ISSN22773061

Experimental Evaluation of an Industrial Technique for the Approximation of Software Functional Size

Khaled Almakadmeh[1], Alain Abran[2]

[1]Assistant Professor, Faculty of Prince Al Hussein Bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

Email: khaled.almakadmeh@gmail.com

[2]Professor, École de technologie supérieure, Université du Québec, Montréal, Canada

Email: alain.abran@etsmtl.ca

ABSTRACT

The Early & Quick sizing techniques, built based on ISO standards, have been proposed to derive an early approximation of software functional size when only high-level and incomplete requirements specifications are available. In the literature, there is a lack of research to evaluate the performance of such approximation sizing methods. This paper presents an experimental study to evaluate their reproducibility and accuracy. The experimental results show both poor reproducibility and large inaccurate approximations. In particular, the analysis of the findings indicates that the practitioners could not classify the functional requirements specifications in accordance to their levels of granularity using the rules and the concepts of the Early & Quick COSMIC technique.

Keywords

Early & Quick COSMIC, software functional size, size approximation, incomplete requirements, Function Points, ISO 19761.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 10, No 3

editor@cirworld.com
www.cirworld.com, member.cirworld.com

1. INTRODUCTION

Software project managers and technical leaders use all the available information, including the approximation of the software functional size, to estimate the cost and duration of software projects [1-7]. Estimation of software projects based on measuring software functionality was first proposed by Albrecht [8] in 1979. Several methods containing refinement of Albrecht's concepts and rules are proposed in order specify its use and applicability have been standardized by ISO: COSMIC Function Points [9] and Function Points Analysis (FPA) [10]. Although the functional size of software can be measured accurately with these ISO standards when all the functionality details are available, size measurement is much more challenging and imprecise when the initial requirements are high level and lack details: under these conditions functional size can only be approximated and not measured accurately. Desharnais et al. [11] recommend using functional size approximation techniques for such 'partially documented' software functional requirements specifications.

Functional size approximation techniques can be classified in two (2) main categories, according to Meli [12]:

A. Direct approximation techniques

Direct functional size approximation techniques adopt the "expert opinion" approach, which depends completely on the expertise of the individuals responsible for the approximation of software functional size. This means that these approximations may be influenced by many subjective factors, like personal relationships in the case of collaborative teams, contractual aspects of the task which commonly affect team performance. The direct approximation techniques it may result in reasonable functional size approximations, but it is challenging to recognize when they reasonable, and when they are not. Examples of such techniques include the following:

- Analogy-based approximation technique [13], in which a repository of measured software applications is used. The approximator looks for 'similar' pieces of software, calculates their average size, and then assigns an approximate value to the piece of software that he is approximating. However, the accuracy of this technique is poor [14].
- Delphi technique [15], which considers a group approximation approach, rather than an individual one. For example, each individual involved in the approximation constructs an anonymous approximation, and then these individual approximations are combined to achieve an overall size approximation as a group estimate. However, the results are difficult to justify, and this technique is not recommended for software enhancement projects [12].
- Three-point approximation technique [16], in which the functional size approximations are collected from experts, and then calculates the final approximate functional size is calculated using the formula: ApproxSize = (Min + 4 × MostLikely + Max) / 6, with a standard deviation σ = (Max – Min) / 6. However, the approximator faces the same challenges as with the Delphi technique [12].

B. Derived approximation techniques

Derived approximation techniques are algorithmic or structured, and based on theoretical or statistical models. A few derived algorithmic functional size approximation techniques have been proposed:

- Extrapolative approximation technique [17], which is applied by asking each individual involved in the task to approximate one functional component, and derive the remaining approximations through statistical or theoretical means. However, the accuracy of this technique is poor, strongly depends on distribution profiles, and not recommended for enhancement projects [12].
- Average complexity approximation technique [18], in which functional components are identified in accordance to FPA method [10] in order to approximate functional size according to these components.
- Early & Quick techniques [19-21], which was initially published in 1997 for the original Function Points Analysis sizing method [10]. In this context, the term "Early" refers to the need to obtain functional size approximation before a significant portion of the software requirements is detailed enough for precise measurement, and the term "Quick" means that typically such size approximation must be obtained rapidly, since they must be provided to management within a short time, in spite of the obvious constraints. As the COSMIC measurement method [9] became adopted as an international standard, the initial design of the Early & Quick technique was extended to the COSMIC measurement method. The initial design of the Early & Quick COSMIC technique was proposed in 2000 [17], and subsequently generalized by Conte et al. [21] in 2004. The Early & Quick techniques are presented in more detail in the next section.

Functional size approximation techniques are in great demand to tackle the lack of precise and detailed software requirements specifications at early phases of the software development life cycle. However, a key finding from our literature survey is that while there are 'opinions' on the performance of these approximation techniques, but there is no experimental research evaluating their performance, especially for those based on ISO standards, such as the Early & Quick techniques. The Early & Quick COSMIC technique is selected to evaluate its performance since it refers to the 2nd generation of functional size measurement methods which was developed in the early 2000 to correct weaknesses of the 1st generation of FSM initially developed at the end of the 1970s.

In the ISO International vocabulary of basic and general terms in metrology [22], reproducibility and accuracy are defined as follows:

- Reproducibility, as a condition of measurement: "condition of measurement, out of a set of conditions that includes different locations, operators, measuring systems, and replicate measurements on the same or similar objects"; and

- Accuracy, as applied to measurement: "closeness of agreement between a measured quantity value and a true quantity value of a measurand".

This paper reports on an experiment to evaluate the reproducibility and accuracy of the approximation results with the latest variant of the Early & Quick techniques, the Early & Quick COSMIC technique. This paper is organized as follows: Section 2 presents the Early & Quick techniques. Section 3 presents the context of the experiment designed to evaluate the approximation reproducibility and accuracy of the Early & Quick COSMIC technique. Section 4 presents the experimental results. Section 5 discusses validity threats. Discussion and directions for future work are presented in Section 6.

2. EARLY & QUICK TECHNIQUES: AN OVERVIEW

The Early & Quick (E&Q) techniques [19, 21] define a set of concepts and procedures which combine various functional size approximation approaches to derive an approximation for the functional size of software. They classify functions (i.e. functional processes and data groups in the FPA variant, E&Q FPA) in an analogical and an analytical fashion. E&Q techniques provide the opportunity to use different levels of detail of the software during the functional size approximation process. Therefore, the total amount of functional size uncertainty – within a range of values (minimum, most likely, and maximum) – will be the weighted sum of the uncertainty values of individual components.

An E&Q functional size approximation starts with a breaking down of the structure of the software system under study, an example of which is shown in Figure 1 for the FPA variant, E&Q FPA. This figure depicts the elementary functional processes, as well as the logical data groups and their aggregations that represent different levels of detail. These heterogeneous levels of knowledge make it possible to take advantage of all the information available: in other words, the E&Q techniques enable the use of all the available non detailed information in the functional size approximation process. The elementary functional processes can be grouped into 'small', 'medium', or 'large' typical and general processes. General processes in turn can be grouped into 'small', 'medium', or 'large' macro processes, and the elementary logical data groups can be grouped into multiple data groups.

The functional processes in the E&Q FPA technique correspond to the elementary processes of the standard FPA method: (i.e. External Input (EI), External Output (EO), and External Query (EQ)), and in the E&Q COSMIC technique, they correspond exactly to the functional processes of the standard COSMIC method, without distinction as to their type. Both the IFPUG and COSMIC versions [19, 21] of the Early & Quick technique provide large range of size values, with no reference to the specific rules of either sizing methods. The Early & Quick technique also provides generic definitions that do not exactly map the detailed definitions and rules of either methods. By design, the Early & Quick technique does not have the same level of details as the ISO standards.

Typically, the root is the highest level in the hierarchy (i.e. the application level), and lower levels stem from that root, based on the number of software artifacts under study. The method is applied down through the levels until the approximator decides that it is not useful to proceed with further decomposition (i.e. at the functional process level). It is worth mentioning that all the functions provided by the application must be at leaf level, since there is no explicit functionality at higher levels of the hierarchy. Therefore, a functional approximation of all the leaves provides a bottom-up approximation of the whole tree (i.e. the software application).

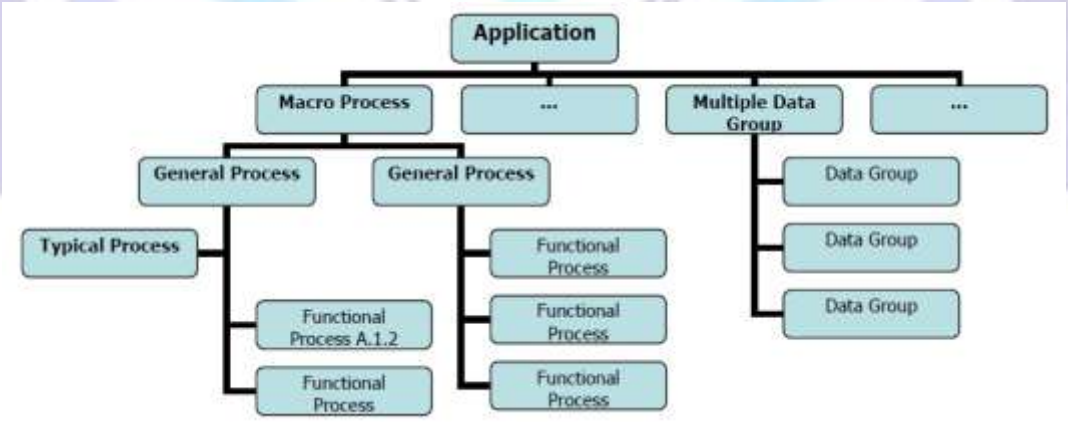

Figure 1. Functional hierarchy in the Early & Quick FPA – an example [19, 21]

Table 1 provides the descriptions and acronyms of the functional levels in the E&Q techniques:

- a functional process (FP) represents the smallest software process with autonomy and significance, and corresponds to the functional process in the standard COSMIC method;
- a general process (GP) consists of a set of two or more average functional processes;
- a typical process is a particular case of a general process and normally consists of a set of the most frequently occurring operational transactions;

- a macro process (MP) consists of two or more average general processes;
- a logical data group (LDG) represents a group of logical data attributes; and
- a multiple data group (MDG) consists of a set of two or more logical data groups.

Table 1. The Early & Quick functional levels [19, 21]

Functional Level	Brief Description
Macro Process (MP)	A set of two or more average GPs. The MP can be likened to a relevant subsystem, or even a bounded application, of an overall Information System.
General Process (GP)	A set of two or more average FPs. The GP can be likened to an operational subsystem, which provides an organized, comprehensive response to a specific application goal.
Typical Process (TP)	A particular case of a GP: the set of the most frequently occurring operational transactions. The TP can be found in two "flavors": CRUD (Create, Retrieve, Update, and Delete), and CRUD plus (CRUD with the addition of List and Report).
Functional Process (FP)	The smallest software process with autonomy and significance. Its FP allows the user to achieve a single business objective at the operational level.
Multiple Data Group (MDG)	A set of two or more LDGs. Its size is evaluated based on the approximated number of LDGs included.
Logical Data Group (LDG)	A group of logical data attributes, it represents a conceptual entity which is functionally significant as a whole for the user.

The E&Q techniques assign a set of size values (minimum, most likely, and maximum) to each leaf in the hierarchy, based on the analytical and analogical table mentioned earlier. Then, these size values are summed to provide the overall approximation result (minimum, most likely, and maximum). It is worth mentioning that the E&Q FPA technique assigns numerical size values to logical data groups and multiple data groups, whereas the E&Q COSMIC technique identifies 'objects of interest', but does not assign any numerical size value to them. Table 2 presents both the component ranges and the size values for the Early & Quick COSMIC approximation technique.

Table 2. Early & Quick COSMIC components and size values [21]

Type	Ordering	Ranges/COSMIC Equivalent	Minimum CFP	Most likely CFP	Maximum CFP
Macro Process	Small	2-4 General Processes	120	285	520
	Medium	4-6 General Processes	240	475	780
	Large	6-10 General Processes	360	760	1300
General Process	Small	6-10 Functional Processes	20	60	110
	Medium	10-15 Functional Processes	40	95	160
	Large	15-20 Functional Processes	60	130	220
Typical Process	Small	CRUD; and CRUD + List	15.6	20.4	27.6
	Medium	CRUD; CRUD + List; CRUD + List + Report	27.6	32.3	42
	Large	CRUD; CRUD + List; CRUD + List + Report	42	48.5	63
Functional Process	Small	1-5 Data movements	2	3.9	5
	Medium	5-8 Data movements	5	6.9	8
	Large	8-14 Data movements	8	10.5	14
	Very Large	14+ Data movements	14	23.7	30

A reference manual for approximating function points at early phases of the software development life cycle using of the E&Q FPA technique is documented in [23]. This manual describes the E&Q FPA technique without mentioning the need for any other guidelines for its application. More specifically, the goals of this reference manual [23] are:

- to provide an exhaustive and clear description of the FPA variant of the E&Q technique (i.e. E&Q FPA); and
- to promote comprehensive and homogenous application of E&Q FPA technique by providing a guideline to approximate function points at early phases of the software development life cycle.

The authors of this reference manual [23] mention that it was designed to be applied by practitioners with an 'average' to 'good' knowledge of standard function point counting (i.e. the standard FPA method) [10]; however, a detailed knowledge of function point counting is not needed, since applying E&Q FPA technique does not need extensive knowledge of the standard FPA rules and practices.

3. EXPERIMENTAL DESIGN

This section presents the eight steps of the experiment reported in this paper - see Figure 2.

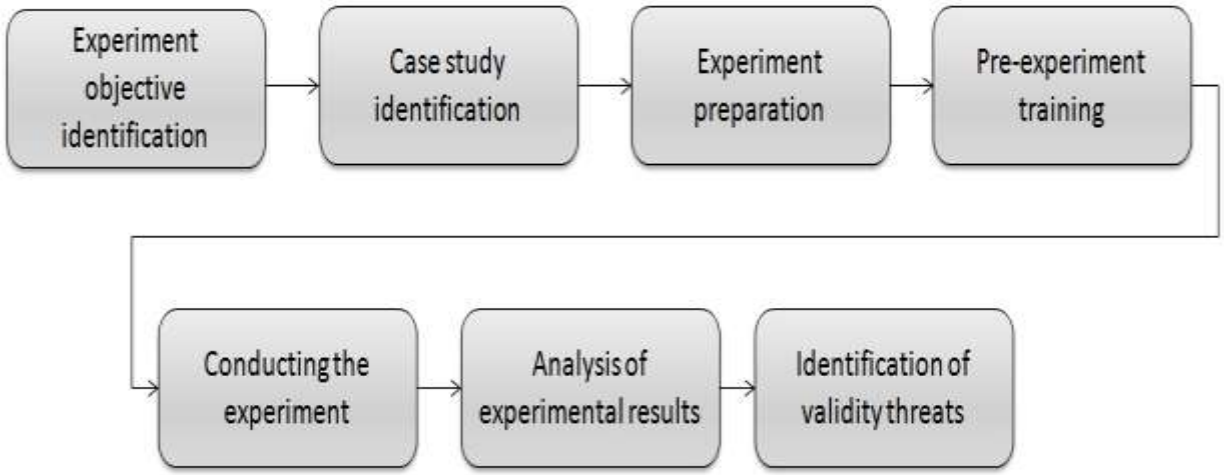

Figure 2. Steps of the experiment

3.1 Identification of experiment objective

The specific experiment objectives were to evaluate the reproducibility and accuracy of the approximation results using only the concepts and rules of the Early & Quick COSMIC technique. In this experiment, 'reproducibility' refers to the degree of closeness between the results of functional size approximation calculated by different approximators on the same case study and 'accuracy' refers to the level of closeness between the results of functional size approximation calculated by different approximators of the case study calculated as described below.

It is worth noting that two preliminary steps are recommended in [24] for the use of the E&Q COSMIC technique [21]. However, no details are provided in [24] on how to apply these steps in practice:

- identification of the levels of granularity of the requirements specifications; and
- identification and use of the size scaling factors.

The experiment is conducted without taking into consideration the two preliminary steps mentioned above for the usage of the E&Q COSMIC technique [21] after fifteen years of the initial publication of the Early & Quick technique [19], such guidelines are still not available to the industry.

3.2 Identification of the case study

The case study in [25] presents the software requirements specifications (SRS) of the first release of the "uObserve" system. This system is intended as a proof of concept for usability testing, and was developed for a research laboratory at the École de technologie supérieure (ÉTS) in Montreal, Canada. The SRS document used in the experiment is written in accordance to the UML 2.0 specifications [26] and IEEE-Std-830 [27] in terms of content and structure. This SRS document [25] consists of 18 pages of textual specifications, divided into three main sections:

- section 1 provides introductory information, including background, software purpose and scope, software objectives, and references;
- section 2 provides a high-level description of the software to be developed, a list of the software functionality and features, and the characteristics of the users; and

- section 3 provides the software functional and non-functional requirements, along with the user interfaces, the hardware interfaces, and the software prototype.

The functional size of the original document of this case study had been previously measured by a team of measurement experts using the international standard for software functional measurement: COSMIC [9]. The measurement experts had an average of fifteen years of industrial experience, were all COSMIC Certified Entry Level practitioners [28] were experienced in functional size measurement, and were active members of the COSMIC Measurement Practice Committee.

Table 3. Functional size of the original case study measured by experts [28]

Expert code	Measured functional size	Average functional size
Expert #1	81 CFP	79.3 CFP
Expert #2	71 CFP	
Expert #3	68 CFP	
Expert #4	97 CFP	

Table 3 presents the functional size calculated by each member of this team of four experts, and the average functional of 79.3 CFP. The differences in the functional size measurement results of each individual measurer are due to measurement assumptions made by the four experts, and do not represent the existence of measurement errors. They do, however reflect the various 'flavors' that develop owing to the assumptions that can be made by different development teams during the development phase of the software [28].

The case study document from [25] describes the functionality of the software system based on 15 use-cases that specify software system functionality in textual form. Table 4 presents the reference classification of the functional components of the software system by applying the E&Q COSMIC technique prepared by the designer of this experiment:

- 8 use-cases are classified as ‘small’ functional processes;
- 5 use-cases are classified as ‘medium’ functional processes; and
- 2 use-cases are classified as ‘large’ functional processes.

On the right-hand side of Table 4 is the approximation of the software functional size in CFP by applying the E&Q COSMIC table (i.e. Table 2). The functional size approximation ranges (Min: 57 CFP, Most-likely: 87 CFP, Max: 108 CFP).

Table 4. The reference classification & size approximation

Macro Process			General Process			Typical Process			Functional Process				Approximation of functional size with E&Q COSMIC (min, most-likely, max)
S	M	L	S	M	L	S	M	L	S	M	L	V. L.	
-	-	-	-	-	-	-	-	-	8	5	2	-	(57 CFP, 87 CFP, 108 CFP)

3.3 Preparation of the experiment

This activity consisted of three sub activities: materials preparation, pilot testing, and call for participation, as follows:

3.3.1 Materials preparation

Prior to the experimental session, the E&Q COSMIC technique was reviewed by the experiment designer in order to provide a description of the concepts and rules, as well as a procedure for applying the technique. The experiment materials included:

- a description of the E&Q COSMIC technique;
- the case study document;
- a defined set of rules; and
- a defined set of participant roles.

The original case study document [25] was considered to be detailed and complete [28] and it specifies in detail the functionality that has to be delivered by the software system. Therefore, it allows a standardized functional size measurement method to be used to obtain an accurate measurement of software system functional size. size. To meet the objective of this experiment in an approximation context, the case study document was modified as follows:

- 6 use-cases were kept ‘as is’ (i.e. without any modification of their specifications);
- 4 use-cases were partially modified, by removing portions of specifications; and

- 5 use-cases were significantly modified by removing use-case specifications entirely.

The average size of 79.3 CFP is the functional size of the uObserve software as developed and implemented: in this experiment, is it indeed the right reference size to evaluate - for approximation purposes - the accuracy of any combination of deletions of details from the set of implemented use-cases of the uObserve software. An alternate strategy - for experimental purposes – would have been to use a set of incomplete requirements which had never been implemented, but this would not provide any basis for evaluating the accuracy of the E&Q COSMIC technique. Opinions of the participants about the altered requirements specifications were not sought since they did not have access to the documentation of the uObserve software as ultimately implemented. This is therefore representative of current practices in industry: approximation is typically done based on incomplete information and no prior knowledge of which specific details are missing.

3.3.2 Pilot testing

A preliminary run of the experiment was performed by the designer of the experiment and an independent expert to identify the potential challenges in the experimental procedures, including:

- the applicability of the SRS to the experiment, in terms of scope and objective;
- an estimate of the time required to conduct the experiment;
- the usability of the data collection forms to be used by the participants in the experiment; and
- verify the correctness of the reference classification of the functional components of the software system (see Table 4) by having an independent expert to perform the experiment activities.

3.3.3 Participants in the experiment

This experiment was stage as part of the 2nd International Symposium in Software Engineering Management (ISSEM 2011). Twelve participants volunteered to conduct the experiment.

The industrial experience profile of the 12 participants involved in the experiment is presented in Figure 3, based on their industrial experience in software engineering topics: the participants had an average of nine years of experience in software requirements analysis and modeling, software development, software documentation, software quality assurance, and software project management. It can be also observed in Figure 3 that participants A1 to A8 have an average of 12 years of industry experience, while participants A9 to A12 have very limited industry experience. In summary, two-thirds of the participants had significant industry experience.

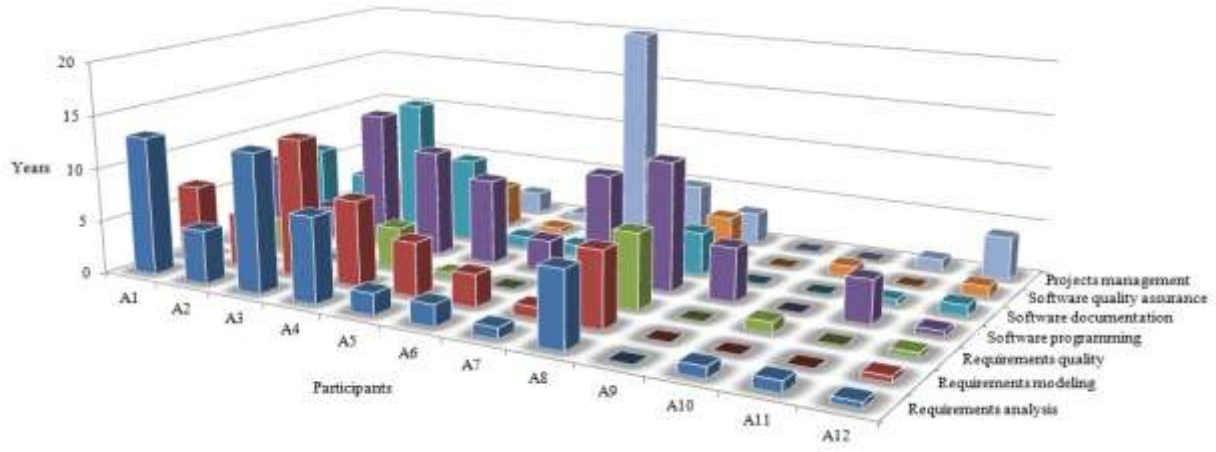

Figure 3. Industrial experience (in years) of the 12 participants in the experiment

3.4 Pre-experiment training

The participants in the experiment were given a one-hour training session, to familiarize them with the E&Q COSMIC technique, the rules to follow, and the roles that would govern the participants' behavior during the experiment. They were then given 30 minutes to read the SRS document.

3.5 Conducting the experiment

The participants were handed a printed copy of the case study document and given one (1) hour to:

- classify the set of software requirements specifications as E&Q COSMIC functional components in accordance to their level of granularity; and then
- use the statistical table of the Early & Quick COSMIC technique (i.e. Table 2) to calculate an approximate functional size of the software system presented in the modified SRS document (see Figure 4).

The following experimental data were to be captured on forms designed for this purpose:

- software process types: Functional, General, Typical, or Macro;
- total number of software processes for each process type;
- total functional size for each process type and total functional size of the software system; and
- total effort required to approximate the functional size.

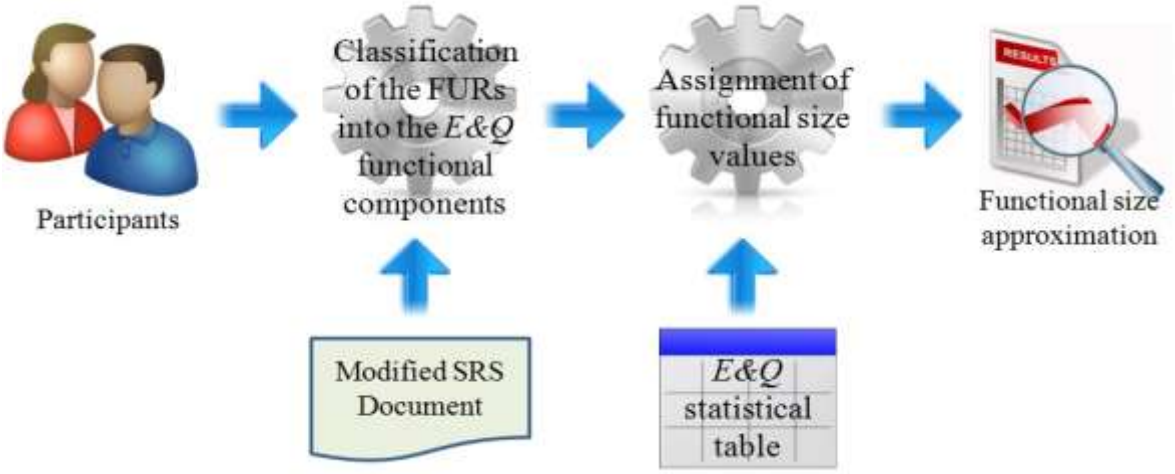

Figure 4. Overview of the activities of the participants in the experiment

4. EXPERIMENTAL RESULTS

This section presents the evaluation of the reproducibility of the functional size approximations calculated by the participants in the experiment, as well as, the evaluation of the accuracy of their functional size approximations with relative to the reference functional size of the case study (see Table 3).

4.1 Descriptive data from the Experiment

To explore whether or not the experience of the participants had an impact on the results of the experiment, the results are presented in two groups:

- results of the 8 participants in Table 5 with an average of 12 years of industry experience; and
- results of the 4 participants in Table 6 with an average of 1 year of industry experience.

Tables 5 and 6 present the classification of the software processes (i.e. functional components) of the case study used in this experiment (i.e. the uObserve software system) and the functional size approximation calculated by each participant using the E&Q COSMIC table. These tables also present the effort expended in minutes by each participant in conducting the experiment.

Table 5. Experimental results of the 8 participants with an average of 12 years of experience

Participant Code	Macro Process			General Process			Typical Process			Functional Process				Functional size (min, most-likely, max) in CFP	Effort (minutes)
	S	M	L	S	M	L	S	M	L	S	M	L	V. L.		
A1	-	-	-	8	-	-	-	-	-	12	17	-	18	(521, 1071, 1616)	55
A2	-	-	-	7	3	-	-	-	-	-	-	-	-	(250, 705, 1250)	50
A3	-	-	-	4	2	-	4	-	-	8	-	-	-	(238, 543, 910)	40
A4	-	-	2	-	-	5	2	4	-	2	3	-	-	(1181, 2369, 3957)	45
A5	-	1	-	1	-	-	1	-	-	4	3	-	-	(299, 592, 962)	30
A6	-	-	-	-	-	-	-	-	-	11	3	1	-	(45, 74, 93)	26
A7	-	2	-	1	8	2	-	-	-	12	-	-	-	(964, 2077, 3450)	45
A8	3	-	-	3	1	-	6	3	-	18	5	-	-	(697, 1454, 2472)	45

Table 6. Experimental results of the 4 participants with an average of 1 year of experience

Participant Code	Macro Process			General Process			Typical Process			Functional Process				Functional size (min, most-likely, max) in CFP	Effort (minutes)
	S	M	L	S	M	L	S	M	L	S	M	L	V. L.		
A9	-	-	-	4	2	-	2	1	-	-	3	2	-	(250, 545, 909) CFP	32
A10	-	9	-	-	1	1	-	-	-	-	1	-	-	(2265, 4510, 7408) CFP	60
A11	1	1	-	-	2	1	-	-	1	3	5	1	-	(581, 1185, 1972) CFP	23
A12	-	-	-	1	-	-	1	-	-	5	2	-	-	(57, 114, 179) CFP	40

4.2 Evaluation of the reproducibility of the functional size approximation

To evaluate the reproducibility of the E&Q COSMIC technique, the approximations of the functional size of the 12 participants are compared to the median functional size approximation. For this data set, the median is represented by approximation of participant A2 (see Table 7). Therefore, the percentage difference in functional size approximation for participant A2 is (Min: 0%, Most-likely: 0%, Max: 0%), and the average percentage difference in approximation is calculated using the percentage difference of the other 11 participants. The plus sign in Table 7 indicates an increase in the percentage difference of the functional size approximation, and the minus sign in Table 7 indicates a decrease.

Table 7. Percentage difference in functional size approximation

Participant code	Approximate functional size using the E&Q COSMIC technique (Min, Most-likely, Max) (in CFP)	Percentage difference in functional size approximation (Min, Most-likely, Max)
A6	(45, 74, 93)	(−82%, −90%, −93%)
A12	(57, 114, 179)	(−77%, −84%, −86%)
A3	(238, 543, 910)	(−5%, −23%, −27%)
A9	(250, 545, 909)	(0%, −23%, −27%)
A5	(299, 592, 962)	(+20%, −16%, −23%)
A2	(250, 705, 1250)	(0%, 0%, 0%)
A1	(521, 1071, 1616)	(+108% , +52%, +29%)
A11	(581, 1185, 1972)	(+132% , +68%, +58%)
A8	(697, 1454, 2472)	(+179%, +106%, +98%)
A7	(964, 2077, 3450)	(+286% , +195%, +176%)
A4	(1181, 2369, 3957)	(+372% , +236%, +217%)
A10	(2265, 4510, 7408)	(+806% , +540%, +493%)
Minimum		**(−82%, −90%, −93%)**
Maximum		**(+806%, +540%, +493%)**
Average percentage difference relative to the functional size approximation of participant A2 (for all 12 participants)		**(+158%, +87.4%, +74%)**
Average percentage difference in the functional size approximations of participants A3, A9, A5		**(+5%, −20.6%, −26%)**

Of the twelve 12 participants in the experiment, the functional size approximations calculated by participants A3, A9, and A5 look like 'reproducible' approximations relative to the median, which is represented by the approximation of participant A2. Even though the functional size approximation of participants A3, A9, and A5 look like 'reproducible' approximations, their approximations of the functional size yield the following average percentage difference of (Min: +5%, Most-likely: −20%, Max: −26%).

Overall, the average percentage difference for the 12 participants is (Min: +158%, Most-likely: +87.4%, Max: +74%) which indicates non-reproducible results for most of the participants. The sources of large variations in the approximations of the functional size presented in Table 7 which yield an average percentage difference of (Min: +158%, Most-likely: +87.4%, Max: +74%) are the incorrect identification of the number of software processes and the incorrect classification made for the software processes (i.e. the functional components) in the case study. Overall, the results presented in Table 7 indicate that the use of the rules and concepts of the E&Q COSMIC technique by the 12 participants does not provide a 'reproducible' approximation of the functional size of the case study used in the experiment.

4.3 Evaluation of the accuracy of the functional size approximation

The functional size approximations of the 12 participants Tables 5 and 6 are first compared with the average functional size of 79.3 CFP (see Table 3) which was measured by the team of experts using the original version of the SRS document. The Magnitude of Relative Error (MRE) equation [29] is used to calculate the accuracy of the functional size approximations (see Table 8) as follows:

- the 1st column presents the functional size values approximated by the 12 participants in the experiment;
- the 2nd column presents the average functional size value measured by the team of experts (considered here as the reference value for accuracy); and
- the 3rd column presents the MREs calculated using the approximate functional sizes from the 1st column and the average measured functional size from the 2nd column.

Table 8. Accuracy of the functional size approximation - 12 participants

Participant code	Approximated functional size using the E&Q COSMIC technique in CFP (min, most-likely, max) (1)	Reference functional size for accuracy criteria (2)	MREs calculated using the values in (1) and (2) (min, most-likely, max)
A1	(521, 1071, 1616)	79.3 CFP	(557%, 1251%, 1938%)
A2	(250, 705, 1250)		(215%, 789%, 1476%)
A3	(238, 543, 910)		(200%, 585%, 1047%)
A4	(1181, 2369, 3957)		(1389%, 2887%, 4890%)
A5	(299, 592, 962)		(277%, 646%, 1113%)
A6	(45, 74, 93)		(43%, 7%, 17%)
A7	(964, 2077, 3450)		(1115%, 2519%, 4250%)
A8	(697, 1454, 2472)		(779%, 1733%, 3017%)
A9	(250, 545, 909)		(215%, 587%, 1046%)
A10	(2265, 4510, 7408)		(2756%, 5587%, 9241%)
A11	(581, 1185, 1972)		(633%, 1394%, 2387%)
A12	(57, 114, 179)		(28%, 44%, 126%)
Average MREs on functional size approximations (all 12 participants)			**(684%, 1502%, 2546%)**
Average MREs on functional size approximations (except participants A6 & A12)			**(814%, 1798%, 3041%)**

Of the functional size approximations calculated by the 12 participants in the experiment, only those calculated by participants A6 and A12 look like 'reasonable' approximations relative to the reference average functional size of 79.3 CFP:

- the functional size approximations of A6 and A12 resulted in 'most-likely' MRE values of 7% and 44%, respectively, while the MREs of the remaining 10 participants vary wildly from +500% to +5000%;
- for these 10 participants, the average MREs of the functional size approximations are: Min: 814%, Most-likely: 1798%, Max: 3041% (bottom line of Table 8).

Overall, for the latter 10 participants, the functional size approximation range of MRE values is (Min: 814%, Most-likely: 1798%, Max: 3041%). Overall, the average MRE for the 12 participants is (Min: 684%, Most-likely: 1502%, Max: 2546%), which indicates extremely highly inaccurate results for most of the participants.

The source of these inaccurate results is the high level of misclassification of the software processes by the participants. For instance:

- participant A1 identified data movements in two software processes as a set of 18 ‘very large’ functional processes and 8 ‘small’ General Processes. This large number of software processes identified by this participant leads to a range of size approximations (Min: 521 CFP, Most-likely: 1071 CFP, Max: 1616 CFP) using the statistical table of the E&Q COSMIC technique (i.e. Table 2). This participant (A1) had 13 years of experience in requirements analysis and modeling at the time of the experiment.
- participants A2 to A8 classified most of the software processes, if not all, at higher levels of classification, and neglected all the available functional specifications of the software processes (i.e. the functional components). Participant A4 had 8 years of experience in requirements analysis and modeling at the time of the experiment, but he classified 13 software processes at the Macro, General and Typical processes levels, and only 5 software processes as functional processes. Participant A4 identified 2 large macro processes as well as five (5) large general processes, leading to a very large range of approximations (Min: 1181 CFP, Most-likely: 2369 CFP, Max: 3957 CFP) using the statistical table of the E&Q COSMIC technique.
- participant A10 identified twelve (12) software processes in the case study, and failed to classify them in their correct class of the E&Q COSMIC functional components, in accordance to their level of granularity. This led to a very large range of approximation sizes: (Min: 2265 CFP, Most-likely: 4510 CFP, Max: 3957 CFP) using the statistical table of the E&Q COSMIC technique.

Tables 9 and 10 present the number of software processes identified by each of the 12 participants in the experiment. The Magnitude of Relative Error (MRE) equation [29] was used to calculate the accuracy of the number of software processes identified. That identified number of software processes was then compared with the reference value of 15 software

processes prepared by the designer of this experiment and verified for correctness by the independent expert. Tables 9 and 10 present:

- the number of software processes identified by the 12 participants (1st column);
- the correct number of software processes identified by the designer of the experiment and verified by the independent expert (2nd column); and
- the corresponding Magnitude of Relative Error (MRE) values (3rd column) calculated using the values from the 1st and 2nd columns.

Only participant A6 in Table 9 was able to identify the correct number of software processes explained in the case study. However, this participant could not classify them in accordance to their levels of granularity in the correct E&Q functional classes – see Table 5.

In addition, only participant A11 in Table 10 was able to identify the correct number of software processes explained in the case study, but he misclassified them, which led to a large range of size approximations (Min: 581 CFP, Most-likely: 1185 CFP, Max: 1972 CFP). Furthermore, the functional size approximations of participant A12 look 'reasonable' (see Table 6). However, participant A12 identified only 9 software processes, instead of the correct number of 15 software processes and could not classify them in accordance to their levels of granularity in the correct E&Q functional classes.

It is worth mentioning that the average MRE of 17% of the participants in Table 10 (i.e. participants with limited industry experience) is great deal better (i.e. a smaller MRE) than the average MRE of 74% for the participants in Table 9 (i.e. participants with 12 years of industry experience). This is because the participants with limited industry experience identified less software processes than the participants with 12 years of industry experience.

Table 9. Number of software processes identified by participants A1 to A8

Participant code	Number of software processes identified (1)	Reference no. of software processes (2)	MRE using values from (1) & (2)
A1	55	15	266%
A2	10		33%
A3	18		20%
A4	18		20%
A5	10		33%
A6	15		0%
A7	25		67%
A8	39		160%
Average MRE on software processes			**74%**

Table 10. Number of software processes identified by participants A9 to A12

Participant Code	Number of software processes identified (1)	Reference no. of software processes (2)	MRE using values from (1) & (2)
A9	14	15	7%
A10	12		20%
A11	15		0%
A12	9		40%
Average MRE on software processes			**17%**

Most of the participants in Tables 5 and 6 calculated inaccurate functional size approximations, since they had incorrectly identified and classified the software processes (i.e. the functional components) of the case study. Overall, the results presented in this subsection indicate that use of the rules and concepts of the E&Q COSMIC technique by the 12 participants did not help them arrive at an 'accurate' approximation of the functional size of the case study used in the experiment.

4.4 Summary of findings

The experimental results presented in sections 4.2 and 4.3 lead to the following findings:

- the functional size approximations calculated by the 12 participants using only the rules and the concepts of the E&Q COSMIC technique did not lead to reproducible or accurate results in this experiment.
- the incorrect identification and classification of the functional components had in impact on the reproducibility and accuracy of the functional size approximations of the functional components.
- no relationship was observed between misclassification of the functional components and amount of details available in the use-cases.
- the participants with extensive industry experience and those with limited industry experience made similar mistakes, in terms of incorrectly identifying and classifying of the software processes in the case study. In other words, the participants with extensive industry experience did not perform better than those with limited industry experience.

5. VALIDITY THREATS

5.1 Construct validity threats

A construct validity threat is associated to the failure of the experimental setting to reflect the conditions of the technique under study (i.e. the E&Q COSMIC technique). In the case of the experiment reported here, the type of reference manual mentioned in [23] for the E&Q COSMIC technique was not available. To mitigate the risk of this type of threat occurring, the experimental material that was made available to the participants in the experiment was designed to contain equivalent information to that in the reference manual of the E&Q FPA technique [23]. In other words, the material used in the experiment contains a complete description of the E&Q COSMIC technique, including the functional size approximation rules and procedures. The participants in the experiment were not able to correctly classify the software processes in accordance with their levels granularity, as described 'as is' in the proposed E&Q COSMIC technique. The preliminary steps recommended in [24] were not taken into account in the experiment design, owing to the unavailability of related guidelines from the literature fifteen years after the initial publication of the E&Q technique in [19] and eight years after the publication of the COSMIC variant in [21].

A second construct validity threat is the restricted time available in which to conduct the experiment. This lack of time prevented the participants from asking for clarification from their colleagues, or from experts in the field, which is common practice in software development organizations.

A third construct validity threat is the inexperience of the participants with the E&Q COSMIC technique. On the one hand, the E&Q COSMIC technique includes few simple rules and concepts. In theory, the eight (8) participants who had an average of twelve (12) years of industrial experience at the time of the experiment should be able to apply the technique correctly without extensive training. On the other hand, if those participants have such difficulty to apply consistently the technique, then it could be that it is the E&Q COSMIC technique itself that may be immature and in needs of much further refinements in its definitions and rules.

Another construct validity threat is about the different expectations of the participants about the SRS document they were given. On the one hand, the participants were handled a modified version of the SRS document where some of the use-cases lack details (i.e. functional specification) which is indeed a challenge. On the other hand, the E&Q COSMIC technique is indeed designed to be used at the early phases of the software development life cycle when the inputs to an approximation come from the customers of the software have ambiguous and incomplete expectations of the software to be developed: the approximators - using the E&Q COSMIC technique - will use such incomplete information to approximate the functional size of software for effort estimation purposes. Some variance in approximators results are expected but a good approximation method should lead to a minimum of variation. Therefore, the experiment presented in this paper reproduces such a context. It must be noted that when all the details (i.e. the functional specifications) of the use-cases are available, the E&Q COSMIC technique becomes irrelevant since with such details, the full detailed ISO measurement rules of standard measurement methods like COSMIC [9] can be applied and precise measurement results can be obtained rather than approximations with ranges of values.

5.2 Internal validity threats

An internal validity threat is associated with any changes in the design of the experiment, such as lack of discussion or clarification during the experimental period, lack of clear data collection procedures, or description of the concept(s) to be evaluated in the experiment, that could affect the validity of the experimental results. To mitigate the risk of this type of validity threat occurring, a one-hour tutorial session was held prior to the experiment to describe its objectives, scope, and rules, as well as the roles of the participants. The designer of the experiment explained the E&Q COSMIC technique in detail, and opened the door to discussion to clarify the activities and materials of the experiment, including a complete description of the E&Q COSMIC technique, a participant experience survey, and data collection forms.

Moreover, the designer of the experiment conducted a pilot test of the experiment by performing the experimental activities of the prior to running the actual experiment. This was done with the help of an independent expert, in order to identify any potential challenge in the experimental procedures, including the applicability of the SRS to the experiment, the time required to conduct the experiment, and the usability of the data collection forms to be used by the participants in

the experiment, as well as to verify the correctness of the reference classification of the functional components of the software system. The independent expert had 20 years of experience in requirements analysis and modeling, 6 years of experience in software documentation and software quality assurance, and 3 years of experience in functional size measurement using the COSMIC measurement method.

The independent expert identified the correct number of software processes (15) explained in the requirements document in [25], and classified 13 of them in the reference classification proposed by the designer of the experiment. However, the independent expert classified 1 of the software processes as a large functional process, whereas this functional process was deemed by the designer of the experiment to be a medium functional process. The independent expert identified 3 more data movements than the designer of the experiment, and this affected the total number of data movements identified in that functional process and resulted in its classification as a large functional process.

Similarly, the independent expert classified another software process as a medium functional process, whereas this functional process was deemed by the designer of the experiment to be a large one. The independent expert identified 2 data movements fewer than the designer of the experiment, and this affected the total number of data movements identified in that functional process and its classification.

The differences in the classification of the 2 software processes were caused by assumptions made by the independent expert for elements in the Graphical User Interface (GUI) of the software system. This affected the identification of the data movements in each software process. These differences should not be considered as misclassifications, because they reflect the various 'flavors' of functional behavior – as a result of the assumptions – of the software [28]. Also, the differences in the classification of these software processes did not affect the final functional size approximation of the software.

Next, the Magnitude of Relative Error (MRE) equation [29] was used to calculate the accuracy of the functional size approximation in Table 4 relative to the reference average functional size of 79.3 CFP (see Table 3). This gives a range of MRE values of (Min: 28%, Most-likely: 8.4%, Max: 36.2%). It is worth noting that the approximate 'Most-likely' functional size of 87 CFP is close to the reference average functional size of 79.3 CFP (i.e. it yields an MRE value of 8.4%). The experiment was designed to apply the E&Q COSMIC technique using a single case study (i.e. uObserve requirements specifications) with a group of 12 participants. In other words, the experiment tested the reproducibility of the classification process with multiple subjects (i.e. participants) using the same requirements document, in order to obtain multiple ranges of functional size on the same requirements document. Assessment of the ranges introduced in the analytical/statistical table was outside the scope of the design of this experiment.

Another potential threat is that all the software processes described in the case study document were only functional processes, and none were higher-level processes (Macro, General, or Typical). To mitigate this threat, future experiments will be designed to apply the E&Q COSMIC technique using multiple case studies with higher process levels (i.e. Macro, Generic, Typical) in order to assess the ranges introduced in the analytical/statistical table (i.e. minimum, most likely, and maximum size values).

5.3 External validity threats

One external validity threat here is associated with the failure to be able to generalize the experimental results beyond the experimental setting. The number of participants in the experiment was limited to 12. However, the experiment involved participants with 2 profiles: experienced participants, and participants with limited experience. Participants A1 to A8 had significant experience in software requirements analysis, modeling, and quality assurance, while participants A9 to A12 had limited experience in these areas. In spite of this, the classification results showed that they all committed similar errors in classifying the software processes of the software system.

6. DISCUSSION AND FUTURE WORK

This experiment looked into the application of the Early & Quick COSMIC technique using a single case study (i.e. uObserve requirements specifications). The experiment tested the reproducibility and accuracy of the functional size approximations with multiple subjects (i.e. participants) using the same requirements document. The functional size approximations produced by 12 participants from the software engineering industry using only the rules and concepts of the E&Q COSMIC technique currently available to the industry did not lead to results that were either reproducible or accurate:

- the average MRE of the functional size approximation of the 12 participants relative to the reference average functional size of 79.3 CFP is as follows: Min MRE 684%, Most likely MRE 1502%, Max MRE 2546%.
- the average percentage difference in functional size approximation relative to the median approximation is (Min: +158%, Most-likely: +87.4%, Max: +74%).
- only 2 participants were able to identify the correct number of software processes: the average MRE of the number of identified software processes of participants with 12 years of industry experience is 74%, and the average MRE of the number of identified software processes of participants with limited industry experience is 17%.
- none of the 12 participants in the experiment classified the identified software processes in the correct E&Q functional classes, in accordance with their levels of granularity.

This experiment could not take into consideration the two preliminary steps recommended in [24] for the application of the Early & Quick COSMIC technique:

- identification of the levels of granularity of the software requirements specifications; and
- identification and use of size scaling factors.

This was mainly because of the unavailability in the literature of guidelines for performing these two steps, even though 15 years has passed since the initial publication of the Early & Quick techniques and 8 years has passed since the publication of the COSMIC variant [21]. The implicit assumption in [24] is that such guidelines would lead to reasonably accurate and reproducible approximations, but there is no supporting evidence that this assumption works as intended.

The industry and the research community recognize the importance of approximate sizing, and size approximation techniques, like the E&Q COSMIC technique, have been proposed, which consist of:

a) a procedural part (i.e. identification and classification of the functional components of software); and

b) assignment of the numerical size values of the classified functional components using tables of size factors, such as the E&Q COSMIC statistical table.

This experiment has used all information available on such a technique, but could not demonstrate that it led to either reproducible or accurate results. All the available information on such a technique has been used in this experiment, but we could not demonstrate that it led to either reproducible or accurate results.

The experiment reported in this paper has been conducted in 2011 with the COSMIC version of the E&Q approximation technique. For the IFPUG version of the E&Q technique, there is an April 2012 edition of a reference manual, version 1.1 of the 'Early & Quick Function Points 3.1 [30]. Commercial training and certification is provided in Italy on the basis of this reference manual. However, there is not yet publicly available information on the performance, in terms of reproducibility and accuracy of approximation results, of practitioners trained or certified on the basis of this reference manual. Organizations providing commercial training should conduct (themselves, or preferably through an independent third party) such experiments with the people they have trained to demonstrate that training improves reproducibility and accuracy of the functional size and the E&Q technique itself can produce reproducible and accurate functional size results.

The Early & Quick COSMIC technique does not require from the participants the knowledge of the detailed rules and definitions of the standard COSMIC and IFPUG method. A future experiment may investigate whether or not people with expertise with either ISO standards would come up with better approximation results.

The software measurement industry has recognized that guidelines are needed, but none has been put into the public domain. Consequently, it has yet to be demonstrated that guidelines lead to reasonably reproducible and accurate results. In summary, there is no documented evidence that:

a) the initial E&Q COSMIC design works as intended; or that

b) the preparatory guidelines designed to support these approximation techniques work as intended.

The software measurement industry and the researchers need to work on developing such guidelines and on verifying that they work as intended. It must be shown that they lead to:

- a reproducible approximation of functional size; and
- a reasonably accurate approximation of functional size.

The methodology used in this experiment can be reused to test the contributions of guidelines as they become available in the public domain. In addition, the case study used in this experiment and the quantitative findings of this research can be used as a benchmark to quantitatively test the contributions of any guidelines that are proposed in the future by researchers or practitioners.

The experiment reported here is part of a research project aimed at designing a framework to identify the levels of granularity of software requirements specifications and to assign scaling factors to them to rank their levels of granularity. In other words, the research objective is to design a framework that takes into account the two preliminary steps recommended in [24], on which no details were provided by the authors of [19, 21] on how to apply these steps in practice.

REFERENCES

[1] Boehm, B. Software Engineering Economics. 1981. 1st ed. USA: Prentice Hall.

[2] Jensen, R. 1983. An Improved Macro-level Software Development Resource Estimation Model. In: Proceedings of 5th ISPA Conference, 88-92.

[3] Park, R. 1988. The Central Equations of the PRICE Software Cost Model. 4th COCOMO Users' Group Meeting, Carnegie Mellon University: Software Engineering Institute, USA.

[4] Putnam, L. and Myers, W. 1992. Measurers of Excellence. Yourdon Press Computing Series.

[5] Jones, C. 1997. Applied Software Measurement. 2nd ed. McGraw Hill.

[6] Chulani, S. 1998. Incorporating Bayesian analysis to improve the accuracy of COCOMO II and its quality model extension. Ph.D. Thesis, University of Southern California, USA.

[7] Boehm, B., Abts, C., Winsor Brown, A., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer D., Steece, B. 2000. Software Cost Estimation with COCOMO II. NJ: Prentice-Hall.

[8] Albrecht, A. 1979. Measuring Application Development. In: Proceedings of the IBM Applications Development, 83-92. Monterey: California.

[9] International Organisation for Standardization. 2011. A Functional Size Measurement Method – ISO19761: COSMIC. Geneva: International Organization for Standardization.

[10] International Function Point Users Group. 2009. Function Point Counting Practices Manual. Princeton, NJ, USA: International Function Point Users Group.

[11] Desharnais, J-M and Abran, A. 2003. Approximation Techniques for Measuring Function Points. In : the 13th International Workshop on Software Measurement – IWSM 2003. Montreal, Canada. p. 270-286.

[12] Meli, R, and Santillo, L. 1999. Function Point Estimation Methods: A Comparative Overview. The European Software Measurement Conference.

[13] Brown, B. 1968. Delphi Process: A Methodology Used For the Elicitation of Opinions of Experts. Santa Monica, California: The RAND Corporation.

[14] Santillo, L. 2012. Easy Function Points – 'Smart' approximation technique for the IFPUG and COSMIC method. The joint conference of the 22nd International Workshop on Software Measurement and the 7th International Conference on Software Process and Product Measurement. Assisi, Italy. 137-142.

[15] Shepperd, M and Schofield, C. 1997. Estimating software project effort using analogies. IEEE Transactions on Software Engineering. 23(1): 736-743.

[16] Keefer, D. and Bodily, S. 1983. Three-Point Approximations for Continuous Random Variables. Journal of Management Science. 29: 595-609.

[17] Tichenor, C. 1998. The IRS Development and Application of the Internal Logical File Model to Estimate Function Point Counts. IFPUG Fall Conference of Use. Orlando, Florida.

[18] Morris, P. 2004. Levels of Function Point Counting. Australia: Total Metrics.

[19] Meli, R. 1997. Early and Extended FP: A New Estimation Method for Software Projects. IFPUG Fall Conference, p. 15-19. Scottsdale, Arizona.

[20] Meli, R. 2000. On the Applicability of COSMIC-FFP for Measuring Software throughout its Life Cycle. In: 11th European Software Control and Metric Conference, 18-20 April, 1-10. Munich, Germany.

[21] Conte, M., Iorio T. and Santillo, L. 2004. E&Q: An Early and Quick Approach to Functional Size Measurement Methods. Software Measurement European Forum – SMEF 2004, Rome, Italy.

[22] International Organization for Standardization. 1993. International vocabulary of basic and general terms in metrology. Geneva, Switzerland: International Organization for Standardization.

[23] Data Processing Organization 2007. Early & Quick Function Points Reference Manual v.1.0 – IFPUG version, Data Processing Organization, Rome, Italy.

[24] Common Software Measurement International Consortium. 2007. Advanced & Related Topics v 3.0, URL: www.cosmicon.com/dl_goto.asp?id=60, last accessed: July 2013.

[25] Trudel, S. and Lavoie, J-M. 2008. uObserve Software Specification v 2.0. Department of Software Engineering and Information Technology, École de Technologie Supérieure, Université du Québec, Montréal, Canada.

[26] Arlow, J. Neustadt I. 2005. UML 2 and the Unified Process. USA: Addison-Wesley, Pearson Education.

[27] IEEE Computer Society.1998. IEEE Std. 830 - IEEE Recommended Practice for Software Requirements Specifications. New York: IEEE Computer Society.

[28] Trudel, S. 2012. Using the COSMIC functional size measurement method (ISO 19761) as a software requirements improvement mechanism. Ph.D. thesis, École de Technologie Supérieure, Université du Québec.

[29] Golub, G. and Van Loan Charles, F. 1966. Matrix Computations. 3rd ed. Baltimore: The Johns Hopkins University Press.

[30] Data Processing Organization 2012. Early & Quick Function Points 3.1 – Reference Manual v.1.1, Rome, Italy.

AUTHORS' BIOGRAPHY

Khaled Almakadmeh

Holds a Ph.D. in Software Engineering from École de Technologie Supérieure (ÉTS) of the Université du Québec since September, 2010. He has a Master's degree in Information Systems Security from Concordia University, Canada (2010), and a Bachelor's degree in Computer Science from Jordan University of Science & Technology, Jordan (2008). He is an Assistant Professor at the Hashemite University (Jordan). His research interests include designing size scaling factors to derive more credible functional size estimates to improve the process of obtaining effort estimates of software systems.

Alain Abran

Holds a Ph.D. in Electrical and Computer Engineering (1994) from the École Polytechnique de Montréal (Canada) and Master's degrees in Management Sciences (1974) and Electrical Engineering (1975) from the University of Ottawa. He is a Professor and the Director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ÉTS) of the Université du Québec (Montréal, Canada). He has over 15 years of experience in teaching in a university environment, and more than 20 years of industry experience in information systems development and software engineering. His research interests include software productivity and estimation models, software engineering foundations, software quality, software functional size measurement, software risk management, and software maintenance management. He has published over 300 peer-reviewed publications and he is the author of the book "Software Metrics and Software Metrology" and a co-author of the book "Software Maintenance Management" (Wiley Interscience, Ed., & IEEE-CS Press). Dr. Abran is co-editor of the Guide to the Software Engineering Body of Knowledge – SWEBOK, and he is the chairman of the Common Software Measurement International Consortium (COSMIC).