Software Engineering Education Symposium SEES'98
Poznań, Poland, 18-20 November, 1998

THE CONTEXT AND THE CONTENTS OF SOFTWARE ENGINEERING EDUCATION IN SLOVAKIA

Pavol NÁVRAT and Mária BIELIKOVÁ*

Abstract. One of the key actors in implementing the technological innovations centered around the progress in information technology is a software engineer who analyses, designs, implements and maintains various kinds of information systems of businessses, organizations, etc. There are, among other things, at least two important aspects of the software engineering related education to be taken into account: the context, both the current one and the historical background of the developments, and the contents of the education. We present contents of our curricula and explain how it responds to the particular context. Experience of developing Software Engineering education within integrated Informatics curricula designed as a blend to cover the fields of Software Engineering, Computer Engineering and Telecommunications Engineering is also reported.

Keywords. software engineering education, context of education, content of education, integrated Informatics curriculum.

1 Introduction

There is a big – and still growing – demand of not only skilled, but also highly educated software engineers in all branches of industry, public service, research and development etc. The statement is probably true in general, and it holds in particular also for a typical Central European country like Slovakia which has been going since 1989 through a period of social and economic transition without a precedent in its history.

It is becoming clear that there are, among other things, at least two important aspects of the software engineering related education to be taken into account. One of the aspects is the context, both the current one and the historical background of the developments in this region during the last fifty years. It is rather different from what has been the development in the rest of Europe or in North America, to name just two regions with the greatest relevance for the region of Central Europe.

The other aspect is the contents of the education. Roughly speaking, it has been similar to a comparable education offered elsewhere. Even more, in some sense it could not have been any radically different. Despite that, naturally also here some differences can be observed. We

* Slovak University of Technology, Department of Computer Science and Engineering, Ilkovičova 3, 812 19 Bratislava, Slovakia, {navrat,bielik}@elf.stuba.sk, www.elf.stuba.sk/~{navrat, bielik}

shall present contents of our curricula and explain how it responds to the particular context. In conclusions, we shall indicate some possibilities of further development of our curricula.

2 The Context

Central and Eastern Europe shared for decades a radically different context characterized by the lack of freedom, strive for collectivism, and an overall responsibility of the state. Although no single unified system of higher education existed, the Soviet influence was dominant also in this area so their model was reflected in the particular national systems of higher education. We do not possess enough data nor know about any comprehensive study to be able to speak about the situation in the region in general. On the other hand, the situation in Slovakia was very similar to other countries in the region, for the reasons indicated above, so its context can be considered representative in some sense.

In Slovakia, there were no credits used and there were only two levels of higher education. The first one was master's level in humanities, social, natural, etc. sciences, or equivalently engineer's (Ing.) level in engineering, technology, economics, transport, agriculture etc. It took typically five to six years to complete and included elaborating a thesis. The second level was called somewhat obscurely and misleadingly candidatus scientiarum (CSc.) as it took at least further three years of study, included elaborating a dissertation and was thus comparable to the PhD. level known elsewhere.

A major change to higher education in Slovakia succeeded in 1989. It is perhaps worth mentioning that one of the two most important demands of the democratic movement – besides abandoning the "leading role" of the communist party – was an immediate halt of the Marxist indoctrination in the education, both of which had been enforced by the corresponding clauses of the constitution. Success in these two demands, i.e. deleting the clauses from the constitution, marked the actual moment of collapse of the regime. The change allowed to pass a new higher education law in 1990. The new law has established an entirely new and different context for the higher education. The law guarantees academic liberties and radically widens the extent of academic autonomy. The law introduces three levels of higher education; however, it leaves their introduction, and indeed their definition largely up to the decision of a particular university.

Practically, problems arisen with the undergraduate (bachelor's) level which has not been backed by any widely shared interpretation nor experience nor subsequent legislation that would define the status of the graduates. As a consequence, some universities decided to maintain the "old" master's courses but to introduce new bachelor's courses running in *parallel*. The idea was to make them largely independent and not transferable with the master's courses intended as technical, non-scientific education preparing for direct entry to industry positions. There seems to be the problem of interpretation still present. The institution that was offering such courses was an university, or an university of technology, and not a *college*, or a *polytechnic* or a *Fachhochschule*. The students were expecting to receive a complete higher education from it and were not prepared to exchange one or two years "saved" for some new title that is largely unknown and lacks any tradition.

Other universities, on the other hand, seized the opportunity offered by the new law and started to offer bachelor's courses that were in *series* with the subsequent master's and PhD. courses. This involves a far reaching conceptual change also in the master's courses. They do not cover the whole interval of the higher education (apart from the PhD. studies) anymore, but contract to the second level only. Implementation of this model has opened, for the first time in decades, door for students to design their education with much more freedom because of the inherent flexibility of the scheme. They start realizing that they can receive two degrees for example in two different fields, but within a period that would before suffice for just one, even if the higher one of them. Or, to put another way, before this scheme has been adopted, it took twice as long time to acquire two specializations. However, the process of shifting the attitudes of students appears to be a rather slow one, because of strong influence of the tradition.

Other major benefit of the increased granularity of the higher education scheme is the option to finish the studies after no more than four years (with the bachelor's degree). At the moment, the problem with this option is once again the lack of tradition; there is no experience with such graduates, and the public will need some time to find a justified and adequate interpretation. With the first bachelors graduated last year, it will soon be possible to start gathering experience.

One comment is pending on further developments of the liberal higher education law. Obviously, the accumulated experience was evaluated and resulted in identifying the need of certain amendments. Alas, the amendment to the law that was passed in 1996 results, besides correcting some inaccuracies of the original law and improving some solutions, in actually reducing the extent of academic self-rule by shifting some powers back to the government.

From the above outline of the context of higher education, there are some conclusions to be drawn. The development in Central and Eastern Europe has been un-natural, forced by the communist rule and forcing severe compromises in form and contents. The compromises in engineering, natural and a few other sciences were among all the fields the least severe and were largely reduced to three or four social and political sciences subjects that were turned into Marxist propaganda. The rest of the curricula i.e., the absolute majority was largely not contaminated. Here, the biggest problem was the insufficient hardware and software equipment of the laboratories, caused by the export restrictions of the western countries and the technological backwardness of the eastern countries.

3 The contents

The contents of the higher education related to software engineering has been developing basically along similar lines and towards similar resulting curricula. Obviously, there are differences caused by different needs of the respective industries, by different levels of development of the technologies, or by different amounts of support for science and education in the respective countries, which in turn may have caused varying relative time delays in adopting the fundamental new scientific results or methodologies into the curricula.

Besides that, the context as outlined above has been influencing the development of curricula. In Slovakia, there has been a rather sharp distinction between "classical" universities and universities of technology. The former traditionally never offer an engineering degree, while the latter never offer other but such degrees. Therefore, the contents of the related curricula have been at the former type of universities very close to pure computer science, and at the latter type a broader computer science and engineering combination.

Traditionally at Slovak University of Technology, the Informatics curricula have been designed as a blend to cover the broad field of Computer Engineering, Computer Science and Telecommunications Engineering. A relatively major revision in 1989 involved a formal integration of Computer Science (reshaped and renamed to Software Engineering) and Computer Engineering tracks with Telecommunications Engineering, but was at the same time accompanied by increasing the curricula's internal flexibility and by implementing the concept of elective subjects and blocks in a far greater scale than before [10].

As mentioned above, the Informatics course's revision of 1989 also reflected the shifting focus of interest of the industry by changing the name of the Computer Science track to Software Engineering. Of course, changing a name does not make any difference in the content, but the measure was accompanied by setting a goal of gradually introducing more subjects supporting this shift. Developing and introducing software engineering related subjects requires faculty with sufficient expertise and experience. To achieve this, we followed several paths. First, we re-oriented our development activities for local industry to become more involved in typical software engineering projects. Presently, several members of the faculty regularly work on development projects for industry [1]. Second, we try to hire experienced professionals from industry as external lecturers, thesis supervisors etc. whenever possible. Last but not least, we made every possible use of European Union projects (under the TEMPUS programme aiming to foster European transfer of expertise in the educational area) that were focused on Informatics [9, 8] to acquire experience from our partners who were already teaching software engineering related subjects, most notably from the Sheffield University [5].

The last major revision of our course took place in 1993. It was connected to the University's decision to make use of the possibility of introducing the bachelor's level of education. Introducing the bachelor's level into a (then) single Informatics course opened several questions that were concerned also with the content of the curricula, thus going beyond the simple acceptance of the new situation of having two Informatics courses. What appeared initially to many to be merely a formal split into bachelor's and master's levels, induced on second thought serious questions.

It quickly turned out that the splitting revision must first identify the different goals and objectives of the respective courses and consequently to design the curricula as careful transformations of the original one. We shall describe our efforts towards this aim, the present shape of the courses and our experience gained so far.

4 Software Engineering Education within Integrated Informatics Curricula

One of the crucial questions concerns the fundamental skills and knowledge that characterize the professional software engineer [12]. The importance of it is best documented by the establishing of a joint ACM/IEEE Task Force to identify the body of knowledge of the software engineering profession. Implicitly, the assumption behind the effort is that a complete and comprehensive answer to the question is not known to date. Ideally, however, the body of knowledge should be identified before a curriculum can be designed that aims to transfer the knowledge to the learner. Luckily, there are other sources that describe the body of knowledge – here belong obviously monographs, but also e.g., the Ford report [4] which

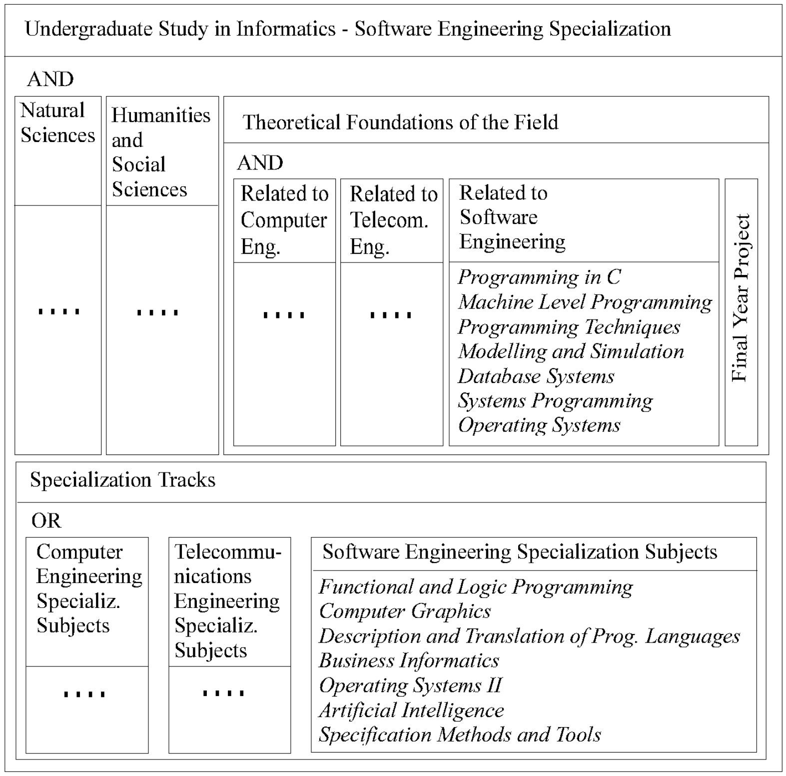

Fig. 1. Undergraduate study in Informatics.

provides a detailed mapping onto a technical content of an undergraduate curriculum. His other report [3] deals extensively with the postgraduate education.

So the situation in knowing what to teach in Software Engineering is not so entirely pessimistic. However, the problem is of a more fundamental nature, viz. the field itself is not mature enough to have its own theory, scientific principles and engineering practices formulated and standardized the way as the other, "older" engineering disciplines have. Some would prefer to strengthen the undergraduate Computer Science curriculum instead of splitting and creating the undergraduate Software Engineering course [15]. Some even deny Software Engineering is a serious discipline [2]. Some envisage software as an engineering discipline [13, 7]. We are strongly committed to developing the discipline of Software Engineering, not least because it quite naturally fits the challenges of a university designated to technology.

Nevertheless, elements from the computing curricula [14] are strongly present in our undergraduate Informatics curriculum. This builds a strong scientific and engineering foundation, but leaves relatively lesser room for subjects specific for software engineering in its "narrow" sense (despite the fact that it typically takes four years to complete).

Figures 1 and 2 give an overview of the undergraduate and postgraduate curricula of Informatics. Software Engineering Specialization Tracks are outlined in more detail.

We offer at least a few subjects (such as Principles of Software Engineering) that provide a detailed introduction to software life-cycles, specification techniques, analysis and design methods, implementation approaches including verification and validation, project

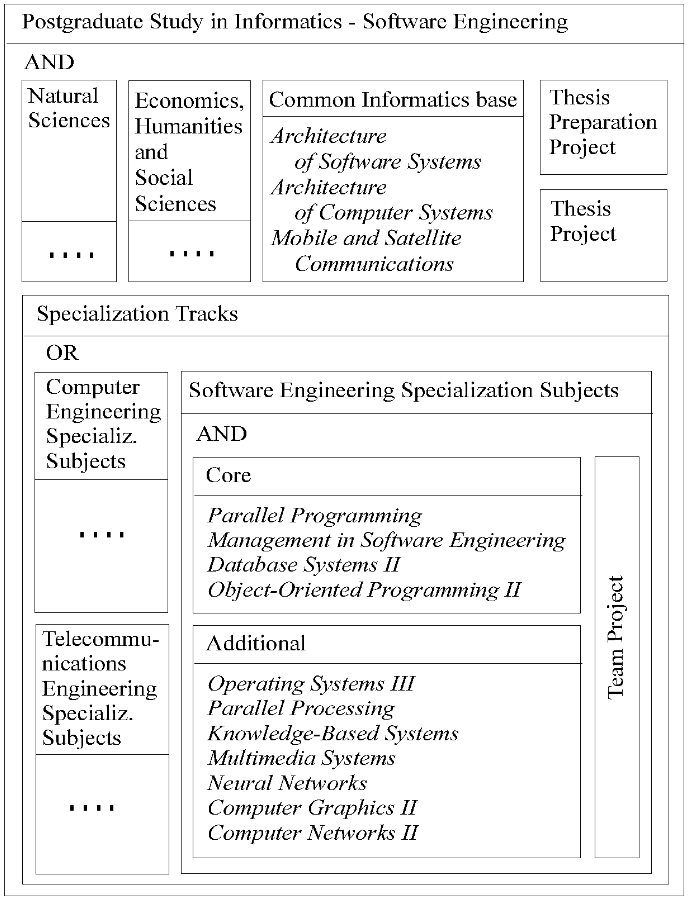

Fig. 2. Postgraduate study in Informatics.

management issues, and cover structured system analysis and design in a considerable depth to educate the students sufficiently in the software engineering profession so that they acquire necessary knowledge to be able to assume adequate jobs if they do not continue in postgraduate study.

Moreover, the context is such – as explained above – that most of the graduates will continue to study the subsequent master's course.

For both of the above reasons, we concentrate the software engineering education mainly to the postgraduate Informatics course. It is a three semester course with one semester devoted entirely to a thesis project. The first two semesters include one Mathematics, one Computer or Telecommunications Engineering subject, a two semester thesis preparation project, and a two semester team project. One humanities/ social sciences elective must be included, and there is also one free elective. The rest are five Software Engineering electives.

From the above outline of the specific background of our revisional efforts towards strengthening the software engineering education offered by our university, it can be seen that it is rather complicated and in some aspects quite different from the normal developments in regions not going through such a transition. On the other hand, many fundamental questions are the same.

One such question asks about relationships between software engineering and other related engineering disciplines. In particular, the relationship to computer engineering is in our view quite close. Many fundamental problems are similar and so are the methods to tackle them. Hardware or software specification and description languages are one example. Human computer interaction seems to be a matter of software, but how fundamentally different is the topic of peripheral devices in computer engineering? Recently, the architecture of software systems is becoming one of the core subjects in software engineering, fully paralleling the long ago established importance of architecture of computer systems. An interaction between the disciplines of the two engineerings takes place within our common department, and it often proves to be fruitful. For the students, the disciplines provide different perspectives, allowing us to generalize and to deepen understanding at the same time.

The distance of both to Telecommunications Engineering is bigger, but even here the common scientific topic i.e., communicating, processing and storing information justifies designing one common integrated course of Informatics with enough flexibility to allow relative autonomy of the three engineerings within it. So at the moment, we have an integrated undergraduate course of Informatics, and similarly a postgraduate one. Of course, we are aware of the possibilities of other solutions, and the configuration may evolve in time.

Our strategy of teaching Software Engineering is strongly based on practical work. As can be seen from the overview of the curricula of Informatics, and particularly of Software Engineering specialization tracks, students have to complete several projects during the study: a year project in the undergraduate study; a team project in the postgraduate study and a thesis project in the postgraduate study.

Projects are devoted to software systems development and/or analysis and maintenance of the existing systems. Team projects in Software Engineering specialization tracks are particularly to be stressed. Moreover, students participate in smaller projects which are linked to particular subjects.

5 Conclusions

We attempted to describe the context and the contents of higher education in Slovakia, which can be considered representative with respect to Central European countries in some sense not least because it is a typical country in a transition period and this fact has consequences to transformation of curricula [11]. We noted that contrary to the common practice in countries with a long tradition of a two level higher education (leaving aside the third PhD. level), where the majority of graduates do not continue towards the master’s degree, our tradition dictates – at least for some time – to an overwhelming majority of graduates to proceed towards the master’s degree. As a consequence to software engineering education, there is not felt any strong need of more specializing already at the bachelor’s level. Rather, we prefer an integrated scheme comprising a broader field of Informatics.

Our scheme, however, clearly identifies and distinguishes three specialization tracks within itself, with one of them being Software Engineering. In such a way, we have pushed more forward the idea of providing a self consistent bachelor’s level study in Software Engineering than e.g., Jackson, Manaris and McCauley [6] who report only on integrating software engineering concepts and techniques into an undergraduate Computer Science curriculum.

We report on an interesting experience of developing Software Engineering education within integrated Informatics curricula. This solution avoids making Software Engineering too narrow or too loosely defined due to the lack of sufficiently comprehensive own theory and methodology. Instead, a firm base is created by widening Computer Science foundations with some fundamental principles and theories of Computer and Telecommunications Engineerings with reference to common scientific problem of algorithmic information processing. Building on that, specific software engineering models, architectures and methods are developed. From among the two Informatics curricula, the postgraduate one offers more breadth and depth specifically with respect to Software Engineering, while the undergraduate one is more general in offering breadth in Informatics and going deeper into Software Engineering only to a relatively lesser extent. With respect to both of them, however, we can conclude that the contents of the curricula does not differ in any significant way from those offered at other institutes in other regions.

6 Acknowledgements

The work reported here was supported by the Slovak Ministry of Education, project No. PL977069S/1 (Commission of the European Communities, project INCO977069).

7 References

1. Bieliková, M., Galbavý, M., Kapustík, I., Molnár, Ľ., Návrat, P.: Using a CASE tool in developing an information system for Slovak Telecom. In: *Information Tools and Technologies*, vol. 1, Int. Academy of Automation (1996), 159-164.
2. Dijkstra, E. W.: On the cruelty of really teaching computing science. *Communications of the ACM* **32**(12) (1989) 1398-1404.
3. Ford, G.: SEI report on graduate software engineering education. Tech. Rep. CMU/SEI-91-TR-2, CMU Software Engineering Institute (1991).
4. Ford, G.: A progress report on undergraduate software engineering education. Tech. Rep. CMU/SEI-94-TR-11, CMU Software Engineering Institute (1994).
5. Holcombe, M.: Software engineering. In: Linkens, D. and Nicolson, R.I., eds. *Trends in Information Technology*, chapter 2. P. Perigrinus/IEE UK (1988), 17-37.
6. Jackson, U., Manaris, B.Z., McCauley, R.A.: Strategies for effective integration of software engineering concepts and techniques into the undergraduate computer science curriculum. In: Twenty-eighth *SIGCCSE Technical Symposium on Computer Science Education*, vol. 29, (1997), 360-364.
7. Leibfried, T.F., MacDonald, R.B.: Where is software engineering in the technical spectrum? *Int. Journal of Engineering Education* **8**(6) (1992) 419-426.
8. Molnár, Ľ., Návrat, P., Vojtek, V.: PARLAB - laboratory for parallel computing and programming. In: *Proc. TEMPUS Workshop on Engineering and Related Sciences*. Bratislava, STU and National Tempus Office (1996), 73-74.

9. Molnár, Ľ., Návrat, P., Vojtek, V.: Promoting education in the field of software engineering. In: *Proc. TEMPUS Workshop on Engineering and Related Sciences*. Bratislava, STU and National Tempus Office (1996), 31-34.
10. Návrat, P.: Intelligent support for software construction, and higher education in informatics at the Slovak TU: the DEC connection. In: *Proc. DECSYM 92 Latest Trends in Computing*. Side-Antalya (1992), 253-263.
11. Návrat, P., Molnár, Ľ.: Curricula transformation in the countries in transition: an experience from Slovakia. *IEEE Transactions on Education* **41**(2) (1998) 88-91.
12. Rösel, A., Bailes, P.: Identifying foundations of software engineering expertise. *SIGCSE Bulletin* **24**(4) (1992) 52 - 63.
13. Shaw, M.: Prospects for an engineering discipline of software. *IEEE Software* **7**(11) (1990) 15-24.
14. Tucker, A.B.: A summary of the ACM/IEEE-CS joint curriculum task force report: Computing curricula 1991. *Communications of the ACM* **34**(6) (1991) 69-84.
15. Wulf, Wm.A.: Computer science and software engineering: splitting is the wrong solution. *Computer Science Education* **3** (1992) 123-134.