Measurement of Workflow System Development

Josef Fiala¹, Jan Ministr¹, Jaroslav Ráček², Jana Sedláčková²

Abstract
Papers deals with cost estimation of workflow system development. A measurement of source code lines (COCOMO – Constructive Cost Model), function oriented metrics (FPA – Function Point Analysis) or metrics based on measurement of individual process activities (ABC – Activity Based Costing) can be used for this purpose. If the workflow system is mail based or document based then should be used COCOMO or Function Point Analysis. In the case of process based system the cost of process analysis and cost of process definition development can be estimated by using principles of Activity Based Costing.

1. Introduction
Papers aims to see on workflow system implementation through the eyes of future user, ergo firm which is in developer position or user which is shopping and implementing the offered software solution. Implementation a new information system is complicated process which has considerable risk and requires careful planning and high rate of responsibility by deciding. It is necessary to specify the cost of project in its start. It is reality for new software development or for offered package implementation. The project budged and activity plan includes also data about software product which has been developed. In other words it is necessary to measure the resultant product often in time when the development of product is not finished and this fact has influence at preciseness of estimation.

2. Workflow System Types
In case of information systems which have build inside the support of firm workflow can to come some situation types. The way of determine the estimation of size implemented system is in accordance with types of situations.
One of factors, which has influence on the way of determine the estimation is the type of system. It is necessary to differentiate between document and process based workflow. Document oriented systems have ability of communication with external applications. They are suitable to administrative workflow (supporting and repeated processes). These systems are usually designed as three layer architecture which contains presentation, application and data layer. Data of these systems can be share with other information systems. In point of design and implementation view this conception WfMS (Workflow Management System) is not different from majority of information system which firms uses.
Over again the process oriented systems contain the definitions of firm processes according these are established and controlled single process instances. Process oriented WfMS can imagine as “Process Inter pret” which has the wide scale of interface and iterates with others applications. This workflow use own communication canals and databases which are not directly access able to other applications. Process oriented systems are usually suitable in production workflow (simultaneously decision making

¹ Faculty of Economics, Technical University of Ostrava, Czech Republic
² Faculty of Informatics, Masaryk University in Brno, Czech Republic.
and repeated processes). Architecture of these processes often comes out from WfMC (Workflow Management Coalition) standards where the base made handled WfMS which works with imported process definitions that are designed concrete needs of firm [Hol95, Hol99].

3. Methods of Cost Calculation

Another factor that has influence on way of determine the estimation the size of system is availability of information about the size the source code and other source data. That means configuration files or process definitions. If are available the data about the size of source files that means that solvers of project are able to determine this size. In others words its means that we can go out from consideration that laboriousness, cost and time of creating the new project are directly proportional to the size of source codes. Then can use for calculate these quantities for example the COCOMO (Constructive Cost Model) method that first time B. Boehm published in 1981 and he several times modified the method and so this method of laboriousness estimation is useable for present techniques of software development [Boe81]. Calculating by help COCOMO or other similar model which come out from size of source files is suitable for new developed systems which are based on three layer architecture.

Case when is COCOMO model not useable is situation when designers are not able to determine size of source files. That happens usually when the team has lack of opportunities from previous similar projects. Especially in first phases of software solution are the determining the size of source files substantially hard and not accurate. In this case it is offered another way of determine the size of software by help the required functionality which should be known before the start of system development. Probable the widespread technique in this type is the FTA - Function Point Analysis. It is metrics that describes the software functionality from point of end user view and it completely covers the point of developer view. The principle of this technique is based on evaluating the single user functions which system will offer to a relevant number of production units (functional points). Then it is determined the laboriousness one unit [Jon91]. Function Point Analysis was originally determined to define the estimation a new developed systems based on three layer architecture. Principle of function point can be used to evaluate the workflow processes.

By forecasting the laboriousness and costs of implementation the process oriented systems are usually separately make forecasting the implementation of WfMS and creating and implementing of process definitions. In case of implementation the WfMS it is usually implementation of finished commerce offered product. Then the price and time of implementation are determined by the conditions of the provider contract.

The separate activity in frame of project planning is laboriousness define of creating and implementing process definitions in to realized WfMS. There are existing two basis variants of procedures. The first variant comes out from size of source code (size of process description). In this case it has to disposition data about amount and size the firm processes. This information is for disposal in time when the firm has the processes mapped. Than is possible to evaluate the laboriousness of implementation the single processes by help the ABC (Activity Based Costing) method. It is means the implementation of single activities and from this it is possible to deduce the laboriousness of whole systems.

In situation when the firm has not mapped processes it can use for implementation the estimations based on function point principle. Method of function points can be used both for the estimation the laboriousness of process definition implementation and for the estimation the mapping of processes.
4. COCOMO Method Assessment

Assessment of laboriousness and costs of the software development by COCOMO method comes out from idea that effort $E$ (number of spend personal-months) and time $T$ (time of project duration) are proportional to a size of the source code. In this way is possible to estimate financial costs of the project. The lack of this approach is particularly the accuracy of first assessment of product size which is presented in KSLOC (thousands of source code). The assessment of size of source code can greatly differentiate from reality (Boem present even 4x) in both directions. The accurate assessment of future source code is not possible without the previous opportunities and other source of empirical data.

Before own calculation it is necessary to set other parameters of model which are detail level of the calculate model and used development mode. The level of the detail says if and in how scale will be to include other factors than size of source code. It exist three detail levels which are basic model, middle model and advanced model. Basic model uses only size of code. Middle model takes into consideration other aspects of project. Advanced model uses the same attributes as middle model and moreover takes to consideration the phase of life cycle of project. Detail level takes in consideration the conditions in which is software developed the own complexity of developed product will be displayed in elected developed mode. In this place origin COCOMO also differentiate three levels which are organic mode, immediate mode and bounded mode. Organic mode is preferred in preparation of simple, good resoluble projects by where are minimum risks and size of product is to 50 KSLOC. Immediate mode is used by middle extensive projects that have usually to 300 KSLOC and where good understanding of requirements of users and it are is not impending big risks. In case extensive project (over 300 KSLOC) or risks project it is elect bounded mode. Basic relations for calculate of effort and time have next form.

$$E = a \times (\text{KSLOC})^b \quad (1)$$

$$T = c \times E^d \quad (2)$$

Symbols $a$, $b$, $c$, and $d$ represented empiric values of parameters which are elected according level of model and used development mode from predefined tables. Concrete for basic model $a \in [2,4 ; 3,6]$ for middle and advanced model $a \in [2,8 F_c ; 3,2 F_c]$. Others values are moving in next boundaries $b \in [1,05 ; 1;20 ]$, $c = 2,5$, $d \in [0,32 ; 0,38 ]$. The taking to consideration other factors of project than size of code are in middle and advanced model masked in correction factor $F_c$, which is the product of next 15 values which describes software attributes of product, hardware attributes, attributes of development team and attributes of project. In normal conditions are every from these 15 values equal 1. If has the given attribute influence to increase the cost then grows its value. If it is the attribute to its value is not necessary to take into consideration than its value vice versa go down. COCOMO offers tables which contain concrete values of attributes and defined the way of its determination.

Model COCOMO passed over its issue the big development. Nowadays it exist not only origin (above described version) but even the version for incremental development cycle, version which takes in consideration periods of development or version which uses statistical assessment KSLOC and quantitative approach to uncertainty. COCOMO can be use to assessment of costs by modification the existing applications.

The most used application now is COCOMO II from 1995 year which uses three models which are APM (Application Composition model) for projects which use modern tools and GUI, EDM (Early Design Model) for general assessments in introduction phases when is the architecture of product developed and PAM (Post Architecture Model) for assessments when the architecture is specified. COCOMO II uses also new attributes of project which usually come out from combination early used attributes.
5. **Assessment by Function Points**

If it not to disposal the assessment of future size code then COCOMO system is not useable. In this case is possible the extent application and consecutively also laboriousness its creation to estimate by help of function points. Function points are normalized metrics of software project which measure the application area (application function and data) and it not investigates technical area (not measuring the code). With dash of exaggeration can to say that function points present unit of production by process of software development. In this way we can to estimate the laboriousness of development not only the new products but also the laboriousness of modification the existing systems.

At first is necessary to calculate “not adjusted points”. These are associated with transaction functions (EI – External Inputs, EO – External Outputs, EQ – External Enquiry) or with data functions ((ILF – Internal Logical Files, EIF – External Interface Files). All identified EI, EO, EQ, ILF and EIF applications are sorted into groups according its type and complexity. Number of elements every group is multiplied by relevant weight and then is all added together. Result of total adding gives number of not adjusted functional points \(NFP\).

The procedure of calculation the functional point accurate determines coefficients of single weights and procedure how identified points to separate to single groups. Furthermore it is find out 14 factors (general characteristic of system) as for example rate of required credibility of back up data and revive or rate required on-line data inputs. Every factor is evaluated with integer number from 0 to 5 according degree of influence on application. Resultant number of adjusted functional points \(FP\) is possible to obtain from relation.

\[
FP = 0.65 \times 0.01 \sum F_i \times NFP
\]  

Symbol \(\sum F_i\) presents the total of 14 system characteristic evaluations. Then is necessary to determine the laboriousness and price one of function point and then to determine the total costs of software project.

Adjusted functional points can to use as input value for calculate by COCOMO methods. Caper Jones presents coefficients according these can functional points approximately recalculate to number of commands in elected programming language [Jon91]. To one functional point matches circa 13 commands in SQL language, 64 commands in C++, 128 commands in C or 320 commands in assembler.

Functional points can to use also to next assessments. According the statistics the value \(FP^{1.15}\) forecasts approximately number of paper documentation pages of project, \(FP^{1.2}\) approximately number of testing examples, \(FP^{2.25}\) approximately potentiality of errors by new software projects, \(FP^{0.4}\) approximately plan of development in calendar months, \(FP/150\) approximately number of employees needed to solution of application, \(FP/750\) forecasts approximately number of employees to maintain of application.

Great differences can be discovered also in productivity of single development teams. It depends on experiences, equipment. Not experienced team which not uses structured methods and CASE tools has according Jones productivity 2.5 FP in person-month but. Team with experience with structural methods and CASE tools have approximately FP on level 40 in person-month.

6. **Costs of Process Definitions Creating**

COCOMO and FPA are methods which can use with success to estimate the laboriousness and costs of information system developing based on three level architecture. From point of dividing the workflow systems view there are document and mail based workflow systems which executed administrative, collaborate and ad hoc workflow [Car03]. Keys firm processes are usually come under the production workflow area and they are supported by process based workflow systems. These systems come out from architecture which is defined workflow reference model [Hol95]. The costs of creating included not only the cost
of obtaining programs but also costs of process definition creating. Single process definitions are according David Hollingsworth suitable to associate greater workflow models which are defining whole workflow of the firm [Hol99, Rac02].

Workflow model is description of firm workflow which enables its automatic handling as is for example modeling or controlling execution through workflow system. Workflow model contains particularly the description of activities, actors, applications and data.

By estimate of size the process definitions can to come out from principles of Activity Based Costing method. If the identification of basic entities which made workflow model is successful and also the assessment of code size of their description is successful too (for example in XPDL format) is possible to determine whole code size of workflow model WfSLOC.

\[
WfSLOC = A_a N_a + A_t N_t + A_u N_u + A_p N_p + A_d N_d
\]

Symbols \(A_a, A_t, A_u, A_p, A_d\) presents average numbers of attributes which are used in workflow model for description of activities, transitions, actors, applications and data.

With this assessment of codes size which is obtained in this way can be more work similar as with assessment of KSLOC that is used in COCOMO model. It means the substitution to relations (1) and (2). Values \(a, b, c, d\) will be another. Nowadays are lack of empiric information about and their accurately estimation. Also the next attributes of project as for example existing the process maps have to be take in considerate by assessment the effort \(E\) and time \(T\) in calculate of the correction factor \(F_c\).

7. Conclusions

Workflow system concept is much spread. By taking solution how way of cost assessment will be used for concrete workflow system is need to determine the type of system. In case of systems which are based on document sharing or mail sharing (exclusive) comes de facto about classic information systems where is the support of processes masked in the system functionalities. In these cases is possible the costs to estimate by help of COCOMO model or AFP (if is lack of solid code size assessments).

In case of process oriented systems which are based on workflow reference model principles the system is composed from two parts. The first part presents own software which controls process executing. Here is possible to use COCOMO or AFP again. Second part of the system made workflow model which is associating all of firm process definitions. Assessment of size the workflow model can execute by techniques which use ABC method principles and ten calculate needed effort and time in similar way as in COCOMO.

Bibliography


