

# SETT-Net: A Network for Software Engineering Training and Teaching

Katerina Zdravkova

*Institute of Informatics, University "Sts Cyril and Methodius" – Skopje, Macedonia*

Klaus Bothe

*Institute of Informatics, Humboldt University – Berlin, Germany*

Zoran Budimac

*Department of Mathematics and Informatics, University of Novi Sad, Serbia and Montenegro*

*E-mails: keti@ii.edu.mk; bothe@informatik.hu-berlin.de; zjb@im.ns.ac.yu*

**Abstract.** *The intention of the SETT-Net is to enable the usage of shared training and teaching resources for a software engineering course in eight schools and in four countries. The network is completely Web-based. It consists of a Centre Point at Humboldt University in Berlin and four National Centres, in Berlin, Novi Sad, Plovdiv and in Skopje. This paper presents the syllabus, the teaching materials of the joint SE course, the network architecture and its contents.*

**Keywords:** Computing Curricula, eLearning

## 1. Introduction

In 2001, the project "Software Engineering Education and Reverse Engineering" supported by DAAD: Deutscher Akademischer Austausch Dienst (German Academic Exchange Service) [7] within the Stability Pact for South Eastern Europe [8] was launched. Initially, it involved four schools in four countries. These schools make the core project groups.

At the moment, ten schools from seven universities collaborate on several tasks. The project has attracted a great interest in other countries from the Balkans, which will attend the next project workshop in Ohrid, at the end of August 2003.

The project addresses many tasks dealing with the software engineering training and teaching. The essential are:

- Introduction of a software engineering course in South Eastern Europe
- Creation of a joint software engineering course,
- Preparation and implementation of the training and teaching materials
- Realization of representative case studies
- Dissemination of the project results, and
- Preparation of the prerequisites for the project maintenance and upgrading

## 1.1. Setting up the scene

The first project workshop was held in Novi Sad in October 2001. In order to prepare a basis for a joint SE course, computing curricula and syllabi, as well as the infrastructure at the nine South Eastern European schools were discussed. Surprisingly, it was found that software engineering was a regular course in two schools, and to some extent, in other two schools. Most computing curricula were outdated and the infrastructure in all Yugoslav schools was poor.

The workshop was a catalyst for immediate changes. As a result, one school has already restructured and two are in the process of restructuring the computing curriculum. The software engineering course has become, or it is intended to become, a basic course at the undergraduate or at the postgraduate studies in the most of the schools involved in the project.

## 2. Joint software engineering course

One of the most important issues of the project is the creation of a joint course on software engineering for the schools involved in the project. The motivation for such a course in several schools was:

- Knowledge dissemination and transfer
- Reduction of expenses (reuse of the slides the assignments, the case studies, ...and others)
- Transfer of credits
- Joint discussion forums (on contents, didactics, literature, assignments, ...and others), and
- Background for joint education and research

Software engineering course has a long tradition at Humboldt University in Berlin. After five years of extensive work, the course is becoming stable. Its syllabus is being developed in accordance to the recommendations of IEEE Computer Society and ACM.

With 60 lecture hours, 30 lecture hours for exercises and 30 lecture hours for assignments (i.e. 60 + 30 +30), it covers the most of the CC2001 core topics, as well as many elective topics. It also gives a good theoretical basis for the future research and development. The course is enriched with many case studies based on real-life problems [1]. They give the opportunity to exploit all software engineering elements, including teamwork and software system management. Finally, it presents many practical aspects of the software engineering itself.

## 2.1. Course syllabus

The joint course is divided into 5 basic parts. Each part consists of several separate topics. The parts within the course are divided and the smaller topics are given in the following list:

- Introduction to software engineering:
  - P1/1: What is software engineering (2)
  - P1/2: Quality criteria for software products (1)
  - P1/3: Software process models (1)
  - P1/4: Basic concepts and software development documents (1)
- Requirements engineering:
  - P2/1: Results of the “analysis and definition” phase (1)
  - P2/2: Cost estimation (2)
  - P2/3: Function-oriented view (1)
  - P2/4: Data-oriented view (1)
  - P2/5: Rule-oriented view (1)
  - P2/6: Structured analysis (2)
  - P2/7: State-oriented view (1)
  - P2/8: Scenario-oriented view (1)
  - P2/9: Object-oriented analysis (6)
  - P2/10: Formal software specification and program verification (3)
- Software Design:
  - P3/1: Overview of design activities (2)
  - P3/2: Structured design (1)
  - P3/3: Object-oriented design (2)
- Implementation and testing
  - P4/1: Implementation (2)
  - P4/2: Systematic testing (6)
  - P4/3: Functional testing (2)
- Advanced problems:
  - P5/1: Software metrics (4)
  - P5/2: Maintenance (2)
  - P5/3: Reverse engineering (4)
  - P5/4: Quality of software development process and its standardization (3)
  - P5/5: Introduction to software ergonomics (4)
  - P5/6: Project management (3)

The number in the brackets represents the number of lecture hours allocated to each of the parts. The identifier in front of the topic indicates the part and the topic within that part.

## 2.2. Comparison of the joint syllabus with the recommended CC2001 syllabus

In December 2001, a Joint Task Force made of members from the IEEE-CS and the ACM developed “curricular guidelines for undergraduate programs in computing“. The final report is recognized as CC2001 [4]. It identifies a body of knowledge appropriate to undergraduate computer science programs, which consists of 132 units. CC2001 also defines “a minimal core consisting of those units for which there is a broad consensus that the corresponding material is essential to anyone obtaining an undergraduate degree in this field. Units that are taught as part of an undergraduate program but which fall outside the core are considered to be elective.”

In these curricula, the software engineering course belongs to the computer science body of knowledge. It engages 31 out of 280 core lectures, i.e. it is one of the most important courses. According to CC2001, its syllabus consists of these groups of lectures:

- SE1: Software design (8)
- SE2: Using APIs (5)
- SE3: Software tools & environments (3)
- SE4: Software processes (2)
- SE5: Software requirements & specifications (4)
- SE6: Software validation (3)
- SE7: Software evolution (3)
- SE8: Software project management (3)

Similarly to the joint SE course, the number in the brackets represents the number of lecture hours allocated to each of the lectures.

The course can be enriched with several elective lectures from the following list:

- SE9: Component-based computing
- SE10: Formal methods
- SE11: Software reliability
- SE12: Specialized systems development

The number of the lecture hours for the elective lectures is not allocated.

Compared with the software engineering course recommended by CC2001, the joint SE course has several additional topics (Table 1.). The joint SE course mainly focuses on the theoretical aspects of the software engineering. Therefore, the core CC2001 topics are studied in more details.

In contrast, no attention is paid to the topic “Using APIs” which should consist of 5 lecture hours according to CC2001. All other missing CC2001 topics, such as the “Specialized systems development” are parts of the elective topics. This means that their exclusion will not significantly affect the quality of the course.

Topic label	Allocated hours	Minimum CC2001	CC2001 unit (topic) label
P1/1	2	0	none
P2/2	2	0	none
P2/9	6	1	SE1/5
P2/10	3	0	none
P4/2	6	3-4	SE3/3, SE6
P5/1	4	1	SE4/3
P5/3	4	1-2	SE7/3, SE7/5
P5/5	4	0	none
none	0	5	SE2
none	0	not allocated	SE12

**Table 1. Major differences of both syllabi**

To conclude, the current course covers at least 85% of the recommended CC2001 syllabus. It has still got some small differences that can be either neglected or modified during the course evolution.

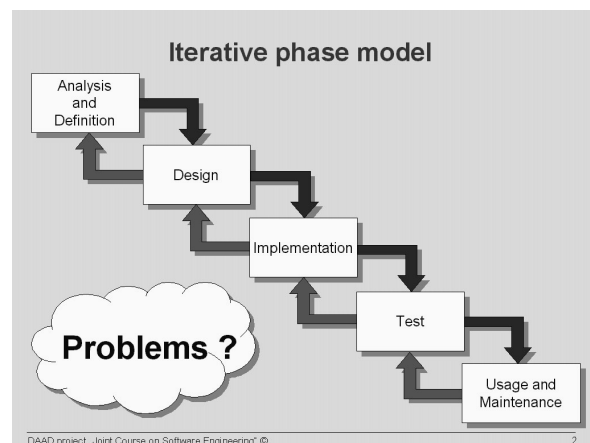
The similarity of the syllabus of the joint SE course with the recommended CC2001 syllabus encourages its further implementation.

### 2.3. Training and teaching materials

The preparation of the basic course started in 1995 in Berlin. The first stage was a continuing accumulation of the training materials, including conventional slides and exercises. The course material was divided into eight sections: overview, contents, exercises, guests, tools, project, literature and examinations.

The core, and at the same time, the most demanding part of the preparation process was the contents section. It consisted of the lecture notes presented in a slide format. From the very beginning, all material, with the exception of the slides was Web-based [5]. After seven years, the slides have become stable, and the first version of Web-based slides was produced. At this moment, there are about 1360 original slides, i.e. in average 45 slides per one session of two lecture hours. They cover all the 26 topics integrated in the course.

The general goal of the slides is to achieve a high standard of the contents, but also to reach an aesthetic presentation of high quality. In order to fulfil the harmony element, the basic layout of the main slide components is consistent throughout the whole course. To reproduce the order of the thoughts during the presentation, the slides are usually animated. This approach accurately reproduces the human lecturing style.



**Figure 1. The slides intend to accurately reproduce the human lecturing style**

There are several special visual effects adopted at this stage of the project. The remarks and the questions appear in clouds (Fig. 1.). The answers to the questions asked during the lecture appear at the same slide, at the next slide, or they are parts of the lecture notes. Some of the questions are left for the discussion during the lecture, while in special cases they are left open and their answer will come up after the discussion. A similar approach is used for the picture explanations.

The slides are intended for use by two kinds of users: the lecturers and the students. Their format is determined by the end user. As far as the students are interested to read and print the slides only, the most appropriate format for them is the portable document format (PDF).

The lecturers are supposed to reproduce the lecture according to the slides with a minimum preparation effort. Therefore, the slides appear as Power Point presentations. Every slide has got short notes with useful instructions for the potential lecturers that will use them.

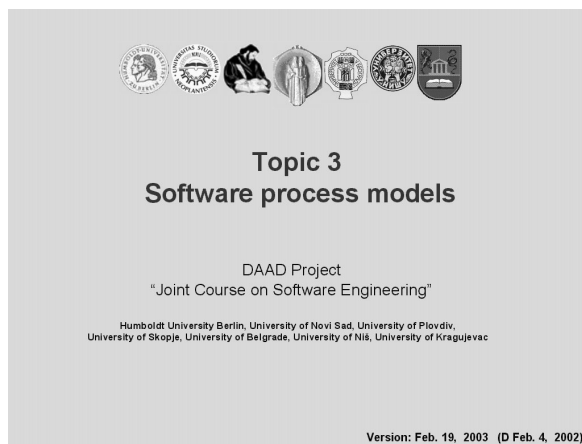
The instructions deal with:

1. The contents of the topic of a particular slide
2. The teaching methodology
3. The useful remarks, and
4. Give the answers to the questions

Whenever a topic has a strong connection to a special book, or the slides are delivered from another author, then this information is presented at the title slide (Fig. 2.). These sources are used with a written copyright permission.

In order to enable producing of a printing version of the slides together with the lecture notes, the recommendations for an optimal use for printing have been developed. Here they are:

1. Export the presentation from Power Point into Word
2. Optimise the columns: the first column should be of the same width like the exported slide
3. Optimise the rows: delete the empty lines
4. Draw horizontal and vertical lines between the slides and the lecture notes



**Figure 2. A typical title slide**

## 2.4. Improvement of the slides

All teaching materials are prepared for the purposes of the software engineering course at Humboldt University, and therefore they are in German. Since many project participants are not familiar with the German language, English was chosen to be an intermediate language.

Translation process of the slides into English has already started and the current stage was reported during the second project workshop held in Plovdiv in September 2002. By that time, 12 out of 26 topics were completely finished. Reproduction of the lecture notes according to these presentations was more than successful. It encouraged keeping to this direction.

In parallel with the translation process, the improvement of the original slides is also in progress. Improvement process passes through three stages: extraction, addition and correction.

It is worth mentioning that during the improvement process, new topics are introduced,

and the slides for them are produced. Such an example is the topic P5/2 that has not existed in the basic course at Humboldt University in Berlin. It is expected that in the near future at least three new topics will be introduced.

Global upgrades concerning the joint SE course are made at Humboldt University. The improvement process is carried out within the core project groups, according to the defined slide management rules (Fig. 3.).

1. For each topic holds: There is only one current topic ppt-slide file which is the one on the SE Education Website at HU.
2. For a modification of a topic ppt-slide file, it must be assured that only one party is allowed to modify this file at one moment.  
To this end, during the slide modification process this topic is marked by "in update by ..."
- In such a case, Uli Sacklowski should be informed - and he will pose this message to the website.
3. ppt-slide files at the project web site can only be exchanged by the administrator which is Uli Sacklowski.
4. Exchange of an old version by a new version should be agreed by the modification authors and by HU
5. Handouts (pdf format) will be delivered by HU

**Figure 3. The rules for the slide management**

At the moment, all the teaching materials are free. However, the access to the slides, to the student assignments, and to the examinations is password protected.

## 3. Software engineering network

In February 2002, a project site was created at Humboldt University in Berlin [6]. It gives an overview of all activities concerning the project and detailed information concerning the software engineering course.

The links related to the project point to the schedule of project development, and to the participants of the project. The first link is the best indicator of the project's progress.

Currently, the project is divided into fourteen tasks. The activities corresponding to the last group of links have not started yet, but they are expected to start soon and be completed before the last project workshop scheduled for August 2003. They will present the allowed differences between the national syllabi, the adjusting criteria for the exams and the exercises, and finally, the joint textbook. These tasks need a more intensive collaboration between the Berlin school and those South-East European schools that have already started, or intend to start the

software engineering course soon. Once the differences between the national syllabi and the criteria for the assignments and examinations are adjusted, the credit transfer will be possible.

At the beginning, the credit transfer between Humboldt University in Berlin and other Southeast European schools is a priority. With time, a mutual transfer of the credits will also be established.

### 3.1. Network architecture

In software engineering education, there are examples of cooperative work across universities [2]. Furthermore, the Web provides presentation and delivery of multimedia contents. As a consequence, virtual classrooms are replacing traditional classrooms every day [3]. They enable on-line learning and bring together many students no matter the place and time.

The creation of a network of several virtual classrooms was a necessity of the project, mainly because of the diversity of the national education systems, as well as the language barrier. Furthermore, the prepared materials are approved and Web transferable.

The Centre Point is the heart of the whole network, particularly because of its leading role in the creation and in the presentation of the joint SE course. As a result, all the core project groups contribute in its creation.

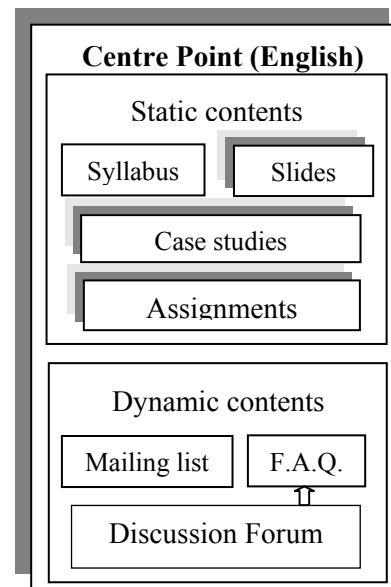
The Centre Point is completely Web based [5]. The process of transforming the Web site into a virtual classroom is in progress. It concerns with two very important aspects:

- The frequency of changes of the contents that determines the virtual classroom dynamism, and
- The level of interactivity of its Web site.

The frequency of changes is a very important issue since it determines the structure of the basic Web site, but also the structure of its potential mirrors. This aspect highly affects the transformation of the network into a virtual classroom network. The method for its effective and correct mapping from central to national level will be selected during the project meeting in Berlin in May 2003.

All the contents at the Centre Point are in fact dynamic. However, the frequency of changes concerning the training and teaching materials is not often, and therefore it can be treated as static (Fig. 4.). The other elements, particularly the discussion forum, the mailing list, and the F.A.Q. are dynamic.

Currently, the interactive element has not been included yet, mainly because of the concentration of the efforts to the creation of the training and teaching materials. In future, the interactivity will be added into the case studies, the student assignments, and particularly into the exams.



**Figure 4. Division of the contents in relation to the frequency of changes**

### 3.2. A National Centre

The Centre Point is connected with the four national sites, called National Centres. At the moment, they are in Berlin, Novi Sad, Plovdiv, and in Skopje.

Basically, the National Centres are mirrors of the Centre Point. The main objectives of its presence are to help relieve the Berlin server of some of the traffic, and the need of a local access to all materials related to the network. Furthermore, it facilitates translation, because the framework remains unchanged.

At local level, each mirror site is coupled by another site, which is an exact translation of the static contents of the mirror in the official language of the country. Translated site can be accessed directly from the National Centre home page. Furthermore, all contents from the English version are connected with the corresponding national version. This makes the National Centres completely dependent on the Centre Point. At the same time, they are independent units within the network.

At the National Centres, independent are the syllabus, the selection of the materials relevant for the course, the assignments, the place, the time, the list of lecturers and students, and some other elements concerning the national course. The selection of the training and teaching materials coming from the Centre Point is supervised, but not restricted by Humboldt University in Berlin. The selection policy has not been established yet, but it is one of the tasks of the next project meeting. The policy will highly depend on the infrastructure in the Southeast European schools involved in the project. The National Centres have a possibility to insert individual links.

Communication between the Centre Point and the National Centres is centralized, except for the individual communication. This means that the global changes concerning the materials are inevitably reflected on the National Centre.

National Centres are intended to be the focal points for software engineering training and teaching in the SEE countries that participate, or will participate in the project. This means that new schools that join the network will become National Points, with the National Centre as a by-pass to the Centre Point.

Currently, the only National Centre with an own network of National Points is the school in Novi Sad. The material related to the joint SE course has been included into the Web pages of the school. Besides the links to the site at Humboldt University, this Web page also contains local student assignments and outlines of some standard software documents.

#### 4. Conclusion

The network for software engineering training and teaching initiated by the project “Software Engineering Education and Reverse Engineering“ is the first international virtual classroom network in South Eastern Europe.

Supported and supervised by Humboldt University in Berlin, the network connects three South Eastern European countries and many schools. Its goal is to define a framework for all potential joint courses within the participating schools. All that courses will be in line with the European standards, thus the project will be a good basic for establishing the European Credit Transfer System in South Eastern Europe.

Besides Humboldt University, as a part of the software design course, in October 2002, the University of Novi Sad started using selected

lectures and slides based on our joint project. This experience is the first verification of the concept of the network for the software engineering course. The course was also offered to the industry in order to gain more feedback. Some topics gained greater interest in industry than they do among the students, e.g. software process models and project management, which can give us a clearer picture about the main problems of the software development practice in South Eastern Europe. This important feedback will be used in the further development of the course.

#### 5. Acknowledgements

We acknowledge the support of DAAD under the auspices of the Stability Pact for South Eastern Europe through which the work reported here is being funded.

#### 6. References

- [1] Bothe K. Reverse Engineering: the Challenge of Large-Scale Real-World Educational Projects. In Proceedings of CSEET 2001, Charlotte, NC, February 2001.
- [2] Brereton O P, Lees S, Bedson R. Student Collaboration across Universities: A Case Study in Software Engineering. In Proceedings of the 13th Conference on Software Engineering Education & Training, Austin, 2000.
- [3] IIL, International Institute for Learning, Inc: Microsoft Project Virtual Classroom eLearning Courses, © 2000-02  
[http://www.iil.com/microsoft\\_project\\_training/microsoft\\_project\\_virtual\\_classroom\\_courses.asp](http://www.iil.com/microsoft_project_training/microsoft_project_virtual_classroom_courses.asp)
- [4] Engel G, Roberts E, editors. The Joint Task Force on Computing Curricula, Computing Curricula 2001, Computer Science, IEEE Computer Society & Association for Computing Machinery, December 2001.
- [5] Bothe K, Sacklowski U, Schuetzler K., Software Engineering Education, 2002  
[http://www.informatik.hu-berlin.de/swt/lehre/SE\\_02/index.html](http://www.informatik.hu-berlin.de/swt/lehre/SE_02/index.html) [02/13/2002]
- [6] Bothe K, Budimac Z. Project Web site  
<http://www.informatik.hu-berlin.de/swt/intkop/se/index.htm> [02/06/2002]
- [7] DAAD: Deutscher Akademischer Austausch Dienst <http://www.daad.org>
- [8] Stability Pact for South Eastern Europe  
<http://www.stabilitypact.org>