Early estimation of manhours in software development

Ravil Muhamedyev [a], A. I. Kupensheyeva [b]
International IT University, IT department
Manas/Jandosov Str., 34A/8A, Almaty, Kazakhstan,

Received 30 June 2012, accepted 15 August 2012

Abstract. To estimate the labor input of software development, an adaptive system of input estimation based on formalized specification, a model of which includes corresponding correctors that ensures system configuration for improvement of calculation accuracy, was proposed. The results of the technique based on the LFC metrics show that error of evaluation results compared with the actual data is relatively small (at least for one module). The maximum error value is equal to 20%, the average is around 9% to be compared to the actual data.

Citations: Ravil Muhamedyev, A. I. Kupensheyeva. Early estimation of manhours in software development – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **2(13)** 2012 – Pp. 8-12.

Keywords: software engineering; labor effort estimation; formal specification; labor input estimation metrics; estimation methods.
Short title: Estimation of manhours.

Introduction

In order to estimate the effort put into software development various models are used, which are basically based on some form of analysis specification of customer requirements or technical project. In particular, *Function Point* (FP) metrics allow to estimate the effort based on the numbers of the function points has been widely known [1,2].

In our opinion, adaptive evaluation system based on formalized specifications of a customer can be used in established system of development (elaboration). This evaluation system can be named as "early", since it does not include the in-depth analysis methods of the projected system. This system of effort estimation should be based on some important parameters of formalized specifications and additional parameters, which include, for example, qualification of software developer, type of module and etc.

For the assessment of basic applicability of formalized specification the analysis based on actual data was performed. The quantity of words and characters were used as the metrics of formalized specification (LFC).

1. Estimation of labor input of software development

The results of the technique based on the LFC metrics show that error of evaluation results compared with the actual data is relatively small (at least for one module). The maximum error value is equal to 20%, the average is around 9% to be compared to the actual data.

Estimation of labor input of software development is one of the difficult tasks in software engineering. It is solved by different methods [1-5]: expert estimation as well as estimation using models.

1. *Expert estimation.* The method is designed for project estimation with an emphasis on the expert's knowledge and experience. The estimation is conducted for the whole project or its separate parts by the experts.

2. *Estimation using models.* This technique is designed for project estimation with the help of specific models. *Estimation using models* technique has the following types: estimation by analogy; use Case Points (UCP); function points (FP); fast Function Points (ffP); early Function Points (EFP).

2.1. *Estimation by analogy.* Project estimation on the basis of historical data. In fact, it is an automated version of expert estimation method. Project estimation based on its "measurement" of forms, reports, subsystems, essence and etc. Conversion of measurement results to labor effort according to accumulated statistics.

2.2. *Use Case Points* (UCP). The method is designed to evaluate the projects, for which requirements definition is applied with the help of use cases (or precedents). The main point of this technique is to determine "actors" and use cases (precedents) and to estimate (evaluate) its complexity.

2.3. *Function Points* (FP). The method is designed to evaluate the project on the base of a concept called "function

[a]email: *r.muhamedyev@iitu.kz*, corresponding author
[b]email: *ainurka999@mail.ru*

point". Its essence is presented as follows:

a) identification of all information objects and all operations on data exchange between the system and "actors" (users, other systems) and estimation of its complexity;
b) correction of the results - account of non-functional requirements;
c) using the result (in FP) in a COCOMO (Constructive Cost Model) calculator (conversion to labor effort with the partition according to project phases and processes) or conversion to code size, and further to labor efforts.

2.4. *Early Function Points* (EFP). Variety of a Function Points method allowing application in the absence of detailed requirements. The main point of this technique could be formulated as follow:

a) identification of all information objects and all operations on data exchange between the system and "actors";
b) correction of the results - account of non functional requirements.

2.5 *Fast Function Points* (ffP). Variety of a Function Points method allowing application in the absence of detailed requirements. The essence is as follows: - Identification of all information objects and all operations on data exchange between the system and "actors" (users, other systems) without estimation of their complexity (an average value is used) In the case of the established software development process estimation of labor input can be conducted using the statistics based on the results of the previous tasks. Particularly, in this regard, size-oriented metrics has worked well. The given approach described in Ref. [1] is based on LOC (lines of code) estimation by analogy with COCOMO.

Software development input estimation on the basis of formal specification The techniques considered above are not easy to implement, and in some cases, for example, when modifying system modules, their application can be redundant.

Here comes the question: is it possible to move away from the given techniques for input estimation and replace them with the simplified method that allows performing immediate estimation of software development input at an early stage, the stage of task formulation? We assume that, in some cases, in the well-organized development process, in "conservative" environment, where specific statistics of software development is given and the groups of developers are stable, it is possible to use formal specification of software development (FC - formal specification) for calculation of labor input (WD - work days) in man-days.

$$FC => WD \tag{1}$$

In other words, it can be assumed that formal specification can be sufficient for relatively accurate labor input calculations of software development.

In this case, the task is to find a function fwc

$$WD = fwc(FC, P), \tag{2}$$

where P is a vector of additional parameters, including, for example, a developer's qualification, a type of the designed module and etc. It is supposed that in some instances it is possible to perform convolution of P vector to the correction factor that depends on the specified parameters.

The general model of a customizable (adaptive) system of input estimation is shown in Fig. 1. Block 1 is a block for definition of significant parameters of formal specification, block 2 is a block for definition of a vector of additional parameters, block 3 is a module for definition of fwc function, block 4 is used for calculation of predicted input, block 5 performs a comparison of actual indicators of labor input with the calculated ones, Kfc, Kp, $Kfwc$ are corresponding correctors that ensures the system configuration to improve an accuracy of input calculations. The factor analysis, clustering and pattern recognition algorithms, and also algorithms for semantic analysis can be applied in order to construct the correctors.

The experiments with an actual data were conducted in order to verify the possibility of using formal specification for software development input calculations. At the same time, the number of words and symbols (which, in this case, plays the role of significant parameters) is used as a simple metrics of formal specification.

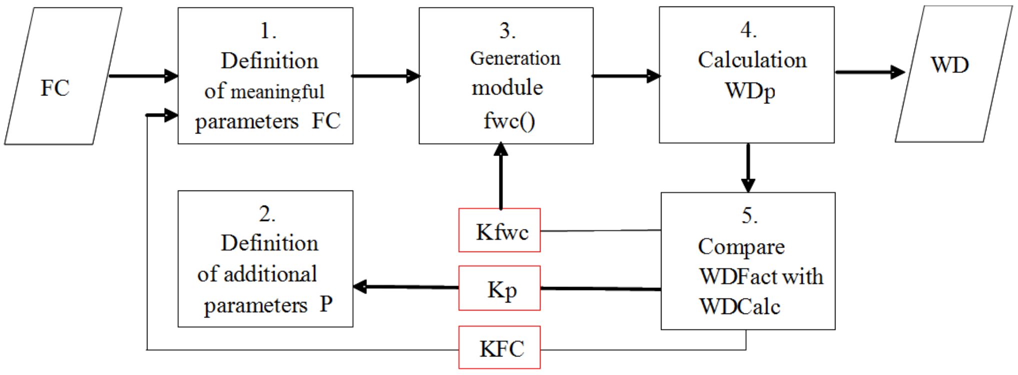
FC
1. Definition of meaningful parameters FC
3. Generation module fwc()
4. Calculation WDp
WD
2. Definition of additional parameters P
Kfwc
Kp
KFC
5. Compare WDFact with WDCalc

Fig. 1. The generalized model of adjusted (adaptive) system of an assessment of labour input.

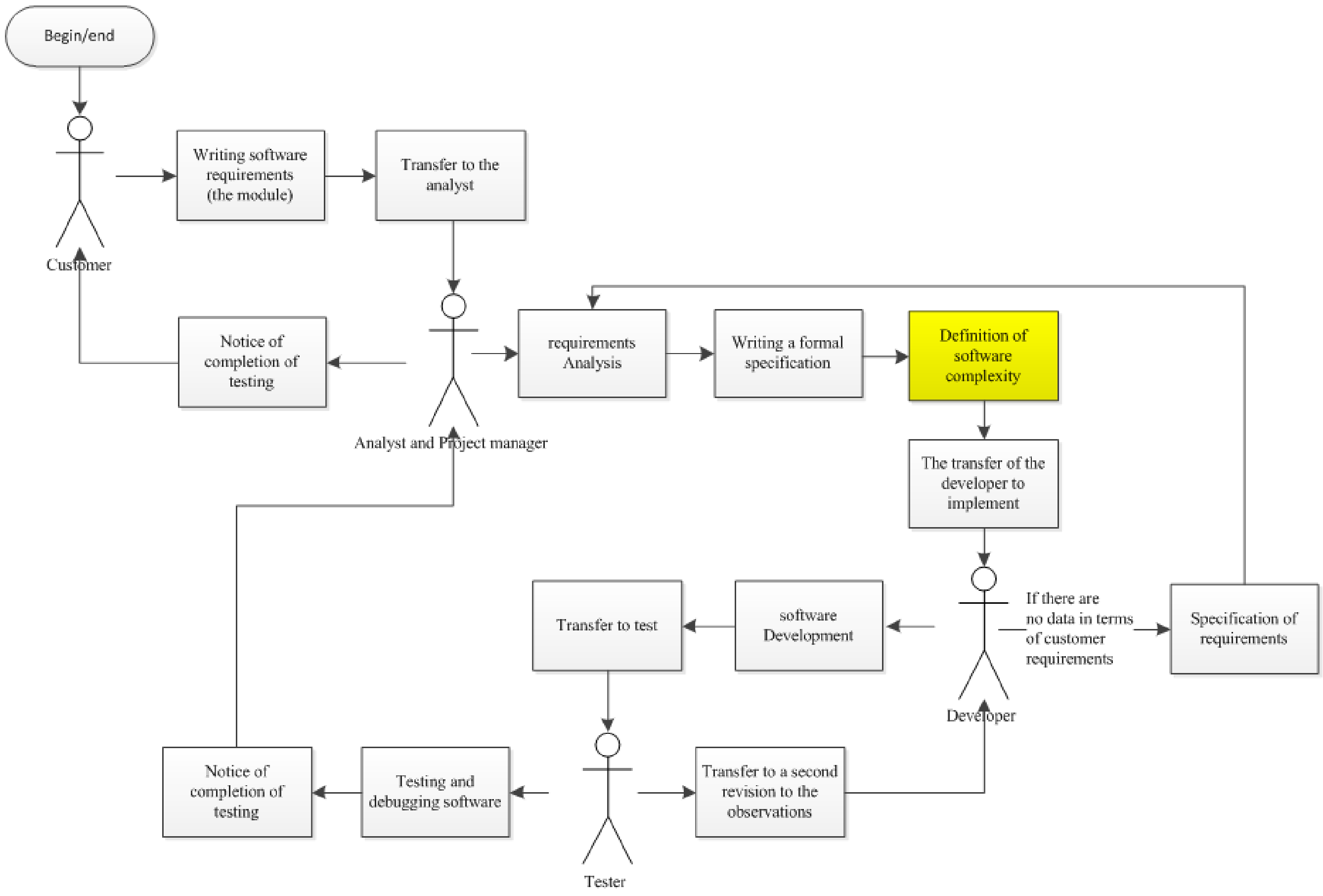
Begin/end
Customer
Writing software requirements (the module)
Transfer to the analyst
Notice of completion of testing
Analyst and Project manager
requirements Analysis
Writing a formal specification
Definition of software complexity
The transfer of the developer to implement
Developer
If there are no data in terms of customer requirements
Specification of requirements
software Development
Transfer to test
Tester
Testing and debugging software
Notice of completion of testing
Transfer to a second revision to the observations

Fig. 2. The development and modification of software in the organization of "HB".

In this instance, the expression (1) can be converted to the following form:

$$WD = LFC(w, c) * p, \tag{3}$$

where LFC is a function depending on the number of words w and symbols c in formal specification, p is a correction factor. In this case, a correction factor is a convolution of P vector.

At first glance, the described approach looks very formal, independent of the semantic content of specification to a software product. However, it can be recalled that the structure of formal specification developed by an expert gives necessary substantial sense to the whole technique.

An example of LFC metrics and formal specifications application for estimation of software development labor input

The applicability analysis of LFC metrics and formal specifications was conducted based on the data accumulated in an organization with an "HB" label. In this organization software development and modification is done as follows - see Fig. 2.

At first, an application form written by a customer in a free form and related to the specific module of a system comes to an analyst.

Based on this application the analyst writes formal specification and passes it to the developer.

At this point, on the basis of the given specification the developer in the "HB" organization estimates his/her labor input. The calculation of labor effort is not formalized and based on expert estimation. But, as shown in Fig.2, it is proposed to calculate efforts at an analysis stage, without bringing up an application to the developer. To formalize the process of effort calculation empirically dependence

$$LFC = f(w, c) \tag{4}$$

was found in the following form where $w1$ and $w2$ are the number of words in the application of the customer and analyst, respectively; $c1$ and $c2$ are the number of symbols in the application of the customer and analyst, respectively.

As a result formula (2) has the following form

$$WD = LFC * p, \tag{5}$$

In order to get the p factor, data about WD accumulated in the development process can be used. In other words, for each application

$$P_i = \frac{WD_i}{LFC_i}, \tag{6}$$

where WD_i is the developer's labor effort for i-th application, LFC_i is LFC calculation using formula (3) for i-th application.

Table 1. Quantitative characteristics of applications for the module “Credits”.
CA - Customer Application; AA - Analyst Application; $w1$, $w2$ - number of words, $c1$, $c2$ - number of characters.

S/n	Name of application	WD_i fact.	CA $w1$	CA $c1$	AA $w2$	AA $c2$
1.	Unblock	11	612	6685	568	4917
2.	Stepped rate	15	853	7657	1058	7208
3.	Calculation of the duration	5	240	1715	292	2564
4.	Cancellation of deposit	13	688	6518	764	6087
5.	Changes mortgage	4	234	1981	319	2722
6.	Delivery of documents	4	127	1114	418	3372
7.	Calculation of the homogeneity	9	130	1146	812	8205

Table 2. The results of calculations of labor input required for application processing by the developer according to the module “Credits”.

S/n	Name of application	LFC	P_i	P	WD_i (per.-min)	WD_i (per.-day)
1.	Unblock	12782	0,38	0,44	5624,08	11,72
2.	Stepped rate	16776	0,43	0,44	7381,44	15,38
3.	Calculation of the duration	4811	0,60	0,44	2116,84	4,41
4.	Cancellation of deposit	14057	0,68	0,44	6185,08	12,89
5.	Changes mortgage	5256	0,27	0,44	2312,64	4,82
6.	Delivery of documents	5031	0,48	0,44	2213,64	4,61
7.	Calculation of the homogeneity	10293	0,23	0,44	4528,92	9,44

Table 3. The comparison of actual and calculated data of labor input required for applications processing by the developer.

S/n	Name of application	Module	WD_i (per.-day) fact.	WD_i (per.-day) calc.	$\Delta(WD_i)$ (per.-day)	$\varepsilon(WD_i)$ %
1.	Unblock	Credits	11	11,72	0,72	6,55
2.	Stepped rate	Credits	15	15,38	0,38	2,53
3.	Calculation of the duration	Credits	5	4,41	-0,59	11,8
4.	Cancellation of deposit	Credits	13	12,89	-0,11	0,85
5.	Changes mortgage	Credits	4	4,82	0,82	20,5
6.	Delivery of documents	Credits	4	4,61	0,61	15,25
7.	Calculation of the homogeneity	Credits	9	9,44	0,44	4,89

The generalized factor P is calculated as an average for all P_i.

Let us consider the application of the given technique in the “HB” organization when modifying modules in one system.

Formal specification for software development (modification) was designed for the use of the technique. Formal specification is based on GOST34 [8].

Table 1 contains the source data for calculation ($w1$, $c1$, $w2$, $c2$)

According to the data given in Table 1, the following results are derived:

Based on the first application “Unblock” LFC = 612+6685+568+4917 = 12782

$$P_i = \frac{WD_i}{LFC_i} = \frac{11*60*8}{12782} = 0.41. \tag{7}$$

where 11 is a runtime of the application in man-days. For accuracy improvement, calculations are done in minutes with account of 8-hour workday.

An average value of a correction factor for the module “Credits” calculated according to 7 applications is P=0.44.

Using the values of p and data from Table 1, the developer’s labor effort (WD_i) is calculated according to 7 applications (Table 2).

Table 3 presents the data that allows comparing the results of actual input required for application processing with the calculated data. The absolute error of labor input required for application processing is calculated by the formula:

$$\Delta(WD_i) = |(WD_i^{actual.}) - (WD_i^{calc.})| \tag{8}$$

The relative error of labor input:

$$\varepsilon(WD_i) = \frac{\Delta(WD_i)}{WD_i^{actual}}. \tag{9}$$

The error between actual $WD_i^{actual.}$ and calculated $WD_i^{calc.}$ is insignificant as shown in Table 3.

The results of using the technique based on LFC metrics show that the error of the results of input estimation compared to actual data is relatively small (at least for one module).

Let us note that the attempts of using the specified approach for labor input estimation required for applications processing that affect multiple modules have not been successful. Therefore, the described technique can be applied only in case of severe restrictions on the subject area and in the presence of reliable data about actual labor input of software development.

Conclusion

To estimate the labor input of software development, an adaptive system of input estimation based on formalized specification, a model of which includes corresponding correctors that ensures system configuration for improvement of calculation accuracy, was proposed.

The analysis based on the actual data was conducted in order to evaluate principal applicability of formal specification. At the same time, the quantity of words and symbols were used as the metrics of formal specification (LFC).

The results of using the method based on LFC metrics show that the error of results of labor input estimation compared to the actual data is relatively small (at least for one module). The maximum value of the error is 20%, the medium is about 9%.

The attempts of using the specified approach for labor input estimation required for applications processing that affect multiple modules have not been successful. Therefore, the described technique can be applied only in case of severe restrictions on the subject area and in the presence of reliable data about actual labor input of software development.

Let us note that although the proposed metrics is formal, it can be recalled that the structure of formal specification developed by an expert gives necessary substantial sense to the whole technique.

Further development of the approach can be a system that computes the correction factor p by the actual results of design and self-defining type of function fwc. The possible approaches can be the factor analysis, clustering and pattern recognition algorithms, and also algorithms for semantic analysis.

References

1. Lipaev V.V. Software Engineering. Methodological Foundations. – Moscow: TEIS, 2006. - 680 p.
2. Orlov S.A. Technologies of development of the software. – St. Petersburg: Piter, 2004. - 527 p.
3. Lipaev V.V. Debugging complex software: Methods, tools, technology. – Moscow: Energoatomizdat, 1993.
4. Boehm B.W. Software Engineering. – Moscow: Radio and Communications, 1985.
5. B. Boehm and R. Turner. Using Risk to Balance Agile and Plan-Driven Methods. – *Computer*, June 2003, 57-66.
6. A. Abran and P.N. Robillard. Function Points Analysis: An Empirical Study of its Measurement Processes. – *IEEE Transactions on Software Engineering* 22 (1996) 895-909.
7. S. Arkhipenkov. Lecture N6. Evaluation complexity and timing of software development, 2009. – <http://citforum.ru/SE/project/arkhipenkov_lectures/11.shtml>, accessed 2012 03 05.
8. Development of documentation in accordance with GOST. – <http://www.rugost.com/index.php?option=com_content&task=category&sectionid=7&id=25&Itemid=62>, accessed 2012 03 05.