International Journal of Engineering Education Vol. 35, No. 1(A), pp. 195–208, 2019
Printed in Great Britain

0949-149X/91 $3.00+0.00
© 2019 TEMPUS Publications.

A Comparison of the Planning Poker and Team Estimation Game: A Case Study in Software Development Capstone Project Course*

MARKO POŽENEL and TOMAŽ HOVELJA
Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia.
E-mail: marko.pozenel@fri.uni-lj.si, tomaz.hovelja@fri.uni-lj.si

Effort estimation is a crucial part of software development projects. Despite the availability of several assessment techniques, accurate assessment still remains an extremely difficult task. Team Estimation Game is a relatively new estimation technique for agile software development methods that has not received significant attention from the scientific community despite its growing popularity between practitioners. In this paper, we attempt to bridge this gap by presenting the results of an empirical study with undergraduate students in which we compare Team Estimation Game with the more established Planning Poker technique. We mainly focus our analysis of the two techniques on the time needed for user story estimation and estimation accuracy. The results of the empirical study reveal that Team Estimation Game produces more accurate story estimates than Planning Poker. Additionally, we found that for the Team Estimation Game, estimation and planning skills of the development teams improve from Sprint to Sprint. Team Estimation Game proved to be a useful estimation method for agile projects within the capstone course. Furthermore, we have shown that the study can be successfully incorporated into a software engineering capstone course without hindering the teaching goals while retaining the validity of research goals.

Keywords: empirical study; capstone course; agile software development; scrum; planning poker; team estimation game

1. Introduction

The software community is still coping with high project failure rates [1–4]. IDC report (International Data Corporation) on improving IT project outcomes in 2009 [5] estimates that 25% of IT projects experience complete failure, while 50% of the projects require various reworks and 20% do not provide ROI (Return on Investment). In Gulla [2] it is stated that project management is the major cause for the IT failure. Alami A. [6] also analyzed possible reasons for project failures and one of the possible reasons found is also poor project management. Standish Group in its 2011 report [7] defines a project as a success, when it is on time, on budget, and with all planned features implemented. Gartner [8] reports that only 16.2% of projects meets aforementioned requirements, 52% projects meets the requirements partially and 31% are complete failures. Nasir et al. [4] made extensive literature survey of critical success factors that impact software projects. They state that five most critical success factors of software projects are clear requirements, realistic estimation of schedule and budget, and effective project management skills and methodologies applied by the project manager. While the numbers of cancelled projects might sometimes be exaggerated [3], there is still room for improvement.

According to the 2011 CHAOS report from the Standish Group [7] agile projects are successful three times more often than non-agile projects, and agile process is the right tool to fight software development project failure. However, El Emam et al. [3] state that despite a clear progress in recent years, estimation skills still remain a key challenge to IT projects, since the practitioners use the tools and techniques that might not be appropriate or they might simply not fully exploit the possibilities of the tools used. Agile software development methods like Scrum [9] offer the possibility to assess work efficiently as a team. Agile methods assume that the required functionality of the new system is described in a form of user stories. In order to prepare a release plan, the complexity of each user story is usually assessed in story points, where each story point represents one man day of work. Story assessment enables us to either determine the indicative release date or to customize the release plan contents according to the required date. Agile processes depend on accurate estimation of user stories in order to prioritize work and create meaningful release plans [10]. Therefore the improvement of estimation techniques is always a relevant issue for researchers and practitioners alike, as well as investigations of the estimation process to discover relevant accuracy factors for effort estimation [11, 10] and develop new techniques for effort estimation [12].

One of the most popular and researched effort estimation method in recent years is Planning Poker

* Accepted 25 September 2018.

[13, 14]. Planning Poker (PP) has been widely used in industry and is well covered in literature [15], there were also several studies about the Planning Poker in academic environment [16–18]. Recently, a new estimation method appeared, named Team Estimation Game (TEG), for which its proponents claim that it enables faster evaluation and more accurate estimates [12]. While the Planning Poker method was already the subject of some scientific studies [19–21], we were not able to find a systematic study about Team Estimation Game except some recommendations of agile practitioners who recommend TEG as a better assessment method [12, 22]. In order to better understand TEG, we wanted to further investigate it and compare it to the widely used PP method. We conducted an empirical study with students (ESWS), where we focused on assessment accuracy and possible differences in the assessment of these two popular methods. The comparison is based on empirical data collected in the study.

An empirical evaluation of innovations in the industrial environment is sometimes difficult. Empirical studies typically require a lot of time and resources, developers are often overburdened with the existing work and may not be motivated to perform additional tasks. For this reason, we can help with studies that involve students as test subjects. In general, empirical studies with professionals are better accepted by researchers and practitioners as ESWSs. ESWSs are often viewed skeptically due to the lack of external validity: (i) students who may not fully embody professional developers are used as subjects (ii) research is done on smaller projects rather than full-size industrial projects [23]. Nevertheless, such studies—if they are conducted in the right way and address relevant validity threats—can significantly contribute to the understanding of new techniques, methods and processes [24] and help convince professionals to participate in future studies [25]. In order to maximize research and pedagogical value of our study, we applied requirements for successful ESWS [23] proposed by Carver et al. We analyzed the study requirements, carefully planned, integrated into the course and finally implemented the study. The complete plan of the study is presented in Poženel et al. [26].

The study was conducted within the scope of software engineering capstone course that takes place in the second semester of the final year of undergraduate studies. The capstone course consists of 15 weeks and requires students to work in project teams where they develop a project assignment strictly following the Scrum method [16, 27]. The course consisted of 18 student teams, of which 12 teams voluntarily participated in the study. All student teams were working on the same project that was prepared in the cooperation with the industry. During the course, data about story estimates and time needed for estimation were systematically gathered in order to enable the identification of possible differences between the studied agile estimation methods, i.e., Planning Poker and Team Estimation Game.

The paper is organized as follows. Section 2 first highlights both popular agile estimation techniques that are used in our study, and then defines the Research Questions. In Section 3 study design and capstone course are presented. Section 4 is dedicated to the key measures used. Section 5 presents the results, which are further discussed in Section 6. Validity concerns are addressed and described in Section 7. Section 8 concludes the paper.

2. Planning poker and team estimation game

An important part of agile development is effort estimation (or work estimation). It represents the central activity of agile planning processes in which estimation of the complexity of user stories, including technical complexity, time and possible difficulties bound to user stories are estimated. Effort estimation represents the best guess about the amount of work that has to be invested in the task, user story or a project in order to complete it. In the past, this task proved to be quite challenging in software projects.

One of the most commonly used agile estimation techniques is Planning Poker. Planning Poker is a group estimation technique where all team members are involved and they actively contribute to the final score. The assessment unit is a user story. The complexity of user stories are estimated in story points, where points are integer values arranged according to a Fibonacci like scale. The development team agrees on the assessment of individual user stories through PP rounds, where they communicate with the customer (i.e., the Product Owner) to get necessary data.

The team starts with a product backlog containing user stories that have to be assessed (and later implemented). User Stories are estimated one by one. Team members discuss them and ask the Product Owner for clarifications where needed. Each team member then expresses his personal estimation of the required effort and waits for all others to do the same. In order to avoid anchored estimates, they reveal individual estimations expressed in story points at the same time. If they agree, the estimation for the user story is complete. If the estimates differ, they discuss the given estimates. Especially team members with the lowest and the highest estimations must explain their assessment in detail. After the discussion is finished, the team members again express their individual estimations and reveal them to the group at the same time. The process is repeated until they all agree about the user story estimate. The estimation process is complete when all remaining user stories from the Product Backlog go through the same process [21].

While PP has proven to be a good effort estimation technique it still has some shortcomings, one of the most crucial is that the activity can be relatively time-consuming [22]. To solve the existing issues with PP different new estimation techniques were developed. One of a relatively new team estimation techniques is Team Estimation Game (TEG) that academics have not yet studied in detail. The aim of TEG is to balance the effort and time spent for estimation with the final result. It has short, intuitive rules and does not need special setup. TEG focuses more on relative complexity rather than on exact complexity of individual user stories. It relies on the fact that it is easier for people to compare individual user stories among themselves even when they do not know all the aspects of the stories involved in such comparison. TEG also helps assessors to not get lost in details, which can present a risk for successful estimation. Thus certain authors claim that TEG requires less effort than estimation techniques that splitting user stories to tasks [28]. Splitting user stories into tasks requires additional time where developers may not have all the necessary information to estimate user story more accurately. The estimation of user stories using relative estimation enables time-saving while retaining comparable quality estimation results [22]. Additionally, it is suggested that TEG better considers effort proportions among different user stories by providing the “whole picture” [22]. Since TEG can be presented as a game, it encourages active participation in the assessment process. Also, new team members that know TEG method and do not know the specific domain can be integrated directly into the estimation process [28].

TEG consists of two stages: (i) an arrangement of user stories from product backlog on a display area (e.g., panel, surface) relative to the required effort to complete (from low to high) (ii) an assignment of story points to each group of user stories. Similar to PP, the team starts with the Product Backlog full of unassessed user stories. The team first prepares a playing surface to place the cards and then draws columns on the playing surface. A playing surface is a visible and easily accessible area (a white board, large table or floor) for rearrangement and placement of individual user stories. The TEG process begins when the first team member takes the first user story card from the product backlog, reads it, mentally assess’ it, and places it on the playing surface. Then the second team member takes and reads the next user story card from the Product Backlog and positions it on the playing surface relative to the previously placed card depending upon whether it is more or less complex. Less complex user stories are placed in columns left to the first card, while more complex stories are placed in columns right to the first card. If the user story complexity is the same as the first one, it is placed in the same column below the first card. The next team member then either: (i) selects, considers and positions the next story card from the Product backlog, (ii) moves an existing user story card on the playing surface to different column, (iii) selects Pass (no action). Team members then repeat previous steps until the Product Backlog is empty and no team member wishes to move any of already positioned user story cards on the playing surface (they all select Pass).

After the first step user story cards are arranged by effort in relation to each other from the smallest to the biggest. In the stage two, estimation units for columns have to be determined. As with PP, points are used as estimation units and the same point sequence is used. The team members work together to assign points to each column to indicate the effort needed to complete user stories that are in that column. Using a round robin approach, each team member can either estimate the required effort for a non-estimated column or change an existing estimate of a column. The process is repeated until they all agree on the column estimates [12].

Despite many advantages, TEG also has some drawbacks. If the estimation team is larger (more than nine people), one round can take several minutes. In such situations processing of user stories may not be fast enough. Without sufficient professional background of the team members, the estimates may also become very speculative. Reinold [28] states that once introduced, the TEG method can be difficult to replace. Other estimation methods use different reference variables and valuation metrics [28].

PP and TEG are both group estimation techniques and they are both suitable for effort estimation in agile software development [22, 29, 30]. Bang states that Planning Poker is an efficient way to do high-level estimation. Haugen [30] suggests that Planning Poker can improve estimation performance compared to an unstructured group estimation process. Both estimation methods ensure that everyone in the project team understands the requirements behind the user stories. PP focuses on each user story individually while TEG focuses more on relations between user stories. PP gives all focus and attention to one user story at a time, thus enabling an in-depth discussion and better assessment. Important part of both assessment methods is good team communication that enables the sharing of team member opinions. It ensures that every team member understands the requirements for a user story and needed effort to implement it. Since they all have to agree on estimates PP and TEG eliminates the excessive influence of individuals. If the product backlog contains lots of user stories, effort estimation using PP can be long and tedious.

3. Study design

Given the above background information, we believe that an empirical study including both assessment techniques would improve the understanding of TEG's strengths and weaknesses. With additional knowledge we could identify individual benefits of these two techniques and be able to choose the appropriate technique for a software project. To the best of our knowledge, an empirical study that compares PP and TEG methods has not been published yet. Thus empirical evidence is sorely needed to confirm or refute the hypotheses about the advantages of TEG.

To study this problem domain, we conducted an empirical study that involved 12 Scrum Teams of students. We considered and addressed validity concerns that arise from empirical software engineering. In order to be able to compare estimates of different project teams and obtain statistically relevant results, student teams were working on the same software project, using the same tools and procedures. We used students as test subjects, since in an industrial environment it would be almost impossible to find a setting, where many teams work on the same set of user stories.

3.1 Research questions

The study was conducted within the capstone course in software engineering in the final year of undergraduate study. The aim of our study was to evaluate the TEG method. We evaluate this method by answering key research questions established in literature as crucial for the evaluation of estimation methods [16, 20, 21, 31]. The research questions are:

- RQ1: Does assessment process using Team Estimation Game take less time than using Planning Poker?
- RQ2: Are user story estimates using Team Estimation Game more accurate than estimates using Planning Poker [20, 21]?
- RQ3: Does the assessment accuracy (BRE) improve from Sprint to Sprint [16]?

The aim of the Research question RQ1 is to determine which method enables faster assessment of the Product backlog. Since effort estimation is not seen as an activity that delivers value to the customer (customer does not attain a tangible or intangible benefit), it should be as time efficient as possible. Björsne et al. [22] state that Planning Poker is a relatively time consuming estimation technique. Thus according to practitioners [22, 28], it is expected that estimation process using TEG takes less time than using PP. The purpose of the second research question (RQ2) is to determine whether the assessment accuracy of TEG remains the same (or even improves) as with Planning Poker, despite TEG allegedly being a faster method. A similar study focusing on Planning Poker was conducted by Moløkken [20]. The aim of the third question (RQ3) is to explore if the team's ability for more accurate assessment improves during the project from Sprint to Sprint. The third research question is similar to the research questions in a study conducted by Mahnič [16]. For the Planning Poker method the authors conclude that the ability to plan improves from Sprint to Sprint. According to the findings, it is expected that assessment accuracy increases from Sprint to Sprint for both methods.

By answering these questions we can comprehensively evaluate both TEG and PP of how fast and accurate they are, and how much potential for improvement they have.

In order to study the differences between PP and TEG and get statistically significant results, it was necessary to conduct a study with a sufficiently large number of teams working on the same project. Our study setup involved 12 student teams who developed Web-based application for supporting patronage service. The patronage service application included multiple user groups (e.g., patients, doctors, nursing sisters, head nursing sisters). It covered support for scheduling patients, more efficient organization and time management of nurses, some basic data analysis and anticipated basic application maintenance module for proper functioning of the application. Each project team consisted of four members and acted as self-organizing and self-managing Scrum Team that independently communicated with the Product Owner and was collectively responsible for the project outcome. Since such studies require a large amount of time and resources that are usually not available in the industry, it is almost impossible to conduct such a study in a professional setting. To circumvent this limitation, we conducted Empirical study with students (ESWS) within the capstone course.

3.2 The Capstone course

The software engineering capstone course at the

University of Ljubljana is taught in the final semester of the final year of undergraduate study. It was first introduced in academic year 2008/2009 and was updated several times [32]. Within the capstone course, we have already conducted some ESWSs [16], [33, 34] where we gained necessary experience in conducting studies. Students are taught agile software development with an emphasis on Scrum in a near real-world environment. The course lasts 15 weeks, the emphasis is on practical work. The students are required to work in teams of four in order to develop working software solution based on product specifications defined by the actual Product Owner.

The course design is adapted for Scrum methodology. The semester is divided into four Sprints: Sprint 0 that serves as a preparatory Sprint before the project and three regular Scrum Sprints, where actual development takes place. The first Sprint (Sprint 0) lasts two weeks, while the actual three Scrum Sprints lasts four weeks. In Sprint 0, formal lectures take place where students learn Scrum, how to use User stories for project requirements specification and how to assess project work. In order to be able to start development in Sprint 1, students get acquainted with initial User stories in Product backlog that was prepared by the Product Owner. Within the Sprint 0 development teams are formed and each team prepares all the necessary steps (e.g., development environment) to be able to start with development at an early stage of the next Sprint. At the end of Sprint 0, each development team estimates the effort required to implement each user story in the Product Backlog and prepares project release plan.

Sprint 1, 2, and 3 are regular Scrum Sprints that have the same structure. At the beginning of each Sprint, Sprint Planning meeting takes place where the project teams together with the Product Owner prepare the Sprint Backlog containing prioritized user stories for the current Sprint. During the Sprint, regular Daily Scrums take place where development teams maintain their Sprints Backlog and update project progress data supervised by the teaching instructors. At the end of each Sprint, Review meeting and Retrospective meetings take place. At the Sprint Review meeting project teams present their results to the instructors and to the Product Owner, who enforce the concept of "Done" for implemented user stories. At the Retrospective meeting teams and instructors discuss about development process in the past Sprint and propose possible improvements for the next Sprints. A more detailed description of the software engineering capstone course can be found in publications, published by Mahnič [16, 35].

In order to maximize the research and teaching goals, the ESWS was carefully designed and integrated in to the capstone course based on Carver's checklist [23] and past experience with conducting ESWSs [21].

4. Calculation of estimation accuracy

There are several ways to calculate estimation accuracy. We selected BRE (Balanced Relative Error) as the measure of estimation accuracy. BRE is proposed by Miyazaki et al. [36] and is calculated as shown in Equation 1:

$$\mathrm{BRE} = \frac{|\text{actual effort} - \text{estimated effort}\,|}{\min\,(\text{actual effort},\ \text{estimated effort})} \quad (1)$$

In the past, MRE (Magnitude of relative error of the estimate) [37], was widely used to measure the accuracy of effort estimations in software engineering but came under the community criticism [38], [36, 39, 40, 41]. The main concern when using MRE is uneven weighting of underestimates and overestimates [38]. The decision to use BRE over MRE as our key measure resulted from the fact that MRE is robust a more balanced measure [36].

In case of MRE underestimation cannot reach 1 or larger amounts, while overestimation has no upper limit, which can yield a very asymmetric distribution of the measure. This can cause uneven treatment of overestimated and underestimated stories, with underestimation being weighted insufficiently. BRE resolves this issue, evenly balancing overestimation and underestimation. It is a robust and sensible measure. Another reason for using BRE was compatibility with several studies that have been conducted in the past and use BRE as their key measure [42, 43], including related studies on PP [19–21]. Since BRE values failed to comply with normal distribution according to skewness (larger than ± 1) and kurtosis (larger than ± 2), we analyzed the BRE results with nonparametric tests instead of parametric tests.

All research questions required analysis of two independent samples. Therefore, the Mann-Whitney-Wilcoxon test [44] was used to answer all research questions. A two tailed p-value of 0.05 or less had to be reached in order to demonstrate that the distributions of the tested variables differ significantly from each other.

To evaluate the importance of the difference between tested variables, we additionally used effect size measure Cliff's delta. We chose the nonparametric measure of the effect size Cliff's delta (δ) [45], which is considered a robust and intuitive alternative to Cohen's d in situations where data are either non-normal, or are ordinal and therefore have reduced variance [46, 47].

Table 1. Basic descriptive statistics for RQ2 and RQ3

		BRE TEG			BRE PP		
		SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 1	SPRINT 2	SPRINT 3
N	Valid	107	88	48	180	149	95
	Missing	37	56	96	36	67	121
Mean		0.96	0.82	0.45	1.31	1.82	1.10
Median		0.50	0.50	0.28	0.61	0.92	0.60
Std. Deviation		1.30	1.13	0.55	2.44	2.88	1.33
Skewness		3.22	3.84	2.31	5.94	3.65	2.40
Std. Error of Skewness		0.23	0.26	0.34	0.18	0.20	0.25
Kurtosis		14.53	19.87	5.72	46.20	16.38	6.79
Std. Error of Kurtosis		0.46	0.51	0.67	0.36	0.40	0.49

5. Results

The study involved 12 student teams that were divided into two groups: 6 student teams estimated user story complexity by Planning Poker method while the remaining 6 student teams used Team Estimation Game for estimation of user stories. Two student teams were later removed from the Team Estimation Game test group due to drop out of some team members leaving us with 4 teams.

Product backlog that student teams were working on included 37 user stories, 36 regular and additional one to prepare working environment and to implement basic application framework. All user stories were divided into three different priority levels: 14 "must have" stories, 10 "should have" stories, and 13 "could have" stories. Student teams were asked to implement all must have stories and should have stories. For a higher grade students had to additionally implement several could have stories.

Basic descriptive statistics for the final student teams sample are presented in Table 1 and Table 2. For RQ2 and RQ3, the left side of the Table 1 presents BRE data for the Team Estimation Game for each Sprint, while the right side of the Table 1 presents data for Planning Poker method. Rows contain measurements from SPSS.

Table 2. Statistics for RQ1

		TEG	PP
N	Valid	12	18
	Missing	0	0
Mean		47.25	47.67
Median		39	41.50
Std. Deviation		29.19	19.92
Skewness		0.56	0.75
Std. Error of Skewness		0.64	0.54
Kurtosis		1.41	0.31
Std. Error of Kurtosis		1.23	1.04

5.1 RQ1: Does assessment process using Team Estimation Game takes less time than using Planning Poker?

To answer the research questions Mann-Whitney U test was performed. The test compared user story estimation time for Team Estimation Game and Planning Poker assessment technique. Basic descriptive statistics for RQ1 are presented in Table 2, while the test results are presented in Table 3.

In the Table 3 we can see that mean value for estimation time is approximately 47 minutes for both, TEG and PP teams. From the similar mean values it can already be apparent that the hypothesis will likely not be confirmed. Statistical test further showed that there is not statistically significant difference in the estimation time between Team Estimation Game and Planning Poker teams.

5.2 RQ2: Are user story estimates using Team Estimation Game more accurate than estimates using Planning Poker?

In order to answer research question RQ2, we tested both estimation methods in different Sprints and analyzed the differences between BRE statistics. A comparison between the two methods was performed by individual Sprints. The results are presented in Table 4, where each block of rows represents the results of one of the three Sprints.

In all three Sprints BRE median is lower for the Team Estimation Game method.

Table 3. Results for RQ1

	TEG = 1 PP = 2	N	Mean Rank	Sum of Ranks	Mann-Whitney U	Asymp. Sig. (2-tailed)	Cliff's d
TEG vs PP	1	12	14.29	171.50	93.50	0.54	–0.15
	2	18	16.31	293.50			
	Total	30					

Table 4. Results for RQ2

TEG vs PP	N	Mean BRE	Mean Rank	Sum of Ranks	Mann-Whitney U	Asymp. Sig. (2-tailed)	Cliff's d
Sprint 1 TEG	107	0.50	149.36	26884.50	8665.50	0.16	–0.10
Sprint 1 PP	180	0.61	134.99	14443.50			
Total	287						
Sprint 2 TEG	88	0.50	129.97	19365	4922.00	0.001	–0.25
Sprint 2 PP	149	0.92	100.43	8838			
Total	237						
Sprint 3 TEG	48	0.28	81.26	7719.50	1400.50	0	–0.39
Sprint 3 PP	95	0.60	53.68	2576.50			
Total	143		143				

In the first Sprint, the BRE median of Team Estimation Game (BRE = 0.50) was 18% smaller (i.e., more accurate) than BRE median of Planning Poker (BRE = 0.61). However, a statistical significant difference between the estimation methods was not confirmed. Cliff's delta of the tested difference was very small (0.10).

In the second Sprint, the Team Estimation Game BRE median (BRE = 0.50) was 45% smaller than Planning Poker BRE median (BRE = 0.92). Statistical test confirmed the significant difference between estimates of both methods (P-value = 0.001). The size of impact was small to medium (Cliff's d = –0.25). Therefore we can conclude, Team Estimation Game was more accurate in the second Sprint as Planning Poker.

Similar results as in the second Sprint, were also obtained for the third Sprint. The BRE median of Team Estimation Game (BRE = 0.28) was 54% smaller than BRE median of Planning Poker (BRE = 0.60). Statistical tests confirmed that difference in estimation accuracy was statistically significant (P-value > 0.001), the effect size measure was of medium size (Cliff's d = 0.39). Thus, we can conclude that for the third Sprint Planning Poker estimates were less accurate than Team Estimation Game estimates.

5.3 RQ3: Does the assessment accuracy (BRE) improve from Sprint to Sprint?

In order to answer the research question RQ3, we compared Each Sprint with those Sprints for each method that are chronologically before them. The results of statistical tests are presented in Table 5.

For the Team Estimation Game the accuracy of estimates between Sprint 1 (BRE = 0.50) and Sprint 2 (BRE = 0.50) did not improve. The difference was not statistically significant (P-value = 0.62) and also the Cliff's delta was very small (0.04). We can

Table 5. Results for RQ3

Method	Sprint	N	Mean BRE	Mean Rank	Sum of Ranks	Mann-Whitney U	P-value. (2-tailed)	Cliff's d
TEG - BRE	Sprint 1	107	0.50	99.81	10680	4514	0.62	0.04
	Sprint 2	88	0.50	95.80	8430			
	Total	195						
	Sprint 2	88	0.50	74.48	6554.50	1585.50	0.02	0.25
	Sprint 3	48	0.28	57.53	2761.50			
	Total	136						
	Sprint 1	107	0.50	84.35	9025	1889	0.01	0.26
	Sprint 3	48	0.28	63.85	3065			
	Total	155						
PP - BRE	Sprint 1	180	0.61	156.74	28212.50	11922.50	0.08	–0.11
	Sprint 2	149	0.92	174.98	26072.50			
	Total	329						
	Sprint 2	149	0.92	128.06	19081	6249	0.12	0.12
	Sprint 3	95	0.60	113.78	10809			
	Total	244						
	Sprint 1	180	0.61	138.22	24879.50	8510.50	0.95	0.01
	Sprint 3	95	0.60	137.58	13070.50			
	Total	275						

conclude that the estimates in Sprint 1 and 2 were equally accurate.

The comparison of Sprint 3 and Sprint 2 shows that estimates in the third Sprint were more accurate than estimates in the second Sprint. BRE median in the third Sprint (BRE = 0.28) is 45% smaller than BRE median in the second Sprint (BRE = 0.50). The results also show statistically significant difference between the Sprints (P value = 0.02). Cliff's delta of the tested difference is low to middle (0.25).

The comparison between estimates of Sprint 3 to Sprint 1 reveals similar findings to the above presented comparison between the Sprint 3 and the Sprint 2. BRE median in the third Sprint (BRE = 0.28) is lower than BRE median in the first Sprint (BRE = 0.50). Estimates in the third Sprint were 45% more accurate than estimates in the first Sprint. The differences between estimates of both Sprints were statistically significant (P-value = 0.01), while effect size is low to middle (Cliff's d = 0.26).

Based on the tests we can conclude that assessment accuracy for Team Estimation Game improves from Sprint to Sprint.

Next, we tested Sprint estimates between individual Sprints for Planning Poker. BRE median for Sprint 2 and Sprint 1 show that estimates in the second Sprint were 49% less accurate than estimates in the first Sprint (BRE median for the first Sprint = 0.61, BRE median for the second Sprint = 0.92). The accuracy of the estimates of Sprints 1 is not statistically different from Sprint 2 (P-value = 0.08) and the size of effect is low (Cliff's d = 0.11).

The comparison between estimates of Sprint 3 to Sprint 2 reveals that BRE median of the third Sprint (BRE = 0.60) was 65% smaller than BRE median of the second Sprint (BRE = 0.92). Therefore, estimates in the third Sprint were 65% more accurate than in the second Sprint. However, statistical tests found these differences neither statistically significant (P-value = 0.12) nor having a substantial effect size (Cliff's d = 0.12).

The last test shows results of the comparison between estimates of Sprint 3 and Sprint 1. The Planning Poker estimates in the third Sprint (BRE median = 0.60) were 2% more accurate than the Planning Poker estimates in the first Sprint (BRE median = 0.61). The results revealed that there is no statistically significant differences between the estimates (Asymp.Sig. = 0.95). Also Cliff's delta was very low (0.01).

In the case of Planning Poker statistical tests found no statistical difference between estimates of Sprints even though we may observe a weak tendency of improvement of the estimates accuracy from Sprint to Sprint. The only exception is the second Sprint, where assessment accuracy decreased. As it turned out in the interviews, this is because student teams under-estimated time in the first Sprint and under that impression they exaggerated in the second Sprint.

6. Discussion

In this section we will discuss the implications of our results on the paper's three research questions.

6.1 Does assessment process using Team Estimation Game takes less time than using Planning Poker (RQ1)?

In general, the difference between the average for TEG and PP is minimal. As a result, we could not find statistically significant difference between PP and TEG. During the study, the available data for the research question decreased due to the drop-out of some student groups. At the available volume of data, the difference between the TEG median and the PP median is too small to get a statistically significant difference. Therefore, this research question could not be statistically significantly answered. However, the tendency found in RQ1 is in line with [28, 22] that claim that estimation process using TEG takes less time than PP.

To the best of our knowledge no other study that compares PP and TEG according to time needed for estimation process has been published yet. TEG is a relatively new estimation technique. As of 2014, Usman et al. [48] performed systematic literature review on effort estimation in Agile Software Development and found out that Planning Poker is one of the most used estimation techniques, while Team Estimation Game was not mentioned at all. In order to compare findings and to further evaluate the RQ1, similar studies should be carried out in the future.

Possible explanation for why TEG is supposed to be a faster assessment technique is that it provides the Scrum Team with a better overview of the already discussed stories on the board and, therefore, the ability to evaluate the new user stories faster. Additionally, team members may find similarities with already assessed stories, which allows for shorter estimation time. On the other hand, using PP they might not be able to link the discussed user story to already discussed stories, which increases discussion and consequently assessment time. Post research interviews with the students appeared to show this to be the case.

As also noted in [21, 49], one possible explanation for the faster assessment with TEG is that students find administrative tasks like assessment process boring and less important. Using TEG, a free-rider [50] group member can hide more easily within the group without being noticed not providing value to the estimation process. In case of PP, team members are forced by the rest of the Scrum Team to put at least some effort in a group discussion. To limit the extent of such behavior, the instructors monitored the assessment process closely and tried to make sure all members of the group were actively involved.

6.2 Are user story estimates using Team Estimation Game more accurate than estimates using Planning Poker?

Regarding the observed difference in estimation accuracy between TEG and PP, as seen in Table 5, the test showed statistical difference in the second and the third Sprint. In the first Sprint, the difference is not statistically significant.

A similar study, comparing PP and individual expert estimates, was conducted by Moløkken-Østvold et al. [20]. They found that PP is in some cases more accurate than unstructured combination of estimates of a group. Their observation that PP influenced team members' work enabling them to see a problem from different perspectives and offer possible increase in estimation accuracy, is in line with our observations. TEG and PP team members believed that group discussion increased their estimation accuracy. This findings disagree with Armstrong's [51] claims, that face-to-face meetings harm forecasting and decision making. Moløkken-Østvold et al. [20] mentioned the lack of studies comparing PP with more structured techniques. Our study addresses this issue.

Another similar study, comparing PP estimates with statistical combination of individual estimates for students and professionals, was conducted by Mahnič and Hovelja [21]. They found that for professionals, PP estimates tended to be better than statistical combination of estimates. This is in line with our observations that PP increases discussion that allows for identification of hidden tasks, which may lead to better estimation accuracy. However, they found that for students, PP estimates were less accurate than statistical combination due to the possible inability of students to fully exploit group discussion. On the other hand, our discussion with the students showed that students believed that the group discussion helped them with a more accurate assessment.

A possible explanation for more accurate TEG is that TEG offers a comprehensive view of the user stories [26]. All user story cards are placed on the playing board and arranged relatively according to user story complexity. Team members also find it easier to compare relative complexity of one user story with another even if all details of the user story are not known [52]. That allows students to identify faster which user stories are similar and which user stories can share components. Since PP focuses on a single story at a time, the Team members' ability to compare different user stories is limited. Another possible reason for less accurate PP is that PP requires more effort than TEG. Using PP, if the number of user stories is high, the students might lose focus after some time and may start assessing stories with less care. Such behavior would be in line with Mahnič's claim alternative explanation of poorer PP performance could also lie in the observation [21] that highest and lowest estimates in the first round of PP act as strong anchoring point for the final estimate and thus lead to less accurate estimate. TEG avoids that by assigning estimates at the end when the Scrum Team has already discussed relations within user stories and estimates are assigned to a group of stories.

6.3 Does the assessment accuracy (BRE) improve from Sprint to Sprint?

With regard to RQ3, statistical tests presented in Table 5 indicate that (i) the assessment accuracy for TEG improves from Sprint 1 to 3 and from Sprint 2 to 3 and (ii) not for PP estimates.

Turning to TEG, we see that the average accuracy as measured by BRE from Sprint 1 and 2 to 3 has improved by 45%. One possible reason is that teams improved their assessment skills over time knowing the project and team better. In principle, team members had not known each other before the project started. They knew only about the topics they went through together in the study process. Their actual hands-on experience with Web development were varied. The more they knew each other, the better their group discussion was. Improved quality of the discussion in a development team enabled the identification of possible hidden tasks, which lead to more accurate assessment. However, the accuracy from Sprint 1 to Sprint 2 did not improve but remained the same. At this stage, we can only speculate that one Sprint is not enough time for the aforementioned reasons to take effect. Based on a discussion with students, the most likely reason is under-estimation of required effort in the first Sprint. In the second Sprint student teams took that dully into account in the assessment process. Unfortunately, they exaggerated in the opposite direction.

As far as Planning Poker is concerned, the results were not statistically significant. The reasons for possible improvements in assessment accuracy are equal to those mentioned in case of TEG, despite the fact that we were not able to prove it with data. PP does not provide a good general overview of all user stories as TEG does. Hence, it is harder to accurately assess user stories. A possible explanation might also be that two Sprints were not enough for students to master the Planning Poker to the level enabling improved assessment significantly. When comparing Sprint 1 and Sprint 2, one can notice that assessments had the tendency to be less accurate in the second Sprint. As with TEG, the discussion with students revealed that the reason is likely under-estimation in the first Sprint, and as a result over-estimation in the second Sprint.

Previous studies have shown that estimation and planning skills improve from Sprint to Sprint. In a capstone course study [49], the authors empirically evaluated students' progress in estimation and planning while using PP. They conclude that after three Sprints, actual achievements almost completely matched the planned. When we look at the RQ3, our study focused only on estimation accuracy between Sprints. For TEG, their findings are in line with our findings.

Tanveer et al. [11] conducted a study in industrial environment that investigates factors that affect accuracy of estimation and the estimation performance in agile development process. They found out that developer's knowledge, experience, and changes on the underlying system affect the estimation accuracy. This findings are in line with our findings for TEG teams. As students acquire experience from Sprint to Sprint, their assessment accuracy improves.

7. Limitations

We have taken into account all the requirements for a successful ESWS. To increase the validity of the study, the ESWS was designed and incorporated into the capstone course in accordance with recommendations, proposed by Carver et al. [23]. Carver presented a checklist that provides guidance for teachers and researchers with activities that must be performed in order to increase research and pedagogical outcomes of the ESWS as much as possible. The checklist consists of ten items (Table 6), where each item has a set of requirements that must be addressed. When designing and running the study, the requirements for each checklist item were satisfied to the greatest extent possible in order to increase study validity.

Table 6. Carver's [23] ESWS checklist

1	Ensure Adequate Integration of the Study into the Course Topics.
2	Integration the Study into the Course Schedule.
3	Reuse Existing Artifacts and Tools as Appropriate.
4	Write up a Protocol and have it reviewed.
5	Obtain Subjects' Permission for Their Participation in the Study.
6	Set Subject Expectations.
7	Document Detailed Information about the Experimental Context.
8	Implement policies for Controlling/Monitoring the Experimental Variables.
9	Plan Follow-up Activities.
10	Build or update a Lab Package.

The detailed design of the presented empirical study with students is presented in Poženel et al. [26]. In the following subsection we will instead focus on study validity.

Evidence based studies in software engineering must possess a certain degree of validity. Validity ensures that study is well defined and that results can be generalized to broader population of interest [53].

(i) *Internal validity*: To eliminate bias and the effect of non-essential variables, we monitored all factors of internal validity. Student teams were involved in the experiment at the same time, working on the same project. During the course of the project no changes were introduced. Except for the two student teams, that experienced problems with dropout of some team members due to inadequate contribution, team members did not change. In order to avoid effects on the study results, we removed those two teams from the study analysis at the end. Students' participation in the study was voluntary, and integrated in the course schedule, what improved motivation throughout the duration of the study and minimized mortality threat.

All student teams were exposed to pre-test training of all assessment methods and software tools. They were no differences in duration of training or tools used between student teams. We believe that such a pre-test training does not affect the study outcome. All student teams used the same tools to collect data. The data was collected automatically and in the same way. No changes were made during the project. We believe that automatic and timely data collection removed instrumentation threat. Concern, that self-selected group might differ too much in skills was redundant. All students previously attended the same classes where they acquired theoretical and practical knowledge, necessary for the project. The class discussion also revealed that team members rarely knew other team members before the project. All student teams worked on the same project, used the same working methods and used same tools for data collection. The only variable was the assessment method used. There was no reason any group felt marginalized or worked with less attractive tools. In order not to inadvertently affect the research, students did not know the research goals, while the student grade did not depend on the outcome of the study. Additionally, we used Mann-Whitney-Wilcoxon test to compare the grades of the student teams using the two different assessment methods. With a P-value of 0.29 we can state that there were no statistically significant differences in performance between the groups using different assessment methods. Such results support our statement that the only important variable was the assessment method used, since it is very likely that if strong variance in intragroup dynamics existed it would importantly affect group performance. The results for grade comparison are presented in Table 7.

(ii) *External validity*: In our ESWS special attention was paid to external validity. As far as generalization is concerned, professionals in various software companies may not get the same results as students. However, we believe that the trend in the results for PP and TEG would be similar. Students in the last semester of undergraduate university studies had all the necessary knowledge to participate in such a study. They previously acquired theoretical and practical knowledge from computer science field, while many students were already involved in various projects in software companies. We believe that findings can be generalized for problems that include team assessment of effort needed to complete the project.

However, ESWS are often viewed skeptically by researchers [54] and industrial environment because students do not produce the same results as professionals and the ESWS projects are usually smaller [23] and therefore ESWS suffer from external validity issues. On the other hand, also studies from specific environments often cannot be generalized without changes. Nevertheless, Carver et al. [23] state that ESWSs can be valuable to the industrial and research communities if they are conducted in an adequate way and take into account threats to internal and external validity. They analyzed criteria to assess usefulness of individual ESWS and listed cases where ESWS could be useful for wider software development community. In order to maximize external validity of our study, we followed requirements presented by Carver et al. [23].

(iii) *Construct validity*: It is focused on whether the theoretical constructs are interpreted and measured correctly [55] and is associated with external validity [53]. Student grades were based on faithful implementation of the activities that were required from all teams enrolled in the capstone course. They were informed that grade depends on conformance to development process and data quality and not from the study results, what minimized potential danger of tendency to guess the hypothesis and confirm it. Student teams were not under any different pressure than Scrum Teams in software companies when they estimate time complexity in order to plan the project work what minimized evaluation apprehension. The researchers were aware of the threat that they can unintentionally influence the student's behavior. Therefore, there was no interaction about study goals.

For complete and reliable collection of data we used proven and tested tools like Scrum project management tool ACScrum [35], SPSS and Microsoft Excel. We benefited from similar studies that were conducted in the past [27], where proven tools for automated estimation process for PP and TEG were developed. Threats to construct validity were also lowered with the fact that the researchers understand the environment where the study was conducted. Construct validity was also increased with the fact that ten student teams were working on the same project resulting in the estimates that are directly comparable. In the literature, some other threats to construct validity can be found [56], but we believe that they are not of any importance to this study.

(iv) *Conclusion validity*: In order to derive statistically correct conclusions based on the collected data, we used balanced measure of relative error (BRE) to calculate estimation accuracy. To analyze differences between the two distributions (PP and TEG) we used non-parametric Mann-Whitney-U tests. For measuring the effect size we used non-parametric measure Cliff's delta, which is used in situations where data are non-normal, ordinal and therefore have reduced variance [46, 47].

In this study, two estimation techniques were compared in order to further highlight new method TEG that is becoming very popular among practitioners. We provided a reasonable explanation that TEG gives more accurate estimates than PP. For TEG, we provided evidence that the accuracy of estimates improves from Sprint to Sprint. Our study could serve as a basis for replication study in different environments what would increase external validity of our study. Complete study design including lab package was presented in Poženel et al. [26]. It could also serve as a stepping stone to further research, how TEG is performing in an industrial environment or in

Table 7. The results of grade comparsion

	TEG = 1 PP = 2	N	Mean Rank	Sum of Ranks	Mann-Whitney U	Wilcoxon W	Z	Asymp. Sig. (2-tailed)
TEG vs PP	1	4	4.25	17	7.00	17.00	–1.07	0.29
	2	6	6.33	38				
	Total	10						

environments where PP did not perform as expected.

(v) *Reliability*: With the well-defined case study protocol based on a checklist for conducting ESWS [23, 26], automatic data collection, careful preparation and monitoring of each step, we increased the repeatability of findings and mitigated this threat. With automatic data collection we minimized potential problem where different researchers recorded observed behavior differently. A lab package allows the study replication in a different setting.

8. Conclusions

While Planning Poker is a very popular agile estimation technique, new techniques are emerging that claim to be better. As such, proponents offer Team Estimation Game as a better alternative to PP. The aim of our study was to compare these two agile estimation techniques focusing on accuracy of produced estimates and the time needed for producing these estimates.

The results of our study show that TEG produces, in most cases, more accurate story estimates than PP. This confirms our research hypothesis RQ2.

With regard to accuracy improvements (RQ3), the results of the study confirm that in most cases, TEG estimation accuracy improves from Sprint to Sprint. This was shown in the second and third Sprint, where TEG was a more accurate estimation method than PP. Additionally, TEG accuracy improved from the second to the third Sprint and from the first to the third Sprint. In case of PP, we found no improvement of estimation accuracy between Sprints and could not confirm the research question.

Unfortunately, we were not able to confirm the hypothesis that Team Estimation Game is less time consuming assessment technique that Planning Poker (RQ1). The results show that there is no significant difference between the two methods in terms of time spent to assess user stories.

Regarding the external validity, the results can be generalized to the population of students and young professionals in the field of Computer Science and Informatics. The results may also be valid for professionals who are not experienced in story estimation in agile software development, however, a follow up study focusing of just experienced professionals would be very beneficial.

We hope that our paper will motivate other researchers to replicate the study in different settings to confirm or complement our findings. Possible future replications of the study could be done on a different group of undergraduate students at a similar undergraduate study program with different constraints. A valuable study replication would include graduate students in a master's study program of Computer science who are usually already part-time employed and have even more experience in software development. It would also be important to replicate the study with practitioners in the industry. Finally, we also believe that it is worth investigating how TEG compares to other agile estimation techniques.

References

1. K. R. Linberg, Software developer perceptions about software project failure: a case study, *Journal of Systems and Software*, **49**(2), 1999, pp. 177–192.
2. J. Gulla, Seven reasons why information technology projects fail, in *SHARE Conference*, Orlando, USA, 2011.
3. K. El Emam and A. G. Koru, A replicated survey of IT software project failures, *IEEE software,* **25**(5), 2008, pp. 84–90.
4. M. H. N. Nasir and S. Sahibuddin, Critical success factors for software projects: A comparative study, *Scientific research and essays*, **6**(10), 2011, pp. 2174–2186.
5. J. C. a. W. D. Pucciarelli, Improving IT Project Outcomes by Systematically Managing and Hedging Risk, An IDC Insights Research Document, 2009.
6. A. Alami, Why Do Information Technology Projects Fail?, *Procedia Computer Science*, **100**, 2016, pp. 62–71.
7. C. Manifesto, The laws of CHAOS and the chaos 100 best PM practices, in *The Standish Group International*, Boston, 2011.
8. L. Mieritz, Gartner Survey Shows Why Projects Fail, 1 June 2012. [Online]. Available: http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/. [Accessed 1 July 2015].
9. K. Schwaber, Agile project management with Scrum, Redmond, WA: Microsoft press, 2004.
10. [F. Raith, I. Richter, R. Lindermeier and G. Klinker, Identification of inaccurate effort estimates in agile software development, in *Software Engineering Conference (APSEC), 2013 20th Asia-Pacific*, Bangkok, Thailand, 2013.
11. 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*, New York, NY, USA, 2016.
12. H. L. Johnson, How to play the Team Estimation Game, 8 May 2012. [Online]. Available: http://www.agilelearninglabs.com/2012/05/how-to-play-the-team-estimation-game. [Accessed 18 July 2018].
13. J. Grenning, Planning Poker or how to avoid analysis paralysis while release planning, *Hawthorn Woods: Renaissance Software Consulting,* **3**, 4 2002, pp. 22–23.
14. K. Moløkken-Østvold and M. Jørgensen, Group processes in software effort estimation, *Empirical Software Engineering*, **9**(4), 2004, pp. 315–334.
15. M. Cohn, Agile Estimating and Planning, Englewood Cliffs, New Jersey: Prentice Hall, 2005.
16. V. Mahnič, A capstone course on agile software development using Scrum, *IEEE Transactions on Education,* **55**(1), 2012, pp. 99–106.
17. M. Paasivaara, V. Heikkilä, C. Lassenius and T. Toivola, Teaching students Scrum using LEGO blocks, in *Companion Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, India, 2014.
18. G. Rodriguez, Á. Soria and M. Campo, Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum, *Computer Applications in Engineering Education,* **23**(1), 2015, pp. 147–156.
19. K. Moløkken-Østvold and N. C. Haugen, Combining estimates with Planning Poker—An empirical study, in *Software Engineering Conference, 2007. ASWEC'2007, 18th Australian*, Melbourne, Australia, 2007.

20. K. Moløkken-Østvold, N. C. Haugen and H. C. Benestad, Using Planning Poker for combining expert estimates in software projects, *Journal of Systems and Software*, **81**(12), 2008, pp. 2106–2117.
21. V. Mahnič and T. Hovelja, On using planning poker for estimating user stories, *Journal of Systems and Software,* **85**(9), 2012, pp. 2086–2095.
22. T. Bjorsne, I. Kostial and K.-D. Schmatz, Die Alternative zum Planning Poker: Das Team Estimation Game, Berleb Media GmbH, 2013. [Online]. Available: https://www.projektmagazin.de/artikel/die-alternative-zum-planning-poker-das-team-estimation-game_1077797. [Accessed 20 September 2018].
23. J. C. Carver, L. Jaccheri, S. Morasca and F. Shull, A checklist for integrating student empirical studies with research and teaching goals, *Empirical Software Engineering*, **15**(1), 2010, pp. 35–59.
24. J. Carver, L. Jaccheri, S. Morasca and F. Shull, Issues in using students in empirical studies in software engineering education, in *Software Metrics Symposium, 2003. Proceedings. Ninth International*, Sydney, Australia, 2003.
25. W. F. Tichy, Hints for reviewing empirical work in software engineering, *Empirical Software Engineering,* **5**(4), 2000, pp. 309–312.
26. M. Poženel and V. Mahnič, Studying agile software estimation techniques: the design of an empirical study with students, *Global Journal of Engineering Education*, **18**(2), 2016, pp. 53–58.
27. V. Mahnič, The capstone course as a means for teaching agile software development through project-based learning, *World Transactions on Engineering and Technology Education*, **13**(3), 2015, pp. 225–230.
28. D. Reinold, Team Estimation Game, Berleb Media GmbH, 2015. [Online]. Available: https://www.projektmagazin.de/methoden/team-estimation-game/. [Accessed 19 July 2018].
29. T. Bang, An agile approach to requirement specification, *Agile Processes in Software Engineering and Extreme Programming*, 2007, pp. 193–197.
30. N. C. Haugen, An Empirical Study of Using Planning Poker for User Story Estimation, in *Agile Conference '06*, Minneapolis, MN, USA, 2006.
31. K. Moløkken and M. Jorgensen, A review of software surveys on software effort estimation, in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003, Proceedings*, Rome, Italy, 2003.
32. V. Mahnič, Teaching Scrum through team-project work: Students' perceptions and teacher's observations, *International Journal of Engineering Education*, **26**(1), 2010, pp. 96–100.
33. V. Mahnič and T. Hovelja, The Influence of Diffusion of Innovation Theory Factors on Undergraduate Students' Adoption of Scrum, *International Journal of Engineering Education*, **32**(5A), 2016, pp. 2121–2133.
34. V. Mahnič and T. Hovelja, Teaching User Stories within the Scope of a Software Engineering Capstone Course: Analysis of Students' Opinions, *International Journal of Engineering Education*, **30**(4), 2014, pp. 901–915.
35. V. Mahnič and A. Časar, A Computerized Support Tool for Conducting a Scrum-Based Software Engineering Capstone Course, *International Journal of Engineering Education*, **32**(1), 2016, pp. 278–293.
36. Y. Miyazaki, A. Takanou, H. Nozaki, N. Nakagawa and K. Okada, Method to estimate parameter values in software prediction models, *Information and Software Technology,* **33**(3), 1991, pp. 239–243.
37. L. C. Briand and I. Wieczorek, Resource estimation in software engineering, in *Encyclopedia of software engineering*, New York, John Wiley & Sons, 2002, pp. 1160–1196.
38. T. Foss, E. Stensrud, B. Kitchenham and I. Myrtveit, A simulation study of the model evaluation criterion MMRE, *IEEE Transactions on Software Engineering,* **29**(11), 2003, pp. 985–995.
39. M. Jørgensen, A critique of how we measure and interpret the accuracy of software development effort estimation, in *First International Workshop on Software Productivity Analysis and Cost Estimation. Information Processing Society of Japan*, Nagoya, 2007.
40. B. A. Kitchenham, L. M. Pickard, S. G. MacDonell and M. J. Shepperd, What accuracy statistics really measure, *IEE Proceedings-Software*, **148**(3), 2001, pp. 81–85.
41. M. Shepperd, M. Cartwright and G. Kadoda, On building prediction systems for software engineers, *Empirical Software Engineering,* **5**(3), 2000, pp. 175–182.
42. K. Moløkken-Østvold and M. Jorgensen, A comparison of software project overruns-flexible versus sequential development models, *IEEE Transactions on Software Engineering,* **31**(9), 2005, pp. 754–766.
43. K. Moløkken-Østvold and K. M. Furulund, The relationship between customer collaboration and software project overruns, in *Agile Conference (AGILE 2007)*, Washington, USA, 2007.
44. H. B. Mann and D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, *The Annals of Mathematical Statistics*, **18**, 1947, pp. 50–60.
45. N. Cliff, Ordinal Methods for Behavioral Data Analysis, New York: Psychology Press, 2014.
46. M. R. Hess and J. D. Kromrey, Robust confidence intervals for effect sizes: A comparative study of Cohen's d and Cliff's delta under non-normality and heterogeneous variances, in *Annual Meeting of the American Educational Research Association*, San Diego, April 12–16, 2004.
47. R. D. Ledesma, G. Macbeth and N. U. R. I. A. Cortada de Kohan, Computing effect size measures with ViSta-the visual statistics system, *Tutorials in Quantitative Methods for Psychology*, **5**(1), 2009, pp. 25–34.
48. 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*, New York, NY, USA, 2014.
49. V. Mahnič, A case study on agile estimating and planning using Scrum, *Elektronika ir Elektrotechnika,* **111**(5), 2011, pp. 123–128.
50. P. Levin, Running group projects: dealing with the free-rider problem, *Planet*, **9**(1), 2003, pp. 7–8.
51. J. S. Armstrong, How to Make Better Forecasts and Decisions: Avoid Face-to-Face Meetings, *Foresight: The International Journal of Applied Forecasting*, **5**, 2006, pp. 3–15.
52. S. Bockman, Team Estimation—An Alternative (Often Better) Estimation Method than Planning Poker, Berleb Media GmbH, 2009. [Online]. Available: http://www.netobjectives.com/files/team-estimation-game.pdf. [Accessed 6 December 2017].
53. T. D. Cook, D. T. Campbell and A. Day, Quasi-experimentation: Design & analysis issues for field settings, **351**, Boston: Houghton Mifflin Company, 1979.
54. M. Höst, B. Regnell and C. Wohlin, Using students as subjects - a comparative study of students and professionals in lead-time impact assessment, *Empirical Software Engineering*, **5**(3), 2000, pp. 201–214.
55. F. Shull, J. Singer and D. I. K. Sjøberg, *Guide to advanced empirical software engineering*, **93**, Berlin, Heidelberg: Springer-Verlag, 2008.
56. C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell and A. Wesslén, Introduction to Experimentation in Software Engineering, Boston, MA: Kluwer Academic Publishers, 2000.

Marko Poženel is a Teaching Assistant at the Faculty of Computer and Information Science at the University of Ljubljana, Ljubljana, Slovenia. His teaching and research interests include agile software development methods, empirical software engineering as well as Web data mining and user behavior analysis. He received his PhD in Computer Science from the University of Ljubljana in 2010.

Assist. Prof. Dr Tomaž Hovelja received his bachelor's degree, master's degree and his PhD in Business Administration from the Economic Faculty at the University of Ljubljana. He is employed as an assist. prof. at the Faculty of Computer and Information Science at the University of Ljubljana. His research areas are social, economic and organisational factors of IT deployment in enterprises, IT projects success criteria and management and organization of the software development process. His research has appeared in journals such as Journal of Systems and Software, Computer Science and Information Systems, Technological and Economic Development of Economy.