IMPLEMENTATION OF MUTUALLY SUPPORTING COURSES AND PROJECT-ORIENTED LEARNING IN "SOFTWARE ENGINEERING" BACHELOR'S PROGRAM

Dmitrii Kuzin

Associate Professor of Automation and Computer Systems Department

Andrei Zapevalov

Associate Professor, Head of Automation and Computer Systems Department

Inna Shukurova

Associate Professor of Foreign Languages Department

Valeriya Bezuevskaya

Vice-Rector on Development of Surgut State University

Sergei Kosenok

Rector of State-funded Educational Institution of Higher Education «Surgut State University»

ABSTRACT

This article describes the way used and experience of making explicit connections between several mutually supporting courses and learning outcomes as a "learning track" of Software Engineering bachelor program according to CDIO Standard 3 "Integrated Curriculum". A part of the curriculum considered in the article consists of seven disciplines that form a long-term interdisciplinary project. By the authors' opinion it contributes to increase student's activity in building products, processes and systems.

KEYWORDS

Interdisciplinary project, Software engineering, Mutually supporting courses, Learning track, Standards: 2, 3.

INTRODUCTION

Software engineering bachelor program has been implemented at the university since 2001. About 120 alumni graduated from this program between 2006-2017. But the more experience we got, the more clearly we saw the need to modernize the program. The main challenges for us become globalization and tighter competition in higher education, widening gap between graduates' skills and employers' demands, rapid technological changes and the beginning of the transition to a post-industrial economy. Given this, we considered the CDIO concept as a framework for modernizing the educational program and joined the CDIO initiative in June, 2017.

It should be noted that in addition to external challenges, there were several circumstances, inherited from the Russian higher education system and impeding the modernization of the educational program as a lot of state regulations and rules governing the educational process. Due to these circumstances, the CDIO implementation takes more time than it could, but already in 2017 we started to train students on the updated curricula. By this point, we had a large experience in monodisciplinary project-based learning, however, interdisciplinary projects implementing was the greatest challenge.

CONSISTENCY OF COURSES GOALS AND PREREQUISITES

Before joining the CDIO initiative Software Engineering bachelor program was implemented in the university as a traditional higher educational program with following disadvantages:

1. Large amount of courses (up to 60 pcs. for 4-year bachelor program)
2. Monodisciplinary curriculum
3. Inadequate consistency of courses with each other
4. Domination of theoretical training
5. Lack of active learning methods

Disadvantages 2 and 3 led to the facts that often courses were taught in wrong sequence, changing course content wasn't properly monitored. Even the addition or removal of some courses from the program was ignored by other faculty members, because each course was a "thing in itself." Why does it happen? Although official courses objectives and prerequisites defined, there was no special procedures to guarantee courses content consistency.

The CDIO Standard 3 "Integrated Curriculum" defines the mutually supporting courses as courses that make explicit connections between related content and learning outcomes (Crawley et. al., 2011). Establishing that connections has required detailed specification of learning outcomes for all courses and harmonization of the course content (Malmqvist et. al., 2006). So, the very important step to implement CDIO Standard 3 "Integrated Curriculum" is harmonization of courses content from the point of view of learning objectives and prerequisites. Notice, that here we are talking about disciplinary knowledge and skills.

DESIGN LEARNING TRACK PROCESS

As it was mentioned above, the harmonization of courses content is an important task for building an integrated curriculum. But due a large number of courses in the program the alignment of goals and prerequisites for all courses is a too complicated task. So, according to the article authors' opinion, it is necessary to find chains of disciplines with closely related disciplinary knowledge and skills and start harmonization process for those disciplines. We called such chains of disciplines *learning tracks*. Identifying learning tracks is **a first stage** in the overall learning track design process showed in Figure 1. For example, one of the learning tracks of Software Engineering bachelor program described in the article consists of seven disciplines, listed in Table 1.

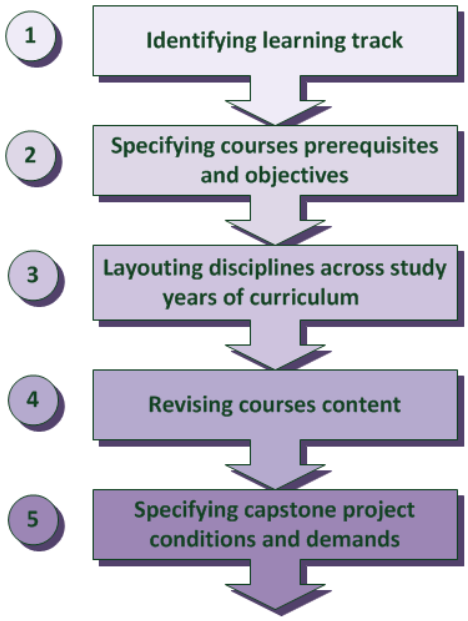

Figure 1. Learning track design process in stages

According to Figure 1 **the second stage** of the learning track design is specifying prerequisites and objectives for learning track courses (see Table 1). In Table 1 the learning outcomes of each course (also used as prerequisites for other courses) colored the same color as the course. For example, learning outcomes of Web-Technology Basics course consist of HTML, CSS, JavaScript and PHP knowledge, that is necessary to start learning Web-Programming.

Table 1. One of the learning tracks for Software Engineering bachelor program

Prerequisites	Course	Objectives (outcomes)
programing basics, mathematics	**Project Activity Basics**	software development management, teamwork, client-server software architecture, software lifecycle
programing basics, client-server software architecture, software lifecycle	**Web-Technology Basics**	HTML and CSS design, JavaScript programming, PHP programming
HTML and CSS design, JavaScript programming, PHP programming	**Web-Programming**	HTTP-cookies, PHP-sessions, advanced JavaScript and AJAX programming
HTML and CSS design, JavaScript programming, PHP programming, advanced JavaScript and	**Web-Application Development Technology**	JQuery programming, MVC-architecture, PHP-frameworks using

AJAX programming		
HTML and CSS design, JavaScript programming,	**XML-Technologies**	XML, XSD, XSLT, Data presenting and transformation
programing basics, mathematics	**Database Management Systems**	relation database development, SQL querying, DB-server programming
PHP-frameworks using, relation database development, DB-server programming, Data presenting and transformation	**Capstone Project**	***Web-based information system development***

The third stage of the learning track building is making allocation of disciplines across the academic years. Figure 2 shows the example of courses allocation (used the same color for the courses as in the Table 1). Usually, courses follow in the sequence according to the courses outcomes and prerequisites consistency. If learning outcomes of one course are used as a prerequisite for another course, the first course is learned before the second one, and in the diagram the previous course is included entirely in the area occupied by the subsequent course. The area delineated by the bold line in the diagram shows the assignment of the course to the academic semester.

As shown above, some courses have explicit relations via prerequisites and objectives and they should be studied consequentially. But for some courses restrictions on consistency of prerequisites and objectives are not strong, so such courses can be studied simultaneously, for example Database Management Systems and Web-Application Development Technology. In some cases such courses are mutually supporting and have relations via learning content. For example, according to interdisciplinary approach, studying Database Management systems students develop user interfaces using web-technologies that they learn in Web-Application Development Technology course. Providing of such mutually supporting courses requires precise synchronization of course contents and revision of content is the **fourth stage** of the learning track building.

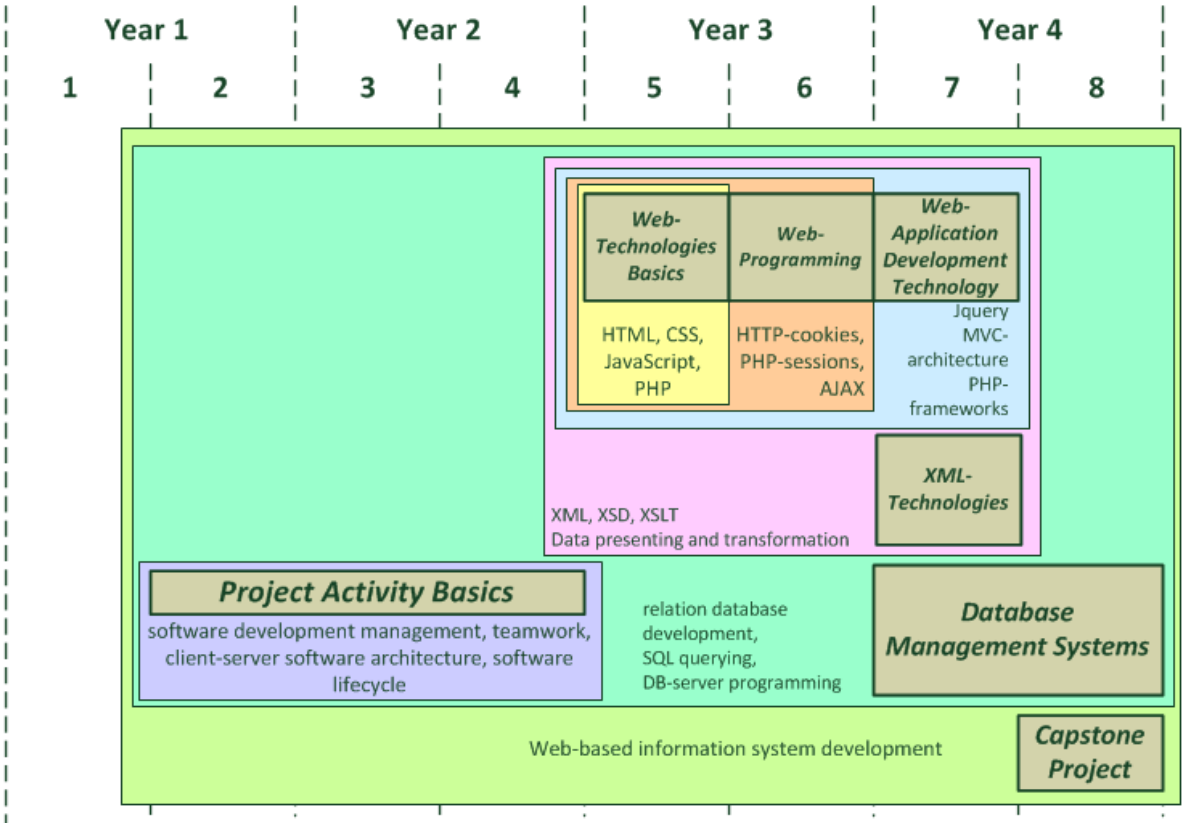

Figure 2. Learning track diagram with courses objectives

GENERAL INTERDISCIPLINARY PROJECT

Each course content in the learning track consists of a sequence of exercises that let students develop their own information system step-by-step. Therefore, studying each course students design a software product that will be developed in further courses. So going through the learning track students make a general interdisciplinary project (capstone project). Each project is an information system with typical architecture shown in Figure 3 (each architecture component colored as the course with the appropriate learning outcomes in Figure 2). The developed information systems are the prototype for different application fields such as logistic, warehouse, trade, booking etc. Developing an information system, students can work both individually and in small groups. To define conditions and demands for the general interdisciplinary project is the **fifth stage** of the learning track building.

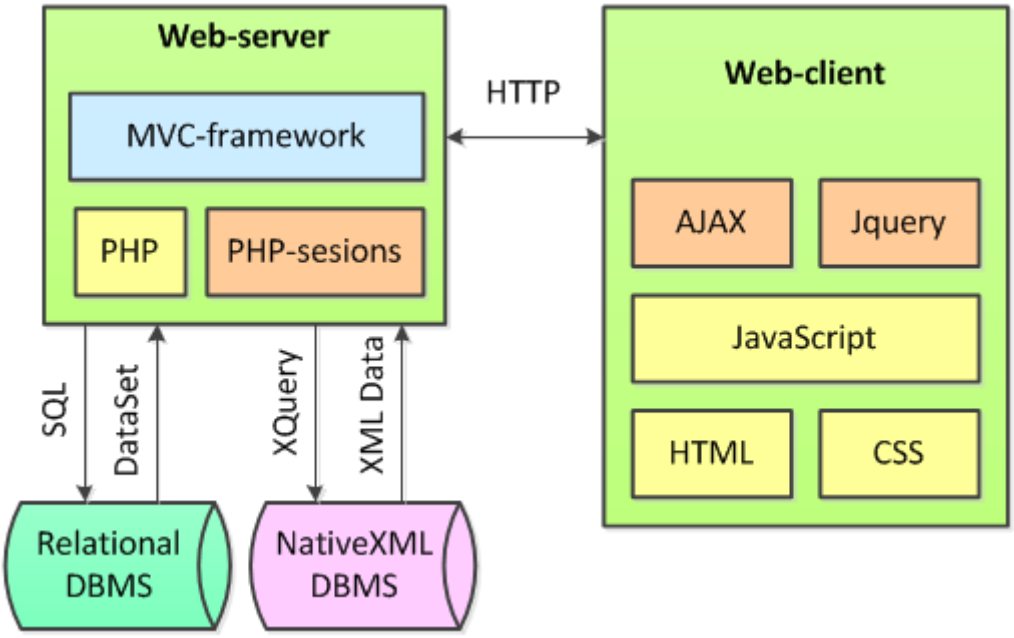

Figure 3. Typical architecture of web-based information system developed by a student

CONCLUSION

This article describes the organization of project-oriented learning using an interdisciplinary approach and mutually supporting courses in Software Engineering bachelor program. The scope of mutually supporting courses in the article considered as a learning track. In fact, it is necessary to make the following sequence of steps to provide a learning track in the educational program:

1. To identify and name learning tracks.
2. To specify prerequisites and objectives for each course.
3. To layout disciplines for reasons of better consistency of learning objectives and outcomes.
4. To revise the content of courses.
5. To define conditions and demands for general interdisciplinary project as a result of learning track passing.

Using these steps, it is possible to allocate 6-8 such learning tracks in the curriculum that will simplify the implementation of the CDIO-standards in the program and will allow to do this in a piecemeal manner. The next task is to develop a detailed specification of soft skills mapped to the CDIO Syllabus 2.0 and the modernization of educational content for the development of such skills.

REFERENCES

Crawley, E.F., Malmqvist, J., Östlund, S., Brodeur, D.R., & Edström, K. (2011). *Rethinking Engineering Education: The CDIO Approach.* Switzerland: Springer International Publishing.

Malmqvist, J., Östlund, S., Edström, K., “Using integrated program descriptions to support a CDIO programme design process”, World Transactions on Engineering and Technology Education 5(2), 259-262, 2006.

BIOGRAPHICAL INFORMATION

Dmitrii Kuzin, Ph. D. in Computer Science, Associate Professor of Automation and Computer Systems Department. His current research focuses on information representation and processing with ontology-based intelligent information systems.

Andrei Zapevalov, Ph. D. in Automation, Head of Automation and Computer Systems Department. His current scholarly activities focus on curriculum development methods and teaching methodology.

Inna Shukurova, Ph. D. in Pedagogical Science, Associate Professor of Foreign Languages Department. Her current scholarly activity focuses on the soft skills development through teaching foreign languages.

Valeriya Bezuevskaya, Ph. D. in Chemistry, Vice-Rector on Development. Her current research focuses on project management and international education program development.

Sergei Kosenok, Doctor of Pedagogical Science, Rector of State-funded Educational Institution of Higher Education «Surgut State University». His current research focuses on management and policy in education.

Corresponding author

Dmitrii Kuzin
State-funded Educational Institution of Higher Education «Surgut State University»
1 Lenin Avenue, Surgut, Khanty-Mansi Autonomous Okrug, 628412, Russia
kuzin_da@surgu.ru
Phone: +7(922)258-82-16