The Effectiveness of Business Software Systems Functional Size Measurement

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

***Abstract*—Execution of software Development and Enhancement Projects (D&EP), particularly those delivering Business Software Systems (BSS) as a product, encounters many problems, which still makes fulfilling of client requirements appear a big challenge for BSS providers. This may be proved by numerous analyses indicating exceptionally low effectiveness of BSS D&EP as compared to other types of IT projects, what - with their significant costs being considered - leads to the substantial financial losses. Author's analysis of fundamental BSS D&EP success factors indicates that one of the most important of them is objective and reliable measurement of such projects' product size, with particular consideration given to client's perspective. Further analyses led author to the conclusion that these conditions are only met by the software product size measure based on its functionality, what is confirmed by the acknowledgement of the so-called software Functional Size Measurement (FSM) concept along with several FSM methods by the ISO and IEC. The paper analyzes and evaluates the potential of effective usage of FSM with regard to BSS, with particular consideration given to the two most popular normalized FSM approaches, namely International Function Point Users Group (IFPUG) method and Common Software Measurement International Consortium (COSMIC) method. This issue is important not only for practical, but also for theoretical reasons, which are caused by the need to satisfy requirements of software engineering as a knowledge discipline having scientific grounds.**

Keywords-business software systems development and enhancement projects; functional size measurement; IFPUG method; COSMIC method; software engineering

I. INTRODUCTION

Majority of application Development and Enhancement Projects (D&EP) fail to meet criteria of their execution effectiveness. As indicated by the results of the Standish Group analyses, success rate for such projects has never gone beyond 35% [1]. It means that majority of them either end up with total failure, or they exceed costs and/or time estimated as well as they lack critical functions/features.

Analyses by T.C. Jones plainly indicate that those application D&EP, which are aimed at delivering Business Software Systems (BSS), have the lowest chance to succeed [2]. 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 [3]. 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. Interesting comparisons of resolution results, cost overrun, time overrun, and expected ROI, made by the Standish Group with regard to three types of order processing application D&EP, are presented in Table 1 [4]. All these sample analytical results unequivocally indicate that, from the provider's perspective BSS D&EP are particularly difficult in terms of management.

Meanwhile BSS are not only one of the fundamental IT application areas; also their development/enhancement often constitutes serious investment undertaking: spending on BSS may considerably exceed the expense of building offices occupied by companies commissioning such

TABLE I. COMPARISONS OF RESOLUTION RESULTS, COST OVERRUN, TIME OVERRUN, AND EXPECTED ROI FOR THREE TYPES OF ORDER PROCESSING APPLICATION D&EP

Resolution	New application development	Package application with modifications	Application modernization
Resolution results comparison			
Successful	4%	30%	53%
Challenged	47%	54%	39%
Failed	49%	16%	8%
Cost overrun comparison			
Below 20%	43%	22%	46%
20% to 50%	21%	36%	29%
51% to 100%	10%	29%	14%
Over 100%	26%	13%	11%
Avarage overrun	44%	47%	34%
Time overrun comparison			
Below 20%	38%	27%	59%
20% to 50%	19%	32%	21%
51% to 100%	30%	31%	12%
Over 100%	13%	10%	8%
Avarage overrun	44%	45%	29%
Expected ROI comparison			
High	11%	34%	52%
Average	66%	57%	37%
Low	23%	9%	11%

Source: [4, pp. 4-6].

systems, and in extreme cases, even 50-storey skyscraper, roofed football stadium, or cruising ship with a displacement of 70.000 tons [5]. Yet quite often client spends these sums without supporting his decision on getting engaged in such investment by proper analysis of the costs, based on the rational, sufficiently objective and reliable basis. The above situation manifests itself in the difference in costs spent by various organizations on similar applications that may be even fifteen fold [6]. It comes from the fact that objective and reliable BSS D&EP effort estimation still appears to be a great challenge to the software engineering. Thus rational *ex ante* as well as *ex post* pricing of BSS, being of key significance to clients, encounters serious problems in practice. In the author's opinion the main reason of that problem is effort calculation on the basis of resources, while such activity should ground on the required (*ex ante* pricing) and actually delivered (*ex post* pricing) BSS size, which determines work effort, what was proved in [7].

Exceptionally low effectiveness of BSS D&EP as compared to other types of IT projects (e.g., maintenance, support, package acquisition, implementation projects, projects delivering other types of software), especially with their costs being considered, 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 (see e.g., [8][9]). What is more, analyses of The Economist Intelligence Unit 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].

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 [11]:

1) Realistic planning, with particular consideration given to the reliable and objective estimates for key project attributes (work effort, execution time and cost), what requires BSS size measurement (see e.g., [12]).

2) Proper project scope management, above all consisting in undertaking small projects, that is projects whose product is characterised by relatively small size (expressed in appriopriate software product size units), what also requires BSS size measurement.

3) Authentic involvement of client in the project (both users and managers) - thus BSS size measurement should be carried out by taking into consideration mainly the perspective of the client, that is with the use of software product size units that are of high significance to him.

The above considerations lead to the conclusion that significant factor of BSS D&EP effectiveness is objective and reliable measurement of their product size, with particular consideration given to client's perspective. As proved in [11], these conditions are being fulfilled only by the measure of product size based on its functionality.

What underlay the search for the right measure of software product size, having been undertaken for several decades already, was not only the need to increase the effectiveness of software D&EP execution but also the requirements of software engineering as a discipline of knowledge where quantifiable approach to software development/enhancement should be of key significance. Definition of software engineering adopted by the Institute of Electrical and Electronics Engineers (IEEE) reads that it is „the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [13]. Quantifiable approach means that the measurement of software processes and products should constitute immanent feature of this discipline of knowledge. That's why over the last couple of years significant intensification of works could have been observed, which aimed to standardize the best practices, especially with regard to software products. Various ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission) norms have been developed as a result of these works, filling an important gap in the software engineering [14]. One of the most important groups of ISO/IEC standards concerns software product size measurement.

II. STANDARDIZATION OF FUNCTIONAL SIZE MEASUREMENT

Many years' verification of reliability and objectivity of various approaches towards software size measurement 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. The concept of the so-called software Functional Size Measurement (FSM) was normalized in the six-part standard ISO/IEC 14143 [15]. First of all, this standard specifies definition of functional size, which is understood as „size of the software derived by quantifying the Functional User Requirements", while Functional User Requirements (FUR) stand for the „sub-set of the User Requirements describing what the software does, in terms of tasks and services" [15, Part 1]. Hence functional requirements in this norm, due to their importance and need to ensure objectivism of measurement, are treated disjointly when combined with other requirements of non-functional character. The elementary unit of FUR used for measurement purposes is called Base Functional Component (BFC). The example of a FUR could be "Maintain Customers", which may consist of the following BFC: "Add a new customer", "Change customer details" and "Delete a customer" [15, Part 1]. On the other hand, Functional Size Measurement Method (FSMM) in the discussed standard was defined as a specific FSM implementation defined by a set of rules, which conforms to the mandatory features of such measurement.

According to the ISO/IEC 14143 norm the process of using FSMM should comprise the following steps [15, Part 1]: (1) defining the scope of FSM, (2) identifying the FUR contained within the scope of FSM, (3) identifying the BFC contained within the FUR, (4) classifying the BFC with regard to their type, (5) assigning appropriate value to each BFC, and (6) calculating functional size.

There are about 25 variants of the FSM techniques having been developed, however only five of them have been now acknowledged by the ISO/IEC as conforming to the rules laid down in the ISO/IEC 14143 norm and certified as international standard, namely: (1) International Function Point Users Group (IFPUG) method, which is approved in the ISO/IEC 20926 standard [16]; (2) Mark II (MkII) function point method proposed by the United Kingdom Software Metrics Association (UKSMA), which offers more detailed measurement comparing to the IFPUG method and is normalized in the ISO/IEC 20968 standard [17]; (3) Netherlands Software Metrics Association (NESMA) function point method, being the simplified version of IFPUG method, which is approved in the ISO/IEC 24570 standard [18]; (4) Common Software Measurement International Consortium (COSMIC) method, which is certified in the ISO/IEC 19761 standard [19]; and (5) FSM method developed by the Finnish Software Metrics Association (FiSMA), which is normalized in the ISO/IEC 29881 standard [20].

The first three methods listed above are accepted by the ISO/IEC not in full versions, as proposed by the organizations developing them, but in part, however in the most important part with respect to the software functional size measurement [15, Part 6] – that is why they are called the first-generation FSMM. In the approaches proposed by IFPUG, UKSMA and NESMA these methods involve also delineating of the so-called Value Adjustment Factor (VAF), which is supposed to adjust functional size being measured with the use of Unadjusted Function Points (UFP) to the environment of specified project by taking technical and quality requirements (i.e., requirements of non-functional character) into consideration [21, Part 5]. Yet this part of these methods has not been approved by the ISO and IEC – as these organizations' assumptions exclude the fact of FSM depending on requirements of this type. On the other hand, the COSMIC and FiSMA methods were recognized as international standard entirely [15, Part 6][20] – that is why they are called the second-generation FSMM.

FSM methods accepted by ISO/IEC differ in terms of software measurement capabilities with regard to various categories of software (i.e., so-called functional domains). Thus prior to choosing given method one should assess its adequacy to the type of software product. In the ISO/IEC 14143 norm it is stated, that [15, Part 6]:

- There are no functional domains constraints for the accepted part of the IFPUG and NESMA methods, although they were developed as intended for measurement of BSS functional size, nor for the FiSMA method.
- The UKSMA method is adequate for the measurement of any type of software provided that the so-called logical transactions may be identified in it. The rules were developed as intended for BSS therefore the method supports neither complex algorithms characteristic of scientific and engineering software nor the real-time systems.
- The COSMIC method is adequate for: data-driven systems (i.e., BSS), time-driven systems (i.e., real-time systems), and hybrid solutions combining both the above (e.g., real-time systems of airline tickets booking). On the other hand there are constraints for software with complex mathematical algorithms or with other specialized and complex rules (e.g., expert, simulation, weather forecasting systems) and for software processing continuous variables (e.g., computer games, musical instruments software).

The ISO/IEC 14143 norm adheres to the ISO/IEC 15939 standard [22], determining general rules and procedures for the software measurement process in compliance with the ISO/IEC 15288 norm [23], which, on the other hand, defines processes of the system's life cycle. One of the steps of the size measurement process defined in the ISO/IEC 15939 standard is procedure of selecting a method that will be used to measure its size. According to this procedure, selection of FSM method being best tailored to the client's needs should consist of the following activities: (1) characteristics of organizational units of software user with regard to the measurement process, (2) identification of their information needs towards measurement process, and (3) selection of appropriate FSMM on the basis of prospective methods identification (for more details see [24]).

Requirements towards appropriate FSM method vary depending on the organization's character. For example, financial institutions usually choose the method, which correctly measures the BSS while chemical company, by reason of its basic activity, would rather require measurement method being suitable for the real-time systems. Choosing method adequate to the needs would also depend on how its result is planned to be used. If an organization intends to use the measurement results also for the purpose of comparing its productivity against industry data, it is recommended to choose the method being relatively popular in the given industry, for which such data exist. In the case it only needs cursory, rough estimation of functional size, the requirements towards appropriate method of its measurement will get reduced (see also [24]).

ISO and IEC allow for selecting method other than the methods approved by them yet they recommend that it conforms to definitional part of the ISO/IEC 14143 norm. It is also recommended to carry out measurement with the use of relevant supporting tools (see e.g., [12]).

III. Functional Size Measurement of Business Software Systems

Thus to measure BSS functional size one may use all FSMM normalized by the ISO/IEC. What's more, this is the need to measure BSS size that was at the basis of FSM concept and methods development. In the context of their FSM it is assumed that software systems of this type are characterized by the following properties [25]:

- Basic purpose of BSS is to acquire, collect and make available data concerning business activity to support this very activity by: keeping data in the ordered way, enabling execution of various inquiries and delivering information supporting the decision-making.

- Functionality of BSS usually is dominated by the need to collect business data of differentiated level of structure complexity and to ensure their integrity and availability in the long run.
- Overwhelming majority among the so-called functional users of BSS are persons (in contrast to things: other software, devices, hardware) who usually enter into direct interaction with the system through the input/output devices. It means that considerable part of functionality is directed towards right proceedings in case of mistakes being made by this type of users and towards helping to use BSS efficiently. If other, equivalent applications or their components cooperate with the measured system, then they also – next to persons – are functional users of this application.
- Different BSS may cooperate (e.g., exchange data) either on-line or in a batch mode.
- In BSS data are usually collected historically, i.e., after the events that took place in real world, taking into consideration the time of current answer and the fact it is given by a person. Data may be processed also in the batch mode. As a rule BSS do not include software used to drive the real-world events in the real time, which is characteristic of real-time systems although it happens that they receive data in the real time (e.g., prices on the market) – as a result they are forced to respond in similar way.
- Business rules governing data manipulation may be sometimes complicated however BSS rarely include a large number of complex mathematical algorithms.
- BSS usually reside in one layer of software, however application layer software depends on software located in other layers (e.g., operating system, device drivers) – otherwise it could not have been used.
- BSS perceived by functional users being persons as an individual application in fact may consist of several equivalent components. Thus separate measurement for each of them may turn out necessary. This applies to the situation where the goal of the product FSM is to get its size for the effort estimation while each component is based on different execution technology.

All FSM methods normalized by the ISO/IEC allow, among others, for:

- Expressing BSS size from the perspective of its functionality - software attribute being of first priority to the client, what promotes his involvement in the project – and this is a fundamental factor of BSS D&EP success (see e.g., [1][8][9]).
- Comparing the actually delivered BSS size with the size required by client, what enables to evaluate the realized project with regard to the actual value of product delivered (for more details see [7]).
- Making BSS size independent of technology used in the project execution – since functional size reflects actual capabilities of the system, which are independent of programming language used.
- The way of calculating BSS size that is independent of the development methodology and of the project's life cycle models as well as of project constrains and developer's experience [26].
- Obtaining sufficiently objective and reliable estimates not only for BSS size, but also for D&EP work effort, cost and completion time relatively early in the project's life cycle [27] – since early estimates of BSS functional size can be based on incomplete FUR (see e.g., [12][26]).
- Estimating size, effort, cost and time of each change to the BSS functional user requirements.
- Determining the effort, cost and time of all project stages - since the BSS size is based on FUR and these are them that decide on the effort.
- Obtaining appropriate economic indicators - since the use of BSS functional size indicates increased productivity in case of the reduction of total costs, resulting from using more efficient programming language (withdrawing the paradox of software size programming units).
- Supporting CMMI-DEV (Capability Maturity Model Integration for Development [28]) - since the FSM is a factor that makes it easier for an organization to achieve subsequent levels of maturity [29].

IV. THE IFPUG METHOD VERSUS COSMIC METHOD

The two most popular normalized FSMM dedicated to business software systems are IFPUG method and COSMIC method. There are obviously certain similarities between them, which most of all include (see e.g., [30][31][32]):

- Common FSM concept, based on similar understanding of the measurement purpose and scope as well as definition of the measured software boundaries.
- The rules of both methods are based on similar, yet not identical, meaning of the terms related to data. What also is convergent is the concept of data transformation as well as of users perceived as recipients of the measured software functionality.
- Occurrence of two phases of measurement: identification of elements, on the basis of which the functional size is determined, and actual measurement, in which they are mapped into this numerically-expressed size. In the IFPUG method, the first of these phases is not described explicite yet it assumes that the measurement is based on the FUR - data models, functions/processes models or windows, screens, forms and reports designs may also be used for this purpose. In the phase of actual measurement, the explicitly described rules of this method are employed towards these elements. In the COSMIC method, the measurement phase proceeds solely on the basis of FUR.
- Similar way of expressing FUR. In both methods, FUR are expressed by means of BFC. In the

approach developed by IFPUG there are 5 types of BFC: Internal Logical Files (ILF), External Interface Files (EIF), External Inputs (EI), External Outputs (EO), and External inQuiries (EQ) [16], whereas in the COSMIC method there are 4 types of BFC: entry, exit, read, and write [19]. However, there is no simple analogy between them as in the COSMIC method data are not measured explicite and they are not distinguished as a type of BFC.

- Both approaches, although in a different way, meet the requirements imposed on FSM methods in the ISO/IEC 14143 norm therefore both were recognized as international standards of FSM (the IFPUG method not in full version - see [16] vs. [21]).

Differences between the discussed methods mostly concern the following:

- Rules of measurement. Fundamental difference in this respect is the fact that the IFPUG method includes general system characteristics (VAF), representing the influence of technical and quality requirements (i.e., requirements of non-functional character) on functional size. This is the reason why this approach has not been approved by ISO/IEC entirely, however taking them into account in calculations is not necessary. What's more, studies have revealed low practical usefulness of VAF to increasing the quality of prognoses. Characteristics of this type do not exist in the COSMIC method where measurement is based solely on FUR.
- Size boundaries for processes/functions. In the IFPUG method, the size of all five BFC is arbitrarily limited thus the software size depends on their number. While in the COSMIC approach there is no upper limit for the process size as it is determined by the number of data movements. On the other hand, the size of COSMIC data movement is 1 CFP (COSMIC Function Point) and does not depend on data to which it pertains, which is the case of processes in the IFPUG method.
- Data inclusion manner. In the IFPUG method, data are included in calculations in a twofold way: separately as internal/external logical files and as file type referenced affecting the process size. In the case of COSMIC method, data are included with each data movement of read or write type of BFC. Thus the use of IFPUG method requires constructing of data model, which in the COSMIC approach is not indispensable however proves useful. In the IFPUG method, data model also provides basis for early estimates while in the COSMIC approach this is process model that is employed for the approximation purposes.
- Benchmarking data resources. Current version of the largest repository with benchmarking data concerning software FSM, that is repository of International Software Benchmarking Standards Group (ISBSG) [33], includes data in nearly 80% pertaining to the software products being measured with the use of IFPUG method while in only 8% to those measured with the use of COSMIC method.

Moreover, in the subject literature, however, in most cases being supported by COSMIC, the following features of this method are pointed out as deciding on its advantage over IFPUG method:

- Broader range of application. The IFPUG method was developed in order to measure BSS, however in its current version no constraints with regard to the measurement of other functional domains were imposed by ISO/IEC. Meanwhile it is often argued that this method does not prove useful in the case of real-time systems size measurement – unlike COSMIC method [31]. According to the author of this paper, such conclusion goes too far both from theoretical and practical point of view although measurement of this type of software using IFPUG method undoubtedly is more complicated as compared to the COSMIC method and therefore it may be less accurate. In publications on the IFPUG method one may find not only the rules but also the examples of employing it in the measurement of real-time systems size (see e.g., [34]).
- Compliance with object-oriented analysis and programming. In this case it is argued that if the COSMIC method was developed much later than IFPUG method it then takes into account modern techniques of FUR description and systems construction, paying attention mostly to the object-oriented approach [35]. However, this in no way proves that there is no possibility to calculate functional size using object-oriented approach to the development based on the IFPUG method – rules of the method and practical examples do indicate such a possibility exists (see e.g., [36]).
- Broader measurement perspective. With the use of IFPUG method, functional size is measured from the perspective of end user while with the use of COSMIC method – from the point of view of the so-called functional user that next to an end user includes also developers, who perceive other applications and devices interacting with the measured software [31]. Perspective limited to an end user only carries some danger of skipping in the calculations of such functionality, which is imperceptible to an end user, however on condition that it is assumed that only a user being a person can be a recipient of functionality. Meanwhile, recognizing the IFPUG method as complying with the ISO/IEC 14143 standard means that definition of user it currently employs is consistent with this notion's definition included in this norm, wherein a user is understood not only as a person but also as a thing (e.g., other applications, devices) that interacts with the measured software [15, Part 1].
- COSMIC approach assumes that typical software is made of layers, for which the rules of proceedings were expressed explicite therefore this method can

be used to measure complex, layered architectures [26].

- In COSMIC approach there are no artificial limits imposed on the size of functional process, that's why the integrity of measure is very good, while in the IFPUG method artificial limits (e.g., weights) limit the size of BFC, so the integrity is limited [26].
- Possibility of faster delivery of results. COSMIC method happens to be regarded as more intuitive, more concise and simpler than IFPUG approach, which should result in quicker delivery of the measurement outcome. Yet this has not been confirmed by the surveys, which indicated that there are no significant differences in the quickness of measurement made with the use of both methods [32]. What is more, even authors of the COSMIC method admit that in case one needs quick measurement with low-quality user requirements, it is simpler (and quicker) to employ IFPUG method – which results from the limited scope of its BFC size, which are easier to be predicted correctly [35]. In this situation using the COSMIC method would require an expert in order to obtain result on the same level of reliability, while this would increase the effort of measurement process. It is worth noting that it applies to the possibility of employing both methods for the estimation purposes: in the original COSMIC method there are limited possibilities to carry out approximate calculations at the early stage of the project's life cycle, or the way of obtaining such calculations is time-consuming, which results from the necessity to base on FUR specification of high level detailness. However, there are some its variants that allow for early estimates of functional size on the basis of incomplete FUR (e.g., Object-Based Approach, Story-Based Approach, and Event-Based Approach) [26].

Organizations employing IFPUG method may face the need to converse results obtained with the use of this method into the results expressed in COSMIC function points (CFP), e.g., due to client's requirements or because of this other method being recognized as more adequate in specific application conditions. And inversely: it may happen there will be the need to converse sizes expressed in CFP into the functional size expressed in IFPUG function points, e.g., due to the need to use generally available benchmarking data for comparison purposes as a serious argument in negotiations with client. However, there is no possibility of exact conversion of the results of both methods using mathematical formula. This results from the fact that BFC of the IFPUG method cannot be exactly translated into BFC of the COSMIC method, and inversely, as well as from the above mentioned differences in measurement rules.

One of the approaches towards conversion is conversion based on statistical formula. Many studies have been initiated in this area, which aimed at gaining adequate statistical formula that would be expressing correlation between the sizes obtained with the use of both methods, however their outcomes differ to a large extent, which may be seen in the examples shown in Table 2. Thus it is advised that an organization facing the need of conversing sizes of its applications makes its own analysis using the regression method in order to derive equation specific to itself, at the same time relying on the right size of the sample of benchmarking data concerning measurement as well as on statistically representative examples, being specific to given organization.

TABLE II. EXAMPLES OF STATISTICAL FORMULAS FOR THE CONVERSION OF THE IFPUG/NESMA METHOD RESULTS INTO THE COSMIC METHOD RESULTS[A]

Author (year)	**Sample size**	**Size range (UFP IFPUG/ NESMA)**	**Formula (regression analysis)**	R^2
Fetcke (1999)	4	40–77	$CFP = 1.1 \times UFP - 7.6$ (UFP – number of IFPUG Unadjusted Function Points)	0,97
Vogelezang Lesterhuis (2003)	11	39–1424	$CFP = 1.2 \times UFP - 87$ $CFP = 0.75 \times UFP - 2.6$ (≤ 200 UFP) $CFP = 1.2 \times UFP - 108$ (>200 UFP) (UFP – number of NESMA Unadjusted Function Points)	0,99
Abran, Azziz, Desharnais (2005)	6	103–1146	$CFP = 0.84 \times UFP + 18$ (UFP – number of IFPUG Unadjusted Function Points)	0,91
Desharnais, Abran (2006)	14	111–647	$CFP = 1.0 \times UFP - 3$ (UFP – number of IFPUG Unadjusted Function Points)	0,93
Van Heeringen (2007)	26	61–1422	$CFP = 1.22 \times UFP - 64$ (UFP – number of NESMA Unadjusted Function Points)	0,97

[A] R^2 is a coefficient of determination describing degree to which the model explains the shaping of the variable being explained – in this case expressing the proportion of deviation in the COSMIC size (in CFP) being explained by the change in the IFPUG/NESMA size.

Source: Author's own analysis based on [32] and [37].

The issue of conversion of the results of both methods requires further investigations, even more as there are no works concerning their latest versions.

Naturally, the discussed methods are not devoid of shortcomings, though. Among objections being most often raised towards both approaches is their relatively high complexity. In fact, when analyzing the rules of both methods (see [21] and [25]) it is hard not to agree with this argument, even despite the existence of various tools supporting them (see e.g., [12]). However, the studies reveal that the time devoted by a specialist (the so-called scope manager) to the delineating of functional size is estimated to be less than 1% of the project life cycle duration [38]. As specialist makes use of the information, which is collected within any methodically-conducted project (especially of data model and functions/processes model), regardless of whether or not the estimation of its attributes is intended. Whichever option is used, the work effort, cost and time involved in the execution of functional size measurement is

 ISBN: 978-1-61208-139-7

meager comparing to these attributes for the entire life cycle of BSS D&EP, not to mention comparing them with possible effects of erroneous decisions. What's more, it's hard to expect that measurement of products of high complexity - and BSS undoubtedly are such products – is going to be effective yet simple task. Other methods of product size measurement are either simple and ineffective (methods based on programming units) or they are neither simple nor effective, especially with regard to the planning purposes (methods based on construction complexity units), not to mention that they lack usefulness from the client's perspective (see also [11]). Hence it is worth to treat the costs of employing FSMM as an investment in the improvement of the software processes in an organization.

V. The Objectivity and Reliability of Business Software Systems Functional Size Measurement

FSM methods, despite relatively high complexity, are used worldwide more and more often (see e.g., [39]), clearly due to their proved effectiveness, especially in case of BSS. For instance, in UK, the Mark II method is a method recommended in the execution of D&EP for the needs of public administration. On the other hand, COSMIC method is a national standard in Japan and in Spain. What's more, these methods are widely employed not only by providers but by clients as well [26].

Research into the objectivity and reliability of software size estimation in functionality units early in D&EP life cycle was carried out by the ISBSG [27]. It analyzed data for 130 projects having product size ranging from 11 to 20 000 function points – and the method decidedly most often used for calculation purposes was the IFPUG method. The researches concerned both prognosis of thus expressed size on the basis of data model as well as methodical calculation of the size on the basis of FUR specification. The ISBSG argued that the product size expressed in functionality units in both cases is estimated in sufficiently objective and reliable way comparing to the calculations made on the basis of end product, whereas estimates derived on the basis of FUR specification are characterized by higher reliability than those obtained on the basis of data model: in the first case 70% of estimates proved not lower than actual product functional size while in the second case this coefficient amounted to 62% of such estimates. In both cases allowable estimation error was assumed only at the level of ±10%. With regard to the objectivity of product size estimation using functionality units it is stated that two specialists obtain results that differ only by ±10%, however on the condition that requirements are specified properly [40]. Conclusions coming from the ISBSG research are also confirmed, among others, by M. Parthasarathy [41, p. 292], who pointed out objectivity and relatively high reliability of the product size estimation based on FSM.

Attempt to carry out similar research was also made by the author of this paper. Within the surveys that aimed at analysing the level of FSMM usage by the Polish BSS providers as well as the reasons behind this *status quo*, she managed to obtain data, which in the case of three BSS providers (small IT company, medium-sized IT company and IT department in a bank) allowed for the IFPUG method reliability analysis (for more details see [39]). For this method the analysis of prognoses accuracy was made in comparison with actual end product size based on the number of unadjusted function points: (1) estimated on the basis of data model and function model, with average complexity being assumed for the function depending on its type; (2) calculated in accordance with method's recommendations on the basis of FUR specification. All products considered in the analysis (provider 1: 11 products, provider 2: 14 products, provider 3: 11 products) are rather relatively small business applications (up to 600 IFPUG UFP). When analyzing reliability, prediction accuracy indicator *PRED(RE)* was employed, which serves to express what in the surveyed cases is the percentage share of these projects/products whose estimates are contained within the assumed estimation Relative Error (RE) related to the actually received value [42]. Thus in order to consider a method reliable the *PRED(30)* was assumed on the level not lower than 80% [43]. What also was calculated is the *PRED(10)*, in order to compare prediction accuracy level with surveys conducted by the ISBSG, in which allowable estimation error was assumed on the level of ±10%.

As indicated by the author's analysis, the IFPUG method in the case of calculations made - according to method's recommendations - on the basis of FUR specification meets the assumed reliability condition. Yet this method does not meet the prediction accuracy condition in any of the analyzed cases, if estimation is made on the basis of data model and function model with average complexity being assumed for the functions. Thus the research results confirm that better effects may be achieved if calculations are made on the basis of FUR specification, which is consistent with the conclusion coming from the ISBSG analyses. Yet the obtained results appear worse in comparison with the ISBSG report - this may result from the fact that in the survey the Polish providers presented the author with data coming from BSS D&EP chosen by chance (not from the best projects, which was probably the case of ISBSG) as well as from scantier experience in using FSMM in Poland.

The FSMM standardized by the ISO/IEC provide also 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 [44] and International Software Benchmarking Standards Group [27], indicate that BSS D&EP in case of which the FSMM were used for effort planning, are characterized 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%. 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, revealed that currently the highest accuracy of effort estimations is delivered by the effort estimation methods based on BSS size expressed in functionality units [12].

VI. Conclusion and Future Work

The measurement of software products is an area of software engineering, which cannot be considered as sufficiently mature not only in terms of practice, but also in terms of knowledge maturity [45]. That's why over the last couple of years significant intensification of works could have been observed, which aimed to standardize the best practices of software products measurement. Some of the undertakings have only just gained recognition, which may be proved by the fact that the latest version of the CMMI-DEV model was strongly focused on measurement. Also various ISO/IEC norms have been developed as a result of these works, filling an important gap in the software engineering (for more details see [14]). One of the most important groups of ISO/IEC standards concerns the software product FSM. Normalization of such measurement is aimed mainly at reducing unnecessary diversity in the area of software size measures, ensuring compatibility between the standardized approaches as well as their usefulness, especially for business software systems.

The ISO/IEC standards for the FSM methods, like the ISO/IEC 14143 norm for the FSM concept, adhere to other standards. The ISO/IEC 15939 offers help in defining the set of measures being adequate to the specific informational needs yet it neither provides the list of such measures nor it recommends specific set of measures for the D&EP. Therefore one may find the opinion that although employing of rules described in this standard is necessary for the measurement process implementation in the organization, these rules *per se*, however, are not sufficient for this purpose [46]. Thus this standard should be linked with other normalized measurement approaches, e.g., the IFPUG method or the COSMIC method.

As indicated by the above analyses, it is hard to unequivocally decide on the advantage of the COSMIC method over the IFPUG method (or inversely) – both have strengths and drawbacks, coming up in the specified problem areas, both have supporters and adversaries. Most probably, COSMIC approach will not totally replace the IFPUG method in the nearest future as this first-generation method has proved being sufficiently objective and reliable approach, at least with regard to the business software systems [47]. Since both approaches prove useful to BSS, this is not the author's intention to solve this dilemma. In any case, from the perspective of requirements made for the methods of BSS size measurement there are no significant differences between the COSMIC and IFPUG method. Generally speaking, functional size obtained with the use of both methods constitutes sufficiently appropriate measure of BSS size and the basis for the estimation of BSS D&EP work effort, cost and duration. These methods, however, are not free of disadvantages therefore they need further improvement, which should benefit to higher accuracy of prognoses being obtained through the methods. Yet the differences between them are significant enough so that they cause problems in proper conversion of their results.

That's why in further works attention should be mostly paid to the possibility of working out the rules of conversion between the results gained with the use of various FSMM, especially these two most popular methods of BSS size measurement. Also surveys that aimed at analyzing the level of using the FSMM by the Polish BSS providers as well as the reasons behind this *status quo* [39] 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 BSS D&EP scope management aspects, with particular consideration of the FSMM reliability.

Undoubtedly, the issue of BSS size measurement is important both for pragmatic as well as for theoretical reasons. Pragmatic reasons rise from the need to increase effectiveness of the execution of BSS D&EP. On the other hand, theoretical reasons are provoked by the need to satisfy requirements of software engineering as a discipline of knowledge – without the possibility of measurement of its basic objects, ensuring objective and reliable analytical criteria, it is hard to regard it as a discipline having scientific grounds. Hence strong significance of appropriate software products' size measurement methods arises, especially with regard to BSS having the lowest chance to succeed.

References

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

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

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

[4] Standish Group, "Modernization – clearing a pathway to success", West Yarmouth, Massachusetts, 2010, pp. 1-16.

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

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

[7] 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.

[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 (03.12.2008).

[11] B. Czarnacka-Chrobot, "The economic importance of business software systems size measurement", Proc. 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.

[12] 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.

[13] IEEE Std 610.12-1990: IEEE standard glossary of software engineering terminology, The Institute of Electrical and Electronics Engineers, New York, 1990.

[14] 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, 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.

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

[16] ISO/IEC 20926 Software and systems engineering - Software measurement - IFPUG functional size measurement method 2009, edition 2, ISO, Geneva, 2009.

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

[18] 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.

[19] ISO/IEC 19761 Software engineering – COSMIC: a functional size measurement method, edition 2, ISO, Geneva, 2011.

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

[21] IFPUG, "Function point counting practices manual, release 4.3", Part 0-5, International Function Point Users Group, NJ, January 2010.

[22] ISO/IEC 15939 Systems and software engineering - Measurement process, ISO, Geneva, 2007.

[23] ISO/IEC 15288 Systems and software engineering - System life cycle processes, ISO, Geneva, 2008.

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

[25] COSMIC, "The COSMIC functional size measurement method, version 3.0, guideline for sizing business application software (version 1.1)", Common Software Measurement International Consortium, Québec, May 2008.

[26] G. Rule, "The most common Functional Size Measurement (FSM) methods compared", Software Measurement Services, St. Clare's, Mill Hill, Edenbrige, Kent, UK, 2010, pp. 1-8.

[27] ISBSG, "The ISBSG report: software project estimates – how accurate are they?", International Software Benchmarking Standards Group, Hawthorn, Australia, 2005, pp. 1-8.

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

[29] C. A. Dekkers and B. Emmons, "How function points support the Capability Maturity Model Integration", in CrossTalk. The Journal of Defence Software Engineering, February 2002, pp. 21–24.

[30] J. Cuadrado-Gallego, D. Rodríguez, F. Machado, and A. Abran, "Convertibility between IFPUG and COSMIC functional size measurements", Proc. of the 8th International Conference on Product-Focussed Software Process Improvement, PROFES 2007, Riga, July 2007, pp. 273–283.

[31] G. Xunmei, S. Guoxin, and Z. Hong, "The comparison between FPA and COSMIC-FFP", Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, 2006, pp. 109–117.

[32] H. van Heeringen, "Changing from FPA to COSMIC. A transition framework", Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, 2007, pp. 143-154.

[33] ISBSG, "Data demographics release 11", International Software Benchmarking Standards Group, Hawthorn, Australia, June 2009, pp. 1-24.

[34] International Function Point Users Group: http://www.ifpug.org/publications/case.htm, Case 4 (03.04.2011).

[35] Common Software Measurement International Consortium: http://www.cosmicon.com/advantagecs.asp (8.06.2009).

[36] International Function Point Users Group: https://www.ifpug.org/publications/case.htm, Case 3 (03.04.2011).

[37] COSMIC, "The COSMIC functional size measurement method, version 3.0, advanced and related topics", Common Software Measurement International Consortium, Québec, December 2007.

[38] P. Morris, "Functional size metrics", Total Metrics, October 2001.

[39] 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.

[40] International Software Benchmarking Standards Group: http://www.isbsg.org/Isbsg.Nsf/weben/Functional%20Sizing%20Methods (21.04.2008).

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

[42] D. V. Ferens and D. S. Christensen, "Does calibration improve predictive accuracy?", in CrossTalk. The Journal of Defence Software Engineering, April 2000, pp. 14—17.

[43] A. Abran and P. N. Robillard, "Reliability of function points productivity models for enhancement projects (a field study)", Proc. of Conference on Software Maintenance 1993-CSM-93, Montreal, IEEE Computer Society Press, Los Alamitos, 1993, pp. 80-97.

[44] 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.

[45] L. Buglione and A. Abran, "The software measurement body of knowledge", Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, January 2004, pp. 84–94.

[46] L. Bégnoche, A. Abran, and L. Buglione, "A measurement approach integrating ISO 15939, CMMI and ISBSG", Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, 2007, pp. 111–130.

[47] F. Vogelezang, "COSMIC Full Function Points. The next generation of functional sizing", Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, March 2005, pp. 281–289.