TEACHING GLOBAL SOFTWARE ENGINEERING AND INTERNATIONAL PROJECT MANAGEMENT

Experiences and Lessons Learned from Four Academic Projects

Florian Matthes, Christian Neubert, Christopher Schulz
TU München, I19, Boltzmannstr. 3, D-85748 Garching
{*matthes,neubert,schulzc*}*@in.tum.de*

Christian Lescher
TU München, I1, Boltzmannstr. 3, D-85748 Garching
lescher@in.tum.de

José Contreras
Universidad Tecnica Federico Santa Maria, Avenida España 1680 Casillo 110-V Valparaíso, Chile
jose.contreras@usm.cl

Robert Laurini, Beatrice Rumpler
INSA de Lyon, 7 avenue Capelle, F-69621 Villeurbanne
{*robert.laurini,beatrice.rumpler*}*@insa-lyon.fr*

David Sol
Tecnológico de Monterrey, campus Puebla Vía Atlixcayotl 2301, 72800 San Andrés Cholula
dsol@itesm.mx

Kai Warendorf
Hochschule Esslingen University of Applied Sciences, Flandernstr. 101, 73732 Esslingen
kai.warendorf@hs-esslingen.de

Keywords: Global software engineering, software project management, education approach, experience report, international project management

Abstract: As part of the ongoing globalization process, software is no longer developed by a sole enterprise which is based at one single location only. In turn, distributed engineering teams are continuously modifying software by bringing in their local knowledge and country-specific expertise. Due to this cooperation on a global-scale, today's software engineers require distinct skills and capabilities allowing them to face a paradigm called Global Software Engineering (GSE). However, regarding today's universities curricula, the teaching of GSE can be seen as an emerging discipline which is increasingly gaining attention. This paper depicts the progression and lessons learned from four different globally distributed software engineering projects executed by late bachelor and master students from five different universities. In doing so, the article facilitates future GSE endeavors in academia and industry.

1 Introduction and background

Many technological, organizational, and economic factors have led to the increased globalization of development projects (Sangwan et al., 2006). Drivers for *Global Software Engineering* (GSE) include cost competitiveness, access to talent regardless of location, the need for a globalized presence, as well as mergers and acquisitions (Carmel, 1999). Globally-distributed projects are rapidly becoming the norm for large software systems (Herbsleb, 2007). Thereby, GSE imposes new challenges on software engineers, in which coordination over distance (Damian et al., 2010; Herbsleb, 2007; Lescher et al., 2007) can be considered as one of the major ones. In this respect, there is a strong need for educating and training IT employees to act jointly in an international context. Not only those software engineers are geographical

separated from each other by working at multiple locations on interdepending modules, they are also subject to cultural differences including national culture, native language, as well as organizational culture (Carmel, 1999; Hofstede and Hofstede, 2004). In this regard, technical expertise and know-how are essential but the ability to collaborate effectively in globally distributed teams is equally important (Gotel et al., 2009).

When considering the personal requirements today's software engineers are facing in their daily work life, it is surprising to see that teaching GSE at universities is still in its infancy. Besides its recent occurrence and emerging importance in industry, the sparse offering of teaching GSE in academia maybe caused by the additional coordination work hosting institutions have to cope with when organizing such types of courses. Furthermore, different semester schedules combined with an unequal grading and evaluation schemes complicate the cooperation among global distributed universities. In the light of aforementioned situation, five chairs of five different universities speaking three different languages conjointly organized a practical course during the winter semester 2009/10, namely:

- Institute National des Sciences Appliquées de Lyon (INSA) in France
- Tecnológico de Monterrey campus Puebla (TEC) in Mexico
- Technische Universität München (TUM) in Germany
- Universidad Tecnica Federico Santa Maria (USM) in Chile
- University of Applied Sciences Esslingen (UE) in Germany

Coined by the name *Network of Engineering universities Educating in Intercultural Design* (NEREID) in July 2008, the cooperative course aimed at teaching students how to work on software engineering projects given a globally distributed team.

Originally, the NEREID course was launched by Prof. Robert Laurini in the Information Technology department of Institute National des Sciences Appliquées de Lyon (INSA). Considering the typical syllabus of the master computer science graduates of INSA de Lyon six students work conjointly on six projects during a given time frame of one year. In this vein, each student has to take on the role of the project lead, following the thought that an expert in computer science must not only be a good programmer, but moreover has to obtain distinctive skills in project management, especially for the design and coding of huge software products. However, the main limitation is that local students possess the same culture, apply similar methodologies, and speak identical languages. Regarding the future work of those students, they will much likely be in contact with colleagues having different cultures, speaking different languages, as well as mastering different methodologies.

Since INSA de Lyon regards international project experience as a key aspect of each student's curriculum[1], in fall 2007 the idea came up to organize cross-country projects, in which students from different countries would have to work conjointly on one project. Thereupon, Prof. Robert Laurini and his colleagues contacted several international universities in 2008. With regards to participation constraints, the relationships between students within a project team had to be on the same level, i.e. each software engineering task could be assigned to each student. Additionally, the time schedule of both institutions had to be congruent. Although, the contacted institutions considered student exchange as being of the utmost importance, they either denied the proposal or could not meet the criteria. As for the reasons stated, either project management was not their priority, time period and duration were not relevant, or project management was a priority, but organization could not comply.

In July 2008, Prof. Robert Laurini was invited to the TEC were the colleagues of the Mexican chair were immediately enthusiastic about his project ideas. As a fruitful consequence, one student project topic was selected, and two groups of France-Mexican students were organized in order to work together from autumn to winter 2008. The subject consisted in the creation of a small information system for a tourism office based on Google mash-ups. As a follow up of this experimentation, a first post-mortem report was compiled (Laurini and Sol., 2009), analyzing the different aspects encountered during the project execution in an international context. In addition, the two partners jointly decided to extend the NEREID course in terms of academic partners and number of student projects facing the upcoming year 2009.

In this paper, we now document and reflect the experiences when teaching GSE during the winter term 2009/10. In detail, we describe the general approach being taken by the five universities as well as the students including roles and deliverables (Section 2), the

[1] Institute National des Sciences Appliquées de Lyon (INSA) is a very international-oriented institution with more than 25% of its students being from foreign countries. Moreover, 75% of INSA de Lyon students are spending one or two semesters abroad.

final outcome and experience students made within three projects (Section 3), and the experience as well as lessons learned from a teaching staff's perspective (Section 4). In doing so, our results are grounded in the experience we as the teachers made when interacting with the participants as well as in the information we gained when conducting four semi-structured interviews with students involved in projects Technische Universität München (TUM) were tutoring. Having taught GSE to 43 students organized in nine different software teams during a time span of 4 months, our main goal is to inspire the reader how to teach GSE in an academic context. More precisely, we attempt to help teaching staff to plan, implement, and guide software engineering projects realized by globally distributed teams who are compelled to use a communication language other than their mother tongue.

2 An approach to teach global software engineering

This section delineates the approach which was applied when carrying out the NEREID course. It originates from the first cooperation between France and Mexico in 2008 and was refined before the second iteration of the course started in October 2010. Figure 1 illustrates the main phases and their duration by distinguishing between course preparation, execution, and post processing stage. Since the NEREID cooperation involved universities' teaching staff equally to students, a differentiation between both groups is made through the two gray horizontally aligned rounded rectangles. In the following, each phase's main activities are explained.

2.1 Introduce & coordinate

The preliminary phase consisted of two main activities: first of all, the participating universities had to be selected. Afterwards, those members agreed on how to put the NEREID course into practice during a kick-off meeting. Since NEREID initially was launched by Prof. Robert Laurini from INSA de Lyon the acquisition and the selection of the participating universities were managed by him and his colleagues. In June 2009, he announced that the members for the winter semester 2009/10 consisted of the five institutions (cf. Section 1). After the participants had been identified, a kick-off meeting was summoned. Date, means of communication, and the agenda of this initial meeting were mainly arranged through e-mail exchange. As communication device, an IP-videotelephony system was used being available at all universities. The two-hour meeting was held in English, the professors and all respective research assistants were participating. Main purpose was to get to know all members and to discuss the boundary conditions for the second iteration of NEREID course. In addition, mutual trust and rapport was build on a teaching staff level.

The overall course goal was to allow student teams to work conjointly on distinct software engineering projects in order to improve their project management and communication skills taking differences in geographical location, academic curriculum, and culture into account. Less focus was put on technical experience and tooling skills students would gain through the implementation of the assigned projects. All academic institutions agreed that participating students would start without any introductory lectures. Hence, no specific project management and cooperation methods or models were taught in beforehand.

It was up to each single member to fit the course in the individual degree program and grading scheme. For instance, each institution independently determined the type of the course (e.g. seminar, internship), the credit points which could be achieved (e.g. four credit points regarding the European Credit Transfer System ECTS), as well as the experience and knowledge representing prerequisites for a participation in the course (e.g. attended software engineering lecture, language classes, project management).

All members settled on that the student teams had to consist of 4-6 students who were enrolled in computer science and business informatics at their according university. The set of team participants had to originate from 2-3 nations differing in their first languages which made communication in a foreign language inevitable. In terms of workload, ten hours per week were estimated by the members, hence 100-120 hours would be accumulated by each student taking three month project time as a basis. The universities agreed that each individual project topic was addressed by one student team only.

Due to its world-wide acceptance in terms of written and spoken communication, English was chosen as the official project language, for both the organizing members as well as the participating students. Since none of the members came from an Anglophobe country, each participant was equally challenged in speaking a foreign language. Regarding communication devices, no explicit directives were stated in advance. This allowed the teaching staff and

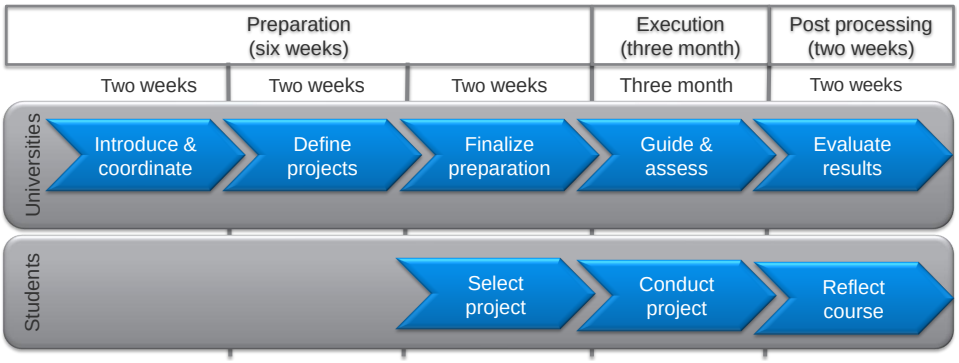

Figure 1: NEREID course execution stages

students to take advantage of a broad lineup of communication devices ranging from simple e-mail and phone calls to more sophisticated video and web conference solutions.

Regarding the different project roles following distinctions were made [2]:

Project sponsor: Initiated the project by proposing a specific topic one student team would work on. During the execution, the sponsor played the role of a local or remote business customer, hence formulated functional and non-functional requirements considering the software engineering product. He or she was also in charge of validating the final outcome from a customer perspective. In most cases the role of the sponsor was filled by a professor of the participating academic institutions. If local students were involved in the fulfillment of the project, the sponsor could simultaneously act as the supervising tutor.

Supervising tutor: Once students enrolled in a software engineering project, they were automatically assigned to at least one local supervising tutor who was appointed by each university in beforehand. The tutor's role was to assist the students not only for technical issues with regards to the results and the communication means, but also help them to overcome organizational obstacles. Even if the local supervising tutor often did not know the desired project outcome in detail, he or she was able to provide helpful support considering general project management techniques as well as the concerned deliverables. In addition to escalate objections raised by the local students to other tutors or to the project sponsor, the local supervising tutor was also responsible of evaluating the students he was looking after.

Student project lead: Directly at the start of each project, all student teams had to appoint one project lead who was responsible for classical project management tasks. In particular, he or she organized the internal team and external presentation meetings, assigned different project activities to the team members, and kept track of the overall project goals achievement. Furthermore, the student project lead represented the single point of contact to the project sponsor by being in regular discussion with regards to the system's functional and non-functional requirements. Steering the team through the course of the project in addition to interacting with the customer, made the lead a pivotal but also a tough role.

Student project member: As mentioned, every student team consisted of 4-6 students who conjointly worked on the software project. The individual student role within this team was determined internally by each group regardless of the project sponsor and supervising tutor. The spectrum of duties ranged from quality, design, testing, communication, etc. tasks and was mostly contingent on the experience and know-how the respective student could contribute. Each student was assigned to a local supervising tutor who could aid with the project work and approved the student's overall performance expressed in participation and deliverables.

Another item discussed by the members consisted in a common web page (NEREID-Page,) enabled by a Wiki (infoAsset,) in order to generate a form of corporate feeling. Besides a presentation of all participating universities, this site also contained a short description of each student project. For further information considering a specific project, every member was in charge of providing details on their local web site. It was also intended that the common Wiki would serve as registration platform where students could express their interest in a certain project while sharing

[2]The term *local* refers to roles whose assigned persons are working at the same place, hence may have physical contact to each other. For instance, for a student the local supervising tutor represents the tutor who is available onsite, thus at the same institution. In the contrary, *remote* signifies that the person playing a specific role can be only communicated to by using telecommunication equipment.

their main contact information. In this vein, the Wiki would facilitate composition and initial contact of the team in advance. Finally, all members were asked to contribute at least one relevant software engineering project during a time frame of two weeks. After this phase, a synchronization meeting was scheduled by applying IP-videotelephony again.

Regarding the output, the members decided on subsequent software development artifacts every student team had to provide during the individual project. Each deliverable had to be written and presented in English language, all students were urged to contribute equally to the final outcome. The results were submitted to the local tutor, who either reviewed them in case of a document or attended them as for oral presentations. With respect to the deliverables, all student teams had to provide the following items:

Initial presentation: During a period of 10 minutes, each team presented the general project content in addition to a project plan and anticipated risks. Afterwards, a 5 minutes question & answer section was held allowing the audience and speakers to discuss.

Final presentation: Students had 10 minutes to present their project as well as the lessons learned. 5 minutes were reserved for a short live demonstration of the main results complemented by another discussion round.

Final report: In the end, each student team was obliged to compile a report composed of approximately 10-15 pages. The document summarized their problem and context, the approach which was taken, lessons learned, as well as possible future work. On the one hand, the report served as a specification by delineating the solution which was implemented by the students. On the other hand, it also contained personal experience and suggestions for developing software in international teams.

In terms of the initial and final presentation, local talks only including the students of the tutoring university were preferred by the supervising tutors as well as the respective project team members. This was mainly caused by organizational and technical issues, i.e. time differences coupled with problems considering video communication making it difficult to gather all stakeholders working on the specific project.

2.2 Define projects

Main activity in this phase was to propose and to decide on the different project topics based on the agreements and general project conditions elaborated during the kick-off meeting. Thereby, all topics were restricted to software engineering tasks related to the current research focus of the universities by targeting at late bachelor and master students. The complexity of the projects where adequately adjusted, bearing in mind the time constraint of three months all student teams were exposed to. In case of the specific work packages, all proposals consisted of the implementation or enhancement of a software system, however there was no overlapping in terms of specific project content. The respective sponsor composed a general project description in addition to a requirement specification containing *must* as well as *nice-to-have* requirements serving as a foundation for the teams. Concluding, the project descriptions were published in the Wiki and a second synchronization meeting among the universities was scheduled.

2.3 Finalize preparation

Main goal of the synchronization meeting as part of the finalization phase was the presentation of the final project topics in addition to their start and end dates. In mid-September 2009, a IP-videotelephony system was set up as means of communication allowing all universities to participate. Due to the different semester schedules, the members decided to start the projects independently from each other. Hence, once enough students were registered for a specific project matching the criteria agreed upon in the preliminary phase, the team could get down to business. For the registration process, students were asked to make use of the project specific Wiki page. Furthermore, e-mails between the teaching staff were exchanged since not all universities took advantage of the Wiki.

In addition to above mentioned activities, preliminary discussions were hosted at the particular universities addressing students who were interested in the NEREID project. The 30 minutes meeting took place at the beginning of the semester and provided a quick rundown of the course. Besides the general course of action and required deliverables, the different software engineering projects were presented. After the audience got familiar with the multiple topics and related questions were answered on the part of the teachers, a first allocation of students to the projects was carried out. In this process, students had the freedom to select the topic of their choice.

2.4 Guide & assess

When the preparation phase was finished at the end of September 2009, universities could pass over to supervise the different projects which were now executed by the student teams. Since students did not attend any preparative lecture sessions in advance, strong support by the supervising tutors as well as a close cooperation with the project sponsor were indispensable throughout the project's implementation phase. As depicted in Figure 2, universities undertook the execution of three parallel activities all serving to guide and assess the different student teams.

An initial kick-off meeting was held by the project sponsor which was destined for all student team members and targeted at building trust as well as rapport among the participants. During this meeting, the sponsor presented a more detailed project description including the overall context, a technical description (e.g. specification, architecture, technology) in addition to further organizational information (e.g. links to the website, e-mail addresses). The main activities of this phase were instantiated in the following manner:

Evaluate results: Each software engineering artifact delivered by a student team was assessed by taking advantage of a simple excel spreadsheet. In this sheet, supervising tutors made note of strong points and flaws considering the specific deliverable as well as the way latter was presented also by distinguishing between the different team members. Thereby, the evaluation of the final report was part of the post processing stage. Besides, a continuous assessment based on student-tutor interaction took place. The overall assessment of the student teams also includes an evaluation of the source code (quality of the implementation, functional test) and recommendations of the partner universities. A second activity of this phase consisted in the evaluation of the overall course. In addition to the feedback given by the local students all members of the participating universities shared their experience and lessons learned within a 60 minutes long final video-telephony session.

Tutoring and organize: Supervising tutors not only provided technical help in introducing and explaining different software engineering technologies to the student teams, they also made recommendations with respect to organizational issues and deliverables. For instance, this comprised suggestions for presentation and final report structure, hints regarding escalation paths in case of team-internal problems, and resolving of project holdups which could originate from a lack of team coherence.

Define and validate system requirements: After presenting the project topic during the kick-off meeting, the project sponsor monitored the overall progress with regards to the afore specified requirements and constraints. In doing so, the sponsor also validated the deliverables and checked the source code against the initial demands. Furthermore, questions were clarified raised by the students considering the system.

2.5 Conduct project

After the organizational preparation was performed by the universities, the students were able to tackle their specific software engineering project within the given time frame of three months. The execution phase (Figure 2) was subdivided into four distinct activities. Final outcome of each activity was the deliverable as mentioned above.

Prepare: Right after the projects have been assigned, the student teams started to get familiar with the other team members as well as the overall project topic and constraints. Furthermore, they defined the internal project organization including roles and responsibilities, set up a project plan, and identified work packages. Afterwards they distributed these packages among the members and prepared the initial presentation representing the first deliverable, which was discussed together with the local supervising tutors during the initial talk.

Design & implement: In the core working phase the student teams had a total of seven weeks for system design and implementation. Besides, other software engineering tasks e.g. testing, documentation as described in (Rausch and Broy, 2008) were executed. Each team proposed a system design based on the project sponsor's requirements complemented by the results of the initial presentation. This design was discussed and double-checked during the synchronization meeting (cf. Subsection 2.1) in the presence of the local supervising tutors. After the design was approved, students independently implemented the working packages and integrated them in a subsequent step. After the test cases were carried out, the overall results were incorporated in the final presentation which was held at the end of the implementation phase.

Compose & summarize: Finally, a time frame of three weeks was scheduled enabling the student teams to write their final report. In order to facilitate the

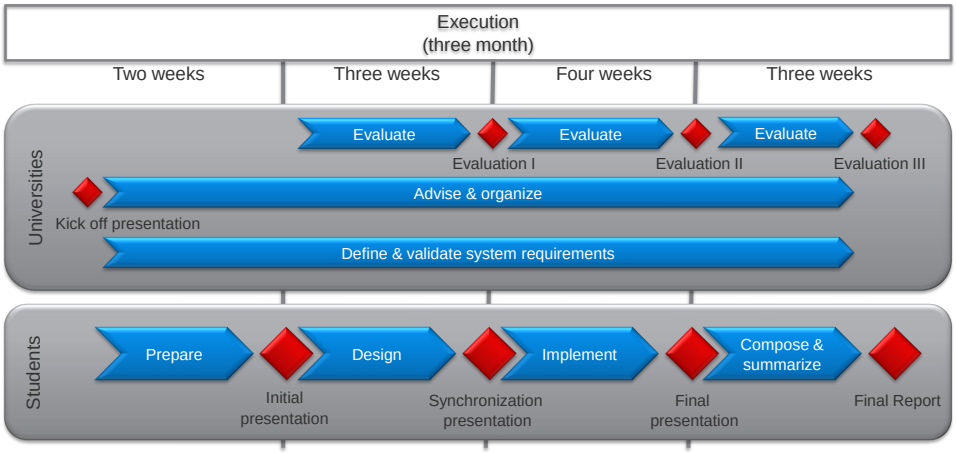

Figure 2: Execution stage details

writing and enhance the quality of the deliverables, students had the possibility to send in their intermediate report. This draft was reviewed by the local tutor and potential suggestions for improvement were commonly discussed during a short meeting at the chair. After the final version of the report was submitted, the students received their final grade based on the evaluation criteria as explained in Subsection 2.4.

2.6 Reflect course

To improve the quality of future courses the 43 students could voluntarily provide feedback during 20 minutes long interview sessions together with the local supervising tutors.

3 The student projects

In the following, we describe three selected student projects conducted in the NEREID course. To capture the experiences and lessons learned, we carried out semi-structured interviews with the students teams from TUM, after the projects were completed and evaluated. Hence, the statements in this Section reflect the perspective of the student teams and the collaboration between them.

3.1 Project 1: University foreign relations map

Project 1 aimed at developing a web application to graphically display partner university relationships. The application should make it easier for students to find partner universities and compare possible exchange partners. The scope included displaying an university relations map as an overview of partner universities, basic administration capabilities to edit university relationship data as well as provisioning of an online forum for communication between students and uploading experience reports.

The project team consisted of students of three sites with two students from each location: INSA de Lyon, TUM, and UE. The latter was the initiator of the project thus also the site of the project sponsor. Project 1 had a student project lead from the project sponsor site who was appointed already before the project was started and had a close contact to the project sponsor. The worksplit was defined by the lead, the development tasks were separated by subsystems: The responsibility for the web page was at UE, the online forum was assigned to INSA de Lyon and the database development was the task of the team at TUM. The requirements for the project were provided by the project sponsor at UE. All communication with the project sponsor was handled by the project lead at UE; there was no direct communication between the team at TUM and the project sponsor. For project communication, weekly phone calls were scheduled, however, due to different reasons, the project team was never fully present at these meetings. Later, e-mail communication was used instead. The team at TUM experienced long delays to receive an answer to e-mails – sometimes up to one week. Another issue of distribution was the access to the web server at UE, as the team at TUM developing the database had no direct access to the server and always needed to go via the team in UE.

A major problem occurred with the INSA de Lyon student site, which was responsible for the online forum. According to the TUM, there was low involve-

ment from the INSA de Lyon team and they were most of the time arguing why they cannot proceed. The TUM team kept sending e-mails to the project lead and asked for an escalation. However, the problem was not solved. At the very end of the project, the INSA de Lyon team posted rudimentary code for an online forum as their contribution, but it was not integrated with the overall system and therefore not usable. The TUM student team felt annoyed about their student colleagues at INSA de Lyon.

The student team at TUM reported in the interview that they saw there were different motivations of the sites involved. In particular the team at UE treated the project lightly. When there were discussions about problems with the project lead, he referred to clarifying all issues with his professor in a personal and informal way. According to the TUM team, there was no formal final presentation at UE, but only an informal meeting with the professor, and the project report of the TUM team was used instead of writing an own report.

3.2 Project 2: Picture-based itineraries

Task of project 2 was to develop a picture-based navigation system for pedestrians. While existing navigation systems usually use street names and a map to explain the route, many pedestrian ways do not have names. Therefore the goal of this project was to create a system which uses pictures augmented with arrows to explain the route, e.g. view of a crossroad where one needs to turn left.

The project team involved students from three sites in three countries – INSA de Lyon, TUM, and USM – with two students at each site [3]. The supervising professor of the INSA de Lyon site was in the role of the project sponsor. The student team felt there was a great latitude with respect to the requirements, as only high level requirements were given. The students organized weekly meetings to exchange on the current status and to make upcoming decisions. The team split their work according to subsystems to minimize the need for communication: the server part was taken by the INSA de Lyon team, the client part was chosen by the TUM team, and the navigation arrows were assigned to the USM team. In contrast to project 1, this project had no explicitly appointed student project lead. While a student from the INSA de Lyon team suggested himself in his first e-mail as project lead, the TUM team answered the decision should be deferred to the first meeting, where the team members get introduced to each other. Later at the meeting, no decision was taken. As the project task originated from INSA de Lyon, they gave an overview of the high level requirements during the first meeting. As many aspects of the task were still underspecified, the INSA de Lyon team took over the responsibility to further clarify the requirements and provide a specification.

Communication was arranged in weekly meetings and exchange of e-mails. While the first meetings had no pre-defined agenda, the meetings became more structured over time. Chat was used as communication medium as the available network bandwidth was not sufficient for voice-over-IP or video communication. The team reported that they made good experiences using this mean, as it allowed for a well-regulated communication.

From the beginning of the project, there were problems in the cooperation with the team in USM. The students from USM did not attend the first meeting, as they "totally forgot the meeting date". This lead to additional effort, as the information provided and the decisions made needed to be re-discussed with the USM team. The TUM team did not escalate the problem in order not to squeal on their colleagues, only some e-mails were exchanged within the team. There was no contribution from the USM team visible until two days before the final presentation: A code delivery was provided on December 20th, which was not integrated and therefore not working at the final presentation. Also there were problems with the time synchronization between the three locations. While the TUM team had to give its final presentation and demonstration of the system on December 22nd, the INSA de Lyon team went already on holiday from December 19th, so it nearly happened that the server needed for the demonstration was not available. Fortunately, the INSA de Lyon team could run the server during their holiday from the students' hall of residence.

3.3 Project 3: Cadaster System

Project 3 focused on developing a system which uses geographical data of the cadaster system of Puebla (Mexico) to map articles to specific geographic locations. Users should be able to visualize, create and modify articles related to a specific reference point on

[3] A second team consisting of students from INSA de Lyon and TEC has been working on the same project. However, only the interview statements of the team in which students from the TUM were involved are reflected in this article

the map.

In the project team were three students from TEC, two from INSA de Lyon and one student from TUM. The topic was defined by the supervising professor from TEC who acted as project sponsor. A high level requirements specification was already existent at the project beginning. Project 3 had a student project lead from TEC who was designated before the project actually started. The team members jointly organized the distribution of work. They decided to have a functional worksplit across locations: One member of the INSA de Lyon team was responsible for organizing meetings (e.g. invitation and agenda) and the documentation (e.g. meeting minutes). TEC was accountable for the conversation and clarifications with respect to the requirements. The student from TUM took over the role of the technical implementation and integration as lead developer. Thereby, each site was in charge for a part of the system. Furthermore, for each location the team defined one so-called team manager who acted as the single point of contact in case of problems or bug reports with the part of the system, for which the site was responsible. The team had weekly meetings, including both one video conference meeting for organizational issues and status reports and one chat for technical clarifications. As and when required, the frequency of meetings was increased. The supervising professor from TEC who was in the project sponsor role occasionally participated in the video conferences. Overall the collaboration went well, although it was noticed that there are cultural differences between the countries: the team in TEC was perceived to be more relaxed, while the student colleagues at INSA de Lyon sometimes expressed their concerns when progress was not as fast as they wanted. The team had to bear the time difference in mind, which was 7 hours, but it was not a serious problem as there was a sufficient overlap in work time of Mexico and Europe and they could agree on time slots for the meetings, where all sites were available. Sometimes it was challenging to find a date for the video conference, as it was difficult for both the INSA de Lyon and the TEC team to get a video conference room due to limited resources. One major issue occurred with respect to understanding of the requirements: The Europeans and the Mexican students had different imaginations of a cadaster system: While the students at INSA de Lyon and TUM assumed that they need to implement a cadaster system with legal registers – as known in Europe –, the Mexican expected to build a system with a simple geographical map. As the team members detected that they have different domain knowledge, they were able to resolve this issue.

3.4 Experience and feedback

Altogether, the problems encountered by the teams very well reflect real-life issues, in particular communication problems as well as technical, organizational, and people aspects (Lescher et al., 2009). Due to the geographic and organizational separation, the students perceived themselves being in local sub-teams instead of one global team. In critical situations this resulted in a characteristic “us” versus “them” attitude. However, all of the NEREID projects were finally able to present good results and made an important learning experience. Being asked for their lessons learned and their feedback towards the seminar, the students mentioned the following points:

Separation by subsystems: The modular architecture and separation by subsystems has proven well, as it reduces complexity and communication needed across sites. Project tasks should be defined in such a way, that a modular separation is possible.

Harmonize deliverables and schedule: It was suggested by the students to better harmonize the deliverables and the schedule of the projects. In particular, the dates and the format for the final presentation and the final report along with the evaluation criteria should be established in an unique way for all involved locations.

Joint kick-off meeting: The students suggested to have a joint kick-off meeting with all involved sites and also with the project sponsor in the role of the customer. Actually, the sponsor was only available during the final presentation. It would improve understanding and morale, if this customer contact could happen already at the kick-off meeting. Instead of a kick-off meeting at the very beginning of the project, this could be also organized as a milestone meeting after 1-2 weeks of the project. This would allow for a more funded clarification of open questions, as the students already had the chance to get into the matter of their project.

Escalation path: A clearly defined escalation path would help the students to deal with problems which are outside of their responsibility. The current experience has shown that the students are reluctant to run down their colleagues in front of their supervising professor.

All teams interviewed emphasized that the project was a very valuable learning experience for them, those were aspects which “cannot be learned by listening to lectures only.”

4 Lessons learned and recommendations

This section points out major lessons learned of the universities acting in both roles: as supervising tutor as for the three projects presented in Section 3 and as project sponsor. At the same time, recommendations for future GSE courses in academia are provided substantiated by a short explanation.

Generally, we suggest that all universities agree upon the three presented deliverables. Not only project's performance can be compared more easily after every student team has to deliver the same results, standardized software engineering artifacts also facilitate the composition of a common time schedule which can be followed by the teams irrespective the individual project topic. The time schedule should be ready as early as possible (preferably two months before semester start) allowing students to fit in the NEREID course in their other academic activities. Furthermore, we consider a joint kick-off meeting including at least all student team members, the project sponsor, and possibly, all supervising tutors, as being mandatory in order to eliminate initial difficulties endangering the participants ambition and engagement.

Assigning 4-6 participants in total to a student team involving 2-3 locations in which each location provides exactly two students has been proven to be successful throughout the four conducted GSE projects. The limited size of the team induces the team members to equally contribute to the project's success by simultaneously giving the tutors the possibility to personally advice and monitor each individual student. While communication overhead and project's content complexity is reduced for students and supervising tutors, the project sponsor only needs to keep track of a very limited project scope of 4-6 engineers working for three months on a prior defined outcome. Regarding deviating interests on team and on location level (therefore students from one location pursue a goal which differs from the overall team goal) we propose that all involved sponsors and tutors consistently state the main objective right at the beginning and commonly evaluate the entire team performance against the actual achievement of this objective.

When it comes to the project content, we recommend to closely link a project sponsor's focus of research with the proposed topic. Not only the professor in charge would be acquainted with the project's underlying body of knowledge, the sponsor would also have an increased interest carrying out this project since latter's successful finalization may positively contribute to the chair's research activities. In addition, either the individual content or the type of the project should ensure intensive communication within each student team. For instance, topics requiring a organizational divide & conquer phase in order to split up and later on integrate the different work packages are suitable since they entail communication among the involved students. Furthermore, projects with an explicit need for country-specific knowledge and local information generate cross-national exchange, too.

Regarding the specific target group, participants should be at least in the 5th semester of their computer science study program, hence late bachelors and master students. On the one hand, those team members are already familiar with elementary software modeling and design techniques (e.g. UML, EPK, Petri nets, and ER models), on the other hand those participants were already in contact with functional and object oriented programming languages almost always required for the fulfillment of the implementation phase of a project. Being less engaged in the well-known technical related aspect of a GSE project would allow participants to focus on the organizational and communicative part of the course in more detail.

Nevertheless, we do not deem extensive preparatory classes taught before the actual project start as an indispensable necessity. Tying in with the previous point, participants have been already taught to work on software problems locally and therefore the introductory courses would only convey knowledge regarding project management with regards to global distributed projects. In turn, we propose that a short but crisp GSE overview session in advance would help to make a start more smoothly for all actors by also saving planning and coordination time for the students. This session, which could be coupled with the kick-off meeting during the execution phase, should also address escalation paths in the case of team internal and external problems (e.g. definition of project lead, team internal friction, lack of communication with the sponsor), thus whenever the team cannot solve the issue autonomously.

Furthermore, we suggest to establish a common evaluation scheme for all participating universities making each student's work transparent, comparable, and traceable by generating a more objective picture of the participant's performance at the same time. This scheme, which could be implemented through a simple electronic spreadsheet with the columns representing the three deliverables and rows depicting the universities, is exchanged among all supervising project tutors after the execution phase is finished.

Nevertheless, we deem important to keep organizational overhead low for the tutors. Hence, completing the mentioned sheet should not take longer than 10-15 minutes for each project team and deliverable.

Not surprisingly, we noticed a higher communication and coordination overhead in comparison to similar local courses when conducting the different GSE projects. However, this increased engagement was paid off by insights in research groups of foreign countries as well as the international experience we gained ourselves by carrying out those project. Being in the role of the supervising tutors and the sponsor, we learned how to organize, execute, and coordinate distributed projects by interacting on a cross-country level with different interest groups. We are satisfied with the solid results delivered in the short amount of time through the students in hoping to prepare them for upcoming multi-cultural endeavors in the industry. We are also confident, that besides academic exchange programs courses like ERASMUS, NEREID will help future software engineers to succeed in an increasing globalizing work environment. In addition to foreign language mastering, industrial placement in companies located in other countries and student mobility, we propose and recommend to fully integrate GSE and international project management in their syllabi.

5 Summary and outlook

Presently, graduated software engineers receive education in technology and management. However in the arising context of globalization, the international dimension should be included. In this article, we presented our experiences gained when offering global software engineering projects to university students. We described the applied approach involving 43 participants at five distributed academic institutions, pointed out three concrete student projects in detail, and presented the lessons learned and key recommendations from the teaching staff and student perspective. We are confident, that the introduced approach in combination with the suggestions will serve as a solid foundation for similar GSE endeavors at universities.

As the NEREID course was seen very successful, we will re-offer it in upcoming winter semester by taking the lessons learned into consideration. In particular, the deliverables, the time schedule, and the evaluation scheme will be harmonized across all universities and a clear escalation path for the students will be set up. Furthermore, the set of partner universities will be extended by a South-Korean university. More information on NEREID can be found on our website:

http://wwwmatthes.in.tum.de/wikis/nereid/home

We express our gratitude to all universities taking part in the NEREID course during the winter term 2009/10. We are looking forward to jointly organizing and hosting a second edition of the course next winter.

REFERENCES

Carmel, E. (1999). *Global Software Teams (High Performance Cluster Computing)*. Prentice Hall.

Damian, D., Kwan, I., and Marczak, S. (2010). *Requirements-Driven Collaboration : Leveraging the Invisible Relationships Between Requirements and People*, volume 2010, chapter 3. Springer-Verlag, Heidelberg, Germany, 1st edition.

Gotel, O., Kulkarni, V., Say, M., Scharff, C., and Sunetnanta, T. (2009). Quality Indicators on Global Software Development Projects: Does "Getting to Know You" Really Matter? In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'09)*.

Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. In *FOSE '07: 2007 Future of Software Engineering*, pages 188–198, Washington, DC, USA. IEEE Computer Society.

Hofstede, G. and Hofstede, G. J. (2004). *Cultures and Organizations – Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. McGraw-Hill.

infoAsset. Website. http://www.infoasset.de; visited on March 7th 2010.

Laurini, R. and Sol., D. (2009). Post mortem report, assessment of the first experimentation of the nereid international project. page 5. INSA-Lyon, Tec de Monterrey/Campus de Puebla.

Lescher, C., Herbsleb, J. D., and Bass, M. (2007). Collaboration in Global Software Projects at Siemens: An Experience Report. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'07)*.

Lescher, C., Herbsleb, J. D., and Bass, M. (2009). A Coordination Risk Analysis Method for Multi-Site Projects: Experience Report. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'09)*.

NEREID-Page. Website. http://wwwmatthes.in.tum.de/wikis/nereid/home; visited on March 26th 2010.

Rausch, A. and Broy, M. (2008). *Das V-Modell XT : Grundlagen, Erfahrungen und Werkzeuge*. dpunkt.verlag.

Sangwan, R., Mullick, N., and Paulish, D. J. (2006). *Global Software Development Handbook*.