An Approach to Assessing Learning Outcomes over Consecutive Courses

Irene Rothe[1], Marco Winzker[1] and Cathleen Salega[2]
[1]Dept. of Electrical Engineering, Mechanical Engineering and Technical Journalism
[2]Dept. of Strategic Planning and Management Control Services
Bonn-Rhein-Sieg University of Applied Sciences
Sankt Augustin, Germany
irene.rothe@h-brs.de
marco.winzker@h-brs.de
cathleen.salega@h-brs.de

***Abstract*— This Research to Practice Work in Progress Paper describes an outcome-based evaluation performed over three consecutive courses taught to electrical engineering students. The courses are delivered in the study entry phase: Computer Science 1 and 2 in the first and second semester and Digital Design in the third semester. The outcome-based evaluation was introduced to evaluate the progress made by students during the courses. Covering three semesters taught by two different lecturers, it gives a long-term view of the learning process in the study entry phase. The "outcome-based evaluation" methodology is proposed in literature and was introduced at our university at the end of 2016. Particular attention needs to be paid to formulating the questionnaire. The lecturers therefore cooperate with an evaluation officer from the university.**

Keywords—evaluation, learning outcomes, self-assessment, cross-evaluation

I. INTRODUCTION

Evaluation is important for improving the quality of the lectures given during courses and for obtaining feedback that can be used by the lecturer as input for future changes and improvements to the course material. Teaching quality can be measured in various ways applying many different criteria. Some criteria such as course structure, teaching materials, practical relevance, and connections between lectures and tutorials are analysed using a standard questionnaire for all courses in the Department of Electrical Engineering, Mechanical Engineering and Technical Journalism.

Another measurement is the evaluation of student learning outcomes. The examinations at the end of the course can be good indicators of these. With the help of the examination results, the lecturer can obtain information about which topics were well placed in the heads of the students or at which points the students felt unsafe and produced poor examination results. The lecturer can then take this information as the basis for improvements to future lectures.

However, there is also a downside to this kind of evaluation. Examinations are usually taken in critical situations, with time restrictions, on a specific day and under pressure. Additionally, not all examinations represent exactly the learning outcomes of the curriculum. There is also no way to find out about the prior knowledge the students had when they entered the course. Examinations mostly reflect a snapshot of the student's competencies. The implication is therefore that examinations only give a limited reproduction of the quality of the learning outcomes achieved.

All this implies that apart from analysing examination results and conducting surveys with the standard questionnaire, there is a need for other methods or tools that can be used for evaluation purposes with the aim of obtaining more specific, accurate information about the student's learning outcomes [1][2].

The method used in this paper is described in the literature as an evaluation of student self-assessments pre and post lecture [3][4]. For research into the effects of prior knowledge in consecutive courses see [5].

Students perform self-assessments of their prior knowledge, skills and professionalism and post knowledge via evaluation sheets. An overview of the validity and accuracy of self-assessments can be found in [6] and [7]. The basis of the assessments is a questionnaire relating to the learning objectives defined in the curriculum. The wording of the learning outcomes in the questionnaire has to be as concrete as possible. It is important to make the scope for interpreting the wording as narrow as possible. In this way, the evaluation results can be utilised more extensively and effectively. The goal for the evaluating teachers should be to facilitate identification of the strengths and weaknesses of their particular course.

This paper focuses on how learning outcomes can be measured for different courses which interact consecutively over three semesters and what conclusions can be drawn from the evaluation results. This kind of empirical data has a pivotal role in initiating change processes [8].

This article is organized as follows: the next section describes the initial situation from where the conception of the outcome-based evaluation started. This section is followed by a

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

description of the actual implementation of the evaluation. The third section explains the implementation and presents the results of the learning outcome based evaluation. The conclusion gives a general summary of the usefulness of outcome-based evaluations and future ideas for implementing these.

II. INITIAL SITUATION

The new method of learning outcome based evaluation was first applied to three consecutive courses taught to electrical engineering students. The courses in question were: Computer Science 1 and 2 in the first and second semester and Digital Design in the third semester.

The learning outcomes described in the Computer Science curriculum for electrical engineering students are:

- basics of programming (how to get from a problem to an algorithm for implementation in programming code)
- knowledge of the general hard and software structure of a computer
- algorithms (sorting and searching algorithms, cryptology)
- programming language C (elementary data types, control structures, functions, addresses and pointers, arrays and strings)
- programming language C (memory allocation, structured date types, lists, bit manipulation)
- efficiency of algorithms

Digital Design is taught to the same electrical engineering students in an undergraduate course in the third semester. The learning outcomes described in the curriculum are:

- understanding digital circuits and the ability to design them
- knowledge of VHDL programming language and the ability to employ it for digital design
- knowledge of FPGAs and the ability to use them to implement digital circuits

These three courses are taught by two different lecturers.

III. DESCRIPTION OF THE OUTCOME-BASED EVALUATION

A. *Formulating learning outcome statements*

The outcome-based evaluation was newly introduced at the university in fall 2016. Two lecturers decided to use this kind of evaluation from the summer term of 2017.

The learning objectives of the course were used as the basis for formulating the learning outcome statements in the questionnaire. The lecturers received support from an evaluation officer while formulating the learning outcome statements.

The wording had to be in the singular, i.e. the statements had to start with “I am/know…”.

The learning outcomes were defined according to five rules: specific, measurable, attainable, relevant and time-bound [2]. They had to be formulated concisely and in a manner that was easy to understand.

A positive learning outcome is measured in percentage of the difference of the calculated mean values of the pre and post amount value of the student’s self-assessed learning outcome [9]:

$$learningoutcome\ [\%] = \frac{\mu_{pre} - \mu_{post}}{\mu_{pre} - 1} \times 100$$

B. *Learning outcome statements for Computer Science 2*

This contribution looks at the learning objective statements for Computer Science 2. The learning objectives were in German and the English translation is provided here:

1. I know the difference between call by value and call by reference.
2. I know the three most important data types in C.
3. I am able to implement a function in C.
4. I am able to work with the while-loop.
5. I understand what a pointer is.
6. I am able to work with a struct.
7. I am able to allocate memory dynamically in C.
8. I am able to explain at least three algorithms for sorting.
9. I am able to draw a flow chart for a given algorithm.
10. I am able to program a branch by using the if-structure.
11. I am able to define the term algorithm.
12. I am able to work with arrays in C.
13. I know what int **p means.
14. I know the difference between *(p++) and (*p)++ if p is a pointer.
15. I can describe the biggest open problem in Computer Science.
16. I know the difference between lists and arrays.
17. I am able to work with bit operators.
18. I am able to convert a decimal number into a dual number.
19. I am able to calculate the complement of a dual number.
20. I am able to add dual numbers and their complements.

For all statements, students reported their self-assessment pre and post lecture in the questionnaire. This assessment was given at the time proposed in [2][4]. All statements were answered on a scale of 1 (strongly agree) to 6 (strongly disagree).

C. *Overlapping learning outcome statements*

The analysis of three courses provided the opportunity to compare student feedback on similar topics. One overlapping question was therefore included in the evaluation sheets for cross-evaluation of the results. Statement 20 (dual numbers) was chosen for this purpose.

IV. OUTCOME-BASED EVALUATION

This section presents the evaluation results of the student self-assessments pre and post lecture. They are then analysed and discussed in terms of teaching effectiveness and to postulate future course improvements.

A. *Analysis of Computer Science 2 Course*

The evaluation results mainly discussed relate to the Computer Science 2 course described in the previous chapter. It took place at the end of the 2017 summer term and 28 students filled out the forms.

Figure 1 shows the results in graphical form for the 20 learning outcomes. Each question has two values, the upper is before the lecture (pre), and the lower is after the lecture (post). In the percentage column, the colors indicate the scale from ‘strongly agree’ in green to ‘strongly disagree’ in red.

The analysis of the Computer Science 2 course shows that some learning outcomes were rated very poorly, for example the statements about sorting algorithms and calculations with respect to dual numbers (statements 8, 20). This could be because those topics were only taught in the first half of the first semester and never really used again at any time during the course. The statement about calculating the complement of a dual number (19) was rated especially poorly and this could be because this calculation was practiced only once during both courses.

One interpretation might be that the students remembered learning something about topics such sorting algorithms and dual numbers, but when performing the self-assessment, they did not feel that they were able to solve a task in this area at the moment the evaluation took place.

Other learning objectives scored much higher for learning success, e.g. the about pointers, call-by-value and call-by-reference issues (1, 5, 13) because they were practiced very intensively during the course.

Last but not least, statements about the while-loop, if-structure, data types in C, arrays and use of functions scored very highly in the evaluation (2, 3, 4). This is learning content which serves as the basis on which code is built. It is the most important competence (in the curriculum), which is encouraged and supported throughout both semesters in Computer Science.

To summarise, it can be said that the learning outcome rating was closely related to the average number of exercises done in connection with the specific course content. Many of these exercises related to pointers, and so the students rated their own competences highly. Only one exercise relating to the complement calculation of dual numbers was done, so the competence rating was low.

Some values are interesting in the sense that it is clear where students can build new knowledge and skills on their prior knowledge and skills from high school days, for example all learning outcomes relating to elementary data types and programming structures (while and if) (2, 4, 10). It is also interesting is that some students had prior knowledge of sorting algorithms. This seems like a good example of why high schools should be motivated to explain algorithms. Another interesting point is the good knowledge of flow charts shown by some students, which meant that their high school teachers attributed just as much importance to them as university teachers.

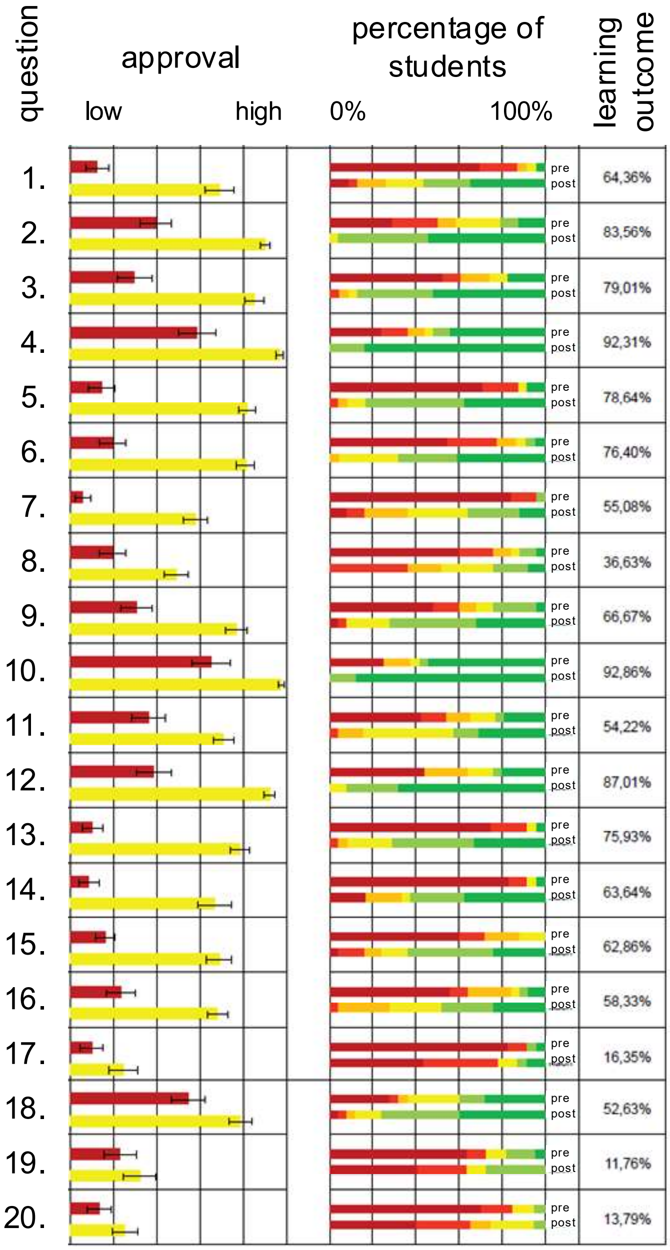

Fig. 1. Results of 20 learning outcomes for the course Computer Science 2 in the summer term 2017

The very low result for the learning outcome relating to bit manipulation skills (17) was due to the fact that the relevant lecture was given *after* the outcome-based evaluation was performed.

B. *Analysis of Computer Science 1 Course*

The evaluation of the Computer Science 1 course took place in the winter term 2017/18 with 72 students participating. The learning outcome values varied between 30% and 84%.

It was observed that the learning success for programming skills was much higher than for theoretical content. This is probably due to the motivation of the students, who seem to be more interested in practical knowledge than basic theoretical knowledge.

C. *Analysis of the Digital Design Course*

The evaluation of the Digital Design course took place in the winter term 2017/18 with 29 students participating. Learning outcome values varied between 25% and 72%.

It was observed that the learning outcome scores for statements with a higher level of difficulty were lower, e.g. “With the help of data sheets, I can decide which memory component (SRAM, DRAM, Flash) to use for a digital system” which achieved 28%. Statements with lower levels of difficulty achieved better learning outcome scores, e.g. “I can design a combinational circuit with a Karnaugh map”, which scored 66%.

D. *Overlapping Question for Computer Science 2 and Digital Design Course*

In order to analyse whether students see a relation between different courses and carry over their acquired knowledge, one statement was duplicated for the courses taught by different lecturers. The statement “I am able to add dual numbers and their complements” was given for Computer Science 2, taught in the second semester and Digital Design, taught in the third semester. The results can be seen in Figure 2.

Fig. 2. Comparison of the same learning outcome in Computer Science 2 and Digital Design

In theory, students should have the same proficiency after Computer Science 2 (post, lower value) and before Digital Design (pre, upper value). They might also forget some skills and show less proficiency before Digital Design.

However, the results indicate the opposite. At the beginning of the third semester, students rated their dual number calculation skills rather poorly, but for the beginning of their third semester, they rate these skills as highly as their prior knowledge. Two explanations are possible and have been discussed by the lecturers.

- The students sit a computer science examination between the second and the third semester and improve their skills in preparation.
- When students rate their “before” skills in the third semester, they appraise their skills differently.

V. CONCLUSIONS

This article outlines a newly introduced learning outcome based evaluation of three consecutive courses taught to electrical engineering students. This method compares student self-assessments pre and post lecture and establishes the learning outcome.

Particular attention was paid to formulating the questionnaire. The basis of the statements in the evaluation sheets was the learning objectives listed in the curriculum of the courses.

The exemplary results of the evaluation give a strong hint that engineering students are well capable of performing self-assessments and that the values obtained from the evaluation sheet can by all means be used by the educators to improve the courses.

The outcome-based evaluation tool is easy to use and helps the teachers clarify their learning goals and constantly improve their lectures.

One improvement planned for the Computer Science 2 course in the upcoming summer semester 2018 will be to continue paying attention to flow charts (question 9) while developing algorithms. This will an important part of the programmer’s life later on but had obviously been forgotten by the students in the second semester.

ACKNOWLEDGEMENT

Part of this work was supported by the German Federal Ministry of Education and Research (BMBF), contract number 01PL11067 and 01PL16067.

REFERENCES

[1] K. A. Gibson, P. Boyle, D. A. Black, M. Cunningham, M. Grimm, and P. McNeil, “Enhancing Evaluation in an Undergraduate Medical Education Program,” Acad Med 2008; 83: 787-793 10.1097/ACM.0b013e31817eb8ab.

[2] T. Raupach and S. Schiekirka, “Handreichung zur Lernerfolgsevaluation Electric Paper Evaluationssysteme GmbH”.

[3] T. Raupach, J. Brown, S. Anders, G. Hasenfuss, and S. Harendza, “Summative assessments are more powerful drivers of student learning than resource intensive teaching formats,” BMC Medicine 2013; 11: 61.

[4] S. Schiekirka, D. Reinhardt, T. Beißbarth, S. Anders, T. Pukrop, and T. Raupach, “Estimating Learning Outcomes From Pre- and Posttest Student Self-Assessments: A Longitudinal Study,” Acad Med 2013; 88: 369-375.

[5] R. Hoz, D. Bowman, and E. Kozminsky, “The differential effects of prior knowledge on learning: A study of two consecutive courses in earth sciences,” Instructional Science 29: 187–211, 2001, Kluwer Academic Publishers. Printed in the Netherlands.

[6] J. T. Fitzgerald, C. B. White, and L. D. Gruppen, “A longitudinal study of self-assessment accuracy,” Med Educ 2003; 37: 645-649.

[7] M. J. Gordon, “A Review of the Validity and Accuracy of Self-assessments in Health Professions Training,” Acad Med 1991; 66: 762-769.

[8] M. Winzker, M. Grein, S. Himmel, M. Kaul, and C. Luppertz, “Using Evaluation Data to Initiate Change in the Study Entry Phase,” Zeitschrift für Hochschulentwicklung (ZFHE) 2014; 9/2: 118-126.