Importance and Contribution of Software Engineering to the Education of Informatics Professionals

Dr. Tick, József
Budapest Polytechnic, Hungary, tick@bmf.hu

Abstract: As a result of the Bologna process a new form of higher education will be introduced in the European Union. Under this two-tier degree structure, bachelor and master degrees can be gained after finishing each cycle. Hungary will also join the Bologna process and the new form of higher education will also be introduced in the years to come. In order to gain experience, a pilot project will be launched in certain colleges and universities at certain faculties and majors in 2004. The Budapest Polytechnic will launch the first Technical Informatics BSc course with 330 students in September 2004. In the light of the already defined competencies and the short- and long-term market needs the piloting project encourages higher education professionals to reformulate those units of knowledge and skills and their ratio, which make up the professional knowledge of a graduate in informatics. In this paper I will summarise the corner points of the software engineering course as a contribution to the general knowledge of professionals in informatics.

Keywords: Software Engineering, Education

1 Introduction

The new two tier degree structure that was accepted in Bologna by the European Union is made up from new types of bachelor and master courses. A significant step in the introduction of this new higher education system is the introduction of the bachelor course (first degree course) in the frame of a pilot project, on the basis of the criteria laid down by the Hungarian authorities, in certain colleges and universities at certain faculties and majors in the autumn of 2004. The John von Neumann Faculty of Informatics at the Budapest Polytechnic will participate in the pilot project and will launch the technical informatics BSc course this September.
A high level training of informatics has already been going on at the Budapest Polytechnic and at its ancestor institution for several years. The curriculum and thus the professional learning material taught have continuously been revised and upgraded. The introduction of the new higher education system has given a handle to rethink the fields and ratio of the professional learning material to be taught in the light of the competencies.

This paper highlights the software engineering aspects of this revision. The paper referred in [1] examined the necessary knowledge to be taught and skills to be trained from the aspect of software engineering specialisation. This paper, however, examines software engineering as part of the general informatics-engineering learning material taking it as an element of the basic professional culture. The question is then, how competent the graduates - finishing their bachelor course in the framework of the pilot project first in 2008 - must be at software engineering, what kind of knowledge and skills they must have, regardless of their specialization such as hardware design, informatics network, mobile informatics or robotic techniques.

In order to answer this question we must take off from the basic competencies set up in the course requirements [2] for BSc informatics engineers:

- Programming in object oriented and visual programming environment
- The knowledge and implementation of software development methodologies, the usage of developing tools
- The installation, configuration, tuning, error location, error shooting, operation and upgrade of modern, general purpose operating systems
- Development, creation, modification of data bases in modern data base system
- In the relevant field of the training specialization the knowledge of basic practical methods and solutions, and the basic skills necessary for engineering activities considering design, development, implementation and operation of systems.

Taking software engineering into account, primarily the second and also the first and the third of these requirements are to define the objectives and goals. From these requirements the basic competencies referring to software engineering can be deducted.
2 Key-competencies of software engineering education

On defining basic competencies, not only the above mentioned requirements must be considered but also the fact that implementation will be realised in a separate subject, which must make up a unit itself. Thus, the competencies to be defined are as follows:

- The knowledge of a necessary theoretical background of the science of software engineering,
- Skills to be able to solve complex, large, software-intensive projects
- Skill level knowledge of software project management
- The knowledge and skill level usage of modelling based software development methodologies
- Practice in using Computer Aided Software Engineering (CASE) tools
- Practice in team work

Beyond these basic competencies some further aspects are to be raised to make the unit complete:

- The training must have an adequate theory/practice ratio due to the specialities of software engineering
- The learning material must be relatively long lasting and taking life long learning into consideration it has to provide a stable base
- The learning material must be up to date, valid and „immediately usable”
- The usage of such a CASE tool is needed in case of a project work that is widespread, while practice in its handling partly raises the value of a student in the labour force market and partly gives a good base to learn the usage of further other tools.

Taking the above into consideration the subject frameworks, the curriculum and the methodological elements can be set up.

3 Elaboration of subjects supporting the realisation of the competencies relevant to software engineering

Beyond the above defined competencies, both the environment and the framework, - which the subjects that teach software engineering learning material,
train skills and sharpen students’ faculties must fit in, - have to be considered in order to define the learning material, the framework subjects as well as the content and the methodological elements of the relevant subjects.

The accepted curriculum of the technical informatics BSc course concerning the general professional learning material is divided as follows:

- **Basic core:**
  - Basic knowledge of Mathematics and Natural Sciences
  - Economic and human learning

- **Professional core:**
  - Programming module
  - System technique module
  - Information systems module

The software engineering subject is placed in the programming module thus it has to adjust most to this narrower environment. The content of this module weighted with the total number of lessons follows below:

- **Introduction to information technology** 30 lesson lab
- **Programming paradigms and techniques** 45 lesson lecture
- **Imperative programming** 45 lesson lab
- **Object oriented programming** 30 lesson lab
- **Basics of WEB programming** 30 lesson lab
- **Visual (event driven) programming** 45 lesson lab
- **Declarative programming** 30 lesson lab
- **Software development and engineering** 45 lesson lecture
- **Software engineering practice** 30 lesson lab
- **Advanced algorithms** 30 lesson lab

The number of lessons well indicates the subject’s weight as well as its role within the module. Software engineering is taught in two subjects (software development and software engineering practice) meaning that it plays a significant role with its total of 75 lessons in the module.
4 Contribution of software engineering to the building up of an informatics approach

The significance of software engineering in the programming module, furthermore in the whole bachelor course is larger than the ratio of the number of lessons would indicate. That is because this is the subject, in which, apart from theory, practical and managerial learning material can be taught, thus creating a certain approach and attitude of the students. Beyond the strictly taken software making and software developing tricks, software engineering contributes to the general knowledge and skills of graduates in informatics by developing a certain approach.

4.1 Development of a product based approach referring to development and production

It is true not only in case of software production that a significant percentage of the graduates in informatics deal with development, product elaboration and production. This activity cannot be for its own sake, it must meet real market demands. Apart from the professional solutions, three factors must be considered, which the students must be prepared for:

- **Software as a product.** The developed software product must be marketable and should meet the requirements of the market and the profession (informatics and the information domain). It must stand its ground compared to a competitive product. It means that at the phase of design professional aspects cannot be exclusive and the market situation must also be taken into account.

- **The possible models of development process.** Students must know the steps of the development process, their sequence and the benefits and drawbacks of the possible paradigms. Students must have practice in the documentation of the process and in supporting CASE tools, they must know about the opportunities of standardisation, reuse and quality assurance. Students must get to know the set experience gained through the application of models.

- **Consideration of the characteristics of the development process.** Students must be prepared for the fact that development is such a process that contains several such elements that are not strictly connected to the professional side of software development, although, they are quite important parts of the development. From professional point of view the most outstanding part of software development is design, however, other fields are also significant in case of development:
Communication with the customer, effective information collection, organisation, and control of the necessary information are all essential to a successful project.

Invitation of the experts of the information domain; nowadays a successful project cannot be carried out without working and cooperating with them.

The impact of project management on development considers not specifically professional aspects. That is why it is crucial to be able to make compromises and deal with trade-offs in case of professional tasks.

Consideration of market and economic aspects partly in the course of analysis and partly in the course of design.

4.2 Skill level training of team work

In order to give a better chance for a better job for those graduates who will not continue their studies on the master course and will work after gaining their BSc degree, the expectations of the market and the employers are especially important. In the list of these expectations practice in team work is regularly takes the first place. That is why fundamental requirements for students are to consider the team work feature of software development, to practise team work and to train such a type of work up to a skill level. This field includes several unique aspects:

- Learning of team organisation possibilities
- Learning of tasks and roles as a team member
- The features of teamwork

4.3 The effective usage of object based modelling in the course of analysis and design

Software engineering discusses the development methods of medium and large size software-intensive systems. The analysis and design of such systems can be realised only if it happens at a higher abstraction level and the system is modelled. Modelling, however, is a general method, which can be used not only in the field of software development but in other fields as well. Together with object oriented approach, modelling can be especially effective in the solution of a general problem. Software engineering is especially significant because this subject gives a chance to learn and practise this method and to gain experience in it. UML, which is taught in the subject, is excellent from such a point of view that it is widespread, can be used in several fields and that the system’s models can be defined from several different viewpoints. These viewpoints are as follows:
• Architecture modell
• Behaviour modell
• User modell (usecase)
• Implementation modell

Several diagramms defined in the framework of the above modells give a good hand to support the entire scale of the development (from analysis to testing) not only in software engineering but in several fields of informatics.

The large number of labs gives an opportunity to turn the theoretical professional and managerial knowledge of students in the students’ professional practice into skill level practice. The following features are highlighted in the course of labs:

• Complex problem to be solved as a project in team work
• The documentation of the whole project from starting through phases to the final report must be well elaborated
• Project management to be based on a certain organisation method
• Roles and responsibilities in the team must be well defined
• Use of object-oriented modelling method is obligatory in the course of development
• Use of modern CASE tool is obligatory
• Evaluation of interim and final reports is based on individual performance and responsibility
• Students must present the project on completion

The practical and technical knowledge, and experience gained in the project carried out according to the criteria set above can be well applied not only in the fields of software engineering but in other fields of informatics as well.

Conclusions

Software engineering education plays a significant role in the informatics BSc course since this subject can teach and train such basic skills to graduates finishing their studies at the bachelor level, which are, by all means, necessary in software development and production, can be used well in the entire informatics profession and thus raise the chance of getting a better job after graduation.

References
