Fully Automated Electronic Mock Examination in an Introductory Computer Science Course

Andreas Kämper*, Felicitas Loepp**
Munich University of Applied Sciences
* Project Team HD MINT, Dachauer Str. 100a, D-80636 München, Germany
** Department of Engineering and Management, Lothstr. 64, D-80335 München, Germany
E-mail: andreas.kaemper@hm.edu, felicitas.loepp@hm.edu

***Abstract*—A mock examination is an exam that does not count for credit. It is taken before an official examination and gives students the opportunity to practice for the later, important exam. The mock examination provides students with information on their actual learning progress and gives them in this way the opportunity to fill knowledge gaps. An electronic mock examination is the online variant usually taken unsupervised in the students' own time. In this work we present the setup of an electronic mock exam in the course "Basics of Computer Science". We discuss question types other than multiple-choice that allow for fully automated marking. Furthermore, we present an in-depth analysis of the results of the mock exam. These results demonstrate that students consider the electronic mock exam as valuable help for exam preparation. Half of the students take the mock exam even more than once and use different strategies to make the most out of it. On the other hand, lecturers get invaluable feedback from the results of the mock exam. This feedback covers all learning objectives of the entire course, comes at the right time, and is ideally suited for planning the contents of a recapitulation lecture.**

Keywords—Electronic Learning; Blended Learning; Computer Science Education; Mock Examination

I. Introduction

A mock examination is "used to practice for the real one" [1] and thus it is taken before an official examination. In such a mock exam the marks do not count, i.e. this "trial examination" does not count for credit. A mock examination is often given to the same standards and under the same controlled conditions as the official examination [2]. Hence, it gives students the opportunity to practice for the later, important examination, to get an idea of the types of questions asked [3], and to get used to regulations and locations of the official examination. Furthermore, students can practice to work under time pressure [3] with only those aids allowed to be used in the official examination. Another, even more important aim of mock examinations is to provide the students with information on their actual learning progress. Students get an idea of what they still need to learn [4, p. 190]. Due to the time gap of usually a few days [5] to a few weeks [4, p. 170] between mock and official examination, there is still the opportunity to fill these knowledge gaps. It is up to the student to use this opportunity [2].

This work was funded by the German Federal Ministry of Education and Research (BMBF), grant no. 01PL12023F "HD MINT", under the program of "Qualitätspakt Lehre" ("Teaching Quality Initiative").

An electronic mock examination, i.e. online mock exam is a variant that differs significantly from those taken on paper. Electronic mock examinations can be performed unsupervised. Students can take them at home at a time of their choice. In addition, it is way easier to allow students to take electronic mock exams more than once than for the paper-based version. Students take advantage of this opportunity, for instance, to practice only certain parts, e.g. theory questions, of the exam. In all cases of electronic mock examinations, feedback is usually also given electronically.

In this work, we describe the setup for an automatically marked electronic mock examination for the course "Basics of Computer Science". We discuss question types other than multiple-choice questions that allow for automated marking. We then evaluate the results of a pilot study with three study groups (two taught by F.L., one taught by A.K) carried out in summer term 2015. We provide a detailed analysis of the results of the mock exam for one of these study groups. In particular, we discuss how well students accepted the mock exam. We also focus on how well mock exam results correlate with results of the final examination and whether students who took the mock examination get better grades in the final exam than those who did not take part. In addition, we analyze student's strategies, to make use of the opportunity to take electronic mock exams more than once. We conclude with recommendations on the use of this type of mock examination for introductory Computer Science classes.

II. State of the Art

While mock examinations are frequently used throughout disciplines at numerous schools and universities, we could identify only few studies that perform research on their actual usefulness in higher education. The only studies on mock examinations in a Computer Science related area we are aware of are the work of Thomas et al. [2] and a series of studies by Schagaev, Bacon, and co-workers [6–8].

Thomas et al. [2] performed an experiment using an electronic mock exam in a postgraduate "Computing" course. The students could perform the mock examination in two ways. Students who sat the exam in the conventional 3-hour slot got feedback on their answers. For this purpose, a tutor marked these exams. Those students, who chose to take longer than three hours, only received sample solutions. Thomas et al. [2]

978-1-4673-8633-3/16/$31.00 ©2016 IEEE 10-13 April 2016, Abu Dhabi, UAE
2016 IEEE Global Engineering Education Conference (EDUCON)

report that the outcome of their experiment was extremely positive. The authors claim that they "put in place a system that provided additional benefits for the student."

Schagaev et al. [6] propose the Multiple Choice Answers Approach (MCAA) for Computer Science and similar disciplines. MCAA is a new assessment methodology that enables the effective use of Information Technology for assessment. Besides automated marking, the MCAA approach also allows for the automated formation of questionnaires comprised of multiple-choice questions. In a web-based enhancement these assessments can be easily introduced into the students' learning process in a training mode for self-assessment [7,8]. The training leads to increased self-assurance and confidence of the learners. In a scheduled assessment students have much more confidence regarding the subject and process of assessment [7,8].

A. *Benefits for the Student*

In order to provide some numerical data on the state of the art on the benefits of mock exams for the students, we have to leave the field of Computer Science, because we are not aware of any such study. However, there are research results available in other areas. Jacobs et al. [9], for instance, investigated, whether a mock exam can help to improve the results of the qualifying exam of the seminar "Statistics 1". The authors compared 25 randomly selected students who took a mock exam (experimental group) to 29 students who got an exam preparation lecture as an alternative (control group). While 66.0% (SD 11.7) of the experimental group passed the exam (pass: 50% of learning goals), only 60.1% (SD 15.2) of the control group passed it. This difference is not significant. However, if the failure rate is compared, 2 failed the exam in the experimental group, while 9 failed in the control group. This result is significant in both the chi-square test (5% significance level) as well as on the Fisher exact probability test ($p = 0.04$). Based on these results, the authors claim that in particular weak learners benefit from a mock exam.

Görlich analyzed the participation in a voluntary electronic mock examination [4, p. 190] for the course "Educational Psychology". Of the 685 participants of the qualifying exam, 420 (61%) took part in the mock exam. In a questionnaire, students who participated in the mock exam were asked, how useful they found it [4, p. 191]. Between 92% and 95% of the students found the mock exam useful for preparation on the qualifying exam; only 2% found it completely useless.

Peat and Franklin [10] report on the use of a mock exam in a "First Year Biology" course. While the examination was paper-based during class time, students needed to enter their answers in their own time into an interactive web form and having a program mark their performance, in order to get feedback. While almost all students participated in the mock exam, only 37% used the web-based marking scheme. During the pilot phase, those 46 students with a score less than 45% in the mock exam (these were perceived to be "at risk" of failure) chose to attend face-to-face remedial tutorials. Three-quarters of these students passed the final examination.

Dotson et al. [5] investigated what they call "mock exam study sessions" in a "Principles and Procedures of Behavior Modification and Therapy" course. Students could participate in a 45 minutes mock exam under test-like conditions led by a teaching assistant. Then the teaching assistant handed out a grading key and discussed grading criteria question by question. Students had to review and evaluate their answers themselves and were encouraged to disclose their answers for discussion and analysis. At the end of this discussion, the teaching assistant briefly explained a correct sample answer. The 60 to 90 minute lasting discussion portion of the mock exam study session helped students to gain a deeper understanding of grading criteria as well as issues to be addressed when procedures had to be applied across different situations. On a scale of 1 (= waste of time) to 5 (= vital) students rated the mock exam study sessions as being helpful (mean: 4.70). Students who attended the mock exam study sessions scored significantly higher (two-tailed t-test, $t(2) = 8.558$, $p = 0.013$) on the qualifying exam than the students who did not attend.

B. *Structure and Realization of Electronic Mock Exams*

Good sources for mock exam questions are previous actual examinations [2]. This ensures that questions are similar to actual examination questions. The mock exam as a whole should be a good representative of actual exams and should give a good coverage of the course overall [2]. Deutsch et al. [11] provide some detail, how they realized an electronic mock examination from the technical perspective. The authors propose a mock exam with a structure identical to the final exam. In their setup, questions were arranged on several screens with an average of three questions per screen. Navigation between screens was feasible at any time. Each screen could be edited as often as needed and incompletely edited screens were highlighted. In analogy to the final exam, the time to complete the mock exam was limited once it was activated by the student. The time left to complete the exam was displayed throughout. Once the time was over, the exam was automatically handed in for assessment.

III. Goals

The primary goal of the present study was to provide an experience report on the development of an electronic mock examination. Requirements for the mock exam were (a) that it is as similar to a real exam as possible, (b) that it is marked in a fully automated fashion, and (c) that it can be taken by students via the Internet at any time and as often as the students want. Of these requirements, item (b) and item (c) were hard requirements, i.e. they had to be fulfilled in either case. Requirement (a) had to be fulfilled as good as possible. Nevertheless, realizing requirement (a) is the most challenging: The mock exam has to provide the same structure as a real exam. Furthermore, the mock exam has to use the same weighting scheme between the questions, identical difficulty levels as well as the same distribution of both question types and competence levels covered in the questions.

Study regulations of our university allow multiple-choice questions only under special circumstances and there are legal restrictions when using them. For this reason, we do not use any multiple-choice question in our qualifying exam; students have to write down text (or numbers) in all questions. Thus, in

the realization of the mock examination we also had to use question types where students have to enter text as often as possible, as long as the fully automated marking does allow for this. Due to the fact that we did not use any multiple-choice questions, we also did not use any strategy for automated generation of questions such as the one by Schagaev et al. [6]. Instead, all questions had to be conceptualized and implemented manually.

The second goal of the study was the evaluation, what the students make out of the mock exam and how well they accept it. Here we wanted to analyze the students' participation as well as the correlation of the scores obtained in the mock compared to the qualifying examination. Furthermore, we wanted to analyze the strategies used by those students who take the mock exam more than once. Finally, we interviewed those students who came to view their exam papers on their assessment of the mock examination.

IV. Methodology

A. Instructional Setup

The introductory course "Basics of Computer Science" in the Department of Engineering and Management covers one semester with four hours of lecture per week. The course is mandatory for every student in the first semester in all three undergraduate programs of the department: Engineering and Management (B.Eng.), Logistics Engineering and Management (B.Eng.), and Automotive Engineering and Management (B.Eng.). The students in the Engineering and Management program can specialize in the third semester in the field "Information Technology" and will then get additional Computer Science courses. For all other students "Basics of Computer Science" remains the only "core" computer science course within their curriculum.

The course provides an introduction to the theoretical foundations of computer science as well as an introduction to programming. Currently Visual Basic for Applications (VBA) as available within Microsoft Excel is used as programming language. The course is taught in a computer lab with one personal computer per student. This setup provides the opportunity to alternate between theoretical input and practical hands-on-sessions throughout class.

Since winter term 2013 we started using Just-in-Time Teaching (JiTT) [12] as well as Peer Instruction in this course [13]. Another introductory Computer Science course taught at our university where also JiTT is used is described and evaluated in [14]. All course material as well as the JiTT material (reading material and questions) is provided by the e-learning platform Moodle [15].

B. Setup of the Official Examination

There is a written final exam at the end of the semester. It lasts 90 minutes and is the only contribution to the final grade of the course. Grades are on a scale from 1.0 (very good), 2.0 (good), 3.0 (satisfactory), 4.0 (sufficient), to 5.0 (failed), with intermediate .3 and .7 gradings between 1.0 and 4.0. A fraction of this final grade contributes to the grade of the Bachelor of Engineering degree. To simplify the time management for the students, we use a scheme of one point per minute (= 90 points for the entire exam). Hence, if an exam question is weighted 8 points, students should solve the question within 8 minutes.

Despite some efforts in our university to go towards electronic examinations in front of a computer, our final exam is still a classical exam with paper and pen. Even the programming questions have to be solved on paper only. Without the help of an integrated development environment with such helpful tools like automated syntax checking, programming becomes difficult. Although we practice programming on paper a lot in class, the "programming-on-paper" questions are considered as being hard by most students. Students are allowed to use a pocket calculator and a formulary as aids in the examination.

The exam is structured according to the learning goals. It usually starts with "Functionality of Computers and Networks" (Section 1), followed by "Numbering Systems and Coding" (Section 2), "Algorithms and Data Structures" (Section 3), "Programming in VBA" (Section 4), and ends with "Design of an Algorithm and its Implementation in VBA" (Section 5). The questions of the examinations cover all competence levels of the taxonomy of Anderson and Krathwohl [16], a revision of Bloom's taxonomy [17]. In every exam we use questions on the levels "remember", "understand", "apply", "analyze", up to the highest level, "create". Only the level "evaluate" does not occur as a stand-alone question in the exam. However, in questions of the level "create", competency on the level "evaluate" strongly helps students to provide a correct solution.

C. Setup of the Mock Examination

We used Moodle to provide all course material. Throughout term the "Quiz activity" of Moodle was used for the JiTT questions. For this reason, we wanted to realize the mock exam also with the same activity. This has the benefit that students are very familiar with the system and accordingly, there were no technical questions regarding the mock examination.

We set up a quiz with a time limit, analogously to the final exam. After 90 minutes the quiz is automatically submitted and marked. Moodle provides an individual timer for each student that is always visible on the top of the screen. The timer is activated once the student clicks on a button to start the quiz.

For the structure of the mock exam we followed the proposal of Deutsch et al. [11]. Questions were arranged in the same order as in the final exam and we also used the same sections. Each section corresponded to one screen. This resulted in 4 questions in Section 1, 2 in Section 2, 4 in Section 3, 3 in Section 4 and 1 in Section 5, respectively, totaling 14 questions. The average number of questions per screen was 2.8 in good agreement with the 3 questions per screen proposed in [11]. Each screen had a "Description" field on top with the name of the respective section. This allowed for easy navigation within the quiz environment.

The electronic mock examination was activated in Moodle after the last lecture of the course (week 14 of the semester). It was available for 13 days including the day of the qualifying examination.

D. Time Requirement for Mock Exam Setup

Coming up with an initial concept for the mock exam, taking our collection of exam questions for the last six examinations as a start, took about 3 hours. This included the individual assessment whether a given question can be used or has to be adapted for fully automated marking. This also included the elimination of questions that were too close to questions the students had seen before. Rephrasing questions, entering them into Moodle, and formatting them nicely took about 10 hours. A significant fraction of this time was needed for formatting the cloze texts and assigning writing variations for the correct answers.

Finding errors in the mock exam questions and fixing them, as will be outlined in Section V. E., took another 5 hours. The most time consuming part was the manual inspection in order to verify that all students' answers were marked correctly. Including the time for some discussions, the time for setup and verification of the entire mock exam was about 20 hours in total.

V. Example Questions of the Electronic Mock Examination

Due to space restrictions we cannot discuss all 14 questions of the mock exam in detail. Instead, we present representative examples of question types on a range of competence levels. In our mock examination we use questions in the German language. In this work, we provide English translations instead for clarity.

A. Questions of the Competence Levels "Remember" and "Understand"

Questions of the competence level "remember" can be formulated fairly easily for exams marked manually. For instance, the question "What is the definition of [technical term]?" can be used. However, in an automatically marked exam this is not that easy, due to the many variations for writing down such a definition. Nevertheless, this can be handled for automated marking, when the "direction" of the question is changed. Instead of asking for the definitions of technical terms, we provide a set of statements and ask for a term that fits to these statements. Fig. 1 exemplifies this for a question from Section 3 "Algorithms and Data Structures". In this question we provide statements that fit to certain properties of algorithms and the students have the name the respective properties. This is realized with the "embedded answer (cloze)" question type with short answers in Moodle. In the example, we arranged the form fields within an HTML table and thus this question does not have the "appeal" of a typical cloze question. Questions of the competence level "understand" can be realized in a similar fashion.

B. Questions of the Competence Level "Apply"

Typical question of the competence level "apply" include calculations, where students have to apply equations and have to calculate values in order to solve a given problem. Fig. 2 shows an example question from Section 1 "Functionality of Computers and Networks". It is achieved with the "numerical" question type of Moodle.

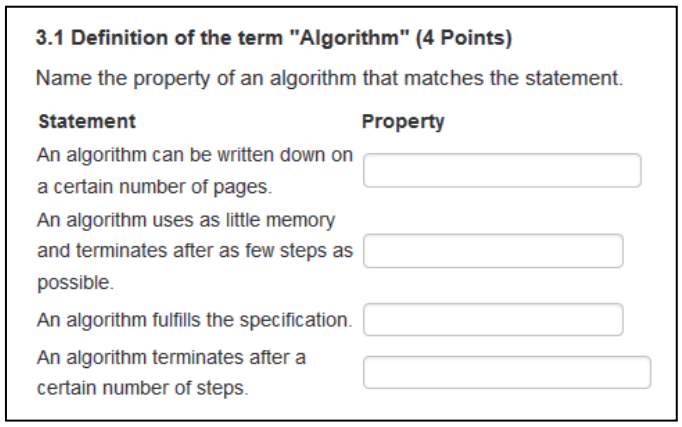


Fig. 1. Example question for the competence level "remember".

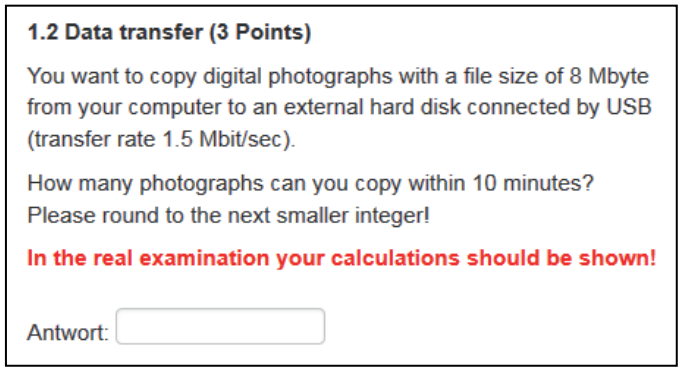


Fig. 2. Example question for the competence level "apply".

In the qualifying exam it is important that students show their calculations. The correct answer gives 1 point; the method of arriving at the answer another 2 points. In an automatically marked exam, it is difficult to let the students input the path of the calculation, unless we provide form fields that give too many hints on how to obtain the result. For this reason we inform the students in the mock exam that in the real examination they have to state how they arrived at the answer.

C. Questions of the Competence Level "Analyze"

Understanding what a given algorithm or computer program does is an important learning objective. Reading code is important for the overall learning progress. We developed many questions on the competence level "analyze" for our JiTT sessions. In the exam, typical questions on this level include the interpretation of Nassi-Shneiderman-diagrams as well as examples of entire VBA sub procedures or VBA functions.

In Fig. 3 a typical example question of Section 4 "Programming in VBA" is shown. Students have to figure out, how the given program for the calculation of the greatest common divisor works. The students have not seen this program before in class and have to figure out the "inner working" of the code. Toward this end, students have to specify the entire output of the program in the correct order.

4.3 Analysis of algorithms and programs (9 Points)

The following VBA program is given. Assume you enter the number **144** in the first `InputBox` and the number **84** in the second one. Indicate the entire output of the program, i.e. what is the output of all the code lines with a `MsgBox`?

```
Sub gcd()

    Dim a As Long, b As Long, g As Long


    a = InputBox("Numerator?")

    b = InputBox("Denominator?")


    ' Processing: Euclidean algorithm

    If (a = 0) Then

        g = b

    Else

        While (b <> 0)

            If (a > b) Then

                a = a - b

                MsgBox ("a = " & a)

            Else

                b = b - a

                MsgBox ("b = " & b)

            End If

        Wend

        g = a

    End If


    MsgBox ("Greatest common divisor = " & g)

End Sub
```

Output:

[ ] = [ ]
[ ] = [ ]
[ ] = [ ]
[ ] = [ ]
[ ] = [ ]
[ ] = [ ]

Greatest common divisor = [ ]

Fig. 3. Example question for the competence level "analyze".

This question is again realized with the "cloze" question type of Moodle. Instead of the "short answer" type, we use the "multiple-choice answer" type here. In this variant, students have to select the correct answer from dropdown menus. This has practical reasons: If we would allow for short answers, many students would fill in their answers in a fashion that the automated marking would not recognize as correct. Selecting answers from a dropdown menu, however, points students to the correct formatting of the answer and thus makes fully automated marking possible. In case of the question shown in Fig. 3, the first dropdown menu per line in the output section has the entries "a =" and "b =". The second dropdown menu per line provides the answers "0", "2", "8", "10", "12", "24", "30", "36", and "60". The same set of answers is contained in the dropdown menu for the greatest common divisor.

D. Questions of the Competence Level "Create"

The realization of automatically marked questions on the highest competence level "create" is the most difficult. Since the learning objective is the creation of something new, also in the exam the students have to create something new.

In our course we use two types of questions on this competence level. In the first type, students have to develop an algorithm for a given specification and express their result in a notation for algorithms. For this purpose, we use the Nassi-Shneiderman-diagram. In the other type of question, students have to go a step further and implement the algorithm in the VBA programming language. Such types of questions are placed in Section 5 "Design of an Algorithm and its Implementation in VBA" of our exam. In the mock examination this creation process has to be mapped into an automatically marked question. We achieved this with a "trick". As shown in Fig. 4, we provide the specification of a program with example input and output and let the students develop their implementation on paper. Once the students come up with a solution, they have to fit this solution into a given cloze text (Fig. 5). We consider this process of writing and mapping to be of the same difficulty as questions in the final exam on the same competence level. Of course, students may cheat here by taking a look at the cloze text before writing down their solution. However, in order to cheat, students have to scroll down to the bottom of the current screen. In addition, many of the gaps in this cloze test can only be filled when the students have thought about the problem in depth on at least the competence level "analyze".

E. Some Hints for Phrasing Questions

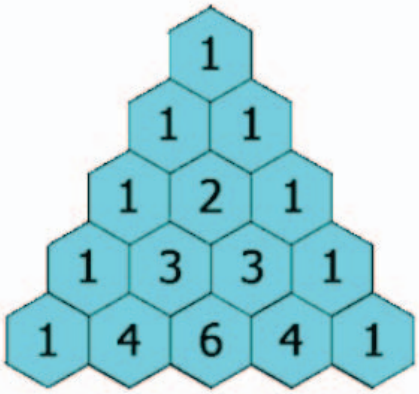
The automated marking worked very well. Despite some easy to fix typographical errors, we had only two "severe" cases where we had to intervene. Both cases happened in the same question where an Excel table had to be filled with entries using a VBA program. Fig. 6 shows this question after correction. The correct answers for the third and fourth gap are "n–i+1" and "n+i+2", respectively. We included some writing variations of these answers. While we anticipated "–i+n+1" and "i+n+2" for the two cases, we had to add the variants "1+n–i" and "1–i+n" for the first case as well as "2+i+n" and "2+n+i" in the second case, respectively (including a set of variants with spaces around operators).

5 Design of an Algorithm and its Implementation in VBA

The exam question would look like this (20 Points):

Background

Pascal's Triangle is a pattern of numbers that is constructed using a simple rule. You start with a row with a 1 and a second row with two 1's. Each subsequent row starts with a 1 and ends with a 1. The numbers in between are calculated by adding the two numbers above to the left and the right. In this way, the rows of the triangle go on infinitely.



Your Task

Implement a complete VBA function in correct syntax that fulfills the following specification:

- Implement a VBA function `NextPascalRow` . This function has a single parameter, an array `row` containing the numbers of the current row.
- From this given row, the function has to calculate all the numbers of the next row of Pascal's Triangle.
- The function should be able to process any row (beginning with row 1). The result should be returned as an array of numbers.
- All arrays should start with index 0.

Example Input and Output

Call with the actual parameter: array with the numbers [1; 2; 1]

Return value: array with the numbers [1; 3; 3; 1]

In the exam you have to implement the entire function. Here you should do the same: please implement the function with paper and pen!

Fig. 4. Example question for the competence level "create" (part 1). The figure used for Pascal's Triangle in the question was created by the user Hersfold on the English Wikipedia and is released into the public domain (https://commons.wikimedia.org/wiki/File:PascalTriangleAnimated2.gif).

Are you ready with your answer on paper? Really? Then compare your solution with the provided cloze test. Try to fit your solution into all the gaps:

```
Option Explicit

Function NextPascalRow(row() [          ]) _
[          ]

    ' Declaration of variables

    Dim n As Integer

    Dim nextrow() [          ]

    Dim i As Integer


    ' Counter of numbers in current row

    n = [          ]


    ' Dimension of the output array

    ReDim [          ]


    ' Calculate numbers of next row

    nextrow(0) = [     ]

    nextrow([     ]) = [     ]


    If (n >= 2) Then

        For i = 1 To ([          ])

            nextrow(i) = row([          ]) + row(i)

        Next i

    End If


    ' Return result

    [          ] = nextrow()

End Function
```

Fig. 5. Example question for the competence level "create" (part 2).

A second problem in this question occurred due to our assumption that it is the easiest to fill the table from top to bottom. However, a student came up with a correct solution in the opposite order. He filled the table from bottom to top, which resulted in a completely different but nonetheless correct answer. Simply marking theses answers as correct would result in another problem: some gaps could be filled with the correct answer when working from top to bottom, while others are correct when working in the opposite direction. To exclude this mixture of answers, we had to provide the order how the rows of the table have to be filled in the text of the question.

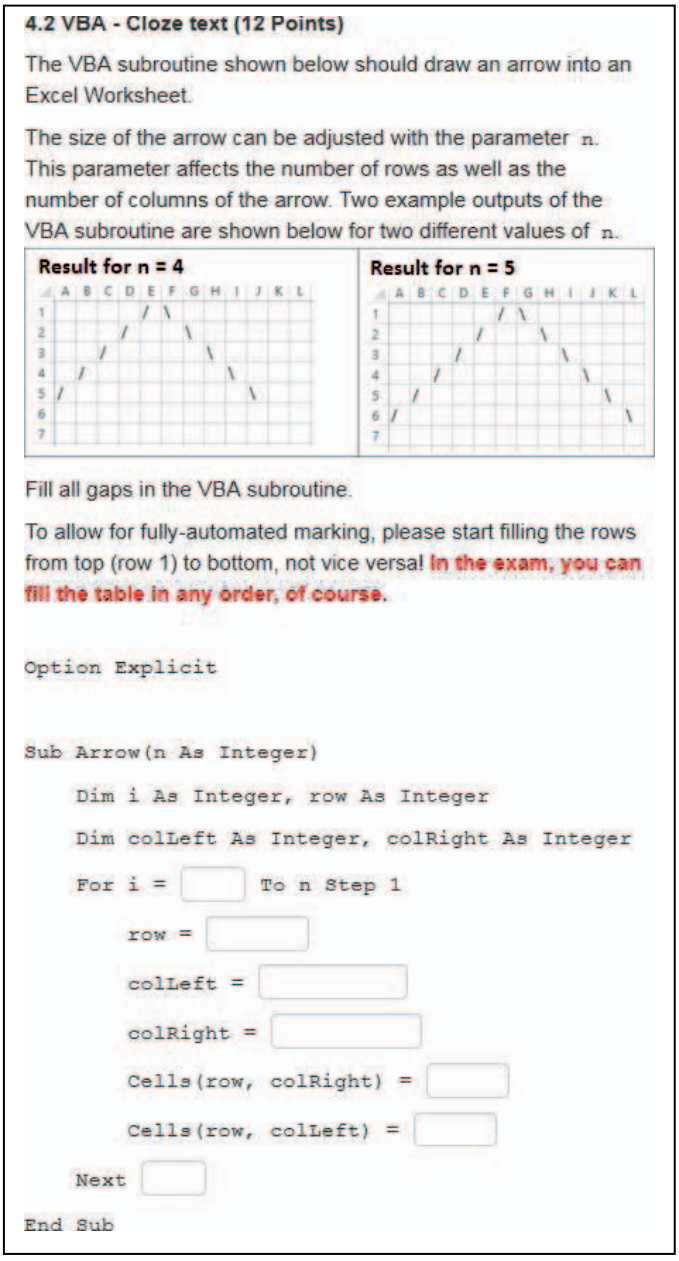


Fig. 6. The only question (cloze text) that had to be changed during the pilot study to allow for fully automated marking.

VI. Evaluation

A. Participation in Mock Examination

In order to analyze the participation rate of the mock examination, we could compare this number with the number of students who carried out the electronic exam application. In our university, however, the exam application has to happen early in the semester around mid-term. Of the 60 students who applied for the exam, 22 students (37%) either postpone their exam for one term or dropout completely (Fig. 7). This number can be compared with data of another first-year Computer Science class at our university: Thurner et al. [18] observed a significant dropout of students between exam application (81 students) and participation in the exam (55 students). This dropout rate of 32% is in good agreement with our findings.

A better way to calculate the participation rate is the method used by Görlich [4]. He directly compared the participations in the mock and the qualifying examinations. In our case, the participation rate in the mock examination is 20 of 24 students (83%) for 1st term students, 7 of 13 students (54%) for 2nd term students, and 0 of 1 student (0%) for 3rd term students, respectively. The overall participation rate of 27 of 38 students (71%) is slightly higher than the rate of 61% observed by Görlich [4, p. 190]. These numbers in voluntary mock examinations, of course, are lower than the participation rate for a paper-based mock examination during a scheduled class, where virtually every student participates [11].

Two students who participated in the mock exam did not take the final examination (red arrows in Fig. 7). These two students obtained scores of 0.0 and 3.5 points, respectively. It is likely that these students based their decision on not taking the final examination also on their disastrous results in the mock exam. Nine students that had not registered in the study groups where the mock examination was offered, to our surprise, took the mock exam. These students must have heard of the mock exam, asked students of our study groups for the Moodle password, registered, and took the exam. This effort can be taken as an indicator, how valuable a mock examination is regarded among students. Thus, in total 38 students participated in the electronic mock examination.

While 19 students (one half) took the mock exam only once, the other half of the students attempted the mock exam (as a whole or in part) two or more times (Fig. 8), sometimes without answering questions. One student even tried the mock exam 5 times. This summed up to a total number of 73 attempts.

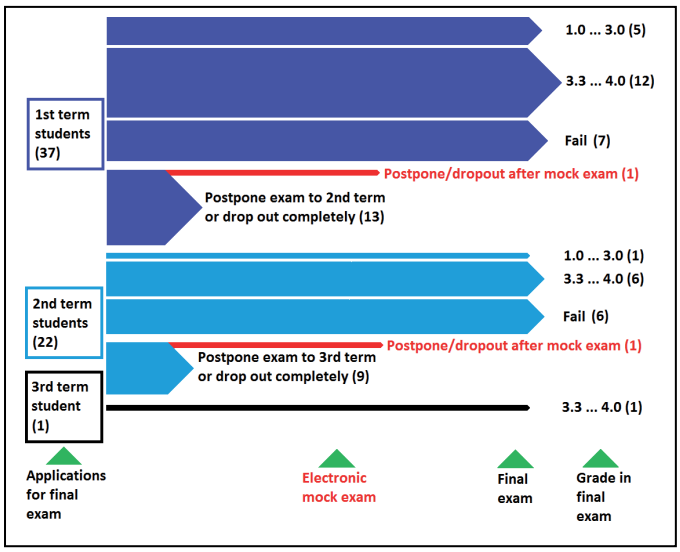


Fig. 7. Applications for final exam and participation in final exam indicating the rate of students who pospone their exam or dropout completely. Grades are clustered in the same fashion as in [18].

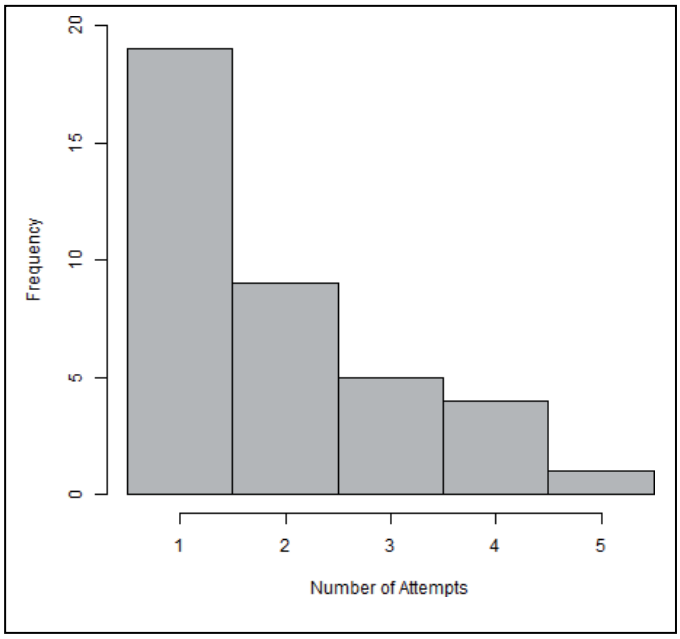


Fig. 8. Histogram of numbers of attempts per student in the electronic mock examination.

B. Correlation between Scores in Mock and Qualifying Exams

A scatter plot showing the correlation between the score in the mock examination versus the score in the qualifying examination for the 27 students who took part in both types of examinations is shown in Fig. 9. The linear regression (also depicted in Fig. 9) indicates that both scores are weakly correlated ($R^2 = 0.38$). The intercept of about 30 indicates that students who scored really badly in the mock exam used their time to learn and were able to improve their results in the final exam by about 30 points. This result agrees with findings of Jacobs et al. [9] that weak learners take the most advantage out of a mock examination.

C. Effect of Mock Exam on Results in the Qualifying Exam

In order to assess the effect of the mock exam on the results of the qualifying exam, we compared the final exam scores obtained for the two groups (Fig. 10). The average score of the group who took the mock examination (experimental group) of 49.4 points (SD 11.9, N = 27) is higher than that of the control group without mock examination of 42.2 points (SD 13.6, N = 11). However, this result is not significant on neither the one-tailed T-test (T = 1.579, p = 0.06) nor on the Mann-Whitney U test (p = 0.09). If failure rates are compared in accordance to Jacobs et al. [9], 8 of 27 (29.6%) students fail in the experimental group while 5 of 11 (45.5%) fail in the control group. This difference is also not significant (Fisher exact probability test, p = 0.46).

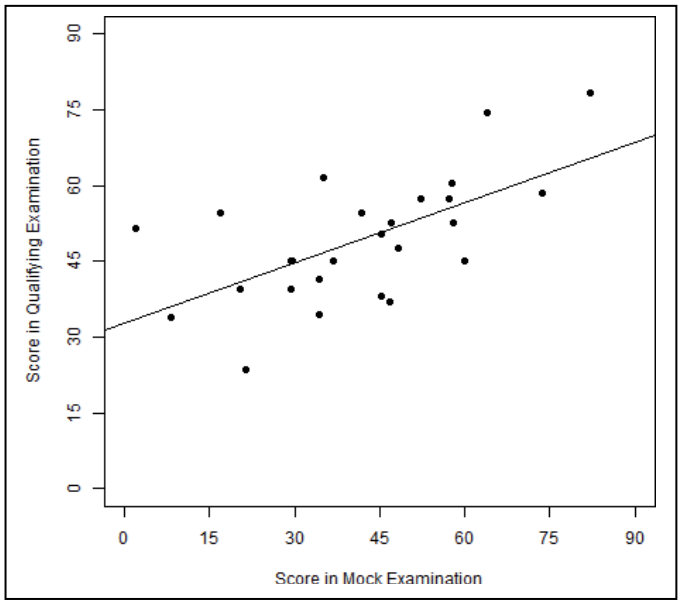


Fig. 9. Score in mock examination versus score in qualifying examination. See text for details on the linear regression.

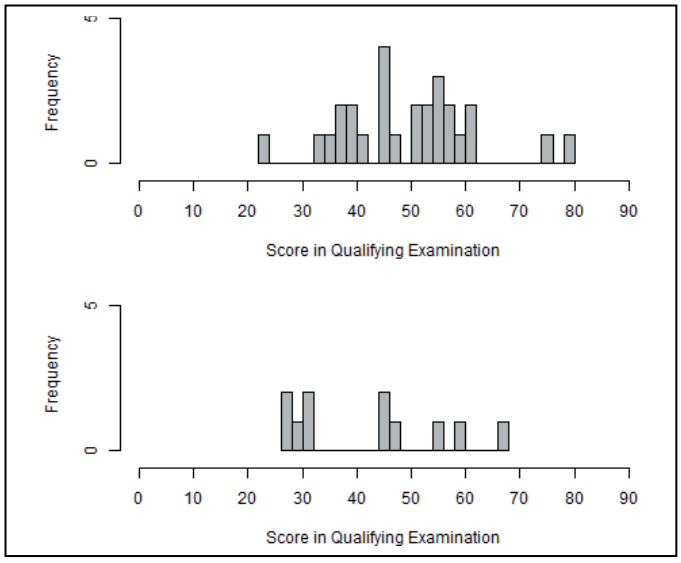


Fig. 10. Histograms of scores in qualifying examination for students who took the mock exam (top) and students who did not take the mock exam (bottom).

D. Feedback by Students

It is difficult to reach students after the last lecture. Nevertheless, we took the chance to briefly interview two students who came to our office hours between the last lecture and the qualifying exam. Of course, in this phase of the immediate preparation for exams, student's time is very limited. Nevertheless, both students praised the mock exam as a valuable aid for their self-assessment. One of the two students also lauded the time pressure as being helpful. Both students agreed that the electronic mock examination should be made an integral part of the course in the semesters to come.

A further five students who came to view their exam papers were interviewed two weeks after the final exam. Of these

students, 3 failed the exam and 2 passed with grades 1.7 and 3.7, respectively. In order to avoid keeping other students waiting, the interviews had to be short. All five students took part in the mock examination, however, in very different fashions. While three students finished the entire mock exam, one student only answered the programming questions, and the last answered programming questions as well as those questions where values had to be calculated, i.e. questions on the competence level "apply". All five students agreed that the mock exam was helpful for their exam preparation. They also agreed that next time the mock exam should be made available earlier in the term.

We are well aware that the small number of students we could interview is not representative and more or less anecdotal. Nevertheless, these interviews gave valuable insight into the strategies that students employ in the final phase of exam preparation. Of particular interest in this respect is the statement of one student: She explained in detail how she made use of the mock exam. Knowing that the mock exam can be taken as often as needed this student answered all questions in the first attempt. With the feedback obtained, she subsequently recapitulated the topics of the wrongly answered questions. In her second attempt, she then answered only these questions again. She repeated this procedure for two additional iterations. Despite this effort, unfortunately, this student failed the exam. However, this led to the analysis of student's strategies outlined in the next section.

E. Student's Strategies for Repeated Participation in the Mock Examination

Students who take the mock exam more than once employ a number of strategies. In order to dissect these strategies, we analyzed the score obtained in the different attempts as well as the time needed for completion of each attempt. We distinguish two cases of attempt. In the first case, a student takes the full exam (F). In the second case, a student either does not solve any question or only a small fraction of them. Hence, in this second case the student either inspects the questions or solves a small increment of the exam (I). In the following, strategies in each attempt are denoted with the letters F and I, respectively.

Students who took the mock exam twice used two different strategies:

- Strategy FF: Seven students took the entire mock exam twice. All of them improved their score from attempt 1 to attempt 2.
- Strategy IF: Two students inspected the questions in a first attempt without answering. In their second trial they then took the full mock exam.

Students who took the mock exam three times used a variety of strategies.

- Two students took the entire exam two times and improved their score in every attempt. In their respective third attempts, one of them took the entire exam a third time and again improved the score (strategy FFF), while the third student just inspected the exam in his third trial (strategy FFI).
- Three students first inspected the questions in trial 1 and took the complete exam in trial 2. In the third attempt, two students took the entire exam (strategy IFF) while the third inspected the questions (strategy IFI).

The five students who took the mock exam four or five times used five different strategies with varying combinations of taking the full exam or inspection/incremental improvement. These were FFFF, FIFI, FIII, IIIF, and FIIIF.

A comparison of mock exam scores obtained by the students who took the exam multiple times returns an interesting result. Students who only take the mock exam in full at least two times (strategies FF, FFF, and FFFF) obtain higher scores (58.1 points, SD = 15.1) in their best (= last) full attempt than students who use mixed strategies with at least one inspection/incremental attempt in between (38.6 points, SD = 11.2). This result is significant on the Mann-Whitney U test ($p = 0.004$). Unfortunately, the corresponding difference between these two groups (mean 58.1 points, SD = 13.5 versus mean 44.1 points, SD = 12.2) is not significant on the final exam scores ($p = 0.15$). However, all students who applied any of the strategies FF, FFF, or FFFF and also showed up in the final exam passed it.

VII. Conclusions and Lessons Learned

Due to the small number of students in our study, all our results have to be treated with some care. Nevertheless, our observations clearly demonstrate the value of mock examinations and, in particular, the fully automated electronic variant. There are numerous advantages for the students.

- Students have the opportunity to test their learning progress under exam conditions. From the score obtained and the feedback they get, they can decide, which part of the teaching content they have to recapitulate or practice. Students agree that this is helpful for their exam preparation.
- In the fully automated setup, students get the feedback immediately after completing the electronic mock examination. This is a big advantage over a mock exam on paper that has to be handed in, marked manually, and returned.
- From the feedback of the students and the participation rate of 71% we can conclude that students see a benefit that is at least worth spending 90 minutes of their time.
- A big advantage of the electronic mock examination is that it easily can be taken more than once. Students can decide how often they take parts of the exam and which parts they focus on. The opportunity to take the mock exam more than once was used by half of the students. Our analysis shows that students use a number of strategies to make the best out of this opportunity.
- Of those students, who take the mock exam more than once, taking it in full at every attempt gives significantly better mock exam results than any strategy that involves at least one inspection/incremental attempt. Also, all students who applied any of the strategies FF, FFF, or FFFF and showed up for the final

examination also passed it. Based on these findings, we recommend that students should try the mock exam only in full several times without any incremental/inspection attempt in between.

- Finally, taking the mock examination is voluntarily. This goes hand in hand with the way, how students make use of the mock examination. They can decide whether they take it under exam conditions or – in case they still have to learn certain parts – use all course material in the mock exam.

Changing to the lecturers perspective, there are also many advantages. The most important one is to get an overview of the students' learning progress. There is a wide range of feedback instruments available that a university teacher can make use of to assess teaching effectiveness [19]. Among them, an electronic mock examination provides the same features as a real exam. The most important feature is that the mock exam is as similar to a real exam as possible. For this reason the entire learning objectives on their respective competence levels are covered [19]. Thus, the automatically marked mock exam provides an ideal feedback instrument for lecturers. When the mock examination takes place a few weeks before end of term, university teachers can analyze results of the mock exam. This feedback comes exactly at the right time before the last lecture. Lecturers get the opportunity to address problematic questions and recapitulate the respective course material in detail in the last lecture.

Once the electronic mock examination is set up, there is almost no work involved for lecturers to use it. Only when learning objectives change, both mock as well as final examinations have to be changed accordingly. This is in great contrast to a mock examination on paper, where material has to be printed, tutors for marking are necessary, and the exams have to be handed back to the students. The fact that the electronic mock exam is taken in the students' own time is also a plus: the always "precious" presence time is not needed for this variant of mock examination.

Currently, we are in the process of setting up the electronic mock examination for all eight study groups of all the six lecturers of the course "Basics of Computer Science". This time, we will activate the mock examination earlier in the semester. We can then make optimal use of the results of the mock examination for preparation of the recapitulation lecture. Doing so, we hope to further optimize the immediate exam preparation phase of our lecture.

ACKNOWLEDGMENT

The authors thank Anne C. Spindler and Gisela Prey for valuable feedback on an early version of this manuscript and Barbara Hank for fruitful discussions on statistics and on visualization of data. The authors are grateful to the referees for many helpful comments.

REFERENCES

[1] A. S. Hornby, Oxford Advanced Learner's Dictionary of Current English, 8th ed. Oxford: Oxford University Press, 2010, p. 985.

[2] P. Thomas, B. Price, M. Petre, L. Carswell, and M. Richards, "Experiments with electronic examinations over the internet," in Proceedings for the 5th Computer Assisted Assessment (CAA) Conference, 2001. Available online: http://caaconference.co.uk/pastConferences/2001/proceedings/q1.pdf.

[3] L. Parmer, "Helping students prepare for qualifying exams; a summary of WCRA Institute III," in Proceedings of the 8th Annual Meeting of the Western College Reading Association, Anaheim, CA, 1975.

[4] S. Görlich, Fundierung und Integration von E-Learning-Komponenten in die Präsenzlehre (in German), Ph.D. Thesis, University of Gießen, Faculty 06 – Psychology and Sports Science, Gießen, Germany, 2006.

[5] W.H. Dotson, J.B. Sheldon, and J.A. Sherman, "Supporting student learning: improving performance on short-essay exams using realistic practice opportunities," Journal of the Scholarship of Teaching and Learning, vol. 10(3), pp. 106-118, November 2010.

[6] I. Schagaev, N. Folic, N. Ioannides, and E. Bacon, "Multiple Choice Answers Approach: Assessment with penalty function for Computer Science and similar disciplines," International Journal of Engineering Education, vol. 28(6), pp. 1294-1300, 2012.

[7] E. Bacon, B.R. Kirk, G. Hagel, G. Kravtsov, M. Charnine, I. Schagaev, and R. Foggie, "Web-enhanced design of university curricula," in Proceedings of the 2013 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS), pp. 288-294.

[8] I. Schagaev, B. Kirk, and L. Bacon, "Essential knowledge aggregation, delivery, and assessment," ACM eLearn Magazine, May 2014.

[9] B. Jacobs, H. Bernd, and A. Fey, "Die Wirkung einer Probeklausur auf Klausurleistung und Angst in einer Statistikklausur" (in German). Available online at URN: urn:nbn:de:bsz:291-psydok-2720, July 2004.

[10] M. Peat and S. Franklin, "Supporting student learning: the use of computer-based formative assessment modules," British Journal of Educational Technology, vol. 33(5), pp. 515-523, November 2002.

[11] T. Deutsch, K. Herrmann, T. Frese, and H. Sandholzer, "Implementing computer-based assessment – a web-based mock examination changes attitudes," Computers & Education, vol. 58(4), pp. 1068-1075, May 2012.

[12] G.M. Novak, E.T. Patterson, A.D. Gavrin, and W. Gavrin, Just-in-Time Teaching: Blending active learning with web technology. Upper Saddle River, NJ: Prentice Hall, 1999.

[13] E. Mazur, Peer Instruction: A user's manual. Upper Saddle River, NJ: Prentice Hall, 1997.

[14] A. Böttcher, A. Kämper, and V. Thurner, "On analyzing the effectiveness of Just-in-Time Teaching," in: Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON), pp. 453-461.

[15] M. Dougiamas and P. Taylor, "Moodle: Using learning communities to create an open source course management system," in: Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003, D. Lassner and C. McNaught, Eds., 2003, pp. 171–178.

[16] L.W. Anderson and D.R. Krathwohl, Eds., A Taxonomy for learning, teaching, and assessing. A revision of Bloom's taxonomy of educational objectives, abridged ed. New York, NY: Longman, 2001.

[17] B.S. Bloom, M.D. Engelhart, E.J. Furst, W.H. Hill, and D.R. Krathwohl, Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York, NY: David McKay Company, 1956.

[18] V. Thurner, A. Böttcher, and A. Kämper, "Identifying base competencies as prerequisites for software engineering education," in Proceedings of the 2014 IEEE Global Engineering Education Conference (EDUCON), pp. 1069-1076.

[19] A. Böttcher, A. Kämper, and V. Thurner, "On feedback techniques for the evaluation of teaching effectiveness," in Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON), pp. 668-675.