Informatics in Education, 2018, Vol. 17, No. 1, 21–44

DOI: 10.15388/infedu.2018.02

Competency Assessment in Problem-Based Learning Projects of Information Technologies Students

Agnė BRILINGAITĖ[1] , Linas BUKAUSKAS[1], Anita JUŠKEVIČIENĖ[2]
[1]*Vilnius University Institute of Computer Science*
Didlaukio str. 47, LT-08303 Vilnius, Lithuania
[2]*Vilnius University Institute of Data Science and Digital Technologies*
Akademijos str. 4, LT-08663 Vilnius, Lithuania
e-mail: {agne.brilingaite, linas.bukauskas}@mif.vu.lt, anita.juskeviciene@mii.vu.lt

Received: August 2017

Abstract. The rapid development of new technologies requires a new set of skills from all users in operating, using, and maintaining digitised environments. Curriculum design in the field of Information and Communication Technologies (ICT) undergoes a rapid change as technological achievements do influence education.

The aim of the article was to present research results on the mappings of learning outcomes to taxonomies to evolve from requirement-based assessment to competency-based assessment. The research was carried out on the problem-based learning (PBL) module. The article presents a novel PBL model design with activities and evaluation schema. The developed model could be used to guide the curricula design of information technologies for generic competencies, and to foster skills essential for the future ICT sector.

Keywords: problem-based learning, competency assessment, taxonomies, information technologies.

1. Introduction

The rapid development of new technologies and the appearance of new ubiquitous environments require a new set of skills from all users in operating, using, and maintaining environments. A whole new block of competencies is required for the future employees to broaden frontiers of research and development of new services and applications. Educational institutions play an essential role in preparing future developers and entrepreneurs employing the right setup of curricula and methods that would be attractive to the nowadays learner.

Current technological, socio-economic, geopolitical, and demographic developments influence the current and future jobs. Reports of the World Economic Forum (WEF, 2016) emphasise that previously disjointed fields like artificial intelligence, mobile and telecommunication, 3D printing, biotechnologies, genetics, nanotechnologies now intersect enriching each other. Thus, new types of jobs are created now and going to be created in the near future. The current business sectors must adapt to changes fostering human resources and their multidisciplinarity. The technological revolution in business is highly affected by the development of information technologies (IT) sector. In many sectors, employees already need or soon will require technical skills, especially IT skills. Business has to work with big data, cloud computing, mobile internet, etc. IT skills become essential in many industrial sectors such as industrial engineering as well as in the humanities and social sciences. Thus, traditional IT distinction among other fields becomes meaningless unless the explicit IT professional competencies are highlighted.

By 2020, the Information and Communication Technology sector will need employees having five essential skills: complex problem-solving, critical thinking, cognitive flexibility, mathematical reasoning, and active learning (WEF, 2016). Of course, technological skills are essential, but the industry will require IT specialists having strong generic skills. IT people work with professionals from various sectors. Thus, they have to stand out in the crowd to be understood and valued by the society, co-workers, and collaborators. Educational institutions must adapt to the changing needs, too.

Educational and quality processes, curricula design and assessment transfer are regulated by various national and international legislative documents, the Bologna process, international accreditation companies, as well as by the labour market and associations. The Association for Computing Machinery (ACM) and Computer Society of the Institute of Electrical and Electronics Engineers (IEEE-CS) have been working on the new recommendations for curricula of undergraduate programmes of Information Technology (Sabin *et al.*, 2017). The ACM curricula emphasise Information Technologies programs to be more applied than theoretical. The curricula implementation recommendations include sharpening oral and writing skills, student engagement in team-oriented projects that extend over a reasonably long period, and student experience with non-IT people to understand the application domain. Learning by doing is an essential and very appreciated learning style of IT students. Problem-based learning (PBL) is an instructional (and curricular) learner-centred approach that empowers learners to conduct research, integrate theory and practice, and apply knowledge and skills to develop a viable solution to a defined problem (Savery, 2015, p.5). Thus, PBL enables complementary development of multiple generic and subject-specific competencies. Additionally, in the problem-based approach, complex and real world problems are used to motivate students to identify and research the concepts and principles they need to know to work through those problems (Duch *et al.*, 2001). While in traditional classes, students lack motivation because classes and their projects are organized using subject-specific tasks.

This article presents a novel implementation of a PBL model together with a set of activities, mapping of competency levels in various taxonomies, and assessment strat-

egy of competencies of IT students. The model considers recommendations for curricula design of information technologies for to generic competencies, and it follows the trends of the current industry. PBL model as an education module (PBL module) simulates the software development cycle. Thus, students make groups aiming at a software or technology blend as a final result. The PBL module is enhanced with activities to cover the development of technical and generic skills from various perspectives. For example, peer-review activity obligates students to demonstrate systemic thinking and communication skills. The task requires to delve into the other group's problem at hand, understand, and to present positive, critical and recommended aspects of the project.

The model was experimented with in the study programme Information Technologies at Vilnius University. The study programme has two PBL modules, during autumn and spring semesters, respectively. The PBL modules have three times larger volume credit-wise than typical subjects at the university. This article considers only spring semester module PBLII. The module concentrates on the software development with the emphasis on *Information Technologies*, not pure computer science or software engineering. Thus, developed technological competencies are related to usage of technologies, software project management, interdisciplinary attitude, system integration, and system deployment in the cloud infrastructure.

The assessment strategy and mapping of competency that levels it to various taxonomies is an essential quality when building and maintaining the curricula. The learning outcomes of the study programme are presented using Bloom's taxonomy. The main level of learning outcomes is application – ability to apply knowledge. But the assessment criteria are discussed concerning SOLO, Marzan taxonomies, and understanding by design framework (UbD). The experimental case study shows that Bloom taxonomy limits flexibility to distinguish students due to different levels of implemented projects. The learning outcome is the goal while assessment criteria must go beyond the Bloom taxonomy and consider knowledge application levels in a close-to-reality situation. Thus, the goal of the work was to show the possible mapping of learning outcomes to other taxonomies to validate study activities for competency-based assessment.

The article is structured as follows. Section 2 covers literature on problem-based learning and taxonomies applied. Section 3 presents a case study of the study programme. Schema of module activities and assessment structure are covered in Sections 4 and 5, respectively. Experimental results with mapping to taxonomies are described in Section 6. The paper ends up with conclusions.

2. Background

In this section, the underlying theory of problem-based learning is laid out. It is subsequently followed by the presentation of the notion of taxonomies that eventually oriented the author(s) of this paper to structure their model and research tool.

Problem-based learning

PBL has its origin in medical education, and it is widely adopted for different disciplines such as engineering, geography (Xian, Madhavan, 2015, p. 282), mathematics (Letchumanan, 2009), and computer science (Koch, Teege, 1999). It has also been successfully used in primary, basic, secondary, vocational and higher education (Torp and Sage, 2002), and in different subjects, in different contexts, learners of different ages are trained (Maudsley, 1999), and for different purposes (e.g. curriculum design and developments, learning assessment, students motivation (Xian, Madhavan, 2015, p. 290).

PBL is also one of the methods of constructivist teaching and learning theory. The pioneer of the constructivist theory, S. Papert, has expanded the theory of constructivism by stating that learning is best when the learner actively develops objects of the real world, and not just ideas or knowledge that deliberately engages in design. New knowledge is created from interrelation with the world through perception and action (Hendry *et al.*, 1999).

Since problem-based learning is learner-centred, it means that the learner is therefore responsible for his/her learning and solving problems on his/her own (Savery, 2015). Thus, learners have work to independently enough to acquire needed skills and to choose appropriate learning methods or methodologies. This independence encourages learner motivation (Savery and Duffy, 1995). However, learning autonomously, especially in new situations is quite challenging for students. Students could learn and do discrete things but not be able to put it all together in the context. Due to this, the transformation of a content-based curricular model into competency-based is needed. This can be achieved by using the Understanding by Design framework (Sabin *et al.*, 2017, p.28). In the competency-based IT curricular framework, competency is a measure of significant learning, achieved through aligning performance goals with active, inquiry, collaborative, experiential, and reflective learning.

In PBL, the teacher is a facilitator of collaborative knowledge construction often is referred to as the tutor. Facilitation techniques are crucial to the success of PBL. To this end, McCaughan (2015) develops a PBL tutoring framework as recommendations for the communication techniques, strategies, actions of the PBL tutor. Likewise, to facilitate the learning well the role of the teacher is critical. Thus, the tutor must be well equipped with different skills to inspire students for self-direct learning, teamwork, communication, social skills and lifelong learning. PBL is student-centred learning, not teacher-centred teaching, hence introducing a new teacher role and a new student-teacher relationship. Subsequently the similarities of three theories were explored that include Howard Barrows' principles for the PBL tutor's actions, Dewey's theories that address teacher behaviors and Carl Rogers's conceptual frameworks that support the therapeutic behaviors of the client-centered therapist; and extracts the common facilitation techniques in order to show the support of the psychological and educational principles.

In more detail, PBL can be characterised as an environment with such key elements that include student-centred, ill-structured problems, multi-disciplinary focus, self-regulation, collaboration, reflection, and evaluation, and, closing analyses (Nelson, 2010).

Thus, students must have the responsibility for their learning, and the problems they are solving must be from the real world. The learning process should be integrated from a wide range of disciplines and involve information sharing, self and peer assessment.

Based on these core features, Ho and Chan (2015) develop the PBL model further. This model consists of seven-step cyclical PBL learning process: (1) Present the problem statement; (2) Analyse the problem statement; (3) Formulate the investigation plan; (4) Present the investigation plan; (5) Carry out the investigation plan; (6) Present results, conclusions and recommendations; (7) Evaluate and propose new inquiry.

To have a successful PBL implementation, the problem statement must be ill-structured. Therefore it means that the problem is complex and cannot be solved by a simple algorithm. Such problems do not necessarily have a single correct answer but require learners to consider alternatives and to provide a reasoned argument to support the solution that they generate (Hmelo-Silver, Barrows, 2006). If the problem is well-structured, it consequences less motivation and effort of the students in the development of the solution (Savery, 2015, p.8). The analyses phase can be divided into two sub-activities: identification of what is known and unknown, and determination of what needs to be known. Thus, students must have the responsibility for their learning. In order to formulate the investigation plan, the goal must be set as well as the hypotheses. Methods, resources, activities, procedures are also decided at this phase. Also, the roles of group members and the timeline are set up. Next, the plan is presented and refined. At step 5 the data is collected, the progress is monitored and evaluated, and the used strategies are regulated. Next, the findings are analysed, the conclusions are drawn, and recommendations are suggested. The last step involves self and group evaluation, future investigation proposition.

PBL enhances problem-solving, independent learning and teamwork, cooperation and collaboration skills (Prince *et al.*, 2005). Those are known as general competencies. Additionally, PBL facilitates the development of key professional competencies such as critical thinking, communication skills, interpersonal relations, and self-assessment (Chaves *et al.*, 2006). In a broader interpretation, the *competency* indicates sufficiency of knowledge and skills that enable someone to act in a wide variety of situations. It can be defined as "a cluster of related abilities, commitments, knowledge, and skills that enable a person to act effectively in a job or situation". [1]

Research on Competency models conducted by Markus *et al.*, (2005) showed that the published competency definitions can be grouped into three distinct approaches: educational standards (knowledge, skills and attitudes), behaviour repertoires (knowledge, motives, traits, self-images and social roles and skills), and organizational competencies (organisational competencies for competitive advantage).

Broadly, the competencies can be generic (universal) and specific. This division is usually used in competency models – as a descriptive tool that identifies the knowledge, skills, abilities, and behaviour needed to perform effectively in an organisation (Markus *et al.*, 2005; Letelier *et al.*, 2003). As mentioned above PBL enhances general competencies, however it is found that this approach is also suitable for specific competencies

[1] http://www.businessdictionary.com/definition/competence.html

development (such as, the ability to reason scientifically about concepts and techniques, and able to communicate the scientific problem, contributions to its solution), often combining it with learning taxonomies, for example Bloom's, PBL Aalborg model (Dolog *et al.*, 2016).

Sometimes competencies are considered as learning outcomes – the abilities related to professional performance (Letelier *et al.*, 2003), for example, engineering design, project management; either are mapped to learning outcomes such as ACM Competency Model classification system mapped to Information Technology Curriculum (Hawthorne *et al.*, 2014). Competency models should provide an operational definition for each competency and sub competency, together with measurable or observable performance indicators or standards against which to evaluate individuals.

Thus, learning assessment can be done clearly and without ambiguity by verifying competency achievement in students as well as using learning taxonomies – educational learning objectives models.

2.1. *Taxonomies*

Taxonomy in a broad sense is defined as a science of classification or a systematic framework. Consequently, it means the quality assurance of proper assessment because it leads to the elimination of the mismatch between what is intended and what is achieved. In education, it is crucial to have clear links between learning objectives, assessment and outcomes. Therefore the need of taxonomies is obvious.

Two widely used taxonomies for assessment of learning are the Bloom taxonomy (original and modified) and the SOLO taxonomy (Meerbaum-Salant *et al.*, 2013). Modified taxonomies such as those proposed by Anderson *et al.*, (2001), and the New Taxonomy by Marzano and Kendall's (Marzano and Kendal, (2008) were developed based on original Bloom taxonomy (Bloom *et al.*, 1956) due to flaws and inconsistencies (Marzano and Kendal, 2008, p.1), and drawbacks to it use in computer science field (Fuller *et al.*, 2007).

In the computer science (CS) field, the assessment design is most often carried out by using original and revised Bloom's taxonomy. One of such revisions is presented by Johnson, Fuller (2007): they incorporated higher application capstone level arguing that in computer science (CS) modules the focus of assessment appeared to be at the application level. However, the question about such the reformulation of Bloom's taxonomy (for more helpful descriptions of cognitive levels in CS) was not answered.

Another solution was presented by (Fuller *et al.*, 2007) to separate Bloom's six levels into two dimensions: Producing (incorporating apply and create) and Interpreting (incorporating remember, understand, analyse and evaluate). Thus, the generated matrix can be used to identify a range of different learning trajectories and hence to guide students on how to improve their skills and understanding. Therefore, to illustrate the proposed by Fuller *et al.*, taxonomy application, a list of problem-solving activities were provided with descriptions related to programming, which included: *Adaptation* for the modification of a solution of other domains/ranges. The competency close to *Create* on the verti-

cal scale, and at least, *Understand* on the horizontal scale, because modifying involves production, and, knowing what and how to modify requires understanding.

The proposed taxonomy by Fuller *et al.* was then developed based on the literature review on educational taxonomies and their use in computer science education with the special focus on the problems identified. Incidentally, the need of a new taxonomy was presented in (Meerbaum-Salant *et al.* (2013). It was identified that there were some difficulties in categorising programming activities within the existing taxonomies. As a consequence, they developed a new one by combining the revisions made by Anderson *et al.* (2001) Bloom and SOLO taxonomies. To this end, in that study, three intermediate categories of unistructural, multistructural, and relational from SOLO and three from Bloom taxonomy (understanding, applying, and creating) were selected to focus on. The proposed taxonomy application was implemented by developing the tests for middle-school students' assessment who were learning concepts of CS from Scratch. The taxonomy was augmented by observations and interviews.

The original Bloom's taxonomy was also used for specifying learning outcomes in CS before assessment and for exploration, specification, and refinement of assessable learning objectives in CS courses. As expressed by Starr *et al.* (2008), a process for taxonomy application in computer science (and other disciplines) was presented by specifying assessable learning objectives throughout the CS curriculum focused on programmatic assessment, as well a case study of how taxonomy may be applied in a CS course.

3. Case Study

This section presents the structure of the *Information Technologies* study programme and covers semesters that implement the PBL modules. The section lists learning outcomes of the PBL modules. The programme belongs to the study field of Computer Science due to the national study classification.

3.1. *PBL Modules in the Study Programme*

Problem-based learning projects (PBLP) are integrated into two semesters of the Information Technologies study programme. Both semesters are an integral part of the second year of studies, and it is proposed as a freely elective specialisation. The structure of the programme is presented in Fig. 1. The study programme contains seven semesters. During the first year (semesters I and II) all study subjects are worth 5 credits (represented by squares). After the first-year, students can choose the specialisation of the programme. The specialisation lasts two semesters (III and IV). The specialisation includes PBLP of 15 credits (PBLI and PBLII) and three subjects of 5 credits as PBLP follows the idea of PBL Aalborg model. Afterwards, PBL learning style can be noted in semester projects (S), professional practice (PP), and final bachelor thesis (BSC). During the PBLI students have a lot of freedom to choose the solution of the problem while Management

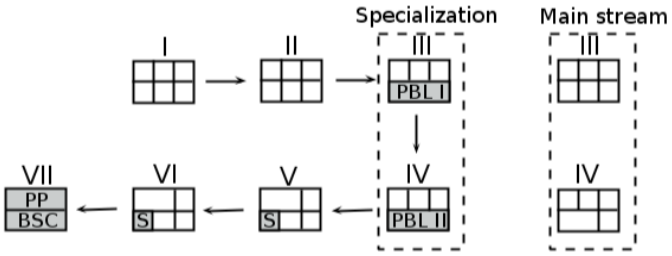

Fig. 1. Structure of the study programme.

Systems PBLII emphasises human-computer interaction, web technologies, and applications on the client-server architecture. This paper considers only PBLII project.

During PBLII project students must consider knowledge and skills obtained during some specific subjects to solve the problem and present the solution professionally. Table 1 presents lists of subjects that are integrated into projects. The first column contains subjects from the previous semesters (number identifies the semester), and the second column lists the subjects of the fourth semester. The subjects in grey must be reflected. During the fourth semester, Web Technologies and Human-Computer Interaction (HCI) subjects are mandatory. Thus, they must be reflected in the projects. Database Management Systems, Basics of Virtualization, and Software Engineering are taught during the third semester and make the basis for Three-Tier or Two-Tier client-server software architecture development. Any other knowledge, typically, is used indirectly. E.g. UNIX skills are needed to set up a virtual machine, or object-oriented programming skills are required to program in PHP or Java.

During the PBLII semester, students work in teams of 2–3 students and create solutions within various disciplines of their choice. The topics are suggested by the teaching staff and by the partners (other university departments or external industrial partners). For example, students developed the Epilepsy Register and Medical Genetics Information System in collaboration with staff from the Faculty of Medicine, Bone Fossil System with researchers of Faculty of History and Faculty of Medicine, Student Academic System Module for the faculty information system, and Park Tourist System with industrial partners. While the Publication Review System, Time Management System, University Game (SimTower style) were suggested by the teaching staff of the PBLII module.

Table 1

Subjects that are considered during PBLII

Previous semesters	Current semester
Data structures and algorithms (2)	**Human-Computer Interaction**
Database management systems (3)	**Web Technologies**
UNIX operating systems (2)	Optional Course
Basics of Virtualization (3)	
Software Engineering (3)	
Object-Oriented Programming (2)	

3.2. *Competencies and Learning Outcomes*

The PBLII module develops a list of competencies defined in the description of the study programme. The programme was an illustrative example in the National Tuning project (TL09-12) for the computer science field. The competencies were defined and updated several times during the projects (IISPDCS II, AMIPA) and before the accreditation (ERITSP2013). The committee of the study programme defined the competencies and learning outcomes, as well as their mapping to subjects. Based on Vilnius University regulations, it is study program committee's responsibility to ensure study quality and to support the description of competencies for the study programme.

The competencies of the programme are distinguished into generic (G) and subject-specific (S) as Vilnius University defined recommendations for such distinction in the Tuning project (TL09-12). The study programme has seven generic and nine subject-specific competencies. PBLII develops six for each group. Also, each competency is associated with one or several learning outcomes (LO). The list of competencies and discussion of learning outcomes is not in the scope of this paper, and a full list of learning outcomes is not provided. The used notation is adapted to the needs of the article.

Generic learning outcomes (GLO) to be developed in the PBLII module are as follows:

- GLO1 **Practical Thinking**. Ability to use existing theoretical models, terminology, recommended programming, modelling, and system administration principles and tools in applied sciences and everyday life.
- GLO2 **Communication.** Ability to explain own subject area and work specifics in Lithuanian and in English no matter what the profession of the listener is.
- GLO3 **Information Management and Systemic Thinking**. Ability to sum up and systematise received/provided requirements or the process of the solution of the task/work.
- GLO4 **Problem-Solving**. Ability to define a problem in application area and apply the the existing solutions.
- GLO5 **Usage of Information and Communication Technologies**. Ability to distinguish pros and cons of the software according to the properties of the user interface, support, installation, friendliness with other software; knowledge about the components of hardware (computer and network).
- GLO6 **Collaboration and Time Management**. Ability to plan own or group activities and ability to distribute tasks among group members based on task difficulty or according to the recommendations or comments of the expert.

The module does not develop ethical behaviour that is also a part of study program's generic competencies. Ethical sense is important in the course, but there is no specific emphasis on it. The generic competencies of the course are related to the team work on a specific task that is a close-to-reality situation. Thus, problem-solving, collaboration and a practical approach make the basis of a student professional development during the project.

The subject-specific learning outcomes (SLO) to be developed in the PBLII module are as follows:

- SLO1 **Software Design and Development: Programming.** Ability to write programs in the application area using programming languages of various programming paradigms (imperative, object-oriented, functional).
- SLO2 **Software Design and Development: Component Integration**. Ability to generalise the interface of the software, dependency from the other software or hardware and to provide the specification of integration.
- SLO3 **Computational Thinking**. Ability to present the algorithm using various techniques (pseudo-code, schema, etc.) for the given task or program; ability to implement others' algorithms; ability to formulate the task in different levels of abstraction.
- SLO4 **Production of Software Documentation.** Ability to write software (service) specification and user manuals.
- SLO5 **Testing**. Ability to define the testing environment and testing requirements for the projects of the application area, to create a testing scenario and to automate testing partially.
- SLO6 **Project Management**. Ability to apply project management standards participating in the project work.
- SLO7 **Data Management**. Ability to apply traditional data structures and modelling methods in the application area.

The discussed module does not develop software support, computer network construction, and security management learning outcomes directly. Thus, they are not assigned to the module. The information and infrastructure security is considered in the course, but with no specific emphasis. The module develops competencies using a practical approach.

The description of the course has the list of learning outcomes that reflect the assigned competencies and learning outcomes of the programme. Each subject-specific learning outcome of the study programme is related to one or more learning outcomes of the module. The learning outcomes of the module were defined by the module coordinators and approved by the committee of the study programme. The programme committee members, as well as coordinators of PBL modules, have experience in PBL Aalborg style from both, student and teacher perspectives. Some learning outcomes of generic competencies are integrated into several LOs of the module. The mapping of module LOs to study programme LOs is presented in Table 2.

In Table 2, LO1 is defined as an ability to understand the essence of the problem, to distinguish and analyse the requirements and restrictions, to find the existing solutions and to organise them, to foresee the possible solutions or the use of the existing solutions; ability to solve problems by applying knowledge in practice. LO1 is related to problem-solving ability (GLO4). The problems that are presented for students require practical thinking, prior knowledge of existing solutions and tools (GLO1), ability to systemize the received information (GLO3). This LO stresses the importance of problem-solving in a more theoretical level without emphasis on the practical skills. Thus, it is closely related to software design (SLO1).

Table 2

Mapping of module learning outcomes to learning outcomes of the study programme

	GLO1	***GLO2***	***GLO3***	***GLO4***	***GLO5***	***GLO6***	*SLO1*	*SLO2*	*SLO3*	*SLO4*	*SLO5*	*SLO6*	*SLO7*
LO1	**x**		**x**	**x**			x						
LO2		**x**								x			
LO3						**x**						x	
LO4					**x**		x	x		x			
LO5	**x**	**x**	**x**						x				
LO6			**x**								x		
LO7	**x**		**x**										x

LO2 is defined as an ability to present ideas, explain problems and their solutions fluently, clearly, and in detail in written or in oral form. It represents generic competency of communication. The student must be able to present problem-solving case in oral and written form (GLO2), including software documentation (SLO4).

LO3 is defined as an ability to work in a group, participate in the planning of the group activities, take responsibility for group work, show the initiative to distribute group tasks. It also includes the ability to carry out the individual or group tasks on time and knowledge of the main principles of project management as well as the ability to use version control systems. LO3 reflects teamwork, effective collaboration, and time management (GLO6), as well as project management (SLO6). During the IT project, specific tools must be used to demonstrate each team member's input to the project, e.g. version control systems.

LO4 is defined as an ability to implement the programming part of the project in the programming language(s) chosen by the group, ability to generalise the interface of own or used software and dependencies on other hardware or software; ability to write user manuals. LO4 is a very subject-specific learning outcome. It covers software design and development (programming and integration) with documentation preparation (SLO1, SLO2, SLO4). This LO requires the ability to use information and communication technologies (GLO5). The LO stresses problem-solving from a very practical approach.

LO5 is defined as the ability to present the algorithmic solutions in pseudo-code or schemas, to explain them, and to evaluate their correspondence to the programming part. LO5 learning outcome combines computational thinking (SLO3) with communication abilities (GLO2) for practical and systemic thinking (GLO1 and GLO3). The algorithmic solution must be chosen and implemented, as well as correctly presented with the argumentation related to existing theories and requirements.

LO6 is defined as an ability to foresee test cases of own software, to set and implement them. It is mapped as a Testing learning outcome (SLO5). To define testing cases the systemic thinking must be involved (GLO3).

LO7 is defined as an ability to choose the suitable data model and the ability to apply standard data structures. Thus Such this learning outcome represents the ability to define a data model (SLO7). Naturally, data modelling is influenced by some prior requirements, and, information must be managed correctly (SLO3).

4. Methodology

This section presents the PBL model to evolve from the requirement-based assessment to competency based assessment. We designed the schema and timeflow of activities that enabled development and assessment of learning outcomes with a strong emphasis on generic competencies.

4.1. *Module Activities*

The PBL module was designed to simulate real software development project. Modern IT teams use the AGILE methodology to follow the status of the project. Weekly sprints with what was done, what is under development, and what will be done, are discussed. To some extent some module activities reflect such methodology but with no expressed and regular sprints. The final result of the PBL project is the report and the programming part as a software product.

During the module, a set of activities is organised. Fig. 2 presents the time flow of activities. The time flow was divided into 16 weeks as one semester of the Vilnius University. Thus, projects have a rather long period as recommended in Sabin *et al.* (2017). The first week of the PBL module has no activities as students must start with other parallel subjects to get understanding how their material or activities can be integrated into the PBL module. The final week is dedicated to final touches of the project: report and programming.

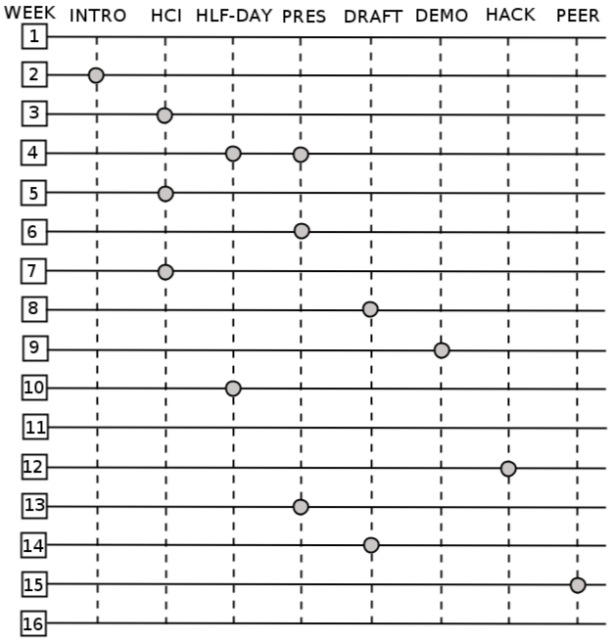

Fig. 2. Time flow of activities.

The types of activities are an introductory seminar, specific seminars (technological skills, use cases), half-day workshops, presentations, submissions of drafts, demonstration seminar, hack-the-project, and peer-review. This model assumes HCI seminars as specific seminars, but they can be replaced with other computer science topics.

The introduction to the module, project market, group formation, and work plan are made during Week 2 seminar. PBLII follows PBLI from the previous semester. Thus, the introduction seminar covers only module requirements, plan of the semester, and project topic descriptions. The PBL learning style is covered in a semester before PBLI (details are presented by Brilingaitė and Bukauskas, 2017).

HCI seminars with case studies, making of prototypes, etc. are planned for the first part of the semester, weeks 3, 5, and 7. Half day workshops (HLF-DAY) are organised to foster the start of the project (analysis and possible directions) and to foster programming tasks during weeks 4 and 10, respectively.

Presentations of groups enable the regular status check of the group project. The first two presentations are organised during the first part of the semester. They are dedicated to the analysis of the problem and design of the software. The last presentation can already cover very specific implementation details. As groups are formed of 2 to 3 students, groups that have a pair of students use this seminar just for the status check and can choose when to make a presentation.

Drafts of the reports are submitted on weeks 8 and 14. In the middle of the semester, there is an important status check of the project. The first draft is submitted, and the demonstration seminar is organised afterwards. On week 12, the hack the project activity shows the development phase of the project. Each student could try to hack some project and report findings. The second significant status check is close to the end of the semester. The second draft is submitted to get much feedback from the colleagues and supervisors. Thus, the peer-review is done in week 15 after the second draft is submitted.

4.2. *Assessment Structure*

We designed the evaluation strategy that includes individual and group assessment. The summative assessment and formative assessment types are applied during the semester. The assessment process is spread during the semester and the exam day. Fig. 3 illus-

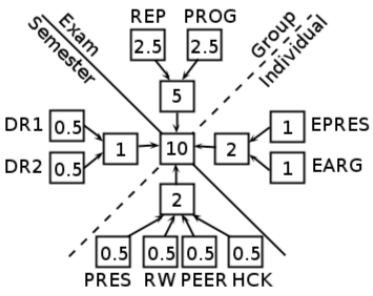

Fig. 3. Point distribution among assessment elements.

trates the point distribution among elements of the summative assessment. To observe the development of student competence and the achieved level, the formative assessment is applied during the activities that are not included in the final evaluation. The total sum of points is 10. The solid line from the left-top corner separates exam points from the points earned during the semester, whereas, the dashed line from the right-top corner separates group and individual assessment activities, respectively.

4.3. *Individual Assessment*

During the semester four individual assessment elements are formed. All elements are equal, i.e. 0.5 points.

Oral Communication. Once per semester, each student must present the group project based on the current stage. The presentation (PRES) includes preparation and upload of the slides. Also, once per semester, each student must review (RW) presentations during the presentation seminar. Each group assigns delegates for the presentation and review activities for each seminar. Students must be able to prepare slides using professional dictionary, tools, visual elements, and reflect the current situation of the project.

Written Communication. There are two individual tasks related to personal written communication. Each student must write a peer-review (PEER) of another group's report. Also, each student must present the hack-the-project report (HCK) with her findings. Students must be able to distinguish advantages and disadvantages of projects, argument missing, unnecessary, or advanced solutions, principles, etc. The dictionary must be professional, the flow of thought must be logical and consequent.

During the exam two evaluations are set up. Each part is worth 1 point. During the exam day, each student is evaluated individually. The student must be able to prepare and integrate slides nicely into the group's presentation. One point is related to slide quality and presentation (EPRES). Another point covers the defence of the project (EARG). During the defence, the student must be able to make arguments, discuss and substantiate points of view.

4.4. *Group Assessment*

During the semester, two drafts (DR1 and DR2) of the report are uploaded and evaluated. Each draft evaluation is based on the current project stage. Each part is worth 0.5 points. Skills of the professional communication are evaluated, as well as the ability to reach the required level in the project stage.

The final result of the project is the report (REP) and the programming part (PROG). Each part is worth 2.5 points. The programming part includes programming code, installation guide, binary executable project. The evaluation is presented at exam time. The criteria for the report are the same as for other written parts, but the final project report must cover the whole project and must include all parts from the introduction

with a motivation and problem description to the implementation with testing phase via analysis and design. The final programming part is a prototype that can be accessed, installed, and that includes the planned functionality. The report is assessed based on several criteria: form, structure, and fluency, an organisation of the related work, citations, complexity, clearness, practical part, and significance of the problem solution.

5. Results

This section presents the three-year observation of student performance during the implementation of the PBL model. The model enabled mapping of LOs to activities. To justify the level of students, the LOs were mapped to several taxonomies.

5.1. *Observation*

The PBL model of this paper was tested three times within the study programme Information Technologies. Table 3 provides statistics on a total number of students and a number of failed students with the reasons of underachievement.

In Fig. 4, the assessment of intermediate drafts and the final version of the report is presented in percentage ranges. The data series is an average for each year from all students where each assessment element is scaled from 0 to 100. It should be noted, that drafts were group activities and students are evaluated as a group whereas final report is evaluated as a group and only with exceptions a few deviations are individual.

In Fig. 4, one can see interesting fact that in 2015 and 2016 intermediate drafts were evaluated in average better than the final report at the end. At the same time years, 2015 and 2016 were having a much higher evaluation of reports than in 2017. The reasons were that students did not have clear guidelines in advance for draft versions to be expected and students tried to postpone the work till the latest report. As organisers of the PBLII we noticed that students right from the beginning should receive very clear way recommendations for each draft version and in 2017 we have successfully modified descriptions, requirements and helpful comments of intermediate reports.

In Fig. 5 the average performance of students during the final exam with assessment elements and the final grade is shown. It should be noted that the final grade consists of final assessment elements during the final exam and the performance during the semester. As one can observe for implementation of the project mostly subject-specific competencies are evaluated. It is very clear what to evaluate and each year students have a very similar background when working on PBLII projects. Thus, the assessment of the implementations does not differ very much. However, the percentage of a larger number of failing students (see Table 3) pushes back the average of all implementations and final grade. On the other hand, presentation during the exam and the demonstration of the actual project implementation is mostly showing generic competencies.

Table 3
Number of underachievements

	# of students	# of failed students	Reasons for underachievement
2015	19	2	Students were identified with an insufficient contribution to the team project. As a group performance was satisfactory, members of teams did not provide any cover-up for burden students.
2016	16	1	The student did not engage in the teamwork even after encouragement. Contribution to the project was nil. Early identification of tailing students and the right communication to the teams by supervisor helped to compensate the loss of the member.
2017	21	5	Students did not develop sufficient skills neither process nor result was satisfactory for the assessment. Those failed students were identified by the group members and requested to be separated as burden or latecomers who ignored group work. In each case, a cycle of negotiations was done to mitigate the situation.

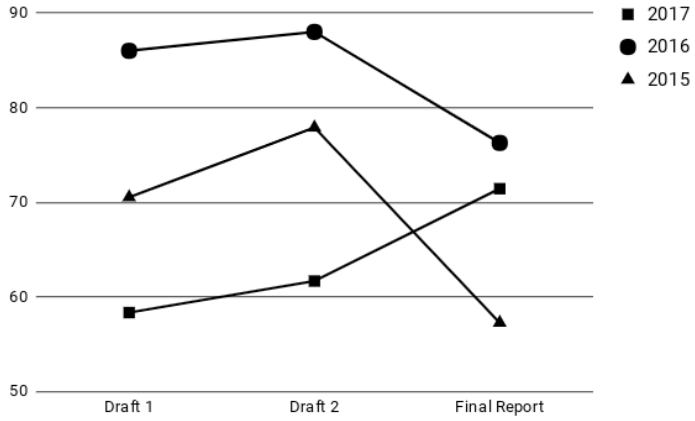

Fig. 4. Assessment of the final project report and intermediate drafts versions 1 and 2.

In general, when defending the project, if lagging students burdened a team then it might fall behind the implementation. If we remove failed groups out of the equation, successful teams outperform our expectations, see Fig. 6.

The three-year observation of student results showed that introduction of guidelines for the intermediate assessment was very beneficial, leading students to produce much higher quality reports and software. Also, it showed that burdened students in our model cannot hide behind the strong group members and are identified very early. Our problem-based learning model ensures evenly spread workload for students with correctly planned activities and assessment strategies.

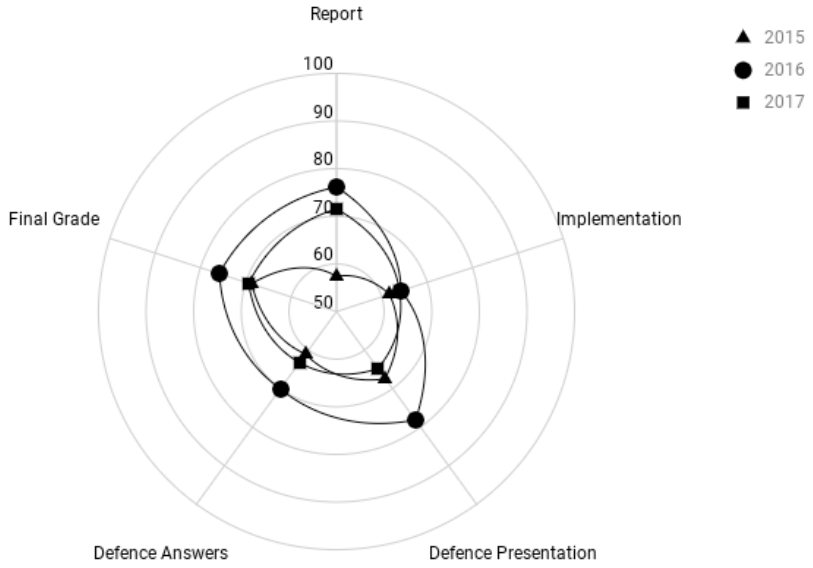

Fig. 5. Average performance of assessment elements during the final exam.

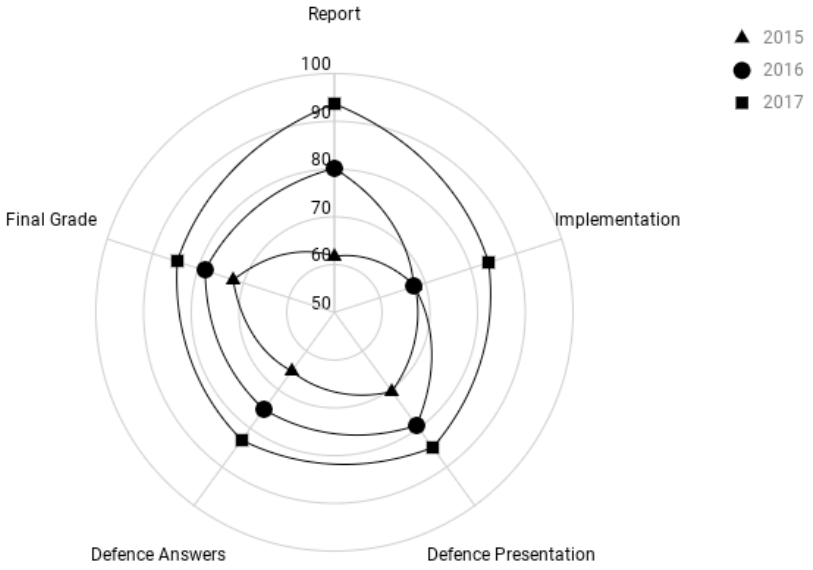

Fig. 6. Average performance of assessment elements during the final exam with failing groups removed.

Table 4
Mapping of LOs to assessment elements

	DR1	DR2	REP	EARG	EPRES	PRES	RW	PEER	HCK	PROG
LO1	x	x	x		x	x				
LO2	x	x	x	x	x	x	x	x	x	
LO3			x		x					x
LO4	x	x	x			x				x
LO5		x	x	x	x	x	x			
LO6			x		x			x	x	
LO7		x	x	x	x			x		x

5.2. *Mapping of Learning Outcomes to Assessment Elements*

Each learning outcome of the PBLII course was mapped to the activity of the proposed PBL model. The mapping done was based on the previous experience during PBLI and PBLII by the coordinators of the PBLII module as the core teaching staff of both modules is the same.

Table 4 represents the mapping of module LOs to the assessment elements. As LO1 is related to problem-solving skills at a theoretical level it was mapped to drafts, final report, and presentations. LO2 is the communication LO. It was mapped to all assessment elements except programming because most elements require the ability to communicate in written or oral form. For example, a peer review must be clear and includes suggestions for the reviewed group. The hack-the-project task must be well structured to be informative and repetitive. How well students work in a group, how they manage their project (LO3) is assessed by the report (e.g. based on meeting notes), exam presentation (task load graphs), and programming (version control system reports). Drafts, report, and programming part represent how students design, program, and document their software (LO4). It is a group responsibility, but during presentations, students must include project design elements. Computational thinking (LO5) must be represented in the second draft, final report, and all presentations, as well as during the review. Testing is the final stage of the report. Thus, testing related skills (LO6) are assessed in the report, exam presentation, peer review, and hack-the-project task. Report and exam presentation helps to understand group creativity for testing cases. The peer review and hacking tasks could show how well students understand testing types and cases for other groups. Data management is assessed in the second draft, final report, final presentation, programming elements, and during peer review as students must understand data models of other groups.

5.3. *Mapping to Taxonomies*

Bloom and SOLO

The competencies and learning outcomes of the study programme and PBLII study subject were defined concerning Bloom taxonomy. Learning outcomes of PBLII refer to

the *Application* level. It is assumed that without *Knowledge* there could be no *Understanding*, and *Application* is not possible without *Understanding*. In IT field, Knowledge and Understanding are cognitive skills while Application is a practical aspect: ability to implement using some programming language, ability to use existing libraries in the software, ability to integrate some module into the existing system, ability to deploy the developed software on a virtual machine (create a machine, setup configurations, and install/prepare software). However, during the defence of the PBLP, it is clear that students developed non-trivial specific software, but they were not able to explain how the web framework ensures secure login to the system or to list essential properties of PHP programming language. Student projects showed that some project groups developed even higher levels (Analysis, Evaluation) even if all groups build a software product (Application). There are several examples of students going beyond the expectations for the level of the LO with the observation statements:

- … implement the programming part of the project (LO4).
  Students have estimated the highest load of the system, prepared the system for effective paging, evaluated system load with a large amount of synthetic data. (Evaluation)
- … choose the suitable data model (LO7).
  Students have analysed example data and discovered a set of data entities with their relationships. The data model contains more than 20 entities with various relationships as well as detailed log information about data changes in the system. (Analysis)
- … define and implement test cases (LO6).
  Students prepared the survey to get user feedback, found users (outside the university, of various professions), analysed the comments and evaluations, drew conclusions, and made system modifications based on the results. (Analysis)

Based on three years' experience, it is very difficult to make the distinction between students that achieve higher level competencies and those that do not. If a group satisfies the requirements and achieves the learning outcome the assessment is high, and those who go higher cannot get higher assessment marks as the assessment is not proportional.

If IT competencies are defined concerning SOLO taxonomy it is clear that in many cases a practical approach comes before the cognitive one. For example, students can apply the object-oriented programming principles without knowing the keystones of the paradigm and even without understanding how the principles work. The LOs of PBLII module are at the Qualitative phase in the Relational level from the practical perspective. Students must create functional software; thus, they combine various technologies and apply knowledge to the final product. Separate and scattered sub-elements that are not combined into one entity lead to a failed exam: even if documentation comprises a large part of the assessment the good report cannot be made having a good analysis, design, development results. Even a small project contains all required elements, and again, there is a problem of defining the criteria for the same level of the taxonomy.

Due to the PBL nature, the assessment criteria cannot be defined based on a very specific computer science field or the technology in mind. Projects are very different. The application area could be very specific, e.g., medical genetics, and defined problems

with solutions could be very diverse. Thus, the criteria cannot include a number of UML diagrams or the requirement to draw the entity-relationship diagram for the database model. The essence is how and in what ways the problem with its solution is presented and how this is done in a team.

Understanding by Design

The attitude is also essential for the IT student as direct satisfiability of the requirements could lead to misunderstood competency. One of LOs of the PBLII is related to teamwork. The example presents the excerpt of LO3 and the case how an effective group work is not convincing and not seen clearly by the module coordinators and supervisors. The example includes the student observation by the research team.

- ... work in a group, responsibility for group work, carry out the individual or group tasks on time; knowledge of the main principles of project management; ability to use version control systems (LO3)

  Students frequently meet in some place outside the university. They do programming in pairs or even in 4-tuples altogether by extreme programming style. The version control system shows regular and sizable commits of one member, one member seems to commit from time to time, and the other two students have only several low-value commits.

One of the requirements in PBLII is to use some version control system. Usually, students follow the requirement in order not to lose the current version of the project (programming or documentation). But version control systems store all the historical data. Each commit to the system is recorded. Each version of the particular file is recorded and can be compared to the other previous versions. Thus, the supervisors and module coordinators can check students' activities, times of activity and nature of contribution to the project. Extreme programming on one computer (one account) leads to the unseen input of some colleagues even if from the perspective of students, the teamwork is effective. Thus, understanding of version control system power is poor, and the requirement is not understood correctly.

Recommendations for curricula of undergraduate information technology study programmes focus on the transfer of learning and targeted understanding. They suggest using understanding by design framework (UbD) that is based on understanding revealed in action (Sabin *et al.*, 2017). This perspective makes allowance for the combination of knowledge, skills, and attitude. The UbD has six facets: Explanation, Interpretation, Application, Perspective, Empathy, and Self-Knowledge. If UbD is followed, then the PBLII LOs could have more diverse levels. LO1 (find and use existing solutions) is associated with *Self-Knowledge* facet as students must be able to recognize similar problems and/or similar solutions as well as to be aware of what they do not know or do not understand. LO2 (present and explain problems and solutions clearly in oral and written form) is associated with *Perspective* as students must demonstrate a perspective, a broad-view by seeing a big picture. Students must express possible ways and argue for the chosen solution. Also, students must be open to suggestions of supervisors and colleagues. LO3 (take responsibility for group work, show the initiative) is associated with *Empathy* as students must assume how others feel in a group, show empathy, and play a particular role in a group. LO4 (implement using programming languages, generalise

interface and dependencies, write manuals) is associated with *Application* as students have to apply knowledge in a unique situation that goes beyond the typical cases. LO5 (present algorithmic solutions) is associated with *Perspective* as algorithms must cover various boundary cases as well as they must be evaluated for correct and effective working. LO6 (foresee test cases) is associated with *Empathy* as students must try to walk in the shows of the user of their system. Some test cases might be even not logical, but still, they must be covered. LO7 (choose suitable data model, apply standard data structures) is associated with *Perspective* as a big picture must be seen - one decision can influence further steps or effectiveness, reusability.

UbD enables distinction of assessment criteria concerning the level in *ability to apply.* If a student does not see a broad picture (perspective) for the algorithmic solution, then it might be that some solution is applied even without an open-view, etc.

Marzano and Kendall's New Taxonomy

New Taxonomy has six levels, and each of them has a number of operations (Marzano and Kendal, 2008). Mapping of PBLII LOs to Marzano levels showed that all LOs are interrelated, and only the whole set of LOs showed the level of student competencies.

Level 1 is Retrieval concerning information recognition and information validation. It is included in all LOs and activities of the course. Level 2 is Comprehension with Integration and Symbolization operations. Students are expected to integrate knowledge from previously attended courses and self-obtained knowledge into the projects, as well as to use standard representations for diagrams and algorithms. This level applies to all LOs. Level 3 is Analysis, and it contains five operations: matching, classifying, analysing errors, generalising, and specifying. Students of PBLII are expected to find similar problems and existing solutions, distinguish categories of related work, analyse their own and other project errors (hack the project, peer-review), generalise the interface of the software, write specifications. Testing LO also requires analysis as some testing is done by programmers themselves and backtracking must be analysed. Level 4 is Knowledge Utilization, and it includes decision making, problem-solving, experimenting, and investigating operations. PBLII students must make decisions on technology usage, application of existing solutions, and usage of standard data structures. Problem-solving is the essence of the PBL project. But students do need to generate and test a hypothesis. Investigation operation could be conducted during the hack-the-project activity and during analysis of this activity's results to have correspondence among the specification, design, and implementation.

Levels 5 and 6 are less technical concerning IT competencies. Level 5 is Metacognition. It contains operations such as specifying goals, process monitoring, monitoring clarity, and monitoring accuracy. This level is closely related to project management and work in a group. Students must be able to define the goal (problem definition, LO1) and to monitor the status of the group project (LO3). PBLII module assessment criteria do not require full meeting notes, detailed task status presentation, thus, this level could be integrated into the PBLII module LOs partially. Level 6 is *Self-System Thinking* that means student understanding of competencies, clear motivation, and self-evaluation. This level was not considered in this article as we observed student *believes* in competency improvement very informally.

6. Conclusions and Future Work

Early exposure of a student to the style of work and close-to-reality projects demonstrates competencies and skills required by the industry. However, student centred problem-based learning should not and cannot leave the student alone to self-control and self-motivation. Our approach allows the PBL project to be extended with activities to ensure student competency development and the intermediate assessment. By following the recommendation for IT curricula to include communication activities our model enables early feel of the IT profession of other business areas. Analysis of the learning outcome mapping shows that typical taxonomies are not sufficient to define IT LOs. But if used altogether they justify the achieved competency levels by individual students appropriately.

Shortly, we will further work on student knowledge transfer activities to make students aware and self-aware of the reasons behind the activities. We are planning to investigate options to involve the future students in the design of new activities decided upon the group to allow peer and self-validation. The PBL modules could further be enhanced to strengthen communication with external entrepreneurs and stages of final intellectual property management.

References

AMIPA – Renewal of study programmes and deployment of new study programmes of the first and second cycles in the study fields of informatics and informatics engineering, 2010–2013. European Social Fund Agency page: http://www.esparama.lt/paraiska?id=22708&order=&page=&pgsz=10

Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., Wittrock, M.C. (Eds. 2001). *A Taxonomy for Learning and Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Addison Wesley Longman, Inc.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). *Taxonomy of Educational Objectives: Handbook 1 Cognitive Domain*. Longmans, Green and Co Ltd, London.

Brilingaitė, A., Bukauskas, L. (2017). Application of Problem-Based Learning in Education of Computer Science: Case Study in Vilnius University, INTED2017 Proceedings, 2791–2800.

Chaves, J.F., Baker, C.M., Chaves, J.A., Fisher, M.L. (2006). Self, peer, and tutor assessments of MSN competencies using the PBL-evaluator. *Journal of Nursing Education*, 45(1).

Dolog, P., Thomsen, L.L., Thomsen, B. (2016). Assessing Problem-Based Learning in a Software Engineering Curriculum Using Bloom's Taxonomy and the IEEE Software Engineering Body of Knowledge. *ACM Transactions on Computing Education (TOCE)*, 16(3), 9.

Duch, B.J., Groh, S.E., Allen, D.E. (2001). Why problem-based learning. *The power of problem-based learning*, 3–11.

ERITSP2013 – The Centre for Quality Assessment in Higher Education Roland, N. Ibbett, Philippos Pouyioutas, Jürgen Dorn, Aleksej Kovaliov, Justinas Petravičius, *Evaluation Report of Information Technologies (612i10003) Study Programme*, Vilnius 2013.
http://pluto.skvc.lt/_layouts/ListAttachment.aspx?Attachment=Lists%2fPublicUnderwayStudyProgram%2fAttachments%2f1237%2fVU_Informacin%c4%97s_technologijos_BA_2013.pdf

Fuller, U., Johnson, C.G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Thompson, E. (2007). Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4), 152–170.

Hawthorne, E. K., Campbell, R. D., Tang, C., Tucker, C. S., & Nichols, J. (2014). Information Technology Competency Model of Core Learning Outcomes and Assessment for Associate-Degree Curriculum.

Hendry, G.D., Frommer, M., Walker, R.A. (1999). Constructivism and Problem-based Learning. *Journal of Further and Higher Education*, 23(3), 369–371, DOI: 10.1080/0309877990230306

Hmelo-Silver, C.E., Barrows, H.S. (2006). Goals and strategies of a problem-based learning facilitator. *Interdisciplinary Journal of Problem-Based Learning*, 1(1), 4.

IISPDCS II – Increasing Internationality in Study Programs of the Department of Computer Science II (orig. Kompiuterijos katedros studijų programų tarptautiškumo didinimas), project number VP1–2.2–ŠMM-07-K-02-070, funded by The European Social Fund Agency and the Government of Lithuania, 2012–2014. European Social Fund Agency page: http://www.esparama.lt/paraiska?id=32507&order=&page=2&pgsz=10

Johnson, C.G., Fuller, U. (2006). Is Bloom's taxonomy appropriate for computer science?. In: *Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling 2006*. ACM, 120–123.

Koch, J.H., Teege, G. (1999). Problem Based Learning in Computer Science. In: *Proc. of 2nd International Conference on New Learning Technologies*. Bern, Switzerland. http://citeseer.nj.nec.com/369966.html

Letchumanan, M. (2009). Problem based learning in mathematics. *Math Digest: Research Bulletin Institute for Mathematical Research*, 2(2), 21–25.

Letelier, M.F., Herrera, J.A., Canales, A.M., Carrasco, R., López, L.L. (2003). Competencies evaluation in engineering programmes. *European Journal of Engineering Education*, 28(3), 275–286, DOI: 10.1080/0304379031000098247

Markus, L., Thomas, H.C., Allpress, K. (2005). Confounded by competencies? An evaluation of the evolution and use of competency models. *New Zealand Journal of pschology*, 34(2), 117.

Marzano, R.J., Kendall, J.S. (Eds.) (2008). *Designing and Assessing Educational Objectives: Applying the New Taxonomy*. Corwin Press.

Maudsley, G. (1999). Do we all mean the same thing by "problem-based learning"? A review of the concepts and a formulation of the ground rules. *Academic Medicine*, 74(2), 178–85.

McCaughan, K. (2015). Theoretical anchors for Barrows' PBL tutor guidelines. In: A. Walker, H. Leary, C. Hmelo-Silver, P. Ertmern (Eds.), *Essential Readings in Problem-Based Learning* . West Lafayette, IN: Purdue, 57–68.

Meerbaum-Salant, O., Armoni, M., Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239–264.

Nelson, E. (2010). Elements of Problem-Based Learning: Suggestions for Implementation in the Asynchronous Environment. *International Journal on E-Learning,* 9(1), 99–114.

Prince, K.J., Van Eijs, P.W., Boshuizen, H., Van Der Vleuten, C.P., Scherpbier, A.J. (2005). General competencies of Problem-Based Learning (PBL) and non-PBL graduates. *Medical education*, 39(4), 394–401.

Sabin, M., Alrumaih, H., Impagliazzo, J., Lunt, B., Zhang, M., Byers, B., Newhouse, W., Paterson, B., Peltsverger, S., Tang, C., Veer, G., Viola, B. (2017). *Information Technology Curricula. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology*. A Report in the Computing Curricula Series (Report Version 0.85, April 20 2017). Association for Computing Machinery (ACM), IEEE Computer Society (IEEE-CS)

Savery, J.R., Duffy, T.M. (1995). Problem based learning: An instructional model and its constructivist framework. *Educational technology*, 35(5), 31–38.

Savery, J.R. (2015). Overview of problem-based learning: Definitions and distinctions. *Essential Readings in Problem-Based Learning: Exploring and Extending the Legacy of Howard S. Barrows*, 5–15.

Starr, C.W., Manaris, B., Stalvey, R.H. (2008). Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. *ACM SIGCSE Bulletin*, 40(1), 261–265.

TL09-12, *Development of the Concept of the European Credit Transfer and Accumulation System (ECTS) at the National Level: Harmonization of the Credit and Implementation of the Learning Outcomes Based Study Programme Design*. Vilnius University and Tuning Association (No VP1-2.2- MM-08-V-01-001) in 2009–2012. Retrieved from http://tuningacademy.org/tuning-lithuania/?lang=en

Torp, L., Sage, S. (2002). *Problems as Possibilities: Problem-Based Learning for k-16 Education* (2nd edition), ASCD.

World Economic Forum (WEF). (2016). The future of jobs: Employment, skills and workforce strategy for the fourth industrial revolution. World Economic Forum, Geneva, Switzerland.

Xian, H., Madhavan, K. (2015). *A Scientometric, Large-Scale Data, and Visualization-Based Analysis of the PBL Literature*. Essential Readings in Problem-based Learning, 281.

A. Brilingaitė holds a PhD in computer science from Aalborg University, Denmark. She is an assistant professor at Vilnius University in the Institute of Computer Science. Her research interests focus on spatial data modelling, location-based services, cybersecurity training, and education in computer science. She is involved in the processes of quality assurance in studies at the university. She has been taking part in EU-funded projects related to the development of student-centred learning, teaching, assessment, and internationalisation.

L. Bukauskas is a PhD in computer sciences from Aalborg University, Denmark. He is an associate professor at Vilnius University in the Institute of Computer Science. His research interests focus on Data Mining, Cyber Security and Digital Forensics. He is also the head of the Information Technologies study program that was awarded honorary award of Investors' Spotlight in 2017. Also, he led the national team within the study field of Computer Science in the development of the concept of the European credit transfer and accumulation system (ECTS) at the national level: harmonisation of the credit and implementation of the learning outcomes based study programme design.

A. Juškevičienė is a doctor of technological sciences (informatics engineering). She is the researcher at the Vilnius University Institute of Data Science and Digital Technologies. The areas of her scientific interest focus on technology enhanced learning, intelligent and adaptive systems, recommender systems, semantics and ontology, evaluation of quality of learning software and learning process. She has been working very active on several national projects, helps to organize seminars and conferences. She has published a number of scientific papers and publications in popular magazines, participated in a number of large scale EU-funded R&D projects.