Expert Estimation and Historical Data: An Empirical Study

Gabriela Robiolo
Universidad Austral
Av. Juan de Garay 125
Buenos Aires, Argentina
grobiolo@austral.edu.ar

Silvana Santos
Universidad Nacional de La Plata
Calle 50 y 20
La Plata, Argentina
silvanasantos@gmail.com

Bibiana Rossi
Univ. Argentina de la Empresa
Lima 717
Buenos Aires, Argentina
birossi@uade.edu.ar

***Abstract*— Expert estimation is the estimation strategy which is most frequently applied to software projects; however, this method is not very much reliable as the accuracy of the estimations thus obtained is always influenced by the level of experience of the expert. As part of the experts' experience is made up by the information they obtain from historical data, we wanted to learn about the value such historical data has for an expert estimator. To do so, we designed an empirical study. We compared the accuracy of the estimations made with several estimation methods based on productivity, size, and analogies which use historical data, to that obtained with expert estimation. We used two similar applications; one was used as the target application and the other one was used to obtain historical data. The results show that the accuracy of expert estimation is affected by the expert's work experience, the level of experience he/she has in the technologies to be used to develop the applications, and his/her level of experience in the domain of the applications. The use of historical data may improve the intuitive expert estimation method when the work experience, the experience in the technologies to be used to develop the application, and the experience in a given domain is low, as well as when the team velocity is unknown.**

Keywords—Expert; Expert Estimation; Effort Estimation; Empirical Study; Historical Data.

I. INTRODUCTION

Expert estimation is the estimation strategy which is most frequently applied today to estimate the effort involved in the development of software projects, and this is so because there is evidence in favor of using it [1]. However, the estimations thus obtained are far from being as accurate as we would like them to be so, if we expect to improve estimation accuracy, further research should be carried out in order to understand how the estimation process works.

With this goal in mind, we found out that the compilation of information about cost estimation made by Jørgensen and Shepperd [2] in 2004 is extremely valuable, since they systematically reviewed papers already written on cost estimation studies and they provided recommendations for future research. They found out that there are few researchers working in this field and that there is no adequate framework to develop high quality research projects that may lead to concluding evidence. Consequently, they suggested the following improvements in the field of research: (a) deepen the study of the basic aspects of software estimation, (b) widen the research on the current, most commonly used estimation methods in the software industry, (c) perform studies that support the estimation method based on expert judgment, instead of replacing it with other estimation methods and (d) apply cost estimation methods to real situations.

As we completely agree with their diagnosis, we believe research on expert estimation has become mandatory, if more accurate estimations are to be obtained.

As far as we know, expert estimation may be said to be based on both intuition, which is acquired by the developer through his daily work experience, and analogy. In fact, such analogy will be made by using both the information the estimator has in his memory and the historical data he may obtain [2]. Although all experts are expected to have some experience, the types of experience they have may be very different, and their estimation performances will surely be different too. Besides, even in cases in which the expert is supposed to have wide experience, there will be factors that will undoubtedly affect his estimations. For example, the domain where the software estimation must be made could be new to him, the team he would work with may have been recently created or the technological environment may not have been previously used.

In Agile contexts, in particular, there is another critical aspect to be dealt with: not knowing the velocity at which the developing team works. Actually, Cohn [3] suggested that one of the challenges when planning a release is estimating the velocity of the team. He mentioned three possible ways to estimate velocity. Firstly, estimators may use historical averages, if available. However, before using historical averages, they should consider whether there have been significant changes in the team, the nature of the present project, the technology to be used, and so on. Secondly, estimators may choose to delay estimating velocity until they have run a few iterations. Cohn thinks that this is usually the best option. Thirdly, estimators may forecast velocity by breaking a few stories into tasks and calculating how many stories will fit into the iteration.

Bearing in mind the present working conditions, as described in the two previous paragraphs, and in order to deepen our knowledge about expert estimation, as recommended by Jørgensen and Shepperd [2], we decided to research on the importance of historical data when performing expert estimations in agile contexts in which the project domains and the technological environments are new to the team, and the teams -with little experience in Agile contexts- have recently been created, so the team velocity is unknown.

 ISBN: 978-1-61208-304-9

In this scenario, we tried to answer the following research question: *when may the accuracy of an expert estimation made in a context of agile software development be improved by using historical data?* The results we obtained through our empirical study have led us to conclude that historical data may improve the accuracy of an intuitive estimation made by an expert when the estimator has limited experience in the job to be performed, the technologies to be used and the domain to be dealt with, and when the team velocity is unknown.

In section two, we will introduce three estimation methods: Expert Estimation (ExE), Analogy-Based Method (AbM), and Historical Productivity (HP). In section three, we will describe an empirical study and analyze the results obtained. In section four, we will investigate related work to see if there is any other evidence of improvement in expert estimation accuracy when using historical data, and finally, in section five, we will draw conclusions as regards the evidence of the benefits of using historical data.

II. ESTIMATION METHODS

This section will describe the three estimation methods used in our empirical study: ExE, AbM and HP. However, before doing so, it is important to focus on the definition of certain expressions used to define such methods. For example, when defining expert, Jorgensen [1] used a broad definition of the phrase, as he included estimation strategies that ranged from unaided intuition ("gut feeling") to expert judgment supported by historical data, process guidelines, and checklists ("structured estimation"). In his view, for an estimation strategy to be included under the expert estimation category, it had to meet the following conditions: first, the estimation work must be conducted by a person recognized as an expert in the task, and second, a significant part of the estimation process must be based on a non-explicit and non-recoverable reasoning process, i.e., "intuition". In our study, however, a narrower definition of the concept of expert was used: that which refers only to intuition. This way, we made a difference between intuitive ExE, and the methods that involve the use of historical data: AbM and HP. It is important to note that in our study, when we used Planning Poker –an ExE method-, no historical data was taken into account.

To further clarify the terms used, we must say that by AbM we meant the estimation performed by an expert, who is aided by a database containing information about finished projects [4]. As regards HP, which is another way of using historical data, it is worth mentioning that in our empirical study we focused on the size characteristic of the products, as suggested by one of the authors that inspired this article [4].

A. Expert Estimation Method (ExE)

When estimating the effort of a software development task, an expert estimation may be obtained either by a single expert, whose intuitive prediction will be considered an expert judgment, or by a group of experts, whose estimation will combine several experts' judgments.

A very frequently used way to obtain group expert judgment is called Planning Poker, a technique that combines expert opinion, analogy, and disaggregation. It is based on the consensus that is reached by the group of experts who are performing an estimation; in fact, it is considered a manageable approach that produces fast and reliable estimations [3][5][6]. This method was first described by James Greening [8] and it was then popularized by Mike Cohn through his book "Agile Estimating and Planning" [3]. It is mainly used in agile software development, especially in Extreme Programming [7]. To apply Planning Poker, the estimation team should be made up of, ideally, all the developers within the team, that is, programmers, testers, analysts, designers, DBAs, etc. It is important to bear in mind that, as this will happen in Agile contexts, the teams will not exceed ten people [3]. In fact, Planning Poker becomes especially useful when estimations are taking too long and part of the team is not willing to get involved in the estimation process [8]. The basic steps of this technique, according to how Grenning described them, are:

"The client reads a story and there is a discussion in which the story is presented as necessary. Then, each programmer writes his estimation on a card, without discussing his estimation with anyone else. Once every programmer has written down his estimation, all the cards are flipped over. If everybody has estimated the same, there is no need for discussion; the estimate is registered and the next story is dealt with. If the estimates are different, the team members will discuss their estimates and try to come to an agreement" [8].

Mike Cohn further developed this technique: he added a pack of cards especially designed to apply this technique and he shaped the whole process: each estimator is given a pack in which there are cards that have numbers written on them Those numbers represent a valid estimation, such as 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100. Each pack has to be prepared before the Planning Poker meeting and the numbers should be big enough to be seen from the other side of the table. There is a raison d'être for the estimation scale presented above. There are studies which have demonstrated that we are better at estimating things which fall within one order of magnitude [9][10], so these were the cards that were employed when Planning Poker was used in the empirical study reported in this article. It should be noted that no historical data was used when estimating with Planning Poker for our study.

B. Analogy-Based Method (AbM)

The idea of using analogy as a basis to estimate effort in software projects is not new: in fact, Boehm [11] suggested the informal use of analogies as a possible technique thirty years ago. In 1988, Cowderoy and Jenkins [12] also worked with analogies, but they did not find a formal mechanism to select the analogies. According to Shepperd and Schofield [13], the principle is based on the depicting of projects in terms of their characteristics, such as the number of interfaces, the development methodology, or the size of the functional requirements. There is a base of finished projects which is

used to search for those that best resemble the project to be estimated.

So, when estimating by analogy, there are *p* projects or cases, each of which has to be characterized in terms of a set of *n* characteristics. There is a historical database of projects that have already been finished. The new Project, the one to be estimated, is called "target". Such target is characterized in terms of the previously mentioned *n* dimensions. This means that the set of characteristics will be restricted to include only those whose values will be known at the time of performing the prediction. The next step consists of measuring similarities between the "target" and the other cases in the *n*-dimensional space [14].

Such similarities may be defined in different ways, but most of the researchers define the measuring of similarities the way Shepperd & Schofield [13] and Kadoda, Cartwright, Chen & Shepperd [14] do: it is the Euclidean distance in an *n*-dimensional space, where *n* is the number of characteristics of the project. Each dimension is standardized so that all the dimensions may have the same weight. The known effort values of the case closest to the new project are then used as the basis for the prediction.

In our empirical study, we applied AbM in its simplest version. The participants compared the user stories of two projects: one considered "historical" and the other one "target". The Estimated Effort (EE) of the user story of the target project was, in fact, the Actual Effort (AE) of the "most similar" user story of the historical project. Actually, no specific characteristics of the user stories were specially taken into account.

C. Historical Productivity

Jørgensen, Indahl, and Sjøberg [4] defined Productivity as the quotient of Actual Effort (AE) and Size, and the EE as the product of Size and Productivity. In this empirical study, COSMIC [15] was used as a measure of Size, and EE was calculated as the product of Size and Historical Productivity (HP). The HP is the value of productivity of the project to be used as historical project, that is, the quotient of the AE and the Size of the historical project.

To measure size, COSMIC was selected because it is an international standard [16] that is widely recognized in the software industry, and also because there is a previous study that used it in an Agile context [17]. With the COSMIC software method, the Functional User Requirements can be mapped into unique functional processes, initiated by functional users; in fact, user stories are actually used in this paper. Each functional process consists of sub-processes that involve data movements. A data movement concerns a single data group, i.e., a unique set of data attributes that describe a single object of interest. There are four types of data movements: a. an Entry moves a data group into the software from a functional user, b. an Exit moves a data group out of the software to a functional user, c. a Read moves a data group from persistent storage to the software, and d. a Write moves a data group from the software to persistent storage.

In the COSMIC approach, the term "persistent storage" denotes data (including variables stored in central memory) whose value is preserved between two activations of a functional process.

The size expressed in CFP is given by the equation CFP = Entries + Exits + Reads + Writes, where each term in the formula denotes the number of corresponding data movements. So, there is no concept of "weighting" a data movement in COSMIC, or, equivalently, all data movements have the same unit weight.

III. Description of our Empirical Study

Our empirical study is described in this section, considering its conception, how it was planned, the particularities of its execution and the results obtained.

A. Definition

This empirical study was designed in order to establish when the accuracy of an expert estimation made in a context of agile development, under the circumstances that will be described below, may be improved by using historical data. Such circumstances are: the project domain and the technological environment must be new to the estimator, and the team would have recently been created, so that the team velocity will be unknown.

The development steps of this empirical study may be summarized as follows:

The study was developed in the context of graduate education for IT practitioners from different educational and work backgrounds. The participants attended a workshop which had two objectives, one oriented to the subjects and another one oriented to the development of this empirical study. The workshop gave the participants the opportunity to: a. understand both how a historical database is built, and under which circumstances such database will give value to the estimation process, b. estimate using three methods and c. compare their results with other participants' results. Later on, the same workshop was conducted for undergraduate students.

The workshop participants were asked to re-estimate the first spring of an application that had been previously developed by a group of undergraduate students who did not participate of the workshop. The selected application had been developed using a development language unknown by the workshop participants and the application belonged to a domain the latter knew little of. The original team velocity was not reported to the participants, to simulate that it was unknown.

The re-estimations were made using three different estimation methods: ExE, based on the participants' intuition, and two other methods which use historical data. The historical data was obtained from a similar application that had been developed by a third undergraduate group –a group that had neither developed the original application nor participated of our empirical study-.

To guarantee the best results, we followed the recommendations of Juristo and Moreno [18] and Wohlin et al. [19] in order to develop this empirical study. To report it, we took into account Jedlitschka, Ciolkowoski

 ISBN: 978-1-61208-304-9

and Pfahl's guidelines for reporting empirical research in software engineering [20].

As previously stated, the objective of this empirical study was to analyze when the accuracy of an estimation made by an expert, based on his personal intuition, may be improved by using historical data. This objective was achieved by comparing the estimation errors obtained by two different groups: undergraduate students and practitioners, when estimating using three different methods: ExE, AbM, and HP.

In fact, the hypotheses to be tested were:

H_0: The mean value of the MRE calculated with the ExE is equal to the mean value of the MRE obtained when calculating with AbM or HP.

H_1: The estimated mean value of the MRE calculated with the ExE is lower than the mean value of the MRE obtained when calculating with AbM or HP.

B. Planning

The *experimental subjects* were IT graduate students and undergraduate advanced students of Informatics Engineering. In fact, all of the graduate students were practitioners. So, in this paper, when we say "participants" we mean both the graduate and undergraduate students, and by "practitioners" we refer only to the graduate students.

The participants were asked to give some information about themselves regarding the following aspects:

- If graduate or undergraduate student
- Professional experience (they had to state the number of years they had worked in software development)
- Experience with COSMIC
- Experience with user stories (they had to inform the number of user stories that they had written/read (fewer than 20, 20-100, more than 100)
- Experience with Ruby [21] language.
- Experience in Database development
- Experience in working in Agile development contexts.
- Level of prior knowledge about the productivity of the teams that developed the experimental objects (high, medium, low)
- Level of experience in the technologies used to develop the experimental objects (high, medium, low)
- Level of experience in the domain of the experimental objects (high, medium, low)

The *experimental objects were* two similar applications (P1 and P2), which were social networks. The first application was a system through which users may conduct surveys. The system classifies users into several categories, builds different groups and instantly surveys those users who fall within the right categories. It was developed by a team of undergraduate students who registered the estimated and actual hours using the Scrumy tool [22], and who were supervised by two professors.

The second application, which we identified as the "target project", was a network where different types of events may be published. For example, an event may be a party, a meeting or a football game. Events are the core elements in this application, not people. It works with event and friend suggestion algorithms and gives the option of buying a ticket for an event online.

The data corresponding to the experimental objects are displayed below. Table 1 shows the user stories of P1 and the Actual Effort (AE) of each user story measured in man hours. As some user stories were not functional processes, they were discarded. Table 2 shows the user stories of P2, which are the user stories of only the first sprint, as it was the only sprint for which effort was estimated.

As regards the counting of the man-hours worked on P1 and P2, one of the tasks within the assignment the undergraduate students that developed the projects had to undertake was to register the hours worked. These two groups did not participate in the empirical study; in fact, they were undergraduate students from a university different from the one where the undergraduate participants studied. The applications were developed in an Agile context, as an assignment in a practical subject. They first estimated the work to be done and then compared their estimations to their real effort. Two professors supervised these tasks. This empirical study used the actual effort of P1 and P2 and the estimated effort of P2 (obtained by the original development group), so that they may be compared to the participants' results.

The aspects of the development process that were controlled to facilitate such comparison were:

- Similarity: Two similar applications that had been developed in Agile contexts were selected as experimental objects. They had been developed in an academic context by advanced undergraduate students, who had been requested to develop an application for an assignment in which a company environment was simulated.
- Experience in team velocity: Since in Agile contexts developers learn from previous estimations, and in this case the estimators were expected to have no previous experience, only the first sprint of the target application could be estimated in order to be compared to the actual effort estimation of P2, as it was only for the first sprint that the original P2 estimators did not have experience in team velocity.
- Language experience: Participants with experience in Ruby language, in Agile contexts, and / or COSMIC were equally distributed.

In order to obtain comparable results in this study, man-hours had to be used to unify the unit of measurement of effort, as the historical values had been previously measured in man-hours, instead of in story points or ideal hours, which are the measures usually used to make effort estimations with Planning Poker in Agile contexts [3].

The workshop was run following these steps:

1) The participants were given a set of materials that included: Brief Vision Documents [23] of P1 and P2, the

professor's slides explaining the empirical study, and an Excel file where each sheet was a step of the empirical study.

TABLE I. DATA OF THE APPLICATION TO BE USED AS HISTORICAL INFORMATION

P1	Actual Effort [man-hours]
Create survey	18
Sign up	15
See user's profile	9
Answer survey	9
Log in/Log out	6
Comment on survey	12
Search for survey	9
Eliminate user	3
Edit personal data	6
Search for user	9
Generate and publish statistics	30
Follow user	30
Select user segment	18
Sort the content according to date	18
Upload pictures	21
UPR (User Popularity Ranking)	36

TABLE II. USER STORIES OF THE TARGET APPLICATION

P2
Create, Modify and Eliminate User
Log in (Log out)
Create event
Search for event

2) Each one of the empirical study steps was explained to the participants. The participants were trained to perform each activity. Also, two examples of COSMIC measurement were included.

It is important to note that the participants worked with an Excel file that was designed to facilitate the understanding of the activities, and the sequence in which they had to do them. The following are the activities presented sequentially in each one of the sheets in the file:

a) *Perform the expert estimation*, based on their intuition, they estimated the man hours to be worked on the target application (P2). Based on the Vision Document of P2, the participants estimated the EE of each user story described in Table 2.

b) Build the historical database. The participants measured the size of the user stories of the historical application (P1) by using COSMIC, as shown in Table 1. The Excel sheet automatically calculated the Historical Productivity (HP) of P1 as the quotient of AE_{P1} and $Size_{P1}$, where AE_{P1} is equal to the sum of the AE of each user story of P1, and $Size_{P1}$ is equal to the sum of the Size of each user story of P1. The data movements of P1 were indentified for each user story, based on: the information included in the Vision Report, the name of the user story, and the explanation given by the leader of the workshop when asked for it. The measurement of the user stories, using COSMIC, was performed in a way similar to that of [17].

c) *Measure the size of the target application (P2), by using COSMIC to measure the size of the user stories.* These size values were automatically used to calculate $EE_{P1,}$ which was calculated as the product of $Size_{P1}$ and Historical Productivity (HP_{P2}).

d) *Estimate the effort for the target application (P2) using AbM.* The participants had to select for each one of the user stories in P2 the most similar user story from the set of user stories in P1 -though based on their characteristics, not on their size- and then assign to the Estimated Effort (EE) of each user story in P2 the AE of the similar user story in P1.

e) *Individually compare and analyze the EE values obtained using ExE, AbM, and HP methods*. The Excel sheet automatically presents a Table which displays the three EE values –those obtained by applying the three different estimation methods- for each user story in P2.

3) The participants estimated the effort of the target application following the steps listed above, and completed the worksheets.

4) The data was collected and the results were analyzed with the participants. A rich discussion about the comparison of the MRE obtained by applying the three estimation methods (ExE, HP and AbM) was conducted by the leader of the empirical study.

C. Execution

The characteristics of the participants are described in Table 3.

Forty nine undergraduate students, who were distributed in fourteen groups of 3-4 students, participated in the two workshops. The median work experience of the students was three years. No one had experience using COSMIC, and they had little experience with user stories. All of them had approved the course "Database" and only 8 had experience in working in an Agile context, that is to say, a small proportion of them. The Level of experience of the development teams in the technologies to be used and in the domain of the experimental objects was low. In one of the workshops, fourteen practitioners worked on their own. The median work experience of the practitioners was fourteen years. No one had experience in using COSMIC, and five of them had experience with user stories.

Their median experience in "Database" was ten years and only three of them had experience in working in an Agile context, which is a small proportion. The Level of experience in the technologies and in the domain of the experimental objects was medium-low.

Table 4 shows the effort estimation values of the target project, obtained by the two groups applying the three estimations methods: ExE, HP, and AbM. Moreover, the AE of the student group that developed the target application (P2) was 35 man-hours.

Figure 1 shows the boxplots of the residuals and Figure 2 the boxplots of the MRE for the target project. To obtain the MRE, the actual value registered for the first sprint of P2 by the group that actually developed the project was used as AE.

 ISBN: 978-1-61208-304-9

TABLE III. WORKSHOP PARTICIPANTS

Type	Number	Work Expe-rience (Years)	Expe-rience using COSMIC	Number of User Stories [<20, 20<US<100, >100]	Database Experience	Experience with Ruby Language	Work expe-rience in Agile context	Experience in the technologies	Expe-rience in the domain
Under graduate	49 (14 groups)	[0-13] Median: 3	No one	<20: 44 20<US<100: 3 >100: 2	All had approved the Course "Database"	No one	Only 8	Low: 47 Average: 2 High: 0	Low: 43 Average: 4 High: 2
Practi-tioners	14	[4-36] Median: 14	No one	<20: 9 20<US<100: 3 >100: 2	Database experience measured in years [0-36] Median:10	Only one	Only 3	Low: 9 Average: 5 High: 0	Low: 11 Average: 3 High: 0

The boxsplots show the different results obtained by each group of participants. The undergraduate participants obtained better estimation results when applying the AbM, rather than the ExE and HP methods. Figure 2 shows the median values, but it must be noted that a more significant difference was observed when comparing the values obtained for the mean MRE in the undergraduate group: AbM: 69.80, ExE:151,43 and HP:175,04. On the other hand, the practitioners group obtained the best results when applying ExE, instead of HP and AbM, as shown by the boxplots. Also, their mean values were ExE: 29.09, HP: 205.16, and AbM: 87.14.

D. Threats to validity

The difference in background of the experimental subjects is the major weakness of this empirical study. However, this drawback may be transformed into a strength if we consider that in this empirical study the experience of the expert is stressed, showing that the accuracy of an expert estimation depends on the estimator's expertise, which is measured by his work experience, his level of experience in the technologies used to develop the experimental objects and his level of experience in the domain of the experimental objects.

Another threat is that the expert estimations were made in two different manners: either alone or in groups. The practitioners worked alone and the undergraduate students formed groups of three or four persons and used Planning Poker to obtain the expert values. However, we think that this combination of expert methods, that is, using Planning Poker or not, did not introduce bias in this study, in accordance with what was reported in [24].

Unfortunately, only a brief explanation about COSMIC was given to the undergraduate students, since it was not possible to give an extensive explanation, as there was not enough time to do so (the whole workshop was three hours long). Thus, the little available time was devoted to those COSMIC characteristics that were necessary for them to know in order to make a correct measurement. However, this did not seem to be a big problem, as the concept of data movement is quite intuitive for all the participants and the medians of the errors shown in both Figure 1 and 2 for the HP method are similar.

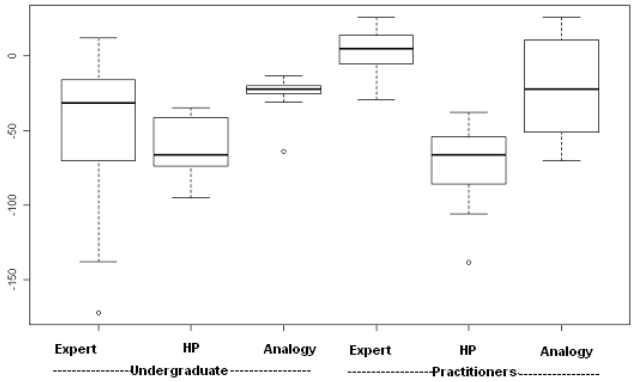

Fig. 1. Boxplots of the residuals of the target project

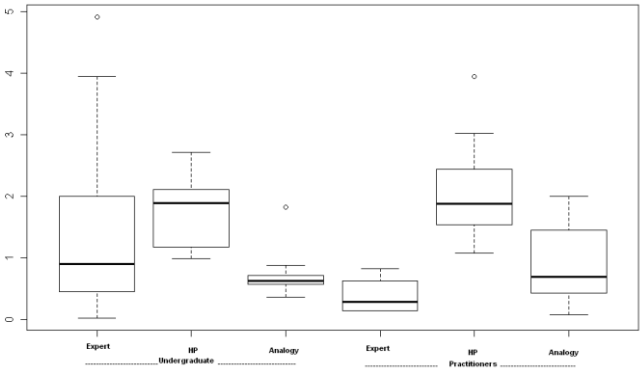

Fig. 2 Boxplots of the MRE of the target project

Also, the use of examples and previous training in Function Points made it easier for the participants to understand how to use this measuring method. On the other hand, the practitioners had been previously trained in COSMIC, so they presented no difficulty. Besides, if anybody had any doubts, the person who led the empirical study would give further explanations.

The order in which the estimations were performed could have introduced bias in the result, so it would have been more convenient if the participants had not performed the estimations in the same order, except for ExE, which must always be performed in the first place.

 ISBN: 978-1-61208-304-9

The selection of a similar application to make up the historical database is clearly an advantage in order to obtain a better estimation, but the problem is that sometimes the estimator does not have data about similar applications at hand, so she or he has to use an application from a different domain. This circumstance may vary the results obtained in this empirical study.

The experimental subjects were identified either as undergraduates or practitioners. However, it may be argued that more categories would have been necessary, as some of the practitioners had more experience in the domain or in the technologies than some others. Consequently, to obtain more evidence of the benefit of using historical data, it is necessary to have a bigger number of estimators, which would allow us to identify different levels of expertise, for example, three expertise levels for practitioners and three for undergraduates.

To conclude, as the experimental objects used in the empirical study came from a particular environment and the experts' experience did not cover the big spectrum of expertise that exists, general conclusions cannot be drawn because there may be different estimation problems in different environments and experts' performances.

IV. DATA ANALYSIS AND INTERPRETATION

To answer the research question posed above, it is important to understand the circumstances under which the use of historical data may improve the expert estimation accuracy. In this empirical study, two types of experts were involved: we called them undergraduate and practitioner participants. Consequently, each group will be separately analyzed first, then the statistical significance of the results will be dealt with, and afterwards, the research question will be answered. Finally, this will be completed with the discussion of aspects omitted in the previous sections.

A. Result analysis

1) Undergraduate

We noticed that there were three aspects that affected the intuitive expert estimation: the work experience, the level of experience in the technologies used to develop the experimental objects, and the level of experience in the domain of the experimental objects. The undergraduate participants' work experience measured in years varied from 0 to 13, with a median of 3. This shows that the "experts" had little experience in estimations and also, that the level of experience in the technologies used and in the domain was low.

Their best result was obtained when using AbM: the MRE median was 63% within a [37%-183%] range. The lack of experience, in this case, was compensated for by the historical data.

By using HP, the MRE dispersion was increased: the MRE values ranged from [99%-272%]. The MRE of the 14 groups had a median of 189% and a standard deviation of 54%.

2) Practitioners

When compared to the undergraduate participants, the most significant difference was their work experience:

TABLE IV. EE OF THE TARGET PROJECT

Participants	Number of estimations	ExE %	HP %	AbM %
Undergraduates	14 (made by groups of 3-4 undergraduate students)	161.00	110.00	57.00
		61.00	74.70	57.00
		34.00	76.30	60.00
		65.00	69.72	48.00
		207.00	84.12	48.00
		85.00	106.13	55.00
		173.00	90.21	66.00
		68.00	102.84	57.00
		79.00	101.15	57.00
		56.00	101.44	51.00
		51.00	72.00	57.00
		32.00	130.15	57.00
		105.00	108.93	99.00
Practitioners	14	11.00	108.94	11.00
		30.00	173.22	24.00
		21.00	84.77	20.00
		30.00	122.15	60.00
		9.00	90.55	9.00
		64.00	85.96	39.00
		30.00	120.61	105.00
		29.00	111.87	86.00
		16.00	72.88	32.00
		30.00	105.05	95.00
		40.00	88.93	57.00
		40.00	97.07	94.00
		49.00	92.37	70.00
		57.00	140.94	57.00

measured in years, it varied from 4 to 36, with a median of 14. Ten practitioners were project leaders or managers, three were senior developers and only one was a junior developer. This shows that these "experts" had experience in project management and, of course, in estimations.

The practitioners' level of experience in the technologies used to develop the experimental objects and the level of experience in the domain of the experimental objects was medium-low. These characteristics justify the results obtained when using ExE.

During the study, three of them did not perform the expert estimation because they considered that they were no "experts", while two of them assigned to the expert estimation the same value they had assigned to the AbM estimation. Seven of the eleven practitioners that applied pure expert estimation estimated with an MRE lower than 25%.

The estimation by AbM had a MRE median of 70 % in a range result of [8.57%-200%], which is a result similar to that obtained by the undergraduates.

By using HP, the MRE dispersion was increased: [108.22%-384.91%]. The MRE of the 14 practitioners had a median of 189% -similar to that of the undergraduate value-and a big standard deviation of 75%, which may have been caused by the subjectivity introduced by COSMIC, originated by the practitioners' different backgrounds.

3) The statistical significance of the results

The Wilcoxon rank test, at a significance level of 0.05, was used to analyze the statistical significance of the

 ISBN: 978-1-61208-304-9

results. This non-parametric test was selected because the distributions of the variables were not normal. It was applied to test the accuracy of ExE versus that of HP or AbM, according to the results obtained by each group (practitioners and undergraduate participants). The MRE and the absolute residuals were used. Table 5 shows the p-value of each subset, when using the MRE. The results obtained when using the absolute residuals are not shown because there is no significant difference.

TABLE V. STATISTICAL SIGNIFICANCE

Groups	ExE vs:	MRE
Undergraduate	HP	0.162
	AbM	0.948
Practitioners	HP	0.000
	AbM	0.022

When analyzing the MRE obtained by:

- the practitioners, when comparing ExE to HP, it was possible to reject H0 in favor of H1.
- the practitioners, when comparing ExE to AbM, once again, it was possible to reject H0 in favor of H1.
- the undergraduates, when comparing the ExE method to HP, it was not possible to reject H0 in favor of H1.
- the undergraduates, when comparing the ExE method to AbM, it was not possible to reject H0 in favor of H1.

It should be noticed that the three practitioners who did not use the method, as they did not consider themselves to be "experts", were also included in the table. However, later on, the Wilcoxon rank test was also computed, but this time only for the eleven practitioners who made the estimations, and the results did not vary.

Now we can answer the research question: W*hen may the accuracy of an expert estimation made in a context of Agile software development be improved by using historical data?*

These results show that the expert estimation was not improved by the use of historical data when the expert had some work experience, and his level of experience in the technologies used to develop the application, and his level of experience in its domain were medium-low.

However, we found out that historical data may improve the expert estimation when the estimator's work experience, his level of experience in the technologies used to develop the application, and his level of experience in the domain of the application to be developed is low.

4) Discussion

There are some aspects that have not been mentioned yet, but it is worth doing so now. One of them is the little experience in Agile development contexts that the two groups had. We think that this fact did not affect the results obtained because, as the work experience of the undergraduate group was small, their experience in Agile contexts was small too. On the other hand, practitioners were experienced in project management and estimations, so this compensated for their little experience in Agile contexts. On top, as the empirical study was designed to only use the first sprint of a software product development, no estimations were made for the rest of the sprints -which would be usually done when using an Agile method- so their little experience in Agile contexts had no impact on our study.

Another interesting aspect is that most of the effort calculations proved to be underestimated, which may be seen in Figure 1. This could be explained by the fact that almost all the participants did not have previous experience with the Ruby language. On the contrary, the group that developed the target application had previous knowledge of the velocity that they could achieve because they had done a Ruby on Rails tutorial before. Consequently, the level of experience of this group in the technologies used to develop the target application and the level of experience in the target application domain was medium-high, which justifies the accuracy of the estimation: 3% MRE, which was high. At the same time the group that developed the target application had a higher velocity than the group that developed the historical application. Obviously, the bigger the difference in the velocity, the bigger the error in the effort estimation.

One question that may arise is: how would the participants be able to make meaningfully expert estimations if they did not have any knowledge about the developers? This condition was part of the scenario that we were simulating; as it was stated in the introduction of this paper, the team velocity was unknown.

Figure 2 shows that the medians obtained by the two groups when estimating with HP were similar, but their standard deviations were not: the standard deviation of the MRE for the undergraduate group was 53.7 and 75 for the practitioners. This is a consequence of the subjectivity introduced by the COSMIC measurement of both the historical user stories and the user stories to be estimated. The estimation was affected by the subjectivity of the measurements and by the difference between the historical productivity of P1 and the actual productivity of P2.

Figure 2 shows that the MRE medians obtained when the two groups used the AbM method were similar but their MRE distributions were quite different. It was surprising to see that the results obtained by the practitioners using the AbM were worse than those obtained by the undergraduates. As the AbM is based on the selection of a "similar" user story, we may conclude that the undergraduate participants had a comparable concept of "similarity" to that of the original undergraduate group that developed the target application.

The estimation results obtained with the AbM and HP method would have been better if the historical data had been obtained from a similar project –one developed using Ruby on Rails- , but unfortunately, there was none available. Besides, the fact that the user stories that were not functional processes were discarded may have also influenced the results. In addition, another interesting factor that may have been considered is team size.

In our study, the empirical objects were two similar applications, but what would have happened if they had

 ISBN: 978-1-61208-304-9

not been similar? Obviously, the results of the undergraduate group would have been affected, as their best results were obtained using the AbM. The reason is that such method is based on analogy, so if the degree of similarity between the application from where the historical data was to be obtained and that of the target application had been low, the accuracy of the estimation would have been poor too.

Moreover, although we only used the estimates of the first sprint of the target application this time, we believe the estimates of the following sprints could be used in future replications to evaluate if (and to what extent) expert estimations improve while participants gain knowledge of the projects (while AbM and HP are expected to yield constant accuracy throughout the sprints).

Finally, we may wonder about the participants' characteristics included in Table 3 and the reason why other characteristics were not included. To begin with, database experience is related to work experience, so it was necessary to check it because the COSMIC measurement would have been affected if experience in database had been small. In fact, the experience in using COSMIC was defined as a controlled variable. Moreover, the number of user stories the participants had written/read was included because it is related to their work experience in Agile contexts: in fact, there was a correlation between the number of user stories read/written and their experience in Agile contexts, which proved the consistency of the information. In addition, the level of experience with Rugby language and the level of experience in the technologies to be used had to be tested in order to verify if the participants fit our empirical study. Besides, the impact of the level of experience in the application domain was previously analyzed by [25]. We think that these characteristics have made the main differences between the two groups clear.

V. RELATED WORK

Apparently, this has been the first article to have been written about whether using historical data in an agile context improves expert estimation.

However, regarding expert estimation in general, there are some authors that have already reported evidence about the importance of the developers' level of maturity when evaluating the accuracy of estimations, which is in line with the conclusions of our study. For example, SCRUM pioneers believe it is acceptable to have an average error rate of 20% in their results when using the Planning Poker estimation technique, but they have admitted that this percentage depends on the level of maturity of the developers [25]. Another study [26] agrees with this statement, as it indicates that the optimism bias which is caused by the group discussion diminishes or even disappears as the expertise of the people involved in the group estimation process increases.

On the other hand, another study [27] has already examined the impact of the lack of experience of the estimators in the domain problem, as well as that in the technologies used in a software development project. In fact, what was studied was the accuracy with which the effort of a given task was estimated. Such estimation was performed by a single expert by comparing the estimated and the actual efforts. The reason for researching on this aspect is that, occasionally, organizations do not have in their staff experts that have relevant prior experience in some business or technology related aspect of the project they are working on. This research investigates the impact of such incomplete expertise on the reliability of estimates.

It is important to note that Jorgensen [1] has both defined a list of twelve "best practices", that is to say, empirically validated expert estimation principles, and suggested how to implement these guidelines in organizations. One of the best practices he proposed is to use documented data from previous development tasks and another one is to employ estimation experts with a relevant domain background and good estimation records. Actually, our article headed in the same direction; we focused on historical data and we analyzed the impact of the difference in experts' skills.

An aspect that should be taken into account when performing expert estimations is excessive optimism, as it is one of the negative effects that influences the most when a software project fails. Jørgensen and Halkjelsvik [28] have discovered something that seems to be important to understand what may be leading estimators to excessive optimism: the format used to word the question that asks about effort estimation. The usual way to ask about effort estimation would be: "How many hours will be used to complete task X?". However, there are people who would say: "How many tasks could be completed in Y hours?". Theoretically, the same results should be obtained by using any of the two formats. Nevertheless, according to Jørgensen and Gruschke [29], when the second option is used, the estimations which are thus obtained are much lower than those obtained when the traditional format is used, that is to say, the time to fulfill a task will be shorter, and consequently, the estimation will be much more optimistic. Thus, in our study, the expert estimations were made using the usual question. In fact, the final recommendation of this study is that the traditional format should always be used, as this does not contain any deviation imposed by the clients who ask the developers for more than they can pay for.

VI. CONCLUSION AND FUTURE WORK

This paper specifically focuses on an agile context in which the project domain and the technological environments are new to the estimators, the teams have recently been created, and the team velocity is unknown. As under these circumstances historical data may become important, we tried to answer the following research question: *when may the accuracy of an expert estimation made in a context of agile software development be improved by using historical data?* To find out whether there is any advantage in using historical data when the historical velocity is unknown, an empirical study was developed in an Agile software development context.

Historical data seems to be valuable when the work experience, the level of experience in the technologies to be used to develop an application, and the level of

experience in the domain of the application to be developed are low.

So, for estimators who have the restrictions described above, and who have no option but to work with them, we may suggest the following:

- Use intuitive expert estimations when your work experience, your level of experience in the technologies to be used to develop the application, and your level of experience in the domain of the application to be developed are not low.
- Use historical data when your work experience, your level of experience in the technologies to be used to develop the application, and your level of experience in the domain of the application to be developed are low.

In order to generalize this conclusion, a replication of this empirical study is recommended, especially if different software life cycle models [30], application domains, expert profiles, and levels of performance are included. Also, different estimation methods, such us linear regression or Analogies –next time, using the size characteristic- may be used. Finally, in order to enrich this empirical study, it would also be convenient to compare the estimation performed by an expert who has deep knowledge of this domain, and also knows the team velocity, to the estimations obtained by the participants of our study.

AKNOWLEDGMENTS

Our thanks to the Research Fund of Austral University, which made this study possible, and to Luigi Lavazza for his opportune comments.

REFERENCES

[1] M. Jorgensen, "A review of studies on Expert estimation of software development effort," Journal on System and Software, Vol. 70, No. 1-2, 2004, pp. 37-60.

[2] M. Jorgensen, and Shepperd, "A systematic review of software development cost estimation studies," IEEE Transactions on Software Engineering, Vol. 33, No. 1, January 2007, 2007, pp. 3-53.

[3] M. Cohn, Agile Estimating and Planning. Addison-Wesley, 2005.

[4] M. Jørgensen, U. Indahl, and D. Sjøberg, "Software effort estimation by analogy and regression toward the mean," Journal of Systems and Software, 68(3), 2003, pp. 253-262.

[5] T.J.Bang, "An Agile approach to requirement specification," Agile Processes in Software Engineering and Extreme Programming, SE:35, VL:4536, Lecture Notes in Computer Science, G. Concas,E. Damiani, M. Scotto, G.Succi, Eds., Springer Berlin Heidelberg, 2007, pp. 193-197.

[6] J. Choudhari and U. Suman, "Phase wise effort estimation for software maintenance: an extended SMEEM model," in Proceedings of the CUBE International Information Technology Conference, ACM, 2012, pp. 397-402,

[7] N.C. Haugen, "An empirical study of using Planning Poker for user Story estimation," Proceedings of AGILE 2006 Conference, Computer Society, IEEE, 2006, 9 pp. – 34.

[8] J. Grenning,. "Planning Poker or how to avoid analysis paralysis while release planning," 2002, DOI=, http://sewiki.iai.uni-bonn.de/_media/teaching/labs/xp/2005a/doc.planningpoker-v1.pdf: August, 2013.

[9] E. Miranda. "Improving Subjective estimates using paired comparisons," IEEE Software, 18(1), 2001, pp. 87–91.

[10] T. Saaty, Multicriteria decision making: the Analytic Hierarchy Process. RWS Publications, 1996.

[11] B. Boehm. Software Engineering Economics. Prentice Hall. 1981.

[12] A.J.C. Cowderoy and J.O. Jenkins, "Cost estimation by analogy as a good management practice," in Proc. Software Engineering 88. Liverpool: IEE/BCS, 1988, pp. 80-84,

[13] M. Shepperd, and C. Schofield, "Estimating Software Project Effort Using Analogies," IEEE Trans. on Software Eng., vol. 23, no. 11, 1997, pp. 736-743.

[14] G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd. Experiences using Case-Based Reasoning to predict software project effort. Empirical Software Engineering Research Group Department of Computing Bournemouth University Talbot Campus Poole, BH12 5BB, UK. 2000.

[15] COSMIC – Common Software Measurement International Consortium, 2009, The COSMIC Functional Size Measurement Method - version 3.0.1. Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003).

[16] ISO (2011) ISO/IEC19761:2011, Software Engineering -- COSMICFFP– A Functional Size Measurement Method, ISO and IEC.

[17] J. Desharnais, L. Buglione, and B. Kocatürk,. "Using the COSMIC method to estimate Agile user stories," *in* Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, ACM, New York, 2011, pp. 68-73.

[18] N. Juristo and A.M. Moreno, Basics of Software Engineering Experimentation. Kluwer Academic Publishers., 2001.

[19] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslen, Experimentation in Software Engineering: an Introduction. Kluwer Academic Publisher, 2000.

[20] A. Jedlitschka, M. Ciolkowoski and D. Pfahl, "Reporting experiments in Software Engineering," in Guide to Advanced Empirical Software Engineering, Section II, 2008, pp. 201-228.

[21] Ruby on Rails. http://rubyonrails.org/: August, 2013.

[22] Scrumy, http://www.scrumy.com: August, 2013

[23] K. Bittener and I. Spence. Use case Modeling. Addison Wesley, 2003.

[24] K. Molokken-Ostvold, N.C. Haugen and Benestad, H.C., "Using planning poker for combining Expert estimates in software projects," Journal of Systems and Software, 81 (12), 2008, pp. 2106-2117.

[25] O. Ktata, and G. Lévesque, "Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?," in Proceedings of the Third C* Conference on Computer Science and Software Engineering (C3S2E '10), ACM, New York, NY, USA, 2010, pp. 101-107.

[26] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," J. Syst. Softw. 85, 9 (September), 2012, pp. 2086-2095.

[27] S. Halstead, R. Ortiz, M. Córdova, and M. Seguí, "The impact of lack in domain or technology experience on the accuracy of Expert effort estimates in software projects," in Proceedings of the 13th international conference on Product-Focused Software Process Improvement (PROFES'12), Springer-Verlag, Berlin, Heidelberg, 2012, pp. 248-259.

[28] M. Jorgensen, and T. Halkjelsvik, "The effects of request formats on judgment-based effort estimation," Journal of Systems and Software, 83 (1), 2010, pp. 29-36.

[29] M. Jorgensen and M. Gruschke, "The Impact of lessons-learned sessions on effort estimation and uncertainty assessments," IEEE Transactions on Software Engineering, Jan. IEEE computer Society Digital Library. IEEE Computer Society, 2009, pp. 368 - 383.

[30] A. M Davis, E. H. Bersoff and E. R. Comer, "A strategy for comparing alternative software development life cycle models", Software Engineering, IEEE Transactions on (Volume: 14 , Issue: 10), 1988, pp. 1453 – 1461.

 ISBN: 978-1-61208-304-9