An Effort Estimation Support Tool for Agile Software Development: An Empirical Evaluation

Conference Paper · July 2019

DOI: 10.18293/SEKE2019-141

Emanuel Dantas
Universidade Federal de Campina Grande (UFCG)

Alexandre Costa
Universidade Federal de Campina Grande (UFCG)

Mirko Perkusich
VIRTUS

Hyggo Oliveira de Almeida
Universidade Federal de Campina Grande (UFCG)

An Effort Estimation Support Tool for Agile Software Development: An Empirical Evaluation

Emanuel Dantas§ ¶ ‖, Alexandre Costa ‖ ¶, Marcus Vinicius ¶, Mirko Perkusich§ ¶ ,
Hyggo Almeida¶ , Angelo Perkusich¶
§ Federal Institute of Paraíba
Monteiro, Paraíba, Brazil, 58500-000
{emanueldantas, mirkoperkusich}@ifpb.edu.br
¶ Intelligent Software Engineering (ISE) Group, Federal University of Campina Grande
Campina Grande, Paraíba, Brazil, 58429-140
{marcusbarbosa, alexandrecosta, almeida, perkusich}@embedded.ufcg.edu.br
‖ CAPES Foundation, Ministry of Education of Brazil, Brasilia - DF, Zip Code 70.040-020

***Abstract*—Accurate effort estimation is an important part of the software process. In Agile Software Development, the techniques for predicting effort are mostly based on expert judgment, but there are approaches based on Machine Learning. The theme continues to be challenging and a subject of further studies given the difficulty of finding accurate solutions to the problem. This paper proposes and evaluates a tool based on the decision tree method for effort estimation in agile projects. We evaluated our tool given its accuracy and ease of use collecting data from four projects. To evaluate the accuracy, we compared the values of Magnitude of Relative Error from the teams' estimations with the values provided by the tool. To evaluate the ease of use, we used the Technology Acceptance Mode. The initial results show that the tool can be reliably used with minimal training. In terms of accuracy, the tool achieved lower error compared to the estimates provided by the teams (mean: 19.05% vs 33.32%), and the evaluation means in TAM were higher than 4.0 in ten of the eleven variables analyzed on a Likert scale. From this work, we conclude that estimation by decision tree is a viable technique that, at the very least, can be used by project managers to complement current estimation techniques.**

Keywords*—Agile Software Development; Effort Estimation; Machine Learning; Decision Tree.

I. INTRODUCTION

Since 2001, there has been an increasing interest in agile methodologies. Nowadays, these methods are widely used in software industry projects, especially, because of its volatility and flexibility. Agile software development (ASD) focuses on the needs of the rapidly changing environment by embracing the proposal of iterative and incremental development [7]. According to a recent study [2], Scrum stands out as an agile development process with 56% of preference. Scrum is a framework for developing and sustaining complex products [1]. One of its main events is the Sprint Planning Meeting in which occurs the task breakdown. During this process, the Scrum team defines tasks to be developed and estimates the effort to complete them.

In ASD context, effort estimation consists of predicting the effort needed to fulfill a given task [24], which is an important part of the development process, since it is one of the many steps that can lead to successful project completion. Due to the dynamic nature of ASD [8] and limited documentation [36], effort estimation is considered a complex and critical task [4]. Traditionally, it depends heavily on the expert experiences [36] and the estimated values usually are far away from the real ones, resulting in low accuracy. On the other hand, accurate estimations can improve the development planning by enabling optimal assignments of both stories and developers [22]. Additionally, high accurate values can be used in different prediction models to improve the outputs. For instance, to increase project velocity [14], to optimize developer effort across different projects in the organization, and different projects inside the organization can be centrally coordinated to increase efficiency.

Over the past years, there has been a growing interest in effort estimation researches [21], [37], [32]. In previous work [9], we identified different approaches proposed to estimate effort using historical data and expert opinions based techniques. A significant amount of these used Artificial Intelligence or Machine Learning techniques to support effort estimation in ASD, which contributed to achieve higher accuracy. Despite the contributions of recent studies, low accurate prediction is still considered a gap in effort estimation.

In this study, we focus on the empirical evaluation of an effort estimation support tool for agile software development. The proposed tool is based on historical data and uses a Decision Tree to estimate effort during software development. In the evaluation, we aim to measure (i) the accuracy of the proposed tool and (ii) its usefulness and ease of use. To do so, we used Magnitude of Relative Error (MRE) to compare the estimations made by the project team members and those ones provided by the tool with the real effort (in hours). Furthermore, through a questionnaire based on the Technology Acceptance Model (TAM) [10], professionals with experience in software development used the tool and reported their perceived usefulness and perceived ease of use. As a result, the proposed tool achieved lower MRE values compared to ones provided by the teams (mean: 19.05% vs 33.32%). As also, most of the professionals who participated in the study found the approach to be useful and easy to use to support the estimation process (evaluation mean was higher than 4.0 in ten of the eleven variables analyzed on a Likert scale [16]).

The remainder of this paper is organized as follows: Section

DOI reference number: 10.18293/SEKE2019-141

II presents the background information. Section III details the proposed tool. Section IV presents the empirical evaluation, followed by the results and discussion in Section V. Finally, threats to validity and conclusions are described in Sections VI and VII, respectively.

II. Background and Related Work

This section presents the background and related work on this paper. A considerable number of studies has been published on this subject. A Systematic Literature Review (SLR) [39] performed in 2014 aggregated and described the state of the art related to estimation techniques, effort predictors, accuracy measures and agile methods used. In 2018, Dantas et al. [9] updated this review and observed a strong indication of solutions based on Artificial Intelligence (AI) and Machine Learning (ML) methods for effort estimation in ASD. The purpose of the identified studies is to support human judgment during effort estimation. The main techniques observed were: Bayesian Network and Decision Tree.

In general, the studies use different factors in their approaches to estimate effort more accurately, either to support traditional estimation methods such as Planning Poker or in ML and AI solutions. These factors are known as Cost Drivers and represent personal or project factors that influence the estimated values [9]. For instance, to support Planning Poker, Lenarduzzi [19] considered technical ability, competence level, and managerial skill. While Grapenthin et al. [13] used factors related to the project: complexity and flexibility.

Artificial Intelligence is a branch of computer science formed by techniques that support activities of optimization or knowledge discovery [45]. A popular AI technique is the Bayesian Network which is used to estimate values considering the uncertainty of the variables (cost drivers) and can be constructed using historical data and experts. In the context of effort estimation, Mendes [25] proposed a Bayesian Network for Web effort estimation using knowledge from a domain expert. Karna and Gotovac [17] presented another model, including the relevant cost drivers with turnover, priority, and severity. Zahraoui et al. [46] adjusted story points using: priority, size, and complexity. The complexity and importance of a user story were considered by Lopez et al. [20], while Dragicevic et al. [11] considered the skills of the developers and requirements complexity. These approaches use historical data and expert knowledge to estimate effort with better accuracy.

Machine Learning is a current application of AI based on the idea that a system can learn from data rather than through explicit programming. Thus, the ML methods aim to improve the performance at certain tasks learning from the observed data. Many studies use ML to estimate effort in software projects. For instance, in Satapathy and Rath [32] Decision Tree, Stochastic Gradient Boosting and Random Forest were compared to assess effort estimation given a dataset composed by 21 projects. Porru et al. [29] used Support Vector Machine, K-Nearest Neighbors and Decision Tree for the same purpose given data from eight open source projects. Several Neural Networks were used in Panda et al. [27]. The purpose of these works is to support human judgment during effort estimation.

In this study, we use a Decision Tree which is a decision support method indicated for establishing classification or regression based on multiple covariates for developing prediction algorithms for a target variable [35]. This method constructs prediction models by recursively partitioning the data and fitting a simple prediction model within each partition [6], till a halting paradigm is fulfilled [32]. For this purpose, it uses a sequential process to identify the predictor variables that best differentiate groups along with the outcome variable of interest.

Effort estimation is still attractive to researchers since a reliable estimation process is crucial for correct project planning and a good management of the resources [33]. Several other studies on the subject can be found in the literature, mainly focus on designing a good estimator [18], [3], new cost metrics [26] or other aspects [15].

III. Proposed Tool

A. Overview

The effort estimation support tool presented in this paper uses historical data to construct a decision tree predictive model. The purpose of the tool is to support teams during task effort estimations. Therefore, it assists in the development process during the planning phase. The tool consists of a web application with an interface implemented in Angular that can be accessed from computers or smartphones. The key parts of the tool are discussed below.

B. Dataset

Agile projects often manage user requirements with models called User Stories (US). These artifacts are used to describe features that deliver value to the customer. USs are written in natural language, which makes it difficult to retrieve information [30]. Therefore, to build a dataset we have defined a model to represent the main features of software projects.

The model was built by specialists and is organized in a hierarchical structure of 03 levels to represent each feature. The first level is called *Module*, the second is *Operation*, and finally for each operation we have different types of *Tasks* in the third level. Tho whole model is composed by 03 *Modules*, 18 *Operations* and 131 *Tasks* types. Table I shows an example of operations defined to the "Authentication"module.

Table I: Examples of previously defined operations for Authentication module

Module	Operation
Authentication	Perform login with username and password Perform login with OAuth Password recovery First login Validate user permissions Update profile Create account Remove account

With the defined model, we analyzed the backlog of different organizations to create our dataset. All companies work with agile processes based on Scrum and develop projects on the web platform. We have analyzed 26 backlogs, containing a total of 530 USs and 1879 tasks. At the end, we were able to map 52% of tasks according to the model, which results in 977 items that form our dataset. For each record, in addition to the information representing the feature (*Module*, *Operation*, *Task*), we extract information from the following cost drivers: effort, human resource and technology. The information was taken from the teams during the planning meetings.

C. Prediction Algorithm

The proposed tool adopts the M5P algorithm for building the Decision Tree [43]. This algorithm represents an appropriate choice because it implements as much decision trees as linear regression for predicting a continuous variable. To train the algorithm, we used a 10-fold cross validation [5] and choose Effort (in hours) as the dependent variable. In Figure 1 we can see the relations of the cost drivers used to construct the predictive model. Category corresponds to the feature according to the model, human resource is the professional responsible. Finally, the predominant technology in the feature is also used to forecast the varying effort. Several Cost Drivers were analyzed using the Weka software, but these were chosen for achieving a result with the least error.

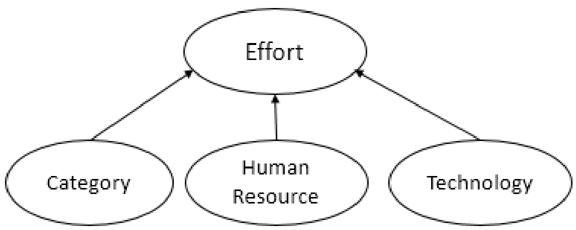

Figure 1: Cost Drivers Relation.

D. Design

The architecture of the proposed tool consists of a processing layer which executes the Decision Tree algorithm and a serving layer which exposes a REST API. The database has been implemented in MySQL and contains tables to store the historical data of the projects. To perform the processing, we used the class M5P[1] in Weka, version 3.8.0. Its interface was build using Angular framework, version 6.1.2. The tool is available [2] and can be accessed in any web browser.

The tool was developed for supporting agile teams during Sprint Planning. As shown in Figure 2, the tool has a view that shows all the tasks types and the structure is represented in three hierarchical levels (see the area of the screen indicated by 1). These tasks are in agreement with the model created in the construction of our dataset. Once selected the task type, the tool processes the decision tree algorithm and displays an effort estimation (see area of the screen indicated by 2). Finally, clicking on "detail", the user can have access to the historical values of this task.

IV. Empirical Evaluation

This section presents the case study performed to evaluate the proposed tool, which is detailed in the following subsections.

A. Case Study Planning

The evaluation is oriented towards design science in software engineering [42]. The improvement problem is the low accuracy from effort estimation tasks. The artifact used is a historical data based tool to support decision making during planning meetings. Therefore, we seek to answer the following research questions:

- RQ1: *Category*, *Human resource* and *Technology* can be used to construct a decision tree predictive model capable to reduce effort estimation errors during planning meetings?
- RQ2: The perceived usefulness and ease of use have a positive impact on the tool adoption?

To investigate the research questions, we conducted an exploratory case study [28] in a Brazilian software development company. Four agile software projects, based on Scrum, were available to participate in the study. Each project was composed of six developers and one Scrum Master who also performs the project manager role, each developer works exclusively on one project. According to the company practice, the Scrum Master does not participate in the effort estimation process, only the developers. All the projects belong to web development platform and were implemented with similar technologies.

The study lasted six sprints with 15 days each, resulting in three months of evaluation. Before the beginning, the participants were submitted to 15 minutes of training to get familiar with the proposed tool. At this moment, they received instructions and examples of use to learn how to utilize the tool.

Most participants were graduate developers (84.61%), who worked full time at the company and had more than three years of experience in software development. The remaining participants (15.49%) were undergraduate students from a computer science course, who worked part-time and had at least a year of experience in software development.

B. Case Study Execution

During the case study, the tasks were created by the teams and recorded in a spreadsheet. The authors of this paper were responsible for analyzing each task and for mapping them to the corresponded category of the model presented in Section III-B. Thereafter, the project teams estimated the effort of the tasks. At this moment, they could estimate the values by their own or use the tool to assist in the process. At the end of the study, 76% of the tasks were mapped, which corresponds

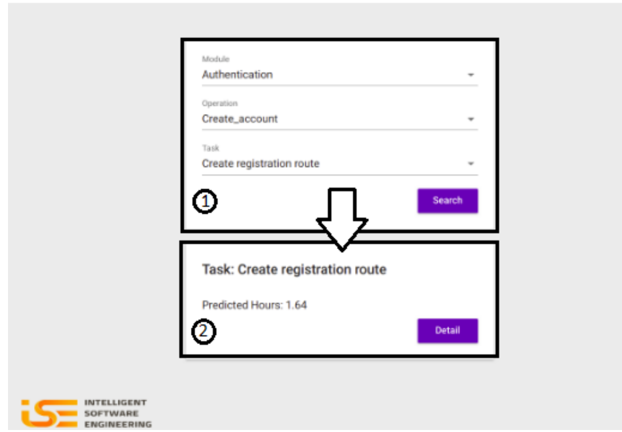

Figure 2: Screenshot of tool.

[1] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/M5P.html
[2] https://mot-client-web.herokuapp.com/

to 121 items. For each task, we recorded three values: (i) the team estimation, (ii) the tool estimation and (iii) the real effort.

To answer the RQ1, we choose the MRE (Equation 1) which measures the difference between real and estimated effort relative to the real one. Therefore, we compared the team MRE values with the tool MRE values, resulting in two value lists. Then, we use a confidence interval (with 95% confidence level) to do a statistical analysis. This procedure is recommended to characterize the uncertainty associated with a certain parameter estimated [41].

$$MRE = \frac{Real\ effort - Estimated\ effort}{Real\ Effort} \quad (1)$$

To answer the RQ2 we applied a questionnaire based on the indicators presented in the TAM [10]. The questionnaire aims at assessing users' beliefs about the usefulness and ease of use of a technology that is expected to support their work. It has been used extensively to explain and predict users' acceptance of information technology [34]. According to TAM, two variables impact adoption: perceived usefulness and perceived ease of use. Perceived usefulness refers to the degree to which an individual believes that using a particular technology would enhance his or her job performance. Perceived ease of use refers to the degree to which an individual believes that using a particular technology would be free of physical and mental effort [40]. Table II shows the applied questionnaire. The questions are divided by type: Perceived usefulness (PU), Perceived ease of use (PEoU), External variables (EV) and Attitude (AT) . The type of response follows a Likert scale [16] with five possible answers ranging from strongly disagree (mapped to number 1) to strongly agree (mapped to number 5). The participants were asked to respond to each statement in terms of their own degree of agreement or disagreement [23]. The questionnaire was applied at the end of the six sprints and all the participants submitted their answers.

V. RESULTS AND DISCUSSION

This section outlines the results with respect to the research questions, which are answered in different sub-sections.

A. RQ1: Category, Human resource and Technology can be used to construct a decision tree predictive model capable to reduce effort estimation errors during planning meetings?

After collecting data from the case study, we were able to calculate MRE values from 121 tasks, resulting in two types of errors: team MRE and tool MRE. As mention before, we used the confidence interval to analyze the data. In several regression applications, these intervals are computed by supposing the errors follow a normal (Gaussian) distribution or another more general distribution with a number of parameters [31].

In Table III, are shown the mean, standard deviation and confidence intervals (with 95% confidence level) for the MRE values. We can observe that the tool has achieved lower MRE mean values when compared to each project team values. The overall result shows a 19.05% MRE mean value for the tool and 33.32% for the teams, i.e., the proposed tool can provide more accurate estimations. Also, the lower overall standard deviation obtained by the tool indicates that the estimated values tend to be close to the mean, generating more homogeneous predictions. The overall confidence intervals (14.87% - 23.62% and 25.55% - 41.42%) do not intersperse, thus the two group values can be considered statistically different.

Regarded to the RQ1, we can conclude that the selected cost drivers can be used to construct a decision tree predictive model capable to reduce effort estimation errors during the planning meeting. Therefore, the proposed tool can provide accurate estimations when compared to the ones given by the project teams.

B. RQ2: The perceived usefulness and ease of use have a positive impact on the tool adoption?

In order to carry on a qualitative evaluation, we used the TAM. Table II shows the assigned variables for this study.

Table IV describes the values for median, mean, standard deviation (SD) for questions of the perceived usefulness. All average values were higher than 4.0, indicating that participants generally had positive attitudes toward the tool.

Table IV: Perceived usefulness

Variable	Definition	Mean	Median	SD
V1	Using the tool is useful for estimating task effort.	4.32	4	0.712
V2	Using the tool allows quick access to a historical basis	4.82	5	0.743
V3	The tool is accessed at all planning meetings	4.12	4	1.04

In Table V, all mean values related to the ease of use are above the midpoint and the standard deviations are within the range from 0.716 to 0.926 indicating a narrow spread around the mean.

Table V: Perceived ease of use

Variable	Definition	Mean	Median	SD
V4	Learning to use the tool was easy for me	4.74	5	0.786
V5	I often get confused in researching and understanding information in the tool	3.94	4	0.926
V6	Access to the tool is simple	4.78	5	0.716

The variable *V9* in Table VI has the lowest mean value. This occur because the tool was recently adopted, thus the motivation among co-workers was expected to be low. Also, the standard deviation of *V9* was greater than one, representing a high dispersion of the information.

Table VI: External variables

Variable	Definition	Mean	Median	SD
V7	The navigation features (menus, icons, links and buttons) are all clear and easy to find	4.34	4	0.696
V8	The tool has a nice interface	4.74	5	0.556
V9	My co-workers encourage me to use the tool	3.42	3	1.213

The results of Table VII show that the participants perceived the attitude of the information systems, since, for all variables, the average has resulted in more than 4.0, showing agreement on all the statements.

Table VII: Attitude

Variable	Definition	Mean	Median	SD
V10	I believe it's best to use the tool instead of traditional planning.	4.46	4	0.879
V11	My intention is to use the tool to better plan my project tasks	4.24	4	0.786

Table II: Questionnaire Statements on: Perceived usefulness, Perceived ease of use, External variables and Attitude

Type	Definition	Variables
Perceived usefulness (PU)	The level at which a person believes that using the tool improves the performance of their tasks.	V1: Using the tool is useful for estimating task effort.
		V2: Using the tool allows quick access to a historical basis
		V3: The tool is accessed at all planning meetings
Perceived ease of use (PEoU)	Level at which the person presents their perception of the tool in terms of ease of learning and operation.	V4: Learning to use the tool was easy for me
		V5: I often get confused in researching and understanding information in the tool
		V6: Access to the tool is simple
External variables (EV)	External variables provide a better understanding of what influences perceived utility and ease of use.	V7: The navigation features (menus, icons, links and buttons) are all clear and easy to find
		V8: The tool has a nice interface
		V9: My co-workers encourage me to use the tool
Attitude (AT)	Intention of the individual to use the tool	V10: I believe it's best to use the tool instead of traditional planning .
		V11: My intention is to use the tool to better plan my project tasks

Table III: Results of the MRE calculation

	Team MRE				Tool MRE			
	Mean (%)	**Standard Deviation (%)**	**Lower Confidence Interval (%)**	**Upper Confidence Interval (%)**	**Mean (%)**	**Standard Deviation (%)**	**Lower Confidence Interval (%)**	**Upper Confidence Interval (%)**
Project A	43.27	55.12	38.43	42.12	25.67	15.23	23.16	32.19
Project B	26.56	38.77	22.48	31.02	14.78	10.12	13.21	19.43
Project C	31.45	42.24	26.38	33.10	18.21	27.34	16.33	22.91
Project D	32.05	58.12	30.21	36.39	17.54	26.21	15.88	20.32
Overall	33.32	51.33	25.55	41.42	19.05	22.92	14.87	23.62

When using Likert-type scales it is imperative to calculate and report Cronbach's alpha coefficient for internal consistency reliability [12]. The Cronbach' s alpha is an index of inter-item homogeneity that describes how related a set of items is as a group [38]. In table VIII, we can see the coefficient for each variable. All multi-item constructs should meet the guidelines for a Cronbach's alpha of greater than 0.70. Table VIII demonstrates that the coefficient values, for all constructs in the measurement model, exceeded the recommended threshold. Therefore, based on the presented results related to the RQ2, we can conclude that the tool is easy to use and has utility for the teams.

Table VIII: Descriptive statistics

Construct	Variables	Mean	Cronbach's
Perceived usefulness	V1	4.32	0.8829
Perceived usefulness	V2	4.82	0.9084
Perceived usefulness	V3	4.12	0.8779
Perceived ease of use	V4	4.74	0.8745
Perceived ease of use	V5	3.94	0.8238
Perceived ease of use	V6	4.78	0.8779
External variables	V7	4.34	0.8989
External variables	V8	4.74	0.8779
External variables	V9	3.42	0.8229
Attitude	V10	4.46	0.9208
Attitude	V11	4.24	0.8779

VI. Threats to Validity

This section presents the main threats to validity identified in the study, according to Wohlin et al. [44]. The analysis of the threats aims to examine the relationship between the conclusions reached and the reality.

A threat to external validity identified is related to the platform of the selected projects. All of them were web development projects with similar technologies, thus our findings can not be generalized. To address this threat, we intend to conduct a new case study with projects from mobile and desktop platforms. Another identified threat refers to internal validity. The teams that participated in the case study were composed by graduate developers and undergraduate students, with different levels of experience, which can impact the error measurement. Therefore, the results from this study must be considered as indicators and further evaluation with different contexts must be conducted.

VII. Conclusion and Future Work

This paper proposed and evaluated an effort estimation support tool for agile software development. The tool uses historical data to construct a decision tree predictive model to provide effort estimations during the planning meeting. To evaluate the tool, we performed a case study in a Brazilian software development company, in which four agile projects (based on Scrum) and 24 professional participated. The case study lasted three months and thereafter we performed a quantitative and qualitative analysis.

The achieved results indicate that the tool can support the planning meeting, reducing the errors related to the task estimation process. Also, we concluded that the majority of the professionals who participated in the study found the tool useful and easy to use for supporting the effort estimation, and they would regularly use the tool for future plannings in their job.

As a limitation, we realize that the low numbers of teams and the types of projects are prohibitive to generalize our findings. In the future, we plan to conduct another case study with more projects and different contexts. Additionally, we would like to evaluate the usage of the tool with more experienced teams.

Acknowledgment

The authors acknowledge the financial support given by CAPES/Brazil.

Referências

[1] Sutherland j, schwaber k. the scrum guide. https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf. accessed in 23/02/2019.

[2] Versionone. 9th annual state of agile development survey results. https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf. accessed in 23/02/2019.

[3] M. A. Ahmed, I. Ahmad, and J. S. Alghamdi. Probabilistic size proxy for software effort prediction: A framework. *Information and Software Technology*, 55(2):241–251, 2013.

[4] M. S. S. Basha and D. Ponnurangam. Analysis of empirical software effort estimation models. *CoRR*, abs/1004.1239, 2010.

[5] N. Bhargava, S. Dayma, A. Kumar, and P. Singh. An approach for classification using simple cart algorithm in weka. In *Intelligent Systems and Control (ISCO), 2017 11th International Conference on*, pages 212–216. IEEE, 2017.

[6] L. Breiman. *Classification and regression trees*. Routledge, 2017.

[7] M. Brhel, H. Meth, A. Maedche, and K. Werder. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 61:163–181, 2015.

[8] A. Cockburn. *Agile Software Development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2007.

[9] E. Dantas, M. Perkusich, E. Dilorenzo, D. Santos, H. Almeida, and A. Perkusich. Effort estimation in agile software development: an updated review. In *Proceedings of the30th International Conference on Software Engineering and Knowledge Engineering*, 2018.

[10] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[11] S. Dragicevic, S. Celar, and M. Turic. Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, 127:109–119, 2017.

[12] J. A. Gliem and R. R. Gliem. Calculating, interpreting, and reporting cronbachs alpha reliability coefficient for likert-type scales. Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education, 2003.

[13] S. Grapenthin, M. Book, T. Richter, and V. Gruhn. Supporting feature estimation with risk and effort annotations. In *Software Engineering and Advanced Applications (SEAA), 2016 42th Euromicro Conference on*, pages 17–24. IEEE, 2016.

[14] P. Hearty, N. Fenton, D. Marquez, and M. Neil. Predicting project velocity in xp using a learning dynamic bayesian network model. *IEEE Transactions on Software Engineering*, 35(1):124–137, 2009.

[15] M. Jørgensen. The influence of selection bias on effort overruns in software development projects. *Information and Software Technology*, 55(9):1640–1650, 2013.

[16] A. Joshi, S. Kale, S. Chandel, and D. Pal. Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, 7(4):396, 2015.

[17] H. Karna and S. Gotovac. Estimating software development effort using bayesian networks. In *Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on*, pages 229–233. IEEE, 2015.

[18] E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung. Exploiting the essential assumptions of analogy-based effort estimation. *IEEE Transactions on Software Engineering*, 38(2):425–438, 2012.

[19] V. Lenarduzzi. Could social factors influence the effort software estimation? In *Proceedings of the 7th International Workshop on Social Software Engineering*, pages 21–24. ACM, 2015.

[20] J. López-Martínez, R. Juárez-Ramírez, A. Ramírez-Noriega, G. Licea, and R. Navarro-Almanza. Estimating user stories complexity and importance in scrum with bayesian networks. In *World Conference on Information Systems and Technologies*, pages 205–214. Springer, 2017.

[21] J. López-Martínez, A. Ramírez-Noriega, R. Juárez-Ramírez, G. Licea, and S. Jiménez. User stories complexity estimation using bayesian networks for inexperienced developers. *Cluster Computing*, pages 1–14, 2017.

[22] O. Malgonde and K. Chari. An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*, pages 1–39, 2018.

[23] J. McIver and E. G. Carmines. *Unidimensional scaling*, volume 24. Sage, 1981.

[24] E. Mendes. *Cost Estimation Techniques for Web Projects*. IGI Global, Hershey, PA, USA, 2007.

[25] E. Mendes. Knowledge representation using bayesian networksa case study in web effort estimation. In *Information and Communication Technologies (WICT), 2011 World Congress on*, pages 612–617. IEEE, 2011.

[26] T. Menzies, D. Port, Z. Chen, J. Hihn, and S. Stukes. Validation methods for calibrating software effort models. In *Proceedings of the 27th international conference on Software engineering*, pages 587–595. ACM, 2005.

[27] A. Panda, S. M. Satapathy, and S. K. Rath. Empirical validation of neural network models for agile software effort estimation based on story points. *Procedia Computer Science*, 57:772–781, 2015.

[28] M. Q. Patton. *Qualitative research evaluation methods*. SAGE Publications, Inc; 3rd edition, 2003.

[29] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli. Estimating story points from issue reports. In *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, page 2. ACM, 2016.

[30] M. Robeer, G. Lucassen, J. M. E. van der Werf, F. Dalpiaz, and S. Brinkkemper. Automated extraction of conceptual models from user stories via nlp. In *Requirements engineering conference (RE), 2016 IEEE 24th international*, pages 196–205. IEEE, 2016.

[31] F. Sarro, A. Petrozziello, and M. Harman. Multi-objective software effort estimation. In *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*, pages 619–630. IEEE, 2016.

[32] S. M. Satapathy and S. K. Rath. Empirical assessment of machine learning models for effort estimation of web-based applications. In *Proceedings of the 10th Innovations in Software Engineering Conference*, pages 74–84. ACM, 2017.

[33] E. Scott and D. Pfahl. Using developers' features to estimate story points. In *Proceedings of the 2018 International Conference on Software and System Process*, ICSSP '18, pages 106–110, New York, NY, USA, 2018. ACM.

[34] J. Silva, D. Ramos, and M. S. Soares. On criteria to choose a content management system: A technology acceptance model approach. In *SEKE*, 2016.

[35] Y.-Y. Song and L. Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[36] B. Tanveer, L. Guzmán, and U. M. Engel. Understanding and improving effort estimation in agile software development: An industrial case study. In *Proceedings of the International Conference on Software and Systems Process*, ICSSP '16, pages 41–50, New York, NY, USA, 2016. ACM.

[37] B. Tanveer, L. Guzmán, and U. M. Engel. Effort estimation in agile software development: Case study and improvement framework. *Journal of Software: Evolution and Process*, 29(11):e1862, 2017.

[38] M. Tavakol and R. Dennick. Making sense of cronbach's alpha. *International journal of medical education*, 2:53, 2011.

[39] M. Usman, E. Mendes, F. Weidt, and R. Britto. Effort estimation in agile software development: A systematic literature review. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, PROMISE '14, pages 82–91, New York, NY, USA, 2014. ACM.

[40] L. G. Wallace and S. D. Sheetz. The adoption of software measures: A technology acceptance model (tam) perspective. *Information & Management*, 51(2):249–259, 2014.

[41] L. Wasserman. All of statistics: a concise course in statistical inference. 2013.

[42] R. J. Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[43] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[44] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

[45] S. Zahraee, M. K. Assadi, and R. Saidur. Application of artificial intelligence methods for hybrid energy system optimization. *Renewable and Sustainable Energy Reviews*, 66:617–630, 2016.

[46] H. Zahraoui and M. A. J. Idrissi. Adjusting story points calculation in scrum effort & time estimation. In *Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on*, pages 1–8. IEEE, 2015.