CREATIVE MATH.
13 (2004), 127 - 132

Software Development Emphasis in Informatics Engineering Curriculum

Cristian Rusu and Jorge Bozo

Abstract. A six-year undergraduate program in Informatics Engineering was initiated in *Escuela de Ingeniería Informática* of *Pontificia Universidad Católica de Valparaíso* back in 1997. Software development process was considered a major topic from the very beginning. That is why four courses were designed to cover it: *Information Systems I, Information Systems II, Software Systems Development,* and *Software Engineering.* The titles of the courses are the same, but the content changed. The *Information System Research Group* of the above mentioned academic unit worked lately to improve not only the content of each course, but especially its logical continuity, all over the four courses. The experience of redesigning the contents, the requirements, the evaluation forms are presented.

1. Introduction

Pontificia Universidad Católica de Valparaíso (*PUCV*) is a private - subsidized very prestigious and long tradition university, being considered one of the best universities not only in Vina del Mar – Valparaiso region, but in Chile. It has more than 75 years of history, maybe not too long in European terms, but quite a long history in South America reality. It continuously developed itself, as academic and research establishment, having nowadays more than 13,000 students, enrolled in over 65 undergraduate (*carreras*) and 40 graduate programs (*diplomado, magister,* and *doctorado*).

In 1972 was created *The Center of Computer and Information Sciences* (*Centro de Ciencias de Computación e Información*) of *Universidad Católica de Valparaíso.* Later, in 1981 it was incorporated into the *Faculty of Engineering*, and became *Informatics Engineering School* (*Escuela de Ingeniería Informática*) in 1982. A four-years undergraduate program of *Informatics Engineering* started the same year, being the first program of the new academic unit. Fifteen years later, in 1997, a new six-years undergraduate program was initiated. The two programs are presently running, *Ingeniería de Ejecución en Informática - INF* (8 semesters), and *Ingeniería Civil Informática - ICI* (12 semesters), with over 650 students.

The research activity inside of *Escuela de Ingeniería Informática* is developed in two working groups, (1) *Information System Research Group*, and (2) *Communications & Distributed Systems Research Group.* A *Master Degree program in Informatics Engineering* will be initiated in 2005.

Received: 20.09.2004. In revised form: 1.10.2004.
2000 *Mathematics Subject Classification.* 68U99.
Key words and phrases. *Computer science higher education, informatics engineering curriculum, software development.*

2. ICI Curricula Structure

We will now analyze the curriculum of the long cycle undergraduate program of *Escuela de Ingeniería Informática*, *Ingeniería Civil Informática* (*ICI*). The ICI program is quite similar to the Romanian long cycle programs in Computer Science (Engineering), existing in the Romanian Technical Universities.

The ICI program normally lasts 12 semesters, but not more than 18 semesters. Actually all Chilean undergraduate programs last an average of 2 semesters more than the similar programs in Romania. The explication consists mainly in the higher complexity of the fundamental sciences programs (*Mathematics, Physics, Chemistry*) in the Romanian high schools. For instance, teaching *Calculus* in the Romanian high schools is a rule, but it is an exception in the Chilean high schools.

The ICI curriculum is divided in 4 areas: (1) *Fundamental Sciences & Engineering*, - FSE (2) *Managerial Sciences* - MS, (3), *Information Technology* – IT, and (4) *Specialized Informatics* - SI. Actually a fifth area is present, consisting of *General Studies* - GS, which complement the formation of the students not only as engineers, but also as persons. The distribution of the above mentioned areas by semesters is showed in table no. 1.

Semester	**FSE**	**MS**	**IT**	**SI**	**GS**
1	12		6		
2	13		4		2
3	15		3		2
4	14		4		2
5	10		7		
6	3	3	12		
7	4	3	11		
8		7	11		
9		7	6	3	2
10			7	6	2
11			7	9	
12		2	4	8	
TOTAL	**71**	**22**	**82**	**26**	**10**
%	**33.65**	**10.43**	**38.86**	**12.32**	**4.74**

Table 1. Distribution of thematic areas of ICI curriculum (number of credits).

As table 1 shows, the distribution of subjects by area is as following: FSE - 33.65%, MS – 10.43%, IT - 38.86%, SI - 12.32%, GS - 4.74%. We included in the SI group the optional subjects, and the projects (that are designated to prepare the graduation thesis). All the optional subjects are computer science subjects.

Joining the groups IT and SI, we obtain a total percentage of approximately 50% specialized preparation in Informatics. So, half of the subjects are designed to prepare professionals in Informatics, and the other half are designed to complete their preparation, in order to be able to work in a wide range of fields, not only as computer scientist, but also as engineers, or managers.

3. Software Development Courses in ICI Curricula

Software development process was considered a major topic from the very beginning of the ICI program. That is why four courses were designed to cover it: *Information Systems I, Information Systems II, Software Systems Development,* and *Software Engineering.* The four courses may be grouped under the generic name of *Software Development* (SD). SD is a subgroup of the IT group. The distribution of the courses by semesters and the number of credits are shown in table no. 2.

Course	**Semester**	**Credits**
Information Systems I (ICI441)	7	4
Information Systems II (ICI444)	8	4
Software Systems Development (ICI543)	9	3
Software Engineering (ICI542)	10	4

Table 2. Distribution of Software Development Courses by semesters.

Each course is an input requirement for the following one. The input requirements for the first course of the SD group (*Information Systems I*) are *Systems Theory* (4 credits) and *Databases* (4 credits). Both courses are "theoretically" positioned into the 6^{th} semester. We say "theoretically" because of the flexibility of the ICI curricula.

The major problems of the SD group courses in the past were redundancy of contents (both in theoretical and practical parts), on one hand, and lack of some important topics, on the other hand. This was basically due to the fact that various professors were teaching the SD courses, each of them having its own vision. Moreover, many of them were part-time professors, and the change of the professor from one year to other was quite common.

The revision process of all SD courses, not only as content of each course, but especially its logical continuity, all over the four semesters, was one of the task of the *Information System Research Group (ISRG)*, formed a couple of years ago. Both authors of the present paper are members of ISRG, and were directly involved into the changing process.

4. Changes in Software Development Courses

The present structure of SD courses is as following.

Information Systems I

- Introduction to software systems development,
- Structured analysis,
- Object-oriented analysis and data modeling,
- Alternative analyses techniques and CASE tools.

Information Systems II

- Structured design,
- Object-oriented design.

Software Systems Development

- Implementation,
- Inspections and tests,
- Maintenance,
- Project management.

Software Engineering

- Software engineering paradigms,
- Feasibility,
- Requirements engineering,
- Software systems design,
- Software evolution,
- CASE tools,
- Software quality.

Two professors are currently teaching SD courses: one is responsible of *Information Systems I, Information Systems II*, and the other one is teaching *Software Systems Development,* and *Software Engineering*. This considerably facilitated the task of redesign SD courses, and brought more coherency to the whole SD teaching process. *Information Systems I* mainly focuses on analysis, as *Information Systems II* mainly deals with the design process. *Software Systems Development* has a practical orientation. These three courses cover all phases of the software development process. *Software Engineering* brings a general overview of the whole software development process, offering alternative approaches, and dealing with main topics as software quality. The present structure follows the cascade paradigm only for didactic reasons. Actually software development phases are not seen as separate stages; on the contrary, there is strong integration, not only at theoretic level, but also at practical level. Students have to develop one complete software system, as course projects, during three semesters, corresponding to the first three SD courses. The evaluation of the final version of the system gives a high percentage of the final grade in *Software Systems Development* course (70%).

As software development is manly a team work, a special emphasis is given to the group work. That is why the project developed during the three semesters involves groups of 2-3 students. This way they are also able to put in practice basic elements of project management.

In order to get a better distribution of theoretical concepts over the whole SD courses cycle, part of the content of *Software Engineering* course was transferred to the *Software Systems Development* course: *Inspections and tests, Maintenance, Project management*. This way the cycle of software development process is closed in three semesters. Meanwhile students are also completing the development of their project (a fully functional software system), which is the basic requirement of approving the *Software Systems Development* course. The above mention transfer left more room for special topics into the *Software Engineering* course: *Software quality, Software evolution, CASE tools*. Someone could think that part of the *Software Engineering* course repeats topics of the previous courses: *Requirements engineering, Software systems design*. There is no redundancy actually, as *Requirements*

engineering chapter focuses on requirements recollection techniques, and *Software systems design* deals with particular cases as *real-time systems, components-based design,* or *user-interface design.* The *CASE tools* chapter illustrates our intention to get to a closer relationship between theory and practice. As part of the *IBM Scholars Program* and the recently established cooperation agreement between *IBM Chile Ltd.* and *Pontificia Universidad Católica de Valparaíso*, we will have full access to IBM software products, for academic purposes. This way we will be able to focus the above mentioned chapter on *Rational Software*, getting it to the practice. Software development is a complex process, which can not be fully studied, even in a four-semester cycle. That is why, apart of the SD group courses, we are also delivering optional courses related to the filed. A good example is the *Human-Computer Interaction* course, which is taught in English.

5. Conclusions

An Informatics Engineering Curricula has to be dynamic, to adapt itself to the continuous IT changes. SD courses are a very important part of the formative process of the future Informatics Engineers. That is why the optimization of these courses, both as content and coherency as a whole, is a constant concern.

Changing curricula is a long a difficult process. Improvements are faster and much easier to implements by a better structure inside of the existing frame. The actual stage is better than a few years ago. Changes that we have made allow a better integration along the four-semester SD cycle. Task is not over yet, that is always room for more. A better structure of these courses remains a priority for the *Information System Research Group* in the future.

References

[1] *** *Reglamento General de Estudios*, Universidad Católica de Valparaíso, Chile, 1998

[2] *** *Reglamento Interno*, Escuela de Ingeniería Informática, Universidad Católica de Valparaíso, Chile, 2002

[3] Bozo J., *Las Asignaturas Optativas y la Actualización de los Curriculum en las Carreras de Ingeniería en Informática*, submitted to VI Congreso Chileno de Educación Superior en Computación (JCC 2004), Arica, Chile, 2004

[4] Rusu C., *Computer Science Higher Education in Chile*, Creative Mathematics (formerly Lucrările Seminarului de Creativitate Matematică), published by Department of Mathematics and Computer Science, North University of Baia Mare, vol. **12** (2003), ISSN 1584-286X, Rumania, 121-128

[5] Rusu C., *Informe Final, Proyecto de Investigación PUCV No. 209.731/2003, Estudio comparativo entre los sistemas de Enseñanza Superior en Informática entre Rumania y Chile*, Pontificia Universidad Catolica de Valparaiso, Chile, 2004

[6] Rusu C., Rusu V., *Business Emphasis in Computer Science Curricula*, Proceedings of Central and East European Conference on Business Information Systems (CEECBIS), "Babes-Bolyai" University, Cluj-Napoca, Romania, May 20-22, 2004, 429-436

[7] Rusu C., Rusu V., *El sistema de enseñanza en informática en Rumania*, Ingenerare, Revista Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, Vol. 15 (2003), ISSN 0717-5035, Chile, 54-60

[8] Rusu V., Rusu C.: *Sistemul de învăţământ superior în Statistică în Chile. Studiu de caz: Universidad Católica de Valparaíso*, Buletinul Ştiinţific al Universităţii de Nord Baia Mare, Rumania, Fascicola ŞTIINŢE ECONOMICE, Seria A, vol. XX (2003), 45-50

[9] http://www.ucv.cl/ - The official web site of Pontificia Universidad Católica de Valparaíso

[10] http://www.inf.ucv.cl/ - The official web site of Escuela de Ingeniería Informática (Pontificia Universidad Católica de Valparaíso)

Pontificia Universidad Católica de Valparaíso
Escuela de Inginería Informática
Av. Brasil no. 2241, Valparaíso, Chile
E-mail address: cristian.rusu@ucv.cl,
E-mail address: jbozo@ucv.cl