ABSTRACT
The importance of Human-Computer Interaction (HCI) education for software professionals should be evident and well understood, when designing Computer Science (CS) programs, at all levels. Unfortunately this is not the case, and usually the professors involved in CS programs consider HCI to be a secondary matter. This could be an explanation for the lack of HCI courses, at least in Chilean CS programs. These are rather exceptions, due to enthusiasts, being usually optional, but not compulsory subjects. The paper presents the experience of Pontificia Universidad Católica de Valparaíso in implementing HCI in CS undergraduate and graduate curricula, using a practical approach, fitted to the necessities of the local software companies, based on the feedback received from the ex-students. We focus on preparing HCI practitioners in undergraduate programs, and both practitioners and researchers in graduate programs.

Categories and Subject Descriptors
H.5.2 [User Interfaces]: Evaluation/methodology, Theory and methods, User-centered design.
K.3.2 [Computer and Information Science Education]: Computer science education, Curriculum.

General Terms
Design, Human Factors, Theory.

Keywords
Human-Computer Interaction, Computer Science curricula, HCI practices.

1. INTRODUCTION
The Special Interest Group on Computer-Human Interaction (SIGCHI) of the Association for Computing Machinery (ACM) defines Human-computer Interaction (HCI) as a discipline concerned with the design, evaluation, implementation of interactive computing systems for human use, and with the study of major phenomena surrounding them [17]. HCI is a highly interdisciplinary area, involving mainly Computer Science (application of design and engineering of user interfaces), Psychology (application of theories of cognitive processes and the empirical analysis of user behavior), Sociology/Anthropology (interactions between technology, work, and organization), and Industrial Design (designing of interactive products). HCI focuses on Computer Science, and belongs mainly to Computer Science area. Other disciplines serve as supports. HCI uses supporting knowledge on both machine and human side. Actually HCI may be considered as a combination of Science, Engineering and Design. Basically, HCI tries to make people’s experience with computers more productive, more time-efficient, and more pleasant.

The importance of (HCI) education for software professionals should be evident and well understood, when designing Computer Science (CS) programs, at all levels. It sounds good in theory, but is difficult in practice. Unfortunately the professors involved in CS programs usually consider HCI to be a secondary matter. This is probably a reflection of the well known conflict between HCI specialists and software engineers [7]. It has serious consequences for the training of the software professionals, as they will focus on the inner part of the software systems, ignoring the importance of the outer part (the interface). What will we get then? Top models dressed in poor clothes, and not in the “haut couture” creations they deserve!

The extraordinary development of the HCI field, both as theory and practice, is poorly reflected in South America, with the notable exception of Brazil [1], [2]. This unfortunate “rule” is also followed by Chile. There is an obvious lack of HCI courses in Chilean CS programs. These are rather exceptions, due to enthusiasts, being usually optional, but not compulsory subjects. What should we do? How can we prove the importance of HCI, in an environment where trust lacks? Should we try to change the mentality of our colleagues, or should we try to influence the software professionals? We should probably act on both sides.

Our interest in getting better user interfaces came from practice, as we have a rather extensive previous experience in software development [14], [15], [16]. Later on, we discovered HCI as a well established field of CS. So, we make our way from practice to theory, and we have the natural impulse to get back to practice. We tried to do this in our HCI classes [13].
2. THE HCI COURSE IN A COMPUTER SCIENCE UNDERGRADUATE PROGRAM

Once we discovered HCI as a well established science, we were tempted to share our experience with others, and especially with our students. As there were no HCI subjects in CS curricula in Pontificia Universidad Catolica de Valparaiso, Chile (PUCV), the only way to do that was to propose an HCI course as optional subject. We did that for the very first time back in 2003. It was a double challenge, as it was our first experience in teaching HCI, and we did it in English, not in Spanish, which is the official language in Chile (and in PUCV). The experience was highly successful, so we repeated it every year. We never intended to teach an English course; English was just a tool, but not a goal. The course was designed from the very beginning as a CS course. However, we intended to also offer an opportunity to practice applied technical English.

When we tried to teach the same course in Spanish, the interest among students was even higher, as only a relatively small number of students have enough knowledge of English to participate in HCI classes taught in English (see table 1). This second version of our HCI course was, however, not a simple translation of the course previously taught in English. We had to rethink the course, especially due to the rather poor Spanish references. So we came to realize the disparity of HCI bibliography in English and Spanish.

Teaching HCI every year offered us the possibility to improve it, and to adapt it to the necessities of local software companies, mainly based on the feedback received from ex-students. We kept the general structure of the course more or less the same, but we changed the weights of different topics. The structure that we are using is as follows:

- Field of Human-Computer Interaction,
- The Nature of Human-Computer Interaction,
- Computer System and Interface Architecture,
- Usability,
- Interaction Design,
- Help and Documentation,
- Web Design.

The most important area of HCI, from a software developer point of view, is the user interface design and development. This mainly involves user-centered design and task analysis, dimensions of interface variability, dialogue tools and techniques, interface design, interface implementation, interface quality and evaluation methods.

Our course mainly focuses on Usability: Usability Paradigms and Principles, Usability Engineering, Usability Evaluation Methods, Usability in Practice [8], [10], [11]. The ISO 9241 standard defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Designing for maximum usability should be the main goal of the software design. There cannot be software without software users! That is why we consider usability as the most important software attribute. Two main problems occur: (1) how can an interactive system be developed to ensure its usability? and (2) how can the usability of an interactive system be demonstrated or measured?

We strongly believe in a user-centered approach for interface design and software development [3], [4], [9]. A user – centered development methodology differs from traditional software engineering methodologies in three key areas: (1) is user centric, not data centric, (2) is highly interdisciplinary and draws knowledge from many areas, and (3) is highly iterative and involves as much testing and revision as possible.

The main changes that we have done since 2003 were into the balance theory – practice. We gradually increased the weight of practical activities, and we came to focus more and more on teaching the students how to put the HCI theory into practice. Our belief is that we have to prepare HCI practitioners in CS undergraduate programs. That is why we are trying to include as many practical exercises as possible. For example, students have to perform at least 3 – 4 heuristic evaluations during the semester. They also have to practice brainstorming sessions and user interviews, among others. They are always encouraged to interact, to expose their ideas, to make proposals, to analyze and evaluate them.

Students have to prepare a final HCI project (preferable a group project), consisting in the development of user interface prototypes. They have to apply the usability concepts, to cross-evaluate the prototypes, and to improve them based on the evaluations they performed. They have to highlight the changes and the improvements they have made, in public presentations. In order to encourage interdisciplinarity and to make connections with other subjects of CS curricula, students are encouraged to choose HCI project subjects that can help them in improving some other software development projects.

Students’ feedback was excellent; they really enjoyed the HCI course. This is proved by the increasing number of the students that are choosing the course (from a wide range of available courses), the increasing number of students that are choosing HCI related subjects for their final (graduation) thesis, and the contact that they are maintaining with us over the time.

Moreover and more important, ex-students are keeping contact with us after graduation. They offer a valuable feedback from practice, from their practical experience as software developers. They are trying hard to apply in practice what they learned in HCI classes. Even if it is difficult, and sometimes discouraging, they are actually developing, step by step, an awareness of the importance of HCI practices into their professional environment.

Table 1. Number of students of HCI undergraduate course

<table>
<thead>
<tr>
<th>Year</th>
<th>No. of students</th>
<th>Language of teaching</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>15</td>
<td>English</td>
</tr>
<tr>
<td>2004</td>
<td>33</td>
<td>Spanish</td>
</tr>
<tr>
<td>2005</td>
<td>26</td>
<td>English</td>
</tr>
<tr>
<td>2006</td>
<td>27</td>
<td>English</td>
</tr>
</tbody>
</table>

Principles, Usability Engineering, Usability Evaluation Methods, Usability in Practice [8], [10], [11]. The ISO 9241 standard defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Designing for maximum usability should be the main goal of the software design. There cannot be software without software users! That is why we consider usability as the most important software attribute. Two main problems occur: (1) how can an interactive system be developed to ensure its usability? and (2) how can the usability of an interactive system be demonstrated or measured?

We strongly believe in a user-centered approach for interface design and software development [3], [4], [9]. A user – centered development methodology differs from traditional software engineering methodologies in three key areas: (1) is user centric, not data centric, (2) is highly interdisciplinary and draws knowledge from many areas, and (3) is highly iterative and involves as much testing and revision as possible.

The main changes that we have done since 2003 were into the balance theory – practice. We gradually increased the weight of practical activities, and we came to focus more and more on teaching the students how to put the HCI theory into practice. Our belief is that we have to prepare HCI practitioners in CS undergraduate programs. That is why we are trying to include as many practical exercises as possible. For example, students have to perform at least 3 – 4 heuristic evaluations during the semester. They also have to practice brainstorming sessions and user interviews, among others. They are always encouraged to interact, to expose their ideas, to make proposals, to analyze and evaluate them.

Students have to prepare a final HCI project (preferable a group project), consisting in the development of user interface prototypes. They have to apply the usability concepts, to cross-evaluate the prototypes, and to improve them based on the evaluations they performed. They have to highlight the changes and the improvements they have made, in public presentations. In order to encourage interdisciplinarity and to make connections with other subjects of CS curricula, students are encouraged to choose HCI project subjects that can help them in improving some other software development projects.

Students’ feedback was excellent; they really enjoyed the HCI course. This is proved by the increasing number of the students that are choosing the course (from a wide range of available courses), the increasing number of students that are choosing HCI related subjects for their final (graduation) thesis, and the contact that they are maintaining with us over the time.

Moreover and more important, ex-students are keeping contact with us after graduation. They offer a valuable feedback from practice, from their practical experience as software developers. They are trying hard to apply in practice what they learned in HCI classes. Even if it is difficult, and sometimes discouraging, they are actually developing, step by step, an awareness of the importance of HCI practices into their professional environment.
We supervised the first undergraduate thesis developed in PUCV in the usability field: “Metodología de evaluación de usabilidad de sistemas de software” (Methodology for Usability Evaluation of Software Systems), finalized on 2005. Since then, more and more students are choosing a large range of HCI or HCI related subjects for their graduation thesis:

- two ex-students will soon present their final version of thesis on usability or usability related subjects: “Evaluación de la usabilidad por métodos formales” (Usability Evaluation by Formal Methods), and “Optimización del uso de técnicas del tipo SIG en la planificación territorial” (Optimization of GIS techniques in urban planning),
- one student is working on a thesis directly related to usability evaluations, since August 2005: “Prototipo de herramienta de apoyo en evaluaciones de usabilidad” (Tool Prototype for Usability Evaluations),

3. THE HCI COURSE IN A COMPUTER SCIENCE GRADUATE PROGRAM

A major step forward was the introduction of HCI as compulsory course into the curriculum of the Master Degree (MD) program in CS, a program that PUCV offers. We consider that the introduction of the HCI as a compulsory course into the new curricula was a major “battle” we won. Some of the arguments that helped us to win this battle were:

- the successful experience of teaching HCI as optional course in undergraduate program,
- the interest of the students,
- the feedback of the ex undergraduate students (now software professionals) that have graduated the HCI course,
- the demand of the potential MD students.

The last two arguments were proved by an inquiry that was made during the preparation of the MD program curriculum. Surprisingly, the software professionals are more convinced of the importance of HCI than our own colleagues (professors of CS programs). This belief comes from practice, from their professional experience. We find this extremely important, as it proves an overwhelming awareness of the importance of HCI practices. This is stronger and more important than any bureaucratic argument that may occur when designing CS curricula. It is particularly encouraging for the Latin American context.

The new MD HCI course represents a new challenge. Potential students are coming either from professional environment (being experienced software professionals) or from school (recently graduating a CS program, without practical experience). The audience is more heterogeneous than in the case of the undergraduate HCI course. Obviously, the new course has to be different. It has to include some HCI basics (for students without prior experience in HCI), without entirely repeating the content of the undergraduate HCI course. In addition, it has to bring the concepts to a higher level. We have to focus on preparing both HCI practitioners and researchers in CS graduate program. We are building the new course over the structure of the old undergraduate one, but with emphasis on different topics. The teaching process is more personalized, tanking advantage of the reduce number of students in a graduate program (around 15, comparing to approximately 30, in undergraduate programs).

We intend to keep the main focus on usability, and usability evaluation, but we introduce new topics, as elements of semiotic engineering, which can help to bridge the gap between HCI practitioners and software engineers [6]. Besides classical evaluation methods we will also introduce the communicability evaluation method [5], [12]. Regarding the infrastructure, a usability laboratory becomes necessary.

4. CONCLUSIONS

A CS curriculum has to be dynamic, to adapt itself to the continuous IT changes. We consider HCI as a basic part of the formative process of the future software professionals. Changing curricula is usually a long and bureaucratic process. Improvements are faster and much easier to implements by offering new or redesigned optional subjects.

We proposed an HCI course as optional subject back in 2003. The experience was highly successful, so we repeated it every year, in CS undergraduate programs. As we strongly believe in a user-centered approach, we are mainly focusing our HCI course on usability. We try to improve the course each year, and to adapt it to the necessities of local software companies, mainly based on the feedback received from ex-students. We gradually increased the weight of practical activities, and we came to focus more and more on teaching the students how to put HCI theory into practice.

A major step forward was the introduction of HCI as compulsory course into the curriculum of the MD program in CS, a program that PUCV offers. We focus on preparing HCI practitioners in undergraduate programs, but on preparing both HCI practitioners and researchers in graduate programs.

Surprisingly, the software professionals are more convinced of the importance of HCI than our own colleagues (professors of CS programs). This proves an awareness of the importance of HCI practices in the software professional environment, and is particularly encouraging for the Latin American context.

5. ACKNOWLEDGMENTS

Our thanks to SERG (Semiotic Engineering Research Group), of Computer Science Department of Pontificia Universidad Catolica de Rio de Janeiro (Brazil), especially to Prof. Dr. Clarisse de Souza, for all the support offered during our visit on November 2005, for her continuous encourages and collaboration opening.
6. REFERENCES


