Adapting the Course "Quality of Information Systems" to E.H.E.A Guidelines

Mario Piattini and **Félix García**
ALARCOS Research Group
Technologies and Information Systems Department
University of Castilla – La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain
{Mario.Piattini, Felix.Garcia}@uclm.es

Abstract

Currently, organizations depend increasingly on the quality of information systems (IS) for the achieving of their main strategic goals. This is the driving force behind providing Computer engineers with the necessary training. We present some ideas and experiences about how the teaching of the course Information System Quality, which belongs to the 5th year of MSc degree in Computer Science, has been adjusted and adapted so as to comply with the directives of the European Higher Education Area (EHEA). Important lessons and points to consider in the future have come about from the above.

Keywords: Software Engineering, Quality of Information Systems, European Higher Education Area (EHEA)

1. Introduction

Organizations are becoming more and more dependent on information system (IS) quality in the quest to reach their main strategic goals. Given the importance that this topic is acquiring in organizations, Computer Engineers must have the training that is necessary for the tasks required of them. To this end, there are optional courses in a number of universities which provide the possibility of going into more depth in topics related to IS quality, and, to be more precise, in those which are to do with the quality of software.

In the Escuela Superior de Informática –"ESI"- (Faculty of Computer Engineering) of the University of Castilla-La Mancha the course of Quality of Information Systems, with 4.5 ECTS (European Credit Transfer System) credits and which forms part of the 5th year of the Plan for Computer Engineering passed in 1998, has been on the curriculum for the last four years. In this work we present some ideas and experiences on how the teaching of this course has been made to fit in to the guidelines of the European Higher Education Area (EHEA)[1].

Some basic aspects of the course appear in summary form in the next section, with special attention given to its content. We then have a look at some suggestions for considering when the task of adapting teaching methods to EHEA directives is being undertaken and we explain the changes we made in the course to make room for the use of these new methods. Lastly, we present the lessons learned after the first year of experience and some conclusions and matters that may be considered in future academic years.

2. The Course of Quality of Information Systems

The process of change undergone by quality of IS has seen it go from being a treatment which focused essentially on inspection and the detecting of errors in the programs, to becoming a more systematic approach. That is all thanks to the importance that the aspect of quality has taken on in the engineering of systems and of software. In [6] we carry out a detailed study of different proposals for international curricula in relation to IS quality, among which we highlight two: the main "academic" proposal as set out in SWEBOK (Software Engineering Body of Knowledge) [7]; and the most important professional proposal undertaken by the ASQ (American Society for Quality) [2]. We also offered a general overview on the teaching of software/IS in Spanish universities and their correspondence with the areas of importance in the two proposals that we have just talked about. When we came to adapt these proposals to the course of IS Quality, with the EHEA guidelines in mind, we found the need to put forward the list of topics set out in Table 1.

In previous years, we taught all of the above topics to students through lectures. In addition, those attending received handouts of articles on the topics, along with acetates and bibliographic references, should anyone wish to look at the courses in more detail afterwards.

[1] http://www.eees.ua.es/en/introduccion.html

Table 1. IS Quality Course Syllabus

Part 1. Introduction to Quality
1. Concept of Quality
2. Tools Used in Quality
3. Quality Models and Standards
Part 2. Information Systems Quality
4. Quality of Computer Systems
Part 3. Quality of the Product
5. Models of Software Product Quality
Part 4. Process Quality
6. Software Process
7. Life-Cycle Process Models
8. Models of Evaluation and Improvement
Part 5. Other Aspects of IS Quality
9. Measurement of Information Systems
10. Quality of Information

3. Teaching Methods: General Considerations

Theory and research in Educational Psychology tell us that the student ought to be guided towards building up sturdy knowledge structures about the course being studied. What is most important is not so much the quantity of information. The quality of knowledge constructs produced is what really matters, especially if the student is taking on a progressively greater prominence in that construction process.

The majority of experts agree on the fact that a social climate of participation, cooperation, and mutual respect should be fostered in the classroom. In that atmosphere several approaches to group teaching (presentation of work carried out, group projects) ought to be combined with methods focused on the individual. Taking Vygotsky's ideas as a starting point and with the social dimension of learning to the forefront of our minds, we can underline the key role played by group work. It lets students get farther on than when they work all alone as individuals.

When the pupil interacts with his or her peers, he or she has a greater cognitive production. He can express his doubts and clear up any confusion he may have (students teach each other) [8]. Business managers also highlight what a good idea it would be for the university to foster the development of social habits. These would include teamwork, expressing one's ideas in public, the ability to listen actively, with empathy, as well as to be able to ask questions, request help, negotiate, and take receive ideas.

The situation existing at present in the majority of courses for university degrees is, however, very far-off answering the above demands. Traditional teaching models, which are teacher-centered or product-based, are what we observe most.

The changes being made to qualifications and courses, following the directives of the EHEA, aim to improve the overall training and education of the students, as well as that of the teaching staff. Its main focus is students and not the practice of teachers and lecturers.

For several years now, people in the field of Computer Science have been pushing for an educational methodology that emphasizes learning, as opposed to teaching, and which would thus challenge students to think independently [1].

4. Organization of learning in the course of IS Quality.

Faced with the goal of re-adjusting the course of Information System Quality to bring it into line with the EHEA, we have taken on board the greater part of the suggestions and techniques described in the above sections. Our aim has been to offer student-centered education, with a focus on understanding and transfer of knowledge, rather than on mere repetition and application.

To that end, we gave lectures on the content set out in Table 1 in the first two weeks. The point of this was to give the student the foundational knowledge required for him or her to understand the course content, as well as offering that individual an overview of IS quality. A complementary reading list was provided, in case the student wished or needed to go into the content in more depth.

Students also received four articles, which they were to read outside class and they were told that the articles would be commented on during the presentation of the corresponding topics. In what could be considered laboratory classes, we got the students to work in groups of 4/5 on basic techniques of quality (on Ischikawa's diagram and the process control graph) applied to software. We made them see the difficulties of working in groups and commented on the need for them to "choose" their possible group colleagues. These would be the people with whom they would carry out the work that would be at the centre of their learning throughout the rest of the duration of the course.

In addition, students had put before them a job offer that had appeared in the newspaper a couple of weeks before the beginning of their introduction to this course. This advertisement was looking for experts in assessment and improvement of software processes. They were also told about similar tasks being carried out in the Soluziona software factory in Ciudad Real (Spain) as well as about work being undertaken by the Novasoft firm in Castilla-La Mancha and which had been presented a couple of months beforehand in a talk given to the Faculty of Computer Engineering by the firm's technical manager.

By doing all this we aimed to motivate the student, making him/her see the competitive advantages which the course could offer them in the IT market, with reference to the knowledge they possessed up until that point (this is

especially true for Software Engineering, as this is the course that has most to do with IS Quality).

In an attempt to simulate a work setting, and taking the job offer in the newspaper as a starting point, we did a mock recruitment of students in the firm, offering them the possibility of working in five work groups. The groups are Group A: ISO 9000/90003; Group B: EFQM; Group C: CMMI stage representation; Group D: CMMI continuous representation; Group E: ISO 15504.

The work groups set out to consist of five "engineers" /students in each one. Their goal was to prepare an "assessment framework" to measure the maturity of the software processes of the organization, based on the standards and norms which were applicable in each group.

When it came to setting up the groups, we let the students be the ones to choose their workmates. We did force the groups to be made up of five students, since it meant that the kind of problems they had to face are not the same sort as when they work in pairs. Working in groups of five students gives them additional learning content such as negotiation techniques, the ability to coordinate, cooperate, plan out tasks, and so on. In pair work one person tends to dominate and the need to develop those skills doesn't arise.

We observed, however, that we were not able to fulfill all the recommendations given by Felder and Brent about cooperative work [5]. This was because we were not able to set up groups with heterogeneous ability levels: this fact may have led to the formation of student groups in which all of them were "the better ones", while other groups were composed of weaker individuals. Neither were we able to form student groupings in which women (Felder even talks about "or other minorities") made up a significant proportion. In the course of IS quality, this is just not possible because 29 students enrolled and only two were women. This low percentage of women in engineering studies is an important problem: in the case of computer engineering, the situation in this sense is dramatic.

Students had to meet up to work together on many different occasions and as many of them live hundreds of kilometers from the Computer Engineering Faculty (University of Castilla-La Mancha). Because of this, we thought it would be better for them to organize their groups according to the relative ease of getting together with workmates (in addition, of course, to considering with whom they got on better). There were 29 students enrolled for this course: 21 chose to follow the course in the form that had been adapted to the EHEA and two students asked for an exam just as the teaching guide stipulates (when this guide was produced it was not known that the course was going to be chosen for adjustment to EHEA guidelines). The rest of the students neither attended classes, nor sat the exams. Of those who chose to follow the version that had been made to fit in with the EHEA, four groups of 5 students were formed (A, C, D and E), while there was one student assigned to group B.

We should note that in our case we have adopted the "cooperative base group" type, rather than "formal cooperative learning groups". The reason was that we preferred the group to stay together for the whole course. Even though the second option might have been more enriching for students, we really believe that it was better to keep the group intact, since we had just one term to work in.

We stressed to the students that the important thing was to reach the objective of the project they we assigned them: to prepare an "assessment framework" for the maturity level of the software processes of an organization. In order to give them guidance towards the achieving of this goal, a spiral development was proposed to them, based on four stages:

1. Give an oral presentation for 30 minutes (with slides only) of the norm /standard that corresponds to each work, with the aim of analyzing the basis on which to build the assessment framework. This would be for after Easter (the beginning of April).
2. Give an oral presentation for 60 minutes (with slides only) of the assessment framework of each group-this would be for the beginning of May. The remaining groups present, as well as the teacher, could make comments and give the feedback necessary for the framework presented to be refined.
3. Give in a detailed document about the assessment framework.
4. Hand in an analysis report to the company management about the assessment frameworks produced by the other groups.

5. Lessons Learned

First, it ought to be underlined that, this being the very first year in which students faced this kind of learning methods, they found themselves a little disorientated. Not all of them understood what we were going to ask them.

In a sense, the feeling of safety into which they settle in the traditional system is threatened (in the exam they are only asked about what the teacher /lecturer has explained in a class/lecture and about which they have notes).

Students have remarked that the new method seems more appropriate to them. It brings them closer to the "real world" and, as they are in fifth year, this seems more worthwhile to them than yet another exam. They also express the opinion that this system could work with students from the final academic years of a University degree course, but not with first year students who are still much more in the state of "not knowing know where they are".

Another comment they made is one that we see repeatedly in the literature on this issue. It is that the groups need to manage themselves and that they feel loaded-down with this burden.

Although it has been proven that cooperative learning raises self-esteem and contributes to developing positive

attitudes such as respect, mutual aid and collaboration, we ourselves have not been able to demonstrate this.
However, we have seen an increase in teacher-student communication as well as a greater involvement of students in the course. It must be remembered that, as Bamberger and Hook point out, "software quality is not usually seen as a glamorous concept, though its growing importance is recognized" [3]. This new type of teaching makes for a higher level of engagement from the students than they had shown before.

Apart from the lessons mentioned above, there are still a number of questions that it might be a good idea to tackle in future editions as: to simulated the work setting more faithfully, as regards distribution of tasks (doing this in more depth and clarity); marking out roles explicitly within the groups, allowing them on a rotating basis; to give some kind of lesson on techniques for improving group performance, as well as on time management- this latter aspect is one that some students are not trained in; to perform some kind of study on the personality of the students in order to understand the differences in learning styles and to produce teaching methods that are appropriate to different personality types [4] would be something else to attempt, too.

6. Concluding Remarks

We can observe that "teaching is not just about the transmission of knowledge", since this would imply that the learner plays out a passive role as a simple receptacle and that is certainly not what we want. This aspect becomes even a more serious issue in the case of Computer Engineering; still more so in areas such as IS Quality. These fields are in a state of ongoing change. Not only are new features being incorporated every day, thanks to the way IS is constantly taking giant leaps forward and to its mass use by all sorts of organizations- IS also sets up models for other areas at a pace that is difficult to take in or keep up with.

Only those students who are mature and who are sure of their bearings will be able to face up to the challenges this situation places before them. That is why we are fully convinced that new learning techniques which are active, collaborative and project-based must be used.

In spite of the hindrances that we may find at the level both of culture and of infrastructure, we believe that the guidelines of the EHEA represent a unique opportunity for us to achieve the incorporation of this type of technique in our courses. We trust that this input will manifest its effects in the computer engineers that we are involved in training.

We should remember that adjusting teaching to the EHEA makes the evaluation of the teaching program by the teacher him/herself an even more pressing need. We are talking about an assessment that should be a meta-cognitive process and regulatory in character. The teacher ought to re-design his/her program progressively, according to the information coming from students and circumstances.

References

[1] ACM, "ACM Computing Curricula 2001". Computer Science. Final Report, The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, 2001.
[2] ASQ. Certified Software Quality Engineer. American Society for Quality. En http://www.asq.org/cert/types/csqe/bok.html, 2005.
[3] Bamberger, J. y Hook, J. "Teaching Software Quality and Leadership: Experiences and Successes", *Proc. of the 1995 Asia Pacific Software Engineering Conference (APSEC´95)*, pp. 333-343, 1995.
[4] Capretz, L.F. "Implications of MBTI in Software Engineering Education", *SIGCSE Bulletin*, ACM 34 (4), 2002, pp. 134-137.
[5] Felder, R.M. y Brent, R. "Cooperative Learning in Technical Courses: Procedures, Pitfalls and Payoffs", ERIC Document Reproduction Service Report ED 377038, 1994.
[6] García, F. y Piattini, M. "Propuesta de Contenidos para asignaturas sobre Calidad de Software y Sistemas de Información", *XI Jornadas de Enseñanza Universitaria de la Informática (JENUI´2005)*, Madrid, 14-16 de julio de 2005, Thomson.
[7] IEEE. "SWEBOK. Guide to the Software Engineering Body of Knowledge", *IEEE Computer Society*, 2004.
[8] McKeachie, W.J. "Teaching Tips, Strategies, Research and Theory for College and University Teachers". 9ª ed., Lexington. Heath and Co, 2002.