Scaffolding a Team-based Active Learning Course to Engage Students: A Multidimensional Approach

Dr. Mohsen M Dorodchi, University of North Carolina at Charlotte

Dr. Dorodchi has been teaching in the field of computing for over 30 years of which 20 years as educator. He has taught majority of the courses in the computer science and engineering curriculum over the past 20 years such as introductory programming, data structures, databases, software engineering, system programming, etc. He has been actively involved in computer science education research and has a number of publications and active NSF grants in this field including learning and predictive analytics for student success, S-Stem NSF grant, Research Practitioner Partnership NSF grant, and Spatial Reasoning Impact Study in CS1.

Nasrin Dehbozorgi, University of North Carolina at Charlotte

Researcher and Ph.D. candidate in the department of Computer Science at University of North Carolina at Charlotte. Conducting research in the area of CSE by applying AI/NLP to do learning analytics, developing models to operationalize attitude in collaborative conversations and pedagogical design patterns.

Aileen Benedict, University of North Carolina at Charlotte

Aileen Benedict is a Ph.D. student and GAANN Fellow at UNC Charlotte, who has been mentored in teaching since 2016. Her work mainly focuses on CS education and learning analytics, with specific interests in reflective practices and predictive analytics. More recently, she has also been learning more about various topics in machine learning, recommender systems, and mental health.

Erfan Al-Hossami, University of North Carolina at Charlotte

Erfan Al-Hossami is a Ph.D. student at UNC Charlotte. Erfan has been mentored in teaching CS1 since 2016 and then in CS education research. His work mainly focuses on predictive learning analytics. His research interests include Machine Learning, NLP, and Conversational A.I. and mental health. Recently, he's been learning more about code generation, transfer learning, and text classification.

Alexandria Benedict, University of North Carolina at Charlotte

Alexandria Benedict is an undergraduate student at the University of North Carolina at Charlotte pursuing her Bachelor of Science in Computer Science. She is a recipient of the STARS Scholarship, and is a research assistant under the RPP STEM Ecosystem Project which helps study the effects of computational thinking inside classrooms. Furthermore, she has been a teaching assistant under Dr. Mohsen Dorodchi for the Introductory Computer Science course at UNCC for the past 2 years.

Scaffolding a Team-based Active Learning Course to Engage Students: A Multidimensional Approach

Mohsen Dorodchi, Nasrin Dehbozorgi, Aileen Benedict,
Erfan Al-Hossami, Alexandria Benedict

Abstract

In this evidence-based practice paper, we present our experiences with different scaffolding techniques to improve student engagement in active learning classes. Scaffolding of course content enables learners to achieve the expected course learning outcomes smoothly from lower to higher challenge levels. Also, in active learning classes with an emphasis on group activities, the activities can be scaffolded in different ways to promote a higher level of engagement and provide more diversity in students' learning process. Since students in large activity-based active learning classes (ABAL) complete the assigned activities at different times and in different places, the collaborative work may not become as effective anymore. This phenomenon of falling behind in collaborative learning and team-based activities are observable through late and missing submissions, in which, both are consequential to student performance. In this paper, we present our Introductory Computer Science (CS1) course model, particularly highlighting the process of group work and collaborative learning. Next, we introduce a novel multidimensional scaffolding methodology focused on the following dimensions: (1) chunking by difficulty, (2) chunking by time, (3) chunking by focus, and (4) chunking by collaboration. This approach focuses on refining instructor-to-student mediums through diversifying activities, balancing the challenge levels, including pre-class and post-class assignments, and chunking instruction time. Our approach rethinks scaffolding by incorporating the teaching strategy of think-pair-share as a scaffolding technique to guide learners through student-to-student learning mediums as well. To assess the effectiveness of our approach, we report on various student engagement metrics, including on-time, late, and missing submissions. Our multi-semester findings indicate a significant increase in student on-time submissions and a substantial decrease in overall missing submissions.

1 Introduction

The use of active learning methodologies has been growing considerably in recent times [1]. Active learning is a method of teaching and learning that strives to involve students more directly in the learning process, as "students participate when they are doing something besides passively listening" [2]. Active learning enables students to engage in different tasks such as reading, writing, and discussing while performing "higher-order" thinking (e.g., evaluating, analyzing, and synthesizing) [2], socializing, and reflecting on their problem-solving experiences in small groups [3]. Prince overviewed some various active learning approaches, including problem-based learning (PBL) and collaborative and cooperative learning, along with a study of the effectiveness of such techniques [4]. In this work, the core of each of the above models is defined. Based on the evidence reviewed, collaborative and cooperative learning are promoted with the emphasis on individual accountability. In other words, it has been recommended to practice a mixture of individual exercises and not only team-based ones in order to encourage students to be responsible for their own learning as well. One major challenge in active learning courses is to choose and design these activities in a way to enhance student understanding. An essential component of this design process is to include scaffolding, as noted by Ge et al [5].

Scaffolding is the process of providing students with a smooth learning pathway towards intended course goals through manageable steps [6]. Scaffolding is done by guiding learners through their zone of proximal development defined by Lev Vygotsky, a Russian teacher and psychologist, as "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem-solving under guidance, or in collaboration with more capable peers" [7].

This guidance can be done in various ways, hence a multidimensional aspect of scaffolding. By multidimensional, we mean mix and matching diverse methods of activities and various configurations of similar forms of activity. We call each one of these a dimension, which will be explained in detail in later sections as our contribution. Students' late work is, in a way, a symptom of negative experiences. Therefore, we explore measures regarding late work to try to show whether or not these scaffolding techniques are improving experiences.

In the next section, we review related work on (1) scaffolding in education, and (2) the effects of late work on students' learning experiences. In section 3, we then describe our course and define some of the key types of activities. In section 4, we review the various dimensions of scaffolding and how we implement each one in our course. In section 5, we assess the effectiveness of our scaffolding methods over the course of three different semesters. Finally, in section 6, we conclude with our overall experiences in scaffolding the CS1 active learning class, provide some final recommendations, and also discuss some limitations.

2 Related Work

2.1 Scaffolding in Education

Scaffolding was first introduced by Wood [6] as the process of allowing a learner to achieve a goal through guided efforts. According to Pol, while the concept of scaffolding has received considerable attention in education research, there is no consensus about its definition [8]. However, the majority of researchers have agreed that scaffolding is about providing "dynamic intervention finely tuned to the learner's ongoing progress" and "construed as support" for students "when performing a task that the student might otherwise not be able to accomplish" [8].

Moreover, the amount and type of support need to be regulated as an individual learner assumes more responsibility for learning. One of the critical characteristics of scaffolding is the process of slowly shifting the learning responsibility to the students over time [8]. Referring to the difference between what a student can and cannot do without support, the common objective of scaffolding is to help students eventually become independent of assistance. This assistance is gradually "faded" as responsibility for learning is transferred to the student at some point during the learning process [8].

2.1.1 Scaffolding Techniques

One of the pedagogic scaffolding systems in CS education, especially programming courses, is achieved through well-timed prompts and feedback during students' coding processes from a peer, tutor, or an automated system. In one study, researchers suggest a system called "Test My Code" (TMC) to collect data from students' coding processes and to provide built-in scaffolding messages during their work. These messages were designed to help students achieve the programming goal, which was not easily attainable by themselves [9].

As an alternative for automated scaffolding tools, in another study, the researchers scaffold the code tracing skills in novice programmers by using a sketching approach. In this study, they have the student simulate the computer's memory table by tracing code blocks line by line while thinking aloud about the outcome and sketching them. The authors believe this method helps novices to apply their learned syntactic and semantic knowledge through an external representation of the memory. They theorize that this method can be considered as a scaffolding approach because it reduces the load from students' working memory by making the program state visible. They ran the experiment on 21 students in the CS1 college level. Their study shows that the students who did systematic code tracing by thinking aloud the sketching had a 15% higher performance [10].

In a series of work by Fitzgerald, et al., the authors reported on various successful scaffolding techniques to help diverse groups of students with a wide range of abilities and needs to learn. In particular, they developed a framework known as scaffolded reading experiences (SRE) to support English language learners with the complexities of reading in a new language in several essential ways. The proposed framework slices up the learning goals into manageable tasks through a set of pre, during, and post-reading activities (using diverse forms of reading materials) to break down a complex reading task into smaller chunks. This framework also eases the cognitive demands on the English language learner by structuring them in distinct stages [11].

Designing tasks into manageable chunks was also discussed by Wood, who first introduced the term "scaffolding" [6]. It is the instructor's goal to "control" the tasks that are beyond the learner's capacity so that the learner can concentrate on only the parts that are within their range of competence. It may be worth mentioning that our course model [12] and activity/exam design approach [13] promoted different models of scaffolding inline with the aforementioned reviewed literature on scaffolding in education.

2.1.2 Scaffolding in the Social Learning Context

Scaffolding can also be highly relevant to collaborative learning, as it can be an instructor or peer who guides the learner through the "Zone of Proximal Development" (ZPD) [14], or proper challenge levels for the individual's most effective learning. On top of this, Stone observed a fascinating attribute of learning inspired by ZPD theory stating that learning first takes place on a social level before an individual level [15]. This idea describes a paradigm shift in learning theories over the past several decades. Less often are learners perceived as individuals who go solitarily through the cognitive process [16]. Instead, there is a new emphasis on the social experience that the individual learners acquire and apply.

The recent socio-cultural theory defines learning as a socially engaging experiment in a community of practice [17]. This theory highlights the importance of collaboration in active learning where we observe the learners as a group of students who interact, question, and assist each other toward a common learning goal rather than a collection of single individuals [16]. The socio-cultural theory of learning explains the concept of the Zone of Proximal Development (ZPD), which differentiates the level of independent performance of the individual and the performance level achieved after the instructional guidance (scaffolding) of experts or peers [16]. Dixon-Krauss also states that "from a Vygotskian perspective, the teacher's role is mediating the child's learning activity as they share knowledge through social interaction" [18]. These theories also support the idea of collaborative learning, further suggesting that by having groups with diverse skillsets, members can help guide each other through their own ZPDs.

2.2 Effects of Late Work on Students' Experiences

Despite multiple interventions, many educators are still facing the attrition and late submission issues present in introductory programming courses [19]. Numerous studies have been conducted relating to students' late submissions in programming courses [19, 20].

Fenwick et al. findings show that students who start working later on the assignments have lower scores compared to their peers who start earlier [21]. A possible reason for this could be that the students who start work earlier on the assignments have more time to seek help from other resources [19]. In another study [22], researchers studied the submission behavior of 1,900 students. The result of their research shows that student's performance at the end of the course is linked to their submission patterns earlier in the semester. Moreover, the findings show that students who submit their first assignment late are more likely to develop a late submission pattern in the following years [20]. This highlights the importance of helping students develop time management skills in introductory CS courses to prevent low performance in future years. Edwards et al. further emphasize the importance of assisting students in developing these skills and investigating various interventions aimed at helping students end procrastination [23].

In another study, the researchers claim that, since it is a challenge to get access to logs of all data related to students, especially in large classes, there is a need to have a metric that helps in identifying poor-performing students in earlier stages of the course [22]. They identified a measure of at-risk students based on their timeliness in earlier submissions in the course [22]. Their study on submission patterns of over 220,000 records determines that students' submission patterns in earlier stages are a reliable measure to identify at-risk students. In introductory-level CS courses, it is highly essential to identify and help at-risk students earlier in the semester before it leads to drop out and failure issues. The problem of late submission behavior calls for methods to help students overcome this challenge and to apply interventions that improve students' time management skills and their performance as a result of that.

The other side of late submission problems could be task complexity. There should be a delicate balance between the activity challenge level and the required time to submit the assignment. This

Table 1: Background Information

		Fall 2016	Fall 2017	Fall 2018
Activity Breakdown	Pre-class	0	14	15
	In-class	20	36	48
Total Activities		20	50	63
Total Students		99	113	105

balance is especially important early on in courses when students are struggling to learn new concepts. Scaffolding methods are a way to alleviate these problems. Educators have applied diverse methods of scaffolding in CS to help students have a smooth progression through the course. In the following, we describe what scaffolding is and how it is defined in a social context, and present several scaffolding tools that researchers applied in introductory CS courses.

3 Introducing Our Course

The experiences described in this section are from our introductory programming course (CS1), which follows the activity-based active learning model since the Fall 2016 semester, and has been evolving since then [12, 24]. In the analysis section, we specifically use the semesters of Fall 2016, Fall 2017, and Fall 2018 as the majority of the class population were traditional first-semester freshmen. As noted in Table 1, we had a total of 99 students in Fall 2016, 113 students in Fall 2017, and 105 students in Fall 2018. We also had 20 activities in Fall 2016 before scaffolding, 50 activities in Fall 2017, and 63 Activities in Fall 2018.

Our CS1 collaborative active learning course model follows Kolb's iterative learning style [25] model. In the iterations, the students complete preparation work before attending the class, work on activities in groups during the class, and then answer more advanced problems as individual homework after class. In this class, all activities have due dates. If students submit the activity within a few days after the due date, they would get a late penalty but can submit their work as a late submission. After that time frame, students will not be able to submit their work and their work will be considered as a missed submission.

Learning is a complex phenomenon, and there are many different psychological theories to explain the mechanisms of learning. Lev Vygotsky proposed one of the very first learning theories through socialization [14]. He argued that learning occurs through interactions and communications with others, and further examined the impact of social environments in the learning process. Consequently, he proposed that a learning environment needs to promote and maximize collaboration, peer instruction, and social learning through discussion, collaboration, and feedback. Furthermore, Bandura proposed a social learning theory arguing that people can learn new information and behaviors through socializing [26]. This theory guides educators to recognize how important it is to practice proper models of study skills and teamwork in the classroom to construct self-efficacy of the learners.

Social learning can be implemented in different ways, however, the main notion of that is about learning through interactions of the learners. The common model of instructor-learner social learning is complemented by the learner(s)-learner(s) interactive learning which is promoted in team-based activity-based active learning.

There are many methods and techniques in social learning which could be considered scaffolding since they provide a step-wise approach toward engaging every member into constructive social learning. One of these methods is known as the think-pair-share method [27, 28]. This method scaffolds social learning into steps. First, it attempts to bring all group members into the same level of awareness regarding a problem. This step is known as the think step, in which everyone would ponder about the issue. During the pair step, the learners would discuss the problem/solution in pairs, which is the smallest unit in social learning and is easy to form and manage. For the next level, learners would discuss as a larger group to exchange ideas and conceptualize the activity goal through socializing in a larger group. We observed that without this scaffolding process, some students never engage themselves in larger group discussions.

In this section, we discuss the different types of activities in our course including: (1) Active Readings, (2) Video Quizzes, (3) In-Class Group Activities, (4) Personal Response System (PRS) Quizzes, and (5) Scratch-off quizzes. Numbers of activities per semester are shown in table 1. Active Readings and Video Quizzes are both types of pre-class work while the rest are in-class work.

3.1 Active Readings

Prior to class, students are assigned Active Readings which consist of a series of questions associated with a given reading. In our class, we used Canvas Quizzes to create these assignments and specified the textbook's page numbers to be read along with it.

3.2 Video Quizzes

Video quizzes are another type of pre-class work in which students are tasked with watching several short videos (with a length of about eight minutes max) throughout the course. Every one to two minutes, there is a stopping point to make sure students take a moment to think about the critical concepts of what they just saw. At each of these stopping points, the students are given a question related to a crucial topic or a concept the video had just demonstrated. If the answer is incorrect, they need to repeat the video and answer the question again until they get the correct response. Almost all essential topics in the course are touched on using these video quizzes, especially areas that have been historically challenging for students (such as binary numbers and conversions, methods, and arrays).

3.3 In-Class Group Activities

During the class, students are assigned a variety of in-class activities. In these activities, students work in small assigned groups on a variety of tasks including: Flowchart, code completion, code snippet debugging, answering MCQs (Multiple Choice Questions), writing pseudocode, and/or programming from scratch using Java.

3.4 Personal Response System (PRS)

Personal Response Systems, PRS for short, are increasingly being adopted within higher education over the last six years [29]. One major benefit for PRS is that students have an opportunity for continuous and formative feedback rather than periodic summative feedback in the form of tests and assignments [29]. There are various implementations of PRS such as Kahoot! which can be gamified to engage students through pre-made or impromptu quizzes, discussions, and surveys [30]. In our course we use Kahoot! for formative assessment in class and as a way to pace students in the class. These quizzes are composed of multiple-choice questions and students are tasked with answering each of the questions within a selected timeframe (20,30,60, or 90 seconds). The instructor often encourages students to discuss the questions prior to answering them. Students submit their answers using their personal devices (mobile phone or laptops).

3.5 Scratch-Off Quizzes

Scratch-Off quizzes are multiple-choice quizzes where students submit their answers on scratch-off cards. Students complete the scratch-off quiz individually at first, and then discuss their solutions with their groups to select a final answer. Lastly, students scratch off their chosen answer. If it is correct, a star will appear on the card, providing students with immediate feedback. Students get full points if they get the star from the first attempt. Students are also allowed to attempt multiple times for 60%/40%/10% of the possible points accordingly.

4 Multidimensional Scaffolding in Action

In our systematic practice of active learning since 2016, we have experienced many challenges with designing the course model and activities to promote the best possible learning experiences for students. During this time, we were able to see how students were impacted differently based on each of the various active learning techniques we experimented with. This process of continuous development has led us to the course model for introductory programming which was later expanded to undergraduate software engineering courses [12, 31].

Sustained student engagement was one major challenge in the active learning course design. As we explained in our course model [12], we found that we could only achieve consistent engagement with an aligned set of scaffolded activities. The final course model [12] provided students the opportunity to engage with the course content in small chunks before, during, and after the class.

Throughout our iterative design process we have made several observations regarding patterns in student performance, the first being that of consistent late and missed submissions for various students. Students who had a higher number of late and missed submissions, on average, demonstrated lower performance in the course. Furthermore, we have also observed that students were often concerned with high complexity, inconsistent challenge level of activities, and not having enough time to complete the activities during the class. Some possible reasons for these common problems are a lack of time management skills (such as procrastination), study skills, or soft skills.

Figure 1: The various dimensions of scaffolding.

We argue that one goal of active learning, especially in introductory courses, should be helping our students develop these proper skills. We hoped that our scaffolding methods would help to accomplish this by ultimately enhancing students' soft skills and decreasing the occurrences of late and missed submissions. Therefore, we decided to use this indicator as our primary metric to measure the improvement of our multidimensional scaffolding techniques.

We have broken down our scaffolding techniques into various dimensions as shown in Figure 1: (1) balancing the challenge levels of activities ("chunking by difficult"), (2) breaking activities down into manageable chunks ("chunking by time"), (3) providing a diverse array of activities ("chunking by focus"), and (4) providing a more effective social learning environment ("chunking by collaboration"). In this section, we will discuss these various dimensions of scaffolding in our course model, along with examples of how we applied them.

4.1 Balancing the Challenge Level

As discussed, scaffolding is the process of assisting a learner to achieve a goal through guided efforts. Through scaffolding, a learner is guided through their zone of proximal development, which is a level that is challenging enough to push the learner forward, but not so difficult that they cannot progress without the help of another (such as an instructor, tutor, or peer). In other words, a very large part of scaffolding is to understand where the learner's proper challenge levels are and to design course activities with that in mind. This idea is illustrated in figure 2. As shown the learner being guided through their zone of proximal development. It is through these continuous steps where growth occurs.

To help progress students through proper challenge levels at each step of the way, we developed an algorithm to assess challenge levels of the tests [13] and expanded that to apply to all the activities of our course model. The challenge determination is done based on cognitive domain of

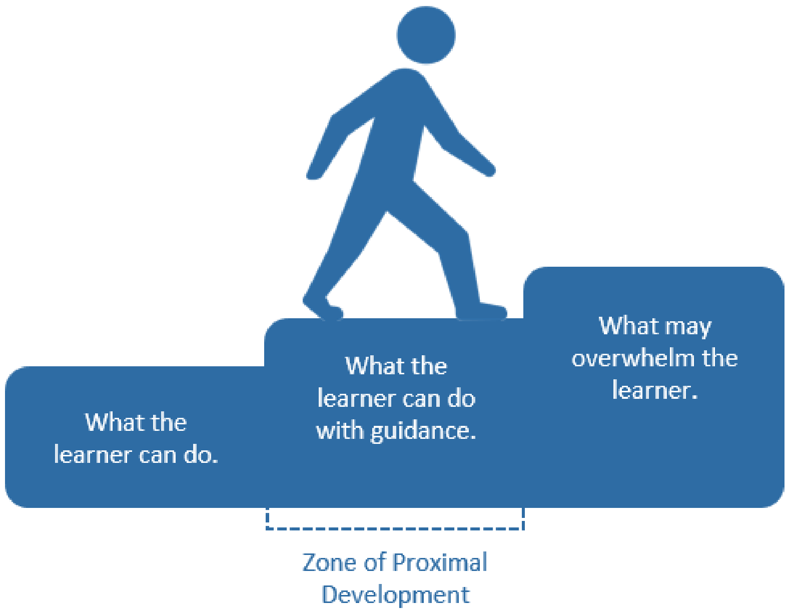


Figure 2: Learner's growth via guided steps through own zone of proximal development

Bloom's Taxonomy (BT) and modified Bloom's Taxonomy for computer science [13]. Based on our course model, the students (1) complete preparation work before attending the class (called pre-labs or pre-work in general), (2) work on both group and individual activities during class (called in-class work), and then (3) complete more advanced problems as individual homework after class (called post-labs or post-work in general) [12].

Scaffolds require the teachers "controlling those elements of the task that are initially beyond the learner's capability, thus permitting him to concentrate upon and complete only those elements that are within his range of competence" [6]. Therefore, we make sure that each step, the pre-, in-class, and post-work, builds off of each other in terms of material and difficulty. The pre-work is necessary to provide students the foundational knowledge of the topic so that groups can work efficiently for the in-class work and ask the instructor, teaching assistants, or other peers questions about any confusion from the pre-work. Some examples of the pre-work given include:

- *Active Readings*, where we ask students to read specific sections of the textbook and answer questions (mostly multiple choice or fill in the blank);
- *Video Quizzes*, where students watch a video describing a given concept with questions included throughout the video; and
- *Pre Labs*, where students are given very basic programming exercises in preparation for their lab.

Most of these exercises focus on the *remember* and *understand* levels of Bloom's Taxonomy. Pre Labs may also reach the apply level, but this usually occurs after at least one iteration of lecture (including the other types of pre work).

Not only do we implement this balancing of levels throughout the overall course model by having these iterations from easier to harder difficulties (pre work being the easiest, and post work being the most challenging), but we must also pay attention to the challenge levels within each step. For example, within the in-class work, the activities themselves must be scaffolded as well. We may have the following activities in one class session:

1. *Kahoot!*, where we ask students questions relating to their pre-work and falling within the remember and understand levels of BT, allowing them to also consult with their group members; then
2. A *group activity*, where teams of four students work together on answering questions (e.g., filling in snippets of code, explaining an algorithm from the reading, or developing a flowchart for a given problem) that may fall more within the understand, apply, and analyze levels of BT; and finally
3. A *scratch-off quiz*, where students must first work on their own on a wider range of questions relating to the material covered in the pre-work and in-class work leading up to this point, and then review again in their groups.

Once again, attention must be paid to the challenge levels of the activities and the individual problems within each activity.

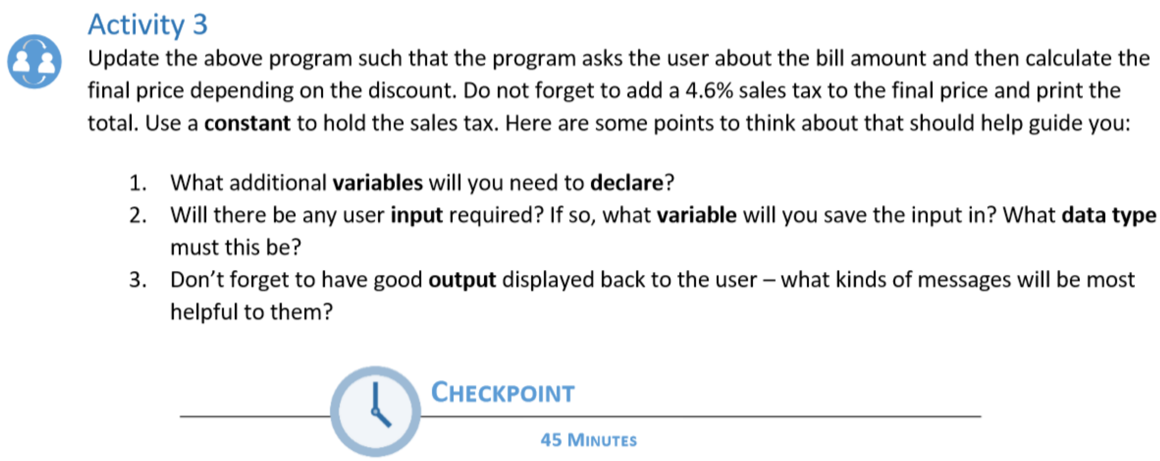
It is also noted that with scaffolding, the amount of support given for a particular subject is slowly reduced as it becomes unnecessary [6]. This is seen in the above example where by the end of class, students are practicing what they have learned individually. This idea is also applied in the post-work, as it is another individual assignment to both provide the students with more practice, challenge them to solve similar problems without the aid of peers, and to assess their understanding before moving on to the next module.

4.2 Breaking Activities Down into Manageable Chunks

Another one of the guidelines for scaffolding instruction discussed by Silver was to "break a task into small, more manageable tasks with opportunities for intermittent feedback" [32]. As mentioned previously, we had observed many students feeling overwhelmed with the amount of course material and being unable to finish in-class activities on time. We believed that by breaking tasks down into smaller chunks, we could decrease that overwhelming feeling and also help students build their time management skills. By having smaller, more specific goals, it is easier to identify and focus on what needs to be done next. Studies have also shown a direct correlation between goal specificity and level of performance, further showing the effectiveness of this approach [33]. This can be included in numerous ways, as discussed in the following subsections.

4.2.1 Pre-, In-Class, and Post-Work

The pre-, in-class, and post-work described in the above section is one such method to help break tasks down into smaller pieces. Not only does this help prepare students for the next section, but it provides instructors and TA's to provide feedback before moving on to the next activity or topic.



Activity 3

Update the above program such that the program asks the user about the bill amount and then calculate the final price depending on the discount. Do not forget to add a 4.6% sales tax to the final price and print the total. Use a **constant** to hold the sales tax. Here are some points to think about that should help guide you:

1. What additional **variables** will you need to **declare**?
2. Will there be any user **input** required? If so, what **variable** will you save the input in? What **data type** must this be?
3. Don't forget to have good **output** displayed back to the user – what kinds of messages will be most helpful to them?

Figure 3: This is an example of a checkpoint provided within one of the lab documents.

4.2.2 Checkpoints within Activities

In the next step, we broke down the long class activities into smaller parts by adding checkpoints. For example, in lab documents, we added a specific checkpoint image with the estimated time it would take to get to this next point, as shown in figure 3.

We hoped that by adding checkpoints: (1) students could use these to pace themselves throughout the entire class, and (2) the instructors would have stopping points to check the students' progress and pause the class, if needed, for additional instruction (e.g., answering questions or giving demos of a solution). By presenting students with these checkpoints, we thought we could help raise their awareness of the time and their progress throughout the activity, once again, to help them pace themselves. Another goal of these checkpoints, especially in our active learning classroom, was to help keep the whole class on the same track. Having the time limit associated per section allows us to enforce the timely completion of different types of activities so that it is possible to switch effectively. Furthermore, by giving the instructor these openings to check on the whole class, they are then able to adapt to the students' needs and hopefully keep everyone on the same page. When designing these checkpoints, it is extremely important to take the students' current knowledge into account to create proper and realistic time limits. This is something we have evolved over time based on testing and timing the activities ourselves, observing the students in the class, and listening to their feedback.

4.2.3 Kahoot!

Kahoot! is a gamified Personal Response System [30] which we believed could serve a similar purpose to the checkpoints in between activities. We incorporated these throughout the class to help provide stopping points to assess students' understandings of the completed activities, to provide low-stake practice problems, and to measure the effectiveness of our activity and checkpoint break-ups.

There was a significant performance gap between the lab tests, which were practical programming exams, and the lecture tests, which were more theoretical and based on core concepts. The averages were 77.46% for lab tests and 64.81% for lecture tests in the Spring 2017 semester. In hopes of reducing this performance gap, we designed a series of Kahoot! quizzes to provide students with constant and consistent feedback and practice in between activities. With a more consistent assessment of the learners' knowledge and experiences, we are better able to determine and meet the students' learning needs.

4.3 Diversifying Activities

Through student surveys and class observations, the instruction team noted that students lost focus over the semester due to the repetitive delivery method of course content. Furthermore, a plateau in student learning was seen after a few weeks into the semester, given a lack of diverse activities. In this case, scaffolding, through diversifying activities, helps to smoothen the flow of learning by chunking attention spans on new and refreshing tasks. We refer to this scaffolding process as chunking by focus.

Throughout the course, we expose students to a mixture of activities: (1) active reading assignments, (2) in-class activities, (3) Kahoot! quizzes, and (4) scratch-off quizzes. Active readings serve as pre-work, as mentioned in section 3, to prepare students for completing in-class activities by enhancing their foundational knowledge, especially the remembering and understanding levels of Bloom's Taxonomy (BT) [34, 35] for relevant concepts. Next, we have a range of in-class activities where students follow guided instructions and discuss with peers in their small groups. During each of these in-class activities, students perform at least one of the following tasks: code completion, code snippet debugging, answering multiple-choice questions, writing pseudocode, and drawing flowcharts. Each of these tasks falls under different BT's categories, demands the use of different skills, and requires students to pay attention to new and different concepts. By diversifying the in-class tasks, students are less likely to feel the work is repetitive, enhancing their attention overall. Third, we include Kahoot! quizzes, as described in section 3, where students discuss with their groups and answer questions individually. Kahoot! quizzes serve as post-work (as mentioned in section 4.1) and for pacing the class (as mentioned in section 4.2). Using Kahoot! quizzes in between in-class activity checkpoints helped students switch their attention to a new task and also get quizzed on a fresh concept they just learned in the activity. Lastly, we use scratch-off quizzes, where students complete a quiz individually at first and then discuss final answers with their groups. After group discussions, students then submit their answers on scratch-off cards. These quizzes were offered only one time before each test. The scratch-off quizzes served as another method for test preparation and reduced Kahoot! redundancy, enhancing overall student attention.

In this section, we explored the scaffolding technique of diversifying activities aiming to create manageable student attention spans by providing a variety of learning experiences. We expose students to over four different types of activities, each with various tasks, cognitive demands, and learning outcomes. After using this diversification process in the class, the instruction team observed that students were more prepared for the course, refreshed, focused, and lastly, students were able to work, guide, and learn from each other, enhancing the environment for collaboration.

4.4 Scaffolding Social Learning

Students in our classroom come from various cultures and backgrounds, have different skill sets, and communicate differently. These factors play a part in creating differing levels of soft skills, such as communication and teamwork, making it a challenge to construct an environment that promotes collaboration. In our course, we do not want to simply assign students into groups and expect them to learn and socialize in a productive manner. Instead, we believe in the importance of creating an appropriate environment for disciplined collaboration through guidance and chunking.

Scaffolding seems to focus on content delivery from instructors to students. We wanted to rethink scaffolding and implement a way to guide learner-to-learner interactions as well. To do so, we assign students into groups to promote social construction of knowledge while also focusing on individual assessments of learning rather than assigning a group grade. In this case, we aim to use scaffolding to smoothen the collaboration flow, chunk discussions and group sizes, and ensure that students work together efficiently by teaching and learning from each other using the think-pair-share model. We refer to this scaffolding process as chunking by collaboration.

To provide a more effective social learning environment, we arrange students into teams of 4 to 5 students. Students are guided through the in-class activities to perform think-pair-share. While performing think-pair-share, students would individually read and complete a portion of the activity, then pair up with another person in their group (if the group size is odd with another two), then, the pairs are instructed to discuss with the bigger group. This way, we believe social interactions are chunked into individually thinking about the content, sharing with one person to practice communication, and then discussing in a larger group. Furthermore, students have more resources to get help from their peers after undergoing the think-pair-share process [27, 28]. After rethinking and implementing think-pair-share as a scaffolding methodology, the instruction team observed decreased student help requests during the class period because students seemed to be more independent and capable of resolving questions and issues within their groups. This observation was compared to when the class arranged students into larger groups (groups of 8-10) and did not use think-pair-share.

5 Analysis

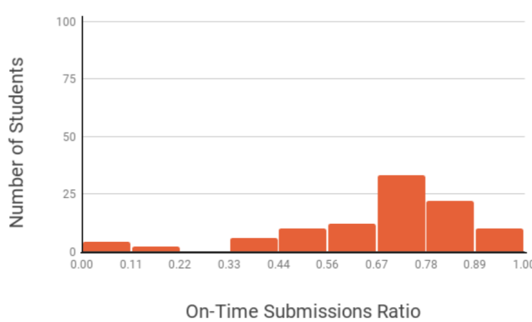
In this section, we analyze various student engagement measurements before and after implementing the scaffolding interventions mentioned in section 4. The data was collected from the Canvas Learning Management System (LMS) for a CS1 course offered by the same instructor. Next, the data is de-identified. This data was collected over Fall 2016 (before scaffolding), Fall 2017 (during scaffolding), and Fall 2018 (after fully implementing the scaffolding techniques described). We compare engagement metric aggregates of time spent on the course page, page views, on-time, late, and missed submission ratios (out of total possible submissions) using the mean, median, and standard deviation across the semesters in table 2.

Furthermore, we illustrated the histograms of the three different submissions. The histograms shown in figure 4 illustrate the ratio of on-time submissions before and after scaffolding. Fall 2016 denotes the term before scaffolding, and Fall 2018 denotes the term after scaffolding. In

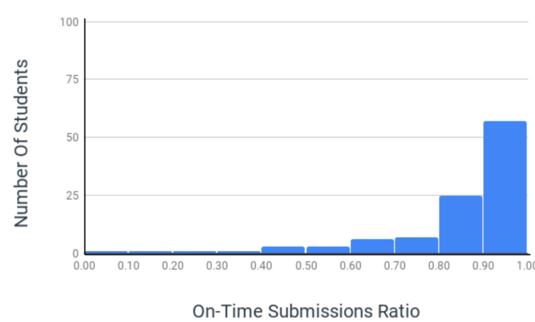
Table 2: Descriptive Statistics of Student Engagement Metrics

		Fall 2016	**Fall 2017**	**Fall 2018**
On-time submissions	Mean	0.67	0.86	0.84
	Median	0.71	0.9	0.9
	STDEV	0.20	0.14	0.18
Late submissions	Mean	0.09	0.09	0.08
	Median	0.10	0.06	0.05
	STDEV	0.07	0.08	0.06
Missed submissions	Mean	0.24	0.06	0.08
	Median	0.19	0.02	0.03
	STDEV	0.22	0.11	0.14
Time Spent On Course Page	Mean	28:08:48	51:06:38	48:05:51
	Median	24:15:32	44:28:18	43:44:21
	STDEV	23:26:34	27:44:37	22:51:28
LMS Course Page Views	Mean	470.21	728.12	887.7
	Median	457	660	796
	STDEV	200.53	351.87	494.76

these histograms, the vertical axis shows the number of students, and the horizontal axis shows the ratio of on-time submissions in figure 4 and missed submissions in figure 5 based on the total number of submissions in each class. We notice a big shift towards the right, indicating increased on-time submissions after scaffolding (Fall 2018) compared to before scaffolding (Fall 2016). The histograms of missed submission rates of Fall 2016 and Fall 2018 are presented below in figures 5a and 5b respectively.



(a) Before Scaffolding

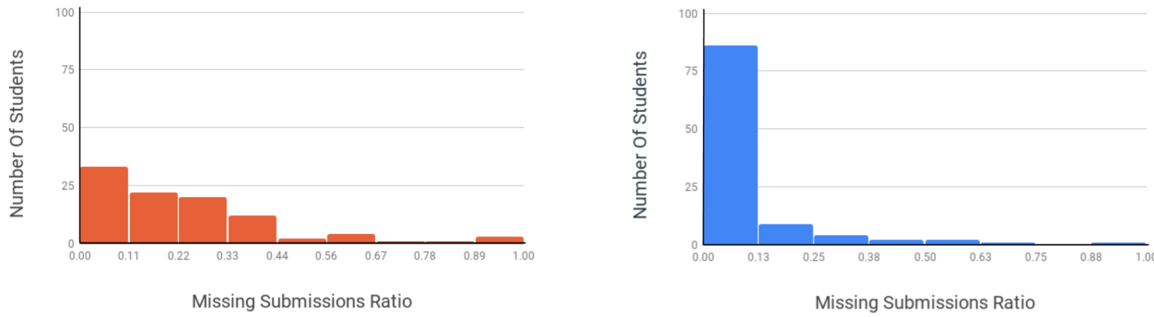


(b) After Scaffolding

Figure 4: Histogram of on-time submission ratios before and after scaffolding

When comparing the histograms in figure 5, the histogram shifts to the left, indicating a sharp decrease in missing submissions after scaffolding. We also observe that over 70% of students had missed less than 10% of all submissions after scaffolding (Fall 2018).

For in-depth analysis, we performed a test of the hypothesis (*t*-test) as well as Cohen's effect size.



(a) Before Scaffolding

(b) After Scaffolding

Figure 5: Histogram of missing submission ratios before and after scaffolding

As the first step, we conducted Levene's Test on our dataset, and the requirements for homogeneity were met. Later, we conducted a two-sample *t*-test to determine the change significance before scaffolding (Fall 2016) and after scaffolding (Fall 2018), as demonstrated in table 3. The null hypothesis is that there is no significant difference between the populations of Fall 2016 and Fall 2018 in terms of the engagement metrics: time spent on the course page, student course page views, on-time, late, and missed submissions. Next, we perform a one-tailed test. We compute the *t*-test and *p*-value, and significance for each of these metrics. We report statistically significant results amongst all engagement metrics and reject the null hypothesis.

After applying multidimensional scaffolding in Fall 2018, students missed fewer submissions overall even though the number of activities almost tripled from 20 to 63. It is also important to note that this increase in activities is primarily due to the scaffolding techniques applied, such as breaking activities down into smaller chunks. We also report significant increases in page views and time spent on the course page. This increase can likely be attributed to the rise in activities and required submissions for the class. For example, by breaking activities down into smaller chunks, students may then need to access the course page more frequently. This frequency can be seen as a benefit since students will have more repetition with the material. Lastly, on-time submissions increased significantly, perhaps due to enforcing in-class checkpoints, as mentioned in section 4.3. To measure performance significance, we measured normalized exam grades and have found no significance and did not reject the null hypothesis.

In the analysis of our experiments, the hypothesis test of significance (and our case, the *t*-test) may not always provide a generalizable (from sample to population) and interpretable, quantitative description of the size of an effect. For this reason, the effect size is used in parallel to the *t*-test. Studies have shown that effect size provides a more robust measure compared to statistical significance in the analysis of experiments since it describes the size of the observed effect independent of the possibly misleading influences of sample size [36]. To further showcase the effect of our scaffolding intervention on the engagement metrics of on-time, late and missed submissions, we calculated Cohen's *d* effect size based on the following equation [37]:

$$d = \frac{\bar{x_1} - \bar{x_2}}{s}, \tag{1}$$

Table 3: Significance of Engagement

Metric	Mean Dif	t	df	Sig. (1-tailed)
On-time Submissions	-0.175	-6.463	202	$<$**.00001**
Missed Submissions	0.159	6.123	202	$<$**.00001**
Late Submissions	0.018	1.895	202	**0.03**
Time Spent on Course Page (In Seconds)	-71823.3	-6.154	202	$<$**.00001**
LMS Course Page Views	-417.483	-7.812	202	$<$**.00001**

where *d* is the normalized distance between the means of the two populations [38]; $\bar{x_1}$ denotes the mean value of late, missed or on-time ratios for all students of the control group; $\bar{x_2}$ denotes the mean for the same engagement metrics for the treatment group; and s denotes the pooled standard deviation for both groups. Cohen suggested that the *d* values of 0.8, 0.5, and 0.2 would represent large, medium, and small effect sizes [37].

Table 4: Submission pattern effect size (*d*) by semester

Semester	On-Time Submissions Cohen's D	Missed Submissions Cohen's D	Late Submissions Cohen's D
Fall 2016 vs Fall 2017 (Impact of 1st scaffolding)	1.39	1.0	0.08
Fall 2017 vs Fall 2018 (Impact of chunking by collaboration)	0.4	0.2	0.15
Fall 2016 vs Fall 2018 (Impact of overall approach)	0.84	0.78	0.25

We calculated the effect sizes of the three parameters of late, missed, and on-time submissions for Fall 2016, Fall 2017, and Fall 2018 in three different scenarios, as shown in table 4.

The first scenario is shown in the first row of table 4, where Fall 2016 represents the control group and Fall 2017 as the treatment focusing on the impact of the first scaffolding attempt. Analysis of these effect sizes reveals that the first level scaffolding has mostly impacted the on-time and missed submission rates in such a way that we are observing more on-time and less missed submissions in Fall 2017 compared to Fall 2016. However, the late submission rate does not suggest significant change across the two semesters. The same result is also observed in the third row of table 4, where overall scaffolding impacted mostly the on-time and missed submissions and did not show a large effect on the late submissions. The analysis of the impact of "chunking by collaboration" between Fall 2017 and Fall 2018 shows only a medium effect size for on-time submissions.

6 Summary, Recommendations and Limitations

Scaffolding is a process that guides learners towards course goals through manageable chunks or steps. In this paper, we introduced our multidimensional scaffolding methodology for a CS1 course through what we called "chunking". By chunking, we strive to create a series of related and manageable activities to give students a smooth learning experience. Our methodology proposes four dimensions of scaffolding: (1) chunking by difficulty, (2) chunking by time, (3) chunking by focus, and (4) chunking by collaboration. We determine the effectiveness of our approach and assess whether it improves student engagement by measuring on-time, late, missing submission ratios, time spent on the course, and course page views. We report on statistically significant differences across all engagement measurements between a semester with no scaffolding (Fall 2016) and a semester with the proposed scaffolding approach fully implemented (Fall 2018). Based on our evidence-based practice in CS1, we recommend the following scaffolding practices:

- balancing and scaling up the challenge level of activities and questions within activities (chunking by difficulty),
- reducing activity size to take less than an hour; breaking down activities by using checkpoints (chunking by time),
- using short PRS quizzes in between activity checkpoints to help with class pacing and providing feedback,
- diversifying activity types in a session (chunking by focus), and
- using think-pair-share to guide learner individual and social learning through discussions (chunking by collaboration).

6.1 Limitations

One significant limitation of this paper is the fact that all the data has been collected from only one course and one instructor over several semesters. Therefore, these results may not be generalizable to other situations and under different circumstances. For example, course page views and time spent seem to be suitable engagement measures only for courses that heavily utilize Learning Management Systems (LMS). On the other hand, the submission time is also impacted by the way LMS is used in the course. It is also important to note that, for example, we did use the LMS more in Fall 2017 and Fall 2018 compared to Fall 2016 because students had to complete more activities on the LMS as opposed to outside of it (e.g., using an IDE, Google Docs, etc.). Such variation in the utilization of LMS probably skewed the metrics of page views and time spent on course between Fall 2016 and the other reported semesters. Also, these metrics may include different situations, such as idle time, where students may have loaded the course page and were inactive for a while.

References

[1] N. Dehbozorgi, S. MacNeil, M. L. Maher, and M. Dorodchi, “A comparison of lecture-based and active learning design patterns in cs education,” in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–8.

[2] C. Bonwell and J. A. Eison, “Active learning: Creating excitement in the classroom,” *ASHEERIC Higher Education Report No. 1, George Washington University, Washington DC.*, 1991.

[3] C. Meyers *et al.*, *Promoting Active Learning: Strategies for the College Classroom. 1st ed.* Jossey-Bass, 1993.

[4] M. Prince, “Does active learning work? a review of the research,” *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[5] X. Ge, K. A. Yamashiro, and J. Lee, “Pre-class planning to scaffold students for online collaborative learning activities,” *Educational Technology & Society*, vol. 3, no. 3, pp. 159–168, 2000.

[6] D. Wood, J. Bruner, and G. Ross, “The role of tutoring in problem solving,” *Journal of Child Psychology and Child Psychiatry*, vol. 17, no. 89, pp. 5–18, 1976.

[7] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978.

[8] J. Van de Pol, M. Volman, and J. Beishuizen, “Scaffolding in teacher–student interaction: A decade of research,” *Educational psychology review*, vol. 22, no. 3, pp. 271–296, 2010.

[9] A. Vihavainen, T. Vikberg, M. Luukkainen, and M. . P”artel, in *Scaffolding students’ learning using test my code*. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education . ACM, pp. 117–122.

[10] N. G. L. . K. A. J. Xie, B., “An explicit strategy to scaffold novice program tracing,” in *49th ACM Technical Symposium on Computer Science Education*, 2018, pp. 344–349.

[11] J. Fitzgerald and M. F. Graves, “Reading supports for all,” *Educational Leadership*, vol. 62, no. 4, pp. 68–71, 2005.

[12] M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. MacNeil, and N. Dehbozorgi, “Design and implementation of an activity-based introductory computer science course (cs1) with periodic reflections validated by learning analytics,” in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–8.

[13] M. Dorodchi, N. Dehbozorgi, and T. K. Frevert, “”i wish i could rank my exam’s challenge level!”: An algorithm of bloom’s taxonomy in teaching cs1,” in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–5.

[14] L. Vygotsky, “Thought and language.(edited by a. kozulin),” *Cambridge, MA: Massachusetts Institute of Technology.(Original work published in 1934)*, 1986.

[15] C. A. Stone, “What is missing in the metaphor of scaffolding,” *Contexts for learning: Sociocultural dynamics in children’s development*, pp. 169–183, 1993.

[16] R. Althauser and J. M. Matuga, *On the pedagogy of electronic instruction, In C. J. Bonk and K. S. King (Eds.), Electronic Collaborators: Learner-centered Technologies for Literacy, Apprenticeship and Discourse*. Mahawah, NJ: LEA Associates, 1998.

[17] T. Kahn, “A learning agenda: Putting people first,” *Palo Alto, CA: Institute for Research on Learning*, 1993.

[18] L. Dixon-Krauss, *Vygotsky in the Classroom: Mediated Literacy Instruction and Assessment.* ERIC, 1996.

[19] S. H. Edwards and P.-Q. M. A. A. A. K. D. . T. B. Snyder, J., “Comparing effective and ineffective behaviors of student programmers.” *In Proceedings of the fifth international workshop on Computing education research workshop (). ACM*, pp. 3–14, August 2009.

[20] C. Ott, A. Robins, P. Haden, and K. Shephard, “Illustrating performance indicators and course characteristics to support students’ self-regulated learning in cs1,” *Computer Science Education*, vol. 25, no. 2, pp. 174–198, 2015.

[21] J. B. Fenwick, C. Norris, F. Barry, J. Rountree, C. Spicer, and S. Cheek, "Another look at the behaviors of novice programmers." Acm Sigcse . New York and Ny: ACM, 2009, pp. 296–300.

[22] N. Falkner and K. Falkner, "A fast measure for identifying at-risk students in computer science." ICER'12 . , NY: ACM, 2012, pp. 55–62.

[23] S. H. Edwards, J. Martin, and C. A. Shaffer, "Examining classroom interventions to reduce procrastination," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 2015, pp. 254–259.

[24] M. Dorodchi, A. Benedict, and E. Al-Hossami, "Cs1 scaffolded activities: The rise of students' engagement," in *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ser. ICER '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 299. [Online]. Available: https://doi.org/10.1145/3291279.3341209

[25] D. A. Kolb, "Experiential learning: Experience as the source of learning and development," *FT press*, vol. 2014, 2014.

[26] A. Bandura and R. H. Walters, *Social learning theory*. Prentice-hall Englewood Cliffs, NJ, 1977, vol. 1.

[27] W. D. Carss, "The effects of using think-pair-share during guided reading lessons," Ph.D. dissertation, The University of Waikato, 2007.

[28] J. McTighe and F. T. Lyman Jr, "Cueing thinking in the classroom: The promise of theory-embedded tools." *Educational Leadership*, vol. 45, no. 7, pp. 18–24, 1988.

[29] K. Moss and M. Crowley, "Effective learning in science: The use of personal response systems with a wide range of audiences." Computers Education 56, 2015, pp. 36–43.

[30] R. Dellos, "Kahoot! a digital game resource for learning," *International Journal of Instructional Technology and Distance Learning*, vol. 12, no. 4, pp. 49–52, 2015.

[31] M. Dorodchi, E. Al-Hossami, M. Nagahisarchoghaei, R. S. Diwadkar, and A. Benedict, "Teaching an undergraduate software engineering course using active learning and open source projects," in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–5.

[32] D. Silver, "Using the 'zone' help reach every learner," *Kappa Delta Pi Record*, vol. 47, pp. 28–31.

[33] H. J. Klein, E. M. Whitener, and D. R. Ilgen, "The role of goal specificity in the goal-setting process," *Motivation and Emotion*, vol. 14, no. 3, pp. 179–193, 1990.

[34] D. R. Krathwohl and L. W. Anderson, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman, 2009.

[35] C. W. Starr, B. Manaris, and R. H. Stalvey, "Bloom's taxonomy revisited: specifying assessable learning objectives in computer science," *ACM SIGCSE Bulletin*, vol. 40, no. 1, pp. 261–265, 2008.

[36] C. O. Fritz, P. E. Morris, and J. J. Richler, "Effect size estimates: current use, calculations, and interpretation." *Journal of experimental psychology: General*, vol. 141, no. 1, p. 2, 2012.

[37] J. Cohen, "Statistical power analysis for the behavioral sciences. 2nd edn. hillsdale, nj: Lawrence erlbaum associates," 1988.

[38] G. Cumming, *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. New York: Taylor Francis, 2012.