Teaching HCI in SE Curriculum

KRISTINA MOROZ LAPIN, SAULIUS RAGAISIS
Software Engineering Department
Faculty of Mathematics and Informatics
Vilnius University
Naugarduko 24, LT-03225 Vilnius
LITHUANIA
http://www.miu.lt/se/

Abstract: - This paper reports the experience of Software Engineering educatn in Vilnius University, Lithuania. There are emphasized the collaboration with industry and the learning objectives related to Human-Computer Interaction (HCI) when establishing a separate undergraduate Software Engineering study program as well as implementing it. 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 two stages approach: in-home simulation of the real world projects and the projects provided by industry. This evolution has influenced also the teaching of HCI. This initially stand-alone 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 Curriculum, HCI education, team projects, collaboration with industry.

1 Introduction

Traditionally 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 a strong toward development tasks, 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]. They also value tools and implementation and tend to bypass paper prototypes. They also lack fundamentals of both communication and collaborating skills that were asserted as essential professional attributes [2]. Several HCI topics, related to interface design, was taught in software engineering course, paying a little attention to usability education. Students learned some theory that underpins usability, but did not have a possibility to experience working with real users that have different needs and perceptions from them [3].

It was understood that "Software Engineering Programmes Are Not Computer Science Programmes" [4] 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 [5];
- regulations of Lithuanian Ministry of Science and Education and Vilnius University;
- recommendations of international professional organizations ACM, IEEE and authorities, e.g. [4, 6], and Software Engineering Body of Knowledge under development [7];
- requirements of accreditation boards for study programs in computing CSAB and ABET [8, 9];
- 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 systems development 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. Especially representatives of foreign IT companies emphasized this opinion.

Further we address the integrating of HCI into software engineering curriculum, the content of HCI lectures and assignments. We deal with it from both theoretical (on lectures) and practical points of view (in the projects). Next we summarize our experience of collaborating with industry and analyse how the collaboration influenced curriculum enhancements.

2 Integrating HCI into software engineering curriculum

Developing the new software engineering curriculum, the need to improve both the practical skills of working in software development group and the user-centred education was recognised [1, 2, 10, 11, 12]. HCI course was introduced as a stand-alone course with its own assignments.

Developed software engineering curriculum was the result of employer's expectations and academy realisation abilities (Table 1). Industry influenced the rise of ten courses that concern the above mentioned problems. These courses are emphasized in Boldface type, electives are in Italics.

Table 1. Software Engineering Curriculum at Vilnius University

Mathematics and Basic Sciences	Computing Fundamentals	Software Engineering
Discrete Mathematics	Computer Architecture	**Personal Software Process**
Mathematical Logics	Programming Fundamentals	Software Engineering I
Linear Algebra and Geometry	Introduction to IT	Software Engineering II
Mathematical Analysis	Algorithms and Data Structures	**Team Software Process**
Theory of Graphs	Object-Oriented Programming (C++)	Internet Technologies
Probability Theory and Statistics	Object-Oriented Programming (Java)	**Human Computer Interaction**
Statistical Data Analysis	Database Management Systems	**Team Project I**
Physical Foundations of Electronics	Theory of Algorithms	Object-Oriented Design
Physics for Informatics	Operating Systems	**Software Projects Management**
Mathematical Modelling	Computer Networks	Course Work in Software Engineering
Numerical Methods	Bioinformatics	Software Process
Optimization Methods	*Documents Management Systems*	Software Testing
Humanities and Social Sciences	*Coding Theory*	Software Quality Management
Communication Skills	*Logic Programming*	**Professionalism and Ethics**
English Language	*Computer Hardware*	**Team Project II**
Introduction to Philosophy	*Information Systems*	Bachelor Work in Software Engineering
Introduction to Psychology	*Artificial Intelligence*	*Distributed Systems*
Basics of Economics	*Geographical Information Systems*	*Real Time Systems Design*
Fundamentals of Management	*Information Systems Audit*	*CASE Method*
Accounting	*Compiling Methods*	
Informatics Law	*Computer Graphics*	
Free Electives		

After the first 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 could be conducted under supervision of department but with user and customer from this company.

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 [13]. We understood that HCI course should be integrated with team projects.

Integrating HCI into existing computing curricula, the two approaches can be applied:

- a separate introductory HCI course [13, 14], or
- HCI topics incorporated in existing courses [15, 16, 17].

IS/EC curriculum combines these approaches [11]:

- 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:

- Human Computer Interaction (HCI) and Team Software Process (TSP) courses have separate lectures;
- the assignments are performed in the frame of Team Project I (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 significant change [15].

3 HCI content

Integration influenced also the changes in both sequence of HCI lecture topics and structure of assignments.

3.1 Lectures

According to ACM SIGCHI Curricula for Human-Computer Interaction [12], HCI course should overview the following aspects:

- (N) the nature of HCI,
- (U) the use and context of computers,
- (H) human characteristics,
- (C) computer system and interface architecture, and
- (D) development process.

Initially the sequencing of course topics accorded with above-mentioned recommendations. However, further integration with Team Project I required reorganizing the sequence of lectures. Topics related to iterative user-centered development are discussed deeper and are shifted towards the beginning. Alike T-model [18], we deal with the whole breadth of HCI topics in overview; while incorporation the user-centered design techniques into existing software design methodologies are treated deeper (Table 2).

Table 2. HCI course content

Lecture	Topic	HCI aspect
1	Introduction to HCI	(N)
2–3	Humans abilities and constraints	(H)
4	Usability of interactive systems	(U)
5 – 6	User–centred iterative design techniques.	(D)
7 – 9	Usability evaluation methods	(D)
10	Testing with users	(D)
11	Interface building tools	(D)
12 – 13	Computer systems and interface architecture	(C)
14	Computer supported cooperative work	(U)
15	HCI theories, models, principles, standards and recommendations.	(N)
16	Final team project presentations	

3.2 Assignments

Student teams receive the task that is unified with TSP. HCI deliverables cover user and task analysis, interface requirements, low fidelity and high fidelity prototyping, and usability evaluations (Table 3). First prototype is evaluated using Keystroke Level Model and Cognitive Walkthrough. Detailed prototype is evaluated using heuristic evaluation.

Table 3. HCI assignments

Week no.	Assignments
3	1. User and task analysis, usability goals, interface requirements
5	2. Low fidelity prototype with early evaluations (KLM, CW)
7	3. Project requirements
10	4. High fidelity prototype with evaluation (HE)
16	5. Working implementation and user documentation.

The final project must due to the end of class. Project outcome is the complete system with user documentation. Programming can be done in any language or system. Final project is defended in public oral presentation, in which participate students, HCI and TSP lecturers as well as representative of industry partner.

Before the integration HCI with the TP1, one team member rather than the 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 TSP 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.

4 Experience of the collaboration with industry partners

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 [19]. We have about 120 SE students per year. Organising student internships, Vilnius University collaborates with about twenty partners (Table 4).

Table 4. Distribution of industry partners in Spring 2005

Industry partners	Number	Percentage
IT Companies	13	68,5 %
Banks	2	10,5 %
Other	4	21,0 %

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 user.

Current curriculum offers two team project courses. Both projects 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 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 user-centred 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. 1) that was even better than the marks which we gave

assessing student internship outcomes (Fig. 2). Comparing with earlier criticism the opinion of our customers is significantly better.

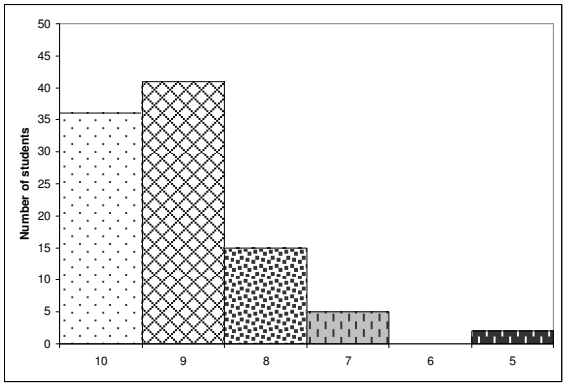

Fig. 1. Distribution of marks proposed by customers (hosting companies) in Spring 2005

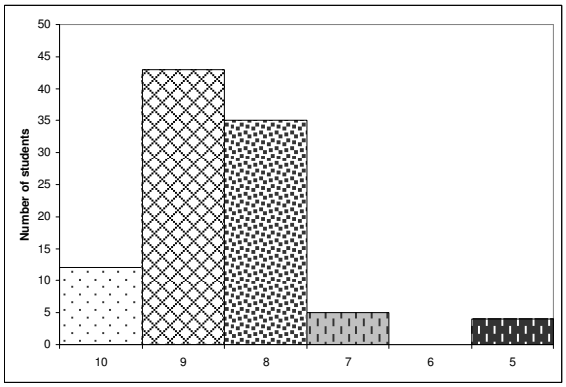

Fig. 2. Distribution of final marks for Team Project II performed in Spring 2005

5 Conclusions

In our opinion, the completely stand-alone HCI course is not ideal solution. Our experience with HCI course before the integration with Team Project I approved this observation. In order to have an impact on usability, students have to learn incorporating the user-centred design concepts to the SE projects.

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 user-centered 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.

We state that our proposed approach with separate lectures and joined projects is better solution than completely separate courses. This approach motivates students to apply HCI skills in other courses. 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, 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 the software development companies deemed both team projects to be extremely successful. Their opinion was that both introduction of HCI course and integration it with projects highly improved student' skills of field studies and usability engineering.

References:

[1] A.N. Ribeiro and J.C. Campos and F.M. Martins. Integrating HCI into a Software Engineering course. In *Proceedings of the HCI Educators 2007,* March 2007, pp.49–58.

[2] D.K. Lidtke and 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. Available online: www.comp.lancs.ac.uk/computing/users/dixa/papers/HCI-educ98/

[4] D.L. Parnas, Software Engineering Programmes Are Not Computer Science Programmes, *Annals of Software Engineering*, Volume 6, Issue 1-4, 1999, pp. 19-37.

[5] G. Karen, Why do Universities Fail Teaching Software Engineering?, 2001. Available online: users.actcom.co.il/~choo/lupg/essays/softwareengineering-and-uni.html

[6] B. Meyer, Software Engineering in the Academy, *IEEE Computer*, No. 5, 2001, pp. 28-35.

[7] SWEBOK - Guide to the Software Engineering Body of Knowledge, Stone Man Trial Version. IEEE, Computer Society, 2001.

[8] Accreditation Criteria for Software Engineering, IEEE Computer Society and ACM, Software Engineering Coordinating Committee, 1998.

[9] Criteria for Accrediting Engineering Programs. Effective for Evaluations during the 2000-2001 Accreditation Cycle. ABET (Accreditation Board for Engineering and Technology, Inc.), 2000.

[10] Gorgone, J.T., and Gray, P. (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.

[11] S.S. Chan, R.J. Wolfe, and X. Fang, Teaching HCI in IS/EC Curriculum, *Proceedings of the Eighth Americas Conference on Information Systems*, 2002, pp. 1011 – 1020.

[12] Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantel, M., Perlman, G., Strong, G., and Verplank, W. Curricula for Human-Computer Interaction. *Association of for Computing Machinery*, 1992, 1996. Available online: http://sigchi.org/cdg/index.html

[13] S. Ludi, Undergraduate software engineering curriculum enhancement via Human-computer interaction, *ICSE Workshop on SE-HCI*, 2003, pp. 72-75. Available online: http://www.se-hci.org/bridging/icse/p72-75.pdf

[14] K. Moroz Lapin, The comparative analysis of the teaching of human-computer interaction (in Lithuanian), *Informacijos mokslai (research papers Information sciences)*, Vilnius University, Vol. 30, 2004, pp. 66-73.

[15] 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.

[16] B. Khoo, An Interactive Case Scenario for Teaching User Interface Design, *Proceedings of the Fifth American Conference on Information systems*, 1999, pp. 918–920.

[17] J.A. Polack-Wahl, Teaching HCI in Software Engineering, *Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference*, October 20–23. October, Savannah, GA. 2004

[18] A. Dix, HCI Education - mastery, *SIGCHI Bulletin*, Volume 35, Number 2, March 2003. Available online: http://www.comp.lancs.ac.uk/research/publications/publications.php?publication_title=hci

[19] 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