Assessment & Evaluation in Higher Education
Vol. 35, No. 4, July 2010, 367–380

Routledge
Taylor & Francis Group

The difficult bridge between university and industry: a case study in computer science teaching

Jan Schilling[a]* and Ralf Klamma[b]

[a]*Department of Industrial and Organisational Psychology, RWTH Aachen University, Aachen, Germany;* [b]*Department of Computer Science, Informatik V (Information Systems), RWTH Aachen University, Aachen, Germany*

Recently, there has been increasing criticism concerning academic computer science education. This paper presents a new approach based on the principles of constructivist learning design as well as the ideas of knowledge transfer in communities of practice. The course 'High-tech Entrepreneurship and New Media' was introduced as an interdisciplinary project management lab in which students collaborate in groups to develop software solutions for authentic problems. Main goals were the tighter integration of university and local start-up companies, an intense knowledge transfer on software engineering methods, as well as the implementation of constructivist learning principles in academic teaching. This paper presents the background and structure of the course as well as the results of a formative evaluation. While being successful in introducing a course based on digital-media assisted, constructivist learning arrangements, establishing lasting communities of practice between university and industry is still an open issue. After discussing several reasons, the paper concludes with a list of general recommendations on how to improve the approach and its implementation.

Keywords: collaborative learning; situated learning; communities of practice; constructivist learning design; course evaluation; formative evaluation; interview; qualitative methods

Introduction

While some progress has been achieved, computer science (CS) teaching is still mainly based on the traditional lecture format (Van Gorp and Grissom 2001). Renkl, Mandl, and Gruber (1996) have pointed out that this educational approach often results in 'inert' knowledge that is not used for problem solving and concrete professional practice. In recent years, there have been increasing efforts to advance computer science education by extending new approaches like project-based (e.g. Coppit 2006; Fincher and Petre 1998) or problem-based learning (e.g. Nuutila, Törmä, and Malmi 2005). These efforts share the same concern to advance more active and authentic learning (Stein, Isaacs, and Andrews 2004) in computer science education. For this purpose, they mainly draw on two lines of theoretical development in the field of learning theory: constructivist (e.g. Collins, Brown, and Newman 1989) and situated or socio-cultural learning theories (e.g. Brown and Duguid 1991; Lave and Wenger 1991). Van Gorp and Grissom (2001) argue that they can be characterised by

*Corresponding author. Email: Jan.Schilling@psych.rwth-aachen.de

ISSN 0260-2938 print/ISSN 1469-297X online

DOI: 10.1080/02602930902795893
http://www.informaworld.com

the fundamental aspects of context (i.e. solving authentic problems or possibly simplified versions), construction (i.e. constructing the knowledge necessary to solve the problem by engaging in meaningful activities) and collaboration (i.e. working together with peers to solve the problem). Situated learning (cf. Cousin and Deepwall 2005) describes the process by which novices become participants within a community of practice (CoP). A community of practice can be described by three main features (Wenger 1998): domain (i.e. the members of the community share a domain of interest), community (i.e. the members engage in joint activities and discussions, help each other and share information) and practice (i.e. members develop shared practices to solve problems: experiences, stories and tools). Learning is a matter of participating in practices and actions of a community and in the end becoming a legitimate practitioner (Paavola, Lipponen, and Hakkarainen 2004). Constructivist and situated approaches do not exclude each other, but can be combined and are even considered to supplement each other (e.g. Mandl, Huber, and Renkl 1996). Combining the two approaches seems especially important, as Ben-Ari has convincingly argued that situated learning approaches alone cannot be simply adopted for computer science education as they ignore the gap 'between the world of education and the world of high-tech CoPs, which demand extensive knowledge of both CS subjects and application areas' (2004, 98). The present paper describes the evaluation of the course 'High-tech Entrepreneurship and New Media' (Klamma, Rohde, and Wulf 2004; Klamma et al. 2003) which is now in its fifth year of existence and was held at two German research universities. The goals for the course were two-fold:

(1) The first goal was to advance software development practice on part of the students by using a constructivist learning design. This implies that groups of students work together on authentic problems of local companies to allow the development of shared knowledge and thereby mutual learning (cf. Gruba and Sondergaard 2001).
(2) The second goal was to establish communities of practice between students, academics and local companies by providing entrepreneurs with a platform to reflect upon their experiences, to sensitise students for entrepreneurial activities and, in return, to transfer new concepts and methods of software engineering from university into company practice. By this, a durable network of institutionalised relationships is built (i.e. building of social capital, Bourdieu 1983; Cohen and Prusak 2001).

Course structure

Course design

Most participating students were enrolled in German Diplom studies[1] in computer sciences or international students from the master programmes in software systems engineering and media informatics. As a consequence, the language of the course was English.

During the first meeting, the students were introduced to the basic concept of the course. They selected one of the presented project tasks and formed corresponding project teams (with seven and eight members, respectively). This rather big group size became necessary because only two of three companies, which had formerly announced their interest, actually participated in the course. It was calculated that the

work effort would comprise of 50 hours of student time for the course itself and 150 hours for the project lab. At the end of the term, the students presented an alpha prototype as a feasibility study. The successful development of the prototype was the precondition for the students to pass the course.

Very early in the term, the lab members held meetings with their companies to develop the project goals and ideas on necessary working practices. The project groups and tutors compiled and agreed upon precise project objectives for the execution of the practical tasks and a project plan. In the following week, project management techniques and software engineering methods were presented to the students in a two-day workshop (called 'mini-lab'). They carried out a small scale project leading through all steps and methods of the planned course.

With regard to the accompanying lectures, speakers from academia and practice rotated. The external speakers were predominantly entrepreneurs and contributed to the course with their practical experiences. In three lab reviews during the course, all project groups presented their results to the other participants and discussed the further course of action together with the lecturers and experts. Therefore, these review sessions also served as opportunities to exchange experiences and to benefit from the progress and findings of the other project groups.

To support the participants in the development of presentation skills, a combined training and coaching programme was organised. A half-day presentation training was aimed at providing and practising basic knowledge of the preparation and realisation of business presentations. In three coaching sessions, the participants were given detailed video-based feedback on their review presentations to improve specific aspects of their presentation behaviour. At the end of the term, all students, lecturers and entrepreneurs came together in a final meeting to present and discuss the projects' outcomes.

Community information system and software methodology

Group-oriented learning processes, especially among the student teams as well as between them and their academic instructors, were facilitated by a community system. The community system is meant to serve as a discussion forum among students and guest lecturers from the industry and academia. Weekly lectures – stored also in the community repository – encourage students' reflections on their tasks. Ultimately, the community system should be an archive of the work done during the course. It was decided to deploy the BSCW system (Basic Support for Collaborative Work: Appelt 1999) for several reasons. It is equipped with a login system to protect the course members' interactions and offers – compared to other systems like Moodle, Blackboard or WebCT – a lot of functionalities (e.g. awareness function) as well as multilingual interfaces. Other programming tools necessary to support community-oriented work settings such as a source code management system and various editors were also installed. For their work, the students were able to use a database lab room.

In order to choose an appropriate software methodology, it was decided to introduce a combination of project management methods and extreme programming (XP) techniques (Beck 1999). XP is regarded as a methodology suitable for short projects with small software development teams (cf. Johnson and Caristi 2002). XP is one of the agile software development methods that are characterised by an emphasis on customer needs, short releases and heavy testing along the software development process (Bergin et al. 2004). The companies were very interested in the XP approach.

A reflective session on the use of XP was introduced to make students and practitioners aware of its advantages and difficulties (Stephens and Rosenberg 2003).

Evaluation of the course

In a formative evaluation (Scriven 1967) of the course, qualitative interviews were conducted with three different stakeholder groups. Qualitative evaluation seemed more suitable for studying learning processes than traditional objectivist evaluation methods (cf. Hadjerrouit 2005). At three times, random samples of participants were asked about their expectations of and experiences during the course. At Time 1 (shortly after starting the course) eight participants, at Time 2 (midterm) seven and at Time 3 (after the end of the course) six participants were interviewed. In total, 21 interviews were conducted with 13 of the 15 participants. Furthermore, the two course instructors were interviewed as well as the two contact persons of the companies involved after the end of the course. The interviews were conducted by four psychology students who were trained and instructed for this task. The interviews lasted between 30 and 60 minutes, depending on how elaborately the interviewees answered the questions. Tape recordings were used to retain the statements of the interviewees which were then transcribed into written protocols. The open-ended questions posed to the participants, the course instructors and the company tutors were partly overlapping (see Table 1). Thereby, the impressions of different stakeholders regarding the course could be compared.

In a stepwise procedure of data analyses and control checks, all interview material was analysed for its content (cf. Schilling 2006). The first step was to reduce the interview material to its basic content (‘paraphrasing’) by deleting all the words not essential for an understanding of the statement and transforming the sentences into a short form. Afterwards, all statements not important with regard to the question at hand as well as all recurrent statements of an interviewee were deleted (‘reduction’). To make sure that no important content would be excluded from further analysis, a second person checked the resulting text material with reference to the original texts. In the next step, all remaining material was coded in a process of inductively categorising the answers according to the different questions posed. Again, this categorisation was checked by a second person for accuracy. Ultimately, descriptive statistical analyses were carried out to complement the qualitative analysis. Here, the basic measure of person frequency (i.e. how many of the subjects address a certain theme) was regarded to evaluate the importance of a certain statement.

Results

As stated above, the study was aimed at evaluating the implementation and effectiveness of the course. From a product-oriented perspective the course can be rated as successful. Both project groups developed fully functional technical solutions for the tasks. For the current purpose, the main interest concerns the question, if the course was able to achieve its goals of fostering sustainable cooperation, communication and collective learning between the different stakeholder groups. Especially the chances and limitations of the approach to implement communities of practice at university level need to be examined more closely. As the questions were open-ended, the interviewees were able to give more than one answer to each question. Therefore, the frequencies of a question do not simply add up to 100%. Furthermore, not all the

Table 1. Interview questions.

Participants	Academic tutors	Company tutors
Personal and academic background	Personal and occupational background	Personal and occupational background
Experience with project work, entrepreneurship and software development		Reasons for joining the course and relation to the topic
Reasons for participating, expectations and experiences	Expectations of the course	Involvement into the course and expectations
Evaluation of the course components: mini-lab, presentation training and coaching, lectures, extreme programming	Evaluation of the course components: mini-lab, presentation training and coaching, lectures, reviews, extreme programming	Evaluation of the course concept and components: interdisciplinary approach, lectures and reviews
Evaluation of the task: content, time-table		Evaluation of the task: effort, time-table
Cooperation in the project team (arrangement of tasks, atmosphere, hierarchy, group cohesion, cultural diversity, relationship to the other team)	Cooperation in the project team (atmosphere, technical and social skills, motivation, hierarchy, group cohesion and learning community, cultural diversity, relationship between the teams, observable learning processes)	Cooperation in the project team (atmosphere, technical and social skills, motivation, cultural diversity, observable learning processes)
Cooperation with the company (contact person, insight into company policies, learning from and of the company)	Cooperation between students and company (insight into the company, learning from and of the company, further cooperation with the companies)	Cooperation with the students (contact persons, students' insight into the company, learning from and of the company, further cooperation)
General conditions (BSCW, linkages between lectures and project, relationship between tutors and students)	General conditions (BSCW, cooperation with external lecturers, cooperation with the students, contact with the other company and related project team)	General conditions (BSCW, cooperation with lecturers and academic tutors, contact with the other company and related project team)
Suggestions for improvement	Suggestions for improvement	Suggestions for improvement

interviewees responded to every question. Consequently, the total number of respondents varies across the questions and is always indicated in the results.

General expectations and experiences

When asked for their *reasons*, most of the students stated that participation in courses like these was mandatory for them (see Table 2). But at the same time they were also interested in this specific course because of its content or its project character. These reasons were also reflected in their manifold *expectations of the course* which included a wide variety of aspects (see Table 2) with 'teamwork' and 'technical skills acquisition' as most frequently mentioned topics. When asked *how far these*

Table 2. General expectations and experiences.

Reasons for participation?	Mandatory course	7 of 13
	Interested in the content of the course	7 of 13
	Project character of the course	2 of 13
Expectations for the course?	Experiencing teamwork	8 of 11
	Acquiring technical skills	6 of 11
	Learning more about project management	4 of 11
	Acquiring presentational skills	3 of 11
	Getting to know corporate practices	3 of 11
Expectations fulfilled?	Acquiring technical skills	9 of 13
	Acquiring presentational skills	5 of 13

expectations were met, the students placed emphasis on the acquisition of technical skills and presentational skills while teamwork was regarded inconsistently. While four students stated that working with others was a learning experience, the same number of participants evaluated teamwork negatively.

The company practitioners approved of the concept of the *course*, although they claimed that the focus was more on technology than on entrepreneurship. While one of them did not have specific *expectations* of the course, the other expected and experienced a results-oriented cooperation with the students.

Evaluation of course components

In evaluating the different components of the course, the students revealed very specific opinions (see Table 3). The BSCW system was evaluated positively as helpful for the exchange of information and files. Likewise, the introduction to software engineering methods ('mini-lab') at the beginning of the course was predominantly seen as helpful, especially in getting to know the task and the other participants. Also, the *training* (i.e. the half-day workshop on presentation behaviour) and *coaching* of *presentational skills* (i.e. the three video-feedback sessions) received very positive ratings. The practice of XP was evaluated inconsistently. Four of 10 students welcomed the procedure as helpful, four disliked it. The lectures that aimed at complementing the project work were seen as negative by most of the students. Many maintained that planned lectures did not take place at all or that their content did not relate to the project.

Table 3. Evaluations of course components.

Positive evaluations:	
BSCW	10 of 11
Presentation skills training	9 of 11
Introductory workshop ('mini-lab')	8 of 10
Video feedback coaching	8 of 11
Extreme programming (XP)	4 of 10
Lectures	2 of 11
Negative evaluations:	
Lectures	8 of 11
Extreme programming (XP)	4 of 10

The company tutors did not attend the *lectures*, both claiming that their involvement in the course should focus on presentations, meetings with the group and the use of BSCW. The BSCW was evaluated as a very helpful tool for exchange of information, but not as necessary for project management. They preferred communication via mail and phone.

The academic tutors were generally pleased with the introduction to applied software engineering methods ('mini-lab'), only one of them reckoned that preparation of the presentations was not always successful. *Coaching and presentation training* were evaluated positively as the quality of the presentations improved during the course. The use of XP was rated inconsistently: while one tutor was disappointed, the other was more positive. He admitted that the students in his group should have used the method more than they actually did, which he attributed to the great time pressure during project work. Both tutors shared the view that important *lectures* unfortunately did not take place. Those that proceeded were rated as very good. From their point of view, the BSCW was the central tool for communication between students and academic tutors.

Evaluation of the tasks

Regardless of the specific project, the members of both teams almost unanimously approved of their *tasks* (10 of 13). Especially the chance of learning new software tools (5 of 13) and the high degree of practice-orientation (4 of 13) were seen as positive. Although some students mentioned high pressure of time and workload during the project, they predominantly regarded the *time-frame* as appropriate for accomplishing the task (9 of 13). Six of eight students remarked that their task was not precise from the start and needed to be clarified with the company tutors. While this was known to the students as an integral part of the course as described before, it was nevertheless evaluated as a deficiency.

Cooperation in the project teams

Concerning the actual *group size*, most of the interviewed students took up a neutral position (6 of 11), but stated that a project group should ideally not exceed six to eight group members (7 of 11). The *atmosphere* within the groups was almost unanimously regarded as good (8 of 8), only one person changed his rating in a second interview to neutral. While *cooperation* was considered as mainly high, evidence suggests that it became more difficult over time (see Table 4).

Table 4. Cooperation in the project teams.

1. Survey (beginning of the course):	
Positive	7 of 11
Neutral	1 of 11
Negative	0 of 11
2. Survey (midterm):	
Positive	0 of 11
Neutral	3 of 11
Negative	3 of 11

The programming work was mainly done alone or in pairs, so that weekly team meetings (7 of 11) only took place to coordinate the activities, discuss problems and make decisions (8 of 11). Reported difficulties particularly concerned the problem of social loafing (i.e. letting others work without investing much effort oneself) (6 of 11). When asked about the *experience of team belonging*, the interviewees answered unanimously. Most often, this feeling only occurred at certain events (e.g. preparation of presentations: 3 of 13; presentations, team meetings, client meetings and mini-lab: 2 of 13, respectively). Four of 13 students stated that they did not feel a sense of group belonging during the course. In general, the students had a neutral (4 of 13) or positive opinion (3 of 13) concerning *cultural diversity* in the teams. Participants viewed diversity particularly as an opportunity for cultural exchange and to learn more about different working styles. While 5 of 13 interviewees experienced no difficulties in interacting with intercultural groups, seven participants reported of occasional language barriers. The interviewed students could not agree on the question, if any kind of *hierarchy* existed within the project groups. Four students maintained their team had no leader, while six mentioned that their team included a member who coordinated and distributed the tasks, counselled other members and headed the meetings. Merely two students claimed that a single experienced member decided on matters by himself without involving the group. Concerning the exchange with the other project team, the participants said that they did not have insight into the work of the other team (three of six) or only a very rough impression of their work (two of six).

The company tutors stated that all the participants were *motivated*, but differed with respect to their initiative in the group. Both interviewees perceived a clear *hierarchy* within the groups regarding team leaders and followers. The *cooperation* in the group was generally successful with only minor problems concerning team arrangements in one group. *Cultural diversity* was seen as a source of communication problems. Participants also differed with respect to their technical skills.

Concerning their evaluation of *skills* and *motivation* of the group members, the academic tutors differed clearly. While one of them found that technical skills in his group were very heterogeneous, he stated that all participants were motivated throughout the project. The experiences of the second tutor in his group were directly opposed to this: all team members had very good technical skills, but their motivation was inconsistent. Like the students, the academic tutors related the *feeling of team belonging* in their group to certain special occasions (e.g. a situation where one group refused a proposal made by their informal team leader for the first time). The *cooperation* was high, but one tutor stated that the group divided their work and rarely worked together as a team. While the *cultural diversity* was generally seen as positive with little conflicts between the different members, one tutor observed a separation of sub-groups based on their nationality. From the academic tutor's point of view, the *arrangement of the tasks* ran smoothly, transparently and without significant difficulties. Like the contact persons from the companies, the academic tutors spoke of a clear *hierarchy* with two informal team leaders within the groups, respectively. As major *learning results* in the groups, the tutors mentioned better internal communication and cooperation as well as listening to the customer. In their opinion, the *relationship between the two project teams* was without any sense of competition.

Cooperation between project teams and companies

General cooperation between the project team and the company was experienced as inconsistent with only a few students giving a clear-cut rating (3 of 11: positive; 1 of 11: negative). Specific aspects can be found in Table 5.

The *contact* to the company tutor was rather loose. The majority of students said that they had no contact (2 of 11), rare contact with him (4 of 11) or said they had met him twice or thrice (3 of 11). The *platforms for communicating* with the company tutor were: meetings in the company (5 of 11), meetings at presentations of the group (4 of 11), via mail (4 of 11) and BSCW (2 of 11). While some students responded (especially in the last interview wave) that it was an informal *relationship* (4 of 11), others experienced it as business-like (2 of 11). The students thought that they gained no (4 of 6) or only very limited (2 of 6) *insight into the company's policies or practices*. Reported reasons for this were manifold: no interest on part of the students, no direct contact with the company, the current removal of the company, focus on technical problems and working on the project as homework (1 of 6, respectively). Therefore, it is not surprising that the interviewees named very few things they *learned* communicating with the company: a new software (2 of 5), customer-orientation (2 of 5) and teamwork (1 of 5). Vice versa, they did not think that the company had learned anything (3 of 4) from them.

The company tutors experienced the *contact* with the group as good, open, productive and professional. Both stated that they were referred to particular *contact persons* (two and one, respectively) in the groups. Communication took place after the presentations, in meetings (two meetings in the company and regular meetings, respectively) and via mail. From their point of view, *learning of the group* members concerned technical skills in particular. Insight into the business of the companies was very limited and mainly focussed on the development of the product. Both interviewees did not see any *learning on the part of their companies*. They got to know or already knew XP, but it is currently not applied or will not be used in their companies. Also, they agreed that the students were not *involved in the company*. Both stated that there was a clear division of labour: the company defined the problem (as a customer), the students solved it (as service providers). Both agreed that *contact to the other group and company* only took place during the presentations. Concerning *future contact* with the students, the opinions were inconsistent. One company tutor was interested in further cooperation, the other was not.

The academic tutors stated that from their point of view the students were not *integrated into the daily business* of the companies. Therefore, the group members gained only minor insight into the company's policies and practices, e.g. the communication

Table 5. Cooperation between project teams and companies.

Positive aspects:	
Professional advice	3 of 11
Favourable behaviour of the company tutor	2 of 11
Negative aspects:	
Absence of the company tutor during group presentations	3 of 11
Not reacting to questions from the group	3 of 11
Lack of feedback for the group	2 of 11
Problems because of the relocation of one company	2 of 11

between companies and customers, marketing strategies and the strategic orientation of products. The academic tutors were not convinced that the students were able to *learn* much from their cooperation with the companies. Concerning learning experiences of the companies, one tutor had the impression that the company was interested in XP and further cooperation with the university and the students. The other tutor stated that the company showed little interest in the students' practices of software development. Therefore, he saw no learning experiences on the part of the company. Further cooperation with the university might be possible.

Cooperation between project teams and academic tutors

There were only very few statements concerning the cooperation between the students and the academic tutors. The academic tutors were seen as responsible for organising and planning the course (two of six), as contact persons for the presentations (one of six) and as lecturers (one of six). Their support was evaluated equally often as appropriate or not sufficient (two of six, respectively). The academic tutors perceived cooperation with their project teams as constructive and reserved. The presentations during the reviews were evaluated very positively by one tutor who emphasised that the students were praised and motivated for the achievements by the company. The academic tutors also stated that they had frequent or steady contact (at least during the reviews when the groups presented their preliminary results) with the other project group. Both academic tutors knew the companies well and stayed in contact with both of them during the course.

Discussion

In this study, the implementation of a teaching concept for project work was analysed. We particularly wanted to investigate to what extent the establishment of communities of practice and social networks had been successful and which factors had influenced its success or failure. The results are manifold, but lead to some general conclusions concerning the effectiveness of the course with regard to its different goals.

Communities of practice between students, industry and university

The establishment of communities of practice between employees of a company and university students was not successful. This can be gathered from the statements of students, company representatives and academic tutors. The results offer some explanations for this conclusion. First of all, the relationship between the companies and the course participants was one of customers and service providers. The company tutors were not involved in the actual development of the product but explained their requirements, so that the students were able to work on their own. Also, their engagement in the course was reduced to a minimum necessary (i.e. not attending all the lectures, absence at presentations and regular but rather loose contact with the group). The students on the other hand were not involved in the business policies and practices of the company. A possible reason for this can be found in the interests of the different stakeholders. The main focus of the companies was on the development of a product, while the students were primarily interested in passing a mandatory course. On behalf of the academic tutors, the development of a product was clearly the most important incentive for attracting companies to participate and engage in the course. Therefore,

they organised reviews during the course of the project that were focussed on the advances in developing a product which also was a necessary condition for the students to pass the course. Thus, the academic tutors found themselves in an ambivalent role. On the one hand, they were moderators in building communities of practice; on the other hand, they reviewed and evaluated the performances of the project groups. The results suggest that the students regarded the academic tutors as 'normal' instructors who organised the course and evaluated their outcomes. The moderator role was not mentioned by the interviewees. As the ratings on the collaboration between students and academic tutors were rather mixed, it seems unlikely that a community of practice was established in this relationship. Hence, as the implementation of a community of practice does not mean learning about practice (what the students definitely did), but becoming a practitioner (i.e. part of a professional culture; cf. Solomon 2007), this goal clearly was not reached. Following the arguments above, this result seems to be the logical consequence of the stakeholders' interests and the resulting roles and structure of the course.

Communities of practice within the project groups

The cooperation between the students was rated as positive. However, the results refer to some difficult points as well. First, the evaluation of the intercultural diversity of the groups was rather positive with merely the problem of language being addressed by the different stakeholders. Thus, this seems only a minor aspect for the establishment of communities of practice. Second, the actual group size was rather large. This led to a high division of labour. Third, the students mainly worked on their own or in pairs of two (partly due to the guidelines of XP). Therefore, it is not surprising that only few of the students reported feelings of team belonging. Single events like the preparation of presentations and team meetings are not representative of the 'typical' working style in the teams. These results shed some doubt on the assumption that communities of practice developed within the project groups. As XP was not used extensively in the groups, it remains unclear if the project groups actually developed common practices. They organised their work and cooperation by building specialised roles for the task. This included creating a team hierarchy with informal leaders (coordinating the activities and communicating towards the company and academic tutors). One consequence of this organisation is the problem of social loafing as mentioned above. Since the students' goal is not to become a member of a community, but to develop a joint product, it is rather rational for the individual to reduce its personal commitment as long as no negative consequences are to be expected. As a result, the groups concentrated on those activities which were necessary to accomplish their goals, just like the company tutors. Therefore, it is not surprising that the two project groups did not interact with each other, not because of competition, but because there was no need to do so. Hence, it remains at least uncertain if a community of practice has been established within the project groups. A long-term evaluation seems necessary to answer the question, if the relationships, built in the course, will turn out to be enduring.

Learning results

As communities of practice could not be established, the preceding conclusions are somewhat disappointing. Nevertheless, the approach chosen also entailed some

advantages and was successful with respect to the other aim of advancing software development practice. The students acquired technical as well as presentational skills and succeeded in the development of a product in a limited period of time, while working in a culturally diverse team. Although team work was evaluated inconsistently, it definitely was a part of the learning experience. Both project groups developed fully functional technical solutions for the complex tasks. Therefore, communication problems obviously were not insurmountable. On the contrary, cultural diversity was mainly seen as an opportunity for cultural exchange and learning of different work styles. Only the goal of giving the students an understanding of entrepreneurship was clearly not reached. Important lectures in the course were missing. The remaining ones obviously were not linked closely enough to the task of the project teams. Concludingly, these results show that the main problems with the course concern the claim of establishing communities of practice between the industry and university. Obviously, this is a difficult aim, and some future refinements in the course structure are necessary to approach it.

Future perspectives

To conclude, the course was accepted by the students and became an integral part of the curricula of diverse master and diploma programmes in computer sciences. Nevertheless, in order to better meet the original expectations, there are several necessary steps to re-design the course as follows:

(1) The roles of the coach and reviewer on the part of the academic tutors should be divided: e.g. one tutor is the reviewer in one and coach (especially for the training of the presentational skills and the reflections on cooperation) in the other group. The former mix of mentoring, coaching and reviewing confused the students and did not prove helpful in building trustful relations. Therefore, the switching of roles should help to establish relations of trust between the group and the mentor who can take care of his or her group, particularly in defending the group in the review process. This would serve as a step to establish communities of practice between the groups and their mentors.
(2) Advanced student tutors should be integrated into the course (at least two per group) as experts for the project work. Since student tutors are conversant with programming and project management techniques necessary for the course, they will form the expert core of a community of practice within the project teams.
(3) The projects should take place in the facilities of participating companies. The project should include students as well as programming experts of the company. Concerning the main goal, as to establish a common practice between academia and companies, it is essential to only cooperate with companies which are able to offer working places.
(4) The task should be less extensive (a ‘smaller’ product with less time-consuming programming) to facilitate the process of developing common practices. It should be discussed with the companies which tasks are relevant and can be conducted together with programming experts of the companies. The long-term perspective (jointly supervised thesis work, research and development contracts in funded projects) should be made clearer to the companies to avoid one-shot relations between the academia and industry.

(5) More coaching sessions should be organised to offer the groups a platform to reflect upon their cooperation. This idea was already established in the last course. The feedback of the students was encouraging to organise more coaching sessions in future courses. After an initial training, these events can be organised by the student tutors or even by the students themselves.
(6) A stricter use of the XP principles (including a rotation of the programming pairs) should be enforced to make sure that common practices are developed. The project management techniques lead to an efficient division of labour and tightly working pairs inside the groups. Rotation principles should help to distribute knowledge and establish apprenticeship learning within the groups.
(7) The lectures should be more directly linked with the steps of the project work. The course begins with a two-day tutorial ('mini-lab') introducing all necessary technologies and methodologies in a very condensed way. During the course, different knowledge and competencies are relevant at different times. Therefore, it should be helpful to introduce the techniques at the time when they are required. More precisely, the lectures should start with writing a business plan and managing a start-up, followed by project management techniques. Before the first review, presentation techniques should be introduced. When the programming work begins, XP techniques can be addressed intensively.

Although some of the course goals were not yet met, the development and implementation of new approaches in academic teaching are necessary to improve computer science education and connect academia and industry more closely.

Note

1. The German Diplom is equivalent to a degree spanning the first and second cycle of study (i.e. undergraduate and graduate level) and includes about five years of study.

Notes on contributors

Jan Schilling is an assistant professor in the Department of Psychology, RWTH Aachen University. His main research interests are organisational learning, organisational cynicism, and leadership with a specific focus on implicit leadership theories, destructive leadership and the attribution of leadership success.

Ralf Klamma is an associate researcher in the Chair of Information Systems, Department of Computer Science, RWTH Aachen University. His research covers information systems theory, the application of information systems in science, engineering and the humanities, communities of practice, social software, technology enhanced learning and entrepreneurship.

References

Appelt, W. 1999. WWW based collaboration with the BSCW system. Paper presented at the proceedings of the 26th annual conference on Current Trends in Theory and Practice of Informatics (SOFSEM '99), November 27–December 3, in Milovy, Czech Republic.

Beck, K. 1999. *Extreme programming explained: Embrace change.* Boston: Addison-Wesley.

Ben-Ari, M. 2004. Situated learning in computer science education. *Computer Science Education* 14, no. 2: 85–100.

Bergin, J., J. Caristi, Y. Dubinsky, O. Hazzan, and L. Williams. 2004. Teaching software development methods: The case of extreme programming. Paper presented at Special

Interest Group on Computer Science Education (SIGCSE '04), March 3–7, in Norfolk, VA, USA.
Bourdieu, P. 1983. Forms of capital. In *Handbook of theory and research for the sociology of education,* ed. J.G. Richardson, 241–58. New York: Greenwood.
Brown, J.S., and P. Duguid. 1991. Organizational learning and communities of practice: Towards a unified view of working, learning, and evaluation. *Organization Science* 2, no. 1: 40–58.
Cohen, D., and L. Prusak. 2001. *In good company: How social capital makes organizations work.* Boston: Harvard Business School.
Collins, A., J.S. Brown, and S.E. Newman. 1989. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In *Knowing, learning, and instruction,* ed. L.B. Resnick and R. Glaser, 453–94. Hillsdale, NY: Lawrence Erlbaum.
Coppit, D. 2006. Implementing large projects in software engineering courses. *Computer Science Education* 16, no. 1: 53–73.
Cousin, G., and F. Deepwall. 2005. Designs for network learning: A communities of practice perspective. *Studies in Higher Education* 30, no. 1: 57–66.
Fincher, S., and M. Petre. 1998. Project-based learning practices in computer science education. *Frontiers in Education Conference* 3: 1185–91.
Gruba, P., and H. Sondergaard. 2001. A constructivist approach to communication skills instruction in computer science. *Computer Science Education* 11, no. 3: 203–19.
Hadjerrouit, S. 2005. Learner-centered web-based instruction in software engineering. *IEEE Transactions on Education* 48, no. 1: 99–104.
Johnson, D.H., and J. Caristi. 2002. Using extreme programming in the software design course. *Computer Science Education* 12, no. 3: 223–34.
Klamma, R., M. Jarke, M. Rohde, and V. Wulf. 2003. New approaches to media-supported project work at the university level. Paper presented at the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT '03), July 9–11, in Athens, Greece.
Klamma, R., M. Rohde, and V. Wulf. 2004. Making sense of communities of practice at the university level: Connecting academia and industries. Paper presented at Multi-Conference Business Information Systems (MKWI '04), March 9–11, in Essen, Germany.
Lave, J., and E. Wenger. 1991. *Situated learning: Legitimate peripheral participation.* Cambridge: Cambridge University Press.
Mandl, H., H. Gruber, and A. Renkl. 1996. Communities of practice towards expertise: Social foundation of university instruction. In *Interactive minds: Life-span perspectives on the social foundation of cognition,* ed. P.B. Baltes and U.M. Staudinger, 324–35. New York: Cambridge University Press.
Nuutila, E., S. Törmä, and L. Malmi. 2005. PBL and computer programming: The seven steps method with adaptations. *Computer Science Education* 15, no. 2: 123–42.
Paavola, S., L. Lipponen, and K. Hakkarainen. 2004. Models of innovative knowledge communities and three metaphors of learning. *Review of Educational Research* 74, no. 4: 557–76.
Renkl, A., H. Mandl, and H. Gruber. 1996. Inert knowledge: Analyses and remedies. *Educational Psychologist* 31, no. 2: 115–21.
Schilling, J. 2006. On the pragmatics of qualitative assessment: Designing the process for content analysis. *European Journal for Psychological Assessment* 22, no. 1: 28–37.
Scriven, M. 1967. The methodology of evaluation. In *Perspectives of curriculum evaluation,* ed. R. Tyler, R. Gagne, and M. Scriven, 39–83. Chicago: Rand McNally.
Solomon, Y. 2007. Not belonging? What makes a functional learner identity in undergraduate mathematics? *Studies in Higher Education* 32, no. 1: 79–96.
Stein, S.J., G. Isaacs, and T. Andrews. 2004. Incorporating authentic learning experiences within a university course. *Studies in Higher Education* 29, no. 2: 239–58.
Stephens, M., and G. Rosenberg. 2003. *Extreme programming refactored: The case against XP.* Berkeley, CA: Apress.
Van Gorp, M.J., and S. Grissom. 2001. An empirical evaluation of using constructive classroom activities to teach introductory programming. *Computer Science Education* 11, no. 3: 247–60.
Wenger, E. 1998. *Communities of practice. learning, meaning, and identity.* Cambridge: Cambridge University Press.