Security Factors in Effort Estimation of Software Projects

Jana Sedláčková[*]
Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
xsedla23@stud.fit.vutbr.cz

Abstract

This contribution deals with problems related to an effort estimation of the software projects which are connected to the development of secured products. The aim of this contribution is to present an option of an effective consideration of the effort which is connected with the software products development. The FPA and COCOMO II methods for cost estimation are extended by the security factors, which are considered depending on requested security level of the resulting software product. The FPA method extended by security factors is called FPA&SF (Function Analysis & Security Factors) and the extended COCOMO II method is named as COCOMO II&SF (COCOMO II & Security Factors). An example of determination of the effort estimation by new methods FPA&SF and COCOMO II&SF is part of this contribution. These methods are used on an actual project and the acquired results are compared with values which were evaluated after a successful completion of the relevant project.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

FPA, COCOMO II, Effort Estimation, FPA&SF, COCOMO II&SF, Product Security, Evaluation Assurance Level - EAL

1. Introduction

A development of software project is a complicated process which brings along considerable financial risks and requires an in-depth planning and a great amount of responsibility during the decision making. It is necessary to estimate the effort for project realization as accurately as possible already in the initial phases of the development, and determine the project cost from this value subsequently. The main goal of the estimation is the accuracy, because based on the acquired values, the basic development plan or the price for the resulting software product, which is submitted to the customer, is determined. There are several approaches and methods for estimating the cost of software projects. One of the most important ones must surely be the method of function point analysis (FPA) and another commonly used method is COCOMO II.

The system security is another very important topic connected with software development, specifically with development of information system with the usage of IT. At present the information systems security field is constantly evolving. Information systems often store or process a very sensitive data, information and knowledge which must be secured properly from an unwanted monitoring or attacks. A design and subsequent implementation of appropriate security mechanism must be part of the development of such information system. The cost for implementation of security mechanisms form a significant part of the total cost in case of some information systems with higher security demands. The cost for ensuring the information systems security on requested level should therefore be considered already during the phase of project development cost estimation [1].

However, none of the available methods for cost estimation of project development considers cost connected with ensuring the resulting software product security during the estimation. The methodology of the FPA and COCOMO II methods extended by security factors will be presented in the following chapters.

2. Methods of cost estimating

There are various methods of estimating the cost of software projects development. Among the most famous and commonly used ones belongs the model COCOMO II, which is based on the knowledge of the developed product size, and Function Point Analysis (FPA), which is based on description of the final product functionality [7], [6].

[*] Recommended by thesis supervisor:
Assoc. Prof. Jitka Kreslíková
Defended at Faculty of Information Technology, Brno University of Technology on November 10, 2010.

Sedláčková, J. Security Factors in Effort Estimation of Software projects. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 2 (2011) 12-17

2.1 FPA

The method was invented by Allan Albrecht in 1983 and the organization IFPUG (International Function Point User Group) is engaged actively in further evolvement of this method since 1986. The method leaves out the problems related to determination of expected code amount. Function points are a normalized metrics of software project, which measure an application field and does not consider a technical field. At the same time, it measures application functions and data, and does not evaluate the source code [10].

Function points calculation process according to IFPUG [5] consists of the following steps:

1. Identification of the subsystem boundaries
2. Identification of the data functions (internal logical files ILF and external interface files EIF)
3. Identification of transactional functions (external inputs EI, external outputs EO and external queries EQ)
4. Calculation of the Unadjusted Function Point (UAF) Count
5. Determination of the Value Adjustment Factor (VAF) using General System Characteristics (GSC)
6. Calculation of the final Function Point Count

- ILF Internal Logical Files: Any potentially unrestricted data sequence generated, used or maintained by an application can be considered as a logical file.
- EIF External Interface Files: Similar to FILE, however, the given logical file is shared by several programs. It includes each large group of user data or leading information used in an application. This information has to be maintained by a different application.
- EI External Inputs: These input statements concern only user input orders, which are related to changes in the internal data structure. They do not concern user input orders, which are aimed only at control of program implementation.
- EO External Outputs: The calculation scheme is similar to that ones related to the input orders. All unique user data or control data leaving the external frontier of the measured system count as output orders.
- EQ External Queries: Orders in the form of enquiries are related to outputs carrying out the program implementation and do not change the internal data structure. EQ is similar to EI and EO under the condition that these are enquiries in the form of question.

Depending on the environment, in which the given system is being developed, there are general system characteristics specified – the value adjustment factor (VAF) is determined [5], [4], [6].

The equation for determination of influence of the general system characteristics

$$VAF = (TDI * 0.01) + 0.65 \tag{1}$$

Where TDI is a factor of technical complexity, so called resulting degree of influence. It concerns a calibrating parameter of effort, which indicates an influence of all the 14 factors. Each of them is rated by a six-point scale (0 – 5) according to a relevant degree of influence (DI) on application.

$$TDI = \sum_{i=1}^{14} DI_i \tag{2}$$

Evaluated system characteristics:

- Data communications
- Distributed data processing
- Performance
- Heavily used configuration
- Transaction rate
- On-line data entry
- End-user efficienc
- On-line update
- Reusability
- Complex processing
- Installation ease
- Operational ease
- Multiple sites
- Facilitate change

An estimation of the code size and a value of total costs for developing software project can be determined from the calculated adjusted function points. At first, it is necessary to specify the effort and a price of one function point. Based on this, the total costs necessary for the project development can be determined. Depending on the programming language in which the software project is carried out, the size of the source code, which corresponds to one function point, is assessed.

The estimation of code size for given programming language [5], [6]:

$$LOC = FP * \frac{LOC}{FP} \tag{3}$$

The effort of new project is estimated as a share of number of points for new project divided by number of points per months:

$$E = \frac{FP}{\frac{FP}{M}} \tag{4}$$

Where $\frac{FP}{M}$ is an average amount of function points falling on one person-month and it is determined from the finished projects.

An estimation of project development duration in months can be calculated by relation [9]:

$$T = FP^{0.4} \tag{5}$$

2.2 COCOMO II

The model was developed by Barry Boehm in 1995 as an extension of model COCOMO 81, which had been no longer sufficient for determination of price estimation of newer and more demanding software projects. COCOMO II consists of 2 models – Early Design Model (EDM) and Post-Architecture Model (PAM) [6].

EDM is used in the initial phases of project, when only the software architecture is being designed and detailed information about the actual software and its overall development process are not yet available. PAM works in more details and is used in the phase when an in-depth specification is finished and the software is ready for its development. The model PAM is primarily significant for the purpose of this contribution therefore it is described in more details [3], [2].

2.2.1 Post-Architecture Model

The model contains a file of 17 effort multipliers and file of 5 scale factors. Multipliers have 6 possible levels of evaluation (very low, low, normal, high, very high, and extremely high). Every evaluation corresponds with a positive number, which is determined by calibration from previous projects.

Scale factors [2]:

- PREC Precedentness
- FLEX Development / Flexibility
- RESL Architecture / Risk resolution
- TEAM Team cohesion
- PMAT Process maturity

Scale factors [2]:

- RELY Required software reliability
- DATA Data base size
- CPLX Product complexity
- RUSE Required reusability
- DOCU Documentation match to life-cycle needs
- TIME Execution time constraint
- STOR Main storage constraint
- PVOL Platform volatility
- ACAP Analyst capability
- PCAP Programmer capability
- PCON Personnel continuity
- AEXP Experience with application
- PEXP Platform experience
- LTEX Language and tool experience
- TOOL Use of software tools
- SITE Multisite development
- SCED Required development schedule

The equation of effort calculation:

$$E = A * KLOC^B * EM \tag{6}$$

where A is a constant, whose value is estimated around 2.94 and EM is calculated from 17 effort multipliers,

$$EM = \prod_{i=1}^{17} EM_i \tag{7}$$

$$B = 0.91 + 0.01 * \sum_{i=1}^{5} W_i \tag{8}$$

where W_i are scale factors with relevant degree of influence.

The equation for estimation of preliminary plan of project development duration:

$$T = [C * \bar{E}^{(D+0.2*0.01*\sum_{i=1}^{5} W_i)}] * \frac{SCED\%}{100} \tag{9}$$

where T is a calendar time in months from definition of basic project requirements to approval of all activities, C is a constant, whose value is around 3.67, E is estimated person-months (PM) without effort multiplier SCED, D is a constant, whose value is around 0.28, b is calculated according to formula stated above, SCED% is percentage of required plan compression/expansion.

3. Software Security Criteria

Common Criteria are used for evaluation of information technologies security and are defined by an international standard ISO/IEC 15408. The criteria specify Evaluation assurance level, which compose of special-purpose sets and serve to an even arrangement of the individual security requirements resulting in a balanced perspective on the extent of confidence in the correctness of the considered product.

Security levels in accordance with the standard ISO/EIC 15408 (Criteria for evaluation of information technologies security) [8]:

EAL1: Functionally Tested - the lowest assurances level, where the aim is to provide a certain amount of trust without using a more complex analysis of security risks. The base of the assurances are a functional specification, interface definition and processing of security documentation.

EAL2: Structurally Tested - compared to EAL1, it extends the requirements by an independend testing. At the same time it is necessary to extend the product development by an informal architecture discription and deal with the commonly known product security attacks. The level provides low to medium independently verified security in case that a complete information from the development phase is not available.

EAL3: Methodically Tested and Checked - enables application of the maximal assurances which are based on a proved approaches to development process without a significant increase of effort. Further, on contrary to EAL2, a wider testing of security functions and mechanisms, development environment monitoring and ensuring the configuration administration is requested. The level is recommended as a medium level of security verified independently.

EAL4: Methodically Designed, Tested, and Reviewed - is based on a quality development methods which are highly reliable, however they don't require an extensive involvement of specific knowledge, skills or resources. An independent vulnerability analysis must prove a resistance against attackers with low attack potential.

EAL5: Semi-Formally Designed and Tested - during development, this level requires usage of special methods and techniques of security engineering in medium extent. The assurances are based on an existence of formal security model which is completed by semi-formal representation of the overall design and concealed channels analysis. The resulting products can resist attacks that use the sources with medium performance.

EAL6: Semi-Formally Verified Design and Tested - is based on a modular and layered design. The resulting product is highly resistant to attacks and is developed in a strictly managed and controled environment. The concealed channels analysis must be systematic.

EAL7: Formally Verified Design and Tested - the most demanding level which is based on a fully formal design. The design complexity is deliberately decreased. A full formalization, a formal model of security policy, a formal presentation of functional specifications and global design, a semi-formal detailed design, and a formal and semi-formal corespondence presentation are required.

4. Extension of FPA and COCOMO II methods by security factors

The FPA method extended by security factors is further called FPA& SF (Function Analysis& Security Factors) and the extended COCOMO II method is further named as COCOMO II&SF (COCOMO II& Security Factors).

4.1 FPA&SF

A 15^{th} factor - product security factor - is added to general system characteristics which are described in chapter 2.1. By adding this factor, the formula for calculation of value adjustment factor (VAF) is modified. To specify it more precisely, the calibrating parameter of effort (TDI= factor of technical complexity), which is now represented by influence of 15 factors of general system characteristics, is extended.

$$VAF = (TDI * 0.01) + 0.65 \tag{10}$$

$$TDI = \sum_{i=1}^{15} DI_i \tag{11}$$

Each of the general systems characteristics factors is rated by a six-point scale (0 - 5) according to the relevant degree of influence (DI) on application. The added 15^{th} factor is rated by the same rules.

4.1.1 Method of assessing the influence of product security factor

The assessment of product security factor influence to the system is determined according to the level of assurances that must be accomplished by the resulting software product. The Common Criteria, the criteria for evaluation of information technologies security which are defined by ISO/IEC 15408 standard, are used as criteria for assurances evaluation. A detailed description of the criteria including relevant levels of assurances can be found in the chapter 3.

Table 1: Determination of influence of product security factor

Degree of Influence	Determination of influence
0	EAL1
1	EAL2, EAL3
2	EAL4
3	EAL5 (EAL4)
4	EAL6
5	EAL7 (EAL6)

4.1.2 Determination of influence degree of product security factor

A degree of influence of product security factor on the system is determined on the basis of the individual classes (levels of assurance evaluation), see table1. The degree of influence was assigned to the relevant classes on the basis of information acquired from projects finished previously. See table 1.

4.2 COCOMO II&SF

By the extension of COCOMO II method it is understood that there is a new 18th effort multiplier of product security - SECU added to the Post-Architecture Model (PAM) which is specified in chapter 2.2. The effort multiplier is included in attributes of the software product. The formula for effort calculation (E) is modified by adding the new effort multiplier. To be more precise, the calculation of the total effort multiplier (EM), which is now calculated by product of 18 components of effort multiplier, will be extended:

$$E = A * KLOC^B * EM \tag{12}$$

$$EM = \prod_{i=1}^{18} EM_i \tag{13}$$

The constant A and variable B will remain unchanged. All the effort multipliers, including the newly added one (SECU), have 6 possible levels of evaluation (very low, low, normal, high, very high, and extremely high). Evaluation of degree of SECU influence is presented on Table 2.

4.2.1 Method of influence assesment

The assessment method of influence in COCOMO II&SF is the same as in case of FPA&SF, i.e. depending on security level in accordance with the standard ISO/EIC 15408 (Criteria for evaluation of information technologies security). See chapter 4.1.

4.2.2 Determination of influence degree

A degree of influence of product security factor on the system is determined on the basis of the individual classes (levels of assurance evaluation). See table 3.

5. Experimental results acquired by FPA&SF and COCOMO II&SF methods

Project - Internet Banking:
A web application provides the clients of a bank institution a direct access to control defined products. The

Table 2: Degree of SECU influence evaluation

Attribut	Very Low	Low	Normal	High	Very High	Extremely High
SECU	0.90	0.94	1.00	1.19	1.36	1.88

Table 3: Determination of influence of SECU multiplier

Degree of Influence	Determination of influence
Very Low	EAL1
Low	EAL2, EAL3
Normal	EAL4
High	EAL5 (EAL4)
Very High	EAL6
Extremely High	EAL7 (EAL6)

application provides the logged-in client with the options to administer his account, monitor and enter the transactions, manage the payment orders, submit requests for new products and payment card issuance, possibly request a blocking of the account and payment card, etc. All the active operations are confirmed by a SMS authorization. Technical specification: The module required security on level EAL6 according to Common Criteria (standard ISO IEC 15408). Further, high reliability, very fast processing of inquiries, operations and batch actions, and easy control for the end user, were required. The software was developed in Java language and a very experienced team of analysts and developers was participating in its development. Actual code extent: 25.995 KLOC.

5.1 Effort and time estimation by method FPA&SF

For estimation of UFP and EM see Table 4 and Table 5.

Table 4: Estimation of unadjusted function points

Weigh	Simple	Avarage	Comple	Total
EI	2 * 3 +	5 * 4 +	16 * 6 =	122
EO	0 * 4 +	2 * 5 +	4 * 7 =	38
EQ	2 * 3 +	2 * 4 +	6 * 6 =	50
ILF	1 * 7 +	2 * 10 +	9 * 15 =	162
EIF	0 * 5 +	2 * 7 +	6 * 10 =	74
			UFP	446

Table 5: Estimation of EM

Characteristic	Evaluation
Data communications	4
Distributed data processing	2
Performance	3
Heavily used configuration	2
Transaction rate	3
On-line data entry	3
End-user efficiency	4
On-line update	4
Complex processing	3
Reusability	3
Installation ease	3
Operational ease	4
Multiple sites	3
Facilitate change	4
Product security	4

$$TDI = \sum_{i=1}^{15} DI_i = 49 \tag{14}$$

$$VAF = (49 * 0.01) + 0.65 = 1.14 \tag{15}$$

$$FP = 1.14 * 446 = 508 \tag{16}$$

$$E = \frac{508}{11.45} = 44.37[PM] \tag{17}$$

$$T = 508^{0.4} = 12.09[M] \tag{18}$$

5.2 Effort and time estimation by method COCOMO II&SF

Used model: Post-Architecture Model Estimation total effort multiplier and scale factor in accordance of Table 6 and Table 7.

Table 6: Estimation of Effort Multiplier

Characteristic	Evaluation	Value
RELY	Very High	1.26
DATA	High	1.14
CPLX	Normal	1.00
RUSE	Normal	1.00
DOCU	Low	0.91
SECU	Very High	1.36
TIME	Normal	1.00
STOR	Norma	1.00
PVOL	Norma	1.00
ACAP	High	0.85
PCAP	High	0.88
PCON	Very Highl	0.81
AEXP	High	0.88
PEXP	Very High	0.85
LTEX	Very High	0.84
TOOL	High	0.90
SIDE	Normal	1.00
SCED	Normal	1.00

Table 7: Estimation of Scale Factor

Characteristic	Evaluation	Value
PREC	High	2.48
FLEX	Normal	3.04
RESL	High	2.83
TEAM	Normal	3.29
PMAT	High	3.12

$$W = \sum_{i=1}^{5} W_i = 14.76 \tag{19}$$

$$E = 2.94 * 25.995^{1.0576} * EM = 56.15[PM] \tag{20}$$

$$T = [3.67 * 56.15^{0.28+0.2*0.01*14.76}] * \frac{SCED\%}{100} = 12.77[M] \tag{21}$$

5.3 Influence of product security to the development time

Comparison of calculated project development time and effort, with and without consideration of product security factor, is presented below. Further, the results are compared with real values that were acquired during the project realization.

Method FPA:

$$FP = 1.10 * 446 = 491 \tag{22}$$

$$E = \frac{491}{11.45} = 42.88[PM] \tag{23}$$

$$T = 491^{0.4} = 11.92[M] \tag{24}$$

Method FPA&SF:

$$FP = 1.14 * 446 = 508 \tag{25}$$

$$E = \frac{508}{11.45} = 44.37[PM] \tag{26}$$

$$T = 508^{0.4} = 12.09[M] \tag{27}$$

Method COCOMO II:

$$E = 2.94 * 25.995^{1.0576} * 0.4478 = 41.29[PM] \tag{28}$$

$$T = [3.67 * 41.29^{0.28+0.2*0.01*14.76}] * \frac{SCED\%}{100} = 11.61[M] \tag{29}$$

Method COCOMO II&SF:

$$E = 2.94 * 25.995^{1.0576} * EM = 56.15[PM] \tag{30}$$

$$T = [3.67 * 56.15^{0.28+0.2*0.01*14.76}] * \frac{SCED\%}{100} = 12.77[M] \tag{31}$$

Real values:
Real development time: T = 12.65 month
Real development effort: E = 53.76 person-month
Real teme used to ensuring product: Ts = 1.23 month

6. Conclusions

The contribution covered the problems related to determination of effective effort estimation of projects for security product development. An approach, which describes an option of effort consideration, that forms a basic element for cost determination related to the developed product security on the requested security level, was designed.

A new FPA&SF method, which was created by the FPA method extension by product security factor that forms 15th attribute of general system characteristics, was introduced. Also the COCOMO II&SF method, that is a result of an extension of COCOMO II method by product security factor in the form of 18th effort multiplier called SECU, specifically in the model PAM (Post-Architecture Model), was presented. Further, the methodology for assessment of the product security factor influence on the system depending on assurances level that must be accomplished by the resulting software product, was presented. The so called Common Criteria defined by the standard ISO/IEC 15408, were used for evaluation of assurances. For both the methods, the relation between the individual levels of assurances and the degree of product security factor influence on the system was concluded on the basis of previously acquired results from successfully finished projects.

The designed methodology was verified experimentally on tens of actual projects where the resulting software product required various degrees of assurance levels, or various security levels. The calculated results were compared with real values acquired after the project completion. There was a calculation based on one actual project presented in the contribution for illustration. It was proved that in case of low security level, the difference is insignificant. In case of software products that require assurance level EAL5 - EAL7, the effort invested in implementation of security mechanisms is very substantial. The COCOMO II method provides more realistic results due to more accurate calibration of evaluation of SECU attribute influence depending on the degree of complexity.

References

[1] N. Abdullah. Estimating Software Cost with Security Risk Potential, 2009.

[2] W. Boehm. COCOMO 2.0 –Model Definition Manual, Center for Software Engineering, 2000.

[3] W. Boehm. Cost Models for Future Software Life Cycle Process: COCOMO 2.0, 2005. ISSN 1022-7091.

[4] C. Ebert. Software Measurement. Springer, 2007. ISBN 978-3-540-71648-8.

[5] IFPUG. Function Point Counting Practices Manual, Release 4.3.1. Technical report, International Function Point User Group, 2010. ISBN 978-0-9753783-4-2.

[6] J. Král. Informační systémy, 1998. ISBN 80-86083-00-4.

[7] S. McConnell. Software Estimation - Demystifying the Black Art, 2006. ISBN 978-0-7356-0535-0.

[8] NBU. Informace o hodnocení bezpečnosti informačních technologií - Common Criteria, 2005.

[9] J. Sedláčková. Cenové odhady softwarových projektů. Master's thesis.

[10] P. Vickers. An Introduction to Function Point Analysis, 2003.

Selected Papers by the Author

J. Sedláčková, J. Kreslíková. Security Factors in Effort Estimation. *31th International Conference Information Systems, Architecture, and Technology*, Poland, 2010.

J. Sedláčková, J. Kreslíková. Improvement Estimation of Software by Security Factor. *Work in Progress Session of the SEAA /DSD conference*, France, 2010.

J. Sedláčková, J. Kreslíková. Improvement of Function Point Analysis by Security Factor. In *Procesný manažer, Sapria*, Slovac Republic, 2010.

J. Sedláčková, F. Huňka. Odhady nákladů workflow projektů metodou FPA. In *Proceeding 3. Summer School of Application Informatics*, pages 94-98, Czech Republic, 2006.

J. Sedláčková. Function Point Analysis and Workflow Projects. In *Proceedings of the 16th Conference and Competition STUDENT EEICT 2010 Volume 5*, pages 97-101, Czech Republic, 2010.

J. Sedláčková. Využití CSP při odhadování nákladů workflow procesů. In *Proceeding 3. Summer School of Application Informatics*, pages 12-18, Czech Republic, 2007.

J. Sedláčková, J. Raček, F. Huňka, J. Ministr. Použití metody funkčních bodů v oblasti workflow procesů. In *Informační technologie pro praxi*, pages 96-100,Czech Republic, 2006.