

# Taking Global Software Development from Industry to University and Back Again

Igor Cavrak  
University of Zagreb  
Faculty of Electrical Engineering and  
Computing  
HR-10000 Zagreb, Croatia  
+385 1 6129 861  
igor.cavrak@fer.hr  
<http://www.rasip.fer.hr/icavrak>

Rikard Land  
Mälardalen University  
Department of Computer Science and  
Engineering  
PO Box 883, SE-721 23 Västerås, Sweden  
+46 21 10 70 35  
rikard.land@mdh.se  
<http://www.idt.mdh.se/~rld>

## Abstract

*As global software development (GSD) establishes itself as a software engineering practice, it should be taught to the software developers of the future. But can the problems of GSD really be transferred to the controlled environment of a university course? Will the students attending the course be able to cope with the problems associated with GSD as professionals? This paper elaborates on these questions by presenting approaches, methods, and goals in a planned GSD university course.*

## 1. Introduction

Universities should equip computer science students with not only theory and technology skills but also knowledge in the engineering environment they will be faced with as future software engineers. Many such issues are addressed by specific courses in software engineering, often organized as practical projects in which the students will have to tackle “real” problems [1,2,5,6,7,8,9,12]. But as global software development (GSD) is becoming widespread in today’s international enterprises and virtual organizations [3,10], universities should incorporate this trend into their curriculum [11]. To our knowledge, university courses or student projects involving global software development are very rare, and are restricted either to existing software engineering courses [2] or case studies [1]. Perhaps the problem to give a course in this area is due to the inherent properties of global software development: it is difficult to present a university course that faces the students with global development specifics such as telephone- and videoconferencing, distributed configuration management, using a foreign language (English), coping with another culture, and collaborating

across time zones. Despite this, we have accepted the challenge and are developing a course in distributed software development, to be simultaneously held in Västerås, Sweden, and Zagreb, Croatia.

## 2. Course Description

The students will be introduced to the problems of GSD through lectures and/or self-studies but the larger part of the course will consist of a joint software development project. In the project, the students will encounter many GSD problems but not all: Sweden and Croatia are e.g. located in the same time zone. Besides combining theory and practice through lectures and the joint project, the problem domain addressed by the project will as well be in the GSD domain: we plan to develop and/or enhance tools to support GSD. Each time the course is held, experiences and suggestions of more product features will be “inherited” from the previous year. In this way requirements and products are “bootstrapped” along subsequent courses.

### 2.1. Relation to Community Development

As products will be primarily based on free software (due to the financial reality), we plan to submit course products back to the public community thus employing a large number of users in testing the product usability and quality, and gathering feedback to use in next year’s requirements. Maintenance of the product will be (at least partially) ensured by assigning product-related graduation thesis to some of the students participating in the project.

The type of development the students will perform only vaguely resembles community development: although the work is distributed and somewhat individualized, there will be a much stronger project

management, and some of the developers will meet physically.

## 2.2. Relation to GSD in Industry

If the course involves a real customer, he will presumably invest both time and money [2], enabling both a real-world problem and traveling possibilities [2]. However, there is a risk that focus will be more on the end product than on the educational elements of the project.

We have therefore chosen to avoid industrial involvement in favor of a slight shift towards community software development. This means that we can focus on education, but it also means that we will have to use low-cost communication media and will not be able to let the students travel. Especially the latter is a great disadvantage, since being able to meet face to face has been emphasized as the most important means to enable successful GSD [3,4,10]. However, by the start of the course some of the course leaders will have worked together for six months, which will help alleviating the cultural and physical distance [2,3,4]. Also, since this is a university course, should the worst happen and there is no product at all at the end of the course, this is evidence that the students encountered problems – which is educational (at least to some extent).

The students will form groups at each university and work part of the time together, as is the case in the office environment at most companies, but will also work at different hours and locations (due to their other obligations and us not providing personal workplaces at the university).

## 2.3. University Specifics

There are other university environment specifics as well such as students having different background and experiences, adjusting the course to the existing curriculum at two universities (which can present a major problem as duration of the course, start and end dates must be coordinated [2]). Each university also has a responsibility towards its own students in the first place – how should students working on the same project/product be graded in different rating systems? And what happens if there are not enough students at one of the universities involved? A careful balance must be struck not to overwhelm students with project work on one of the sites due to differences in obligations regarding the number of other courses held at their universities.

## 3. Main Issues

We can discern two major challenges for our course. First, how do you transfer the problem domain of GSD

from industry environment to a university course? Second, how useful will the outcomes of the course be to industry: solutions and software professionals?

## 4. References

- [1] Brereton P., Lees S., Bedson R., Boldyreff C., Drummond S., Layzell P., Macaulay L., Young R., “Student Collaboration across universities: A Case Study in Software Engineering”, *Proceedings of 13<sup>th</sup> Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 2000.
- [2] Bruegge, B., Dutoit, A.H., Kobylinski, R. and Teubner, G. “Transatlantic project course in a university environment”, *Proceedings of 7<sup>th</sup> Asia-Pacific Software Engineering Conference (APSEC)*, 2000.
- [3] Carmel E., *Global Software Teams: Collaborating Across Borders and Time Zones*, ISBN 0-1392-4218-X, Prentice-Hall, Upper Saddle River, NJ, 1998.
- [4] Carmel E., Agarwal R., “Tactical Approaches for Alleviating Distance in Global Software Development”, *IEEE Software*, volume 18, issue 2, IEEE, 2001
- [5] Crnkovic I., Larsson M., and Lüders F., “Implementation of a Software Engineering Course for Computer Science Students”, *Proceedings of 7<sup>th</sup> Asia-Pacific Software Engineering Conference (APSEC)*, 2000.
- [6] Crnkovic I., Land R., and Sjögren A., “Is Software Engineering Training Enough for Software Engineers?”, *Proceedings of 16<sup>th</sup> Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 2003.
- [7] Daniels M., Faulkner X., and Newman I., “Open ended group projects, motivating students and preparing them for the ‘real world’”, *Proceedings of 15<sup>th</sup> Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 2002.
- [8] Dawson R.J., Newsham R. W., and Fernley B. W., “Bringing the ‘real world’ of software engineering to university undergraduate courses”, *IEE Proceedings In Software Engineering*, volume 144, issue 5, 1997.
- [9] Dawson R., “Twenty Dirty Tricks to Train Software Engineers”, *Proceedings of 22<sup>nd</sup> International Conference on Software Engineering (ICSE)*, ACM, 2000.
- [10] Karolak D., *Global Software Development: Managing Virtual Teams and Environments*, ISBN 0-8186-8701-0, IEEE Computer Society Press, Los Alamitos, CA, 1998.
- [11] Shaw M., “Software Engineering Education: A Roadmap”, *Proceedings of the 22<sup>nd</sup> International Conference on Software Engineering*, ACM Press, New York, NY, 2000.
- [12] Wohlin C. and Regnell B., “Achieving industrial relevance in software engineering education”, *Proceedings of 12<sup>th</sup> Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 1999