Integrating team projects into the SE Curriculum

KRISTINA LAPIN, SAULIUS RAGAISIS Software Engineering Department Faculty of Mathematics and Informatics Vilnius University Naugarduko 24, LT-03225 Vilnius LITHUANIA {kristina.lapin, saulius.ragaisis}@mif.vu.lt_http://www.mif.vu.lt/se/

Abstract: - This paper reports the experience of developing and modifying a Software Engineering study program in Vilnius University, Lithuania. The aim of a new curriculum is improving the collaboration with industry and emphasizing the team projects. Industry partners participated in defining both, the outcomes of the curriculum and aspects that should be emphasized. The evolution of team projects' implementation has resulted in three stages approach: introductory internal university team project for the second year students, in-home simulation of the real world projects for the third year students and the projects provided by industry for the last year undergraduates. The initially stand-alone HCI course was integrated with team projects in which students are expected to apply acquired HCI skills. The intent of our SE curriculum is developing extended software engineering skills and in-depth understanding of the user-centred design techniques.

Key-Words: - SE education, industry-university collaboration, team projects, integration of courses.

1 Introduction

New Software Engineering (SE) Department was established in the Faculty of Mathematics and Informatics at Vilnius University (VU) on purpose develop the new Software Engineering to Curriculum. The implementation of the curriculum started at 2001. Currently we gained experience with two crops of SE undergraduates. This paper discusses the reasons for developing and implementing the new curriculum and outlines the modifications after we gained the first experience with a new curriculum.

Further we address the motivation for development of the new software engineering curriculum. We deal with curriculum improvements. Next we summarize our experience of collaborating with industry and analyze how the collaboration influenced curriculum enhancements.

2 Motivation for the new curriculum

Software Engineering (SE) was taught as 1-2 courses in Computer Science (CS) study program. Such courses were introduced at Vilnius University (VU) about 20 years ago. After completing the CS programme, students possess strong development skills, e.g. programming, data structures, and databases.

However, such approach to education of software professionals is not satisfactory: graduates do not receive the knowledge and skills needed for industrial software development. It results in low quality and unusable software systems.

Students are weak in both visual design and understanding of user-centred design process. They tend to approach design and analysis from system developer perspective, rather from user's perspective [1, 2]. User perspective was involved, dealing with interface design in software engineering course. Students did not have a possibility to experience working with real users that have different needs and perceptions from them [3]. CS study programme focused on individual contributions rather than on group efforts [4].

It was understood that "Software Engineering Programmes Are Not Computer Science Programmes" [5] and as a first step VU has started Software Engineering track in CS study program since 1997. Initially the track contained 4 additional SE courses on requirements analysis, design, project management, and software process. Later on, the team project in collaboration with industry was introduced.

In 2002, the first students were accepted into separate undergraduate Software Engineering study programme. The development of SE curriculum was based on:

- many years of experience providing CS education and particular SE courses;
- university staff experience in a variety of industrial software projects for Lithuanian and foreign customers that is very important for software engineering education [6];
- regulations of Lithuanian Ministry of Science and Education and Vilnius University;
- recommendations of international professional organizations ACM, IEEE and authorities, e.g. [5, 7], and Software Engineering Body of Knowledge under development [8];
- requirements of accreditation boards for study programs in computing CSAB and ABET [9, 10];
- a lack of vertical and horizontal integration of the individual courses throughout the old CS curriculum. The courses were poorly coordinated, some material repeated in the different courses. Students lacked the confidence how to apply skills learned in a course to new situations. Practical exercises were poorly coordinated and integrated with theoretical material.
- a need to accommodate the new modern courses, such as human-computer interaction, bioinformatics, project management and others;
- a need to take advantage of various teaching technologies such as project-based learning, cooperation with industry;
- industry concerns and students demand.

Industry expectations helped defining the main curriculum outcomes and the emphasized aspects. Management of Lithuanian IT companies has qualified the graduates of our CS study programme as enough professional software developers. At the same time they indicated the lack of some knowledge and skills important for industrial projects. Their comments and wishes are expressed in different words but in fact they state the need of true software professionals that undertake the entire development systems life cycle, integrate competencies of programmer, designer, system analyst and project manager, and apply sound methods for each role. They stress that software engineers need skills of teamwork, communication with customers, application area analysis, and project management. The project planning skills are estimated as the very important factor for the team project success [11]. Especially representatives of foreign IT companies emphasized this opinion that conforms to Peter Denning observations [12].

Developing new software engineering the curriculum, the need to improve the practical skills of working in software development group and the user-centred education was recognised [1, 2, 13, 14, 15]. Students have to receive a special training on how to function collaboratively [16]. The new curriculum should have a substantial team project component but simply assigning more team projects is not sufficient in addressing the need for students to learn teamwork skills [17]. The special courses, such as Team Software Process and Team Project, should be delivered and supervised by teachers in order to train students how to work in team. Preparing undergraduates for the transition to industry, industry partners provide for faculty projects both real industry tasks and communication with users [18, 19].

Introducing the team projects we discussed with our industry partners about team size. Lithuanian IT companies are not big; the most industry projects are performed by teams up to 8-10 members. Therefore the goal of our curriculum should be the ability to function in such groups. Assessing the team project results, the process quality should be measured as well as the quality of produced product. Younessi and Grant suggest using CMM for the process evaluations [20].

In our opinion team size should increase gradually. In the second study year project team consists of four members, in third year project team is formed up to six members, in the last year team is composed of 7- 8 members.

3 Curriculum improvements

Developed software engineering curriculum was the result of employer's expectations and academy realisation abilities [21]. Regarding the need to improve collaboration skills and industry expectations, the following courses were included in the new curriculum:

- SE subjects: Personal Software Process (PSP), Team Software Process (TSP), Human-Computer Interaction (HCI), Team Project I (TP1), Software Process Management, Team Project II;
- Humanities and Social Sciences: Communication Skills, Professionalism and Ethics, Introduction to Psychology, Fundamentals of Management.

The first study year contains the individual Computing Fundamentals and SE courses (Fig. 1). The horizontal bands within the following figures represent respectively autumn and spring semesters. The second year introduces team projects in SE course that lasts two semesters. The introductory group work skills are acquired in this course. The team in this project consist of 4 students. This is internal university project where instructor is both guiding the students work and defining the desired outcome.

The teaching of the third semester courses was modified after the first crop of our undergraduates completed the first internship. After that internship, industry partners pointed out the real problem with the organisation of team work in student projects. They indicated the lack of practical experiences in communication and collaboration both inside the team and with people in industry. One of the biggest IT company in Lithuania "Blue Bridge" suggested introducing the intermediate project before the projects in industry. Such project should be conducted under supervision of department but with user and customer from this company. Therefore third study year was augmented with TP1 that was integrated with Team Software Process course.

The first internship highlighted also another problem. Despite of the introduced HCI course, team project outcomes have not improved because many students limited the use of acquired knowledge only to HCI assignments [23]. We understood that HCI course should be also integrated with Team Software Process and Team Project I (Fig. 3). In TSP students acquire skills in software-process engineering [22], while HCI emphasize how to integrate user-centred design techniques to the software process.

The industry partner defines the desired outcome and acts as a customer and user, while the HCI and TSP instructors guide the team works. Third year project team consists of 6 students.



Fig. 1. Computing fundamentals and SE (in Bold) obligatory courses delivered in 1st study year







Fig. 3. Computing fundamentals and SE (in Bold) obligatory courses delivered in 3rd study year



Fig. 4. Computing fundamentals and SE (in Bold) obligatory courses delivered in 4th study year

The lectures of TSP and HCI are delivered separately. Integrating HCI into existing computing curricula, the two approaches can be applied:

- a separate introductory HCI course [23, 24], or
- HCI topics incorporated in existing courses [25, 26, 27].

IS/EC curriculum [14] combines these approaches:

- separate course provides an overview of the user-centred development process with emphasis on Web site design, prototyping, usability testing, and special consideration for the novice user;
- specific deeper analysed topics in appropriate existing courses, such as Web Usability topic in the course of Design and strategies for e-commerce.

Our SE curriculum also combines these approaches, but otherwise:

- HCI and TSP courses have separate lectures;
- the assignments are performed in the frame of TP1: TSP course covers general project activities, HCI exercises incorporate the user-centred techniques into the team activities;
- the task is provided by industry partner that participates in team projects activities as the user and the customer.

Integrating the HCI assignments with TP1 motivates students to apply acquired knowledge into software engineering projects. Students gain a possibility of working with real users and their notion of what is "common sense" or "obvious" will significantly change [25].

Before the integration of HCI with the TP1, one team member rather than the whole team often designed the user interface. HCI assignments allow avoiding such behavior. After introduction of HCI, the HCI assignments motivate students to apply user-centered design techniques before any coding for the project assignments. Performing HCI assignments, students fulfill two iterations with creation and evaluation of low and high fidelity prototypes. In this case students learn to incorporate iterative user-centered techniques into the projects that they perform in other courses.

We expect that after applying HCI practices to their first project, they will apply them and to the subsequent projects. Such expectations are accomplished also in the grading of students projects. The HCI grade is partially included into the SE project grade.

The final team project is integrated with internship in the industry environment (Fig. 4). In the final project team consists of up to 8 members. The quality of the project outcomes as well as project process attributes is evaluated by industry partners and university supervisors.

4 Experience of the collaboration with industry partners

For preparing students to be effective professional software engineers, working experience in an industrial context is required. Students profit from tours of industry, seeing the information systems at work. They are hearing computing professionals give presentations and answer their questions about various applications and careers in computing [2].

The gap between the industry and academia can be bridged through steady close cooperation with one big industry partner [28]. We have about 120 SE students per year. Organising student internships, Vilnius University collaborates with about twenty partners (Table 1).

Spring 2005		
Industry	Number	Percentage
partners		
IT Companies	13	68,5 %

2

4

10,5 %

21.0 %

Table 1. Distribution of industry partners inSpring 2005

Banks

Other

Initially software engineering study programme offered one team project course in which students performed internships in the industry environment. Discussing with industry partners the experience of the first internship, project director of one of the leading Lithuanian IT companies "Blue Bridge" first software company in Lithuania certified according ISO 9000 - suggested introducing one more team project course. This company supports TP1 by both formulating the task akin to realistic industry problem and acting as a customer and user. Current curriculum offers two team project courses. Both project courses collaborate with industry partners but in different ways. TP1 involves industry partner as a customer, project activities are supervised by TSP and HCI lecturers. It is like pilot project before the final internship in industry that is fulfilled in the frame of Team Project II.

First implementation of TP1 (without HCI) took place in the autumn semester 2003. Team project was integrated then with TSP lectures. This implementation highlighted problems that caused integration of HCI course with TP1. The lectures were rearranged. User-centred development techniques are treated deeper than other HCI aspects. The focal attention is paid for placing usercentred development techniques in the software engineering life cycle.

The next internships that took place next year showed the success of the enhancements. Industry partners stated that student skills improved comparing with earlier internships. This opinion can prove the marks proposed by our customers (Fig. 5) that was even better than the marks which we gave assessing student internship outcomes (Fig. 6). Comparing with earlier criticism the opinion of our customers is significantly better.

5 Discussion

Integration of third year courses improves the teaching quality. Horizontal coordination of three courses decreases student workload. However such integration requires more organisational efforts from the teaching staff than separate courses. Team

registration must be one for all integrated courses. Integration is successful if the crop of students in integrated courses is the same. This opinion coincided with an experience of prof. Franz Kurfess from California Polytechnic State University who visited Vilnius University in Spring, 2008. He made a point about an integration of HCI and SE courses. He teaches HCI and SE courses and also tried to integrate labs of these courses. According to his opinion the integration was not successful because part of students chose only one of these courses.

It could be noted that enough similar approach is used in the SE curriculum at the University of Melbourne [29]: a core component of the curriculum is a pair of team projects conducted with industry in 3rd and 4th study years.



Fig. 5. Distribution of marks proposed by customers (hosting companies) in Spring 2005



Fig. 6. Distribution of final marks for Team Project II performed in Spring 2005

6 Conclusions

This paper describes the SE curriculum enhancements to provide new undergraduate students with a better preparation for the real world. In our opinion, the following curriculum enhancements determined the good results:

- introduction of the internal university team projects since 3rd semester;
- team project courses are performed each year, from the second to forth;
- team size grows in the subsequent projects gradually, from 4 students in the second year project to 6-8 in the last year;
- integrating former stand-alone HCI course with Team Software Process and Team Project I into one team project train students to apply their skill in real software development projects;
- introduction of the courses of humanities and social sciences improved the communication and management skills;
- introduction of the intermediate third year team project under the supervision of the department, but with a user and customer from industry trains the communication skills within industry environment;
- horizontal integration third year stand-alone courses to one third year team project reduced the huge amount of practical works from three separate assignments to one team project;
- vertical coordination of the team projects allow gradually increasing team size and degree of collaboration with industry across curriculum;
- special courses train the abilities to function in a group.

In order to train software engineers who place value in creating usable software, both department and industry takes their part. Department prepares software engineers by training them to use usercentered techniques at every phase of development. Students are expected to apply HCI concepts in the team projects. Collaboration with industry provides feedback for curriculum enhancements. Industry influences the introduction of team projects to our curriculum by formulating the project tasks, participating in the final presentation and assessing the produced product.

The team project courses are designed to hone the practical skills of each student to function better as a member of team, to understand the team work and to communicate with a user.

The industry partners stated that after the Team Project I, students are better prepared to perform their final real projects in industry. The representatives from software development companies deemed both team projects to be extremely successful. Their opinion was that curriculum enhancements highly improved students' skills of performing group works.

References:

- [1] A.N. Ribeiro, J.C. Campos, F.M. Martins, Integrating HCI into a Software Engineering course, *Proceedings of the HCI Educators* 2007, 2007, pp.49-58.
- [2] D.K. Lidtke, G.E. Stokes, Information Systems-Centric Curriculum ISCC'99, *Journal* of Systems and Software, Elsevier Sciences, Vol.49, No.2, 1999, pp. 191-175.
- [3] A. Dix, Personal position statement on HCI situation and training, Proceedings of Workshop on Effective Training and Education for HCI, 1998, www.comp.lancs.ac.uk/computing/users/ dixa/papers/HCI-educ98/
- [4] T.B. Hilburn, D.J. Bagert, A Software Engineering Curriculum Model, *Proceedings of the 1999 ASEE Annual Conference*, ASEE, 1999. Available online:

citeseer.ist.psu.edu/hilburn99software.html

- [5] D.L. Parnas, Software Engineering Programmes Are Not Computer Science Programmes, Annals of Software Engineering, Vol.6, Is.1-4, 1999, pp. 19-37.
- [6] G. Karen, Why do Universities Fail Teaching Software Engineering?, 2001. Available online: <u>users.actcom.co.il/~choo/lupg/essays/softwaree</u> <u>ngineering-and-uni.html</u>
- [7] B. Meyer, Software Engineering in the Academy, *IEEE Computer*, No. 5, 2001, pp. 28-35.
- [8] SWEBOK Guide to the Software Engineering Body of Knowledge, Stone Man Trial Version. IEEE, Computer Society, 2001.
- [9] Accreditation Criteria for Software Engineering, IEEE Computer Society and ACM, Software Engineering Coordinating Committee, 1998.
- [10] Criteria for Accrediting Engineering Programs. Effective for Evaluations during the 2000-2001 Accreditation Cycle. ABET (Accreditation Board for Engineering and Technology, Inc.), 2000.
- [11] G. Wilstrand, J. Börstler, Success Factors for Team Project Courses, Proceedings of 19th Conference on Software Engineering Education and Training, 2006, 95-102.

- [12] P.J. Dening, Educating the New Engineer, *Communication of the ACM*, ACM, Vol.35, No.12, 1992, pp. 83-97.
- [13] J.T. Gorgone, P. Gray (eds.), Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. *Report of the Joint ACM/AIS Task Force on Graduate IS Curriculum*. Association for Computing Machinery, January 2000.
- [14] S.S. Chan, R.J. Wolfe, X. Fang, Teaching HCI in IS/EC Curriculum, Proceedings of the Eighth Americas Conference on Information Systems, 2002, pp. 1011-1020.
- [15] T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantel, G. Perlman, G. Strong, W. Verplank, Curricula for Human-Computer Interaction. Association of for Computing Machinery, 1992, 1996. Available online: http://sigchi.org/cdg/index.html
- [16] B.R. Swan et al., A Preliminary Analysis of Factors Affecting Engineering Design Team Performance, *Proceedings of the 1994 ASEE Annual Conference*, 1944, pp. 2574-2589.
- [17] E. Lingard, E. Berry, Teaching Teamwork Skills in Software Engineering Based on an Understanding of Factors Affecting Group Performance, 32nd Annual Frontiers in Education (FIE'2002), Vol.3, 2002, pp.S3G-1– S3G-6.
- [18] D. McDonald, J. Devaprasad, P. Duesing, A. Mahajan, M. Qatu, M. Walworth, Reengineering the Senior Design Experience with Industry-Sponsored Multidisciplinary Team Projects, *Proceedings of 26th Annual Conference Frontiers in Education (FIE'1996)*, Vol.3, 1996, pp.1312-1316.
- [19] R. Dowson, Twenty Dirty Tricks to Train Software Engineers, *Proceeding of the 2000 International Conference on Software Engineering*, 2000, pp. 209-218.
- [20] H. Younessi, D.D. Grant, Using CMM to Evaluate Student SE Projects, *Proceedings of the International Conference on Software*

Engineering: Education and Practise, 1996, pp. 386-391.

- [21] K. Moroz-Lapin, S. Ragaisis, Teaching HCI in SE Curriculum, 4th IASME/WSEAS International Conference on Engineering Education EE'2007, 2007, pp. 236-241.
- [22] S. Khajenoori, Process Oriented Software Education, *IEEE Software*, Vol. 11, Is. 6, Nov. 1994, pp. 99-101.
- [23] S. Ludi, Undergraduate Software Engineering Curriculum Enhancement via Human-Computer Interaction, *ICSE Workshop on SE-HCI*, 2003, pp. 72-75. Available online: <u>http://www.se-hci.org/bridging/icse/p72-75.pdf</u>
- [24] K. Moroz-Lapin, The Comparative Analysis of the Teaching of Human-Computer Interaction (in Lithuanian), *Informacijos mokslai (research papers in Information sciences)*, Vilnius University, Vol. 30, 2004, pp. 66-73.
- [25] J. Lazar, J. Preece, J. Gasen, and T. Winograd, New Issues in Teaching HCI: Pinning a Tail on a Moving Donkey, *Extended Abstracts, Proceedings of ACM SIGCHI Conference CHI*'2002. 2002, pp. 696-697.
- [26] B. Khoo, An Interactive Case Scenario for Teaching User Interface Design, *Proceedings* of the Fifth American Conference on Information systems, 1999, pp. 918–920.
- [27] J.A. Polack-Wahl, Teaching HCI in Software Engineering, *Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference*, 2004
- [28] A.J. Kornecki, S. Khajenoori, D. Gluch, On the Partnership Between Software Industry and Academia. *Proceedings of the 16th Conference* on Software Engineering Education and Training CSEET'2003, pp. 60-69
- [29] T. Maqmood, K. Lister, S. Karunesekera, E. Kazmierczak, Industry Base Learning in Software Engineering: Successes and Challenges. Proceedings of Workshop on Software Engineering Education, December 4, 2007, Nagoya, Japan, pp. 16-21.