Validation of Supplier Estimates Using COSMIC Method

Francisco Valdés-Souto

National Autonomous University of Mexico
Science Faculty
CDMX, Mexico City, Mexico
fvaldes@ciencias.unam.mx

ORCID: 0000-0001-6736-0666

Abstract. In the software development industry, it is well known that software development organizations (suppliers) need a better and formal estimation approaches in order to increase the success rate of software projects developed. Considering a systematic view, any project requested by a customer needs to be validated in the estimation provided by the supplier, regardless of how formal or not the estimation method utilized was.

However, very often the customers do not know the information used by the suppliers to make their estimations. The software decision makers must face a validation estimates problem where the more useful solution is used the expert judgment, with several problems related to it.

In this paper, a real case study is described where a validation estimates model was generated from a reference database based in COSMIC method. The defined model using a density function helps to the customer to define validation criteria considering the probability that the supplier estimate will be met according to an industry reference database.

Keywords. COSMIC ISO 19761, FSM, Software Estimation, Validation Estimates.

1 Introduction

In the competitive software development industry, it is well known that software development organizations need a better and formal estimation approaches in order to increase the success rate of software projects.

In the literature several techniques and models have been proposed to improve the estimation capabilities in software projects over decades. However, most of the approaches and techniques have been proposed considering a development organization point of view. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]

For decades considerable research has been directed at building and evaluating software effort estimation methods and tools for the organization to develop software projects. The estimation accuracy is a major factor for accepting or declining the estimation methods. [11], [12] Considering a systematic view, any software development organization has a customer which requests the software to be developed and there exists an economic or work interrelation between a customer and supplier.

The customer-supplier interrelation creates economical transactions where the asymmetry of information between customers and suppliers is implicit as described by Valdés et al. [13] As a consequence there is also a risk that needs to mitigate. Two generic pricing modes exists currently in the industry aiming to do that– with several variations – these pricing modes were described by Abran [12]: time and materials billing mode or fixed price contract.

The customer-supplier interrelation is a process where the customer request for a software project to the supplier (a development organization). The supplier estimates the effort or cost considering the inputs available in early stages related to product, process, and constraints [12], [14]. The supplier considers the technical development perspective (internal), producing a project budget and contingency fund which and is communicated to the customer. The customer validates (accepts) or rejects the budget [12], according to the pricing models described. The budget could be a fix price or several work hours with a unit price.

When the customer needs to validate the estimation provided by the supplier, very often they do not know the information used by the suppliers to make their estimations. In the actual competitive business context, software projects are often time-to-market driven, then the decision makers face the validation estimates problem. A more useful solution is used “the expert judgment” but it is not formal and presenting several problems well known.

In mature disciplines it is possible to observe international consensus on measurement, as evidenced by established measurement methods and their respective etalons. In the software domain, expert judgment (‘experience-based’) is the estimation approach typically employed in industry [15] and exists only for functional size measurement international standards. To date ISO/IEC has recognized five (5) Functional Size standardized Measurement Methods (FSMM), where COSMIC is the only FSMM for the second (2nd) generation.

Since the functional size is the only standardized measure in software, it could be considered a fundamental element to all economic actors in software development industry to perform their transactions. It is a required element for all the software development roles to perform their functions. The functional size could be considered as a basic “understanding” element used by the suppliers to make their estimations and for the customers to validate the estimations. Even when they do not know other elements or data used for the supplier to make their estimations, all the parts could know and consider a comparison standardized element that represent the quantity of functionality to be developed (software functional size).

This paper describes a case study about using the COSMIC method to validate the estimations from the customer perspective through a validation estimates model created based in a reference database.

The remainder of this paper is organized as follows. Section II describes overview information related to estimation models. Section III presents information about the construction of validation estimates model based on the use of the COSMIC method.

Section IV describes the application of the validation estimates model defined in a real case between a customer (government entity) and the supplier (software development organization) in the Mexican industry. Section V sets the conclusions

2 Software estimation

2.1 Estimation and databases

It is generally recognized that requirements define the project size (scope), which impact the effort needed to develop it hence drives the project duration [14].

In the past decades, many estimation models and tools have been developed [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]: most of these models focus on estimating effort.

Trendowicz et al. [10], describe some of the main objectives regarding software effort estimation as: project planning & tracking, process improvement, negotiating project costs, etc.

In the literature, several approaches of estimation techniques/methods/models classification were found.

The first approach [4], [5], [16], establishes a classification with three categories: Expert judgment, Algorithmic models, and Analogy.

The second approach [4], [5], establish a classification into two major categories: Algorithmic models, and Non-algorithmic models.

It is interesting to note that in the previous classifications, except for that of expert judgment, all the types described are utilizing mathematical algorithms. This has been considered recently by Abran [12], who defines a common view of the estimation process, identifying only two types of Estimation Models: Expert Judgment and Mathematical Models. In these classifications, any models that are derived from statistical, numeric analysis or more general mathematical algorithms are included.

Any estimation model possesses a strong relationship with the measurement process of the input variables employed to generate the estimate. When the measurement of input variables for an estimation model is reliable, greater confidence is generated in the use of the estimation model [12], [17].

As mention below, the only software feature that could be measure using international standards, is the software functional size, however, is well known that the functional size does not represent all the effort or cost involved to the software development. There are other elements such as Non-Functional Requirements (NFR).

The correlation between functional size and effort/cost has been proved and could explain most of the effort or cost required to develop software. This is in a range from > 75% of the effort/cost and in some cases explaining up to 99% [12]. For the additional effort or cost that does not explain the functional size, there are several possible approaches aims to measure NFR, however currently there is no standard. There is a problem explaining the additional effort or cost with no formal measures.

Any organization that aims to have or develop estimation approaches needs historical data. Very often the organizations do not have this information [18].

Most of the estimation models developed are dependent on the representativeness of the samples (databases) used.

In order to generate the estimation models, researchers and practitioners have used databases documented based on past completed projects they participated in. This information is usually not available to everyone, is difficult to acquire or it has elements that do not make sense for all the database's users. [17]

As pointed out by Morgenshtern, "Algorithmic models need historic data, and many organizations do not have this information. Additionally, collecting such data may be both expensive and time-consuming". [19]

There is an international database managed by a non-profit organization, the International Software Benchmarking Standards Group (ISBSG). This organization generates reports considering an international sample of data [20], the ISBSG database established a standardized data collection for benchmarking and estimating.

The ISBSG database is not the only database. There are other databases in organizations within some countries. For example, in Mexico exists the Mexican Association of Software Metrics (AMMS) that generates a local reference database. The data related to the FSMM in the Mexican reference database is only COSMIC because is a national standard.

2.2 Functional size approximation

As pointed out below, collecting data to create a reference database, may be expensive and implies being time-consuming [19]. In general, this is related to collecting information about past projects that not always exists or is complete.

Functional size measurement methods work best when the information of the functional user requirements is fully known [21].

If the information is complete, there is another problem where the requirements were not measure while the project was being developed. The project information exists but additional effort is required to measure the functional size and must be performed in efficient way in order to gather a reference database that could be useful.

According to the COSMIC Guideline for Early or Rapid COSMIC Functional Size Measurement [22], there are three main situations in which a functional size approximation is needed:

1. When a size measurement is needed quickly, and an approximate size measurement is acceptable if it can be measured much faster than with a standard measurement method. This is known as "rapid sizing."

2. In the early stages of a project, before the requirements have been specified in sufficient detail, for an accurate measurement of size. This is called "early sizing."

3. In general, when the quality of the documentation of the requirements is not good enough for an accurate measurement of the size.

The COSMIC Guideline for Early or Rapid COSMIC Functional Size Measurement [34] integrates several techniques for the approximate sizing of new, 'whole' sets of requirements. The majority of the techniques presented in [22] are based on the existence of historical data to determine the scaling factor (average, or size bands) or another calibration. The techniques based on historical data are of little use for organizations without such data.

Aiming to tackle this situation, the EPCU approximation technique was proposed by Valdés et al. [23], [24]. This approach based in fuzzy logic, does not require local calibration and is useful when there are no historical data available. In addition, it is less expensive than those that requires calibration.

Research on the EPCU size approximation technique has focused on two granularity levels of the Functional User Requirements (FUR) description: Functional Process [23] and Use Case [24], with different EPCU context definitions, especially regarding changing the domain of its output variable function.

2.3 Estimation quality criteria

For decades, estimation accuracy was a major factor for accepting or declining a certain effort estimation method. The more frequent way to compare the accuracy of estimation models in the literature and define the confidence about them, can be called quality criteria [12]: Magnitude of Relative Error (MRE), Standard Deviation (SDMRE), Prediction level (PRED) and Mean Absolute Residual (MAR) – see [16], [3], [4], [25], [11].

Because MMRE has been shown to be a biased estimator of central tendency of the residuals of a prediction system and it is an asymmetric measure [26], [27], [28], [29], the Mean Absolute Residual (MAR) [11] arose aiming to provide more acceptable measure of accuracy.

To apply all these quality criteria, the original dataset used to generate the estimation needs to be known. To evaluate the accuracy or confidence for a specific estimation, the database with which the estimate was made needs to be available in order to validate the confidence of the estimate.

Usually the suppliers, a development organization creates their estimation models correlating the functional size against the effort or cost for a project considering their past projects. The managers calculate the budget according to the profits and the risk identified. It is not common that all this information will be shared with the customer.

Usually, the final proposition shared with the customer has a budget, not the data or the steps to define the budget.

The customer could at most, measure the functional size for the requested project. With this data, it is not possible for the customers to calculate the quality criteria, resulting in most of the validations being made using expert judgment.

2.4 Estimation model vs. validation model

There is a need for estimation models from the perspective of the supplier for several objectives as was mentioned in [10]. Most of the literature is related to this approach and is based in the use of historical data (database). This allows generation of productivity models [30] that correlate functional size with the resources data of the projects (effort and cost). It also represents a particular way of work for the company that record the projects in the database. The productivity models were used to generate an estimation model where the confidence of the estimation could be certainly established. [12]. With the final estimation the supplier defines the budget (effort or cost) to be proposed to the customer.

The evaluation of the estimation model's accuracy and confidence is important because the suppliers are looking to provide information for the decision makers to define the final budget (effort or cost) hence the expected profits.

Once the supplier delivers to the customer the budget (effort or cost) there is a need for a validation estimates models from the customer perspective. The customer does not know the supplier database, and if they did it does not represent anything useful because the productivity model, or the estimation model, used in the representation of the effort/cost to develop the software. It is not the budget where the software is quoted to the customer.

If the customer collects the set of projects they buy from distinct suppliers and generates their own database, that correlate the functional size against the budget (effort or cost) offered by the suppliers (not the real effort or cost to develop the software), the customer could then identify the accuracy and confidence. They could then provide information for the decision makers in order to evaluate if the budget could be accepted. This is a validation estimate model.

The validation estimate will enable the customer to validate if the budget provided by the supplier is valid considering the constraints established by the customer aligned to their goals.

The difference between estimation model and validation estimate model is as follows. The estimation model will predict a possible value, or range of values, of the cost/effort to construct the software. It considers the inputs and the historical projects from the software development organization to generate a budget. In other words, the estimation model could be seen as a regression model as defined in [12] and is referenced to a local historical database that very often is not public.

The validation estimate model must to be used to validate if the estimation provided by the supplier – budget- accomplish the specific constraints defined by the customer. The validation estimate model is also an estimation model developed using the customer perspective information.

Having the customer, a reference database from their perspective, a probability distribution model could be developed. The customer could evaluate, with the data provided by the supplier, the probability that the estimate provided will be met and if the project accomplishes the criteria defined over the validation estimate model.

Ideally, validation models should be built from an accessible reference database aiming to reduce the asymmetry of information as is defined in [13] "Organisms that would generate and transmit this information to potential buyers are clearly needed in an industry".

Both models, the estimation and the validation model, work better if there is a link between them ("understanding" element), this function could be developed by the FSMM.

In this case study the use of COSMIC is selected because is the national standard in Mexico, and the reference database from the AMMS was built over this standard. The COSMIC is defined by the AMMS with three qualifiers defined as BTT:

- BASIC because they are internationally generally accepted standards that allow the generation of derived metrics in the future;
- TRANSVERSAL because they serve to all economic actors and software development roles, to perform their functions (development, test, D&A, self-management, etc.) and transactions (buying and selling, bids, using, etc.);
- TRANSCENDENT because being basic is intended to allow comparison over time (forward and backward) and through different practices, technologies, which are changing.

COSMIC facilitates the way of work for the two models enabling a comparison between them, considering the software functional size identified by the supplier or customer (Fig. 3). The functional size could be verified as described by Hassam et al. [31] using a three-phase protocol to make the verification of the COSMIC measurements.

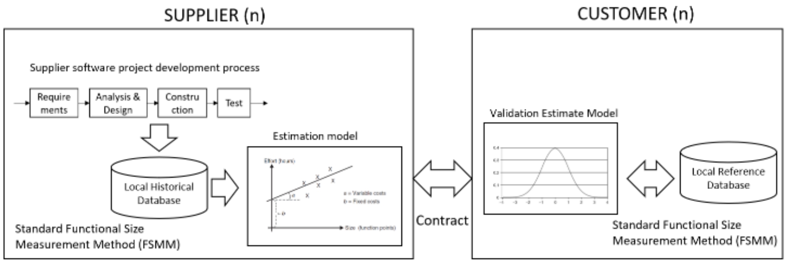

Fig. 3. Estimation model and validation estimate model

Even if the FSMM used is distinct, the validation estimate works in a similar way.

3 Defining a validation estimate model

3.1 Gathering information for a reference database

In Mexico at the end of 2015, a call for project was issued by the Mexican Software Metrics Association (AMMS) through an online platform, with the purpose of gathering information for the realization of the Baseline Study of Productivity and Cost of the Mexican Software Development Industry (IMDS, Industria Mexicana de Desarrollo de Software). The purpose was to obtain information related to software projects carried out in Mexico (already concluded) from the customer perspective. This enables defining the baseline of productivity and cost of the IMDS, aiming to improve the knowledge of the IMDS from different points of view, such as the technical aspect and the economic aspect.

The first guest participants were, private companies and government entities that would like to share and contribute to determining a database containing relevant information to the IMDS, which would serve as a reference to economic agents. To complete the survey related to obtaining information, it was necessary that the software projects had concluded.

The survey was divided into three sections that complement the information necessary to carry out the study:

- The first section relates to the identification of the project for an adequate classification of the information, the classification data were like used by ISBSG, i.e. technical issues (life cycle, architecture, organization type, etc.), organization issues, etc.
- The second section consists of answering a questionnaire that allows the application of EPCU functional size approximation technique [23,24], [22], to obtain an approximate size in the international standard ISO 19761.
- The third section was optional and was for those companies/entities that have used some measurement standard of functional size. No answer received with this section for any project submitted.

The data collected present software projects already completed and carried out in Mexico. In total, detailed information was gathered from 398 software projects, 96 (24%) from the government sector and 302 (76%) from the private sector. For all the projects, was possible to obtain the approximate functional size using the EPCU approximation technique, and with the effort, a Product Delivery Rate could be obtained [12].

All the detailed data and analysis including frequency analysis of the categories considered in the study, distribution analysis of the data set by category and descriptive analysis for each category are presented in the Baseline Study of Productivity and Cost of the Mexican Software Development Industry [32].

3.2 Finding a probability distribution for the productivity phenomena in the reference database

With the data collected in the AMMS study, it was proposed to find an adjustment of some probability distribution (density function) to the phenomena of productivity that is the inverse of PDR (WH/CFP).

When the information was analyzed in intervals, the number of elements and the frequency of ranges are determined within the entire range of values generated the frequency histogram (Fig. 4). If the number of intervals were increased considerably, the concept of probability of occurrence is designed to indicate how often it is that each of these values occurs, and consequently there is a probability associated to each value (or point). Thus, the frequency histogram is transformed into the density function, derived by the intervals. Fig. 4.

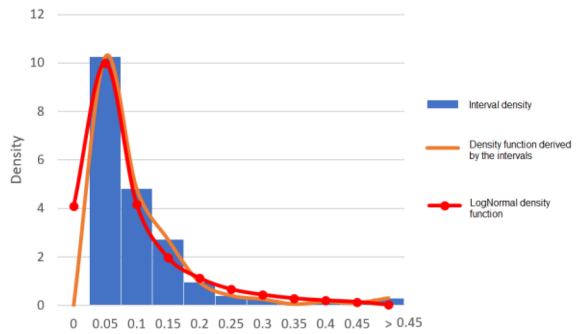

Fig. 4. Lognormal density function by productivity [CFP/Effort], Adapted from [32]

The productivity, already considered as a variable, directly implies that the values obtained from the database collected only represent specific realizations. A sample of all the possible values, does not allow knowing the total behavior of the phenomenon. To achieve a complete characterization of the variable, that allows an empirical inspection of it by means of a sample, it is necessary to introduce a random factor that is precisely in the density function. In the theory of probability there are several known density functions that can be adjusted to specific phenomena without major problems by these densities.

As can be seen in Fig. 4, for the case of productivity, it is relevant to consider the extremes of its range. At the far right it is observed that the value of the density is low, although it does not definitively fall to zero. This behavior is known as heavy tails and is fundamental when proposing a distribution. Sergey et al. [33] addresses several of the discussions surrounding this topic and proposes a solution that combines the adjustment of a Lognormal distribution and the adjustment of a Pareto-type distribution.

For the study carried out by the AMMS, the Lognormal distribution was chosen because it will allow modeling in a simple way the phenomenon of interest: productivity and in consequence the PDR, which we know is the inverse of productivity.

From Fig. 4, it is possible to see that Lognormal distribution fits better to the data.

By associating this density function with some phenomenon, for example, the productivity or the PDR from the IMDS study. It is possible to identify the average behavior (μ) that this industry presents. Additionally, it will be possible to determine where 68% of the productivities fall ($\mu \pm \sigma$), or 95% of them ($\mu \pm 2\sigma$), etc. In most cases it will be possible to identify which probability corresponds to a related point of productivity or PDR with a specific project. Also identify and define ranges of probabilities according to the criteria of each organization, that are named validation criteria.

The Lognormal distribution keeps a close theoretical relationship with the most important of all probability densities: the normal distribution. This is fundamental because, based on a normal model, which is sometimes better known, we can establish the validation criteria for the estimations from a customer perspective.

3.3 Defining the validation estimates model and validation criteria

For this case study, we will describe the validation of estimates of a software development organization (supplier) from a government entity (customer).

In Table 1, the data distribution includes only projects carried out in government entities from the reference database gathered from the Baseline Study of Productivity and Cost of the Mexican Software Development Industry [32]. The sigma intervals for probability distributions of PDR values are shown. It is possible to observe the sigma intervals for probability distributions -very often used in normal distribution-, however, considering a Lognormal distribution, the upper limits grow at a fast step. Table 1 and Fig. 5.

From Table 1, it is possible to observe that only a few projects have a high productivity (low PDR). See the left side of the Fig. 5, with very low probability (column 4) for the evaluated project to be in that side. Similarly, few projects have very low productivity (high PDR) as seen on the right side of Fig. 5, with a high probability (column 4) for the projects to be there. This implies a more expensive cost, because is related to more effort.

Considering this and with the advice of the researcher, the customer identifies the criteria that the estimations must accomplish.

After the explanation of the density function, the customer decides to expect a range between more or equal to 64.4% (Expected Value) and less or equal to 84.1% ($\sigma 1$) of probability that the supplier estimate is met.

In other words, lower limit value accepted by the customer as PDR is the Expected Value = 13.60 [WH/CFP] but the upper limit value accepted by the customer as PDR is $\sigma 1$ = 21.68 [WH/CFP]. These two elements define the validation criteria for the estimates. See the dotted line in Fig. 5. The definition of the validation criteria using the PDR and the probability for the supplier estimate is more than 64%, while there are studies where the success rates of the projects are lower than 40%. [34]

4 Validating estimations in a formal way in government entity

4.1 Context

A government entity in Mexico has a contract with a software development organization to develop nine (9) projects in 2018. For confidentiality the names of the entity and the supplier company are not shown, they were referred as customer and supplier respectively. The amount for that contract was $33,928,580.00 MNX equivalent approximately to $1,785,714.74 USD in twelve months.

For the operation of the contract, the customer request projects to be developed for the supplier. The supplier estimates the projects using the technique they use. In this case the supplier also uses Expert Judgment; the supplier must present the effort estimated for each project and the assumptions.

The customer validated the estimates against the model defined in Section III.C, Table 1 and Fig. 5. To develop the validation, the customer gathers the functional size in

COSMIC units (CFP) using the EPCU approximation technique [23,24] for each project and obtains the PDR considering the effort estimated by the supplier.

Table 1. Lognormal model for productivity and PDR by sigma intervals for government projects, adapted from [42]

LogNormal Model element	Productivity [CFP/WH]	PDR [WH/CFP]	Probability %
- σ3	0.01	1.12	0.1%
- σ2	0.02	2.35	2.3%
- σ1	0.04	4.93	15.8%
Geometric Mean	0.97	10.34	50%
Expected Value	0.074	13.60	64.4%
σ1	0.20	21.68	84.1%
σ2	0.42	45.45	97.7%
σ3	0.89	95.28	99.9%

Because the use of validation estimates model was being tested while this contract is executed, no decisions that impact to the projects where taken, the possible decisions were documented and evaluated when the projects were finished.

4.2 Validating estimates

The nine projects developed by the supplier for the government entity are shown in Table 2, the description of the table is as follows:

- Column 1 identifies the projects from 1 to 9,
- Column 2 indicates the number of elements of FUR,
- Column 3 indicates the type of the element at the corresponding granularity level (Use cases (UC) or Functional Process (FP)) to be used to apply the EPCU approximation technique [23,24] .
- Column 4 presents the total effort estimated by the supplier,
- Column 5 the functional size approximated by the customer, who knows the FUR
- Column 6 the PDR is calculated
- Column 7 the result of the validation of the estimates.

If the PDR is in the valid range [13.60 WH/CFP, 21.68 WH/CFP], the estimation is accomplished using the validation criteria. The probability of success is between 64.4% and 84.1%, according to the Lognormal density function with the data of the IMDS.

Column 8 presents the success probability according to the Lognormal distribution.

In Table 2 there are two projects (22.2%) that accomplished the validation criteria (Project 1 and 6) with a PDR = 18.92 WH/CFP and PDR = 18.96 WH/CFP respectively, equivalent a probability of success of 79.3% and 79.4 % respectively. For the seven projects (77.8%) that do not pass the validation, it is possible to observe three cases:

The first is a set of five projects presenting less than 50% of probability of success, been the lowest probability in this set a 27% (Project 9 with PDR = 6.56 WH/CFP) and the highest probability a 45.5% (Project 3 with PDR = 9.51 WH/CFP).

Table 2. Projects developed by the supplier and the estimates validation

Project	Number of UC or FP	Type (UC / FP)	Total effort [WH]	Approximated Functional Size (CFP)	PDR [HH/CFP]	Validation estimate range [13.60 , 21.68]	Success Probability [%]
1	35	CU	12,994	686.6	18.92	✔	79.3%
2	12	CU	3,370	126.2	26.71	✘	90.0%
3	21	CU	8,634	907.5	9.51	✘	45.5%
4	97	PF	7,648	910.4	8.40	✘	38.9%
5	26	CU	4,294	580.0	7.40	✘	32.6%
6	26	CU	8,665	456.9	18.96	✔	79.4%
7	19	CU	5,897	476.1	12.39	✘	59.6%
8	73	PF	6,280	807.4	7.78	✘	35.0%
9	134	PF	11,460	1,745.9	6.56	✘	27.0%
TOTAL			69,242	6,697.0			

The second case is Project 7 that presents a PDR = 12.39 WH/CFP with a probability of success of 59.6%. This probability is higher than the geometric mean (50%) but less than the minimum value of the validation criteria defined by the customer.

The third case is Project 2, which presents the highest probability of success with 90% with a higher PDR = 26.71 WH/CFP. This PDR is 23.2% higher than the maximum value accepted (21.68 WH/CFP).

Fig. 5, depicts the position of the nine projects over the Lognormal distribution and the validation criteria identified by the dotted line. This is a graphical way to observe the validation estimate operation.

Considering the estimates gather from the supplier and the validation realized by the customer with a reference database, the theoretical decision had to be:

1. Accept two projects (Projects 1 and 6) because they are between the validation criteria.
2. Reject six projects because were underestimated, the problem, in this case, is not the cost because will be lower, the problem arises because the low probability of the estimations will be met. The entity could not receive the value from the project as was planned and there is a risk for the project to be more expensive in order to finishing it.
3. Reject one project (Project 2) because was overestimated, this project could be 23.2% more expensive than the limit, of course, the probability success was very high (90%).

For the rejected projects a new estimation should be requested to the supplier several times, including an advisory looking to avoid misunderstanding about the project

FUR's. It is well known that estimations were made in early stages and the high uncertainty about the project with unformal estimation techniques generates poor estimations.

The real results for the projects recommended to be accepted (point 1) were that the project finished according to the estimated effort.

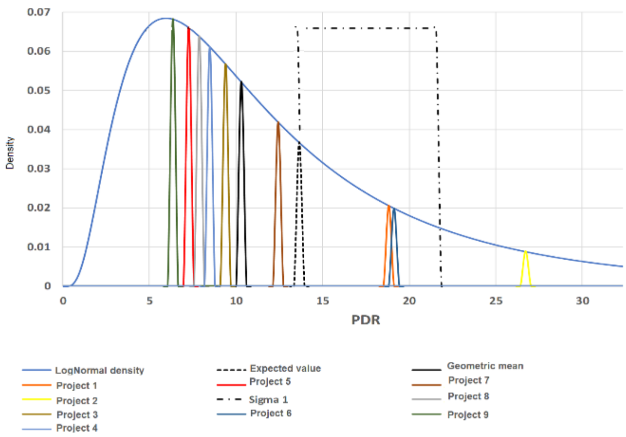

Fig. 5. Validation of the nine projects graphically over Lognormal distribution

For the project overestimated, the supplier reports the consumption of all the effort, there is no way to check if the supplier uses less effort.

The six projects underestimated (point 2) present problems because the extra effort was required, for 4 of them (with the high underestimation) more effort was obtained through several changes request, for two projects the cost was assumed by the supplier.

4.3 Threats to validity

There are several validity threads to discuss for the case study implementation.

The use of COSMIC as FSMM was established because in Mexico COSMIC is the National Standard: NMX-I-19761-NYCE-2017 before NMX-I-119-NYCE-2006 for software functional measurement, in consequence the reference database generated by AMMS use the COSMIC method.

Intuitively, the proposed approach in this paper could be generalized for other FSMM like NESMA or IFPUG but must be proven.

It could be assumed that if the customer and the supplier use the same FSMM as a "understanding" element, the application of the approach could be simplified, but must to be proven.

For this case study, the use of EPCU approximation approach was used because the size measurement is needed quickly (rapid sizing). The granularity level in the FUR's for the 9 projects, the two different EPCU context definitions (Functional Process [23] and Use Case [24]) were used. At the end a functional size in COSMIC units is obtained and compare.

Because the use of validation estimates model in this case study was only for testing and no decisions were taken, the results need to be validated against real context. Hence taking the decisions and comparing with the real results of the projects, after the application of the validation criteria defined.

5 CONCLUSIONS

In this paper a validation estimates model was proposed in order to allow the customers of software development contracts to validate the supplier estimation and improve the decision making about the feasibility of the project using a reference database based in the functional size measurement method COSMIC.

The validation estimates model was derived from a Mexican reference database, that was created in the realization of the Baseline Study of Productivity and Cost of the Mexican Software Development Industry. The purpose was to obtain information related to software projects carried out in Mexico (already concluded) from the customer perspective, aiming to define an industrial reference.

The validation estimate model was based in a Lognormal density function that fits better to the data gathering in the Baseline Study. The validation criteria was defined by the customer expecting a range between more or equal to 64.4% and less or equal to 84.1% of probability that the supplier estimate is met. This probability is higher than several studies in the industry about software projects success [34]. Another relevant issue is that both under-estimation as well as over-estimation are considered as disqualifiers.

With this formal way to validate the supplier estimation, a customer could improve the decision making about the feasibility of the project because the model was defined by a reference database and the probability that the supplier estimate is known, in consequence the probability of the project success is increase.

Using the validation estimates model defined with their validation criteria, nine real projects from a Mexican government entity were validated in their estimates generated by the supplier.

The possible recommendations derived by the model and its validation criteria could be useful for the customer to avoid some problems in the project's execution for the underestimated projects and could help to save resources and control the supplier for the overestimated project.

References

[1] A.J. Albrecht, J.E. Gaffney, Software Functions, Source Lines of Codes and Development Effort Prediction: A Software Science Validation, IEEE Trans. Softw. Eng. 9 (1983) 639–648.

[2] T.K. Abdel-Hamid, S.E. Madnick, Impact of Schedule Estimation on Software Project Behavior., IEEE Softw. 3 (1986) 70–75. doi:10.1109/MS.1986.234072.

[3] A. Idri, A. Abran, Towards A Fuzzy Logic Based Measures for Software Projects Similarity, in: MCSEAI'2000 - Maghrebian Conf. Comput. Sci., Fez, Morocco, 2000: pp. 1–12.

[4] A. Idri, A. Abran, Fuzzy Analogy : A New Approach for Software Cost Estimation, in: 11th Int.

Work. Softw. Meas., Montréal, Canadá, 2001: pp. 93–101.

[5] A. Idri, A. Abran, T.M. Khoshgoftaar, Estimating Software Project Effort by Analogy Based on Linguistic Values, in: Proc. - Int. Softw. Metrics Symp., IEEE Computer Society, Ottawa, Canada, 2002: pp. 21–30. doi:10.1109/METRIC.2002.1011322.

[6] R. Jeffery, M. Ruhe, I. Wieczorek, Using Public Domain Metrics to Estimate Software Development Effort, in: Seventh Int. Softw. Metrics Symp. METRICS 2001, IEEE Computer Society Technical Council on Software Engineering, London, England, 2001: pp. 16–27. doi:10.1109/METRIC.2001.915512.

[7] Barry .W. Boehm, Software Engineering Economics, 1st ed., Prentice Hall, Englewood Cliffs, NJ, USA, 1981.

[8] B.S. Barry W. Boehm , Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, Software Cost Estimation with COCOMO II, 1st ed., Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[9] F. Valdés-Souto, A. Abran, Industry Case Studies of Estimation Models Using Fuzzy Sets, in: Reiner Dumke (Ed.), Softw. Process Prod. Meas. Int. Conf. IWSM-Mensura 2007, UIB-Universitat de les Illes Baleares, Illes Baleares, Spain, 2007: pp. 87–101.

[10] R.J. Adam Trendowicz, Jürgen Münch, State of the Practice in Software Effort Estimation: A Survey and Literature Review, in: J.Z. Zbigniew Huzar, Radek Koci, Bertrand Meyer, Bartosz Walter (Ed.), Lect. Notes Comput. Sci. Softw. Eng. Tech., Springer-Verlag, Brno, Czech Republic, 2008: pp. 232–245. doi:10.1007/978-3-642-22386-0.

[11] M. Shepperd, S. MacDonell, Evaluating prediction systems in software project estimation, Inf. Softw. Technol. 54 (2012) 820–827. doi:10.1016/j.infsof.2011.12.008.

[12] A. Abran, Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers, 1st ed., John Wiley & Sons, Hoboken, NJ, USA, 2015.

[13] F. Valdés-Souto, A. Abran, Using the ISO 19761 COSMIC Measurement Standard to Reduce “Information Asymmetry” in Software Development Contracts and Enable Greater Competitiveness, Technol. Invest. 04 (2013) 261–268. doi:10.4236/ti.2013.44031.

[14] P. Bourque, S. Oligny, A. Abran, B. Fournier, Developing project duration models in software engineering, J. Comput. Sci. Technol. 22 (2007) 348–357. doi:10.1007/s11390-007-9051-5.

[15] J. Gómez, El Laboratorio de las TI, Result. Del Estud. Sobre Medición Del Tamaño Del Softw. 2014. (2015) 1. http://www.laboratorioti.com/2015/02/11/resultados-del-estudio-sobre-medicion-del-tamano-del-software-2014/ (accessed February 10, 2016).

[16] M. Shepperd, C. Schofield, Estimating Software Project Effort Using Analogies, Softw. Eng. IEEE Trans. Softw. Eng. 23 (1997) 736–743. doi:10.1109/32.637387.

[17] F. Valdés-Souto, Creating an Estimation Model from Functional Size Approximation Using the EPCU Approximation Approach for COSMIC (ISO 19761), in: C. Mario, Z. Jaramillo, C. Elena, D. Vanegas, W.P. Charry (Eds.), Softw. Eng. Methods, Model. Teaching, Vol. 4, Editorial, Bpgotá, Colombia, 2017: p. 468.

[18] F. Valdés-souto, Creating a Historical Database for Estimation Using the EPCU Approximation Approach for COSMIC (ISO 19761), 4th Ed. Int. Conf. Softw. Eng. Res. Innov. (2016). doi:10.1109/CONISOFT.2016.32.

[19] O. Morgenshtern, T. Raz, D. Dvir, Factors Affecting Duration and Effort Estimation Errors in Software Development Projects, Inf. Softw. Technol. 49 (2007) 827–837. doi:10.1016/j.infsof.2006.09.006.

[20] C. Symons, The Performance of Real-Time, Business Application and Component Software

Projects, 2011.

[21] L. Santillo, Early and Quick COSMIC FFP Overview, in: A.A. Reiner Dumke (Ed.), Cosm. Funct. Points Theory Adv. Pract., CRC Press, Boca Raton, FL, USA, 2011: pp. 176–191.

[22] Common Software Measurement International Consortium (COSMIC)., Guideline for Early or Rapid COSMIC Functional Size Measurement, 2015.

[23] F. Valdés-Souto, Analyzing the performance of two COSMIC approximation sizing techniques at the functional process level, Sci. Comput. Program. 135 (2017) 105–121. doi:10.1016/j.scico.2016.11.005.

[24] F. Valdés-souto, Analyzing the Performance of Two COSMIC Sizing Approximation Techniques Using FUR at the Use Case Level, in: M. Salmanoglu, A. Coşkunçay (Eds.), Jt. Conf. Int. Work. Softw. Meas. Int. Conf. Softw. Process Prod. Meas., Beijing, China, 2018: pp. 70–108.

[25] I. Myrtveit, E. Stensrud, A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models, IEEE Trans. Softw. Eng. 25 (1999) 510–525. doi:10.1109/32.799947.

[26] L. Lavazza, Accuracy Evaluation of Model-based COSMIC Functional Size Estimation, in: ICSEA 2017 Twelfth Int. Conf. Softw. Eng. Adv., 2017: pp. 67–72.

[27] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, M.J. Shepperd, What accuracy statistics really measure, in: IEE Proc. - Softw., 2001: pp. 81–85. doi:10.1049/ip-rsn:20010506.

[28] T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, A simulation study of the model evaluation criterion MMRE, Softw. Eng. IEEE Trans. 29 (2003) 985–995.

[29] I. (Norwegian S. of M. Myrtveit, E. (Buskerud U.C. Stensrud, M. (Bournemouth U. Shepperd, Reliability and validity in comparative studies of software prediction models, IEEE Trans. Softw. Eng. 31 (2005) 380–391. doi:10.1109/TSE.2005.58.

[30] T. Zaw, S.Z. Hlaing, M.M. Lwin, K. Ochimizu, Automated Size Measurement of Embedded System based on XML using COSMIC FSM, in: A.C. Murat Salmanoglu (Ed.), Proc. Acad. Pap. IWSM Mensura 2018 "COSMIC Funct. Points - Fundam. Softw. Effort Estim. Held Conjunction with China Softw. Cost Meas. Conf. (CSCM 2018), CEUR Workshop Proceedings (CEUR-WS.org), Beijing, China, 2018: pp. 14–19. doi:ISSN:1613-0073.

[31] H. Soubra, A. Abran, Verifying the Accuracy of Automation Tools for the Measurement of Software with COSMIC – ISO 19761 including an AUTOSAR-based Example and a Case Study, in: Frank Vogelezang; Maya Daneva (Ed.), 2014 Jt. Conf. Int. Work. Softw. Meas. Int. Conf. Softw. Process Process Prod. Meas., Conference Publishing Services (CPS), Rotterdam, the Netherlands, 2014: pp. 23–31. doi:10.1109/IWSM.Mensura.2014.26.

[32] Asociación Mexicana de Métricas de Software, Estudio Línea Base de Productividad y Costo de la Industria Mexicana de Desarrollo de Software, 1a Edición, Asociación Mexicana de Métricas de Software (AMMS), México, CDMX, 2018.

[33] S. Nigai, A Tale of Two Tails: Productivity Distribution and the Gains from Trade, J. Int. Econ. (2016). doi:10.1016/j.jinteco.2016.10.006.

[34] Standish Group International, CHAOS Manifesto 2012 Coll., 2012.