International Journal of Engineering Education Vol. 27, No. 1, pp. 14–23, 2011
Printed in Great Britain

0949-149X/91 $3.00+0.00

A Student-Centered Introductory Programming Course: The Cost of Applying Bologna Principles to Computer Engineering Education*

CAMINO FERNÁNDEZ,[1] DAVID DÍEZ,[2] TELMO ZARRAONANDÍA[2] and JORGE TORRES[3]
[1] Escuela de Ingenierías Industrial e Informática, Universidad de León, León 28071, Spain. E-mail: camino.fernandez@unileon.es
[2] Departamento de Informática, Universidad Carlos III de Madrid, Madrid 28911, Spain. E-mail: {ddiez,tzarraon}@inf.uc3m.es
[3] Campus Querétaro, Tecnológico de Monterrey, Queretano 76130, Mexico. E-mail: jtorresj@itesm.mx

The Declaration of Bologna proclaims to change both the academic organization and the way of teaching that are currently implemented in European Higher Education. With regard to teaching methods, an instructional strategy focused on lecturers' teaching is being replaced by methodologies focused on students' learning. However, the application of a student-centered instructional strategy, known as 'active learning', is a formidable task, requiring a great deal of effort from all those involved, plus a suitable instructional process and a set of specific instructional activities. This paper shows the pilot case study of an introductory programming course of the degree in Computer Engineering based on an active learning strategy. Such a pilot case study was carried out to analyze the outcomes of applying the Bologna principles and, based upon those outcomes, to identify the needs and distinctiveness of the Declaration as well as the methodological changes required to implement such principles. During a whole semester, learners followed a continuous assessment process and a student centered instructional strategy that involved an extra effort from both students and teachers. The results show learners benefited from the experience but the system is not entirely prepared to support the new instructional process and several adaptations should be made. Throughout this paper, in addition to the explanation of the pilot study, a set of recommendations, instructional principles, educational tools, and methodological directions will be outlined in order to provide a discussion framework that stimulates and focuses the debate on the application of the Bologna principles to the learning programming activity.

Keywords: Bologna Process; computer education; programming course; pilot study

1. Introduction

Historically, higher education has provided people with the knowledge to fulfill their role as future social leaders. However, it is increasingly recognized that traditional educational models are not sufficient to meet the needs of the new social reality [1]. Because of this situation, higher education institutions are involved in a process of continuous improvement aimed at experimenting with innovative learning methods that allow students not only to acquire knowledge but also to develop appropriate skills and abilities. Responding to this trend, the European Union (EU) has accomplished a set of agreements—such as '*The Declaration of Sorbona*' [2], '*The Declaration of Bolonia*' [3], '*The European Council of Lisboa*' [4], and '*The Declaration of Berlin*' [5]—whose objective is to build a European Higher Educational Area (EHEA) for our society to become a more competitive and dynamic economic area [6]. Such a challenge demands both a new academic organization and new instructional approaches to teach and train in the life-long learning frame [4, 7].

The EHEA context will facilitate the mobility of scholars, graduates and higher educational staff as well as prepare students for their future careers and life-long learning [3]. One of the main priorities of the Bologna Process is to develop a credit system, called European Credit Transfer and Accumulation System (ECTS), which can be used for the purpose of transfer and recognition. ECTS is a student-centered system based on the principle that sixty credits measure the workload of a fulltime student over one academic year [8]. Furthermore, the very spirit of the Bologna Declaration [3] inspires continuous lecturing and outcome monitoring evaluation. This means that traditional lecturing will be transformed in such a way as to require students to study independently. This modification will give rise to several tier adaptations referring to syllabus and curriculum. The Bologna process is a far-reaching reform with many implications and a considerable impact on our society. With the purpose of identifying organizational constraints, methodological limitations and implementation missteps of the Bologna Process, different workshops, seminars [10], official reviews [10, 11, 12] and case studies have been carried out. A pilot study applying Bologna principles to the first academic year of the

* Accepted 2 September 2010.

degree in Computer Engineering at Carlos III University of Madrid was performed.

The pilot study includes a total of nine subjects from the fields of Physics, Mathematics, Computer Architecture and Software Engineering. In this paper, we focus on an introductory programming course taught in the first semester of the degree in Computer Engineering. Our experience in such a course has revealed both positive and negative issues. On the one hand, the results of the course have been improved if compared to those achieved in previous years. On the other hand, the dedication of teachers as well as the amount of instructional resources needs to be increased in order to achieve these goals. In addition, a set of lessons related to the way of both implementing the Bologna principles and applying a student-centered instructional strategy were formulated based on the experience of all key players. The rest of the paper is organized as follows. Section 2 introduces the pilot study, its features, conditions and pedagogical principles. The section describes both the design and conducting phases of the pilot study, focusing on the instructional principles that addressed the design of our course. Section 3 explains the most relevant outcomes of the pilot study. In accordance with the objectives of the pilot study, three main issues will be analyzed: academic results, opinions of participants and the human resources needed to apply the Bologna principles. The results of the course will be compared with similar courses and experiences in order to identify pros and cons of our instructional methodology. Finally, conclusions are drawn in the last section. Such conclusions show a set of facets or characteristics that should be considered when applying a suitable student-centered instructional strategy.

2. Description of the pilot study

The Carlos III University of Madrid has been the first Spanish University to adapt all of its bachelor degrees to the EHEA principles. With the purpose of identifying educational limitations and testing different instructional methodologies, the Carlos III University of Madrid performed a pilot experiment previously to the definition of new curriculums. Such a pilot case study was carried out in the first year of the degree in Computer Engineering. The pilot experiment, called Bologna Project, was not conceived either as a proof of concept or a case-control study, but as a case study that investigated an instructional phenomenon within a real educational context. The following sections describe the design and conducting phases of this pilot experiment.

2.1 Designing the pilot study

A pilot experiment is a small scale preliminary study conducted before the performance of the real situation in order to check the feasibility of such a situation [13]. Before conducting a pilot experiment, its circumstances and surroundings should be designed to guarantee the validity and application of the results. The design deals with three main issues: (i) the objectives of the pilot experiment; (ii) the conditions under which the pilot will be performed; and (iii) the information that should be collected in order to draw useful conclusions.

2.1.1 Objectives

The Bologna Project was designed by the Carlos III University of Madrid before the first new bachelor was introduced. The pilot experiment was carried out at an early stage in order to evaluate in advance non-trivial issues that may make the application of new syllabus and curriculums unfeasible. With the purpose of achieving this objective, two main goals were defined:

- Knowing the reaction of students to the new learning and teaching methodologies. A student-centered methodology represents a break from traditional strategies so that the response of students, their opinions and views, should be collected.
- Assessing the resources needed to start up with the new bachelor. A way of learning based on continuous lecturing and evaluation requires a greater number of teachers as well as specific instructional resources to monitor students. The pilot experiment would allow us to estimate these needs and to compare them with the means currently available.

In addition, the workload of students should be anticipated. The elaboration of a preliminary study in a significant context would allow us to estimate the amount of work required for a student to acquire a number of well-known educational objectives in a given period. In order to achieve this objective, the study of the academic results should be carried out.

2.1.2 Educational context

A pilot experiment is usually carried out on members of the relevant population, but not on those who will form part of the final situation [14]. In particular, the Bologna Project of the Carlos III University of Madrid was performed during the first year of the degree in Computer Engineering. The syllabus and schedule of this degree did not exactly conform to the requirements of the Bologna Process (for instance, the workload of students was not

measured in ECTS). However, the total number of students, their regular profile and the type of subjects involved were particularly appropriate for applying a student-centered methodology. As a consequence of that, the first year of the degree in Computer Engineering was considered to be suitable to carry out the pilot experiment.

A total of nine subjects, divided into two semesters, were selected to conduct the pilot experiment. The target of our work was an introductory programming course given in the first semester of the academic year. The course lasts fifteen weeks with a total of five teaching hours per week –including lessons and laboratory work. The course was taught by a coordinator and eight teachers, with the help of another two extra teachers for monitoring tasks. The course was followed by 141 students, split into three groups. The majority of the students, approximately eighty per cent, were taking the subject for the first time.

2.1.3 Data collected

Finally, once the objectives and the context have been defined, the design phase should determine the data to be collected. In our case, five types of data were considered as relevant:

- Opinion about the course. The opinion of the students about the development of the course should be requested. The aim is to know whether or not the structure, materials and schedule of the course are suitable to take a programming subject.
- Opinion about the instructional methodology. Both instructional activities and assessment tasks should be evaluated by students in order to determine their capability to support and encourage self-learning.
- Workload. The amount of time dedicated to the course by students should be collected. The workload of students should correspond to the number of ECTS credits defined for the introductory programming course.
- Human resources. With the purpose of estimating the human resources needed to apply the Bologna process principles, the amount of time that teachers dedicate to the course should be monitored.
- Academic results. The academic results of participants in the pilot experiment should be analyzed and compared with previous years in order to identify the pros and cons of the new instructional methodology.

Opinions about the structure of the course were gathered using both anonymous questionnaires and personal interviews. In the case of workload and human resources, the estimation of involved participants was directly requested by the course coordinator.

2.2 Conducting the pilot study

A pilot study is often used to test the design of the full-scale experiment; in our case, the design of an introductory programming course adapted to the Bologna philosophy. With the purpose of applying such a philosophy correctly, teachers involved in the pilot study were trained on how the student-centered system should be implemented. According to the instructional experts and pedagogues consulted, four main concerns should be considered:

- Students learn more effectively through interactive instruction rather than passive traditional verbal (lecture) instruction.
- The most significant initiative was that any learning activity that implies an effort for students should be measured. Such a way of learning would motivate students and encourage their participation.
- Feedback should be continuous. Furthermore, the result of assessments tasks should be returned to students as soon as possible. Feedback helps to correct errors and to have a better appreciation of the deficiencies that should be overcome.
- Monitoring activities might observe the evolution of the subject. The final objective would be that teachers identify knowledge gaps among students in order to have a better understanding of potential corrective activities.

Our introductory programming course was designed taking these concerns as a basis. The following subsections explain the pedagogical principles and the instructional process that addressed the design of such a course.

2.2.1 Pedagogical principles

Instruction is a discipline that belongs to the education domain, whose purpose is to ease the learning experience in an intentional and oriented way, to achieve identified learning objectives, which is an important aspect of formal education [15]. To build a learning experience in a deliberate way, instruction must be designed [16]. The following paragraphs describe the instructional principles upon which the instructional design of our course was based. The course was aimed at promoting active learning as a way of implementing a student-centered strategy.

Although a universally accepted definition of the term 'active learning' does not exist, the one proposed by Bonwell and Eison [17] captures its underlying main idea. According to them, active learning could be anything that involves students in doing things and thinking about the things they are

doing. The basic idea is to get students involved in their own learning process, promoting observation, the gathering of information and the autonomous study of the collected data. Instructional activities that promote student active attitudes have been acclaimed as being more effective in learning production terms than traditional lectures where students play a passive role. This point has subsequently been confirmed since there currently exists strong empirical evidence that an active involvement of the student in his/her own learning process has a powerful impact both on deep understanding of the contents and on the long-term retention [19, 20]. Even though there exist many different ways to promote students active learning [18, 21, 22], the key elements for its successful implementation include the introduction of active learning activities in the classroom and the achievement of students' engagement in the learning process. Following this idea, the lectures of our course were always planned ahead, not only to include short exercises but students' interventions as well. Both handed in exercises and students interventions were always annotated and, regardless of their results, student grades always came out slightly higher. In order to promote student engagement, formative assessment techniques were introduced. These not only increase student motivation but also help both student and teacher to identify strengths and weaknesses, and it has been demonstrated that it can improve both learning and exam results [23]. With the purpose of achieving this objective, lecture exercises were promptly corrected and handed back to the student so they obtained continuous feedback about their learning progress. The continuous assessment suggested by the Bologna directives was designed to serve this purpose as well. In this way, learning objectives for each course week were clearly stated, and at the end of every week their attainment was evaluated through a practice exercise. Great effort was made in order to inform students about the exercise results at the beginning of the next week, so in case they had not correctly mastered the previous weeks' learning objectives, corrective action could be taken immediately. Even so, according to the student-centered paradigm proposed by the Bologna directives, it is not only necessary for the students to adopt an active attitude about their learning but they must also become progressively more autonomous. This way, the teacher's role will shift from one of simply imparting information to facilitating the student learning. Following this objective, each of the projects proposed during the course included advanced activities that the student could optionally solve with the help of the extensive bibliography references provided. Students were also encouraged to read around the subject before the relevant lecture, and the lecture would begin with a student outlining the main arguments on that particular issue.

2.2.2 Instructional process

The instruction process is determined by the educational objectives, the profile of students, the instructional strategy, and the assessment method being applied on the course [23]. In addition, in order to identify the instructional domain in which our pilot experiment was carried out, the description of the contents will be included.

A common approach in programming education is first to teach the basics of a programming language and then guide students towards effective strategies for the whole programming process [24]. The aim is to gradually develop skills such as abstraction, comprehension, transposition or application. In accordance with these ideas, a reference programming language must be selected and a set of programming concepts and strategies should be learnt—in particular, our course focused on introducing Java technology and the object-oriented paradigm. The course compiled a set of programming concepts such as data types, control sequences, arrays and strings, as well as basic knowledge of algorithmic and object-oriented design. As mentioned previously, our introductory programming course is taken by students during the first semester of the first year of the degree in Computer Engineering; thus, such students can be defined as novice programmers. In general, students did not have any knowledge of programming or it was very incomplete and jumbled. Moreover, a significant number of students were either not interested or motivated enough. Regarding the instructional strategy, our course compiled lessons and laboratory work. Lessons lasted three hours a week and they were designed to promote the participation of students. Lessons included theoretical classes, such as lectures and tutorials, practice-oriented classes, such as case examples and design exercises, and discussions. On average, one hour of class work required approximately another hour of personal work. On the other hand, laboratory work was made up of several programming exercises that were solved in groups of two students in order to promote teamwork. Such programming exercises were handed in at regular intervals throughout the course. Laboratory work implied around a working time of five hours a week.

Finally, concerning the assessment method, the final grade was obtained by combining the results obtained in theoretical examinations (sixty percent of the final grade) with those from practical laboratory sessions (forty percent of the final grade). The

course included seven multiple-choice exams and eleven practical exercises distributed during the fifteen weeks of the course. Both types of assessments were included in the continuous evaluation. At the end of the semester, students took their final written exam and implemented a small application in Java (around 1000–2000 code lines). In addition, and in order to encourage the participation of students, active class contributions were also measured and considered in the calculation of the final mark; however, this part of the grade was only considered for those students that had passed both the tests and the laboratory work.

3. The analysis of the results

The data compiled through the pilot study was obtained from two different sources. On the one hand, the coordinator of the introductory programming course collected information about the workload of students and the human resources required to carry out a student centered instructional methodology. On the other hand, the heads of the Computer Science Department requested the opinion of students and the academic results achieved in the different subjects in order to determine the usefulness of the pilot study. The following sections show the analysis of the results. The description starts with the revision of students' opinions. It continues with the study of participants' workload, and it ends with the analysis of the academic results.

3.1 The opinion of students

Regarding the development of the course, students were asked about the quality and usefulness of the material used in the organization of the course and the schedule of the subject. Figure 1 shows the average of the results to these questions where at least four out of nine succeeded. Our interpretation is that the deep change in the structure of the course that the Bologna process implies was suitable for the programming subject.

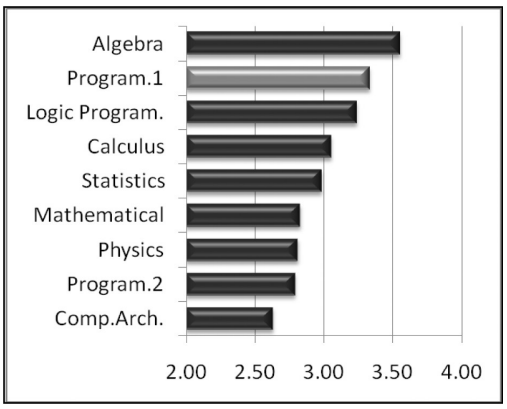


Fig. 1. Students' evaluation: Course quality.

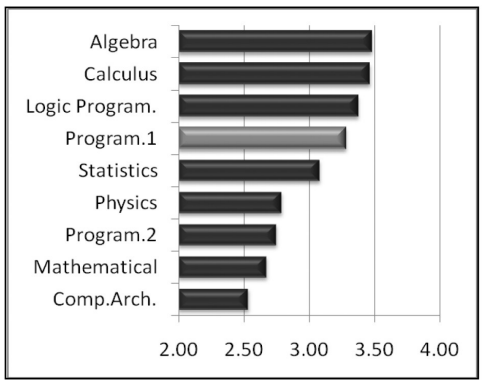


Fig. 2. Students' evaluation: Assessment process.

Regarding the assessment process, students were asked if it had properly measured the skills and knowledge achieved during the course. Figure 2 shows the results to that question and we find that Program.1 is not one of the best, but its results are close to the best ones. Taking into account that a '5-point Likert scale' was used; there is a clear distinction between those subjects below the central value, and the ones above it. Our subject is included in the first group, so we can assume that our assessment process was positively considered by students.

In short, students were asked if they would like to take another course using the same methodology. Figure 3 shows the answers obtained. Despite previous results, considering all the aspects of the course as a whole, the instructional methodology followed in Program.1 is the one that students prefer, even though our course was the one that involved the higher number of working hours per week for students, as is shown in the next section.

3.2 The workload of students

The first semester includes four subjects corresponding to the first four bars (see Fig. 4). During that semester, the estimated number of hours per

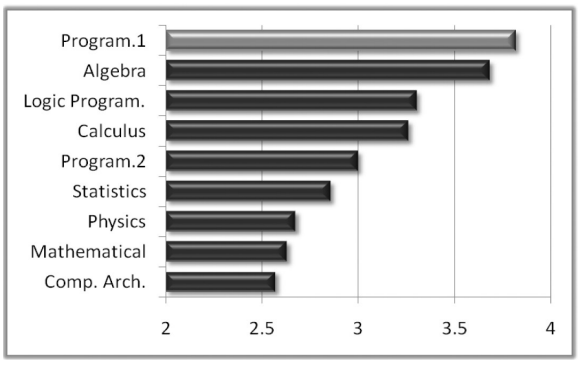


Fig. 3. Student's responses to 'Would you like to take another course using the same methodology?'

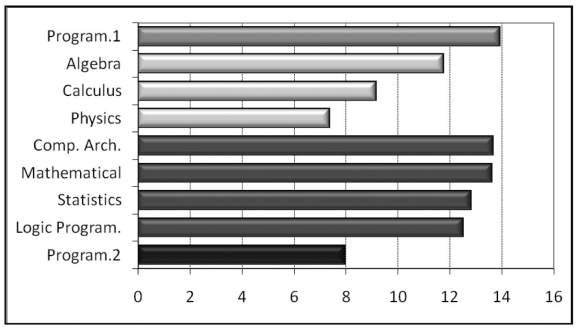


Fig. 4. Students: Working hours per week.

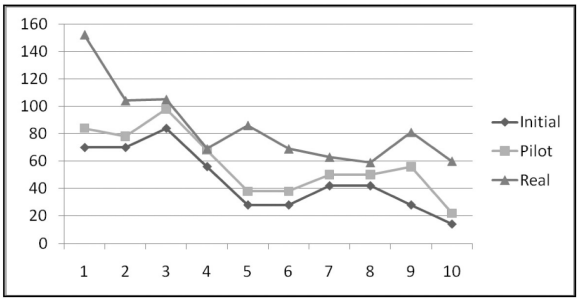


Fig. 5. Human resources.

week for each student was 42.15. In the second semester there were five subjects resulting in a total of 46.89 hours per week. In both cases data exceed the theoretical forty weekly hours, but the excess is around 5% for the first semester, where our subject is included. We can conclude that the load is balanced in this first case, although Program.1 is the most time consuming one for students, but that is a basic assumption for the first programming course in a Computer Engineering degree. The reason lies in the fact that the rest of the subjects—Algebra, Calculus and Physics—are not completely new to students.

The first semester at University is different to the rest of the semesters, especially for engineering degrees. Students are not used to working regularly from the very beginning, and that is one of the main problems. It is only when they fail exams in the first semester that they change their habits, as the bars of the second semester show. In this context, it is even more difficult to try to involve students in the course from the first day for a completely new and relatively demanding subject such as programming. Fig. 4 shows how Program.1 is the only one in the first semester that achieved results similar to the ones obtained in the second semester.

Taking this assumption, in the second semester we find that there is a course with significantly less working hours per week than all the rest. This subject is Program.2, which is considered to be the continuation of our course. This fact can be due to many other factors, but also to the sound basis achieved in the first semester. As we will show later, more than two thirds of students passed the course. The results obtained clearly show that the reason for students preferring the methodology in Program.1 was not inversely related to the workload of the subject.

3.3 Human resources

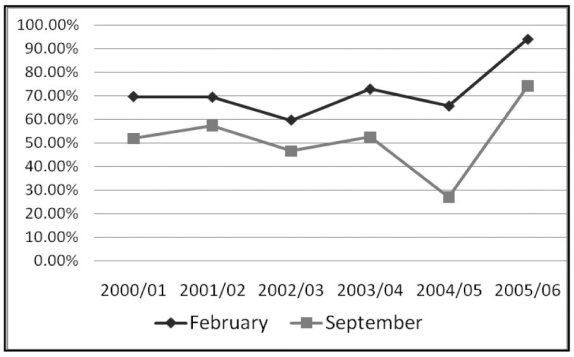
Regarding human resource needs, Fig. 5 shows the total amount of hours dedicated to this pilot project by every one of the ten teachers involved in it. The initial assignment of teacher hours was increased by 17.8% as extra help afforded to the pilot project by the university. However, real data collected shows that the final amount of time was more than doubled (increase of 118.8%). The most important rise was related to coordinating (teacher 1), laboratory work (teachers 5 and 6) and monitoring activities (teachers 9 and 10). And inside monitoring activities, the most consuming one (teacher 10) was dealing with the assessment process.

Besides real data collected, teachers completed a questionnaire in which they expressed their opinion about the extra work that the pilot project had involved for them in the following tasks: preparing course materials, assessment process and monitoring students. Their answer, in the case of our course, was 9.4% below the average for the rest of the courses included in the same pilot case. If we consider this value together with a standard deviation of 13%, we can deduce that the extra work needed in the majority of the rest of the courses was even more than that 118.8%.

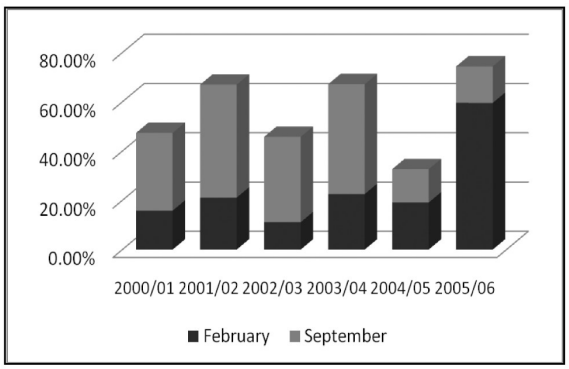
3.4 Academic results

Although it was not the main objective of the pilot study, the academic results of the course were analyzed in order to determine the suitability of a student-centered instructional methodology. With the purpose of drawing conclusions about such a methodology, three different kinds of data were evaluated:

- The results of the course against time. As shown in Figs. 6a and 6b, the performance of the pilot study entailed an increase in the number of students who both passed the subject and took the exams. In particular, two facts should be highlighted: (i) The number of students who passed the exam at the first opportunity (sat in February) was multiplied by three; and (ii) There was a reduction in the dropout rate. Such a reduction was especially significant in the second opportunity (sat in September), which shows a greater interest in the subject. Thanks to the new instructional methodology, not only were students more

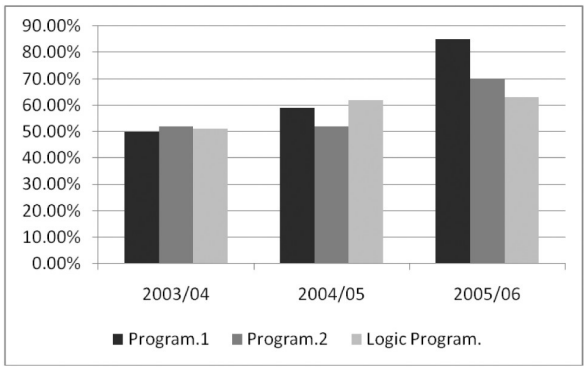


(a) Students taking the exam.

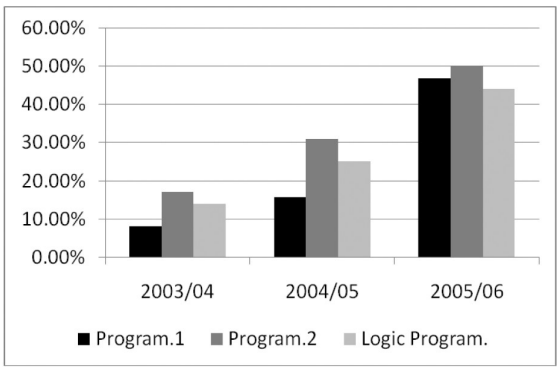


(b) Students passing the exam.

Fig. 6. The results of the course against time.



(a) Students taking the subject.



(b) Students passing the subject.

Fig. 7. The results of the course against time compared with other programming courses.

prepared but they also felt more self-assured about their knowledge and wisdom.

- The results of the course against time compared with other programming courses. Figure 7 shows that until the change in instructional methodology, the academic results of the course were worse than the results obtained in other subjects with the same content and similar characteristics. This situation was aggravated by the fact that our subject is taught in the first semester of the first year, compared to other courses which are taught in the second semester as follow-on courses of our subject. The application of a student-centered methodology helped to reduce the relevance of having previous knowledge about programming.
- The results of the course compared to other subjects involved in the pilot study. Although the existing differences between syllabuses of the subjects avoid drawing conclusions about the way of teaching, it can be concluded that, as shown in Fig. 8, changes in the course had a greater impact on academic results than in the rest of subjects studied. The greatest improvement in the outcomes of the course demonstrates that such an improvement was not only motivated by the performance of the pilot study, and the favorable context created around it, but also by the application of an appropriate methodology to teaching introductory programming courses.

Likewise, even though not shown in the charts below, there was a better performance in all groups in which the subject was divided into, regardless of the number of students or the group of teachers which taught it.

The data collected shows that the instructional methodology applied in the course improved academic results. One possible reason for this lies in the fact that there was a continuous monitoring during

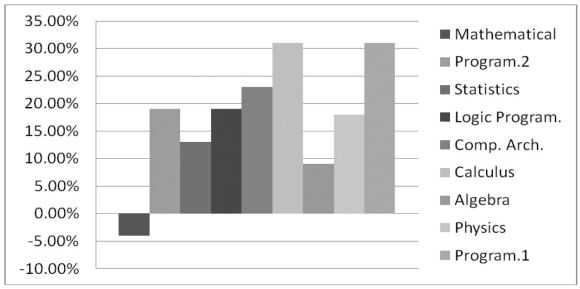


Fig. 8. The improvement of the academic results compared to other subjects involved in the pilot study.

the course. According to diverse authors [24, 25, 26], one of the main constraints to beginners in programming courses is their inability to recognize their knowledge gaps. As a result, beginners have difficulties in solving simple exercises, understanding the cause of their mistakes and determining future learning points. The application of a student-centered methodology allows both students and teachers to become aware of knowledge gaps as soon as possible in order to have a better understanding of potential corrective activities. In addition, students feel more confident about their way of learning, improving their attitude and marks.

4. Conclusions

This paper has presented the results of a pilot study carried out by the Carlos III University of Madrid in the first academic year of the Computer Engineering degree. The pilot study includes a total of nine subjects; however, in this paper we have focused on describing and analyzing an introductory programming course taught in the first semester of the degree in Computer Engineering.

The study pursues several goals to be fulfilled. First, this study aims to determine the reaction of students to the new learning and teaching methodologies that Bologna principles imply. This reaction is especially important for a subject such as programming, which has traditionally represented one of the greatest obstacles for new students, generating significant levels of absenteeism in the classrooms. Second, the amount of human resources assigned to the course should be revised in order to identify new requirements or maintain the actual ones. In order to implement an adequate and continuous evaluation, a review of teachers' weekly assignment to subjects should be considered. And third, the workload of students also has to be measured in order not to exceed their forty weekly hour limit. For the first time, students' work outside the class should be computed, so it should be measured and not only estimated.

The main lesson to be learned from this pilot study is that preliminary studies are crucial in situations like this one, where a whole process changes, and this change involves a variety of stakeholders. The main features of that change such as, for instance, moving the attention to student's effort or emphasizing the assessment process, were clear from the beginning. Nevertheless, a significant number of small but essential practical details which can make a new initiative fail or succeed do not come to light until a real case startup. In addition, even though reduced groups were one of the main issues in the Bologna process, the extra work for the course coordinator should be considered in order to provide either extra help or suitable coordination tools. Regarding the way of teaching programming courses based on a student-centered methodology, four main conclusions could be drawn:

- The course development should be systematic and increasing. Learning programming requires an incremental problem solving process in order to learn the basics of a programming language and to gather the programming strategies in applying these notions: This is already widely acknowledged. But this goal is very hard to achieve, mainly due to the low students' attendance to classes. In our implementation of Bologna principles, the daily attendance measured exceeds 90%.
- Continuous monitoring and tracing are required to recognize students' knowledge gaps and to apply corrective instructional activities. This general principle, if not properly implemented, can have a great impact on the number of hours of individual work for the student. As already mentioned, one of the main constraints of beginners in programming courses is their inability to recognize their own knowledge gaps. As a consequence, the monitoring task has to be performed in a way that allows an immediate response to students' problems, or the burden outside the classroom can rise sharply. In our implementation of Bologna principles, the workload of students stood at an acceptable level.
- The feedback from students should be continuous and elaborative. Feedback helps to correct errors, to have a better appreciation of the deficiencies that should be overcome, and to get students involved in their own learning process. Our formative assessment techniques did increase students' motivation—our methodology was the one they most liked to repeat, and also improved exam results—the number of students that passed the exam at the first opportunity was multiplied by three.
- There is a relationship between the duration of the assessment-process and the feedback efficacy. If the assessment-process is too long, the feedback won't be helpful enough. This is a clear need, and in our study, the assessment task turned out to be one of the more time consuming ones, which makes it difficult to shorten that period of time within the teachers' current weekly schedule.

In summary, a student centered methodology has proved its usefulness in introductory programming courses; however, in order to apply the Bologna principles successfully, the dedication of teaching staff to the subject should be doubled. In the absence of such changes, and to alleviate pressure on existing teaching staff, they should be provided with tools

that simplify the instructional process: Tools to facilitate coordination amongst teachers are important but it is even more essential to improve the whole assessment process.

References

1. J. Delors, I. Al Mufti, I. Amagi, R. Carneiro, F. Chung, B. Geremek, W. Gorham, A. Kornhauser, M. Manley, M. Padrón-Quero, M. Savané, K. Singh, R. Stavenhagen, M. Won Suhr and Z. Ñanzhao, *La Educación Encierra un Tesoro: Informe a la UNESCO de la Comisión Internacional sobre la Educación para el Siglo XXI*, Ediciones UNESCO, París, Francia, UNESCO Edition, 1996.
2. C. Allegre, L. Berlinguer, T. Blackstone and J. Rüttgers, *Sorbonne joint declaration*, Communication, Ministers for France, Germany, Italy and United Kingdom, Paris, May 1998.
3. European Commission, *The bologna declaration*, Communication, European Commission, Directorate-General for Education and Culture, Bologna, Italy, June 1999.
4. European Council, *Agree a new strategic goal*, Communication DOC/00/8, European Union, Lisbon, Portugal, March 2000.
5. European Commission, *Conference of Berlin*, Communication, European Com- mission, Ministers responsible for Higher Education, Berlin, September 2003.
6. European Commission, *Making a european area of lifelong learning reality*, Communication COM(2001) European Commission, Directorate General for Education and Culture, and Directorate-General for Employment and Social Affairs, Brussels, Belgium, November 2001.
7. European Commission, *El papel de las universidades en la europa del conocimiento*, Communication COM(2003) 58 final, European Commission, Directorate-General for Education and Culture, Brussels, Belgium, February 2003.
8. Office for Official Publications of the European Communities. *ECTS Users' Guide*, 2009. http://ec.europa.eu/education/lifelong-learning-policy/doc/ects/guide_en.pdf, Accessed 19 June 2009.
9. Bologna Process. Bologna Seminars 2007-2009. http://www.ond.vlaanderen.be/hogeronderwijs/bologna/Bologna Seminars, Accessed 19 June 2009.
10. ENIC NETWORK Council of Europe/UNESCO. Recognition Issues in the Bologna Process—Final Report, 2001.
11. Bologna Working Group on Qualifications Frameworks. *A Framework for Qualifications of the European Higher Education Area*. Bologna Main Documentation, 2005 http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/050218_QF_EHEA.pdf, Accessed 19 June 2009.
12. P. Zgaga, *Looking out: The Bologna Process in a Global Setting*. Bologna Main Documentation, 2007, http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/WGR2007/Bologna_Process_in_global_setting_finalreport.pdf, Accessed 19 June 2009.
13. M. Haralambos, R. M. Heald and M. Holborn, *Sociology: Themes and Perspectives*. HarperCollins Publishers, Glasglow, 20 March 2000.
14. J. Gerring, *Case Study Research: Principles and Practices*. Cambridge University Press, Cambridge, 11 December 2006.
15. M. P. Driscoll, *Psychology of Learning for Instruction*, 3rd edition, Allyn & Bacon Publishers, Boston, 13 August 2004.
16. S. Dijkstra, *Theoretical foundations of instructional design: Introduction and overview*, in: Instructional Design: International Perspectives Vol. 1: Theory, Research, and Models, R. D. Tennyson, F. S., and S. Dijkstra, Eds., vol. I, chapter 2, pp. 19–24. Lawrence Erlbaum Associates, New Jersey London, 1 August 1997.
17. C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom*, ASHE-ERIC Higher Education Report No. 1. The George Washington University, School of Education and Human Development, Washington, DC., 1991.
18. J. M. Braxton, W. A. Jones, A. S. Hirschy and H. V. III Hartley, The role of active learning in college persistence, *New Directions for Teaching and Learning*, **115**, 2008, pp. 71–83.
19. M. Prince, Does active learning work? A review of the research. *Journal of Engineering Education*, **93**(3), 2004, pp. 223–231.
20. C. Meyers and T B. Jones, *Promoting Active Learning. Strategies for College Classroom*, Jossey-Bass Publishers, San Francisco, 11 May 1993.
21. M. Silberman, *Active Learning, 101 Strategies to Teach Any Subject*, Allyn and Bacon, Boston, 1 March 1996.
22. P. Black and D. Wiliam, Assessment and Classroom Learning, *Assessment in Education*, **5**(1), 1998, pp. 7–74.
23. G. R. Morrison, S. M. Ross and J. E. Kemp, *Designing Effective Instruction*. 4th Edition, John Wiley & Sons, Hoboken, 30 April 2003.
24. A. Robins, J. Rountree and N. Rountree, Learning and Teaching Programming: A Review and Discussion. *Journal of Computer Science Education*, **13**(2), 2003, pp. 137–172.
25. L. E. Winslow, Programming pedagogy—A psychological overview. *SIGCSE Bulletin*, **28**, 1996, pp. 17–22.
26. D. Diez, P. Diaz, I. Aedo and C. Fernandez, C. DEICHECK. Automating the assessment process to improve the informative feedback, *Proceedings of the 38th Annual Frontiers in Education Conference*, Saratoga Spring, 22–25 October 2008, 22 December 2008.

Camino Fernández received her degree in Computer Science from the Polytechnic University of Madrid in 1994, MSc degree in Artificial Intelligence in 1995, and PhD in Computer Science in 2000 from the same University. She joined the Computer Science Department of the Universidad Carlos III de Madrid in 1995, where she worked as associate professor in Computer Science and Artificial Intelligence. In 2008 she moved to University of León, where she teaches courses that include subjects as object oriented programming and design patterns.

David Díez holds an MSc in Computer Science and Technology (2007) and a PhD Thesis in Computer Science (2009) from the Universidad Carlos III of Madrid. From 1998 to 2005, he worked as software engineering and project manager for different multinationals companies. Currently, he works as an assistant professor at the Universidad Carlos III de Madrid. He is co-author of several publications on international journals and communications in international research conferences.

Telmo Zarraonandía holds a degree and a PhD in Computer Science from the Universidad de Deusto and the Universidad Carlos III de Madrid respectively. From 1996 to 2002 he worked in private sector companies from Spain and UK as software developer engineer. He joined the Computer Science Department of the Universidad Carlos III de Madrid in 2003 as assistant professor. He has participated in several projects related to the field of computer supported education and interactive systems. He has published several scientific papers in international journals and conferences.

Jorge Torres received her degree in Computer Systems Management (1994) and Master Degree in Information Systems Management (1998) from the Technological of Monterrey. He hold an MSc in Computer Science and Technology (2005) and dissertation of PhD Program in process from the Universidad Carlos III of Madrid. He joined the Computer Science Department of the Technological of Monterrey in 1998, where he worked as full-time professor in Software Engineering. In 2008 he founded and directs the research group: The Distributed and Adaptive Systems Lab for Learning Technologies Development form Technological of Monterrey. His research interests include technologies to support education and learning, software engineering, web services and distributed systems. He has participated in several projects related to the field of Web Services Enhanced Learning and computer supported education.