Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(7):2617-2626

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Research and application of estimation method for software cost estimation based on Putnam model

Ning Jingfeng[1], Jiang Yan[2] and Yu Honglei[1]

[1]*Changchun University of Technology, Changchun, China*
[2]*Changchun Finance College, Changchun, China*

ABSTRACT

Effective software estimate is the most challenging and the most important activity in a software development project management. Only when using scientific methods to reasonably and reliably estimate the scale, the amount of work and cost of the software project, do the people implement good plan and control. The paper introduces several typical ways of software cost estimate, deeply analyses the model of Putnam, uses the examples to research the scale, the number of labors at the peak and the time of the project to provide information for managers to make positive decisions.

Key words: cost of software; estimation; the model of Putnam

INTRODUCTION

Estimation and cost budget are always made together, for cost budget is the purpose of estimate, without estimate, there is no possible to make cost budget. In software industry, software cost estimate is always an arduous task[1].Estimate of software development includes both scale and cost estimate. According to the reasonable estimate result, the paper sets out feasible objective, and weighs the cost, schedule and quality of software, implement effective risk management, and provides powerful basis for managers to make investment decisions[2]. The software project estimate runs through the whole software life development cycle, and also needs continuous improvement, planning, review and follow-up[3、4]. With the information accumulated, compare the actual data with the estimate data, and improve the next estimation[5].

CLASSIFICATION OF SOFTWARE COST ESTIMATION

The earliest software cost estimate can be traced back to SDC (System Development Corporation) linear model in the 1960s, went through 40 years of development [6]. Mr.Boehm, the Professor of Southern California, pointed out,“understanding and controlling of software cost bring us not only more software but also better software”[7]. At present, some domestic software enterprises also gradually pay attention to the thought and methods of software engineering, and put these technology and methods into practice. China’s software project management is China's software project management is also moving towards standardization and normalization, but start late in terms of software project estimation,for estimation methods such as various software cost estimation, estimation of the amount of work, and estimation of function points are only in the starting stage. So it is relatively difficult to estimate the software project in the software industry, and the estimation method of software engineering based on experience is still being adopted. The domestic software project estimation techniques are still on the original state to a great extent. There are many classification models for software cost estimation methods [8,9], which can be classified three types from the point of whether using algorithm model: based on algorithm model, not based on algorithm model and the combination of both methods. The basic idea of method based on algorithm model is[10]: to find the cost affected factors for software amount of work, and judge how it influences on the workload, to be

additive, multiple or exponential, in order to get optimized model algorithm expression form. The typical methodsnot based on the algorithm model are experts estimation, analogy estimation and regression analysis.The combination method of software cost estimation obviously applies various techniques and analysis methods in estimation techniques, which is the trend of software cost estimation, and also the better selection that combines various estimation methods advantages and disadvantages, and adapts to different estimation situations and demands.

Putnam estimation model is the dynamic and multivariable estimation model of workload, which is deduced from manpower distribution of large project. Putnam further studies the original model and improves methods, and explores the potential relations among various factors by collecting data. It uses data identification factor and model to explain the performance of software project, and give enough information for the administrator to make decisions for seeking more profit. Putnam model combines many concepts, such as manpower structure, techniques, environment and process productivity.

ESTABLISHMENT OF PUTNAM MODEL

During the software development project, although the quantity of problems is unknown and limited, it also needs manpower to solve the problems. Take indicatrix K of quantity of problems as total labor cost. The accumulative labor cost involved in the project $C(t)$, expressed as man year. $C(t)$ is null at the beginning of the project, and monotonously increases towards the total labor cost K. The change rate of accumulative labor cost dC/dt, shows the number of men put into at random time during development. So the accumulative cost at time t is shown as:

$$C(t) = \int_0^t m(\zeta)d\zeta \quad (1)$$

Suppose at any given time, the optimized number of men put into the project team is in direct proportion to the number of problems to be solved, and then get the following equation:

$$\frac{dC(t)}{dt} = k[K - C(t)] \quad (2)$$

Among which k is proportionality coefficient.

Because during the life cycle of the project, one group of engineers are gradually increased their efficiency, the concept of project team study is introduced. The growth trend of solving problems ability is also a favorable factor for improving work efficiency, which is shown as function $p(t)$. So the change rate of accumulative labor cost can be supposed to be in direct proportion to $p(t)$. This assumption is described through the following differential equation of first order.

$$\frac{dC(t)}{dt} = p(t)[K - C(t)] \quad (3)$$

For this equation from the beginning of the project, i.e., $t = 0$ to get any given time t integration, get the following equation:

$$C(t) = K\left[1 - \exp\left(\int_0^t p(\tau)d\tau\right)\right] \quad (4)$$

Separate variables and make replacement of $u(t) = K - C(t)$, then get $du = -dC(t)$, when $u(0) = K$, then:

$$Log\left(\frac{K - C}{K}\right) = -\int_0^t P(\tau)d\tau$$

Further assumption on the learning function $p(t)$, most of the expression of learning functions are linear, i.e., in direct proportion to time, this linear relationship may be expressed as follow:

$$P(t) = 2at \quad (5)$$

Suppose parameter a is a positive number. By integration on the equation (4), at any given time t, the formula of

accumulative labor cost $C(t)$ becomes:

$$C(t) = K[1 - \exp(-at^2)] \quad (6)$$

By accumulative cost function By accumulative cost function against time differential, the number of labor of the project can be calculated, namely, manpower function:

$$m(t) = 2Kat\exp(-at^2) \quad (7)$$

This function represents Rayleigh curve, at the beginning of the project (t= 0), its value is 0. Then increase toward the employment peak direction, after the peak, gradually reduced to zero. See Figure 1

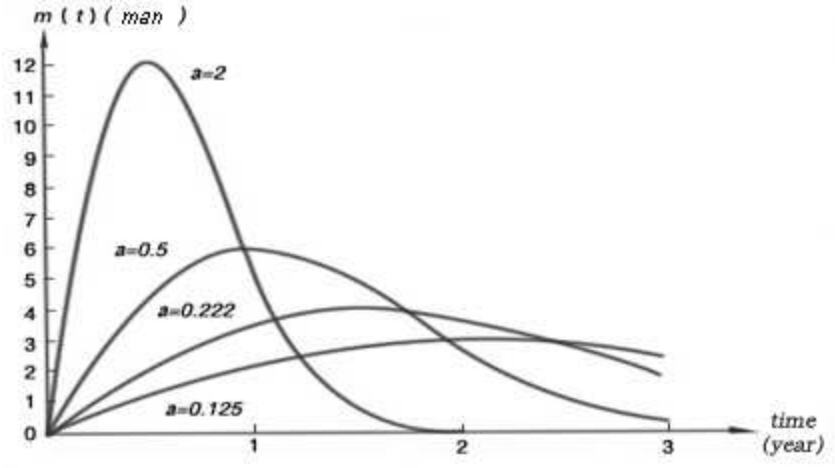

Fig. 1: Parameter a's effect on the labor distribution

From the above Figure we can see: parameter a has the dimension of inverse square of time, which plays an important role in determining of labor peak. The larger a value is, the earlier the occurrence time of peak, and the steeper the labor curve is. Derive manpower function against time, to get its zero value, then the relation between peak hour t_d and a can be found.

$$t_d^2 = \frac{1}{2a} \quad (8)$$

ESTIMATION OF LABOR PEAK

During measurement of large software development project, the manpower distribution curve has maximum value, which occurs at the time very near delivery time t_d, before that time, manpower used in specification description, design, coding, test and qualified verification; after that, manpower demand is used in maintenance, modification and other site work.

According to equation (8), the labor peak time is related to a. Accordingly, acan be obtained from the peak time as follows:

$$a = \frac{1}{2t_d^2} \quad (9)$$

So the number of men put into the project at peak time is determined by using $1/2t_d^2$ to replace a in Norden/Rayleigh model.

By replacement, equation (7) becomes:

$$m(t) = \frac{K}{t_d^2} t \exp(-\frac{t^2}{2t_d^2}) \quad (10)$$

When $t = t_d$, labor peak can be obtained, which is also expressed in m_0 . So the expression on labor peak of project is expressed in:

$$m_0 = \frac{K}{t_d\sqrt{e}} \quad (11)$$

Among which K is the total project cost in the unit of man year, t_d is the delivery time in the unit of year, e is the base number of natural logarithm, m_0 is the number of men employed at the peak.

DIFFICULTY ESTIMATION

A difficult project mainly refers to a very urgent task which needs to arrange a number of persons to work concurrently, thus resulting in the development plans difficult to control. Compare with the project not quite difficult, difficult project is more sensitive to disturbance. Generally for the project with relatively longer time limit, the manpower distribution curve is relatively flat. The Norden/Rayleigh function against differential of time gets the following equation:

$$m'(t) = 2Ka(1-2at^2)\exp(-at^2)$$

When $t = 0$

$$m'(t) = 2Ka = \frac{K}{t_d^2}$$

The rate of K/t_d^2 is called the degree of difficulty (also known as the coefficient of difficulty), expressed in D:

$$D = \frac{K}{t_d^2} (unit: man/year) \quad (12)$$

This relational expression shows, when there are more labor demands or maybe time arrangement is tight（the value of t_d is small）, the development of project is more difficult. According to formula （11）

$$D = \frac{K}{t_d^2} = m_0 \frac{\sqrt{e}}{t_d}$$

From degree of difficulty D, we can see the difficult project normally demand the relatively higher labor peak number in given development time.

LABOR INCREMENT ESTIMATION

The labor increment is also called labor accelerated speed. It shows the project's bearable and progressive manpower applied maximum allowable growth rate.

D relative to the derivative of t_d and K：

$$D'(t_d) = -\frac{2K}{t_d^3} \quad (13)$$

$$D'(K) = \frac{1}{t_d^2} (年^{-2}) \quad (14)$$

In fact $D'(K)$ is always much smaller than $D'(t_d)$.

The degree of difficulty relative to the derivative of time plays an important role in describing software development。If the project scale increases , the development time also increases to such extent , thus making

parameter K/t_d^3 focuses on a value and remain the value ,which might be 8, 15 or 27. This parameter is expressed in D_0 as：

$$D_0 = \frac{K}{t_d^3}\left(man/year^2\right) \text{（man/year）} \quad (15)$$

The value of D_0 and the developed software property have the following relations：

$D_0 = 8$ new software has many interfaces and is interactive with other software systems.

$D_0 = 15$ new and independent system.

$D_0 = 27$ software re-established from the existing software.

Parameter D_0 is called labor increment, which is also called labor accelerated speed. D_0 has great influence on the shape of the labor distribution curve. The bigger D_0 is, the steeper the labor distribution curve is, the faster the labor growth accelerate.

ACCUMULATIVE LABOR COST ESTIMATION

Define accumulative labor cost:

According to formula (6) $C(t) = K[1-\exp(-at^2)]$

Among which C (with man year as unit) gives the labor cost from the beginning of the project to t time, is the total labor cost of software life cycle (with man year as the unit) , while a is Norden coefficient defined by Putnam, it equals $\frac{1}{2t_d^2}$ (see equation 9), so the accumulative labor cost in the whole cycle is expressed as:

$$C(t) = K[1-\exp(-\frac{t^2}{2t_d^2})] \quad (16)$$

While the accumulative labor cost in development sub-cycle is expressed as:

$$C_d(t) = K_d[1-\exp(-\frac{t^2}{2t_{0d}^2})] \quad (17)$$

During the period of software project, the accumulative cost increases from zero, then one well known S-curve (see Figure 2) labor costs in delivery time in the management of the project can be obtained in equation (6) by replacement of $t = t_d$:

$$C(t_d) = K\left(1-\frac{1}{\sqrt{e}}\right) = 0.39K \quad (18)$$

From this equation we can see, the developed software system once delivered to the target processor, on the initial operate stage, it only spends 39% of the total cost K, the rest labor amount uses in acceptance test, maintenance and modification aspects, etc.

DEFINE SOFTWARE EQUATION

The definition of two important parameters of Software projects K and t_d , and by means of Rayleigh function, see how labor change with the time, now discuss the relations between K and t_d and software products, i.e., to find the relations between labor cost K and development time t_d and the delivered software scale S .

From the relations between productivity and degree of difficulty, it would get the following equation:

$$\mathrm{Pr} = C_n D^{-2/3} \tag{19}$$

Pr is productivity, D is the defined degree of difficulty, C_n is a proportional constant, which depends on the technical condition in the data collecting environment.

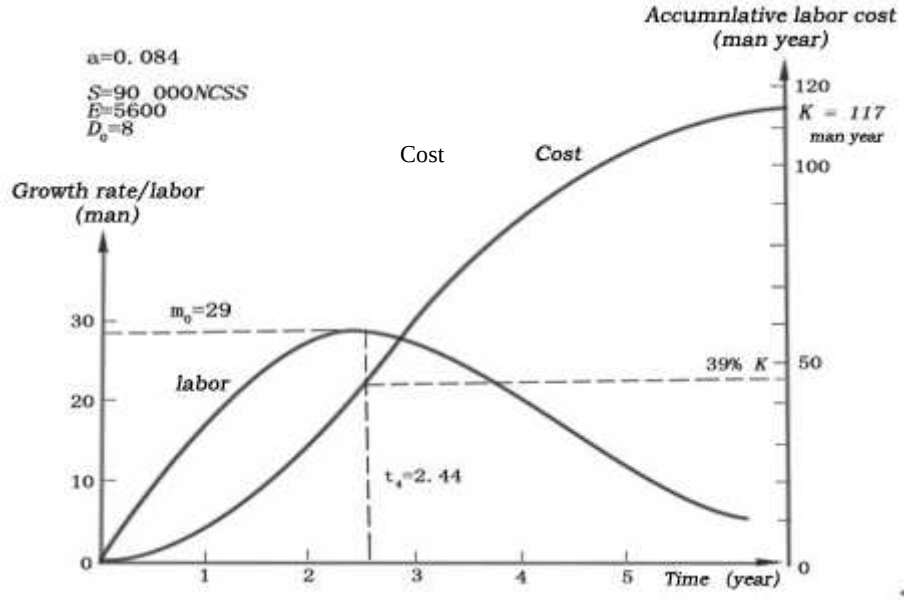

Fig. 2: Accumulative labor cost

While the degree of productivity is the rate of input to output:

$$\mathrm{Pr} = \frac{Input}{Output}$$

Set productivity Pr is the proportion of the total delivered source codes constituted product and the total labor needed to produce these source codes:

$$\mathrm{Pr} = \frac{\text{The size of delivered source code}}{\text{The total labor of code}}$$

Total labor includes labor cost spent from the beginning of the project to the delivery time t_d. From equation（18）, we get:

$$C(t_d) = 0.39K \tag{20}$$

Thus, at time t_d, the delivered software size is S, use the source program statement number as the unit expressed in:

$$S = \mathrm{Pr} \times 0.39K \tag{21}$$

From equations （19）and（21）, we get:

$$S = 0.39C_n K D^{-2/3}$$

According to equation（12）,we get:

$$S = 0.39C_n K \left(\frac{K}{t_d^2}\right)^{-2/3}$$

after systemizing, we get:

$$S = 0.39C_n K^{1/3} t_d^{4/3} \tag{22}$$

Use coefficient E to replace $0.39C_n$, we get the software equation as follows:

$$S = EK^{1/3}t_d^{4/3} \quad (23)$$

E becomes environment factor, it determined by the technical condition in the production environment. The more efficient of the method and tool of software development, the more skillful the analysts and programmers and their administrators, the higher the environment factory E is. K is the labor cost in the complete period (with man year as the unit), t_d is development time (with year as the unit), S is the scale of the software (with NCSS as the unit) .

APPLICATION CASE

Take second-hand car trading system as the example. Driven by the state second-hand car policies and logistic promotion policies, the second-hand car markets also have increased gradually, in order to grasp and control the state second-hand car price trend and the region differences, the demands for the second-hand car service platform facing the whole country and with complete functions are also increasing. Because the second-hand car market is characterized as one car one price, one place one price, one person one price, and one time one price, how to judge a car's actual quality truly and correctly is the inherent problem to increase the second-hand car market close rate to be solved, secondly, how to complete car transaction in different places conveniently, fast, and with high quality to increase the second-hand car market close rate is the key problem to be solved. This project is a project of a certain Changchun second-hand car trading company, mainly realizing the functions such as the management of second-hand car, individual user's management, enterprise user's management, and website management. For its cost estimation, relatively mature Putnam model estimation method is used. According to the previous project experience, the estimated scale of this project is 9000NCSS, not more than 18000NCSS (Non-comment source statement), so it belongs to small project, which is developed under the condition of environment factor 1200. This software is an independent data processing typed program, and its labor acceleration $D_0 = 15$. Determine the development sub-cycle:

The software equation given by equation (23) over E and the third power of which get:

$$\left(\frac{S}{E}\right)^3 = Kt_d^4$$

The right side multiplies and divides t_d^3 and replaces K/t_d^3 by labor acceleration D_0, we get the following equation:

$$\left(\frac{S}{E}\right)^3 = D_0 t_d^7$$

To solve the development time t_d, it would get:

$$t_d = \left[\frac{1}{D_0}\left(\frac{S}{E}\right)^3\right]^{1/7}$$

This D_0 value corresponds to the shortest development time, for the above given value, it would get the shortest development time:

$t_d(\min) = 1.6 year$, this value also approximately equals to 1 year and 7 months.

Estimate the labor needed in the complete period. From equation （15） to solve K, it would get:

$$K = D_0 t_d^3 = 62.7 \text{man year}$$

From the degree of difficulty defined by equation (12), it would estimate:

$$D = \frac{K}{t_d^2} = 24.5 man/year$$

man/year

Seen from the above, according to the current practical experience of our team, programming tasks initially increase

at the growth rate of two persons monthly.

Human distribution of software development is a sub –cycle of the complete cycle, it is also Rayleigh function, according to equation (10), described again as follows:

$$m_d(t) = \frac{K_d}{t_{0d}^2}.t.\exp(-\frac{t^2}{2t_{0d}^2}) \quad (24)$$

In the formula, t_{0d} is labor peak hour (unit: year), is the labor cost (unit: man year) relevant to the sub-cycle of the product development, the manpower spent on the development activities of the products, starts after the initial design recheck and stops at the system test and is accepted. Thus it covers the subsystem design, module specifications, design, coding, test, subsystem test, system test and acceptance. K_d and t_{0d} have the following specific value relation:

$$K_d = \frac{k}{6} \quad (25)$$

$$t_{0d} = \frac{t_d}{\sqrt{6}} \quad (26)$$

Thus it would get K_d =10.5 man year. 95% of which would spend on the development period of t_d.
According to the accumulative labor cost expression of the time t given by equation (17), it is described again as follows:

$$C_d(t) = K_d\left[1-\exp\left(-\frac{t^2}{2t_{0d}^2}\right)\right] \quad (27)$$

Substitute $t = t_d$ into equation, and divided by K_d on both sides of the equation, then it would get:

$$\frac{C_d(t_d)}{K_d} = 1-\exp\left(-\frac{t_d^2}{2t_{0d}^2}\right)$$

Then based on equation (26), it would get:

$$\frac{C_d(t_d)}{K_d} = 1-\exp(-3)$$

thus get:

$$\frac{C_d(t_d)}{K_d}\times 100\% = 95\% \quad (28)$$

This means within the delivery time t_d, 95% of the total development/project labor cost has consumed away, 5% of K_d leaves for the site installation and qualification testing expenses.

So $C_d(t_d) = 9.9$ man year. The rest 0.6 many year (7.2 man months)would spend on after-sales services.

According to the above data, the labor characters used in the development peak hour is calculated. The peak labor time is given from equation (26), so we have:

$$t_{0d} = \frac{t_d}{\sqrt{6}} = 0.65 year$$

or 7.8 months.

Substitute $t = t_{0d}$ into equation (24), it would get:

$$m_{0d} = m_d(t_{0d}) = \frac{K_d}{t_{0d}}\exp(-\frac{1}{2})$$

i.e. $m_{0d} = \frac{K_d}{t_{0d}\sqrt{e}}$

So number of men in peak labor time $m_{0d} = 9.8man$. From the obvious actual consideration, we think the number of men in the development team would increase to 10 within 7 months after the project begins.

From the above result,we find, if the time period is short, first, the higher the Norden coefficient α value is, the steeper the labor distribution curve is; secondly, labor peak value formula $m_0 = K/t_d\sqrt{e}$ gives higher value, which is conformed with the former point. This means the more difficult the project is, the more demand to concentrate more persons, and the earlier the concentrate time is. The implicit meaning is, a very urgent task would need to arrange more persons to work concurrently, and making the development plan not easy to control. Compare with not very difficult project, difficult project is more sensitive to the interruption, the slightest demands change or response result in management will cause the decrease of $\alpha = 1/2t_d^2$, i.e., for the project with relatively longer time limit, the manpower distribution curve is relatively flat. Naturally, if the manpower demand is higher, this effect is more obvious.

ESTIMATION RESULT AND COMPARATIVE ANALYSIS

Compare the estimation result with the actual result of the case using Putman model as follow, the accumulative labor cost of the development sub-cycle is 9.9 man year, while the actual one is 12 man year; the estimation result of labor time at the peak is 7.8 months, while the actual time starts in the 10th month after the project begins; the estimation result of the labor number of people at the peak is 10, the actual result is 8. The actual value is basically near the estimated value. In the existing software estimation methods, using the aforementioned experts estimation methods and analogy and other methods, can also estimate the software project. But the expert estimation methods mainly depend on the expert to make the individual judgment by predicting the object future development trend and condition; estimation is able to be made according to the specific software project status, with stronger pertinence, it only depends on the estimation experts' knowledge and experience, with more subjectivity and lack of objectivity, in actual work, for software project of the same type or even the same software project , different experts might evaluate obvious different results. While during the estimation of software by analogy method, it is the scale estimation by the comparison between new project and history project, which is suitable to evaluate some projects of equivalent application fields, environment and complexity with history project, the precision of its estimation results depend on the integrity and correctness of the history project data.

CONCLUSION

Putnam find that the results get from projects of the same type basically obey the normal distribution, then make linear treatment, then get relevant benchmark graph, using these graphs, development capacity of different software enterprises can be compared, which is very useful for the bidding and progress estimation of project. The paper collects relevant data, makes deeper analysis on Putnam model, and estimates the actual project in second-hand car industry. During using the model estimation, the paper reveals the relations among the source program code length of software project, the development time of software and the workload, with very important theoretical meaning, can help the software administrator to implement effective risk management and decision. The software project estimation methods researched in the paper have verified in many software project, and the results show that these methods have higher reliability and better application potential. But Putnam model also has the properties such as a relatively low correctness, not reflecting the software product, project, participants, and software and hardware resources. In the future estimation process of software project using Putnam model, collecting and accumulating different historical data related to cost estimation, combined by the statistics and analysis of the data to get the model parameter suitable to the organization itself. Software cost estimation is based on probability model, thus it is very difficult to produce precision, but if providing good historical data and using systematic technology, better result can be obtained, completely correct estimation of software cost still needs the continuous efforts from the broad science researchers.

Acknowledgments

The authors wish to thank the Jilin Provincial Education Department"The 12th Five Year Plan" science and technology research project "Research and application based on software cost estimation" (Project No.: JJKHZi [2014] No.144), under which the present work was possible.

REFERENCES

[1]T. C. Jones.*Estimating Software Costs, New York：McGraw—Hill*, **1998**.

[2]B. W. Boehm, R. E. Fairley. *Software estimation perspectives, IEEE software*，v.17, n.6, pp.22-26, **2000.**

[3]B. Londeix *Cost Estimation for Software Development*．*Reading, MA：AddiSon—Wesley*, **1987**.

[4]R. E. Fairle. *Recent advances in Software estimation techniques, In*：*Proc*．*1992 Int'l Conf*．*Software Engineering, New York, ACM Press, pp.*382-391, **1992**.

[5]B. W. Boehm. *Software cost estimation with COCOMO* □．*Englewood Cliffs, NJ：Prentice_Hall*, **2000**.

[6]Boehm BW, Valerdi R. *Achievements and challenges in software resource estimation.Technical Report, No.USC-CSE-2005-513*， **2005.** http://sunset.usc.edu/publications/TECHRPTS/2005/usccse2005-513/usccse **2005**-513.pdf

[7]Boehm BW. *Understanding and controlling software costs. IEEE Trans. on Software Engineering***,**v.14, n.10, pp.1462-1477, **1988.**

[8]R．Agarwal, M．Kumar, Yogesh, et al．*Estimating software projects*．*ACM SIGSOT Software Engineering Notes*, v.26, n.14, pp.60-67, **2001.**

[9]B. W. Boehm. *Software Engineering Economics*, *Englewood C1iffs*，*NJ：Prentice-Hall*, **1981.**

[10]Boehm BW, Valerdi R. *Achievements and challenges in software resource estimation, Technical Report, No.USC-CSE-2005-513*, **2005**.
http://sunset.usc.edu/publications/TECHRPTS/2005/usccse2005-513/usccse 2005-513.pdf