Enhancing creativity in HCI education

KRISTINA MOROZ-LAPIN
Software Engineering Department
Faculty of Mathematics and Informatics
Vilnius University
Naugarduko 24, LT-03225 Vilnius
LITHUANIA
kristina.lapin@mif.vu.lt    http://www.mif.vu.lt/~moroz/

Abstract: - In this paper, we describe our experience of modifying the teaching of Human Computer Interaction in Vilnius University, Lithuania. Recent modifications aim at fostering student creativity and inventiveness. Software engineering students often lack the preparation to invent creative solutions because they are trained to perform structured analysis and development. Creative solutions do not mean only aesthetically pleasing user interfaces. We understand this concept as innovative ideas that implement various solutions for users' needs. After introduction of Human-Computer Interaction course in 2002, we have modified our teaching. Initially separate HCI course was integrated with SE project. This allowed balancing student workload and gaining experience to perform user-centred design activities in SE projects. The next modifications that change beginning of the course and lecture content stimulate students to be active and creative during interface design. To illustrate our approach we present some examples of student prototypes.

Key-Words: - HCI education, interface design, user-centred design, creativity.

1 Introduction
This paper outlines the way to enhance creative invention in Human Computer Interaction (HCI) course that is delivered at Vilnius University (VU). Significant part of any HCI course covers usability concept and usability evaluations. But it is important to stress that course should aim not only at usable solutions but at solutions that enhance quality of interaction. The narrow orientation to prototyping and usability evaluations does not motivate students to be creative. Such an approach often lacks for methods that invent better solutions and designs [1].

Further we summarize our experience with HCI teaching and SE curriculum enhancements. A brief analysis of existing HCI courses reveals different teaching trends in motivating students creativity. Next we deal with recent modifications that concern stimulation of creativity in student projects. We deal with it from both theoretical (on lectures) and practical points of view (in the projects). To illustrate results, we show some exemplar prototypes followed by conclusions.

2 Modifications in SE curriculum
Before the implementation of the new software engineering (SE) study programme, several HCI topics, related to interface design, were taught in SE 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 [2]. Graduates possessed strong development skills, e.g. programming, data structures, and databases.

However, such approach to education of software professionals was not satisfactory: graduates did not receive the knowledge and skills needed for industrial software development. It results in low quality and unusable software systems. Students were weak in both visual design and understanding of user-centred design process. They tended to approach design and analysis from system developer perspective, rather from user’s perspective [3]. They also value tools and
implementation, and tend to bypass paper prototypes. Students lacked for both communication and collaborating skills that are asserted as essential professional attributes [4].

In 2001, the first students were accepted into separate undergraduate Software Engineering (SE) study programme [5]. The development of SE curriculum was based on experience of delivering SE courses in existing Computer Science (CS) curriculum. University staff has been also involved to industrial projects for Lithuanian and foreign companies that is very important for software engineering education [6].

A lack of vertical and horizontal integration of the individual courses throughout the old CS curriculum was realized [10]:

- 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.

Industry expectations helped defining the main curriculum outcomes and the emphasized aspects [5]. Industry partners stressed that our graduates lacked skills of teamwork, communication with customers, application area analysis, and project management. The need to improve both the practical skills of working in software development group and the user-centred education was recognised [3, 4, 7, 8, 9]. In order to solve these shortcomings project-based approach was introduced [10].

3 Fostering creativity in HCI courses

The techniques of creativity are abundantly used in the industrial, commercial, publicity domains. They make it possible to bring the new ideas and concepts in a minimum time and with a best performance. According to Dondon [11], creativity could be introduced through appropriate management in educational projects by facilitating individual creativity and using Herrman Human behaviour modelling [12] in creativity seminars.

3.1 Stimulating creativeness in lectures

According to Altshuler [13], creativeness ranges from minor improvements to new concepts, and to true discoveries. Creativeness needs time space for social interaction because "creativity is individualistic characteristic, and innovation is a social activity" [14]. Computer science students are trained to use structured methods in analysis and development. The rigid methodologies contradict to fuzzy creativity processes. The challenge is integration of highly organised engineering processes with fuzzy activities that facilitate creative invention [1].

Creativity can be summarised as partly talent but also solid skills, planning, understanding the requirements, implementation, theory, testing, that apply to the problem and the solution [15].

Wong [13] provided the practical suggestions how inventiveness could be incorporated into HCI, which among other stated the following:

- students should be expected to generate ideas, present them in sketches and explain how these ideas influenced the final result;
- the more realistic settings are needed which replicate real-world cases with enough low-level details in order to develop sound understanding of the domain;
- before the entering HCI courses students should be aware of tools and frameworks for creativity, that assist them in discovering, generating and refining new ideas;
- create opportunities for ‘eureka moments’;
- expose students to design solutions from different domains;
- learning environment should be shaped to collaboration and sharing of ideas within a student groups.

According to Lars Oestreicher, successful teaching should shake student minds by making them aware of a constructive curiosity [16]. It is not sufficient to provide just good or bad design examples. These examples should make the students to reflect on the further consequences emanating from the more general problems illustrated through the examples. Our experience and introduced course modification are very near with above mentioned findings.

3.2 Tools and methods for inventiveness

In this section we outline some tools that could help students to generate ideas. On conference in Limerick the following methods were suggested [1]:

- on-line lab-diaries allow the public brainstorming of ideas about the problem and design concept [14];
- the contextual interviews, the affinity diagrams and the development of personas and scenarios help students to understand the context and nature of the problem [17];
- the record of cooperative evaluation, in which partners think aloud, provides a source of reflection and insight to designers and could be used to focus on creative thinking [18];
- project-based approach is a suite of methods and techniques which incorporates design journal,
the morphological box with different combinations of all possible solutions, sketching and representation techniques, project-planning, scenario-based design and other user-centred methods [19];

- Bad Ideas Toolkit suggests using systematic methods to critically review bad ideas from different perspectives, it creates new opportunities by making them good ideas that helps avoiding design fixation and trains students to explore and understand the extent and constraints of the design space [20].

Fonseca, Jorde et al. [21] point out that significant obstacle that restricts creativity occurs when students are forced to go from the initial task analysis directly to prototyping. In their opinion it is very brusque step in the interface development methodology because students start thinking in interaction styles and screen layout before thinking about the solution for users' needs. The authors suggest including the conceptual modelling phase between the task analysis and low-fidelity prototypes.

Creativity can be encouraged by introducing to engineering curriculum special courses e.g. visual literacy course [22]. Such courses prepare engineering students to be effective visual communicators. Students are encouraged to brainstorm thoroughly and not settle with first idea that comes in mind. During the lectures the different ways to think creative, such us divergent thinking, are introduced to them.

4 Teaching HCI

HCI course was introduced as a stand-alone course with its own assignments in 2003. After the first student internship in industry we realised that despite of the introduced HCI course, subsequent team project outcomes have not improved: many students limited the use of acquired HCI knowledge only to HCI assignments [23]. These results induced the first modifications: HCI course was integrated with SE team projects, combining separate lectures with joint practical classes [5]:

- HCI and Team Software Process (TSP) courses have separate lectures;
- 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;
- a 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 [24]. Students gain more realistic settings and can develop sound understanding of the domain.

The modifications consolidated student skills using user-centred design techniques in SE projects, but results lacked creative solutions. In most cases the first low-fidelity solutions were elaborated to high-fidelity design throughout the project. Though results were usable enough but they also showed up creativity gap between the definition of usability goals and prototyping.

Second modifications made last year aimed to break creativity gap and concerned both lectures and practical classes:

- bad design examples are presented in lectures, critical review provides a way to think of how to make good initially bad solutions;
- list of assignments is augmented with an individual homework, in which students are expected to notice interactive devices (not necessary software) that caused difficulties in use, why they are unusable and how students would improve them;
- unconventional solutions receive extra points in assessment.

These modifications do not teach creativity. They aim to shake student minds and induce a change in their mindset. The bad examples in lectures and practical classes are not just shown, but also elaborated from various perspectives. Students like to present problems in searching required information in web pages or remember their experience when they learned to use new mobile phones. Their examples concern not only software but also other products, such as lifts, copiers, control devices in the cars, etc. These presentations stimulate discussions about further consequences that arise for user while using these devices.

4.1 Theoretical classes

The basic ideas about course structure are implemented according to ACM SIGCHI Curricula for Human-Computer Interaction [9]. 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-centred development are discussed deeper and are shifted towards the beginning. Alike T-model [25], we deal with the whole breadth of HCI topics in overview; while incorporation the user-centred design techniques into
existing software design methodologies are treated deeper. The recent changes did not influenced course structure [5] but rather content. Presented poor design examples illustrate lecture topics. It is thought-provoking material that activates students during lectures.

4.2 Practical classes
After the integration HCI with Team Project I student the task for students’ teams is unified for both courses. HCI deliverables cover user and task analysis, usability goals and interface requirements, brainstorming session, low-fidelity and high-fidelity prototyping, and usability evaluations (see Table 1). After recent modifications we begin the course with homework. Students present inconvenient solutions in interactive devices. They analyse consequences for users and propose the way, how to make them good.

Table 1. HCI assignments

<table>
<thead>
<tr>
<th>Week no.</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1. Individual homework.</td>
</tr>
<tr>
<td>4</td>
<td>2. User and task analysis, usability goals</td>
</tr>
<tr>
<td>5</td>
<td>3. Low-fidelity prototypes with usability evaluations</td>
</tr>
<tr>
<td>6</td>
<td>4. Interface requirements and project recommendations</td>
</tr>
<tr>
<td>7</td>
<td>5. High fidelity prototype with heuristic evaluation</td>
</tr>
<tr>
<td>11</td>
<td>6. Working implementation and user documentation</td>
</tr>
</tbody>
</table>

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 behaviour. After applying HCI practices to their first project, students 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.

Individual homework motivates students to rethink obvious solutions and reflect them from various perspectives. Students comment that they change their minds when they see solutions that are asserted by colleagues as inconvenient for use. Creativeness is social activity, so alternative solutions are generated in brainstorming session. Our experience also endorses that it is hard to admit creative process to rigid time on the classes [14]. Therefore, students use wiki, installed on faculty server, to continue discussions and elaborating created ideas. Discussion on faculty wiki also help them to present how alternative solutions, generated during brainstorming sessions influenced the design of high-fidelity prototype.

4.2 Examples
Unconventional solutions gain extra points assessing the deliverables that also motivates to generate more alternative designs. Creative solution has to match with user needs and usability goals.

Below we present two examples of student projects. The example from Fig. 1 illustrates a prototype of credit limit assurance assessment system. The idea of this project is to help insurance analyst to assert credit limit in a specific case. While making the credit limit insurance decision, the analyst gains the information from the internal and external data sources. Internal data bases provide information of the previous insurance operations. External sources include financial indexes, such as status, turnover, sales, buying, the capital, spending, liquidity ratio, the development, and the events from the past contracts. Queries to internal sources are processed immediately. Queries to external sources block the decision-making process until all answers will be received. Error prevention is essential.

Appropriate visualisation informs, which queries are already completed, which are still in progress, and how many applications for the credit insurance have been not evaluated yet.

Fig. 1. Prototype of credit limit assurance assessment system

In Fig. 2 we can see interface for air traffic control that aims to enhance decision support for aircraft landing procedure. Display presents the one-aircraft landing model where the actual position with relevant parameters is shown. Situational awareness in this
display encompasses trajectory view in 3D and 2D projections, velocity variations and true/false diagram about current situation. Sliders on the right part of window allow adjusting safe parameters to the aircraft type (light, medium, heavy).

Fig. 2. Decision support for aircraft landing control

6 Conclusion

We have presented our approach of teaching an introductory HCI course in SE curriculum. Introduced modifications enhanced HCI teaching by integrating assignments with SE projects. Last year modifications encouraged creative thinking.

Students generally like present homework with founded bad designs. This exercise is perceived as funny activity. In previous year the first assignment was user and task analysis, followed by definition of usability goals. The first assignment was asserted by students as hard because it was not easy to think from user perspective without preparation.

Homework presentations usually raise discussions, activate imagination and involvement. It is important at the beginning of the course. After discussions students easier understand the user perspective and define the usability goals. So, in our opinion good and bad design examples stimulate creative thinking in human-computer interaction design.

Brainstorming session and creation of at least two alternative solutions fosters to try various interactions styles for defined usability goals. It increases amount of creative thinking. Positive aspect is available wiki on faculty server that facilitates discussions after the class hours.

Presenting project requirements students show how alternative solutions influenced their decisions.

We feel satisfied that student response to the recent changes in the content of lectures and assignments has been mainly positive.

We plan further curricular improving by dedicating more space to emerging interaction techniques and styles such as mobile devices and ubiquitous computing as well as to emphasise a Web development. We have also to rethink the assessment of student projects with more sound dependence between innovative solutions and project marks.

References:


