development effort prediction: a software science validation", IEEE Trans Software Eng.
- [9] Attarzadeh, I. Siew Hock Ow," Improving the accuracy of software cost".
- [10] Braz, M. R. and S. R. Vergilio. "Software Effort Estimation Based on Use Cases". Computer Software and Applications Conference, COMPSAC '06. 30th Annual International,2006.
- [11] Idri, A., S. Mbarki, et al. "Validating and understanding software cost estimation models based on neural networks". Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004.
- [12] Jørgensen, M. "Practical guidelines for expert-judgment-based software effort estimation", IEEE Software, 2005.
- [13] Li, J., J. Lin, et al." Development of the Decision Support System for Software Project Cost Estimation. Information Science and Engineering", 2008. ISISE '08. International Symposium on,2008.
- [14] Mahmud S. Alkoffash, Mohammed J. Bawaneh and Adnan I. Al Rabea," Which Software Cost Estimation Model to Choose in a Particular Project", Journal of Computer Science, 2008.
- [15] Maged A. Yahya, Rodina Ahmad, Sai Peck Lee," Effects of Software Process Maturity on COCOMO II's Effort Estimation from CMMI Perspective", IEEE, 2008.
- [16] Musilek, P., W. Pedrycz, et al. "On the sensitivity of COCOMO II software cost estimation model". Software Metrics, Proceedings.Eighth IEEE Symposium on,2002.
- [17] Moløkken, K., & Jørgensen, M." A review of surveys on software effort estimation. International Symposium on Empirical Software Engineering", 223-231. Retrieved from ACM Digital Library database,2003.
- [18] Yahya, M. A., R. Ahmad, et al." Effects of software process maturity on COCOMO effort estimation from CMMI perspective".Research, Innovation and Vision for the Future, RIVF. IEEE International Conference on,2008.
- [19] Ashita Malik, Varun Pandey, Anupama Kaushik et al. (2013), "Analysis of Fuzzy Approaches for COCOMO II", Vol-6, India, pp 68-75
- [20] Barry Boehm, Chris Abts, Sunita Chulan et al. (2000), "Software Development Cost Estimation Approaches", Annuals of Software Engineering, 10, pp 177-205
- [21] Barry Boehm, Chris Abts, Sunita Chulani et. al., "Software Development Cost Estimation Approaches- A Survey", IBM Research
- [22] Abdel-Hamid, T.K., Adapting, Correcting, and Perfecting Software Estimates: A Maintenance Metaphor. IEEE Computer, 1993. 26(3):p. 20-29. 1-OiEm, 2-RgSi, 3-DmSi, 4-Nr, 5-No. (S)
- [23] Abdel-Hamid, T.K. and S.E. Madnik, Impact of Schedule Estimation on Software Project Behaviour. IEEE Software, 1986. 3(4): p. 70-75. 1-Oi, 2-Si, 3-Si, 4-Nr, 5-On
- [24] Software Cost Estimation Hareton Leung Zhang Fan Department of Computing The Hong Kong Polytechnic University
- [25] D. V. Ferens, and R. B. Gumer, "An evaluation of three function point models of estimation of software effort", IEEE National Aerospace and Electronics Conference, vol. 2, 1992, pp. 625- 642.7