Effort Estimation in Agile Software Development: The State of the Practice in Colombia

Diana Gomez[1], **Juan Cubillos**[1], **Jairo Aponte**[1], **Edwar Rojas**[1]

[1]Universidad Nacional de Colombia. Facultad de Ingeniería. Bogotá - Colombia

{dgomezcar, jmcubillosa, jhapontem, edrojasb}@unal.edu.co

Abstract. *Effort estimation is fundamental for developing software projects and critical for their success. This paper focuses on how Colombian agile practitioners perform effort estimates in agile projects. For this purpose, we conducted an exploratory survey study that involved 60 respondents. The main findings are the following: (1) Agile practitioners prefer estimation techniques based on Expert Judgement. (2) Most of the respondents perceive that their estimates have a medium accuracy level. (3) The determining effort drivers are features of the project team and the software to be built. (4) The use of datasets for estimation is not common. (5) Most of the results of related studies are similar to ours, with differences in terms of the roles involved and the techniques used.*

1. Introduction

Software Development Effort Estimation (SDEE) is the process of assessing or predicting the most realistic effort, duration, and cost of a software endeavor [1], [2]. In the software engineering domain, effort estimation and planning are two closely related activities since the first one is essential to carry out the second one [3]. Accurate and precise estimates influence several planning tasks that help complete the project properly, such as creating a detailed schedule, breaking a project down into phases, and prioritizing functionalities for delivery.

Although effort estimation has been widely studied, the iterative nature of agile methods requires specialized estimation techniques [4]. Despite the numerous approaches proposed so far, the effort estimates in Agile Software Development (ASD) are far from perfect and in most cases the tendency is to underestimate the effort required. Requirements, management, and team-related issues are cited as the top causes for large differences between the actual and estimated effort [5].

Expert Judgment techniques have been by far the most common estimation approach used in Agile Effort Estimation (AEE) to predict software development effort. Expert Judgment is the assessment based on the experience of an application area, knowledge field, discipline, or industry, for the activity being performed. Such expertise may be provided by groups or individuals with specific education, knowledge, skills, experience, or training [6]. The most widely used estimation technique in ASD is Planning Poker, which is a variation of Expert Judgment.

A recent update of a Systematic Literature Review (SLR) on effort estimation in ASD summarizes the results of primary studies published from December 2013 to January 2020. The review focuses on 73 papers that provided new evidence for characterizing estimation activities in agile projects. These results confirm that Planning Poker has become the most widely used estimation technique, and it is very closely related to the most frequently used size metric, namely Story Points (SP). Moreover, SP estimate the size and complexity of user stories, which are the main artifacts for specifying user requirements in ASD [5].

The aim of this work is to analyze the current state of the software estimation process for agile software development in Colombia. To the best of our knowledge, there are no similar studies conducted in this region on AEE. Even more important, our interest is to verify the main findings of other studies regarding the effort estimation process in ASD, specifically concerning techniques used, existing problems, and opportunities for improvement.

We collected and analyzed data from five areas of interest: (1) Techniques used to carry out effort estimation; (2) Accuracy level of effort estimation in ASD; (3) Effort predictors used in this process; (4) Datasets, i.e. a collection of information containing estimations from software projects, used to perform effort estimation activities and (5) Differences between our findings and previous studies conducted in other regions. The data collection instrument was an online survey. It allowed us to gather information related to the aforementioned areas of interest, along with the demographic information of people who play various technical and managerial roles in ASD.

The paper is organized as follows. Section 2 presents related work of software development effort estimation on agile contexts. Section 3 describes the research methodology. Section 4 presents the most significant results of the study. Section 5 discusses the principal findings. Threats to validity and conclusions are described in sections 6 and 7, respectively.

2. Related Work

There is extensive literature of effort estimation based on surveys. Most of these studies have been carried out on traditional software development. Molokken and Jorgensen reported studies of this type since 1984 [7]. As traditional software methods are still in used, there are recent publications with the same scope [8], [9]. On the contrary, the literature review indicates that there are few studies, regarding effort estimation on ASD, similar to ours.

The first of them is the research by Usman et al., which validates a previous study by the same authors [10]. In this work of 2014, they surveyed 60 agile practitioners with experience in effort estimation from 16 different countries belonging to the five continents. Results showed that Planning Poker, Analogy, and Expert Judgment are frequently used estimation techniques in ASD, while SP is the most frequently employed size metric. Moreover, the expertise level of the team and prior experience are the most commonly used cost drivers. More than half of the respondents believe that their effort estimates on average are under/overestimated by an error of 25% or more. Finally, they found that most agile teams take into account implementation and testing activities during effort estimation, and it is mostly performed at iteration and release planning levels in ASD.

The second study was carried out in 2018 by Vera et al. [11]. They conducted an empirical study that identified the most frequent approaches to effort estimation in Chilean companies. Their main findings reveal that companies use Expert Judgment mainly due to their inability to count on trustworthy historical data. This inability to record reliable information is usually a consequence of the unplanned human resource sharing performed by the small software companies. It is worth noting that this study included all kinds of software projects.

In 2019, Prokopova et al. [12] conducted a study aimed at analyzing the project estimation process in Czech companies. 91 companies, belonging to six information technology areas were surveyed. They found that the most widely used estimation technique is Expert Judgment with 57% incidence. Overall, 48% of companies report implementation time as a key factor for the estimation. The most common cause (39%) of delivery delays in software projects is the requirements volatility. In general, 54% companies failed to complete a quarter of projects in the estimated timeframe. It is important to note that this study considered both agile and traditional projects.

Regarding effort estimation accuracy levels in ASD, Ramessur et al. [13] analyzed 15 projects in a well-known Mauritanian company and uncovered eighteen factors that influence iterations effort estimation and promote better estimation accuracy levels. These factors mainly refer to stakeholders, team characteristics, code base features, requirements aspects, and managerial factors.

Finally, there are several studies about effort estimation on User Stories [14], [15], [16], [17]. Majchrzak et al. [14] propose factors influencing User Story estimation. Their results show three major factor groups: management, knowledge, and requirements issues. Additionally, there are post-release defect factors that are directly related to inaccurate estimates: demotivation, stress, missing requirements, and inappropriate tests. Abrahamsson et al. [15] propose an ASD method for predicting development effort based on well structured user stories. Mahnič and Hovelja [16] carried out an empirical study using Planning Poker for estimating user stories, whose main finding is the correlation between the expertise of estimators and the estimation accuracy. Haugen [17] shows that the introduction of Planning Poker improves the estimation performance of the team on Extreme Programming (XP) environments.

3. Research Methodology

This work is a survey study that follows the approach suggested by Linaker et al. [18] who define the fundamental activities that must be carried out in studies based on surveys, within the software engineering field.

In 2015, Usman et al. [10] conducted a survey study aimed at validating the results of a previous SLR where ASD practitioners were consulted. Following a similar approach, we identified an SLR performed by Fernandez-Diego et al. [5] that was published just before the beginning of this research. Aside from being a recent SLR, there are several circumstances that make it suitable for our survey design. First, it is an update of an SLR carried out by Usman et al. and published in 2014 [19]. Second, it covered primary studies published between December 2013 and January 2020. Lastly, it keeps the research questions from 2014 SLR which are included in our study.

3.1. Research Questions

The main motivation for our study is to know how Colombian ASD practitioners estimate effort. In the absence of studies in this regard, we consider it pertinent and useful to carry out a survey study that, in the context of agile methods, inquires about the similar aspects investigated by other studies at a global level.

With the above in mind, we selected four research questions from Usman and Fernandez-Diego [19] [5]. We adapted two of them to focus only on effort estimation. We added a fifth question about how the results of our study compare with similar ones in other regions of the world. As a result, through this study, we answer the following research questions:

- RQ1: What techniques have been used to estimate effort in ASD?
- RQ2: What is the accuracy level of effort estimation techniques in ASD?
- RQ3: What effort predictors have been used for effort estimation in ASD?
- RQ4: What are the characteristics of the datasets used for effort estimation in ASD?
- RQ5: Are there significant differences between the results of our study and previous studies conducted in other regions?

3.2. The Survey and the Respondents

An online survey was sent to 146 ASD practitioners from diverse technology areas and regions of the country who have worked on ASD projects performing, among others, software estimation activities.

To determine the target population, we made a list of people who had academic or work relationships with us. In addition, we made sure that all of them were working in companies or departments involved in software development. This approach to select the participants can be classified as "convenience sampling", where the sample is taken from a group of people easy to contact [20].

The survey has 23 questions distributed into five sections. The first section, related to the demographic profile, inquiries about the profession, educational level, age, gender, and software development experience of the respondents. The other sections are all related to software effort estimation and collect information to help answer the research questions. These sections request information regarding techniques, tools, standards, and effort drivers, among others.

The survey was available for three weeks and 60 complete responses were collected. These answers correspond to the 41% of the sample initially planned. The survey was distributed in Spanish and an English version is available online [21].

4. Results

4.1. Demographics

Sixty Colombian ASD practitioners from different regions answered the survey. Geographically, almost all the respondents work in the Andean region of the country, where the most densely populated cities are located.

In respect of *age and education level*, 50% of the respondents are over 36 years old, 45% are between 25 and 35 years old, and 5% of them are under 25. Moreover, more than 90% of them own at least a bachelor's degree and 63% earned a postgraduate degree.

Regarding *roles*, the respondents are mostly developers (46.7%), software designers (18.3%), and business analysts (11.7%). There is minor representation of other roles,

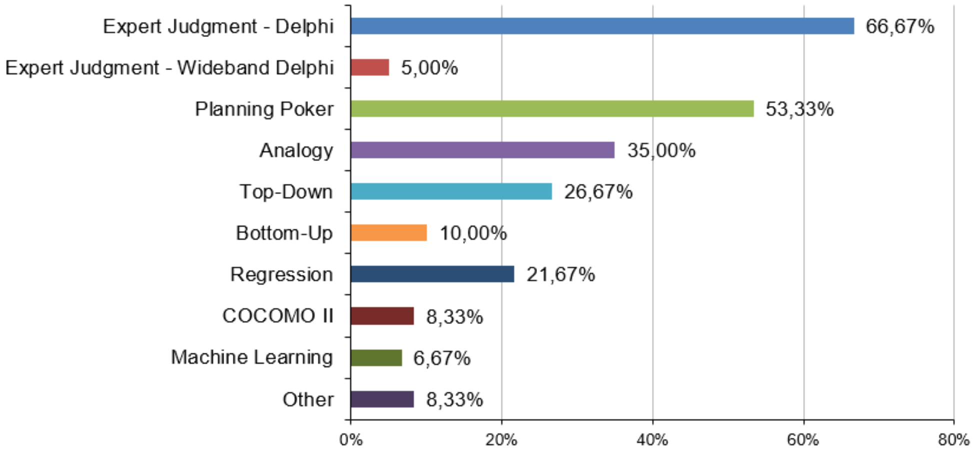

Figure 1. Techniques used for Effort Estimating

such as software architects (5%), scrum masters (5%), project leaders (5%), IT managers (2%), among others.

Regarding the *business area* where they work, most of them are in the information technology (59%) and financial sectors (27%). Other minor areas reported were government and medical and healthcare industries.

Finally, concerning *experience in software engineering*, 81.7% of the respondents reported more than 5 years working in the software industry and regarding *effort estimation experience*, 48.3% of respondents reported more than 5 years of experience, 26.7% have at least 2 years, and 13.3% less than one year doing estimation tasks.

4.2. Effort Estimation

This section presents the survey results grouped by research question.

4.2.1. RQ1: What techniques have been used to estimate effort in ASD?

There are six questions in the survey oriented to answer this research question. First, we asked them about the effort estimation techniques used in their agile projects. The results show that the top 5 techniques used by the respondents are Expert Judgment - Delphi (66.7%), Planning Poker (53.3%), Analogy (35%), Top-Down (26.7%), and Regression (21.7%). Figure 1 shows the results for all effort techniques.

Second, we inquired about agile frameworks used by the respondents. The most used frameworks are Scrum (100%), Kanban (41.7%), Extreme Programming (36,6%), and Test-Driven Development (15%). Other agile methods are used by less than 10% of the respondents, including Lean Software Development, Crystal, Dynamic Systems Development Method (DSDM) and Feature Driven Development (FDD). Frameworks for scaling agile such as SAFe were reported in only a few cases.

Third, we asked at what planning levels the respondents perform the effort estimation. The results indicate that 78.3% of the respondents estimate at the User Story level, 66.7% estimate at project level and 45% estimate at release and iteration level.

Fourth, we inquired about the software lifecycle activities where the respondents performed effort estimation. The most relevant software activities where estimation is performed are development (88.3%), followed by analysis (60%), design (53.3%), testing (51.7%), documentation, maintenance (36.7%), and training (18.3%).

Fifth, we inquired about tools to support the effort estimation process. We found that spreadsheets are the most used (90%), followed by customized tools (20%). Specialized tools for estimation are used to a lesser extent (20% for open source tools and 10% for proprietary ones). 5% of the respondents use no tools and only 3.3% of them use project management or application lifecycle management tools such as Kanban, MS Project or Azure DevOps tools.

Finally, we asked about software measurement units used in agile projects. The most used unit is Story Points (58.3%), followed by Use Case Points (33.3%), Function Points (30%), and Lines of Code (21.7%). A significant percentage of respondents (16.7%) use no unit and some of them reported 'working hours' as an additional measurement unit, even though this one is not defined as a software measurement unit.

4.2.2. RQ2: What is the level of accuracy of effort estimation techniques in ASD?

To answer RQ2, the survey includes three general questions and one specific for the Planning Poker technique.

The results show that the accuracy level is high for 23.3% of the respondents, medium for 53.3%, and low or very low for 11.7% of them. Additionally, 11.7% of the respondents do not have a clear idea about the accuracy of their estimates.

With respect to techniques for measuring the effort estimation accuracy. Most of the respondents (70%) use no technique to calculate the estimation accuracy levels, while 18.3% use variance analysis, 8.3% Prediction at level X or PRED(X), and 8.3% Magnitude of Relative Error (MRE).

Regarding factors affecting accuracy of effort estimation. Results show that the most relevant factors are requirements definition (80%), followed by the size and experience of the project team (58.3%), user involvement (45%), excessive optimism or pessimism (36.7%), project management (35%), and formality of the estimation process (23.3%). Figure 2 shows all factors that somehow influence effort accuracy level.

Table 1. Results of Accuracy Level for Planning Poker

Type of estimation	Accuracy level					
	Very High	High	Medium	Low	Very low	Not Apply
Duration	12%	41%	**44%**	0%	3%	0%
Cost	6%	12%	**56%**	14%	3%	9%
Resources	3%	38%	**45%**	6%	0%	8%

Focused on ASD and considering that Planning Poker is the most used effort estimation technique [10], we formulated questions to find out the accuracy level of duration, cost, and resources estimates. The data collected indicate that 56.6% of the respondents

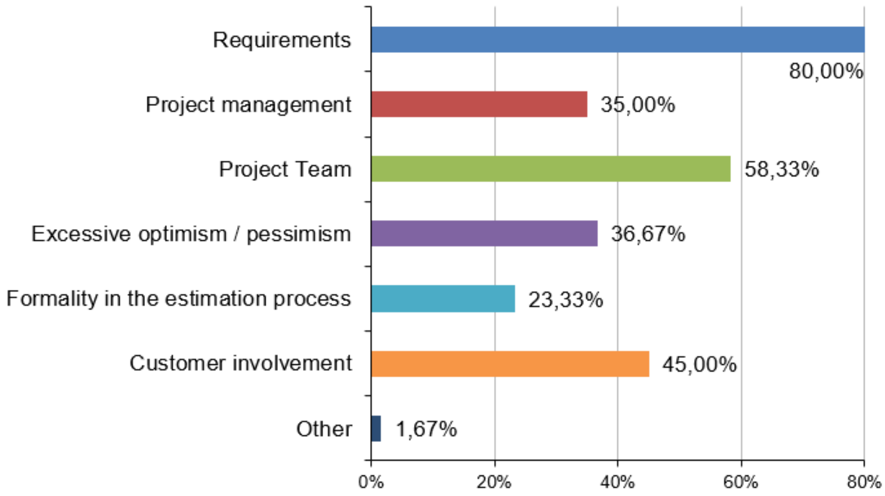

Figure 2. Factors Influencing Effort Accuracy Level

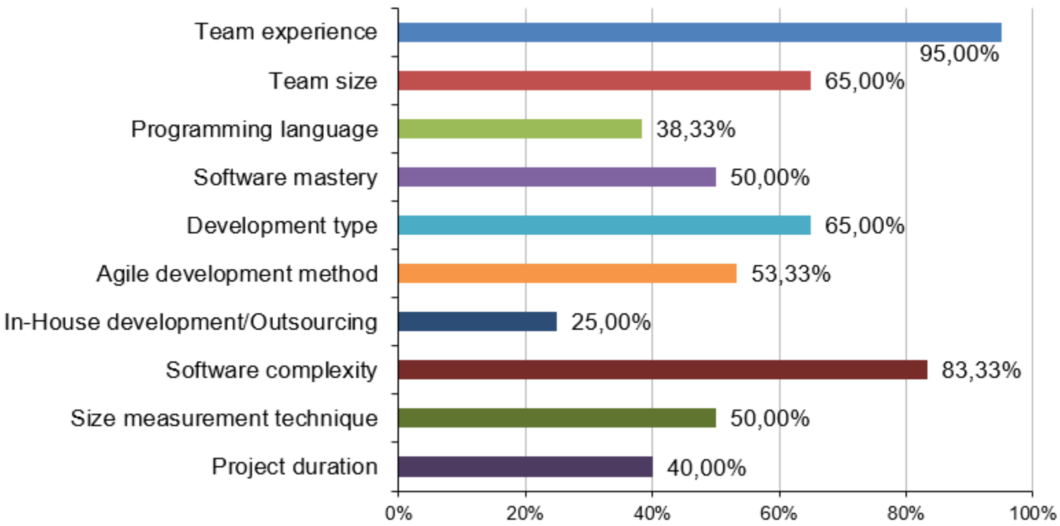

Figure 3. Effort Predictors in ASD

have used Planning Poker as an effort estimation technique. Table 1 shows the results distributed by type of estimation and accuracy where Planning Poker is used.

4.2.3. RQ3: What effort predictors have been used for effort estimation in ASD?

We formulate two questions to know which effort predictors are considered fundamental in ASD. Regarding the perception of the predictors' importance in the effort estimation activities, results show that the team experience is a fundamental factor for almost all of the respondents (95%). The second most selected factor is software complexity (83.3%). The third, fourth, and fifth places in the list are type of development (65%), team size (65%), and agile development method (53.3%), respectively (Figure 3).

Concerning the predictors actually used by the respondents, the resulting list is almost the same. They consider team experience as the main factor (86.7%), followed by software complexity (70%), development type, i.e., new or maintenance (60%), and team size (55%). Other factors are considered but to a lesser extent, including development method (agile or traditional), software domain (business sector), and programming language, among others. The four most important effort predictors are the most used in practice by the respondents. Two noteworthy differences refer, on the one hand, to the agile method used, and on the other, to the project duration. The agile approach adopted by the team seems to be less important in practice, while the project duration turns out to be more important in practice than in theory.

4.2.4. RQ4: What are the characteristics of the datasets used for effort estimation in ASD?

First of all, the most important finding is that the majority of the people surveyed (83.6%) have never made an effort estimate based on project datasets. However, from the responses of people who have used datasets, we identified two characteristics of the datasets used. First, the organizations where the respondents work own these datasets. Second, the datasets are used to estimate by analogies. Furthermore, only 3.6% of the respondents have used licensed datasets, 7.2% free access datasets, and 10.9% datasets owned by the company.

Regarding how the datasets have been used, we found that 12.7% of the respondents use them to make comparisons among projects of the same company, 9.09% apply parametric effort estimation techniques by using the datasets, and 5.4% use the datasets to estimate effort based on comparisons with projects outside their companies.

4.2.5. RQ5: Are there significant differences between the results of this study and previous studies conducted in other regions?

Estimation techniques. In this regard, Usman et al. [10] conclude that Planning Poker and Analogy are the most used techniques in ASD. Vera et al. [11] report Expert Judgment as the main technique for software effort estimation and, in some cases, Expert Judgment combined with Analogy. In our study, Expert Judgment and Planning Poker are the most used techniques for effort estimating. Our study and the one by Usman et al. [10] are entirely comparable, as the respondents chose from the same list of estimation techniques. The comparison with the study by Vera et al. [11] is partial because the list of techniques provided by their survey is unknown. In the study by Prokopova et al. [12], respondents had the option of choosing Expert Judgment and Analogy but not Planning Poker. Therefore, only the results obtained in relation to Expert Judgment are comparable in all studies, as shown in Figure 4.

Effort predictors. Our results point out that the most relevant is team experience, followed by software complexity, and development type (new or maintenance). Ramessur et al. [13] list similar factors affecting the effort estimates and conclude that the prime factors that may affect the estimation accuracy are team experience, technical skills, and requirements complexity.

Estimation Accuracy Level. Our results indicate that the perception of accuracy level of estimates is high for 23.3% of the respondents, medium for 53.3%, and low or very low for 11.7% of them. On the other hand, 11.7% of the respondents do not have a clear idea about the accuracy of their estimates. Usman et al. [10], as many other authors, report a tendency to effort underestimation. They also found that the combination of techniques and Story Points have a positive correlation with estimation accuracy. Vera et al. [11] indicate that almost half of companies overestimate the project effort, independently of the life cycle being used. This happens in some cases because the company uses a multiplier to overestimate projects in order to reduce the endeavor risk. Additionally, companies get no feedback from their estimations, and consequently they do not know the estimations accuracy level.

Prokopova et al. [12] investigate the number of projects that are delivered on time. They found that more than half of the companies (54%) do not finish 25% of their projects within their estimations, which is consistent with the trend identified in the other related studies.

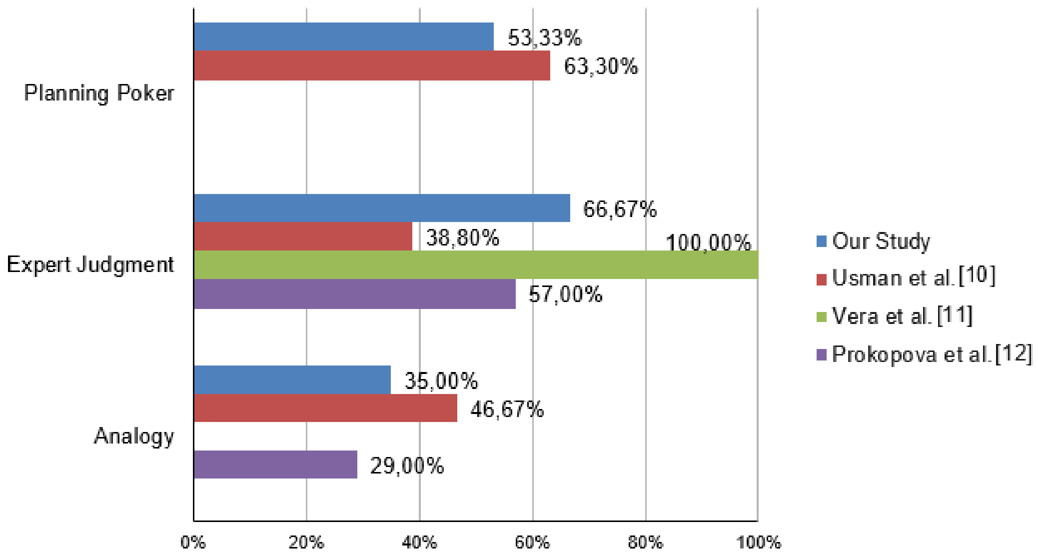

Figure 4. Estimation Techniques in Related Studies

4.3. Correlations

A correlation analysis was performed to identify some of the main interdependencies between significant variables. We used JASP, a free and open-source program for statistical analysis supported by the University of Amsterdam [22]. Most of the variables are nominal categorical with more than two values. Contingency tables, contingency coefficient, and Chi-squared test were used to determine whether there is a statistically significance between selected variables. Table 2 shows the results of the correlation analysis. Our results are similar to the Mahnič and Hovelja in terms of dependencies between respondents experience and effort estimation accuracy level [16].

5. Discussion

5.1. Effort Estimation Techniques

In ASD there is a preference for non-algorithmic techniques because project managers and teams favor approaches that aid collaboration and consensus for the estimation of the

Table 2. Correlations between Significant Variables

Variables	p-value	Contingency Coefficient	Explanation
SWE Experience and SDEE Experience	p<0.001	0.653	Significant association (moderate positive correlation) between the Software Engineering Experience and Experience in Software Development Effort Estimation.
SWE Experience and Accuracy Level	p=0.012	0.497	Significant association (moderate positive correlation) between the Software Engineering Experience and the Perception about Estimation Accuracy Levels.
SDEE Experience and Accuracy Level	p<0.001	0.610	Significant association (moderate positive correlation) between the Software Development Effort Estimation Experience and the perception about Estimation Accuracy Levels.

effort required and/or the size of the software to be built [5]. In this category, Expert Judgment oriented techniques are most commonly used by ASD practitioners. Planning Poker and Delphi are the most used. Analogy and Top-Down techniques are at the second level of use.

In regard to algorithmic estimation techniques, Regression and COCOMO are the most relevant, although they are used to a lesser extent than non-algorithmic techniques. Interestingly, techniques based on function points are not commonly used.

On the other hand, learning-oriented techniques are not sufficiently widespread in agile contexts. Instead of asking for specific techniques, our survey asks only for the use of learning-oriented techniques. Only 6.7% of the respondents have used them for software estimation.

Another important finding is that almost all respondents use at least one estimation technique. This fact suggests that effort estimation is a frequent and valuable process in agile projects.

Regarding estimation accuracy, the perception of respondents of high, average, and low levels of accuracy are 23.3%, 53.3%, and 11,7%, respectively. Moreover, 70% of the respondents use no technique for analyzing the accuracy of their estimates. Among the respondents who analyze accuracy, the most representative technique for measuring it is variance analysis(18.3%). In less than 17% of cases, the techniques used are oriented to measure deviation or error magnitude. The aspects that most influence estimation accuracy are requirements (definition, specification, size, etc.) and team size and experience.

5.2. Predictors for Effort Estimation

The factors that most influence the estimation process can be divided into two main groups: Those related to the team carrying out the software project and those that are connected to the characteristics of the project.

On the one hand, team experience includes aspects such as the technical skills of the members, their maturity as a team, as well as the quantity and quality of their training and the know-how of the team members about the software process to be used. That experience is the most important factor to consider in estimating effort, according to the results. On the other hand, the characteristics of the project have similar relevance. In this category, the complexity of the software to be built is the most significant factor. Another influential factor is whether the project is a new development, maintenance, or enhancement, mainly due to the level of uncertainty involved in each type of development. Lastly, in a third-place of relevance is the software domain, i.e., the targeted subject area of the software to be built.

5.3. Datasets and Tools for Effort Estimation

We found that spreadsheets are the most popular tool for supporting the effort estimation process, possibly due to the ease and flexibility to build models with them. Spreadsheets are suitable for building simple applications to implement effort models such as regression curves based on public or proprietary datasets. The use of proprietary tools is not common perhaps due to the licensing prices of this type of tools.

With respect to datasets, they are rarely used. This can be a strong reason to have low usage rates of parametric and learning-oriented effort techniques. In cases where datasets are used, proprietary datasets stand out, i.e., datasets based on info from previous projects of the company. Finally, datasets are employed mainly for productivity comparison purposes between past and current projects within the organization.

5.4. Planning Level in ASD

The most frequent planning level is User Stories / Requirements / Use Cases (78.3%). Likewise, there are high levels of planning on projects (66.7%), releases (45%), and iterations (45%). This might suggest that there is an interest in ASD for effort predictions at iteration, release, and project level. This in turn would confirm one of the main effort estimation goals in the early stages of the software life cycle: planning, as accurately as possible, the resources required by the software project. It would be interesting to know how planning is performed at the project level, as one of the main characteristics of ASD projects is uncertainty in scope. Lastly, it is worth noting that more than 98% of the people surveyed perform some type of planning. Only one respondent admitted that she makes no planning, at least at the levels considered in the survey.

5.5. Comparison with other Studies

Table 3 shows the summary of related studies, in terms of the characteristics of the data collection processes.

Our results show that almost all surveyed people (88.3%) estimate coding activities, while analysis (60%), design (53.3%), and testing (51.6%) activities are taken into account for effort estimation in similar proportions. This is in line with Usman et al. [10], where coding activities are considered to be the most important for effort estimation (86.67%). However, they found that analysis (43,33%) and design (53,33%) activities are considered for effort estimation in a smaller proportion, while testing activities are the second most important factor.

With respect to factors that can affect effort estimation, they are similar in all cases. Likewise, the relevance assigned to these factors is also similar. Finally, team

Table 3. Characteristics of Related Studies

Study	Instrument	Analysis Unit	Observations
Usman et al. [10]	Survey	Individual	Online questionnaire 60 people surveyed.
Vera et al. [11]	Interview	Company	Semi-structured interview, One person per company 10 people surveyed.
Prokopova et al. [12]	Survey	Company	Questionnaire sent to 91 companies.
Our study	Survey	Individual	Online questionnaire 60 people surveyed.

experience is the most important factor to consider in effort estimation according to our results and related studies [10], [13], [14].

As for estimation accuracy, we used a 5-point scale: very high, high, medium, low, and very low. For the same purpose, Usman et al. [10] use 2 major categories: overestimated or underestimated to express estimation accuracy. Vera et al. [11] do not report how accurate the effort estimates are, instead, they concluded that the effort is always overestimated due to a multiplying factor that magnifies the estimates. In Prokopova et al. [12], only the percentage of projects not delivered on-time is reported.

6. Threats to Validity

Validity determines how much we can trust that the study findings accurately reflect the reality we want to describe or explain. For the particular case of a survey, many factors can affect that validity and they are discussed in this section, as well as, the actions we took to avoid or mitigate their effects.

External validity. First, the participants were mostly developers or people who play specific roles within software teams such as designers, analysts, architects, or scrum masters. Most of them have more than 5 years of experience in the software industry, with ages ranging from under 25 to over 45 years. Thus, the sample is quite heterogeneous in terms of experience, age, and job role. Second, the vast majority of them are in the Andean region where the largest population and the greatest industrial development of the country are concentrated. Despite the above, the results of our study cannot be fully applied to the entire population. Consequently, it is feasible that a replication of this study yields somewhat different results, in some of the analyzed aspects.

Internal validity. First, to reduce the threat that the subjects may not be competent enough to answer the questionnaire proposed, a list of potential respondents was built to ensure that participants had worked for software companies and had at least basic knowledge of effort estimation. Second, we collected specific demographic information regarding their career to be sure that the answers given by our respondents are reliable and suitable. Third, we conducted a pilot experiment with two graduate students and obtained feedback about the questionnaire itself and the time required to respond.

Construct validity. In this regard, we mitigated the associated threats by designing the questionnaire based on the results of an SLR. In addition, we guaranteed the relevance of each question included by associating it with one of the research questions formulated in this study. Moreover, the questionnaire items were iteratively improved so that we ensure the questions are valid, clear, unambiguous, and bias-free. Lastly, we write an introduction to the survey where we explained the purpose of the study, and clarified the concept of effort estimation.

7. Conclusions

We present a survey study aimed at determining how Colombian ASD practitioners carry out the effort estimation process within agile projects. The designed survey inquires about the effort estimation techniques used, the accuracy of the estimates, the factors that influence the accuracy of the estimates and whether datasets are used to support the estimation processes in any way. The survey was answered by 60 people who play various roles within Colombian software companies. We used convenience sampling and ensured that all participants work in the software industry.

With respect to estimation techniques, the ones based on Expert Judgment are the most frequently used. Within this category, Planning Poker stands out because it is used by more than half of the respondents and its effectiveness is considered medium-high when estimating duration, costs, and resources in ASD projects. Furthermore, the use of parametric techniques is representative (regression, COCOMO, Function Points Analysis). A minority of the respondents report using learning-oriented techniques and the use of proprietary techniques is marginal.

Regarding the agile processes, Scrum has been used by all participants, while Kanban and XP were reported by less than half of the respondents. The predominance of Scrum makes it the typical framework where the application of effort estimation techniques takes place. Moreover, effort estimation occurs primarily within the fine-grained planning process, that is, at the User Story or Use Case level. Data also suggests that estimation results are used, to a lesser extent, in planning iterations, releases, and the entire project. These results are consistent with studies that describe the way agile projects are planned and managed [10].

The respondents also reported that the activities for which effort estimates are carried out are development (90.9%), analysis phase (61.8%), design tasks (58.2%), and software testing work (56.4%).

Regarding software sizing, Story Points and Use Case Points are used to a great extent as units for estimating software size. Approximately, one-third of the surveyed people use Function Points, and LOC to a lesser extent. Additionally, about 20% of the respondents reported that they use no size measurement technique.

As for tooling, the use of specialized software to perform software estimation is not common. This activity is dominated by spreadsheets (90.9%) and, to a lesser extent, by software tools tailored to the needs of each organization, which may be based on spreadsheets.

Estimation accuracy is considered high for a fifth of the surveyed people, average for slightly more than half of them, and low or indeterminate for the rest of the respondents. However, about 71% of the respondents report that a formal accuracy analysis process is not carried out. The use of variability analysis was reported by 20% of them, while MRE and PRED(X) were reported by a small proportion of people. All factors affecting the accuracy of estimates suggested in the survey are recognized as significant.

With respect to effort predictors, the most relevant in order of importance are team experience, software complexity, development type, team size, software domain, and software size measurement technique used.

With respect to datasets, we found that, in most cases, they are not used to support the effort estimation process. This indicates a low level use of parametric or learning-oriented techniques, which require historical data for calculating estimates or training the models. In the few cases where datasets are used, they are mostly public sources or historical data from the organization itself.

The general results suggest that although estimations are performed as an input for project planning, their accuracy is not formally evaluated. In this regard, the majority of respondents simply perceive that their estimates are acceptable or good, but there is no evidence of any effective process that supports such perceptions. We believe that the support of tools that facilitate data capture, as well as its subsequent analysis, could substantially improve the quality of the estimates.

References

[1] M. Chemuturi, 2009.Software Estimation Best Practices Tools & Techniques, Broward-Florida, J. Ross Publishing.

[2] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic Literature Review of Machine Learning Based Software Development Effort Estimation Models," Inf. Softw. Technol., vol. 54, no. 1, pp. 41–59, 2012. https://doi.org/10.1016/j.infsof.2011.09.002

[3] A. Trendowicz, 2014, Software Project Effort Estimation. Berlin, Germany, Springer-Verlag.

[4] O. Malgonde and K. Chari, "An Ensemble-Based Model for Predicting Agile Software Development Effort," Empirical. Software. Engineering., vol. 24, no. 2, pp. 1017–1055, 2019. https://doi.org/10.1007/s10664-018-9647-0

[5] M. Fernandez-Diego, E. R. Mendez, F. Gonzalez-Ladron-De-Guevara, S. Abrahao, and E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review," IEEE Access, vol. 8, pp. 166768–166800, 2020. https://doi.org/10.1109/ACCESS.2020.3021664

[6] PMI, A Guide to the Project Management Body of Knowledge, 6th ed. Newtown Square, Pennsylvania, USA: Project Management Institute, 2017.

[7] K. Moløkken and M. Jørgensen, "A review of software surveys on software effort estimation," in Proceedings - 2003 International Symposium on Empirical Software Engineering, ISESE 2003, 2003, pp. 223–230.

[8] T. Vera, S. F. Ochoa, and D. Perovich, "Survey of Software Development Effort Estimation Taxonomies," no. March, 2018.

[9] A. Saeed, W. H. Butt, F. Kazmi, and M. Arif, "Survey of software development effort estimation techniques," in ACM International Conference Proceeding Series, 2018, pp. 82–86.

[10] M. Usman, E. Mendes, and J. Börstler, “Effort Estimation in Agile Software Development: A Survey on the State of the Practice,” in ACM International Conference Proceeding Series, 2015, vol. 27-29-Apri, p. 10. https://doi.org/10.1145/2745802.2745813

[11] T. Vera, S. F. Ochoa, and D. Perovich, “Understanding the Software Development Effort Estimation in Chilean Small,” 2018.

[12] Z. Prokopova, P. Silhavy, and R. Silhavy, Analysis of the Software Project Estimation Process: A Case Study, vol. 984. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-19807-744

[13] M. A. Ramessur and S. D. Nagowah, “Factors Affecting Sprint Effort Estimation,” Adv. Intell. Syst. Comput., vol. 1089, no. January, pp. 507–518, 2020. https://doi.org/10.1007/978-981-15-1483-8_43

[14] L. Madeyski and M. Majchrzak, “Factors influencing User Story estimations: an industrial interview and a conceptual model,” Cent. East. Eur. J. Manag. Econ., vol. 4, no. 4, pp. 261–280, 2016.

[15] P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, and W. Pedrycz, “Predicting Development Effort from User Stories,” Int. Symp. Empirical Software Engineering. Meas., pp. 400–403, 2011. https://doi.ieeecomputersociety.org/10.1109/ESEM.2011.58

[16] V. Mahnič and T. Hovelja, “On Using Planning Poker for Estimating User Stories,” J. Syst. Softw., vol. 85, no. 9, pp. 2086–2095, 2012. https://doi.org/10.1016/j.jss.2012.04.005

[17] N. C. Haugen, “An Empirical Study of Using Planning Poker for User Story Estimation,” Proc. - Agil. Conf. 2006, vol. 2006, pp. 23–31, 2006. https://doi.org/10.1109/AGILE.2006.16

[18] J. Linaker, S. M. Sulaman, R. Maiani de Mello and M. Host, ”Guidelines for Conducting Surveys in Software Engineering,” Department of Computer Science, Lund University, 2015.

[19] M. Usman, E. Mendes, F. Weidt, and R. Britto, “Effort Estimation in Agile Software Development: A systematic literature review,” in ACM International Conference Proceeding Series, 2014, pp. 82–91. https://doi.org/10.1145/2639490.2639503

[20] M. Saunders, P. Lewis, and A. Thornhill, Research Methods For Business Students, 8th ed. Pearson Education, 2019.

[21] “Survey Translation” 2022. [Online]. Available: http://it-jmca.com/UN/Paper2/Survey.pdf

[22] JASP Team. (2022). JASP (Version 0.16.4)[Computer software]. Retrieved from https://jasp-stats.org/