2010 23rd IEEE Conference on Software Engineering Education and Training

Customers' Role in Teaching Distributed Software Development

Ivana Bosnić, Igor Čavrak, Mario Žagar
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia
{ivana.bosnic, igor.cavrak, mario.zagar}@fer.hr

Rikard Land, Ivica Crnković
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
{rikard.land, ivica.crnkovic}@mdh.se

***Abstract*—This paper describes different aspects of teaching distributed software development, regarding the types of project customers: industry and academia. These approaches enable students to be more engaged in real-world situations, by having customers from the industry, local or distributed customers in universities, distributed customers in software engineering contests or being involved in an ongoing project, thus simulating the company merging. The methods we describe are used in a distributed project-oriented course, which is jointly carried out by two universities from Sweden and Croatia. The paper presents our experiences of such projects being done during the course, the differences in each approach, issues observed and ways to solve them, in order to create a more engaging education for better-prepared engineers of tomorrow.**

Keywords-software engineering; global software development; education; industry cooperation; project customers

I. INTRODUCTION

It is not rare to hear the complaints of software engineering companies about the practical knowledge of the students who start working after obtaining their degree. While the students can have a high level of theoretical knowledge, they often lack the practice of solving real-life problems, which are being handled outside of the university [1]. Constant changes in the Software Engineering (SE) field are also an issue for the educators who should define the desired knowledge, as students need to get used to quick changes in required technical skills. As a result of globalization, the companies face new concerns regarding distributed development, collaboration, outsourcing, merging, social and intercultural issues. These topics are rarely considered a part of the educational process, which becomes a problem after the students start working; as such skills are necessary to succeed [2].

An effort to help solving the above-mentioned problems is creating global software development courses, such as [3],[4],[5],[6]. To help better prepare the students of our environments, a distributed, project-based course called Distributed Software Development (DSD) has been created, which started in 2003 and has been given continuously for 7 years already. It is jointly led by Mälardalen University (MDH), Västerås, Sweden and University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Croatia, with students from both sides attending the same joint lectures and, more important, working together in distributed teams [7],[8].

Over the years, various project settings have been used. This paper describes three particular types of project settings we have recently tried, where the role of the customer is played not by the teachers and supervisors, but by external people. The reason to try new project settings in the course is to provide the students with as varying, challenging and motivating project settings and tasks as possible. In general, it is our impression that motivation is one of the most important factors for successful projects. Hopefully, our experiences from these settings would be useful for others developing similar courses.

First, an overview of DSD course is described in chapter 2, where also a general role playing model of teaching staff as customers and supervisors is explained, along with a difference of local and remote teaching staff approach. Three different new approaches to projects are described in chapters 3, 4 and 5: Customers in companies, contests and company mergers. Chapter 6 brings a set of questions and experiences observed using these approaches, regarding the project and types of customers.

II. DSD COURSE

A. DSD course introduction

Distributed Software Development course is an elective course for students of Computing and Software Engineering Masters programs. The main goal of the course is to prepare students to work distributed on software engineering projects, covering all the SE project phases – defining the problem, gathering the requirements, creating the project plan, dividing the roles in the team, developing, testing, and documenting. Through all of these project stages, they have to present their project status to all sides to assure the customers of the project's final success. All these phases, more-or-less usual in project-oriented SE courses, are being augmented by the fact that everything needs to be done in a distributed manner, with teams and teachers at both universities. Both the theoretical part (the lectures) and practical part (the project work) of the course are conducted remotely, using communication technologies. In such environment, students face challenges which they would not notice in a local setting, such as:

- Educational background differences – the involved universities emphasize different elements of software engineering and computing education, which leads to unequal knowledge levels;

This work is supported by the Croatian Ministry of Science, Education and Sport, under research project "Software Engineering in Ubiquitous Computing"

1093-0175/10 $26.00 © 2010 IEEE
DOI 10.1109/CSEET.2010.27

IEEE computer society

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

- Communication problems – developing the project and dividing the work among people whom they never met in "real life", using only e-mail, instant messengers and audio/video conferences is much more difficult. A good project definition, specification and interfaces between the system parts are crucial for project success [1];
- Social and intercultural issues – differences in background, views on work, life and communication habits are even more augmented when dealing with different nations. Besides Croatian and Swedish students, at MDH there are many foreign students, mostly from Europe and Asia (Iran, India, Pakistan, China...) Sometimes, mostly on Croatian side, students are faced with different cultures for the first time, and they may have a hard time accepting behavior not similar to theirs, which can lead to lack of team-spirit and trust;
- Language barriers – the majority of the students do not speak English as a native language. As English is the official course language, and project work requires a large amount of communication, various problems can arise, from misunderstandings to mild exclusion of team members (or the whole remote team) who can not cope well with the foreign language communication.

Due to differences in the structure of academic years in Sweden and Croatia, the course duration is rather short, from the beginning of November to the end of January. Therefore a lot of students' focus on their tasks is required. The course starts with the lectures of professors from both sides on global SE themes. In some weeks there are guest lecturers from the industry, who explain their experience with distributed work and pinpoint the main obstacles in such collaboration.

The main part of the course is project work, where students are divided into groups of 6-8 members (3-4 from each side). The projects proposed to them are described briefly and the technologies to be used are proposed. Each project has a supervisor (a member of teaching staff) and customers (usually other teaching staff members). Usually at the local side of the supervisor, one of the team members is assigned a role of Project Leader, who is in charge of communicating with customers, organizing and leading the project work. He/she must also keep close relations with the remote side and their Team Leader, who is in charge of the remote team [9].

Besides various technological skills needed to finish the project, students need to very quickly learn how to assess their and other's skills, adapt to a new environment, divide the jobs, establish trust with non-familiar students at the other side, communicate effectively, solve problems which may arise in the group and practice their social skills. A part of the project work is presenting the project status; they need to practice their presentational skills, in foreign language and in a distributed, international setting.

The project topics are usually defined by the teaching staff, to fit the area of interest of the faculty research groups or to develop software systems to help DSD course. However, to better answer the current needs for educating the future software engineers, we are examining new types of projects and DSD scenarios. In those projects, the main emphasis is moving from the teaching staff as customers to involving real customers from SE industry, other interested parties, or other environments; to transform the simulation of a possible event to a real situation, very much needed in SE. As we are making a step forward from the safe university setting to the "outer world" we still need to take care of the students' abilities and motivation, and never forget that they are still educating themselves, so the feeling of failure could be counterproductive. Therefore we have to balance between letting the students try and learn from their own mistakes, and successfully cooperating with the external customers, who need the final product while they are less interested in education part.

B. Roles of the teaching staff

One of the DSD course specifics, as opposed to classical teaching courses, is role-playing of both teaching staff and students. The teaching staff is trying to move away from teacher-student relation, to become project supervisors and customers, which relates to real life of a company as persons who are responsible for the project outcome (above the Project Leader) and persons who actually buy the product.

Each DSD project gets one supervisor, most often a teaching assistant, who keeps an eye on the project status, meets with students regularly, gives technical advices (although not to a big extent), takes care of the social or intercultural issues which can arise out of distributed work and tries to be a close help to the team if anything goes wrong, either from technical or from collaborative point of view.

Besides supervisor, in DSD course there are also the project customers. In the real-life projects, customers often do not have enough specific knowledge nor will to be very concrete in the requirements, do not know what exactly they want, but are very interested in deadlines and ways of paying for the job done. The customers also prefer a good marketing of the good product, to present them the fully-finished project, with answers to questions "Why is it good for me? What can I do with that?" along with documentation, examples, support etc. DSD customers are very similar: they have a broad idea what do they need, they have some non-avoidable requirements, but they leave the rest to the project team, to talk and negotiate with the customers, make compromises, wrap it in a kind of a "use case story". They are more concerned with practical questions about the product, and less about the technical background. This role is performed by other teaching assistants and professors of the course, but due to the short course duration, the project supervisor is also involved as a customer, for example to discuss requirements or practical usability issues of the product.

Although this is an elaborated role system, it should never be forgotten that DSD is a part of education process [12]. While we pretend to be customers and supervisors, we still have to carry out our teaching roles, assess the project status from the "student" perspective, watch for signs of students' concerns or discouragement which sometimes occur out of

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

overwhelming teamwork situations, and follow the process of each student's personal advancement which can be different from the project group evolvement. This part is generally not present (or strong) in real-life situations or the border of roles is clearer, but in DSD projects there is a duality.

C. Basic approach: internal university customers

Playing a customer role in a university setting is not an easy task. Taking into account all the reasons mentioned above, the customer role can not be carried out to its full extent. The university staff is used to define broader and more general requirements, with different project aspects left for students to decide. The requirements can be more easily changed, as the project purpose is usually a proof-of-concept on some research implementation or an attractive project concerning usage of new technologies. Although the customers and supervisors should be different teaching staff individuals, this model is in practice usually transformed to a situation where professors and teaching assistants – namely customers and supervisors – together play both roles. This affects the objectivity of role playing, as teaching staff at the same time must balance between supervising the team, giving advices and helping with the problems – which would be the roles of a supervisor – and defining the requirements, asking for the features, checking the system usefulness, and finally paying for the work (in this situation with grades only) – which would be the roles of a customer. In the beginning of the course, students can also find this somehow confusing. They are not sure how to approach the staff, but during the project they find it much easier to solve difficulties and make changes in project plan due to problems which appear.

The teaching staff who proposed the project resides on the main project site. This university becomes the project *home*, and in most cases a Project Leader is on the main site, also. This enables the project team, mostly the Project Leader, to efficiently communicate with the customer, especially during the project beginning. This is important as first face-to-face meetings with a customer in most cases give better results and show easier understanding.

For other team members, the meetings with the customers can be either online or live. Both have advantages and disadvantages: online meetings offer the students from the remote side a better insight in the big picture of the project, which can keep them more motivated [11]. Such meetings, on the other hand, can often be perceived as indirect and do not provide an adequate level of human interaction between sides. Local meetings are usually more effective, as the proximity of participants offers easier discussing about the details. Unfortunately, local live meetings can introduce inequality between the local and remote team, while providing the global information to local students only, which puts them in a higher position in the overall team. Therefore, it is essential to ensure a good information flow between the project sites [1].

Due to imbalance in number of students on each site, in some years there are project teams who do not have a remote site, but all the team members come from the same university. To ensure a simulation of distributed development, this team is assigned a remote supervisor and customer, on another university.

This poses a new set of issues for students to solve: although they are all on the site, they have to gather requirements and deliver a project on a distance, so a lot of communication between the students and the customers is needed, especially in the first weeks. The problems of cultural differences and not knowing the supervisor can be crucial, as students can easily feel discouraged or isolated. All sides involved, teaching staff and students, need to maintain close contact and communicate very often. Even then, students do not feel like they know their customer, which is an interesting point: distributed teams, where the students on both locations spend a lot of time informally talking and building team spirit, show a deeper involvement in the project and form a better relation with the supervisor. To solve this difficulty, it would be very useful for a remote supervisor to spend a few days at the remote, main site, with his new project team. This would not only offer them a better global image of the project proposal and requirements, but also provide a way to create a more personal relation, which would help the project progress.

D. New approaches

In the paper, we propose three approaches of enhancing the project work and making it more realistic than before, with different customer scenarios:

- SE companies – the customer involved is a company representative, who needs an SE product and is willing to spend time with the students discussing the project proposal and status, giving advices and leading them to a final product;
- SE contests – various contests exist which motivate students for developing software. Some of them can be used in DSD course, for instance a contest which provides the project themes and customers (project proponents) who are interested in supervising the students teams during the contest;
- SE company mergers – acquisitions, mergers, outsourcing and subcontracting occur in SE companies very often. Dealing with teamwork issues which arise in newly combined teams, simulating such a situation in a course can be a useful students' experience.

The first two approaches are successfully conducted in DSD course, in 2008. The third approach is currently being tested for the first time, in 2009. The description of scenarios, solved issues and the points to be addressed further are given in the following chapters of the article.

Our goal for the DSD course is not to restrict ourselves to one approach which we find the most appropriate for the course, but to combine different customer scenarios during the same year, for different projects. In that way, the students are offered a broader insight in these highly-probable events in their future careers. Even if they can not participate in all the scenarios, by listening to other teams' presentations and sharing experiences, they can also profit and gain new knowledge.

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

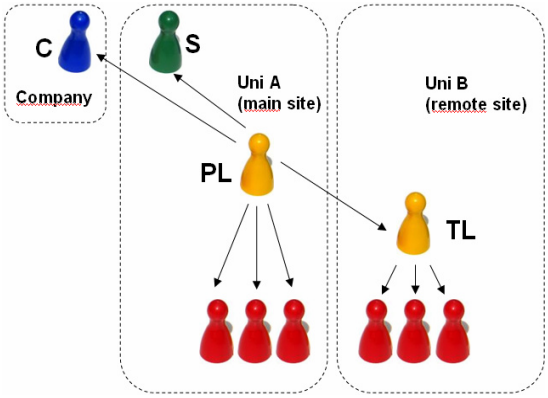

Figure 1. Company customer structure. C = Customer, S = Supervisor, PL = Project Leader, TL = Team Leader

III. APPROACH 1: SE COMPANIES

The first scenario to be described is a distributed students' project done with the company as an external customer. In this case, the teaching staff from both universities keeps the role of project supervisors, but the company representative has both the role of the customer and additional supervisor. An organization schema is shown in Figure 1, with only the main supervisors and customers for better clarity. The project goal is solving a real problem or a part of the problem, in which the company is interested, with a "proof of concept". The students are provided with a Requirements Document, which is more specific and detailed than in "usual" DSD course project. The company should provide students with a constructive support, ranging from providing the documentation important for the project, holding regular meetings, giving technical and organizational advices to enabling them to see the way work is being done in the company. During the project, the company should also provide students with Acceptance Test Plan, to test the features of the product. The customers should, besides on-site working and meetings, also be involved in educational activities of the course, such as attending the joint presentations given by the students and providing their comments on the students' work, during and after the course. This will give the teaching staff a better insight in the work process and help with the objective students' evaluation at the end of the semester. The requirements on educational activities pose a problem due to busyness and priorities of company employees involved, which are different than those of teaching staff.

In our pilot-project involving a real company customer for the first time, we were working with one company; a small Croatian enterprise oriented on telecom billing and fraud detection systems. This company consists mostly of engineers educated at FER, thus familiar with the level of education and experience of our students. Also, it is a representative of a small business partner in one Tempus project, which is aimed at providing better cooperation between Croatian universities and the industry, from education to research level.

Some of the issues which were expected beforehand or occurred during the project are described below.

A. *Project type*

It is very important to select a proper project type for the course. As stated before, the goal of the course is not to make "a perfect" product, at all costs, but to focus on the process of creating a product and project work, which should be of a good quality, to result in a good final product. The reader should have in mind all the difficulties which occur in a distributed, multi-cultural, multi-educational setting, which should be less intense, or even non-existent, when working in a usual, local environment. That being said, it is vital that the project is a feasible one, without very complicated components, such as deriving special algorithms or developing a deep theory in order to make things work. As for motivation, it would be good if the project task being offered is interesting and "intriguing" in a way, but at most times it is hard to find a theme so motivating in the business-oriented sector, which would be of a proper size for this course.

While proposing the projects from the industry side, it is clear that these goals should not be "business-critical", as there is no guarantee that the final product will be equal or similar to the expected one. The project's failure could also sometimes lead students to important experiences and conclusions, which proves to be important for their future careers. If that happens, and the "moral of the story" has been learnt, the course objectives are still fulfilled, even if the product is not functional or finished. So, a company should be prepared to take a risk, while providing the students with peripheral tasks, extensions or improvements, excluding the work on core components. Decomposing the work in smaller parts (such as GUI, business logic, different tiers of an application etc) enables the students to divide the components to work on between the working teams in Croatia and Sweden, define clear interfaces and reduce the information overflow in communication between the teams.

The project done at DSD course was developing a tool to help converting the billing data for mobile telephony systems. Specifically, it was a parser generator from a formal grammar called ASN1 to CSV format. As the companies dealing with billing systems usually have very big amounts of data to process, one of the most important requirements was the speed of converting. Therefore, the tool basis had to be written in ANSI C, and it was the only requirement on programming languages. That fit well into the course, as although the different universities from which the students originate offer programming courses based on different programming languages, C language is still widely taught and used for explaining basic programming techniques. The customer had no particular requirements on other technologies used in the system.

B. *Communication issues*

Besides communication problems observed in the past years of this course, a new set of issues has arisen with the company involvement. As the company is not primarily focused on the students, but is occupied with their everyday work, a different communication rules should be provided. In this project, a typical information *push-pull* issue occurred. Being proactive and pushing the project information to the customers seems to be better solution than waiting for the

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

customer to react or ask for information. Regular Week reports, which should be written by every project team, now get a very practical value – how to state in a few sentences what is the *real* project status? Also, to prevent the one-way communication which can sometimes be present in students-supervisor relations, besides reports which usually do not provide questions to be answered, regular meetings should be organized.

In our project, meetings were held usually once in one or two weeks. The meetings were local only, which made the remote team very dependant on the local members, which resulted in a kind of isolation feeling. Although it is good to have "one point of entry", usually a Project Leader who is at the main project site, the steps will be taken next year to ensure online meetings at least in the beginning of the project. Concerning the push/pull of information, an interesting issue occurred: although the team was regularly publishing the reports on the project Web site, they didn't find it important enough to "push" the data to the customers or invite them to take a look at the project Web page. Both sides were presuming conclusions based on their current work: students presumed that the customers are used to visit the project page just as teaching staff does, while the customers presumed that they would be informed of new events and documents by e-mail or other communication channels.

Experience of a project with a company showed this duality as one of the issues. While the university staff is mostly concerned about the process of getting work done, the final students' experiences and lessons learnt, it is understandable that the customer is more interested in the final product which should be functional enough to be useful. The project supervisor was still at the university, taking care of the educational part of the team work, but was not too well informed about the technical requirements or detailed specifications, which were discussed directly with the customer, so it was hard for him to be fully involved. On the other hand, the customer was not, although having a good will, thoroughly informed about the "social" and educational part of the teamwork, nor he was included in the final grading. While a part of this issue could be solved with more intense communication between real customer and project supervisor at the university, it remains open how to reconcile different goals of academia and company world.

C. *Legal issues*

Ownership and copyright issues are becoming more complicated as more parties are involved. It should be stated that the practice of University of Zagreb is that the Faculty, not the author – student – owns the projects done as a part of education. Such attitude can cause big problems in trying to connect students' projects with the industry, as it is not possible to pay the students for any work done in the course, and companies do not have too much interest in supervising the project which will be owned by FER. Legal issues on continuing the cooperation with the company and commercializing the project also exist. The rules on these topics are currently not completely defined in this environment, so this is a kind of "gray zone". A step to the final solution, which should enable easier cooperation, could be licensing software products with open source license similar to BSD licenses. This group of "non-viral" open source licenses does not force users to apply the same license to derivative works. As the products developed in DSD do not contain big scientific achievements, the companies could agree more easily to develop and use open source tools, as the company in our pilot-project pointed out.

D. *Company interest*

An important topic in cooperation with the companies is their benefit in DSD participation. Surely, a lot of energy and time is invested in proposing the project, supervising the team and evaluating the project. Students also need more than a minimal amount of attention to make a good product. Nevertheless, the benefit exists, especially when looking more far-sighted.

The company involved in DSD course has direct contact to the students of their team, and indirect contact to the students of other project teams. As they are the students of final years of study, who were prepared to take such an intensive elective course and who gathered a lot of real-work experience during the course, enhanced by familiarity with particular concepts and requirements, this company-students relation can easily become an opportunity for employment. An obstacle can be the fact that most students of final years already work somewhere, which reduces their interest for new employers.

Current situation in Croatia shows the serious lack of computer science engineers on the labor market, with no significant signs of improvement. Closeness to the university and better visibility there can enable a company to market themselves to the students in a more concrete way. Such connections could also lead to better cooperation between research groups and the industry, which is currently of a low intensity, as the companies' goals are mostly not focused on supporting or implementing the research done at the university. In these topics, much could be learnt from North- and West-European universities, while having in mind other factors such as country's economy or tradition impact.

IV. APPROACH 2: SE CONTESTS

To enhance the students' interest in software engineering, various project competitions around the world exist, with different approaches to the students, individuals or groups. Some of them are *Google Summer of Code*, where students help to improve one of the popular open source products; *Microsoft Imagine Cup*, where students design their own product which fits in a very broad theme selected every year; *SCORE Software Engineering Contest* where students solve one of the proposed problems, while having to deal with an external customer, who proposed the project.

The second scenario described in this article integrates the contest in the DSD course, involving the student teams in such software engineering competitions, while retaining the similar structure of the course team projects. In such scenario, depending on the contest, students can have external supervisors, work on their own or proposed projects, and gain better motivation to succeed in their project due to the nature of the contest.

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

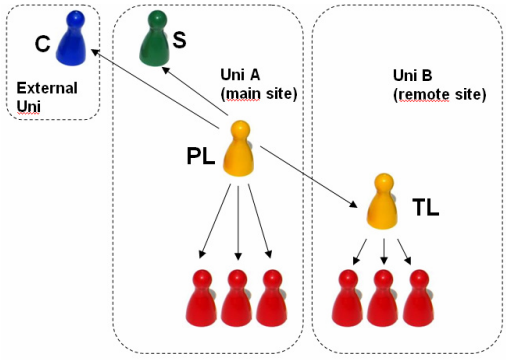

Figure 2. SE contests structure

In DSD course, we have integrated the SCORE software engineering contest. This contest, which was held for the first time, is a part of *International Conference on Software Engineering (ICSE)*. The main contest idea is to propose a set of projects, with very different themes and concepts, ranging from purely software to a mixture of software and hardware implementations, generally without prescribing the exact technologies which need to be used, in order to accommodate a bigger audience. Each project has a description, a set of initial requirements and the external customer, who proposed the topic. The contest period lasts for more than a year, which ensures that the contest can be integrated into courses held in different parts of an academic year.

Four of the DSD projects teams participated in three SCORE projects. The teams consisted of 6-7 members, which was a bit more than advised number of persons for SCORE (5 members), but was supplemented with shorter time for developing than in SCORE (two months effectively). Two teams were distributed, with their university supervisors and Project Leader on the same side, the third team was local in Sweden with their main supervisor in Sweden also. The fourth team was localized in Sweden, but their main supervisor was in Croatia. This approach to the course is the only one where customer and the main supervisor are not on the same site. The organization schema for the approach with SE contests is shown in Figure 2.

Each project was given an external customer, provided by SCORE, and it was expected that students communicate with them in topics such as gathering requirements, solving main technical problems, reporting the project status, etc. Of course, all the other elements of "local" course had to be fulfilled, as regular reporting to the teaching stuff, presenting the project status and so on. The preferred method of communication was e-mail, due to different time zones of the external customers and amount of involvement needed from the customers. It was interesting to observe the different level of engagement of the customers in different projects, which were ranging from offering lots of technical advices, regular discussing and working with the group, to referring the students to find the real group of users in order to elicit the product requirements. This was an opportunity for the students to observe different kinds of customers which can be found in real life projects, and practice working with them. Also, the students managed to create a relationship with the completely unknown person on the other side of the world, which in the end they did meet in person[1], adapting to the communication and exchange of ideas while building trust, in quite short course duration.

Having to deal with external customers and requirements, the students had to be prepared for unpredictable situations, For instance, one of the projects was supposed to be done with real-world stakeholders, who would be interested to use this application. The students had to find them, interview them, discuss the requirements, invite them to participate in the project testing and evaluate the project, etc. In those teams, one of the members had a role of a "public-relations" member. He would contact students at the University interested to help, particular groups of people such as people with disabilities, regular public transport commuters, bicyclists; he would create offline or online questionnaires and discuss stakeholders' needs and proposals.

In the end of the course, together with duties which needed to be performed for the course (final presentation, different reports and documents), SCORE groups had to submit a final report to the SCORE, which was a 20-page document describing the process of creating a software product, methodologies used, issues and problems recognized and dealt with, requirements and technical overview of the system. This report was partially similar to our course project reporting. Still, that was another opportunity to learn how to create official reports, summarize key points and make a critical view of their work, but also to show the good points of their solution and advertise their product.

V. APPROACH 3: SE COMPANY ACQUISITIONS AND MERGERS

A practice often seen in this time of globalization is buying, selling and merging of the companies, as well as companies cooperating or forming business relationships like partnership, outsourcing and subcontracting. All of this involves an element of distributed development which brings issues like:

- the team which is originally developing the product bears more project insight in the beginning
- the documentation amount and quality can be inadequate for another team to grasp the knowledge

This year, we are cooperating with a third university, Universität Paderborn, Germany, in our DSD course, in company merger scenario. Students there are involved in their local software engineering, project-oriented course, which is showing some similarities to DSD, but differences like duration of the course (1 full academic year) currently prevent full integration with DSD. Therefore, a scenario of involvement is created, where some of the Croatian and Swedish students would join the ongoing project in Germany. A part of the

[1] Among some 50 teams registered for the SCORE contest with final in 2009, all of the four projects from our course were accepted into a second round of evaluation of 10 teams. Out of the selected 6 finalist teams, which were invited to ICSE conference, 3 were from the DSD course, and the overall winner was the DSD course project *BTWmaps: if you go, my advice to you* [10].

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

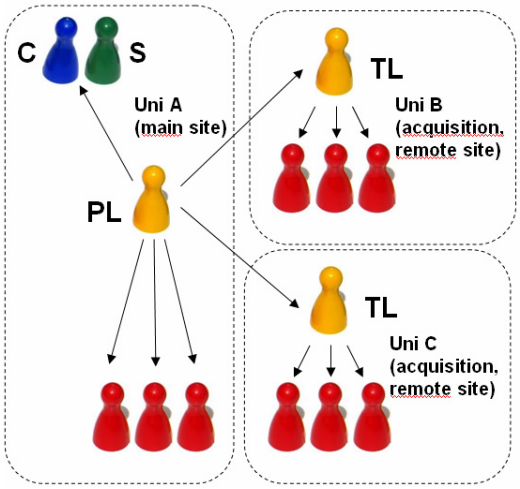

Figure 3. Structure resembling company acquisition and subcontracting; note that at University A, the stakeholders have had a long-term relationship while Universities B and C are newcomers.

German team has started working distributed on a non-crucial, add-on part of their 1-year project, which would be integrated in the full solution. This puts our students in the real-world position of a new-coming team, where they would be faced with the abovementioned problems. Project Leader is, as usual, on the main site, with the customers and main supervisor, while each remote site has its own Team Leader. The organization schema is shown in Figure 3.

In the first weeks of the project work, we have observed the following issues:

- the roles of different groups in the whole project appeared unclear (is it outsourcing, merging, developing in a distributed company);
- the project requirements had to be made according to requirements of the local, bigger project, so the remote teams felt isolated in making the first decisions;
- communication, collaboration and role management proves to be harder involving three student sub-teams on a project;

As we develop the DSD course as a "process-based", where the lessons learnt are more important than the final product, special efforts from the teaching staff in all the countries involved are needed to create such a project, form and monitor such teams, in order to give students a positive experience of this kind of work.

VI. CUSTOMER TYPES AND PROJECT CHARACTERISTICS

Our experiences from the differences between the University-defined projects and those defined by industrial customers can be described using various project characteristics. The differences are those of exploratory projects and new development efforts on one hand (which in our case has equaled academia), and evolution of existing software or larger and therefore more planned projects on the other hand (our industrial customers). Exploratory projects, mostly had loosely defined project specification (customers described only a project vision); initial requirement definitions were weak, so there were lots of revisions during the project work; the choice of technology was free whenever it was possible to comply with the project vision (sometimes there were projects where that was not possible due to basic requirements).

On the other hand, the projects with the industrial customers had more detailed initial specification, with more detailed requirements definition, which put more constraints to the choice of technology.

While dealing with the different types of customers, - academia / industry, different project roles - supervisors / customers, and even different geographical location of project roles – co-located / remote, many questions appeared:

- What are the advantages/disadvantages in academia, where the same people play the role of both customer and supervisor?
- What are the characteristics of university customers? What are the overall characteristics of such projects?
- How does the customer location influence the project work (co-located with the main site; no co-location)?
- How does the supervisor role change, when the customer is external? Can a supervisor influence the customer? Can he have influence on students?
- What are the motives of industry customers? What are the motives' implications on the project?

Some of the questions are easily answered with our current 6-year experience gained through this course, and are explained in particular chapters of this paper. But the questions involving more complex approaches, like external customers in industry and the university staff role change in such projects need to be addressed after a more thorough research experience, which would include a higher number of such projects, with subtle variations of roles. We hope to accomplish this future work in this and coming years, by increasing the number of projects developed in this way.

VII. CONCLUSION

Teaching global software development today should be more challenging and motivating for the students, to give them a more practical basis for their future careers. Project-based courses are one of the good ways to lead students to a higher level of involvement. Different additional scenarios, role-playing and external cooperation can prepare students better for coping with similar situations afterwards, as long as the teaching staff keeps in mind their primary goal – education of engineers. In this way, with the demanding work of staff to still maintain a level of students' safe environment, students learn more effectively, show big motivation and evaluate such courses as very useful and successful.

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.

VIII. ACKNOWLEDGMENTS

We would like to thank Prof. Wilhelm Schäfer and his group from Universität Paderborn, Germany, for collaborating with us on this course, and Mr. Branko Beslać from Ring datacom company for participating in the course as an external customer.

REFERENCES

[1] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish and J. Kazmeier, "Global Software Development Handbook", Auerbach Publications, September 2006..

[2] D. Bagert et al., "1999 Guidelines for Software Engineering Education Version 1.0," tech. report CMU/SEI-99-TR-032, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 1999.

[3] N. Mullick, M. Bass, Z. Houda, P. Paulish, and M. Cataldo, "Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure". In *Proceedings of the IEEE international Conference on Global Software Engineering* (October 16 - 19, 2006). ICGSE. IEEE Computer Society, 203-212, Washington, DC, USA, 2006

[4] B. Bruegge, A.H. Dutoit, R. Kobylinski, G. Teubner, "Transatlantic project courses in a university environment," *Asia-Pacific Software Engineering Conference*, pp. 30, Seventh Asia-Pacific Software Engineering Conference (APSEC'00), 2000.

[5] Jesús Favela, Feniosky Peña-Mora, "An Experience in Collaborative Software Engineering Education," *IEEE Software*, vol. 18, no. 2, pp. 47-53, March/April, 2001.

[6] Bertrand Meyer, Marco Piccioni, "The Allure and Risks of a Deployable Software Engineering Project: Experiences with Both Local and Distributed Development," *Software Engineering Education and Training, Conference on*, pp. 3-16, 2008 21st Conference on Software Engineering Education and Training, 2008.

[7] I. Crnković, I. Čavrak, J. Fredriksson, R. Land,; M. Žagar, M. Akerholm, "On the Teaching of Distributed Software Development" *25th International Conference on Information Technology Interfaces, Proceedings of the,* pp 237-342, 2003. Cavtat, Croatia, 2008

[8] M. Žagar, I. Bosnić and M. Orlić, "Enhancing software engineering education: a creative approach." In *Proceedings of the 2008 international Workshop on Software Engineering in East and South Europe* (Leipzig, Germany, May 13 - 13, 2008). SEESE '08. ACM, New York, NY, 51-58.

[9] D. W. Karolak, "Global Software Development: Managing Virtual Teams and Environments (Practitioners)", Wiley-IEEE Computer Society Pr, December 1998.

[10] "BTWmaps: By the way, if you go, my advice to you", DSD SCORE project, ICSE 2009, http://btw.rasip.fer.hr

[11] J. Contreras et al., "Improving Collaborative Learning by Supporting Casual Encounters in Distance Learning," Proc. SITE 2000 Int'l Conf., Association for the Advancement of Computing in Education, Norfolk, Va., 2000, pp. 136–142.

[12] I. Čavrak, R. Land, "Taking Global Software Development from Industry to University and Back Again", *Proceedings of ICSE 2003 International Workshop on Global Software Development* (GSD 2003), Portland, USA

Authorized licensed use limited to: Malardalen University. Downloaded on July 30,2010 at 11:50:41 UTC from IEEE Xplore. Restrictions apply.