An Intercultural Engineering Module for Software Engineers

Nicole Hornbrook
Purdue University
hornbroo@purdue.edu

James C. Davis
Purdue University
davisjam@purdue.edu

ABSTRACT

The world is continuing to intertwine more and more because of technology, and, as a result, it is becoming more important to develop software products that are culturally inclusive and acceptable. It is important to educate future software engineers about intercultural engineering, both in terms of requirements and in terms of engineering collaboration. In this poster we discuss a learning module for a software engineering course on Intercultural Engineering. We describe three case studies for use in assignments or in-class discussion.

KEYWORDS

Intercultural learning; Software engineering

1 INTRODUCTION / PROBLEM

Computer software plays an integral role in realizing technical solutions to business and societal problems. However, if the engineering process does not take intercultural issues into consideration, then engineering solutions may lack cultural inclusivity. A technology may solve an issue in one society but not another. As the world continues to intertwine more and more because of technology, it is more important now than ever to develop software products that are culturally inclusive and acceptable.

To bridge this gap between culture and engineering, we are developing an intercultural engineering module for use in an upper-level software engineering course. Our goal is to make future software engineers aware of different mindsets and different requirements that arise during intercultural engineering. Our approach is to present students with case studies on cultural differences that affect software engineering, and the solutions proposed by engineers and researchers.

2 BACKGROUND and RELATED WORK

Geert Hofstede's cultural framework is useful for analyzing and describing observable cultural differences; therefore, we require it in our study of culture and software engineering as it is widely used and understandable. Hofstede acquired a large survey database from IBM where he came up with six cultural dimensions [1] based on a culture's tendencies, values and preferences. These dimensions include power distance, uncertainty avoidance, individualism, masculinity, long term orientation, and indulgence.

Many studies have been conducted on teaching software engineering students to work collaboratively with international teammates using globally distributed projects. Altin et al. [2] for example, conducted a course where the participants completed a project with students from a university in Turkey. Throughout the semester, the students collaborated virtually in different time zones and came together two times in Turkey. Clear et al. [3] mentions many of these types of courses such as [4], [5], [6], and [7] in their literature search. Clear et al. found that international teams in educational settings typically struggle with time-zone, trust, and communication issues. Though this method is effective in training students for working with other cultures, collaborating with other schools and traveling internationally is out of the scope of our project.

Other researchers have similarly attempted to educate the next generation of software engineers with culture inclusive course materials. Jesiek et al. [8] came up with realistic scenarios to prepare students for working in multicultural teams in the Global Engineering Competency (GEC) project. Each scenario depicts a real-world engineering issue within the context of working with international managers and collogues. The scenarios can be used as assessments or discussions. The researchers' goal in this project is

similar to ours; however, it is more general as it is directed to students in many different fields of engineering. Therefore, the project contains only a few scenarios relating to software engineering. In addition, the GEC project focusses on dealing with engineers from other cultures, where we hope to also equip students to build software for those of a different culture.

Furthermore, some researchers have focused more on in-class exercises to simulate international teamwork. Li et al. [9] created a classroom exercise to simulate teamwork across different time-zones. In the exercise, each section of the class represented a different location with a different time-zone. The students had to complete a project within 1.5 hours with the given challenges of communication. Davis et al. [10] used a classroom exercise to simulate communication challenges when interacting with various cultures. Each group of students were assigned a room with a specific culture and were to visit other rooms and adapt to the other room's culture while also keeping their own. Clear et al. also mentioned a couple studies using in-class exercises ([11], [12]) attempting to simulate global software engineering collaboration. These exercises similarly focused on communication, culture, and time-zone related issues. Though we also discuss the issues of international teamwork in our case studies, unlike these other exercises and courses we also discuss how culture affects the end-product itself as different cultures may have different requirements.

3 OVERVIEW / METHODS

To begin our research, we conducted a literature search to find some cultural issues relating to software that have already been investigated and potentially resolved. At first, we did a general overview of the IEEE and ACM conferences with search terms such as "culture and software engineering", "culture inclusivity", and "religion and culture".

Later, we searched more specifically through individual conferences for any papers that cited Geert Hofstede. We then visualized our search by creating a table of conferences and the number of proceedings they had citing Hofstede. By doing this, we were able to get a general idea of where the most research connecting culture and software engineering was being conducted.

4 RESULTS

After a systematic literature search, we created three case-studies on some of the most interesting topics we found, accompanied by discussion questions to facilitate reflection. We summarize each case study next.

User interfaces

This case study discusses a few software solutions to cultural barriers and preferences using Hofstede's six cultural dimensions. Specifically, the case study explores how user interface preferences can differ between cultures. First, we provided a background to Hofstede's study and included a table comparing Hofstede's scores between the USA and China. We then mainly focused on a program called "Modeling Culture for Cultural Adaptivity" (MOCCA) [13] that generated personalized user interfaces based on the countries the user had lived in. In the study, we summarize how the MOCCA can generate three different versions of eight interface aspects. These versions are determined based on if the countries score high medium or low on specific Hofstede dimensions. The case study ends with discussion questions about the MOCCA, using Hofstede's dimensions beyond user interfaces, and gender inclusivity.

Intercultural teamwork

This next case study discusses common issues that arise in multicultural software developing teams. The case study first analyzes team conflict using stereotypes, and then discusses the issues through the lens of Hofstede's cultural dimensions. The study ends with some discussion questions that touch on the topics of time, stereotypes, and trust.

KosherOS

This study focuses on how religious requirements can also impact software engineering as religion is a part of culture. First, the study introduces the topic of Orthodox Judaism and briefly mentions some software solutions to their Sabbath requirements, such as home automation [14], to make the laws easier to obey. Next, the case-study discusses a kosher solution developed by SafeTelecom called the KosherOS and summarizes an interview we conducted with their senior developer. In the interview, the senior developer explained the

purpose and design of KosherOS, what kinds of applications KosherOS support, and technical aspects of the implementation. Finally, we provide discussion questions relating to KosherOS, kosher cellphones, and culturing technology in general.

5 FUTURE WORK

The case studies will potentially be used in the Intercultural Engineering module to stimulate the students' creativity and critical thinking skills. Each case-study ends with a set of discussion questions the students are to answer and discuss with one another. The hope is that after reading and completing the questions of the case studies, the students will become more aware of current cultural needs and will be able to identify other cultural issues and develop creative solutions for them.

As seen in our study of the KosherOS, culture can definitely influence the technical requirements of software products; however, our Hofstede-specific literature search reflected that more research has been conducted on the impact of culture on the teamwork aspect of engineering than the software aspect. Culture can impact every step of the development process; therefore, there is much room for more research in the developmental stage of software engineering. Continuing this project, we plan on searching for more products like the KosherOS and will look out for more culture requirements that impact the technical aspects of software engineering.

ACKNOWLEDGMENTS

This project is supported by a grant from the Purdue VEIL program.

REFERENCES

[1] G. Hofstede, "Dimensionalizing Cultures: The Hofstede Model in Context," *Online Read. Psychol. Cult.*, vol. 2, no. 1, 2011, doi: 10.9707/2307-0919.1014.

[2] R. Altin *et al.*, "Working across time zones in cross-cultural student teams," *Proc. Conf. Integrating Technol. Comput. Sci. Educ. ITiCSE*, p. 360, 2009, doi: 10.1145/1562877.1563002.

[3] T. Clear *et al.*, "Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review," in *Proceedings of the 2015 ITiCSE on Working Group Reports*, Vilnius Lithuania, Jul. 2015, pp. 1–39. doi: 10.1145/2858796.2858797.

[4] J. Favela and F. Pena-Mora, "An experience in collaborative software engineering education," *IEEE Softw.*, vol. 18, no. 2, pp. 47–53, Apr. 2001, doi: 10.1109/52.914742.

[5] O. Gotel, C. Scharff, and S. Seng, "Preparing Computer Science Students for Global Software Development," in *Proceedings. Frontiers in Education. 36th Annual Conference*, San Diego, CA, USA, 2006, pp. 9–14. doi: 10.1109/FIE.2006.322632.

[6] B. Bruegge, A. H. Dutoit, R. Kobylinski, and G. Teubner, "Transatlantic project courses in a university environment," in *Proceedings Seventh Asia-Pacific Software Engeering Conference. APSEC 2000*, Singapore, 2000, pp. 30–37. doi: 10.1109/APSEC.2000.896680.

[7] E. Doerry, R. Klempous, J. Nikodem, and W. Paetzold, "Virtual student exchange: lessons learned in virtual international teaming in interdisciplinary design educatior," in *Information Technology Based Proceedings of the FIfth International Conference onHigher Education and Training, 2004. ITHET 2004.*, Isanbul, Turkey, 2004, pp. 650–655. doi: 10.1109/ITHET.2004.1358252.

[8] B. K. Jesiek, N. T. Buswell, and Q. Zhu, "Global Engineering Competency: Assessment Tools and Training Strategies," *ASEE Annu. Conf. Expo. Conf. Proc.*, 2018, doi: 10.18260/1-2--30559.

[9] Y. Li, S. Krusche, C. Lescher, and B. Bruegge, "Teaching Global Software Engineering by Simulating a Global Project in the Classroom," *SIGCSE 2016 - Proc. 47th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 187–192, 2016, doi: 10.1145/2839509.2844618.

[10] K. A. Davis, A. R. Taylor, D. Reeping, H. G. Murzi, and D. B. Knight, "Experiencing Cross-Cultural Communication on a Home Campus: Exploring Student Experiences in a Cultural Simulation Activity," *J. Excell. Coll. Teach.*, vol. 30, no. 4, pp. 187–214, 2019.

[11] R. van Solingen, K. Dullemond, and B. van Gameren, "Evaluating the Effectiveness of Board Game Usage to Teach GSE Dynamics," in *2011 IEEE Sixth International Conference on Global Software Engineering*, Helsinki, Aug. 2011, pp. 166–175. doi: 10.1109/ICGSE.2011.24.

[12] C. Lescher, Y. Li, and B. Bruegge, "Teaching Global Software Engineering: Interactive Exercises for the Classroom," in *2014 IEEE 9th International Conference on Global Software Engineering*, Shanghai, China, 2014, pp. 163–172. doi: 10.1109/ICGSE.2014.14.

[13] Harvard School of Engineering and Applied Sciences, K. Reinecke, A. Bernstein, and University of Zurich, "Knowing What a User Likes: A Design Science Approach to Interfaces that Automatically Adapt to Culture," *MIS Q.*, vol. 37, no. 2, pp. 427–453, Feb. 2013, doi: 10.25300/MISQ/2013/37.2.06.

[14] A. Woodruff, S. Augustin, and B. Foucault, "Sabbath day home automation: 'it's like mixing technology and religion,'" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*, San Jose, California, USA, 2007, pp. 527–536. doi: 10.1145/1240624.1240710.