When May the Accuracy of Expert Estimation Be Improved by Using Historical Data?

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*— As expert estimation is the estimation strategy most frequently applied to software projects today, it is important to focus the research on effort estimation methods on it. This is the estimation method used in Agile contexts so we are interested in deeply understanding the value of historical data in this context, in particular when the project domains and the technological environments are new to the team, and when the teams -with little experience in Agile contexts- have recently been created. We designed an empirical study in order to find out when the accuracy of expert estimation made in a context of agile software development may be improved by using historical data. Our empirical study has shown that the use of historical data may improve the intuitive expert estimation method under the following circumstances: 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. However, the estimations thus obtained are far from being as accurate as desirable. If we expect to improve estimation accuracy, further research should be carried out in order to understand how the estimation process works. At present, we are particularly interested in expert estimation in Agile contexts, so we would like to learn if historical data may bring any improvement to expert estimation. To pursue this objective, we are now extending a paper we wrote last year [1], which was presented at ICSEA 2013, in order to add more evidence in favor of using expert estimation. This new paper will also confirm the evidence reported by other authors in [2].

Having decided on our goal, we found out that the compilation of information about cost estimation made by Jørgensen and Shepperd [3] in 2007 was extremely valuable, since they systematically reviewed papers 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 conclusive evidence.

Consequently, they suggested the following improvements in the field of research:

- *Deepen the study of the basic aspects of software estimation.* Jørgensen and Shepperd focused on two basic aspects: the evaluation of the accuracy of an estimation method and the appropriate selection of an estimation method.
- *Widen the research on the current, most commonly used estimation methods in the software industry.* The leading estimation method today is that based on expert opinion (ranging from analogies to experiences and intuition), but research on expert estimation is still scarce.
- *Perform studies which support the estimation method based on expert judgment, instead of replacing it with other estimation methods.* Given the fact that expert judgment is the most widely used method in the software industry today, it would be convenient to improve such judgment by supporting it with the use of formal estimation methods.
- *Apply cost estimation methods to real situations.* There are few studies in which the estimation methods are evaluated in real situations, since most of such methods are applied to laboratory, non realistic contexts.

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 [4] 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 [3], 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.

It is important to note that this study does not take into consideration the effect of non-functional user requirements on effort; hence it will not address the estimation of effort which is necessary to satisfy non-functional requirements.

In this scenario, we have 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, both those in [1] and the ones in this new paper, in which we have included a different way of applying one of the estimation methods reported in [1] and more detailed results, 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 the following Section, we will investigate related work to see if there is any other evidence of improvement in expert estimation accuracy when using historical data. In Section III, we will introduce three estimation methods: Expert Estimation (ExE), Analogy-Based Method (AbM), and Historical Productivity (HP). In Section IV, we will describe an empirical study and in Section V we will analyze the results obtained. Moreover, the threats to validity will be discussed in Section VI and finally, in Section VII, we will draw conclusions about the evidence which shows the benefits of using historical data.

II. 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, if we consider 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 [5]. Another study [6] 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 [7] 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 sometimes 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 [2] has both defined a list of twelve "best practices", that is to say, empirically validated expert estimation principles, and also 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 goes in the same direction; we have focused on historical data and 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 [8] have made a discovery 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 [9], 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.

Besides the papers mentioned above, Jorgensen has written several studies that include other aspects that may affect expert estimations. Although such aspects were not taken into account in this study, we believe they may enrich our conclusions. These aspects are:

a. *high degree* of *inconsistency and an improper weighting of variables* [2], he believes that if these negative aspects could be reduced, the accuracy of the estimations would be much better.

b. *the level of interdependence (focusing on relations, social context and interconnections) introduces a deviation in the estimation process* [9], according to Jørgensen, the estimations performed by software developers are also affected by human relationships. Besides, he points out that such deviations take place under every circumstance.

III. 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 [2] 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: firstly, the estimation work must be conducted by a person who is considered an expert in the task, and secondly, 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 [11]. 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 [10].

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-, which is a variation of the Delphi method. Planning Poker 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 [4][11][12]. This method was first described by James Greening [14] and it was then popularized by Mike Cohn through his book "Agile Estimating and Planning" [4]. It is mainly used in agile software development, especially in Extreme Programming [13]. 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 [4]. 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 [14]. The basic steps of this technique, according to how Grenning described it, 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 all estimates are equal, 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" [14].

Mike Cohn further developed this technique. He added a pack of cards especially designed to apply it. Each pack has to be prepared before the Planning Poker meeting and it will contain cards with numbers written on them. Such numbers should be big enough to be read from the other side of a table. Those numbers represent a valid estimation, such as 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100. There is a raison d'être for such an estimation scale: there are studies which have demonstrated that we are better at estimating things which fall within one order of magnitude [15][16]. Planning Poker as has been here defined was used in the empirical study reported in this article. It should be noted that no historical data was used when Planning Poker was employed in 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 [17] suggested the informal use of analogies as a possible technique thirty years ago. In 1988, Cowderoy and Jenkins [18] also worked with analogies, but they did not find a formal mechanism to select the analogies. According to Shepperd and Schofield [19], 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 [19].

Such similarities may be defined in different ways, but most of the researchers define the measuring of similarities the way Shepperd & Schofield [19] and Kadoda, Cartwright, Chen & Shepperd [20] 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 estimated effort by applying AbM in two distinct ways: our first approach was to take into account the characteristics of each user story in a general manner (AbM-1), and our second approach was to use only one characteristic of the n characteristics which could be used. In this case in particular, such characteristic was size (AbM-2s).

When using AbM-1 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.

When using AbM-2s, the project characteristic which was taken into account was the size of the project measured using COSMIC [21]. The EE of the user story of the target project was the AE of the closest historical user story –i.e., the user story whose size distance was the smallest ($|\text{UserStorySize}_t - \text{UserStorySize}_h|$**)**. When multiple user stories were at the minimum distance, the participants calculated the mean of the AE of these user stories.

C. Historical Productivity

Jørgensen, Indahl, and Sjøberg [10] 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 [21] was used as a measure of Size, and EE was calculated as the product of Size and Historical Productivity (HP). HP was the productivity of the project which was 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 [22] that is widely recognized in the software industry, and also because there is a previous study that used it in an Agile context [23]. With the COSMIC software method, Functional User Requirements -possibly represented via user stories- can be mapped into unique functional processes. 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:

- *Entry* moves a data group from a functional user into the software
- *Exit* moves a data group out of the software to a functional user
- *Read* moves a data group from persistent storage to the software
- *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. Moreover, the size of a software application is given by the sum of the sizes of its functional processes, and the size of the functional processes is given by the sum of Entries, Exits, Reads and Writes, where each term in the sum indicates the number of corresponding data movements, expressed in CFP. So, the concept of "weighting" a data movement does not exist in COSMIC; in other words, all data movements weigh the same.

IV. Definition and planning of our Empirical Study

Our empirical study is described in this section, considering its conception and how it was planned.

A. Definition

This empirical study was designed in order to establish when the accuracy of an expert estimation made in an agile development context, 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 sprint of an application –the "target" application, i.e., P2- which had been previously developed by a group of undergraduate students who did not participate in the workshop. Both the development language and the application domain were unfamiliar to participants. Initially, participants had no idea of the developing team's velocity.

The re-estimations were performed by using four different estimation methods: ExE, based on the participants' intuition, and three other methods which use historical data. The historical data was obtained from an application which was similar to the target application, which had been developed by a third undergraduate group – a group that had neither developed the original application nor participated in our empirical study-. Such application, P1, will be called "historical application".

To guarantee the best results, we developed this empirical study following the recommendations of Juristo and Moreno [24] and Wohlin et al. [25]. To report it, we took into account Jedlitschka, Ciolkowoski and Pfahl's guidelines for reporting empirical research in software engineering [26].

As previously stated, the objective of this empirical study was to analyze when the accuracy of an estimation made by an expert, a role played by undergraduate students and practitioners in this study, may be improved by using historical data. This objective was achieved by comparing the errors the experts obtained by estimating with a method based on "pure" intuition (ExE) to those they obtained by estimating with three different methods: AbM-1, AbM-2s and HP.

Figure 1 summarizes this definition.

The hypotheses to be tested were:

H_0: The mean value of the MRE calculated with the ExE method is equal to the mean value of the MRE obtained when calculating with AbM-1, AbM-2s or HP.

H_1: The mean value of the MRE calculated with the ExE method is lower than the mean value of the MRE obtained when calculating with AbM-1, AbM-2s 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.

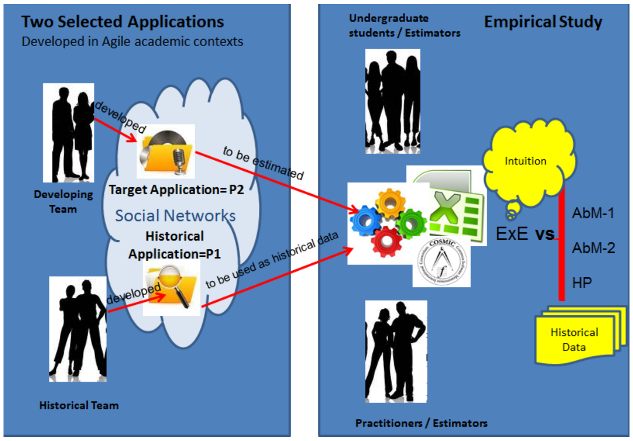
Two Selected Applications
Developed in Agile academic contexts
developed
to be estimated
Developing Team
Target Application= P2
Social Networks
Historical Application=P1
to be used as historical data
developed
Historical Team
Undergraduate students / Estimators
Empirical Study
Intuition
ExE vs
AbM-1
AbM-2
HP
Historical Data
Practitioners / Estimators

Fig. 1 Summary of the empirical study.

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 [27] 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), namely social networks, which had been developed by two groups of students before the empirical study was designed. For the sake of clarity we will say that P2 was developed by the "developing team" and P1 by the "historical team", as shown in Figure 1. It is important to note that these two groups did not participate in our empirical study; in fact, they were undergraduate students from a university different from the one where the undergraduate participants of the study were studying. P1 and P2 were developed to fulfill an assignment in a certain course. Both teams used Agile methodology to do so. They were instructed to register the hours worked per user story using the Scrumy tool [28]. Two professors supervised all of these tasks.

Application P1 is 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.

Application P2, which we have identified as the "target" project, is 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 I shows the user stories of P1 and the Actual Effort (AE) of each user story measured in person hours. As some user stories were not functional processes, they were discarded. Table II shows the user stories and the AE of P2. This empirical study used the actual effort of P1 and P2. These are the user stories of only the first sprint, as it was the only sprint for which effort was estimated.

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, person-hours had to be used to unify the unit of measurement of effort, since the historical values had been previously measured in person-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 [4].

The workshop was run following these steps:

1) The participants were given a set of materials that included: Brief Vision Documents [29] 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.

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 person-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 II.

b) Build the historical database. Each team created its own historical dataset by measuring the size of the user stories of the historical application (P1), using COSMIC, as shown in Table I.

TABLE I. USER STORIES OF THE HISTORICAL APPLICATION (P1)

User stories	Actual Effort [person-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)

User stories First Sprint	Actual Effort [person-hours]
Create, Modify and Eliminate User	8
Log in (Log out)	18
Create event	6
Search for event	3
Total	35

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. So, the HP_{P1} is calculated at application level, which will be used to automatically calculate the EE_{P2}.

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 [23].

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 $EE_{P2,}$ which was calculated as the product of $Size_{P2}$ and Historical Productivity (HP_{P1}), which had been obtained in the previous step.

d) *Estimate the effort for the target application (P2) using AbM in two different ways: AbM-1 and AbM-2s.* For AbM-1, 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 the stories' general characteristics, not on their size or on any other specific characteristic- and then assign to the EE of each user story in P2 the AE of the similar user story in P1. For AbM-2s, the participants had to use the sizes which had been previously calculated for each one of the user stories in P2, as described in c) above, and the sizes which had been calculated for P1, described in b) above. They had to compare the size of each user story in P2 to the size of all the user stories in P1 in order to find the smallest distance between $|\text{UserStorySize}_{P2} - \text{UserStorySize}_{P1}|$. Once the smallest distance had been found, the AE of the user story in P1 which was the closest to P2 was assigned to the EE of such user story in P2. In those cases in which the participants found that the smallest distance was repeated, they assigned as EE of P2 the mean value of AE of the user stories in P1 which shared the same distance values.

e) *Individually compare and analyze the EE values obtained using ExE, AbM-1, AbM-2s and HP methods*. The Excel sheet automatically presents a table that displays the four EE values –those obtained by applying the four 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 four estimation methods (ExE, HP, AbM-1 and AbM-2s) was conducted by the leader of the empirical study.

Figure 2 summarizes the steps of the empirical study.

C. Execution

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 attended the course "Database" and passed the exam 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.

The characteristics of the participants are described in Table III.

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.

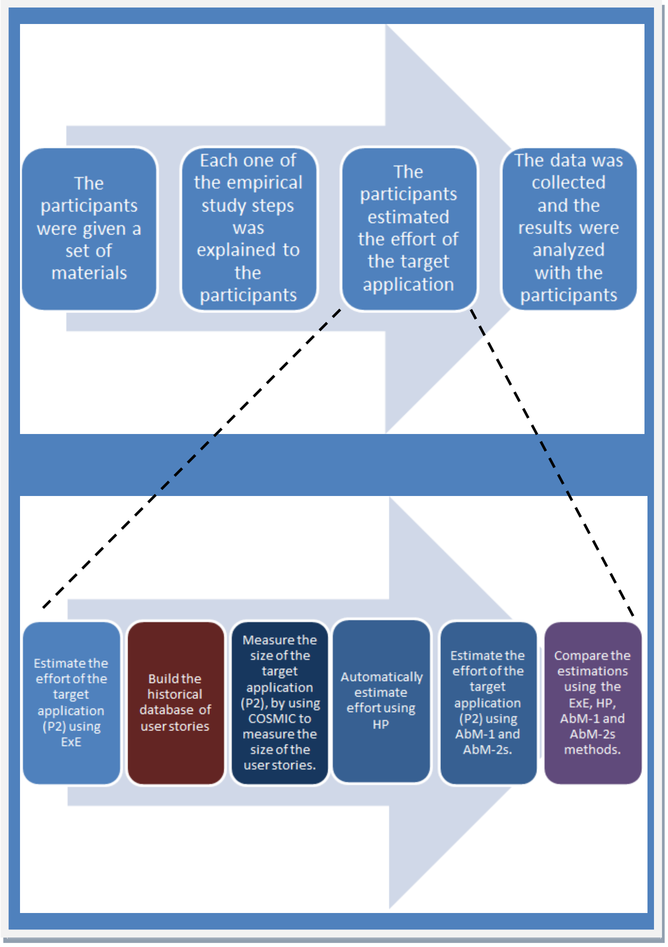
The participants were given a set of materials
Each one of the empirical study steps was explained to the participants
The participants estimated the effort of the target application
The data was collected and the results were analyzed with the participants
Estimate the effort of the target application (P2) using ExE
Build the historical database of user stories
Measure the size of the target application (P2), by using COSMIC to measure the size of the user stories.
Automatically estimate effort using HP
Estimate the effort of the target application (P2) using AbM-1 and AbM-2s.
Compare the estimations using the ExE, HP, AbM-1 and AbM-2s methods.

Fig. 2 Steps of the empirical study.

In one of the workshops, there were fourteen practitioners worked on their own and their median work experience was fourteen years. No one had experience in using COSMIC, and five of them had experience with user stories. Their median work experience with databases 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, that is, not definitely low, but it could not be classified as fully medium.

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

V. RESULTS

Before answering the research question posed above, it is important to understand the circumstances under which

TABLE III. WORKSHOP PARTICIPANTS

Type	Number	Work Expe-rience (Years)	Number of people familiar with COSMIC	Number of User Stories [<20, 20<US<100, >100]	Work experience with Database	Number of people familiar with Ruby Language	Number of people familiar with 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 of them had attended the course “Database” and passed the exam	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 use of historical data may improve expert estimation accuracy. To do so, this section will first describe the results obtained by the two types of participants -undergraduates and practitioners- and then analyze them. Afterwards, the statistical significance of such results will be dealt with, and later on, the research question will be answered. Finally, this analysis will be completed with the discussion of aspects omitted in the previous sections.

1) Description and analysis

Table IV shows the effort estimation values calculated for the target project, obtained by the two groups applying the four estimations methods: ExE, HP, AbM-1 and AbM-2s. Moreover, the AE of the target application (P2), which was developed by the undergraduate group was 35 person-hours.

Figure 3 shows the boxplots of the residuals and Figure 4 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. Also, Table V shows the statistical functions of residuals and the MRE.

The boxsplots show the different results obtained by each group of participants. The undergraduate participants obtained better estimation results when applying the AbM-1 or AbM-2s, rather than the ExE or HP methods. Figure 4 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-1: 0.70, AbM-2s: 0.71, ExE:1.51 and HP:1.75. On the other hand, the practitioners’ group obtained the best results when applying ExE, instead of HP or AbM, as shown by the boxplots. Also, their mean values were ExE: 0.38, HP: 2.05, AbM-1: 0.87 and AbM-2s: 0.60.

The best result of the undergraduate group was obtained when using AbM: the MRE median of AbM-1 was 0.63 within the [0.37-1.83] range and AbM-2s was 0.62 within [0.13-1.14]. 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 [0.99-2.72]. The MRE of the 14 groups had a median of 1.89 and a standard deviation of 0.54. Moreover, by using ExE we obtained a higher dispersion: MRE values ranged from [0.03-4.91], with a median of 0.90 and a standard deviation of 1.55.

Both the practitioners’ level of experience in the technologies used to develop the experimental objects and their level of experience in the domain of the experimental objects were medium-low. These characteristics justify the results obtained when using ExE: the median of the MRE was 0.29 in a [0.14-0.83] range of values.

During the study, three of the practitioners assigned to the expert estimation the same value they had assigned to the AbM-1 estimation. This may have been a coincidence, or they may not have felt confident to perform an estimation based on their intuition.

Eleven out of fourteen practitioners obtained MRE less than 0.25 via ExE. The estimation by AbM-1 had a MRE median of 0.70 in a range result of [0.09-2.00] and that by AbM-2s obtained 0.51 in a range of [0.06-1.37], which are results similar to those obtained by the undergraduates. Moreover, the subtle differences between the MRE medians and the standard deviations of AbM-1 and AbM-2s may be justified by the fact that AbM-1 is based on subjective criteria, while AbM-2s is based on the concept of size. In fact, the practitioners had worked in very different contexts, which naturally affected their subjective comparisons.

TABLE IV. EE OF THE TARGET PROJECT

Participants	Number of estimations	Id Participants	ExE	HP	AbM-1	AbM-2s
Undergraduates	14 (made by groups of 3-4 undergraduate students)	1	161.00	110.00	57.00	-
		2	61.00	74.70	57.00	48.75
		3	34.00	76.30	60.00	75.00
		4	65.00	69.72	48.00	-
		5	207.00	84.12	48.00	-
		6	85.00	106.13	55.00	63.17
		7	173.00	90.21	66.00	52.03
		8	68.00	102.84	57.00	52.50
		9	79.00	101.15	57.00	56.79
		10	56.00	101.44	51.00	51.40
		11	51.00	72.00	57.00	72.00
		12	32.00	130.15	57.00	39.60
		13	105.00	108.93	99.00	73.83
		14	23	120.05	63	71.50
Practitioners	14	15	11.00	108.94	11.00	59.60
		16	30.00	173.22	24.00	45.00
		17	21.00	84.77	20.00	51.30
		18	30.00	122.15	60.00	60.73
		19	9.00	90.55	9.00	47.67
		20	64.00	85.96	39.00	57.00
		21	30.00	120.61	105.00	38.00
		22	29.00	111.87	86.00	82.80
		23	16.00	72.88	32.00	68.00
		24	30.00	105.05	95.00	52.75
		25	40.00	88.93	57.00	73.88
		26	40.00	97.07	94.00	52.50
		27	49.00	92.37	70.00	-
		28	57.00	140.94	57.00	37.00

TABLE V. STATISTICAL FUNCTIONS OF RESIDUALS AND MRE

Participants	Statistical Functions	Residuals				MRE			
		ExE	*HP*	*AbM-1*	*AbM-2s*	*ExE*	*HP*	*AbM-1*	*AbM-2s*
Undergraduate	Mean	-50.71	-61.27	-24.43	-24.69	1.51	1.75	0.70	0.71
	Median	-31.50	-66.30	-22.00	-21.79	0.90	1.89	0.63	0.62
	Standard deviation	56.40	18.80	12.43	12.03	1.55	0.54	0.36	0.34
Practitioners	Mean	2.43	-71.81	-19.21	-20.86	0.38	2.05	0.87	0.60
	Median	5.00	-66.06	-22.00	-17.75	0.29	1.89	0.70	0.51
	Standard deviation	16.25	26.30	32.65	13.35	0.26	0.75	0.61	0.38

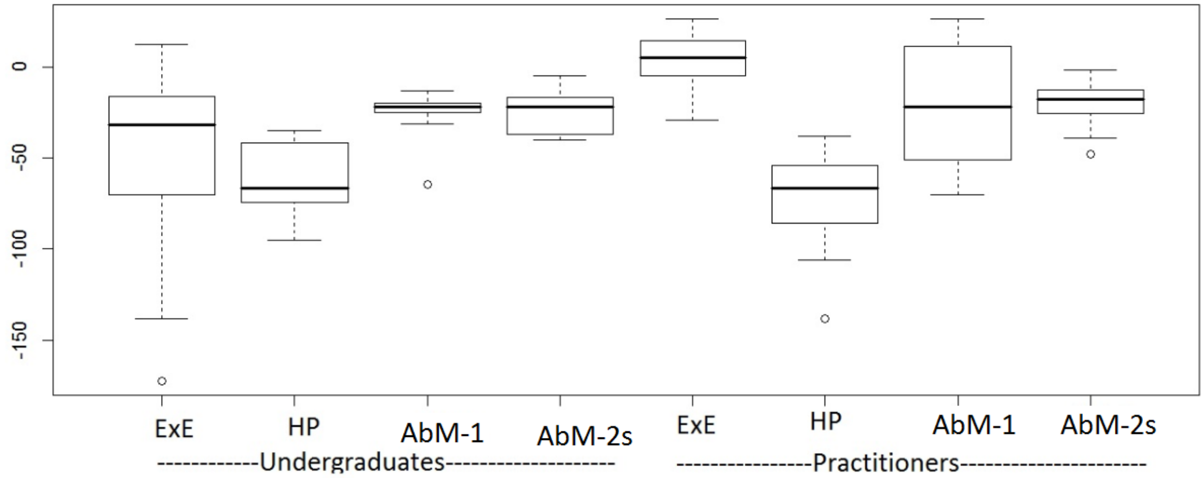

Fig. 3. Boxplots of the residuals of the target project.

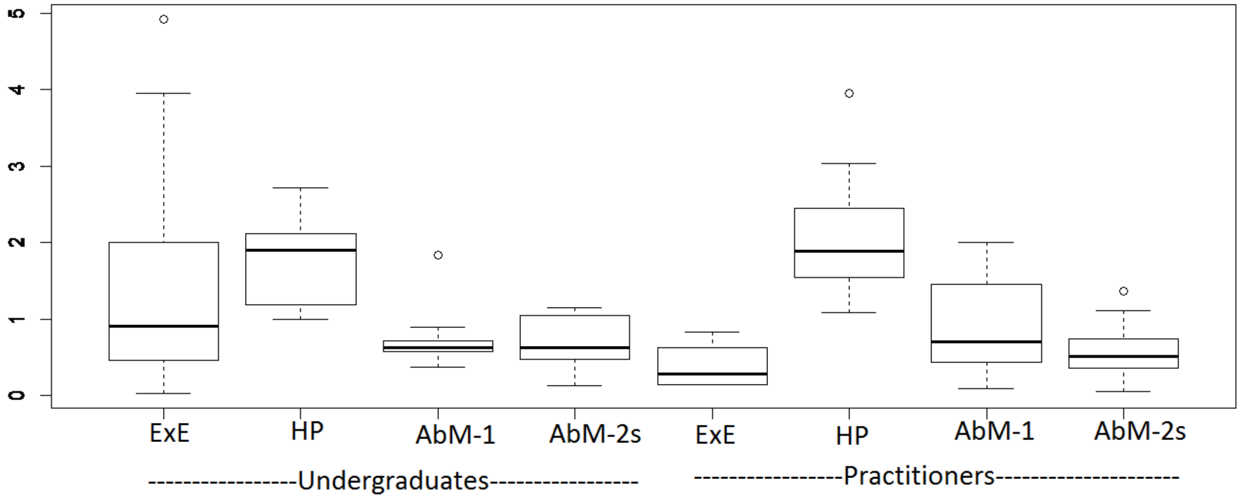

Fig. 4 Boxplots of the MRE of the target project.

By using HP, the MRE dispersion was increased: [1.08-3.85]. The MRE of the 14 practitioners had a median of 1.89 -similar to that of the undergraduate value-and a big standard deviation of 0.75, which may have been caused by the difference in productivity between P1 and P2.

2) Statistical significance

In order to test the hypotheses presented in Section III, the Wilcoxon rank test, at a significance level of 0.05, was used to analyze the statistical significance of our 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, AbM-1 or AbM-2s, according to the results obtained by each group (practitioners and undergraduate participants). The MRE and the absolute residuals were used. Table VI shows the p-value of each subset, when using the MRE. The results obtained when using the absolute residuals are not shown because they presented no significant difference. 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-1, once again, it was possible to reject H0 in favor of H1.
- the practitioners, when comparing the ExE method to AbM-2s, it was not 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-1 and AbM-2s, it was not possible to reject H0 in favor of H1.

TABLE VI. STATISTICAL SIGNIFICANCE

Groups	ExE vs:	p-value
Undergraduate	HP	0.162
	AbM-1	0.948
	AbM-2s	0.793
Practitioners	HP	0.000
	AbM-1	0.022
	AbM-2s	0.083

The statistical significances obtained by the practitioners using AbM-1 and AbM-2s are similar, but it was not possible to reject H0 in favor of AbM-2s, although we did reject it for AbM-1. The main reason is the differences in the distributions between AbM-2s and AbM-1, as shown in Figure 4. Besides, the calculation may have been affected by the lower number of available instances.

It should be noticed that the EE values reported by the three practitioners who presented the same values when using ExE and AbM-1 were also included in the table. However, later on, when the Wilcoxon rank test was run, we only considered the values reported by the other eleven practitioners, 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 together with his level of experience in its domain were medium-low.

However, we have found out that historical data may improve 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.

1) Discussion

There are some aspects that have not been mentioned yet, but we believe they are worth being discussed at this point. 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, although the work experience of the undergraduate group was limited, so was their experience in Agile contexts. On the other hand, the fact that practitioners were experienced in project management and estimations compensated for their little experience in Agile contexts. Furthermore, 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 3. This could be explained by the fact that almost all the participants did not have previous experience with the Ruby language.

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 stated in the introduction of this paper, the team velocity would be unknown.

Figure 4 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 0.54 and 0.75 for the practitioners. The estimation was affected by the subjectivity of the measurement which may be explained by the differences between means and median, for the two groups: a. undergraduate: mean: 62, median:62 and b. practitioners: mean:56, median:53. In Table VII and VIII the measurements made by each group of participants are reported. It is important to note that the standard deviations are quite similar: 12.25 and 11.97.

TABLE VII. MEASUREMENTS MADE BY UNDERGRADUATES USING COSMIC

Id User Story	2	3	6	7	8	9	10	11	12	13	14
1	10	4	6	5	4	4	5	6	3	7	4
2	7	5	4	5	3	4	4	5	3	4	5
3	2	3	2	3	2	3	3	3	2	3	2
4	11	3	4	6	4	4	4	5	3	3	3
5	3	4	3	5	4	5	5	5	3	3	4
6	5	3	4	4	2	3	4	4	2	3	3
7	3	4	3	4	3	4	3	5	3	4	4
8	8	5	4	5	2	5	3	4	3	3	4
9	4	5	4	5	4	6	4	6	4	3	4
10	3	4	3	4	3	3	3	3	3	4	4
11	7	6	5	5	3	6	4	11	3	8	3
12	4	4	4	5	2	4	3	5	2	3	5
13	4	4	4	4	3	4	2	3	2	3	3
14	3	1	3	3	3	2	3	5	3	3	3
15	4	5	4	4	2	4	2	5	3	4	3
16	2	2	4	2	2	3	2	8	2	6	2

TABLE VIII. MEASUREMENTS MADE BY PRACTITIONERS USING COSMIC

Id User Story	14	15	16	17	18	19	20	21	22	23	24	25	26	28
1	4	3	8	3	3	2	8	4	4	3	4	4	3	2
2	5	3	1	3	3	2	4	4	3	2	4	4	4	3
3	2	3	2	3	3	2	4	3	3	3	3	3	3	3
4	3	2	2	2	4	2	5	5	5	3	6	3	5	4
5	4	4	4	5	3	5	6	5	5	2	5	4	3	5
6	3	2	2	2	3	2	7	5	4	2	5	2	4	4
7	4	3	3	3	3	3	5	3	4	2	3	3	3	2
8	4	4	4	2	3	2	7	5	4	2	5	3	5	4
9	4	3	2	4	6	3	4	5	5	3	5	4	5	4
10	4	3	2	3	3	3	3	4	4	3	4	3	4	3
11	3	3	2	3	3	3	9	3	6	3	3	8	3	3
12	5	4	2	2	3	3	4	4	4	2	4	4	4	4
13	3	3	3	3	3	3	5	4	5	3	4	3	3	3
14	3	3	3	3	3	3	3	3	5	3	3	3	3	3
15	3	2	3	3	3	3	6	5	4	2	4	2	5	4
16	2	3	3	3	4	3	4	2	4	3	2	3	2	2

Some of the measurements (1, 4, 5, 27) are not reported because they are not available.

Figure 4 shows that the MRE medians obtained when the two groups used the AbM-1 method were similar but their standard deviations of MRE were quite different. The practitioners' standard deviation was bigger than the undergraduates' standard deviation. This may be a consequence of the variety of persons that made up the practitioners' group: when they had to select the "most similar" user story, they applied their own criteria, based on their different work experiences, which were definitely subjective.

On the other hand, the undergraduates and practitioners got similar distributions with AbM-2s. The reason may be that the two groups used an objective measure of size: COSMIC.

It was not surprising that the results obtained with HP were clearly worse than those obtained with AbM. This was expected, since HP is a method that estimates at application level, while AbM estimates at user story level.

The estimation results obtained with the AbM and HP methods 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 not been similar? Obviously, the results of the undergraduate group would have been affected, as their best results were obtained using 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 III 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. Furthermore, the impact of the level of experience in the application domain was previously analyzed by [31]. We think that these characteristics have made the main differences between the two groups clear.

VI.THREATS TO VALIDITY

The difference in the 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.

The productivity rate of academic developments is usually quite different from the one of professional settings. This fact has obviously affected the results obtained, but as it is reflected in the error values, it does not invalidate the empirical study.

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. In spite of this difference, we think that combining expert methods, that is, using Planning Poker or not, did not introduce bias in this study, in accordance with what was reported in [30].

Unfortunately, only a brief explanation about COSMIC was given to the undergraduate students since there was not enough time to give an extensive explanation (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 serious problem, as the concept of data movement was quite intuitive for all the participants and the medians of the errors shown in both Figure 3 and 4 for the HP method are similar.

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 gave them further explanations.

The order in which the estimations were performed may 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.

When building a historical database, the selection of an application similar to the one to be estimated is clearly an advantage in order to obtain a better estimation. In this empirical study, we used as historical application one that had not been developed in the same language the application to be estimated had been. Obviously, this circumstance may have enlarged the estimation error of the method that used historical data. At the same time, as the context defined for this empirical study was one in which the project domains and the technological environments were new to the team, we interpreted that the application used as historical was well selected.

The accuracy of AE registered by the students that developed P2 was controlled by two professors. In fact, the students registered in a web application the user stories and the tasks done, the EE and the AE of each user story and the EE and AE of each task assigned to each user story. This detailed registration facilitated the control for accuracy.

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 only one 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.

VII. 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. In our study the estimations performed by two different groups –undergraduate participants and practitioners- which first used intuitive expert estimations (ExE) and then three different estimation methods which use historical data (HP, AbM-1 and AbM-2s) were compared in order to find out whether there is any advantage in using historical data under these circumstances.

We may conclude that 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.

Consequently, 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.

As historical data is not frequently available [32], we expect the results of this empirical study may motivate novice developers to give importance to collecting such data in their daily work.

In order to generalize this conclusion, a replication of this empirical study is recommended, especially if different software life cycle models [33], application domains, expert profiles, and levels of performance are included. Also, different estimation methods, such us linear regression may be used. Finally, in order to enrich this empirical study, it would also be convenient to compare an 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] G. Robiolo, S. Santos, and B. Rossi, "Expert estimation and historical data: an empirical study," in Proceedings of The Eighth International Conference on Software Engineering Advances, ICSEA 2013, October 2013, pp. 336-345.

[2] 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.

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

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

[5]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, 2010, pp. 101-107.

[6] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," J. Syst. Softw., vol. 85, no. 9, September 2012, pp. 2086-2095.

[7] 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, 2012, pp. 248-259.

[8] M. Jorgensen, and T. Halkjelsvik, "The effects of request formats on judgment-based effort estimation," Journal of Systems and Software, vol. 83, no.1, 2010, pp. 29-36.

[9] M. Jorgensen and M. Gruschke, "The Impact of lessons-learned sessions on effort estimation and uncertainty assessments," Software

Engineering, IEEE Transactions on, vol. 35, no. 3, 2009, pp. 368 - 383.

[10] M. Jørgensen, U. Indahl, and D. Sjøberg, “Software effort estimation by analogy and regression toward the mean,” Journal of Systems and Software, vol. 68, no. 3, 2003, pp. 253-262.

[11] T.J.Bang, “An Agile approach to requirement specification,” Agile Processes in Software Engineering and Extreme Programming, Springer Berlin Heidelberg, 2007, pp. 193-197.

[12] 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,

[13] 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.

[14] 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: May, 2014.

[15] E. Miranda, “Improving Subjective estimates using paired comparisons,” IEEE Software, vol. 18, no.1, 2001, pp. 87–91.

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

[17] B. Boehm, Software Engineering Economics. Prentice Hall, 1981.

[18] A.J.C. Cowderoy and J.O. Jenkins, “Cost estimation by analogy as a good management practice,” in Proc. Software Engineering 88, Second IEE/BCS Conference, 1988, pp. 80-84,

[19] M. Shepperd and C. Schofield, “Estimating software project effort using analogies,” IEEE Trans. on Software Eng., vol. 23, no. 11, 1997, pp. 736-743.

[20] G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd, “Experiences using Case-Based Reasoning to predict software project effort,” Proceedings of the EASE conference keele, UK., 2000.

[21] COSMIC – Common Software Measurement International Consortium, The COSMIC Functional Size Measurement Method - version 3.0. Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003), 2007

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

[23] 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, 2011, pp. 68-73.

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

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

[26] 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.

[27] Ruby on Rails, doi: http://rubyonrails.org/: May, 2014.

[28] Scrumy, doi: http://www.scrumy.com: May, 2014.

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

[30] K. Molokken-Ostvold, N.C. Haugen, and H.C. Benestad, “Using planning poker for combining Expert estimates in software projects,” Journal of Systems and Software, vol.81, no.12, 2008, pp. 2106-2117.

[31] M. Jorgensen, “Selection of strategies in judgment-based effort estimation,” Journal of Systems and Software, vol. 83, no. 6, 2010, pp.1039-1050.

[32] C. Mair , M. Shepperd, and M. Jørgensen, "An analysis of data sets used to train and validate cost prediction systems," ACM SIGSOFT Software Engineering Notes, ACM, vol. 30, no. 4, 2005, pp.1-6.

[33] 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, vol. 14, no.10, 1988, pp. 1453 – 1461.