

Teaching New Technologies and Life-Long Learning Skills: A Sample Approach and Its Evaluation

<http://dx.doi.org/ijac.v4i4.1728>

Christina B. Class

German Jordanian University, Amman, Jordan

Abstract—This paper presents a didactical approach to combine relevant elements of discovery learning and the project method in courses that include semester work and written examinations. This approach comprises three promising aspects: it increases student motivation, strengthens their self-efficacy and eases the teachers' task to integrate new technologies into their courses. The concept has been evaluated to verify the assumption that it can be applied to teaching new technologies. The course evaluation equally shows that students' self-efficacy can be strengthened by the described course design.

Index Terms—discovery learning; project method; life-long learning; self-efficacy

I. INTRODUCTION

In many engineering fields, especially in computer science and computer engineering, technological changes happen at a fast pace and technologies and frameworks are constantly evolving. While many concepts remain the same, they are embedded in ever changing environments and frameworks, combined with increasing numbers of different features and accessible through modern programming languages with increasingly specialized application programming interfaces. Employers look for graduates that are competent in these new technologies and students are eager to learn the new frameworks and languages during their studies. If possible, university education should, thus, include the new languages and technologies into their courses.

The main focus of university education remains on teaching relevant concepts that have long term validity. But those concepts may be illustrated and applied using different technologies and environments that implement these concepts. Using up to date technologies to gain experiences in practical settings during university education is a very promising approach to increase student motivation and their competitive advantages as graduates on the job market. At the same time it is vital to not only prepare students for the current technological environments and market needs. After graduation, students will be part of the workforce for dozens of years in which technologies will continue to change. It can be expected that they will receive training courses during their work life, but a big part of their life-long learning will equally happen as self-directed learning. Therefore, university education must help students to foster and further develop general skills which are relevant for life-long learning.

As many concepts remain stable, it is important that students understand and grasp those basic concepts. This

enables them to faster adapt to technological advances based on the same invariants. It is therefore mandatory for university education to carefully balance between teaching basic concepts and technologies that are used to implement these concepts. As a simple example let's mention the concept of inheritance in object-oriented languages. If students learn simply how to implement inheritance in a given programming language without understanding the concepts and the characteristics of inheritance, it will be difficult for them to transfer their knowledge to a second object-oriented language. If students, on the other hand, have mastered the concept during their (university) education, it will be much easier for them to identify the concept in the new language and apply previous experiences during their learning process even if inheritance is realized in a slightly different way.

Basic concepts must therefore remain in the core curricula of university education and preferably modern technological approaches be chosen to illustrate and teach these concepts. This poses a tremendous challenge for the faculty as next to research and teaching load they have to keep up to date with the newest technological developments. Becoming and being an expert in *diverse* technologies is a challenge that can hardly be met. Major software companies are aware of the challenges and support students and faculty with additional training and training resources to help them incorporate new technologies into the classroom. The question arises whether it is necessary, that the professor, next to being a domain expert and an experienced teacher, also is an expert in the details of the latest technologies.

In order to be prepared for the market, students need good skills in understanding documentation and acquiring knowledge in new technologies. They equally need to develop strategies how to test for specific characteristics of a programming language or an environment. It is therefore essential to confront students with situations of incomplete information and uncertainty during their university education. Methods of discovery learning are well suited for this purpose as they use existing knowledge of the students and foster their skills and strategies to discover new knowledge. Discovery learning teaching methods date back to the 1960s and have been applied successfully since. The project method is a method of discovery learning and allows students to define freely their project and the detailed learning content. This is in general very motivating for students. The project method is also a good way to confront students with new technologies and foster competences in dealing with incomplete information and problem solving. Discovery learning may therefore be

used to compensate for missing expertise from the faculty in the details of some technology. The students can take part in the research process and thus equally strengthen personal thinking and problem solving skills.

University institutions might have structural requirements for courses like the number of lecture and lab hours or the number and form of examinations. Those requirements might not comply with the requirements of specific teaching methods like e.g. the project method. Careful adaptations must be made that allow applying a teaching method within the given structural frame *without* compromising its main characteristics and advantages.

It is common practice to combine different teaching approaches within one course. If several methods are applied in the same course, they may influence each other and the teaching methods must be well aligned to each of the subset of learning targets they aim at. As different combinations might be more advantageous than others, it is important to also evaluate the combinations of different teaching methods.

In this paper we describe a didactical approach that combines elements of discovery learning and the project method with classical well-structured classroom teaching. This combination allows for the integration of new technologies and for fostering students' motivation and skills for life-long learning. The method can be applied in different course settings that equally include well-structured courses with several written examinations throughout the term. This concept has been successfully applied in two different courses using new technologies (Graphical User Interfaces using MS Expression Blend in 2009 and 2011 and Special Topics in Software Engineering using Cloud Computing with Google App Engine in 2010). We have evaluated the teaching methods. The results of the evaluation support our two main claims:

- a) The method allows for integration of new technologies in education.
- b) The method is helpful to strengthen students' relevant general skills, their self-confidence and self-efficacy.

II. KNOWLEDGE, SKILLS AND COMPETENCES

Two types of knowledge, declarative and procedural knowledge, can be distinguished. *Declarative knowledge* refers to facts that we know and is often explicit whereas *procedural knowledge* refers to skills that we can perform and is often present as implicit knowledge.[1] In general both types of knowledge are tackled within the same course and setting. Declarative knowledge can easily be assessed in examinations. Procedural knowledge generally requires a more careful setup of the assessment and is tested by providing tasks that allow for the application of the skills to be tested. These tasks must be defined carefully in order to allow for a valid assessment.

[2] provides a definition of the terms skill, competency and competence. A *skill* is "the ability to carry out a particular activity consistently", a *competency* "the ability to carry out a complex task that requires the integration of knowledge, skills and attitudes" and a *competence* "the ability to perform a role effectively within a context. This requires a range of competencies." [2]

Using these terms we can describe one main goal of university education as to enable students to acquire

knowledge, skills and competencies to fulfill their future roles in the workplace and society. In the fast changing technological world of today, important competencies are also related to the ability of information gathering, reading skills, testing, and lifelong learning. These are key skills, i.e. "essential, generic skills which are the basis of all successful lifelong learning and personal development." [3] Key thinking skills are related to managing personal states, resource management, distinguishing relevant from irrelevant, communicative skills, self-management, information gathering and organization, determination, hypothesis generation and testing, etc. [4] Many of those skills are also referred to as transferable skills which are equally "of particular interest to the business sector". [2]

University education must also focus on the development of these relevant skills and provide course settings that allow for embedding the training of relevant skills in adequate learning situations and settings.

The achievement of people highly depends on their self-esteem as well as on their self-efficacy, i.e. "people's perception and evaluation of their own ability within specific areas" [2] and is influenced by psychological states, like e.g. "learned helplessness [...] that results when an individual expects that life's outcomes are uncontrollable." [2] "Helplessness can devastate even the brightest learner" [5] and should therefore be taken seriously. Such factors are influenced to a large extend by former experiences and learning experiences as well as success and failure.

In today's technological world, it is not possible to know every detail and also experts may need to ask for advice, research for more detailed information and/or test assumptions and hypotheses. This is not necessarily a sign of failure or lack of knowledge or confidence. It is important to strengthen students in their confidence to know and accept the limitations of their knowledge, to gather and evaluate information and to carry out meaningful tests. Students must achieve confidence in these areas and learn and train the relevant associated skills. Setting high but achievable standards in education, communicating learning outcomes and grading schemes transparently and providing helpful feedback to students can help them to become more confident and develop their own strengths and skills. [5]

It is a responsibility of professors and instructors to provide a respectful and agreeable learning environment that allows students to experience acknowledgement and equally develop their skills the best possible.

It is important that the learning methods and the learning situations are well aligned to the expected learning outcomes and the different taxonomy levels and take into account the situation of the students.

III. THE DIDACTICAL BASICS

The presented teaching method is a combination and adaptation of different methods. It is based on and influenced by these didactical approaches. Before the method will be presented, we will summarize the main characteristics of some teaching methods and introduce basic terms.

With *ex cathedra teaching* we refer to the classical teacher centered method of presenting knowledge to students using presentations, small assignments and allowing students to ask a limited number of questions. This method is especially suited for clearly specified teaching

content that mainly consists in the knowledge, comprehension and partly application cognitive domain of Bloom's taxonomy [6].

Inquiry-based learning is "driven by the need to solve a problem [...] and] learners need [to] engage in new experiences, such as research"[2] It is therefore well suited for learning sequences that require students to find out more details and discover own approaches to solve a specific problem. One specific form of inquiry-based learning is *problem-based learning*, which "makes students responsible for exploring the problem, identifying learning issues, researching materials, and presenting solutions." [2] Problem-based learning is very promising as it "can open up new possibilities for effective student learning [...] and its processes can develop skilled lifelong learners, equipped to work in a rapidly changing world." [7] A specific method of problem-based learning is the *project method* that was already defined in the middle of the 19th century.[8] Students choose and define themselves the project to work on.[8] Regular interactions help the students to keep track with the project goals and identify problems as soon as possible. It is important to note that the learners mainly define the learning outcomes and content when the project method is applied in its pure form. It is therefore not so well suited if students shall acquire learning content that is specified in detail [8] and the learning outcomes cannot be directly tested in written examinations. In the remainder of this paper we refer to inquiry-based learning as *discovery learning* methods. In [9] the term "inductive methods" is used.

In order to fulfill their promises, the methods must be applied in the right context. As [10] suggests, pure discovery learning methods are less effective in terms of immediate and delayed retention as well as transfer to solve new problems than methods of guided discovery learning.

IV. THE INFLUENCING SQUARE: CHOOSING THE TEACHING METHOD

There exist numerous didactical methods and whenever designing a course or a learning sequence, an adequate didactical approach must be chosen. Of course, this approach is to be mainly chosen based on the kind of knowledge and skills that are central to the course outcomes. But the choice of the method is also influenced by four other important factors: the students, the faculty, the institution and the technology. In this section each of these factors will be discussed further.

A. Students

Many of today students are very much aware of what happens in the job market. They study not only based on their interests and to gain knowledge but also to have good chances of an interesting and challenging career in the job market. Being aware of the current situation in the job market and requirements they are very interested in technological advances. They want to receive up to date technical education and gain skills relevant to the job market. Equally students expect to have interesting courses and many still depend (at least partly) on extrinsic motivation and encouraging feedback. Also success in learning is an important factor for student motivation.

B. Faculty

Faculty is generally not only active in teaching but equally in administration and research. The pressure to keep up to date with research as well as with teaching content can be enormous and makes it difficult to balance the workload. For each course the basic concepts must be identified and illustrated, suitable technologies must be chosen to demonstrate and teach these concepts and the teaching methods must be defined. It is important to carefully align the teaching methods with the expected learning outcomes.

In order to allow faculty to base their teaching on the newest technologies, the teaching method should not depend on the thorough expertise of faculty in all specifics of the applied technology. Faculty is supposed to be experts of the taught concepts and be confident with the applied technology but it should not be required that they know all details and specific cases of the used platforms. Otherwise they will hardly be able to manage their teaching load and might also tend to stick quite long with technologies they master well but which are no longer up to date or which are less asked for by the market.

The teaching experience also has to be taken into account in planning the courses and choosing the instructional methods, as "inexperienced instructors [...] can easily be overwhelmed by the additional challenges imposed by inductive methods" [9]

C. The Institution

Based on the course descriptions and the university regulations, the institution defines a frame for the application of teaching methods. E.g. a given course might require written examinations and thus does not allow to be built solely around projects with a graded presentation at the end of the term. The teaching method must therefore also be chosen with regards to the university regulations. Those regulations may include course structure, grading policies, lab and lecture times, attendance, term grades, final examination policies etc. E.g., the German Jordanian University applies a course structure that grades students based on the term achievements (course work) (30 %), a midterm (30 %) and a final (40 %) examination. Midterm and final examinations are generally written examinations with a short duration (between 1 and 2 hours) and thus not suited for, e.g., elaborating on long case studies.

D. Technology

In many technical disciplines, an essential choice is related to the technology used to illustrate and practice the concepts. Learning outcomes related to concepts and competences often can be taught using different technologies.

In some cases the module descriptions of a given course might prescribe the technology to be taught. This is especially the case for many basic courses, as more advanced courses might be based on the technology introduced. For these courses the technological choices are reviewed and adapted on higher levels, e.g. within one department or one school, in regular intervals, e.g. when the curricula are revised.

But in many courses teachers are given the freedom to define the technology they use to teach a given concept. The choice of the used technology is critical and follows among others the following criteria:

- a) maturity: it must be mature enough to concentrate on the teaching content and not mostly on technical problems
- b) resources: documentation and teaching resources need to be available
- c) availability: the technology must be available in time and be affordable for the institution and the students

Above mentioned factors influence the course setting and the choice of the teaching methods which have to balance the requirements and interests of institution, faculty and students while adapting a suitable technology.

V. INTENDED LEARNING OUTCOMES

We have applied the method described in this paper in two different courses. In order to understand the choice of the different teaching methods presented in the next section, it is helpful to know the targeted learning outcomes of these courses.

Table 1 summarizes main outcomes for the two courses on the knowledge, skills and competencies level.

The learning content ranges from very general concepts and process knowledge over specific programming languages towards knowledge of specific technological environments. Necessary skills that are developed are related to personal skills, information gathering as well as documentation skills.

VI. THE TEACHING APPROACH

We propose to separate the teaching outcomes into two sets: knowledge and skills that can be tested in written examinations form the first set whereas a second set is defined by more complex tasks that combine application

of the knowledge as well as personal and generic skills and their development. As [3] mentions, those skills have to be trained on a general level and are relevant for life-long learning.

The first set of learning content is taught using *ex cathedra* teaching (lectures, classroom teaching) combined with *assignments* to apply the knowledge. Especially for application related knowledge as programming, such assignments are essential. If the knowledge can be related to formerly known concepts, the knowledge is presented using a *differential and conceptual* approach. E.g. in both courses that we will discuss, the students learned a new programming language (C# and Python, respectively). As all students had very good working knowledge in C and Java, the programming languages have been presented by specifying the main differences from the known languages and presenting the syntax. This differential presentation was interrupted by common assignments that allowed students to apply the new syntax and directly experience the differences. In this way the new programming languages could be introduced in 4 lectures only. In presenting syntax and main differences, we concentrated on the concepts of the programming language. The main goal was to enable students to read and understand given code, write code segments and research for more detailed information in the API specifications on the Internet or in relevant literature. The introduction remained conceptual as we never intended to cover the complete languages in these lectures. In [11] a similar approach is proposed “when technology is taught, code examples are used, but the technical details are left to the students as reading assignments.” [11]

The technologies / frameworks have been introduced by *demonstrating* students shortly how to develop an application and interact with the tools. These introductions focused on the main interaction points and development

TABLE I. LEARNING CONTENT OF THE TWO COURSES

		Graphical User Interface	Special Topics in Software Engineering
knowledge	technology	MS Expression Blend	Google App Engine
	general concepts	user interface elements usability user interface design internationalization + localization	software engineering process models cloud computing software requirements testing documentation requirements
	programming	C#	Python
XML and XAML		Google Datastore GQL	
skills	documentation	design document	software requirements specification
		Handbook	test document
		project assessment and review / usability test document	
	information gathering	reading and understanding of online documentation	
		development of specific tests to check out the behavior of the languages or the environment	
personal skills	project management time management presentation skills interpersonal skills dealing with incomplete information dealing with new environments dealing with uncertainty		
competencies	applying the skills and knowledge to build a prototype for a given project		

steps to allow students to undertake first steps with the technologies and gain own experiences. An important part of this element was also to provide links to online help, APIs, FAQs, forums, books etc. and give students some tips how to use these. The technologies themselves have not been examined in detail during the course but have been used as a vehicle to implement the projects and discuss, apply and understand the concepts.

The knowledge and skills that have been taught using classical classroom teaching have been applied in a *project* that was based on the introduced technology. We followed the project method as students had a large degree of freedom to define their own projects. The projects were graded within the 30 % course work. The learning outcomes relevant for the written examinations were not dependent on the project, thus the success in examinations did not depend on defining specific topics for the projects. This is especially important as “different students can learn different things from solving the same problems”. [12] Of course the project work had an influence on the knowledge and skills that were taught and how much students mastered specific aspects as it allowed students to further practice and apply them.

All projects have been implemented in student teams fostering interpersonal skills. By choosing the project details and distributing the tasks students also chose which skills to practice most and which knowledge to apply. So students were empowered to make decisions related to their learning contents and learning process which increases student motivation. The student groups received some guidelines that framed the project (e.g. the use of the introduced technology) and ensured that learning during the project phase was directed towards the course content. A detailed *grading scheme* for the project task has been made available in time to the students. They could, thus, align their activities in order to receive good grades.

During the project students also practiced additional skills like *documentation* skills by preparing specific documents. We did not ask students to provide all documents that are normally part of a software project, but depending on the course content two different types of documents have been specified. This gave students more time to spend on each of the required documents. The grading scheme for the documentation equally has been made available in time to the students and they received detailed feedback in order to improve their documentation skills.

During the project phase some lectures were reserved for the work on the project. Student attendance also was required during these lectures and the teams worked on their projects, and discussed questions and proposals in teams or with the lecturer. This time was also used to discuss common problems and the lecturer helped solving problems mainly by pointing students to the right resources. As students defined and realized a project, they faced many concrete questions about which library functions to use, specific settings of the product, etc. During this phase students worked a lot with the information in books (as far as available), the available documentation on the Internet as well as with posts in blogs and technical discussion groups. This helped students to strengthen their information gathering skills, practice research tasks, and to develop strategies to test the given environment for specific behavior that was not described in enough detail. Those lectures were oriented towards discovery-learning,

team work and strengthening communication skills. A student forum has been created on our e-learning platform¹ to allow students to exchange tips, code samples and request for help. The project tasks were mainly oriented towards to application, analysis and synthesis cognitive domain of Bloom’s taxonomy [6].

The *written examinations* were based on the knowledge and skills that were taught using classroom teaching methods and assignments. The questions have been more difficult in the final examination allowing for inclusion of the experiences students gained during the practical project work. The *learning outcomes* tested in the examinations were handed out to the students in written form well ahead of the examinations. This allowed them to specifically prepare for the examinations. It also helped to reduce uncertainty and confusion that might arise as the student projects have been very different and students’ experiences quite individual. In such cases often the question arises among students whether the examination is based on some content that only part of the student projects covered.

The projects were *presented* to the whole group at the end of the term. This presentation included a lessons learned part. Common reflections of the project and experiences concluded the presentation to make sure that students learn as much as possible from this practical part of the course. The final evaluation of the project tackles the evaluation cognitive domain after Bloom [6].

VII. REVIEW OF PROJECTS AND DISCOVERY LEARNING

In both courses students have worked on projects in small student teams. In the Graphical User Interface course we specified elements that the user interface should have, like multiple window areas, interaction with mouse and keyboard, use of menus and dialog boxes, use of buttons, text entry and text read-only controls and at least two different kinds of selection controls. Students could define their own application area to implement a user interface that includes above mentioned elements. The student projects had a large variety, ranging from an administration software for study related documents over a health club simulation, a horoscope information system and a tool to administer ID3 tags of MP3 files to an implementation of the game “who wants to be a millionaire?” The ideas of the students were surprising and challenging. As the students themselves chose their application area they were highly motivated and spent a lot of effort to implement *their* project. The motivation was constantly high and the results of all groups were noteworthy. Especially groups with students whose GPAs are generally weaker were performing surprisingly well. And most of all, students really had fun implementing their projects and were proud to present them at the end. The grading scheme has been made available to the students during the project phase and was aligned to the project assignment.

The project was accompanied by required documentation, in this case a handbook as well as a project assessment and review document (2009). In 2011 students have carried out a usability test on a project of another student group and submitted a short report as assignment. The criteria for these documents equally have been made transparent. Not surprisingly students did not find these

¹ <http://www.moodle.org>

assignments so interesting and spent less energy and care in preparing the documentation. It was a good idea to not ask for too many different documents, in order to allow students to concentrate on selected documents. This also allowed us to be more stringent in our expectations and grading.

In the second course, Special Topics in Software Engineering, the project topics were more precisely specified. This course was mainly given in German language to confront our students with the technical terms prior to their exchange year in Germany. The project itself (a Google App Engine application) was more technology oriented and had less possibilities for nice user interface interactions. We presented four possible applications areas on a higher level (e.g. an application that stores information links) that could be useful for students during their exchange year. We opened up the perspective that upon successful completion of the project they could upload the application itself to the Google cloud to allow for communication and information exchange during their exchange year. It could be observed that these projects were less motivating for the students, as their focus was more directed and not completely chosen by the students. But also the design of the projects was more on the architectural and internal level as with a user interface project that allows students to create attractive and fancy pages, dialogs, and effects which also influences the students' motivation. Nevertheless it could be noted that the students were very engaged in the projects and one group uploaded their application to the Google cloud.

During this project students were required to develop a software requirements specification (SRS) and a test document. It is always difficult to motivate students to document, but requiring students to write the SRS before the implementation and developing and document the tests with reference to the requirements helped students better understand the role of these documents. This was also mentioned in the student feedback, even if students did not start loving documentation tasks.

In both courses students were required to find information in the Internet and the API descriptions and did not have complete code samples they could use. This experience helped students to better appreciate good documentation. They spent a lot of time searching for information and were motivated to solve their problems on their own – and proud when they did. They worked much more independently than expected and performed very well in information research. They tested many things and used the technologies and frameworks very well.

VIII. STUDENT FEEDBACK AND EVALUATION

Students in summer 2010 (Special Topics in Software Engineering) and in spring 2011 (Graphical User Interfaces) filled in a detailed evaluation related to the teaching method. As only 13 resp. 12 students participated in this course, the results are not conclusive but allow for a *first assessment* of the teaching method.

The first part of the evaluation consisted of closed questions related to the students' expectations of the course and the professor. As we propose the described teaching method as a way to include new technologies in the curricula in which the professors are not yet fully experts, we wanted to verify our assumption that students would accept this approach. To which extend do students expect a

professor to be an expert? How acceptable is it for students to search information in FAQs and in the Internet? Do students accept to have different topics which are partly only introduced on a conceptual level (like the programming language in the course)? Do students require textbooks that cover all the topics? What is the role of the professor in the teaching process? Who is responsible for the students' motivation?

Fig. 1 depicts the 23 questions /statements of this part of the evaluation. Some questions target a similar topic to cover multiple dimensions of specific expectations. Students could choose from 4 different possibilities that have been translated to integer numbers (strongly agree = 4, agree = 3, disagree = 2, strongly disagree = 1). Depending on the question a higher value can denote a positive or a negative statement.

Fig. 2 shows the numeric results for the questions of the first part of the evaluation.

Results of this part of the evaluation indicate that:

- It is okay for students to have many topics covered in one course without a specific textbook and to have to read online documentation. Students like to use new technologies and accept

1. It is important for me to have a specific textbook to revise the content of the lectures.
2. It is important to have mainly one topic that is taught (not different topics).
3. A whole course should be focused on teaching a new programming language. It is not sufficient to show the differences to other languages.
4. I expect my professor to have profound knowledge of all material she is covering.
5. A professor is a teacher who conveys knowledge to me.
6. A professor is a teacher who helps me to build my own knowledge.
7. A professor is directing my learning process.
8. A professor is an enabler of my learning process.
9. I like to discover details on my own by testing programs, looking in the internet or into documentation.
10. I prefer clear instructions step by step.
11. Class projects must be specified in detail.
12. I want to try things out myself. I do not need to have detailed examples.
13. I do not really learn when we only scratch topics. I need to spend time on details.
14. If a professor first has to try out some details before she can answer questions, she is not well enough qualified for teaching this subject.
15. A professor should never learn the technology more or less in parallel to the students but well in advance.
16. I like to use new technology in courses, even if I face sometimes some problems.
17. In order to learn something, I need an environment in which I have complete control and can observe everything. The applied environment was not so good for studying.
18. A professor is mainly responsible for my motivation to study.
19. I need to find the motivation for studying in myself.
20. I learn better if I can discuss the content with my colleagues.
21. I expect from the software and learning environment we use that there will be no problems.
22. Reading FAQs or software documentation in the internet should not be required in undergraduate university courses.
23. A professor should not have expectations and grading schemes for projects (documentation and presentation).

Figure 1. Questions related to students' expectations towards the professors and course in the first part of the evaluation

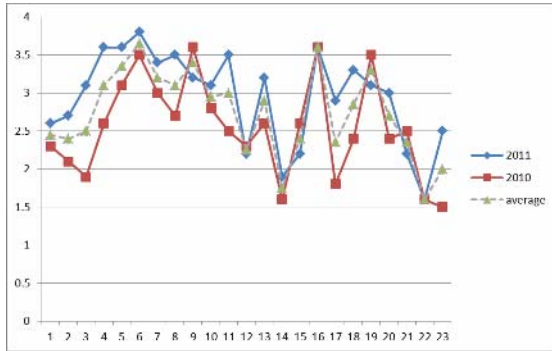


Figure 2. Results of the evaluation of the first part (students' expectations)

that this might pose some problems. They equally accept when they do not have complete control over the environment used. They like to discover details on their own but on the other hand, students are not so eager to try things out by themselves but want detailed examples to build on.

- Students mainly expect professors to have profound knowledge in the teaching topic, but the fact that the professor needs to try out specific details before giving an answer has no influence on the perceived qualification of the professor. But they expect the professor to learn the technology well in advance to the students, i.e. to the course.

Related to the role of the professor in the teaching process the answers do not give a clear picture. The professor is perceived as conveying knowledge and, in the same time, helping to build own knowledge. On one hand the professor is seen as directing the learning process on the other hand as an enabler. Many students believe that the professor is mainly responsible for their study motivation whereas they all agree that they have to find the motivation for studying in themselves. This reflects the many roles and expectations that we phase during the teaching process.

The average values of both courses differ most in questions 3, 4, 8, 11, 17 and 23. This may be explained by the differences in the student groups. Students in the summer course of 2010 were in the 3rd to 4th year and preparing for their German exchange year whereas students in the course of spring 2011 were in the 2nd and 3rd year, some realizing their first real course project. Therefore it is not surprising that more mature students have less expectations and a lower demand for detailed explanations.

The second part of the evaluation was related to the skills and attitudes of the students, their perception of their own skills and perceived changes due to the course. With this block we gathered information about the students' attitudes and self-efficacy. Fig. 3 shows the 17 questions of this part of the evaluation.

We evaluated general thinking skills (reading and information gathering skills), time management and team skills, the willingness to professionally tackle software projects (planning, designing, documenting and testing), their feeling of responsibility for their own learning process and success (which also refers to the topic of learned helplessness), their confidence and self-efficacy (to be

1. I am able to try out things that are not clear to me by creating small programs / examples.
2. I have good reading skills of documentation / FAQs, etc.
3. I feel confident even if I do not know all details of a product / environment.
4. I am curious to test out new products.
5. I am willing to spend time on the planning, the design and the testing of a project.
6. I understand the importance of documentation.
7. I have good team skills.
8. I know how to tackle a new project.
9. My time management skills are good.
10. I feel responsible for the outcomes of my learning process.
11. I feel confident that I can keep up with new technologies and technological change.
12. I feel confident that I will be a good computer engineer / computer scientist.
13. I have a good potential for a successful career.
14. I am confident in my own general skills.
15. I combine and apply a lot of knowledge (theoretical and practical) that I learned in different other courses.
16. I have a good overview of the interconnectedness of the different topics of CS and CE.
17. I want to be involved in large projects.

Figure 3. Average values of the second part of the evaluation

successful in the fast changing technological environment and create a successful career), as well as their wish to be involved in large projects. In all the mentioned fields students had a good to high confidence / positive attitude. The points with the lowest ratings were the confidence when using a product / environment they do not know in detail (3), the time management skills (9), the impression that they combine and use knowledge of different courses they already took (15) and their understanding of the interconnectedness of different fields and topics (taught in different courses) of their study plan (16).

Fig. 4 shows the average values of the second part of the evaluation (as of the end of the term). Students could choose from 4 different possibilities that have been translated to integer numbers (strongly agree = 4, agree = 3, disagree = 2, strongly disagree = 1).

The points with the best scores in summer 2010 reveal students that are confident to be able to try things out (discovery) (1), understand the importance of documentation (6), accept their responsibility for the outcomes of the learning process (10), are curious to test new products (4),

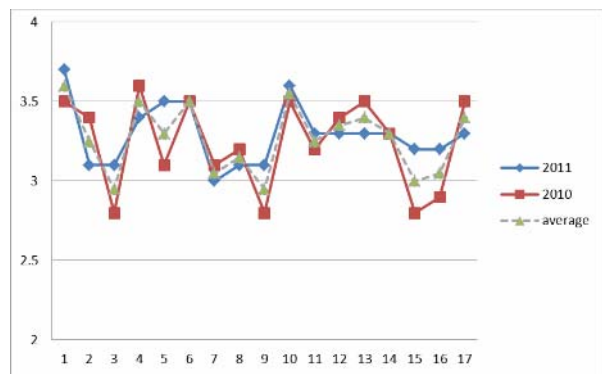


Figure 4. Questions related to students' thinking skills, confidence and self-efficacy

want to be involved in large projects (17), feel confident that they will be a good computer scientist / computer engineer (12) and have a good potential for a successful career (13). The group of students in spring 2011 (2nd and 3rd year) were most confident in trying things out (1), willingness to plan, design and test a project (5), understood the importance of documentation (6) and felt responsible for their own learning process (10).

We equally were interested in how the students' interests and confidence changed during the term. In summer 2010 we assessed the individual perception of the students. We have added an additional column to all statements of Fig. 3 and asked whether they agreed more with the statement after the end of the term. Not all students answered that question. In order to evaluate whether negative changes might occur, in spring term 2011 we assessed the current level of agreement with the statements of Fig. 3 in the first week of the term and in the last week in the term. By comparing the values we identified existing changes. Fig. 5 depicts the results of both terms. It is obvious that the changes in 2011 seem to be less, but we need to take into account that the assessments were different. The values of 2010 depict the changes that the students themselves perceived. As self-efficacy has a lot to do with self-perception, such changes are important. On the other hand the values of 2011 depict the differences in the self-perception of students at two different points in time. The result of 2010 depicts whether and to which extend the course could change students' attitudes, their *perceived* confidence and their self-efficacy. The points in which the course seems to have achieved a change in many students are the following: Students are more willing to spend time on the planning, design and testing of a project. And they understand the importance of documentation better. This shows that the teaching methods helped

to change the mindset of students and teach skills and attitudes that are relevant for success in the professional field. The course also increased students' curiosity to test out new products and their confidence to deal with the new environments even if they may not yet know all details. These attitudes are relevant in the fast changing technological world as our students will be confronted with many new products and concepts during their professional career. This confidence is equally an important factor to master self-studying. One important change is that students are more confident that they can keep up with new technologies and technological change. This confidence is relevant for students to look forwards to their professional career as they are all aware of the fast pace of technological changes and advances.

As in spring 2011 we have assessed the values at the beginning and at the end of the term, we could make a more detailed analysis. Fig. 6 depicts the average level of agreement at these points in time. We can observe that all average values have remained the same or improved. But a more detailed analysis reveals that at the end of the term some students agreed *less* to some statements than at the beginning of the term.

We will shortly discuss two different student 'profiles' from the evaluation.

Some students improved a lot their confidence and self-efficacy. In Fig. 7 one such student profile is depicted. In 11 from the 17 statement we could observe higher values. Especially four changes are noteworthy. At the beginning of the term the student disagreed with the statements "I am curious to test out new products.", "I know how to tackle a new project." and "I feel confident that I can keep up with new technologies and technological changes." After the course he strongly agreed with these statements. The biggest change is related to the self-perception of the students' time management skills. While at the beginning of the course he strongly disagreed to have good time management skills he strongly agreed after the course.

Many students show a more mixed student profile where they agree more to some statements than before while agreeing less with some other statements. The profile of one such student is depicted in Fig. 8. After the term he agrees less to 5 different statements that he has *strongly agreed* at the beginning of the term. A striking difference can be observed with the statement "I feel confident that I can keep up with new technologies and tech-

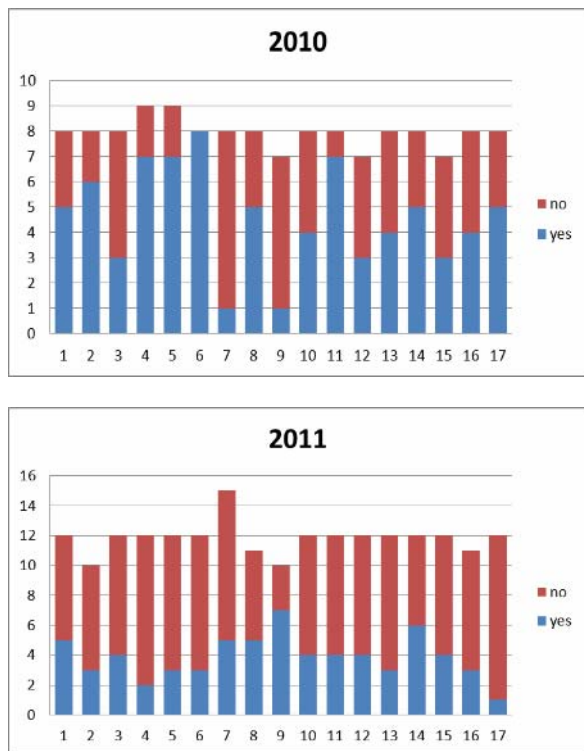


Figure 5. Changes in students' self-perception

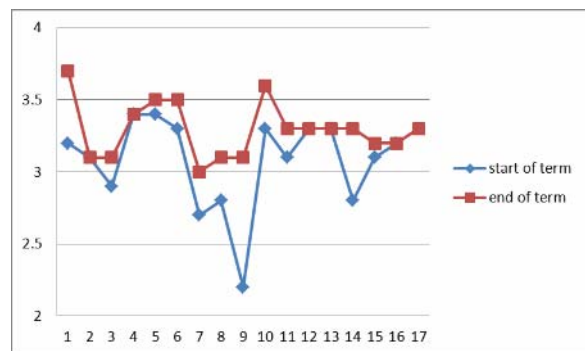


Figure 6. Average values of students' self-perception at the beginning and end of spring term 2011

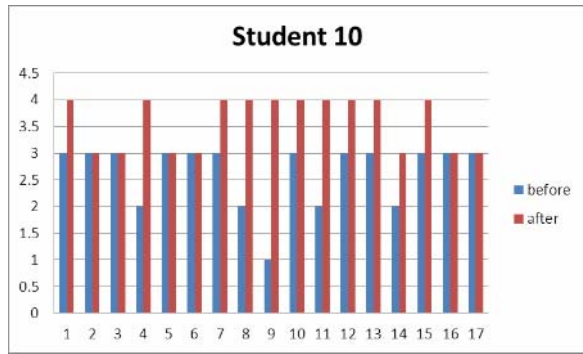


Figure 7. Student profile 1

nological changes” that fell down from “strongly agree” to “disagree”.

It is obvious that the evaluated values do not describe precise profiles. They depict momentary perceptions. But our evaluation results clearly show that the teaching method has an effect on students’ self-perception and thus on students’ self-efficacy. While the average effects are very positive the different evaluation method in spring 2011 and the detailed analysis have shown, that the teaching method and uncertainty can shake some students’ self-confidence and convictions related to some aspects. While this is a necessary process for learning and self-improvement care must be taken that students are not discouraged and learn helplessness.

The third part of the evaluation consisted of three open questions related to which skills improved, whether the approach to problems changed and whether the course design helped students to become a better computer scientist or computer engineer respectively. Most of the students answered to these questions and the feedback was exclusively positive. Students mentioned the project oriented skills, interpersonal skills, general thinking skills and how attitudes about how to tackle software projects changed.

IX. SUMMARY AND CONCLUSION

In this paper we have described a didactical approach that combines elements of classroom teaching with discovery learning and the project method. We have applied this approach in two different courses on the bachelor level. The learning outcomes on the knowledge as well as on the skills level have been presented in this paper. We have discussed the student projects.

The method has been applied successfully and the stu-

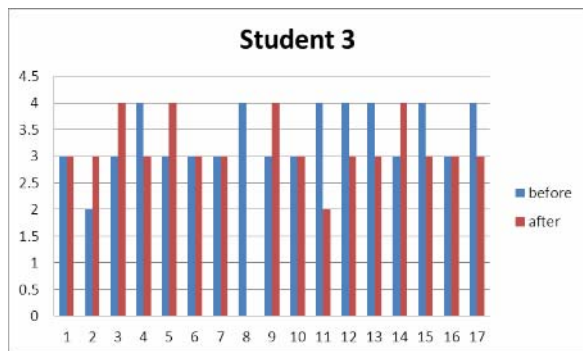


Figure 8. Student profile 2

dents were very motivated. The student feedback indicates that students accept the part of discovery-learning and the fact that the teacher herself is not an expert in all the details related to the used technology.

We equally have assessed students’ self-perception to evaluate the potential of the described approach to change students’ self-confidence and self-efficacy. The evaluations have clearly shown that the didactical approach has the potential to influence students’ attitudes towards important professional tasks and to strengthen their self-efficacy. It equally has revealed that while strengthening the average attitudes, it can shake some students’ convictions. In future work we will gather more data in different courses with student projects to evaluate these effects more closely.

The teaching method has been applied in courses small student groups. As it includes student projects that require detailed feedback, the workload for the professor cannot be neglected. But it is comparable to other teaching approaches that require regular feedback on coursework like e.g. homework. The students worked intensively during the project phases and did a lot of research and problem solving within their teams or with the help of fellow students. This reduces the preparation time related to the details of the applied technology compared to a course in which the lecturer presents new technologies in detail. But it also poses a big challenge to the professor – students expect a professor to be an expert and if he fails this expectation this might have a negative impact on the learning setting.

The two courses fulfilled specific requirements: both courses have been advanced courses and students have had good working knowledge in e.g. programming languages. Otherwise it would not have been possible to present the new languages with the described differential and conceptual approach in few lectures only. Prior to the given courses the professor knew most of the students. In such a setting it can be assumed that students accept the fact that the professor has to research for some specific answers more easily. It might be more difficult to apply the described discovery-learning approach with students that do not know the professor and might easily question his competences. As the course assessment is mainly based on the outcome of the written examinations and the students made very different experiences during the project phase, it was important that students trusted the binding character of the published learning targets for the examinations as well as of the grading schemes for the projects. Otherwise the students’ stress might have significantly increased and the learning experience be disturbed. Equally it is noteworthy to mention that the projects have been graded against the published grading scheme and not with reference to each other. We did not want to introduce this competitive element into the grading to allow especially also the average and weaker students to experience their skills during the discovery-learning process and the project and increase their self-efficacy level without being stressed because “better” students were faster or seemed to do something much more “fancy”. Although the projects were very different, we did not experience that any student group took it “easy going” and was especially lazy.

REFERENCES

- [1] A. E. Lawson: "Neurological basis of learning, development and discovery: implications for science and mathematics instruction", Kluwer Academic Publishers, 2003.
- [2] A. Jordan, O. Carlile and A. Stack, "Approaches to learning: A guide for teachers", Open University Press, 2008
- [3] P. Scales: "Teaching in the lifelong learning sector", Open University Press, 2008.
- [4] E. Jensen: "Teaching with the brain in mind", 2nd Ed., Association for Supervision & Curriculum Development, 2005.
- [5] E. Jensen: "Brain-based learning the new paradigm of teaching", 2nd ed., Corwin Press, 2008.
- [6] Huitt, W. (2009). "Bloom et al.'s taxonomy of the cognitive domain". Educational Psychology Interactive. Valdosta, GA: Valdosta State University. Retrieved 21st of June 2011 from <http://www.edpsycinteractive.org/topics/cogsys/bloom.html>
- [7] P. Stephenson and V. Galloway, "Problem-based learning" in G. Foley (Ed.), "Dimensions of adult learning", McGraw-Hill Education, 2004, pp. 265-275
- [8] K. Frey, "Die Projektmethode", <http://www.uni-leipzig.de/~lbpaed/wb/media/literatur/Frey%20-%20Die%20Projektmethode.pdf>, retrieved on 21st of June 2011.
- [9] M. Prince and R. Felder, "The many faces of inductive teaching and learning", Journal of College Science Teaching. March / April 2007
- [10] R. E. Mayer, "Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction", American Psychologist, January 2004, pp. 14 - 19. <http://dx.doi.org/10.1037/0003-066X.59.1.14>
- [11] O. Barzilay, O. Hazzan and A. Yehudai. "A multidimensional software engineering course", IEEE Trans. Edu. vol. 52, no.3, pp. 413-424, Aug. 2009. <http://dx.doi.org/10.1109/TE.2008.930094>
- [12] J. M.M. Sanden, S. van der Vosniadou, and J. Terwel, "New learning in science and technology" In: P.R.J. Simons, J.L. van der Linden & T.M. Duffy. (Eds.). "New Learning: three ways to learn in a new balance", Kluwer Academic Publishers, 2000, pp. 119-140.

AUTHOR

Christina. B. Class studied Business Informatics at the University of Mannheim, Germany, and made her PhD in the Computer Engineering and Networks Laboratory at the Swiss Federal Institute of Technology in Zurich, Switzerland. From 1999 to 2008 she was professor for computer science at the Lucerne University of Applied Sciences and Arts in Switzerland. In 2008 she joined the computer science department at German Jordanian University in Amman, Jordan. Dr. Class has several certificates and degrees in computer science didactics. (Christina.Class@gju.edu.jo)

This paper is an extended and updated version of the author's paper "An Approach for Teaching New Technologies and Life-Long Learning Skills", presented at the International Conference EDUCON2011, held in April 2011 at PSUT, in Amman, Jordan. Received 28 June 2011. Published as resubmitted by the author 15 October 2011