The Economic Importance of Business Software Systems Development and Enhancement Projects Functional Assessment

Beata Czarnacka-Chrobot
Department of Business Informatics
Warsaw School of Economics
Warsaw, Poland
e-mail: bczarn@sgh.waw.pl

Abstract—**Execution of Business Software Systems (BSS) Development and Enhancement Projects (D&EP) encounters many problems, leading to the high scale of their failure, which then is reflected in considerable financial losses. One of the fundamental causes of such projects' low effectiveness are improperly derived estimates for their costs and time. In their case, the budget and time frame are determined by the effort being spent on activities needed to deliver product that would be meeting client's requirements. Meanwhile, objective and reliable effort estimation still appears to be a great challenge, what in the author's opinion is caused by effort estimation based on resources, while such planning activity should base on the required software product size, which determines work effort. Estimation of BSS size requires using of the suitable software size measure, which has been sought for several decades now. What's more, it is worth using the capabilities offered by such measure for the BSS D&EP assessment from the perspective being critical to a client, that is from functional perspective. Thus this paper analyses capabilities, being significant from the economic point of view, of taking advantage of suitable approach to the BSS size measurement, what should contribute to the better understanding of the importance of this issue, still being underestimated by business managers – as in the subject literature this issue is usually considered from the technical point of view. Meanwhile, suitable BSS size measurement should constitute the basis for rational activities and business decisions not only for providers, but also for clients needs.**

Keywords-business software systems development and enhancement projects; effectiveness; software size measures; functional size measurement; functional assessment; Software projects Functional Assessment Model (SoftFAM)

I. SCALE OF FAILURES IN THE BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS EXECUTION

In practice, the execution of software Development and Enhancement Projects (D&EP), particularly those delivering Business Software Systems (BSS) as a product, encounters many problems, which makes fulfilling of client requirements still appear a big challenge for companies dealing with this kind of business (see also [1]). This may be proved by the unsatisfactory effectiveness of such projects, revealed by numerous analyses, which manifests itself in the high scale of their failure.

The Standish Group, the US institution providing research reports on this issue from 15 years, estimates that now only 32% of application D&EP worldwide turn out successful while products delivered as a result of nearly 45% of them lack on average 32% of the required functions and features, the planned time of product delivery is exceeded by nearly 80% on average and the estimated budget - by approx. 55% on average [2]. Also, it is worth mentioning the research carried out by government agencies in the USA indicating that 60% of software systems development projects overrun the planned completion time, 50% of these projects overrun the estimated costs while in the case of 46% of them the delivered products turn out useless [3]. Similar – as to the general conclusion – data result from the analysis of IT projects being accomplished in Poland, which was carried out by M. Dyczkowski, indicating that in 2006-2007 approx. 48% of such projects went over the planned completion time while approx. 40% exceeded the estimated budget [4].

Analyses by T.C. Jones plainly indicate that those software D&EP, which are aimed at delivery of business software systems, have the lowest chance to succeed [5]. The Panorama Consulting Group, when investigating in their 2008 study the effectiveness of ERP (Enterprise Resource Planning) systems projects being accomplished worldwide revealed that 93% of them were completed after the scheduled time while as many as 68% among them were considerably delayed comparing to the expected completion time [6]. Merely 7% of the surveyed ERP projects were accomplished as planned. Comparison of actual versus planned expenses has revealed that as many as 65% of such projects overran the planned budget. Only 13% of the respondents expressed high satisfaction with the functionality implemented in final product while in merely every fifth company at least 50% of the expected benefits from its implementation were said to be achieved. Meanwhile (see also [4][7]):

- BSS are one of the fundamental IT application areas.
- BSS development or enhancement often constitutes serious investment undertaking.
- In practice, COTS (Commercial-Off-The-Shelf) BSS rarely happen to be fully tailored to the particular client business requirements therefore their customisation appears vital.

- Rational *ex ante* and *ex post* valuation of unique (at least partially) BSS, being of key significance to clients, encounters serious problems in practice.
- From the provider's perspective, the discussed type of IT projects is particularly difficult in terms of management, which basically results in their exceptionally low effectiveness as compared to other types of IT projects.

The paper is structured as follows: in Section 2 the author presents the selected results of studies concerning losses caused by the especially low effectiveness of BSS D&EP execution and points out main factors of BSS D&EP effectiveness. In Section 3 different BSS size measures are compared, while in Section 4 the concept and methods of BSS functional size measurement are analysed. Section 5 is devoted to the presentation of author's own model dedicated to BSS D&EP functional assessment against a background of existing related methodologies and along with the main conclusions coming from its verification and comparison to those methodologies. In Section 6 the main results of author's own study on the usage of functional size measurement methods by Polish BSS providers are pointed out. Finally, in Section 7 the author draws conclusions and some open lines about future work on functional approach to the BSS D&EP assessment from the economic point of view.

II. LOSSES CAUSED BY THE LOW EFFECTIVENESS OF BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS EXECUTION

Low effectiveness of BSS D&EP execution leads to the substantial financial losses, on a worldwide scale estimated to be hundreds of billions of dollars yearly, sometimes making even more than half the funds being invested in such projects. The Standish Group estimates that these losses – excluding losses caused by business opportunities lost by clients, providers losing credibility or legal repercussions – range, depending on the year considered, from approx. 20% to even 55% of the costs assigned for the execution of the analysed projects types (see e.g., [8][9]). On the other hand, analyses of The Economist Intelligence Unit, which studied the consequences of BSS D&EP delay indicate that there is strong correlation between delays in delivery of software products and services and decrease in profitability of a company therefore failures of BSS D&EP, resulting in delays in making new product and services available and in decreasing the expected income represent threat also to the company's business activity [10]. Meanwhile,"The costs of these (...) overruns are just the tip of the proverbial iceberg. The lost opportunity costs are not measurable, but could easily be in the trillions of dollars. [For instance - B.C.C.] the failure to produce reliable software to handle baggage at the new Denver airport is costing the city $1,1 million per day." [11].

If direct losses caused by abandoning the BSS D&EP result from erroneous allocation of financial means, usually being not retrievable, in the case of overrunning the estimated time and/or costs, however, they may result from delay in gaining the planned return on investment as well as from decreasing it (necessity to invest additional funds and/or cutting on profits due to the overrunning of execution time and/or delivery of product incompatible with requirements).

According to the Standish Group analyses, yearly spendings on application software D&EP in the USA range from approx. 250 to approx. 350 billion USD. In this type of projects, average yearly cost of development works alone ranges from approx. 0,4 to approx. 1,6 million USD, what indicates that they are usually serious investment undertakings. Spendings on such projects may considerably exceed the expense of building offices occupied by companies commissioning them, and in extreme cases, even 50-storey skyscraper, roofed football stadium, or cruising ship with a displacement of 70.000 tons [12]. Yet quite often client spends these sums without supporting their decision on getting engaged in such investment by proper analysis of the costs, based on the rational, sufficiently objective and reliable grounds. The above situation manifests itself in the difference in costs spent by various organizations on similar applications that may be even fifteen fold [13].

The above unequivocally implies a significant need to rationalize practical activities and business decisions made with regard to BSS D&EP, which is only possible when taking into account factors showing influence on this effectiveness. Author's analysis, which concerned numerous studies on factors of BSS D&EP effectiveness, available in the subject literature, leads to the conclusion that among fundamental factors are:

1) Proper project management, including: realistic planning, with particular consideration given to the reliable and objective estimates for key project attributes (work effort, execution time and cost), and proper project scope management, above all consisting in undertaking small projects, that is projects whose product is characterised by relatively small size. Both these factors require product size measurement.

2) Authentic involvement of client in the project – both users and managers. Thus product size measurement should be carried out by taking into consideration mainly the perspective of the client of BSS being developed, that is with the use of product size units that are of high significance to him.

Therefore if fundamental opportunity to increase the chance for effective execution of the discussed types of projects and to decrease the losses caused by low effectiveness lies in accurate estimates of their key attributes, in undertaking small projects and in client's involvement then what appears to be *significant factor of BSS D&EP success is objective and reliable measurement of their product size*, with particular consideration given to client's perspective. "Measurement of software size (...) is as important to a software professional as measurement of a building (…) is to a building contractor. All other derived data, including effort to deliver a software project, delivery schedule, and cost of the project, are based on one of its major input elements: software size." [14, p. 149].

III. BUSINESS SOFTWARE SYSTEMS SIZE MEASURES

One of the fundamental causes of low BSS D&EP success rate are improperly derived estimates for their costs and time. In the case of such projects the budget and time frame are determined by the effort being spent on activities needed to deliver product, which would meet client's requirements. However, sufficiently objective and reliable BSS D&EP effort estimation still appears to be a great challenge to the software engineering. In the author's opinion the main reason for this problem is effort estimation made on the basis of resources whereas such planning activity should ground on the required software product size, which determines the work effort (see Figure 1).

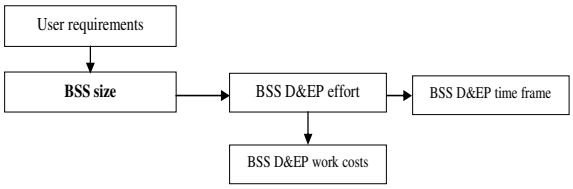

Figure 1. Simplified model of dependencies between BSS D&EP key attributes and the size of project product.

Source: Author's own study.

Basic approaches to the size measurement of every software product may be reduced to perceiving it from the perspective of (see also [7]):

- Length of programmes, measured by the number of the so-called programming (volume) units. These units most of all include source lines of code, but number of commands, number of machine language instructions are also taken into account. However, these units measure neither size of the programmes nor their complexity but only the attribute of "programme length" yet thus far these are them that in practice have been employed most often with regard to the software size [14, p. 149].
- Software construction complexity, measured in the so-called construction complexity units. Most of hundreds of such measures having been proposed are limited to the programme code yet currently these units are used mainly in the form of object points [14, pp. 155-156]. These points are assigned to the construction elements of software (screens, reports, software modules) depending on the level of their complexity.
- Functionality of software product, expressed in the so-called functionality units. They most of all include function points, but also variants based on them such as: full function points, feature points, or use case points. These points are assigned to the functional elements of software (functions and data needed to complete them) depending on the level of their complexity – not to the construction elements as it was the case of object points.

Synthetic comparison of various software size measures against a background of key requirements set for such measures were presented in Table 1.

TABLE I. SYNTHETIC COMPARISON OF SOFTWARE SIZE MEASURES

Requirement towards measures	Programming units	Construction complexity units	Functionality units
Unequivocalness of definition	Freedom in formulating definitions (differences as big as even 5:1)	Depending on the method	In methods normalized by ISO/IEC
Possibility to make reliable prognosis on the size relatively early in the life cycle	Possibility to calculate programme length only for the existing code	None – with regard to programming units and object points	As early as at the stage of requirements specification
Base for the reliable evaluation of the all phases work effort	Final programme length does not fully reflect the whole work done	Final software size does not fully reflect the whole work done	Relatively high reliability as early as at the stage of requirements specification
Software size being independent of the technology employed	Programme length determined by the language employed	Size being dependent on the technology employed	Size depends on functional user requirements
Possibility to compare software written in different languages	Lack of such direct possibility	Lack of such direct possibility	Size doesn't depend on the language used
Measuring size in units being of significance to a client	No significance to a client	Secondary significance to a client	Measurement from the point of view of a client
Possibility to compare delivered size vs. required size	Inability to make reliable prognosis	Inability to make reliable prognosis	Thanks to the possibility of making reliable prognosis
Possibility to measure all software categories	Yes	Depending on the method	Depending on the method
Easiness of use	Yes	No	No

Source: Author's own study.

IV. BUSINESS SOFTWARE SYSTEMS FUNCTIONAL SIZE MEASUREMENT

The right measure of software size has been sought out for several decades now. Many years' verification of various approaches showed that what for now deserves standardization is just the concept of software size measurement based on its functionality – being an attribute of first priority to the client. Due to the empirically confirmed effectiveness of such approach, it was in the last years normalized by the ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission), and turned into the six-part international standard ISO/IEC 14143 for the so-called software Functional Size Measurement (FSM) [15].

Details displayed in Table 1 clearly indicate the reasons why functionality units were recognised as the most appropriate measure of software size not only by the

ISO/IEC but also, among others, by Gartner Group [16] as well as by International Software Benchmarking Standards Group (ISBSG) [17]. They show no limits being characteristic of programming units and construction complexity units – although one may have reservations as to their versatility and relatively high complexity of the methods based on them. However, it is hard to expect that the method of measurement of software products, by nature being complicated, would be effective yet simple.

The set of rules for software FSM enclosed in the ISO/IEC 14143 norm provides key definitions, characteristics and requirements for FSM, and defines Functional Size Measurement Method (FSMM) as a specific FSM implementation defined by a set of rules, which conforms to the mandatory features of such measurement. The first part of this standard defines also indispensable principles upon which the FSMM should be based – fundamental one is the definition of functional size, which is understood as "a size of the software derived by quantifying the functional user requirements", while The Functional User Requirements (FUR) "represent the user practices and procedures, that the software must perform to fulfil the user's needs" [15, Part 1].

After about 30 years of improving various FSM techniques five of them (out of over 20) have been now acknowledged by the ISO/IEC as conforming to the rules laid down in the ISO/IEC 14143 standard, namely:

- International Function Point Users Group (IFPUG) method [18].
- Mark II (MkII) function point method proposed by the United Kingdom Software Metrics Association (UKSMA) [19].
- Netherlands Software Metrics Association (NESMA) function point method [20].
- Common Software Measurement International Consortium (COSMIC) full function points method [21].
- FSM method developed by the Finnish Software Measurement Association (FiSMA) [22].

The FSMM standardized by the ISO/IEC differ in terms of software measurement capabilities with regard to different software classes (functional domains), but all of them are adequate for business software systems (see Table 2 and [23]).

Functional size measurement of BSS supports first of all [15, Part 6]:

- BSS D&EP management by enabling: to make early prognosis on resources necessary for project execution, to monitor progress in project execution, to manage the changes in the required BSS size, to determine degree to which the supplied product meets functional user requirements, as well as to make post-execution project analysis and compare its attributes with other projects.
- BSS performance management by: BSS development, enhancement and maintenance productivity management, quality management, organizational processes maturity and capability management, determining organizational BSS asset value in order to estimate cost of its potential replacement, reengineering, or outsourcing, making prognosis on budget necessary to maintain BSS, as well as BSS supply contracts management.

TABLE II. THE ISO/IEC STANDARDS FOR THE SOFTWARE FSMM VERSUS FUNCTIONAL DOMAINS

FSMM	Functional domains specified in the norm	Constraints indicated in the norm
ISO/IEC 20926 - IFPUG method	All software classes	None
ISO/IEC 20968 - UKSMA method	For any type of software provided that the so-called logical transactions may be identified in it (the rules were developed as intended for business software).	The rules support neither complex algorithms characteristic of scientific and engineering software nor the real-time systems.
ISO/IEC 24570 – NESMA method	All software classes	None
ISO/IEC 19761 - COSMIC-FFP method	- Data-driven systems, e.g., business applications for banking, insurance, accounting, personnel management - Real-time systems (time-driven systems), e.g., telephone exchange systems, embedded software, process control, operation systems - Hybrid solutions combining both above, e.g., real-time systems of airline tickets booking.	- Systems with complex mathematical algorithms or with other specialised and complex rules (e.g., expert, simulation, self-learning systems) - Systems processing continuous variables (audio, video) - For above-mentioned domains it is possible to modify the method so that it may be used locally.
ISO/IEC 29881 - FiSMA method	All software classes	None

Source: Author's own study based on [15, Part 6].

The FSMM standardized by the ISO/IEC provide sufficiently objective and reliable basis for BSS D&EP effort, budget and time frame estimating. Results of numerous surveys, including e.g., those carried out by the State Government of Victoria [24] and International Software Benchmarking Standards Group [17], indicate that BSS D&EP, in case of which the FSMM were used for effort planning, are characterised by relatively accurate estimations. Studies by the State Government of Victoria indicate that pricing of BSS on the basis of product size expressed in functionality units results in reducing the average budget overrun to less than 10% – comparing with current average budget overrun amounting to approx. 55% [2]. The ISBSG report confirms these results: in the situation where the methods based on product functional size are employed in making cost estimation, in 90% of cases the estimates differ from the actual costs not more than by 20%, and among these very cases 70% are accurate to within 10%. Also analysis of the results of 25 studies concerning the reliability of the most important BSS D&EP effort estimation methods, made by the author on the basis of the subject literature [25], revealed that currently the highest accuracy of effort estimations is delivered by the extrapolation methods based on software product size expressed in functionality units.

V. FUNCTIONAL ASSESSMENT OF BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS

It is worth taking advantage of the capabilities offered by software FSM to the assessment of BSS D&EP from the perspective being fundamental to a client – that is from functional perspective.

A. The southernSCOPE and northernSCOPE methodologies

Here is why the FSM concept constitutes basis of the southernSCOPE [13] and northernSCOPE [26] methodologies, supporting the management of BSS D&EP functional scope, i.e., scope measured on the basis of functional size of their product. Fundamental assumptions of these methodologies read as follows (see also [27]):

- Price to be paid by client for software being accomplished within D&EP depends directly on the functional size of project product.
- Estimates are being derived throughout the project's life cycle.
- Structure of changes management promotes proper management of changes being introduced by client to the functional requirements.
- Person responsible for the scope management, the so-called scope manager, ascribed key role in this methodology, should work independently.

Practice shows that the discussed methodologies prove useful in the case of projects aimed at developing or enhancing BSS, regardless of whether or not they have internal or external character. As conditions of the effective use of these approaches are being met in their case; among these conditions are:

- Accomplishment of project within the planned and controlled budget is of key significance, if not a priority, to a client.
- There is an acceptance for the methods of product functional size measurement.
- Functional user requirements can be specified on the level of detailness suitable for the FSMM.
- There is a possibility to reduce the number of changes to the required product functionality appearing upon completion of the requirements specification phase.

Concurrently, with the above methodologies, the author of this paper proposed [27] and verified [28] her own model, designed for the functional assessment of BSS D&EP, named SoftFAM (Software projects Functional Assessment Model). Functional Assessment (FA) of project is understood by the author as its *ex ante* and *ex post* evaluation carried out on the basis of FSM concept. Key attributes of FA include: product functional size (*FS*), work effort, which needs to be spent on *FS* development/enhancement (*E*), and functional productivity (*P*) understood as the ratio of product functional size to the work effort on *FS* development/enhancement (*FS*/*E*) [29], or – being inversion of functional productivity - work effort necessary to achieve functionality unit ($E(u)=E/FS$) that determines work cost per *FS* unit (thus measured with regard to the product size unit, not to the work time unit).

B. Assumptions for SoftFAM

The SoftFAM may occur in the form of full model as well as in one of the simplified variants – thus it has a modular character. The following assumptions were explicitly included to the full variant of the model:

1. Functional assessment consists of at least three stages:
   1.1. Initial functional assessment (FA1). It may take place as soon as at the stage of initiating BSS D&EP thanks to the functional size early estimation rule, having been derived on the basis of benchmarking data [25][30] (the so-called calculations of Function Points Zero – FP0). Yet more accurate estimates are received at the analysis stage where the fundamental FUR are known – they are based on the calculations of FP1 (Function Points One), for which, according to the rules of FSM methods, estimation error up to ±30% is allowed. Estimation made at this very stage should be sufficient for initial planning of project attributes, making initial decision on investment, choosing execution variant as well as for choosing group of providers' offers. Further analytical works involve substantial means, which - according to the ISBSG report [31] - make up even up to approx. 27% of the effort spent during the entire project cycle and thus it is worthwhile to make use of the possibility to rationalize of these activities and decisions already at this very stage.
   1.2. Detailed functional assessment (FA2). For the second time estimation should be carried out when detailed FUR specification is already known, which is upon completion of the analytical stage. At this stage estimations are based on calculating FP2 (Function Points Two), in case of which – in accordance with the FSM methods rules – estimation error should not exceed ±10%. Thus, what should be done is a correction of the initially estimated required functional size and based on this – the required effort and functional productivity. This correction results not only from the fact of FUR changing since the moment of calculating FP1 but also from the change of the error range allowed for *FS* at this very stage and consequently – also for the attributes estimated on the basis of *FS*. Based on estimations being derived at this stage, another functional assessment of the previously selected group of providers' offers should be made so that as a result at most several potential product providers will be chosen following the criteria of such assessment. Selecting one of these providers may depend on other criteria as well – they should regard first of all fulfilling of client's non-functional

requirements. It is important that the required product functional size as well as the offered and approved work cost per functionality unit are reflected in provider's formal commitment to a client, which means formal *ex ante* pricing of the project product.

1.3. Final functional assessment (FA3). For the third time functional assessment should be made upon completion of development/enhancement activities in order to measure the actually delivered *FS*, which is meant to lead first of all to the *ex post* pricing of product on the basis of this size and the approved work cost per functionality unit as well as it is to be used to verify degree of FUR accomplishment by a provider, who thus gains possibility to enhance his software processes. Data obtained this way should be then stored by provider in the organizational benchmarking data repository, especially designed for this very purpose. This is meant for deriving and verifying dependencies being specific to given project organization but also for enhancing FSM methods and effort estimation models. At this stage calculations should take into account the fact that since the moment of making FP2 calculations FUR might have changed. Thus the value of all required attributes needs to be updated.

2. All *r*equired (*FSr*, *Er*, *Pr*), *o*ffered (*FSo*, *Eo*, *Po*) and *re*alised (*FSre*, *Ere*, *Pre*) attributes should be included to the relevant tolerance intervals, dependent on the functional assessment stage, which normalize the ranges of allowed values. The need of taking them into account results both from the limited possibilities to derive accurate estimates, particularly at the initial assessment stage, being caused first of all by the BSS D&EP execution conditions changing over time, as well as by analytical needs. Tolerance intervals should promote rational delineating of required and offered attributes values. They read as follows:

2.1. Product functional size – both required by a client (*FSr*) as well as offered (*FSo*) and realised (*FSre*) by a provider – must be within the range allowed for *FSr*, i.e., [*FS*min, *FS*max], where: *FS*min – stands for minimum while *FS*max – stands for maximum required functional size. Defining of *FS*max results from the fact that, as showed by the Standish Group studies, only about 20% of functions and features specified get ever used [2]. Thus delineating the maximum expected functional size reduces the risk of delivering needless functionality.

2.2. Work effort – both expected by a client (*Er*) as well as offered (*Eo*) and realised (*Ere*) by a provider – must be within the range allowed for *Er*, i.e., [*E*min, *E*max], where: *E*min – stands for minimum while *E*max – stands for maximum effort expected by a client. *E*min should not be lower than the effort enabling for delivering minimum required functional size (*FS*min).

2.3. Functional productivity – both required by a client (*Pr*) as well as offered (*Po*) and realised (*Pre*) by a provider – must be within the range allowed for *Pr*, i.e., [*P*min, *P*max], where: *P*min - stands for minimum while *P*max - stands for maximum productivity required by a client. Having *P*max defined is useful for rational provider offer selection, i.e., from the point of view of limiting the risk of choosing the offer where the productivity would be defined as overstated value. Since such situation would mean that in fact the effort per functionality unit is likely to be exceeded, which would entail the risk of delivering product having functional size lower than the allowed one as the provider would be probably trying not to go over the offered effort. In addition, delineating *P*max is conducive to the increased probability of delivering product of sufficient quality.

Fulfilling these conditions ensures:

- Rationality of client requirements with regard to the functional assessment attributes.
- Conformity of the potential providers offers with rational client requirements concerning functional assessment attributes.
- Conformity of the accomplished project with client requirements concerning functional assessment attributes.

The full variant of SoftFAM comprises at least two stages of estimation (FA1, FA2), within which the ranges of allowed values for functional attributes are being used. Due to the modular character of the presented model there is also the possibility to use its simplified variants, which may be considered for applying in practice keeping in mind, however, the increase of risk caused by such simplification. As indicated by the analysis in [27], level of satisfying client's analytical needs decreases with gradual resignation from, initially, one of the two stages of assessment, next from the intervals of allowed values for functional size, effort and functional productivity, and then with omitting both aspects of the FA. Assessment will be more detailed if a client resigns from the initial stage of estimation thus, however, increasing the risk of making non-rational investment decision due to the estimates being delayed in relation to the possibilities.

C. Verification of SoftFAM

The verification of the full variant of SoftFAM was based on the case study of a dedicated BSS being developed from scratch for the needs of Polish affiliated sales department of some international motor concern and presented widely in [28].

Results of the verification indicate that SoftFAM allows for *ex ante* and *ex post* assessment of BSS D&EP effectiveness, and it also supports *ex ante* and *ex post*

analysis of BSS D&EP economic efficiency. As these results prove that functional assessment allows rationalizing certain practical activities as well as business decisions made on the basis of its criteria. Among such activities are: specification of rational client requirements concerning key project attributes (product size, project work effort, cost and time), evaluation of potential providers offers, comparison of execution variants from the point of view of estimated work costs and the economic efficiency, indicating variant having highest potential efficiency, rational *ex ante* and *ex post* pricing of project product as well as enhancing prognosis concerning future projects by project provider. Among business decisions being supported by functional assessment should be mentioned: client's investment decision about going into the execution of project having expected attributes, selection of the offer being most adequate to his requirements concerning these attributes as well as selection of execution variant having highest economic efficiency.

Moreover, results of the verification also indicate that formal pricing of BSS D&EP product should base on the required size (*ex ante* pricing) and on the actually delivered size (*ex post* pricing) of this product expressed with the use of functionality units and on the work costs per unit being measured with regard to the product size unit – and not on the fixed price contracts nor time and material contracts, most often occurring in the project practice, not only in Poland [14, p. 250], which promote exceeding of the BSS D&EP execution costs.

Because of the above capabilities, the SoftFAM allows for reducing some of the negative phenomena commonly occurring in the Polish practice of such projects execution, showing negative influence on their effectiveness and also on their real efficiency, namely:

- Deliberate lowering of BSS delivery costs by providers in order to win contract for product development (the so-called "price-to-win" technique for product pricing) – thanks to *ex ante* and *ex post* product pricing based on the required and actually delivered product functional size and work cost per functionality unit having been mutually and formally agreed at the stage of provider selection.
- Clients increasing the required functionality during the project lifecycle without relevant reflecting of this change's consequences in the execution costs – as a result of monitoring each change in product functional size and ability to determine this change's influence on total work costs on the basis of the formally agreed work cost per functionality unit.
- Provider in reality delivering product having functionality lower than the required one within the fixed price contracts – client is not obligated to pay for the functionality, which had not been delivered as the *ex post* product pricing is based on its actually delivered functional size.
- Provider delivering functionality (many a time also being lower than the required one) at costs being higher than those expected, which usually takes place in the case of time and material contracts – client does not settle the payment on the basis of project duration but on the basis of actually delivered product functional size and formally agreed work cost per functionality unit.

This is possible thanks to the following rules being used in the full variant of SoftFAM:

- Adopting the allowed tolerance intervals for required, offered and realised FA attributes.
- When choosing offers for project execution, preferring the highest allowed productivity (the lowest allowed effort per functionality unit) instead of the cheapest offers.
- Taking into account the influence of changes in FUR being made during the project lifecycle on product functional size, work effort and functional productivity.
- *Ex ante* and *ex post* pricing of product based on the required and actually delivered product functional size as well as mutually agreed work cost per functionality unit.

Verification of the full SoftFAM indicates that it promotes fundamental factors of the effective execution of BSS D&EP [2] – as it contributes to getting client involved in the project and to the proper management of project scope, as well as to achieving most of the functional measurement goals mentioned in the ISO/IEC 14143 norm, especially in the area of project management [15, Part 6].

Advantage of the full version of SoftFAM over southernSCOPE and northernSCOPE methodologies results from the fact of the model adopting two significant assumptions, not being explicitly specified in these methodologies, namely (see also [27]):

- Need to apply upper bounds of the allowed tolerance intervals for required, offered and realised functional size and functional productivity and lower bounds for work effort.
- Need to employ at least two stages of estimation: first one for proper assessment of the investment decision rationality while second stage – in order to choose suitable software product provider.

Therefore, comparing to these methodologies, using full SoftFAM reduces the risk of choosing inappropriate provider as well as the risk of lowered *ex ante* and overstated *ex post* product pricing, and consequently, it reduces the chance of failing to deliver required functionality and/or to deliver product of insufficient quality. On the other hand, modular character of SoftFAM enables for choosing its variant being most suitable to a given situation – it may be a version based on the simplest criteria, closest to the southernSCOPE and northernSCOPE methodologies.

VI. USAGE OF FUNCTIONAL SIZE MEASUREMENT METHODS BY POLISH BUSINESS SOFTWARE SYSTEMS PROVIDERS

A necessary condition for taking advantage of BSS D&EP functional assessment is to employ software FSM methods. Meanwhile, the author's studies, whose results were widely presented in [4], indicate that the level of using

these methods among Polish BSS providers, although growing, still leaves a lot to be desired.

Surveys that aimed at analysing the level of using the software FSMM by the Polish BSS providers as well as the reasons behind this status quo, were conducted against a background of author's own research concerning the usage of BSS D&EP effort estimation methods. The use of both types of methods was examined in two cycles: at the turn of the year 2005/2006, being the time of economic prosperity, and next at the turn of the year 2008/2009, that is in the initial stage of crisis and increased investment uncertainty associated with it (in order to observe changes, the author originally intended the research to be repeated after 5 years, however radical change in the economic situation worldwide and in Poland persuaded her to undertake it 2 years earlier).

Both research cycles were completed using the method of diagnostic survey: the first cycle analysed responses given in 44 questionnaires (52 questionnaires were sent out) while the second cycle – responses given in 53 questionnaires (62 questionnaires were sent out). Questionnaires were distributed among various Polish dedicated BSS providers, both internal (IT departments in organizations) as well as external (for the most part from SME sector), providing systems for the needs of financial institutions (banks, insurance) departments, trading companies and public administration institutions. In both cycles the overwhelming majority of responses were answered by IT managers or project managers. Each questionnaire included about 30 questions validated by experts; most questions were of open or semi-open character and were divided into two main groups: concerning the usage of the effort estimation methods (answered by all respondents) and concerning the usage of the FSMM (answered only by the respondents familiar with FSMM). It should be stressed that the research was limited only to organizations dealing with D&EP, whose products are dedicated BSS – thus analysis included neither software maintenance, support and integration projects, software package acquisition and implementation projects, nor other software products types.

In the context of the subject matter analysed in this paper fundamental conclusions from these surveys read as follows:

- Considerable part of the respondents declares they do not commonly employ any of the methodology-based approaches to the BSS D&EP effort estimation, in most cases pointing to the "price-to-win" technique as the preferred estimation approach (not methodology-based) when providing software systems for government institutions (because of legal regulations). However, the level of using the BSS D&EP effort estimation methods has increased over the analysed time (from 45% to 53% of the surveyed providers).
- In both research cycles the respondents declared rather widespread usage of at least one of the effort estimation methods, mostly pointing to the expert methods (first cycle: 36%, second cycle: 43% of all respondents), which are burdened with high risk (tests show that the ratio of the effort estimates, being calculated by different experts for the same project may be 1:6 or even 1:12 at the worst [32]).
- FSM methods still place at the penultimate position among five analysed methods used for BSS D&EP effort estimation by the surveyed providers, however the level of using them has increased in the second research cycle (from 20% to 26% of all respondents).
- In both research cycles relatively low popularity of the FSMM results mostly from insufficient familiarity with such methods, but the FSMM awareness has increased over the analysed time (from 27% to 34% of all respondents).
- Percentage of the respondents using FSM methods versus those familiar with them has increased slightly too (from 75% to 78%), which means that the overwhelming majority of those familiar with the FSMM are also employing them.
- In both research cycles as the main purpose of using the FSM methods was considered product size estimation in order to effectively estimate the effort, costs and time frame for the initiated project.
- In both research cycles as the main advantages of the FSM methods were considered the methods objectivity and high usefulness, including most of all possibility to employ them at initial project stages at sufficient accuracy level of estimates, which helps increase the effectiveness of delivering the required functionality on time and within the planned budget. Disadvantages of the FSM methods include first of all high level of difficulty in using them.

As indicated by the above, in the case of all respondents the main reason for relatively low popularity of the FSM methods is that none of the BSS D&EP effort estimation methods is used commonly as well as insufficient familiarity with these methods, whereas among respondents using estimation methods – insufficient awareness of FSMM and at the same time familiarity with other methodology-based approaches. Among providers declaring familiarity with the FSM methods the main reason why they quitted using them is their high difficulty level.

Fundamental purposes for using the FSM methods indicated by the surveyed Polish dedicated BSS providers are presented in Table 3, where they are related to the purposes for using FSM described in the ISO/IEC 14143 norm. Data presented in Table 3 indicate that (see also [4]):

- In both research cycles higher importance is assigned to the purposes of Project Management group.
- Fundamental purpose of using FSMM indicated in both research cycles is product size estimation in order to effectively estimate the effort, costs and time frame for the initiated project, which is the purpose belonging to the Project Management group.
- Among purposes belonging to the Performance Management group, productivity management was indicated as the most important one in both research cycles.

TABLE III. BASIC PURPOSES FOR USING THE FSM METHODS INDICATED BY THE SURVEYED POLISH DEDICATED BSS PROVIDERS

Purpose indicated by Polish BSS providers	2006 (%)	2009 (%)	ISO/IEC 14143 purpose	
Estimation of product size and, based on this, estimation of the effort, costs and time frame for the project being initiated – in order to design own offer as well as for the commissioned applications	100%	100%	Project resource forecasting	Project Management
Supporting decisions about rationality of initiating the projects and way of completing projects (e.g., using own resources or by outsourcing)	56%	64%		
Monitoring progress, costs and time in the project execution	67%	64%	Tracking the progress of a project	
Managing the changes in the required product size and their influence on project work effort	44%	36%	Managing scope change	
Determining degree to which the Commercial-Off-The-Shelf meets functional user requirements	0%	7%	Package functionality fit	
Comparing attributes of the finished project with other projects	44%	50%	Post-mortem analysis	
Managing software development, enhancement or maintenance productivity	78%	86%	Productivity management	Performance Management
Managing software reliability	44%	50%	Quality management	
Managing organization's maturity	0%	7%	Organizational maturity and process capability	
Measuring existing applications in order to determine their value to estimate costs of its potential replacement, reengineering, or outsourcing	56%	64%	Accounting for an organization's software asset	
Making prognosis on the budget necessary to maintain software	33%	29%	Budgeting for maintenance	
Managing the product size and project scope in the client-provider relations	67%	78%	Contract management	
Valuation of applications being executed by other companies	56%	57%		
Determining degree to which the supplied dedicated product meets functional user requirements	0%	14%		

Source: Author's own study with the use of [15, Part 6].

- In 2009, three new items appeared on the list of purposes for using FSMM, namely: managing organization maturity and determining degree to which the supplied dedicated product *or* the COTS meets functional user requirements – in the first cycle they were indicated by none of the surveyed Polish dedicated BSS providers.

The FSM methods stayed practically unknown in Poland until the recession in IT branch that took place in the first years of the 21st century. Although the level of using these methods can be hardly considered high, increase in their popularity, however, may be possibly explained by the four main factors, namely:

- Increasing care about financial means in the times after recession mentioned above (including current crisis where it appears even somewhat stronger).
- Growing competition on the market and increasing market globalization level.
- Growing awareness of clients therefore greater requirements concerning providing justification for the project costs and completion time offered by providers.
- Standardization of the FSM concept and its several methods by the ISO/IEC.

It is hard to compare conclusions coming from the above analysis with the results of other studies carried out worldwide in this area, as the author heard no about studies having similar goals. Yet the fundamental conclusion brought by these surveys agrees with the general conclusion drawn by the Software Engineering Institute (SEI) on the basis of the research attempted to answer the question about today's approach to the measurement of software processes and products: "From the perspective of SEI's Software Engineering Measurement and Analysis (SEMA) Group, there is still a significant gap between the current and desired state of measurement practice. (…) Generally speaking, based on the results of this survey, we believe that there is still much that needs to be done so that organizations use measurement effectively to improve their processes, products, and services." [33].

The research will be continued to keep observing the changes while the research area will be extended as much as possible to other Polish dedicated BSS providers and other economic BSS D&EP aspects.

VII. CONCLUSION AND FUTURE WORK

Summing up it should be stated that the importance of suitable BSS size measurement being significant from the economic point of view results first of all from the necessity to:

1) Increase effectiveness of BSS D&EP execution and reduce losses caused by their low effectiveness. Accurate *ex ante* assessment of project product size, cost and time increases the chance to reach its goal, i.e., on-time delivery of BSS being consistent with client's business requirements without budget overrun. Since the more accurate estimation the lower the risk to go beyond estimates in reality. What's more, such assessment enables to get information about resources that are necessary to deliver product having required functions and features – and it should allow for quitting projects, for which the chance of execution with the resources available proves low, or for correcting resources designed for the projects so that they are closest to the

estimated values. Down to the more accurate investment decisions made on the basis of measurable, objective and reliable criteria it is possible to reduce losses caused not only by abandoned projects and by large scale of overrunning the time and costs of their execution but also resulting from business opportunities lost by clients as a result of delivering products not meeting their requirements.

2) Rational ex ante and ex post pricing of BSS D&EP product. In the Polish practice of the BSS D&EP execution there are two types of client-provider contracts that definitely dominate at the moment, they are: fixed price contract and time and material contract. In the first case price of the project product is calculated on the basis of the assumed fixed costs, which were agreed following the requirements specification. In contracts of another type calculation of the product price is based on the agreed rate for work hour being spent by product provider. It means that work cost per unit is measured not with regard to the unit of product size but with regard to the unit of work time, and therefore this is work time – instead of required or actually delivered product size – that determines the total work costs. Project execution with *ex post* pricing of actually delivered product is still rare, at least in Poland, where we deal with low (however growing) level of the so-called "measurement culture" in software engineering, especially from the functional point of view (see Section 6). Both these approaches to the BSS pricing promote overrunning of budget designed for delivering of product that would meet client's requirements. In case of client-provider contracts based on hourly work rate the provider could extend the time of product execution. Also, there is no guarantee that even extending this time excessively and thus leading to the uncontrolled increase in costs the provider would deliver product of required functionality. In case of fixed price contracts, apart from likely situation where the actually delivered product size may be smaller than the required one, there is also another problem that arises: providers manifest strong resistance to any extension of requirements, being so characteristic of BSS D&EP due to the changeability of business environment. Thus the contracts of this type may prevent cost overrun yet on the other hand they do not guarantee delivering of product having required functions and features at this very cost. Therefore *ex ante* and *ex post* pricing of the BSS, being developed or enhanced, should be based on its size: required (estimated) in the case of *ex ante* pricing and actually delivered (measured) in the case of *ex post* pricing. Consequently, work costs per unit should be related to the product size unit and not to the work time unit. This is what makes pricing have objective and reliable character, as client will get possibility to plan the cost of project execution depending on the outcome this project is expected to bring and, as a consequence of its execution, will pay for the actually delivered size of product and not for his requirements, which provider failed to fulfil (in case of fixed price contracts) or for the provider's extra work time (in case of time and material contracts). It requires adequate measure of software size to be implemented, which may be acquired on the basis of the software functional size measurement concept, having been recently normalised by the ISO/IEC.

3) Proper control over the BSS D&EP execution. Measuring product size and project attributes during project execution helps perceive discrepancies between the reality and the plan, respond to potential threats on a current basis, prevent risk factors and monitor the areas of critical significance.

4) Collecting historical data for BSS D&EP estimation purposes. Measurement of the accomplished BSS D&EP attributes allows for deriving dependencies indispensable for making accurate estimation of similar projects in the future thus leading to the enhancement of estimation models that are based on such dependencies.

5) Improvement of BSS D&EP products and processes. Capability to measure software quality (e.g., reliability, what requires knowing the product size) allows to specify client's quality requirements with the use of quantitative criteria, to carry out measurable assessment of product quality during project lifecycle, thus making it possible to verify whether its level is satisfactory, what may result in undertaking improvement activities, as well as to make quality assessment of the final product. On the other hand, SPA/SPI (Software Process Assessment/Software Process Improvement) models (e.g., CMMI - Capability Maturity Model Integration) are based on the assumption that better software product is achieved by means of the improved software processes [34], whose quality too requires to be assessed. In these models higher and higher importance is attached to the software products and processes measurement.

From the point of view of software organizations the measurement of products and processes should be a standard practice: estimating and measuring product size, process effort, cost and time enable for more effective business activity. Estimating and measurement prove being very important also from the point of view of these organizations' clients, who should be given grounds for making rational investment decision and consequently for choosing variant promoting minimisation of costs at the assumed level of effects (required product size), possibly maximisation of effects (achievable product size) at the assumed costs level (if unexceedable costs were determined *a priori*). Moreover, experience in the Polish market (yet not only in this one) indicates that in the practice of BSS D&EP we still cannot speak about the balance of power between a provider and client. The former often dictates conditions of cooperation, many a time making use of client ignorance, especially with regard to the BSS pricing, imposing – if only client allows for it – contract conditions being favourable for himself.

Change of this situation is possible owing to employing suitable approach to the BSS size measurement, that is functional approach, and thanks to taking advantage of the capabilities offered by FSM concept and methods for the BSS D&EP assessment from the perspective being of key significance to a client. Therefore the author made an attempt to develop SoftFAM – the model of BSS D&EP functional assessment that would allow for evaluating the

effectiveness of their execution, both *ex ante* as well as *ex post*, and for supporting *ex ante* and *ex post* analysis of BSS D&EP economic efficiency. The SoftFAM verification results prove that such model allows rationalizing certain practical activities and business decisions made on the basis of its criteria, as well as it allows for reducing some of the negative phenomena commonly occurring in the practice of such projects execution, not only in Poland.

REFERENCES

[1] B. Czarnacka-Chrobot, "The Economic Importance of Business Software Systems Size Measurement", Proceedings of the 5th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2010), 20-25 September 2010, Valencia, Spain, M. Garcia, J-D. Mathias, Eds., IEEE Computer Society Conference Publishing Services, Los Alamitos, California-Washington-Tokyo, 2010, pp. 293-299.

[2] Standish Group, "CHAOS summary 2009", West Yarmouth, Massachusetts, 2009, pp. 1-4.

[3] David Consulting Group, "Project estimating", DCG Corporate Office, Paoli, 2007: http://www.davidconsultinggroup.com/training /estimation.aspx (20.05.2011).

[4] B. Czarnacka-Chrobot, "Analysis of the functional size measurement methods usage by Polish business software systems providers", in Software Process and Product Measurement, A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, J. Brunekreef, Eds., Proc. of the 3rd International Conference IWSM/Mensura 2009, Lecture Notes in Computer Science, vol. 5891, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 17–34.

[5] T. C. Jones, Patterns of software systems failure and success, International Thompson Computer Press, Boston, MA, 1995.

[6] PCG, "2008 ERP report, topline results", Panorama Consulting Group, Denver, 2008, pp. 1-2.

[7] B. Czarnacka-Chrobot, "Standardization of Software Size Measurement", in Internet – Technical Development and Applications, E. Tkacz and A. Kapczynski, Eds., Advances in Intelligent and Soft Computing, vol. 64, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 149-156.

[8] J. Johnson, "CHAOS rising", Proc. of 2nd Polish Conference on Information Systems Quality, Standish Group-Computerworld, 2005, pp. 1-52.

[9] Standish Group, "CHAOS summary 2008", West Yarmouth, Massachusetts, 2008, pp. 1-4.

[10] Economist Intelligence Unit, "Global survey reveals late IT projects linked to lower profits, poor business outcomes", Palo Alto, California, 2007: http://www.hp.com/hpinfo/newsroom/press/2007/ 070605xa.html (20.05.2011).

[11] Standish Group, "The CHAOS report (1994)", West Yarmouth, Massachusetts, 1995, pp. 1-9.

[12] T. C. Jones, "Software project management in the twenty-first century", Software Productivity Research, Burlington, 1999, pp. 1-19.

[13] State Government of Victoria, "southernSCOPE, reference manual", Version 1, Government of Victoria, Melbourne, Australia, 2000, pp. 1-22.

[14] M. A. Parthasarathy, Practical software estimation: function point methods for insourced and outsourced projects, Addison Wesley Professional, 2007.

[15] ISO/IEC 14143 Information Technology – Software measurement – Functional size measurement – Part 1-6, ISO, Geneva, 2007.

[16] Gartner Group, "Function points can help measure application size", Research Notes SPA-18-0878, 2002.

[17] ISBSG, "The ISBSG report: software project estimates – how accurate are they?", International Benchmarking Standards Group, Hawthorn VIC, Australia, 2005, pp. 1-7.

[18] ISO/IEC 20926 Software and systems engineering - Software measurement - IFPUG functional size measurement method 2009, ISO, Geneva, 2009.

[19] ISO/IEC 20968 Software engineering – Mk II Function Point Analysis - Counting practices manual, ISO, Geneva, 2002.

[20] ISO/IEC 24570 Software engineering – NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis, ISO, Geneva, 2005.

[21] ISO/IEC 19761 Software engineering – COSMIC-FFP – A functional size measurement method, ISO, Geneva, 2003.

[22] ISO/IEC 29881 Information Technology – Software and systems engineering – FiSMA 1.1 functional size measurement method, ISO, Geneva, 2008.

[23] B. Czarnacka-Chrobot, "The ISO/IEC Standards for the Software Processes and Products Measurement", in New Trends in Software Methodologies, Tools and Techniques, H. Fujita and V. Marik, Eds., Proc. of the 8th International Conference SOMET'2009, Frontiers in Artificial Intelligence and Applications, vol. 199, IOS Press, Amsterdam-Berlin-Tokyo-Washington, 2009, pp. 187-200.

[24] P. R. Hill, "Some practical uses of the ISBSG history data to improve project management", International Software Benchmarking Standards Group, Hawthorn VIC, Australia, 2007, pp. 26-30.

[25] B. Czarnacka-Chrobot, "The role of benchmarking data in the software development and enhancement projects effort planning", in New Trends in Software Methodologies, Tools and Techniques, H. Fujita and V. Marik, Eds., Proc. of the 8th International Conference SOMET'2009, Frontiers in Artificial Intelligence and Applications, vol. 199, IOS Press, Amsterdam-Berlin-Tokyo-Washington, 2009, pp. 106-127.

[26] Finnish Software Metrics Association, "nothernSCOPE – customer-driven scope control for ICT projects", FiSMA, March 2007.

[27] B. Czarnacka-Chrobot, "Methodologies supporting the management of business software systems development and enhancement projects functional scope", Proc. of the 9th International Conference on Software Engineering Research and Practice (SERP'10), The 2010 World Congress in Computer Science, Computer Engineering & Applied Computing (WORLDCOMP'10), Hamid R. Arabnia, Hassan Reza, Leonidas Deligiannidis, Eds., Vol. II, CSREA Press, Las Vegas, Nevada, USA, July 2010, pp. 566-572.

[28] B. Czarnacka-Chrobot, "Rational pricing of business software systems on the basis of functional size measurement: a case study from Poland", Proc. of the 7th Software Measurement European Forum (SMEF) Conference, T. Dekkers, Ed., Libreria Clup, Rome, Italy, June 2010, pp. 43-58.

[29] L. Buglione, "Some thoughts on Productivity in ICT Projects, version 1.2", WP-2008-02, White Paper, July 25, 2008.

[30] "Practical Project Estimation (2nd edition): A Toolkit for Estimating Software Development Effort and Duration", P. R. Hill, Ed., ISBSG, Hawthorn, VIC, 2005

[31] ISBSG, "The ISBSG Special Analysis Report: Planning Projects – Project Phase Ratios", International Software Benchmarking Standards Group, Hawthorn, VIC, Australia, 2007, pp. 1-4.

[32] International Software Benchmarking Standards Group: http://www.isbsg.org/Isbsg.Nsf/weben/Functional%20Sizing %20Methods (20.05.2011).

[33] M. Kasunic, "The state of software measurement practice: results of 2006 survey", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2006, pp. 1-67.

[34] CMMI Product Team, "CMMI for Development", Version 1.2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2006, pp. 1-573.