

Web-Based Teaching in Software Engineering

N. Papaspyrou, S. Retalis, S. Efremidis, G. Barlas and E. Skordalakis

*National Technical University of Athens
Department of Electrical and Computer Engineering
Division of Computer Science, Software Engineering Laboratory
Polytechnioupoli, 15780 Zografou, Athens, Greece.
E-mail: {nickie,retal,sef,gbarlas,skordala}@softlab.ntua.gr*

Abstract

The introduction of the new technologies of computer networks and hypermedia systems in education seems promising. However, only through experimentation can the effectiveness of these technologies be demonstrated. This was the main objective of the EONT project, in the process of which the National Technical University of Athens adapted an introductory course in Software Engineering to a novel enriched instructional delivery mode. The existing course material was supplemented by web-based courseware, integrated into a web-based novel networked learning environment. In this paper we report on the results of our research and development, concerning this particular course, and discuss the results that were obtained from our evaluation study.

Keywords: Software Engineering, web-based courseware, enriched instructional delivery mode, novel learning environment.

1. Introduction

Software construction is nowadays in great demand and people need be educated in it.¹ *Software Engineering* (SE) is a branch of engineering which uses computer science and other branches of science and engineering to offer people a methodology for constructing software, in a way that assures the quality of the resulting product and completion of the construction task within predicted estimates of cost and schedule. Many universities world-wide, including the National Technical University of Athens (NTUA), offer courses in SE.

The Department of Electrical and Computer Engineering at the NTUA offers an introductory course in SE since 1986, targeted primarily to ninth semester students. The course components are: lectures, discussions, projects and study material. There is a two-hour lecture session every week and one additional hour is allocated to discussion on the course material. Students gain a lot of experience by working collaboratively in small groups on reasonably large projects. The study material consists of a textbook² entitled “An Introduction to Software Engineering”, written in Greek, as well as review papers from the recent literature.

However, the traditional mode of teaching faces a number of major problems.³ Lecture attendance, which is traditionally not compulsory in most Greek universities, decreases steadily and the percentage of students currently attending the lectures is less than 50%, mainly because many students have already started engaging in professional activities. It is difficult for students to communicate with the instructor (e.g. ask questions and receive feedback) outside

the classroom or the instructor's appointed office hours, although many of them would prefer it. Moreover, the curriculum of the course changes so rapidly that the textbook quickly becomes obsolete.

Recent developments of Information and Computer Technology (ICT) in the fields of computer networks and networked hypermedia systems promise attractive solutions to the above problems. It is now possible to store the instructional material in a central computer and allow it to be accessed by many students through personal computers connected to a local or wide-area network. The problem of interaction can be tackled with the use of synchronous or asynchronous communication.⁴ The use of ICT-based *Open and Distance Learning* (ODL), as an instruction delivery method supplementary to the traditional one, is a way of overcoming the problems of the latter.

We had the opportunity to make an experiment on the effectiveness of ICT-based ODL methods for enriching classroom-based instruction⁵ within a European partnership project entitled "EONT: An Experiment in Open and Distance Learning using New Technologies", partially funded by the European Union under the Socrates programme. Seeking primarily a complement and not a substitute to traditional education, we developed web-based courseware for the existing course "An Introduction to Software Engineering", i.e. a set of learning resources specially designed for the world-wide-web.

The students were also provided with printed material (textbook, up-to-date papers, etc.) and their assessment was based equally on the quality of the team project and the result of the final exam. As a supplement to the lectures and printed material, the students had access to the web-based courseware which contained an enhanced version of the course material in electronic form and useful links to relevant material on the Internet. Moreover, asynchronous communication facilities were offered via a web-based discussion forum and class management was integrated in our web-based learning environment.

In this paper we present our approach to teaching the course "An Introduction to Software Engineering" in our novel web-based learning environment. The paper is structured as follows. The philosophy adopted when designing the contents of the online course material is illustrated in section 2. Section 3 describes the structure of the enriched instructional delivery mode. Section 4 describes the web-based courseware with special emphasis on the online course notes and section 5 presents the case study, which is an important part of the web-based courseware. Finally, section 6 presents results from the evaluation of the enriched instruction delivery mode, and section 7 contains some concluding remarks.

2. Philosophy of the course

People have needs and, in order to satisfy them, must carry out various processes. In the execution of these processes they are assisted by tools and machines, which they construct for this purpose and integrate into systems that we call *artificial*

systems. Thus, artificial systems are human constructions, typically built in a staged fashion as shown in Figure 1a.

According to this figure, the various processes that need to be carried out in real life come out as executions of problem solutions. These solutions have three broad parts: the human part, the soft part and the hard part, carried out respectively by people, computers and various machines. It is the job of a domain expert to formulate the problem and solve it, based on the needs that must be satisfied, and it is the job of a systems engineer to divide the problem solution into the three aforementioned parts. Then, in collaboration with a human engineer, a software engineer and a hardware engineer, who construct the three subsystems, the systems engineer integrates the parts and the whole artificial system is formed.

The soft part of a problem solution defines an information process, which accepts data as its input, processes them and produces again data as its output. Information processes are automated by software systems and this kind of automation is valuable, since such processes are tedious, tiring, unpleasant and time consuming. Software Engineering is the technological branch concerned with the construction of software systems, which can always be thought of as components of larger artificial systems.

This view is taken in our Software Engineering course. Furthermore, it is accepted that Software Engineering accomplishes its mission by translating a description of the soft part of a problem solution from a language directed to humans to a language directed to computers. This translation involves the development processes that are shown in Figure 1b, as well as other processes that are required for management, integration, etc. All these processes together form the *software process*.

The subject matter of Software Engineering encompasses the software process and the methodologies that are needed for carrying it out effectively and efficiently. One such methodology, based on the above assumptions, has been specially developed for our course. This methodology is named YASM (Yet Another Software Development Methodology) and is used for the development of team projects, that is, the course components aiming at skill acquisition.

3. Enriched instructional delivery mode

In this section we describe the enriched instructional delivery mode employed in the course “An Introduction to Software Engineering”. The term “enriched” denotes that ODL methods were used as a supplement to the ex-cathedra delivery vehicle. The aim of the enriched mode was to provide alternative learning paths, based on the advantages of the ICT-based ODL, and improve the overall learning effectiveness.

The enriched instructional delivery mode has several advantages over the traditional mode:

- Students can progress *at their own pace*⁶ and study the instructional material in the order that best suits their abilities or preferences;
- The learning material is stored online and the course is “open” at any time and from any place for the students registered in it.
- The teacher plays the role of a facilitator and assists the learning process rather than imposes his wisdom on the students.⁷

Special emphasis has been put on integrating the web-based courseware into a learning environment, called *Novel Networked Learning Environment* (Nov-NLE).³ At a physical level, Nov-NLE is a distributed system and uses the client/server model.⁸ There is a server, in which all material is centrally stored, and many clients, one for each user accessing the instructional material through the Internet and not necessarily located in one place.

Nov-NLE consists of the following subsystems:

- The administrative and management subsystem, where all data necessary for system’s organization (list of students, of instructors, messages, etc.) are kept.
- The presentation subsystem, which is responsible for the presentation of the web-based courseware to the users. Personal and group-based annotation mechanisms also exist as electronic metaphors of writing on the margins of a book.
- The communication subsystem, which is responsible for enabling the human-human interaction. Nov-NLE facilitates asynchronous communication by two means: an integrated e-mail mechanism and a web-based discussion forum, which involves threaded communication on specific topics.

4. Web-based courseware

The aim of the course is to convey knowledge and skills to the students and enable them, after a successful attendance, to work as novice software engineers. The objectives of the course are to allow students to familiarize themselves with inherent difficulties in software construction, get an understanding of the software development methodologies, and gain enough practical experience through involvement in team projects. The web-based courseware consists of the following components:

- Online course notes (hypermedia metaphor of a textbook);
- A study guide;
- A case study;
- A course description; and
- Specimen of past exam papers with their solutions.

The content of the online course notes, which is shown in Figure 2, has been organized in self-contained blocks according to the rationale of the UK Open University.⁹ Each block corresponds to specific instructional objectives and is

divided in units, each of which includes learning activities (questions, exercises, discussion topics) that allow students to check the extent to which they have understood the material and to discuss with fellow students or tutors various course topics.

The most important component of the web-based courseware is a case study, in which the YASM methodology is demonstrated in practice. The case study was considered essential for showing the students how to apply this methodology to the construction of realistic software systems. During the course, team projects of moderate size and difficulty are carried out by small teams of students using this methodology. The case study is concerned with the development of a software system for assisting the issuance of the monthly bill for the shared expenses charged to the apartments in an apartment complex.

The case study is structured as two parts: a front end and a back end. The front end focuses on the problem at hand and its solution. It consists of the definition of needs, the problem definition, the problem solution and its division in human, soft and hard part. The software requirements specification is also included in the front end. The back end focuses on the design and implementation of the actual system. This structure exposes the underlying philosophy of the YASM methodology. The front end is the *static* part, i.e. the part which describes the problem solution and is not bound to any design or implementation decisions. On the other hand, the back end is the *dynamic* part, i.e. the part which depends on design and implementation decisions, as well as the actual run-time environment. Several back ends can be constructed, based on the same front end.

It is interesting to point out that the software requirements specification is part of the front end although it contains, among other things, the user interface specification. It is crucial that the specification of the user interface defines the functionality of the human-computer interaction without depending on any specific implementation environment. Instead, the choice of implementation environment should be based upon this specification.

Three back ends have been implemented for three different implementation and run-time environments: Microsoft Access, ANSI C, and Microsoft Visual C++. The purpose behind the three different implementations is to stress the argument that the back end can be implemented in various ways, and that it is the responsibility of the software engineer to choose the best implementation under certain considerations.

The staged development which maps the soft part of the problem solution to software requirements specification, then to a software design and finally to an implementation has been made explicit in the case study. The data and the processing steps of the solution are shown to be mapped to more concrete forms as we progress to design and coding.

5. The evaluation study

A first version of the enriched instructional delivery mode was tested and evaluated in the autumn semester 1996-1997. This section describes the evaluation method followed and the results extracted.

5.1 Subjects

50 students (15% women and 85% men) out of the 61 registered for the course responded to the evaluation study. Since the focus of the evaluation study was not to make a comparison between the traditional lecture-based teaching mode with the enriched one, there was no division of students into focus groups. All students that participated in the course were subjects to this study regardless if they used the ICTs for learning. They were asked to justify their choices and their criticism on the enriched mode.

The majority of the subjects had previous experience in using computers but none of them had ever used ICT for learning. In terms of the time spent working with the Nov-NLE system, 48% spent less than one hour per week, 38% spent 1-2 hours per week and 14% spent 3-4 hours per week.

5.2 Evaluation method

The aim of the evaluation study was to examine the learning effectiveness of the enriched instructional delivery mode, the factors that affect the effectiveness, as well as overall criticism on the delivery mode.

For the purpose of this study, two instruments integrated into one questionnaire for data collection were developed: the first instrument was based on a number of standardized questions, reflecting the previously stated theoretical framework, whereas the second one on a number of open-ended questions, reflecting likes and dislikes, problems identified, suggestions, etc. Univariate and multivariate data analysis was performed for the quantitative part of the study and content analysis for the qualitative part.¹²

The criteria for the selection of items composing the variable “learning effectiveness” were:¹³

- *Criterion of appropriateness*: the course and its delivery mode were appropriate to the student body for which it was intended (preparedness for study, expectations, aims and objectives, etc.)
- *Criterion of engagement* (the student found the course and the course delivery interesting, challenging, flexible, etc.)
- *Criterion of students' performance* (students increased their knowledge, skills, developed new habits, etc.)

5.3 Evaluation results

The average score of the learning effectiveness measurement was 3.25 on the five-point Likert scale, where 5 was coded as the highest subjective effectiveness perceptions and 1 as the lowest. On the one hand, considering that the developed web-based courseware was at its first version, this score is relatively high. On the other hand, the effectiveness of the enriched instructional delivery mode reached only a moderate level. This is explained largely by the attachment of students to more human-centred modes of instruction. Most of the students admitted that they prefer to communicate and collaborate face-to-face than via ICTs. However, they mentioned that the ICTs allowed them to work on the course at their own pace and be informed about the course without constraints in time and place.

Secondarily, the preference of the students towards the traditional lecture-based teaching mode was explained by problems related to the design of the web-based courseware as well as the development and administration of the Nov-NLE system. The students expressed the need for more examples in the web-based courseware and especially in the online course notes, which were criticized as lacking interactivity with the learning material. The students either preferred to read the printed textbook or to select some parts of the online course notes and print them.

The analysis of the qualitative data also showed that students found the enriched instructional delivery mode easy and intuitive to follow. The case study was far more useful than the online course notes because it gave the students a hands-on experience with the development and documentation of a relatively large project.

It was also consistently evidenced that, due to the potential for periodic revision, extension and update of the web-based courseware, this instructional mode would provide greater flexibility for meeting the identified educational needs and a variety of learning styles in the future.

Respondents rated highly the need for the integration into the delivery mode of electronic seminars as a form of computer-mediated collaborative learning method as well as the development of an online library of Internet sites and online journals, in which they would find relevant learning material. Other comments more specific to the course material were gathered and analyzed in order to make revisions and develop a more effective instructional mode.

6. Concluding remarks

Teaching is not an easy task. It becomes even more difficult when the subject matter is still evolving, as is the case of Software Engineering. Additionally, contemporary lecture-based teaching has been criticized as having many drawbacks. In particular, conventional universities have been denounced for the constraints they impose on time and place of instruction delivery. Open and Distance Learning seems to be a means for overcoming these constraints and contributing to the improvement of the state of higher education. Higher education

can profit a lot from the use of new technologies in computer networks and hypermedia systems.

Only through experimentation is it possible to have an answer to this expected benefit. The development of web-based courseware and its integration into a web-based learning environment were the cornerstones of an enriched instruction delivery mode. This mode was proven to be beneficiary to the NTUA students who participated in the experiment. Although it was difficult to draw a firm conclusion about the effectiveness of the use of an enriched instruction delivery mode, a tentative conclusion is that Software Engineering in higher education will benefit a lot from development efforts capitalizing on the strengths of new technologies.

Acknowledgements

The EONT project has been partially funded by the European Commission's SOCRATES Program, under the reference number TM-OP-1995-1-GR-88 (1/0).

References

1. Gibbs, N. E. The SEI education program: The challenge of teaching future software engineers, *Communications of the ACM*, 1989, **5**.
2. Skordalakis, E. Software engineering teaching at NTUA, in *Proceedings of the First International Conference on Software Engineering in Higher Education*, ed. G. King, C. Brebbia, M. Ross, G. Staples, Computational Mechanics Publications, 1994.
3. Koutoumanos, A., Papaspyrou, N., Retalis, S., Maurer, H. & Skordalakis, E. Towards a Novel Networked Learning Environment, in *Proceedings of World Conference of Web Society (WebNet'96)*, San Francisco, USA, November 1996.
4. Hilz, R. *The virtual classroom learning without limits via computer networks*, Ablex Publishing Corporation, USA, 1995.
5. Papaspyrou, N., Koutoumanos, A., Maurer, H. & Skordalakis, E. An Experiment in ODL using New Technologies, in *Proceedings of World Conference of Web Society (WebNet'96)*, San Francisco, USA, November 1996.
6. Marshall, A. D. Developing Hypertext Courseware on the World Wide Web, in *Proceedings of World Conference on Educational Multimedia and Hypermedia (ED-MEDIA 95)*, Graz, Austria, July 1995.
7. McCormack, C. & Jones, J. D. *Building a Web-based Education System*, Wiley Computer Publishing, 1997.
8. Sinha, A. Client-server computing, *Communications of the ACM*, 1994, **35**(7).
9. Coats, M. *Learning how to learn*, Kogan Page, Open University, 1995.

10. IEEE recommended practice for software requirements specifications, *ANSI/IEEE Std. 830*, 1992.
11. IEEE recommended practice for software design descriptions, *IEEE Std. 1016*, 1987.
12. Makrakis, V., Retalis, S., Koutoumanos, A., Papaspyrou, N. & Skordalakis, E. Evaluating the effectiveness of an ODL hypermedia system and courseware at the National Technical University of Athens: A case study, *Journal for Universal Computer Science*, **4**(3), pp. 259-272, March 1998.
13. Chambers, E. Course evaluation and academic quality, in ed. F. Lockwood, *Open and Distance Learning Today*, Routledge, London, pp. 342-352, 1995.

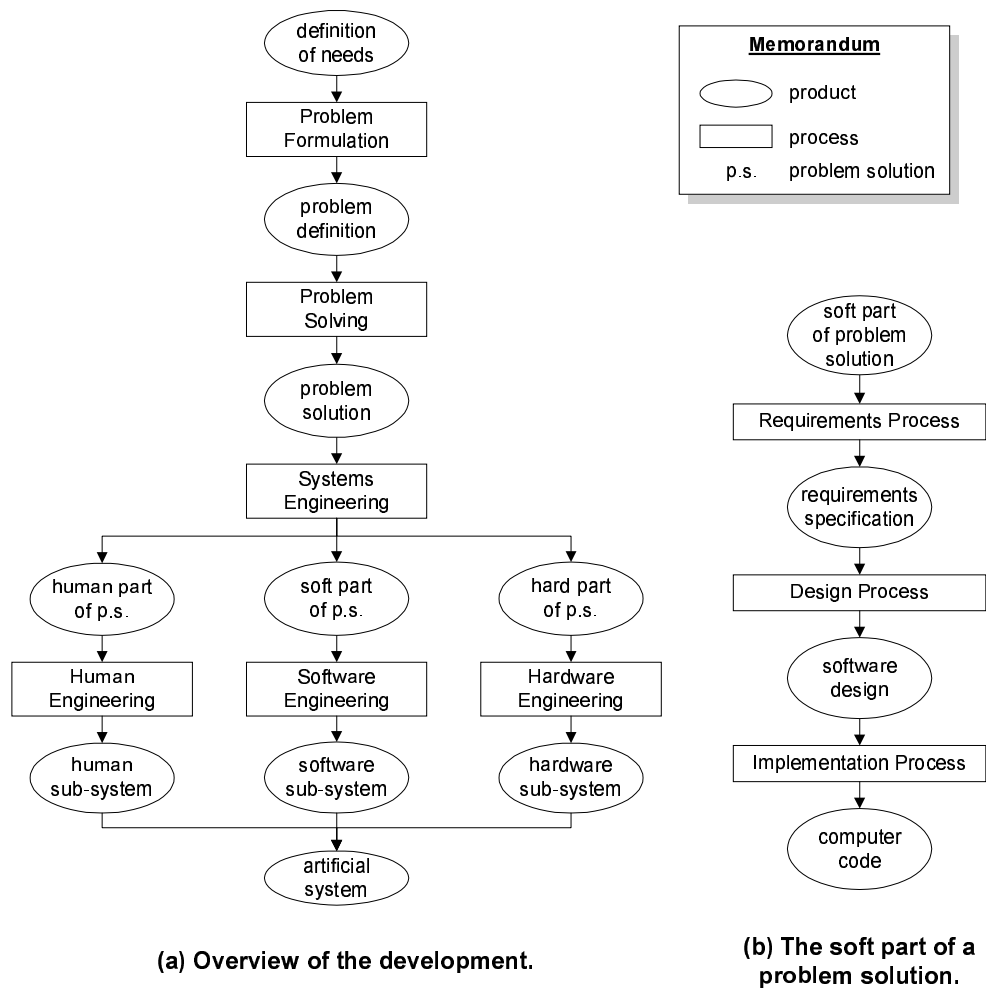


Figure 1: Stages in the development of an artificial system.

An Introduction to Software Engineering

OVERVIEW MAP

BLOCK 1. INTRODUCTORY CONCEPTS

- Unit 1 Information Work Processes and Software Engineering
- Unit 2 Software Process Models
- Unit 3 Software Development Methodologies

BLOCK 2. DEVELOPMENT PROCESSES

- Unit 4 Requirement Process
- Unit 5 Design Process
- Unit 6 Implementation Process

BLOCK 3. MANAGEMENT PROCESSES

- Unit 7 Project Initiation Process
- Unit 8 Project Monitoring and Control Process
- Unit 9 Software Quality Assurance Process

BLOCK 4. INTEGRAL PROCESSES

- Unit 10 Verification and Validation Process
- Unit 11 Software Configuration Management Process

BLOCK 5. TOOLS AND STANDARDS

- Unit 12 Tools
- Unit 13 Standards

BLOCK 6. A CASE STUDY

- Unit 14 A front end of the case study
- Unit 15 A back end of the case study: An implementation in MS ACCESS
- Unit 16 A back end of the case study: An implementation in ANSI C
- Unit 17 A back end of the case study: An implementation in MS Visual C++

Figure 2: Content of the online course notes.