ISSN 2231-3478 (Print)

JUSPS-B Vol. 34(3), 25-36 (2022). Periodicity-Monthly

Section B

(Online) ISSN 2319-8052

JOURNAL OF ULTRA SCIENTIST OF PHYSICAL SCIENCES

An International Open Free Access Peer Reviewed Research Journal of Physical Sciences

website:- www.ultrascientist.org

Estd. 1989

Software Cost Estimation

(Extension of Putnam's Model)

SHUBHA DEVI YADAV

Department of Mathematics
D.A.V. Degree College, Lucknow, U.P., (India)
Corresponding Author Email: Shubha.yadav70@gmail.com

http://dx.doi.org/10.22147/jusps-B/340301

Acceptance Date 18th April, 2022, **Online Publication Date 31st May, 2022**

Abstract

The Putnam Estimation Model is a dynamic multivariable model that assumes a specific distribution of effort over the life of a software development project. The model has been derived from manpower distributions encountered on large projects total effort of 30 man- years or more). However, extrapolation to smaller software projects is possible, In Putnam model there is only one choice with the manager to use for cost estimation. Thus, the estimation is very restricted. Now we want that the manager should have some choice while estimating the value of K & td (and thus cost estimations). So, in this paper, we want is to find a zone solution to the equations, so that manager can choose the value of & as per his requirement within that solution zone. By doing so we can make the estimation more general.

Key words : Putnam Model, software cost estimation.

Introduction

In the beginning of computer era of computing, software costs, represented a small percentage of the overall cost of a computer-based system. Now the days software is the most expensive element in many computer based system. A small cost-estimation error can make the difference between profit and loss. Software cost and effort estimation can never be an exact science. There are so many variables, such as human, technical environmental, etc. that can affect the ultimate cost of software and the effort applied to develop it. The main reasons why it is important to understand and control software costs, are:

(i) Software costs are big and growing. Thus, any percentage of cost saving, will have a big and growing impact on overall system.

(ii) Many useful software products are not getting developed due to huge expense. Helping good software people work more efficiently, will provide time for them to deal with this backlog of needed software.

(iii) Understanding and controlling software, costs can get us better software. As our lives and life style continue to depend more and more on software, this factor has a great impact on all of us.

A number of studies have indicated that software costs are large and rapidly increasing. For the united states in 1980, using two separate approaches and relatively conservative assumptions, (Bochn, 1983) derived a total of 900000 - 1000000 software personnel, with a resulting annual cost of $ 40 billion.

A number of potential options are there to achieve reliable cost and effort estimates, such as:

1- Delay estimation until late in the (Obviously we can achieve 100% accurate estimates after the project is complete).
2- Use relatively simple “decomposition techniques to generate project cost and effort estimates.
3- Develop an empirical model for software cost and effort.
4- Acquire one or more automated estimation tools.

Putnam Estimation Model:

The Putnam Estimation Model is a dynamic multivariable model that ass uses a specific distribution of effort over the life of a software development project. The model has been derived from manpower distributions encountered on large projects total effort of 30 man- years or more). However, extrapolation to smaller software projects is possible,

The Putnam’s approach is more along the lines of the experiments. Here we examine the data and try to determine the functional behavior by empirical analysis.

It has been well established from many empirical evidence that software projects follows a life-cycle pattern described by Norden. This life-cycle pattern happens to follow the distribution formulated by Lord Rayleigh to describe another phenomenon. Norden used the model to describe the quantitative behavior of the various cycles of projects each of which had a homologous (behaviors) character. Accordingly, it is appropriate to call the model the Norden Rayleigh model.

Software system follows a life cycle pattern. These systems are well described by the Nordern-Rayleigh manpower equation

$Y = 2\ \mathrm{Kat}\ e^{-a.t^2}$

where, (t) = number of persons at any time t (in years)
K = Total project effort in staff years
a = Acceleration factor (which determine sharpness of curve)

Acceleration factor is given by:

$a = 1/2\ td^2$

Here td , is development time.

The system has two fundamental parameters: the life cycle effort (k), the development time (td) and a function of these, the difficulty $(K/(td)^2)$. System tends to fall on a life normal to constant difficulty gradient.

Hera, if the management parameters K and td, are known, then we can generate the manpower, instantaneous cost, and cumulative cost of a software project at any time t by using Rayleigh equation.

But we need a way to estimate K and td or better yet, obtain a relation that shows the relation between k and td and the product (the total quantity of source code) Putnam used the Norden-Rayleigh curve to derive a software equation that relates the number of delivered lines of code (Source statements) to effort and development time.

$$\mathrm{SS} = C_k K^{1/3} td^{4/3}$$

where C_k is a state of technology constant and reflects "throughout constraints that impede the progress of the programmer". Typical values might be $C_k = 2000$ for a poor software development environment (e.g. no methodology, poor documentation and reviews, a batch execution mode) and $C_k = 8000$ for a good software development environment (ie; good methodology in place, adequate documentation/reviewes , interactive execution mode) ; and $C_k = 11{,}000$ for an "excellent" environment (e.g. automated tools and techniques). The constant C_k can be derived for local conditions using historical data collected from post development efforts.

Putnam used the s/w equation and the manpower build up equation to find the solution for cost estimation.

Software equation is

$$\mathrm{SS} = C_k K^{1/3} td^{4/3}$$

$$\mathrm{SS} = (C_k Y^{1/3} td^{4/3})/B^{1/3}$$

$$Y^{1/3} = \mathrm{SS}(B^{1/3})/\ C_k td^{4/3}$$

$$1/3 \log Y = \log \{(\mathrm{SS}\ B^{1/3})/C_k\} - \log td^{4/3}$$

$$\log Y = 3 \log\{ (\mathrm{SS}\ B^{1/3})/C_k\} - 4\log td \qquad (1)$$

& Manpower buildup parameter equations is

$K/td^3 = \text{constant}$

$Y/B\ td^3 = \text{constant (MBP)}$

$Y = \text{MBP}.\ B.\ td^3$

$\log Y = \log(\text{MBP.B}) + \log td^3$

$\log Y = \log(\text{MBP.B}) + 3 \log \ td^3$ (2)

Then the solution of equation I & II is the point. This coordinate point is the estimate of Y & td thus for the development cost.

In Putnam model of cost estimation, the manager had no choice. The only solution was the point of intersection of software equation and the manpower buildup equation.

Extention of Putnam Model : Putnam used Norden-Rayleigh curve to derive the software equation:

$SS = C_k K^{1/3} td^{4/3}$

He used the difficulty gradient $K/(td)^2$= constant,

where difficulty gradient is the slope of effort distribution curve at the origin. He solved the two equations to get a point (coordinate point) which represented the estimation of K & td. In Putnam model this coordinate is the only choice with the manager to use for cost estimation. Thus, the estimation is very restricted.

Now we want that the manager should have some choice while estimating the value of K & td. (and thus cost estimations). So, what we want is to find a zone solution to the equations, so that manager can choose the value of K & td as per his requirement within that solution zone. By doing so we can make the estimation more general.

Now we would like to find the zone of cost estimation. We have 5 non-linear equations:

1- $SS = C_k K^{1/3} td^{4/3}$: software equation.
2- $C = (K / td)\, e^{-1/2}$
3- $MBP = K / td^3$
4- Total cost = K cost/MY
5- td = constant.

We will find the zone of solutions to these equations. For this we will find the corresponding Linear equations and then solve them graphically. We will take all five equations one by one:

1- $SS = C_k K^{1/3} td^{4/3}$

$SS = C_k \{(Y^{1/3})/(B^{1/3})\} . (td^{4/3})$

$Y^{1/3} = (SS . B^{1/3})/(C_k . td^{4/3})$

taking log of both sides and solving, we get

$\log Y = 3 \ \{\log(SS\, B^{1/3})/C_k\} - 4\log \ td$ (3)

This is the software equation in linear form.

2- $C = (K / td)\, e^{-1/2}$

$C = (Y/B \ td)\, e^{-1/2}$

$Y = C \ B \ td / e^{-1/2}$

taking log of both sides and solving, we get

$\log Y=\log C\ B + \log td - \log e^{-1/2}$
$\log Y=\log C\ B + \log\ td -(-1/2)$
$\log Y=\log C\ B + \log\ \ +1/2$

Here C is the number of people involved in the project. So, we can take two values of C, one minimum number of people C_1 and second maximum number of people C_2. So, we have two equations as

$\log Y=\log C_1\ B + \log\ td + 1/2$ (4a)
$\log Y=\log C_2\ B + \log td + 1/2$ (4b)

3- MBP= K/td^3
MBP= Y/B td^3

taking log of both sides and solving, we get

$\log MBP= \log Y- \log B- 3\log td$
$\log Y=\log MBP+\log B+3\log td$
$\log Y=\log (MBP \text{ X } B) +3\log td$

Here, MBP is manpower build up parameter.So we can take two values of manpower build up parameter as minimum MBP1 & maximum MBP2:
so , we have two equations as :
$\log Y = \log (MBP1 \text{ X } B) + 3\log$ (5 a)
$\log Y = \log (MBP2 \text{ X } B) + 3\log$ (5 b)

4- Total cost = K cost/MY
C = Y/B const. , where cost/MY is const.
Y = C.B/const.

taking log of both sides and solving, we get

$\log Y = \log C.B - \log \text{const.}$

Here C is the total cost of development from the management. We can know the minimum & maximum limit of total cost C_1 and C_2 respectively. So,we have two equation as:

$\text{Log } Y = \log C_1 B\text{-}\log \text{const}$ (6a)
$\text{Log } Y = \log C_2 B\text{-}\log \text{const}$ (6b)

5- td = constant.

taking log of both sides and solving, we get

$\log td = \log C$

where td is the time of development & we can ask the management for minimum and maximum limit of the time of development, as td_1 and td_2.

so we have two equations as:

$\log td_1 =\log C1$ (7a)
$\log td_2 =\log C2$ (7b)

Now we have nine linear equations in Y & variables. We will solve than graphically.

For this we will take software equation as objective equation & other eight as constraint equations.

Manpower buildup parameter -

Table 1: Values obtained for Manpower buildup parameter at different M.B.J index.

M.B.parameter	7.3	14.7	26.9	55	89	223
M.B.J(index)	1	2	3	4	5	6

Special skill factor:

Table 2: Values obtained for Special skill factor .

Size(in SLOC)	6-15K	20K	30K	40K	50K	770K
B:	0.16	0.18	0.28	0.34	0.37	0.39

Productivity parameter :

Table 3: Productivity parameter vs P-index as detailed in the model.

Productivity parameter	**P-index**
754	1
987	2
1220	3
1597	4
1974	5
2584	6
3194	7
4181	8
6186	9
6765	10
8362	11
10946	12
13530	13
17711	14
21892	15
28657	16
35422	17
46368	18

T.P.P. solution for cost estimation: Now we will solve these nine equations graphically using following parameters :

software equation:

1: $\log Y = 3 \log \{(SS\ B^{1/3})/C_k\} - 4\log\ td$

SS = 20000 LOC

B = .18

Ck = 1220.

$\log Y = 3\{\log (20000 \text{ X } (.18)^{1/3})/1220\} - 4 \log td$

$\log Y = 3 \text{ x } .9664276 - 4 \log td$

$\log Y = 2.899283 - 4 \log td$ (8)

Table 4: Values obtained for equation 8 (stated above)

td	log *td*	log Y
0.2	-0.69	5.69
0.4	-0.39	4.49
0.6	-0.22	3.78
0.8	-0.09	3.28
1	0	2.89
1.2	0.07	2.58
1.4	0.14	2.31
1.6	0.2	2.08
1.8	0.25	1.87
2	0.3	1.69
2.2	0.34	1.52
2.4	0.38	1.37
2.6	0.41	1.23
2.8	0.44	1.11
3	0.47	0.99

II: $\log Y = \log C B + \log td + 1/2$

C1=10

C2=20

B= .18

when C = 10 & B = .18

$\log Y = \log (.18 \text{ x } 10) + \log td + 1/2,$

$10g\, Y = 0.2552725 + \log td + .5$

$\log Y = 0.7562725 + \log td$ (9a)

Table 5: Values obtained for equation 9a (stated above)

td	log *td*	log Y
0.2	-0.69	0.05
0.4	-0.39	0.35
0.6	-0.22	0.53
0.8	-0.09	0.65

1	0	0.75
1.2	0.07	0.83
1.4	0.14	0.9
1.6	0.2	0.95
1.8	0.25	1.01
2	0.3	1.05
2.2	0.34	1.09
2.4	0.38	1.13

when $C = 20$ & $B = .18$

$\log y = \log(.18 \times 20) + \log td + 1/2$

$\log y = 0.5563025 + \log td + 1/2$

$$\log y = 1.0563025 + \log td \tag{9b}$$

Table 6: Values obtained for equation 9b (stated above)

td	log *td*	log Y
0.2	-0.69	0.35
0.4	-0.39	0.65
0.6	-0.22	0.83
0.8	-0.09	0.95
1	0	1.05
1.2	0.07	1.13
1.4	0.14	1.2
1.6	0.2	1.26
1.8	0.25	1.31
2	0.3	1.35
2.2	0.34	1.39
2.4	0.38	1.43

III : $\log Y = \log(MBP \times B) + 3\log td$

$MBPI = 26.9$

$MB\ P2 = 7.3$

$B = .18$

when $MBP = 26.9$

$\log y = \log(26.9 \times .18) + 3\log td$

$\log y = \log(4.842) + 3\log td$

$$\log y = .6830247 + 3\log td \tag{10 a}$$

Table 7: Values obtained for equation 10a (stated above)

td	log *td*	log Y
0.2	-0.69	-1.41
0.4	-0.39	-0.5
0.6	-0.22	-0.01
0.8	-0.09	0.39
1	0	0.68
1.2	0.07	0.92
1.4	0.14	1.12
1.6	0.2	1.29
1.8	0.25	1.45
2	0.3	1.58
2.2	0.34	1.71
2.4	0.38	1.81

when MBP = 7.3
B = .18
$\log Y = \log (7.3 \times .18) + 3 \log td$
$\log Y = \log (1.314) + 3 \log td$
$\log Y = 0.118995 + 3 \log td$ (10 b)

Table 8: Values obtained for equation 10b (stated above)

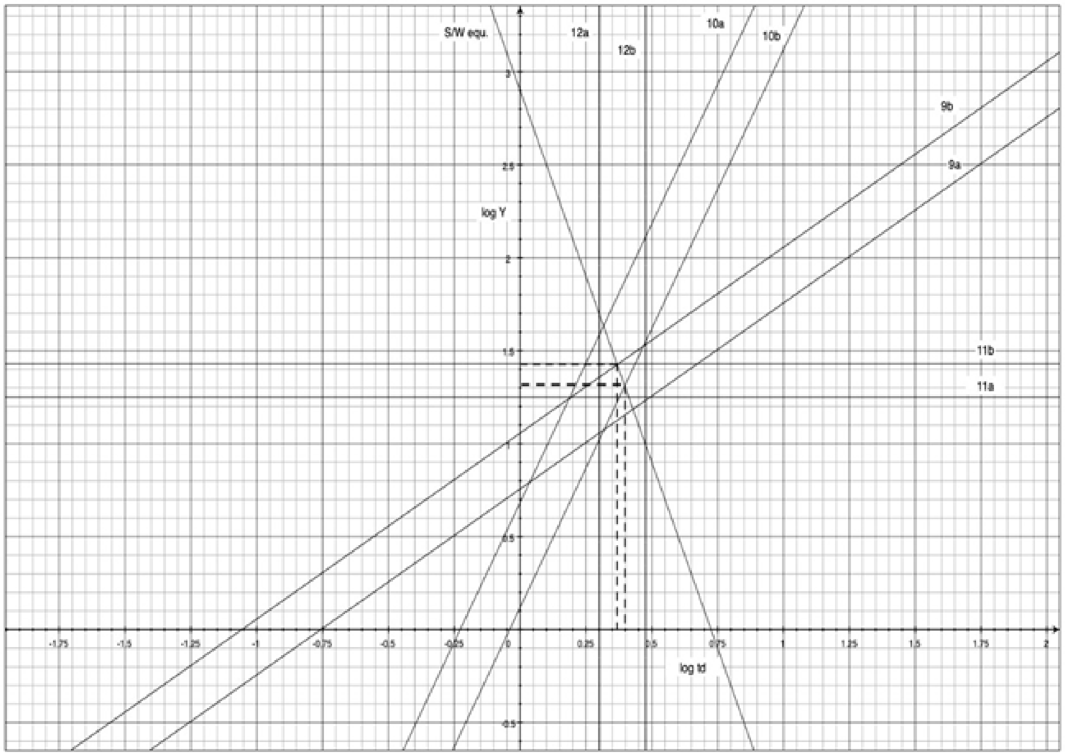
td	log *td*	log Y
0.2	-0.69	-1.41
0.4	-0.39	-0.5
0.6	-0.22	-0.01
0.8	-0.09	0.39
1	0	0.68
1.2	0.07	0.92
1.4	0.14	1.12
1.6	0.2	1.29
1.8	0.25	1.45
2	0.3	1.58
2.2	0.34	1.71
2.4	0.38	1.81

IV : $\log Y = \log C.B - \log \text{const.}$
$C_1 = 2000000$
$C_2 = 3000000$
B = .18
const. 20000

When $C = 2000000$
$\log Y = \log (2000000 \times .18) - \log 20000$
$\log y = 5.5563025 - 4,30\,10$
$\log Y = 1.25$ (11 a)
when $C = 3000000$
$\log y = \log (3000000 \times .18) - \log 20000$
$\log y = 5.7323938 - 4.3010$
$\log y = 1.43$ (11b)

V : $\log td = \log (\text{const})$
$td_1 = 2$ years
$td_2 = 3$ years,

when $td = 2$ years.
$\log td = \log (2)$
$\log td = .3010$ (12 a)
when $td = 3$ years,
$\log td = \log 3$
$\log td = 0.4771$ (12b)



Graph:

Conclusion and Future Work :

As can be inferred from the graph plotted, the manager has a number of choices for y and t thus for cost estimation. As per his requirement he may choose any coordinate point on software equation between A and B. Thus, point A is the point of maximum effort and minimum time. And B is the point of minimum effort and maximum time.

Suppose the manager choose the point A in accordance with his requirement.

So, $\log Y = 1.43$.
& $\log td = .37$
$\log y = 1.43$
$Y = 26.915$
& $\log td = .37$
$td = 2.344$.
so, estimated cost of development.
$= 26.9\,15 \times 20000 = 538300$

Here the total cost lies between the minimum and maximum. limits. Thus, the solution to cost estimation is feasible.

Thus we can say that, by having a choice of coordinates between A and B, it helps us to assess the quality and the productivity of people working on the project. It also helps us to form a baseline for estimation and to help, justify requests for new tools and additional training for better results.

Above discussed is a practical technique for acquiring prompt, trustworthy results established on tested values and comparable sizing. Each item in the Software development cycle can be allocated a number that represents its size and complexity. Enterprises are anticipated to gain some assurance that they will thrive in a "world of the unexpected." That is unattainable for software development teams without a proficient technique at all levels of software development. One of them is software cost estimation.

References :

1. R., Shweta K.; S, Duraisamy; T., Latha Maheswari, Optimized Cost Estimation Model in early stage software- A review. Ilkogretim Online., Vol. *20* Issue 1, p1877-1887. 11p. (2021).
2. Fashina, Alfred,Challenges in the process of Estimating Software Cost Jyvaskyla: University of Jyvaskyla, 50 p. (2021).
3. Shweta K R*1, Dr. S. Duraisamy*2, Dr. T Latha Maheswari, COMPARATIVE ANALYSIS OF ALGORITHMIC, NON ALGORITHMIC AND MACHINE LEARNING MODELS FOR SOFTWARE COST ESTIMATION: A SURVEY, International Research Journal of Modernization in Engineering Technology and Science,Volume:*03*/Issue:03/March-2021.
4. F. Aizaz, U. I. Janjua, H. Zafar, J. A. Khan and I. Kazim, "An Empirical Investigation on Software Cost Estimation Techniques and Barriers on Agile Software Development in Software Industry of Pakistan," 2021 International Conference on Frontiers of Information Technology (FIT), 2021, pp. 194-199, doi: 10.1109/FIT53504.2021.00044.

5. Odey John Adinya* and Agaba Francis, Department of Computer Science, University of Calabar, Cross River State, Nigeria, CURRENT DYNAMICS OF SOFTWARE PRICE DETERMINATION, July 27, 2021,FUW Trends in Science & Technology Journal, www.ftstjournal.com, e-ISSN: 24085162; p-ISSN: 20485170; August, 2021: Vol. *6* No. 2 pp. 466 – 469.
6. Sweta Kumari, Shashank Pushkar 'Comparison and Analysis of Different Software Cost Estimation Methods'. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. *4,* No.1, (2013).
7. Jianglin Huang, Yan-Fu Li, Min Xie 'An empirical analysis of data pre-processing for machine learning-based software cost estimation', Information and Software Technology, Volume *67,* November 2015, Pages 108-127.
8. Z. Levy, "The software estimation dilemma, a small step forward," COMPEURO'90: Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering - Systems Engineering Aspects of Complex Computerized Systems, 1990, pp. 242-249, doi: 10.1109/CMPEUR.1990.113631.
9. P. T. Kane, N. D. Leuci and D. J. Reifer, "A cost model for estimating the cost of developing software in the Ada programming language," [1988] Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences. Volume II: Software track, 1988, pp. 782-790, doi: 10.1109/HICSS.1988.11879.
10. W. K. Wiener-Ehrlich, J. R. Hamrick and V. F. Rupolo, "Modeling Software Behavior in Terms of a Formal Life Cycle Curve: Implications for Software Maintenance," in IEEE Transactions on Software Engineering, vol. SE-10, no. 4, pp. 376-383, July 1984, doi: 10.1109/TSE.1984.5010250.
11. L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," in IEEE Transactions on Software Engineering, vol. SE-4, no. 4, pp. 345-361, July 1978, doi: 10.1109/TSE.1978.231521.
12. J. W. Ulvila, J. E. Gaffney and J. O. Chinnis, "Software metrics decision support system," Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No.98TB100262), 1998, pp. 184-192, doi: 10.1109/METRIC.1998.731245.
13. Chin-Yu Huang, Jung-Hua Lo and Sy-Yen Kuo, "Pragmatic study of parametric decomposition models for estimating software reliability growth," Proceedings Ninth International Symposium on Software Reliability Engineering (Cat. No.98TB100257), 1998, pp. 111-123, doi: 10.1109/ISSRE.1998.730861.
14. Saurabh Bilgaiyan, Samaresh Mishra, Madhabananda Das, "A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques", https://ieeexplore.ieee.org/abstract/document/7556814
15. L. H. Putnam, "Example of an early sizing, cost and schedule estimate for an application software system," The IEEE Computer Society's Second International Computer Software and Applications Conference, 1978. COMPSAC '78., 1978, pp. 827-832, doi: 10.1109/CMPSAC.1978.810575.
16. J. B. Synnott, "Managing software development - requirements to delivery," The IEEE Computer Society's Second International Computer Software and Applications Conference, 1978. COMPSAC '78., 1978, pp. 19-24, doi: 10.1109/CMPSAC.1978.810301.