North Dakota State University--Fargo

From the SelectedWorks of Jeremy Straub

May, 2017

Teaching Software Project Management Using Project Based Learning (PBL) and Group Projects

Dr. Jeremy Straub, North Dakota State University--Fargo
Scott Kerlin
David Whalen

Available at: https://works.bepress.com/jeremy_straub/352/
Teaching Software Project Management Using Project Based Learning (PBL) and Group Projects

Jeremy Straub
Department of Computer Science
North Dakota State University
Fargo, ND
jeremy.straub@ndsu.edu

Scott Kerlin
Department of Computer Science
University of North Dakota
Grand Forks, ND
skerlin@cs.und.edu

David Whalen
Department of Space Studies
University of North Dakota
Grand Forks, ND
whalen@space.edu

Abstract—While not included at some institutions and relegated to 3 (minimum) hours of core coverage by the ACM / IEEE Computer Society model curriculum, project management is becoming an integral component of computer science education. The prevalence of failed software projects dictates a focus on this discipline which provides the tools and processes relevant to effective performance of software creation, research and numerous other activities. Computer science students, however, are typically users of these skills and thus benefit from a practical, hands-on-approach that emphasizes learning usable skills over management theory (which they, generally, lack the foundational knowledge for). This paper compares and contrasts three different approaches taken to given students experience with a software project and its management with three different levels of project management emphasis at the University of North Dakota. Their learning relative to pre-defined outcomes was assessed and other sources of benefit were identified.

Keywords—project management; project-based learning; group projects; computer science; education; experiential education

I. INTRODUCTION

The need for computer science undergraduate program graduates to have project management skills is becoming apparent. Studies [1, 2] reporting project failure rates as high as 50% are not new. These expensive failures are not only expensive, they erode public trust in computers and software. While some projects fail due to purely technical considerations (i.e., trying to do something that cannot be implemented), many others fail due to project management issues, including – as Keil, et al. [1] note – project escalation. While project management skills are not a panacea for all of these problems, they are a critical part of the solution. It is not just managers and team leads that require these skills. Most developers are given tasks to perform that are themselves small projects (and would benefit from proper management); others will benefit from understanding the needs of their manager or team lead via an understanding of project management techniques. Consequently, astute students seek more project management experience than the three (minimum core) hours required by the ACM / IEEE Computer Society model curriculum [3].

This paper compares and contrasts three approaches to providing this type of educational experience. All three approaches utilize project-based learning (PBL); however their format and coverage differ. Student learning is compared via course completion statistics as well as the use of pre- and post-tests. Numerous ancillary benefits and increases in student perception of their skills and knowledge related to project management are also discussed.

II. BACKGROUND

Two areas of background inform the work described herein. An overview of these areas, computer science project management need and project based learning is now provided.

A. The Need for Project Management in Computer Science

Several studies [1, 2] have demonstrated the pressing need for a better understanding of project management by computer science practitioners. They discuss project failure rates as high as 50% for information technology and software projects. Notably, many of these failures are attributed to project management issues. Given this, that a problem exists is clear. However, no easy solution to this problem exists. Several factors bear consideration: de Bakker, et al. [4] note that project management knowledge is not being used and Bannerman [5] suggests that project management techniques, themselves, may be creating project risks.

Keil, et al. [1] demonstrate the importance of knowledge of project management techniques and the use of estimation, monitoring and control in preventing escalation (a key source identified for project failure). In fact, they proffer that the use of project management techniques in identified “key areas” may be sufficient to reduce the occurrence of escalation-attributable project failures.

In each of the foregoing, thus, the education of practitioners (and those serving as project managers, perhaps with limited or no applicable training) serves to provide the requisite knowledge and information required to reduce the frequency of and mitigate project risk. Thus, educating students that will be – almost inevitably – required to manage their own work and (perhaps not that much later) the work of others may form part of a solution to this significant problem.
B. Project-Based Learning

Project-based learning (also known as problem-based learning or PBL) is a form of students learning by doing, a form of education that has been used extensively throughout history [6, 7]. PBL has shown effectiveness across educational levels [8, 9] and in numerous fields. Areas of demonstrated utility include project management [10], entrepreneurship [11], computer science [12] as well as numerous engineering sub-disciplines including aerospace [13], computer [14], electrical [15] and mechanical [16]. The use of PBL has been demonstrated to increase student in-program retention [17], retention of knowledge [18], understanding [19] and workforce readiness [19]. It has also been shown to deliver increased levels of student creativity [20] as well as to enhance students’ self-image and motivation [21] and even to aid in students’ job placement [22].

O’Grady [23] notes, however, that PBL’s use in computer science is “shallow”. PBL’s prior use for project management instruction [10] has generally involved projects that are small and self-contained. These factors limit the utility of the approach and the ability of PBL to provide students with hands-on experience relevant to numerous critical project management concepts.

III. LEARNING OUTCOMES

This section discusses the learning objectives relative to project management in the ACM / IEEE model curriculum. All three approaches to teaching CSCI 297 were responsive to these objectives, to varying degrees. The topics covered by the model include [3]:

- Team management
- Scheduling
- Measurement / Estimation
- Risk
- Quality Assurance
- Configuration management / version control / release management
- Project management tools
- Process models

The ACM / IEEE model curriculum also identifies learning objectives, which include [3]:

- Team project involvement (team building / management)
- Project plan creation (including estimation, scheduling, resource allocation, configuration and change management and risk identification / management)
- Determine a risk approach
- Compare / contrast quality assurance techniques

These topics are allocated a coverage time of three hours (three contact hours plus associated out-of-class time) by the ACM / IEEE curriculum.

Three different approaches have been used as part of CSCI 297 to provide students with experience relevant to conducting a real-world software project. Historically, the CSCI 297 course was an opportunity for students to pick a topic (either of their own volition or from those suggested by the course facilitator) and work on it. However, the low success rate in students completing the course using this approach drove consideration of an alternate approach. Under this approach, explicit project management content was included, relating to the following topics:

- Project Initiation / Definition and Planning
- Work Breakdown Structures
- Budgeting
- Control (which is related to several of the model’s points but not directly stated)
- Change Management
- Deliverable Management
- Issue Management
- Additional Risk Management areas
- Additional Quality Management areas
- Project Leadership
- Communications Management
- Expectations Management
- Performance Management
- Conflict Resolution
- Project Closeout

Two different approaches were utilized which covered the foregoing topics. Under the first, students acted as project management consultants to software development teams on an ongoing spacecraft software development project. Incorporating feedback from student participants from the first approach, the second approach allowed students to participate in a project from start to finish. Under this approach, students worked on smaller real-world projects, including the development of a 3D Scanner and classroom response ‘clicker’ software.

Under both approaches, students participated in, learned from and demonstrated knowledge acquisition and understanding via ‘flipped classroom’ style discussions. They also got to put these skills to use in managing their respective project (albeit, the scope of the second approach projects was more limited). The specifics of course implementation are discussed in the following section and assessment is discussed in Sections V (experimental design) and VI (results).

IV. COURSE APPROACHES AND OVERVIEW

CSCI 297 is an open-format course that provides students with experiential learning opportunities. This course is targeted at sophomore and junior level computer science major students. As mentioned in the previous section, three approaches have been taken to the operation of this course. Each is now discussed in more detail.

A. Historic Approach

The first approach was very ill-defined. Students could propose a topic to work on or could ask the course facilitator to
provide one. They would define a scope of work and were expected to perform this work. Limited mentoring was provided by the facilitator and, in some cases, other faculty members who proposed a particular topic.

B. Revised Approaches

The second and third approaches shared several common features. First, they were better defined. The course had a set meeting period each week and students were given topics to prepare to discuss during that meeting period each week. They were also given key project-management related deliverables to produce and work to aid their subject understanding.

Limited lectures were used to provide an overview of the course and set expectations; however, the majority of the course used the PBL technique. Multiple ‘flipped classroom’ discussions were used to gauge student understanding and class preparation. These also served as opportunities for students to discuss their project-specific work and get feedback from their classmates and the course facilitator (in addition to project-specific meetings that were held on a recurring basis).

Both approaches 2 and 3 utilized the textbook Project Management: Absolute Beginner’s Guide (by Gregory M. Horine). This book was selected due to its accessibility and the order in which it presented the material. Also beneficial was the fact that subjects were presented in self-contained sections. The book also incorporates concept maps that provided overviews of each topic area for students to use to gain a quick overview or for reviewing. Students read one or two chapters per week from the text to cover the requisite material during the semester. The book also served as a starting point for those who needed to conduct more detailed research in project-specific areas.

C. Second Approach

For the second approach (the first project management-integrated approach), students selected a spacecraft program software team and acted as a project management consultant to the team lead. In this context, they:

- Created a project definition document
- Created an overview of deliverables and their elements
- Created a work breakdown structure for the group
- Used the work breakdown structure for estimation
- Created a schedule for the group (including dealing with dependent tasks and fixed start/end date tasks, where applicable)
- Used Microsoft Project to build some of the foregoing
- Presented their work as a poster at a local event related to space robotics
- Broke into two-person teams which each developed two operations-phase (risk / issue / change / etc. management plans)
- Adapted their plans based on supplied additional information that they were required to consider
- Provided critical feedback to their classmates via verbal discussions and anonymous feedback cards
- Interacted with team members from the team that they were supporting to gather information and gain buy-in for their proposed plans

Student participants attended a one hour course meeting as well as the weekly meeting of the software team that they were working with. Some students opted to participate in their group beyond the level required for course activities, while others did not.

D. Third Approach

Under the third approach, where students participated in projects from the start, they performed similar tasks, including:

- Creating a project definition document
- Creating an overview of deliverables
- Creating a work breakdown structure
- Using the work breakdown structure for estimation
- Creating a schedule for the group
- Using Microsoft Project
- Presenting work at one or more local conferences

Unlike the second approach, however, where each student was working on deliverables for a separate team, students in this approach were (in some cases) working together on a shared set of deliverables. Students participating in this format were also responsible for implementing the work that they managed (and received a greater number of credits for the course corresponding to this additional workload).

V. EXPERIMENTAL DESIGN OVERVIEW

The experimental design consisted of three primary evaluation mechanisms: a comparison of student retention rates, a pre-test and a post-test. The pre- and post-tests, which were utilized only for approaches 2 and 3, also asked students to characterize their current (at that time) status with regards to several metrics.

The pre- and post-tests were identical. Two test forms were developed using questions from test banks from two higher-level project management textbooks. These necessarily included some questions requiring a depth of understanding beyond the scope of the class, to prevent students from ‘maxing out’ the scale. One-half of the four students were given Form A of the pretest and one-half were given Form B. The students who received each form were recorded. All students were given both forms for the post-test. This facilitates de-confounding differences in the comparative difficulty of the two forms as well as the possibility that students might remember questions from the form that they took as the pre-test and study the information in particular.

Students in approach 2 were also asked to assess the benefits that they received as part of an assessment protocol for the spacecraft program. From areas of benefit, identified in previous work, they were asked to identify which were the most important. The results of these assessment activities are presented in the following section.
VI. RESULTS

The completion rates of students in the three approaches to the course were tracked. Table I presents these results. Note that two values are presented for approach 3. The first includes all students that participated in this approach. The second includes only students that participated in one of the recommended projects. Using a one-tailed t-test, the performance of approaches 2 (p=0.018) and 3-recommended (p=0.018) is significantly better at p < 0.05 (approach 3 is not significantly better with p=0.229).

<table>
<thead>
<tr>
<th>Table 1. Retention Rates for All Approaches.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach 1</td>
</tr>
<tr>
<td>Approach 2</td>
</tr>
<tr>
<td>Approach 3</td>
</tr>
<tr>
<td>Approach 3 - Recommended</td>
</tr>
</tbody>
</table>

Students were asked to assess themselves in regard to five metrics, on both the pre- and post-tests. A Likert-like scale ranging from 9-Expert to 1-Average to 1-Novice was utilized. Questions, phrased in the format “on a scale of 1 to 9, please rate your current level of ___” asked students about their leadership skills, leadership confidence, project management skills, project management confidence and time management skills. Table II presents the pre- and post-test and difference values for approach 2; Table II presents these values for approach 3. It is notable that the difference between the average gains between approaches 2 and 3 (0.2) is not practically significant. In both cases, students reported (on average) gains in all five categories.

<table>
<thead>
<tr>
<th>Table 2. Student-Reported Results for Approach 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
</tr>
<tr>
<td>Leadership Skills</td>
</tr>
<tr>
<td>Leadership Confidence</td>
</tr>
<tr>
<td>Project Management Skills</td>
</tr>
<tr>
<td>Project Management Confidence</td>
</tr>
<tr>
<td>Time Management Skills</td>
</tr>
<tr>
<td>Average</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3. Student-Reported Results for Approach 3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
</tr>
<tr>
<td>Leadership Skills</td>
</tr>
<tr>
<td>Leadership Confidence</td>
</tr>
<tr>
<td>Project Management Skills</td>
</tr>
<tr>
<td>Project Management Confidence</td>
</tr>
<tr>
<td>Time Management Skills</td>
</tr>
<tr>
<td>Average</td>
</tr>
</tbody>
</table>

Students in approach 2 were also asked to identify areas of greatest benefit that they received from participation (this is presented for the whole project in Straub & Whalen, 2013). A relevant assessment tool was not available for approach 3. Figure 1 presents the areas of most important benefit received.

VII. DISCUSSION OF RESULTS

The results presented in the foregoing section demonstrate the attainment of key learning outcomes by students enrolled in the software project management class. The work, as expected, has demonstrated the efficacy of PBL in providing students with knowledge and understanding of key project management concepts and their application in the context of a software project. Among the three approaches, the second (spacecraft program participation) was shown to have the best retention (tied with the approach 3 case where only those significantly better (p=0.017) for approach 2 at p<0.05; however, significance cannot be asserted for approach 3 (p=0.247).

<table>
<thead>
<tr>
<th>Table 4. Evaluation Results for Approach 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
</tr>
<tr>
<td>Form A</td>
</tr>
<tr>
<td>Form B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5. Evaluation Results for Approach 3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
</tr>
<tr>
<td>Form A</td>
</tr>
<tr>
<td>Form B</td>
</tr>
</tbody>
</table>

Fig. 1. Top Areas of Benefit Identified by Participants [24].
participating in recommended projects are included) as well as statistically significantly better performance on the post-test as compared to the pre-test.

The use of the large spacecraft program project as a context in which to perform this work allowed students to perform at all levels of Bloom’s taxonomy [25, 26]. This caused the internalization of knowledge as demonstrated by the improved performance on the unannounced post-test and the artifacts that were created throughout the course of the course. Regrettably, there is no globally accepted standard for which to compare student performance in this discipline against, thus it is impossible to quantify the learning gains of the students in any sort of absolute terms.

It is notable that both approaches 2 and 3 demonstrated improvement in both student perception relative to key metrics. Students in approach 3 enjoyed a greater opportunity to have ‘hand-on’ participation in implementation as well as the opportunity to participate in a project from start to finish; however, their performance on the pre- and post-tests was not found to be statistically significantly different and was less than half of the level of improvement shown for students in approach 2.

The course (in approaches 2 and 3) also presented students with the opportunity to gain leadership experience. A wide variety of other benefits were obtained by students. As demonstrated in Figure 1, real world project experience was identified as a top area of benefit. It was also identified as a key area of benefit sought in the pre-survey by nearly 80% of project participants [27].

Students in both approaches 2 and 3 also identified improvement in their leadership skills and confidence and project management skills and confidence. They gained an average of 1.25 points under approach 2 and 0.93 points under approach 3 (14% and 10% of the 9-point scale, respectively) during their participation. An average gain of 1.5 and 1.38 was reported, for approaches 2 and 3 respectively, for leadership confidence. This was also larger than expected; however, given that this was (for many participants) a first leadership / project management experience, the level of increase in skill and confidence (from untested to demonstrated) is not unexplainable. It is also notable that students perceive that they still have significant room for growth in these areas. The largest single gain, however, was in project management skills: 2.07 average improvement was reported by students in approach 3 (as opposed to only 1.25 improvement in approach 2). Insufficient data exists to determine the attribution for this difference in gain, this will serve as a subject for future work.

VIII. CONCLUSIONS AND FUTURE WORK

This paper has presented a project-based learning approach to teaching software project management, a critical skill for graduates of computer science programs going into industry or pursuing graduate-level studies. It has considered three ways to provide students with hands-on skills in this area and compared them in terms of retention, student perception of their skills and confidence and tested performance. The work has demonstrated the particular efficacy of using involvement in a large group project as a mechanism for imparting these knowledge and skills to students. All three approaches made very limited use of the lecture instructional mechanism and provided significant opportunities for students to learn and internalize knowledge and skills from actually performing project management. However, the retention levels between approach 1 and those from approaches 2 and 3 show the importance of imposing greater structure on participants.

Future work will focus on the longer-term impact of participation in this course and the utility of the concepts presented in both future academic work and in the students’ career entry. The application of this same approach with additional types of projects and projects of different levels of scope is also planned.

ACKNOWLEDGMENT

A limited subset of the material covered herein was presented in a paper entitled “An Experiential Education Approach to Teaching Software Project Management” at the Midwest Instruction and Computing Symposium [24].

REFERENCES


