Teaching Global Software Engineering by Simulating a Global Project in the Classroom

Yang Li, Stepan Krusche, Christian Lescher and Bernd Bruegge
Technische Universität München
Chair for Applied Software Engineering
Munich, Germany
{liya, krusche, lescher, bruegge}@in.tum.de

ABSTRACT

Globalization has long since found its way into software engineering. Many companies transfer part of their development activities to distributed countries in order to ensure their global competitiveness, gain access to local markets and react to the prevailing lack of specialized workforce. The global distribution of project teams introduces new challenges: Geographic separation, different time zones, remote communications, and culture and language barriers make the collaboration between team members more difficult.

Instructors in universities are faced with the problem of how to make students with little or no experience aware of the challenges of Global Software Engineering and equip them with skills to deal with them. International practical courses are effective but require high organizational effort. In this paper, we describe an exercise for teaching Global Software Engineering in a single classroom and report on our experiences. The exercise simulates a global software project within three sites. Through the exercise, students experienced some of the aforementioned challenges and tried to deal with them in a simulated environment.

Keywords

Global Software Engineering; Simulation; Classroom; Education; Project Management

1. INTRODUCTION

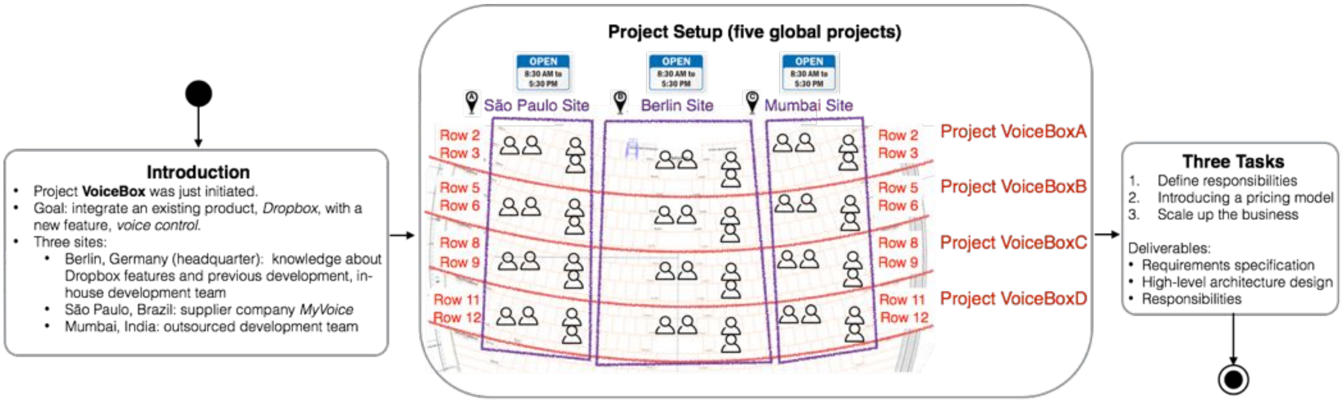
Software engineering is increasingly a globally-distributed undertaking [14]. Development teams are often distributed over multiple sites and countries. A variety of motivations, including proximity to markets or customers, access to talent regardless of location and cost competitiveness are driving this move. While the advantages are tempting, Global Software Engineering (GSE) also introduces new challenges: Geographic separation, working in different time zones and different cultures make the collaboration between team members more difficult than in co-located projects.

Typical collaboration problems in GSE include ineffective communication, unclear responsibilities, and low team cohesion. Geographically distributed teams lack effective communication such as face-to-face conversations to effectively deliver messages in multiple channels (e.g. voice, emotions and body language) and get instant feedback. Miscommunication occurs repeatedly in ineffective communication, which is one major cause for issues such as cost overruns, quality problems and project delays [8]. Additionally, responsibilities between teams are unclear, due to a reduced level of awareness about who is currently working on which topic. This delays task assignment and other types of decision making especially when change occurs. Finally, team cohesion – team members stick together to pursue a common goal – is often used to measure team performance. High team cohesion leads to enhanced motivation, increased morale, greater productivity, harder work, more open communication, and higher job satisfaction compared to non-cohesive groups [5, p.52]. However, team cohesion is more difficult to achieve between globally distributed teams.

Instructors in universities are faced with the problem of how to teach GSE, imparting not only theoretical knowledge, but also practical skills required for globally distributed projects. Previous work [13, 6] presented a systematic review on teaching GSE and the learning environments used. A common approach is to set up an international practical course, involving students from multiple universities who then work jointly over distance on a software project.

We reported on our experience with teaching three distributed software engineering project courses [2], namely JAMES [1, p.725], PAID and STARS. Students from Carnegie Mellon University in Pittsburgh, USA and from Technische Universität München (TUM) in Munich, Germany collaborated to specify, design, and realize software for a real industrial client located at Stuttgart, Germany and Detroit, USA. The three project courses aimed to address team management, client collaboration and rationale capture in a global software project setup. In the Global Studio Project students from Carnegie Mellon University, International Institute of Information Technology Bangalore, Monmouth University, Pontifical Catholic University of Rio Grande de Sul, University of Limerick and TUM jointly worked on a software project [14, p.183]. We also conducted NEREID (Network of Engineering univeRsities Educating in Intercultural Design), involving students from Mexico, Chile, France and Germany [12]. Gloor et al. [7] described a global project course they taught for six years that involves 30–40 students from two to four sites, located in the USA and Europe.

To appear in the 47th ACM Technical Symposium on Computing Science Education - SIGCSE, March 2016, Memphis, TN, USA.



Introduction
- Project **VoiceBox** was just initiated.
- Goal: integrate an existing product, *Dropbox*, with a new feature, *voice control*.
- Three sites:
  - Berlin, Germany (headquarter): knowledge about Dropbox features and previous development, in-house development team
  - São Paulo, Brazil: supplier company *MyVoice*
  - Mumbai, India: outsourced development team

Three Tasks
1. Define responsibilities
2. Introducing a pricing model
3. Scale up the business

Deliverables:
- Requirements specification
- High-level architecture design
- Responsibilities

Figure 1: Exercise workflow

Although the approach of an international practical course was found to be effective, it has the disadvantage of requiring high organizational effort. To set up such a course is a globally distributed project in itself, where instructors from multiple universities need to agree on common objectives, a time schedule which fits to the semester timetable of all involved sites, a harmonized evaluation scheme, etc.

In this paper, we present an alternative way of teaching GSE in the classroom, simulating a global project. The simulation exercise can be carried with low overhead and has shown to be effective in addressing typical GSE challenges and in applying first strategies to overcome them. The rest of the paper is organized as follows: Section 2 introduces the workflow and detailed design of the simulated global project. Then we share our findings and lessons learned in Section 3. Finally we conclude the paper in Section 4.

2. COURSE DESIGN

To simulate a global project with competitiveness of different talents and technical skills, our strategy is to simulate GSE projects with three sites: one site is the in-house development team at headquarters; a second site is an outsourced development team; and the third site has a unique set of technologies and domain knowledge. The goal is to integrate a *special technology* provided by the third site with an *existing product* developed by the other two sites.

One of the biggest challenges of the simulated project is how to prepare students to become a domain expert within a short amount of time and only by using the materials provided in the exercise. This is especially hard for the third site. We have brainstormed for ideas and researched for available resources that we can provide to students for fast learning. After study and discussions, we have chosen "voice control" as the *special technology* that the third site provides. For students in computer science, the idea and main functionalities of voice control are not difficult. However, only few of them are aware of the approaches and technologies behind. For example, how speech recognition works and how vocabularies are customized. Nuance is a leading speech-recognition technology company. It provides rich documentation[1] for developers who want to integrate Nuance speech services. We provide students the Nuance developers documentation to obtain domain knowledge. As for the *existing product* that will be integrated with speech recognition, we chose Dropbox[2], due to its popularity. Most students are already familiar with Dropbox, so the overhead of gaining knowledge to understand the product and requirements is low.

The exercise workflow is shown in Figure 1. At the beginning of the 1.5 hours exercise, the instructor gives an overview. Afterwards four global projects (i.e. A, B, C, D) are set up in the classroom. Three different tasks are assigned and conducted by students. In the following, we elaborate the exercise introduction, project setup, and each of the three tasks.

2.1 Exercise Introduction

Before the project starts, the instructor first introduces the context, goal, and organization of the global project. In our course, we defined the exercise context as follows:

"Project *VoiceBox* was just initiated. The goal of the project is to integrate an existing product, *Dropbox*, with a new feature, voice control. To gather domain expertise in the new voice control technology, your company recently acquired the supplier company *MyVoice*[3] in São Paulo, Brazil. The main development sites are in Berlin, Germany (headquarters) and Mumbai, India. While the team in Berlin has rich knowledge about Dropbox features and previous development, the team in Mumbai consists of good coders who are inexperienced with Dropbox however."

In addition, we specified the following constraints within the global projects:

- Project members may not leave their site. Exception: each project has one and only one liaison from the Berlin site, who can travel to the other development sites. He/she can talk to the developers only during local business hours (8:30 am - 5:30 pm).
- Each project may select their own communication mechanism (phones, Skype, and chat), but they can respond only during local business hours.

[1] Documentation is available on Nuance developers homepage: http://dragonmobile.nuancemobiledeveloper.com

[2] Dropbox allows users to securely share and synchroize files. Dropbox official website: www.dropbox.com

[3] MyVoice is an imaginary technology we named. The description and materials about MyVoice were adapted from Nuance documentation.

- Use the project Confluence[4] space to document meetings and deliverables.
- Projects may request extra hours after usual business hours, however this will lead to additional costs.

2.2 Project Setup

After exercise introduction, global projects are organized in the classroom. We have grouped the students into four projects. Each project reflected the introduced three-sites organization, i.e. São Paulo, Berlin and Mumbai sites. We used a chart to illustrate the project setup (see Figure 1), a map of the classroom where the course took place. Depending on the number of students and the classroom layout, this "matrix-organization" may be adjusted. Additionally, the instructor showed a time zone map (see Figure 2) to visualize the time zones at the three sites. After students join the projects, each project has five minutes to decide and setup the project communication mechanisms. Then the first task is introduced by the instructor.

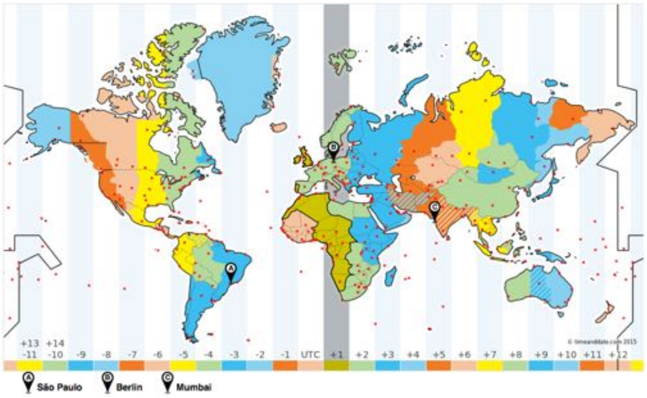

+13 +14
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 UTC +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12
© timeanddate.com 2015
A
B
C
São Paulo
Berlin
Mumbai

Figure 2: Time zone map of the three sites (created on timeanddate.com)

2.3 Task I

Duration: Students have in total 20 minutes to work on this task which correspond to four real world hours[5]. The simulated time for this task is at Berlin Time 10:00 am. This corresponds to São Paulo time 05:00 am and Mumbai time 1:30 pm. Project members at São Paulo site are still sleeping and do not work at the beginning of the exercise. The instructor shows Figure 3a to students that visualizes current local time in the three different time zones.

Task description: At the headquarters in Berlin, the team first specifies the requirements. Afterwards, the Berlin team arranges a meeting with the Mumbai team to discuss requirements, architecture, and responsibilities of each site. Open questions to the São Paulo site are listed.

Instructions (work breakdown structure): The following instructions were given to the students during the task.

- Berlin Site:
  - Specify early requirements for VoiceBox.
  - Provide information about existing architecture of Dropbox.
  - Organize meetings.
  - Lead other sites to refine requirements, architecture and responsibilities.
- São Paulo Site:
  - Read the provided materials about MyVoice.
  - Prepare yourself to be the MyVoice experts.
- Mumbai Site:
  - Read the provided programmer's guide of MyVoice.
  - Give input about how to integrate MyVoice into the current product.

Process: After clarifying the task and instructions, the instructor distributes initial requirements to the Berlin site and programmer's guide to the Mumbai site. Since it is at 5:00 am in São Paulo, this site is not involved in meetings with the other two sites. Instead, students at this site use the time to become experts in the MyVoice technology. Custom vocabularies guide and programmer's guide of MyVoice and a pricing model (see Table 1) are provided to the São Paulo site. In this task, the goal is that students from each site understands the setup, the roles they are playing and obtain domain knowledge. Especially students will experience for the first time in software projects, that one site (São Paulo) is not reachable due to time difference.

	Silver Platinum ($999/year)	**Gold Platinum** ($3999/year)	**Platinum** ($9999/year)
Cloud-based automatic speech recognition (ASR) for dictation and web-search	yes	yes	yes
Vocabularies for ASR customization	Up to 200 phrases Up to 40 vocabularies	Up to 1000 phrases Up to 80 vocabularies	Up to 300 phrases Up to 160 vocabularies Vocabularies update
Cloud-based Text-to-Speech (TTS)	English	English, Spanish, Arabic, French, German, Chinese and Japanese	40+ languages
Platforms	iOS and Android	iOS, Android, and HTTP REST interface	iOS, Android, and HTTP REST interface
Production Pricing	Free up to 2k transactions/Day/App Above 2k: $.008/transaction	Free up to 10k transactions/Day/App Above 10k: $.008/transaction	Free up to 20k transactions/Day/App Above 20k: $.005/transaction

Table 1: Pricing model of MyVoice (material provided to the São Paulo site)

2.4 Task II

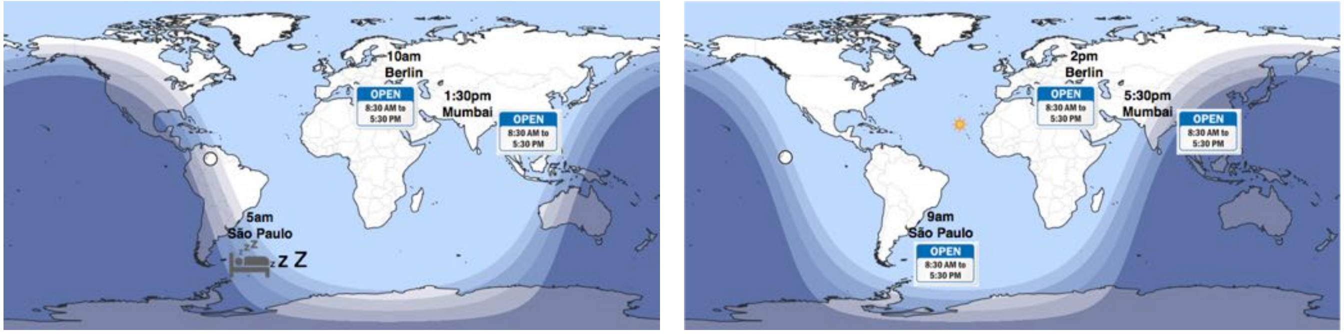
Duration: Students again have in total 20 minutes to work on this task. The simulated time for this task is at Berlin Time 2:00 pm. This corresponds to São Paulo time 09:00 am and Mumbai time 5:30 pm. This means that Mumbai just finished local business hours. The instructor shows Figure 3b to students that visualizes current local time in the three different time zones.

Task description: The São Paulo site receives the listed questions of Task I and answers them right away. The pricing model of MyVoice is also introduced by the São Paulo site. Requirements, architecture and responsibilities need to be refined based on the new information about MyVoice. Be careful, the office hour at the Mumbai site is about to end.

Instructions (work breakdown structure): The following instructions were given to the students during the task.

[4] Confluence is a Wiki provided by Atlassian. We set up the space for the students before the exercise.

[5] Every 5 minutes one real world hour passes.



(a) Local times for Task I

(b) Local times for Task II

Figure 3: World map showing daylight and the current time of the three sites

- Berlin Site:
  - Make business decisions of VoiceBox based on new information from other sites.
  - Provide information about existing architecture of Dropbox.
  - Lead other sites to refine requirements, architecture and responsibilities.
- São Paulo Site:
  - Answer the questions about MyVoice.
  - Introduce the MyVoice Pricing Model.
- Mumbai Site:
  - Give input about how to integrate MyVoice into the current product.

Process: After Task I, students are prepared for diving into details about the project. In this task, the instructor expects the students to organize meetings cross sites. The pricing model introduced by the São Paulo site is a surprise to the other two sites as they do not know about it. Due to the pricing model, the project might change the development plan they discussed before. Cross-site negotiation might be involved and delay in communication and decision-making will occur. Students will also need to decide if the Mumbai site needs to take extra working hours for the meeting or not. Until now, students should also use at least one mechanism to communicate and to document the artifacts such as meeting minutes and decisions in a global setup. The goal is, that students will face common GSE challenge such as ineffective communication, miscommunication between sites, unclear requirements and lack of transparent information.

2.5 Task III

Duration: Students have in total 20 minutes to work on this task. However, this time students will define the current time and inform the instructor about the meeting time slots.

Task description: After the previous meeting, the Berlin headquarters identified new requirements for VoiceBox to support the increasing business. Define a time slot for a meeting, where you address these new requirements. Refine requirements, architecture and responsibilities.

Instructions (work breakdown structure): The following instructions were given to the students during the task.

- Berlin Site:
  - Schedule a meeting with other sites.
  - Address new requirements.
  - Lead other sites to refine requirements, architecture and responsibilities.
- São Paulo Site:
  - Answer questions about MyVoice.
  - promote different MyVoice services.
- Mumbai Site:
  - Give input about how to integrate MyVoice into the current product.

Process: The instructor distributes new requirements to the Berlin site. The new requirements include supporting a larger user base and localized languages. It also requires that VoiceBox will work on multiple platforms. The Berlin site is expected to discuss with the Mumbai site about the feasibility and difficulty of realizing the new requirements. They might also want to discuss with the São Paulo site about different services come with different prices.

In order to find a suitable time slot to meet, students will propose different strategies. They might request the Mumbai site to take more extra hours and discuss the issues as soon as possible. To avoid taking extra hours, an alternative is, that the Berlin site can meet with São Paulo site first and wait until the second day to meet with the Mumbai site. This is a typical extra working costs vs. delay costs situation. Via the meeting scheduling, students will take serious account of working hours and time differences in global projects. This is a valuable point that is worth discussing in a retrospective session. In particular, the instructor can interview students about their meeting time and strategy. He can relate the situation to global projects that often have shifted local business hours (different from other local companies) to maximize overlapping hours with other sites.

3. FINDINGS AND LESSONS LEARNED

We conducted the GSE exercise in the course of *Software Engineering II: Project Organization and Management* in summer term 2015. The target audience of this course were students who had basic knowledge about software development and wanted to enhance their knowledge in project management. The course covered concepts such as agile

#	Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
S1	... it is difficult to coordinate time	34%	50%	11%	4%	2%
S2	... there is communication overhead	23%	52%	18%	5%	2%
S3	... there is a lack of informal communication	27%	43%	23%	5%	2%
S4	... it is difficult to communicate changing requirements	21%	45%	23%	11%	0%
S5	... expertise is not shared with relevant stakeholders	18%	46%	18%	16%	2%

Table 2: Evaluation results of the GSE exercise with Likert scale responses to statements: The exercise addressed the challenge in global projects that ...

methods, configuration management and continuous delivery [9, 10], as well as team-based software development [3, 4]. 450 computer science students registered for the course, including half bachelor students in their second year and half master students. About 100 students participated in the GSE exercise. In the following we present the findings and lessons learned from conducting the exercise.

3.1 Findings

We were amazed at how diverse the documents and communication logs were. Each project had established its own communication mechanism. Project A and C used Confluence to share important information and questions to other sites. Project B was communicating mainly on an online chatting platform. Project D had a liaison who traveled between sites throughout the whole project, while Project A had a liaison only for Task III. Project D also created a Facebook group so that everyone could communicate online. Among the four projects, we have observed that Project D had the most effective communication, the best team performance, and high quality documentation. The liaison played an important role in face-to-face communication and motivating other sites when visiting them. Project A was frustrated at the beginning, but was able to catch up after all sites had improved the communication. In Project C, the team was underperformed and many of the GSE difficulties were shown. Students had ineffective communication (only using Confluence to share documents but no chatting or visiting). The Mumbai site got unmotivated. Project C didn't have a leader to motivate and organize the team.

We evaluated the exercise in an online questionnaire. 56 students took part in the questionnaire and reported about their experience of GSE challenges in the exercise by assessing between "strongly disagree" to "strongly agree" (five-level Likert scale) to statements shown in Table 2. 84% respondents agreed that the exercise addressed time coordination challenges. This indicates that despite being in the same classroom, the students were able to experience synchronization issues and that the simulation of the time differences worked well. 75% respondents agreed that the exercise addressed communication overhead because of the geographical separation of project teams. The respondents reported communication challenges they experienced between the three different sites. For instance, one project reported that they have agreed on using one communication channel, but some team members were still trying to initialize other communication mechanisms. Another reported communication challenge was caused by the different time zones. When developers at one site wanted to discuss with other sites, but unluckily they were out of business hours or they had to request for extra hours.

About 70% respondents agreed that the exercise had presented the lack of informal communication in global projects. This result corresponds to our observation during the exercise. 66% respondents agreed that in the exercise it was difficult to communicate changing requirements between the different sites. However, in another question, 46% respondents reported that they were able to address changing requirement. They reported that creating a page in the Confluence wiki which includes a notification mechanism for changes, was an effective measurement to deal with the problem. The challenge of sharing expertise with relevant stakeholders was addressed in the exercise for 64% respondents. They reported that it was difficult enough to find the right person and still they have to overcome the communication barriers.

We also asked which communication channels the students used. They could choose multiple answers. 21% used Skype as chatting application. 61% used Facebook, some of them used its chatting feature, others used the group feature that offered commenting and like functionality. 48% used Confluence as Wiki to create pages with important knowledge and communicating with comments. 39% traveled between the different sites, especially for negotiation tasks.

Finally, 53% respondents agreed that they liked the global software engineering exercise, in particular they liked to experience the different challenges in a condensed but yet very practical way and in real time. One student wrote: "The whole execution was interesting, effective and fun! Never thought it would be possible to emulate a global project management inside a class". For the unsatisfied students, they disliked the exercise because of its high complexity and the frustration caused by the GSE challenges. This implies again the exercise was effective to address GSE issues that often reoccur in the real world. Nevertheless, a retrospective discussion with the students about the frustration that reflects GSE challenges and how to address them would be helpful.

On the other hand, the students also suggested possible improvements that we consider for the next GSE exercise. Some indicated that the overall exercise was too complex and should be simpler, especially regarding the technical documentation that was handed out. Others had problems to understand the exercise and were confused about the different tasks. They asked for more time and information and more explicit requirements. One student suggested to also include cultural differences into the exercise.

3.2 Lessons Learned

From our own observation and students' feedback we identified the following lessons learned.

Project Context. The project context was well defined. The idea of integrating voice control in Dropbox was easy to understand, but domain knowledge is required to specify requirements and architecture. Therefore, the three sites were able to play distinct roles in the project. According to students' performance, we consider the defined context is at an appropriate level of difficulty for the exercise.

Time Management. The exercise time was a little too tight. We did not have time to conduct a retrospective discussion to reflect GSE issues students experienced. We recommend to plan two hours for the exercise, which include a 10 minutes retrospective session. We also suggest to allocate 30 minutes for Task I instead of 20 minutes as we did. This allows students to have sufficient time to understand the project and read the provided materials.

Multimedia Support. We used three projectors in the classroom to present world clock, task description and instructions. This helped us greatly in handling the complexity of the exercise. We distributed hardcopies of the pricing model to the São Paulo site. We also created tinyurl links for the downloadable developer's guides, so that students could easily fetch the materials. All of these need to be planed in advance. The practice we suggest is to write a script for screen projection and schedule material distribution.

Curriculum Design. The GSE exercise is suitable for bachelor / master level software engineering courses. In our case, we designed the tasks with a focus on project management. Depending on course goals, the tasks can be adjusted. For example, in a course that focuses on modeling and software design, tasks can be to decompose subsystems and draw UML class diagrams for each subsystem. For courses that have only short time slots for exercises, we recommend other smaller in-class GSE exercises that are described in [11].

Privacy. While privacy is an important concern in industrial projects, we did not focus on it in our exercise. The students were free to choose their own preferred communication mechanism. For example, some students used services such as Facebook or Skype which might not be allowed in enterprises. However, enterprises usually have other mechanisms for secure communication such as private wikis or customized chat services that can only be used internally.

4. CONCLUSIONS

In this paper we described a global software engineering exercise that can be carried out in the classroom. We shared the exercise design and our experience. We consider the first conduction of the exercise successful. The students had experienced work with different timezones, lack of effective communication and other GSE issues that are typical in the real world. In an online questionnaire we found anecdotal evidence that the simulated global project in the classroom was able to address key GSE challenges. We have also identified areas to improve for the next exercise. For instance, we plan to refine the task instructions to make them more precise for students to understand what exactly needs to be done. Additionally, the time management needs to be improved, so a retrospective session can be included. We hope the simulation exercise serves as an example for other university instructors. The exercise can be further adapted and customized to fit into their curricula.

5. ACKNOWLEDGMENTS

We want to thank all students taking part in the exercise.

6. REFERENCES

[1] B. Bruegge and A. Dutoit. *Object-Oriented Software Engineering Using UML, Patterns and Java: International Version*. Prentice Hall, 3rd edition, 2010.

[2] B. Bruegge, A. H. Dutoit, R. Kobylinski, and G. Teubner. Transatlantic project courses in a university environment. In *Proceedings of the Seventh Asia-Pacific Software Engineering Conference, APSEC 2000*, pages 30–37. IEEE, 2000.

[3] B. Bruegge, S. Krusche, and L. Alperowitz. Software engineering project courses with industrial clients. *ACM Transactions on Computing Education.*, 2015.

[4] B. Bruegge, S. Krusche, and M. Wagner. Teaching Tornado: from communication models to releases. In *Proceedings of the 8th Educators' Symposium*, pages 5–12. ACM, 2012.

[5] E. Carmel. *Global Software Teams – Collaborating Across Borders and Time Zones*. Prentice Hall, 1999.

[6] L. Fortaleza, T. Conte, S. Marczak, and R. Prikladnicki. Towards a GSE international teaching network: Mapping Global Software Engineering courses. In *Proceedings of Collaborative Teaching of Globally Distributed Software Development Workshop*, pages 1–5. IEEE, 2012.

[7] P. Gloor, M. Paasivaara, C. Lassenius, D. Schoder, K. Fischbach, and C. Miller. Teaching a global project course: experiences and lessons learned. In *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*, pages 1–5. ACM, 2011.

[8] D. J. Hoch, S. K. Lindner, C. R. Roeding, and G. Purkert. *Secrets of Software Success: Management Insights from 100 Software Firms around the World*. Harvard Business School Press, 1999.

[9] S. Krusche and L. Alperowitz. Introduction of Continuous Delivery in Multi-Customer Project Courses. In *Companion Proceedings of the 36th ICSE*, pages 335–343. IEEE, 2014.

[10] S. Krusche, L. Alperowitz, B. Bruegge, and M. Wagner. Rugby: An agile process model based on continuous delivery. In *Proceedings of the 1st Workshop on RCoSE*, pages 42–50. ACM, 2014.

[11] C. Lescher, Y. Li, and B. Bruegge. Teaching global software engineering: Interactive exercises for the classroom. In *Proceedings of the IEEE 9th International Conference onGlobal Software Engineering (ICGSE),*, pages 163–172. IEEE, 2014.

[12] F. Matthes, C. Neubert, C. Schulz, C. Lescher, J. Contreras, R. Laurini, B. Rumpler, D. Sol, and K. Warendorf. Teaching Global Software Engineering and International Project Management: Experiences and Lessons Learned from Four Academic Projects. In *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU)*, 2011.

[13] M. Monasor, A. Vizcaino, M. Piattini, and I. Caballero. Preparing students and engineers for global software development: a systematic review. In *5th IEEE International Conference on Global Software Engineering*, pages 177–186. IEEE, 2010.

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