Cross-Course Project-Based Learning in Requirements Engineering: An Eight-year Retrospective

Joshua C. Nwokeji and Stephen T. Frezza, PSEM
Department of Computer and Information Science, Gannon University, Erie, Pennsylvania 16501 USA
Email: [nwokeji001,frezza001]@gannon.edu

***Abstract*—This paper focuses presents a study of eight years of cross-course project-based learning (CC-PjBL) in an upper-level requirements and project management (RPM) course. Project-based learning (PjBL) is a method of instruction in which students learn by investigating and solving real-world problems in and open-ended, time-limited context[1]. Our instantiations of CC-PjBL matched paired students in an introductory software Requirements and Project Management (RPM) with students in different technology- oriented software development course(s) to utilize the requirements deliverables created by their RPM-course colleagues. The paper includes a review of relevant related PjBL literature, descriptions of our initiative and experience in applying CC-PjBL to RPM topics over eight years, and the lessons learned thereof. This paper reports summary experiences from student evaluations of these courses to evaluate the CC-PjBL experiences. The discussion also includes problems encountered with CC-PjBL assessment, faculty participation, and course hand-off, which may be useful to instructors that are considering to apply CC-PjBL as a method of instruction and to those that are currently practicing PjBL.**

I. Introduction

Requirements Engineering (RE) is a critical part of a software engineering (SE) curriculum, but is also routinely taught as a component of computer science (CS) or information systems (IS) majors [3, 4, 5]. The critical roles played by RE courses in software development have been emphasized in multiple studies [6, 7, 8, 9]. RE helps in identifying the functional capabilities that should be implemented in a system, so that the system can meet the objectives and needs of all of its stakeholders, both client/customers and developers [10, 11, 12]. There is a strong correlation between requirements engineering and software quality, such that many software failures are directly and indirectly attributable to badly engineered requirements [3, 8]. Yet RE faces enormous challenges in industry, including a poor understanding of RE practices [13] and shortage of skills [14, 15]; additional challenges have been identified and discussed in [16].

Requirements Engineering Education (REE) aims to educate and train professionals to address the many of these issues in industry [13, 17]. Through REE, students can learn the tools and methods, as well as acquire the skills to elicit, analyze, specify, and validate software and system requirements [6, 17, 18]. The term 'requirements encompasses all capabilities (*i.e.*, functions, operations, behavior, quality attributes, constraints and features) to be encoded in a software, so as to meet stakeholders needs [10, 11, 12]. These capabilities are normally derived from the stakeholders *e.g.*, human beings that have consequential interests in the system being developed, including software owners and users [13, 19, 15].

In addition to the technical skills of modeling, analyzing, specifying and validating software capabilities, REE learning outcomes should also include social skills and the development of professional competencies to enable students to interact with and manage stakeholders and their expectations effectively [6, 14, 13]. Some examples of the social skills we want our students to experience include conflict resolution, negotiation, social perceptiveness and persuasion [14]. Technical skills include the knowledge and practice of requirements elicitation, documentation, modeling, analysis, change management and negotiation [10].

The basics of these technical skills can be acquired in a traditional classroom or laboratory with computers and other learning activities. However, to actually value these skills in practice [20], or more importantly acquire the associated social skills, students need the experience of actually interacting with "*stakeholders who are either real or behave in as realistic / authentic manner as possible. Only if these stakeholders have a real stake in what is elicited, documented and validated well, do the students face the whole range of emotions when interacting with them*" [15, 21]. In order to deliver its learning outcomes and be most effective, REE should incorporate student interaction with real and authentic stakeholders.

Finding and engaging these types of stakeholders in RE courses remains a major challenge [21, 3]. One alternative is to adopt the software internship or residency model [22, 23]. However, the student population typically outnumbers the internship opportunities available in a local set of industries and internships settings can limit the ability of tracking learning outcomes and can limit how students leverage and develop a breadth RE skills. A feasible alternative to addressing these limitations and challenges is to incorporate project-based

[1]*There is no consensus in the literature on whether or how to distinguish PBL (Problem-Based Learning) from project-based learning. To avoid confusion, we use PjBL to clarify that we are referring to Project-Based Learning. For an interesting discussion of the relationship between PBL and PjBL, see both [1] and [2].*

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

learning (PjBL) into RE courses.

PjBL is a learner-centered instructional method where students acquire knowledge, skills, and experiences by investigating and solving real world problems on more open-ended projects [1, 2, 24, 25]. The pedagogical benefits of PjBL are widely acknowledged [24, 25, 26, 27]. Popular among these are its support for cognition and experiential learning as well as the ability to engage students and bridge the gap between theory and practice [26]. Furthermore PjBL helps students to contextualize acquired skills and knowledge [27] and to develop creative thinking and problem solving skills [24]. In a typical PjBL setting, students are required to manage and closely interact with stakeholders to complete a given project. Such interaction can help students develop professional competencies [24] such as communication, negotiation, and persuasion, required for effective requirements engineering practice. Therefore, PjBL can be beneficial to REE and help to address some of its challenges.

In REE, one of the major advantages and difficulties of a PjBL approach is the need for live representation of multiple stakeholder perspectives. The common viewpoint is that of the potential user or client of an application or system. Another key perspective is of the development team, that is, those who actually need to use the requirements documentation in order to design and develop the application or system to fulfill the needs. These two viewpoints have very different needs from requirements documentation, review for different purposes, and use the documentation differently. We have utilized cross-course project-based learning (CC-PjBL) to allow students to experience both stakeholder perspectives.

Our instantiation of CC-PjBL matches project-based learning to an introductory requirements and project management (RPM) course wherein upper-division RPM student teams are paired with students in different technology-oriented software development course. Each pair is expected to deliver a software product whose stakeholders can be local business owners, university professors/instructors/staff, or students. The development course students then engage the project as the developers of the RE deliverables created by their RPM-course colleagues. Specifically, we report the summary experiences from student evaluations of eight years of these efforts, including some of the problems encountered in assessment, faculty participation, and project hand-off. These can be useful to instructors that are considering to apply PjBL as a method of instructions and also to those that are currently practicing PjBL.

II. Related Work

Project-based learning (PjBL) has been successfully applied to, and benefited, some other courses in the software engineering (SE), information systems (IS), and computer science (CS) curricula, for instance see [23, 25, 26, 28, 29, 30]. Yet, we found very few studies [6, 31] that report the application of PjBL to REE, the challenges and prospects as well as the lessons learned from such initiatives. Few other related studies [13, 32, 33] report a software-supported PjBL, where a software simulation plays the role of client in a virtual project.

Portugal *et al* report the application of PjBL to their requirements engineering (RE) course [6]. Students were allowed to choose their teammates and organize themselves into project teams. Each team plays alternating roles of client, builder, and auditor. Client teams develop a business need, problem or opportunity that could be solved using a software. Then the builder teams perform requirements elicitation and develop the functional and non-functional requirements that can be implemented to address the need of the client teams. Finally the audit teams are responsible for quality control and inspection of the requirements and solutions designed by the builder teams.

This approach is similar to the technique used by Liang and de Graaf in a requirements engineering course at the University of Groningen. The key difference however is that a requirements engineering wiki (RE-Wiki) was used to enhance team collaboration [31]. Both Quintanilla Portugal *et al* [6] and Liang and de Graaf [31] applied the role-playing approach to PjBL introduced by Zowghi and Paryani [34]. The results and evaluations from these initiatives [6, 31, 34] show that role-playing PjBL can be beneficial to REE. However, role-playing is only a substitute for what students can learn from the experiencing the emotion and resistance of authentic project stakeholders. In role-playing PjBL, most projects are owned and initiated by fellow students, who may have (or perceived as having) no real stake in these projects. Hence, role-playing PjBL may not be the most effective technique to help students acquire the social skills required from the RE professional.

Taichi Nakamura *et al* [32, 33] report a hybrid approach to PjBL, where role-playing is combined with a software system and used to educate 4th year RE course. A hybrid-PjBL approach has also been used to train corporate RE workers involving role-playing, a business domain expert system, and a software agent [33]. Students interacted with these components to complete a virtual project. In the role-playing component, students were organized in teams to play the roles of requirements engineers or business analysts working on a virtual project. In the second component, a business domain expert software system acts as the client or owner of the project. Finally, a software agent plays the role of an instructor or adviser to the students working on the project. Similarly, Regev *et al* [13] used a business game board to simulate real work environment for experiential learning in RE course.

These related initiatives [13, 32, 33] provide a means for students to learn RE concepts from a remote or distant location. But, simulating a client can limit the effectiveness of PjBL. While overcoming the limits of stakeholder access, expert systems and game boards limit interaction. Social skills can be better acquired through interaction with human beings and a more realistic work environment.

To address these limitations, encourage and support the application of PjBL to REE, we have implemented Cross-Course Project-Based Learning (CC-PjBL). In contrast to existing approaches, CC-PjBL allows students to interact and experience emotions from multiple stakeholders, who are real and authentic. Additionally, in CC-PjBL we require students to complete real world projects to be used by local businesses, charity organizations, business units within the university, or students organizations. By collaborating to work on real world projects with multiple stakeholders, including a development team, students can acquire the necessary professional skills and affect required of contemporary requirements engineers or business analysts. In the section that follows, we describe our CC-PjBL approach.

III. Description of CC-PjBL

CC-PjBL is the means that we use to translate two sets of related software-engineering materials, namely introductory requirements engineering and introductory software project management into active, experiential learning experiences. This hybrid course is titled *(*Requirements and Project Management). The mechanism is to find stakeholders with *Mjligheter* [35] (opportunities and/or problems) for which the faculty-of-record can find cooperating faculty-of-record teaching appropriate software development courses in our institution. For these development courses, different approaches have been tried, but the important constraint to scaffold for the RPM students is that there is a course where they will be able to help recruit a development team capable of producing a prototype. Typically these are web or mobile-application development courses, occasionally user-interface design or software maintenance courses.

Fig. 1: RPM Course Description

From a content-delivery perspective, the first phase of the course focuses primarily on RE tools and techniques, while the second phase of the course focuses primarily on quality and PM course outcomes. The course outcomes for both aspects of the course are shown in Figure 2. These four course outcomes emphasize skill in applying RE and PM knowledge.

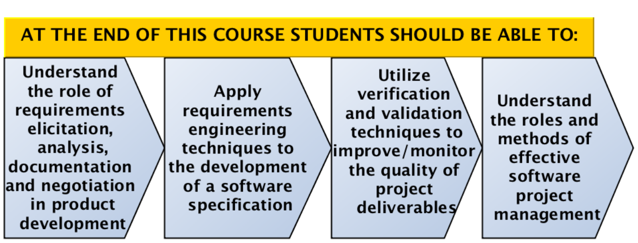

Fig. 2: RPM Student Learning Objectives/Course Outcomes

Normally in traditional REE, students focus on the development of a software specification. With PjBL, the difficulty with live stakeholders is that they normally want more for the investment of their time - they want a prototype or testable system, or user-interface design that they can use for testing usability and/or attracting funding - not a just a student-generated specification of questionable quality.

This is a difficulty in REE - assessing the quality and usefulness of the student-developed specifications. Without development stakeholders immediately tasked with building something from the specification, it is significantly more difficult to review their work well, or to find others who have a sufficiently vested interest in the quality and completeness of their specification. CC-PjBL changes this, as the specification has vested stakeholders - the development students who are tasked with building a prototype or testable UI Design from their requirements. Other students, who have their own motivations to do good work, or have a good team project in their development course, care that they build what the customer wants.

The hand-off of the requirements specification to the development team is a key learning milestone. Most often, the development teams come back with significant questions, and are often very critical of what the RPM students thought were good specifications. This also sets up key REE learning in both the difficulty and value of clear and concise requirements, as well as the importance of effective requirements change management. Another key professional skill that this helps develop is the process of negotiating when the development on the project actually begins - the sooner the first specification hand-off starts, the sooner the development team can start their work and get their team organized; but this also means that there is less time for exploring requirements. This is an important trade off and 'schedule' negotiation is one of the reasons that both RE and PM topics are covered in the course. 'Product' requirements are the focus of the specification but the 'Project' requirements affect the project, especially the when, who and how of the tasks. In the RPM course, as in many types of projects, experiencing the crossover between project and product requirements is often rather blurry, and also one of the motivations for CC-PjBL approach.

The second part of our RPM course covers introductory project management topics starting from project initiation to project closure. Topics include risk management, quality assurance, task and work-package identification and planning. Project management knowledge areas (PMKA) and PM lifecycle (PMLC) models are also covered. The PM part of the course usually starts immediately after hand-off, so that RPM students will become project managers of the development team; thereby getting practical PM experiences. During the development phase, the RPM students are expected to organize joint project planning meetings (JPPM), develop work and requirements breakdown structures (WBS/RBS), identify project risks drivers, and develop and periodically update a project risk management plan. Additionally, we expect them to negotiate with their development teams and develop plans for both product (software) and process quality. In the last phase of the course, the students are expected to help manage the product or its prototype. Finally, each project group needs to arrange for product demonstrations and hand-off to the customer (as appropriate), and prepare and submit a project report for assessment.

The PM portion of the course is the portion of the course with the most risk. Once their project starts, the RPM student teams are immediately put into a 'matrix management' situation where they have no real authority over the other students who are doing the development, and may or may not have any relationship with the faculty-of-record for the development development course with which they are paired. But it is exactly this sort of unscripted difficulty that makes the students' experience both rich and rewarding; Their project may fail, through their mistakes or the mistakes of others. The project may also succeed if they learn to navigate the particular personalities and issues of their project, not just a scientific, classroom application of particular RE or PM techniques. For RPM instructors, finding the correct level of scaffolding to allow the project to have good potential for success, but also leave the responsibility to the RPM student managers is one of the more challenging aspects of the CC-PjBL approach.

A. *Managing CC-PjBL*

Managing a course potentially running multiple student projects with multiple stakeholders can be difficult. Managing that in coordination with potentially multiple other faculty teaching other courses is harder. Our experience is that while students often have potential stakeholders for prototype projects, it is necessary that the faculty-of-record find and groom stakeholders and projects for use by the students. Similarly, it is important to have developed relationships with other faculty teaching development courses that have end-of-course projects of more than just a few weeks duration.

In our institution, we are particularly blessed in that most development-oriented courses use small-team course projects as part of their pedagogy. Multiple courses that could potentially incorporate and RPM project run every semester. Our FA semester usually has at least one web-development course, an iOS development course, a usability/user-interface design course, and a software maintenance course that have the potential for hosting one or more RPM projects each. Our spring (SP) offerings normally have at least one web-development and one Android development course that can accommodate RPM projects. The key to establishing the cross-course collaboration is to get the agreement from the faculty-of-record for each of the development courses, and to understand their plan for how projects will run in their course(s), when they allow the students to start on their projects, and how they want to assign students to projects. These decisions are always left to the faculty leading the development course to determine for their students.

Integrating an RPM project into a development course requires effort. The advantage to the faculty leading development courses is that the RPM students are responsible to identify the problem and manage the clients for the project. This allows their development students to focus on the technical aspects, which are typically better aligned with their course outcomes. The disadvantages include: 1) having to work with another faculty's timetable, typically not available until right at the start of the semester, and 2) having to address questions and issues from the RPM students working with their students. For students in the development course(s), there is always the potential that they are also taking the RPM course. The normal rule is that students are not allowed to be on both the requirements and the development sides of the same project. This has an advantage, in that development students who are also RPM students have another channel of communication with the RPM students managing their project.

The basic order of steps for the cross course collaboration is presented in Table I. The pre-course work includes recruiting project stakeholders, and finding development course faculty willing to consider one or more RPM projects for development course projects. These dev teams limit the projects available to the technologies used in those courses. At the beginning of the RPM course, the faculty introduces the available stakeholders as well as the available development technologies. Subject to faculty approval, students are encouraged to propose other projects for which both live stakeholders and suitable development teams are available.

TABLE I: Cross-Course Coordination Steps and Milestones

Week	RPM Activity	Dev Activity	Responsibility
-?	Recruit project stakeholders	–	RPM faculty
-1	Agree to collaborate		Both faculty
1	Introduce stakeholders & technologies	–	RPM faculty
1-2	Form RPM teams	–	RPM faculty & students
2+	Elicit requirements	–	RPM students
4-5	Project proposals		RPM students
5	–	Project & team selection	Dev faculty & students
5-6	Model review & requirements hand off		RPM students
7-9	Initial planning & project Kickoff		Both students
8+	Proj management & requirements change	–	RPM students
8+	–	Proj development	Dev students
14	Project closure & reporting		Both students

At this point, RPM teams are formed around the available projects, normatively 3-4 students per RPM team. When there is more than one project available, a student voting scheme is used, where students select their top 3 projects, and the RPM faculty assigns student teams they think will be successful. With teams formed, they then begin the requirements work: planning for, identifying and meeting with project stakeholders. Usually a requirements model plus an initial scope statement are early deliverables. Around week 4-5, these are turned into 'elevator' talks to introduce the project and to the collaborating development (dev) courses to help recruit development teams. At this point, the collaborating dev faculty assign students to the RPM projects. In some cases, this is a complete match - all students in the development course are assigned to one of several RPM projects. In other cases, the RPM projects are among others eligible for use in the dev course. A requirements hand-off is scheduled, where RPM students hold a formal or informal review of their current documentation with the development team. This usually leads to questions and changes. Around mid-semester (apx. week 7) the initial planning for the project and a project kick-off is scheduled. These may be pushed back in the semester, depending on when the development faculty actually wants their students to begin work on the project, but typically kickoff takes place before week 9. After project initiation, the RPM team proceeds with project management, further requirements work, and managing requirements change and product quality activities. The development teams proceed with building prototypes from the negotiated requirements, using the techniques and technologies of their development course. Development continues to the end of the semester, when an integrated prototype is presented by each team in their respective course. Then each team completes the project report(s) and reflection(s) expected in their respective courses.

B. Retrospective on CC-PjBL

The RPM course originally started as a small lecture/activity course focused on the core knowledge content of requirements engineering and project management. The initial two instances of the course did not have integrated projects. Consistent with the findings in the literature, these courses had mixed reviews from the students in terms of their effectiveness. CC-PjBL was first tried in SP 2009 to overcome the teaching-and-learning issues encountered in the first runs of the course. The origins of the cross-course course collaboration were in a small single-project course, where the number of students worked together to build useful requirements models and specifications, and manage a project to some form of prototype success.

As shown in Table II, the first several years that the course ran, it was with a single cross-course collaboration on a single project within the development course. In many cases, these were web development courses, and instructor for the development course would subdivide the requirements and/or components across small teams within the course, with the intent of having students integrate each other's code, and the RPM team there to help the development teams organize and integrate their work, supporting the development instructor in tracking the various teams' work. As enrollment expanded, in SP'13, a new model was attempted, where the RPM students were assigned to three different projects, each of which was assigned a development team in a single course. In this case, there were other small-team projects within the development course, but three were managed by the RPM students.

As Table II shows, this changed radically when the course was opened up as an introductory graduate course in FA'14, and the enrollment jumped to 26. Now the project needs were much more significant, and three different courses were recruited to include one or more RPM projects. With these next offerings of the course, communication with the development faculty members was mixed; some projects worked significantly better than others corresponding to the strength of communication with the development faculty. RPM team sizes were generally kept to 3-4 students, and generally corresponded to the team size in the development course.

TABLE II: RPM CC-PjBL Sections 2009-2016

Section	Year	Instructor	Enrolled	# Projects	# CC
SP	'09	Frezza	5	1	1
SP	'10	Frezza	5	1	1
SP	'11	Frezza	9	1	1
SP	'13	Frezza	2	3	1
FA	'14	Frezza	26	7	3
SP	'15	Frezza	32	5	2
FA (x2)	'15	Vitolo	43	11	3
SP-1	'16	Grande	15	4	2
SP-2	'16	Frezza	11	3	2
FA	'16	Nwokeji	21	5	2

Historically, as Table II indicates, we've managed with as many as two different RPM instructors, as many as eleven different RPM projects and with as many as three different development courses collaborating. One of the most difficult aspects of maintaining the CC-PjBL has been maintaining the collaboration relationship with suitable target development courses. While this has been easy with faculty who've taught the RPM course, this has been less consistent with faculty unconnected to the course. Faculty turnover has also been a contributor to the problem.

IV. Data Collection

To evaluate our CC-PjBL approach, we extracted data from our institution's course evaluation tool (EvalTools) for analysis. Towards the end of each semester, students are expected, but not required, to complete survey questions regarding the course effectiveness. In Figure 3, the lower line shows the number of students that completed the survey while the upper line shows number of students registered for that section.

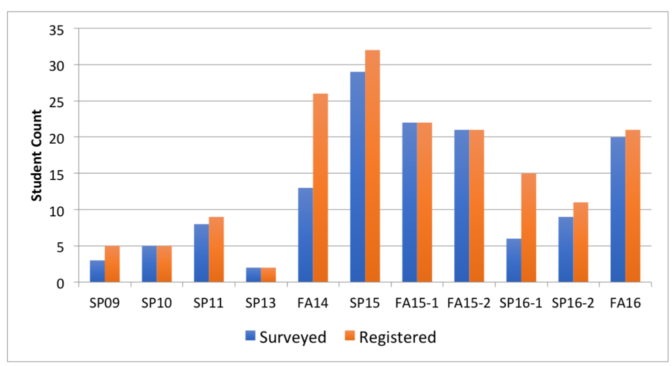

Fig. 3: CC-PjBL enrollment and survey response rates

Comparison data from course evaluations for eleven different RPM sections offered over a period of eight years are shown, including the first CC-PjBL attempt in 2009. Relative to CC-PjBL, five questions were selected to rate the effectiveness of the course and to understand if/how our CC-PjBL instructional approach has been beneficial in achieving the instructional objectives. The five survey questions corresponding to these components were:

1) *How would you rate the overall quality of this course?*
2) *How would you rate the faculty's overall performance in this course?*
3) *How would you rate your overall learning experience in this course?*
4) *How effective has this course been in achieving its instructional objectives and/or student learning outcomes?*
5) *Compared to other courses, the amount of work required in this course was:?*

For each of these questions, students were required to select one response from a list of five options ranging from a highest value to the lowest value. A sixth N/A ('not applicable') was included to allow for questions that students choose not to answer. Answer prompts matched the questions, *e.g.* ratings as 'Excellent' (5) to 'Poor'(1) or 'Highly Effective' (5) to 'Ineffective' (1). Along with sets of qualitative information, as these on-line surveys were double-blind, students were encouraged to provide qualitative feedback as well.

V. Findings and Discussion

For the past eight years, we have educated 169 students in our RPM CC-PjBL course. The major rationale for combining requirements engineering topics with project management was to enable our students acquire the RE and PM knowledge, skills and affect necessary to manage software development project. Three questions relate to the perceived quality of the course, which we used to evaluate the course over time. Using a weighted scoring of a '5' for excellent and a '1' for poor, an overall trend per section presented in Figure 4 depicts differences over the 11 RPM sections ordered by time.

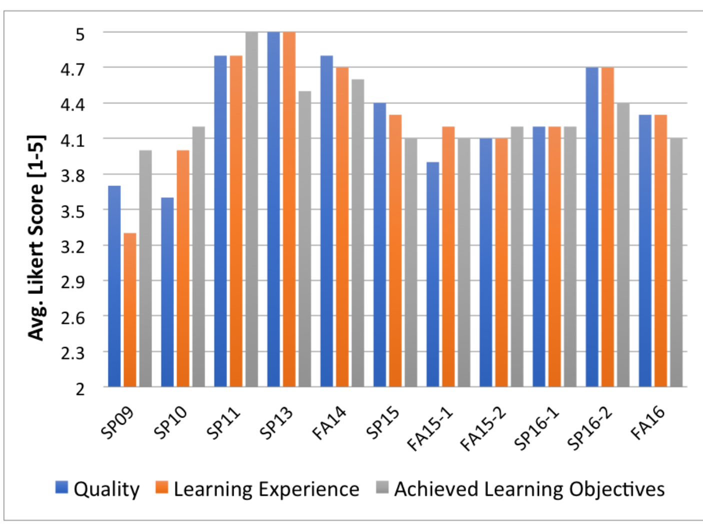

Fig. 4: Overall Student Evaluation Over Time

What was most obvious in Figure 4 was a 'ramp' up in the first runs of the course. This was expected with a new technique, and the faculty learning how to manage the student expectations and quality of the experiential learning. The other interesting aspect is that even with a large jump in the number of projects, collaborating courses and faculty new to the course and to the CC-PjBL approach, the student experiences did not appear to suffer greatly, and the overall evaluations remained better than the first runs of the course.

Another analysis view was to look at all eleven cohorts as a whole. Our analysis examines results from one hundred and thirty eight (138) students who took the survey, representing the one hundred and sixty nine (169) students registered for the course. Figure 5 depicts the aggregate eight-years of responses from all eleven sections.

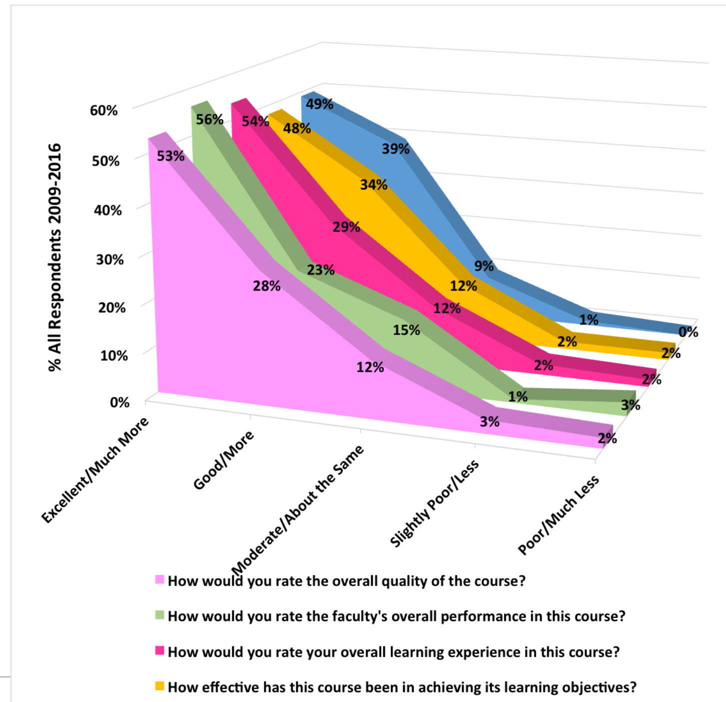

Fig. 5: Overall Student Evaluation

Among the more interesting information, this figure shows that 82% of students agreed that the course was effective or very effective in achieving its instructional objectives. We were also not very surprised on the response in Figure 5 where 49% of the students agree that the amount of work required to complete this course was very high, while 39% said high. Furthermore, 47% rated the overall quality of this course as excellent. Similarly, 46% of the students reported their overall learning experience was excellent, while only 2% reported poor.

The qualitative data shown in Figure 6, consistent with the overall data in Figure 5 shows that students acquired core RE and PM skills, with room for improvement, particularly with the student workload.

From the data presented in Figures 5 & 6 we report the following findings:

POSITIVE COMMENTS

Spring 2011 Very challenging coursework, but the completion of the work did offer a unique insight into the material and probably a glimpse of real world business rigors.	Spring 2010 Requirements textbook was cool and good basis of what shou;d be understood before design begins	Fall 2015 Would prefer assignments and project work instead of exams to this course more effective learning experience Fall 2015 Coursework is so helpful when we go to live projects
Spring 2011 Got to be a lot of work near the end of the semester, but understandably so. I really enjoyed working hands on with the developers (for the most part)	Spring 2013 Real world,hands on experience Fall 2014 I think it is perfect we don't need to change anything in this course	Fall2014 How my instructor interacts with the students and try to explains everything Fall 2015 It was really helpful to understand about the steps involved in the software project
Fall 2016 It is a very intersting course and I get a chance to communicate regular meetings with stakeholders and development team and learn new things about the project	Fall 2014 If you are in computer science than you have to do this course to learn gathering requirements . Because if you don't know how to derive requirements than you can not be a good engineer . For me, I Learned a lot's from this class , before this class i don't know anything about requirements but now I can say like I can do any kind of requirement with project management	

NEGATIVE COMMENTS

Spring 2009 I believe that there is too much time on requirements portion of the class and not enough in the project management area. I feel that our project was rished at the end of the class	Spring 2010 Way to much work was put into trying to actually manage a group of students.There isn't enough time in the semester to both do book work and manage a class.	Spring 2011 the book on software requirements was not helpful at all. the chapters were not ordered to develop a knowledge, rather it was made to jump around to different chapters, and wasn't thourough in explaining key definitions and topics.project management book was excellent choice
Spring 2013 Not enough resources to complete course project during semester	Fall 2016 The course is too long, there is some not benificial information	Spring 2015 The text books are good but they are very detailed
Fall 2014 This course is excellent but I think it is too long because we have to study two books with 23 chapters from one book and 9 chapters from another book	Spring 2009 Little to no feedback on individual assignments. Multiple choice on exams are unclear and ambiguous	Fall 2014 I think, we have more than 16 assignment during class , that's too much for each student because we have to do assignments for another class too.

Fig. 6: Positive and Negative Sample Comments from RPM Students 2009-2016

(a) The CC-PjBL instructional approach appears effective in achieving our RPM course outcomes. As shown in Figure 4 and the aggregate data in in Figure 5 85% of the students agree to this. This is also supported by the positive comments in Figure 6.

(b) The faculty that delivers the learning outcomes of our RPM have consistently shown excellent performance over the years. This is supported by the chart in Figure 5 where 53% of the students rated faculty's overall performance as excellent, while another 33% rated it as good with only 1% rating it as poor. We acknowledge that this is remarkable, because despite the recent shortage skills and lack of RE professionals and educators [14, 15], our institution has consistently identified and retained faculty with greater efficiency in delivering RE and PM learning outcomes.

(c) The overall quality of our course and its instructional approach as well as students overall learning experience are very good. As shown in Figure 5, 47% of the students rated the overall quality of this course as excellent, while 34% rated it as good. This is very similar to the rating for student overall learning experience in our course.

A. Challenges

Results from our data collection and analysis show that our CC-PjBL approach can benefit REE in many ways, especially in achieving course objectives and outcomes. Yet, the following challenges should be addressed in order to achieve optimal results:

(a) Identifying projects with real-stakeholders: Stakeholders are central to requirements development, they are the owners and users of the requirements being elicited and thus the software being developed. For students to acquire the interaction, social, and other skills through RE-PjBL, they need to interact with people who have real and authentic stake in a project. Finding such people is a major challenge in our RE-PjBL course. Without scaffolding by the instructors in advance, it takes students more than 3 weeks to identify a stakeholder with project. This is a considerable delay relative to the the total length of the course.

There are multiple project owners that our students currently depend on for projects. These are local business owners or charity organizations, functional units within our university, professors or staff, and students. Yet, these are not always guaranteed. Most local businesses and charity organizations require software to streamline business operations and transaction. But due to funding constraints, they usually do not initiate such projects, even when our students send proposals to them.

(b) Finding development-course instructors: Collaboration with other faculty, especially those that teach software development related courses, is central to the success of our CC-PjBL approach. We depend on these faculty to organize the teams of students in the corresponding development courses, map each team to a corresponding team in our RPM course, and be willing to attend stakeholders meeting as well as contribute to the overall success of the project. Without the involvement and commitment of development course faculty and students, our approach can hardly be successful. Yet, over the years, we have experienced great difficulties in finding other faculty who are willing to collaborate. One reason is the time, effort and risk required for CC-PjBL. Some development faculty may find it easier for students to work on hypothetical project, which involves little or no meeting with stakeholders, and thus lesser resource consuming. It is more easier when the RPM instructor is also teaching a development related course, however this is not always the case. Hence, other institutions and REE instructors who desire to try our approach, must first and foremost find and convince other faculty in development related courses to collaborate and provide the teams of development students.

(c) The workload line in Figure 5 and some negative comments in Figure 6 shows that our approach involves a greater amount of workload when compared with other instructional approaches. One reason for the relatively higher workload is because this is a hybrid course that covers learning outcomes and contents from both RE and PM. As students are expected to read two textbooks, this adds to the workload. Furthermore, students are required complete a software requirements development project - including scheduling and traveling to stakeholder locations, and dealing with the reality that stakeholders cancel or reschedule meetings, change requirements often, and can become difficult for students to manage. While excellent 'real-world' learning experiences, students often find these situations frustrating and time consuming. We agree with this evaluation, and continue to work to reduce the workload without watering-down the experience.

Overall, the data from these course evaluations does not show results for many interesting questions, such as the effect of these RPM projects on Capstone and MS projects that usually follow. While anecdotally several projects each year get translated into MS or undergraduate capstone projects, we lack consistent records to assert how effective the course was in preparing students for these experiences. Similarly, this data on learning outcomes does not include information about project outcomes, or how well the stakeholders found the experience. These are areas for future work.

VI. Conclusion

The aim of this paper is to report and share, CC-PjBL, an innovative approach to applying project-based learning (PjBL) to requirements engineering (RE). We have used CC-PjBL for eight years to educate and train RE & PM students. To evaluate the effectiveness of this approach, we collected both qualitative and quantitative data from EvalTools. Quantitative data analysis proved that CC-PjBL has been effective in achieving instructional objectives and student learning outcomes. Furthermore, this analysis also shows that students are very interested in our RPM course and its instructional approach (CC-PjBL). Data from student overall evaluation shows very good performance by the faculty that taught this course. Moreover, students gave a high rating to their overall experience as well as the quality of our course and its approach. The results obtained from quantitative data is consistent with the positive comments made by students regarding the course.

In addition to the recorded success of our approach, this paper also identified and discussed the challenges encountered in this course and its instructional approach. A key challenge, from the students' perspective, is the higher amount of workload required to complete this course in relation to other courses. This can be attributed to the hybrid nature of this course, the two voluminous textbooks required for the course, and the software requirements development projects. Another key challenge, from faculty perspective, is the difficulty in finding and convincing other faculty in development related courses to participate in this approach. We also observed that most students consider RE and PM related courses boring and would rather take courses where interactive technologies, such virtual reality and mobile devices, are used as methods of instructors.

We will continue to use this approach, and encourage other REE faculty to try CC-PjBL and see if similar results can be achieved. Yet, we acknowledge various areas of improvement which provides scope for future our work. In the future we will carry out more research to identify how we can re-design this course to reduce the significant amount of workload without affecting the learning outcomes and course objectives. Further, we will investigate how virtual realty, wearable technologies, mobile devices, and other interactive technologies can be integrated into our CC-PjBL instructional method, to meet the expectations and demands of students, and also make our course more interesting, and less cumbersome.

References

[1] E. De Graaff and A. Kolmos, "History of problem-based and project-based learning," *Management of change: Implementation of problem-based and project-based learning in engineering*, pp. 1–8, 2007.

[2] M. Savin-Baden, "Challenging models and perspectives of problem-based learning," *Management of change: Implementation of problem-based and project-based learning in engineering*, pp. 9–30, 2007.

[3] S. Mohan and S. Chenoweth, “Teaching requirements engineering to undergraduate students,” in *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’11. New York, NY, USA: ACM, 2011, pp. 141–146. [Online]. Available: http://doi.acm.org/10.1145/1953163.1953207

[4] V. Uskov, M. Yalamanchili, U. Singh, and A. Penumatsa, “Innovative business analysis curriculum for computer science, it and engineering programs,” in *Proceedings of the 46th Annual Frontiers in Education (FIE) Conference*. IEEE Computer Society, Oct 2016, pp. 1–6.

[5] S. Scepanovic and L. Beus-Dukic, “Teaching requirements engineering: Euroweb experience,” in *Proceedings of the 2015 European Conference on Software Architecture Workshops*, ser. ECSAW ’15. New York, NY, USA: ACM, 2015, pp. 38:1–38:5. [Online]. Available: http://doi.acm.org/10.1145/2797433.2797473

[6] R. L. Quintanilla Portugal, P. Engiel, J. Pivatelli, and J. C. S. do Prado Leite, “Facing the challenges of teaching requirements engineering,” in *Proceedings of the 38th International Conference on Software Engineering Companion*, ser. ICSE ’16. New York, NY, USA: ACM, 2016, pp. 461–470. [Online]. Available: http://doi.acm.org/10.1145/2889160.2889200

[7] F. Chen, “From architecture to requirements: Relating requirements and architecture for better requirements engineering,” in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Aug 2014, pp. 451–455.

[8] A. Idri, S. Ouhbi, J. L. Fernndez-Alman, and A. Toval, “A survey of requirements engineering education,” in *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*, April 2012, pp. 1–5.

[9] Z. S. H. Abad, A. Shymka, S. Pant, A. Currie, and G. Ruhe, “What are practitioners asking about requirements engineering? an exploratory analysis of social q a sites,” in *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, Sept 2016, pp. 334–343.

[10] K. Wiegers and J. Beatty, *Software Requirements*. Pearson Education, 2013.

[11] J. C. Nwokeji, T. Clark, and B. S. Barn, “Towards a comprehensive meta-model for kaos,” in *2013 3rd International Workshop on Model-Driven Requirements Engineering (MoDRE)*, July 2013, pp. 30–39.

[12] ——, “A proposal for consolidated intentional modeling language,” in *Proceedings of the Second Workshop on Graphical Modeling Language Development*, ser. GMLD ’13. New York, NY, USA: ACM, 2013, pp. 12–22. [Online]. Available: http://doi.acm.org/10.1145/2489820.2489826

[13] G. Regev, D. C. Gause, and A. Wegmann, “Requirements engineering education in the 21st century, an experiential learning approach,” in *2008 16th IEEE International Requirements Engineering Conference*, Sept 2008, pp. 85–94.

[14] D. Zowghi, “Requirements engineering education and training: Key challenges and practical solutions,” in *2009 17th IEEE International Requirements Engineering Conference*, Aug 2009, pp. 358–358.

[15] G. Gabrysiak, H. Giese, A. Seibel, and S. Neumann, “Teaching requirements engineering with virtual stakeholders without software engineering knowledge,” in *2010 5th International Workshop on Requirements Engineering Education and Training*, Sept 2010, pp. 36–45.

[16] R. Faudou and J. M. Bruel, “An industrial feedback on model-based requirements engineering in systems engineering context,” in *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, Sept 2016, pp. 190–199.

[17] R. N. Memon, S. S. Salim, and R. Ahmad, “Identifying research gaps in requirements engineering education: An analysis of a conceptual model and survey results,” in *2012 IEEE Conference on Open Systems*, Oct 2012, pp. 1–6.

[18] L. Beus-Dukic, “Final year project: A test case for requirements engineering skills,” in *2011 6th International Workshop on Requirements Engineering Education and Training*, Aug 2011, pp. 5–8.

[19] J. C. Nwokeji, T. Clark, B. Barn, V. Kulkarni, and S. O. Anum, “A data-centric approach to change management,” in *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, Sept 2015, pp. 185–190.

[20] S. Frezza, “Issues in student valuing of software engineering best practices,” in *Proceedings of the 46th Annual Frontiers in Education (FIE) Conference*, October 2016.

[21] G. Gabrysiak, M. Guentert, R. Hebig, and H. Giese, “Teaching requirements engineering with authentic stakeholders: Towards a scalable course setting,” in *Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences*, ser. EduRex ’12. Piscataway, NJ, USA: IEEE Press, June 2012, pp. 1–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=2663678.2663679

[22] A. Sampaio, C. Albuquerque, J. Vasconcelos, L. Cruz, L. Figueiredo, and S. Cavalcante, “Software test program: A software residency experience,” in *Proceedings of the 27th International Conference on Software Engineering*, ser. ICSE ’05. New York, NY, USA: ACM, 2005, pp. 611–612. [Online]. Available: http://doi.acm.org/10.1145/1062455.1062565

[23] S. C. dos Santos and F. S. F. Soares, “Authentic assessment in software engineering education based on pbl principles a case study in the telecom market,” in *2013 35th International Conference on Software Engineering (ICSE)*, May 2013, pp. 1055–1062.

[24] N. Hosseinzadeh and M. R. Hesamzadeh, “Application of project-based learning (pbl) to the teaching of electrical power systems engineering,” *IEEE Transactions on Education*, vol. 55, no. 4, pp. 495–501, Nov 2012.

[25] P. Letouze, J. I. M. d. Souza, and V. M. D. Silva, “Generating software engineers by developing web systems: A project-based learning case study,” in *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, April 2016, pp. 194–203.

[26] D. Gernimo, J. Serrat, A. M. Lpez, and R. Baldrich, “Traffic sign recognition for computer vision project-based learning,” *IEEE Transactions on Education*, vol. 56, no. 3, pp. 364–371, Aug 2013.

[27] K. Gary, “Project-based learning,” *IEEE Computer*, vol. 48, no. 9, pp. 98–100, Sept 2015.

[28] S. C. dos Santos, “Pbl-see: An authentic assessment model for pbl-based software engineering education,” *IEEE Transactions on Education*, vol. PP, no. 99, pp. 1–7, 2016.

[29] M. Jazayeri, “Combining mastery learning with project-based learning in a first programming course: An experience report,” in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, May 2015, pp. 315–318.

[30] C. M. Kellett, “A project-based learning approach to programmable logic design and computer architecture,” *IEEE Transactions on Education*, vol. 55, no. 3, pp. 378–383, Aug 2012.

[31] P. Liang and O. de Graaf, “Experiences of using role playing andwiki in requirements engineering course projects,” in *2010 5th International Workshop on Requirements Engineering Education and Training*, Sept 2010, pp. 1–6.

[32] T. Nakamura, U. Kai, and Y. Tachikawa, “Requirements engineering education using expert system and role-play training,” in *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Dec 2014, pp. 375–382.

[33] T. Nkamaura and Y. Tachikawa, “Requirements engineering education using role-play training,” in *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Dec 2016, pp. 231–238.

[34] D. Zowghi and S. Paryani, “Teaching requirements engineering through role playing: lessons learnt,” in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, Sept 2003, pp. 233–241.

[35] S. Frezza and K. Goodman, “Finding mjligheter: Creativity and ill-structured problems,” in *Proceedings of the 124th ASEE Annual Conference and Exposition*, June 2017.