Experiences in Teaching New Technologies in Computer Science University Degrees

Claudia A. Queiruga [1,a], Laura A. Fava [1,b,*] and Ana M. Ungaro [2,c]

[1] Laboratory of Investigation in New Information Technologies (LINTI), Computer Science School, University of La Plata, Calle 50 y 120, La Plata, Argentina

[2] Pedagogical Department, Computer Science School, University of La Plata, Calle 50 y 120, La Plata, Argentina

[a] claudiaq@info.unlp.edu.ar, [b] @info.unlp.edu.ar, [c]@info.unlp.edu.ar

Keywords: discovery learing, new technologies.

Abstract. The digital technological era in which we live makes our students native technological users, which radically redefine the ways in which teaching and learning take place. In this context, the Computer Science School of the UNLP describes a discovery learning proposal for two mandatory subjects, *Software Development Workshop*[1] and *Software Laboratory*[2]. The students in these subjects carry out projects related to local demands and the field of mobile applications, and are encouraged to form interdisciplinary aptitudes, owing to the emphasis that university education places in structural contents that have long term validity.

1. Introduction

We are going through a technological era of dizzying growth. Young students applying to universities are now digital natives, and therefore experienced users of web 2.0 technologies, as shown by their use of social networks to communicate and share knowledge. Facebook, Twitter and Flickr are examples of worldwide virtual communities. This special way of communicating gives these subjects an epochal identity, which means that many of these students were born into this phenomenon and grew up using these digital and virtual tools as their natural way of reading, seeing, understanding and conceptualizing the world. This has a great impact in university education, especially in degrees that focus on computer knowledge, such as engineering and computer science. Acknowledging this new landscape is a challenge for the teachers involved, which must become acquainted with these new technologies, not only as an additional means of communication in their courses, but also for their incorporation to course contents. The teachers in software development courses must interact with constantly evolving technologies, such as open APIs, services, current development frameworks, and new programming languages adapted for mobile devices, without neglecting core concepts that contribute to the graduates' professional careers.

Likewise, our country's production development is in need for graduates and advanced students that can contribute knowledge and skills in new software development technologies. This situation has considerable impact in how long it takes for students to get their degrees and how often they attend class. Taking into account this regional demand and its impact in degree obtention, the *Software Development Workshop* and *Software Laboratory* teachers are orienting their teaching proposals towards what students need to learn to become successful professionals. This is

[1] Obligatory third year course of the B.S. Computer Science at Computer Science School (UNLP)

[2] Obligatory fourth year course of the B.S. Computer Science at Computer Science School (UNLP)

particularly challenging to our educational practice, as it acknowledges a need to create an articulate balance between the epistemological dimension of knowledge specific to the computing discipline and the epistemological dimension of pedagogical knowledge [1][2]. In this way, this results in a new complex knowledge (in the framework of a teaching and learning process), which allows the teachers to build up a methodological solution of "teaching specific contents" within the area of software engineering.

However appropriate it may be to incorporate use and development of new technologies in computer science and engineering courses, the main goal of university education remains the same: providing access to structural concepts that have long-term validity. Modern technologies must be incorporated to better illustrate these fundamental concepts and place students in an environment that emulates real, everyday work conditions. For this reason, it is necessary to create a balance between teaching the concepts that make for competent and successful professionals and offering interaction with the technologies that implement these concepts. To illustrate this idea, let us consider the concepts of inheritance and polymorphism in object orientation: students learning to program in an object-oriented language such as Java or Smalltalk and who have understood these fundamental principles can benefit from code reusability and generic behavior, among others, and will be able to identify these concepts when learning to use another programming language.

Teaching these theoretical bases and analyzing their structural dimension allows taking the educational practice farther from a simple concept explanation, thus constructing a didactic solution that exposes the differences between theory and practice in order to overcome them by considering them necessary constituents of one single piece of knowledge.

This article presents a didactic proposal that articulates conceptual instruction with elements of discovery learning.

As the authors cite in [3], "in discovery learning, students are confronted with a challenge and left to work out the solution on their own". In the subjects presented in this article, discovery learning is articulated with previously acquired core knowledge in conceptual instructions. This combination gives each method influence over the other, boosting integration in the teaching of new technologies that are motivating to the students, with fundamental concepts for their long-term training as professionals.

2. Our Experience

2.1 Teaching Methods Applied

In the last few years, *Software Production Technologies Workshop* and *Software Laboratory* teachers have experimented with the use of a certain amount of teaching methods with the goal of increasing attendance, stimulating participation and, above all, helping students gain knowledge and skills that are relevant over time. We have used a traditional didactic approach to explain key concepts that are closely related with proposals that will later allow students to tackle complex problems, with interesting challenges that demand the application of these essential concepts. We consider this methodology articulation to be appropriate because the traditional method allows us to cover the most relevant topics, offering students a first contact with them and motivating their application in real-life situations, where the students' complex problem-solving skills will be put into play.

With the goal of achieving a proactive attitude in the students of these subjects, the teachers propose the development of projects related to specific needs raised by local public institutions or the implementation of games and mobile applications, which motivate our young students, as is well known. In these contexts, the teachers also face the students with incomplete information, to stimulate thinking and searching information in API (Application Programming Interface) documentation, specifications or Java developer community forums. The subjects presented in this article focus on two types of Java applications: server-side and mobile. For the development of server-side applications, the students work on multiple open source frameworks that are widely accepted in the market and in the Java developer community. The students are shown use and scalability of each tool in the framework. As to mobile application development, software development focuses on matters such as user oriented design, mobility and integration with existing applications that offer maximum experience to the user [4]. For the development and assessment of these application, *Software Laboratory* students are provided with multiple high-end and mid-range mobile devices (telephones) with touch screens and sensors (GPS, accelerometer, thermometer), with Android 2.2 and 2.3, Windows Mobile 6.1 and Maemo 5 operating systems.

Health care projects have been developed in the framework of the *Software Production Workshop.* A system was developed to improve interaction among health care network professionals, thus improving primary health care and avoiding the unnecessary referral of infants [5]. Likewise, in *Software Laboratory*, GWAP development was undertaken, for example in the case of the mTagATune mobile application that encourages voluntary work in its users [6]. Teachers in these courses orient development through guides, monitoring tools and links to APIs, online forums and FAQs. During project development, students log their activities through public access wikis (http://wiki.labmovil.linti.unlp.edu.ar) that encourage knowledge and experience sharing.

The aforementioned scenarios promote the students' desire to learn by searching for knowledge in multiple sources. In this sense, discovery learning methods adjust well, generating the need to discover new strategies and knowledge on the basis of already acquired concepts. Teamwork is encouraged in these projects, which generates interpersonal skills, promotes solidarity and leads to a significant amount of pre-professional practice for future engineers and computer scientists.

2.2 Assessment methods

Another key aspect related with the teaching activity is its assessment. All classroom activities are potential assessment activities, which can result in useful information for basic assessment functions: didactic feedback and learning accreditation. For this reason, a number of assessments take place during the teaching sequence:

2.2.1 Initial Diagnostic Assessment

The goal of this assessment is to collect information on the students at the beginning of the course, in order to be able to effectively plan the teaching-learning process.

2.2.2 Continuous Diagnostic Assessment

Its purpose is to facilitate learning, not simply to measure how much students have learned. For this purpose, when some key units have been covered, students are required to hand in individual work based on the topics of the units, with a slightly higher complexity in relation to exercises seen in class. This is an individual assessment and it takes place in the laboratory where the practical applications take place.

As we have mentioned, these works motivate discovery learning and promote self-evaluation, while allowing teachers to diagnose and solve difficulties in the learning process.

2.2.3 Summative assessment

Its goal is to take stock of what the students have learned in the process and it has two main purposes: to check whether a student has reached the minimum goals, and to grade them in accordance with their performance along the course. At this instance, some students complete a project they started during the course, while others choose to develop entirely new projects.

The resulting projects are presented to the teachers and remaining students at the end of the courses. These instances constitute important assessments in the courses.

3. Conclusions

Teaching theoretical bases and analyzing their structural dimensions allows us taking the educational practice farther from a simple concept explanation. We have been able to show that our pedagogical proposal, based on discovery learning and use of new technologies, results in great motivation for the students to learn more about the concepts presented and solve more complex problems.

In Argentina, technology-oriented degrees are an area of great national interest, which is why our students enter the IT labor market in their first years of study. This fact has substantial impact in how long it takes them to graduate, which is a major concern to the teachers of all subjects, specially those in intermediate and upper years, when students begin to be absent from class and more focused on their work activity. The use and development of innovative technologies constitute appropriate and motivating elements to incorporate in software engineering subjects, due to the fact that university education pursues the communication of concepts that have long-term relevance.

Modern technologies allow us to illustrate those fundamental concepts while placing the students in a working environment similar to genuine everyday conditions. In addition, teaching methodologies used in the *Software Production Workshop* and the *Software Laboratory* reflect the interest of the students through high attendance percentages. The quality of the projects that were presented by the students and the feedback obtained through polls bear witness to achievements accomplished.

References

[1] G. Edelstein, Formar y formarse en la enseñanza. *Editorial Paidos*, 2011, ISBN: 9789501261615.

[2] A. Díaz Barriga, Didáctica y curriculum. Editorial Paidos, 1997, ISBN: 968-853-362-9.

[3] M. Prince; R. Felder, The Many Faces of Inductive Teaching and Learning. Journal of College Science Teaching Vol. 36, No. 5, March/April 2007.

[4] J. Díaz; C. Queiruga, et al. Building Mobile Mashup Applications. Some Challenges Encountered in Computer Science Degrees. Accepted for publishing in WorldComp'12, The 2012 World Congress in Computer Science, Computer Engineering and Applied Computing. Currently undergoing registration.

[5] J. Díaz; L. Fava, et al. A Patient Referral and Counter-Referral Management System for Hospitals, HCist 2011, ENTERprise Information System, Parte III, págs. 185-194, ISBN = 978-3-643-24351-6, Springer.

[6] J. Díaz; C. Queiruga, et al. mTagATune: mobile TagATune. ICBM 2011, IEEE Computer Society, ISBN: 978-1-4577-0497-0. Páginas: 331–339.