Journal of Information Systems Education

Volume 35
Issue 2
Spring 2024

A Process-Focused Experiential Approach to IT Education Using Peer Learning

Jeremy St. John and Karen St. John

Recommended Citation: St. John, J., & St. John, K. (2024). A Process-Focused Experiential Approach to IT Education Using Peer Learning. *Journal of Information Systems Education*, 35(2), 148-159. https://doi.org/10.62273/ORES9068

Article Link: https://jise.org/Volume35/n2/JISE2024v35n2pp148-159.html

Received: June 26, 2022
First Decision: August 19, 2022
Accepted: August 16, 2023
Published: June 15, 2024

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 35(2), 148-159, Spring 2024
https://doi.org/10.62273/ORES9068

A Process-Focused Experiential Approach to IT Education Using Peer Learning

Jeremy St. John
Norris-Vincent College of Business
Angelo State University
San Angelo, TX 76909, USA
jeremy.stjohn@angelo.edu

Karen St. John
School of Information Technology and Computing
Abilene Christian University
Abilene, TX 79601, USA
karen.stjohn@acu.edu

ABSTRACT

This study provides an overview of efforts to improve experiential learning outcomes by integrating the curriculum of an upper-level Project Management (PM) course with an Introductory Programming (IP) course using a game-making project. Students in the PM course applied PM methods and techniques while supervising teams of students in the IP course in making a gaming app. Paradoxically, it was found that transitioning from complex, real-world client projects to more structured and guided student projects increased realism and enhanced experiential learning. Positive results were not automatically derived by employing experiential learning tools. More important was a focus on the process through which student experiences resulted in knowledge acquisition and creation. The findings of this study encourage educators to pay special attention to the implementation and integration of learning tools to create synergy in the learning process.

Keywords: Experiential learning & education, Game-based learning, Peer learning, Project management, Introductory programming

1. INTRODUCTION

Gartner's 2019 survey of senior executives found that 63% of respondents cited concerns about a talent shortage, which could pose considerable risk to operations (Lavelle, 2019). Project Management Institute's (PMI's) 2017 Project Management Job Growth and Talent Gap Report corroborates this trend, revealing that the global economy has become increasingly project-based; it is projected that demand for skilled project managers will grow by 33% through 2027 (PMI, 2017). Simultaneously, as the need for skilled workers outpaces the supply, employers have shifted the competencies they seek in IT workers, with an emphasis on soft skills in addition to technical skills (Dawson & Thomson, 2018). This misalignment between the skillsets students acquire in their education and the skills employers are seeking is one cause of the IT skills gap (Draus et al., 2022; Taylor-Smith et al., 2019).

Several researchers have reported results that address this issue. In their qualitative study on the gap between project management (PM) education and requisite PM for success in the contemporary complex work environment, Ramazani and Jergeas (2015) found a major disconnection. They identified a need to shift the focus from teaching technical skills to fostering interpersonal and leadership skills. Further, they recommended that PM education focus on critical thinking abilities and preparing students for the complexities of real-life projects. Additionally, Magano et al. (2020) identified some of the soft skills employers are seeking, such as leadership, problem-solving, communication, negotiation, and teamwork. Considering both the IT skills gap and the rising demand for more soft skills alongside technical skills, there is ample justification to reconsider IT education and training. In recent years, experiential learning techniques have been touted as the solution for addressing this skills gap. These methods typically involve students working through the same or very similar complex real-world problems that professionals work through. The acceptance of experiential learning as a key solution for keeping education relevant is evident in AACSB's 2020 Curriculum Standard 4.3, which requires that the "school provides a portfolio of experiential learning opportunities" (AACSB, 2020, p. 43).

The use of experiential learning tools and methods in IT education and training is not a new phenomenon, and techniques have been refined over the decades. Programming course instructors have long used experiential learning techniques. These techniques typically take the form of programming assignments where students learn on their own by writing code, making mistakes, and correcting those mistakes.

However, the problem with this unguided "sink or swim" approach is that students can become overwhelmed trying to troubleshoot these errors independently. Notably, IP courses have traditionally suffered from high dropout and failure rates (Malik & Coldwell-Neilson 2017; Zingaro & Porter, 2014). Watson and Li (2014) found a fairly consistent mean worldwide pass rate of only 67.7%. IP students can easily lose motivation due to the complexities of learning programming (Dawar, 2021). The level of student success or failure is often a reflection of motivation (Alturki, 2016).

Experiential learning techniques have also been applied in IT PM courses, typically through team-based real-world client projects. Moreover, game-making is an experiential learning technique whereby students learn by creating games. Several benefits are associated with client projects and game-based learning in IT education. Karanja and Grant (2020) suggest that a learner-centered pedagogy increases enthusiasm in PM coursework. Koivisto and Hamari's (2018) review of the literature reveals that enjoyment and fun, intrinsic elements of motivation, are the second most frequent outcomes of game-based learning studies. Tynjälä et al. (2009) demonstrated that client projects require students to address complex issues while negotiating team dynamics, which improves their critical thinking and communication skills.

Game-based learning has historically provided realism in learning while also increasing motivation (Geithner & Menzel, 2016). However, much like client projects, game-based learning, particularly simulations, has high development costs and is extremely time-consuming to implement and manage (Bronner & Kollmannsperger, 1998; Geithner & Menzel, 2016). In a PM class, these team-based experiential learning projects can be difficult to assess and may prove burdensome in terms of time requirements, scalability, and costs. In their content analysis of PM course syllabi at AACSB-accredited universities, Karanja and Grant (2020) found that only 7% of syllabi included any mention of "real world" or experiential learning experiences. These implementation difficulties may help explain their lack of use in the classroom. Similarly, difficulties novice programmers face in implementing programming may also explain the poor outcomes associated with IP courses.

At the onset of this study the most popular experiential learning tools had been implemented in previous semesters with limited improvements in terms of outcomes and student satisfaction. Rather than asking what new experiential tools should be employed to improve outcomes and student satisfaction, this study asks how should these tools be implemented and integrated to enhance the process of transforming experience into knowledge.

To address this, a team-based game-making project using peer learning was devised, with a focus on process. Peer learning, a form of collaborative learning, involves students working in pairs or small teams to find solutions to problems. Zhang et al. (2020) reported a number of unexpected benefits from having peer tutors in the classroom, including a noticeable improvement in the classroom atmosphere. However, most studies on peer learning have only studied the impact on mentees, with little research examining the challenges and benefits from the mentor's perspective (Carvalho & Santos, 2022; Marshall et al., 2021).

Accordingly, for this study's purposes, data was collected and analyzed from the viewpoint of peer mentors. The implementation heavily relied on peer learning, a form of active learning pedagogy rooted in constructivism (Falchikov, 2001).

The current study examines curriculum changes made at a small, private university, reflecting the trend among most universities today that offer several computer-related degrees. These degrees include Computer Science (CS), Digital Entertainment Technology (DET), Information Technology (IT), and Information Systems (IS). An integrated project was devised, extending across two interdisciplinary courses, a lower-level IP course and an upper-level PM course. A semester-long team project required teams to plan, design, code, and create a simple video game. Professors from each of the computer-related degree programs who taught basic coding and design techniques throughout the semester co-taught the IP course. Meanwhile, an IS professor who covered PM tools and techniques taught the PM course.

The next section focuses on theory and provides a brief overview of experiential learning and Experiential Learning Theory (ELT). Then, observations are made regarding teaching both IP and PM courses, including a description of the challenges identified in both courses. The implementation and results are followed by the conclusion.

2. THEORY

Experiential learning can be defined as a pedagogical approach that helps students apply knowledge and conceptual understanding to real-world problems (Wurdinger & Carlson, 2010). A broad definition of experiential learning can be understood as learning from experience or learning by doing.

In addition to AACSB's requirement that schools integrate experiential learning, AACSB's 2020 Curriculum features Standard 4.1, which demands that the curriculum content cultivates agility with current and emerging technologies (AACSB, 2020). Undoubtedly, technology's role in experiential learning experiences is often central. Students and instructors typically interact directly with the technology, often obscuring the implementation and learning process. This scenario has the danger of focusing on the technology and tools while neglecting the implementation and learning process.

Kolb and Kolb (2005) highlight that Experiential Learning Theory (ELT) defines learning as "the process whereby knowledge is created through the transformation of experience" (p. 194). More simply stated, experiential learning can be thought of as a method of learning by doing as opposed to passive learning methods such as listening to a lecture. This definition stresses the importance of processes over tools. Kolb and Kolb (2005) state that "experiential learning is often misunderstood as a set of tools and techniques to provide learners with experiences from which they can learn" (p. 193). Given this consideration, the focus shifts from choosing experiential learning tools that work best to choosing strategies that best enhance the process of transforming experience into knowledge. This study employs a process that combines courses and fosters peer learning. This methodological modification, based on peer learning, includes synchronous course scheduling, a combination of instructor-led and project-specific classroom time, and team-based presentations.

Experiential learning approaches within PM typically include a group project that can be combined with other techniques and approaches, as Kruck and Teer's (2009) interdisciplinary group project illustrates. Game-based pedagogies are becoming more attractive to educators because of their scalable and assessable nature compared to other methods of providing "real-life" experiences. They are also less expensive to implement and manage than real-life client problems or projects, saving time and making them more practical. This factor explains why simulation games are being increasingly used in PM classes (Lee, 2016). Several different constructivist learning approaches use a game-based pedagogy, such as learning by using educational games, learning by using entertainment games, learning by making games, or using gamification in learning (Nousiainen et al., 2018).

Constructivism is a learning approach that actively encourages students to use their experiences in the world to build knowledge (Gaeta et al., 2019). An integral component of constructivism is that it empowers individuals to construct and develop their own knowledge (Bakan & Bakan, 2018). Game-making, used in this study, is a constructivist approach that involves learning through creation and design (Gaeta et al., 2019; Kafai, 2009). This approach shares the benefits of other experiential learning tools, including improved creativity and problem-solving skills (Gaeta et al., 2019). Similar to the current study, most constructivist gaming approaches to learning have focused on teaching programming (Kafai & Burke, 2015). Notably, game-making, a distinct variant of game-based learning, has also been found to be enjoyable and motivating for students (Dalal, 2012).

Teague and Roe (2008) found that collaborative learning could potentially make studying programming more engaging, interactive, and enjoyable. For an IP class, the amalgamation of this "fun" factor from collaborative learning and the enjoyment of a game-based project could improve students' attitudes and motivation. Moreover, Vogel et al.'s (2006) meta-analysis of 32 studies on the use of computer gaming and interactive simulations for learning found enhanced cognitive and attitudinal results from using games for learning. Additionally, Kafai and Burke (2015) noted improvements in attitudes toward learning, specifically for students involved in game creation.

The process of game creation requires that students solve unstructured, complex problems. The successful ability of students to solve unstructured, complex problems has been associated with improved communication, problem-solving, and critical thinking skills (Nousiainen et al., 2018; Yang & Chang, 2013). Given the improvements in problem-solving, critical thinking, and learning attitudes, game-based learning is an optimal choice for teaching IP. However, the benefits associated with game-based learning are not automatically realized through its introduction in the classroom, especially regarding student motivation. For instance, Wouters et al. (2013) found no evidence that the use of games motivated students more than conventional instruction, despite games being found to be more effective for learning. In another meta-analysis of studies on the use of games for learning, Clark et al. (2016) found that, compared to non-game methods, game-based learning significantly enhanced student learning. That meta-analysis revealed that much of the research focused on media comparisons. While acknowledging the importance of the medium, their results underscored the importance of design within the medium. They concluded that design determines a learning environment's ability to produce the desired result.

3. TEACHING

3.1 Teaching Introductory Programming

Programming is commonly taught by having students program, a typical constructivist "learning by doing" approach. However, the problem with this intuitive approach is that it exposes students to programming problems, client projects, and other experiential techniques in IT education that tend to be quite complex, leading to frustration and failure. The use of this traditional approach in teaching programming is often referred to as a "sink or swim" approach. Some instructors deem this approach appropriate, anticipating that the lower-performing half of the class will be weeded out when, given little support or guidance, students cannot demonstrate proficiency in programming (Argent et al., 2006). The typically high student-to-teacher ratio in IP classes exacerbates the problem of students receiving little support or guidance, potentially leaving instructors feeling overwhelmed. Most learning theorists agree that the unguided "sink or swim" approach is not effective, proposing that complexity in IT education actually increases the need for guidance (Guzdial, 2009; Merrill, 2002).

3.2 Challenges Teaching IP

With failure rates often as high as 50% (Margulieux et al., 2020), innovative teaching methods may help reduce these statistics. For example, Chase and Okie (2000) reduced an IP course's withdrawal/fail rate by 32% by applying peer learning. Additionally, Porter et al. (2013b) dramatically improved failure rates by implementing peer learning. IP students generally report overwhelming satisfaction with peer tutors and perceive it as beneficial to their learning experience (Crabtree et al., 2022).

Working individually, students frequently "get stuck," which wastes time and leads to frustration. Conversely, working in groups and in peer learning has been shown to positively impact student failure rates in IP courses (Bakare & Orj, 2018; Porter et al., 2013a). Peer learning not only encourages students to engage in the learning process but also provides guidance and feedback, allowing for more opportunities for students to receive immediate feedback (Zhang et al., 2020). Teams are considered an important aspect of software development efforts because they have been shown to enable coordination, collaboration, and communication (Sharp & Robinson, 2010). Software development teams have also been shown to improve both learning and project success (Janz, 1999; Lindsjørn et al., 2016). In the gaming context, group work has outperformed individual work in fostering learning (Wouters et al., 2013).

Leveraging cooperation by having students work in groups to solve programming problems may address the lack of guidance. However, it is difficult for instructors to assess individual contributions. Teams are typically assessed together, which encourages free riders (El Massah, 2018). Free riders contribute less than their colleagues (Dyrud, 2001) and are the most often mentioned disadvantage of group work—their loss of productivity and lack of contribution negate the benefits of teamwork (Roberts & McInnerney, 2007). Past research has indicated that the free rider problem not only involves some members being less productive but also results in engaged and performing team members who stop trying in response to receiving the same grades as free riders on the team (Lee & Lim 2012; Lin, 2018; Narmaditya et al., 2022). Furthermore, free riders are more of a problem when the group exhibits more diversity in skills, experience, and ability, as is often the case with interdisciplinary IP (Sanz-Martinez et al., 2019). Instead of fostering positive outcomes with peer learning, the result is social loafing, which occurs when individuals perform less and learn less in a group than they would individually (Loughry et al., 2007). For example, learning to write code often occurs by making and correcting errors, tasks or effort that might be avoided by free riders.

Changes to the IP class were made as part of the ongoing Assurance of Learning (AOL) process. For example, in response to complaints from DET majors that the programming course lacked relevance for them, a game-making project was introduced in the fall of 2014 to add an element of fun to the IP course and make it more relevant for DET majors given the DET program's focus on game development. Although IP students enjoyed this project, they often expressed excessive frustration with team dynamics, particularly the before-mentioned free-rider problem, where some students did most of the work while others hardly did any.

It became clear to the instructors that students needed more guidance than the instructors could provide, given the large class size. One proposed solution was made by designating one student as a team leader. Unfortunately, first-year team leaders lacked experience in PM and had difficulty managing the team. Another attempt involved encouraging senior students to mentor first-year students in the IP course voluntarily. However, it soon became clear that the leadership ability among the volunteers varied widely, and few seniors were capable or empowered to manage the projects effectively. In many cases, volunteer seniors took the strain off the teams by doing work intended for team members. Instructors felt that this approach was undermining student learning in the IP class and was unfair for the volunteers by placing an unreasonable burden on them.

Over the course of three years of implementing the mobile app/game group project in the IP class, a common theme emerged. Student comments were positive regarding the project assignment, noting that it was fun and interesting. Negative comments mostly addressed team dynamics and aspects of team management. Learning was inconsistent among students because of varied skill sets and imbalanced workloads. Despite senior volunteer leadership, the problem persisted: some individuals would shoulder most of the work while others slacked off. The common problem across teams was not related to coding or creativity but was attributed to poor PM.

3.3 Teaching PM

Project Management (PM) education needs to adapt rapidly to accommodate growing demands. To illustrate, PMI's (2017) Project Management Job Growth and Talent Gap 2017–2027 report noted that the demand for project managers over the next decade is expected to grow faster than the demand for workers in other occupations. As project work in IS continues to displace routine business operations, PM skills continue to grow in significance and are seen as a vital asset for graduates (Venkatesh & Maruping, 2017). Success in PM relies on soft skills and teamwork (Sabin et al., 2017). This emphasis on soft skills rather than hard skills is only expected to increase with AI and robotics. Nimmo and Usher (2020) found that the impacts of these technological advancements will potentially necessitate significant changes in requirements for PM education and practice, specifically a dramatic shift in focus from hard to soft skills.

Advanced PM courses teach a complex mix of hard and soft skills, including communication, critical thinking, leadership, collaboration, and teamwork. Client projects often serve as simulations for soft skills in the "real world" but fall short in key areas. Specifically, within the classroom context, teams are homogeneous, and only one student per team can actively practice PM skills as the project manager.

3.4 Challenges in Teaching PM

Notably, the PM course involved in this study had the same fundamental problem many PM courses have—teaching students real-world soft skills to meet industry needs. This course used the client project model of instruction, whereby student teams chose from several real client projects. Student teams worked on these projects throughout the semester and presented their final projects to the class at the end. The PM course in this study used client projects to create a realistic environment conducive to experiential learning.

Much like programming, many PM concepts are difficult to teach. For example, Tabatabaei (2014) notes that in teaching PM, the most significant and frequently mentioned negative factor is teamwork issues. Moreover, Pollard (2012) recommends teaching PM soft skills, including effective communication, leadership, collaboration, analytical skills, and problem-solving using real-world situations in a learner-focused environment.

In the real world, projects are temporary and created to address unique situations. Such projects usually involve interdisciplinary teams composed of individuals with diverse backgrounds, functional roles, and varying degrees of skill and experience. Often, these individuals are unaccustomed to collaborative work. An example might be an outspoken salesperson and designer working with introverted software engineers to create an application. The difficulties encountered with diverse teams allow students to learn soft skills. The challenge for the PM instructor lies in simulating an inexperienced, difficult, and diverse team, enabling each PM student to learn how to manage such a team. Although client projects are realistic, they lack realism in the classroom environment, primarily due to problems stemming from team composition. Teams assigned to client projects consist of homogeneous senior IS majors. Only one student on the team assumes the role of project manager, while the other students, being familiar with expectations, play along. Consequently, the student performing the role of team leader has less authority, making handling conflicts between team members more difficult.

Despite incorporating real-life client projects into the PM course, the teams were imbalanced, with an overabundance of project managers assigned to manage each other. It was an unrealistic, demotivating problem of "too many generals and no foot-soldiers." Conversely, the IP course had the opposite problem of "too many foot-soldiers and no generals." The client project in the PM course focused on the tool, represented by a real-life project. However, as Kolb and Kolb (2005) note, the problem can be a focus on tools rather than processes. ELT requires a focus on the process of converting experience into knowledge. The game-making project is arguably a less realistic learning tool. However, the combining of courses revealed that both classes had a better learning experience because the project experience was more realistic.

The previous PM course design employed experiential learning tools such as group client projects, but it failed to realize the full potential of experiential learning. The cost of implementing real client projects in both time and effort for the students and the instructor was very high. For instance, communications had to be maintained between not only clients and student teams but also between clients and the instructor. Teams collaborated with several clients on different projects in various industries. While clients appreciated the work that students invested into the projects, they understandably withheld sensitive or private project information, making PM decisions more difficult. As Richmond et al. (2008) note, these problems were fundamentally of design. From a theoretical perspective of using ELT, experiential learning was misunderstood as a tool to be implemented rather than a process of creating knowledge through experience (Kolb & Kolb, 2005).

4. IMPLEMENTATION AND RESULTS

4.1 Implementation

It is necessary to recognize the prior problems associated with the PM and IP courses to understand the process changes employed to address them. Table 1 lists the problems encountered within the PM Teams, IP Teams, and Course Structure over the three-year period prior to the semester when process changes were implemented. Table 1 also includes the associated process changes (solutions) made to address these problems.

As previously stated, experiential learning is learning by doing. For PM students, this learning paradigm meant managing a team working on a project. After combining the courses, the IP class served as the source of teams for the PM students to manage. This arrangement effectively addressed the problem in the PM course where teams consisting of four to five managers managed each other. Additionally, first-year students, who typically have little experience working on teams, were naturally good at simulating real-life interpersonal team problems for the PM students to manage. For first-year students in the IP course, their experiential learning entailed learning by writing code and creating a game. They also learned how to be managed. An indirect benefit of the experience was their observation of proper team management using PM tools and techniques. The senior PM students could work with first-year students individually, allowing them to tailor instruction depending on the student's ability, which maximized the learning for all students. Similarly, PM students learned how to manage teams of first-year students, which provided a more authentic learning experience than managing a group of seniors with similar knowledge.

Collaboration between instructors of IS, CS, DET, and IT resulted in an interdisciplinary, game-based project approach, effectively addressing many problems within both the IP course and the advanced PM course. Brookes (2017) defines interdisciplinary learning as occurring when students from mixed disciplines teams collaborate in teams, thereby creating greater collaboration between disciplines. Greater collaboration occurs when students incorporate their own discipline's perspectives into the team's common goal. Brookes (2017) notes that interdisciplinary learning grants students more exposure to and knowledge of methods from other disciplines. This learning paradigm is important for all computer-related disciplines where employers use diverse, cross-functional teams to work on projects.

Area	Problem	Process change (Solution)
PM Team Experience	Homogeneous teams: The PM course teams consisted of 4-5 seniors rotating managing each other.	Each PM student was assigned to a first-year IP team, facilitating a more realistic experience and an opportunity to practice PM tools and techniques.
IP Team Experience	Heterogeneous teams: The IP course teams consisted of 4-5 first-year students of different majors with no management experience.	Each team of 4-5 first-year students was assigned a senior PM student, directed by the PM instructor.
Course Structure	Teams coordinated meetings outside of class, with members frequently missing team meetings.	Mandatory Thursday in-class meetings. Instructors took attendance, which allowed instructors to monitor and improve student meetings.
Team Structure	Unequal contributions led to different individual learning experiences, strife among team members, and difficulties for instructors.	Project managers divided project work into smaller tasks, assigning responsibility to individual team members using a Work Breakdown Structure (WBS) and a work responsibility matrix. Project managers verified that individual team members completed their work.
Supervision	Two CS/IT/DET instructors assisted teams and individuals. The high workload on instructors resulted in students waiting for help and becoming frustrated.	A senior PM student experienced with the project managed each IP team. Two CS/IT/DET instructors plus one IS instructor assisted project managers & teams, which prevented instructor overload.

Table 1. List of Problems and Associated Process Changes in IP and PM Courses

Pollard's 2012 study on the use of client projects in an undergraduate PM class identified the biggest problem encountered with respect to time management—the difficulty in finding time for weekly team meetings and time for meetings with clients. This issue also emerged in the experience of the instructors involved in this study and was a common complaint in past student evaluations. To address this problem, the PM course and the IP course were aligned to occur on the same days (Tuesdays and Thursdays), at the same time, and on the same floor of the business building. Given that the PM class was roughly one-quarter the size of the IP class, it was relatively easy to balance the two classes. Each student in the PM course managed a team of 4-5 first-year students, who worked on a semester-long project making a gaming app of their choice.

The PM class in this study was mandatory for IS majors and an elective for other majors. At some institutions, PM is required for all business majors, which naturally results in much larger PM courses, making balancing more difficult. One solution could be to seek out other first-year courses with large group projects, such as Introduction to Business, or similar courses for business majors. Although the courses in the context of this project were both technical in nature, it is not required that they be technical.

The PM tools and methods used to manage the first-year student teams focused on planning, scheduling, and controlling the project. Examples include the Work Breakdown Structure (WBS), Responsibility Matrix, and Critical Path Method (CPM). These tools and techniques can be applied universally to manage any team, provided the project is sufficiently long, complex, and involved. However, the PM tools and methods are not as useful for simple, short activities.

The pedagogical approach in the PM course—primarily textbook and instructor-centered teaching—follows the Project Management Institute's well-established PMBOK® methodologies and processes. This framework focuses on PM hard skills, and while related PM tools/techniques are critical for students to learn and can be taught in an instructor-centered setting, they are best learned through experience. On Tuesdays, the PM course met as a class for instructor-centered teaching that followed the textbook's progression, conveniently moving through the phases of a project from beginning to end. The semester started with planning the project and ended with its closure. Also on Tuesdays, the IP course held instructor-centered lectures on programming, game design, and game-making techniques.

In both courses, every Thursday was dedicated to teamwork. On these teamwork days, the PM class started as usual with an instructor-led debriefing for five to ten minutes. After debriefing, students joined their teams from the IP class for the remainder of the class period. Teams met in both classrooms, and the instructors of both classes moved from team to team, primarily observing but also helping when needed. Having the courses offered synchronously and across the hall from each effectively eliminated scheduling problems, with attendance being mandatory.

Senior-level PM students led the IP student teams by implementing PM tools and methods they had learned in the PM course. Use of these particular tools and methods, such as WBS, CPM, and the responsibility matrix, was specifically required, with the documentation of their proper use contributing to the students' grades. Notably, this was the third year for the IP class to use the gaming app project as a semester assignment. Consequently, senior PM students, who had already taken the lower-level IP course with its game-making project alongside other courses required for their degree, had previous expertise on all aspects of the project. Additionally, they also had first-hand experience of poorly managed and/or dysfunctional teams in the IP course.

Based on student comments and instructors' observations, students in the IP course learned basic programming principles pertinent to their discipline faster and with less frustration than before. Furthermore, they learned how to manage and be managed and also learned effective teamwork and team management by observing the PM techniques employed by the senior PM student assigned to their team.

For the instructors, having a student project manager lead each team dramatically reduced their workload. Instructors felt like their time could now be spent on encouraging and guiding student learning rather than "putting out fires." Throughout the semester, the PM instructor and both IP instructors could spend the entire class time on team days working with individual groups. Before that, such interactions were sporadic and usually happened before or after class, typically involving interpersonal team issues related to poor project management.

At the end of the semester in the IP course, teams presented their final projects in a competitive showcase open to the entire school. Attendees had the opportunity to go from table to table, exploring posters describing each game's function and trial each gaming app. The IP teams were supported by their senior project manager from the PM class, who attended but did not actively present. Figures 1 through 3 show screenshots from Team Reaper's game project presentation. Figures 4 and 5 show screenshots from team On Guard's game project presentation. Students were free to choose their teams, names, and themes. These two examples illustrate the variety of interests and approaches to meeting project goals.

Figure 1. Team Reaper's Screenshot 1

Figure 2. Team Reaper's Screenshot 2

Figure 3. Team Reaper's Screenshot 3

Figure 4. Team on Guard Screenshot 1

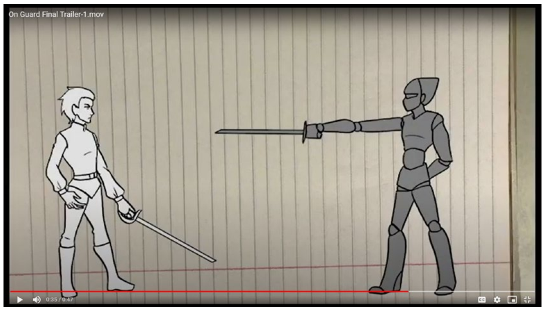

Figure 5. Team On Guard Screenshot 2

At the end of the semester, PM students gave a "Lessons Learned" presentation about their experience managing the game-making projects. Presentations focused on highlighting the difficulties and solutions they encountered in managing their teams. Before this presentation, it was stressed that negative experiences had as much learning potential as positive experiences.

Along with the "Lessons Learned" presentation, students submitted an end-of-semester reflective essay that summarized their experiences in the course. The data used in this study was collected from these reflective essays. Reviewing lessons learned is part of the final closing phase of PM. The discussion of problems encountered was encouraged, with every student receiving a completion grade of 100% for this assignment. This is important to note because assessment and evaluation methods must be addressed to avoid confusing "how students learn and how they are assessed" (Salmon, 2000). The feedback from the reflective papers was far more informative than the data derived from course surveys. This finding aligns with other researchers who have also found that end-of-semester reflective papers yield more and richer information than course evaluation surveys (Deggs & Weaver, 2009; YuekMing & Manaf, 2014).

4.2 Results

Using a grounded theory approach, NVivo 12 software was used to perform thematic synthesis following the steps proposed by Cruzes et al. (2014). The first step involved extracting relevant data from the student papers. Next, both researchers coded all 27 open-ended student responses. Subsequently, the researchers developed a thematic framework of higher-order themes and sub-themes from the codes. This exercise resulted in the identification of four major themes: positive team experiences, positive class experiences, negative team experiences, and negative class experiences. Table 2 presents these four themes along with their related sub-themes. Although there are no statistical tests for validating qualitative research on open-ended responses, various methods can increase the coding's credibility. One method of increasing credibility is having multiple researchers code the results and analyze the data (Côté & Turgeon, 2005; Nowell et al., 2017; Sutton & Austin, 2015). Additionally, software tools can also improve validity by maintaining rigor and reducing researcher bias (Cruzes et al., 2014; Woods et al., 2016). Braun and Clarke (2006) suggest embedding the narrative with extracts of raw data to convince the reader of the validity of the analysis.

Theme	Number of Participants Who Mentioned Each Subtheme	Total Number of Times Mentioned
Positive Team Experiences		
• Mentoring was beneficial	14	16
• Realistic management experience	8	9
• Improved team communication	3	3
Positive Course Experiences		
• Combining the classes was a good idea	15	22
• Both courses benefitted	12	12
• Class structure was unique	11	11
• Combining classes should be a tradition	8	10
• Fun experience	7	8
Negative Team Experiences		
• Negative team experience	6	6
• Unequal effort by team members	5	7
• Poor team communication	5	5
Negative Course Experiences		
• Room for improvement	6	6
• Not enough meeting time	3	4
• Project assignment lacked structure/details	2	2

Table 2. Distribution of Qualitative Codes by Theme and Sub-Theme

Student feedback predominantly indicated positive experiences from the combined classes, primarily due to the opportunities presented in the team management and mentoring. For example, of the 27 student responses, 25 reported positive course experiences, despite the encouragement for students to report both positive and negative experiences. Notably, while eleven students reported only positive experiences, one student reported only negative experiences, and one student reported neither positive nor negative experiences, while 14 reported both positive and negative experiences. The most frequently mentioned sub-theme, reported by more than half the students, was that combining the classes was a good idea. To illustrate, one student comment indicating it was a good idea said, "…opportunity to manage a team in a learning environment I thought was a genius idea." The second most commonly reported sub-theme was that mentoring was beneficial. For example, one comment was, "We were able to offer something I wish I had my freshman year of taking this class." Under the positive team experiences theme, 14 students mentioned that mentoring was beneficial and eight mentioned that the management experience was realistic. Both these aspects are direct results of combining the courses.

In examining the negative experiences theme, sub-themes such as unequal effort among team members and poor communication emerged as common group project problems. An example of a comment indicating a negative experience was, "a lot of the difficulty in our project came from some students just not being ready for the challenges that this class and project can bring." Although these are valid problems, they are not intrinsically linked to the game-making project or the combined course implementation. The sub-theme "Not enough meeting time" is also featured in the negative category, as illustrated in this student's comment: "we need to meet with our group more than just once a week." Although providing one day a week for in-class team meetings increased team meeting time, it was not enough in the opinion of three students. The sub-theme "Project assignment lacked structure/details," although clearly perceived as a negative by a few students, actually reflects the choice built into the game-making project where teams formulated their own innovative designs.

In general, the positive themes received significantly more mentions than the negative themes. This response reflects the overall positive class learning environment the instructors observed. The negative team experience "unequal effort" corresponds with the free-rider effect. The feedback from students indicates it still occurred but was greatly reduced. Evidence suggests that the project was well received by students and was perceived as successful.

5. LIMITATIONS AND CONCLUSIONS

5.1 Limitations

One limitation of this study is its restricted generalizability due to the small sample size. Another limitation is that both courses were conducted as traditional, live courses. Thus, many team issues were avoided in the mandatory weekly team meetings during class time. The authors of this study did not attempt combining both courses in an online environment; however, this should be feasible with the proper execution. Both courses would need to hold synchronous sessions simultaneously, as they did in this study. Students from the IP and PM classes could still have meetings but would need to employ technology that permits virtual meetings, such as Discord, which would allow Instructors to still monitor student meetings and answer questions or guide interactions as appropriate. Therefore, future research should investigate implementing this model in an online learning environment.

5.2 Conclusions

As a conceptual paper, this paper's focus was on the relationships among constructs. The purpose of conceptual papers is to develop logical arguments about these relationships, not empirically test them (Gilson & Goldberg, 2015).

An important aspect of linking theory to practice is reflective observation, defined as a post-implementation reflection relating what happened back to theory (Drinka & Yen, 2008). The benefits of experiential learning tools and techniques have been demonstrated in past studies, as previously discussed. Initially, the IP course integrated game-making and peer learning in an attempt to improve the pass/fail rate, motivate students, and improve experiential learning. The PM course adopted peer learning and real-life projects to motivate students and improve experiential learning. However, before the IP and PM courses were combined, these experiential learning tools yielding mixed results. Subsequently, the focus shifted from the tools (peer learning and real-life projects) to the learning process and associated problems. In an attempt to enhance the process of transforming experience into knowledge, a strategy was developed that integrated the IP and PM courses to improve the overall learning process.

Positive student feedback regarding combining the courses, peer mentoring, and the realistic management experience resulted from changing the process by combining the courses. This was essentially a focus on "the process whereby knowledge is created through the transformation of experience" according to Experiential Learning Theory (ELT; Kolb & Kolb, 2005, p. 194). Furthermore, we observed a reduction in the instructors' workload and stress levels after this focus on process.

Students in the PM course gained real-world experience by applying PM tools and methods to manage a team making a gaming app. Similarly, students in the IP course enjoyed being mentored and guided by a seasoned peer possessing both game-making experience and PM skills. The implementation of a game-making project had to be adjusted and combined with other learning tools before it yielded positive results.

In terms of generalizability, one of the primary findings based on ELT was that focusing on the process of knowledge creation through the team experience positively impacted teaching and learning in many ways. In the case of PM, this pedagogical experience involved both "soft" and "hard" skills. The focus on the process of knowledge creation instead of specific tools is generalizable to teaching other topics but specific implementations would be expected to vary. When generalizing aspects of this study's specific implementation or choosing a course to combine with a PM course, it should be noted the PM students in this study had two advantages that might not be present in other situations. First, all but one PM student had taken the IP course as a first-year student and therefore had personal experience with the IP gaming project and familiarity with the associated common technical problems. Second, because PM was the capstone course for IS majors, all

PM students had additional technical expertise from undertaking upper-level technical and managerial courses. As impending graduates, they also exhibited more interpersonal skills gained from their experiences working on team projects in previous courses. In the PM context, experience is considered an important part of competence, which also enhances other components of competence, such as knowledge and skills (Frame, 1999).

Students in both classes demonstrated a positive perception of working on a mentored team while making a game. PM students repeatedly mentioned the "real-world" aspects of the project as particularly positive. The inclusion of a game-making team project, interdisciplinary teaching/learning, and having PM students manage teams resulted in a fun learning experience. The modifications resulted in two very specialized classes having a much broader and richer learning experience. For instance, PM students not only learned how to manage through experience, but also the first-year students learned experientially how to be managed. Additionally, they could observe as PM tools and techniques were applied to keep everything progressing according to plan.

Consequently, results suggest a focus on process design when transitioning from traditional classroom learning to experiential learning. In a similar vein, existing experiential learning implementations suffering from high overhead in terms of time requirements, scalability, costs, and assessment difficulties should re-examine process design and consider the incorporation of peer learning. For the experiential learning process in general and peer learning in particular, the seniority of the PM students who guided the teams seems to have empowered them to mitigate motivational issues associated with the free rider effect and the "sink or swim" approach to experiential learning.

6. REFERENCES

AACSB. (2020). *2020 Guiding Principles and Standards for Business Accreditation*. AACSB. https://www.aacsb.edu/-/media/documents/accreditation/2020-aacsb-business-accreditation-standards-jul-1-2022.pdf?rev=b40ee40b26a14d4185c504d00bade58f&hash=9B649E9B8413DFD660C6C2AFAAD10429

Alturki, R. A. (2016). Measuring and Improving Student Performance in an Introductory Programming Course. *Informatics in Education*, 15(2), 183-204. https://doi.org/10.15388/infedu.2016.10

Argent, L., Depper, B., Fajardo, R., Gjertson, S., Leutenegger, S. T., Lopez, M. A., & Rutenbeck, J. (2006). Building a Game Development Program. *Computer*, 39(6), 52-60. https://doi.org/10.1109/MC.2006.189

Bakan, U., & Bakan, U. (2018). Game-Based Learning Studies in Education Journals: A Systematic Review of Recent Trends. *Actualidades Pedagógicas*, 72, 119-145. https://doi.org/10.19052/ap.5245

Bakare, J., & Orji, C. T. (2018). Effects of Reciprocal Peer Tutoring and Direct Learning Environment on Sophomores' Academic Achievement in Electronic and Computer Fundamentals. *Education and Information Technologies,* 24(2), 1035-1055. https://doi.org/10.1007/s10639-018-9808-1

Braun, V., & Clarke, V. (2006). Using Thematic Analysis in Psychology. *Qualitative Research in Psychology*, 3(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Bronner, R., & Kollmannsperger, M. (1998). Planspiele als Hochschuldidaktische Lehrmethode [Business Games as University Teaching Method]. *Wirtschaftswissenschaftliches Studium*, 27(4), 218-220.

Brookes, W. (2017). Transdisciplinary Learning in Technology Degrees. *The 16th International Conference on Information Technology Based Higher Education and Training* (pp. 1-6). https://doi.org/10.1109/ITHET.2017.8067823

Carvalho, A. R., & Santos, C. (2022). Developing Peer Mentors' Collaborative and Metacognitive Skills With a Technology-Enhanced Peer Learning Program. *Computers and Education Open*, 3, 100070. https://doi.org/10.1016/j.caeo.2021.100070

Chase, J. D., & Okie, E. G. (2000). Combining Cooperative Learning and Peer Instruction in Introductory Computer Science. *ACM SIGCSE Bulletin*, 32(1), 372-376. https://doi.org/10.1145/331795.331888

Clark, D. B., Tanner-Smith, E. E., & Killingsworth, S. S. (2016). Digital Games, Design, and Learning. *Review of Educational Research*, 86(1), 79-122. https://doi.org/10.3102/0034654315582065

Côté, L., & Turgeon, J. (2005). Appraising Qualitative Research Articles in Medicine and Medical Education. *Medical Teacher*, 27(1), 71-75. https://doi.org/10.1080/01421590400016308

Crabtree, J., Zhang, X., & Ray, D. (2022). Utilizing Peer Tutors in Introductory Programming Education: An Exploratory Investigation. *International Journal of Teaching and Learning in Higher Education*, 33(3), 429-445.

Cruzes, D. S., Dybå, T., Runeson, P., & Höst, M. (2014). Case Studies Synthesis: A Thematic, Cross-Case, and Narrative Synthesis Worked Example. *Empirical Software Engineering*, 20(6), 1634-1665. https://doi.org/10.1007/s10664-014-9326-8

Dalal, N. (2012). Using Rapid Game Prototyping for Exploring Requirements Discovery and Modeling. *Journal of Information Systems Education*, 23(4), 341-344.

Dawar, D. (2021). Towards Improving Student Expectations in Introductory Programming Course with Incrementally Scaffolded Approach. *Information Systems Education Journal*, 19(4), 61-76.

Dawson, J., & Thomson, R. (2018). The Future Cybersecurity Workforce: Going Beyond Technical Skills for Successful Cyber Performance. *Frontiers in Psychology*, 9(744), 1-12. https://doi.org/10.3389/fpsyg.2018.00744

Deggs, D., & Weaver, S. W. (2009). Using Reflection to Evaluate Course Outcomes. *Journal of College Teaching & Learning*, 6(2), 41-48. https://doi.org/10.19030/tlc.v6i2.1171

Draus, P., Mishra, S., Slonka, K., & Bromall, N. (2022). Exposing the IT Skills Gap: Surveying Employers' Requirements in Four Key Domains. *Information Systems Education Journal*, 20(2), 4-14.

Drinka, D., & Yen, M. (2008). Controlling Curriculum Redesign With a Process Improvement Model. *Journal of Information Systems Education*, 19(3), 331-342.

Dyrud, M. A. (2001). Group Projects and Peer Review. *Business Communication Quarterly*, 64(4), 106-112. https://doi.org/10.1177/108056990106400412

El Massah, S. S. (2018). Addressing Free Riders in Collaborative Group Work. *International Journal of Educational Management*, 32(7), 1223-1244. https://doi.org/10.1108/ijem-01-2017-0012

Falchikov, N. (2001). *Learning Together: Peer Tutoring in Higher Education*. Routledge. https://doi.org/10.4324/9780203451496

Frame, J. D. (1999). *Project Management Competence: Building Key Skills for Individuals, Teams, and Organizations*. Jossey-Bass.

Gaeta, E., Beltrán-Jaunsaras, M. E., Cea, G., Spieler, B., Burton, A., García-Betances, R. I., Cabrera-Umpiérrez, M. F., Brown, D., Boulton, H., & Arredondo Waldmeyer, M. T. (2019). Evaluation of the Create@School Game-Based Learning–Teaching Approach. *Sensors*, 19(15), 3251. https://doi.org/10.3390/s19153251

Geithner, S., & Menzel, D. (2016). Effectiveness of Learning Through Experience and Reflection in a Project Management Simulation. *Simulation & Gaming*, 47(2), 228-256. https://doi.org/10.1177/1046878115624312

Gilson, L. L., & Goldberg, C. B. (2015). Editors' Comment: So, What Is a Conceptual Paper? *Group & Organization Management*, 40(2), 127-130. https://doi.org/10.1177/1059601115576425

Guzdial, M. (2009, October 8). How We Teach Introductory Computer Science Is Wrong. *BLOG@CACM*. https://cacm.acm.org/blogs/blog-cacm/45725-how-we-teach-introductory-computer-science-is-wrong/fulltext#

Janz, B. D. (1999). Self-Directed Teams in IS: Correlates for Improved Systems Development Work Outcomes. *Information & Management*, 35(3), 171-192. https://doi.org/10.1016/s0378-7206(98)00088-3

Kafai, Y. B. (2009). *Minds in Play: Computer Game Design as a Context for Children's Learning*. Routledge.

Kafai, Y. B., & Burke, Q. (2015). Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist*, *50*(4), 313-334. https://doi.org/10.1080/00461520.2015.1124022

Karanja, E., & Grant, D. (2020). Evaluating Learner-Centeredness Course Pedagogy in Project Management Syllabi Using a Content Analysis Approach. *Journal of Information Systems Education*, 31(2), 131-146.

Koivisto, J., & Hamari, J. (2019). The Rise of Motivational Information Systems: A Review of Gamification Research. *International Journal of Information Management*, 45, 191-210. https://doi.org/10.1016/j.ijinfomgt.2018.10.013

Kolb, A. Y., & Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning & Education*, 4(2), 193-212. https://doi.org/10.5465/amle.2005.17268566

Kruck, S. E., & Teer, F. P. (2009). Interdisciplinary Student Teams Projects: A Case Study. *Journal of Information Systems Education*, 20(3), 325-330.

Lavelle, J. (2019, January 17). *Gartner Survey Shows Global Talent Shortage Is Now the Top Emerging Risk Facing Organizations*. https://www.gartner.com/en/newsroom/press-releases/2019-01-17-gartner-survey-shows-global-talent-shortage-is-now-the-top-emerging-risk-facing-organizations

Lee, H., & Lim, C. (2012). Peer Evaluation in Blended Team Project-Based Learning: What Do Students Find Important? *Educational Technology & Society*, 15(4), 214-224.

Lee, W. L. (2016). Scrum-X: An Interactive and Experiential Learning Platform for Teaching Scrum. *The 7th International Conference on Education, Training and Informatics* (pp. 1-7).

Lin, J. (2018). Effects of an Online Team Project-Based Learning Environment With Group Awareness and Peer Evaluation on Socially Shared Regulation of Learning and Self-Regulated Learning. *Behaviour & Information Technology*, 37(5), 445-461. https://doi.org/10.1080/0144929x.2018.1451558

Lindsjørn, Y., Sjøberg, D. I. K., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork Quality and Project Success in Software Development: A Survey of Agile Development Teams. *Journal of Systems and Software*, 122, 274-286. https://doi.org/10.1016/j.jss.2016.09.028

Loughry, M. L., Ohland, M. W., & DeWayne Moore, D. (2007). Development of a Theory-Based Assessment of Team Member Effectiveness. *Educational and Psychological Measurement*, 67(3), 505-524. https://doi.org/10.1177/0013164406292085

Magano, J., Silva, C., Figueiredo, C., Vitória, A., Nogueira, T., & Pimenta Dinis, M. A. (2020). Generation Z: Fitting Project Management Soft Skills Competencies—A Mixed-Method Approach. *Education Sciences*, 10(7), 187. https://doi.org/10.3390/educsci10070187

Malik, S. I., & Coldwell-Neilson, J. (2017). Impact of a New Teaching and Learning Approach in an Introductory Programming Course. *Journal of Educational Computing Research*, 55(6), 789-819. https://doi.org/10.1177/0735633116685852

Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing Withdrawal and Failure Rates in Introductory Programming With Subgoal Labeled Worked Examples. *International Journal of STEM Education*, 7(19), 1-16. https://doi.org/10.1186/s40594-020-00222-7

Marshall, M., Dobbs-Oates, J., Kunberger, T., & Greene, J. (2021). The Peer Mentor Experience: Benefits and Challenges in Undergraduate Programs. *Mentoring & Tutoring: Partnership in Learning*, 29(1), 89-109. https://doi.org/10.1080/13611267.2021.1899587

Merrill, M. D. (2002). First Principles of Instruction. *Educational Technology Research and Development*, 50(3), 43-59. https://doi.org/10.1007/bf02505024

Narmaditya, B. S., Annisya', & Yunikawati, N. A. (2022). Providing Assessment Model to Diminish Free Rider and Enhance Students' Cooperative Skills. *Journal of Higher Education Theory and Practice*, 22(7), 84-90. https://doi.org/10.33423/jhetp.v22i7.5272

Nimmo, L., & Usher, G. (2020). 'Job-Ready' Project Managers: Are Australian Universities Preparing Project Managers for the Impact of AI, ML And Bots? *Project Management Research and Practice*, 6. https://doi.org/10.37938/pmrp.vol6.0014

Nousiainen, T., Kangas, M., Rikala, J., & Vesisenaho, M. (2018). Teacher Competencies in Game-Based Pedagogy. *Teaching and Teacher Education*, 74, 85-97. https://doi.org/10.1016/j.tate.2018.04.012

Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic Analysis. *International Journal of Qualitative Methods*, 16(1), 160940691773384. https://doi.org/10.1177/1609406917733847

PMI. (2017). *Project Management Job Growth and Talent Gap 2017–2027*. https://www.pmi.org/learning/careers/job-growth

Pollard, C. (2012). Lessons Learned from Client Projects in an Undergraduate Project Management Course. *Journal of Information Systems Education*, 23(3), 271-282.

Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013a). Success in Introductory Programming: What Works? *Communications of the ACM*, 56(8), 34-36. https://doi.org/10.1145/2492007.2492020

Porter, L., Lee, C. B., & Simon, B. (2013b). Halving Fail Rates Using Peer Instruction: A Study of Four Computer Science Courses. *SIGCSE '13: Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, (pp. 177-182). http://doi.org/10.1145/2445196.2445250

Ramazani, J., & Jergeas, G. (2015). Project Managers and the Journey from Good to Great: The Benefits of Investment in Project Management Training and Education. *International Journal of Project Management*, 33(1), 41-52. https://doi.org/10.1016/j.ijproman.2014.03.012

Richmond, W., Banerjee, D., & White, B. J. (2008). Integrating Curriculum Across Courses in the Same Semester and Across Semesters Using a Service Learning Project. *Decision Sciences Journal of Innovative Education*, 6(2), 509-513. https://doi.org/10.1111/j.1540-4609.2008.00191.x

Roberts, T., & McInnerney, J. (2007). Seven Problems of Online Group Learning (and Their Solutions). *Educational Technology and Society*, 10(4), 257-268.

Sabin, M., Alrumaih, H., Impagliazzo, J., Lunt, B., Zhang, M., Byers, B., Newhouse, W., Paterson, B., Peltsverger, S., Tang, C., van der Veer, G., & Viola, B. (2017). *Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology*. Association for Computing Machinery**.** https://doi.org/10.1145/3173161

Salmon, G. (2000). *E-Moderating: The Key to Teaching and Learning Online*. Kogan Page Limited.

Sanz-Martinez, L., Er, E., Martínez-Monés, A., Dimitriadis, Y., & Bote-Lorenzo, M. L. (2019). Creating Collaborative Groups in a MOOC: A Homogeneous Engagement Grouping Approach. *Behaviour & Information Technology*, 38(11), 1107-1121. https://doi.org/10.1080/0144929x.2019.1571109

Sharp, H., & Robinson, H. (2010). Three 'C's of Agile Practice: Collaboration, Co-ordination and Communication. In T. Dingsøyr, T. Dybå, & N. B. Moe (Eds.), *Agile Software Development* (pp. 61-85). Springer. https://doi.org/10.1007/978-3-642-12575-1_4

Sutton, J., & Austin, Z. (2015). Qualitative Research: Data Collection, Analysis, and Management. *The Canadian Journal of Hospital Pharmacy*, 68(3), 226-231. https://doi.org/10.4212/cjhp.v68i3.1456

Tabatabaei, M. (2014). Online Teaching and Learning Project Management. *Journal of the Southern Association for Information Systems*, 2(1), 42-58. https://doi.org/10.3998/jsais.11880084.0002.104

Taylor-Smith, E., Smith, S., Fabian, K., Berg, T., Meharg, D., & Varey, A. (2019). Bridging the Digital Skills Gap. *ITiCSE '19: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 126-132). https://doi.org/10.1145/3304221.3319744

Teague, D., & Roe, P. (2008). Collaborative Learning: Towards a Solution for Novice Programmers. *Proceedings of the Tenth Australasian Computing Education Conference (ACE2008)* (78, pp. 147-153).

Tynjälä, P., Pirhonen, M., Vartiainen, T., & Helle, L. (2009). Educating IT Project Managers Through Project-Based Learning: A Working-Life Perspective. *Communications of the Association for Information Systems*, 24. https://doi.org/10.17705/1cais.02416

Venkatesh, V., & Maruping, L. (2017). Information Systems Projects and Individual Developer Outcomes: Role of Project Managers and Process Control. *Information Systems Research*, 29(1), 127-148. https://doi.org/10.1287/isre.2017.0723

Vogel, J., Vogel, D., Cannon-Bowers, J., Bowers, C., Muse, K., & Wright, M. (2006). Computer Gaming and Interactive Simulations for Learning: A Meta-Analysis. *Journal of Educational Computing Research*, 34(3), 229-243. https://doi.org/10.2190/FLHV-K4WA-WPVQ-H0YM

Watson, C., & Li, F. (2014). Failure Rates in Introductory Programming Revisited. In *ITICSE '14: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 39-44). Association for Computing Machinery. https://doi.org/10.1145/2591708.2591749

Woods, M., Macklin, R., & Lewis, G. K. (2016). Researcher Reflexivity: Exploring the Impacts of CAQDAS Use. *International Journal of Social Research Methodology: Theory & Practice*, 19(4), 385-403. https://doi.org/10.1080/13645579.2015.1023964

Wouters, P., Van Nimwegen, C., Van Oostendorp, H., & van der Spek, E. (2013). A Meta-Analysis of the Cognitive and Motivational Effects of Serious Games. *Journal of Educational Psychology*, 105(2), 249-265. https://doi.org/10.1037/a0031311

Wurdinger, S. D., & Carlson, J. A. (2010). *Teaching for Experiential Learning: Five Approaches That Work*. Rowman & Littlefield.

Yang, Y., & Chang, C. (2013). Empowering Students Through Digital Game Authorship: Enhancing Concentration, Critical Thinking, and Academic Achievement. *Computers & Education*, 68, 334-344. https://doi.org/10.1016/j.compedu.2013.05.023

YuekMing, H., & Manaf, L. A. (2014). Assessing Learning Outcomes Through Students' Reflective Thinking. *Procedia - Social and Behavioral Sciences*, 152, 973-977. https://doi.org/10.1016/j.sbspro.2014.09.352

Zhang, X., Crabtree, J. D., Terwilliger, M. G., & Jenkins, J. T. (2020). *Journal of Information Systems Education*, 31(2), 106-118.

Zingaro, D., & Porter, L. (2014). Peer Instruction in Computing: The Value of Instructor Intervention. *Computers & Education*, 71, 87-96. https://doi.org/10.1016/j.compedu.2013.09.015

AUTHOR BIOGRAPHIES

Jeremy St. John is an associate professor of management information systems and Department Chair of Accounting, Economics, and Finance at Angelo State University's Norris-Vincent College of Business. He received a Ph.D. in Business Computer Information Systems from the University of North Texas. His research has been published in *Information Systems Management* and the *European Journal of Innovation Management*. Prior to joining academia, he was a consultant/instructor for the Treasury Department's Computer Audit Specialist (CAS) training program.

Karen St. John is an assistant professor of information technology at Abilene Christian University with publications in the *Journal of Education for Business* and the *European Journal of Innovation Management*. She also serves as the Director of Pinecrest Cemetery. Previous work experience includes consultant/instructor for the Treasury Department's Computer Audit Specialist (CAS) training program.

INFORMATION SYSTEMS & COMPUTING ACADEMIC PROFESSIONALS

iscap

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)