



Innovation in research and engineering education:
key factors for global competitiveness
*Innovación en investigación y educación en ingeniería:
factores claves para la competitividad global*

CHANGES IN ENGINEERING CURRICULA: ALGORITHMS AND PROGRAMMING COURSES

Ángela Villota Gómez, Juan Manuel Reyes García, Lorena Castañeda Bueno

Universidad Icesi
Cali, Colombia

Abstract

The algorithms and programming courses are constantly questioned and impacted during the engineering curricula changes. Nevertheless, such changes represent enormous opportunities to include the development of engineering skills such as problem solving and teamwork, as well as knowledge in new technologies that are required in the competitive market. Moreover, the design and implementation of the changes generate great impact for the faculty and the students, thus is a challenge during this process. In this paper we present our findings in the change of the algorithms and programming courses after the construction of our new outcome-based syllabus for the program of Computer Engineering. In this paper we present the results we have obtained during this curriculum change process. They include a description of the problems about our traditional courses, the analysis of the approaches that impacted the decisions we made, and the proposal of a continuous improvement process for enhance the courses.

Keywords: engineering curricula; algorithms and programming courses

Resumen

Los cursos de algoritmos y programación son constantemente examinados y afectados durante las reformas de los programas académicos de ingeniería. Sin embargo, estas reformas representan enormes oportunidades para incluir el desarrollo de habilidades de ingeniería, tales como la resolución de problemas y trabajo en equipo, así como el conocimiento de nuevas tecnologías que son requeridos en el mercado Laboral. Por otra parte, los cambios producidos por el diseño y puesta en marcha de una malla curricular reformada, impactan también tanto a los estudiantes como al cuerpo docente, añadiendo un reto más al proceso de cambio. En este artículo se presentan los resultados obtenidos en el cambio de los cursos de algoritmos y programación después de la construcción del nuevo programa académico basado en competencias para el programa de Ingeniería de Sistemas. Entre los resultados obtenidos en este proceso tenemos: una caracterización de los problemas de los cursos tradicionales, el análisis de los aspectos que influyeron en la toma de decisiones, y la propuesta de un proceso de mejora continua de los cursos.

Palabras clave: *programas académicos de ingeniería; cursos de algoritmos y programación*

1. Introduction

The traditional engineering curricula had included at least one algorithms and programming course in the freshman year in order to provide, among other things, the basis of problem solving skills desired in a professional engineer (Koffman, 1984). However, this course is usually underrated by students (except for those in computer science related programs) due to the lack of an explicit relation with their professional expectations. Over time this course gets disconnected from other courses in the curricula, and in many cases even dated. Consequently, to overcome obsolescence and to rise the relevance of these type of courses, an engineering curriculum is redesign to include new knowledge, affecting the program structure and the syllabuses. This is the case of transition from a content-based to an outcome-based program. Undeniably, there is a strong dependency between the constant evolution of an engineering curriculum and the teaching and learning process. Hence, algorithm and programming courses become a milestone in this process not only for the academic significance, but for the engineering skills in the following courses of the engineering program.

In this paper we present our findings after the construction of our new outcome-based syllabus for the program of Computer Engineering as it influenced the algorithms and programming courses that are offered to all our engineering programs. Among these findings, we highlight the experience of the new syllabus implementation, as well as the changes we have observed in the capability of students while building solutions in different contexts. In our most rewarding results, the algorithms and programming courses had raised relevance among students, not only because of the programming itself, but because it is the first approach students have while facing problem from the engineering point of view, that is, following the stages of analysis, design and implementation.

With this experience, we aim at motivating engineering programs to consider active learning teaching and learning strategies to enhance the students experience and results in the algorithms and programming courses.

This paper is organized as follows: in the 2th section there is a description about the context of the changes in the computer engineering curriculum. Section 3, exposes the challenges of proposing a new syllabus for the first programming courses. In the 4th section we present the aspects that influenced the new syllabus elaboration. In the 5h section we discuss some findings about the integration of the programming course with all engineering programs and in section 6 we expose some concluding remarks.

2. Context and Background

During 2009 the curricula-change committee of the computer engineering program studied the structure of the academic syllabus to identify those key courses with fundamental content for the program. During this evaluation, the Algorithms and Programming academic block was categorized as fundamental for the entire program. At that time, such block was composed by three algorithms and programming courses, and one data structure course. However, as a result of the program intensity evaluation, it was decided to cut the number of courses of this block removing the third algorithms and programming course. To this end, the

content of the original four courses should be evaluated and eventually reorganized to fit the new three courses.

Likewise, any possible change in the algorithms and programming courses must consider the fact that some of these courses are shared with other two engineering programs besides the one being changed. Hence, it was critical to guarantee a positive change without interfering with the learning outcomes of the other programs. Consequently, the first step into this change implied to compare other approaches with similar conditions to ours. Out of this study we summarized the following facts: (1) the structure of the programming courses were more oriented to problem solving than to programming language itself, (2) the teaching techniques were design to achieve progressive goals, and (3) the student's expected results were towards the development of skills as well as knowledge.

The following step was to study algorithms and programming courses approaches that would couple with these findings (i.e, Andes University Cupi2 project [ref:<http://cupi2.uniandes.edu.co>, Teach Scheme Reach Java [ref:<http://www.teach-scheme.org/>]and Microworlds [ref:<http://www.microworlds.com/>]). After this study and somediscussion, the faculty decided to *adopt* and later *adapt* the Cupi2 project into the algorithms and programming block of courses. This project is described in the next section.

The first part of the change was the adoption of this project, which implied to join the Cupi2 community followed by experiencing the dynamic of the class. In order to do that, a delegation of two instructors moved for a couple of weeks to Bogota (which is Andes University city) to take the first courses in the role of students. Later on, all seven instructors gathered in intensive meetings to learn the material and decide the academic details of the course, such as: specific coverage of a topic, evaluation strategies, teaching techniques, projects and assignments, report the progress of their classes, evaluate the students response, and make choices over ongoing events.

The other important part of this change was the adaptation of this project according to the objectives of our curricula and the university professional profile. For this purpose, we study the learning objectives of Cupi2 with respect to our engineering programs, and define the final objectives of each course as well as the proper materials and strategies to support them. Nevertheless, this implementation process requires an iterative and constant evaluation to gradually adjust the elements of the courses, specially to fit the learning process of the students and the particular university plans.

3. The drawbacks of the traditional courses

As mentioned before, the curriculum-change committee decided that the algorithms and programming block of courses was critical and required some adjustments. To decide the proper changes, the committee studied the results of the previous study against the current courses (Report for the faculty, 2010). Equally important, we realized that our current issues on this matter are shared among fellow faculties from other universities who still teach their programming courses based on the traditional courses. As a result of such conventional style we have identified disadvantages listed in the following subsections:

3.1 The whole is more than the sum of its parts

Based on the idea of building computer programs, the ground basis of the traditional algorithms and programming courses had been to teach a particular programming language, technique and paradigm.

Moreover, focussing on the details of a language and how to use them in a computer program. Hence, a course focusing on the programming language is not appropriate to the challenges of today (Sarria, 2009).

During the development of the course, the students repeat and practice assignments without being truly aware of the overall goal of what they are developing. While studying only some parts that are disaggregated, students don't see the whole because they are distracted learning interesting little issues. The result is the frustration due to a lack of vision of the whole problem and therefore ending up with low motivation in the course.

3.2 Technical Details vs Essential Concepts

The teaching of the programming language allows to carry out the practice and testing after accomplishing the analysis and modeling of a solution to the posed problems. However, the teaching of the programming language technical details has occupy a large part of the course time leaving little time to develop such skills. Furthermore, such technical subjects often are studied as main topics detrimenting the study of the main concepts.

In addition, the implementation of isolated procedures to support a concept of an algorithm not always contribute to consolidate the notion of a complete solution to a problem of engineering. Indeed, traditional teaching relies in the instructor's individual decisions and personal feelings towards what is essential and what is detailed. As a result, traditional courses lose objectiveness while achieving the learning objectives.

3.3 About the User Interface

Traditionally, the user interface for these courses had been underestimated while developing programs. Despite the fact that using text console for input and output of messages is in some cases sufficient, it has been noted that today's students can't relate to this type of human computer interface. As a consequence, students don't find the appealing of developing applications.

It is reasonable to think that new technologies had changed the graphic interactions of the users with simple applications. In fact, such interfaces represent an important element while developing a software solution, specially, those within the problems of these first-year courses.

3.4 The lack of connection between the programming courses and a software development courses

It is worth to mention that the knowledge of the software development process is important in the professional profile of a computer engineer. Hence, the curriculum itself should provide a consequent and logical structure to integrate such knowledge. However, traditional first-year programming courses are centered in the programming skills without a proper introduction and cohesion to the development process. As a result, in the following courses it is hard to reveal the importance of such process when the students had learned to develop software without it.

4. The building of the first programming course syllabus

As a result of the changes in the computer engineering curriculum it was proposed an Algorithms and Programming academic block--composed by three courses--and defined as basic for the computer engineering program at Icesi University. In addition, the first course of this block is common to all the engineering programs at the university. As a consequence, the learning objectives, course content, and

teaching methodologies, must guarantee the academic requirements of all these programs. Moreover, students of all these programs are mixed in the same classroom.

Under these circumstances, the elaboration of the syllabus for the first programming course was guided by two reference points: the Cupi2 Methodology for Teaching/Learning Programming proposed by Villalobos *et al* (2005) and the Context in Engineering Education by Crawley *et al* (2008)

4.1 Cupi2 teaching/learning programming methodology

Many of the challenges described in Section 3 are also exposed by Villalobos *et al* (2005), Villalobos and Casallas (2006) among other publications because different universities have characterized or are characterizing the teaching programming problem. These initiatives represent an important advance in the teaching of programming and go in the same direction of interesting initiatives like code.org.

Cupi2 is an Andes University project which aims at finding new ways to tackle the problem of teaching programming considering some known student difficulties like lack of issues for solve a problem, motivational and methodological problems contributing through an active learning approach based on problems and the construction of a teaching/learning pedagogical model, Villalobos and Casallas (2006).

Some of the advantages of such decision include: (1) the Cupi2 project belong to a community with other universities in Colombia, this would provide a collaborative work to enhance the courses, (2) the previous experience of a faculty member brought insights in the dynamicity of the classes to asses the elements that would fit in our model according our curricula changes, and (3) the Cupi2 project defines a structure and methodology acceptable according our expectations, such as, a structure of two programming courses and one of data structures, with a problem-solving learning orientation. Thus, such change in the courses will follow an incremental implementation in which the first step would affect the first course of algorithms and programming. Later on, the second algorithms and programming course, and finally the inclusion of key elements from Cupi2 into our original data structures course.

The Universidad Icesi joined the Cupi2 to the first two programming courses contents and also adapted the teaching / learning process proposed in this methodology. The adaptation was a response to the particular university needs and our experience with students in the application of the methodology in the first year this was applied.

The first change was the course objectives adaptation towards competency-based objectives. The amount of assessments during the course period as well as its structure and depth were important aspects that were changed. The teachers team began then to prepare their own assessment to support these changes. Following the same basic guidelines of Cupi2, we adapted the methodology to change fundamental aspects like place greater emphasis on the analysis and design. Likewise we decided to start earlier the switch of only complete a part to develop the whole solution by the student. Based on the above, the teachers continued developing teaching material including previous study test and more learning reinforcement activities.

4.2 Context in Engineering Education

One of the key aspects within the active learning and the cupi2 methodology was very helpful for sorting out the problem of the first programming course proposal: the context. The context accordingly the Merriam-Webster dictionary is defined as “the parts of a discourse that surround a word or passage and can throw light on its meaning”. There can not be understanding without context, as instructors, we know that our

discourses (called lectures or courses) are in vain if we can't put them in context, that students may pass the courses, but don't will remember anything in the next term.

As proposed in Crawley *et al* (2008), context in engineering education refers to the inclusion of the contextual elements of the professional engineering context that have remain in the past five decades. In this order of ideas, they point out the context elements that must be part of the teaching/learning environment in engineering for achieving motivation to learning because the closeness to their future professional practice. The context elements for the engineering education proposed in Crawley *et al* (2008) are: (1) A focus on the needs of customers (2) Delivery of products and systems (3) Incorporation of new inventions and technologies (4) A focus on the solution, not disciplines (5) Working with others (6) Effective communication (7) Working within resources. All of this context elements are close related with the professional in engineering development, and are part of the CDIO initiative.

5. The Algorithms and Programming courses joint with the context for the engineering education

One of the particularities of the redesign of the first programming course syllabus is that it was intended for being the first of a block considered base for the computer engineering formation but also that first programming course as part of the core of all engineering programs. However the students from the other carriers did not find relevant, important or even interesting a course that they believe is designed just for computer engineers, specially because programming is not in their professional practice. We already have tools against that difficulty, because the cupi2 project was designed for students from different disciplines. But we still have students asking the instructors, and asking themselves: why they should be learning all these things?. Therefore the student motivation and the understanding for the relevance of this course in their education become one of the concerns for instructors.

For this purpose we leave the expression "programming course" from the algorithms and programming course vocabulary, instead we give more importance to the outcome of problem solving naming the algorithms and programming course as a "problem solving course". Thereby we are taking into account some context elements for the engineering education proposed in Crawley *et al* (2008). We have experienced the impact in the student motivation when we change from construction of a program to development of a problem solution. In the middle of this process the students are closer and involved to their professional practice as engineers in the context of the delivery of products and systems, working with others and effective communication.

6. Concluding Remarks

There are many factors influencing the actual changes in the engineering curriculum, especially the programming courses. The difficulties from an old fashioned syllabus, the new needs for the incoming curriculum, the connection between all the courses in the curriculum and the challenge that the new syllabus should be as equal of important and connected with the courses in the curricula of the students from other engineering careers (that wasn't changing).

Curricula changes also influence the teaching and learning strategies. The active learning principle defines the students as active actors in their learning process and the instructor the actor who motivates and assures the learning experience. We have experienced that during the first programming course based on an

active learning methodology, the students recognize the importance of being autonomous, disciplined and acquire the needed abilities for being *active students*, skills that are valuable to every course in their careers.

We highlight the experience of the new syllabus implementation, as well as the changes we have observed in the capability of students while building solutions in different contexts. In our most rewarding results, the algorithms and programming courses had raised relevance among students, not only for the programming itself, but as the first approach students have while facing problem from the engineering point of view. That is, following the stages of analysis, design and implementation.

7. References

Articles

- Crawley, E.F. and Cha J. and Malmqvist J. and Chalmers J. and Brodeur D. (2008) The context in Engineering Education. in proceedings of 4th International CDIO Conference in Gent, Belgium
- Villalobos, J. and R. Casallas.(2006) Teaching/Learning a First Object-Oriented Programming Course outside the CS Curriculum. 10th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts - ECOOP (European Conference on Object-Oriented Programming)
- Villalobos, J., Casallas, R., and Marcos, K. (2005, Octubre). *El Reto de Diseñar un Primer Curso de Programación de Computadores*. XIII Congreso Iberoamericano de Educación Superior en Computación, Cali, Colombia.
- Sarria, G.M. (2009, October). Introduction to Programming for Engineers following the Parachute Paradigm. 39th ASEE/IEEE Frontiers in Education Conference M1C-1.
- Koffman, E., P. Miller and C. Wardle, "Recommended Curriculum for CS1, 1984", Communications of the ACM, Oct. 1984 (27:10), pp. 998-1001.

Electronic sources

- Cupi2 Project <http://cupi2.uniandes.edu.co/>
- Teach Scheme Reach Java <http://www.teach-scheme.org/>
- Microworlds <http://www.microworlds.com/>
- Merriam-Webster Dictionary available at: <http://www.merriam-webster.com/dictionary/context>
- CDIO initiative <http://www.cdio.org/>
- CODE <http://www.code.org/>

About the authors

- **Ángela Villota Gómez**. Bsc in Computer Science, MSc in Engineering from Universidad del Valle. Profesora tiempo completo Universidad Icesi. apvillota@icesi.edu.co
- **Juan Manuel Reyes García**. Bsc in Computer Science, MSc in Engineering from Universidad del Valle. Profesor Universidad Icesi. juan.reyes3@correo.icesi.edu.co
- **Lorena Castañeda Bueno**. Bsc in Computer and Telematics Engineering, Master in Telecommunications and Information Management from Universidad Icesi. Profesora tiempo completo Universidad Icesi. lcastane@icesi.edu.co

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería y de la International Federation of Engineering Education Societies

Copyright © 2013 Asociación Colombiana de Facultades de Ingeniería (ACOFI), International Federation of Engineering Education Societies (IFEES)