Maintaining quality of software engineering education by a shared repository of course materials in a multilateral setting

Conference Paper · April 2014

DOI: 10.1109/EDUCON.2014.6826145

Klaus Bothe
Humboldt-Universität zu Berlin

Zoran Putnik
University of Novi Sad

Mirjana Ivanovic
University of Novi Sad

Stanimir Stoyanov
Plovdiv University "Paisii Hilendarski"

Maintaining Quality of Software Engineering Education by a Shared Repository of Course Materials in a Multilateral Setting

Klaus Bothe
Institute of Computer Science
Humboldt University Berlin
Berlin, Germany
bothe@informatik.hu-berlin.de

Ioan Jurca
Institute of Informatics
Technical University Timisoara
Timisoara, Romania
ionel@cs.utt.ro

Zoran Budimac, Zoran Putnik, Mirjana Ivanovic
Institute of Mathematics and Informatics
University of Novi Sad
Novi Sad, Serbia
{zjb,putnik,mira}@dmi.uns.ac.rs

Novica Nocovic
Dept. of Computer Science
University of Sarajevo,
Sarajevo, Bosnia and Herzegovina
nnosovic@etf.unsa.ba

Stanimir Stoyanov, Asya Stoyanova-Doyceva
Institute of Informatics
University “Paisii Hilendarski”
Plovdiv, Bulgaria
{S.Stojanov, a.stojanova}@isy-dc.com

Damir Kalpic
Faculty of Electrical Engineering and Computing
University of Zagreb,
Zagreb, Croatia,
damir.kalpic@fer.hr

Katerina Zdravkova, Boro Jakimovski
Institute of Informatics
University "Sts. Cyril and Methodius"
Skopje, FYR Macedonia
{keti,boro}@ii.edu.mk

Betim Cico
Faculty of Electrical Engineering
Polytechnic University Tirana, ,
Tirana, Albania,
bcico@abcom-al.com

Abstract— **Developing teaching materials is a time-consuming and expensive activity. Thus, over the years, several consortia have created joint materials to benefit from them. It is expected that shared teaching materials are a means to save effort in its development, to transfer methodological and technical knowledge between different university staff, and to exchange experience in practical application. However, does it really pay off considering the diversity of different educational environments and the difficulties of using externally produced materials, rather than dedicated individual ones? This paper reports on the experience gained in a multi-country project. Both success factors and problems are outlined. It turned out that sharing an educational repository by several partners supports maintaining quality of teaching materials by joint efforts and distributed contributions.**

Keywords— Software engineering; shared teaching materials; multilateral co-operation

I. INTRODUCTION

Creating, collecting and sharing software engineering course materials was the goal of the multi-lateral SWENET project, consisting of six American universities and supported by an NSF Grand [10]. The website was open to the public such that other interested sites might also have profited from it. Another project, the so-called MuSofT project [7], with participants from eight German universities had similar intentions. Another bigger consortium is the International Software Engineering University Consortium, ISEUC, covering dozens of partners from Australia, Canada, U.K. and the US [9]. Besides these projects specialized in software engineering, more general projects can be found in the literature in which shared teaching materials have been created covering a lot of areas, e.g. Ariadne [1] and Merlot [8].

As an excellent summary of the basic motivation of all that work, in [10] is found: “As software engineering educators, we are all too aware of how difficult it is to start from scratch. In particular, the lack of examples, teaching materials, and exercises can significantly impede the inclusion of software engineering concepts into undergraduate courses”.

All of these projects above are based on the reasonable paradigm of developing learning units of varying types to suit the individual needs of different lecturers. These learning units are expected to be combined in order to be used in different courses.

978-1-4799-3190-3/14/$31.00 ©2014 IEEE 3-5 April 2014, Military Museum and Cultural Center, Harbiye, Istanbul, Turkey
2014 IEEE Global Engineering Education Conference (EDUCON)

Since 2002, teachers from ten universities in eight countries - Germany, Serbia, Bulgaria, FYR Macedonia, Bosnia & Herzegovina, Croatia, Romania and Albania – have developed teaching materials in the field of software engineering within the JCSE (Joint Course on Software Engineering) project. Our purpose was to jointly develop, discuss, share and reuse these materials to save capacity and to transfer technical and methodological knowledge. The consortium covers groups experienced in software engineering, as well as groups without any previous experience in the field. It was decided that English would be the common language for the development of course materials.

As opposed to the approach in the projects mentioned above [7, 9, 10], our philosophy was to provide a complete course in software engineering consisting of a sequence of dependable modules, rather than a set of independent modules that have to be combined into a course by lecturers. This form is easier to reuse, in particular for lecturers not so familiar with the field. In addition, this dependence allows for more coherent materials.

There were fundamental questions connected with this project.

- Is it profitable to reuse existing teaching materials instead of producing them individually from scratch?
- Is it really possible for a staff, without deeper experience in the field, to establish a new course only on the basis of materials (mainly) developed by other parties?
- Do existing shared teaching materials allow for enough flexibility to satisfy different educational environments?
- Does sharing an educational repository by several partners support maintaining quality of teaching materials by joint efforts or will it lead to different problems instead?

This paper focuses on the answers to these questions by evaluating the experience gained in the development and (re-)use of the JCSE teaching materials at ten universities. In particular, we investigate points which influence maintaining the quality of teaching materials within this particular multilateral setting.

The rest of the paper is organized as follows: in the next section, an overview of the project is given. In section 3, the different deliveries of the course at ten universities are described. Section 4 focuses on aspects which were found important for the success of the project, and in section 5, the concentration is on critical points which had to be coped with. Section 6 provides the summary.

II. PROJECT OVERVIEW: BACKGROUND AND COURSE MATERIALS

On the basis of a project supported by DAAD (German Academic Exchange Service), in 2002, a group of five universities (Humboldt University Berlin, Novi Sad, Plovdiv, Skopje and Belgrade) started the development of shared teaching materials for software engineering compliant to general recommendations [6]. In 2003, the first stable version was completed. During the next few years, there was a continuous evolution of the materials through improvements, extensions and new topics. From 2004 to 2007, new groups joined the project (Universities of Tirana, Timisoara, Zagreb, Sarajevo, Podgorica, Kragujevac, Rijeka and the Polytechnic University Tirana). All of them contributed to the project by diverse activities: producing materials, discussing materials, writing review reports, applying slides in lectures and writing usage reports, translating materials to national languages, and others [3].

Figure 1 provides an overview of our project website covering teaching materials for the shared course, as well as organizational issues (cf. JCSE website [12]).

Based on the syllabus, the lecture slides constitute the core of the teaching materials by providing the basic technical knowledge. This knowledge is illustrated by case studies: software projects with documents covering all development phases (requirements specification, software architecture, code, test cases, cost estimation, etc.). Lecture notes provide additional information for lecturers, i.e. technical information in addition to the slide content and hints on the methodology. Assignments are closely connected with the content delivered in lectures and refer to software tools as well as case studies.

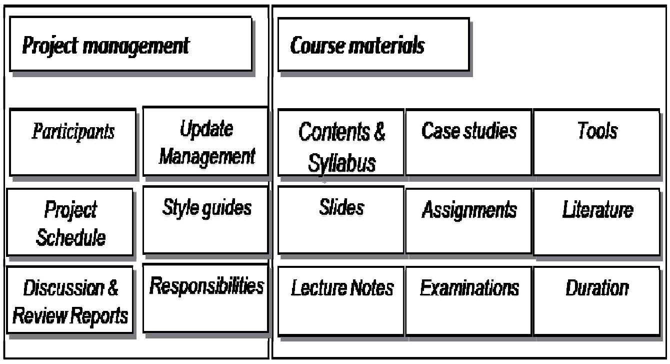

Figure 1. Overview of JCSE website: course materials and project management

As a main point, the structure of the JCSE website reflects the general success factor of projects like that one: We have to consider project management activities as well as the course material structure as two separate topics to get a high-quality product in that multilateral environment. On one side, course materials have to be much more than only lecture slides. Thus, for example, even examination questions are useful to be reused. On the other side, the whole development and maintenance process has to be planned by a dedicated project management. Questions like the following ones have to be answered: Who is responsible for which activities? How to deal with modifications of existing materials? How to organize the exchange of experience? In other words, the whole course material project has to be managed like an ordinary software project.

III. Delivery of the Course at Different Sites

Already the fact that there were a lot of real-life applications of the course at different sites has been an important point of the project. In that way, there was a continuous evolution of the materials: by a development of the materials or by exchange of experience (e.g. in a discussion forum or by filling in review reports).

These joint efforts were rather meaningful for an economical maintenance of the quality of teaching materials.

The course has been presented at ten universities in eight countries based on our JCSE teaching materials (Figure 2). It has been presented as a whole, in parts, and in variants. At some universities, the materials have even been applied in different internal variants, e.g. for different audiences or curricula. It turned out that there is a wide range of usage and enough flexibility to satisfy different educational environments.

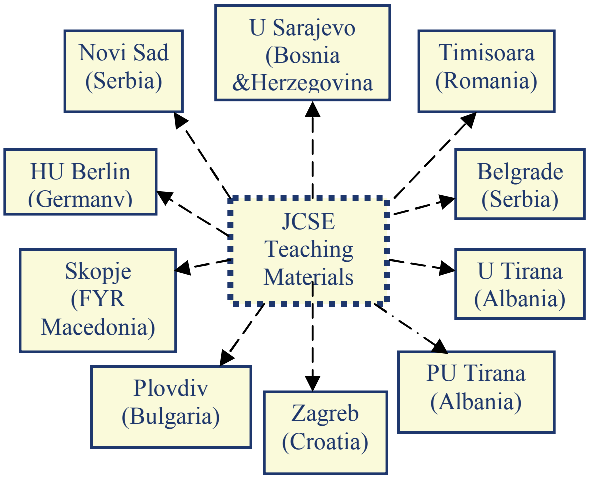

Figure 2. Universities using JCSE teaching materials.

Finally, Software engineering as an educational discipline was introduced at several universities only after our teaching materials were produced: Novi Sad, Skopje, Plovdiv, Tirana, Belgrade and Kragujevac.

The following is a brief summary of the delivery of courses based on the shared materials.

Humboldt University Berlin: an early variant of the course was introduced in 1996. Since 2002, the developed materials have been used for about 80 students per year. The complete materials are presented, i.e. all topics, assignments, tools and case studies. Exams are oral ones.

University of Novi Sad: this university was the first one of the South Eastern European partners that has applied the materials since 2002. From 2002 to 2004, the course was offered to a special audience (e.g. from industry for vocational training). Since 2004, software engineering has been introduced as a compulsory course in Bachelor studies [4]. Slides are used in English, and the oral delivery is given in Serbian. Most of the topics and assignments are presented. Only two of the five tools are used. Exams are in a written form using our project material.

University 'Paisii Hilendarski' Plovdiv: this group has presented all the course topics since 2003, basically in Bachelor studies and as induction modules for Master studies. About 100 students take part each year. Plovdiv decided to translate the English slides into Bulgarian. The assignments are not taken from the project. Exams are in a written form using their own (private) questions. The group derived a Bulgarian textbook from the JCSE materials.

University 'Sts. Cyril and Methodius' Skopje: our English materials were selectively used in two existing general CS courses between 2004 and 2006. In 2007, there was the first official software engineering course where the JCSE materials were used selectively (only 15 out of 25 topics, and 3 of 8 assignments). During the next years, the content has been extended to 21 topics and 6 assignments, and a new case study of their own was introduced. This new case study is being offered to the other partners for reuse.

University of Belgrade: at this faculty, software engineering was introduced in 2005. 12 topics were taken from the JCSE material and combined with their own (local) materials. Slides are used in Serbian translation. About 130 students take part every year.

University of Sarajevo: this university applied the course materials only since 2011. However, there is a unique interesting delivery by a combination of JCSE materials and topics taken from the book written by Sommerville [11].

Polytechnic University Tirana: the delivery of the course at this university is unique in our project. The JCSE was delivered as an intensive course over six days by guest lecturers from Berlin and Novi Sad. Both the slides and presentation were in English. Afterwards, in the weeks after the lectures, four assignments were given, solved and submitted via email to the lecturers. Even the exams were assessed in a distance education mode: written in Tirana, and corrected in Berlin/Novi Sad. In this way, it was the first time that software engineering has been taught at a university in Albania. The goal is that the local staff takes over the course step by step. To that end, two assistants from Tirana presented selected topics at the deliveries since 2010.

University Tirana: for a special curriculum ("Informatics in Economics"), 12 selected topics have been presented by local teachers without any external support, only by using the materials from the project website.

University Zagreb, Technical University Timisoara: only some of the selected JCSE topics have been presented as part of other established CS courses.

The original JCSE materials have been applied in nearly the original produced form, as well as in several variants. Summing up, the diversity in the delivery of the course covers the following main points:

- extent of presented topics
- language of teaching materials

- differences in the audience
- extent of using tools
- assignments used
- examination types (oral, written)
- familiarity of the staff with the subject of software engineering

Other organizational issues of diversity are: team work in assignments or not; additional case studies in lectures and/or assignments; mode of final grading: might be a combination of final examination, course work (assignments), and additional tests; own topics added: e.g. RUP (Belgrade), extreme programming (Novi Sad), i.e. the course is open to new subjects; publishing slides for students before/after the lectures; delivery of handouts before the lectures (most imported slides with detailed information have to be available in the lecture); value of the course (how many credits).

In summary, it is just this diversity that contributed to efficient development of the course materials covering ideas of rather different sources.

IV. Success factors

Already Joint development of shared materials was challenging and had to be supported by dedicated measures. Several important aspects concern project management:

- **Slide style guides:** these were necessary to assure a unified appearance of the slides throughout all course topics. For example, the master slide had to be defined, the form of headlines, the kind of colors, etc.
- **Version control:** since all participants were requested to contribute to the materials, a restrictive update procedure had to be defined. A very simple one was adopted: at any particular time, there was only one current slide file for each topic which was on the project server. Only one party was allowed to modify a topic at any one time. The website administrator was the only one who could physically exchange the current version for a newer one.
- **Roles:** in addition to the website administrator, other roles with dedicated rights were introduced: project manager, developer, user and modifier.
- **Review reports:** as special kinds of documents, course materials could be assessed by the well-defined review technique. This process was supported by a review report form which was the basis of the discussion forum.
- **Personal contacts:** workshops, guest visits, and mutual help were permanent organizational and personal issues in order to stay in contact and to support each other.

All in all, these measures were not too sophisticated. However, they were sufficient to guide a distributed development process. Details of the development process can be found in [3].

Reuse of shared materials was non-trivial and had to be supported by dedicated measures connected with the philosophy of the course:

Complete course: in our compilation of teaching materials, there are fixed dependencies in the sequence of topics, as well as strong connections between the topics and other parts (case studies, assignments and tools). This form is easier to reuse, in particular for lecturers not so familiar with the field: they do not have to combine single learning units by themselves.

Short versions: there is the freedom to leave out parts (topics, assignments, tools or case studies). Two default recommendations were provided for a long variant, as well as a short variant of the syllabus. Moreover, as is pointed out in sections 3 and 4, there is still much more flexibility for SE teachers. It was necessary to document dependencies between different topics (which one is a prerequisite of the actual one) to be able to cancel particular topics and select only special ones for presentation.

Pool of adaptable assignments: assignments deeply connected with the lectures should be available. Moreover, since solutions to the assignments are often not unique in software engineering, exchange of ideas connected with this issue had been organized in our project.

Lecture notes: not all background information is available on the lecture slides, in particular, concerning additional technical knowledge and the methodology of how to apply the slides. Lecture notes were thus introduced as part of the slide files, i.e. as part of power point notes.

Case studies throughout the whole course: in SE education, case studies belong to the heart of the course. Case studies have to illustrate the delivered knowledge and should be close to real life. In our case, two software projects were selected to be used as case studies throughout the lectures and in assignments.

Tools: modern software development without tools is impossible. For that reason, tools should be included in lectures as well as in assignments. Main fields of typical tools should be covered. The following fields are supported by tools in our course: OOA (UML), formal specifications, metrics, testing and version control.

Examination questions: a common pool of examination questions, together with solutions, should be offered to the staff to give them an idea of how to test the students.

Teachers mobility – guest lecturing: Due to different capacities at different local universities with educators with different levels of expertise in the field it turned out to be helpful to include teachers mobility as a strategic aspect.

V. Challenges

Several inherent challenges, which could not be foreseen in the beginning of the project, turned out to be crucial. The following decisive ones should be mentioned:

Language: although English is generally quite well understood, since it is the technical language in engineering, the use of English in everyday lecturing is challenging and demanding in countries which are non-native English speaking.

Localization: according to the level of knowledge of the English language, several groups decided to translate the materials into their national languages. The use of a national language allows for more subtle teaching and discussion with students. A tool was implemented and used in our project to support the process of technical translation [2]. By means of that tool, the translation process was supported by an internal technical multilingual dictionary which guarantied the technical correctness of the translation. This tool, however, is not in everyday usage. Its benefit is the standardization of translating technical terms; however, translating by hand is simply faster.

Maintainability: materials which exist in the original English variant, as well as in several national languages, complicate maintenance and version control dramatically. Thus, when there was translation to the national language, the development life at that site very often stopped and did not include further improvements of the English baseline materials.

Local staff: additional efforts are necessary to also allow lecturers with less background in the field to apply teaching materials properly and competently: lecture notes with supplementary information on the subject and on teaching tips to the slides was attached – information which is normally in the mind of the experienced lecturer. In this way, there are several examples where a new course on software engineering was introduced with less experience in the field, only by using the JCSE materials [13].

Tools: the reduced usage of tools seems to be a serious drawback, since modern software development without tool-support cannot be successful. There are two main reasons for not using tools so intensively at partner universities: it takes staff capacity (time) to become familiar with the tool, and tools might not be free of charge even for universities.

Assignments: the most important reason to drop a suggested assignment was the non-availability of tools, e.g. testing tools. To cope with assignments in software engineering and, in particular, with their solutions, requires a lot of experienced staff, for example, to assess a textual review report of a given requirements specification is not an easy task for the novice educator.

VI. Conclusions

Several The primary purpose of the project was to disseminate experience and to reduce effort in creating a new course on software engineering. To that end, shared teaching materials have been developed covering all parts (lectures, assignments, case studies, examinations and tools). The materials have been maintained and have evolved over the years.

At the beginning of the project, there was agreement on the philosophy to establish a "complete course". This is different from other projects such as Swenet [9] and Musoft [7], in which a set of (rather) independent modules were provided. In our case, this complete course made it manageable for teachers who were new to that field to take the materials as they were and deliver it (Skopje, Plovdiv and Tirana).

This paper focused on the experience in using these materials at ten universities at eight different countries. The materials developed made it possible at several sites to introduce software engineering as a new teaching discipline. The course has been delivered as it is, i.e. as a complete package of topics with associated materials (slides, assignments, case studies, etc.), as well as in diverse variations. The pool of material allows for enough flexibility in that process. Thus, the course exists in several variants: different in the extent of presented topics, in assignments, used tools, examinations, experience of teachers in that field, kinds of delivery (local teachers / guest lecturers in international courses), basic curriculum, and others.

It turns out that sharing of teaching materials pays off: it saves capacity for the production of teaching materials and it allows for the transfer of technical and educational knowledge.

Thus, the project was a contribution to maintaining quality of teaching materials by joint efforts and joint contributions at rather different local universities across different countries.

References

[1] Ariadne Project, http://www.ariadne-eu.org

[2] Bothe, K. and Joachim, Interactive Tool-based Production of Multilingual Teaching and Learning Materials. Proceedings of the 5th IEEE International Conference on Advanced Learning Techniques, ICALT (Kaohsiung, Taiwan, 2005).

[3] Bothe, K., Schützler, K., Budimac, Z. and Zdravkova, K. 2005. Collaborative Development of a Joint Web-Based Software Engineering Course across Countries. Proceedings of the 35th ASEE/IEEE Frontiers in Education Conference (Indianapolis, IN, October 19 – 22, 2005).

[4] Budimac, Z. and Putnik, Z. 2006. A short report on yet another run of the JCSE in Novi Sad, 6th Workshop "Software Engineering Education and Reverse Engineering" (Ravda, Bulgaria, September 18 - 23, 2006).

[5] Budimac, Z., Putnik, Z., Ivanovic, M., Bothe, K. and Schützler, K. 2008. Conducting a Joint Course on Software Engineering Based on Teamwork of Students. Informatics in Education, vol. 7, No. 1, 2008.

[6] CC2001, "Computing Curricula 2001", ACM and the Computer Society of the IEEE, http://www.acm.org

[7] Doberkat, E.-E., Kopka, C. and Engels, G. 2004. MuSofT – Multimedia in der Softwaretechnik. Softwaretechnik-Trends, Volume 24, Issue 1, 2004

[8] Merlot project, http://www.merlot.org

[9] Modesitt, K. 2002. "International Software Engineering University Consortium (ISEUC), A Glimpse into the Future of University and Industry Collaboration", 15th CSEET (Covington, Kentucky, 2002).

[10] Hilburn, T., Hislop, G., Lutz, M., Mengel, S. and Sebern, M., Software Engineering Course Materials Workshop, 16th CSEET (Madrid, 2003).

[11] Sommerville, I.: Software Engineering, 9th edition, 2011.

[12] JCSE course website, http://www2.informatik.hu-berlin.de/swt/intkoop/jcse/

[13] Zdravkova, K. Experience with the delivery of JCSE in Skopje in 2007/08, 8th Workshop "Software Engineering Education and Reverse Engineering" (Durres, Albania, September 8 - 13, 2008).