Case Study: COSMIC Approximate Sizing Approach Without Using Historical Data

Francisco Valdés Souto
École de Technologie Supérieure, University of Québec
Dept. of Software Engineering
Montréal, Canada
francisco.valdes@spingere.com.mx

Alain Abran
École de Technologie Supérieure, University of Québec
Dept. of Software Engineering
Montréal, Canada
alain.abran@etsmtl.ca

Abstract—In mature engineering disciplines, international consensus can be reached on measurement, as evidenced through established measurement standards. In software engineering, there are 5 functional size measurement standards. These standards work best when the functionality to be measured is fully known, although this usually doesn’t happen in the early phases of software development.

The techniques most often used to approximate the sizing of the software to be developed in the early phases involve historical data. However, gathering historical data is a challenge in itself. This paper proposes the use of a fuzzy logic model to approximate the functional size of a piece of software.

Keywords—EPCU; COSMIC; Approximate Sizing; Fuzzy Logic; Functional Size; FSM.

I. INTRODUCTION

In mature engineering disciplines, there is international consensus on measurement, created through established measurement standards and their respective etalons 1.

In the software domain, quantitative international standards exist only for functional size measurement (FSM), as illustrated by the ISO 14143 series, in which key concepts of the entity and the attribute to be measured are prescribed. To date, the ISO has recognized five FSM methods for software as compliant with ISO 14143:

- One of these is referred to as a 2nd generation FSM method: COSMIC – ISO 19761.

The recognized FSM methods work best when the information to be measured – the defined functionality (i.e. the functional requirements) – is fully known. This is most often not the case in the early phases of software development [22], when only the non-detailed information is commonly found at this stage in the software life cycle [23].

Desharnais et al. point out that, due to a lack of good software documentation, it is not always possible to apply all the detailed rules, and measurers must fall back on approximation techniques for the subsets of requirements without enough details [20]. While the Desharnais et al. research work was related to the IFPUG method, it applies to any recognized functional size measurement methods for software.

Desharnais et al. [20] identify a number of instances where the detailed measurement rules cannot be used:

- The documentation is not precise enough for the application of the detailed measurement rules.
- The amount of work required to apply the detailed measurement rules to obtain precise measures of the software, and the work required subsequently to update the measurement results, is perceived by management as being too expensive.
- Qualified measurers are not available.

Santillo [22] further states that the “functional size of software to be developed can be measured precisely [only] after the functional specification stage: this stage is often completed relatively late in the development process.”

Consequently, a number of researchers have attempted to devise approximation approaches for measuring software functional size by analyzing data from completed projects.

In 1992, Henderson et al. [26] investigated the relationship between function points and lines of code (the Backfiring method), but found wide variability in the results.

In 1997, Meli [28] designed a technique which can produce two distinct types of size approximation:

- Early Function Points (EFP), which is a faster version of the FP IFPUG 4.0 estimation method, and

---

1 The Etalon, or Measurement Standard, is the realization of the definition of a given quantity, with a stated quantity value and associated measurement uncertainty, used as a reference [36].
• Extended Function Points (XFP), which is derived from the EFP after the application of three correction factors. (Note that XFP is not compatible with FP IFPUG 4.0.)

In 2004, Conte et al. [23] applied the Early & Quick (E&Q) technique to the COSMIC method, and indicated that further tests would be needed to make adjustments to the proposal, or to confirm it. Briefly, this technique is based on (direct) analogy and (derived) analysis: it is a human-based size estimation method, and is impacted by the size estimator’s ability to “recognize” the components of the system as belonging to the proposed classes.

In 2003, Deshmukh et al. [20] analyzed two techniques often used in the industry, Function Points Simplified (FPS) [24] and Backfiring [25], using two verification criteria selected from ISO 14143-3. They reported that, in the organizational context of the study, the FPS technique could be used instead of the detailed measurement with an accuracy range of 5%.

In 2007, Vogelezang et al. [14] reported on a study of 50 projects designed to define size bands using the quartile approach, and they analyzed the influence of distinct factors in approximate size estimation. They found that the only factor that had a substantial influence on the size of an average functional process in each of the quartiles is the number of functional processes [14].

In 2011, Santillo [22] proposed using the Analytic Hierarchy Process [27], which is a technique that provides a means for making choices among alternatives, particularly where a number of concurrent objectives have to be satisfied.

The COSMIC document, Advanced and Related Topics [6], describes two approaches for approximate sizing:

• Early sizing
• Rapid sizing

In both these approximate sizing approaches, the first task is to identify artifacts of the software at some higher level of granularity (the standard level of granularity for the COSMIC method is the functional process), and to size them using a locally calibrated scaling factor. These locally calibrated measures can then be converted into COSMIC units (i.e. CFP) using a scaling factor. Obtaining an adequate scaling factor requires an organization’s historical data.

However, as pointed out by Morgenshtern, “Algorithmic models need historic data, and many organizations do not have this information. Additionally, collecting such effort and cost data may be both expensive and time consuming” [2].

So, how can a software developer (an individual or a company) use the approaches for approximate sizing promoted by COSMIC without historical data?

This paper proposes a solution using the fuzzy logic model proposed in [3, 4, 5], which is referred to as the Estimation of Projects in a Context of Uncertainty – EPCU model.

In section II, we describe the COSMIC method and the approaches for approximate sizing promoted by COSMIC, focusing in the Equal Size Bands approach. In section III, we present the EPCU fuzzy logic model. In section IV, we define the EPCU context for estimating the functional size of a specific functional process using the proposed approach to COSMIC approximate sizing. In section V, we describe the experimentation process, and in section VI, we present the results obtained in the experiments. Our conclusions are presented in section VII.

II. COSMIC METHOD

A. COSMIC Method Basics

The COSMIC method applies a set of rules and procedures to software in order to measure its functional size [22]. The concept, definitions, and basic measurement practices can be found in the COSMIC 3.0.1 Measurement Manual [7]. An exhaustive description of the COSMIC method can be found in resources available online at www.cosmiccon.com.

According to the COSMIC Generic Software Model, the following general principles apply to any software that can be measured with the COSMIC method [7]:

a) Software receives input data from its functional users and produces output, and/or another outcome, for the functional users.
b) Functional user requirements of a piece of software to be measured can be mapped into unique functional processes.
c) Each functional process consists of sub-processes.
d) A sub-process may be either a data movement or a data manipulation.
e) Each functional process is triggered by an Entry data movement from a functional user which informs the functional process that the functional user has identified an event.
f) A data movement moves a single data group.
g) A data group consists of a unique set of data attributes that describe a single object of interest.
h) There are four types of data movement. An Entry moves a data group into the software from a functional user. An Exit moves a data group out of the software to a functional user. A Read moves a data group from the software to persistent storage. A Write moves a data group from persistent storage to the software - See Figure 1.
i) A functional process shall include at least one Entry data movement and either a Write or an Exit data movement that is shall include a minimum of two data movements.
j) As an approximation for measurement purposes, data manipulation subprocesses are not separately measured; the functionality of any data manipulation is assumed to be accounted for by the data movement with which it is associated.

All measurement methods have their own functional size unit (FSU); for the COSMIC method, the FSU is the Cosmic Function Point (CFP).

B. COSMIC Approximate Sizing

The COSMIC document, Advanced and Related Topics [6], describes two approaches to approximate sizing:
1) **Early sizing:** this approach is for use early in the life cycle of a project, before the Functional User Requirements (FUR) are detailed and specified.

2) **Rapid sizing:** this approach is for use when there is not enough time to measure the required software using the standard method.

These two approximation approaches can be considered in the early phases of a development project. The general approach to approximate sizing is that some scaling for the type(s) of artifact(s) of the FUR of the software to be measured must be defined locally [6], requiring, for instance, that an average size of the artifacts to be measured be established locally – see Figure 2.

This solution needs historical data in order to produce an adequate scaling factor.

This scaling factor represents the size expected to be measured when the functional user requirements are at the level of detail where an accurate measurement can be made because all the necessary details are available [14].

In [6], four examples of approaches to approximate sizing of new ‘whole’ sets of requirements are presented. Each example is based on two main assumptions:

- There exist historical data to determine the scaling factor (average, or size bands).
- The whole set of requirements is described, or at least there is a commitment about the scope of the software to be developed defined by the requirements described. This means that the requirements are stable.

The four examples described in [6] are as follows:

- **Example 1:** the Average Functional Process approach. The approximate size of the new piece of software is estimated to be equal to (Number of Functional Processes x Average Size from historical data).
- **Example 2:** the Fixed Size Classification approach. A statement of requirements is analyzed to identify the functional processes and to classify each of them according to their size in one of three or more size classes, called, for instance: Small, Medium, and Large. A corresponding scaling factor is then assigned to each functional process.
- **Example 3:** the Equal Size Bands approach. The functional processes are first classified into a small number of size bands. In the next step, the average sizes of each band are calculated (preferably calibrated locally), and then these average functional sizes are multiplied by the number of functional processes of the new piece of software, in each band respectively, to obtain the total estimated approximate size.

This approach can be refined further to give a more accurate result, if sufficient size data are available for calibration.

In the experiment developed here (section V), the values used were those determined by Vogelezang [14] – see Figure 5.

- **Example 4:** the Average Use Case approach. This example extends Example 1 to a higher level of granularity, the use case.

In measurement, there are five scale types: nominal, ordinal, interval, ratio, and absolute [15]. Note that the examples described in [6] use an interval approach:

- for Examples 1 and 4, there is only one interval (the average size), and
- for Examples 2 and 3, several intervals are defined.

Vogelezang et al. [14] name these four examples as follows:

- Average Functional Process approach (Example 1)
- Size Classification approach (Example 2)
- Refined Approximate or Quartile approach (Example 3)
- Average Use Case approach (Example 4)

Example 3, the Equal Size Bands approach, was selected because there is a documented study that supports the data in this example. For Examples 1 and 2, there is only expert opinion to support them, but no documented evidence. For Example 4, the focus is on the size of the use cases, and this approach is similar to that in Example 1.

It is important to note that the intervals mentioned in the examples are stated only qualitatively, and are classified only on an ordinal type of scale, without specified quantitative intervals.

The ordinal scale type used to classify the size bands (Small, Medium, Large, and Very Large) is transformed next into a ratio scale type, represented by an average value calculated from the set of functional processes in the historical data associated with each size band.
III. THE EPCU MODEL

A. Fuzzy Logic

Fuzzy logic (FL) is a superset of conventional (Boolean) logic that has been extended to handle the concept of the partial truth – i.e. truth values between "completely true" and "completely false". It was introduced by Lotfi Zadeh in the 1960s as a means to model the uncertainty of natural language [12, 13, 16].

Fuzzy logic (FL) is a superset of conventional (Boolean) logic that has been extended to handle the concept of the partial truth – i.e. truth values between "completely true" and "completely false". It was introduced by Lotfi Zadeh in the 1960s as a means to model the uncertainty of natural language [12, 13, 16].

Some elements proposed by FL make it useful in the management of uncertainty and imprecision, such as fuzzy set theory, which is basically a theory of classes with boundaries that are not sharp. Some features expressed by Zadeh [12, 13] are considered an extension of classical set theory. Other elements are linguistic variables and the "if-then" rules. The key idea behind these elements is information compression.

A fuzzy system is a mapping between linguistic terms, such as ‘very small’, for example, attached to variables. Thus, an input into a fuzzy system can be either numerical or linguistic, and the same applies to the output [8-11].

Mamdani defined the first fuzzy inference method, with applications for control systems, synthesizing a set of linguistic rules. The Mamdani method is one of the most commonly encountered in the FL literature [34, 35].

B. Overview of the model

In 2007, Valdés et al. [3] proposed an FL-based model referred to as the Estimation of Projects in Contexts of Uncertainty (EPCU) model. This model takes into account:

• the linguistic variables used by experts to describe the input variables in their experience-based estimation process (when these inputs are based on the vague or ambiguous information available to estimate a project); and

• the way experts combine these linguistic values to estimate a project.

This EPCU model has been designed to be used in the early stages in the project life cycle when there is a great deal of uncertainty and the information about the project is vague (i.e. usually described by linguistic variables).

The EPCU model [3-5] is made up of six steps: identification of the input variables, specification of the output variable, generation of the inference rules, fuzzification, inference rule evaluation, and defuzzification (Figure 3).

The first three steps are related to the configuration of the estimation process: this process generates an estimation model, or “EPCU context”. An EPCU context is "a set of variables (inputs and output) and the relations that affect a specific project or a set of similar projects" [4, 5].

a) Identification of the input variables

The goal of this step is to elicit the most significant variables for a project (or a kind of project) from the expert practitioners within an organization.

Also required is to define the membership function domain for representing the opinions of the experienced practitioners about these input parameters. The fuzzy set could be classified as low, average, or high.

Mamdani [34] mentions that the linguistic value determines how fine a decision is obtained, not the cardinality of the support universes (domain functions), and defines the criterion for determining the number of variables to include, such as how many distinct values a human can cope with in charting protocol.

Also required is to define the membership function domain for representing the opinions of the experienced practitioners about these input parameters. By the end of this step, the most significant parameters have been generated, along with their fuzzy sets and the ranges available to each of them.

b) Specification of the output variable

The previous step is repeated for the selected output variable. A classification for the output has also had to be defined using the fuzzy sets that represent it.

c) Generation of Inference Rules

All the fuzzy sets belonging to each input variable must be combined into if-then form:

\[ \text{If } x \text{ and } y, \text{ then } z \]
\[ \text{If } x \text{ or } y, \text{ then } z \]  

(1)
where \( x \) is a fuzzy set for one input variable, \( y \) is a fuzzy set for another input variable, and \( z \) is the fuzzy set for the output variable.

All the fuzzy sets for each input variable must be combined to generate the rulebase.

d) **Fuzzification**

The goal of this step is to obtain fuzzified values as a consequence of opinions put forward by an expert practitioner.

Once the membership function is defined for all the input variables, an expert opinion is requested for each variable. This process will create fuzzy values to be used in the next step, which is to execute the rulebase.

If three fuzzy sets are used for the input variable, the membership function can look something like the example in Figure 4.

e) **Inference Rule execution**

The fifth step consists of executing the rulebase by substituting the fuzzy values obtained for each input variable fuzzy set. Inference Rule execution must follow the rules of fuzzy logic, such as:

\[
\begin{align*}
\text{Value (P or Q)} &= \max \{\text{value (P), value (Q)}\}; \\
\text{Value (P and Q)} &= \min \{\text{value (P), value (Q)}\}
\end{align*}
\]  

(2)

Figure 4. Example of a fuzzy membership function and defuzzification

f) **Defuzzification**

The sixth step is defuzzification, the objective of which is to obtain a crisp value for the final estimate. Examples of defuzzification methods are: Max-Min, Max-Dot, Max-Product, Centroid Average, and Root Sum Square (RSS).

The EPCU estimation generated in the case studies presented next was developed using the RSS, and then computing the fuzzy centroid of the area.

This method was selected because it combines the effects of all applicable rules, scales the functions at their respective magnitudes, and computes the fuzzy centroid of the composite area. It is more complex mathematically than the other methods, but it gives the best weighted influence to all the Inference Rules involved (the Inference Rule “fires” the fuzzy set, in the specialized vocabulary).

The steps for obtaining the crisp value are:

1. Obtain the strength for each fuzzy set belonging to the output membership function (RSS). Considering the values obtained in the Inference Rule execution step, the strength for each fuzzy set defined for the output variable is obtained with the following formula:

\[
FS_k = \left( \sum R_i \right)^{0.5}
\]

(3)

where: \( FS_k \) is the fuzzy set defined by the same linguistic value, \( R_i \) is the rule that fired a specific fuzzy set, \( i \) is the number of rules defined in the rulebase.

2. Obtain the fuzzy centroid of the area. The weighted strength of each output member function is multiplied by its respective output membership function center points, and then all these results are summed. The area obtained is divided by the sum of the weighted member function strengths, and the result is taken as the crisp output.

\[
\text{Crisp Value } (FS_k) = \text{Centroid} = \frac{\sum (FS_k \_center * FS_k \_strength)}{\sum (FS_k \_strength)}
\]

(4)

where:

- \( FS_k \) is the fuzzy set defined by the same linguistic value,
- \( k \) is the number of fuzzy sets for the output variable.

C. **Benefits of the EPCU Model**

In the early phases of the project, most of the variables are linguistic, or qualitative, and more often than not the estimates are developed in an environment of uncertainty. The EPCU model can be used in the early phases of the software development life cycle: in the experiments reported [5] the performance of the EPCU estimation process for most of the projects is significantly better than that of the experience-based estimation approach, based on the quality criteria used. When the performance is better, the difference is small using the experienced-based estimation approach, and so the performances can then be considered equivalent.

Consequently, the use of the EPCU model in the early phases is preferable to the experience-based estimation approach, under similar experimental conditions.

The systematic replication of estimation experience is defined as “the use of the expert’s experience by other people with distinct experience and skills” [5]. By comparison, it can be observed that the EPCU model enables a systematic replication: whatever the level of skill of the people who assign the values for the input variables, the EPCU model generates estimates with less dispersion than the experience-based approach for the projects analyzed.

It is important to note that, while accurate historical data are not required when using the EPCU model, expert experience is required for setting up the configuration of the EPCU contexts to be used for estimation purposes. Once configured, the EPCU
estimation model can be used by those who lack experience in the type of projects to be estimated in a specific EPCU context.

IV. COSMIC APPROXIMATE SIZING USING THE EPCU MODEL

In this section, the EPCU context for approximating the functional size for a specific functional process is defined.

As mentioned above, the approximate sizing approaches defined by COSMIC and described in the examples [6] are based on two main assumptions:

- Historical data exist for calculating the scaling factor (average, or size bands).
- The whole set of requirements is described, or at least there is a commitment, defined by the requirements, about the scope of the software to be developed.

It is clear that, in practice, meeting both assumptions early on in the project is difficult; therefore, the approach proposed in this paper is aimed at addressing the first assumption.

Relative to the two approaches for approximate sizing stated in COSMIC’s Advanced and Related Topics document [6], the proposed approach is related to the Rapid sizing approach, to be used when there is not enough time to measure the required software using the standard method.

To tackle the historical data issue, and considering that there is no universal average functional process from which a scaling factor for early size measurement can be derived [14], we take the Refined Approximate, or Quartile, approach (Example 3), defined by Vogelezang et al. [14], as the basis for the COSMIC approximate sizing task using the EPCU model approach for business applications.

Vogelezang [14] used measurements on 37 business application development projects, each having a total size greater than 100 CFP. The quartile values from this dataset are as follows: Small = 4.8 CFP, Medium = 7.7 CFP, Large = 10.7, and Very Large = 16.4 CFP [14] – see Figure 5.

In order to manage the uncertainty and to define a model that could be used without historical data, a fuzzy logic context is defined, using the EPCU model.

1) Identification of the input variables

Two input variables were considered necessary to evaluate a functional process, based on the experience of the researcher:

1. Variable 1: the functional process size (subjective, experience-based), and
2. Variable 2: the quantity of objects of interest related to the functional process (subjective, experience-based).

Even when input variable 1 is the same as the variable to be estimated, the input value is assigned in terms of the linguistic values defined, not in terms of functional size (CFP).

The definition of an Object of Interest in COSMIC is: “Any ‘thing’ that is identified from the point of view of the Functional User Requirements.

That ‘thing’ could be any physical thing, as well as any conceptual object or part of a conceptual object in the world of the functional user about which the software is required to process and/or store data. [7]

For the two variables defined, linguistic values (fuzzy sets) were defined: for variable 1, Small, Medium, Large, and Very Large; for variable 2, Low, Average, and High, and the domain for the membership function is in a range from 0 to 5 ∈ R.

The membership function for the input variables is defined as in Figure 4.

2) Specification of the output variable

We define the output variable considering a range that includes the values defined for the quartiles, that is, the average sizes of bands, where each band contains 25% of the number of functional processes of the software [6] – see Figure 5.

Considering all the possible values across the dataset in Vogelezang et al. [14], the range defined by the quartiles are [4.8 CFP to 16.4 CFP] and the maximum size (28.9 CFP for the Very Large category).

According to Santillo [22], “theoretically a functional process can be assigned any size expressed in CFP (from 1 to no theoretical limit – they are not bounded, but, in practice, they are expected to have some sort of “natural” upper boundary, or cutoff).”

Because the Vogelezang [14] dataset is not accessible, the assumption about the last category (Very Large) is that the average value (16.4) correctly represents the full dataset, which means that most of the sizes are around the average, and the 28.9 CFP size could be an outlier.

Two distinct contexts have been determined. For the first context (Context A), the range for the output variable membership function is from 2 CFP to 28.9 CFP, and for the

---

2 Mobile technology excluded
3 Principle (i) of the COSMIC Generic Software Model [7]: a functional process shall include a minimum of two data movements.
second context (Context B), the range for the output variable is from 2 CFP to 16.4 CFP. Both contexts are defined by four linguistic values (fuzzy sets): Low, Average, High, and Very High.

The idea is to identify a natural cutoff for the functional process size for the output variable range, in particular for the largest category.

3) Generation of Inference Rules
All the fuzzy sets belonging to each input variable are combined into if-then form.

With the definition of the first three steps, the EPCU context for estimating the CFP for a specific functional process has been defined: the final value is in the 2 CFP to 28.9 CFP range from for Context A, and in the 2 CFP to 16.4 CFP range for Context B, with four fuzzy sets representing four linguistic values for the output variable. The EPCU context has two input qualitative variables that must to be assigned a value in a 0 to 5 unit range, this value is fuzzified into three linguistic values.

V. EXPERIMENT
In order to analyze the performance of the proposal described in section IV, an experiment was designed.

Because the performance of estimation models is usually evaluated with finished projects, such projects are needed.

As a result, we designed the experiment to test the model in a context similar to that of the early phases.

The steps in the experiment are the following:

A. Defining a Measurement Reference
In order to obtain a reference with known functionality, we selected the C-Registration System (Course Registration System). The requirements for this software system were proposed as an initial measurement reference (etalon) for COSMIC in [19]. The idea was to obtain a whole set of (stable) requirements for a known project, along with its measurement results in COSMIC CFP.

The C-Registration System is documented in the Rational Unified Process (RUP Version 2003.06.00.65) document as an example of a website project.

1) C-Registration System
The C-Registration System will enable students to register for courses online. It will also allow professors to select their teaching courses and to maintain student grades.

The C-Registration System Use Case Diagram is shown in Figure 6.

The functional processes identified in the case study are listed below:

1. Actor types his/her name and password on the login form
2. Add a Professor
3. Modify a Professor
4. Delete a Professor
5. Professor selects/de-selects his/her courses to teach
6. Add a Student
7. Modify a Student
8. Delete a Student
9. Create a Schedule
10. Modify a Schedule
11. Delete a Schedule
12. Registrar starts the Close Registration functional process
13. Professor submit grades
14. Student View Report Card

The functional size of the C-Registration System is 107 CFP. This case study is available at www.cosmicon.com.
was given to provide them with a preliminary idea about the system functionality with stable requirements.

In addition to the requirements section, a form with a list of fourteen functional processes identified by the case study was provided to the practitioners – see Table 2.

The form with the list of functional processes was sent by mail to 25 practitioners; however, only nine answers were received.

### Table 2. Experiment Information Request

<table>
<thead>
<tr>
<th>Functional Process</th>
<th>Classification (SMALL, MEDIUM, LARGE)</th>
<th>The functional process size (subjective, experience-based)</th>
<th>The quantity of objects of interest related to the functional process (subjective, experience-based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Actor types his/her part</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Add a Professor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Modify a Professor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Delete a Professor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Professor selects/does not selects his/her courses to teach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Add a Student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Modify a Student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Delete a Student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Create a Schedule</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Modify a Schedule</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Delete a Schedule</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Registrar starts the close Registration functional process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. Professor submits grades</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Student View Report Card</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C. Gathering Information

Once the practitioners had become familiar with the C-Registration System documentation and the survey form (Table 2), they were asked to perform the following (full data shown in Appendix A):

1. Classify each of the fourteen functional processes using the linguistic values: Small, Medium, Large, and Very large.

2. Assign values for the two input variables previously defined from the EPCU context (section IV) for each of the fourteen functional processes: the subjective relative functional size of the functional processes and the subjective number of objects of interest in each functional process, each within the range of 0 to 5 i.e., $R$.

The practitioners did not know the EPCU model, nor did they participate in the definition of the EPCU context, and they were not familiar with the COSMIC method.

VI. DISCUSSION

A. Information Analysis

Analysis of the information gathered from the experiment uses the following quality criteria: Magnitude of Relative Error (MRE) and Standard Deviation (SDMRE) – see [8, 9, 31, and 32].

1) Quality Criteria

- The Magnitude of Relative Error (MRE) is usually defined by:

\[
MRE = \left| \frac{Actual - Estimated}{Actual} \right| \tag{5}
\]

- The SDMRE is usually defined by:

\[
SDMRE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( Actual_i - Estimated_i \right)^2} \tag{6}
\]

2) Comparing the results of the two EPCU-defined contexts.

The results obtained in the experiment are presented in Table 3. The first column gives the practitioner’s ID, the second column gives the real value for the CRS size (the reference), columns 3 and 4 show the approximate size using the Equal Size Bands approach and its MRE, columns 5 and 6 present the approximate size using the EPCU model with Context A and its MRE, and columns 7 and 8 present the approximate size for Context B and its MRE. The last column on the right shows the difference between the Equal Size Bands approach MRE and EPCU model (Context B).

Table 3 indicates that the Context A size ($MRE = 148\%$) is less accurate than the Context B size ($MRE = 45\%$). It also indicates that most of the values in the Vogelezang experiment [14] for the last category are near the average value. The 28.9 CFP size is, indeed, an outlier, and it is therefore not possible to consider it as a representative value for the output variable range.

The context used to compare the results against the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).

3) Comparing results using the Equal Size Bands approach against the real value.

As shown in Table 3, the MMRE for the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).

Table 3 indicates that the Context A size ($MRE = 148\%$) is less accurate than the Context B size ($MRE = 45\%$). It also indicates that most of the values in the Vogelezang experiment [14] for the last category are near the average value. The 28.9 CFP size is, indeed, an outlier, and it is therefore not possible to consider it as a representative value for the output variable range.

The context used to compare the results against the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).

As shown in Table 3, the MMRE for the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).

Table 3 indicates that the Context A size ($MRE = 148\%$) is less accurate than the Context B size ($MRE = 45\%$). It also indicates that most of the values in the Vogelezang experiment [14] for the last category are near the average value. The 28.9 CFP size is, indeed, an outlier, and it is therefore not possible to consider it as a representative value for the output variable range.

The context used to compare the results against the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).

As shown in Table 3, the MMRE for the Equal Size Bands approach is Context B in Table 3 (the columns related to Context A and its MRE are not considered).
The maximum MMRE is 24% (Practitioner 5) and the minimum is 1% (Practitioner 1) – see Figure 7.

The approximate size values are very good; however, the values for the bands must be defined locally [6], and so it is possible that the values do not correspond to the bands identified by Vogelezang [14]: the average values are strongly related to a specific dataset, as the literature indicates that gathering historical data is a problem in itself [2].

Table 3. Experiment Results

<table>
<thead>
<tr>
<th>Practitioners</th>
<th>MMRE</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57%</td>
<td>1%</td>
</tr>
<tr>
<td>2</td>
<td>37%</td>
<td>56%</td>
</tr>
<tr>
<td>3</td>
<td>30%</td>
<td>46%</td>
</tr>
<tr>
<td>4</td>
<td>31%</td>
<td>45%</td>
</tr>
<tr>
<td>5</td>
<td>39%</td>
<td>57%</td>
</tr>
<tr>
<td>6</td>
<td>36%</td>
<td>31%</td>
</tr>
<tr>
<td>7</td>
<td>31%</td>
<td>46%</td>
</tr>
<tr>
<td>8</td>
<td>34%</td>
<td>49%</td>
</tr>
<tr>
<td>9</td>
<td>37%</td>
<td>52%</td>
</tr>
</tbody>
</table>

The maximum MMRE value with the EPCU model is 77% (Practitioner 8) and the minimum value is 1% (Practitioner 5) – see Figure 8.

The original conceptual basis of the cone of uncertainty was developed by Barry Boehm, who referred to the concept as the Funnel Curve [33]: this means that, in the early phases, the variability in the estimates is higher than in the later phases: the variation proposed by Boehm [33] in the early phases is [-25%, 400%] in this cone of uncertainty.

Based on this assumption, the results are not as good as those of the Equal Size Bands approach using the Vogelezang [14] bands; however, they are within a good range for the early phases, especially considering that not all the practitioners are familiar with the COSMIC method.

They use only the experience-based approach to assign a value for the input variables. This suggests that the use of COSMIC approximate sizing using the EPCU model is less expensive than the sizing method proposed by Vogelezang [14], since it does not require the collection of historical data.

An important feature of the FL approach is that the context does not have to be calibrated: it does not use bands, but rather a continuous range in \( \mathbb{R} \), which is represented by a membership function.

5) Comparing EPCU context results against the Equal Size Bands approach.

From the data in Table 3, it can be seen that the use of Context B (columns 7 and 8) leads to an SDMRE with the EPCU model of 23% and an MMRE of 47%. This MMRE is higher than that of the Equal Size Bands approach value of 36%, and the SDMRE value is also higher, at 14% – see Figure 9.

The maximum MMRE difference is for Practitioner 8 (at 65%), and the minimum difference is for practitioner 6 (at 8%).

VII. CONCLUSIONS

The C-Registration System is a reference system consisting of a new, ‘whole’ set of (stable) requirements that were previously well known for approximating functional size. The scope of this set of requirements is fully defined, which is not very common in reality [20]. However, as the performance of
estimation models is usually evaluated with finished projects, the functional size for the CRS is 107 CFP.

An experiment with 9 practitioners was developed. The practitioners were not familiar with the COSMIC method, they had no historical data for approximating the FSM using COSMIC, nor did they participate in the definition of the EPCU context or know the EPCU model. The only information they had was the requirements from the CRS case study. The experiment constituted a simulation of the early estimation step.

In the experiment, the practitioners used an EPCU approximation approach, within an EPCU context previously defined by the researchers. The output variable was defined using a continuous range of possible values with a “natural” upper boundary, or cutoff, at 16.4 CFP.

Because the Vogelezang [14] dataset was not accessible, this cutoff was derived from Table 3. Furthermore, the researchers took into account that most of the sizes in the Very Large category were around the average (16.4 CFP), and that the size of 28.9 CFP could be an outlier.

Since the output variable was defined in a continuous range, it did not need to be calibrated. At most, the upper limit would need to be updated, depending on the application type. For instance, Vogelezang [14] defined a different upper boundary for mobile applications, of 38.8 CFP.

Other benefits of this definition of the output variable is that there is no need for historical data to define size bands.

The approximation using the EPCU approximation approach shows that the MMRE (Mean Magnitude of Relative Error) = 47%, with a STD DEV = 23%. Compared with the results gathered using the Equal Size Bands approach (MMRE = 11%, STD DEV = 9%), this MMRE is higher than 36%, and its SDMRE is higher too, at 14%.

It is possible to see in Figure 9 that the Equal Size Bands approach and the EPCU Approximation approach behave in a similar way.

As Vogelezang points out [14], “a rapid size measurement will be acceptable if it can be produced faster and still can deliver a reliable approximation of the detailed size measurement.”

Considering the experimentation developed, we can conclude that the use of the EPCU approximation approach is useful when there are no historical data available, in addition to being less expensive than the approach proposed by Vogelezang [14], which does require historical data.

With this approach, an organization can collect historical data about the quality criteria values in the early phases of software development without the need to perform detailed measurements. Then, when the set of software specifications is completed and verified, it can be validated with detailed measurements.

The quality criteria values need to be identified through extensive experimentation and a reliability report must be generated for those using this approach, in order to approximate the FSM. Further studies are needed, particularly with real projects, before the whole set of requirements is known and when the functional process has not been defined by a COSMIC measurer, as was the case with the C-Registration System experiment.

REFERENCES


## Appendix A Information Acquired in the Survey

<table>
<thead>
<tr>
<th>Practitioner 1</th>
<th>Practitioner 2</th>
<th>Practitioner 3</th>
<th>Practitioner 4</th>
<th>Practitioner 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action type 1</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 2</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 3</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 4</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 5</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 6</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 7</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 8</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 9</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 10</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 11</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 12</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 13</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 14</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 15</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 16</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 17</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 18</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 19</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 20</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 21</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 22</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 23</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 24</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 25</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 26</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 27</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
<tr>
<td>Action type 28</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
<td>SMALL</td>
</tr>
<tr>
<td>Action type 29</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Action type 30</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
<td>LARGE</td>
</tr>
</tbody>
</table>