COOPERATIVE INTERNATIONAL VIRTUAL CLASS TO TEACH INTERCULTURAL SKILLS IN GLOBAL SOFTWARE ENGINEERING

Patricia Brockmann
Nuremberg Institute of Technology Georg Simon Ohm
Germany

Abstract

For today's IT graduates, programming skills alone are not enough. In addition to knowledge of technical skills, intercultural understanding is vital in order to work together in geographically distributed, international teams. Unfortunately, few students have the opportunity to travel abroad to acquire first-hand the international experience necessary to learn how to work together with software developers from different cultural backgrounds. In order to teach students these intercultural skills, a *project-based learning* approach was applied to teach virtual, international courses in global software engineering. Experiences, difficulties and lessons learned from a number of team-teaching co-operations involving Germany, Japan, Mexico, and Mongolia are described.

Keywords: distance learning, international, intercultural, project-based

Introduction

Information and communication technology help people worldwide to learn and work cooperatively. Globally distributed, multi-national teams often develop the software that this technology runs on. The team members working together on one software project may live in different countries, work in different time zones and speak different native languages.

As a result, global software development presents a number of new challenges: geographic distance, different time zones, different languages, a lack of trust and cultural differences. For example, a team made up of members in different countries can take more than twice as long to complete the same task than if all of them were sitting in the same room (Herbsleb, & Mockus, 2003).

Originally, different time zones were seen as a potential driver for increased productivity. A "Follow-the-Sun" time zone model means that three teams distributed around the planet could work around the clock in eight hour shifts (Carmel, Espinosa, & Dubinsky, 2010). Although this theoretically should lead to increased productivity, many projects experience lower productivity. Different work schedules make it very difficult to communicate real-time with team members at different locations (Sosa, Eppinger, Pich, McKendrick, & Stout, 2002).

Team members who speak different languages often have difficulty communicating in real-time. A third common language, such as English, is often used. This third filter language may require extra time and effort. Team members first need to translate their thoughts from their native language to English. Then they need to translate the English answers from the other group back into their own native language. Some non-native speakers may feel anxious about telephone or video conferences (Sangwan, Bass, Mullick, Paulish, & Kazmeier, 2007).

Teams in different countries, who have never actually met, often experience difficulty in establishing trust. Familiarity often grows naturally during informal communication about non-project related matters, such as sports or hobbies. This important team-building activity is often missing in distributed teams (Nguyen, Babar, & Verner, 2006).

Misunderstandings due to cultural differences have repeatedly proved to be one of the most difficult challenges to overcome in successful global software engineering (MacGregor, Hsieh, & Kruchten, 2005). Increasing awareness of cultural differences to work more effectively in international teams is one of the main goals of this project.

These new challenges emphasize the need for additional skills in IT graduates. The stereotype of the isolated computer programmer working alone is no longer adequate. Today's IT graduates need to learn both international project management methods and the intercultural skills necessary to collaborate with team members in other countries. A wide variety of soft skills are necessary for success in global software engineering, such as communication, teamwork and conflict resolution skills. In addition, self-criticism, dealing with uncertainty and ambiguity, appreciating diversity and multi-culturalism, understanding cultures and customs of other countries are vital for the success of international projects. Due to time and budgetary constraints, few students have the opportunity to go abroad for an entire semester in order to gain international experience first-hand.

This research attempts to find answers to the follow questions:

1. Can virtual, team-teaching courses help students learn intercultural skills as part of their regular curriculum at their home universities?
2. Can project-based learning help students discover their own solutions to intercultural misunderstandings?

To answer these questions, virtual, team-teaching courses between the Nuremberg Institute of Technology, Germany and three international partners are presented:

1. Mongolian University of Science and Technology (2012, 2013)
2. National Polytechnic Institute in Mexico (2014, 2015, 2016)
3. Ritsumeikan University in Japan (2017).

Challenges, experience gained and lessons learned are discussed.

Context

One of the first researchers to use empirical methods to investigate the cultural differences in the software industry was Hofstede (Hofstede, Hofstede, & Minkov, 2010), who applied multivariate statistical methods to analyze data collected from thousands of IBM employees worldwide. Hofstede classified differences in cultural perspectives according to six dimensions:

1. Power distance: The attitude of a society to inequalities among individuals in a society (PDI)
2. Collectivism vs. individualism: The degree of interdependence among members in a society (IDV)
3. Masculine vs. feminine: Success vs. harmony and caring (MAS)
4. Uncertainty avoidance, ambivalence: Feeling threatened by unknown situations (UAI)
5. Long-term vs. short-term orientation: Planning for the future vs. living in the present (LTO)
6. Indulgence vs. restraint: The extent to which life is to be enjoyed vs. showing restraint (IND)

These dimensions can be used as a framework to make international project teams more aware of country-specific differences. In addition to these six cultural dimensions, Hall (1990) differentiated between two ways of how different cultures perceive time: M-time (monochromatic) and P-time (polychromatic). Monochromatic cultures, such as Germany, tend to start and end a meeting at a precisely scheduled time. Polychromatic cultures, such as Mexico, may feel that such a meeting is being rushed through and then cut off abruptly, before they have a chance to adequately express their views.

Hall (1990) also differentiated between high and low context cultures Collectivist societies such as Asian countries also tend to be classified as high context cultures. The personal relationships between people are often an intrinsic part of communication. Facial expressions, gestures and pauses can convey more meaning than the actual words spoken. Western countries such as Germany tend to be more individualistic and lower context cultures. Because written and spoken words convey meaning, communication can be verbose.

Teaching Global Software Engineering

Beecham et al. (2017) and Clear et al. (2015) conducted systematic reviews of papers to define the challenges facing global software engineering education, including: global distance, teamwork, soft skills, stakeholders, infrastructure, development process, and curriculum. Hoda, Babar, and Shastri (2016) discuss socio-cultural challenges in global software engineering education. They conducted a case study of 14 participants from 10 different universities in 8 countries. They identified six dimensions that caused significant challenges: language, concept of time, attitude towards grades, assumptions about national culture, autonomy, and influence of the lecturer.

Methodology

The global software engineering classes analysed in this research were all taught at the graduate level. Class sizes were relatively small, with 20 students or less. As a result, sample sizes are not large enough for statistically significant quantitative analysis. This study implements a qualitative approach based on questionnaires and group retrospectives to gather data.

Students were asked about their opinions at the beginning and at the end of each course. Students filled out questionnaires ranking which factors they felt were most important for global software engineering, such as geographical distance, time zones, language or cultural differences. They rated their expectations of differences between themselves and the students at the partner university, according to the cultural dimensions (Hofstede et al., 2010) (Hall, 1990) discussed in the previous Section Context.

At the end of each course, a project review and retrospective based on the moderation method "4Ls" (Gottesdiener, 2010) was conducted. Each student was asked to fill out four sticky notes, one in each color:

1. **Like** (green): What did you like about this project?
2. **Lack** (red): What did you miss? What went wrong?
3. **Learn** (blue): What did you learn during this project?
4. **Long for** (yellow): What would you do differently next time?

Students placed their notes on the board and explained their experiences in each category. These discussions are summarized in the Section Results.

Project-Based Learning

Although the teaching method *project-based learning* is often used interchangeably with the term *problem-based learning*, each method has a slightly different focus. Problem-based learning was first introduced to teach medicine at the McMaster University in Canada (Barrows, 1996). The idea is to replace instructor-centered, frontal lectures, which emphasize the passive consumption of material and the rote memorization of facts. Instead, students work in small, autonomous groups, and the instructor plays the role of a facilitator. An authentic problem is presented, without any introductory lectures. Students actively self-organize to investigate and construct their own solutions in a case study. The central hypothesis is that students learn more effectively when the learning process is centered on a concrete problem. Because the learning is self-directed, students develop problem-solving skills.

Project-based learning (Savery, 2006) is quite similar to problem-based learning. Students work together to complete a project and thereby encounter a number of problems. Each of these problems can be expanded on as "teachable moments." The major difference to problem-based learning is that the project is defined as an external outcome, determined by a product owner or customer. The students' role in defining the problem goals and in evaluating the success of the results is limited. As in real-world projects, external stakeholders have a say.

Rodrigues and dos Santos (2016) discuss a framework for applying problem-based learning to computing education. They emphasize the need to adhere rigorously to the pre-defined processes of the problem-based learning method. In contrast, other researchers found versions adapted to software engineering were more effective. Richardson and Delaney (2009) successfully applied a hybrid project-based learning approach in an undergraduate software engineering class. They found the approach especially effective in teaching soft skills. Mendes Silva, Goes dos Santos, Ribeiro da Silva, Viera Dias, & Marques da Cunha (2011) found that an adapted version of problem-based learning gave more realism to teaching software engineering.

Perrenet, Bouhuijs, & Smits (2000) noted that while problem-based learning is more directed toward the acquisition of knowledge, Project-based learning is more suited to the application of knowledge. They also found that project tasks more closely mirror professional reality, because a single project often lasts several weeks or months, whereas single problems in problem-based learning are often limited to one week. Woodward, Sendall, & Ceccucci (2010) developed instructional modules based on project-based learning to teach information systems. A combination of experiential learning, cooperative learning strategies and dialog-based analysis of cases were shown to have a positive effect on the development of students' soft skills. The experiments presented here implement project-based learning because the goals in software engineering almost always involve external stakeholders.

Description of the Cooperative Virtual, International Courses

Over the last six years, a number of virtual, international global software engineering courses have been conducted at the Nuremberg Institute of Technology in Germany. Each course was conducted together cooperatively with a partner university in a different country: Mongolia, Mexico and Japan.

Before starting the project phase, students were given introductory lectures about different aspects of global software engineering, such as "international project management," "agile software development," "distributed collaboration tools," and "intercultural communication." After this initial orientation lecture period, students from each group attempted to collaborate on an international group project. A realistic, international software development project was simulated as part of a project-based learning approach. Students participated either in an intensive, all-day block seminar, which ran for seven consecutive days, or a course that ran for 12 weeks during most of one semester. Students were assigned a messy, real-world project, without detailed instructions on how to solve it.

During the first five courses (2012 – 2016), students formed homogenous sub-teams at each site. For example, 20 students in Germany were responsible for the requirements engineering, 20 students in Mexico for the programming. The two teams together developed one software project. Each site selected one person as a communication manager. In 2017, for the first time, heterogeneous teams were formed. Each team was made up of four students in Japan and four students in Germany. These mixed teams then competed against each other to see who could develop the best project.

Communication between the two groups was limited to electronic means: video conferences, chat and e-mail. The teams exchanged documents and computer code using cloud-based project management and collaboration software. Due to the relatively large time differences between Germany and

Mexico (7 hours), Mongolia (7 hours) and Japan (8 hours) students only had about a one hour time window each day when they could communicate in real-time via video conference. All other communication was conducted asynchronously via cloud platforms, messaging and e-mail.

Results

This section describes the results of the cooperative classes taught with Mongolia, Mexico and Japan, in chronological order.

Results from the Mongolia – Germany Experiments

A virtual, team-teaching cooperation between two universities in Germany and Mongolia is described in detail by Beier, Bickel, Brockmann, & Choinzon (2012) and Ende, Lämmermann, Brockmann, & Ayurzana (2013). During the first course, four professors from the Mongolian University of Science and Technology flew to Germany for one week. This initial meeting greatly helped to establish a sense of trust between the cooperating professors. To further aid in communication, a student originally from Mongolia who was currently studying in Germany served as an intercultural coordinator. Without this "intercultural bridge builder," communication would have been difficult, even between the professors.

In addition to the expected geographical and temporal differences, enormous barriers in language and culture presented huge obstacles for the students. Solutions to anticipated problems, such as geographical distance and the eight-hour time difference could be taken into account by scheduling video conferences at a time when it was afternoon in Mongolia and morning in Germany. Unstable internet connections slowed down asynchronous communication via e-mail.

Although the language barrier could be somewhat alleviated by translation software, the cultural barrier proved to be almost insurmountable. The Mongolian students were used to a traditional lecture format, where the professor is seen as a person of authority. German students felt comfortable asking questions and participating in lively, heated discussions. The Mongolian students sometimes viewed this behavior as rude. Although the students on both sides appreciated the opportunity to work together with other students from a very different culture, they felt this proved to be too much of a challenge, even for master's degree students.

Results from the Mexico – Germany Experiments

The next cooperative course with the National Polytechnic Institute in Mexico was conducted for three years in a row, from 2014-2016 (Harrer, Brockmann, & Olivares-Ceja, 2014; Olivares-Ceja, Gutierrez-Sanchez, Brockmann, Kress, & Staufer, 2017). Although Mexico and Germany have two very different cultures, it was hoped that there would be enough common overlap to enable a successful cooperation.

According to Hofstede et al. (2010), Germans have a high value for uncertainty avoidance and feel uncomfortable in ambiguous situations. German students usually expect detailed specifications and clear instructions of what is expected of them. To test this hypothesis, students were intentionally assigned a vague, messy project. This intentional ambiguity proved unsettling for the German students. In contrast, the Mexican students looked forward to taking part in a novel experience. The German students immediately tried to establish clarity. During the first video-call with the Mexican students, the German students concentrated on task-oriented organization. This “business first” approach intimidated the Mexican students, who expected an informal phase of social contact to ease team building.

As the project progressed, the two teams identified what the difficulties in this intercultural collaboration were and tried different approaches to alleviate these problems. Meetings were conducted based on a written discussion agenda, which each group received ahead of time. Because e-mail was often not read or answered, the students agreed that any binding agreements had to be made during video conferences. In contrast to typical German inflexibility, the Mexican students improvised ideas and goals quite agilely. The German students were alarmed by requests for new requirements during the project, which they saw as a violation of the initial project specifications. They had to learn to abandon their strict plans and adapt to the agility of their Mexican group members, whose new ideas were often better.

Not until the end of the semester, after the pressure of grades abated, did the German students finally relax enough to communicate informally with the Mexican students. They asked themselves why they didn’t even know the names of most of their Mexican partners. They realized that they could have saved a lot of time and prevented misunderstandings by first building trust.

Results from the Japan-Germany Experiment

During the winter semester of 2017/2018, the students were first introduced to the topic of Global Software Engineering during an initial orientation lecture. The lecture topics presented the theory and methods central to global software engineering, adaptation of agile software project management methods to distributed teams, intercultural aspects of global software engineering, as well as team-building and conflict management.

One goal in this class was to prevent students from clustering together with other students from their own country. Students were assigned to heterogeneous groups, made up of 3-4 students from Japan and 3-4 students from Germany. The intention was to test whether mixed teams could lessen “in-group” vs. “out-group” conflicts. Students found these mixed, cross-site teams much more challenging. They were also quite surprised that the students from the Japanese university were not originally from Japan. As part of an English language master’s degree program, students were from a number of East Asian countries, such as China, Korea and Vietnam. Having to adapt to a number of different Asian cultures proved even more difficult than planned.

The fact that clear requirements were not delivered at the beginning of the project was unsettling to the students in Germany, who feared the ambiguity and were intent on getting good grades. Students from both countries reported extreme difficulty in communicating with the remote half of their teams. Some team members reported a lack of trust in team members in the other country, since they had never met them before. They also learned to takes minutes of every meeting and to set deadlines for each individual activity and to assign one person as responsible for each task. Next time, they vowed to spend more time getting to know each other at the beginning of the project.

Observed Intercultural Differences

After the project reviews and retrospectives with the 4Ls Method (Gottesdiener, 2010), the team members' behaviour was rated according to Hofstede et al.'s (2010) and Hall's (1990) cultural dimensions. See Table 1.

Table 1

Comparison of Cultural Dimensions in Five Countries, based on Hofstede et al. (2010) and Hall (1990)

Cultural Dimensions	Germany	Japan	Mexico	Mongolia
Power Distance (PD)	low	high	high	high
Individualism (IND)	high	low	low	low
Masculinity (MAS)	high	high	high	high
Uncertainty Avoidance (UA)	high	high	high	high
Long-Term Orientation (LTO)	high	high	low	low
Indulgence (IND)	low	low	high	high
Monochromatic/Polychromatic	mono	mono	poly	poly
High Context / Low Context	low	high	high	high

As evident from Table 1, Germany, Japan, Mexico and Mongolia differ considerably on the cultural dimensions of power distance, individualism, long-term orientation, indulgence and time perception. Masculinity and uncertainty avoidance score high for all four countries. Team members from Germany were the only ones who valued a low power distance, high individualism and low context communication.

The teams of students noticed quite rapidly that they had different views on the power distance between students and professors. Although German students were used to self-organizing and asking professors direct questions, this behavior was considered rude in the other three countries. German students needed to recognize that their ambition to maximize their individual grades might not have been as important as the success of the entire group. The dynamic, short-term flexibility of the Mexican and Mongolian teams was unsettling for the Germans, who are used to long-term planning to minimize uncertainty. The polychromatic time perception of some teams conflicted with the monochromatic German view of time. As the only low context culture, German students were often confused by what they perceived as vague answers from the East Asian students, while the East Asian students sometimes found the directness of the Germans rather rude. Mexican students were disappointed that the German students did not seem interested in interacting with them informally on social media. From a high indulgence and collective culture, Mexican students thought the German students focused solely on the tasks, neglecting the social aspects of the group. At the beginning of the courses, students ranked differences in time zones and languages as the most important factors for global software engineering. At the end of the courses, students ranked cultural differences and trust between teams as the most important factors.

Conclusions

After six years of team-teaching experiments in global software engineering, we can report a number of conclusions. First, students reported that project-based learning was much more challenging than a traditional, instructor-based lecture. German students experienced anxiety due to the ambiguity inherent in the lack of detailed specifications. At the end of the class, a number of students complained of exhaustion. These experiments were conducted with students at the master's level. Although project-based learning could theoretically be used with less experienced participants, the danger of cognitive overload (Kirschner, Sweller, & Clark, 2006) should not be ignored.

Project-based learning had quite a positive effect on students' performance and on their learning success. The virtual course conducted in cooperation with another university in a foreign country allowed students to participate in a realistic simulation of an international project. Students had to deal with real-world problems, such as a seven or eight hour time difference. Project-based learning was especially effective in helping students learn to understand and to communicate better with people from different cultures. The review and retrospectives at the end of each course demonstrated that the students felt they had learned more by taking part in real-world experiences than they would have learned by listening to theoretical lectures.

When students were allowed to form homogeneous, same-site sub-teams in the previous courses, the development of an "in-group vs. out-group" mentality was observed. Forcing students to work in heterogeneous, cross-site teams increased the amount of communication necessary. Although potential for conflicts increased, students felt that they learned more.

Finally, from the point of view of the instructors, organization, informal communication and trust were judged to be the most important factors for conducting a virtual, distributed course. Although the students never got the chance to meet each other "in real life," the instructors were able to meet personally for one week before each course. These personal meetings were essential, not only to discuss class organization. More importantly, these meetings gave instructors the opportunity to get to know each other on an informal basis. This informal communication formed the basis for a level of trust, which is vital for the success of a virtual, cooperative course. Other researchers also confirm that trust and a good relationship are vital for collaborations (Hussain & Blincoe, 2016).

Acknowledgements

This research was supported by a grant from the Nuremberg Institute of Technology for the research project "DiverSenta."

References

Barrows, H., (1996). Problem-based learning in medicine and beyond: A brief overview. In L. Wilkerson, & H. Fiselaers (Eds.), *Bringing problem-based learning to higher education: Theory and practice*. San Francisco, CA: Jossey-Bass.

Beecham, S., Clear, T., Barr, J., Daniels, M., Oudshoorn, M., & Noll, J. (2017). Preparing tomorrow's software engineers for work in a global environment. *IEEE Software*, *34*(1), 9-12.

Beier, S., Bickel, M., Brockmann, P., & Choinzon, M. (2012). It takes a global village to teach global software engineering. *Proceedings ICEEE International Conference on E-Learning and E-Technologies in Education* (pp.152-157). Lodz, Poland: IEEE.

Carmel, E., Espinosa, J. A., & Dubinsky, Y. (2010). "Follow the sun" workflow in global software development. *Journal of Management Information Systems, 27*(1), 17-34.

Clear, T., Beecham, S., Barr, J., Daniels, M., McDermott, R., Oudshoorn, M., Avickaite, A., & Noll, J. (2015). Challenges and recommendations for the conduct of global software engineering courses: A systematic review. *Proceedings ACM 20th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 1-39). Vilnius, Lithuania: ACM.

Ende, M., Lämmermann, R., Brockmann, P., & Ayurzana, G. (2013). A virtual, global classroom to teach global software engineering: A Mongolian-German team-teaching project. *Proceedings ICEEE International Conference on E-Learning and E-Technologies in Education* (pp. 229-233). Lodz, Poland: IEEE.

Gottesdiener, E. (2010). The 4L's, a retrospective techniqu*e* (Web log comment). Retrieved from https://www.ebgconsulting.com/blog/the-4ls-a-retrospective-technique/.

Hall, E. T. (1990). *Understanding cultural differences, Germans, French and Americans*. Yarmouth, ME: Intercultural Press.

Harrer, M., Brockmann, P., & Olivares-Ceja, J. M. (2014). Teaching cultural aspects of global software engineering: A virtual Mexican-German team-teaching experience. *Proceedings European Conference on Software Engineering Education* (pp.156-166). Herzogenrath, Germany: Shaker.

Herbsleb, J. D., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, *29*(6), 481-494.

Hoda, R., Babar, M., & Shastri, Y. (2017). Socio-cultural challenges in global software engineering education. *IEEE Transactions on Education*, *60*(3), 173-182.

Hofstede, G., Hofstede, G. J., & Minkov, M., (2010). *Cultures and organizations: Software of the mind* (3rd ed.). New York, NY: McGraw-Hill.

Hussain, W., & Blincoe, K. (2016). Establishing trust and relationships through video conferencing in virtual collaborations: An experience report on a global software engineering course. *Proceedings 11th International Conference on Global Software Engineering* (pp. 49-54). Retrieved from http://doi.ieeecomputersociety.org/10.1109/ICGSEW.2016.14

Kirschner P. A., Sweller, J., & Clark, R. E., (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist 41*(2), 75-86.

MacGregor, E., Hsieh, Y., & Kruchten, P. (2005). The impact of intercultural factors on global software development. *Proceedings Canadian Conference on Electrical and Computer Engineering, 2005* (pp. 920-926). Saskatoon, Canada: IEEE.

Mendes Silva, M., Goes dos Santos, S. R., Ribeiro da Silva, J. C., Viera Dias, L.A., & Marques da Cunha, A. (2011). Problem-based learning a practical approach for software engineering interdisciplinary teaching. *Proceedings of the 8th International Conference on Information Technology: New Generations (INTG), 2011* (pp. 1046-1047). Las Vegas, NV: IEEE.

Nguyen, P., Babar, M., & Verner, J. (2006). Critical factors in establishing and maintaining trust in software outsourcing relationships. *Proceedings of the 28th International Conference on Software Engineering* (626-627). New York, NY: ACM.

Olivares-Ceja, J. M., Gutierrez-Sanchez, B., Brockmann, P., Kress, A., & Staufer, J. (2017). Project-based learning in an international virtual classroom to teach global software engineering. *Proceedings - 9th International Conference on Education and New Learning Technologies (EDULEARN17)* (pp. 6263-6273). Barcelona, Spain: IATED.

Perrenet, J. C., Bouhuijs, P.A. J., & Smits, J. G. M. M. (2000). The suitability of problem-based learning for engineering education: Theory and practice. *Teaching in Higher Education*, *5*(3), 345-358.

Richardson, I., & Delaney, Y. (2009). Problem-based learning in the software engineering classroom. *Proceedings of the 22nd Conference on Software Engineering Education and Training* (pp. 174-181). Hyderabad, India: IEEE.

Rodrigues, A., & dos Santos, S. (2016). A framework for applying problem-based learning to Computing Education. *IEEE Frontiers in Education Conference (FIE)*. Erie, PA: IEEE.

Sangwan, R., Bass, M., Mullick, N., Paulish, D., & Kazmeier, J. (2007). *Global software development handbook*. New York, NY: Auerbach Publications.

Savery, J. R. (2006). Overview of problem-based learning: Definitions and distinctions. *Interdisciplinary Journal of Problem-Based Learning*, *1*(1), 9-20.

Sosa, M., Eppinger, S., Pich, M., McKendrick, D., & Stout, S. (2002). Factors that influence technical communication in distributed product development: An empirical study in the telecommunications industry. *IEEE Transactions on Engineering Management, 49*(1), 45-58.

Woodward, B. S., Sendall, P., & Ceccucci, W. (2010). Integrating soft skill competencies through project-based learning across the information systems curriculum. *Information Systems Education Journal*, *8*(8). Retrieved from http://isedj.org/8/8/.

Author Details

Patricia Brockmann
Patricia.Brockmann@th-nuernberg.de