Prime Number Cost Estimation Criterion

Kavita Agrawal (1), Dr. S. K. Bajpai(2)
[1] Deptt of CSE, IU, UP, India
[2] UPTU, UP, INDIA
B-64 sector H, Aliganj, Lucknow, UP, India kavitalucknow@gmail.com
H-27, sector I, Jankipuram, Lucknow, UP, India skbajpaiiet@hotmail.com

Abstract*: Estimating* cost (effort) of software accurately in the beginning of the software development lifecycle is a difficult task. Function points can be calculated apriori and are independent of the development techniques and tools used. In this paper we have proposed an idea based on cost being a prime number. In our idea we are mapping cost (effort) to prime numbers using actual function points as the count for prime numbers i.e. given function points we calculate actual function points then map actual function points to prime numbers to get the cost and conversely given cost in prime number we can find the function points. Our paper will be useful for software development industry in general.

Keywords: Cost estimation, Function point, Prime numbers, Software effort estimation.

1. Introduction

Apriori cost estimation for any software development is a difficult task [10], [12]. Software cost estimation has been proposed traditionally by many developers based on their experience such as Algorithmic model [6], Expert judgement, Analogy, Parkinson's view, Price to win, or on the basis of top down or bottom up considerations on one hand and using Function point on the other [7],[9]. All the above methods are better from one another depending upon the ground that has been used. Our aim in writing this paper is to establish a mathematical criterion for estimating cost of any project no matter what environment has been chosen except for traditional views used in Function points and reduce the cost error from approximately 102.4% [4] to 42%.

What we have observed is that Function point criterion can be mapped to prime numbers and vice versa remembering that Function point (FP) determines cost at requirement phase. In fact cost of any software developed ultimately turns out to be unique. What we find that most often cost of software is uniquely determined except for the point of view of the difference that one finds in Function points.

We tried to map cost on several other numbers[8] [11] which diverge to infinity and hold some parametric development such as Fibonacci numbers and others but found that they are not suitable for cost determination and we found that best suited mapping is done on prime numbers.

Before we propose the following preposition we explain the difference between FP and Actual FP which we use through out this paper without further explanation.

FP (denoted by X) is the Function points calculated at the requirement phase

Actual FP gives us the number of prime numbers less than equal to the cost taken as a prime number i.e. if cost is 7units (unit refers to man hours) then Actual FP is the number of prime numbers less than equal to 7 i.e. equal to 5 unit (unit refers to count) as there are 5 prime numbers less than equal to 7.

Now we propose the following preposition:

Given X as FP point at the requirement phase we calculate new FP point Y(we call this as Actual FP point) by the following equation:-

$$526.8776+2.237601X = Y \qquad \dots (1)$$

Then mapping this to Yth prime number which corresponds to the cost of the software e.g. suppose Y is 5 then cost of software will be 5th prime number which will be 7. Thus Yth prime number is known.

Conversely, given cost as prime number we calculate Actual FP by counting number of prime numbers less than equal to the cost (for this we used a program in Java) . Having computed Actual FP (denoted by X) we compute FP point (denoted by Y) by the following equation:-

$$319.6800037 + 0.226114097X = Y \qquad \dots (2)$$

Our preposition above thus determines cost of software and having obtained cost we compute FP as well.

Our method apriori need the knowledge of how function points criterion has been established for evaluating cost estimation. We give brief account of this method and then used function points on the basis of the equations illustrated in the above preposition.

We have used most of the available data on cost estimation and found that above preposition develops the best way of finding cost estimation at the requirement phase itself. It is interesting to look into civil engineering project which estimate cost of any project at the requirement phase based on some mechanical engineering methods (stress & strain) for their completion and view that project after completion will be usable for certain number of years without any problem. We also view software cost estimation on the same background of civil engineering project.

Our method will be appreciated if judgement is drawn on some worked out project and their cost estimation. We will illustrate our view on this strong point only. We considered the data available and cost available and compare the evaluation based on our method, remembering that our cost will be unique as prime numbers are unique. We have given cost of software project apriori without the knowledge of the software, by simply converting prime number cost to match it to function points.

2- Function Points

We will illustrate function points as our method depends totally on this concept [1],[3]. Since theory of Function point is well known to the reader, we briefly account the necessary parts in the following.

The function point measure is done in three steps:

Count and classify the five user function points: external input types, external output types, logical internal files, external interface file types, external inquiry types. Each FP is classified and a Weight is associated with it [1] and Total unadjusted function point (UFP) is calculated.
adjust for processing complexity. The degree of influence of each of 14 general characteristics namely: Data communication, Distributed functions, Performance, Heavily used configuration, Transaction rate, Online data entry, End user efficiency, Online data update, Complex processing, Reusability, Installation ease, Operational ease, Multiple sites and Facilitate change, is taken on a scale of 0 to 5. Where 0 is no influence and 5 is maximum influence. All influences are summed (PC, processing complexity) and an adjustment factor is developed. Where Processing complexity adjustment (PCA)= 0.65+(0.01 * PC) [1].
Make the function point calculation.Thus Function Point (FP) = UFP*PCA

3. The Idea

A. We compute equation (1) using following algorithm 1 and then calculate the cost of the software using algorithm 2 on certain specific data given in figure 2 [4].
Using data [4]

Figure 2

Project number	Actual MM	Function point
1	287	1217
2	82.5	507
3	1107.31	2306
4	86.9	788
5	336.3	1337
6	84	421
7	23.2	100
8	130.3	993
9	116	1592
10	72	240
11	258.7	1611
12	230.7	789
13	157	690
14	246.9	1347
15	69.9	1044

MM=number of man months
(=152 working hours) [4]

Using data given in figure 2 for deriving equation 1 we use the following Algorithm 1:

Step 1: consider actual MM*152. Take its nearest prime and find the number of primes less than it.

Step 2: this number becomes the reverse FP.

Step 3: using linear regression we relate FP (denoted by X) and reverse prime (denoted by Y). In linear regression[13] Straight line equation is taken to be: ao+a1*X=Y;
Where ao= $(\Sigma Yi\Sigma Xi^2-\Sigma Xi\Sigma(XiYi))/(n\Sigma Xi^2-(\Sigma Xi)^2)$
And a1=$(n\Sigma XiYi-\Sigma Xi\Sigma Yi)/(n\Sigma Xi^2-(\Sigma Xi)^2)$
For our data ao=526.8776 and
a1=2.237601

Step 4: resulting equation is the equation (1) ie 526.8776+2.237601X = Y

Now we obtain cost estimation using Algorithm 2 as follows:

Step 1: Function point is calculated at the requirement phase.

Step 2: Actual function point(denoted by Y) is calculated using function points (denoted by X) calculated in step 1, in equation (1).

Step 3: Effort (cost) is calculated (estimated MM) as a prime number corresponding to actual function point (Yth prime number).

Step 4: Error % is calculate as
((estimated MM-actual MM*152)/actual MM*152)*100

Step 5: Average error is calculated taking absolute values.

Figure 3

FP (X)	REVERSE FP	ACTUAL FP (cal Y USING EQUATION)	ESTIMATED MM (TAKING ACTUAL FP'S CORRESPONDING PRIME)
1217	4546	3250.04	30059.00
507	1499	1661.34	14083.00
788	1571	2290.11	20261.00
1337	5231	3518.55	32801.00
421	1524	1468.91	12281.00
100	493	750.64	5693.00
993	2242	2748.82	24851.00
1592	2028	4089.14	38803.00
240	1330	1063.90	8527.00
1611	4141	4131.65	39229.00
789	3738	2292.34	20287.00
690	2654	2070.82	18059.00
1347	3974	3540.93	33023.00
1044	1296	2862.93	26021.00

FP(denoted by X) in figure 3 corresponds to the function point data in figure 2.

Reverse FP is the number of prime numbers less than equal to the nearest prime number to actual MM.

Actual FP (denoted by Y) is calculated using the equation (1) and using FP(denoted by X) as x value.

Estimated MM is a prime number where actual FP is the number of prime numbers less than equal to Estimated MM.
Actual MM corresponds to Actual MM in figure 2

MM*152 is Actual MM multiplied to 152 to get effort in hours.

Error % is calculate as ((estimated MM-actual MM*152)/actual MM*152)*100 [figure 4]

B . We compute equation (2) using following algorithm 3 and then calculate the FP of the software using algorithm 4 on certain specific data given in figure 2 .Results are shown in figure 5 and figure 6.

Using data given in figure 2 for deriving equation 2 we use the following Algorithm 3:

Step 1: consider actual MM*152. Take its nearest prime and find the number of primes less than equal to it.

Figure 4

Estimated MM USING CAL Y CORRESPON DING PRIME	ACTUAL MM	MM*152	ERROR (absolute values) %
30059.00	287	43624	31.10
14083.00	82.5	12540	12.30
20261.00	86.9	13208.8	53.39
32801.00	336.3	51117.6	35.83
12281.00	84	12768	3.81
5693.00	23.2	3526.4	61.44
24851.00	130.3	19805.6	25.47
38803.00	116	17632	120.07
8527.00	72	10944	22.09
39229.00	258.7	39322.4	0.24
20287.00	230.7	35066.4	42.15
18059.00	157	23864	24.33
33023.00	246.9	37528.8	12.01
26021.00	69.9	10624.8	144.91
		Average error	42.08%

Step 2: this number becomes the reverse FP.

Step 3: using linear regression we relate FP (denoted by Y) and reverse prime (denoted by X). In linear regression[13] Straight line equation is taken to be: bo+b1*X=Y;
Where bo= $(\Sigma Yi \Sigma Xi^2 - \Sigma Xi \Sigma (XiYi))/(n\Sigma Xi^2 - (\Sigma Xi)^2)$
And b1= $(n\Sigma XiYi - \Sigma Xi \Sigma Yi)/(n\Sigma Xi^2 - (\Sigma Xi)^2)$
For our data bo=319.6800037 and
b1=0.226114097

Step 4: resulting equation is the equation (2) ie
319.6800037 + 0.226114097X = Y … (2)

Now we obtain Function point (FP) using Algorithm 4 as follows:

Step 1:Given cost as a prime number we compute actual FP i.e. we calculate Actual FP by counting number of prime numbers less than equal to the cost (for this we used a program in Java).

Step 2: calculate FP (denoted by Y) using equation (2) using Actual FP as X.

Step 3: Error percentage is calculated as
((cal FP - FP)/FP)*100

Step 4: Average error is calculated taking absolute values.

Figure 5

FP	MM*152	Reverse FP(actual FP) denoted by X
1217	43624	4546
507	12540	1499
788	13208.8	1571
1337	51117.6	5231
421	12768	1524
100	3526.4	493
993	19805.6	2242
1592	17632	2028
240	10944	1330
1611	39322.4	4141
789	35066.4	3738
690	23864	2654
1347	37528.8	3974
1044	10624.8	1296

Figure 6

FP (Y)	cal FP using equation (2)	error %
1217	1347.594688	10.73
507	658.6250348	29.91
788	674.9052497	14.35
1337	1502.482844	12.38
421	664.2778872	57.79
100	431.1542534	331.15
993	826.6278087	16.75
1592	778.239392	51.12
240	620.4117524	158.50
1611	1256.018479	22.03
789	1164.894498	47.64
690	919.7868166	33.30
1347	1218.257424	9.56
1044	612.7238731	41.31
	avg error %	59.75

4. Conclusion

The results obtained are quite encouraging. For the same data average error % using function points is 102.74%, for COCOMO basic it is 610.09%, for COCOMO intermediate it is 583.82% and for COCOMO detailed it is 607.85% [4]. Whereas in our idea we mapped cost to prime numbers and the average error % is only 42.08 which is better than in most of the commonly used techniques [3], [4]. Results based on our idea are better than the previously used techniques.

Our point of view is straight forward and will be advantageous in most of the cases. However Function points used may vary from one person to another performing the analysis, thus it contributes to the variations in the results [2], [5]. Which result will be optimal is still a point of consideration but we feel that if larger set of data is used the above result will be better.

Most interesting point is that no matter what project is taken its cost can be estimated using our method. Thus it will be very useful for software development industry.

References:

[1] Allan J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation", IEEE transactions on software engineering, SE-9, 6, 639-648, 1983.

[2] B.A. Kitchenham, E. Mendes, G.H. Travassos, "Cross versus within-company cost estimation studies: A systematic review", IEEE transactions on software engineering , vol. 33 , no. 5, May2007, pg 316-329.

[3] Chris F Kemerer, "Software project management, readings and cases" McGraw hill company, Irwin book team-1997.

[4] Chris F Kemerer, "An empirical validation of software cost estimation models," Communications of ACM, v-30, no-5, pp 416-429, 1987.

[5] Charles R Symons, "Function point analysis: difficulties and improvement", IEEE transactions of software engineering, 14, 1, 2-11; Jan 1988.

[6] Ian Sommerville, "Software Engineering", 5th edition, Addison-Wesley

[7] Bob Hughes and M. Cotterell, "Software project management", Tata Mcgraw-Hill, 3rd edition.

[8] K. Chandrasekharan, "Introduction to analytic number theory", Springer-Verlag New York Inc. 1968.

[9] Richard Fairley, "Software engineering concepts', Tata McGraw Hill

[10] S. Grimstad, M. Jorgensen, "A framework for analysis of software cost estimation accuracy", ISESE'06, Brazil, copyright 2006 ACM , pg 58-65

[11]S.P.Tripathi, Kavita Agrawal, S.K. Bajpai, "A note on cost estimation based on prime numbers", IJITKM , July-December 2010, vol III, no.2, pp 241-245.

[12] Walker Royce, "Software project management A unified framework", Addison-Wesley.

[13] V. Rajaraman, "Computer oriented numerical methods", PHI, second edition.

Author Biographies

Kavita Agrawal is Mtech in computer science. She is assistant professor, department of computer science and engineering, Integral University, Lucknow, UP, India.

Dr.S.K.Bajpai is Ex. HOD and Professor, department of computer science and engineering, IET, Lucknow, UP, India