New Orleans JAZZED about Engineering Education

Paper ID #15229

Evaluating the Collaboration between a Software Project Management Course and a Software Development Course in Terms of Student Learning and Experience

Dr. Stefan Christov, Quinnipiac University

Stefan Christov is an assistant professor of software engineering at Quinnipiac University. He has experience in teaching introductory computer science as well as upper-level software engineering courses, including software quality assurance, software project management, and software engineering in health care. His current research interests include improving the quality of human-intensive processes (HIPs), such as medical processes, with a focus on detecting human errors before harm is done and preventing such errors. He has used software engineering techniques to formally represent and analyze models of complex HIPs and industrial engineering techniques to elicit and validate models of such processes. He is also interested in human-computer interaction techniques for presenting information to assist process performers during an ongoing process. Stefan Christov holds a Ph.D. in Computer Science from the University of Massachusetts Amherst.

Dr. Mark Hoffman, Quinnipiac University

Mark Hoffman is a professor of computer science at Quinnipiac University. He joined the University in 2001 following a career in industry and has taught a wide variety of courses including data structures, computer architecture and organization, software development, and the senior capstone project. His research interests include communication and critical thinking skills in computer science education, and the impact of technology on work/home boundary management. He received his Ph.D. from Polytechnic University in Brooklyn, NY.

Evaluating the Collaboration between a Software Project Management Course and a Software Development Course in Terms of Student Learning and Experience

Abstract

The ability to manage software teams and the ability to productively interact with a project manager are important skills in many computing professions. The integration into the computer science and software engineering curricula of opportunities for students to learn and exercise these skills, however, could be challenging due to various logistical reasons. The work described in this paper evaluates a newly established collaboration between an undergraduate software engineering course on software project management and an undergraduate computer science course on software development in terms of student learning and experience. The contributions of this work are a methodology for evaluating such collaborations and some lessons learned from applying this methodology for one semester.

1 Introduction

The ability to manage software teams and the ability to productively interact with a project manager are important skills in many computing professions. The integration into the computer science and software engineering curricula of opportunities for students to learn and exercise these skills, however, could be challenging due to various logistical reasons. The work described in this paper evaluates a newly established collaboration between an undergraduate software engineering course on software project management and an undergraduate computer science course on software development in terms of student learning and experience.

Software project management (SPM) is an important skill in software engineering. Software engineering management, including subareas such as project initiation and scope definition, project planning and estimation, and project measurement and control, is one of the knowledge areas in the software engineering body of knowledge (SWEBOK [8]). As a result, SPM has been incorporated in some software engineering undergraduate curricula (e.g., [1, 2]).

There is a large body of SPM theory described in the SWEBOK. Finding ways to enhance student learning of this theory by providing opportunities to apply it in a realistic setting, however, is often challenging. This difficulty stems from the fact that to truly exercise project management skills, one needs to be put in charge of a relatively long-term, non-trivial software project that

involves a team of software developers. Setting up such a project and a team for the purposes of a SPM course is logistically difficult. Thus, many SPM courses naturally resort to using an imaginary project where the course instructor provides project materials (e.g., requirements or design documents) and asks students to apply SPM knowledge to generate another set of documents (e.g., effort/size estimates or a project plan). Such imaginary projects provide a venue for students to apply some of the learned aspects of SPM, but they preclude the opportunity to exercise other important aspects of SPM. These other aspects include the enactment of a project plan, and especially the typically iterative, incremental nature of planning and estimation based on what is learned during an ongoing project, as well as the human aspect of project management, including intensive communication with developers and potentially dealing with intricate personality issues.

In addition to the difficulty to provide students with opportunities to exercise management skills, it could also be challenging to provide students with the experience of being managed and maintaining a professional and productive relationship with a manager. Computing curricula typically have students develop software artifacts on their own or in a team of peers, but students rarely work closely with a manager.

To address the above issues, we established a collaboration between a senior-level software engineering course on SPM and a sophomore-level computer science course on introduction to software development (ISD). The seniors in the SPM course serve as managers of teams of sophomores working on semester-long software projects. The managers meet regularly with their teams and apply the theory they learn in the SPM course throughout various phases of the lifecycle of the software system that the sophomore teams are developing/maintaining.

The contributions of the work reported in this paper are two-fold. We propose a methodology for evaluating the collaboration between an SPM course and an ISD course in terms of student learning and experience along with a set of instruments for collecting data to support such an evaluation. We also report some lessons learned from applying this methodology on a one-semester collaboration between an undergraduate SPM course and an undergraduate ISD course.

The rest of this paper is organized as follows. Section 2 briefly discusses related work. Section 3 describes the proposed methodology for evaluating the collaboration between an SPM and an ISD course, section 4 reports the results of applying this methodology, and section 5 discusses these results. Finally, section 6 concludes the paper and discusses future work.

2 Related Work

Establishing a collaboration between SPM courses and other courses in the curriculum has been previously done. For example, Bavota et al. [3] and Maqsood and Javed [10] report on collaborations between graduate SPM courses and other undergraduate courses. Bavota et al. describe the results of collaboratively teaching for several years an undergraduate software engineering course and a graduate SPM course, where the students from the graduate course act as managers for the students from the undergraduate course. The collaboration between the two courses is indirectly evaluated in terms of the size and the quality of the software systems

produced by the undergraduate students by making an inference about the corresponding experience of the undergraduate students. Besides the resulting software systems, a survey is also used to measure the undergraduate students' experience. This survey was administered years after the students took the undergraduate course (not at the end of completing the course project) and a positive general experience is reported. Detail about specific skills or knowledge that the collaboration between the two courses promoted is not provided.

Maqsood and Javed describe the structure and the execution of Practicum in SPM, a graduate course that aids students in learning practical aspects of software project management. Masters students take the software practicum, a two-semester course (1+2 credits), in which they manage the final project of undergraduate students (in teams of up to 5). The Masters students have previously taken a theoretical SPM course. The SPM practicum course is evaluated by comparing its structure to a capability maturity model based on the Software Engineering Institute's Capability Maturity Model Integration [5].

Unlike Bavota et al. and Maqsood and Javed's work, the work described in this paper evaluates the collaboration between two undergraduate courses. Also, we evaluate the collaboration in terms of specific skills and knowledge that the collaboration promotes for students in both courses.

Project management experience has also been introduced within a single course in software engineering curricula. For example, in recent work Malachowsky reports on observations related to introducing a project manager role in projects for an undergraduate Process and Project Management course [9]. A subset of the students in that course volunteer to serve as managers for the team project in the course, where each team needs to complete a paper and deliver a final presentation. Malachowsky reports increased student satisfaction and improved group dynamics after the introduction of the project manager role. In Malachowsky's work, the project managers and the managed students were students in the same course, whereas in the work described in this paper, the project managers and the managed students were in different courses. In Malachowsky's work, the focus of the project was a paper and a final presentation about the findings from the project; in the work we describe here, the focus of the team project was contributing to the implementation of a software system, along with frequent status report presentations, demos, and creation of artifacts to support a software development process.

There has also been work on establishing collaboration across more than two courses in the curriculum. For example, Fenwick et al. [6] study the feasibility of distributing large software engineering projects across the academic curriculum, Tvedt et al. [12] propose that students from different courses collaborate by taking different roles in a simulated software factory, and Walker and Slotterbeck [13] explore the teaching of large scale teamwork in a small college environment by using multi-semester, multi-course projects that require students to work together in teams. The objectives of Fenwick et al.'s, Tvedt et al.'s, and Walker and Slotterbeck's work are broad and related to bringing an entire curriculum closer to real-world software engineering practices whereas the objectives of the work reported in this paper are focused on knowledge and skills particular to software project management.

3 Methods

To evaluate the collaboration between an SPM and an ISD course, we used a combination of methods to collect data related to student learning and experience. In the rest of this section, we first describe the two courses of interest, then the nature of the collaboration between them, and finally the data collection methods we used.

3.1 The *Introduction to Software Development* and the *Software Project Management* Courses

The two courses that are subject of this study are offered in the context of a computer science program and a software engineering program hosted in the same department. Both programs grant only Bachelor of Science degrees in computer science and software engineering respectively. The computer science program is an established program, whereas the software engineering program was recently started and had its first class of seniors during the academic year we performed the study described in this paper.

Introduction to Software Development (ISD) is the third course in the introductory programming sequence for computer science and software engineering majors. It is a required course for both majors and is typically taken during the first semester of the sophomore year. Students in the course work in teams of four or five on a semester-long software development and maintenance project which serves as a vehicle to exercise various software development practices (prior to this course, students have little to no experience with software development processes). For the first 5 weeks of the semester, student teams install and evaluate a software system to which they will contribute, collect customer requirements and prioritize a list of bugs and enhancements, select project management and configuration tools, and develop a preliminary test plan. For the next 8 weeks, teams develop new features and/or fix bugs using Scrum with two week cycles. Student teams make weekly presentations for the first 5 weeks and product demos at the end of each Scrum cycle. Students regularly report the results of their work using several workplace genres to a variety of audiences [7]. In the semester during which we performed the study, there were two sections of this course with 34 students total—16 students in one of the sections and 18 students in the other.

Software Project Management (SPM) is a required course for software engineering majors and should be typically taken during the first semester of the senior year. The course acquaints students with various aspects of software project management. Students learn about project initiation and scope definition; project planning and enactment; measuring and controlling software artifacts and processes; risk management; and human aspects of software project management. Students are given weekly assignments and need to take a midterm and a final exam. Students in the SPM course have taken the ISD course and are thus familiar with Scrum. Fall 2015 was the first offering of the SPM course in the developing Software Engineering curriculum. There were two students enrolled.

3.2 The Collaboration between the Two Courses

The collaboration between the ISD course and the SPM course consisted of having the students from the SPM course serve as managers for the semester-long software project of the teams of students from the ISD course. Each student from the SPM course was assigned to manage two teams. The SPM students were required to meet with their teams weekly, to inquire about the status of the team project, and to provide advice. The SPM students were explicitly instructed, however, that the final decision about the directions their teams take should be made by the team members and not by the managers. The students in the ISD course were expected to attend the weekly meetings with the managers. The collaboration between the two courses was mainly driven by the managers' initiative (based on assignments in the SPM course described below) and occasional requests for advice by the teams from the ISD course.

The assignments of the SPM students (except for one assignment on ethics in software project management) consisted of two parts. One part of the assignment asked the SPM students to meet with their teams, to discuss issues related to the current stage of the team project, and to provide advice, if necessary. The SPM students were expected to provide advice in various areas, such as making suggestions about how their teams should approach customers to elicit project requirements and how to phrase the elicited requirements; helping teams estimate the time and other resources needed to complete various stages of the project; helping teams choose project management and version control tools; and helping teams assess risk. The SPM students were asked to prepare a meeting agenda before and keep minutes during each meeting with their teams.

The second part of the assignments of the SPM students consisted of exercising SPM skills or concepts in the context of the team projects. For example, SPM students were asked to use external size measures to estimate the size of a software feature their teams proposed; to create a work breakdown structures for some activities their teams needed to perform; to identify risk factors and estimate risk exposure associated with the team projects; or to measure aspects of the software artifacts produced by and the development process followed by their teams.

3.3 Data collection methods

This section describes the methods used to collect data related to the collaboration between the ISD and SPM courses.

3.3.1 Data Collection Methods for ISD Course

Final grades. To quantitatively evaluate the correlation between student performance and whether they were on a managed team, we compared the final grades of students on managed teams with the final grades of students on non-managed teams.

End-of-semester survey. We designed two surveys. One was given to students who were on teams with managers and the other to students who were on teams without managers. Both surveys (shown in Appendices A and B, respectively) contained a common part that asked students to

self-assess their achievement of the course learning outcomes on a 5-point Likert scale. The survey given to students from managed teams contained two additional parts. The first additional part asked students whether having a manager contributed to their achievement of each of the course learning outcomes, measured again on a 5-point Likert scale. The second additional part contained open-ended questions about the ISD students' interaction with the managers.

Course project reflection report. At the end of the semester, the students in the ISD course were asked to write a course project reflection report (CPRR). The CPRR (shown in Appendix C) has been used for a number of years to provide students with an opportunity to reflect on the course and on the course project with guiding questions. No questions were added or modified to facilitate students' reflection on having or not having a manager.

3.3.2 Data Collection Methods for SPM Course

End-of-semester survey. The survey given to the SPM students had two parts. The first part asked students whether the interaction with the teams from the ISD course contributed to the achievement of each of the course learning objectives, measured on a 5-point Likert scale. The second part of the survey contained six open-ended questions about the SPM students' interaction with the managed teams. The survey is shown in Appendix D.

Bi-weekly meetings with SPM students. The instructor for the ISD course and the instructor for the SPM course met with the two managers (students from the SPM course) every other week during the semester. The meetings provided the instructors and the managers an opportunity to periodically discuss how the collaboration was working and what needed to be improved. These meetings were informal and conversational in nature.

Minutes from team meetings. The minutes (described in section 3.2) that the SPM students submitted after each team meeting were also used as a source of information to evaluate the collaboration between the two courses.

End-of-semester interview with SPM students. At the end of the semester, the two instructors conducted an exit interview with the SPM students. This interview was guided by the open-ended questions from the survey administered to the SPM students, but allowed the instructors to ask follow-up questions. The SPM students were also given the opportunity to share any observations or suggestions about the interaction with the managed teams.

4 Results

In this section we report the results of the data collected from the Fall 2015 semester using the instruments described above. The results are organized in three sections: the first reports the results for students in the ISD course, the second reports results for students in the SPM course, and the third reports anecdotal observations made by the instructors of each course.

	Managed Team Members	Not Managed Team Members
Number	17	17
Average	89.13	92.38
STD	2.16	2.59

Table 1: Final Grades for Students in Managed and Non-Managed Teams.

	Managed Team	**Not Managed Team**
Participants	16*	16*
Gender	Male: 16, Female: 0	Male: 14, Female: 2
Computer Science Majors	8	8
Software Engineering Majors	4	6
Game Design & Development Majors	6	4
Interactive Digital Design Majors	1	1
Economics Major	0	1

* The total number of majors exceeds the number of participants because some students double major.

Table 2: Demographics of Survey Participants.

4.1 Results from ISD course

4.1.1 Course grades

Table 1 reports the average course grades for students who were members of a "managed" team and students who were members of a "non-managed" team. Students who were members of a managed team had an average course grade of 89.13 while students who were members of a non-managed team had an average course grade of 92.38. The Excel t.test() function assuming a two-tailed distribution with unequal variance yielded a probability 0.05% that the two data sets come from populations with the same mean.

4.1.2 End of semester survey

The survey detailed in section 3.3.1 was administered to all students in the ISD course during the last week of classes. Of the 34 students enrolled in the course, 32 students completed the survey for a response rate of 94%. One student from a managed team and one student from a non-managed team chose not to participate.

Demographics. Table 2 summarizes the demographics of the participants. Both managed and non-managed teams were overwhelmingly male with two females on different non-managed teams. Majors were predominantly Computer Science (CSC), Software Engineering (SER), and Game Design and Development (GDD). A number of students double major in CSC and GDD, and students in GDD and Interactive Digital Design minor in CSC where the ISD course is a required course.

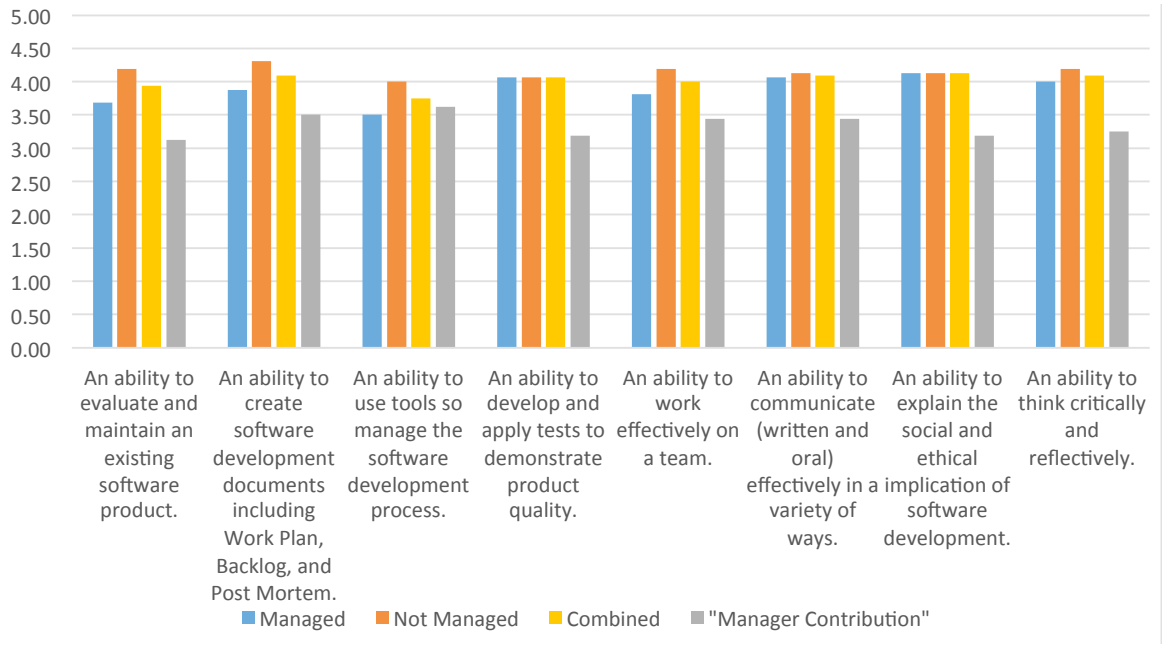

Figure 1: Student Learning Outcomes Achievement.

Student Learning Objectives Achievement. All 32 survey participants answered the question "In the ISD course, I learned the following skills or knowledge" where 16 were members of a managed team and 16 were members of a non-managed team. Figure 1 reports the results for each Student Learning Outcome (SLO) for students on managed teams and non-managed teams, and for all students. None of the differences between the managed and non-managed teams were significant using Excel t.test() function; however, for 6 of the 8 SLOs, the non-managed team result was greater than the managed team.

A second question was asked of students on managed teams only, "Having a manager (student from the SPM course) contributed to my learning of the following skills or knowledge." Of the 16 members on managed teams, 15 completed the question. The results are reported in Figure 1 and labeled "Manager Contribution." The results ranged from 3.13 for "An ability to evaluate and maintain an existing software product" to 3.63 for "An ability to use tools so manage the software development process." All results fell between "Neither disagree not agree" and "Agree".

Engagement with manager (managed team members). Students on managed teams were asked the question, "I engaged with the manager (student from the SPM course) who managed my team by regularly attending meetings and communicating effectively." Of the 16 students on managed teams, 15 completed the question. Twelve students either Agreed or Strongly Agreed with an average response of 4.07 on a 5 point scale.

Open-Ended Questions (managed team members). Students on managed teams were asked "What about having a manager (student from the SPM course) for your team worked best?" Of the 16 students on managed teams, 12 responded. From these respondents, 2 students reported that they found having access to the manager's knowledge of the course valuable. Similarly, 3 students commented that managers were a good resource for questions. The manager's knowledge of

GitHub was useful according to 3 students (as teams started to develop the course project, one of the managers facilitated a GitHub workshop) and 3 students reported that the managers helped keep the teams on track. Managers' knowledge from previous experiences as team members was seen as valuable for this specific issue.

Other questions from the survey. Students on managed teams were asked two questions, "What about having a manager (student from the SPM course) for your team needs to be improved?" and "If you had the opportunity to change how you and your team interacted with a manager (student from the SPM course), what would you change and why?" The first question was answered by 11 students and the second question by 12 students. The respondents are aggregated since they yielded similar responses. Seven students responded that managers need to hold more meetings or make sure all participants attend meetings. Two students commented that managers need to be more engaged in the project planning and development.

4.1.3 Course Project Reflection Reports

All 34 CPRRs were reviewed for comments specific to managers; 10 students made such comments. Of these 10 students, 3 commented explicitly on whether teams should have a manager. Two of these 3 students (both on non-managed teams) recommended that all teams should have a manager, while the third student (who was on a managed team) wrote that the manager was not vital. Of the remaining 7 students, 4 implicitly supported the idea of having a manager, by making suggestions about the responsibilities of a manager. One student on a non-managed team made the following recommendation: "Each team should have a manager to supervise if every team member does their part assigned by scrum master; a team member who does not do their part should be penalized." A student on a managed team recommended that managers should have more authority: "If someone can't get something done, they would have to explain it to the manager, which is a lot like the real world. I think this would make for a fair balance between macro managing and micromanaging, as they can set vague deadlines, which don't necessarily have to be met every single time. I think that's all the manager has to do to become relevant in this class." The remainder of the comments were similar to the improvements and changes suggested through the survey results reported above.

4.2 Results from SPM course

4.2.1 End of Semester Survey

Both of the students in the SPM course agreed that the interaction as managers with teams from the ISD course contributed to their learning of the course SLOs, with the exception of "Ability to develop a software project plan" and "Ability to enact a software project plan" where one or both students disagreed. Through the open-ended questions, the students suggested that managers need to have more authority and suggested possible ways that this might be achieved. Both students commented that the interaction supported their learning of the SLOs "Ability to estimate various software project parameters" and "Ability to lead a diverse team of software developers." The students from the SPM course reported that the ability to lead meetings and to interact with "real

people" on a software project were the most valuable skills/knowledge gained. Further, they reported that version control training and general project guidance were the most valuable benefits for students in the ISD course. When asked about improvements, the students from the SPM course suggested tighter integration between the ISD course and the SPM course assignment and increase of managers' authority.

4.2.2 Bi-weekly meetings with managers

Minutes of the meetings provided the following observations.

- Miscommunication (or lack of communication) among team members was common before the first team presentation in the ISD course. The issue arises again later in the semester for some of the teams.
- Managers provided team members with useful information about requirements gathering and formulation; managers also provided feedback on estimating effort to implement requirements
- Managers provided useful advice for project management and configuration management tools. In particular, managers have pointed out state-of-the art tools and have steered teams away from inappropriate tools (e.g., one team had been planning to use a tool for version control that the manager found out not to be a version control tool)
- Both of the managers provided an overview of different testing methods to their teams and helped them decide which ones to use.
- Managers encouraged reflection among team members and discussions about improving team performance; this was by design of some of the the SPM course assignments

4.2.3 Team meeting minutes submitted by managers

Managers submitted minutes from regular meetings with the teams they managed including a reflection by students in the ISD course collected by managers during their final meeting with the teams. The meeting minutes reinforced several results reported through other methods. The collaboration between the manager and teams helped keep teams focused, coordinated, and clarified team members' responsibilities, and team members appreciated feedback and suggested improvements from their managers. Team members suggested that the role of manager be more clearly defined and some suggested managers become more engaged in brainstorming before each Scrum Cycle and providing feedback after each Scrum Cycle. Generally, the teams reported that the interaction was a positive experience; however, they could have used more help with version control tools, specifically Git.

4.2.4 End-of-semester interview with managers

At the end of the semester, the course instructors interviewed the managers (the students from the SPM course). The managers repeated the need for a clearer definition of the manager's role, and greater and more specific feedback to the teams earlier in the semester. The managers found

meetings with teams and keeping minutes useful for learning how to lead a meeting and how to extract information from team members. The managers considered leading real teams to be a more useful experience than "doing management tasks" on a fictional project. The managers suggested that a team charter could clarify the manager and team's roles, specific team-building activities early in the semester would facilitate the team-building process, and that the assignments for both courses should be more tightly integrated.

4.3 Anecdotal Observations

Throughout the semester the instructors made anecdotal observation based on individual experience with the collaboration in each course and collectively from the collaboration process.

Manager's Authority: When asked at the end of the semester to rate their impact on a team, the managers said it was 3 on a 0-10 scale where 10 is high impact. Their reasons for the rating were that managers felt they were reactive rather than proactive—they answered question rather than directed teams. Originally we minimized managers' authority in grading teams and team members which marginalized their ability to impact team and team member performance. A strategy implemented during the final Scrum cycle where managers were given the responsibility to track teams' report submission as a component of team grading showed promise. In this strategy, students in the ISD course were asked to submit artifacts to their managers instead of to the ISD course instructor; managers were asked to track whether a submission is made before the deadline, to briefly review the submission to verify that its formatting and content adhere to the assignment guidelines, and to forward finalized submissions to the instructor of the ISD course. Other suggestions to vest managers with authority included having managers thoroughly review and evaluate team submissions prior to submission to the course instructor and giving managers more control over planning and tracking tasks of individual team members.

Risk Management: The managers (students from the SPM course) identified risk management as an opportunity to identify, assess, and manage "real" risks. They reported situations where teams attempted to implement a high-risk feature and failed, where the selection of a Scrum Master hindered team performance, and mismatches between team member skills and assigned tasks caused team member frustration and failure.

Managers' Benefit from the Teams from the ISD course: The managers (students from the SPM course) were able to apply project management skills learned in the SPM course. These skills included estimation and planning. The "real" situations provide opportunities to work with "gray area" problems rather than contrived, artificial projects.

5 Discussion

Based on students' answers to direct questions about the collaboration between the ISD and SPM courses (obtained via the end-of-semester surveys, the bi-weekly meetings with the SPM students, minutes from team meetings, and end-of-semester interview with SPM students), on the

information volunteered by students without being explicitly asked (via the course reflection essays in the ISD course), and on the instructors' observations, the collaboration between the ISD course and the SPM course was beneficial for students in both courses[1]. Certain aspects of the collaboration did not work as planned, however, suggesting several areas for improvement in future collaborations between the ISD and SPM courses. In the rest of this section, we first present the main benefits we observed in terms of student learning and experience, we then discuss some issues with the collaboration and potential strategies to address these issue in the future, and finally we discuss the limitations of our study.

5.1 Benefits for students

The main benefits for students in the ISD course were:

- Managers provided ISD students with useful advice on software tools, such as version control and project management systems
- Managers helped with team coordination and with keeping teams focused
- Mangers provided useful feedback, especially on documentation towards the beginning of the semester-long project

The main benefits for the students in the SPM course were:

- Interaction with teams contributed to the learning of the following skills:
  - Ability to determine the scope of a software project by taking into account various constraints
  - Ability to estimate various software project parameters
  - Ability to manage software project risk
  - Ability to lead a diverse team of software developers
- Interaction with teams helped managers learn how to plan and lead meetings
- Interaction with teams provided managers with the opportunity to exercise skills from previous courses, in particular skills related to requirements engineering and quality assurance
- Interaction with teams provided managers the opportunity to experience "soft-skill" aspects of software project management that they would not have experienced without this interaction

5.2 Issues and Lessons Learned

Manager's authority. The most notable issue observed by the instructors and also reported by students in both courses was that the managers (students in the SPM course) had insufficient authority over their teams. As discussed in section 3.2, the students in the SPM course were encouraged to provide advice to their teams, but they were explicitly instructed that decisions about the directions of the team projects should be made by the students in the ISD course. This

[1]The value of the information about the collaboration obtained indirectly (via analysis of course grades and achievement of student learning outcomes) is discussed below.

lack of authority deprived the managers from the ability to direct their teams and, perhaps, hurt the representativeness of the manager-team relationship with respect to such a relationship in a non-educational setting. One of the students from the SPM course reported at the end-of-semester interview that he felt the managers had a reactive role (they were primarily acting as consultants) and not a proactive role as he expected a manager should have.

Furthermore, we did not provide the managers with the ability to evaluate the performance of team members and with the means to effectively encourage good performance or discourage poor performance. This led to situations where managers had difficulty ensuring that all team members attend team meetings or ensuring that each team member performed their duties. Several students from managed as well as non-managed teams in the ISD course indicated that a team should have a manager who should supervise their team's work, should be able to set and enforce deadlines, and should hold team members accountable for their work.

Empowering a group of undergraduate students to evaluate another group of undergraduate students, however, is challenging as it could suffer from various problems. For example, the objectivity of the evaluation could suffer due to conflict of interest for the students who do the evaluation. Given that this was the first time the collaboration between the ISD and SPM courses was attempted at our school, we decided to not allow the managers evaluate the members of their teams precisely to avoid such potential problems. As we gain more experience with and insight about this collaboration, we plan to vest certain degree of authority in the managers, both in terms of directing and evaluating the members of their teams. For example, we are considering allowing the managers to choose a feature their teams must implement during a scrum cycle and the plan for that implementation. Also, we are considering an evaluation scheme, where managers would provide formal feedback on the performance of individual team members and that feedback would be used by the instructor of the ISD course when assigning grades.

Effect of managers on team performance. The lack of authority of the managers resulted in them having little to no influence on the performance of their teams. This observation was confirmed by the students in the SPM course who rated their influence on the performance of their teams as 3 on a scale from 0 to 10 with 10 being highest influence. Thus, we could not use the quantitative data based on final grades of managed vs. non-managed students (presented in Table 1) to confirm or reject hypotheses related to the effect of a manager on team performance (the final grades data are included in this paper for completeness). Similarly, we could not use the data about self-reported achievement of student learning outcomes from students in the ISD course to confirm or reject a hypothesis about the managers' effect on the learning of managed students. We hope that increasing managers' authority in the future would allow us to use such data.

It is interesting to note that students from non-managed teams earned higher average final grade than students from managed teams. This difference is not substantial (it is 3%), but, as discussed in section 4.1.1, it is statistically significant. One possible explanation is that having a manager hurt students' grades. We believe that this explanation is unlikely, however, given the evidence described above that managers had little to no influence on team performance. Another explanation could be the fact that non-managed teams had female students (Table 2), whereas managed teams did not. Recent research suggests that team performance seems to be positively correlated with the proportion of females on a team [14]. Given that there were only two female students in the ISD course, however, we are not confident that the presence of female students

was the main reason behind the difference in final course grades between managed and non-managed students. Another possible explanation is that non-managed teams might have had stronger students during this offering of the ISD course. The effect of managers on team performance is an issue we plan to study further in the future.

Defining the manager's role. Another important issue that surfaced from the collaboration between the ISD and SPM courses was how rigorously the role of the manager should be defined. Our approach during that collaboration was relatively hands-off, allowing the managers and their teams to work out many of the details of their relationship. We made this decision for two reasons: 1) due to the lack of experience with this collaboration in the context of our curriculum, we wanted to avoid over-constraining the relationship without evidence for the constraints we might have imposed; 2) we wanted the managers and their teams to face the challenges of a work environment where the managers need to gauge the amount of oversight they need to provide to their teams based on various factors, such as the team dynamics, the nature of the feature the teams decide to implement, and the stage of the development process.

Our experience with the study indicates that providing a more rigorous definition of the manager's role than what we provided might be useful. One of the managers reported that he had difficulty determining what was expected of him in the beginning of the interaction with his teams and some of the students in the ISD course reported that they wished the duties of the managers were more clearly defined. To address this issue in future collaborations between the ISD and SPM courses, we are considering the use of a team charter that clearly describes the responsibilities of the manager and the team members.

Other lessons learned. Addressing teamwork and leadership early on in the course of the interaction between managers and their teams seems to be necessary. The SPM course was structured in a way such that more technical project management material, such as planning and estimation, was covered earlier in the semester, whereas more "soft-skill" material, such as teamwork and leadership, was covered later in the semester. After getting exposed to this "soft-skill" material, the students in the SPM course noted that they wished they had learned this material in the beginning of the semester. They thought this would have helped them with establishing the relationship with their teams.

Regardless of whether a team in the ISD course was managed or not, communication and teaming issues arose. The instructors observed some interpersonal conflicts in the teams and the managers observed situations of redundant work where more than one team member worked on the same task due to miscommunication. We believe that using a team charter and increasing the authority of managers will help address communication and teaming issues (in particular helping teams overcome the "storming" phase of team development [11]). We are also exploring the possibility to integrate successful project practices observed in academic teams [4].

Closely integrating assignments from the ISD course with assignments from the SPM course and clearly explaining the connection between these assignments to students from both courses might be beneficial in terms of student motivation. One of the managers noted he wished the managers used more of the artifacts produced by the teams they managed and some of the members of the managed teams noted they wished they knew how exactly the managers were using their work. As discussed in section 3.2, the assignments of the SPM students included a component of

applying SPM skills and concepts in the context of the team projects. In the future, we are considering making these assignments accessible to students from the managed teams, so that they can easily see how their work is being used. Also, vesting more authority in the managers should help close the loop, allowing artifacts produced by the managers, such as project plans and risk mitigation strategies, to be used by the managed teams.

5.3 Limitations

The main limitations of the study described in this paper are the small number of students in the SPM course (2 students), the relatively short study duration (1 semester), and the fact that the managers had little influence on the performance of their teams, making it difficult to quantitatively assess results from the impact of the managers.

We plan to continue the collaboration between the ISD and SPM courses to accumulate data from multiple semesters. Based on sophomore and junior class sizes in our program, the number of students enrolled in the SPM course will be substantially higher in the future. Finally, our plan to increase the authority of managers over their teams should increase the impact managers have on team performance, which should allow us to test hypothesis about this impact.

6 Conclusion and Future Work

The initial application of the proposed methodology for evaluating the collaboration between an undergraduate Introduction to Software Development course and an undergraduate Software Project Management course has been promising. This methodology allowed us to gain insight about specific knowledge and skills that the collaboration promotes and also helped us identify several areas for improvement of that collaboration. Overall, the collaboration itself was positive as it seems to have been beneficial for students in both courses.

We plan to continue the collaboration between the ISD and SPM courses. In the next offering of these two courses, we plan to make modifications based on the lessons learned from the study reported in this paper. In particular, we intend to increase the authority of the managers, more clearly define the responsibilities of the managers and of the members of the teams they manage by introducing a team charter, teach teamwork and leadership earlier in the SPM course, and more closely integrate the assignments in the ISD and SPM courses.

The enrollment in the SPM course is expected to substantially increase, which should provide us with larger sample size. We plan to apply again the evaluation methodology described in this paper and to potentially refine it to specifically measure the effect of the course modifications we are planning to implement.

The abilities to manage software teams and to productively interact with a project manager are important skills in many computing professions, but it is often difficult to nurture such skills in academia. Collaborations, such as the one described in this paper, can help students exercise these skills and the proposed methodology can help evaluate and subsequently improve such collaborations in terms of the most important outcomes—student learning and experience.

Acknowledgments

The authors gratefully acknowledge the contributions of Wesley Breisch and Michael Podias who participated in regular discussions about the collaboration between the ISD course and the SPM course and provided useful insights on that collaboration.

References

[1] Software Engineering Curriculum, Rochester Institute of Technology. https://www.se.rit.edu/content/curriculum-overview-0.

[2] Software Engineering Curriculum, Rose Hulman University. http://www.rose-hulman.edu/course-catalog/course-catalog-2015-2016/programs-of-study/software-engineering.aspx.

[3] G. Bavota, A. De Lucia, F. Fasano, R. Oliveto, and C. Zottoli. Teaching software engineering and software project management: An integrated and practical approach. In *Proceedings of the 34th International Conference on Software Engineering*, ICSE '12, pages 1155–1164, Piscataway, NJ, USA, 2012. IEEE Press.

[4] C. Beise, T. A. Carte, C. Vician, and L. Chidambaram. A case study of project management practices in virtual settings: Lessons from working in and managing virtual teams. *SIGMIS Database*, 41(4):75–97, Nov. 2010.

[5] CMMI Product Team. CMMI for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2006.

[6] J. B. Fenwick, Jr. and B. L. Kurtz. Intra-curriculum software engineering education. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '05, pages 540–544, New York, NY, USA, 2005. ACM.

[7] M. E. Hoffman, P. V. Anderson, and M. Gustafsson. Workplace scenarios to integrate communication skills and content: A case study. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 349–354, New York, NY, USA, 2014. ACM.

[8] IEEE Computer Society, P. Bourque, and R. E. Fairley. *Guide to the Software Engineering Body of Knowledge (SWEBOK®): Version 3.0*. IEEE Computer Society Press, Los Alamitos, CA, USA, 3rd edition, 2014.

[9] S. A. Malachowsky. Implementing project managers in the software engineering classroom. In *2015 ASEE Annual Conference and Exposition*, number 10.18260/p.24249, Seattle, Washington, June 2015. ASEE Conferences. https://peer.asee.org/24249.

[10] M. Maqsood and T. Javed. Practicum in software project management: An endeavor to effective and pragmatic software project management education. In *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*, ESEC-FSE '07, pages 471–480, New York, NY, USA, 2007. ACM.

[11] B. W. Tuckman. Developmental sequence in small groups. *Psychological Bulletin*, 63(6):384–399, 1965.

[12] J. Tvedt, R. Tesoriero, and K. Gary. The software factory: combining undergraduate computer science and software engineering education. In *Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on*, pages 633–642, 2001.

[13] E. L. Walker and O. A. Slotterbeck. Incorporating realistic teamwork into a small college software engineering curriculum. *J. Comput. Sci. Coll.*, 17(6):115–123, May 2002.

[14] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone. Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330(6004):686–688, 2010.

Appendix A. End-of-semester Survey Administered to Students from the Introduction to Software Development Course Who Were Not Part of Managed Teams

Demographics

1. Major (Check all that apply.)
2. Minor (Check all that apply.)

Course Objectives

3. In this course, I learned the following skills or knowledge. (5 point "Strongly Disagree" to "Strongly Agree" scale)
   a) An ability to evaluate and maintain an existing software product.
   b) An ability to create software development documents including Work Plan, Backlog, and Post Mortem.
   c) An ability to use tools to manage the software development process.
   d) An ability to develop and apply tests to demonstrate product quality.
   e) An ability to work effectively on a team.
   f) An ability to communicate (written and oral) effectively in a variety of ways.
   g) An ability to explain the social and ethical implication of software development.
   h) An ability to think critically and reflectively.

4. Are there other skills or knowledge (beyond those listed in the previous question) that you learned in this course? If so, describe each in the space provided below. (Open-ended)

Appendix B. End-of-semester Survey Administered to Students from the Introduction to Software Development Course Who Were Part of Managed Teams

Demographics

1. Major (Check all that apply.)
2. Minor (Check all that apply.)

Course Objectives

3. In this course, I learned the following skills or knowledge. (5 point "Strongly Disagree" to "Strongly Agree" scale)
   a) An ability to evaluate and maintain an existing software product.
   b) An ability to create software development documents including Work Plan, Backlog, and Post Mortem.
   c) An ability to use tools to manage the software development process.
   d) An ability to develop and apply tests to demonstrate product quality.
   e) An ability to work effectively on a team.
   f) An ability to communicate (written and oral) effectively in a variety of ways.
   g) An ability to explain the social and ethical implication of software development.
   h) An ability to think critically and reflectively.

4. Are there other skills or knowledge (beyond those listed in the previous question) that you learned in this course? If so, describe each in the space provided below. (Open-ended)

Managed Teams

5. Are you a member of one of the teams that had a manager (team A1, B1, A3, or A4)? (Yes/No)

If yes to question 5, answer the following questions

6. Having a manager contributed to my learning of the following skills or knowledge. (5 point "Strongly Disagree" to "Strongly Agree" scale)
   a) An ability to evaluate and maintain an existing software product.
   b) An ability to create software development documents including Work Plan, Backlog, and Post Mortem.
   c) An ability to use tools so manage the software development process.
   d) An ability to develop and apply tests to demonstrate product quality.
   e) An ability to work effectively on a team.
   f) An ability to communicate (written and oral) effectively in a variety of ways.
   g) An ability to explain the social and ethical implication of software development.
   h) An ability to think critically and reflectively.
7. Are there other skills or knowledge (beyond those listed in the previous question) that you learned by having a manager? If so, describe each in the space provided below. (Open-ended)
8. I engaged with the manager who managed my team by regularly attending meetings and communicating effectively. (5 point "Strongly Disagree" to "Strongly Agree" scale)
9. What about having a manager for your team **worked best**? (Open-ended)
10. What about having a manager for your team **needs to be improved**? (Open-ended)
11. If you had the opportunity to change how you and your team interacted with a manager, what would you change and why? (Open-ended)

Appendix C. Course Project Reflection Report for Students from the Introduction to Software Development Course

Course Project Reflection Report

At the end of your professional role in a project, it is important to reflect on the contributions you made, what you learned, and what you need to change or improve as you move on to your next project. To accomplish this objective, you need to write a reflection paper that reports individual experience working on the course project and assesses individual achievement toward student learning outcomes achievement. Use Weekly Individual Project Reflections, individual Time Sheets, and Scrum Cycle Reports among other documents as evidence to support experience and achievement statements. Assessment the course and course project including what worked well and what did not. For those aspects of the course or course project that did not work well, suggest ways to improve them. In your Course Project Reflection Report you need to consider the following questions.

- What was the *most important thing you learned* in this course this semester? Please explain why you consider this to be most important. Once you complete this, you may add other things you consider important and why.

- What was your *most significant contribution* to the project and the course? Please explain why you consider this to be the most important. Once you complete this, you man add other contributions you consider important and why.

- If you could *change one thing about the course*, what would it be? Please explain why you would change this thing and why it is the one you selected. Once complete, you may suggest other changes and why.

- What Student Learning Outcome (see the syllabus) did you make the most progress toward achieving? Provide evidence from course project documents to support your claim.

- What Student Learning Outcome (see the syllabus) did you make the least progress toward achieving? Provide evidence from course project documents to support your claim.

- Assess your progress in improving your communication skills: reading, writing, speaking, and teaming. (Student Learning Outcomes are coded with RSWT in the course syllabus.) For each communication skill, provide evidence from course project documents to support your claim.

Appendix D. End-of-semester Survey Administered to Students from the Software Project Management Course

Course Objectives

1. The interaction with the teams from the ISD course (in person or via assignments that involved artifacts related to the teams' project) contributed to my learning of the following skills (5-point Likert scale from "Strongly Disagree" to "Strongly Agree"):

- Ability to determine the scope of a software project by taking into account various constraints
- Ability to develop a software project plan
- Ability to enact a software project plan
- Ability to estimate various software project parameters
- Ability to measure and control software products and processes
- Ability to manage software project risk
- Ability to lead a diverse team of software developers

Additional questions

2. Are there other project management skills/knowledge (beyond what is listed above) that you learned from the interaction with the teams? If so, what are these skills/knowledge?
3. What contributions of yours do you think were most helpful to the teams?
4. Overall, what worked in the interaction with the teams?
5. What didn't work in the interaction with the teams?
6. If you had the opportunity to change the interaction with the teams, how would you change it and for what reasons?
7. Is there something that the professors of the ISD course and the SPM course could do to make the interaction with the teams more productive and/or useful for your learning of course concepts?