2016 4th International Conference in Software Engineering Research and Innovation

Developing Software Engineering Competences in Undergraduate Students: A Project-Based Learning Approach in Academy-Industry Collaboration

Reyes Juárez-Ramírez, Samantha Jiménez, Carlos Huertas
School of Chemical Science and Engineering
Autonomous University of Baja California
Tijuana, Baja California, México
{reyesjua, samantha.jimenez, chuertas }@uabc.edu.mx

Abstract—**Project-based learning approach is focused in preparing students to work in industry. In traditional academic environments, it is difficult to reproduce real scenarios such as in industry. Software Engineering courses require practice and learning hands-on; however, with such limitations in academy, it is not easy to develop the competences to work in real world practice. In this paper we present our experiences working with undergraduate students in real projects in collaboration academy-industry. This environment allows us to develop Software Engineering competences to students. We assessed how our students get the set of competences suggested by the Software Engineering Education Knowledge standard, having encouraging results for a Computer Engineering undergraduate program.**

Keywords-software engineering; project; competences, academy; industry;

I. Introduction

Many definitions of engineering suggest that engineering is the profession where principles of science and mathematics are applied to obtain solutions in the benefit of human beings [1, 2, 3, 4]. In [1] is emphasized that: "*Engineering is the process of designing the human-made world. Engineers modify the world to satisfy people's needs and wants*". ABET (www.abet.org) supports this statement emphasizing that the knowledge and principles can be gained by study, experience, and practice, and that they are applied with judgment to develop ways to utilize the materials and forces of nature for the benefit of mankind.

In traditional education practices and lecturing approaches, which are teacher-centered, concepts are learned without practicing their application. In those learning environment, the following situations are present [5]: (1) Students are usually faced with a vast amount of information to memorize, much of which seems irrelevant to their future needs outside of school; (2) students are unable to reason effectively; (3) students are unable to assume responsibility for their own education; (4) students are not prepared to work with others in collaborative team situations; (5) motivation in class is usually low; and (6) produce students who are disenchanted and bored with their education.

In order to acquire the engineer spirit, learning environments must be oriented to real world situations [6]. To address the problem solving and human oriented focus of engineering students, we need to generate a solid learning environment including techniques, methodologies and resources that empower students with the competences necessary to beat the world's problems. Approaches such as Problem-Based Learning (PbBL) [7, 8, 9], Project-Based Learning (PjBL) [10, 11], and Challenge-Based Learning (ChBL) [12, 13] by their nature are appropriate methodologies for engineering education. Furthermore, these approaches are directly related with the Competence-Based Education (CBE) approach [14, 15].

Project-based learning is a methodology that helps to prepare students for real life jobs. Projects expose students to the practical problems. For undergraduate levels, this approach represents a scenario for preparing graduates for industry [16], especially professionals of the 21st Century [17].

Software Engineering is considered one of the most difficult topics in computer science programs. Its difficulty is not like theoretical courses, such as algorithm analysis, nor programming courses, such as data structures. Software Engineering is an empirical course; students should learn Software Engineering methods through hands-on experience, which might include the following aspects [18]: *Real-world software development, real-world customer interaction, real-world planning and estimation, and real decision-making and problem solving*.

Project work may be included in any class, but it is fundamental to Software Engineering courses and senior-level capstone classes [19]. In the case of software development projects, this approach provides dynamic and unscripted learning opportunities. However, given the academy limited resources, without a pool of real projects with real customers, and without professional experienced professors, it is hard for students to learn hands-on experience in a classroom environment.

In our Computer Engineering undergraduate program, we have a specialization area in Software Engineering which is represented for a set of six core courses and another set of supporting courses. We teach these courses using the PjBL approach.

In this paper we present our experiences practicing the PjBL approach, working with projects in collaboration academy-industry in the Innovation Program funded by the Mexican National Council of Science and Technology (Spanish: Consejo Nacional de Ciencia y Tecnología "CONACYT"). We present a resume of the Software Engineering competences proposed by the Software

978-1-5090-1074-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CONISOFT.2016.22

Engineering Education Knowledge (SEEK) [20], which are acquired by our students when they participate in these projects.

This paper is organized as follows. Section 2 contains the basics of Project-Based Learning approach. Section 3 presents the related work. Section 4 presents the set of SEEK competences and the relation with Bloom's Taxonomy. Section 5 describes the nature of our projects in collaboration academy-industry. Section 6 exposes the results of the assessment of SEEK competences acquired by our students. Finally, section 7 contains the conclusions and future work.

II. Basics of the Project-Based Learning Approach

The main element of PjBL is a "project" as the entity to address the learning. This fact makes PjBL an appropriate approach for engineering education since a project, as a unit of work, is the basis of engineering activities in the real practice. This statement is supported by [10], where is argued that "*Almost every task undertaken in professional practice by an engineer will be in relation to a project*".

PjBL takes some basic elements from PbBL, such as a problem to resolve, problem understanding, hypotheses establishment, gathering information related to the problem to answer hypotheses, and proposing solutions. Besides this, PjBL incorporates real-life "challenges" where the focus is on authentic (not simulated) problems or questions and where solutions have the potential to be implemented [21]. In this approach, *projects are realistic, not school-like*; this means that projects are not extracted from a textbook, neither proposed by the teacher.

As the main entity, project gives identity to PjBL, so that, it is necessary to recognize the main characteristics of a project (see Table I). These characteristics allow us to emphasize on an engineering perspective of the problem solving approach, since we have a more systematic and structured scheme to perform the problem solving.

TABLE I. Main Characteristics of a Project

Project's characteristics
PJ-ch_01: The project involves a topic that gives it identity and authenticity.
PJ-ch_02: The project includes the tasks that students do.
PJ-ch_03: The project includes the roles that students play.
PJ-ch_04: The project includes the context within which the work of the project is carried out.
PJ-ch_05: The project specifies the products or outcomes that are produced.
PJ-ch_06: The project identifies the audience for the project's products.
PJ-ch_07: The project states the criteria by which the products or performances are judged.
PJ-ch_08: There is a time specification (or deadline) to finish the project.

In [22], seven features are suggested that are key components of PjBL (see Table II). These features can be used in describing, assessing, and planning for projects. Next we describe each one briefly.

PjBL-fe-01: Student's decision-making and initiative are enhanced throughout the project; feedback and continued assessment are key features to this component.

PjBL-fe-02: Collaboration is intended to give learners opportunities to learn collaborative skills, such as decision-making, interdependence, providing feedback to peers, and working with others as student researchers.

TABLE II. Features of the Project-Based Learning Approach

PjBL Features
PjBL-fe-01: Learner-centered environment
PjBL-fe-02: Collaboration
PjBL-fe-03: Curricular content
PjBL-fe-04: Authentic tasks
PjBL-fe-05: Multiple expression modes
PjBL-fe-06: Emphasis on time management
PjBL-fe-07: Innovative assessment.

PjBL-fe-03: Successful integration of content is achieved through projects. These projects are based on standards, have clearly articulated goals, and support and demonstrate content learning both in process and product.

PjBL-fe-04: Authentic tasks can take on many forms, depending on the goal of the project.

PjBL-fe-05: Learners get opportunities to effectively use various technologies as tools in the planning, development, or presentation of their projects.

PjBL-fe-06: Time management builds on opportunities for learners to plan, revise, and reflect on their learning; projects should include adequate time and materials to support meaningful doing and learning.

PjBL-fe-07: Assessment is an ongoing process, just like learning. PjBL requires varied and frequent assessment; including teacher assessment, peer assessment, self-assessment, and reflection.

Particularly for students, PjBL has five main advantages, which are [22]: (1) Increased motivation; (2) increased problem-solving ability; (3) improved media research skills; (4) increased collaboration; and (5) increased resource management skills. For the teacher, "Projects give an opportunity to shift the primary responsibility for learning and its assessment onto the students' shoulders" [23]. Making the shift so that students take the lead in assessing what and how they are learning enables the teacher to enrich their learning [24].

PjBL can be used in every educational level, from elementary school until undergraduate courses. In this paper we focus on the experiences using PjBL with undergraduate students in a computer science program.

III. Related Work

In [25] is argued that in engineering education, we have to "*empower students to analyze and solve real-world, human problems by integrating scientific, technological, business, and social aspects of these problems, and to communicate innovative solutions to a diverse audience*". Much effort has been oriented to redirect engineering education. Part of this effort is focused on curricula design and implementation [3, 26, 27, 28, 29, 30]. These works agree in that aspects such as attributes and competences (abilities, skills) for an engineer must be emphasized in the curricula creation. Several universities are working with this approach [31]; however, the *attitudes* and *values* that should characterize engineering students remain in lack. Our teaching experience suggests that attitudes and values commonly are not taught in classroom, they are acquired in real practice.

In [21], a proposal for implementing a pedagogical strategy is presented, in order to complement a Software Engineering course. In that study they take into account the knowledge areas (KA) proposed in SWEBOK and also squares up the educational objectives of Bloom's Taxonomy cognitive domain. Graduate students were considered to assess the proposed model.

In [33], the authors propose the addition to SWEBOK of a knowledge area pertinent to the organizational, individual and methodology-related aspects of Software Engineering. This additional area suggested as important and relevant for engineering areas, and for the overall success of a software project.

In the context of Software Engineering education and training, recent studies have addressed specific issues in acquiring skills to get ready for employment. In [34], a study is presented, emphasizing that project failure is due to the lack of Requirement Engineering (RE) practice, and that industry needs to allocate extra costs to send employees for additional training before they can contribute to the job specification. They proposed PjBL as a correct method for delivery mechanisms to enhance Software Engineering undergraduate skills particularly in RE. They also suggest that PjBL is a superset of practices from Problem-Based Learning, Individual-Collaborative Learning and Product-Based Learning.

Later, in [35] the authors present a study promoting the use of PjBL as a mechanism for educators in their task to assist students in enhancing their skills in RE. On the other hand, they identified challenges in the current practice of teaching Software Engineering at the undergraduate level in RE.

In [36], the authors present a proposal for a Software Engineering course for cultivating personal competences to undergraduate students for software development. The authors emphasize that, prior enrolling Software Engineering courses, the students have good programming skills, but not others such as estimation and planning, continuous integration, detailed design, debugging and unit testing. Furthermore, authors emphasize on the necessity of cultivating teamwork abilities, as it is a critical aspect of Software Engineering practice.

In the Latin-American context, also efforts have been done. In [37], a proposal for the design of Software Engineering curricula is presented, considering the expression of a competence as the combination of knowledge, social aspects, personal aspects, and environment conditions. The authors emphasize two types of competences: *Technical skills* and *soft skills*. Soft skills are considered as skills needed for social interaction, such as: communication, leadership, self-motivation, teamwork, etc. They propose a classification of the competences for a software developer: (1) Algorithm and problem resolution, (2) team work, (3) software development processes, and (4) project management.

A specific combination of PjBL and a process model is presented in [38]. The authors present a methodology for developing competences associated to the most representative roles in the software industry to undergraduate students, based on a project-based approach of teaching and following the MoProSoft model structure for the development of software real projects. They use a classification of competences in the following areas: Cognitive, problem solving, self-learning and self-knowledge, social, and motivation to work.

As we can see, there are efforts in the use of SWEBOK guiding curricula design and complementing SWEBOK's KAs. Several studies are focused in graduate programs. On the other hand, some studies have focused on specific areas of Software Engineering such as RE, conducting student preparation through PjBL. However, not many studies describe practical case studies with undergraduate students. On the other hand, there is not consensus about a set of standardized competences in PjBL or Software Engineering. The SEEK proposal can be considered as a standardized guide for the development of Software Engineering competences. We present our experiences implementing PjBL with undergraduate students participating in projects with real customers, assessing the SEEK competences acquired or enhanced. We found that this environment provides a scheme for students to acquire Software Engineering competences in hands-on, enabling them for real employment.

IV. Competences in Software Engineering and Bloom's Taxonomy

The Software Engineering Education Body of Knowledge [20] considers ten fundamental knowledge areas that are essential components of Software Engineering: (1) Computing Essentials (CMP), (2) Mathematical & Engineering Fundamentals (FND), (3) Professional Practice (PRF), (4) Software Modeling & Analysis (MAA), (5) Software Design (DES), (6) Software Verification & Validation (VAV), (7) Software Evolution (EVL), (8) Software Process (PRO), (9) Software Quality (QUA), and (10) Software Management (MGT).

For each topic, a Bloom's Taxonomy level (indicating what competence a graduate should possess) and the topic's relevance (indicating whether the topics is essential, desirable, or optional to the core) are designated. Bloom [39] considers three domains of educational activities or learning:

Cognitive: mental skills (knowledge);

Affective: growth in feelings or emotional areas (attitude or self);

Psychomotor: manual or physical skills (skills).

Domains may be thought of as categories. Instructional designers, trainers, and educators often refer to these three categories as KSA (Knowledge [cognitive], Skills [psychomotor], and Attitudes [affective]).

This taxonomy of learning behaviors may be thought of as "*the goals of the learning process*." That is, after a learning episode, the learner should have acquired a new skill, knowledge, and/or attitude. The cognitive domain involves knowledge and the development of intellectual skills [40]. This includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of intellectual abilities and skills.

In this paper, we only focus on the cognitive domain, which is expressed in six cognitive levels as we can see in Table III. We take into account the revised version of Bloom's taxonomy [41], which considers Synthesis as the highest level of cognition.

TABLE III. LEVELS OF COGNITION IN BLOOM'S TAXONOMY

New Domain	Description
6. Synthesis (Creating)	Builds a structure or pattern from diverse elements. Put parts together to form a whole, with emphasis on creating a new meaning or structure.
5. Evaluation (Evaluating)	Make judgments about the value of ideas or materials.
4. Analysis (Analyzing)	Separates material or concepts into component parts so that its organizational structure may be understood. Distinguishes between facts and inferences
3. Application (Applying)	Use a concept in a new situation or unprompted use of an abstraction. Applies what was learned in the classroom into novel situations in the work place.
2. Comprehension (Understanding)	Comprehending the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.
1. Knowledge (Remembering)	Recall or retrieve previous learned information.

It is important to emphasize that SEEK considers only three levels of cognition for undergraduate programs: Knowledge, Comprehension, and Application. The three higher levels are considered only to graduate programs, as is indicated in the SWEBOK 3.0 [42].

V. THE NATURE OF PROJECTS IN COLLABORATION ACADEMY-INDUSTRY

These projects are implemented in collaboration academy-industry. From 2010 to 2015 we participated in 18 projects with 5 different companies. Some projects involve software systems while others involve a combination of software and hardware. These projects are funded by CONACYT and the Ministry of Economy (Spanish: Secretaría de Economía, "SE"). Companies submit projects to CONACYT in order to get grants for implementing systems that they need in their business processes.

Formalizing the collaboration academy-industry. The participation of universities in these projects is formalized signing an agreement, which indicates outcomes, delivering dates and other conditions. Due to the formality established by the collaboration academy-industry, these projects have the following aspects: (1) A specific scope; (2) well-delimited outcomes; (3) delivery deadlines; (4) human resources for specific roles; and (5) resources and media to use for the systems implementation. In general, these projects involve the following characteristics: *PJ-ch_01, PJ-ch_02, PJ-ch_03, PJ-ch_04, PJ-ch_05, PJ-ch_06, PJ-ch_07*, and *PJ-ch_08.*

Project planning. Company and university's Project Director and Project Administrator together formulate the project, indicating: phases, outcomes, activities, delivery dates, human resources required, material resources. For each phase, activities are expressed in two levels of granularity, indicating only starting and ending dates for each phase. This means that later, team members should do the scheduling of the activities inside each phase, determining starting and ending dates for each activity.

The human resource is a key factor for the success of these projects. Table IV shows the more representative roles. We show some examples of activities performed for each role and who plays each one.

TABLE IV. REPRESENTATIVE ROLES IN PROJECTS

Role	Activities	Human resource
Project Director	Project plan and project scheduling; Managing the financial resources and materials for the project.	Department Director; Professor-Researcher
Project Administrator	Project plan and project scheduling; Providing technical resources and knowledge to the development team.	Department Director; Professor-Researcher
Team Leader	Activity scheduling for each team members. Supervising and monitoring activities.	Student
Hardware Analyst	Hardware system analysis and its documentation.	Student
Hardware Technician	Designing and implementing hardware devices.	Company employer
Software Analyst	Software system analysis and its documentation.	Student
Programmer	Implementing software components and modules.	Student
Documenter	Documenting each artifact of the project. Integrating all documentation of the project.	Student

People selection. University's Project Director and Project Administrator select team members considering the following aspects: (1) Evidenced skills for roles such as programmer, analyst, team leader, tester, documentation; (2) semester where students are enrolled in; (3) Software Engineering courses they have completed; (4) previous experience in past projects.

In the case of the Computer Engineering undergraduate program, the specialization in Software Engineering is integrated with the following courses: Software Systems Analysis and Design (SAD-C); Requirements Engineering (RE-C); Interaction Design (ID-C); Software Engineering (SE-C); Software Quality Assurance (SQA-C), Software

Project Administration (SPA-C). We have other supporting courses such as: Programming (PG-C), Structured Programming (SP-C), Object Oriented Programming (OOP-C), Advanced Object Oriented Programming (AOOP-C), Distributed Applications (DA-C), Web Application Development (WAD-C), Process Engineering (PE-C), Project Administration (PA-C), Mobile Application Development-Android (MAD-C), iOS Application Development (iAD-C).

The students are required to have successfully completed courses such as PG-C, SP-C, OOP-C and AOOP-C, which give to students the fundamentals of programming. Commonly we select students that also have completed courses such as SAD-C, RE-C, ID-C, SE-C, because we consider that these courses give to students the knowledge and skills needed for developing software projects.

The university's Project Managers explain to students the nature and scope of the project, the calendar, the outcomes, delivery dates, the scope of each phase, training needs for special topics, and grants.

Students play different roles based on the courses they previously completed and their skills. Students whom previously have completed a set of programming courses and have programming skills work as programmers. Students whom previously have completed software analysis and design courses work as analysts and documenters. Commonly, a team leader is a role played by students whom have previously completed the SE-C course or whom have strong skills in leadership and teamwork.

Student motivations. We have the modality of "Academy-industry Collaboration Projects with Credits" as extrinsic motivation for students. This modality gives students two curricula credits for participation in a project, and also has the associated credits of at least two courses of the curricula. In the case of the Computer Engineering undergraduate program, we can associate the following courses: SAD-C, RE-C, ID-C, SE-C, SQA-C, SPA-C, DA-C, WAP-C, PE-C, PA-C, MAD-C, iAD-C. This means that in some cases students have not completed already these courses before the projects, so that, they acquire the knowledge and skills in hands-on during the project.

Project initiation: Once team members have been selected, a set of introductory meetings are performed. Company's Project Administrator explains to students the general aspects of the company, products and services they provide, the scope of the project, the connection of the project with the business processes of the company, the calendar and scheduling, outcomes, delivery dates, work conditions for the project, previous systems related with the project, and other aspects.

Developing SEEK competences –Scenario 1: If students participate in projects after they have completed Software Engineering courses, they have the opportunity to practice these knowledges and skills during the project, so that, they have the possibility to enhance their skills gaining competences for software development. Company's Project Administrator also gives practical training in the main topics involved for the system implementation.

Developing SEEK competences –Scenario 2: If students participate in projects before they have completed Software Engineering courses, they should learn the contents of the courses during the project. University's Project Administrator gives a fast training in the main topics of software development, and the students gather knowledge and skills in hands-on during the project. Company's Project Administrator gives practical training in the main topics involved for the system implementation.

In both scenarios, courses from the Software Engineering curricula block are associated to the project, corresponding to the nature of the project and the content of courses. Students obtain the credits of courses depending on project and their role played.

Giving grades to students. Each project has a well-defined calendar, indicating phases, activities, outcomes and deliverable products and deadlines. Teams work based on the calendar, carry on activities and generating outcomes and deliverable products. Teams have working sessions with Project Managers and technical advisors at the university and also have working sessions with Project Managers and technical advisors at the company. We have review sessions weekly, focused on the verification and validation of activities and quality of outcomes and deliverable products. For each phase, the client (company) approves the deliverable products and signs a phase approval document. At the end of the project, university and company's Project Managers assign grades to students based on their performance during the project and the completion of resulting products.

An important aspect of this approach is that students have the opportunities to enhance their soft skills. At the end of each phase, advances of the project are shown in a presentation session. Company's Project Managers, Administrators, technical advisors and end users attend this presentation. To have this presentation, students prepare slideshows and software demonstrations. At the beginning of the project, analysis of technologies and frameworks are presented in those sessions. Students show their analysis and judgment about which technologies are better to use for the system implementation.

Planning and scheduling skills, speech abilities, communication skills, teamwork and interaction skills are practiced and enhanced during the project.

VI. Assessing SEEK Competences: Preliminary Results

In 2015 we had six projects in collaboration with industry, working from April to December, with the participation of 24 students of the Computer Engineering undergraduate program. Table V summarizes the type of software systems implemented and the programming tools employed. The systems implemented are medium and big size.

Two projects used SCRUM from the beginning to the end. The remaining four projects used a hybrid approach, combining elements from waterfall and SCRUM.

TABLE V. SOFTWARE SYSTEMS DEVELOPED IN PROJECTS

Project	Number of students	Process model used	Type of system	Mobile [Android/iOS]	Database	Programming Languages used	Frameworks used
Prj-01	4	SCRUM	Web, WPF Desktop application		Entity Framework, SQL SERVER 2008	C#, Javascript	Microsoft ASP.NET MVC5 Ozeki Camera SDK Emgu Library Web API 2.0
Prj-02	4	SCRUM	Web		SQL	C#, Javascript, (Razor/HTML, CSS for views)	Entity Framework, bootstrap
Prj-03	4	Waterfall/ SCRUM+	Web	Android		JavaScript	Polymer
Prj-04	4	Waterfall/ SCRUM+	Web	Android/iOS	MySQL/MSSQL	PHP, JavaScript	JQuery, Cordova, Crosswalk
Prj-05	4	Waterfall/ SCRUM+	Web Responsive Design	Responsive	SQLite (Pero se adapta a cualquier BD)	Python, Javascript, HTML, CSS	Django, Bootstrap, amCharts
Prj-06	4	Waterfall/ SCRUM+	Web		MongoDB (NoSQL)	C#, JavaScript	.NET MVC, Angular.JS

Table VI shows information about which Software Engineering related courses have been completed by students before the project. For example, 100% of students have completed PG-C before participating in the project.

TABLE VI. COURSES COMPLETED BEFORE THE PROJECT

Course	Percentage of students
PG-C	100%
SP-C	96%
OOP-C	100%
AOOP-C	75%
SAD-C	79%
RE-C	42%
ID-C	79%
SE-C	46%
SQA-C	12%
SPA-C	12%
DA-C	17%
WAD-C	8%
PE-C	58%
PA-C	17%4
MAD-C	8%2
iAD-C.	0%

As strategy, teams were integrated as follows, including:

a) At least one student whom have completed SE-course.

b) At least two students with programming skills.

c) At least one student with analysis/documentation skills.

d) At least one student with leadership abilities.

As we can see on Table VI, based on the percentages presented we have an appropriate number of students for integrating teams.

The student whom has completed SE-C previously serves as a guide to the team in the software process development, having a good understanding of the different activities, outcomes and deliverable products in each phase. Commonly this student also plays as team leader. Having two programmers is a good strategy for us, because the size of the projects and for programming workload distribution. Software analysis is a key activity for building quality software, so that, in each team we have at least one student playing this role.

Taking into account the activities that students perform in the projects, from SEEK we considered the following KAs: CMP, PRF, MAA, DES, VAV, PRO, and MGT. We selected the more presentative specific knowledges in each area, specifically we considered the following knowledges:

CMP.cf: Computer Science foundations
CMP.ct: Construction technologies
CMP.tl: Construction tools
PRF.psy: Group dynamics / psychology
PRF.com: Communications skills (specific to SE)
PRF.pr: Professionalism
MAA.md: Modeling foundations
MAA.tm: Types of models
MAA.af: Analysis fundamentals
MAA.rfd: Requirements fundamentals
MAA.er: Eliciting requirements
MAA.rsd: Requirements specification & documentation
MAA.rv: Requirements validation
DES.con: Design concepts
DES.str: Design strategies
DES.ar: Architectural design
DES.hci: Human computer interface design
DES.dd: Detailed design
DES.ste: Design support tools and evaluation
VAV.tst: Testing
VAV.hct: Human computer user interface testing and evaluation
PRO.con: Process concepts
PRO.imp: Process implementation
MGT.con: Management concepts
MGT.pp: Project planning

MGT.per: Project personnel and organization
MGT.ctl: Project control
MGT.cm: Software configuration management

Due to the space limitations in this paper, we only present results for three KAs (CMP, PRF, MAA), as we can see in Tables VII to IX. Next we show some examples of specific knowledges in these areas (For more information about the description of all knowledges, please see [20]):
CMP.cf.1: Programming Fundamentals (control & data, typing, recursion)
CMP.cf.2: Algorithms, Data Structures/Representation (static & dynamic) and Complexity
CMP.cf.3: Problem solving techniques
CMP.cf.4: Abstraction – use and support for (encapsulation, hierarchy, etc.)
CMP.cf.9: Programming language basics
PRF.psy.1: Dynamics of working in teams/groups
PRF.psy.2: Individual cognition (e.g. limits)
PRF.psy.3: Cognitive problem complexity
*PRF.psy.*4: Interacting with stakeholders
PRF.psy.5: Dealing with uncertainty and ambiguity
PRF.psy.6: Dealing with multicultural environments
MAA.md.1: Modeling principles (e.g. decomposition, abstraction, generalization, projection/views, explicitness, use of formal approaches, etc.)
MAA.md.2: Pre & post conditions, invariants
MAA.md.4: Properties of modeling languages
MAA.tm.1: Information modeling (e.g. entity-relationship modeling, class diagrams, etc.)
MAA.tm.3: Structure modeling (e.g. architectural, etc.)
MAA.rfd.2: Requirements process
MAA.rfd.3: Layers/levels of requirements (e.g. needs, goals, user requirements, system requirements, software requirements, etc.)

Results for these knowledges are shown in Tables VII to IX and in Fig. 1, 2.

We applied a survey to students asking: (1) if they possessed SEEK competences before the project (acquired in previous courses or projects); and (2) the level in what they practiced SEEK competences during the project.

This survey pretends to detect:

a) The most practiced KAs in our projects in collaboration academy-industry, that is, those areas that are practiced from Application level to Synthesis.

b) The less practiced KAs in our projects in collaboration academy-industry, that is, those areas that are practiced from Knowledge level to Comprehension.

We measured three levels of accomplishment with respect to level suggested by SEEK: Achieved (√), exceeded (√+), and not achieved (×). Six levels are considered: Knowledge (K), Comprehension (C), Application (A), Analysis (AN), Evaluation (E), and Synthesis (S).

These results are shown in Tables VII to IX. We indicate the level suggested by SEEK. Also we indicate the level accomplished by our students in the projects, which obtained the higher frequency. For each competence, we indicate the percentage of students that practiced these knowledges from the suggested level by SEEK to Synthesis level.

TABLE VII. COMPUTING ESSENTIALS

KA	SEEK level	Projects in collaboration Academy-Industry (PCAI)		
		Level	**Accompli-shment**	**% of students**
CMP.cf.1	A	A	√	76.5
CMP.cf.2	A	A	√	76.5
CMP.cf.3	A	A	√	82.4
CMP.cf.4	A	A	√	70.6
CMP.cf.5	C	K	×	58.8
CMP.cf.6	C	C	√	76.5
CMP.cf.7	C	A	√+	88.2
CMP.cf.8	C	A	√+	87.5
CMP.cf.9	A	A	√	94.1
CMP.cf.10	C	K	×	58.8
CMP.cf.11	C	A	√+	88.2
CMP.cf.12	C	K	×	64.7
CMP.ct.1	A	A	√	58.8
CMP.ct.2	A	A	√	88.2
CMP.ct.3	A	A	√	88.2
CMP.ct.4	A	K	×	47.1
CMP.ct.5	A	A	√	52.9
CMP.ct.6	A	A	√	76.5
CMP.ct.7	C	C	√	76.5
CMP.ct.8	A	A	√	52.9
CMP.ct.9	A	A	√	70.6
CMP.ct.10	A	A	√	64.7
CMP.ct.11	C	C	√	64.7
CMP.ct.12	A	A	√	64.7
CMP.ct.13	C	K	×	64.7
CMP.ct.14	K	A	√+	100.0
CMP.tl.1	A	A	√	70.6
CMP.tl.2	C	A	√+	88.2
CMP.tl.3	C	A	√+	76.5
CMP.tl.4	C	A	√+	94.1

TABLE VIII. PROFESSIONAL PRACTICE

KA	SEEK level	Projects in collaboration Academy-Industry (PCAI)		
		Level	**Accompli-shment**	**% of students**
PRF.psy.1	A	A	√	64.7
PRF.psy.2	K	A	√+	100.0
PRF.psy.3	K	A	√+	100.0
PRF.psy.4	C	K	×	76.5
PRF.psy.5	K	A	√+	100.0
PRF.psy.6	K	A	√+	100.0

PRF.com.1	A	A	√	64.7
PRF.com.2	A	A	√	64.7
PRF.com.3	A	A	√	88.2
PRF.com.4	A	A	√	70.6
PRF.pr.1	K	A	√+	100.0
PRF.pr.2	C	A	√+	94.1
PRF.pr.3	C	A	√+	70.6
PRF.pr.4	K	C	√+	100.0
PRF.pr.5	K	A	√+	100.0
PRF.pr.6	C	C	√	82.4
PRF.pr.7	K	K	√	100.0

TABLE IX. SOFTWARE MODELING & ANALYSIS

KA	SEEK level	Projects in collaboration Academy-Industry (PCAI)		
		Level	**Accompli-shment**	**% of students**
MAA.md.1	A	A	√	76.5
MAA.md.2	K	A	√+	100.0
MAA.md.3	K	K	√	100.0
MAA.md.4	C	A	√+	82.4
MAA.md.5	K	A	√+	100.0
MAA.md.6	K	C	√+	100.0
MAA.tm.1	A	A	√	70.6
MAA.tm.2	A	A	√	64.7
MAA.tm.3	C	A	√+	94.1
MAA.tm.4	K	K	√	100.0
MAA.tm.5	C	A	√+	87.5
MAA.af.1	A	A	√	58.8
MAA.af.2	A	A	√	47.1
MAA.af.3	A	A	√	64.7
MAA.af.4	C	A	√+	76.5
MAA.af.5	C	K	×	41.2
MAA.af.6	K	K	√	100.0
MAA.rfd.1	C	A	√+	82.4
MAA.rfd.2	C	A	√+	94.1
MAA.rfd.3	C	A	√+	100.0
MAA.rfd.4	C	C	√	82.4
MAA.rfd.5	K	A	√+	100.0
MAA.rfd.6	K	A	√+	100.0
MAA.rfd.7	C	A	√+	88.2
MAA.er.1	C	K	×	56.3
MAA.er.2	C	K	×	56.3
MAA.er.3	K	K	√	100.0
MAA.rsd.1	K	K	√	100.0
MAA.rsd.2	A	K	×	47.1
MAA.rsd.3	K	A	√+	100.0
MAA.rv.1	A	A	√	47.1
MAA.rv.2	K	A	√+	100.0
MAA.rv.3	C	A	√+	70.6
MAA.rv.4	C	A	√+	76.5

Fig. 1 and 2 show the set of SEEK knowledges that were accomplished at Application level. With this result we achieve the goal proposed by SEEK for undergraduate programs. (Due to the space in the paper we only present this level of accomplishment).

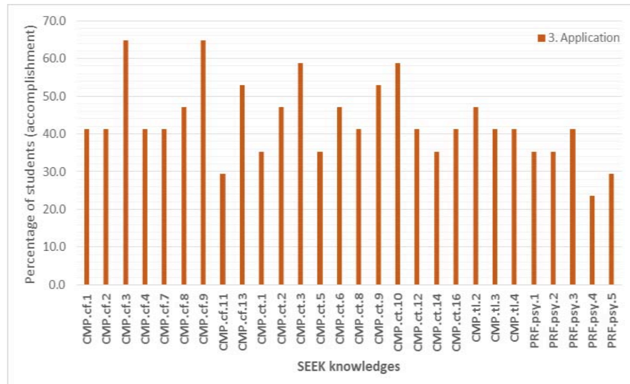

Figure 1. SEEK knowledges accomplished at Application level – Part 1.

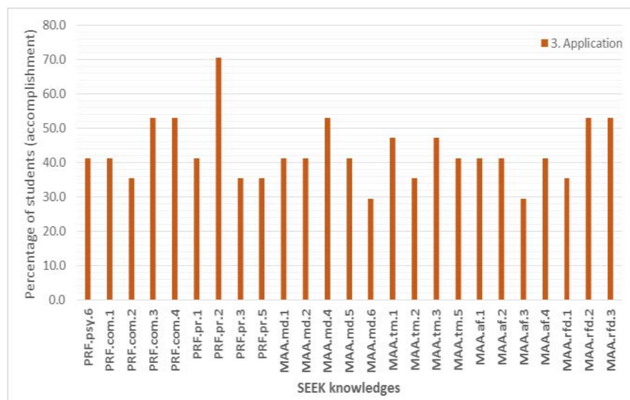

Figure 2. SEEK knowledges accomplished at Application level – Part 2.

Fig. 3 summarizes the accomplishment for each KA. We can argue that the results are supported by the project-work-based environment, involving all the project's features: PjBL-fe-01, PjBL-fe-02, PjBL-fe-03, PjBL-fe-04, PjBL-fe-05, PjBL-fe-06, and PjBL-fe-07.

Main findings:

1. We exceed the levels suggested by SEEK especially in two KAs: Software Process and Software Management. This is because our projects are implemented in a well-structured process, with well-defined phases and activities, outcomes and delivery dates. Besides, in these projects students must manage the time and resources efficiently due that we have delivery dates clearly specified. That is, the project management is well practiced by students.

Also we exceed SEEK's level for Professional Practice and Software Modeling & Analysis. A professional attitude is required to accomplish the project outcomes, and for interacting with different stakeholders in the company. On the other hand, analysis and modeling are the main activities performed in order to assure the software construction.

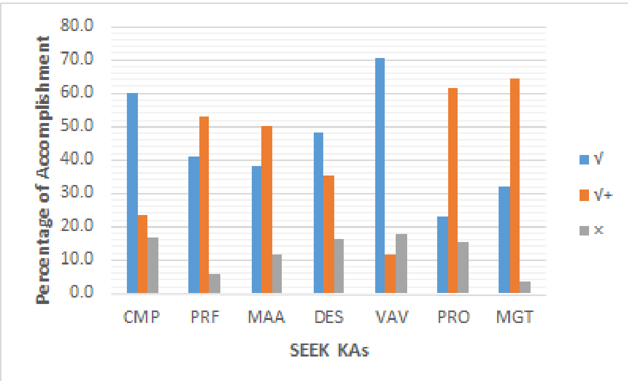

Figure 3. Percentages of accomplishment by KAs.

2. In most of the KAs we achieve the level suggested by SEEK: CMP, PRF, MAA, DES and VAV. Our undergraduate curricula include the basis of computer science. Professional attitudes are well practiced during the project. From VAV, testing is a core practice for code delivering in the project. In the same way, human-computer interface design is a specialty of our research group, so that, user interface design principles are taught.

3. Only in a few cases of KAs we didn't achieve the suggested level by SEEK. Most of the unaccomplished knowledges are related to specialized topics, such as: CMP.ct.4 Parameterization and generics, CMP.ct.13 Constructing heterogeneous systems; hardware-software codesign, MAA.er.1 Elicitation Sources, MAA.er.2 Elicitation Techniques. For some students, this terminology sounds not familiar, for instance, although students did a good requirements gathering, they are not familiarized with "domain experts" and "operational and organization environment" as elicitation sources for requirements, or with "surveys" and "participatory techniques" as elicitation techniques.

Based on the last finding, we detected those areas in which we need to focus our teaching and training efforts in order to cover SEEK KAs in a complete way.

VII. Conclusions and Future Work

In this paper we presented a case of experiences applying the PjBL approach to develop Software Engineering competences, based on the knowledge areas proposed by SEEK. We use as test scenario the case of real projects in collaboration academy-industry, which involves the development of software systems of medium and big size.

We use as basis our Software Engineering curricula block, providing knowledge to our students before their participation in real project, and as a guide for giving training to students during the projects in such topics involved in the curricula courses that they have not completed.

To assesses the accomplishment of SEEK's KAs, we considered activities carried out during the project development. These activities are mapped to the competences to be achieved by students playing each role in the project. These competences are expressed by technical and soft skills, which are adequate for real job, as is proposed in [43].

For this study, we applied a survey to a group of students that participated in real projects during 2015. Students expressed their appreciation about which knowledges they acquired and practiced during the project. We obtained encouraging results, as we can see in the last section; most of the SEEK knowledges were practiced at and over the level specified by SEEK.

As future work we have the following goals: (1) To do a formal assessment of the Software Engineering curricula block based on SEEK guidelines, for our undergraduate program of Computer Engineering; (2) to do an evaluation by role, detecting specific competences based on the SEEK proposal; (3) to make a formal comparison of results with other already existing Latin and Ibero-American proposals and experiences in Software Engineering courses using different teaching-learning approaches, as PjBL, collaborative learning, among others; and (4) to apply this study national wide in Mexico, trying to figure out how Software Engineering related undergraduate programs match SEEK proposal.

Acknowledgment

We want to give special thanks to all our students that participated in projects in collaboration academy-industry in 2015 for their contribution to carry out this study, for their efforts, encouragement, and motivation. These experiences gave us a new perspective of teaching and learning using the PjBL approach with real customers.

References

[1] National Research Council, "*Engineering in K-12 Education: Understanding the Status and Improving the Prospects*", The National Academies Press, Washington, D.C., 2009.

[2] S. Tockey, "Recommended Skills and Knowledge for Software Engineers," In: Proceedings of the 12th Conference on Software Engineering Education and Training, CSEET, 1999, March 22-24, New Orleans, LA, pp. 168 - 176.

[3] National Research Council, "*The Engineer of the 2020: Visions of engineering in the new century*", The National Academy Press, Washington, D.C., U.S.A., 2004.

[4] National Academy of Engineering "Conduct and Ethics in Engineering Practice Related to the North American Free Trade Agreement", Online Ethics Center for Engineering, 6/22/2006, retrieved on December 11, 20013, from

www.onlineethics.org/Topics/ProfPractice/PPEssays/internindex/8872.aspx.

[5] S. Rattanavich, "*Problem-Based Learning"*, In: Proceedings Seminar and Workshop "The Development of Lecturers' Instruction", September 4, 2008, Burapha University, Chonburi.

[6] M. Felder, R. Woods, E. Stice, A. Rugarcia, "The future of engineering education II: Teaching methods that work", Chem. Engr. Education, 34(1), pp. 26–39, 2000.

[7] C. E. Hmelo-Silver, "Problem-Based Learning: What and How Do Students Learn?," *Educational Psychology Review*, Vol. 16, No. 3, September 2004, pp.235-266.

[8] J. R. Savery and T. M. Duffy, "Problem Based Learning: An instructional model and its constructivist framework"*, Educational Technology*, Vol. 35, 1995, pp. 31-38.

[9] E. D. Ragan, S. Frezza, and J. Cannell, "Product-Based Learning in Software Engineering Education", In: Preceedings of the 39th IEEE Conference in Frontiers in Education, 18-21 Oct., 2009, San Antonio, TX, pp. 1-6.

[10] J. E. Mills, and D. F. Treagust, "Engineering Education – Is Problem-Based or Project-Based Learning the answer?" *Australian Journal of Engineering Education*, *Online Publication 2003-04*, pp. 1-16.

[11] H. R. Maier, "A hybrid just-in-time/project-based learning approach to engineering education", Proceedings of the 19th Annual Conference for the Australasian Association for Engineering Education, Yeppoon, Australia, 7-10 December, 2008.

[12] Apple, Inc., "Challenge Based Learning: A Classroom Guide, retrieved", accessed: July 17, 2013, from: http://images.apple.com/education/docs/CBL_Classroom_Guide_Jan_2011.pdf

[13] L. Johnson, and S. Adams, "Challenge Based Learning: The Report from the Implementation Project", The New Media Consortium, Austin, Texas, U.S.A., 2011.

[14] A. Gonczi, P. Hager, and J. Athanasou, "*The Development of Competency-Based Assessment Strategies for the Professions"*, Australian Government Publishing Service, Canberra, Australia, 1993.

[15] A. Gonczi, P. Hager, and L. Oliver, "*Establishing Competency-Based Standards in the Professions"*, Australian Government Publishing Service, Canberra, Australia, 1990.

[16] N. Spinks, N. Silburn, and D. Birchall, "*Educating Engineers for the 21st Century: The Industry View"*, (2006), Royal Academy of Engineering, Henley, England: Henley Management College.

[17] Apple Inc., "Apple Classrooms of Tomorrow—Today: Learning in the 21st Century", 2008, retrieved July 17, 2009, from http://ali.apple.com/acot2/global/files/ACOT2_Background.pdf.

[18] L. Yu, R. Surma, H. Hakimzadeh, "Incorporating Free/Open-Source Data and Tools in Software Engineering Education", Overcoming Challenges Software Engineering Education: Delivering Non-Technical Knowledge and Skills, Ligou Yu (Eds.). pp. 431-441. 2014.

[19] D. Sanders, "Using Scrum to manage student projects", *J. Comput. Sci. Coll.* 23, 1, pp. 79-79, 2007.

[20] IEEE CS and ACM Joint Task Force on Computing Curricula (2004). Software Engineering 2004, "Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering." A Volume of the Computing Curricula Series. August 23, 2004. Copyright © 2004 by the IEEE Computer Society.

[21] J. W. Thomas, "A review of the research on project-based learning", March, 2000.

[22] S. Han, and K. Bhattacharya Constructionism, "Learning by Design, and Project Based Learning", Emerging perspectives on learning, teaching, and technology, 2001

[23] J. A. Bellanca, "Enriched Learning Projects: A practical Pathway to 21st Century skills", Solution Tree Press, Bloomingtong, 2010.

[24] J.A. Bellanca, "*Designing professional development for change: A guide for improving classroom instruction",* 2nd Ed. Corwin Press, Thousand Oaks, CA, 2009.

[25] R. G. Kander, "A 2020 Vision: The Future of STEM", Department of Integrated Science & Technology ISAT), James Dadison University, 2006, retrieved from: *www.jmu.edu/news/KanderCAATalk-1.ppt*, accessed on: January 31, 2015.

[26] F. Maillardet, What outcome is engineering education trying to achieve? In: C. Baille & I. Moore (Eds.), Effective learning & teaching engineering, Routledge Falmer, New York, USA, 2004, pp. 27-35.

[27] P. D. Galloway, *The 21st-Century Engineer, a proposal for engineering education reform*, ASCE Press, Reston, Virginia, U.S.A, 2008.

[28] National Academy of Engineering, *Educating the engineer of 2020: adapting engineering education to the new century*, The National Academy Press, Washington, D.C., U.S.A, 2005.

[29] E. F. Crawley, J. Malmqvist, S. Östlum, and D.R. Brodeur, *Rethinking engineering education: The CDIO Approach,* Springer, New York, U.S.A., 2007.

[30] J. Heywood, Engineering Education: Research and development in curriculum and instruction, Wiley-Interscience, Hoboken, New Jersey, U.S.A., 2005.

[31] L. Pascail, The emergence of the skills approach in industry and its consequences for the training of engineers, *European Journal of Engineering Education,* Vol. 31, No 1, 2006, pp. 55-61.

[32] A. Alarcón, N. Martinez, J. Sandoval, "Use of Learning Strategies of SWEBOK© Guide Proposed Knowledge Areas", Lorna Uden; Francisco Herrera; Javier Bajo Pérez & Juan M. Corchado Rodríguez, ed., 'KMO' , Springer, , pp. 243-254, 2012.

[33] M. Umarji, C. Seaman, "Conceptualizing Software Engineering "People" Issues," retrieved from: http://ceur-ws.org/Vol-163/paper4.pdf, accessed on: January 31, 2016.

[34] N. A. A. Jamaludin, S. Sahibuddin, K. Jusoff, and N. H. Hidayat, Development of a Project-Based Learning Approach in Requierements Engineering, *International Journal of Computer Science and Information Security*, Vol. 8, No. 9, 2010, pp. 6.

[35] N. A. A. Jamaludin, and S. Sahibuddin, Challenges of a Project-based Learning Approach Towards Requirement Engineering, *International Journal of Computer Applications*, Vol. 50, No. 3, 2012, Foundation of Computer Science, New York, USA, pp. 1-5.

[36] T. Reichlmayr, and M. J. Lutz, A Course for Developing Personal Software Engineering Competencies*,* Paper presented at 2012 ASEE Annual Conference, San Antonio, Texas. https://peer.asee.org/20793.

[37] P. Marroquin, K. Olmos, L. F. Fernández, V. González, "Identificación de competencias para el diseño de un modelo educativo en Ingeniería de Software" (In Spanish), CULCYT, Enero-Febrero 2008, año 5, no. 24. pp. 34-40.

[38] M. A. Astorga Vargas, B. L. Flores Ríos, J. E. Ibarra Esquer, J. Mariscal Camacho, L. E. Vizcarra Corral, "Impacto del aprendizaje basado en proyectos implementado en una empresa escolar de Base Tecnológica dedicada al desarrollo de Software" (In Spanish), ReCIBE, Year 4, No. 4, December 2015, pp. 1-47.

[39] B. Bloom, "Taxonomy of Educational Objectives: The Classification of Educational Goals", Mackay, 1956.

[40] B. Bloom (Eds), Taxonomy of Educational Objectives: The Classification of Educational Goals, Mackay, 1956.

[41] L.W. Anderson, D.R. Krathwohl, P.W. Airasian, , K.A. Cruikshank, R.E. Mayer, P.R. Pintrich, J. Raths, M.C. Wittrock, "*A Taxonomy for Learning, Teaching, and Assessing: A revision of Bloom's Taxonomy of Educational Objectives",* New York: Pearson, Allyn & Bacon, 2001.

[42] Bourque, P., Fairley, R. E. (Eds.), SWEBOK 3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, Los Alamitos, CA. (2004). Available on: http://www.swebok.org, accessed on: January 20, 2016.

[43] Australian Computer Society. ICT Profession Core Body of Knowledge (CBOK), V0.6, 27 August 2015.