Applying a Maturity Model during a Software Engineering Course - Experiences and Recommendations

Andreas Bollin, Elisa Reci
Institute of Informatics-Didactics
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
Andreas.Bollin|Elisa.Reci@aau.at

Csaba Szabó, Veronika Szabóová
Department of Computers and Informatics
Technical University Košice
Košice, Slovakia
Csaba.Szabo|Veronika.Szaboova@tuke.sk

Rudolf Siebenhofer
SieITMCi
Siebenhofer. Consulting e.U
Steyr, Austria
siebenhofer@sieitmci.at

***Abstract*—In industry, the benefit of maturity models is uncontested, and models like CMMI are normally taught in at least advanced Software Engineering courses. However, when not being part of real-world projects, the added values are difficult to be experienced on first hand by our students. In this paper we report on a study and teaching approach where, in three successive semesters and at two different institutions, we started rating the process-maturity of students solving tasks in our software engineering courses and transparently related the maturity levels to the task performances. It turned out that there is a non-negligible correlation between the individual process maturity and performances. Considering this finding, the approach might yield to students' process-improvement steps during our courses, help in fostering the understanding of the term process maturity, and finally, also might help in improving the overall students' performances.**

I. Introduction

In a special issue of IEEE's *the institute* on the state of engineering worldwide, Kathy Pretz, editor in chief, starts with a quite emotional foreword. She also states that "today's high-tech companies can't find enough skilled engineering grads and workers, while universities and employers are struggling to keep up with the advances in technology" [1]. While graduates and workers are constantly confronted with acquiring new knowledge, keeping up the quality of courses (and the course content) seems to be one of the challenges for educators.

There are a lot of factors contributing to high quality lectures. One starting point could be the OECD-IMHE project and the report on the quality of teaching in higher education [2]. Another strategy could be to improve the learning outcome by dealing with group formation problems [3] or by flipping the classroom [4] (just to mention two of the many), but one key issue is that teaching can also be seen as a process (or even better: as a set of related processes) involving the educators, the environment and the learner [5]. However, the notion of a process is not always seen by educators (even so by the learners) and key ideas stemming from the field of maturity models (e.g. measurement steps, quality improvement, or generic/specific practices to be followed) are quite often underestimated or neglected. In order to come up with a maturity model for teaching that is finally accepted by all stakeholders, relevant practices and goals have to be identified ... and validated, which is now done in Klagenfurt step by step for several practices and goals [6]. This contribution can be seen as part of a larger effort in identifying (and measuring the impact of) practices and in the presented study we are looking at task solving best practices in our software engineering laboratory classes.

The objectives of this paper are twofold. At first, we want to show up ways in improving the outcomes of (software engineering) lectures. To begin with, we do this by borrowing practices from CMMI, and by looking at those factors that contribute to the students' performances during our classes. Secondly, we want to foster the idea of treating teaching as a process that can be positively influenced by educators and learners. We do this by demonstrating how easy it is to measure at least sub-processes and by showing the significance of process maturity on the final outcome or grades during a course. For this, in 2016, we defined a maturity framework and started a small experiment with 32 students that was then repeated with 140 students in Košice, and lately with 22 students again from Klagenfurt.

The paper is structured as follows. Section II summarizes related work in the field of maturity models and education. Section III presents the structure, details and results of the study. Then, section IV reflects on the findings but also threats to validity, and section V concludes the paper with a summary of the findings and future work.

II. Related Work

In the year 1993, members from industry, government and the Software Engineering Institute (SEI) worked together and created an institutionalized and stable model named CMM (Capability Maturity Model) [7]. The model served as a basis for improving the quality of software processes. Later, SEI presented the integrated version called CMMI (Capability Maturity Model Integration), addressing the quality of a software process in terms of Capability and Maturity levels.

The CMMI model has already served as a framework for creating Capability Maturity models in the educational field. Chen et al. [5] present a capability maturity model which focuses on improving the teaching process for teachers in higher education. Their model is called Teaching Capability Maturity Model (T-CMM) and it is still under development. In Klagenfurt, we are working on a similar model (borrowing from CMMI-Services), called TeaM-Model [6], but we are also including primary and secondary school teachers. The model is also work in progress and currently we are identifying and evaluating first specific and generic practices.

Higher education organizations adapt the CMMI model based on their needs for the improvement of the organization or the syllabus [8],[9],[10], too. However, those models all address the maturity of the organization or the syllabus in higher education. Many other examples address the course design in higher education either in a classroom environment [11] or in online courses design (that would be CMMI in e-learning) [12], [13], [14].

There are also some CMMI-like implementation models for primary and secondary education. Nevertheless, they focus either on the organizational level or on the syllabus, but they do not consider the teachers and their teaching process issue [15], [16], [17].

To summarize, apart from the T-CMM and TeaM models, other approaches either address the maturity on the organizational or on the curricular level. But, to the best of our knowledge, none of them takes a closer look at the different effects of the necessary practices (that are part of teaching processes for teachers and learners).

III. The Study

In order to find out more about the influence of different teaching practices in software engineering, we decided to make use of lecture units that are part of standard software engineering courses at both institutions, the Technical University Košice and the Alpen-Adria-Universität Klagenfurt. The units are using the AMEISE framework [18] to address topics like process models, software quality and project management. As the AMEISE-related part of the lecture (the lab part) is in use since 2003 without major changes, there is the advantage of a rich set of data (baseline) concerning the performance and outcome of the learners, and the teaching skills of the educators. It also has the advantage that the design of this lecture-part is comparable to other practical classes (or labs) at our universities, and we assume that the five factors (see below) that we started to observe in this study are important for the other lectures, too.

A. The TeaM Model Context

Even though the TeaM Model is not in the focus of this work, we are nevertheless looking at some of its practices. It also is related to some of our recommendations at the end of this work, so that it makes sense to take a closer look at the model first.

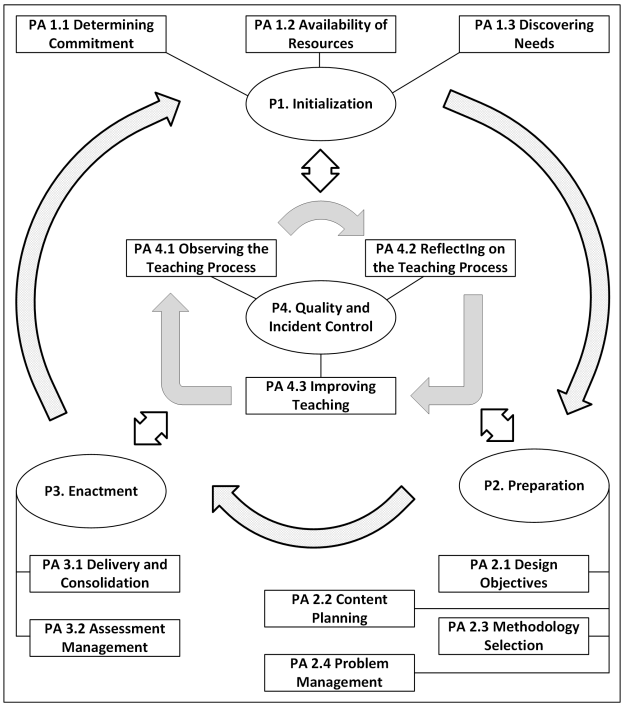


Fig. 1. Process model and phases behind TeaM [6]. The phases P1 to P4 are split into 12 process areas. In this work we are investigating the usability of phase 4 and its influence on phase 3.

The TeaM model is built up from the necessity of some standards to address the quality of teaching. The particularity of the model is on addressing quality by considering the teaching process as a whole with regard to teachers at university, primary and secondary schools. Making use of the model then either helps the educational institution in evaluating and improving its quality of teaching (by, when required, producing a ranking), or it helps teachers to evaluate and improve their teaching process by their own. Within the TeaM model, the teaching process is composed by four phases (see Figure 1): Initialization – where administrative issues are managed and defined; Preparation – where teachers plan and prepare for the course; Enactment – there the implementation of the teaching units takes place (and which is also in the focus of our efforts); Quality and Incident Control – here, possible incidents and the teaching process itself are observed, analyzed and refined (again, this drives partially our research efforts).

For each of these phases, factors related to the quality of teaching are determined, and in the TeaM terminology they are called Process Areas (PAs). In total, 12 PAs are already established as well as the associated goals and activities (practices) for each of the PAs. The TeaM model observes the implementation of these practices and goals by two forms of representations: continues representation (Capability Level), where only one PA is assessed and managed for improvement, and stage representation (Maturity Level), where a set of PAs are assessed and managed for improvement. In such away, a Maturity Level is achieved when all the PAs assigned to that level reach the maximum Capability Level.

A detail description of the TeaM model can be found at the website[1] of the TeaM project. However, for the scope of this work it is just important to know that, firstly, we are conducting this study to learn more about the effects of practices as defined in phases 3 and 4, and secondly, we are collecting experience when applying the model to our own lectures.

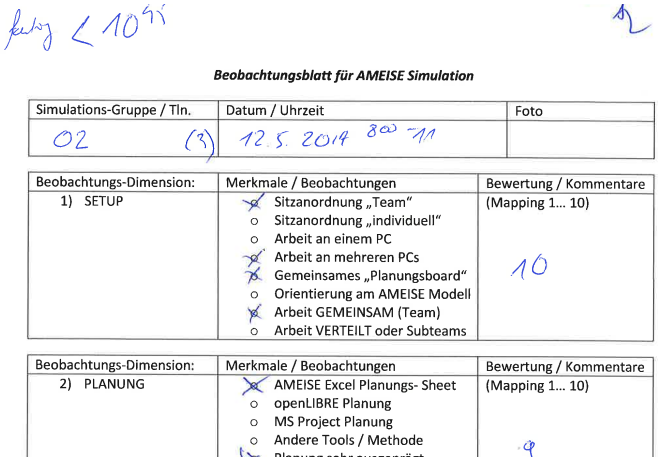
B. Simplified Task-solving Maturity Model

Together with an internationally experienced CMMI expert (one of the co-authors), we developed a simple model for estimating the maturity of a task-solving process (in order to add further evidence to the TeaM model described above). The selection of the different observation factors was driven by a book from Tom DeMarco et al. about patterns of project behavior [19]. And, it was based on some assumptions that we wanted to validate for an educational setting. To be more precise here, our industrial software development project experiences show that the environmental setting and team factors have high influence on the results of the achieved project metrics. Also, planning discipline and especially the updates of the planning are important. The right speed is also seen as an important factor, as this implies that the team takes enough time for facts based discussion but does not spend time in ”endless” discussions. So, to begin with, our model consists of the following five dimensions (being mapped to the educational context):

- Environmental Setup. Here, we assess how well the students are customizing their environment (and how well they adjust to the environment). We look at the seating arrangement, the use of technical equipment, the use of planning boards, and whether they are largely working in teams or more as individuals.
- Planning. Here, we assess how well the students are prepared for the task. We look at their plans (if any and their granularity), if they make notes (how) and whether they keep their plans up to date or not.
- Speed. From a baseline of more than thousand task solution strategies (within the context of our lectures) we extracted the average time that former (successful) teams spent on the (same) tasks. This dimension assesses, for two measure points during the task, if the working speed seems to be reasonable, meaning comparable to other successful groups, or not.
- Discussion. Here, we assess how intense the students are discussing their actions. We look for leaders, analysis discussions, and on which basis decisions are taken (chaotic, fast, fact-based).
- Mood. This dimension looks at the attitude and sentiment of the students. Are they humorous, do they laugh, are they engages and active, or are they bored or anxious.

These dimension are assessed separately by making use of a standardized observation sheet (a small example including

[1]See the TeaM Model Project page for access to the raw data: http://iid.aau.at/bin/view/Main/Projects.



Beobachtungsblatt für AMEISE Simulation

Simulations-Gruppe / Tln.	Datum / Uhrzeit	Foto
02 (3)	12.5.2014 8⁰⁰ - 11	

Beobachtungs-Dimension:	Merkmale / Beobachtungen	Bewertung / Kommentare
1) SETUP	☒ Sitzanordnung „Team“ ○ Sitzanordnung „individuell“ ○ Arbeit an einem PC ☒ Arbeit an mehreren PCs ☒ Gemeinsames „Planungsboard“ ○ Orientierung am AMEISE Modell ☒ Arbeit GEMEINSAM (Team) ○ Arbeit VERTEILT oder Subteams	(Mapping 1... 10) 10

Beobachtungs-Dimension:	Merkmale / Beobachtungen	Bewertung / Kommentare
2) PLANUNG	☒ AMEISE Excel Planungs- Sheet ○ openLIBRE Planung ○ MS Project Planung ○ Andere Tools / Methode	(Mapping 1... 10) 9

Fig. 2. Original observation sheet that we used for our simplified task solving maturity model (in German). You can see the first dimension (”1) SETUP”) with characteristics (”Merkmale/Beobachtungen”) that we specifically were looking for, and the points received (”Bewertung/Kommentare”). The group ”02” organized their environment as a team, was using several computers, they had a common planning board and they also worked together as a team. The assessor valued that facette with 10 points. You also see part of the planning dimension (”2) PLANUNG”). There, the assessor stated that the group was only using a simple AMEISE-Excel sheet for the planning, but it did not use MS Project or Libre-Office – which are also learning goals of the lecture.

a translation to English can be seen in Figure 2) and on the form the observations are mapped to a scale between 1 and 10 points (where 10 points means the fulfilment of all positive characteristics). Finally, the points are summed up, leading to maturity points between 5 and 50. It would have been possible to weight the dimensions, but for reasons of simplicity we decided to treat every dimension the same.

In this section we now present the setting of the study. The generated statistics are also included, however, for reasons of space we have to refer to the TeaM project page (see above) for downloading the raw data used for calculating the statistics and scatter-plots.

C. Research Objectives

Having the above described maturity model in mind, we decided to conduct a study in order to answer the question, if a maturity model can also be applied to our software engineering courses. For this, the following two questions needed to be answered:

- Is there a correlation between the maturity points and the overall performance in the course?
- How strong is the influence of the selected five different dimensions on each other and on the overall lab-course performance?

As a side-effect we also hoped that the findings help in convincing our students that maturity models are useful. Thus, a follow-up question was

- Is the model perceived useful and did the students make use of the findings in their project work?

The simple model introduced above comes with two assumptions. Firstly, we assume that the overall performance in

the lab-course is mainly driven by the fulfilment of the tasks during the lab-course, and the final examination and/or grading is related to the tasks. Secondly, we are aware of the fact that there are many more factors and dimensions that, when considered, lead to good course results and, additionally, lead to good grades at the end of a course. However, we assume that the above mentioned dimensions (borrowed from the field of CMMI-DEV, Product Development) are, among those that are easy to observe and to rate by an educator, the most important ones. Dimensions might be added (or removed) later on.

As mentioned above, we tried to improve the learning outcome of our lectures. In fact, in all of our lectures, the learning goals focus on software engineering and software project management skills. This includes understanding and working with process and maturity models, using quality assurance activities during a software project, planning (developers, schedule, costs, quality) and managing software projects (including tool support). The advantage of the AMEISE framework [18] is that it covers nearly all the topics (except the use of planning tools) for the simulated software project, and that it returns an assessment report containing percentage values for the achievement of the different project goals. At the end of the lecture we combine these results with an assessment of documents (project plans) handed in by the students during the course.

D. Setting

In order to find out whether our model works or not, we decided to start a small pre-study involving 36 students (12 teams) and one assessor in Klagenfurt in the summer term 2016. Then, to avoid a Klagenfurt bias, we repeated the study in Košice with 140 students (66 teams) and a different assessor during the winter term 2016/17. Finally, we looked again at a lecture in Klagenfurt with 20 students (8 teams) and ratings from both assessors during the summer term 2017.

We have been choosing lectures making use of the AMEISE environment as AMEISE provides the students with results of their actions as project manager in form of metrics - especially quality related metrics of their simulated software development. This can be seen as a kind of ”white box analysis”.

For the pre-study in Klagenfurt we used two courses that take place at the beginning and the end of our Masters program in Applied Informatics. In total, 36 students were split (randomly) into 12 teams and informed that they are taking part in a study (without knowing the background. They were also asked for permission for being pictured. After some introduction units, the task of the teams was to plan and prepare for a small-sized (10.000 lines of code) software development project with focus on quality assurance as a homework. After one week of planning (and handing in a mandatory project plan), they had to take over the role of a project manager and had to conduct the project within the AMEISE simulation environment in our laboratories. During the simulation runs they were observed and assessed according to a standard form prepared by our CMMI-expert. After the simulation runs, the students' performances were graded according to a fine-granular grading scheme (that is in use now in Klagenfurt since 2003) yielding points and grades. In a follow-up lecture, the performances of the teams were thoroughly reflected on, and also the background and results from the study were explained. After that, the teams were instructed to repeat the task (planning for a project, conducting a project) and again, at the end, their performances were assessed and reflected on.

The study in Košice followed the same course layout, and the student teams had to fulfil the same tasks as in Klagenfurt. They were also informed about being part of a study and asked for their permission of being pictured. However, due to the large number of teams (and the availability of the lab rooms), the simulation runs took place in Košice on three successive days, and in most cases the students worked together in teams of two. Again, during the lab classes, the teams were observed, assessed according to our standard form, and afterwards their performance was evaluated based on the Klagenfurt's grading scheme.

The last part of our study took again place in Klagenfurt and followed the same pattern. The same tasks had to be fulfilled by the students (randomly assigned to teams of 2 to 3 students), but this time two assessors observed the teams during their work independently. This was done to find out whether the assessment form that we were using helps in avoiding an assessor's bias in the rating.

After that, all the data (group assignments, assessment data, performance data) was collected using Microsoft Excel, and statistical tests were used to identify potential correlations. The following section briefly summarizes the tools that we used.

E. Statistics

Within the scope of this contribution four different statistical tests were used to assess the data: the Shapiro-Wilk parametric hypothesis test, the Pearson's Correlation Coefficient, the Spearman's Rank Correlation Coefficient, and Kendall's Tau Correlation Coefficient.

Assessing the assumption of normality is required by most statistical procedures, e.g. the linear regression analysis that we are using. When the normality assumption is violated, interpretation and inferences may not be reliable or valid. According to Razali and Wah [20], the Shapiro-Wilk test (SW) belongs to the most powerful normality tests available.

The Pearson's correlation coefficient Rho (R_P) measures the degree of association between the variables, assuming normal distribution of the values [21, p. 212]. Though this test might not necessarily fail when the data is not normally distributed, the Pearson's test only looks for a linear correlation. It might indicate no correlation even if the data is correlated in a non-linear manner.

As we will see in the remaining section, not all data is normally distributed. To handle this case the Spearman's rank correlation coefficient Rho (R_S) has been chosen [21, p. 219]. It is a non-parametric test of correlation and assesses how well

a monotonic function describes the association between the variables. This is done by ranking the sample data separately for each variable.

Finally, Kendall's robust correlation coefficient Rho (R_K) can be used as an alternative to the Spearman's test [22, p. 200]. It is also non-parametric and investigates the relationship among pairs of data. However, it ranks the data relatively and is able to identify partial correlations.

When there is no likelihood of confusion, then R will be used to refer to either R_P, R_S or R_K.

In the remainder of this work the correlation R is interpreted as follows:

- When $|R| \in [0.7,\ 1.0]$ it is interpreted to indicate a *strong association*.
- When $|R| \in [0.4,\ 0.7)$ it is interpreted to indicate a *medium association*.
- When $|R| \in [0.0,\ 0.4)$ it is interpreted to indicate a *weak association*.

In addition to the values of the correlation R, also the significance level (p) of the value is provided (checking, within the scope of the null hypothesis, that the probability of the value of R is bigger or equal to the observed value of R). The values in the following tables are rounded to the third decimal place (which means that a value of $p = 0.0005$ would become $p = 0.001$).

F. Results

After the course, the assessment reports of the two assessors were collected and the ratings were checked and verified against the pictures that were taken during the courses. Finally, the ratings and the performance points/grades were recorded electronically. Matlab R2007b was used for testing for normality and the correlations.

Looking at the five different dimensions, the normality tests do not indicate normal distribution. Looking at the maturity points, the test indicates that the sample has been drawn from a normal distribution with a mean of 37.702 and a variance of 14.573. This confirms that the selection of different correlation tests makes sense. We are looking for linear, non-linear and even partial correlation in the data.

In a first step, scatter plots were produced to visualize possible relations between the different dimensions and to get a feeling about how well the resulting dimensions are associated to each other. One of our assumptions (and also assumptions in other maturity models) was that the environmental setup has a major impact on the overall performance. Figure 3 shows a scatter plot where, on the x-axis, we assigned the points (from 1 to 10) for the environmental setup and on the y axis we assigned the points (on a scale between 0 and 200) for the students' overall performances. The plot indicates some positive relation. When looking at the planning dimension we get a quite similar plot (see Figure 4).

Another assumption was that the working speed (also indicating how well students are prepared for a task) has a major impact on the overall performance. However, when looking

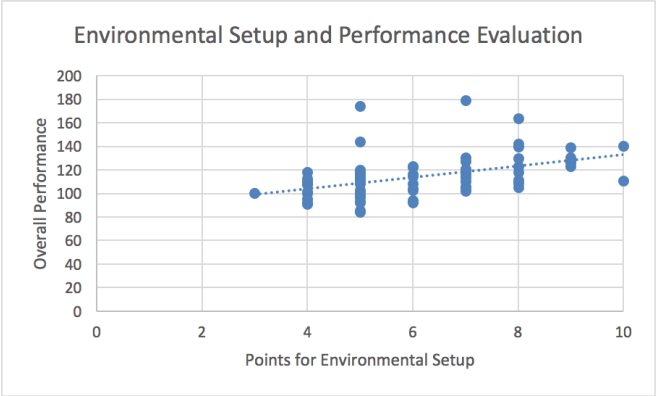


Fig. 3. Scatter-plot for visualizing the correlation between the environmental setup dimension and the course performance/grading. (n = 84)

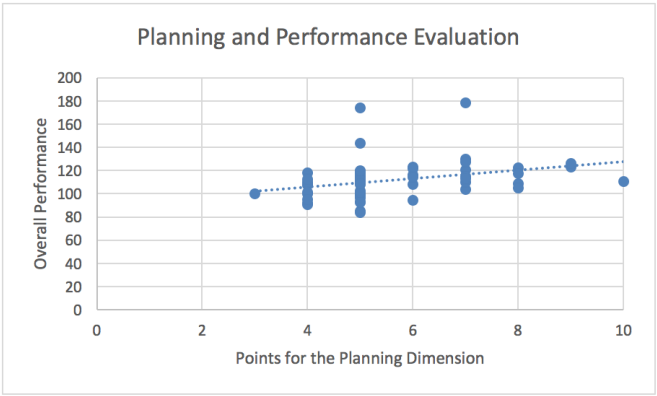


Fig. 4. Scatter-plot for visualizing the correlation between the planning level and the course performance/grading. (n = 84)

at Figure 5, a linear trend-line indicates a slightly negative effect of speed on the result, but one might have difficulties in detecting any positive or negative correlation.

Figure 6 depicts the relation between the atmosphere in the team when working on the tasks and the overall performance. As one can see, there is some relation, not necessarily strong, but also a linear trend-line indicates some increase in performance when the atmosphere is at a higher level. The scatter-plot for the relation between the discussion dimension is quite similar (see Figure 7).

Finally, we looked at the relation between the maturity points (which are linear aggregations of the points for the separate dimensions) and the overall performance. Figure 8 shows that there again seems to be a correlation between these two dimensions, and also the linear trend-line shows that with higher maturity one is very likely to achieve higher points during the lecture.

The scatter plots were very useful for us to get a first feeling for the importance of the different dimensions. However, the data points are still quite close to each other and looking at the results of the statistical tests is necessary. The tables in Figures 9, 10 and 11 summarize the results for the Pearson, Spearman- and Kendall tests. Before going into details (in Section IV) we

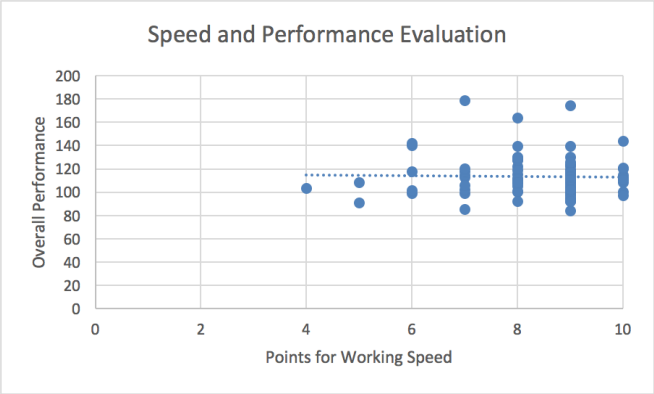


Fig. 5. Scatter-plot for visualizing the correlation between the working speed and the course performance/grading. (n = 84)

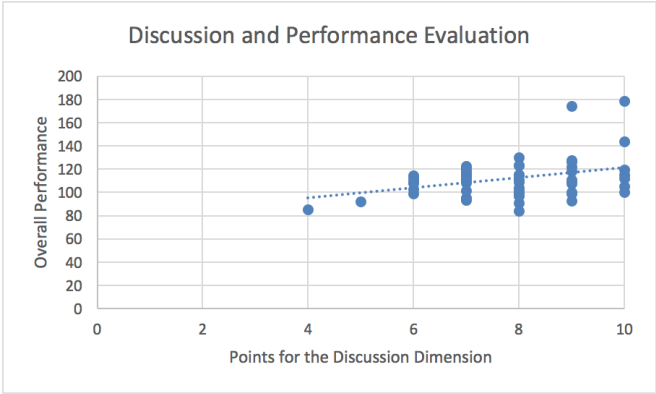


Fig. 7. Scatter-plot for visualizing the correlation between the discussion level and the course performance/grading. (n = 84)

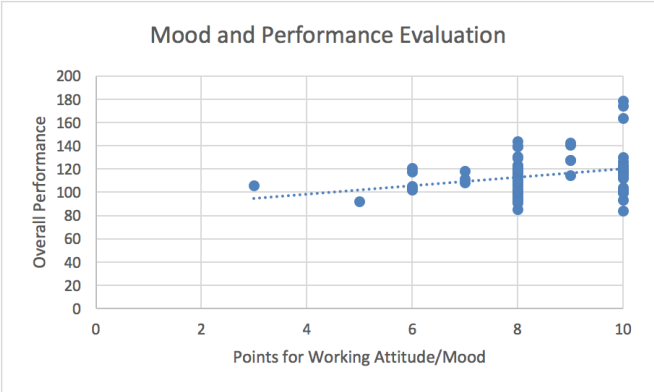


Fig. 6. Scatter-plot for visualizing the correlation between the working attitude/mood and the course performance/grading. (n = 84)

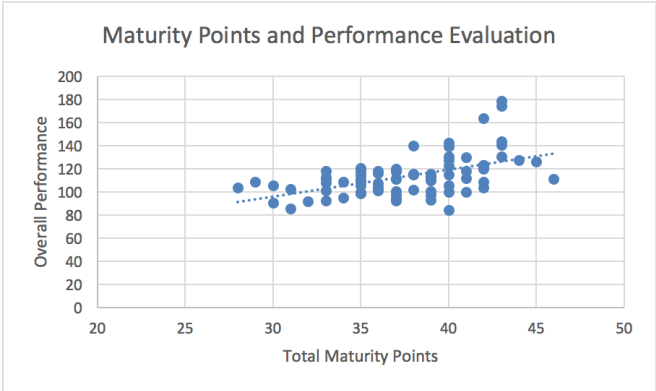


Fig. 8. Scatter-plot for visualizing the correlation between the maturity points and the course performance/grading. (n = 84)

can state that the statistical tests more or less confirmed the observations of the scatter plots. But they also indicate that some of the findings might not be statistically relevant.

The tables also include the results of the tests of the null hypotheses (p-values), and values where the p-val is not within the 5 percentage level are colored red. This indicates that the Rho-values have to be interpreted with care. For all three correlation tests, the speed dimension has the highest chance that the null-hypothesis is confirmed and that the result (that speed has some minor negative influence on the performance) is not relevant. And, for two tests (the Spearman and the Kendall test) the same uncertainty holds for the discussion dimension.

The highest correlation values of R are, for all three tests, to be found between the maturity points and the performance points. They are between 0.524 and 0.549 for the Pearson and Spearman tests, indicating a positive medium linear relation, and 0.394 for the Kendall test, indicating a positive weak partial relation. The next strongest dimensions are the planning and setup dimension, and all three null-hypothesis tests indicate that the results are statistically significant. Their values are within the range of 0.305 and 0.499, indicating again weak to medium (linear) relations.

IV. Reflection

The previous section already gave a first insight into the data set. In this section, we discuss the findings and put them also into the context of our lectures. As mentioned in the introduction, there are many more aspects contributing to course performances, so we also mention those aspects that we took and that we did not yet take into considerations.

A. Discussion

The first sub-question that we tried to answer was if there is any correlation between the maturity points that we observed and the performance in the courses. We were using three different tests for correlation as only the maturity points are normally distributed[2]. In our case, the Pearson test indicates that there might be a medium-size (linear) correlation between the maturity points and the course points ($R_P = 0.524$, $p = 0.000$), and the Spearman test confirms this observation, even with some slightly higher correlation value ($R_S = 0.549$, $p = 0.000$). The Kendall test, looking for at least partial correlations, however, only indicates some weak partial correlation

[2]At the end of the lecture our students, of course, do get grades, mainly based on the performance points. The Shapiro-Wilk test then indicates that the grades are normally distributed.

Pearson Rho	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	1,000	0,281	-0,038	0,173	-0,129	0,514	0,448
Setup		1,000	0,170	0,184	0,281	0,679	0,473
Speed			1,000	0,210	0,133	0,488	-0,022
Discussion				1,000	0,221	0,627	0,250
Mood					1,000	0,508	0,277
Maturity						1,000	0,524
Points							1,000

Pearson p-val	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	0,000	0,010	0,730	0,116	0,242	0,000	0,000
Setup		0,000	0,123	0,094	0,010	0,000	0,000
Speed			0,000	0,055	0,228	0,000	0,842
Discussion				0,000	0,043	0,000	0,022
Mood					0,000	0,000	0,011
Maturity						0,000	0,000
Points							0,000

Fig. 9. Pearson Rho and p values for the five different dimensions, the maturity level, and the performance/grading for the tasks. The colored results have to be taken with care as the null-hypothesis test shows no significance within the 0.05 level. (n = 84)

Spearman Rho	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	1,000	0,252	-0,049	0,179	-0,120	0,496	0,499
Setup		1,000	0,151	0,186	0,309	0,692	0,408
Speed			1,000	0,124	0,038	0,379	-0,019
Discussion				1,000	0,211	0,599	0,202
Mood					1,000	0,534	0,248
Maturity						1,000	0,549
Points							1,000

Spearman p-val	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	0,000	0,021	0,657	0,103	0,278	0,000	0,000
Setup		0,000	0,172	0,091	0,004	0,000	0,000
Speed			0,000	0,260	0,730	0,000	0,862
Discussion				0,000	0,055	0,000	0,066
Mood					0,000	0,000	0,023
Maturity						0,000	0,000
Points							0,000

Fig. 10. Spearman Rho and p values for the five different dimensions, the maturity level, and the performance/grading for the tasks. The colored results have to be taken with care as the null-hypothesis test shows no significance within the 0.05 level. (n = 84)

($R_K = 0.394$, $p = 0.000$). In all, one could say that in our lectures it payed off to reach higher maturity points as the likelihood of earning more performance points (and later on a better grade) for the tasks is higher.

The second sub-question is more difficult to answer as in several cases the correlation tests might not be significant within the 5 percentage level of the null-hypothesis. However, all three tests show that the two dimensions, Preparation/Planning and Environmental Setup have the strongest influence onto the task performances.

- Environmental Setup. In our courses and for the first (simulated) software development projects, we leave it up to the students how well they are prepared (they have to hand-in a project plan, but the plans vary in details) and how they change and use their working environment. One might argue that there might not much to be done at University's lab rooms, but Figures 12 and 13 show two examples of how the *environmental setup* might change – or not. The first group in Figure 12 moved their tables to a wall, made use of the white-board and fixed their project plans and important documents onto it. The second group in Figure 13 just sat down to one of the tables, used their laptops, but did not make their environment more suitable to solving a complex task (lasting at least 4 hours). Taking a closer look at the environmental setup, one can see that it also seems to weakly relate (R between 0.390 and 0.254, on a 1 percent level) to the mood and attitude to the task. From what we knew from industrial experience (and the relation between the working climate and environment), we expected an even higher correlation which we did not find in our courses. However, there at least is a weak (positive) correlation with high significance confirming the general guidelines one finds in common project management and software engineering textbooks.

Kendall Rho	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	1,000	0,200	-0,040	0,145	-0,097	0,397	0,380
Setup		1,000	0,125	0,148	0,254	0,562	0,305
Speed			1,000	0,097	0,036	0,308	-0,018
Discussion				1,000	0,180	0,483	0,157
Mood					1,000	0,431	0,199
Maturity						1,000	0,394
Points							1,000

Kendall p-val	Planning	Setup	Speed	Disc.	Mood	Maturity	Points
Planning	0,000	0,024	0,656	0,101	0,293	0,000	0,000
Setup		0,000	0,164	0,095	0,006	0,000	0,000
Speed			0,000	0,281	0,706	0,000	0,825
Discussion				0,000	0,051	0,000	0,053
Mood					0,000	0,000	0,018
Maturity						0,000	0,000
Points							0,000

Fig. 11. Kendall Rho and p values for the five different dimensions, the maturity level, and the performance/grading for the tasks. The colored results have to be taken with care as the null-hypothesis test shows no significance within the 0.05 level. (n = 84)

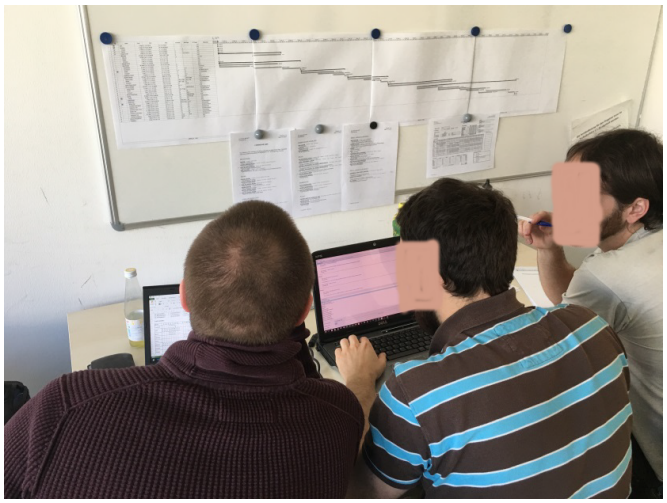


Fig. 12. Example of a team that modified their working environment so that it matches the team's needs. The team moved tables and chairs, made use of the wall and worked closely together.

Fig. 13. Example of a team that did not change the lab's setting. They used their laptops, but no additional aids. The team did not really work closely together.

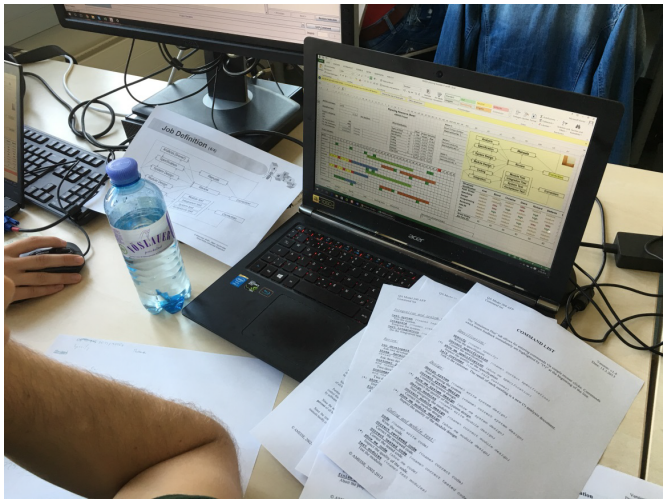


Fig. 14. Example of a group that made use of several planning aids and that kept track of their project (progress).

- Planning/Preparation. As mentioned above, another medium-size and positive correlation can be found between the *planning dimension* and the performance points. The Pearson test indicates that there might be a medium-size (linear) correlation between the planning points and the course points ($R_P = 0.448$, $p = 0.000$), and the Spearman test confirms this observation, even with some slightly higher correlation value ($R_S = 0.499$, $p = 0.000$). The Kendall test, looking for partial correlations indicates some weak partial correlation ($R_K = 0.380$, $p = 0.000$). This strengthens our opinion in choosing the planning quality as being part of our model, and also shows some evidence to our students that investing more effort in preparation really pays off. Looking closer at the planning dimension, one can also observe a weak influence of planning and preparation onto the environmental setup. This is not so astonishing as one might assume that those groups that invested more effort in their plans were also eager to work with their plans. In our case, the data seems to confirm this assumption. Figures 14 and 15 show two examples of how planning and preparation might differ between two groups. Even though the project plans that the groups handed in were comparable (on the level of quality and details), the first team prepared for taking notes, adjusting their plan and re-planning their project, whereas the second group just relayed on their original plan, taking no notes, and not planning for reflections on what they were doing.
- Mood. The next dimension that had some weak (and positive) influence on the team performance was the working climate (mood, attitude) within the team. The Rho values are between 0.277 and 0.199 (on a 1 to 2 percent significance level). Again, here we expected a higher correlation, but in our courses we confirmed at least a weak influence.
- Discussion. Coming to the discussion dimension, only the Pearson test indicates some weak correlation ($R_P = 0.250$, $p = 0.022$), the other tests show comparable (weak) relations, but they are not within the 5 percent significance level. The result thus has to be taken with care. We still think that group discussion is worthwhile during task solving, but we think that our measurement procedure lacks. As one assessor is only able to observe the discussions that is going on within a team at specific point of time (and not constantly for all groups for several hours), the assessment of this dimension as such is a snapshot with a high degree of failure.
- Speed. Our assumption was that the working speed also has some effect on the overall quality and results of the task fulfilment. However, all three test indicate no or only a very small, even negative, effect on the task performance, and this at a very low significance level ($p = 0.862$ for the Pearson test). Working speed, as we defined it, seems to be a useless discriminator for estimating the team's performance. On the other side, we can interpret the result that way: during our courses, the different student groups followed their own optimal time-plan/schedule for fulfilling their tasks, and, if any, being faster than others not necessarily yields better overall performance.

The follow-up question was answered by looking at standardized (quantitative and qualitative) feedback that is collected at the end of all of our lectures. It turned out that the application of the model had a very positive effect on our students. They perceived the lecture units as excellent. However, from an educator's point of view, what was more important is the fact that they, after the results of the study were presented, in the remainder of the lecture they tried to follow the practices, which in turn also raised their performance level compared to the baseline we collected from the lectures the years before.

To summarize, we can answer sub-question one in such a way that there is some medium size and positive linear correlation between the maturity points in our lectures and the task performance (also measures in points). Sub-question two can be answered as follows: apart from the speed and eventually the discussion dimension, there seems to be a weak correlation between the mood dimension and the task performance, and a medium-size influence of the preparation and setup dimension onto the task performance. And, coming to the final question, the model was perceived of being useful and the students tried to make use of the findings.

B. Recommendations

The identification of relations between the different dimensions onto the task solving performance was a first important step. However, the idea behind this measurement effort was not just about collecting data. It was about improving our software engineering lectures and improving the student's motivation when talking about maturity models.

Even though our model is a small part taken from a larger set of specific and generic practices of the TeaM Model, it contains enough evidence to change (improve?) our lectures at bit. We recommend the following and see opportunities on several levels:

- Motivation. Teaching and talking about maturity models is a challenge for educators – and students. One problem is that the models are already quite complex, and from a neuro-didactic point of view hands-on experience (e.g. experiencing what it means when getting from CMMI-level 2 to level 3) would be needed, something that is rarely possible in our software engineering courses. A down-scaled model just focusing on a couple of practices (in our case task solving strategies) might be enough to explain how even small steps might improve the outcome. In two of our three courses, we transparently communicated what we had been doing to the students, presenting the model and also reflecting on the outcome. The qualitative feedback that we collected confirms that the students got the point, also applying the suggested practices with more care in the follow-up tasks.
- Increasing performance. Though our results are not generalizable to all other different types of courses, we think that improving the observed dimensions in courses with a similar layout (tasks with team work, planing and preparation phase, quantifiable outcome/deliverables) will lead to a higher performance of the students. Compared to the same courses (before winter term 2016) where we did not consider and communicate the different dimensions, the results of the students were slightly lower (average points of 110.182 with a standard deviation of $\sigma = 28.193$ compared to 139.42 points and $\sigma = 28.707$.
- Course Quality. The different Teaching Maturity Models all contain some kind of quality control process areas. Quantifying effects of changes are also an important part of our TeaM model. The suggested model could be a starting point for educators to (transparently) evaluate their courses (and recommendations to students) and so to increase the quality of their teaching. In our case, the end-semester evaluations demonstrate that the students are highly satisfied with those parts of the lectures and attest its excellence also again on a quantifiable level.
- Training for the job. Another helpful side-effect of this method to introduce a maturity model (borrowed from CMMI) to the students in combination with our observation and photo documentation was also that one is able to discuss with the students to pay attention (e.g. as a team leader) to the behavior of the team during the development activities. Such soft skills - as a black box view – and being able to analyze the working style of a team by observing the above mentioned dimensions are especially required by a Scrum master in an Agile team.

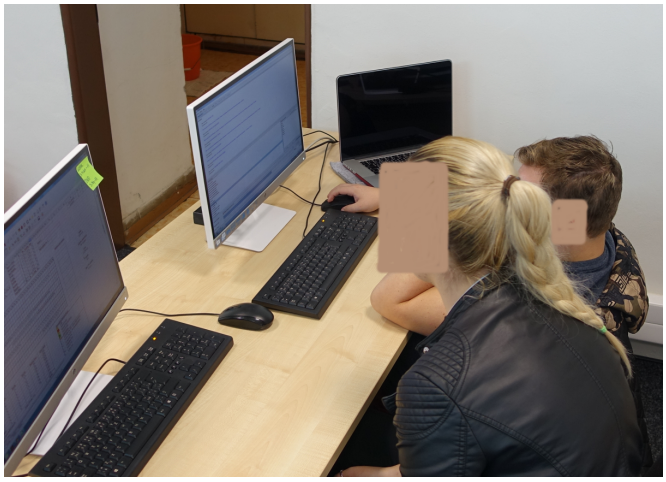


Fig. 15. Example of a group that made no use of planning aids. They also did not keep track of their project (progress).

C. Validity

As mentioned previously, the results of the study have to be taken with some care. There are many more factors contributing to a good lecture and to good course results of our students. Being aware of that, we carefully designed the study in such a way that we at least reduced those factors that we were able to control.

Such factors include naturally the content of the course with all the available materials and tasks. In our case, we made use of a standard lecture that took place in the same manner and basically with the same materials and tasks for several years at both institutions.

The pre-knowledge of the participating students of course varied. From the technical point of view we made sure (based on some questionnaires before the start of the lecture) that the tasks are manageable. However, one notable difference is that the courses in Košice took place in English, and some of the students there had problems with it. In Klagenfurt, we once gave the lecture in English, once in German, without noticing any problems.

One of the most important influence factors is the educator him- or herself. In our case, even though the experiments took place at two different institutions, the educator was the same person. The educator is highly experienced in the topic and knows the background of tasks to be fulfilled during the courses, so that we assume that the lecturer bias can be neglected in our case.

What remains is the fact, that the assessment might lead to different results when done by different assessors. This is a general problem that we also looked at. In our case we had two assessors (one in summer term 2016, another one in winter term 2016) and we checked for the last semester (2017), where both assessors used the evaluation sheet, whether the two – independent – ratings were similar to each other or not. Apart from some exceptional cases where the observation time differed, there were no differences in the rating - demonstrating also the usefulness of the rating sheet.

V. Conclusion

This paper reports on a study and teaching approach, where we were applying a (down-scaled) version of a maturity model on the students work in our software engineering laboratory classes. As part of a larger teaching maturity model, we defined five dimensions (planning, setup, speed, discussion, and mood) to be observed, and looked for associations between these dimensions and the overall course performance. It turned out that, apart from speed, medium-sized correlations between most of the measures and the course performance exist.

Moreover, we noticed that the students were surprised to experience the use(-fulness) of a maturity model at first hand (in lectures that are also about quality and maturity models). They were eager to make use of the practices in their own course works, yielding better results at the end. For that reason, even though our own teaching maturity model is at the very beginning, we recommend to transparently introduce an "observation – reflection – improvement" cycle in any lectures and to start measuring the above mentioned factors.

For future work, we are looking closer at other influence factors (e.g. personality factors), and together with teachers we are collecting further specific and generic practices to be followed. As improving the quality of teaching is an ongoing process, any observation and feedback helps us in moving one step forward.

References

[1] K. Pretz, "Special report engineering education," *the institute*, vol. 40, no. 3, p. 2, September 2016.

[2] F. Henard and S. Leprince-Ringuet, "The path to quality teaching in higher education. A literature review paper on quality teaching," OECD, Institutional Management for Higher Education (IMHE). Retrieved from: https://www1.oecd.org/edu/imhe/44150246.pdf, Tech. Rep., 2008.

[3] A. Mujkanovic and A. Bollin, "Improving Learning Outcomes Through Systematic Group Reformation: The Role of Skills and Personality in Software Engineering Education," in *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '16. New York, NY, USA: ACM, 2016, pp. 97–103. [Online]. Available: http://doi.acm.org/10.1145/2897586.2897615

[4] B. Kerr, "The flipped classroom in engineering education: A survey of the research," in *Proceedings of the 2015 International Conference on Interactive Collaborative Learning*, 2015, pp. 815–818.

[5] C.-Y. Chen, P.-C. Chen, and P.-Y. Chen, "Teaching Quality in Higher Education: An Introductory Review on a Process-Oriented Teaching-Quality Model," *Total Quality Management and Business Excellence*, vol. 25, no. 1–2, pp. 35–56, 2014.

[6] E. Reci and A. Bollin, "A Teaching Maturity Model for Informatics Teachers in Primary and Secondary Education," in *Proceedings of the The 9th International Conference on Informatics in Schools, ISSEP 2016*, 2016.

[7] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "Capability maturity model, version 1.1," *IEEE Softw.*, vol. 10, no. 4, pp. 18–27, Jul. 1993. [Online]. Available: http://dx.doi.org/10.1109/52.219617

[8] C. Lutteroth, A. Luxton-Reilly, G. Dobbie, and J. Hamer, "A maturity model for computing education," in *Proceedings of the Ninth Australasian Conference on Computing Education - Volume 66*, ser. ACE '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 107–114. [Online]. Available: http://dl.acm.org/citation.cfm?id=1273672.1273685

[9] P. V. M. Nuno Duarte, "Towards a maturity model for higher education institutions," in *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering Doctoral Consortium*, 2011.

[10] T. C. Ling, Y. Y. Jusoh, R. Abdullah, and N. H. Alwi, "A Review Study: Applying Capability Maturity Model In Curriculum Design Process For Higher Education ," *Journal For The Advancement Of Science & Arts*, vol. 3, no. 1, 2012.

[11] M. M. L. Petrie, "A Model for Assessment and Incremental Improvement of Engineering and Technology Education in the Americas," in *Proceedings of Second LACCEI International Latin American and Caribbean Conference for Engineering and Technology*, 2004.

[12] C. Neuhauser, "A Maturity Model: Does It Provide a Path for Online Course Design?" *Journal of Interactive Online Learning*, vol. 3, 2004.

[13] ——, "A five-step maturity model for on-line course design," in *Proceedings of the 19th Annual Conference on Distance Teaching and Learning*, 2005.

[14] S. Marshall and G. Mitchell, "Applying spice to e-learning: An e-learning maturity model?" in *Proceedings of the Sixth Australasian Conference on Computing Education - Volume 30*, ser. ACE '04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 185–191. [Online]. Available: http://dl.acm.org/citation.cfm?id=979968.979993

[15] B. Montgomery, "Developing a Technology Integration Capability Maturity Model for K-12 Schools," Ph.D. dissertation, Published thesis. Concordia University, Montreal, Canada, 2003.

[16] B. A. White, H. E. L. Jr., P. M. Leidig, and D. M. Yarbrough, "Applicability of CMMI to the IS Curriculum: A Panel Discussion ," in *The Information Systems Education Conference (ISECON)*, 2003, pp. 1–5.

[17] M. Solar, J. Sabattin, and V. Parada, "A Maturity Model for Assessing the Use of ICT in School Education," *Educational Technology and Society*, vol. 16, no. 1, pp. 206–218, 2013.

[18] A. Bollin, E. Hochmüller, R. Mittermeir, and L. Samuelis, "Experiences with Integrating Simulation into a Software Engineering Curriculum," in *Proceedings of 25th IEEE Conference on Software Engineering Education and Training CSEE&T 2012, 17-19 April 2012, Nanjing, Jiangsu, China*, 2012, pp. 62–75.

[19] T. DeMarco, P. Hruschka, T. Lister, S. McMenamin, J. Robertson, and S. Robertson, *Adrenaline Junkies and Template Zombies: Understanding Patterns of Project Behavior*. Computer Bookshops, 2008.

[20] N. M. Razali, "Yap bee wah," *Journal of Statistical Modeling and Analytics*, vol. 2, no. 1, pp. 21–31, 2011.

[21] D. G. Rees, *Essential Statistics*, 4th ed. Chapman & Hall, 2003.

[22] N. E. Fenton and S. L. Pfleeger, *Software Metrics*, 2nd ed. Thompson Press, 1989.