Industry Case Studies of Estimation Models Using Fuzzy Sets

Francisco Valdés Souto [1], Alain Abran [2]

[1] Mexican Central Bank,Tolona 371, Col Roma Sur, 06760 Mexico D.F., fvaldes@banxico.org.mx, fvaldes@nrim.com.mx

[2] École de technologie supérieure – Université du Québec, Alain.Abran@etsmtl.ca

Abstract. Today, the duration factor in a software project is as critical as it is strategic, since even a slight delay can lead to missing a market opportunity or to generating significant losses. This paper presents two industry case studies on the use of an estimation model using, as input parameters, fuzzy sets of data proposed by practitioners estimating various project attributes.

Keywords: Estimation Model, Time Estimation, Fuzzy Sets.

1 Introduction

Information is acquired in a gradual way throughout the development lifecycle and this poses the known, but unavoidable challenge of making decisions on the basis of incomplete and at times unreliable information. For instance, at the feasibility stage, most of the information is available only at a very high level of abstraction, and is often based on a number of assumptions which can be neither verified nor precisely described. Consequently, cost and duration estimates based on such information should not be expected to be accurate, and should be associated with potentially significant ranges of variance. Still, even at this stage of the process, management must rely on such information for decision making purposes. In such a context, the decision to launch a project is often determined by considering in particular the "subjective" importance of the outcome, that is, delivering the product (a system), while at the same time subjectively minimizing that it may not be possible to do so on time because of a lack of certainty on most of the elements identified and assessed in the feasibility analysis. Any improvement in the estimation technique is therefore welcome in order to improve the decision making process and decrease the ranges of variations.

The estimation technique typically used in industry is the one based on the experience of the employees of the organization. This experience-based approach considers informally an unspecified number of non-quantified variables that other estimation models based on statistical analyses cannot take into account. Of course, there are a number of problems with using experience to make estimations, notably the following ones: experience is specific to the expert and not to the organization; estimation expertise is neither well described nor well understood; and this expertise is hard to assess and cannot be replicated systematically.

However, estimation expertise is still valuable to an organization. The challenge is to figure out how to benefit from it and use other estimation techniques as well, whether they be algorithmic, using simple statistical techniques such as regression, or more sophisticated, such as neural networks or estimation models based on fuzzy logic.

Of course, the challenge is to define an estimation model which takes into account characteristics such as:

- the way practitioners make their estimates;
- the (qualitative) variables that practitioners use;
- the uncertainty associated with the project schedule in the early stages of the life cycle using the information available at the time (often vague or ambiguous).

This paper reports on an approach to combine practitioners' informal expertise at estimating together with a more formal estimation method. This approach consisted in developing an estimation model using fuzzy logic to quantify the software project attributes which, in practice, are often measured in terms of linguistic values (such as low, very low, average, high and very high) based on the experience of the employees of an organization.

The representation of a fuzzy logic function is gradual between the boundaries, rather than abrupt and stepwise as in algorithmic models (Fig. 1).

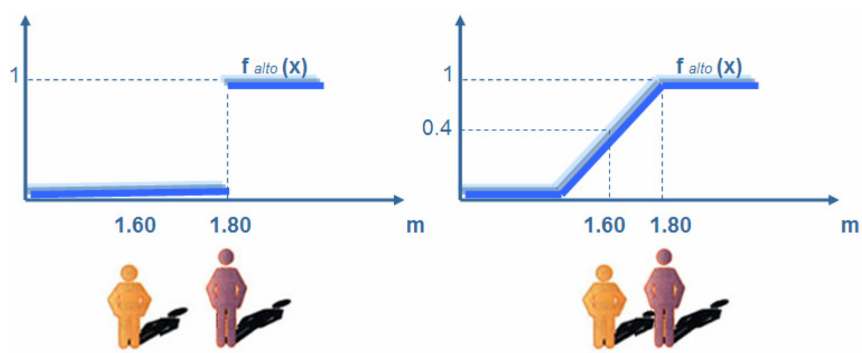


Fig. 1. Stepwise and Fuzzy Functions

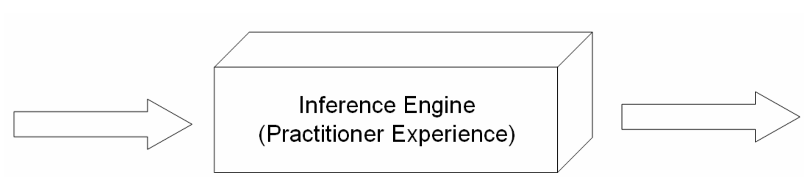
Of the models that use fuzzy logic, a number are based on the case-based reasoning (CBR) or analogy-based estimation approach [4]. The use of fuzzy logic is particularly useful when there is limited domain knowledge and when an optimal solution process to the problem is not known.

The four primary steps comprising a CBR estimation approach are as follows:

- Retrieve the most similar case or cases;
- Reuse the information and knowledge represented by the case(s) to solve the estimation problem;
- Revise the proposed solution;
- Retain the parts of this experience likely to be useful for future problem-solving.

Even when this type of model improves estimate accuracy, there are still two types of uncertainty associated with the estimates derived by these models: the consequence (i.e. ‘have similar costs’) is imprecise and not always deterministic; and it is possible that some applications of CBR will have similar cases, but completely different outcomes [4].

In a software development organization in Mexico, an estimation model based on fuzzy logic was developed to take into account the estimation inputs and outputs of experienced practitioners in the estimation process. The estimation model is referred to in this organization as Estimation of Projects in Contexts of Uncertainty - EPCU. This EPCU model takes as input the way in which experienced practitioners estimate software projects by considering qualitative variables as experienced practitioners use them in making estimates. This is referred to as the process of inference for estimation by experienced practitioners (Fig. 2).



Input Variables: Output Variables

Complexity
Duration
Size
Tool Development Experience
Work Team Motivation
Process Development Experience
Knowledge of the kind of systems to develop
Project leadership skills
Risks
Etc.

Fig. 2. Process of inference for estimation by Practitioner Experience

Our purpose here is not to present the details of this EPCU model, but to report on the actual use of the model through two case studies in two different organizations. The case study is a standard method of empirical study in various 'soft' sciences such as sociology, medicine and psychology. A case study usually looks at a typical project, and can be used to evaluate alternatives.

For each organization where the EPCU model was used, interviews were conducted for case study 1 with the project manager and the rest of the team, and, for case study 2, with the project coordinator only. The interviews were conducted to gather information and data which could not be found in the project documents.

These were face-to-face interviews, and general questions were asked about the projects with a view to identify significant variables and project features such as actual duration, as well as specific questions about the estimated shortest duration for a project in the organization, as well as the estimated longest duration.

This paper is organized as follows: section 2 presents an overview of the fuzzy model used, section 3 presents the two case studies involving the use of this model in industry, and section 4, a discussion.

2 Description of the Fuzzy Model Used

2.1 Overview of the Model

A project is influenced by many parameters at the same time, their impact being distinct for each: some have a major impact in a specific project, while others might be almost irrelevant. The EPCU model includes six steps: identification of input variables, specification of output variables, generation of inference rules, fuzzification, inference rules evaluation and defuzzification.

2.2 Identification of the Input Variables

The goal of this step is to have the experienced practitioners in an organization identify and assess the most significant variables for a project or kind of projects, for instance: software size (expert point of view), software complexity, team skills, knowledge of the software development process or its implementation phase, the leader’s skills, the client or provider organization’s environment, knowledge of the tools to be developed in the project, client commitment, the stakeholders involved, and so on.

It is natural for experts to have differing opinions about some variables. To deal with this diversity, fuzzy logic is used, in a step known as fuzzification (the theory of fuzzy logic is outside the scope of this paper – for further information, see [1-6]), which is described is section 2.5.

In this second step, experienced practitioners must define the fuzzy sets for each variable, which means that they must classify the variables in terms of linguistic values which they can evaluate. For example, for parameter complexity, the fuzzy set could be classified as low, average or high.

Also required is the definition of the membership function domain to represent the opinions of the experienced practitioners about these input parameters.

By the end of this step, the most significant parameters have been generated, together with their fuzzy sets and the ranges available for each of them.

2.3 Specification of the Output Variable

The previous step is repeated for the output parameter (here, the estimate of the project duration). The output has also to be defined in fuzzy sets: usually there are more than three values. Depending on the range defined, the fuzzy sets are needed to describe the relationship between the project duration in the organization and the estimated values provided by the experienced practitioners.

The range of the output variable must begin with the estimated smallest duration for a project in the organization, and extend to its longest duration. If the organization has to develop a larger project than it has ever developed before, the range of the output variable when it is defined must include increments to take into account expectations of projects to be developed in the future. Usually, the output duration variable is defined in weeks or months.

2.4 Generation of Inference Rules

All the fuzzy sets belonging to each input variable must be combined in 'if…, then…' form:

If x and y, then z
If x or y, 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.

2.5 Fuzzification

The goal of this step is to obtain fuzzified values as a consequence of opinions as to those values put forward by an experienced practitioner. This means that the membership function must be defined for the input variables.

If three fuzzy sets are used for the input variable, the membership function can look something like the example in Fig. 3.

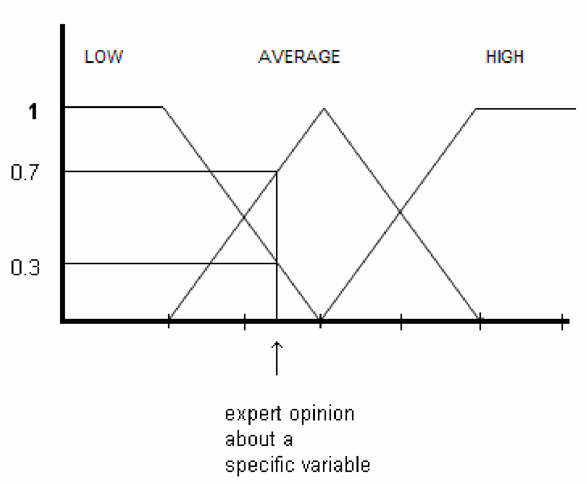


Fig. 3. Example of a fuzzy membership function

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

2.6 Inference Rule Evaluation

The fifth step consists of evaluating the rulebase by substituting the fuzzy values obtained. The Inference Rule evaluation must follow the rules of fuzzy logic, such as:

$$\begin{aligned} \text{Value (P or Q)} &= \max \{\text{value (P), value (Q)}\} \\ \text{Value (P and Q)} &= \min \{\text{value (P), value (Q)}\} \end{aligned} \quad \textbf{(2)}$$

2.7 Defuzzification

The sixth step consists of defuzzification, in order to obtain a crisp value for the final estimate. Examples of such defuzzification methods are: MAX-MIN, MAX-DOT O MAX-PRODUCT, AVERAGE and ROOT SUM SQUARE (RSS). The EPCU estimation generated in the case studies was developed using 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.

Even though it is more complicated mathematically than other methods, it was selected because it gives the best weighted influence to all the Inference Rules involved (eg. 'fired' in the specialized vocabulary).

The steps to obtain the crisp value are:

1. Obtain the strengths for each fuzzy set belonging to the output membership function (RSS). Considering the values obtained in step 2.5 "Inference Rule evaluation" the strength for each fuzzy set defined for the output variable is evaluated. For example for the rulebase of Table1, the strengths are.

"Short" = (R1^2 + R6^2 + R9^2 + R18^2 + R21^2 + R22^2) ^0.5 **(3)**
"Average" = (R3^2 + R5^2 + R8^2 + R10^2 + R11^2 + R13^2+ R14^2 + R20^2 + R24^2+ R25^2) ^0.5
"Large" = (R2^2 + R4^2 + R7^2 + R12^2 + R17^2 + R19^2+ R23^2 + R27^2) ^0.5
"Very Large" = (R15^2 + R16^2 + R26^2) ^0.5

2. Obtain the "fuzzy" centroid of the area. The weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. The area obtained is divided by the sum of the weighted member function strengths and the result is taken as the crisp output.

CRISP VALUE = ("Short" center * "Short" _strength + "Average" _center * "Average" _strength + "Large" _center * "Large" _strength + "Very Large" center * "Very Large" _strength) / ("Short " strength + "Average "_strength + "Large "_strength + "Very Large "_strength) **(4)**

Table 1. Example Rulebase considering the input and output variables in the case study 1

RULE		VAR (P)		LINGUISTIC VALUE		VAR(Q)		LINGUISTIC VALUE		OUTPUT		OUTPUT VARIABLE
R1	IF	SW SIZE	IS	SMALL	AND	FUNCTIONAL COMPLEXITY	IS	LOW	THEN	ESTIMATED DURATION	IS	SHORT
R2	IF	SW SIZE	IS	LARGE	AND	FUNCTIONAL COMPLEXITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	LARGE
R3	IF	SW SIZE	IS	LARGE	AND	FUNCTIONAL COMPLEXITY	IS	LOW	THEN	ESTIMATED DURATION	IS	AVERAGE
R4	IF	SW SIZE	IS	AVERAGE	AND	FUNCTIONAL COMPLEXITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	LARGE
R5	IF	SW SIZE	IS	AVERAGE	AND	FUNCTIONAL COMPLEXITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	AVERAGE
R6	IF	SW SIZE	IS	AVERAGE	AND	FUNCTIONAL COMPLEXITY	IS	LOW	THEN	ESTIMATED DURATION	IS	SHORT
R7	IF	SW SIZE	IS	LARGE	AND	FUNCTIONAL COMPLEXITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	LARGE
R8	IF	SW SIZE	IS	SMALL	AND	FUNCTIONAL COMPLEXITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	AVERAGE
R9	IF	SW SIZE	IS	SMALL	AND	FUNCTIONAL COMPLEXITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	SHORT
R10	IF	SW SIZE	IS	SMALL	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	AVERAGE
R11	IF	SW SIZE	IS	SMALL	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	AVERAGE
R12	IF	SW SIZE	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	LARGE
R13	IF	SW SIZE	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	AVERAGE
R14	IF	SW SIZE	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	AVERAGE
R15	IF	SW SIZE	IS	LARGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	VERY LARGE
R16	IF	SW SIZE	IS	LARGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	VERY LARGE
R17	IF	SW SIZE	IS	LARGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	LARGE
R18	IF	SW SIZE	IS	SMALL	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	SHORT
R19	IF	FUNCTIONAL COMPLEXITY	IS	HIGH	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	LARGE
R20	IF	FUNCTIONAL COMPLEXITY	IS	LOW	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	AVERAGE
R21	IF	FUNCTIONAL COMPLEXITY	IS	LOW	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	SHORT
R22	IF	FUNCTIONAL COMPLEXITY	IS	LOW	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	SHORT
R23	IF	FUNCTIONAL COMPLEXITY	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	LARGE
R24	IF	FUNCTIONAL COMPLEXITY	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	AVERAGE	THEN	ESTIMATED DURATION	IS	AVERAGE
R25	IF	FUNCTIONAL COMPLEXITY	IS	AVERAGE	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	AVERAGE
R26	IF	FUNCTIONAL COMPLEXITY	IS	HIGH	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	LOW	THEN	ESTIMATED DURATION	IS	VERY LARGE
R27	IF	FUNCTIONAL COMPLEXITY	IS	HIGH	AND	SW DEVELOPMENT PROCESS FAMILIARITY	IS	HIGH	THEN	ESTIMATED DURATION	IS	LARGE

3 Case Studies

The EPCU model was tested in two different software organizations, one which develops software for external clients developing software packages, and the other a financial services organization which develops software for its own needs.

These two organizations are relatively small and considered representative of typical software development organizations in Mexico where 80% are small organizations (see Table 2: 'very small' = 63 & 'small' = 117) [7].

Table 2. Size of software organizations in Mexico [7]

Size	Workers	Workers Average	Organizations
Very Small	Less than 15	7	63
Small	16 to 100	60	117
Médium	101 to 250	175	14
Large	251 to 1000	600	11
Corporation	More than 1000	1500	1
Total			206

3.1 Case Study 1

Background on the data set

Organization 1 develops software packages with a staff of 30 and is organized into three areas: Marketing, Engineering and Consultancy.

The project in this case study for organization 1 is a .NET project to develop a B2B system for controlling the operations of shipping, transportation and delivery of packages for specialized organizations such as DHL or UPS. In addition, the B2B system must provide for contract and shipping management, package tracking, and so on.

The characteristics of the project in this case study are listed in Table 3. The project duration estimate of 6 months was made by the area Engineering Manager and Marketing Manager at the conceptualization stage of the project. This estimate was the basis on which management gave the go-ahead for the project, and on which the sale price was calculated. In reality, the project duration turned out to be 16 months, which means that the estimate fell short by 10 months (or 63% of the actual project duration).

Table 3. Characteristics of the project in Case Study 1

Features:	Variables:	Project Duration:
•Small Enterprise •SW B2B •Software Development Process •Actual Duration: 1 year 4 months •Estimated Duration without EPCU: 6 months	•SW Size (Small, Average, Large) •Functional Complexity (Low, Average, High) •Software Development Process Familiarity (Low, Average, High)	•Short, up to 2 months •Average, approx. 6 months •Large, approx. 12 months •Very Large, up to 24 months

Execution of the EPCU model on this data set

The characteristics of this project, as well as the experience of the practitioners were fuzzified using the 6 steps described previously. The inputs to the EPCU model are listed in the first three columns in Table 4: they represent the values assigned by the experienced practitioners for each input variable,(the range of these variables being 0 to 5 on an ordinal-type scale where 0 represents the very low value, and 5 the very high value). The data in the EXPERT 3 row were provided by the project manager, and the data in the next row represent the average for the three practitioners. For instance, the values assigned for the Software Development Process Familiarity (FP) variable are consistent at = 1 (very low) across the people interviewed, because this process was implemented in the organization when the project was developed and at a time when the process was not well understood.

In table 4, the 'EPCU Estimation' column presents the duration estimated using the EPCU model, using the input parameters for each expert, and fourthly the average over the 3 experts of each of the parameters. These estimated duration values are expressed in months. The next column to the right shows the difference between the actual duration and the duration estimated using the EPCU model. The results shown in this column reveal that the largest underestimated duration is the one provided by Expert 2, with an estimate of 11.4 months; this represents an underestimation of 4.6 months This is still better than the 10 months underestimation of the initial feasibility study without the use of the fuzzy logic model.

The input values provided by EXPERT 3 (the project leader) lead to an overestimation at 19.3 months, which means that the project was overestimated by 3.3 months (21%).

Table 4. Case Study 1: estimation outcomes and comparisons

	Software Size	Functional Complexity	SW Development Process Familiarity	EPCU Estimation	Actual Duration - EPCU Estimation	% underestimation	% overestimation
EXPERT 1	3	3	1	14.9	1.1	7%	
EXPERT 2	3	2	1	11.4	4.6	28%	
EXPERT 3 (Project Leader)	4	3	1	19.3	-3.3		21%
AVERAGE	3.3	2.6	1	16.0	0.0	0%	
Actual Duration	16						
Original Estimation	6						
Underestimation	63%						

Next, using the averages of the input values proposed by the practitioners (line 'Average' in Table 4), the estimation duration obtained using the model are very close (eg. the rounding is not shown in Table 4) to the actual values, that is 16 months.

Table 5 presents next an analysis of the potential impact of the use of this model. For illustrative purposes only, the following assumptions are made: the whole project is developed by a single developer who works 8 hours a day (or 160 hours in a month considered to be made up of 4 weeks) at a cost per hour of approximately $120 US$/hour).

With these assumptions, the cost with a 16 months duration is $307,200 (line 1), while the cost estimated at the feasibility stage without using EPCU was $115,200 (line 2): the difference represents then a large underestimation of $192,000,at the feasibility study.

Lines 3 to 5 in Table 5 present next the calculation with the data from the EPCU model based on the inputs of each experienced practitioner: the underestimation is over $87,458 and the overestimation is over $64,220.

By contrast, in line 6 with the use of the average of the inputs using the EPCU model, the estimates would have been very close to the actual duration (bottom line), and the costs very close.

Table 5. Case Study 1: Impact of estimation errors

	EPCU Estimation	COST (with assumptions) [USD]	Losses	Profits
ACTUAL COST		$307,200.0	$0.0	
COST CONSIDERING ORIGINAL ESTIMATION		$115,200.0	$192,000.0	
COST CONSIDERING EPCU ESTIMATION	14.9	$286,264.3	$20,935.7	
	11.4	$219,742.1	$87,457.9	
	19.3	$371,420.2		$64,220.2
	16.0	$306,451.2	$748.8	
Actual Duration	16			
Original Estimation	6			
hrs/months	160			
USD / hr	$120.00			

Verification of potential bias

The input values were assigned by the experienced practitioners taking into account their expertise relative to the organization know-how: it is difficult to know whether the accuracy of the estimates depends on their own expertise or on the estimation model developed.

To investigate this, another round of use of the model was carried out, this time with inputs parameters provided by two system engineering students with no work experience and who did not have any relationship with the organization of case study 1. Both the inputs and the estimated duration for this second round are presented in Table 6.

Table 6. Case Study 1: Estimation outcomes and comparisons by students

	Software Size	Functional Complexity	SW Development Process Familiarity	EPCU Estimation	Actual Duration - EPCU Estimation	% underestimation	% overestimation
STUDENT 1	4	4	3	16.5	-0.5		3%
STUDENT 2	3	4.5	3	13.6	2.4	15%	
Actual Duration	16						
Original Estimation	6						
Underestimation	63%						

The largest underestimated duration with the EPCU model is the one with the inputs provided by STUDENT 2, with an under-estimate of 2.4 months, compared to the 10-month under-estimate without the use of any model. The input values provided by STUDENT 1 resulted in a small 0.5 month of over-estimated duration.

The results of this second round, suggest that there might not be an embedded bias due to inputs from experiences practitioners. Of course, much more data would be required to confirm whether of not there is such bias, and if so, its specific contribution.

3.2 Case Study 2

The data of the second case study come from a small software group in a financial services organization. The project reported in this case is a software enhancement of a production client/server system (i.e. it involved a group of functional modifications to the software and the redeployment in the production environment). The project was carried out by two developers, one of whom had had no hands-on involvement in this software application, and the other was new in the area and was assigned as development leader. Consequently, this individual had no experience with the development tool, although he did have some knowledge of this type of software application. The actual duration of this project was 9 months, while the duration had originally been estimated at 4 months only. A list of the characteristics of the project in case study 2 is provided in Table 7.

Table 7. Characteristics of the project in Case Study 2

Features:	Variables:	Project Duration:
•Internal Area •Client/Server SW •Maintenance •No Experience with the Development Tool •2 individuals (one was new in the area) •Actual Duration: 9 months •Estimated Duration without EPCU: 4 months	•SW Size (Small, Average, Large) •Experience Developing Similar Applications (Low, Average, High) •Experience with the Development Tool (Low, Average, High)	•Short, less than 3 months •Average, 4 to 10 months •Large, approx. 11 months •Very Large, up to 17 months

The inputs for the EPCU model were obtained from the project coordinator (left-hand columns in Table 8). The input data were collected only from the project coordinator, since the individual who was assigned to the project was new, and consequently his opinions as to input values were not considered to be adequate.

The estimated duration obtained with the EPCU model is 9.3 months, with represents a small over-estimation of 0.3 months compared to the actual duration of 9 months.

Table 8. Case Study 2: Estimation inputs and outcomes

	Software Size	Experience Developing Similar Applications	Development Tool Experience	EPCU Estimation	Actual Duration - EPCU Estimation	% underestimation	% overestimation
Coordinator	2	3	0	9.3	-0.3		3%
Actual Duration	9						
Original Estimation	4						
Underestimation	56%						

Again, for illustrative purposes only, consider that the whole project was developed by a single developer who worked 8 hours a day (or 160 hours in a month considered to be made up of 4 weeks), at a cost per hour of approximately $120 USD). The results, based on these assumptions, are shown in Table 9.

Table 9. Case Study 2: Impact of estimation errors

	EPCU Estimation	COST (with assumptions) [USD]	Losses	Profits
ACTUAL COST		$172,800.0		
COST CONSIDERING ORIGINAL ESTIMATION		$76,800.0	$96,000.0	
COST CONSIDERING EPCU ESTIMATION	9.3	$178,560.0		$5,760.0
Actual Duration	9			
Original Estimation	4			
hrs/months	160			
USD / hr	$120.00			

Table 9 shows that the actual cost (using the cost assumptions of a developer) was approximately $172,800, and the cost estimated without using EPCU was approximately $76,800. This means a project loss of approximately $96,000. Using the EPCU model, the estimation was 9.3 months, which translates to a cost of approximately $178,500 and a potential profit generated of approximately $5,760, as against the $96,000 loss without the support of the estimation model used for this case study.

4 Discussion on the use of the EPCU Model

For case study 1, the estimated duration which was the basis on which management gave the go-ahead and on which the sale price was calculated, was 6 months; the actual duration was 16 months, that is, an underestimation of 10 months, or 63% over the original estimate. Underestimation of this magnitude is common in industry, as illustrated in the Standish Group report [8].

The impact of projects with costs over budget and schedule delays is a loss for the organization that asked for the product. In addition, this underestimated project would most likely have deteriorated the organization's relationship with the supplier of the software. Another example is the impact on the organization that developed the product: with a schedule (and a budget) underestimated by 50%, they would not be able to deliver the yield as projected in the feasibility study.

By contrast to the estimate originally based only on the practitioners' experience, the proposed EPCU model would have enabled the generation of multiple estimation scenarios with, even in the worst-case scenario, an under-estimated duration of 4.5 months, or 28% which represents about half the delay experienced in reality. Even in the worst case estimate, the loss incurred by the organization that developed the software would have been lower by at least half, and the potential deterioration in the supplier-client relationship would not have been so great. On the other hand, with the best estimate obtained using the scenario with averages of the inputs estimates, there would have been almost no delay, no additional cost and no deterioration in the relationship with the client.

In case study 2, the estimated duration which was the basis on which management gave the go-ahead and on which the cost was calculated was 4 months, while the actual duration was 9 months, which amounts to an under-estimation of 5 months, or 56%. The duration estimated using the EPCU model was close at 9.3 months, that is, only 3% over actual project duration. In this case 2, the project would have generated a profit and not the loss that occurred.

For case study 2, the original estimate of duration was made by the same person who made the estimation of the inputs to the EPCU model. This individual had identified the input variables required to define the model for his organization, which illustrates that it is not easy, even with the same experienced individuals doing the work, to systematically replicate estimates.

The readers are reminded that case studies provide illustration of the use of models and techniques and help provide insights valuable to understand both the benefits as well as the constraints. These insights can then be used to design experiments to verify in a more systematic manner the quality of the estimation models developed.

Acknowledgements

This research project has been funded partially by the European Community's Sixth Framework Programme – Marie Curie International Incoming Fellowship under contract MIF1-CT-2006-039212.

5 References

1. A. Idri, A. Abran, T.M. Khoshgoftaar, Robert, S. "Estimating Software Project Effort by Analogy Based on Linguistic Values" , 8th IEEE International Software Metrics Symposium, Ottawa, Ontario, 2002, pp. 21-30.

2. Idri, A., Abran, A., Khosgoftaar, T.M., Fuzzy Analogy: A New Approach for Software Cost Estimation, in International Workshop on Software Measurement (IWSM'01), Montréal, Québec, 2001, pp. 93-101.

3. Idri, A., Abran, A., Khoshgoftaar, T., Robert, S., 'Fuzzy Case-Based Reasoning Models for Software Cost Estimation', in Soft Computing in Software Engineering:,Studies in Fuzziness and Soft Computing, Springer-Verlag, 2004, pp. 64-96.

4 Idri, A., Abran, A., Towards A Fuzzy Logic Based Measures for Software Projects Similarity, 6th MCSEAI'2000 – Maghrebian Conference on Computer Sciences, Fez, Morocco, 2000.

5. Seattle Robotics Society, http://www.seattlerobotics.org/encoder/index.php

6. Answer Math , http://www.answermath.com

7. "Programa para el Desarrollo de la Industria del Software (PROSOFT) Versión 1.3", Secretaría de Economía , Mexico.

8. Extreme Chaos, The Standish Group International, Inc. 2000-2004 Reports.