561884-EPP-1-2015-1-DK-EPPKA2-CBHE-JP

Introducing Problem Based Learning in Moldova: Toward Enhancing Students' Competitiveness and Employability

www.pblmd.aau.dk

"Sustainability Strategy"

Teaching and Problem-Based Learning at the Technical University of Moldova

Work Package 5

Developed by: Viorel Bostan (UTM Institutional Coordinator)
Dumitru Ciorba (SPT Leader)
Irina Cojuhari

Evaluated by: John Reilly (External Expert), University of Kent, United Kingdom
Olle ten Cate (External Expert), University of Utrecht, the Netherlands
Ralph Dreher, University of Siegen, Germany
Olga Kordas, Royal Institute of Technology, Sweden
Colin Simpson, University of Gloucestershire, United Kingdom
Olav Jull Sørensen, Aalborg University, Denmark
Ivan Aaen, Aalborg University, Denmark

"This project is funded by the European Commission. Financial support from the European Commission for this project does not constitute an endorsement of the content that reflects only the views of the authors and the Commission can not be held responsible for any use of the information contained in this draft document."

Chişinău, 2019

Summary

The goal of the proposed sustainability strategy is twofold:

- Firstly, it refers to its own development and implementation experience of the new higher education Bachelor’s level program based on PBL - Software Engineering (Section 2).
- Secondly, it refers to the description of the software engineering study program and program implementation vision (Section 3).
- Thirdly, it presents the translation from the PBL Pilot Program Software Engineering to the education based on the PBL 2022 Teaching Methodology (Sections 4 and 5).
- Fifthly, it offers recommendations at all levels to ensure the sustainability of the PBL as a teaching and learning methodology (Section 6).

List of definitions

PBL - student activity model with group task assignment to solve a problem, which is the cumulative result of activities from several courses, constituting an interdisciplinary product, guided by the teacher responsible and evaluated by practitioners in the field.

Student-centered learning - teaching and learning process in which the student becomes a partner in the educational process, and the teacher-student relationship is based on cooperation and collaboration.

Self-guided learning – teaching and learning process that emphasizes the student's responsibility to create learning and experimentation environments, in which they discover knowledge, make discoveries and solve problems on their own.

Learning objectives – general competences by training fields required for graduates of study programmes.

Learning outcomes - clear results, describing the student's knowledge or skills, expected from the teaching-learning process.

Progression – succession of expectations from the teaching-learning process in several stages.

Assessment - multicriterial examination of students' knowledge accumulated in the learning and teaching process.

Projects – are tasks given to students which consist of research and analysis of a problem (both theoretical and practical) and the generation of new approaches or solutions. Projects can be individual and in group.

Semester projects – are the projects carried out by students (usually in the group) during a semester. These projects may have inter-disciplinary character (may refer to two or more disciplines studied during the same semester).

Group/team work – is the joint work of a group of 4-5 students to perform a single task, which is based on communication, collaboration and self-discipline, each member of the group contributing to the achievement of the final result.

Research-based teaching and learning - the process of transmission and accumulation of knowledge, as well as the creation or development of skills that are based on some research tasks and aims to facilitate the learning (including individual) process of students.

Research-based teaching - is the process by which the student is involved in research exercises and is encouraged to reach his/her own conclusions and solutions using the results of the research carried out.

Sustainability strategy – is a long-term vision of an institution aimed at introducing key modifications in order to streamline the teaching-learning process. The strategy includes objectives and concrete actions, the deadline for achievement, as well as the potential outcomes that can be achieved.

(a) the path from a pilot programme to a comprehensive study programme based on problem-based learning (PBL) - the concrete steps to introduce PBL in a study programme

(b) support and promotion of PBL for teaching and learning - performing information and training measures about the advantages and efficiency of PBL.

Credit (ECTS) – the credit is a conventional unit used to calculate the workload performed by the student within a determined time period to achieve certain outcomes and competences. The credit is a tool to ensure the quality of the training.

ECTS (European Credit Trasnfer and Accumulation System) - European system of accumulation and transfer of credits. The Bachelor’s degree studies correspond to 180-240 of transferable study credits, with 30 credits per semetcher.

Profile degree – the educational framework to be known by graduates in order to obtain the title of Bachelor, Master.

Professional development – opportunities offered to the teacher to strengthen their pedagogical skills, competences and approaches; continuous improvement of staff through trainings, internships, etc.

Facilitator – the person who helps a group of students understand their common goals and helps them plan how to achieve the objectives set out in the joint project.

Internship placement (training/practice) – institution/organization where students will conduct internship/training.

Quality assurance – a systematic monitoring and evaluation programme of the different aspects of a project in order to ensure compliance with quality standards.

Table of Contents

1 INTRODUCTION

The Working Package 5 - WP5 - describes the sustainability strategy for the implementation of PBL, student-centered and active teaching and learning at the Technical University of Moldova (TUM). This report details the new bachelor's degree program based on PBL - Software Engineering, and covers a road map and an action plan that will guide staff and university management in their efforts to fully implement PBL, student-centered and active teaching and learning in the respective study program and in the university.

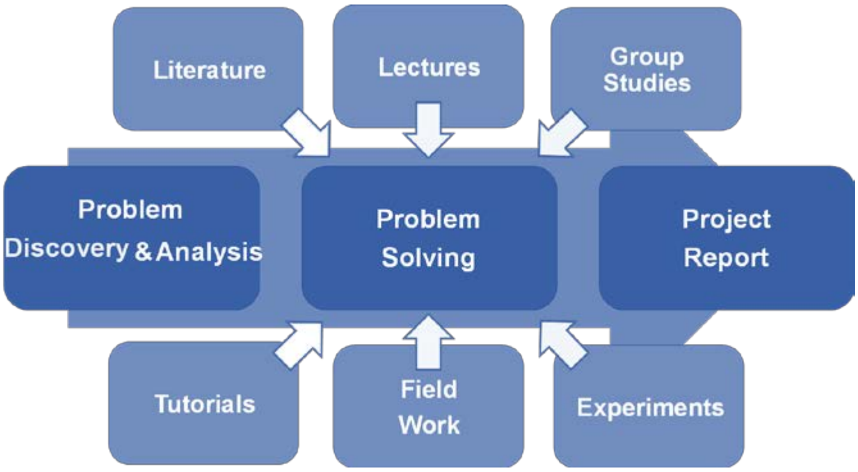
This report is based on WP2-WP4, developed between 2015-2017 and implemented since September 2017 through the launching of the Software Engineering Pilot Program at the Faculty of Computers, Informatics and Microelectronics at the TUM. The report took also into account the experience gained during the study visits and the mobility of the teaching staff at the partner universities in the EU and on the PBL training events carried out in Chisinau by the EU project partners.

1.1 KEY ASSUMPTIONS

Approaching the learning process through the student-centered learning has always been topical; the student is considered an actor of his/her own training, built on the basis of capitalizing on and assuming previous learning experiences. This fact implies the use of active learning methods based on problem solving.

PBL (Problem Based Learning) is a modern learning philosophy that involves students in finding problems and identifying solutions to overcome them. There is no one PBL model suitable for all purposes. However, the PBL-based models are mainly substantiated on two key assumptions. The first hypothesis is that work on the project is in the *center*, and consists of the problem discovery and analysis, problem solving and project report (Figure 1). The second hypothesis is that other teaching and learning (face-to-face) activities, such as literature, lectures, group studies, and tutorials, are designed to *support* work on the project.

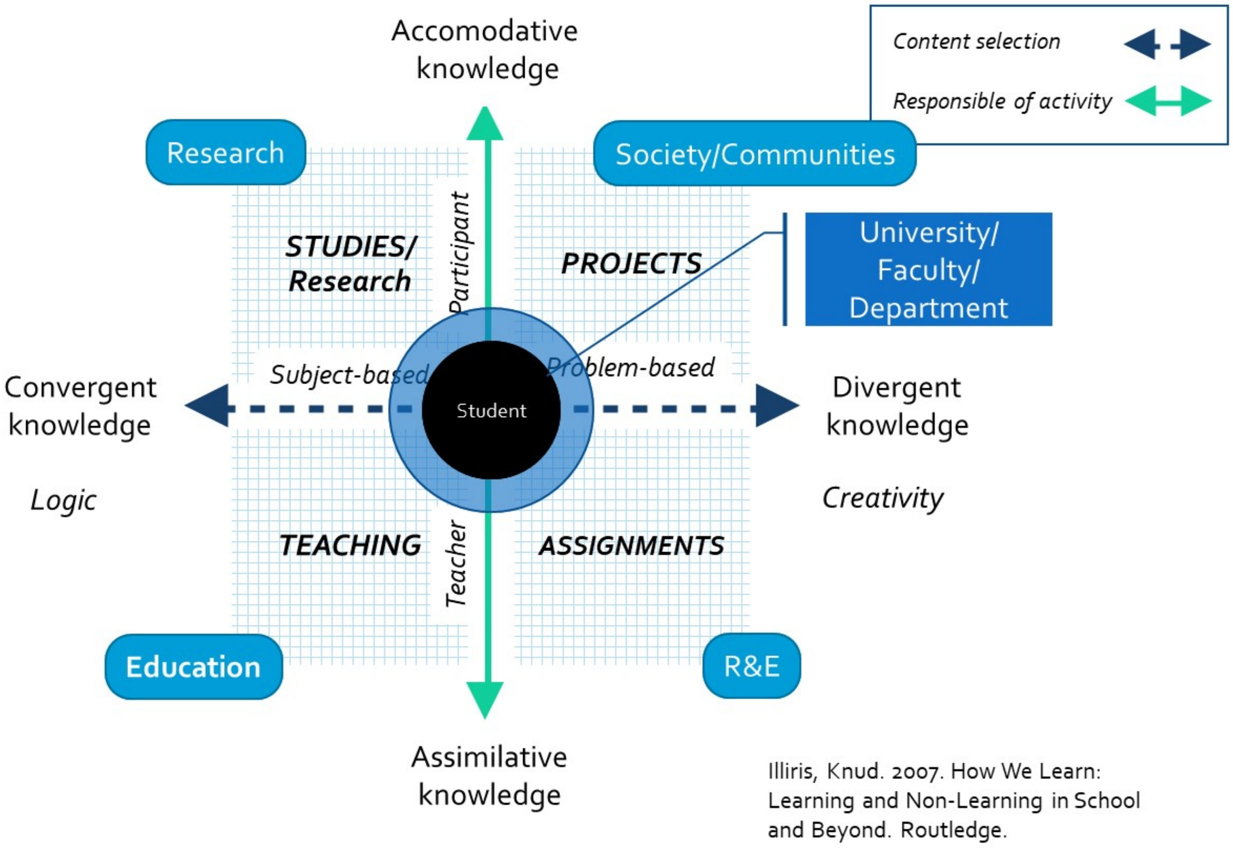
Figure 1: AAU PBL Model: An example



Source: AAU, 2017 (the word 'Discovery' is introduced by Romeo V. Turcan)

The above-mentioned aspects lied the basis of the PBL model development used in the Software Engineering Program Study, inspired from Illiris, Knud [1] (Figure 2). The given model involves problem solving through the implementation of interdisciplinary projects, with the introduction of the research aspects, where the process of education and assimilation of knowledge is accomplished in parallel through the process of teaching the course hours and assigning group or individual tasks contributing to developing student creativity.

Figure 2: The PBL model implemented within the Software Engineering Study Program

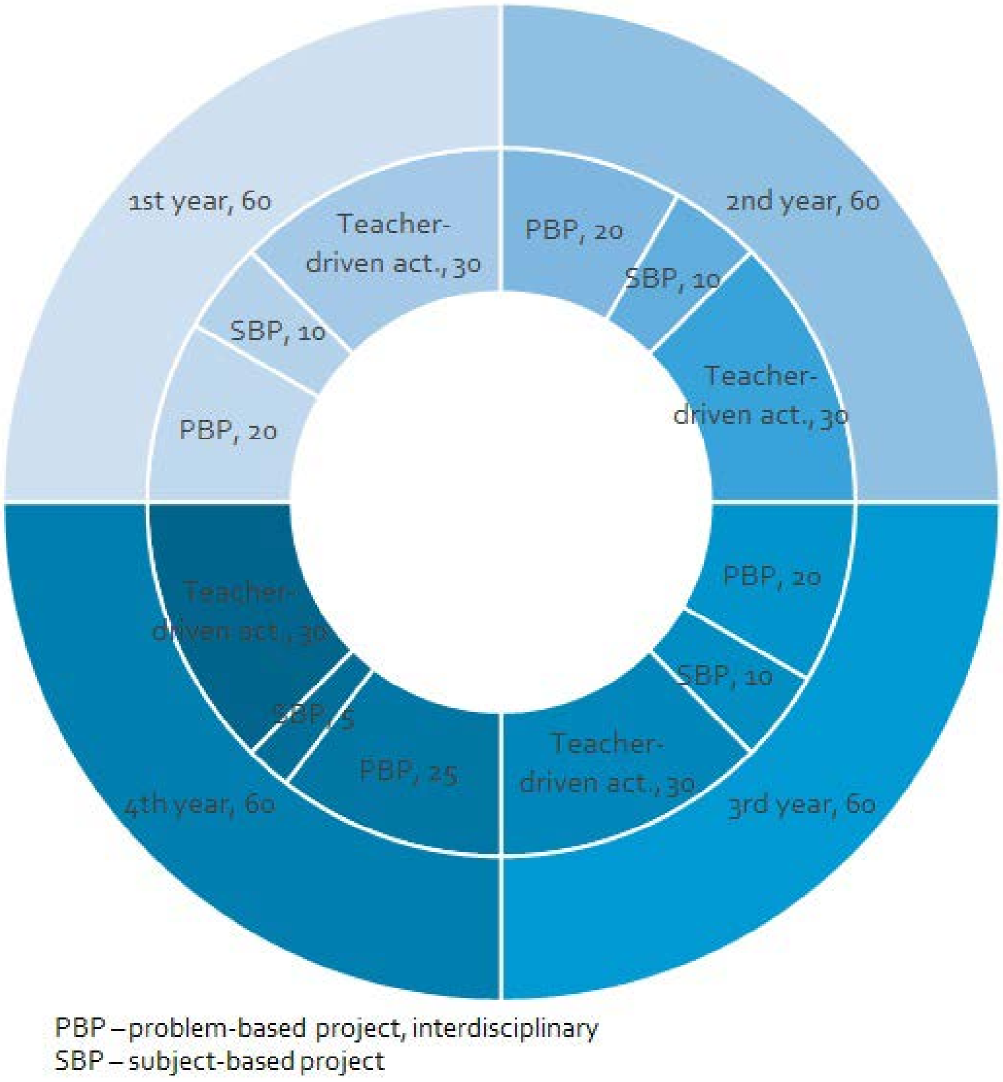


Illiris, Knud. 2007. How We Learn: Learning and Non-Learning in School and Beyond. Routledge.

The model shown in Figure 2 involves work on the project and face-to-face activities such as lectures, seminars, workshops, laboratories and experiments.

Another hypothesis relates to the relationship between work on the project and face-to-face activities. In the context of this report, all references to PBL-based mean a study program with an about 50:50 distribution between student work on the project and face-to-face activities (such as lectures, seminars, workshops, laboratories and experiments). An example of progression is shown in Figure 3. Of course, there are many ways to spread the relationship between work on the project and face-to-face activities during the semesters; the main purpose is to achieve an approximate 50:50 time distribution over the duration of the study program.

Figure 3: Time distribution, 50:50, between project work and face-to-face activities (Software Engineering / TUM)

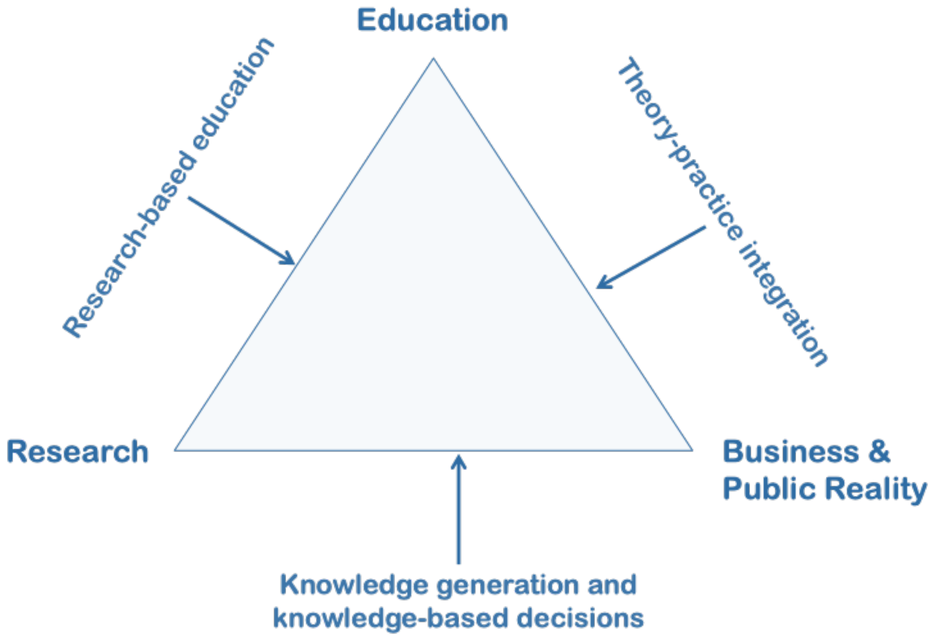


1.2 Expected Results

A number of results are expected to be achieved from the successful implementation of the student-centered Bachelor's Degree Program Software Engineering, based on PBL. It is expected that by 2020, this study program will become internationally recognized, attracting thus European and international students as full-time students or through exchange. Also, other bachelor and master degree programs at the Technical University of Moldova are expected to be redesigned based on PBL, with student-centered teaching methodologies and methods. Also, a better adjustment of students' knowledge, skills and competences is expected to match the needs of labor markets.

Successful implementation of this study program, as well as its dissemination effects across the university will contribute to the further development and enhancement of the integration of collaboration of education, research and business environment / policy makers (Figure 4). Academic staff will excel in engaging in research-based teaching, our students will learn and will become able to apply theories in practice, both in the private and public sector, and our researchers will work with private and public organizations to create and transfer new knowledge.

Figure 4: Socially committed university


Education
Research-based education
Theory-practice integration
Research
Business & Public Reality
Knowledge generation and knowledge-based decisions

Source: Olav J. Sorensen, 2015

1.3 Outline of the Report

This report begins by presenting the TUV vision of the PBL-based Bachelor's Degree Program on Software Engineering, in particular, with a general description of the curriculum, objectives and learning outcomes, and continues with a presentation of each semester, including learning objectives and learning outcomes, the transition from one semester to another, a description of the work on the project and the semestrial projects, including the learning objectives, the results and their evolution. Subsequently, the road map guiding the process of implementing the PBL-based bachelor's degree Software Engineering Program was presented and detailed. The report goes on with presenting and discussing the action plan detailing, for example, the specific activities, resources and internal policies needed to successfully implement the visionary study program. It concludes by providing the university management and university board with a set of policy recommendations on how to improve teaching and learning by introducing PBL, student-centered and active teaching and learning methodologies and methods in our university.

2 LESSONS LEARNED FROM DEVELOPING AND IMPLEMENTING THE SOFTWARE ENGINEERING PILOT PROGRAM

In order to implement the Software Engineering Pilot Program, the Roadmap presented in WP4 was developed. Several activities have been depicted in it for the successful implementation of the Software Engineering Pilot Program. The described activities have been grouped into three periods:

> *I Prepare for the launch of the new study program, where the aim is to prepare the legal framework, the physical environment, the teaching staff for the launch of the new Study Program - Software Engineering.*
>
> *II Implementation, which provides for the launch of the new study program as of September 1, 2017.*
>
> *III Promotion, where the promotion of the ERASMUS + PBLMD Project and the new Study Program - Software Engineering was planned.*

Within these periods, a number of activities were carried out, and namely: developing and adopting the new Software Engineering Education Plan; teacher training on PBL; preparing the infrastructure for teaching based on the PBL methodology; developing educational documents; 2017, 2018 admission processes; initiating students in the new PBL teaching methodology; carrying out the study process based on PBL methodology. Hence, various lessons have been learned over these periods.

2.1 PERIOD 1

The duration of this period is up to 2 years (2015 - summer 2017) and the goal is to prepare the legal framework, the physical environment and the teaching staff for the launch of the new Software Engineering Study Program.

Activities carried out and lessons learned over this period are presented in the table below.

No.	**Activities carried out**	**Lessons Learned**
1.	Training programs for the trainers involved in the teaching process of the Software Engineering Study Program. Training teachers on applying the PBL methodology. Participating in the trainings organized under the project at the	A new pedagogical approach. Getting familiar with the problem-based teaching methodology. Changing the knowledge assessment manner. Switching to interactive, student-centered teaching.

No.	Activities carried out	Lessons Learned
	TUM or the AESM in 2016-2019. Academic mobilities for teachers to the partner universities in the European Union.	Interdisciplinarity is present within each semester, making teachers collaborate with each other. Following the identification of teachers teaching within the Software Engineering Study Program, a resistance to switching from classical teaching to the new teaching method was noted. Use of the ICT in the teaching process. The need for teachers to undergo preventive training prior to being involved in the teaching process within the Software Engineering Program. The need for an as closely as possible interaction between teachers and program managers.
2.	Preparing the infrastructure for the PBL teaching means buying equipment and preparing classrooms, making them suitable for team work.	The PBL teaching methodology cannot be fully implemented without a well-formed infrastructure. This implies the existence of multiple rooms and workspaces where students can do teamwork; availability of ICT equipment and specialized communication media, such as the Moodle platform, or corporate mail; supplying library with literature.
3.	Developing and approving the Software Engineering Education Plan in accordance with: – The Education Code of the Republic of Moldova. – Framework-Plan on Higher Education, approved through the Ministry of Education's Order No 1045 of October 29, 2015. – The Reference Framework of the University Curriculum, approved by the National Curriculum Council within the Ministry of Education of the Republic of Moldova, 2015. – Software Engineering Curriculum Guide, ACM (Association for	The basic lesson learned at this stage was the development of a new Software Engineering Study Program, for which it was necessary to identify the profile of the plan, the skills to be developed within the given study program, the establishment of the skills assessment methods and criteria expected learning outcomes. The adoption of the education plan's structure in compliance with the Framework-Plan and the provision of the 50/50 ratio between theoretical hours (course / lessons), on the one hand, and seminars (internships) and projects, on the other hand, the obstacle being the nomenclature, the framework-plan.

No.	Activities carried out	Lessons Learned
	Computing Machinery) / IEEE Computer Society. – PBL methodology. – The draft of the new Nomenclature of Vocational Training Fields.	Interdisciplinary approach of the study program – to reflect a well-defined theme for each semester in the education plan. Ensure in the education plan the ratio between fundamental course units, general skills and competences development and socio-humanistic orientation.
4	Approving the Study Program at: - Department / Chair. - Faculty - TUM Senate. Internal and external evaluation of the Study Program.	In order for the changes to be introduced at the program level, instructional flexibility and support at the institutional, faculty and departmental levels is needed.
5	Developing educational documents: curriculum by subjects (analytical programs), guides, case studies, evaluations, etc. (for the first year of study).	In the Software Engineering Study Program, interdisciplinarity is provided in each semester, so that a problem encountered in the process of developing course cards and curriculum by subject was related to the formulation of course objectives to ensure interdisciplinarity. The Software Engineering program allows flexibility in carrying out study activities - seminars, design, laboratory work. These activities had to be identified and reflected in the subject curriculum. The semestrial project curriculum had to reflect the objectives and skills reflected in all semestrial subjects. Identification of possibilities to assess the skills developed during courses and project development.

2.2 PERIOD 2

The implementation period foresees the launch of the new study program from September 2017.

Between 2017 and 2019, two admissions took place, so the number of students enrolled in the Software Engineering Study Program is around 100 students.

Activities carried out and lessons learned over these two years are presented in the table below.

No.	**Activities carried out**	**Lessons Learned**
1.	2017 Admission 2018 Admission	Establishing the criteria for the Software Engineering Study Program admission. All students have to pass the English and math admission exam. The number of students admitted to the Software Engineering Study Program varies because of the percentage of students who, after the admission, went to study in Romania.
2.	Formulation and identification of design modules issues.	In two years, two types of projects have been formulated: assignment project, subject project, problem project. Difficulties have been encountered in formulating interdisciplinary problems with the application of research methodologies, so as to achieve the objectives presented in the subject card. Interdisciplinary projects have led to a dialogue among the subject teachers in the same semester with the formulation of common themes or problems.
3.	Teamwork.	In two years, various methods of team formation were tested: random; free choice; by program manager. In the case of the first year students, the recommended option is when teams are formed by the program manager, while the second year students are already capable to independently form their teams. It has been noticed that some students lack group working skills, leading to interpersonal conflicts and conflicts related to the assignment and undertaking of roles within the project, group communication deficiencies and time organization issues. Project monitoring by a supervisor and introduction of the mid-term project assessment helped overcoming these problems.
4.	Project development monitoring	The Software Engineering Study Program has a program manager, and the project development monitoring process is carried out by the supervisors. When the study program was first launched, there were difficulties in relation to fulfilling the role of

No.	**Activities carried out**	**Lessons Learned**
		supervisor, namely in establishing duties and the project monitoring process. Based on the experience gained from the working visits to the partner universities, there have been established ways of monitoring the project development process, involving weekly meetings between supervisors and students, as well as communication on the moodle platform and via the corporate mail. The supervisor in turn has the role of guiding students and monitoring the evolution of their projects based on the objectives specified in the curriculum.
5.	Team assessment	The problem with the team assessment was to identify the input and involvement of each team member in the project. Thus, in two years, different types of team assessment were tested: group assessment, individual assessment, peer-review, team assessment by private sector representatives. It was noticed that an objective assessment was possible in the case of both group and individual assessment, as well as peer review. A questionnaire was provided to the team members to assess the input of each team member to the project.
6.	Internships for students	During these years, the collaboration with the business environment was established. According to the Education Program, there are three internships for the 2nd, 3rd and 4th years of study. Based on the mobility experience, the semestrial project was planned to be carried out during the internship, with supervisors from the university and the firm. It was noted that this type of internship had a positive effect on the development of students' professional skills. Students were assessed both by university and industry representatives.

2.3 PERIOD 3

This period entails the promotion of the ERASMUS + PBLMD Project and the new Software Engineering Study Program. Activities and lessons learned are presented in the table below.

No.	**Activities carried out**	**Lessons Learned**
1.	Promoting the Software Engineering Specialty	Various ways of promoting the Software Engineering Study Program have been applied: mass media, press conferences, informative visits, leaflets, one-day visits by students, magazine articles. Promotion also requires the visibility of the specialty, an activity that has to be carried out throughout the whole year of study.
2.	Promoting the PBLMD Project	The PBLMD Project was promoted with the help of mass media, press conferences, articles in magazines.

3 PBL Software Engineering Study Program Vision

3.1 Overview

In accordance with the PBLMD Project Objectives, a new study program - Software Engineering (SE) was launched within the Department of Software and Automatic Engineering, Faculty of Computers, Informatics and Microelectronics of the Technical University of Moldova, in 2017.

The methodology of the education process in the Software Engineering Program is defined by a set of teaching-learning-assessment strategies corresponding to the problem-based learning principles. This is reflected by the ratio between theoretical hours (courses/lessons), on one hand, and seminars (internships) and projects, on the other hand, equal to about 50/50, the vision of the study program is presented in Figure 3.1, and the education plan in Annex 1.

Figure 3.1: Generic Structure of the Software Engineering Study Program

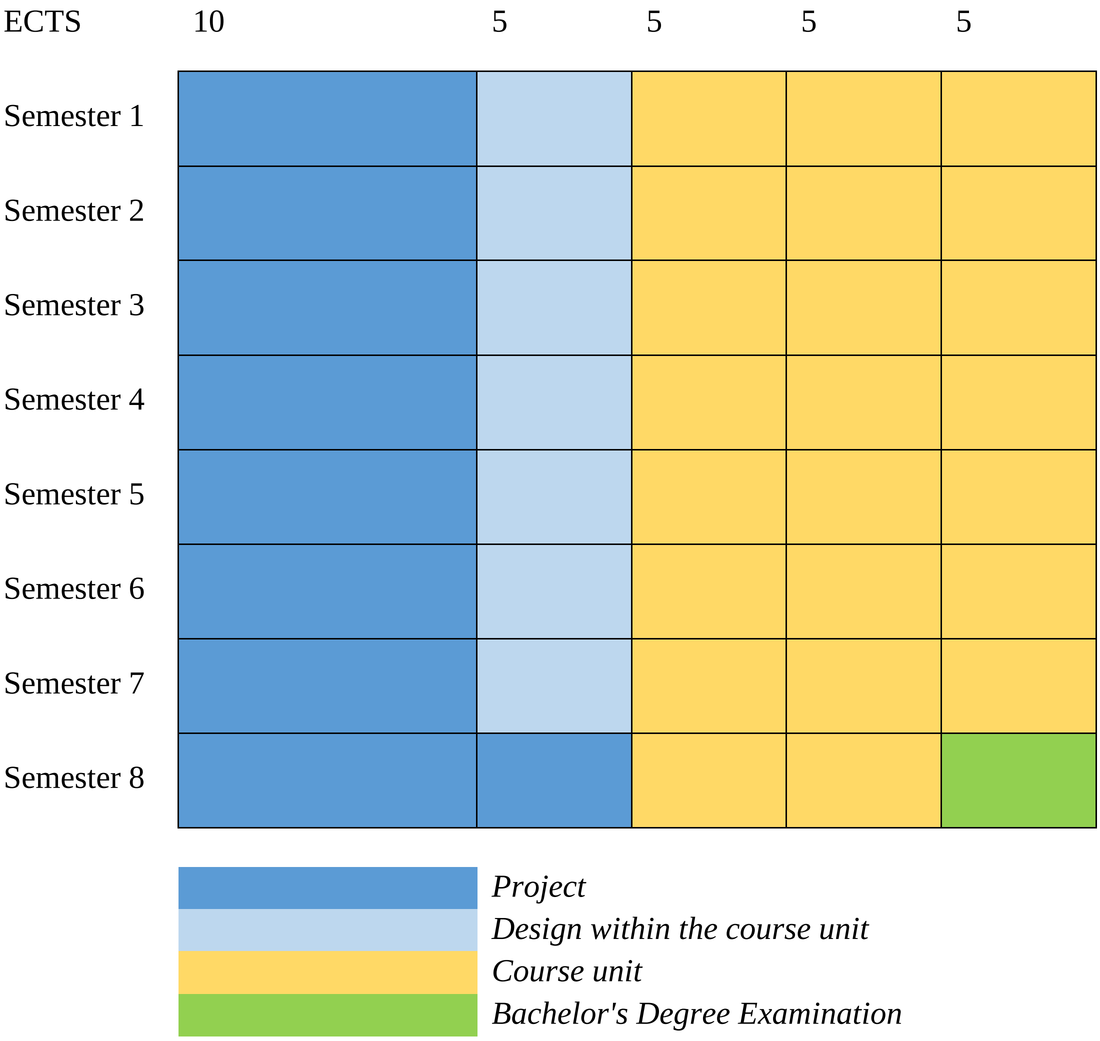
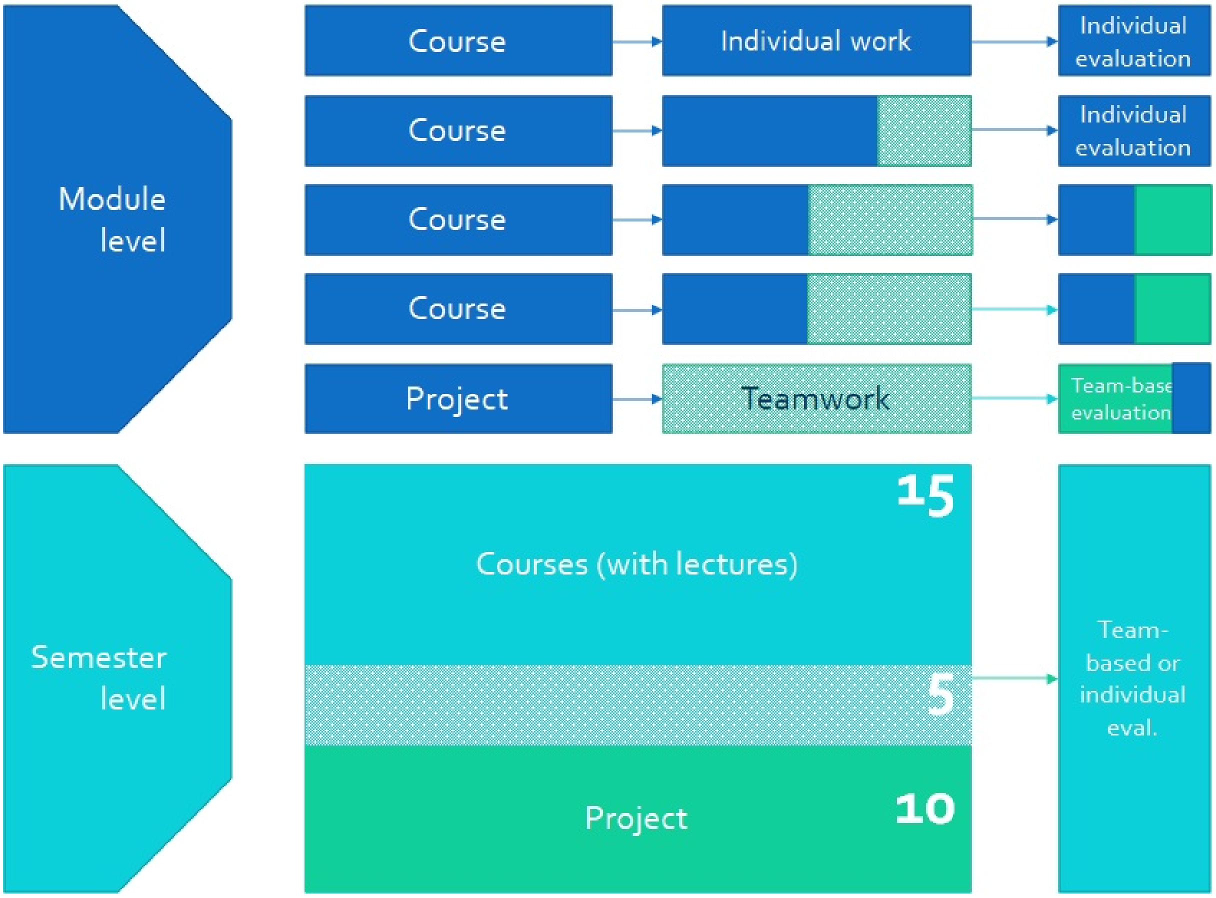


Figure 3.1 shows that it was proposed that every semester should have a separate design module of 10 ECTS, another 5 credits are included as work on design within the semester subjects (Figure 3.2).

Figure 3.2: The vision of organizing the study process within one semester



The Bachelor's Degree Education Plan (Cycle I) of the Software Engineering Program corresponds to the ISCED level 6, being part of:

- The fundamental field of science, culture and technology: *06 Information and Communication Technologies*
- General field of study: 061 Information and communication technologies
- Vocational training area: 0613 Development of program products and applications.

The study program is oriented towards training engineers, allowing them to obtain the qualification corresponding to Level 6 of the National Qualifications Framework / European Qualifications Framework (NQF/EQF). The key features of the professional training are presented in Table 3.1 and correspond to Level 6 of the National Qualifications Framework (National Qualifications Framework: Higher Education, 2013).

Table 3.1 - Essential characteristics corresponding to the NQF Level 6

Level	**Bachelor's Degree (Cycle 1) - NQF/EQF Level 6**
Length of studies	4 years
ECTS study credits	240 credits
Form of organization	Full-time attendance / non-attendance education
Entry requirements	Baccalaureate diploma, secondary specialized education diploma, higher education diploma
Preconditions	Achieving the pre-university education outcomes
Internships	Mandatory (35 ECTS)
Examination and evaluation rules	Current-formative, final-summative evaluation is mandatory; The current-formative evaluation is performed through seminars, practices, self-evaluation and evaluation of individual and / or team work; The methodology of final-summative evaluation is aimed at evaluating the learning outcomes expressed in skills.
Final evaluation mode	Bachelor's degree examination, thesis defense
Certification	Bachelor's degree diploma
Awarded Title	Licensed engineer
Rights for graduates	To apply to master programs; To apply to continuous training programs; To be hired.
Body responsible for authorizing programs	Ministry of Education, National Agency for Quality Assurance in Education and Research

3.2 SEMESTERS

The distribution of subjects on areas of knowledge, the grouping of these on professional skills and their relationships of interdependence are presented in Annexes 2 and 3. The following is a grouping of subjects on semesters defining a common theme.

The project evaluation includes: current evaluation and final evaluation. The current evaluation, being formative and providing students / team with continuous feedback on design activities or integrated modules, ensures the student evaluation with the grade assigned to the team.

The final evaluation, being a summative evaluation, is performed orally on the basis of the team project and individual discussions / interviews (in the presence of the team or not). The evaluation results of the examination are individual and represent 40% of the final grade.

3.2.1 Semester 1

The theme of the introductory semester is Learning based on problems of science, technology and society.

The content areas covered by the subjects taught in this semester are: Exact and Applied Sciences - 10 ECTS, General and Socio-Humanistic Fields - 9 ECTS, Programming - 5 ECTS, Software Development - 4 ECTS, Information Management - 1 ECTS, Architectures, Platforms and Technologies - 1 ECTS.

For the semester project students get 10 ECTS and it is carried out within the *Conceptual Design of an IT Application* module. Course units related to semester design are *Computer Programming* and *Personal and Professional Development / Computer Science and Society*.

Code	**Course unit / module title**	**Total hours**			**Number of hours by type of activity**			**Evaluation form**	**Nr. credits**
		Total	Direct contact	Individual study	Course	Internships	Design		
G.01.O.013	Conceptual Design of an IT Application	300	150	150			150	PA	10
F.01.O.001	Mathematics	150	75	75	45	30		E	5
F.01.O.002	Computer programming	150	75	75	30	15	30	E	5
F.01.O.003	Special Mathematics 1	150	75	75	30	45		E	5
U.01.A.021 *U.01.A.022*	Personal and professional development Computer science and society	150	75	75	30	30	15	E	5
	Total semester 1:	900	450	450	135	120	195	4E, 1PA	30
					450				

3.2.2 Semester 2

The theme of the semester is *The Engineering and Scientific Basics of the Calculation*. The content areas covered by the subjects taught in this semester are: Exact and Applied Sciences - 15 ECTS, Programming - 6 ECTS, Architectures, Platforms and Technologies - 5 ECTS, General and Socio-Humanistic Fields - 4 ECTS.

For the semester project students get 10 ECTS and it is realized within the *Equivalent models* module. Course units related to semester design are *Applied Sciences, Special Mathematics 2* and *Data Structures and Algorithms*.

Code	Course unit / module title	Total hours			Number of hours by type of activity			Evaluation form	Nr. credits
		Total	Direct contact	Individual study	Course	Internships	Design		
F.02.O.004	Equivalent models	300	150	150			150	PA	10
F.02.O.005	Applied Sciences	150	75	75	30	15	30	E	5
F.02.O.006	Special Mathematics 2	150	75	75	30	15	30	E	5
F.02.O.007	Computer architecture	150	75	75	30	45		E	5
F.02.O.008	Data Structures and Algorithms	150	75	75	30	30	15	E	5
	Total semester 2:	900	450	450	120	105	225	4E, 1PA	30
					450				

3.2.3 Semester 3

The theme of the semester is *Application Development Fundamentals*. The content areas covered by the subjects taught in this semester are: Programming - 13 ECTS, Software Development - 4 ECTS, Networks and Data Communications - 3 ECTS, Architectures, Platforms and Technologies - 2 ECTS, Exact and Applied Sciences - 2 ECTS, Information Management - 3 ECTS, General and Socio-Humanist Areas - 3 ECTS.

The semester project is awarded 10 ECTS and is realized within the framework of the *Application Development Fundamentals* module. Course units related to semester design are *Object-Oriented Programming, Computer Networks* and *Databases.*

Code	Course unit / module title	Total hours			Number of hours by type of activity			Evaluation form	Nr. credits
		Total	Direct contact	Individual study	Course	Internships	Design		
S.03.O.027	Application Development Fundamentals	300	150	150			150	PA	10
S.03.O.028	Object-Oriented Programming	150	75	75	30	15	30	E	5
S.03.O.029	Computer networks	150	75	75	30	45		E	5
S.03.O.030	Databases	150	75	75	30	15	30	E	5
S.03.A.039 *S.03.A.040*	Data Analysis and Visualization *Computer Graphics*	150	75	75	30	30	15	E	5
	Total semester 3:	900	450	450	120	105	225	4E, 1PA	30
					450				

3.2.4 Semester 4

The theme of the semester is *Formal Languages and Compilers*. The content areas covered by the subjects taught in this semester are: Software Development - 1 ECTS, Programming - 18 ECTS, Architectures, Platforms and Technologies - 8 ECTS, General and Socio-Humanistic Fields - 3 ECTS.

The semester project is awarded 10 ECTS and is realized within the *Developing Domain-Specific Languages* module. Course units related to semester design are *Formal Languages and Compiler Design, Calculability and Complexity* and *Multimedia Technologies / Simulation and Modeling Techniques*.

Code	**Course unit / module title**	**Total hours**			**Number of hours by type of activity**			**Evaluation form**	**Nr. credits**
		Total	Direct contact	Individual study	Course	Internships	Design		
F.04.O.009	Developing domain-specific languages	300	150	150			150	PA	10
F.04.O.010	Formal languages and compiler design	150	75	75	30	15	30	E	5
F.04.O.011	Calculability and complexity	150	75	75	30	15	30	E	5
S.04.O.031	Operating systems: internal mechanisms and design principles	150	75	75	30	45		E	5
S.04.A.041 *S.04.A.042*	Multimedia Technologies Simulation and modeling techniques	150	75	75	30	30	15	E	5
	Total semester 4:	900	450	450	120	105	225	4E, 1PA	30
					450				

Internships (performed at the student's choice based on the Application Development Fundamentals and Development of Domain-Specific Languages).

3.2.5 Semester 5

The theme of the semester is *Networks and Security*. The content areas covered by the subjects taught in this semester are: Programming - 10 ECTS, Software Development - 6 ECTS, Exact and Applied Sciences - 1 ECTS, Information Security - 4 ECTS, Software Quality - 3 ECTS, General and Socio-Humanistic 6 ECTS.

The semester project is awarded 10 ECTS and is realized within the module *Secure Application Developmen*. Course units related to semester design are *Network Programming, Cryptography and Security* and *Software Design Techniques and Mechanisms / Program Products Verification and Validation.*

Code	Course unit / module title	Total hours			Number of hours by type of activity			Evaluation form	Nr. credits
		Total	Direct contact	Individual study	Course	Internships	Design		
S.05.O.032	Secure application development	300	150	150			150	PA	10
S.05.O.033	Network programming	150	75	75	30	15	30	E	5
S.05.O.034	Cryptography and security	150	75	75	30	15	30	E	5
G.05.O.020	Ethics, communication and law	150	75	75	45	30		E	5
S.05.A.043 *S.05.A.044*	Software design techniques and Mechanisms *Program products verification and validation*	150	75	75	30	30	15	E	5
	Total semester 5:	900	450	450	135	90	225	4E, 1PA	30
					450				

3.2.6 Semester 6

The theme of the semester is *The Internet of Things (IoT).* The content areas covered by the subjects taught in this semester are: Programming - 13 ECTS, Networks and Data Communications - 1 ECTS, Architectures, Platforms and Technologies - 8 ECTS, Exact and Applied Sciences - 5 ECTS, General and Socio-Humanist Areas - 3 ECTS.

The semester project is awarded 10 ECTS and is realized within the module *IoT Projects*. All course units of the semester are related to the semester design, accumulating 255 hours of guided study by the supervisor.

Code	Course unit / module title	Total hours			Number of hours by type of activity			Evaluation form	Nr. credits
		Total	Direct contact	Individual study	Course	Internships	Design		
S.06.O.035	IoT projects	300	150	150			150	PA	10
S.06.O.036	Embedded systems	150	75	75	30	15	30	E	5
F.06.O.012	Signal processing	150	75	75	30	30	15	E	5
S.06.A.045 *S.06.A.046*	Man-computer interaction *Real time programming*	150	75	75	30	15	30	E	5
S.06.A.047 *S.06.A.048*	Mobile application programming *Web programming*	150	75	75	30	15	30	E	5
	Total semester 6:	900	450	450	120	75	255	4E, 1PA	30
					450				

Technology Internship (performed at the student's choice based on the Secure Application Development modules, semester 5, or on the IoT Projects, semester 6).

3.2.7 Semester 7

The theme of the semester is *Information Systems.* The content areas covered by the subjects taught in this semester are: Programming - 8 ECTS, Software Development - 4 ECTS, Software Quality - 5 ECTS, Exact and Applied Sciences - 4 ECTS, Information Management - 2 ECTS, General and Socio-Humanistic Fields / Areas - 7 ECTS.

The semester project is awarded 10 ECTS and is realized within the *Information Systems Design* module. All course units of the semester are related to the semester design, accumulating 225 hours of guided study by the supervisor.

Code	Course unit / module title	Total hours			Number of hours by type of activity			Evaluation form	Nr. credits
		Total	Direct contact	Individual study	Course	Internships	Design		
S.07.O.037	Information systems design	300	150	150			150	PA	10
S.07.O.038	Programming distributed applications	150	75	75	30	15	30	E	5
U.07.A.023 *U.07.A.024*	Software project management Enterprise management	150	75	75	30	30	15	E	5
U.07.A.025 *U.07.A.026*	Electronic marketing Digital entrepreneurship	150	75	75	30	30	15	E	5
S.07.A.049 *S.07.A.050*	Software quality Software requirements analysis and specification	150	75	75	30	30	15	E	5
	Total semester 7:	900	450	450	120	105	225	4E, 1PA	30
					450				

3.2.8 Semester 8

The theme of the semester is *Bachelor's Project.* The content areas covered by the subjects taught in this semester are: Programming - 12 ECTS, Software Development - 12 ECTS, Information Management - 2 ECTS, General and Socio-Humanistic Fields - 4 ECTS.

The semester is dedicated to the Bachelor's Project, which is awarded 15 ECTS. The project shall be publicly defended, in front of a committee including at least one external reviewer, who shall also be the Chairman of the License Committee.

Code	**Course unit / module title**	**Total hours**			**Number of hours by type of activity**			**Evaluation form**	**Nr. credits**
		Total	Direct contact	Individual study	Course	Internships	Design		
S.08.A.051 *S.08.A.052*	Artificial intelligence fundamentals *Unrelated databases*	150	75	75	30	45		E	5
S.08.A.053 *S.08.A.054*	Game development fundamentals *Mixed reality technologies*	150	75	75	30	45		E	5
S.08.O.055	Bachelor's Internship and design	450		450				E	15
S.08.O.056	Theoretical synthesis test: algorithms, programming and databases	120		120				E	4
S.08.O.057	Defending the Bachelor's Project	30		30				E	1
	Total semester 8:	900	150	750	60	90		5E	30

4 ROADMAP

4.1 INTRODUCTION

The Roadmap is a consolidated list of measures, commitments and timelines for implementing actions to ensure the sustainability of the Software Engineering Pilot Program, which implements the problem-based learning. It aims to establish an institutional foundation in order to overcome certain barriers or certain existing threats and ensure the sustainability of the study program.

The objectives are to:

- Extend the PBL teaching methodology to other study programs at the 1st and 2nd cycle of studies.
- Internationalize the Software Engineering Study Program.
- Enhance interactivity with the private environment.
- Provide methodological support for teacher training in PBL.
- Upgrade the teaching infrastructure based on the PBL methodology.
- Revise the Software Engineering Education Plan with performance indicators.
- Promote the Software Engineering Study Program.
- Continuously improve the educational process with performance indicators.
- Integrate research at the teaching and student level into the study process.

Based on the above-mentioned objectives, the Roadmap presented in Annex 4 lists the key activities in the sustainability assurance process. Activities can be grouped in fourt periods:

***Period** 1: Prepare a new study program for implementing the PBL teaching methodology.*

***Period** 2: Internationalize the Software Engineering Study Program.*

***Period** 3: Develop and continuously adjust the Software Engineering Study Program to the needs of the economic environment with the integration of research elements at the teaching and student level.*

***Period 4:** Provide methodological support for teacher training in PBL.*

4.2 PERIOD 1

The first period includes activities related to the extension of the PBL teaching methodology to other study programs in the 1st and 2nd cycles of studies. Thus, over the following two years (2019-2021), it is planned to prepare the legal framework, physical environment, and teachers for the implementation of the PBL methodology within another study program within the Software and Automatic Engineering Department. As a premise, the study program "Automatic and Informatics" was identified.

Automation and Informatics is a bachelor's degree program that is part of the System Engineering field, an interdisciplinary field of science and technology aimed at developing and implementing in a systemic conception the equipment, control, communications and information systems intended for managing processes in different sectors of activity: scientific, technical, industrial and economic.

Over this period, the following results are expected to be achieved:

- Education program adjusted to the needs of the economic environment with the integration of PBL elements.
- Education Plan adopted in accordance with the Framework - Plan so that it is based on a linear progress determined by relations at the semester level rather than at the level of subjects.
- Teachers trained in the PBL methodology, who will be involved in the education process.
- Upgrade the teaching infrastructure based on the PBL methodology.
- Curriculum by subjects, subject cards, guides, case studies, evaluations, etc. developed for the adopted program.

4.3 PERIOD 2

This period will include activities related to the internationalization of the Software Engineering Study Program and will encompass activities related to the establishment of international collaboration relations with the academic environment and the international organizations working in the Republic of Moldova, conclusion of international projects on mobility for both teachers and students, introduction of changes to the Education Plan to ensure mobility with the equivalency of accumulated credits, stimulation of students and teachers to improve language skills.

Over this period, the following results are expected to be achieved:

- Various cooperative and partnership activities carried out based on international projects with foreign universities.
- Interest of students and academic staff for external mobility increased.
- Number of external mobility for students and academic staff increased.
- Number of students with strong English language skills increased.
- Optional English language courses for teachers carried out.
- International compatibility of the study program increased and international aspects and study periods into the study process integrated.
- Study program revised from the point of view of international cooperation (student mobility, double diplomas, etc.).
- International teachers attracted to teach courses.

4.4 PERIOD 3

This period includes activities related to the development and continuous adjustment of the Software Engineering Study Program to the needs of the economic environment with the integration of the research elements at the teaching and student level. This is a continuation of the Software Engineering Study Program and course units' contents adaptation to the needs of students and society achieved through continuous communication (round tables, surveys, etc.) with the institutions interested in the faculty graduates and through the harmonization with other programs of similar prestigious institutions abroad, a factor that can facilitate student mobility.

Integrating the research process into the process of study involves strengthening the scientific research directions made in interdisciplinary (inter-departmental / university) groups with attracting students into scientific activities - by strengthening the students' scientific-practical groups and their involvement in the research activity.

Over this period, the following results are expected to be achieved:

- Interaction with the private environment in order to conduct semestrial projects is enhanced.
- Curriculum on subjects, subject cards, guides, case studies, evaluations, etc. is reviewed, so that students develop their transversal skills needed for a successful employment.
- Teaching infrastructure based on the PBL methodology is upgraded.
- Software Engineering Education Plan is revised and includes performance indicators.
- Software Engineering Study Program is promoted.
- Educational process is continuously improved with the continuous adjustment to the needs of the economic environment.
- Scientific research directions per faculty are strengthened.
- Interdisciplinarity per semester and study years is ensured.

4.5 PERIOD 4

This period implies activities related to the implementation of a university-level strategy for the PBL implementation in other study programs from other faculties. This requires the introduction, at the initial stage, of a compulsory continuous training course on the PBL methodology for teachers.

Over this period, the following results are expected to be achieved:

- Methodological support for teacher training in PBL is developed.
- Compulsory course of continuous teacher training in PBL methodology is initiated.
- Initial course in PBL for students in the first year of study is introduced.
- Methodical guide on project development for students is developed.
- Interdisciplinary year / bachelor's / master's degree projects among the CIM faculty students, as well as from other faculties, are initiated.
- Scientific research directions in groups are strengthened

5 Action Plan

5.1 Introduction

The action plan contains the activities undertaken to ensure sustainability and, as mentioned in Chapter 4, activities are grouped in four periods:

***Period** 1: Prepare a new study program for implementing the PBL teaching methodology.*

***Period** 2: Internationalize the Software Engineering Study Program.*

***Period** 3: Develop and continuously adjust the Software Engineering Study Program to the needs of the economic environment with the integration of research elements at the teaching and student level.*

***Period 4:** Provide methodological support for teacher training in PBL.*

The Action Plan is presented in Annex 5.

5.2 Period 1

In order to implement the PBL teaching methodology within the Automatics and Informatics Study Program of the Faculty of Computers, Informatics and Microelectronics, a series of activities, listed in the table below, are to be carried out:

Actions	**Required resources**
Revise the Education Plan for the training of specialists in Automation and Informatics according to the TUM Regulation on the organization of studies based on the National Education Credit System, taking into account the Regulation on the organization of higher education based on the National Education Credit System, so that the program is linked to the national and international standards of training specialists in the field and is in line with the Framework-Plan. *It is expected that starting with the 2nd year, in each semester, students will have a special subject dedicated to design/projects.*	Support at the department, faculty, University level
Approve the Education Plan within the Software Engineering and and Automation Department; the Faculty of Computers, Informatics and Microelectronics and the TUM Senate.	Support at department, faculty, university level.

Identify the teachers who will be involved in the teaching process under the new study program and train them on the PBL teaching methodology.	Human resourses.
Upgrade the teaching infrastructure based on the PBL methodology.	Financial resources. Support at department, faculty, university level.
Revise curriculum by subjects (analytical programs), subject cards, guides, case studies, evaluations etc. (for the first year of study).	Human resourses.

5.3 Period 2

This period will cover activities related to the internationalization of the Software Engineering Study Program and will include the activities listed in the table below:

Actions	**Required resources**
Carry out various cooperative and partnership activities based on international projects with foreign universities.	Support at University level. Developed infrastructure.
Increase the interest of students and academic staff for external mobility.	Support at University level. Human resourses.
Increase the number of external mobility for students and academic staff.	Support at University level. Human resourses. Foreign partners. Collaborative projects. Financial resources.
Increase the number of students with strong English language skills.	Human resourses. Financial resources.
Deliver optional English language courses for teachers.	Human resourses. Financial resources.
Improve curricular content and teaching skills for all areas of study, increase international compatibility of study programs and integrate international aspects and study periods into the study process.	Human resourses. Support at department, faculty, university level.
Re-design study program on the basis of international cooperation opportunities (student mobility, double diplomas, etc.).	Human resourses. Support at department, faculty, university level. Foreign partners.

Attract international teachers to deliver courses.	Foreign partners. Human resourses. Financial resources.
Launch several international projects providing mobility for both teachers and students.	Support at University level. Human resourses. Foreign partners. Collaborative projects. Financial resources.
Introduce changes to the Education Plan to ensure that mobility is performed with the equivalency of accumulated credits.	Human resourses. Support at department, faculty, University level. Foreign partners.
Enhance interaction with the private environment, attract international companies.	Support at department, faculty, University level. Foreign partners.
Promote the Software Engineering Study Program.	Support at department, faculty, University level.

5.4 PERIOD 3

This period includes activities related to the development and continuous adaptation of the Software Engineering Study Program to the needs of the economic environment with the integration of research elements and covers the activities listed in the table below:

Actions	**Required resources**
Develop research-oriented scientific partnerships with universities, institutions and companies from Moldova and abroad (Europe and the whole world).	Support at University level. Human resourses. Foreign partners. Collaborative projects. Financial resources.
Revise Curriculum by subjects, subject cards, guides, case studies, evaluations, etc. so that students develop the transversal skills needed for a successful employment, taking into account performance indicators.	Support at department, faculty, University level. Human resourses.
Upgrade the teaching infrastructure based on the PBL methodology.	Support at department, faculty, University level. Financial resources.
Participate in international scientific events.	Support at department, faculty, University level. Financial resources. Human resourses.

Attract internationally recognized experts to participate in the events organized within the Software Engineering Study Program.	Support at department, faculty, University level. Human resourses. Financial resources.
Initiate or revive cooperation agreements with international organizations operating in the Republic of Moldova.	Support at University level. Human resourses. Foreign partners. Collaborative projects. Financial resources.
Organize international events with the participation of international partners from international programs / projects / organizations.	Human resourses. Support at department, faculty, University level. Collaborative projects. Financial resources.
Identify companies undertaking the responsibility to support knowledge transfer at the level of content, teachers and internships.	Human resourses. Support at department, faculty, University level. Private Partners.
Integrate research at the teaching and student level into the study process.	Support at University level. Human resourses. Research projects. Financial resources.
Initiate interdisciplinary year / bachelor's / master's degree projects, among the students of the *study programs of the CIM faculty, but also from* *other faculties, by encouraging teamwork.*	Human resourses. Support at department, faculty, University level.
Strengthen scientific research directions carried out in interdisciplinary groups (inter-departmental / university).	Support at University level. Human resourses. Research projects. Financial resources.
Attract students to scientific activities - by strengthening students' scientific and practical groups and their involvement in the research activity *of the teaching staff.*	Support at University level. Human resourses. Research projects. Financial resources.
Promote performance in educational and research processes.	Support at department, faculty, University level.

5.5 PERIOD 4

This period includes activities related to the promotion of the PBL methodology at the university level with a view to implement it in other study programs:

Actions	**Required resources**
Continuous assistance in professional problem solving.	Support at department, faculty, University level.
Provide methodological support for teacher training in PBL.	Support at department, faculty, University level. Human resourses.
Initiate compulsory course of continuous teacher training in PBL methodology.	Human resourses. Support at department, faculty, University level. Financial resources.

6 STRATEGIC RECOMMENDATIONS AT UNIVERSITY LEVEL

6.1 INTRODUCTION

The Technical University of Moldova assumes the general mission of scientific research and permanent promotion of the student-centered educational process. To achieve its mission, the following strategic priorities have been established:

1 Quality and academic excellence (research and education).
2 Development and motivation of human resources.
3 Diversification of educational offer, teaching and learning methods and funding sources.
4 Responsible and transparent university management.
5 Deeper integration with industry / business.

The launch of the PBLMD Project has highlighted some *important issues* related to the implementation of the PBL teaching methodology at the Technical University of Moldova, which are listed below in a SWOT analysis perspective:

Issues	**Notes**
Strengths (internal source)	
TUM is an institution with traditions	Authority transfer and well-established processes
Recognized for the good training provided	USAID Survey
Internationalization actions	A plan developed in the spirit of the international ACM standard Programs delivered in English
Teachers with good professional experience	Teachers are also working for IT companies
High number of students	The relatively higher number of students offers possibilities for optimizing the teaching load
Alternatives in education	Interdisciplinarity, teamwork, etc.
Experience of the Anglophone groups	The existence of the Anglophone Student Community, which allows multiple extra-curricular activities
Weaknesses (internal source)	
Few teachers with PhD	It diminishes the academic value of the program evaluation
Few teachers are fluent in English	

Few teachers are involved in research topics	It diminishes from the scientific value of the program evaluation
Uncompetitive salaries paid to IT teachers compared to specialists in the field	The difference between the salary paid to a beginner in the field and a teacher becomes significant
Insufficient technical equipment to cover new directions	The local industry is already actively seeking for IoT, VR, GameDev, etc. specialists
Insufficient use of institutional collaboration relationships with IT associations / companies	Program engagement, technical endowment, etc.
Opportunities (external source)	
Internationalization of the study program	Cooperative agreements Academic mobility through programs, such as Erasmus +
External financing for technical means	The PBLMD project USAID Collaboration - IoT Laboratory Orange Collaboration - Mobile Technologies
Internships / workshops for staff training	
IT career promotion campaigns by associations related to the field	Choose a Career in IT (ATIC)
Required professional field (a sector developing dynamically in Moldova)	Admission Contest Extensive internship / collaboration base
Threats (external source)	
Reduction in the number of high school graduates	Especially among those with a science profile
Migration of students after the start of the academic year	In particular, to Romania
Confusions with related specialties	Information Technologies, Informatics, Computers, Automation
Competition with "accelerated studies" in IT	Continuous training programs.
Deterioration of the social-political situation in the country	Reduction of budget funding Salaries and scholarships paid with delays

6.2 RECOMMENDATIONS: STUDY PROGRAM LEVEL

To ensure the sustainability of the Software Engineering Study Program at the study program level, the following recommendations are proposed:

Recommendations	**Planned measures**
Continuous adaptation of study programs and content of course units to the needs of students and society.	- Apply the best teaching and examination methods based on the university's experience and specificity. - Consolidate subject groups depending on the areas of knowledge and identify skills supervisors to ensure consistency in the flow of studies. - Apply non-formal education methods (workshops, meetings with specialists in the field / alumni). - Develop cross-cutting skills for a successful ICT employee. - Actively involve employers, as well as graduates, in the review of the study program content. - Consult businesses and economic agents on the content of education plans.
Create a system for collecting feedback from students and graduates on the quality of the study program.	Carry out surveys among students on the quality of the study program.
Encourage the use of ICT in the educational process.	Enhance the use of new e-learning technologies. Promote modern teaching-learning and evaluation methods and technologies.
Promote research-based and student-centered learning.	Apply a proactive approach to motivate and support teachers in scientific research.
Integrate research at the teaching and student level into the study process.	Motivate and support interdisciplinary and applied research. Initiate interdisciplinary projects with research elements.

6.3 RECOMMENDATIONS: AT DEPARTMENT AND COLLEGE LEVEL

To ensure sustainability, the following recommendations and measures are proposed at the department and faculty level:

No.	**Recommendations**	**Planned measures**
1	Active involvement in the teacher training activity and increase the number of teaching staff holding scientific degrees and scientific-didactic titles.	-Teacher training in the PBL pedagogical module. - Expand the continuous training partnerships of teachers. - Increase the number of teachers attending English courses organized by the TUM for teachers. - Support and motivate young tenured teachers to develop and improve psycho-pedagogical skills. Internships / workshops for staff training. - Continuing training of the teaching staff. -Organize the ground for a more active involvement of teachers in the research process. - Conduct scientific seminars at the department. -Oraganize didactic seminars and share the PBL teaching experience.
2	Actively involve teachers and students in research activities.	-Strengthen scientific research directions performed at the department in research groups. - Orientate research directions at the department to the priority research themes Horizon 2020. - Attract students to scientific activities.
3	Promote the image of the department and study program.	- Career promotion campaigns. - Internationalization of the study program. - Strengthen academic partnerships. - Promote the image of the department.
4	Develop the technical and material basis for laboratory works and scientific research.	- External financing for technical means. - Improve the teaching and research infrastructure.
5	Expand the basis for internships.	- Sign new collaboration protocols with businesses to ensure students' internships. - Mobility programs for students.

6	Campaigns to promote mobility with the help of Erasmus students and dissemination of their experience.	Provide assistance and counseling to students and teachers about the opportunity to participate in international mobility programs.
7	Improve faculty-employer interaction.	Conclude or revitalize the cooperation agreements with organizations operating in the Republic of Moldova.

6.4 Recommendations: At the level of teachers

At the level of the teaching staff, the following recommendations and measures are proposed:

No.	**Recommendations**	**Planned measures**
1	Active involvement in continuous training activities.	- Continuous training activities in the PBL training module. - Increase the number of teachers attending English courses organized by the TUM for teachers. - Internships / workshops for staff training. - Continuous training courses.
2	Active involvement in the research process.	- Strengthen scientific research directions carried out in interdisciplinary groups. - Actively integrate teaching staff in the research process. - Actively participate in scientific seminars organized within the department, the faculty. - Actively participate in national and international scientific events. - Attract students to the research process. - Submit research project proposals.
3	Active involvement in mobility programs.	- Initiate several international projects providing mobility for teachers. - Provide assistance and counseling to students and teachers about the opportunity to participate in international mobility programs.
4	Initiate interdisciplinary projects on study programs at the CIM faculty, but also in other faculties.	- Encourage teamwork. - Identify subjects for the implementation of interdisciplinary projects. - Carry out methodological seminars at the department to assure interdisciplinarity.

6.5 Recommendations: At the student level

At the student level, the following recommendations and measures are proposed:

No.	**Recommendations**	**Planned measures**
1	Active involvement in scientific activities.	- Create scientific-practical groups of students within the departments. - Organize Students' Conference.
3	Active involvement in mobility programs.	- Initiate several international projects providing mobility for students. - Provide assistance and counseling to students about the opportunity to participate in international mobility programs.
4	Participate in interdisciplinary projects.	- Encourage teamwork. - Identify subjects for the implementation of interdisciplinary projects. - Provide support in implementing interdisciplinary projects.

6.6 Recommendations: At the level of teacher training in *PBL*

According to the University Charter, the teaching and scientific staff are obliged to permanently improve their professional and cultural level, so at the level of the teacher training in PBL, the following recommendations and measures are proposed:

No.	**Recommendations**	**Planned measures**
1	Perform continuous teacher training.	- PBL teacher training module. - Organize internships / workshops for staff training. - Attract teachers to the research process. - Academic mobility.
2	Develop the technical and material basis for laboratory works and scientific research.	- External financing for technical means. - Adapt areas for active learning.
3	Plan PBL-based education in other study programs.	- Identify study programs and provide support in terms of legal framework, infrastructure.
4	Create methodological support platforms in PBL for students.	- Strengthen the working group creating a PBL support platform for students.

6.7 Recommendations: At the society level

At the society level, the following recommendations and measures are proposed:

No.	**Recommendations**	**Planned measures**
1	Ensure correlation between education and economic development.	- Stimulate employers to invest more actively in the future workforce by creating partnerships between businesses and educational institutions. - Adjust the education system to labor market requirements by involving all stakeholders, especially the state and employers. - Increase investments in education. Streamline them through the financing of the priority sectors, with the gradual increase and in line with the economic growth.

6.8 Recommendations: At the administration and management level

At the level of administration and management, the following recommendations and measures are proposed:

No.	**Recommendations**	**Planned measures**
1	Develop the technical and material basis for laboratory works and scientific research.	- External financing for technical means. - Adapted areas for active learning.
2	Plan PBL-based education in other study programs.	- Identify study programs and provide support in terms of legal framework, infrastructure.
3	Extend institutional collaboration relationships with IT associations / companies.	- Conclude new collaboration agreements with IT associations / companies.
4	Integrate the scientific research element into the study process.	- Motivate and support teachers in scientific research. - Support teachers' participation in scientific events. - Develop mechanisms to motivate and support interdisciplinary and applied research. - Provide laboratories and research centers with equipment.
5	Internationalise study programs.	- Update the regulations for the deployment of Erasmus + mobility programs for both students and staff.

		- Initiate several international projects providing mobility for teachers. - Improve the flexibility of recognition procedures for academic achievements and skills acquired in mobility. - Improve marketing policy in promoting TUM in order to attract foreign students. - Further update the TUM web page with information for potential foreign students, including regular updates with legislative information / changes. - Identify the main recruitment fairs in the geographic areas of interest where the university could participate and encourage faculties to participate in fairs related to their specific areas of interest.
6	Reupdate the library content	- Optimize the structure and content of library collections according to the needs of the study process and research activities.
7	Develop and implement support and motivation mechanisms for teacher development and professional development.	- Organize foreign language courses. - Provide institutional and financial support to participation in various training courses and internships. - Increase the language and practice training offer for English, French or German for the academic staff.

7 REFERENCES

ACM and the IEEE-Computer Society. (2015). *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.* Taken from the Association for Computing Machinery: https://www.acm.org/binaries/content/assets/education/se2014.pdf

Normative acts / Framework Plan for Higher Education. (2015). Taken from the Technical University of Moldova: http://dmc.utm.md/utilsIRDownloads/acteNL/externe/PlanCadruStudiiSuperioare.pdf

Balan, M., Călin, R., and Ciorbă, D. (2016). Bypassing curricula constraints by means of ICT. International Conference When students take the lead: Enhancing Quality and Relevance of Higher Education through Innovation in Student-Centered-Problem-Based Active Learning. Chisinau

National Qualifications Framework: Higher Education. (2013). Taken from the Ministry of Education, Culture and Research: http://mecc.gov.md/sites/default/files/cnc_52_54_55_58_84-inginerie.pdf

European Commission. (2010). *A Digital Agenda for Europe.* Taken from http://eur-lex.europa.eu/legal-content/RO/TXT/PDF/?uri=CELEX:52010DC0245&from=RO

Government Decision. (2011). Strategic Program for Governance Technological Modernization (e-Transformation).

Illeris, K. (2007). How we learn: Learning and Non-learning in School and Beyond. London: Routledge.

Karami, M., Karami, Z., & Attaran, M. (2013). Integrating Problem-Based Learning with ICT for Developing Trainee Teachers' Content Knowledge and Teaching Skill. *International Journal of Education and Development using Information and Communication Technology (IJEDICT), 9*, pg. 36-49.

O'Sullivan, J., & Bercu, I. (2016). *Updating the IT Skills Gap.* Moldova ICT Summit 2016.

Prince, M. (2004). Does Active Learning Work? A Review of the Research. *Journal Engineering Education, 93 (3)*, pg. 223-231.

Technical University of Moldova, Department of Software Engineering and Automation. (2017). *Educational Plans, 2016 Edition.* Taken from the Technical University of Moldova: http://utm.md/studii/planuri/2016/fcim/Plan%20ISW.pdf

Zapater, M., Malagon, P., Goyeneche, J.-M. d., & Moya, J. M. (2013). Project-Based Learning and Agile Methodologies in Electronic Courses: Effect of Student Population and Open Issues. *Electronics Journal, 17*(2), 82-88.

Annex 1: Study Plan implemented from 1 September 2017

MINISTERUL EDUCAȚIEI
AL REPUBLICII MOLDOVA
UNIVERSITATEA TEHNICĂ
A MOLDOVEI

MINISTRY OF EDUCATION
OF THE REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY
OF MOLDOVA

MD-2004, CHIȘINĂU, BD. ȘTEFAN CEL MARE ȘI SFÂNT, 168, TEL: 022 23-78-61 | FAX: 022 23-54-41, www.utm.md

APPROVED
at the Senate Meeting of
Technical University of Moldova
Minutes No. 4
of 27 December 2016
Chairperson of Senate
Rector, PhD _____________ (stamp)
Viorel BOSTAN

COORDINATED
Ministry of Education of the
Republic of Moldova
24 July 2017
Registration No. ISL-01-18130
(stamp)

CURRICULUM
for
Cycle I, Licentiate/Bachelor's Degree (Level 6 according to ISCED)

General field of study:	*061 Information and Communication Technologies*
Field of professional study:	0613 Software and Application Development
Specialty/ Major:	0613.1 Software Engineering
Total number of credits:	240
Degree obtained upon the completion of studies:	Licentiate Engineer/Bachelor's Degree
Certification:	Licentiate Diploma
Basis for Admission:	High school diploma or an equivalent education document; higher education diploma
Language of instruction:	Romanian, Russian, English
Form of education:	Full-time attendance

1. ACADEMIC CALENDAR

Academic year	**Teaching activities**		**Examination period**		**Intern-ships**	***Vacations***		
	Semester I	**Semester II**	**Semester I**	**Semester II**		**winter**	**spring**	**summer**
I	15 weeks	15 weeks	4 weeks	4 weeks	-	*2* weeks	Vacation on Easter – one week (according to the Christian calendar)	10 weeks
II	15 weeks	15 weeks	4 weeks	4 weeks	15 weeks	2 weeks		6 weeks
III	15 weeks	15 weeks	4 weeks	4 weeks	15 weeks	2 weeks		6 weeks
IV	15 weeks	7 weeks	4 weeks	2 weeks	10 weeks	2 weeks		9 weeks

2. Curriculum by semesters/academic years

Year I

Semester I. *Problem Based Learning in Science, Technology and Society*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
G.01.O.013	Conceptual Design of an IT Application	300	150	150			150		PA	10
F.01.O.001	Math	150	75	75	45	30			E	5
F.01.O.002	Computer Programming	150	75	75	30	15	30		E	5
F.01.O.003	Special Math 1	150	75	75	30	45			E	5
U.01.A.021 *U.01.A.022*	Personal and Professional Development *Computer Science and Society*	150	75	75	30	30	15		E	5
GM.O.014	*Foreign Language 1* **	*90*	*45*	*45*		*45*			*E* *	*3*
G.01.O.015	*Romanian (for non-speakers of Romanian) 1* *	*60*	*30*	*30*		*30*			*T* *	*2*
G.01.O.016	*Physical training 1* *	*60*	*30*	*30*		*30*			*T* *	
Total per Semester I:		**900**	**450**	**450**	**135**	**120**	**195**	**0**	**4E, 1PA**	**30**
					450					

Semester II. *Engineering and Scientific Bases for Computing*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
F.02.O.004	Equivalent Models	300	150	150			150		PA	10
F.02.O.005	Applied Science	150	75	75	30	15	30		E	5
F.02.O.006	Special Math 2	150	75	75	30	15	30		E	5
F.02.O.007	Computer Architecture	150	75	75	30	45			E	5
F.02.O.008	Data Structures and Algorithms	150	75	75	30	30	15		E	5
G.02.O.017	*Foreign Language 2* *	*90*	*45*	*45*		*45*			*E* *	*3*
G.02.O.018	*Romanian (for non-speakers of Romanian) 2* *	*60*	*30*	*30*		*30*			*T* *	*2*
G.02.O.019	*Physical training 2* *	*60*	*30*	*30*		*30*			*Ţ* *	
	Total per Semester II:	900	450	450	120	105	225	0	4E, 1PA	30
					450					
Total per Year I:		**1800**	**900**	**900**	**255**	**22 5**	**420**	**0**	**8E, 2PA**	**60**

**** - This is not included in the total sum of evaluation forms (the course units are provided in extracurricular regime, and those 240 credits per program are supplement with additional credits, while the course unit "Physical training" is not quantified with credits).***

T* - Testing assessed as "passed/failed".

Year II

Semester III. *Application Development Bases*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
S.03.O.027	Application Development Basics	300	150	150			150		PA	10
S.03.O.028	Object Oriented Programming	150	75	75	30	15	30		E	5
S.03.O.029	Computer Networks	150	75	75	30	45			E	5
S.03.O.030	Databases	150	75	75	30	15	30		E	5
S.03.A.039 *S.03.A.040*	Data Analysis and View; *Computer Graphics*	150	75	75	30	30	15		E	5
Total per Semester III:		**900**	**450**	**450**	**120**	**105**	**225**	**0**	**4E, 1PA**	**30**
					450					

Semester IV. *Formal Languages and Compilers*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
F.04.O.009	Developing Industry Specific Languages	300	150	150			150		PA	10
F.04.O.010	Formal Languages and Compiler Design	150	75	75	30	15	30		E	5
F.04.O.011	Calculability and Complexity	150	75	75	30	15	30		E	5
S.04.O.031	Operating Systems: Internal Mechanisms and Design Principles	150	75	75	30	45			E	5
S.04.A.041 *S.04.A.042*	Multimedia Technologies *Simulation and Modelling Techniques*	150	75	75	30	30	15		E	5
Total per Semester IV:		**900**	**450**	**450**	**120**	**105**	**225**	**0**	**4E, 1PA**	**30**
					450					
Internship in Production (It shall be carried out at the Student's choice on the basis of Modules *Application Development Basics and Developing Industry Specific Languages)*										
Total per Year II:		**1800**	**900**	**900**	**240**	**210**	**450**	**0**	**8E, 2PA**	**60**

Year III

Semester V. *Network and Security*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
S.05.O.032	Developing Secure Applications	300	150	150			150		PA	10
S.05.O.033	Network Programming	150	75	75	30	15	30		E	5
S.05.O.034	Cryptography and Security	150	75	75	30	15	30		E	5
G.05.O.020	Ethics, Communication and Law	150	75	75	45	30			E	5
S.05.A.043 *S.05.A.044*	Software Design Techniques and Mechanisms *Software Verification and Validation*	150	75	75	30	30	15		E	5
Total per Semester V:		**900**	**450**	**450**	**135**	**90**	**225**	**0**	**4E, 1PA**	**30**
					450					

Semester VI. *Internet of Things (IoT)*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
S.06.O.035	IoT Projects	300	150	150			150		PA	10
S.06.O.036	Embedded Systems	150	75	75	30	15	30		E	5
F.06.O.012	Signal Processing	150	75	75	30	30	15		E	5
S.06.A.045 *S.06.A.046*	Human-Computer Interaction *Real Time Programming*	150	75	75	30	15	30		E	5
S.06.A.047 *S.06.A.048*	Mobile Application Development *WEB Programming*	150	75	75	30	15	30		E	5
Total per Semester VI:		**900**	**450**	**450**	**120**	**75**	**255**	**0**	**4E, 1PA**	**30**
					450					
Technological Internship (It shall be carried out at the Student's choice on the basis of Modules *Developing Secure Applications and IoT Projects)*										
Total per Year III:		**1800**	**900**	**900**	**255**	**165**	**480**	**0**	**8E, 2PA**	**60**

Year IV

Semester VII. *Information Systems*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
S.07.O.037	Information System Design	300	150	150			150		PA	10
S.07.O.038	Programming of Distributed Applications	150	75	75	30	15	30		E	5
U.07.A.023 *U.07.A.024*	Software Project Management *Enterprise Management*	150	75	75	30	30	15		E	5
U.07.A.025 *U.07.A.026*	Electronic Marketing *Digital Entrepreneurship*	150	75	75	30	30	15		E	5
S.07.A.049 *S.07.A.050*	Software Quality *Analysis and Specification of Software Requirements*	150	75	75	30	30	15		E	5
Total per Semester VII:		**900**	**450**	**450**	**120**	**105**	**225**	**0**	**4E, 1PA**	**30**
					450					

Semester VIII. *Licentiate Project*

Code	Name of the Course Unit/Module	Total number of hours			Number of hours by types of activity				Type of final assessment	Number of credits e
		total	direct instruction	individual work	C	S/P	Pr	per week		
S.Q8.A.051 *S.V8.A.052*	Foundations of Artificial Intelligence *Non-relational Databases*	150	75	75	30	45			E	5
S.08.A.053 *S.08.A.054*	Foundations for Game Development *Technologies of Mixed Reality*	150	75	75	30	45			E	5
S.08.O.055	Licentiate Internship and Design	450		450					E	15
S.08.O.056	Summary Theory Exam: *Algorithms, Programming and Databases*	120		120					E	4
S.08.O.057	Defending the Licentiate Project	30		30					E	1
Total per Semester VIII:		**900**	**150**	**750**	**60**	**90**	**0**	**0**	**5E**	**30**
Total per Year IV:		**1800**	**600**	**1200**	**180**	**195**	**225**	**0**	**9E, 1PA**	**60**
Total for the Study Programme:		**7200**	**3300**	**3900**	**930**	**795**	**1575**	**0**	**33E, 7PA**	**240**

3. Internships

Internships*		Semester	Duration, number of weeks/hours	Period	Number of credits
1	Internship in Production	3/4	15	September - December/ February - May	10
2	Technological Internship	5/6	15		10
3	Licentiate Internship and Design	8	10	March - May	15
	Total:		24/990		35

**** Internships shall be carried out on the basis of a Semestrial Design Module.***

4. Free choice (optional) course units

Criterion #	Name	Year	Sem.	Number of hours by types of activity per week			Type of final assessment	Number of credits
				C	S/P	L		
1	Introduction into the Specialty	l	2	30			E	2
2	Psychology of Invention	2	4	30			E	2
3	Cognitive Philosophy	2	4	30			E	2
4	Graphical Representation of Data	3	5	30		30	E	4
5	Programming in Virtual Reality	3	5	30		30	E	4
6	Techniques for Reverse Engineering	3	6	30		30	E	4
7	Managerial Psychology	3	6	30			E	2
8	E-Governance	4	7	30			E	2
9	Romanian (for non-speakers of Romanian) 3	2	3		30		E	2
10	Romanian (for non-speakers of Romanian) 4	2	4		30		E	2
11	Romanian (for non-speakers of Romanian) 5	3	5		30		E	2
12	Romanian (for non-speakers of Romanian) 6	3	6		30		E	2
13	Foreign Language 3	2	3		30		E	2
14	Foreign Language 4	2	4		30		E	2
15	Foreign Language 5	3	5		30		E	2
16	Foreign Language 6	3	6		30		E	2
17	Foreign Language 7	4	7		30		E	2
18	Physical training 3	2	3		30		T*	
19	Physical training 4	2	4		30		T*	
20	Physical training 5	3	5		30		T*	
21	Physical training 6	3	6		30		T*	
22	Physical training 7	4	7		30		T*	

T* - Testing assessed as “passed/failed”.

5. Licentiate Exam

Criterion #	Name of activity	Period	Number of credits
1	Summary Theory Exam: *Algorithms, Programming and Databases*	29.02. - 12.03	4
2	Defending the Licentiate Project	06.06. - 25.06	1
Total:			**5**

Approved at the TUM Senate meeting, Minutes No. 4 of 27.12.2016.

Ion BALMUS	**Dean of CIM Faculty, Dr., Associate Professor**
Dumitru CIORBA	**Head of Department of Software Engineering and Automatics, Dr., Associate Professor**

EXPLANATORY NOTE
to the Curriculum for Licentiate Studies (Cycle I)

Fundamental Area of Science, Culture and Technique: *06 Information and Communication Technologies*

General Area of Study: *061 Information and Communication Technologies*

Area of Professional Education: *0613 Software and Application Development*

Programme of Study: *0613.3 Software Engineering*

Description of Software Engineering Specialty Profile

Software Engineering (SE), along with *Information Technology,* falls within the *Computer Science area,* which pursues the goal to address certain issues related to organising human activities. Relative to Information Technology, *Software Engineering* has got a more theoretical approach focused on training professionals whose essential mission is to develop models and techniques for software production, which scope covers systems infrastructure, as well as organisation and information aspects of enterprises.

This theoretical aspect of Software Engineering stems from the fact that the studied software development procedures have a theoretical sublayer, which is better founded under the Software Engineering Study Programme.

At the same time, the *Software Engineering* Study Programme covers procedures for using the information with the specific aim to design, build and use IT products and services, thus, having established common areas with the Information Technology Programme.

Description of professional training in Software Engineering

Since the beginning of electronic computing in the 40s of the past Century, the computing systems and all the sectors involving them have showed an ever-growing utilization rate. Nowadays, software already defines the essential elements of human activity: governance, communications, production, banks and finances, education, transportation, entertainment, healthcare, agriculture and law. Software products help the world be more efficient and more productive. The OECD data show huge amounts of money spent on software development. Despite such successes, there are serious challenges in terms of development costs, timeliness and quality of many software products. The ACM curricular Guidelines mentions several reasons explaining these challenges, which are definitional for the emergence of a new programme:

- Software products are among the most complex man-made systems, and by its nature, software has got intrinsic, essential properties, which cannot be easily tackled;
- The programming techniques that work well in small teams and for developing moderate products may be not suitable for producing large and complex systems;
- The pace of changes in the area of information and communication technologies (ICT) leads towards new and more advanced/sophisticated products. Therefore, the expectations of beneficiaries and other driving forces put pressure on quality and timeliness of developed products;
- The availability of highly qualified software engineers has not kept pace with the industry demand; therefore, pretty frequently the systems are designed and built by people who lack appropriate training or experience.

The relevance of the new Study Programme is underpinned also by the data of the USAID Survey *(Updating the IT skills gap* - O'Sullivan and Bercu, 2016), which revealed the stringent need for even more professionals in the area of software development.

The high pace of globalization has been largely due to information technologies, which provide opportunities for exploiting the information. The acknowledgment of this fact has been embodied in diverse national and international acts:

- Digital Agenda for *Europe 2020* is part of those seven remarkable initiatives of the European Strategy for Sustainable and Comprehensive Development and is aimed at bringing major economic and social benefits to be generated by a digital single market, which by the middle of its implementation term shall provide the following: a) 50% of population buying online; b) 33% of small and medium-sized enterprises selling online; c) 50% of citizens benefiting from e-Governance services; d) most public services being accessible online in all EU-member countries, etc.
- Transformation of the Republic of Moldova in a modern and effective country is possible only through *technological modernisation* at the level of society, organisations and individuals *(actions that are envisaged also in the Strategic Programme for Technological Modernisation of Governance).*

Therefore, ***the primary goal pursued by the Study Programme*** *is determined by the demand for highly trained engineers in compliance with the area of professional training, who are able to offer advanced software solutions and innovations applied in diverse areas of human activity.*

The Technical University of Moldova (TUM), through its Department on Software Engineering and Automatics, is the *first University that* has trained licentiate engineers in Information Technologies for the national economy *since* 1993. However, the consultations held with its partners (public organisations, private companies and students) have revealed the need for new approaches: *team work and interdisciplinarity.*

As consultations are carried out on a continuous basis, collaboration events are conducted through different workshops and inter-institutional projects involving the Department staff. Among the private companies engaged in the process of consultation or support partnership we can mention Orange, StarNet, Allied Testing, Endava, Pentalog, JMD Planet, Winify, Evisoft, TenerLab, Dekart, etc.

Ensuring Quality Education

The quality of educational activities is a permanent priority of the Department of Software Engineering and Automatics as the unit delivers many study programmes: *Information Technologies, Information Security,* as well as *Automatics and Informatics.* The quality evaluation process comprises a multi-criteria approach, which considers the study results expressed in *knowledge, skills and competences.* The approach components cover the following aspects:

- compliance with the Higher Education Framework Plan *(approved by the Ministry of Education Order No. 1045 of 29 October 2015);*
- compliance with referenced standards *(Software Engineering 2014/ Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, Association for Computing Machinery (ACM), IEEE Computer Society),*
- in terms of the content and structure of the subjects taught – their topicality, integration of research outcomes, developing the skills on how to apply the knowledge in new situations;

Providing with Teaching Staff

Our teaching staff managed to take part in many scientific and didactic activities embedded in *institutional, national and international projects,* by collaborating lately with researchers of similar institutions from Romania, Russia, Denmark, Germany, France, Great Britain, Sweden, Israel, etc. The multidisciplinary feature of this area includes the works of many researchers with academic degrees who have been involved in the study programme:

Total course units/modules	**Number of Course Holders with the functions of**			
	University Professors	Associate Professors	University Lecturers	University Assistants
54	2	11	6 + 1 (with academic degree)	-

Competences developed by the Study Programme and coordination between them and the course units/modules

Professional competences developed by the Study Programme are determined by the definition of Software Engineering Specialty in compliance with the *ACM Standard – Association for Computing Machinery and IEEE Computer Society,* involving a mixture of skills aimed at addressing certain categories of issues outlined through key *competences* related to:

- scientific and engineering foundations of information technologies ;
- organisation and information aspects of systems;

- application technologies;
- software development methods and technologies;
- architecture and infrastructure of computing systems.

The professional and crosscutting competences are covered by fundamental, general, socio-humanistic and specialty subjects, which shares have been set in compliance with the Framework Plan. Further specification of competences developed, as well as their distribution by the content areas is displayed in the matrixes presented in Annexes 1 and 2.

Graduates' Employability

Taking account of the growing need in qualified professionals on the national and regional markets, the TUM graduates have shown a high rate of employability, which is proved also by the USAID Survey. The Classifier of Occupations in the Republic of Moldova, approved by the Government of the RM on 03.03.2014, through the major subgroup *25 Professionals in Information and Communication Technology* with the minor group *251 Software Programmers Analysts* (2511 System Analysts, 2512 Software Designers, 2513 Designers of WEB Systems and Multimedia, 2514 Programmers of Applications, 2519 Software Programmers Analysts not assigned to any of the previous main groups) covers the basic functions/professions of the Software Engineering Programme graduates.

Taking into account the competences acquired following the completion of the *Software Engineer* Programme, the graduates may hold positions other than those mentioned above: from teachers and researchers to director and managers of different levels.

Possibilities for Subsequent Education

Through the stated competences to be attained by the graduates, the Software Engineering Study Programme enables the latter to continue their university studies (Cycle II, Master's Degree) in the ICT area in any local or foreign university within the existing national and international partnerships.

Methods and criteria for competence evaluation

The minimum standards for competence evaluation are displayed in Matrix 1L (Annex 1), the essential evaluation methods comprising: papers/essays, laboratory works intended to develop engineering skills, projects with individual or team tasks with practical completion, tests/exams, licentiate exams and licentiate thesis.

The criteria for competence evaluation, in compliance with the Regulation for organizing the higher education studies on the basis of the National Credit System (the Ministry of Education Order No.726 of 20.09.2010), have been defined through the TUM regulatory acts. Hence, the Regulation for organizing the evaluation of students' learning activity (Order issued by the TUM Rector, entered in force during the 2011/2012 academic year), paragraph *2.3 Evaluation Criteria*, describes in great details the general and specific evaluation criteria (to be supplemented by attitudinal and motivation aspects).

Rules for academic promotion

Promotion to the next year level is conditioned by the accumulation of the mandatory number of credits throughout the academic year foreseen by the Study Plan. It is possible to acquire the allocated credits only when students have been evaluated with marks ranging from “5” to “10”, as per the grading scale outlined in the Regulation for organizing the evaluation of students’ learning activity.

To be awarded the Licentiate Diploma, students shall fulfil the Study Plan and pass the evaluation tests/exams (including the licentiate exams) and defend their licentiate project/thesis with the mark "5" at least.

Foreseen Study Objectives

The Software Engineering Study Programme is aimed at training engineers who shall be able to demonstrate the following qualities:

- Have knowledge and skills in software engineering, be familiar with professional standards required to start the engineering practical activity;
- Demonstrate the understanding and ability to apply theories, models and techniques, which define the foundations for identifying, analyzing, designing, building, implementing, verifying and documenting objectively the industry issues;
- Be able to work both independently and in teams with the aim to develop and deliver high quality software products;
- Demonstrate understanding and pay attention to leadership and communication abilities for negotiations with beneficiaries, which are indispensible components of a typical environment for software development;
- Be able to offer solutions for different areas of application, using software engineering methods and integrating ethic, social, legal and economic aspects;
- Be able to find acceptable solutions, matching the project contradictory objectives, taking into account the existing costs, time, knowledge and systems.

Hence, the Study Programme 0613.3 Software Engineering pursues the following objectives: to train professionals – holders of Licentiate Engineer’ degree, who are able to demonstrate knowledge, skills and crosscutting and professional competences that meet the employers’ requirements, corroborated by the Licentiate Diploma comprising 240 transferable credits and ensuring their employability and/or continuation of Cycle II (Master’s Degree) studies.

Head of Department of Software Engineering and Automatics, Technical University of Moldova

Associate Professor, Dr. Dumitru CIORBA

Annex 2: Matrix 1L – Description of the Study Area/Programme via professional and crosscutting competences.

General Area of Study:	Area of Professional Education:	Study Programme:
061 Information and Communication Technologies	*0613 Development of Software and Applications*	*0613.3 Software Engineering*

Qualification: ***Software Engineering*** Level of qualification: Licentiate/Bachelor's Degree	**Possible occupations (in compliance with the Classifier of Occupations in the RM):** ***25 Professionals in Information and Communication Technology*** *251 Software Programmers Analysts* (2511 System Analysts, 2512 Software Designers, 2513 Designers of WEB Systems and Multimedia, 2514 Programmers of Applications, 2519 Software Programmers Analysts not assigned to any of the previous main groups).

Professional Competences **Level Descrip-tors of structural elements of professional competences**	**C1** **Scientific and engineering foundation of information technologies**	**C2** **Systems organisation and information aspects**	**C3** **Application technologies**	**C4** **Software development methods and technologies**	**C5** **Computing systems architecture and infrastructure**
Knowledge					
D1 Knowledge, understanding the basic industry and specialty concepts, theories and methods; their appropriate use during the professional communication.	C1.1 Identifying and defining *fundamental scientific and applied* concepts, theories and methods supporting the information technology engineering.	C2.1 Identifying and defining concepts, theories and methods used to conduct *human and information focused analyses* on systems operated at the level of organisations.	C3.1 Identifying and defining concepts, procedures and methods for information processing used in *application development depending on the human activity needs.*	C4.1 Identifying and defining concepts and methods focused on *software development, implementation and utilization process.*	C5.1 Identifying and defining hardware, software and communication architecture components, as well as those required for *the description of a computing infrastructure.*
D2 Using the basic knowledge for explaining and interpreting various	C1.2 Explaining engineering solutions by using	C2.2 Explaining concepts, theories and methods used to conduct	C3.2 Explaining technologies appropriate for developing	C4.2 Explaining concepts and methods used for software	C5.1 Explaining the interaction and functioning of architecture and

types of concepts, situations, processes, projects, etc. associated with the industry.	techniques, concepts and principles from pure and applied science.	analyses of systems operated at the level of organisations.	applications required for the organizations activities.	development, implementation and use.	infrastructure components.
Skills					
D3 Applying certain basic principles and methods to address well defined issues/situations, specific for the field under qualified assistance conditions.	C1.3 Addressing the issues related to human activity by applying, in particular, numerical computation techniques and methods.	C2.3 Applying basic concepts, theories and methods to *prepare the information necessary* to develop systems operated at the level of organisations.	C3.3 Using modern technologies to define software applications.	C4.3 Applying programming languages, modelling and development environment, methodologies to produce software.	C5.3 Applying basic methods to specify architecture and infrastructure solutions for typical computing issues.
D4 Appropriate use of standard evaluation criteria and methods to assess the quality, performance and limits/constraints of certain processes, programmes, projects, concepts, methods and theories.	C1.4 Selecting criteria and methods for analysing the advantages and disadvantages of methods and procedures applied in resolving *typical computing issues.*	C2.4 Selecting criteria and methods to assess the quality, performance and limits/constraints of *systems to be developed in* compliance with the needs of the organisation subject to study, including those necessary for defining a quality and security management system.	C3.4 Using criteria and methods determined by the application technologies to assess compliance with interoperability standards.	C4.4 Using criteria and methods to assess the *system development process* in terms of its quality and performance.	C5.4 Using criteria and methods to *assess the functional and non-functional features of system components*.
D5 Devising professional projects using proven industry related principles and methods.	C1.5 Modelling certain standard issues from applied science using math tools.	C2.5 Devising a project (system specification) under the conditions of having a quality and security management system in place.	C3.5 Developing software applications using advanced technologies to convey, store and process data in compliance with the organisation needs.	C4.5 Developing and implementing software for specific problems from diverse areas of human activity.	C5.5 Implementing architecture and infrastructure solutions based on constraints defined by the project.
Minimum Performance	Identifying and applying	Analysing and modelling a	Identifying and using	Analysing, modelling and	Identifying hardware,

Standards for Competence Evaluation.	methods and algorithms learned for standard issues of pure and applied science.	system focused on standard organisation and/or information issues in an area of human activity.	technologies necessary for developing software applications.	devising a functional prototype in compliance with the technological development processes.	software and communication components intended for the applications specific for a selected area.

Level descriptors for structural elements of professional competences	Crosscutting Competences	Minimum Performance Standards for Competence Evaluation
D6. Carrying out professional tasks with due diligence under limited autonomy and qualified support.	CT1. Applying principles, rules and values of professional ethics	Carrying out projects, having complied with the rules of professional deontology.
D7. Getting acquainted with team work specific roles and activities and assigning the tasks to subordinated levels.	CT2. Identifying, describing and unrolling the team activities aimed to develop communication and collaboration skills and to undertake different roles (executive and management roles).	Carrying out team projects, having undertaken different roles.
D8. Acknowledging the need for continuous education; efficient use of resources and learning techniques for personal and professional development	CT3. Demonstrating the spirit of initiative and action for refreshing the professional, economic and organizational culture knowledge.	Devising and applying an individual plan for personal development; communication project in Romanian/Russian and English/French.

Annex 3: Matrix 2L – Coordination between the acquired competences and the course units/module.

Professional Competences	**Competences Explained by Level Descriptors**	**Content Areas**	**Subjects of Study**	**Credits**	
				per sub-ject	**per compe-tence**
1	**2**	**3**	**4**	**5**	**6**
C1 Scientific and engineering foundation of information technologies	C1.1 Identifying and defining *fundamental scientific and applied* concepts, theories and methods supporting the information technology engineering. C1.2 Explaining engineering solutions by using techniques, concepts and principles from pure and applied science. C1.3 Addressing the issues related to human activity by applying, in particular, numerical computation techniques and methods. C1.4 Selecting criteria and methods for analysing the advantages and disadvantages of methods and procedures applied in resolving *typical computing issues.* C1.5 Modelling certain standard issues from applied science using math tools.	Pure and Applied Science	Math	5	**68**
			Special Math 1	5	
			Special Math 2	5	
			Equivalent Models	5	
			Applied Science	5	
			Signal Processing	5	
			Personal and Professional Development/Computer Science and Society	3	
			Project Management/Enterprise Management	3	
			Electronic Marketing/Digital Entrepreneurship	3	
			Cryptography and Security	1	
		Programming	Computer Programming	5	
			Data Structures and Algorithms	5	
			Formal Languages and Compiler Design	5	
			Calculability and Complexity	5	
			Data Analysis and View/ Computer Graphics	1	
			Developing Industry Specific Languages	4	
			Licentiate Internship and Design	2	
			Summary Theory Exam	1	
C2 Systems organisation and information aspects	C2.1 Identifying and defining concepts, theories and methods used to conduct *human and information focused analyses* on	Information Security	Ethics, Communication and Law	2	**17**
			Developing Secure Applications	1	
			Cryptography and Security	1	

	systems operated at the level of organisations. C2.2 Explaining concepts, theories and methods used to conduct analyses of systems operated at the level of organisations. C2.3 Applying basic concepts, theories and methods to *prepare the information necessary* to develop systems operated at the level of organisations. C2.4 Selecting criteria and methods to assess the quality, performance and limits/constraints of *systems to be developed in* compliance with the needs of the organisation subject to study, including those necessary for defining a quality and security management system. C2.5 Devising a project (system specification) under the conditions of having a quality and security management system in place.	Information Management	Project Management/Enterprise Management	1	
			Electronic Marketing/Digital Entrepreneurship	1	
		Software Development	Conceptual Design of an IT Application	3	
			Information System Design	3	
			Summary Theory Exam	1	
			Licentiate Internship and Design	2	
		Software Quality	Software Quality/Analysis and Specification of Software Requirements	2	
C3 Application technologies	C3.1 Identifying and defining concepts, procedures and methods for information processing used in *application development depending on the human activity needs.* C3.2 Explaining technologies appropriate for developing applications required for the organizations activities. C3.3 Using modern technologies to define software applications. C3.4 Using criteria and methods determined by the application technologies to assess	Architectures, Platforms and Technologies	Multimedia Technologies/Simulation and Modelling Techniques	3	52
			Personal and Professional Development/Computer Science and Society	1	
			IoT Projects	2	
			Embedded Systems	3	
			Mobile Application Development/WEB Programming	1	
		Information Management	Databases	5	
			Conceptual Design of an IT Application	2	
			Foundations of Artificial Intelligence/Non-relational Databases	2	

	compliance with interoperability standards. C3.5 Developing software applications using advanced technologies to convey, store and process data in compliance with the organisation needs.	Programming	Basics for Application Development	4	
			Developing Secure Applications	4	
			Equivalent Models	2	
			Developing Industry Specific Languages	2	
			Network Programming	2	
			Object Oriented Programming	2	
			Information System Design	3	
			Programming of Distributed Applications	2	
			Data Analysis and View/ Computer Graphics	2	
			Software Design Techniques and Mechanisms Software Verification and Validation	2	
			Mobile Application Development/WEB Programming	2	
			Foundations for Game Development/ Technologies of Mixed Reality	2	
			Licentiate Internship and Design	3	
			Summary Theory Exam	1	
C4 Software development methods and technologies	C4.1 Identifying and defining concepts and methods focused on *software development, implementation and utilization process.* C4. Explaining concepts and methods used for software development, implementation and use. C4.3 Applying programming languages, modelling and development environment, methodologies to produce software.	Programming	Object Oriented Programming	3	54
			Network Programming	2	
			IoT Projects	4	
			Embedded Systems	2	
			Programming of Distributed Applications	2	
			Data Analysis and View/ Computer Graphics	2	
			Multimedia Technologies/Simulation and Modelling Techniques	2	
			Human-Computer Interaction/ Real Time Programming	3	

	C4.4 Using criteria and methods to assess the *system development process* in terms of its quality and performance. C4.5 Developing and implementing software for specific problems from diverse areas of human activity.		Mobile Application Development/WEB Programming	1	
		Software development	Basics for Application Development	4	
			Developing Secure Applications	3	
			Developing Industry Specific Languages	2	
			Conceptual Design of an IT Application	2	
			Cryptography and Security	3	
			Information System Design	2	
			Foundations of Artificial Intelligence/Non-relational Databases	3	
			Foundations for Game Development/technologies of Mixed Reality	3	
			Summary Theory Exam	1	
			Licentiate Internship and Design	3	
			Defending the Licentiate Project	1	
		Software Quality	Software Design Techniques and Mechanisms Software Verification and Validation	3	
			Software Quality/Analysis and Specification of Software Requirements	3	
C5 Computing systems architecture and infrastructure	C5.1 Identifying and defining hardware, software and communication architecture components, as well as those required for *the description of a computing infrastructure.* C5.2 Explaining the interaction and functioning of architecture and infrastructure components. C5.3 Applying basic methods to specify architecture and	Programming	Programming of Distributed Applications	1	24
			Mobile Application Development/*WEB* Programming	1	
			Network Programming	1	
			Licentiate Internship and Design	2	
		Networks and Data Communications	Computer Network	5	
			IoT Projects	2	
			Computer Architecture	5	

	infrastructure solutions for typical computing issues. C5.4 Using criteria and methods to *assess the functional and non-functional features of system components*. C5.5 Implementing architecture and infrastructure solutions based on constraints defined by the project.	Architectures, platforms and Technologies	Operating Systems: Internal Mechanisms and Design Principles	5	
			Human-Computer Interaction/Real Time Programming	2	

Crosscutting Competences	**Subjects of Study**	**Credits**	
		Per Subject	**Per Competence**
CT1. Applying principles, rules and values of professional ethics	Ethics, Communication and Law	3	**8**
	Conceptual Design of an IT Application	1	
	Personal and Professional Development/Computer Science and Society	1	
	Project Management/Enterprise Management	1	
	Electronic Marketing/Digital Entrepreneurship	1	
	Licentiate Internship and Design	1	
CT2. Identifying, describing and unrolling the team activities aimed to develop communication and collaboration skills and to undertake different roles (executive and management roles).	Equivalent Models	2	**9**
	Developing Industry Specific Languages	1	
	Conceptual Design of an IT Application	1	
	Basics for Application Development	1	
	Developing Secure Applications	1	
	IoT Projects	1	
	Information System Design	1	
	Licentiate Internship and Design	1	
CT3. Demonstrating the spirit of initiative and action for refreshing the professional, economic and organizational culture knowledge.	Equivalent Models	1	**8**
	Developing Industry Specific Languages	1	
	Conceptual Design of an IT Application	1	
	Basics for Application Development	1	
	Developing Secure Applications	1	
	IoT Projects	1	
	Information System Design	1	
	Licentiate Internship and Design	1	
Total per Study Programme			**240**

Annex 4: Grid 1L - Description of the field / study program by professional and transversal skills

Name of qualification: *Software Engineering* Level of qualification: Bachelor		Possible occupations (according to CORM): 25 Specialists in information and communication technology *251 Programmer Analysts in the Software Field* (2511 System Analysts, 2512 Software Developers, 2513 Web and Multimedia Systems Designers, 2514 Application Programmers, 2519 Software Program Analysts Unclassified in Previous Basic Groups)			
Professional skills Descriptors level of the structural elements of professional skills	C1 On the scientific and engineering fundamentals of information technologies	C2 On the organizational and informational aspects of the systems	C3 On application technologies	C4 On software development methods and technologies	C5 On the architecture and infrastructure of computing systems
Knowledge					
D1 Know, understand concepts, theories and basic methods of the field and of the specialization area; proper use of these in professional communication	C1.1 Identify and define concepts, theories and methods *of fundamental and applied sciences* support for information technology engineering	C2.1 Identify and define concepts, theories and methods applied to carry out *analyzes focused on people and information* on systems operating at the organization level	C3.1 Identify and define concepts, processes and methods of processing the information applied to carry out *applications resulting from the needs* of human activity	C4.1 Identify and define concepts and methods focused on *the process of development, implementation and use of software*acquis	C5.1 Identify and define architectural components hardware, software and communications, as well as those needed to *describe a computing infrastructure*
D2 Make use of basic knowledge to explain and interpret various types of concepts, situations, processes, projects, etc. associated with the field	C1.2 Explain engineering solutions by using techniques, concepts and principles in exact and applicative sciences	C2.2 Explain concepts, theories and methods used to analyze systems operating at the organization level	C3.2 Explain the right technologies for producing applications needed in the organization's activities	C4.2 Explain concepts and methods applied to develop, implement and use the software	C5.1 Explain the interaction and operation of architectural and infrastructure components
Skills					
D3	C1.3	C2.3	C3.3	C4.3	C5.3

Apply basic principles and methods for solving well-defined problems / situations, typical for the field, under qualified assistance	Solve problems in human activity fields by applying, in particular, numerical techniques and methods	Apply concepts, theories and basic methods for *preparing the information needed for the development of* systems	Apply modern technologies in defining software applications	Apply programming languages, modeling and development environments, and methodologies for software development	Apply basic methods for specifying architectural and infrastructure solutions for computational problems
D4 Appropriate use of standard criteria and methods of assessment to assess the quality, merits and limits of processes, programs, projects, concepts, methods and theories	C1.4 Select criteria and methods for analyzing the advantages and disadvantages of the methods and procedures applied to solve *numerical computational problems*.	C2.4 Select quality, performance and limits assessment criteria and methods *for the systems to be developed* in accordance with the needs of the study organization, including those required to define a quality and security management system	C3.4 Apply criteria and methods determined by the application technologies to assess compliance with interoperability standards	C4.4 Apply elaboration process assessment criteria and methods a of the systems in terms of quality and performance	C5.4 Apply criteria and methods to evaluate functional and non-functional features of system components
D5 Develop professional projects with the use of established principles and methods in the field	C1.5 Model typical problems in applied sciences using the mathematical system	C2.5 Develop a project (system specification) under the conditions of a quality and security management system	C3.5 Develop software applications using modern technologies for transmitting, storing and processing data in line with what is needed	C4.5 Develop and implement software for concrete problems in various fields of human activity	C5.5 Implement an architectural and infrastructure solution based on constraints stated by the project
Minimum performance standards for skills assessment	Identify and apply learned methods and algorithms to standard problems of fundamental and applied sciences	Analyse and model a system oriented on a organizational and / or informational standard problem of a	Identify and use the technologies needed to develop a software application.	Analyse, model and develop a functional prototype in accordance with technological development processes	Identify the hardware, software and communications components for applications specific to the selected field

		field of human activity.			

Level descriptors of the structural elements of professional skills	Transversal skills	Minimum performance standards for skills assessment
Responsible execution of professional tasks, under restricted autonomy and qualified assistance conditions	CT1. Apply principles, norms and values of professional ethics	Carry out projects in accordance with the rules of professional ethics
Get familiar with roles and activities specific to teamwork and distribute tasks to subordinate levels	CT2. Identify, describe and carry out activities organized in a team with the development of communication and collaboration abilities, as well as with the undertaking of different roles (execution and leadership)	Carry out a project with the team, with the responsible undertaking of different roles
Be aware of the need for continuous training, effective use of learning resources and techniques for personal and professional development	CT3. Show initiative and action spirit to update professional, economic and organizational culture knowledge	Develop and implement a personal development plan; project communication in Romanian / Russian and in English / French.

Annex 5: Grid 2L - Coordination between developed skills and course units / modules

Professional skills	Skills explained by level descriptors	Content areas	Study disciplines	Credits	
				By discipline	By skills
1	2	3	4	5	6
C1 On the scientific and engineering fundamentals of information technologies	C1.1 Identify and define concepts, theories and methods *of fundamental and applied sciences* support for information technology engineering C1.2 Explain engineering solutions by applying the techniques, concepts and principles in exact and applicative sciences C1.3 Solve problems in human activity fields by applying, in particular, numerical calculation techniques and methods C1.4 Select criteria and methods for analyzing advantages and disadvantages of the methods and procedures applied to solve *numerical computational problems*. C1.5 Model typical problems in applied sciences using the mathematical system	Exact and applicative sciences	Mathematics	5	67
			Special Mathematics 1	5	
			Special Mathematics 2	5	
			Equivalent models	5	
			Applied Sciences	5	
			Signal Processing	5	
			Personal and Professional Development / Computer Science and Society	1	
			Project Management / Enterprise Management	2	
			Electronic Marketing / Digital Entrepreneurship	2	
			Cryptography and Security	1	
			Databases	2	
		Programming	Computer programming	5	
			Data Structures and Algorithms	5	
			Formal languages and compiler design	5	
			Calculability and complexity	5	
			|||UNTRANSLATED_CONTENT_START|||Analiza și vizualizarea datelor/ Grafica pe calculator|||UNTRANSLATED_CONTENT_END|||	2	
			Develop domain-specific languages	4	

			Bachelor Internship and Design	2	
			Theoretical synthesis test	1	
C2 On the organizational and informational aspects of the systems	|||UNTRANSLATED_CONTENT_START|||C2.1 Identificarea și definirea conceptelor, teoriilor și metodelor folosite în realizarea de analize focusate pe oameni și informație privind sistemele ce operează la nivel de organizații|||UNTRANSLATED_CONTENT_END||| C2.2 Explain concepts, theories and methods used to analyze systems operating at the organization level C2.3 Apply concepts, theories and basic methods for the preparation of information needed for the development of systems operating at the level of organizations C2.4 Select criteria and methods for assessing the quality, performance and limits of the systems developed in accordance with the organization's needs, including those needed to define a quality and security management system C2.5 Develop a project (system specification) under the conditions of a quality and security management system.	Information security	Ethics, Communication and Law	2	15
			Secure Application Development	1	
			Cryptography and Security	1	
		Information Management	Project Management / Enterprise Management	1	
			Electronic Marketing / Digital Entrepreneurship	1	
		Software development	Conceptual Design of an IT Application	2	
			Information Systems Design	2	
			Theoretical synthesis test	1	
			Bachelor Internship and Design	2	
		Software quality	Software Quality / Analyze and Specify Software Requirements	2	
C3 On application technologies	C3.1 Identify and define concepts, processes and methods of processing the information applied develor *applications resulting from the needs* of human activity C3.2 Explain the right technologies for producing	Architectures, platforms and technologies	Multimedia Technologies / Simulation and Modeling Techniques	3	47
			Personal and Professional Development / Computer Science and Society	1	
			IoT projects	2	
			Embedded Systems	3	

<table>
<tr><td></td><td>applications needed in the organization's activities
C3.3 Apply modern technologies in defining software applications
C.3.4 Apply criteria and methods determined by the application technologies to assess compliance with interoperability standards
C3.5 Develop software applications using modern technologies for transmitting, storing and processing data in accordance with the needs of an organization</td><td></td><td>Mobile Application Programming / Web Programming</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Computer networks</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td>Information Management</td><td>Databases</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Conceptual Design of an IT Application</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td></td><td>|||UNTRANSLATED_CONTENT_START|||Fundamente ale inteligenței artificiale/Baze de date nerelaționale|||UNTRANSLATED_CONTENT_END|||</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td>Programming</td><td>Application Development Fundamentals</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Secure Application Development</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Equivalent models</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Develop domain-specific languages</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Network Programming</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Object-Oriented Programming</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Information Systems Design</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Programming Distributed Applications</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td></td><td>|||UNTRANSLATED_CONTENT_START|||Analiza și vizualizarea datelor/ Grafica pe calculator|||UNTRANSLATED_CONTENT_END|||</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Techniques and mechanisms for software design
Verification and validation of program products</td><td>2</td><td></td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td rowspan="4"></td><td rowspan="4"></td><td>Mobile Application Programming / Web Programming</td><td>2</td><td rowspan="4"></td></tr>
<tr><td>|||UNTRANSLATED_CONTENT_START|||Fundamente ale dezvoltării jocurilor/ Tehnologii de realitate mixtă|||UNTRANSLATED_CONTENT_END|||</td><td>2</td></tr>
<tr><td>Bachelor Internship and Design</td><td>3</td></tr>
<tr><td>Theoretical synthesis test</td><td>1</td></tr>
<tr><td rowspan="11">C4 On software development methods and technologies</td><td rowspan="11">C4.1 Identify and define concepts and methods focused on *the process of development, implementation and use of software*acquis
C2.4 Explain concepts and methods applied to develop, implement and use the software
C4.3 Apply programming languages, modeling and development environments, and methodologies for software development
C4.4 Apply elaboration process assessment criteria and methods of the systems in terms of quality and performance
C4.5 Develop and implement software for concrete problems in various fields of human activity</td><td rowspan="9">Programming</td><td>Object-Oriented Programming</td><td>3</td><td rowspan="11">52</td></tr>
<tr><td>Network Programming</td><td>2</td></tr>
<tr><td>IoT projects</td><td>4</td></tr>
<tr><td>Embedded Systems</td><td>2</td></tr>
<tr><td>Programming Distributed Applications</td><td>2</td></tr>
<tr><td>|||UNTRANSLATED_CONTENT_START|||Analiza și vizualizarea datelor/ Grafica pe calculator|||UNTRANSLATED_CONTENT_END|||</td><td>2</td></tr>
<tr><td>Multimedia Technologies / Simulation and Modeling Techniques</td><td>2</td></tr>
<tr><td>|||UNTRANSLATED_CONTENT_START|||Interacțiunea om-calculator/ Programarea în timp real|||UNTRANSLATED_CONTENT_END|||</td><td>3</td></tr>
<tr><td>Mobile Application Programming / Web Programming</td><td>1</td></tr>
<tr><td rowspan="2">Software development</td><td>Application Development Fundamentals</td><td>4</td></tr>
<tr><td>Secure Application Development</td><td>3</td></tr>
</table>

			Develop domain-specific languages	1	
			Conceptual Design of an IT Application	1	
			Cryptography and Security	3	
			Information Systems Design	2	
			|||UNTRANSLATED_CONTENT_START|||Fundamente ale inteligenței artificiale/Baze de date nerelaționale|||UNTRANSLATED_CONTENT_END|||	3	
			|||UNTRANSLATED_CONTENT_START|||Fundamente ale dezvoltării jocurilor/Tehnologii de realitate mixtă|||UNTRANSLATED_CONTENT_END|||	3	
			Theoretical synthesis test	1	
			Bachelor Internship and Design	3	
			Defend Bachelor's Project	1	
		Software quality	Techniques and mechanisms for software design Verification and validation of program products	3	
			Software Quality / Analyze and Specify Software Requirements	3	
C5 On the architecture and infrastructure of computing systems	C5.1 Identify and define architectural components hardware, software and communications, as well as those needed to *describe a computing infrastructure*	Programming	Programming Distributed Applications	1	20
			Mobile Application Programming / Web Programming	1	
			Network Programming	1	

	C5.1 Explain the interaction and operation of architectural and infrastructure components C5.3 Apply basic methods for specifying architectural and infrastructure solutions for computational problems C5.4 Apply criteria and methods to evaluate functional and non-functional features of system components C5.5 Implement an architectural and infrastructure solution based on constraints stated by the project		Bachelor Internship and Design	1	
		Networks and data communications	Computer networks	3	
			IoT projects	1	
		Architectures, platforms and technologies	Computer architecture	5	
			Operating systems: internal mechanisms and design principles	5	
			|||UNTRANSLATED_CONTENT_START|||Interacțiunea om-calculator/ Programarea în timp real|||UNTRANSLATED_CONTENT_END|||	2	

Transversal skills	**Study disciplines**	**Credits**	
		By discipline	**By skills**
CT1. Apply principles, norms and values of professional ethics	Ethics, Communication and Law	3	17
	Conceptual Design of an IT Application	2	
	Personal and Professional Development / Computer Science and Society	3	
	Project Management / Enterprise Management	1	
	Electronic Marketing / Digital Entrepreneurship	1	
	Equivalent models	1	
	Application Development Fundamentals	1	
	Develop domain-specific languages	1	
	Secure Application Development	1	
	IoT projects	1	
	Information Systems Design	1	
	Bachelor Internship and Design	1	
CT2. Identify, describe and carry out activities organized in a team with the development of communication and collaboration abilities, as well as with the	Equivalent models	2	10
	Develop domain-specific languages	1	
	Conceptual Design of an IT Application	2	

undertaking of different roles (execution and leadership)	Application Development Fundamentals	1	
	Secure Application Development	1	
	IoT projects	1	
	Information Systems Design	1	
	Bachelor Internship and Design	1	
CT3. Show initiative and action spirit to own update professional, economic and organizational culture knowledge	Project Management / Enterprise Management	1	12
	Electronic Marketing / Digital Entrepreneurship	1	
	Equivalent models	1	
	Develop domain-specific languages	1	
	Conceptual Design of an IT Application	2	
	Application Development Fundamentals	1	
	Secure Application Development	1	
	IoT projects	1	
	Information Systems Design	1	
	Bachelor Internship and Design	2	
Total study program			240

Annex 6: Roadmap

2020

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Prepare a new study program for the PBL methodology implementation.												
Revise the Automation and Informatics Education Plan.	🟦	🟦	🟦	🟦	🟦				🟦	🟦		
Approve the Education Plan.											🟦	🟦
Prepare the teaching infrastructure based on the PBL methodology.									🟦	🟦	🟦	🟦
Identify the involved teachers and conduct the teacher training in PBL.												
Review the curriculum by subjects, subject sheets, case studies.												
Admission 2021.												
Internationalization of the Software Engineering Study Program.												
Carry out various cooperative and partnership activities based on international projects with foreign universities.	🟦	🟦	🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦
Enhance international compatibility of the Software Engineering study program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Embark on international projects providing mobility for teachers and students.	🟦	🟦	🟦								🟦	🟦
Introduce changes to the Education Plan to ensure that mobility is performed									🟦	🟦	🟦	🟦

with the equivalency of accumulated credits.									🟦	🟦	🟦	🟦
Carry out optional English language courses for teachers.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Implement external mobilities for students and academic staff.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Attract international companies that will support the knowledge transfer at the level of content, teachers and student practice.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Further develop and adapt the Software Engineering Study Program to the needs of the economic environment by integrating research elements at the teaching and student level.												
Enhance interactivity with the private environment.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Review the curriculum by subjects, subject sheets, guidelines, case studies, evaluations.	🟦	🟦	🟦	🟦	🟦							
Upgrade the PBL-based teaching infrastructure.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Ensure interdisciplinarity.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Develop research partnerships focused on research with other universities, institutions.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Organize scientific events.			🟦							🟦		
Strengthen scientific research directions.			🟦	🟦	🟦				🟦	🟦	🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Provide methodological support for teacher training in PBL.												
Develop methodological support for teacher training in PBL.	■	■	■	■	■				■	■	■	■
Continuous training of teachers in PBL methodology.									■	■	■	

2020

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Prepare a new study program for the PBL methodology implementation.												
Revise the Automation and Informatics Education Plan.												
Approve the Education Plan.	■	■	■	■	■							
Prepare the teaching infrastructure based on the PBL methodology.	■	■	■	■	■	■			■	■	■	■
Identify the involved teachers and conduct the teacher training in PBL.		■	■	■	■	■			■	■	■	■
Review the curriculum by subjects, subject sheets, case studies.						■	■	■	■	■	■	■
Admission 2021.												
Internationalization of the Software Engineering Study Program.												
Carry out various cooperative and partnership activities based on international projects with foreign universities.	■	■	■	■	■	■			■	■	■	■
Enhance international compatibility of the	■	■	■	■	■	■	■	■	■	■	■	■

Software Engineering study program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Embark on international projects providing mobility for teachers and students.	🟦	🟦									🟦	🟦
Introduce changes to the Education Plan to ensure that mobility is performed with the equivalency of accumulated credits.	🟦	🟦										
Carry out optional English language courses for teachers.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Implement external mobilities for students and academic staff.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Attract international companies that will support the knowledge transfer at the level of content, teachers and student practice.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Further develop and adapt the Software Engineering Study Program to the needs of the economic environment by integrating research elements at the teaching and student level.												
Enhance interactivity with the private environment.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Review the curriculum by subjects, subject sheets, guidelines, case studies, evaluations.	🟦	🟦	🟦	🟦	🟦							
Upgrade the PBL-based teaching infrastructure.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Ensure interdisciplinarity.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Develop research partnerships focused on research with other universities, institutions.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Organize scientific events.		🟦	🟦						🟦	🟦	🟦	

Strengthen scientific research directions.	■	■	■						■	■	■	■
Promote the Software Engineering Study Program.	■	■	■	■	■	■	■	■	■	■	■	■
Provide methodological support for teacher training in PBL.												
Develop methodological support for teacher training in PBL.	■	■	■	■	■				■	■	■	■
Continuous training of teachers in PBL methodology.		■	■	■	■				■	■	■	■

2021

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Prepare a new study program for the PBL methodology implementation.												
Revise the Automation and Informatics Education Plan.												
Approve the Education Plan.												
Prepare the teaching infrastructure based on the PBL methodology.	■	■	■	■	■							
Identify the involved teachers and conduct the teacher training in PBL.	■	■	■	■	■							
Review the curriculum by subjects, subject sheets, case studies.	■	■	■	■	■							
Admission 2021.							■	■				
Internationalization of the Software Engineering Study Program.												

Carry out various cooperative and partnership activities based on international projects with foreign universities.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Enhance international compatibility of the Software Engineering study program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Embark on international projects providing mobility for teachers and students.	🟦	🟦									🟦	🟦
Introduce changes to the Education Plan to ensure that mobility is performed with the equivalency of accumulated credits.	🟦	🟦	🟦	🟦	🟦							🟦
Carry out optional English language courses for teachers.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Implement external mobilities for students and academic staff.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Attract international companies that will support the knowledge transfer at the level of content, teachers and student practice.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Further develop and adapt the Software Engineering Study Program to the needs of the economic environment by integrating research elements at the teaching and student level.									🟦	🟦	🟦	🟦
Enhance interactivity with the private environment.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Review the curriculum by subjects, subject sheets, guidelines, case studies, evaluations.	🟦	🟦	🟦	🟦	🟦							🟦

Upgrade the infrastructure adapted to the PBL methodology.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Ensure interdisciplinarity.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Develop research partnerships focused on research with other universities, institutions.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Organize scientific events.			🟦	🟦						🟦	🟦	
Strengthen scientific research directions.	🟦	🟦	🟦	🟦					🟦	🟦	🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Provide methodological support for teacher training in PBL.									🟦	🟦	🟦	
Develop methodological support for teacher training in PBL.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦
Continuous training of teachers in PBL methodology.	🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦	🟦

2022

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
Prepare a new study program for the PBL methodology implementation.										
Revise the Automation and Informatics Education Plan.										
Approve the Education Plan.										
Prepare the teaching infrastructure based on the PBL methodology.										
Identify the involved teachers and conduct the teacher training in PBL.										
Review the curriculum by subjects, subject sheets, case studies.										
Admission 2021.										
Internationalization of the Software Engineering Study Program.										

Carry out various cooperative and partnership activities based on international projects with foreign universities.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Enhance international compatibility of the Software Engineering study program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Embark on international projects providing mobility for teachers and students.	🟦	🟦								
Introduce changes to the Education Plan to ensure that mobility is performed with the equivalency of accumulated credits.	🟦	🟦	🟦							
Carry out optional English language courses for teachers.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Implement external mobilities for students and academic staff.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Attract international companies that will support the knowledge transfer at the level of content, teachers and student practice.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Further develop and adapt the Software Engineering Study Program to the needs of the economic environment by integrating research elements at the teaching and student level.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Enhance interactivity with the private environment.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Review the curriculum by subjects, subject sheets, guidelines, case studies, evaluations.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Upgrade the PBL-based teaching infrastructure.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Ensure interdisciplinarity.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Develop research partnerships focused on research with other universities, institutions.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Organize scientific events.			🟦							
Strengthen scientific research directions.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Promote the Software Engineering Study Program.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Provide methodological support for teacher training in PBL.										
Develop methodological support for teacher training in PBL.	🟦	🟦	🟦	🟦	🟦				🟦	🟦
Continuous training of teachers in PBL methodology.	🟦	🟦	🟦	🟦	🟦				🟦	🟦

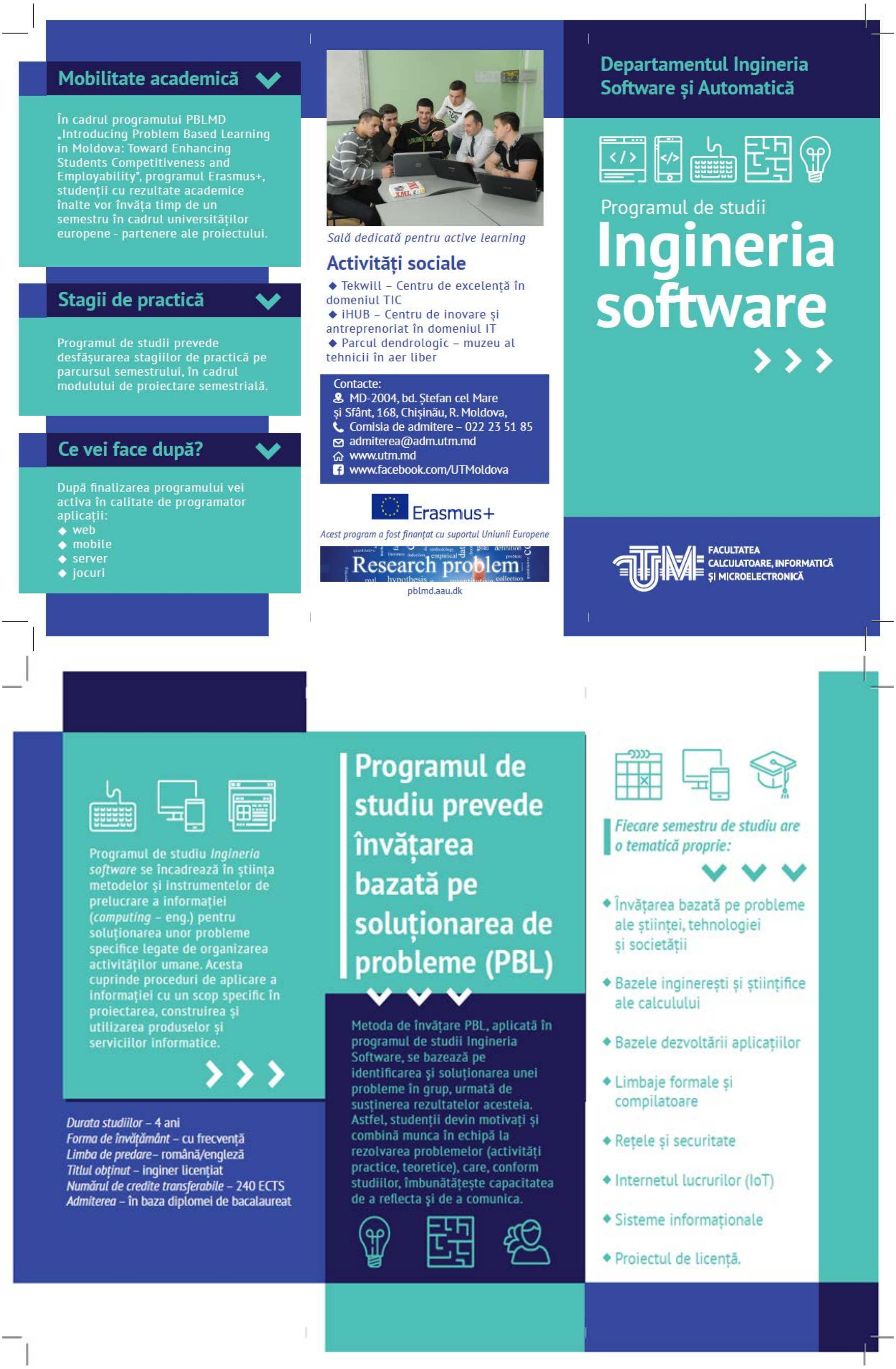
Annex 7: Action Plan

Actions	Required resources
PERIOD 1	
Revise the Education Plan for the training of specialists in Automation and Informatics according to the TUM Regulation on the organization of studies based on the National Education Credit System, taking into account the Regulation on the organization of the higher education based on the National Education Credit System, so that the program is linked to the national and international standards of training specialists in the field and is in line with the Framework-Plan. It is expected that starting with the 2nd year, in each semester, students will have a special subject dedicated to design/projects.	Provide support at department, faculty, university level.
Approve the Education Plan within the Software Engineering and Automation Department; Faculty of Computers, Informatics and Microelectronics and the TUM Senate.	Support at department, faculty, university level.
Identify the teachers who will be involved in the teaching process under the new study program and train them for their use of the PBL teaching methodology.	Human resources.
Upgrade the teaching infrastructure based on the PBL methodology.	Financial resources Support at department, faculty, university level.
Revise curriculum by subjects (analytical programs), subject fiches, guides, case studies, evaluations etc. (for the first year of study).	Human resources.
PERIOD 2	
Carry out various cooperative and partnership activities based on international projects with foreign universities.	Support at University level. Developed infrastructure.
Stimulate the interest of students and academic staff in the external mobility.	Support at University level. Human resources.
Enhance the number of external mobilities for students and academic staff.	Support at University level. Human resources. Foreign partners. Collaborative projects. Financial resources.
Increase the number of students with strong English language skills.	Human resources. Financial resources.
Carry out optional English language courses for teachers.	Human resources. Financial resources.

Improve curricular content and teaching skills for all areas of study, increase international compatibility of study programs and integrate international aspects and study periods into the study process.	Human resources. Provide support at department, faculty, university level.
Re-design study program on the basis of international cooperation opportunities (student mobility, double diplomas, etc.).	Human resources. Provide support at department, faculty, university level. Foreign partners.
Attract international teachers to deliver courses.	Foreign partners. Human resources. Financial resources.
Launch several international projects provide mobility for both teachers and students.	Support at University level. Human resources. Foreign partners. Collaborative projects. Financial resources.
Introduce changes to the Education Plan to ensure that mobility is performed with the equivalency of accumulated credits.	Human resources. Provide support at department, faculty, university level. Foreign partners.
Enhance interaction with the private environment, attract international companies.	Provide support at department, faculty, university level. Foreign partners.
Promote the Software Engineering Study Program	Provide support at department, faculty, university level.
PERIOD 3	
Develop research-oriented scientific partnerships with universities, institutions and companies from Moldova and abroad (Europe and the whole world).	Support at University level. Human resources. Foreign partners. Collaborative projects. Financial resources.
Revise Curriculum by subjects, subject fiches, guides, case studies, evaluations, etc. so that students develop the transversal skills needed for a successful employment, taking into account performance indicators.	Provide support at department, faculty, university level. Human resources.
Upgrade the PBL-based teaching infrastructure.	Provide support at department, faculty, university level. Financial resources.
Participate in international scientific events.	Provide support at department, faculty, university level. Financial resources. Human resources.
Attract internationally recognized experts to participate in the events organized within the Software Engineering Study Program.	Provide support at department, faculty, university level. Human resources. Financial resources.

Initiate or revive cooperation agreements with international organizations operating in the Republic of Moldova.	Support at University level. Human resources. Foreign partners. Collaborative projects. Financial resources.
Organize international events with the participation of international partners from international programs / projects / organizations.	Human resources. Provide support at department, faculty, university level. Collaborative projects. Financial resources.
Identify companies undertaking the responsibility to support knowledge transfer at the level of content, teachers and internships.	Human resources. Provide support at department, faculty, university level. Private Partners.
Integrate research at the teaching and student level into the study process.	Support at University level. Human resources. Research projects. Financial resources.
Initiate interdisciplinary annual/Bachelor’s/Master’s projects among students from the study programs of the CIM faculty, but also from other faculties, by encouraging teamwork.	Human resources. Provide support at department, faculty, university level.
Strengthen scientific research directions carried out in interdisciplinary groups (inter-departmental / university)/.	Support at University level. Human resources. Research projects. Financial resources.
Attract students to scientific activities - by strengthening students' scientific and practical groups and their involvement in the research activity of the teaching staff.	Support at University level. Human resources. Research projects. Financial resources.
Promote performance in educational and research processes.	Provide support at department, faculty, university level.
PERIOD 4	
Provide continuous assistance in professional problem solving.	Provide support at department, faculty, university level.
Develop methodological support for teacher training in PBL.	Provide support at department, faculty, university level. Human resources.
Initiate compulsory course of continuous teacher training in PBL methodology.	Human resources. Support at department, faculty, university level. Financial resources.

Annex 8: Promotional Flier Software Engineering



Mobilitate academică

În cadrul programului PBLMD „Introducing Problem Based Learning in Moldova: Toward Enhancing Students Competitiveness and Employability", programul Erasmus+, studenții cu rezultate academice înalte vor învăța timp de un semestru în cadrul universităților europene - partenere ale proiectului.

Stagii de practică

Programul de studii prevede desfășurarea stagiilor de practică pe parcursul semestrului, în cadrul modulului de proiectare semestrială.

Ce vei face după?

După finalizarea programului vei activa în calitate de programator aplicații:

- web
- mobile
- server
- jocuri

Sală dedicată pentru active learning

Activități sociale

- Tekwill – Centru de excelență în domeniul TIC
- iHUB – Centru de inovare și antreprenoriat în domeniul IT
- Parcul dendrologic – muzeu al tehnicii în aer liber

Contacte:
MD-2004, bd. Ștefan cel Mare
și Sfânt, 168, Chișinău, R. Moldova,
☎ Comisia de admitere – 022 23 51 85
admiterea@adm.utm.md
www.utm.md
www.facebook.com/UTMoldova

Erasmus+

Acest program a fost finanțat cu suportul Uniunii Europene

Research problem

pblmd.aau.dk

Departamentul Ingineria Software și Automatică

Programul de studii

Ingineria software

›››

UTM FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ

Programul de studiu *Ingineria software* se încadrează în știința metodelor și instrumentelor de prelucrare a informației (*computing* – eng.) pentru soluționarea unor probleme specifice legate de organizarea activităților umane. Acesta cuprinde proceduri de aplicare a informației cu un scop specific în proiectarea, construirea și utilizarea produselor și serviciilor informatice.

›››

Durata studiilor – 4 ani
Forma de învățământ – cu frecvență
Limba de predare – română/engleză
Titlul obținut – inginer licențiat
Numărul de credite transferabile – 240 ECTS
Admiterea – în baza diplomei de bacalaureat

Programul de studiu prevede învățarea bazată pe soluționarea de probleme (PBL)

Metoda de învățare PBL, aplicată în programul de studii Ingineria Software, se bazează pe identificarea și soluționarea unei probleme în grup, urmată de susținerea rezultatelor acesteia. Astfel, studenții devin motivați și combină munca în echipă la rezolvarea problemelor (activități practice, teoretice), care, conform studiilor, îmbunătățește capacitatea de a reflecta și de a comunica.

Fiecare semestru de studiu are o tematică proprie:

- Învățarea bazată pe probleme ale științei, tehnologiei și societății
- Bazele inginerești și științifice ale calculului
- Bazele dezvoltării aplicațiilor
- Limbaje formale și compilatoare
- Rețele și securitate
- Internetul lucrurilor (IoT)
- Sisteme informaționale
- Proiectul de licență.