ERP, implementation, cost estimation

Przemysław Plecka[*], *Krzysztof Bzdyra*[**]

ALGORITHM OF SELECTING COST ESTIMATION METHODS FOR ERP SOFTWARE IMPLEMENTATION

Abstract

The article discusses the problem of selecting estimation methods for cost and implementation time for ERP systems, in case when system modifications are necessary. The authors reviewed the methods available in the literature and characterised the stages of strategic phase in the implementation process. On the basis of the analysis of data range and quality required by each method and the data obtained at different stages, a selection algorithm for each stage was proposed.

1. INTRODUCTION

At the moment all significant software producers have their standard product in ERP-class: SAP – Business Suite, Microsoft – Dynamix AX, JD Edwards – EntrepriseOne, etc. During trade talks while the systems the parties reach a conclusion that the organisation of processes in the company does not fully overlap with processes supported by the computer system that is available [1]. There is a group of processes that is not represented in any functionality in standard ERP system. This generates a need for adapting information systems (IS) to a company. The costs of modifications increase the value of the contract (implementation). In some cases it is the company that adapts processes to the system, however, the costs of organisation changes are an additional burden to the client. It is only when clients recognise the costs of system implementation (including adaptations), they incline to consider changes in their organisations. In order to give supplies the basis for negotiations, cost estimation at very early stages of implementation is crucial.

[*] Politechnika Koszalińska, Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, e-mail: przemek.plecka@gmail.com

[**] Politechnika Koszalińska,Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, e-mail: krzysztof.bzdyra@tu.koszalin.pl

While estimating the costs of modifications, software providers encounter difficulties in selecting appropriate method. Usually, they select one that they know best and use it throughout the estimation process. Such an approach causes large estimation errors [2]. A tool suggesting the method that produces the most reliable results would be helpful for software developers. After each stage of selling process, the supplier might verify the data that was gathered and obtain a suggestion which method to choose. The stages of software lifecycle [3] and software valuation methods are known. The question is which of the evaluation methods produces most appropriate results of costs and time at a given stage of strategic phase of IS implementation. The scope of the problem was limited to ERP class information systems implemented in medium-sized enterprises.

The methods facilitating valuation of software production are known and discussed in literature, e.g. by McConell [4]. However, due to changes in information technologies, the popularity of their use has still been changing. The use of algorithmic methods at initial stages of information projects is difficult. At that stage there are no analytic or project documentations, whose components facilitate estimating algorithms. Despite the fact that the uses of algorithmic methods at early stages of information projects can be found in literature [5, 6], the practice of information project suppliers indicates a common use of non-algorithmic methods as faster (i.e. cheaper) and easier. In literature, one can find suggestions for using cost evaluation methods for information projects, starting with statements that any combinations of methods should be used, through views about when and what methods should be used, and finishing with "step by step" procedures [7].

Negotiations with ERP system suppliers and clients concern two elements: costs and implementation time. Consultants estimating the cost of software use such time-consumption measures as man-hour, man-day or man-month. With a given cost of a working unit of time for implementation, it is possible to calculate the cost in a given currency and the time (dates) of implementation, with consideration for possible simultaneousness of certain works.

Chapter 1 of the present article includes the description of stages in strategic phase of implementation project with consideration for the quality of data available for valuation. Another chapter is a review of algorithmic valuation methods. Chapter 3 includes the description of non-algorithmic methods. The final chapter presents the conclusions resulting from the connection of effects from lifecycle stage and the data necessary for software valuation. On this basis an algorithm using a selection questionnaire to choose an evaluation method at each stage of strategic phase.

The use of symbols in Fig. 1, 3, 4, 5 and 6 is in accordance with BPMN 2.0 [8], even if full schemes may not be coherent with the notation.

2. LIFECYCLES STAGES OF ERP SYSTEM IMPLEMENTATION

Numerous authors describe software lifecycles focusing on software production or developing software on individual client's order [3]. None of the presented models corresponds entirely to implementation process of ERP-class software in a middle-sized company. They do not consider "flexibility" of the end of strategic phase (concluding a contract) and possibilities of having one additional stage – feasibility study. Feasibility study is not significant for software lifecycle, however, it provides information for project valuation. The stages of strategic phase are the following:

1. Initial trade talks.
2. Pre-implementation phase.

2' Feasibility study.

3. Project of changes in the system.

The stages of strategic phase and other phases of software lifecycle (implementation, integration, evolution) are presented in Figure 1.

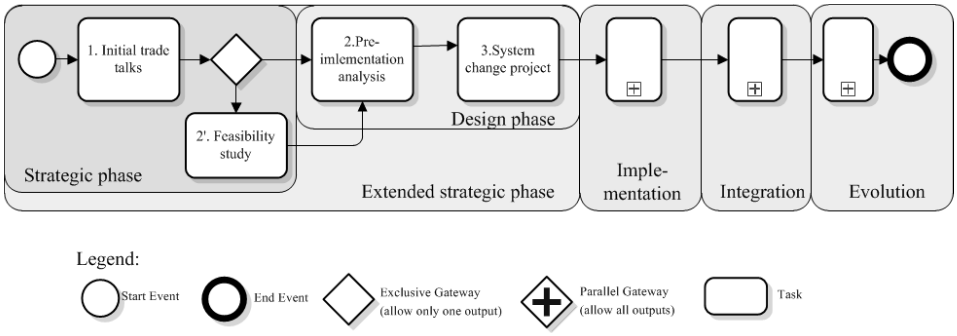

Fig. 1. The stages of strategic phase and other phases of ERP system [source: own study]

Considering cost evaluation, one should remember that in the sales process the moment of contract conclusion is significant. It may happen right after stage 1 but not later than after the end of stage 3. The period is called the strategic phase. It is in IS supplier's interest to get the contract signed as soon as possible, as the implementation of subsequent stages increases the costs with the risk of failure to conclude the agreement at all. However, early estimation of costs involves higher risk of estimation error.

2.1. Initial trade talks

The supplier has meetings with a prospective client in order to define the scale and value of the contract. Usually it is one initial meeting followed by two or three presentation meetings. Some of the elements of work range are identified quickly and precisely. The concern primarily computer hardware, network infrastructure and licences for individual ERP modules. Some elements, e.g. IS modifications that result from non-typical users requirements are difficult to define. At this sate the supplier cannot fully identify the needs that are not satisfied by the standard version of ERP system. As clients' knowledge on IS comes from trade presentations, they cannot define precisely which requirements are not standard. The requirements that supplier is able to obtain from the client are usually incomplete (requirements that the client considered unimportant are missing) and general (client is not able to define the level of specificity).

If the supplier can specify client's expressed requirements and suggest the un-expressed ones, attempts can be made to evaluate the changes. For example, a client defined the requirement in production area concerning separate order for materials from A group of goods for each commission. Such a requirement suggest un-expressed requirement of ordering in the area of logistics, where the management of goods from group A must be excluded from the general plan of orders. Both requirements should be evaluated, even though only one of them was specified by the client. At this stage, single, specified requirements are expressed: reports aggregating the same data in different forms, printout in a specific form used by the client,

2.2. Feasibility study and pre-implementation analysis

If the supplier was unable to evaluate system adaptation (modifications) works clarification and specification of client's needs must be done. Then a pre-implementation analysis or feasibility study is prepared [9]. Although both solutions are aimed at specifying the data for the evaluation, the basic purpose of each is different.

Feasibility study includes information on the company in a form of a systematic document based on economic facts [10]. The information concerns economic, organisational and technical aspects. The aim of the study is to define the range of works and the costs of the project. The document is used by supplier's decision-makers while analysing economic aspects of project implementation.

Pre-implementation analysis does not include other the information than this concerning computer system in the context of a given company and the work. The result is a report including the following components: functional range of the implementation, list and description of business processes, functions and data advised to be included in the functional range of the system, organisational range of the implementation, the proposed aims of the implementation, expected business benefits, schedule of work [11]. At this stage the supplier assumes that the requirements are complete and their level of specificity meet developers' expectations, who rely on this document in their further works.

Even in a medium-sized production company recording all user requirements and processes would be very time-consuming and expensive (from a few thousand up to over a thousand requirement). Moreover, in most cases they would overlap with the records in ERP system documentation. Therefore, suppliers make a differential analysis which includes only those elements that are not covered in a standard IS. Such a procedure shortens the time of stage implementation but also allows the client to see the documentation of a standard version with the pre-implementation analysis.

The supplier should evaluate the quality of requirements that were expressed at this stage for the use of software evaluation method.

2.3. Project of system changes

Project of information system is an intermediate phase between defining the requirements and the implementation. The documentation the is produced exclusively for internal use of the supplier (software departments). Depending of methods of implementation (structured, object-oriented programming, or agile software development, etc), project documents may include different elements [3]. Some ERP system developers worked out their own specific methodologies. In such cases the documentation will be specific. One such example is Select Perspective methodology [12, 13] or ARIS [14]. However, there are always common elements for evaluating software.

The first element of software developing is to specify the requirements resulting from implementation character. The level of requirement specificity must determine the manner of implementation in an unambiguous way. Despite this, project documents include the elements describing data structures and procedures of processes. There is a number of methods for presenting project information: from DFD [15], Entity-Relationship Diagrams, through UML models [16]. Each of them is an appropriate source of date for software evaluation.

2.4. Summary of lifecycle stages

With subsequent stages of software lifecycle supplier's knowledge on the differences between processes in the company and standard software functionalities grows. At first they have only one, incomplete set of general requirements. In subsequent stages requirements are completed and specified. After the project stage, the supplier can additionally use project elements such as: data objects (tables, fields), windows, interfaces, etc for evaluation. On the other hand, supplier's costs grow. If a contract with client is concluded, the costs will be included in the contract value, if not, they will be the supplier's cost. Input information necessary for making valuation at the subsequent stages of project lifecycle is presented in Figure 2.

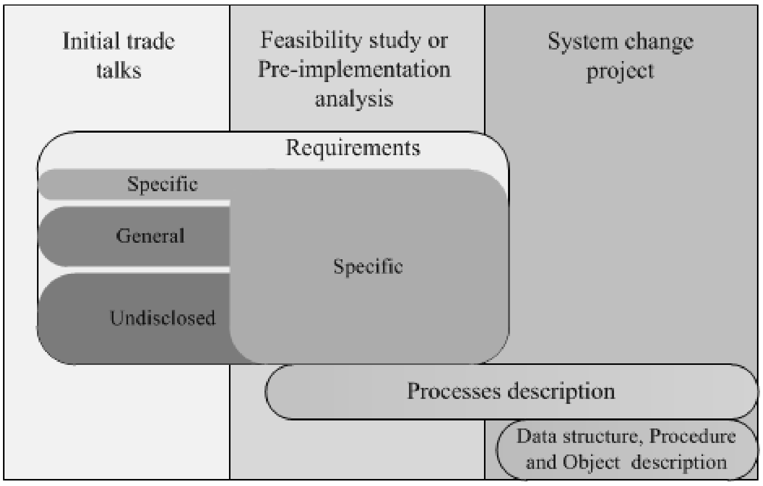

Fig. 2. Input information for evaluation process [source: own study]

3. ALGORITHMIC METHODS OF SOFTWARE EVALUATION

3.1. COCOMO II method

Constructive Cost Model (COCOMO) Method was proposed by Barry Bohem in 1981 [15]. Since then a number of versions and types of this method have been developed, e.g. COCOMO81, COCOMO II [16]. The sequence of processes comprising the evaluation is presented in Figure 3. With the use of COCOMO method Person per Month (PM) can be calculated on the basis of the amount of source code in the programme Kilo Source Line of Code (KSLOC) (process 1 in Figure 3). The information necessary for evaluating the amount of code are obtained from the IS project documentation. The amount of KSLOC are attributed to a number of programme elements, such as procedures, modules, objects, etc. Because for many contemporary uses the amount of code does not correspond to PM, the method was modified by using function point analysis [17] (process 2 in Figure 3) calculated on the basis of compete and specific requirements. The analysis of function points was presented in the next chapter.

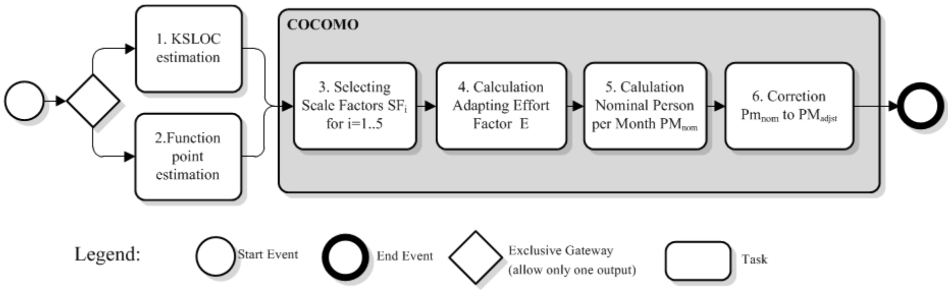

Fig. 3. Sequence of processes in COCOMO method [source: own study]

The first activity is defining five Scale Factors (SF), whose value was determined empirically in five classes, depending on the level of complexity (from very low to very high). Knowing the value of individual factors, the factor adapting effort (E) can be determined from the formula (1):

$$E = B + 0{,}01 \cdot \sum_{i=1}^{5} SF_i \tag{1}$$

where:

B – constant 0.91 for COCOMO II model [23].

Nominal Person per Month PM_{nom} is done in accordance with the formula (2):

$$PM_{nom} = A \cdot (Size)^E \tag{2}$$

where:

Size – the number of code lines In KSLOC unit,
A – constant determined on the basis of previous projects = 2.94 [23].

For models from the first stages of Application Composition Model, Early Design Model [17] nominal time should be corrected with seven coefficients of Person per Month, in accordance with the formula (3).

$$PM_{adjs} = PM_{nom} \cdot \prod_{i=1}^{7} EM_i \tag{3}$$

where:

EM_i - Effort Multiplier.

For the models in another lifecycle stage (Post-Architecture Model) the formula for nominal Person per Month was enriched by 9 indicators (i=1..16). Alike *SF* values, *EM* were determined empirically. The data for SF and EM calculations can be found in method documentation [17].

The literature includes a number of examples of adapting COCOMO method [18, 19] with the use of fuzzy logic, inter alia [20, 21, 22].

3.2. Evaluation with the use of function points

Evaluation method proposed by A.J. Albrecht [23] requires the calculation of the number of function points (FP) on the basis of specific requirements. Then COCOMO method or *Evaluation by Analogy* can be used to calculate the number of FP into Person per Month or costs. The set of user requirements that is used in calculations must be complete and all the requirements must be specific. The process of evaluation with the use of function points is presented in Figure 4.

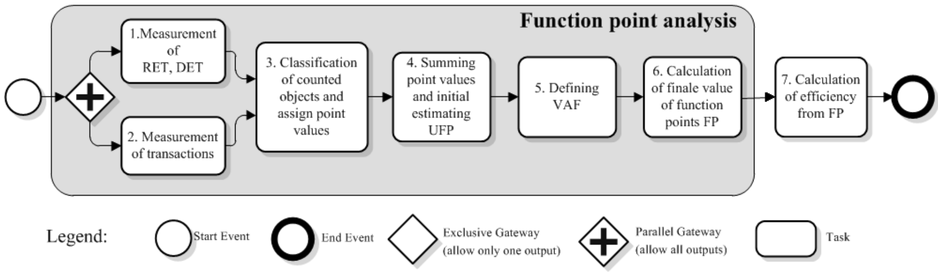

Fig. 4. Evaluation process with the use of function points [source: own study]

Function points method is based on selecting five classes of objects in requirements or the ready program (processes 1 and 2 in Figure 4):

1) Internal Logic File (ILF),
2) External Interface File (EIF),
3) External Inputs (EI),
4) External Outputs (EO),
5) External Inquires (EQ).

The first two classes are related to data, the three other – transactions. To make estimations at this first stage, the following indicators are used (process 3 in Figure 4):

- RET (Record Element Type) - unique, recognisable subgroup of elements given in ILF or EIF, correspond to the record in the table;
- DET (Data Element Type) - Unique, identifiable field in ILF or EIF, correspond to the field in record;

- FTR (File Type Referenced) - recognisable by users, logically related data, correspond to files or relationally connected files.

All objects in classes must be identified and attributed with appropriate value of indicators ILF and EIF are described with RET and DET, while EO, EI and EQ with FTR and DET. In this way the number of Unadjustment Function Points for a given objects is read from the table. Summing *UFP* values of all objects in all classes the total value of Unadjustment Function Points is obtained.

Value Adjustment Factor – VAF considers for internal system complexity, unrelated to its functionality. Defining the value entails giving the impact of 14 factors, which may raise system complexity (process 5 in Figure 4). The list of factors can be found in method documentation [24]. VAF value is calculated from the formula (4):

$$VAF = B + 0{,}01 \sum_{i=0}^{14} C_i \tag{4}$$

where:

B – empirically determined constant value 0.65 [31],

C_i - impact value of i-th factor.

On the basis of VAF the final values of function points are calculated by correcting the Unadjustment Function Points according to the formula (5):

$$FP = VAF \cdot UFP \tag{5}$$

Knowing FP value, efficiency can be determined with two methods (process 7 in Figure 4):

- calculating into KSLOC with empirically determined values from calculation table [25] and then use COCOMO method to define Person per Month,
- If the organisation owns historic data, FP value can be directly calculated into Person per Month, using Estimation by Analogy method.

The source of compete and updated documentation of the method is website of International Function Point Users Group [26].

4. NON-ALGORITHMIC METHODS OF SOFTWARE EVALUATION

4.1. Decomposition and reconstruction

Decomposition and reconstruction is a popular method due to its intuitiveness and universality. It is used in situations when whole project evaluation generates difficulties, e.g. resulting from work heterogeneity. In practice of IT project implementation [29] there are very few project that can be evaluated without this method.

The method involves decomposing the range into a number of components. The method of division is arbitrary and depends on project specifics. Suppliers frequently do evaluation with Work Breakdown Structure (WBS) method [13]. Having done the division, the parts of objects are estimated and undergo further division with the same or other method. The "depth" of division depends on the evaluation methods that is going to be employed in the next stage. Even though the literature lists this method as equal to others [4] its role in the evaluation process is different from others. Project evaluation is started in this method, but after decomposition, other methods of elemental evaluation are selected. A detailed description of decomposition method according to WBS can be found in literature [27, 28, 29, 30].

4.2. Individual expert evaluation

The method of valuation by *individual expert evaluation* is the most frequently used method, not only in software development [31], but also in other IT enterprises such as implementations and modifications. The research conducted in USA in 2002 showed that as many as 72% of the valuations are done with this method [36]. In the first stage, the method requires selecting experts with appropriate knowledge and experience. Then experts evaluate the ranges they were bestowed. In order to reduce the evaluation errors, the method was modified with multiple evaluation for different versions of implementation. Such a technique, called PERT (Program Evaluation and Review) [27, 33], involves analyses of the most optimistic, the most probable case and the worst case. However, it is different from critical path analysis (CPM [34]) because it is used to evaluate independent tasks only. After previous decomposition processes, the information about relations between tasks was lost. The expected evaluation has the following form, then:

$$f(x) = \sum_{i=1}^{N} (Cp(x_i) + 4 \cdot Co(x_i) + Ck(x_i))/6 \qquad (6)$$

where:

C_p – the most optimist value of the i-th task,

C_o – the most likely value of i-th task,

C_k – the most probable value of i-th task.

The specificity of results in this group depends entirely on expert's experience. Selection criteria are imprecisely defined. The influence of personality is significant in as much as experience does not guarantee more precise evaluations. There are undervaluing, overvaluing or unexpected experts.

The method can be used from supplier's first contacts with the client. With appropriate use of experts, evaluation can be done even on the basis of incomplete set of general user's requirements.

4.3. Group expert evaluation

The method involves presenting the same range of work to more than one expert. In unstructured version of the method (group review) the experts decide about the valuation or its range as a group. In a structured version, called Wideband Delphi [35, 15], experts' work is done in a formalised way and its result is a scoring evaluation.

The work of experts in groups is more expensive than individual work, however, method' advantage over *individual expert evaluation* is the decrease of personality factors' importance. In spite of different experience, characters and inclinations, experts will either reach a common ground or, as in case of Widebrand Delphi type, the conclusion of problem is reach by attributing pre-selected points.

Estimation method is used frequently at initial stages of IT projects in situations of high uncertainty of requirements.

4.4. Summing, computing, evaluating

The method concerns searching quantifiable objects, e.g. requirements, functions, use cases, stories, reports, windows, database tables, classes in the project. Each identified object that can be summed is attributed with estimation constituent (cost or time). The estimated values are the function (7) of the objects constituting an information project:

$$f(x) = \sum_{i=1}^{N} C(x_i) \tag{7}$$

where:

x – calculated object,
N – the number of summed objects,
C – computer cost of the object.

The method can be used at every stage of software development or modification. The method is not complex provided the source documentation allows determining the summed objects. One of the failures is high risk of omitting objects or ranges of work that influenced the value of the whole project, for example, ignoring supplementary tables or costs of developing filtering inquiries while evaluating the costs of interface windows. Important stage in this method is the evaluation of individual objects' costs. The stage of individual object evaluation is an important stage of this method. This can be done with help of *Individual expert evaluation* or *Group expert evaluation*. The method is efficient in projects with a small number of object types are identified but they are plentiful, e.g. 30 reports, 25 SQL inquiries and 18 interfaces.

4.5. Evaluation by analogy

The method concerns dividing the project into components that already exist in a completed project. Evaluating selected parts, one may calculate the ratio of two projects' sizes (new and the completed one). Knowing the relations between the sizes and the costs of the completed project, one may estimate the value of the new project.

The difficulty lies in collecting historic data from similar projects and structure as the evaluated project. Additional problem is the selection of a representative part of the decomposed project, which is a basis for multiplicity factor. Ignoring significant objects may increase the evaluation error.

The input data for this method comes from data objects and programmes (interfaces, SQL queries, FP). The use of requirements, even the specific ones, does not allow calculating multiplication factor, thus doing the whole evaluation. Therefore, the method can be used when the effects of programming are known.

4.6. Valuation based on substitution

Alike the previous one, this method requires the knowledge of costs of previously completed in organisation of standard objects (interfaces, reports, etc). Depending on the version of method, the objects can be grouped differently. For example, Putnam [33] and Humphrey [36] selected classes of objects: very small, small, medium, large and very large. Another method of classifying the objects is a standard component method [4] used to valuate object software. If the IS system supplier uses extreme software or close to Agile methods [37], so called "stories" might be a standard element.

Then, the groups of objects are attributed with average cost values, e.g. number of lines of code (LOC), man hours or man days. The objects from a new project must be classified in the same manner. Then their sum can be calculated. Similarly to the previous method, this one should also be used when classes of programming objects are known. One exception is the organisations using extreme or agile programming. In this case, the costs of "stories" that were documented at the stage of talks to clients can be substituted with historic data. The practice of evaluations [2] implies that it can be used at earlier stage (preimplementation analysis), when only specific requirements are known.

4. CONCLUSIONS

Concluding, one should notice that implementation of the first stages of software lifecycle provides more and more information about the planned solution, on the one hand, and there is a number of evaluation methods available on the other hand.

On the basis of the analysis of evaluation methods, the authors propose their own method of selecting precise method of evaluating implementation cost and time (modification of software during implementation).

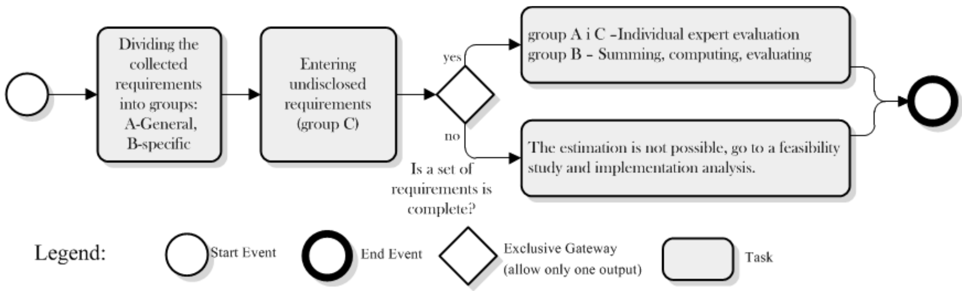

Fig. 5. The method of evaluation at the stage of trade talks [source: own study]

For the trade talks, the algorithms of conduct is presented in Figure 5. As it is presented, for all the groups of requirements time and cost can be estimated only in cases when the set of requirements is complete. As software developers do not mange to complete the set of requirements with a subset of requirements unexpressed at the stage of feasibility study or implementation analysis. In such a case other evaluation methods are available, what is presented in Figure 6.

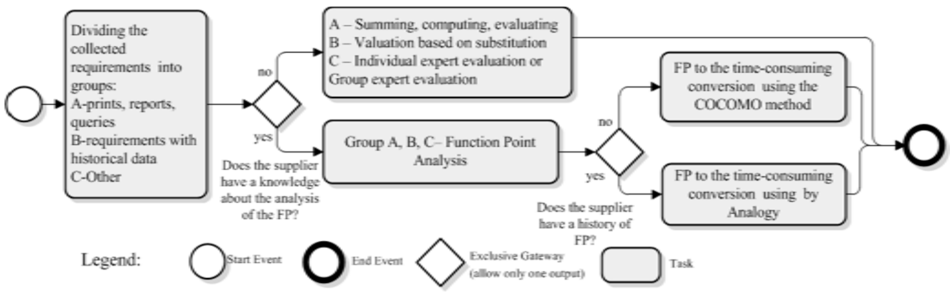

Fig. 6. The selection of evaluation methods at the stage of implementation analysis (feasibility study) [source: own study]

The stage of system change provides, along with additional requirements, the information on the works – data structure, information on the procedures, objects, etc. Alike in previous stages, the supplier should classify the available data. The algorithm of actions is presented in Figure 7.

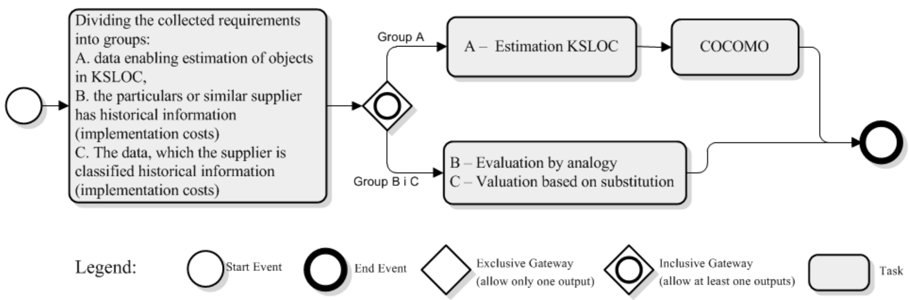

Fig. 7. The selection of evaluation methods at the stage of software project [source: own study].

The above proposition allows using the methods which are most efficient at each stage.

REFERENCES

[1] BURNS M.: How to select and implement an ERP System [online]. 2005. Available: http://www.180systems.com/ERPWhitePaper.pdf

[2] PLECKA P.: Selected Methods of Cost Estimation of ERP Systems' Modyfications. Zarządzanie Przedsiębiorstwem, 2013.

[3] SOMMERVILLE I.: Software Engineering. Pearsom Education Limited, Edingurgh, 2007.

[4] MCCONELL S.: Software Estimation: Demystifying the Blac Art. Microsoft Press, 2006.

[5] MELI R.: Early Function Points: a new estimation method for software project. WSCOM97, Berlin, 1997.

[6] SANTILLO L., CONTE M. I MELI R.: Early &Quick Function Point: Sizing More with Less. Metrics 2005, 11 th IEEE Intl Software Metrics Symposium, Como, Italy, 2005.

[7] BOEHM B., ABTS C. I CHULANI S.: Software Development Cost Estimation Approaches – A Survey. Annals of Software Engineering, vol. 10, no. 1-4, 2000, p. 177-205.

[8] BPMN: Object Management Group [online]. 2013. Available: http://www.bpmn.org
[9] FRĄCZKOWSKI K.: Zarządzanie projektem informatycznym. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2003.
[10] PHILIPS J.: IT Project Management. On Track from Start to Finish. Osborne, 2004.
[11] JUSTYNOWICZ K.: Analiza przedwdrożeniowa coraz popularniejsza [online]. 2007. Available: http://www.bcc.com.pl/akademia-lepszego-biznesu/analiza-przedwdrozeniowa-coraz-popularniejsza.html
[12] ALLEN P., FROST S.: Component-Based Development for Enterprise Systems, Applying the Select Perspective. Cambridge University Press, Cambridge, 1998.
[13] Select Business Solution [online]. Available: http://www.selectbs.com
[14] ARIS [online]. Available: http://www.softwareag.com
[15] BOEHM B.: Software Engeneering Ecomonics, Englewood Clifs, New York, 1981.
[16] BOEHM B. W.: Software Cost Estimation with COCOMO II. Prentice Hall, 2000.
[17] BAIK J.: COCOMO II, Model Definition Manual, Version 2.1. Center for Software Engineering at the University of Southern California, 2000.
18] HELE J., PARRISH A., DIXON B., SNITH R.: Enhancing the Cocomo estimation models. Software, IEEE, vol. 17, no. 6, p. 45-49, 2000.
19] ALJAHDALI S., SHETA A.: Software effort estimation by tuning COOCMO model parameters using differential evolution. Computer Systems and Applications (AICCSA), IEEE/ACS International Conference on, Hammamet, 2010.
[20] FEI Z.: f-COCOMO: fuzzy constructive cost model in software engineering," Fuzzy Systems, 1992., IEEE International Conference on, San Diego, CA, 1992.
[21] SATYANANDA R. C.: An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function. JOURNAL OF SOFTWARE, vol. 4, no. 5, p. 452-459, July 2009.
[22] ATTARZADEH I., Improving estimation accuracy of the COCOMO II using an ada-ptive fuzzy logic model. Fuzzy Systems (FUZZ), 2011 IEEE International Conference, Taipei, 2011.
[23] ALBREHT A.: Measuring Application Development Productivity. Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, USA, 1997.
[24] IFPUG: Function Point Counting Practices: Manual Release 4.1, Westerville, OH: IFPUG, 1999.
[25] The QSM Function Points Languages Table [online]. QSM, 2013. Available: http://www.qsm.com/resources/function-point-languages-table
[26] International Function Point Users Group [online]. Available: http://www.ifpug.org
[27] STUTZKE R. D.: Estimation Software-Intensive Systems. Upper Saddle River, New York, Addison-Wesley, 2005.
[28] TAUSWORTHE R.: The work breakdown structure in software project management. Journal of Systems and Software, vol. 1, 1984.
[29] NORMAN E., BROTHERTON S. I FRIED R.: Work Breakdown Structures. The Foundation for Project Management Excellence, John Wiley & Sons, 2010.
[30] HAUGAN G.: Effective Work Breakdown Structures. Project Management Institute, 2002.
[31] JORGENSEN M.: A Review of Studies on Expert Estimation of Software Development Effort. Journal of Systems and Software, vol. 70, no. 1-2, p. 37-60, 2004.
[32] KITCHENHAM B., PFLEEGER S. L., McCOLL B., EAGAN S.: An empirical study of maintenance and development estimation accuracy. Journal of Systems and Software, vol. 64, no. 1, p. 57-77, 2002.
[33] MYERS P. L. H. W.: Measures for Excellence. Reliable Software on Time, Within Budget, Englewood Cliffs, NY: Yourdon Press, 1992.
[34] FONDAHL J. W.: The History of Modern Project Management Precedence Diagramming Methods: Origins and Early Development. Project Management Journal, vol. XVIII., no. 2, 1987.
[35] NASA: ISD Wideband Delphi Estimation [online]. 2004. Available: http://software.gsfc.nasa.gov/assetsapproved/PA1.2.1.2.pdf
[36] HUMPHREY W. S.: A Discipline for Software Engineering. Addison Wesley, 1995.
[37] COHN M.: Agile Estimating and Planning. Upper Side River, NY: Prentice Hall PTR, 2005.