FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 43 (2018) No. 4
ISSN 0867-6356
DOI: 10.1515/fcds-2018-0016 e-ISSN 2300-3405

Combining Data Analytics with Team Feedback to Improve the Estimation Process in Agile Software Development

Antonio Vetrò [*], Rupert Dürre [†], Marco Conoscenti [‡]
Daniel Méndez Fernández [§], Magne Jørgensen [¶]

Abstract. We apply a mixed research method to improve the user stories estimation process in a German company following agile software development. We combine software project data analytics with elicitation of teams' feedback, identify root causes for wrong estimates and propose an improved version of the estimation process. Three major changes are adopted in the new process: a shorter non numerical scale for story points, an analogy-based estimation process, and retrospectives analyses on the accuracy of previous sprints estimates. The new estimation process is applied on a new project, and an improvement of estimates accuracy from 10% to 45% is observed.

Keywords: Agile estimation, data analytics, process improvement

1. Introduction and motivations

Software process improvement [8] refers to the set of activities that aim to improve the processes, tools and activities of software development and make the whole software project more successful. Project planning is a critical activity for software development, being studied in the scientific literature as a key aspect for the success of a software project (e.g., see [12], [19], [4], [14]), also in agile software development (e.g., [20]). In Scrum, for example, it is very important to make accurate effort estimation in the sprint planning phase [2]: in fact, the estimation of user stories [1]

[*]Nexa Center for Internet & Society, DAUIN, Politecnico di Torino, antonio.vetro@polito.it
[†]Netlight Consulting, rupert@duerre.de
[‡]Nexa Center for Internet & Society, DAUIN, Politecnico di Torino, marco.conoscenti@polito.it
[§]Technische Universität München, daniel.mendez@tum.de
[¶]Simula Metropolitan, magnej@simula.no
[1]A user story is a small, independent feature that deliver value [11]

are then used throughout the Scrum process, from selection of the user stories until updating the burn-down chart. User stories estimates can incur in two types of erroneous predictions: overestimation, where the estimated effort is higher than the actual effort needed, and underestimations, where the estimation is lower than the actual work needed. The impact of overestimation could be different depending on the type [13, p. 1]: for example, overestimations might lead to a reduced productivity rate [15], since the development team expands "the work so as to fill the time available for its completion" [16]. Furthermore, overestimations may result in the belief that the development of a feature or a product in general is not beneficial and, as a consequence, it will be erroneously rejected. Another risk of overestimations is that new opportunities to undertake new projects might be overlooked due to the belief that there is no capacity available. [13, p. 1]. Whereas accurate estimates may increase the productivity, large underestimations cause a lower quality of the product due to the resulting time pressure. [9, p. 27]. Moreover, underestimations might have detrimental effects on the project management, e.g. a wrong plan of the staffing of a development team or a wrong allocation of resources. These effects are often hard to handle: for example it takes time and education to render the added staff beneficial and productive [13, p. 1].

In this work we aim at contributing to the understanding of causes of wrong estimates in agile software development, and tested an improvement proposal targeted to a German software company which develops web applications with agile methodologies: the improvement methodology was based on software project data analytics and project team feedback. We collected quantitative information from the estimation process in four development projects, we analyzed them and involved the project teams in the interpretation of analysis results: we elicited explanations in meetings in which the quantitative results were visually summarized to them. The collected feedback allowed us a deep understanding of the causes of wrong estimates and, on the base of this knowledge, to propose three major changes in the estimation process: the use of a shorter and non-numerical scale for story points; an analogy-based mechanism for estimating new user stories; the introduction of retrospective meetings for retrieving lessons learned from the analysis of estimates accuracy in previous sprints. The new estimation process was applied on a fifth project, and we could observe an improvement of estimates accuracy from 10% to 45% with respect to the previous four analyzed projects.

Our main contribution is an improvement methodology to study estimation issues in agile software development projects. We also provide -on an online appendix- a project characterization schema that should serve as a guide for replications of this study in other contexts.

The remainder of the paper is structured as follows: we enunciate our study goal and research questions in Section 2, followed by metrics and measurement methodology in Section 3 and context description in Section 4. Then we report results in Section 5 and discuss them in Section 6. The new estimation process is introduced in Section 7, the results of tis applications reported in Section 8 and discussed in Section 9. We conclude the paper with the analysis of the study limitations and indication of further work in Section 10. We conclude with a short summary (Section 11).

2. Study Goal and Research Questions

The goal of our study is to improve the estimation process in an agile software development project. Table 1 shows a structured presentation of the goal according to the Goal-Question-Metric template [1]. In relation to the goal, we define six research questions.

Study Goal	Object of Study	Effort estimations
	Purpose	Improve
	Focus	estimation process
	Stakeholder	development teams and products owners
	Context Factors	Software development company developing web applications with agile methods

Table 1. Structured presentation of the study goal

The first two questions investigates the presence of trends in the estimates over the course of the projects: if present, trends have a systematic impact on the accuracy of estimations, that should be taken into account when interpreting the causes of such errors.

RQ1: Are estimates getting more similar during the projects? This research question investigates whether the range of story points used during the sprints is similar throughout the course of the project or not. [2].

RQ2: Are estimates getting smaller or bigger during the projects? The research question investigates whether estimates have a tendency to increase or decrease over the course of a project. This could be consequence of the increasing or decreasing complexity of user stories in the backlog (e.g., bigger user stories in the beginning, and smaller towards the end), or could reveal a tendency of the team in systematically underestimating or overestimating user stories with the progress of the project. Since this question only considers the estimated effort assigned to a user story and not the actual effort needed for its implementation, the results from this research question should be interpreted in light of the results of the following RQs, which involve the actual implementation effort.

The next research questions involve the actual implementation effort of the user stories, which is necessary to: compute the accuracy of the estimates (RQ3); understand whether the relative value of a story point is constant (RQ4); investigate on

[2]A special case occur when group dynamics lead people to estimate in the middle to avoid explaining their estimates (especially the lowest and highest), which is required in planning poker in case estimates differ consistently: this is a case where group dynamic effects dominate the wish to be as accurate and independent as possible. However, we only checked against the variance, see the Section 3

the ability of the team to improve its estimations during the process (RQ5); identify the inaccurate estimates (RQ6). Herein we briefly motivate and explain these four research questions.

RQ3: How accurate are the estimates? We observe the relationship between the estimated and the actual effort, which should linearly correlate: in fact the definition of story points states that the implementation time per story point should be similar between the story point categories, and proportional between different categories (e.g., a 5-point user story is expected to take only half the time of a 10-point user story).

RQ4: Does the effort per story point vary over the course of the project? This research question checks whether the "value" of a story point, i.e. the time spent for a story point, changes over time. In fact, as SCRUM guidelines show, both iteration and release planning often use the velocity as a central measurement. The velocity can be described as the amount of work the development team can complete within one iteration. For example, a team that has an average velocity of 30 story points, can develop as many user stories in one iteration as long as the summed story point values do not exceed 30 story points. However, this approach might be problematic if the story points in a project are not linear, e.g. if six 5-point user stories do take much longer than three 10-point user stories.

RQ5: Does the accuracy of estimates improve during the project? In this research question we investigate whether the performance of the teams in estimating improve during the project. In fact, continuous improvement is often regarded as one of the cornerstones of agile software development: for example, Scrum defines a retrospective meeting in which the team members examine the people, relationships, processes, and tools in order to identify possible problems and their causes; then the team looks for a plan to improve the process and their work in general.

RQ6: How many estimates are inaccurate? We identify the wrong estimates to compute the ratio of inaccurate estimates in each project to discuss them with the projects teams: correctly detecting the wrong estimates is a necessary step to analyse them and understand their causes.

3. Metrics definition

Herein we list the metrics used to answer the research questions and briefly explain them.

RQ1: Are estimates getting more similar during the projects?

M1 Linear regression

- x=Sprint number
- y=variance/CoV

Explanation. We computed the variance and the coefficient of variation of the story point estimates of the user stories sprint-wise (we ordered user stories by their sprint membership). The variance measures the variability from the averagely used story points and therefore gives an indication of the range of story points used. In addition to the variance, we compute also the coefficient of variation (CoV), which is a standardized measure of dispersion of a distribution and is defined as the ratio of the standard deviation σ to the mean μ. The CoV defines the dispersion of the story points in a way that does not depend on the variable's measurement unit as heavily as the variance does (as an example, for the distributions {1,2,3,5} and {5,8,13,20} the difference in the variance is 41, while it is 0.05 for the coefficient of variation, despite a big change in the mean value). CoV has been used for studying performance time predictions (e.g., [5]) and on judgement-based predictions of performance time [6]. The CoV is independent of the unit in which the measurement has been taken and can be used for comparison between data sets with widely different means [21]. Finally, we apply a linear regression in order to combine the calculated values as dependent variables with the sprints as a measurement of the project progress, as explanatory variable (in the form CoV (or Var) = a + b*Sprint_number).

RQ2: Are estimates getting smaller or bigger during the projects?

M2 Spearman correlation coefficient between the story point value of all user stories and the sprint number

M3 Linear regression

- x=Sprint number
- y=Story Point

Explanation. With RQ2, we deepen our research of question 1 and observe whether we can identify a correlation between the story point value of a user story and the project progress. Similar to RQ1, in RQ2 the user stories are ordered by their sprint membership and a regression analysis is conducted with the sprint number as independent variable and the story points as dependent variables. We compute Spearman correlation coefficient [18, p. 396] . For further analysis of this research question we create a box plot showing the estimated effort distribution in each sprint.

RQ3: How accurate the estimates are?

M4 Spearman correlation coefficient [18, p. 396] between the actual and the estimated effort

M5 (Multiple) linear regression:

- x=Story point categories (estimated effort)
- y=Implementation hours (actual effort)

M6 p-values of the Mann–Whitney U test between the implementation times per story points of each story point category

Explanation To answer RQ3, we apply three metrics. Firstly, we compute the Spearman correlation coefficient between the total implementation time and the story points for all user stories. Secondly, we check whether the relative character of story points holds in practice. Thirdly, we calculate the multiple linear regression. Finally, since the definition of story points also states that the implementation time per story point should be similar between the story point categories, we pairwise compare the distributions of the implementation time per story points in the different story point categories by computing pairwise left-sided Mann-Whitney U tests [18, p. 293] to check whether the actual implementation time of user stories of size i is lower than the actual implementation time of user stories of size j, given that $j > i$.

The Mann-Whitney U test is a nonparametric test of the null hypothesis that two populations are the same against an alternative hypothesis, for example that one population tends to have larger values than the other. In contrast to comparable statistical tests like the t-test, the Mann-Whitney U test does not require a special distribution of the dependent variable in the analysis and can be applied to small samples. As a result, each Mann-Whitney U test delivers a p-value that can be interpreted for rejecting or accepting the null hypothesis that two distributions are the same: to be able to manage also categories with few data points without a too conservative approach, we apply level of $\alpha = 0.10$, which is considered strict enough also when rigor is relaxed in favor of pragmatism [7].

We provide the following example to ease the comprehension of the usage of this test. Consider user stories of size 3 and of size 5; we use the Mann Whitney test to check whether implementation times of user stories of size 3 are statistically significantly different (we expect lower) from the implementation times of user stories with size 5. If they are different (test rejected), that implies that the team is properly assigning user stories to right story point category; otherwise, if the test is not rejected (i.e., implementation times of user stories of size 3 are not statistically different from those of size 5), that means that the story points categories 3 and 5 are in the practice indistinguishable: the cause could be either estimation error or the use of too similar categories of story points, that could be merged into one category.

RQ4: Does the effort per story point vary over the course of the project?

M7 Spearman correlation coefficient [18, p. 396] between the story point value and the sprint number (once for all user stories & once per story point category)

M8 (Multiple) linear regression (once for all user stories once per story point category):

- x=Sprint number
- y=Actual effort per story point

Explanation. For a user story with the identifier US_X, the actual effort per story point is computed as follows:

$$actEffortPerStoryPoint_{US\_X} = \frac{totalImplementationTime_{US\_X}}{storyPoints_{US\_X}}.$$

We then ordered the stories once again according to their sprint memberships, computed the Spearman correlation coefficient and applied a multiple linear regression. This was done once for all user stories to observe the overall tendency, and for the user stories in each story point category.

RQ5: Does the accuracy of estimates improve during the project?

M9 Spearman correlation coefficient between the estimation accuracy and the sprint (for under- & overestimated user stories and overall)

M10 (Multiple) linear regression (for under- & overestimated user stories and overall):

- x=Sprint number
- y=Estimation accuracy

Explanation. Our concrete measure for the estimation bias is the relative error (RE) of a user story. For example, a story with the identifier US_X is calculated as follows:

$$RE_{US\_X} = \frac{EstimatedTime_{US\_X} - ImplementationTime_{US\_X}}{ImplementationTime_{US\_X}}.$$

For example, a user story, that was estimated to take 25 hours and needs 20 hours to be completed is overestimated by 25%:

$$RE_{US\_X} = \frac{25-20}{20} = \frac{5}{20} = 0.25$$

Thus, an overestimation by 25% describes that the estimation is 25% higher than the actual value. This measurement limits underestimations to absolute values between 0% and 100% by construction: for user stories, where the estimated effort is slightly lower than the actual effort, the difference in the nominator tends towards 0, and so does the overall result. The 100%-border can be understood by taking in consideration, as illustrative example, an estimated effort of 0.1 hours and actual effort of 10 hours:

$RE_{US\_X} = \frac{0.1-10}{10} = \frac{-9.9}{10} = -99\%.$ [9]

As the teams estimated user stories only with story points and did not break down their estimation to a time estimation of the tasks, we had to use a proxy measure for the estimated time in order to apply the RE. After a preliminary analysis of the data, we decided to use the the median of the implementation time of all user stories in a story point category as estimated time of a user story in that category. For example: the estimate for a 5-story-point user story is the median of the implementation time of all 5-story-point user stories in that project. Due to this necessary work around, the results of these research questions can only be considered as a proxy of a classic estimation error. In addition, as can be noted from the examples, the directional error is always negative for underestimated user stories and positive for overestimated user stories.

We computed the Spearman correlation coefficient [18, p. 396] for estimation accuracy over time for under- and overestimations separately. Moreover, we applied a linear regression for both types of wrong estimates.

RQ6: How many estimates are inaccurate?

M11 Rate of over- and underestimates in a project

M12 RE interval of over- and underestimates

M13 Precision of detected wrong estimates

Explanation. For detecting wrong estimates we computed the accuracy as described in RQ5 and applied the following heuristics based on the interquartile rule for outliers:

1. Calculation of the interquartile range (IQR) for the RE of all user stories
2. Identification of overestimations: A user story US_X is an overestimation if $RE_{US\_X} > Median(REs\_of\_user\_stories\_with\_same\_story\_points) + IQR * 1.5$
3. Identification of underestimations: A user story US_X is an underestimation if $RE_{US\_X} < Median(REs\_of\_user\_stories\_with\_same\_story\_points) - IQR * 1.5$

Due to the approximation of the accuracy measure, we validate the identified wrong estimates with the product owners.

4. Study context: company and projects description

The study was conducted at TechDivision, a medium-size [3] company with approximately 70 employees that develops web applications with agile methodologies. The company headquarters are in Kolbermoor (South Germany) but there is an office located in Munich, where most of the projects analysed in this study have been done. Tech Division's development is mostly based on two frameworks: TYPO3, a widely adopted content management systems in German speaking countries [23], and Magento, a popular content management system for e-commerce websites [17]. The two technologies can be combined to provide solutions for content-commerce-systems. Overall, TechDivision has more than 17 years of experience in developing web applications: they have developed more than 40 Magento projects and 100 TYPO3 projects and are a certified partner for both frameworks. In 2011, TechDivision switched from classic to agile project management. At the time of our analysis, the organization was structured in multiple teams, each including developers, product owners, and Scrum masters.

For our study, we chose projects which used either Scrum or Scrum variations and which estimated with Planning Poker using story points in the Fibonacci-like scale [4].

We analysed data from four web application-based development projects: we describe them through a list of context variables selected from the taxonomy proposed by Kalus and Kuhrmann [10], to whom we added a few additional variables specific to agile development. The resulting characterisation schema involves five dimensions: team, internal development context, team client context, project's objectives, estimation process.

The product owners filled their respective projects' characterisation schemas, that we report in an online Appendix ([5], together with additional details on the course of the projects: we consider this information useful to understand our interpretation of results and for comparing results of similar future studies.

Herein we briefly summarize the projects with a qualitative description:

- Project 1 was developed by 3 to 7 developers and had a duration of approximately 1.75 years. With 393 developed user stories and approximately 12,700 hours of tracked work, it is by far the largest project we consider. The goal of the development was a content-commerce system, i.e. TYPO3 and Magento were utilized, that had to deal with several neighbouring systems. Overall, it was rated to be highly complex.

- Project 2 was about the development of a webshop based on Magento. The development team started with two developers and then reached a constant number of six developers. The final team remained almost unchanged with only

[3]According to European Union recommendation 2003/361/EC categorisation schema: http://goo.gl/eNNVE5

[4]The Fibonacci sequence consists of numbers that are the summation of the two preceding numbers. For example: 0,1, 2, 3, 5, 8, 13, 21. In Agile software development, the Fibonacci scale is used for estimating the relative size of user stories in points

[5]The appendix is available at: `https://researchdata.nexacenter.org/2018fcds/appendix.pdf`

one team member being replaced by another developer. The project took ten months overall, was about 2,500 hours of work, and developed 62 user stories.

- Project 3 was the development of a customized web application and took over a year. Altogether 206 user stories were developed and a total of 5,600 hours of work were tracked. The project used mainly new technologies introduced by one team member and therefore the team gathered knowledge on the technology during the development.
- Project 4 was the smallest project in this study and concerned the relaunch of a TYPO3 website. It covered 47 user stories, took three month, and 1,400 hours of work. The development team was co-located, consisted of fours developers, and was constant during the project. Moreover, the developers were familiar with the technology and have already collaborated on a similar project before.

4.1. Data extraction and preparation

In all projects, JIRA [6] was used for project management. Therefore we were able to extract all user stories of each project through a JIRA query [7]. This query generates an Excel file containing several information of each user story that was developed in a specific project. Specifically, for each user story of the project, the information provided by JIRA is: a unique key identifying the user story; a summary and a detailed description of the user story; the status of the user story (whether it was closed or it was still under development); the resolution of the user story (whether it was fixed or not); the creation time of the user story; the time spent for implementing the user story; the keys of the sub-tasks constituting the user story, if any; the time spent for implementing the sub-tasks, if any; the estimated story points; the sprints which the user story was assigned to. Regarding the implementation time, the JIRA output provides information about the time spent on the specific JIRA ticket and the total implementation time spent, a value including the described time spent plus the time spent on the sub-tasks of the ticket. This information is not automatically provided by JIRA but explicitly entered by a developer when he performs work on a user story: for example, if a developer spends five hours on implementing a user story and another developer spends two hours on testing and integrating it, each developer logs his effort in the corresponding user story in JIRA. If the development team identifies sub-tasks for a user story in the sprint planning, these sub-tasks are generated and linked to the corresponding user story in JIRA. Then the developers can log their implementation time spent to the tasks more specifically.

We noticed that the total time spent was not correct when a related sub-tasks is not closed correctly. Therefore, we developed a Java program that calculates the time

[6]https://www.atlassian.com/software/jira

[7]*project = "projectname" AND type = Story AND (resolution = Done OR resolution = Fixed)*. The JIRA output on the sub-tasks is generated by using the following JIRA query: *project = "projectname" AND type in subTaskIssueTypes()*. Since the JIRA web interface can not export more than 1,000 items at once, it might be required to split the query, e.g. by using the "createdDate" attribute

spent on the sub-tasks and creates an Excel file that is prepared for further calculation with R [8], a programming language for statistical computing and graphics.

Finally, a short note is necessary on the ordering of user stories, which was necessary to correctly compute the metrics for RQ1 and RQ2. Since the observed user stories did not contain the exact estimation date, we decided to order them chronologically according to their sprint membership. We assume that the estimation of a user story is made or reviewed promptly prior to its implementation, for example in the sprint planning meetings. If a user story is assigned to multiple sprints, i.e. if it could not be completed in one iteration, we concentrate on the first sprint it was assigned to. This sorting seems to be closer to reality than using other existing information about the user stories like their creation or resolution dates.

5. Results

Herein we summarize the results in two tables: we provide a qualitative synopsis in Table 2, while we report the metrics for each research question in Table 3. Each row is assigned to a project, while the columns contain the following information:

- RQ1: the linear regression formulas obtained using the variance and the coefficient of variation as dependent variable(M1)
- RQ2: the Spearman correlation coefficient (M2) and the linear regression of sprint number vs story points (M3)
- RQ3: the Spearman correlation coefficient (M4) and the multiple linear regression of estimated effort vs actual effort (M5), and the number of distinct story points categories found by applying pairwise the Mann Whitney U between groups (M6)
- RQ4: the Spearman correlation coefficient (M7) the multiple linear regression of sprint number vs actual effort (M8)
- RQ5: the Spearman correlation coefficient (M9) and the regression line accuracy vs sprint number (M10), both for under and over estimations
- RQ6: the proportion of under and over estimations (M11), the RE range (M12) for both under and over estimations, and the precision (M13) of detected wrong estimates from the validation with the product owners.

Together with the regressions correlation formulas, we report also the obtainedR^2. In addition, an asterisk close to the coefficient correlations, signals that the correlation is significant with p value ≤ 0.05

Given the large amount of data (13 metrics multiplied by 4 projects), we do not report in this paper all the graphs that we computed and analyzed; however, we provide an online website[9], which also contains the following additional information:

[8]http://www.r-project.org/

[9]All results and descriptive statistics are available at https://researchdata.nexacenter.org/2018fcds/

- RQ1: the relation between the variance/CoV and sprint number is visualised with a scatter plot including a regression line to give a more detailed overview of the trend direction. We also provide boxplots of the story points for each sprint and the mean of the estimated story points in the sprints.
- RQ2: the correlation graphs can be consulted.
- RQ3: additionally we report the correlation between the story point size of a user story and its implementation time, the box plots also shown in Fig. 1 together with the table of all pairs of Mann Whitney tests computed between story points categories.
- RQ4: the relation between the implementation time per story point is visualized with a scatter plot including a regression line to give a more detailed impression of the trend direction
- RQ5: we report graphs of the regression lines separately for over- and underestimations.
- RQ6: in addition to the data displayed in the result Table 3, the online Appendix shows in details the outlier detection analysis, also per story point category.

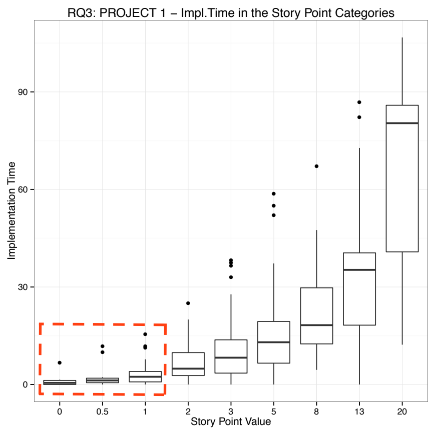
Proj	**RQ1** Estimations variance	**RQ2** Estimations size	**RQ3** Estimation and actual effort	**RQ4** Effort per story point	**RQ5** Accuracy of estimations	**RQ6** Wrong estimations
P1	Decreasing until a team turnover, then increasing	Slightly decreasing	Almost quadratic	Costant	Improving (overest.), constant (underest.)	Very inaccurate, especially over-
P2	Decreasing	Decreasing	Linear	Slightly increasing	Slightly decreasing accuracy	Quite inaccurate
P3	Decreasing	Constant	Linear	Slightly decreasing	Decreasing accuracy	Very inaccurate
P4	Decreasing	Slightly decreasing	Quadratic	Slightly increasing	Improving accuracy	Quite inaccurate

Table 2. Textual synopsis of the analysis results for each research questions

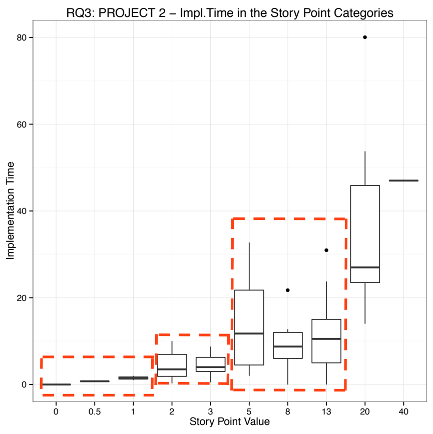
	RQ1 Estimations variance		RQ2 Estimations sizes		RQ3 Estimation and actual effort		
	M1 Regr. Var.	M1 Regr. CoV	M2 Spearman	M3 Regr.	M4 Spearman	M5 Regr.	M6 Categories
Proj 1	26.84−0.61x (R^2=0.18)	0.91−0.01x (R^2=0.17)	-0.117*	5.7−0.08x (R^2=0.02)	0.637*	2.58+1.97x+0.05x^2 (R^2=0.51)	7
Proj 2	57.47−2.74x (R^2=0.23)	0.73−0.01x (R^2=0.04)	-0.352*	10.44−0.33x (R^2=0.07)	0.623*	0.9+1.43x+0x^2 (R^2=0.45)	5
Proj 3	193.35−15.12x (R^2=0.30)	0.89−0.01x (R^2=0.07)	0.008	10.88−0.2x (R^2=0.00)	0.586*	1.46+1.28x+0x^2 (R^2=0.37)	5
Proj 4	18.3−2.81x (R^2=0.46)	0.81−0.02x (R^2=0.02)	-0.135	5.27−0.46x (R^2=0.05)	0.712*	9.88+1.19x+0.38x^2 (R^2=0.76)	5

	RQ4 Effort per story point		RQ5 Accuracy of estimations		RQ6 Wrong estimations		
	M7 - Spear.	M8 - Reg	M9 - Spear.	M10 - Regr.	M11 - Rate.	M12 - Interval	M13 - Precision
Proj 1	-0.013	2.94+0.04x+0x^2 (R^2=0.00)	Over: -0.077 Under: -0.055	Over: 3.37−0.049x (R^2=0.00) Under: -0.35−0.002x (R^2=0.00)	Over: 11% Under: 4%	Over: [1.75, 34] Under: [−0.52, −0.88]	72%
Proj 2	0.106	1.6+0x+0x^2 (R^2=0.01)	Over: 0.064 Under: -0.146	Over: 1.21+0.074x (R^2=0.02) Under:-0.36−0.003x (R^2=0.01)	Over: 6% Under: 3%	Over: [4.875, 13] Under: [−0.55, −0.66]	83%
Proj 3	-0.165*	0.66+0.36x−0.03x^2 (R^2=0.02)	Over: 0.238 Under: -0.142	Over: -0.16+0.314x (R^2=0.09) Under: -0.4−0.011x (R^2=0.02)	Over: 8% Under: 7%	Over: [1.88, 22] Under: [−0.74, −0.95]	77%
Proj 4	0.199	4.4+1.61x−0.23x^2 (R^2=0.01)	Over: -0.34 Under: -0.097	Over: 1.18−0.147x (R^2=0.08) Under:-0.28+0.09x (R^2=0.00)	Over: 6% Under: 4%	Over: [1.33, 2.12] Under: [−0.58, −0.62]	80%

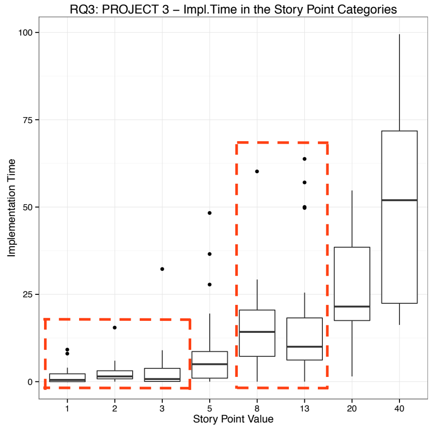
Table 3. Results



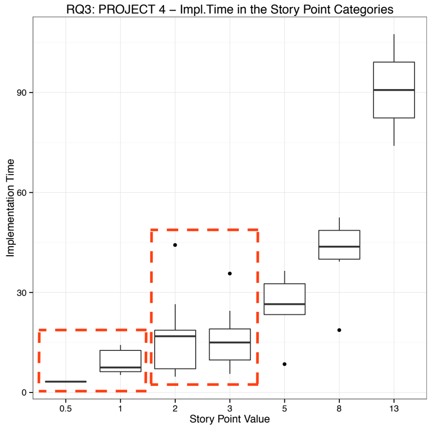
(a) RQ3 - Project 1



(b) RQ3 - Project 2



(c) RQ3 - Project 3



(d) RQ3 - Project 4

Figure 1. Boxplots of implementation times per story point categories. The dotted lines show overlapping categories identified with the left side Mann Whitney Test

6. Discussion

Our interpretation of results are based on the projects contexts and on qualitative feedback from the product owners of the corresponding projects. Concerning the latter one, we usually obtained it by bringing the analysis results printout to discussion with the product owners. For Project 1, the feedback was collected differently: the whole team participated and the results were continuously displayed on a wall in a common space, so that each team member was able to add his/her own ideas and observations at any time: this was intended as a simulation of the approach described in [22]. We could not apply the same strategy to the other projects because not all team members were located in Munich: however all product owners were there, so it was possible to gather their comments. We continue the remainder of the discussion per RQ, with a final short summary.

RQ1: Are estimates getting more similar during the projects? - RQ2: Are estimates getting smaller or bigger during the projects? We observed from RQ1 that the range of story points used got smaller during the course of the projects. Complimentary information to understand the cause of such variation comes from the results of RQ2, which showed that the estimated size of user stories slightly decreased for three projects over the time. In Project 1, the tendency to use less story point values for estimation was a symptom of a conscientious decision the team made after some problematic iterations before sprint 22: team members agreed to avoid the estimation of 13 and 20 story point stories, splitting them into smaller stories. However, this decision was withdrawn due to a change in the development team ($\sim$ 26 sprint) in the further course of the project: this also explains the increase of the variation in a later phase that we reported in Table 2. We do not have similar explanations for the other projects. However when checking descriptive statistics together with the results from RQ2, we observe that Projects 2 and 4 display a (slightly) negative correlation between the estimated size of user stories and the project progress: we know therefore that decrease in the usage of the full range of the story points is due to a higher usage of smaller stories. However, we do not have evidence, as it was for project 1, to interpret it as a symptom of a problem in managing large user stories, or whether this happened simply because the largest stories were at the top of the product backlog. Additionally, we cannot exclude the possibility that team members unconsciously focused on less story points categories with the course of the project.

RQ3: How accurate the estimates are? Regarding RQ3 results, the positive Spearman coefficient (metric M4) confirmed that higher estimates corresponded proportionally to higher implementation times. We remember the reader that the rational of such a question was to check the relative character of user stories, e.g. a 5-point user story should take half of the time of a 10-point user story: therefore one would assume a linear regression line describing the relation between the story points of a user story and its implementation time. This was true for Projects 2, 3 and partially for Project 1, while the fourth project showed a quadratic growth in implementation time: this would mean that, for example, a 10-point user story needs more than twice as much time as a 5-point user story. However, the usage of Fibonacci scale would support also a quadratic relationship, as considered also by a story point analogy described by Cohn [3, p. 52] : in that example, each story point value is illustrated as a glass while the needed effort of a user story is represented by water. A user story may only be assigned to a story point value if its effort fits into the glass: whenever the effort of a user story is too big for one bucket, the next bigger story point category has to be chosen. In case a Fibonacci scale is used, as in the projects analysed, a different behaviour may occur: in Figure 2, the blue line describes a perfectly linear story point meaning where a 40-point user story exactly takes twice the effort of a 20-point story, while the red line indicates a possible story point development according to Cohn's analogy. For this reason, we do not consider a slightly super-linear relationship as symptom of a problem in the estimation process or in the project in general.

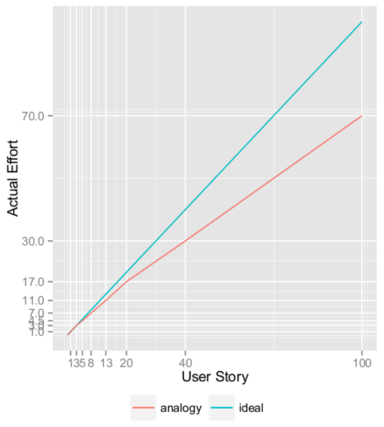


Figure 2. Visualization of the “glass” analogy for story points

RQ4 - Does the effort per story point vary over the course of the project? Concerning RQ4 results, we observed that the general definition of story points as a relative measurement was not valid for three of the four observed projects as the value of a story point was not constant in the different story point categories. In particular, in Projects 2 and 3 the middle-sized user stories (respectively 5 story points and 8 story points) required more in terms of time, while in Project 4 this happened for smaller user stories. Thus, it has to be questioned whether a predefined estimation scale with numerical values is beneficial for following planning activities in agile projects, e.g. a team might be able to deliver a 8-point user story within one sprint while the combination of a 3-point and a 5-point user story can not be completed due to a higher cost per story point in the 5-point category.

RQ5: Does the accuracy of estimates improve during the project? Regarding the capability of the teams to improve in terms of estimation accuracy, we could observe it only in Project 4 (and only for overestimations in Project1). In Project 1, we observed extreme underestimations after sprint 16, which coincides with a change in the software development process: after sprint 15 the team switched from Scrum to an approach Kanban-like (but keeping estimations) for a period of two months before returning to Scrum: thus, we believe that it was the change of approach which has lead to the wrong estimates. Another finding was that the number of extremely wrong estimates decreased with a shortened iteration length of one week from sprint 26 on. What the development team identified as a potential reason for this change was the possibility of a more detailed sprint planning for one than for two weeks. However, this did not hold in Project 2 where the iteration length was similarly shortened to one week. Together with the product owner, we were able to identify potential reasons for the decreasing accuracy in Project 3: after sprint 5 the project was stopped due to the customer's dissatisfaction. After a break, all remaining user stories were estimated more vaguely in order to give the customer an impression of the expected costs. The project was then resumed with a fixed budget and the assurance of the desired functionality. Due to the time pressure the assigned estimates where not re-checked in the following sprints and during the sprint planning meetings the user stories were handled independently from their estimate. In Project 4, the product owner identified the team's learning from complex tasks in the first iterations as a main reason for the improvement in estimating. The team used to have fixed appointments and structures for their sprint meetings, including the grooming meeting. Thus, the approach in Project 4 is the best representation of the targeted continuous improvement through the inspect-and-adapt approach in Scrum. Comparing this to the poor results of Project 3, where the team did not have grooming meetings in the second half of the project, a fixed structure in the meetings seems to be beneficial for the improvement of estimates in a project.

RQ6: How many estimates are inaccurate? Finally, the results of RQ6 show that the Projects 1 and 3 not only have the highest rate of wrong estimates (15%), but also that the estimates are extremely wrong: in Project 1 the RE of the overestimations range from 1.75 to 34, i.e. the estimated effort exceeds the actual effort by a minimum of 175% up to a maximum of 3400%. Project 3 is similar in this regard as the upper border of the overestimations is 2200% here. Projects 1 and 3 are the two projects with the greatest number of changes to the software process as both had a longer break where Scrum was interrupted: while Project 1 interrupted Scrum due to a process change to Kanban-like process, Project 3 had to be stopped due to problems in the negotiation with the customer. As explained in RQ5, both interruptions resulted in problems with the estimation accuracy. Project 4, by contrast, was characterized as the project with the most constant software development process as there were no interruptions to the project and further regular meetings helped to ensure a well-maintained product backlog. From the perspective of the product owner, these established structures were beneficial to the estimation, planning, and the project success in general. The discussions with the product owners about the validation of the wrong estimations detected by our heuristic, revealed that underestimations were often caused by a falsely assumed standard functionality of TYPO3 or Magento, technical problems, e.g. an inevitable refactoring, and or unnoticed side tasks like cross-browser support or intensive testing. In contrast to underestimations, overestimations are often caused by a too high weighting of uncertainties, leading developers to integrate an actually unnecessary buffer. In these cases the teams discuss multiple solutions for a user story and then credits the uncertainty of which solution might be appropriate with a higher overall estimate.

Summary The evidence we collected let us identify a few symptoms of problems:

- the reduction of the number of story points over the course of the projects, with a tendency to use smaller stories (RQ1, RQ2; all projects);
- not constant value of a story point (RQ4; projects 2,3,4)
- a degradation of the accuracy of estimations, with very large errors (RQ5, RQ6; projects 1,2,3).

Therefore, according to our interpretation of results and the feedback collected from the team members, and excluding external factors (e.g., team changes due to leaves), we derived the following explanations:

- **E1** Too many items in the estimation scale are not distinguishable by team members
- **E2** The usage of a numerical scale is misleading
- **E3** Estimating new user stories without an explicit and shared reflection on previous estimations can lead to extreme wrong estimation

7. Improvement proposal

7.1. Treatments selection

Based on discussion of result and our explanations, we derived the following hypotheses:

- **H1**. A non numerical scale with fewer items allow team members to allocate better user stories to estimation categories.
- **H2**. Mechanisms that forces the team members to reflect on previous estimations improve the accuracy of the estimations.

We declined these hypotheses into three treatments to be applied in the estimation process of Tech Division.

T1 Estimation unit and scale. As discussed in previous section, We observed that less and smaller user stories were used over the course of a project. According to the knowledge gathered, the cause is a natural tendency of estimators to handle a few categories and that a numerical scale for user story estimation was misleading. Therefore, we introduced an estimation scale with only five different categories (which was the number of distinguishable story points categories in three out of four projects, as reported in Table 3 and Figure 1d): the categories are abstract and use common T-Shirt sizes to symbolise the amount of effort needed for the completion of a user story. The new estimation scale discussed with the team members resulted in: S, M, L, XL, and XXL. In addition, a "can't or won't estimate“ option is available, which displays that a user story either cannot be estimated due to the uncertainty or that the user story is too large to be assigned to a category.

T2 Analogy-based estimation. We introduced an analogy-based estimation process: we wanted to encourage the team to compare new user stories to already estimated and implemented ones. In the designed process, the developers can use the analogy in their discussion about a user story and they are always forced to check its estimation before finally assigning it to a specific estimation category.

T3 Estimation retrospective. Based on the success factors of Project 4 in terms of accurate estimations, which were a problem for the other projects, and inspired by the inspect-and-adapt principle of Scrum, we introduced an estimation retrospective, which is used to examine the statistical over- and underestimated stories from the last sprint. In our opinion this should recreate a learning process that helps the team to avoid wrong estimates and improve their estimation accuracy during a project.

7.2. Metrics

To quantitatively measure the effect of the treatments and verify our hypothesis, we decided to rely on two metrics from the set of those used in the first analysis of the four projects. The main reason rely on the change of measurement scale from ratio to ordinal scale (but with nominal items). In order to compute most of the metrics, we would have had to transform the nominal scale into a numerical one, assigning for example 1 to S, 2 to M, and so on. However, the measurements would be misleading because in the former measurement cycle user story sizes were expressed in terms of items of a Fibonacci scale. Assigning the T-shirt sizes to the items of a Fibonacci is also a not a valid procedure, because the meaning of the categories in the two scales are totally different. For this reason we opted for relying only on measurements on the implementation times of each T-shirt size category, without regressions. The resulting metric set is reported in Table 4.

Since we could not exclude interdependencies between the treatments, te best way to isolate their effects would be to apply them one by one. However, due to feasibility reasons, the preference of the company and the preference for pragmatism, we tested the treatments in a unique cycle of experimentation. Hence, to identify and isolate

Hypothesis	**Measurement**
H1 - A non numerical scale with fewer items will allow team members to allocate better user stories to estimation categories	M7 Number of undistinguishable estimation categories (identified through Mann Whitney test)
H2 - Mechanisms that forces the team members to reflect on previous estimations will improve the accuracy of the estimations.	M12 Rate of over and underestimates M13 RE interval of over- and underestimates

Table 4. Summary of the treatments

the effect of each treatment, we complemented the measurement on the process with a qualitative evaluation consisting of a short questionnaire filled at the end of an estimation meeting, to gather the perspective of the team members. The questions used are reported in Table 5. Given the low number of team members, we used only descriptive statistics for the analysis.

# ID	Question	Answer options
Q1	"The number and the name of the categories helped me to estimate user stories": in which degree this statement is true for you?	No / Only the abstract category name / Only the number of categories (5) / Yes
Q2	If not, why did categories did not help you ? What could be improved ?	Free text
Q3	"The use of comparison to reference user stories helped me to estimate user stories" : in which degree this statement is true for you?	Likert scale from 1(Strongly disagree) to 5 (Strongly agree)
Q4	If not, why did the comparison did not help you ? What could be improved ?	Free text
Q5	"The grooming retrospective helped us to find a good practice for our estimations" in which degree this statement is true for you?	Likert scale from 1(Strongly disagree) to 5 (Strongly agree)
Q6	If not, why did the retrospective not help you? What could be improved ?	Free text

Table 5. Questionnaire at the end of the estimation meeting

7.3. The new estimation process

Figure 3 displays the course of an estimation meeting according to modified process, which we are going to describe through its main characteristics.

Estimation wall. For the estimation meetings we prepared an “estimation wall”, showed in Figure 4: the center of the wall is a whiteboard, where the estimation categories are visualised as columns where user stories, represented by post-its, can be physically assigned after their estimation. Above the whiteboard, the description estimation process takes place, as well as our definition of the estimation unit. Right to the whiteboard, there is room to present information on the user stories completed in the last sprint. The team will use it for the retrospective to identify best practices for further estimation meetings, which will be inserted in the checklist above.

Estimation meeting - part 1 Each estimation meeting should take place in front of the estimation wall with a computer screen visible to all team members: according to the vision of the fast feedback cycles [22], practitioners should rely on the use of data (interactive) visualisations for their activities. Therefore the team can use the computer to access the original user story in JIRA and for its discussion (e.g.,to estimate user story is necessary to examine the current status of the product increment). In addition, in our process an estimation meeting begins with a retrospective, where the development team reflects on difficult user stories that were developed during the last sprint (e.g. very inaccurate estimates). For this purpose, we wrote an R script which generates box plots for the implementation time per estimation category, one each for all user stories in the project and for the user stories from the last sprint. Moreover, the script produces a list of the last sprint’s user stories with important pieces of information like the estimated category, the implementation time or the number of assigned sub-tasks. In order to structure the retrospective and collect its results, the box plots and the list of user stories are prepared on a flip chart. If the team has identified very good estimations from the last sprint, then it tries to derive best practices and write them on a post-it. If the team has identified a wrong estimation, it can re-assign the user story to the correct category: in this way, the team can avoid wrongly estimated user stories for future comparisons.

Estimation meeting - part 2 In the second part of the meeting the actual estimation of user stories takes place. Similar to planning poker the user stories are estimated sequentially. In the beginning, the product owner presents the user story and shares his knowledge about the requirement with the team. The team then discusses the user story in a way so that each estimator can derive an estimation for himself. Since the estimation meeting is hold in front of the estimation wall, the use of analogy is easier and more prominent. A common problem in analogy-based estimation techniques is that often a initialization phase is needed: if there is no reference user story in the estimation categories, no comparisons can be made. We tackled this issue by conducting a training on the new estimation process, where we let the team

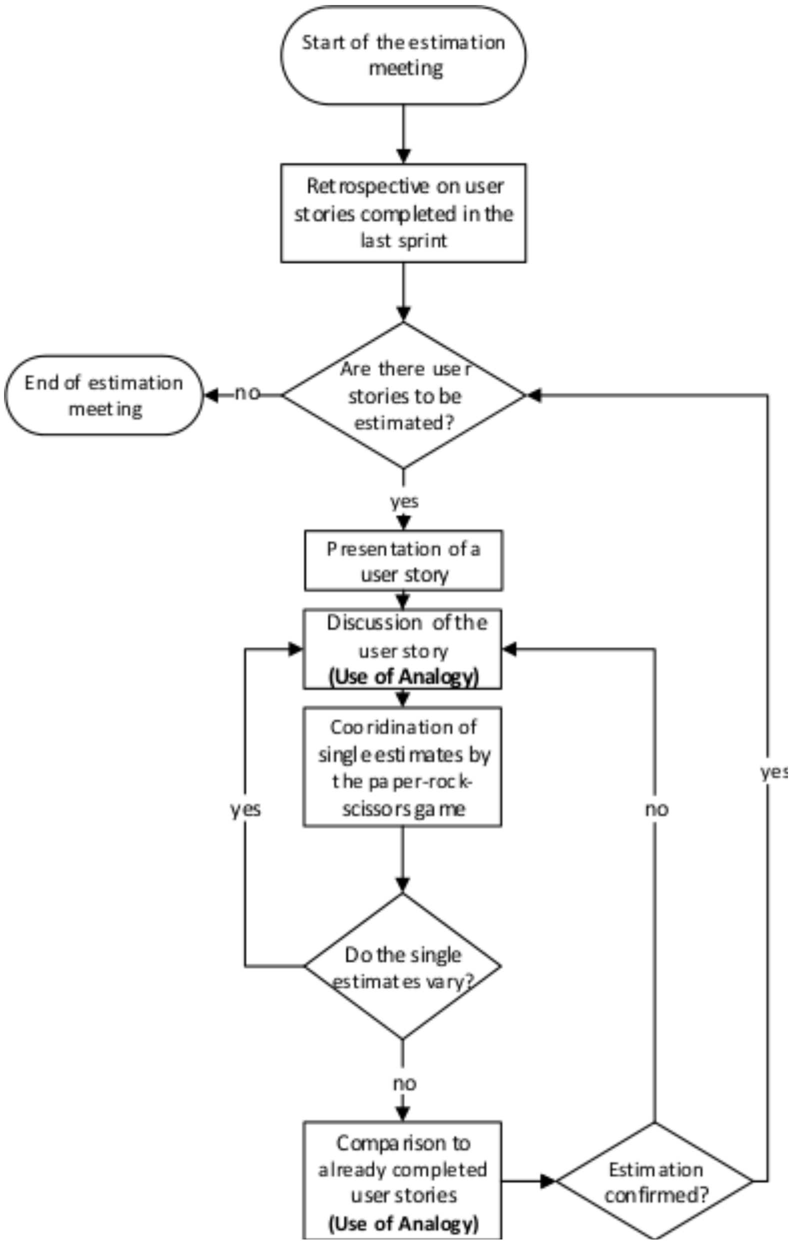

Start of the estimation meeting
Retrospective on user stories completed in the last sprint
Are there user stories to be estimated?
no
End of estimation meeting
yes
Presentation of a user story
Discussion of the user story (Use of Analogy)
Cooridination of single estimates by the paper-rock-scissors game
yes
no
yes
Do the single estimates vary?
no
Comparison to already completed user stories (Use of Analogy)
Estimation confirmed?

Figure 3. The flow of an estimation meeting according to the changed process

estimate user stories from the last sprints: this helped the team not only to become familiar with the new approach, but also provided reference user stories.

The team plays a rock-paper-scissors variant to coordinate the single estimates:

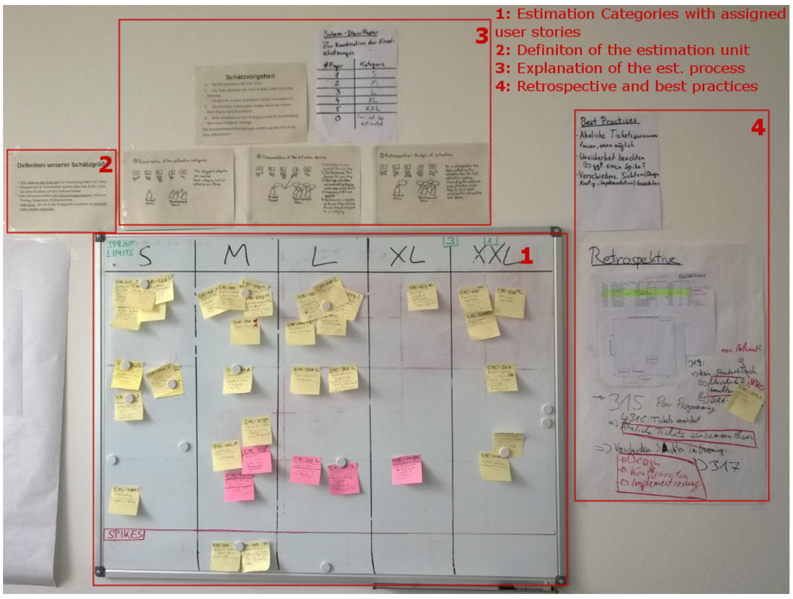

1: Estimation Categories with assigned user stories
2: Definiton of the estimation unit
3: Explanation of the est. process
4: Retrospective and best practices
S
M
L
XL
XXL
SPRINT LIMITS
SPIKES
Retrospektive
Best Practices

Figure 4. Example of the estimation wall at TechDivision

each estimator simultaneously reveals to which of the five estimation categories (s)he would assign the discussed user story by the number of fingers (s)he shows to the group. While one finger indicates a S user story, five shown fingers stand for a XXL user story. Additionally, an estimator can refuse an estimation by showing a fist to the group. This indicates that the user story can not be estimated or that is too big for the completion in one sprint. If all participants agree in their estimation, the estimation of the user story has to be confirmed in a next step by comparing it to already estimated and completed user stories in the according category. If the team members vary in their estimation, the user story has to be discussed again (a good approach might be to have the highest and the lowest estimator explain their estimations to capture the diverging reasons). Moreover, if the estimations of the rock-paper-scissor variant differ in two categories, it might be useful to pin the user story between the categories and compare it to the user stories of each category. This might help to identify which estimation might be more appropriate. After the discussion is completed the team plays the rock-paper-scissors variant once again to see whether it can reach a common estimation now. However, if multiple rounds do not lead to a common estimation for a user story or if the team thinks it cannot be estimated, then the user story has to be “refused” in this estimation meeting and thus is not ready for implementation. Ideally, the results of the discussion and the reasons playing into the estimation decision are noted on the post-it representing the user story. This makes the user stories more comparable for further estimations and might help the team to identify estimation problems during the next retrospective.

7.4. Project hosting the experimentation

The project context in which the treatments were introduced was similar to the one of Project 2: it was the same team developing a comparable webshop with Magento. The efforts for the user stories in the project were first estimated in ideal hours in a Planning Poker-like meeting. But since the team heavily underestimated the user stories with this approach, we got the chance to introduce the estimation approach we had designed. The project had been worked on for about 1.5 months before our change in the estimation process was established.

We prepared the estimation wall in the room where the development team was co-located. The definition of our estimation unit was discussed with the team and finally presented and explained in a blog post in the company's wiki and also added a short description to the estimation wall.

We provide in the Appendix the same contextualisation schema built for the previous projects.

8. Experimentation results

We were able to observe seven one-week iterations that were conducted according to the presented estimation approach. The team was constant over the time except for one developer leaving after the second iteration. It is important to note that our observation did not start with the beginning of the project, nor could we accompany the project to its completion due to the fact that two of the co-authors had to leave. We participated in three of the estimation meetings and assured that they were conducted according to the newly established process.

The qualitative analysis is based on these three estimation meetings: the descriptive statistics on the questionnaire answers are reported in Figures 6, 7 and 8.

The quantitative analysis covers all seven sprints: in that period the development team completed 34 user stories, which covers approximately half of the total course of the project. The results are the following ones.

M7 The Mann-Whitney U test provided p-values greater than 0.05 only in comparisons including the XL estimation category (see Table 6 and Figure 5). Since the XL category can be ignored due to its small sample size, the null hypothesis of similar groups is rejected for all estimation categories. Hence, the actual effort of the different new categories differed from each other.

M12 The rate of over and under estimates are, respectively, the following: 12 % (4/34) and 3 % (1/34)

M13 The RE interval is 1.19 to 2.57 for overestimations, and -0.494 for the only underestimation detected.

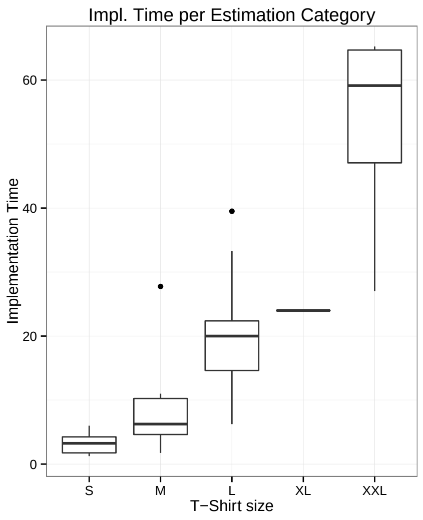


Figure 5. Estimation Approach - Results of RQ3 (M6): Box plots describing the implementation time in each story point category

	M	L	XL	XXL
S	0.0274	1e-04	0.125	0.003
M	-	0.0075	0.25	0.0061
L	-	-	0.1924	0.004
XL	-	-	-	0.2

Table 6. Estimation Approach - Results of RQ3: Pairwise Mann-Whitney U test for the implementation time in the story point category

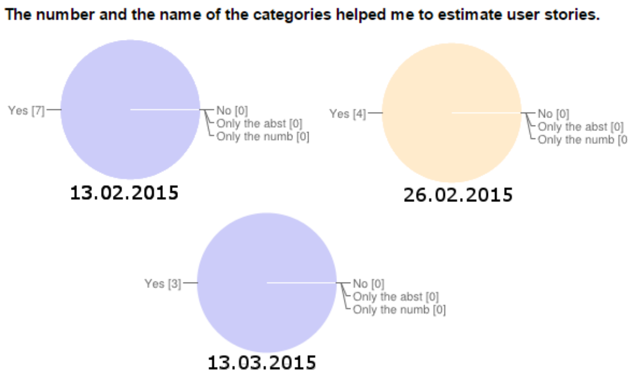


Figure 6. Question on the estimation category

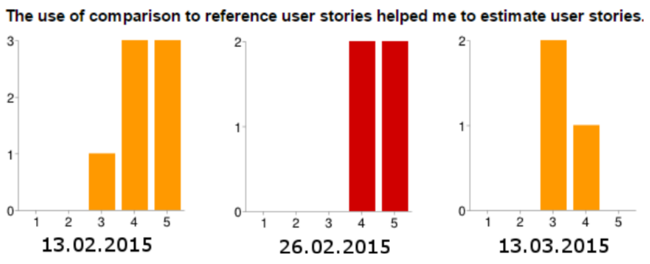


Figure 7. Question on the use of analogy

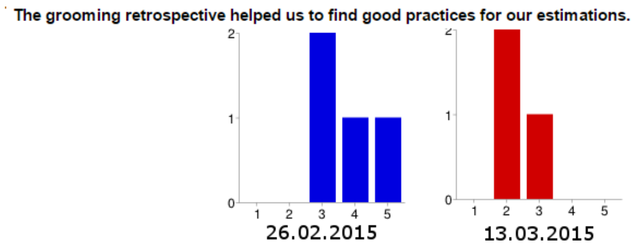


Figure 8. Question on the retrospective

9. Experiment results discussion

The quantitative analysis lead to consider two positive effects of the treatment on the estimations.

Firstly, the implementation time of the five used estimation categories could be distinguished very well, as the box plot and the Mann Whitney test also show. This confirmed that the 5 items might be a sufficient number to use in a scale for estimating user stories. However, we also observed that the category XL was used only once, therefor we do not exclude the possibility that even 4 items could be sufficient in a project like the one we analysed.

Secondly, the detected false estimations were not that extreme compared to the projects in the first measurement cycle. The one detected underestimation has an accuracy of -49.7%, indicating that it actually needed twice the estimated effort. In comparison, we noticed an improvement from 10% to 45% with the previous projects. The overestimations have a relative directional error between 1.19 and 2.57: this value is at the same level with the one from project 4, which ranges from 1.33 to 2.12.

Besides these positive effects, there was a still a high percentage of wrong estimations: in total, 14.7% of the 34 user stories were identified as wrongly estimated, according to our detection algorithm. However when discussing the wrong estimations with the product owner about the results, we discovered that one of detected user stories could be removed from the inaccurate ones, because it was assigned the wrong issue type in JIRA (the overestimation was actually a time-boxed technical spike). Hence, the detection rate was 11.8% (and consequently precision of the heuristic is 80%), but still did not improve with respect to the previous projects analysed.

The survey results showed that the team perceived the changes to the estimation unit and scale (T1) as most useful: in all three surveys the participants agreed that both the number of categories and the abstract name of the categories helped in estimating the user stories. Nevertheless, one answer to the open questions identify a potential threat in the rock-paper-scissors game: the respondent fears that the ordinal scale of the T-Shirts sizes might be undermined by the number of fingers used to indicate the estimated category.

The feedback on the analogy mechanisms (T2) is also considered to be useful by most of the team members: in all three surveys, 11 of the 14 respondents agreed that the use of analogy helped in estimation of user stories.

Finally, the estimation retrospective (T3) seems to have the smallest effect on the results from the quantitative analysis: 70% of the respondents did not agree that the retrospective helped to find good practices for further estimations.

Therefore, by combining the quantitative and qualitative analysis, the evidence collected indicate that the improvements observed are mostly likely an effect of the changes to the estimation unit and the explicit use of analogies during the estimation process. Therefore we accept both our hypotheses, but we exclude the retrospective meeting from the new process.

10. Limitations and future work

Although the improvements observed in the new project and the higher accuracy of estimates, we recognize that the proportion of inaccurate estimates remained at about the same level of the projects with the previous estimation process. We discussed this issues with the company, and decided to devote a follow-up study on this. In this context, and in the purpose of automating the improvement approach presented in this paper[10] we implemented an interactive visualization which displays the distributions of the user stories implementation efforts, highlighting the outliers (according to a different detection algorithm): the team is able to inspect the supposedly wrong estimations, validate them and leave feedback directly on the web application. At the time of writing, the new data analysis is ongoing. A second limitation of this study is related to the duration of Project 5 analysis: due to research staff moving, only half of the project could be analyzed. A third limitation is the lack of treatment isolation (all were tested at the same time): we could not fix this threat, but by conducting a qualitative validation on the quantitative findings we could control it. Finally, our results cannot be generalized to other projects and companies: however we provided in appendix a detailed project characterization that should serve as a guide for applying the proposed improvement methodology in other companies and compare the achieved results with ours.

[10]The work is part of a broader idea of tuning data analysis with fast feedback cycles from research stakeholders. A description of the approach has been presented at the 37th International Conference on Software Engineering in 2015, where we showed the preliminary results [22] of testing the approach on user stories text analysis. We observed that a double feedback mechanism notably improved the precision of the data analysis and the quality of the data gathered

11. Conclusions

We conducted a research to understand the causes of wrong estimates in software agile development and consequently improve the estimation process. This study was conducted in a German company developing web applications with agile processes. Our improvement methodology combined software data analytics (from four projects) with elicitation of teams' feedback, which helped us to identify the following main issues: a tendency to use less story points and smaller stories over the course of the projects; a non constant value of a story point; a degradation of the estimations accuracy, with a presence of very large errors. We identified the possible explanations for those problems and subsequently defined three major changes in the estimation process: a non numerical scale for story points with less items; an analogy-based estimation; retrospectives analyses on the accuracy of previous sprints estimations. The new estimation process was applied on a new project, and we could observe a positive effect, with an improvement of the accuracy from 10% to 45% with respect to the previous analyzed projects.

We consider as our main contributions i) the improvement methodology to identify and solve estimation issues in agile software development projects, and ii) the project characterization schema that should serve as a guide for comparing further replication of this study in other companies.

Acknowledgment

We would like to thank Sacha Storz and the whole team in the partner company: without their restless availability and cooperation this experimentation would have not been possible.

References

[1] Basili V., Caldiera G., and Rombach H. D. The goal question metric approach. In *Encyclopedia of Software Engineering.* Wiley, 1994.

[2] Beck K. and Gamma E. *Extreme programming explained: embrace change.* addison-wesley professional, 2000.

[3] Cohn M. *Agile Estimating and Planning.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

[4] Egorova E., Torchiano M., Morisio M., Wohlin C., Aurum A., and Svensson R. B. Stakeholders' perception of success: An empirical investigation. In *2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, pages 210–216, Aug 2009.

[5] Grealy M. A. and Shearer G. F. Timing processes in motor imagery. *European Journal of Cognitive Psychology*, 20(5):867–892, 2008.

[6] Halkjelsvik T. and Jørgensen M. From origami to software development: A review of studies on judgment-based predictions of performance time. *Psychological Bulletin*, 138(2):238–271, 2012.

[7] Hooper P. What is a p-value?

[8] ISO. ISO/IEC 15504-4:2004 – Information technology – Process assessment – Part 4: Guidance on use for process improvement and process capability determination. Standard, International Organization for Standardization, Geneva, CH, july 2004.

[9] JøRgensen M. A review of studies on expert estimation of software development effort. *J. Syst. Softw.*, 70(1-2):37–60, Feb. 2004.

[10] Kalus G. and Kuhrmann M. Criteria for software process tailoring: A systematic review. In *Proceedings of the 2013 International Conference on Software and System Process*, ICSSP 2013, pages 171–180, New York, NY, USA, 2013. ACM.

[11] Leffingwell D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* Addison-Wesley Professional, 1st edition, 2011.

[12] McConnell S. *Software estimation: demystifying the black art.* Microsoft press, 2006.

[13] Mittas N. and Angelis L. Overestimation and underestimation of software cost models: Evaluation by visualization. In *39th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2013, Santander, Spain, September 4-6, 2013*, pages 317–324, 2013.

[14] Mohagheghi P. and Jørgensen M. What contributes to the success of it projects? an empirical study of it projects in the norwegian public sector. *Journal of Software*, 12(9):751–758, September 2017.

[15] Nan N. and Harter D. E. Impact of budget and schedule pressure on software development cycle time and effort. *IEEE Transactions on Software Engineering*, 35(5):624–637, Sept 2009.

[16] Parkinson C. N. *Parkinson's Law.* Buccaneer Books, 1996.

[17] Robertshaw T. April 2014 ecommerce survey, 2014.

[18] Sachs L. *Applied Statistics: A Handbook of Techniques.* Springer US, 1982.

[19] Sauer C. and Cuthbertson C. The state of it project management in the uk 2002-2003. *Computer Weekly*, 15, 2004.

[20] Serrador P. and Pinto J. K. Does agile work? ? a quantitative analysis of agile project success. *International Journal of Project Management*, 33(5):1040 – 1051, 2015.

[21] UCLA Institute for Digital Research and Education. Faq: What is the coefficient of variation?

[22] Vetrò A., Ognawala S., Méndez Fernández D., and Wagner S. Fast feedback cycles in empirical software engineering research. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 583–586, May 2015.

[23] W3Techs. Web technology surveys.

Received 3.07.2018, Accepted 19.10.2018