Computer-Supported Agile Teaching

A Dissertation

Presented to

The Faculty of the Graduate School of Arts and Sciences

Brandeis University

Department of Computer Science

Timothy J. Hickey, Advisor

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

William T. Tarimo

August, 2016

The signed version of this signature page is on file at the Graduate School of Arts and Sciences at Brandeis University.

This dissertation, directed and approved by William T. Tarimo's committee, has been accepted and approved by the Faculty of Brandeis University in partial fulfillment of the requirements for the degree of:

DOCTOR OF PHILOSOPHY

Eric Chasalow, Dean of Arts and Sciences

Dissertation Committee:

Timothy J. Hickey, Dept. of Computer Science, Chair.

Pengyu Hong, Dept. of Computer Science, Brandeis University

Fred G. Martin, Dept. of Computer Science, University of Massachusetts Lowell

Jordan B. Pollack, Dept. of Computer Science, Brandeis University

Dedication

To my family, and especially my parents, Tadei and Agnes Tarimo, for your unwavering encouragement and support of my academic journey. My father, you worked very hard to make sure that I ultimately understand and embrace my strengths, discipline, academic excellence and hardworking. You all believed in me as you taught me to be influenced, not by our shortcomings and struggles, but in our faith, strengths, milestones and the numerous blessings. Thank you for supporting me the best way that you could.

Acknowledgments

First of all, I give thanks, glory and honor to God for guiding and protecting me. My blessings have been countless and your love is unconditional.

Over the past four years I have received support and encouragement from a great number of individuals. Prof. Timothy Hickey, you have been a mentor, colleague, and friend. Your guidance and gifted personality have made this rewarding journey possible in many ways. You are a remarkable role model - and you have inspired my academic, career and personal lives. Therefore, thank you! I would also like to thank my dissertation committee of Jordan Pollack, Pengyu Hong and Fred Martin for your support and guidance.

I owe thanks to the faculty, fellow students, and to the kind and supportive staff of Brandeis University and the Computer Science Department. I would also like to thank the teachers and students who took part in one way or another in my various studies, you generously shared your time, ideas and passion with me. Thank you Pito Salas, Rebecca Torrey, Maria Miara, Dan Perlman, Myrna Fox, and the many students of the Computer Science Department. Very special thanks go to Prof. Gary Parker of Connecticut College, I am grateful for supporting my undergraduate career and for instilling in me the foundation for research and intellectual curiosity.

Many thanks go to my many other friends, colleagues and relatives from Massachusetts, Connecticut, Waterford, Cambodia and Tanzania. I would not have accomplished my goals without your love and support. Finally, special thanks to Kesey Sar, your endless love, support, patience and sacrifice have made this possible for us.

Abstract

Computer-Supported Agile Teaching

A dissertation presented to the Faculty of the
Graduate School of Arts and Sciences of Brandeis
University, Waltham, Massachusetts

by William T. Tarimo

Agile methodologies have revolutionized the software development process through continuous assessment and adaptation of the development effort, collaboration with customers, and incremental delivery of features towards a complete product. In this dissertation we are proposing an analogous methodology for pedagogy called Computer-Supported Agile Teaching (CSAT). The CSAT framework is founded on the principles of agile teaching, active learning, and a reliance on technology in achieving pedagogy design that is efficient with decision-making that is evidence-based.

In CSAT, the discovery of effective learning and teaching methods is similar to agile-based development, where students, teaching assistants (TAs) and instructors work as a cross-functional and self-organizing team. Just as in agile, the team works in transparency (collaboration among students and the teaching staff around explicit teaching and learning objectives), inspection (of the learning and teaching processes and their effectiveness towards achieving optimal curriculum results), and in adaptation of the pedagogy (teaching and learning methods and resources) in response to the observed needs and potentials of the learners. Constant use of learning analytics, formative assessments and feedback activities are used as learning opportunities and ways to evaluate learning and teaching processes. Throughout this design, we take advantage of the affordances and capabilities of modern information technology in

achieving better efficiency and effectiveness of the supported learning and teaching activities.

In the four years of this doctoral work, we designed and developed the TeachBack web-based application to support the CSAT methodology. This process involved continuous engagement with teachers and students inside and outside of the classroom as we refined our pedagogy ideas and the corresponding TeachBack features. Through the use of TeachBack and various experimental studies we evaluated the feasibility and effectiveness of the proposed CSAT methodology.

We discovered that pedagogy-based use of computers in the classroom can effectively support learning and teaching. Moreover, data generated in computer-mediated classrooms can be used to more accurately discover learning characteristics of the individual students, to better evaluate the effectiveness of learning and teaching methods, to rapidly detect students at-risk, and to adapt the pedagogy to the needs and potentials of the learners using an evidence-based approach. These are the processes that define the implementation of CSAT in the classroom. Based on these results, we believe that the proposed CSAT methodology is both feasible and pedagogically effective, and has the potential to positively transform college teaching and learning.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This dissertation proposes a new model of teaching, Computer-Supported Agile Teaching (CSAT) and provides evidence for its necessity and effectiveness. CSAT is a framework for computer-supported pedagogy design that focuses on dynamic and adaptive teaching as an approach to discover and deliver optimal teaching and learning outcomes for each student in a large class. This thesis sets out to provide an answer to the following question: what is involved in designing a pedagogy for the 21st century classroom that is effective and efficient for the learning and teaching needs of students in large classes?

In presenting a case for CSAT, we argue that such a pedagogy design needs to be a dynamic process where content, teaching and learning opportunities work together in a flexible collaboration in service of optimal outcomes for all involved [**10**]. This is in contrast to the traditional "waterfall" model of pedagogy where the curriculum content is fixed and taught on the same instructional routine that is scheduled around coverage of the content. In this case, the learning outcomes are taken as they are — an aggregate of numeric grades from summative assignments, quizzes and exams all of which barely represent any sense of competencies mastered by the learners. Moreover, learning is taken as the responsibility of the students, they must do what they can to get the most out of the rigid teaching and learning resources.

Educational research presents many teaching and learning theories that have been shown to result in improved learning outcomes [**120**, **121**, **141**]. These include approaches such as student-centered constructivist learning[**16**,**74**,**170**], active learning, collaborative and cooperative learning, and problem-based learning. However, many institutions of higher education are still using the traditional and inferior approaches to pedagogy design [**20**] — instructor-centered teaching with students seen as passive recipients of knowledge. Moreover, literature shows the significant gap between research methods, their results and their actual adoption in classrooms [**166**].

The constantly changing nature of higher education calls for evolution and experimentation in curriculum and pedagogy design. There is an urgent need for the education system to evolve and develop new ways to address the current demands of the professional industry, advances and access to technology, and the needs of the millennial students. Like never before, there is now an unprecedented international, mobile and digital-native student population, large class sizes, increased demand for distance learning and teaching, and the dominance of social media and online technologies that offer access to high quality knowledge and interactions [**26**].

Moreover, the necessary changes must involve shifting towards increased agency for learners and rethinking of learning and teaching within a framework of digital competency [**133**]. The integration and use of technologies in the classroom should provide the opportunities to explore and discover new boundaries of learning and teaching [**133**], instead of enhancing existing traditional teaching and learning methods — a practice that is attributed to the lack of real transformation and innovation in the education system [**71**,**132**,**161**]. The access to anytime and anywhere information and learning, for example, undermines the existing notions of the teacher as the sole expert and distributor of knowledge in the teacher-centered pedagogy. It suggests different roles for the teachers, as coaches of learning and pedagogy engineers who are informed about the suitability of the pedagogy in achieving optimal learning and teaching outcomes based on the needs and potentials of the particular students.

1.1. Introduction to Computer-Supported Agile Teaching (CSAT)

The CSAT framework is founded on the principles of agile teaching, constructivist learning, and reliance on technology in achieving evidence-based pedagogy design and decision-making in the classroom. CSAT defines the process of discovering a pedagogy design with optimal learning and teaching outcomes as being similar to agile software development methodologies. Agile development gets its strength from the effectiveness of self-organizing and self-managing teams that work towards a goal instead of following tightly controlled and directed approaches that leave out the product and the customer during the development process. In Scrum [**7**], a popular approach in the agile methods, the iterative development process uses cross-functional teams, open collaboration, continuous testing, and well-managed chunks of time (sprints) that are designed to achieve specific goals and deliverables towards a whole project. This is founded on *empiricism* — a process control theory which asserts that knowledge comes from experience and making decisions based on what is known, and is based on transparency, inspection and adaptation [**140**].

In CSAT, the discovery of effective learning and teaching methods is similar to agile-based development, where students, teaching assistants (TAs) and instructors work as a cross-functional and self-organizing team. Just as in agile and scrum, the team works in transparency (collaboration among students and the instructor around explicit teaching and learning objectives and methods), inspection (of the learning and teaching processes and their effectiveness towards achieving optimal curriculum results), and in adaptation of the pedagogy (teaching and learning methods) in response to the observed needs and potentials of the learners. When an activity is not effective, then the team must pivot and explore alternative approaches, including changing the objectives, before moving on.

CSAT shifts the focus of pedagogy design from instructor-centered instruction, passive learning, and curriculum content coverage to the learning process, the particular students in the classroom and a collaborative interaction between the teaching and learning processes in a manner that is flexible and adaptive to the learners' needs. This requires experimenting with various forms of active, constructivist, collaborative and problem-based learning approaches where instructional and learning approaches are continuously assessed for their effectiveness and efficiency in achieving the learning objectives. Constant use of learning analytics, formative assessments and feedback activities are used as learning opportunities and ways to evaluate learning and teaching processes. Throughout this design, we take advantage of the affordances and capabilities of modern information technology in achieving better efficiency and effectiveness of the supported learning and teaching activities.

During the four years of this doctoral work, we reviewed research on education technology and pedagogy design in the classroom, developed technology to support dynamic teaching and learning, and implemented computer-supported pedagogy in the classroom. During this time, we designed and developed the TeachBack web-based application to facilitate interactive, active and dynamic teaching and learning in the classroom. This process involved continuous engagement with teachers and students in the classroom as we refined our pedagogy ideas and the TeachBack features to facilitate these ideas. Through the use of TeachBack and various experimental studies we evaluated the feasibility and effectiveness of the proposed CSAT methodology.

1.2. Research Questions

The CSAT framework is based on the hypothesis that teaching and learning can be effectively supported by computers in the classroom. Moreover, we hypothesize that the teachers and students of the 21st century are capable of engaging in a computer-supported classroom in a way that is substantially more beneficial to their learning and teaching outcomes than the current best practices. Furthermore, these technology tools can be used to facilitate learning and teaching activities in such a way that results in data that can be used to continuously assess and adapt the learning and teaching processes with goal of improving pedagogy results. We also hypothesize that the CSAT methodology can be used in online, hybrid and mixed online/physical classrooms. In order to test these hypotheses, we formulated the following set of research questions which assess the various aspects of the CSAT framework.

(1) How does CSAT improve learning and teaching outcomes?
(2) How can we accurately predict and monitor learning outcomes in a CSAT classroom? Moreover, can we proactively detect and support at-risk students?
(3) How can CSAT be effectively implemented in hybrid and mixed online/physical classrooms?

1.3. Contributions of the Thesis

This dissertation proposes Computer-Supported Agile Teaching (CSAT) - a new computer-supported pedagogy that is based on the practices of agile software development.

We also designed and developed TeachBack - a web-based application that is designed to support the CSAT pedagogy in the classroom. The application facilitates pedagogical activities such as individual and collaborative formative assessment, problem-based learning, assessment of affective and cognitive feedback, classroom communication and assistance with questions through a supervised back-channel forum, students' note-taking, and collection as well as analysis of students' data from class participation and performance in the various active learning and assessment activities. TeachBack has been successfully used in several classes and by different instructors over the last three years to support flipped classrooms and the CSAT methodology.

During the four years of this doctoral work, we conducted research studies to evaluate the feasibility and effectiveness of the CSAT model using TeachBack. Among other results, we discovered that computers can be successfully used in the classroom to support learning and teaching without any negative impact on students. Moreover, we discovered that the use of technology in the classroom has to be based on supporting a clearly defined pedagogy plan in order to yield positive outcomes. Data generated in computer-mediated classrooms can be used to more accurately discover students' individual learning characteristics, the most effective learning and teaching methods for these students, early detection of students who are at-risk, and how to best adapt the teacher's pedagogy to the needs and potentials of the learners.

We also show that we can successfully implement and reproduce results of various research-based learning and teaching approaches from the literature within the CSAT framework. These include the benefits of cooperative, collaborative and problem-based learning activities using the Questions and GroupWork tools, as well as the benefits of continuous monitoring and assessment of learning using the Stats and At-Risk features. In our most recent study we showed how to successfully support the CSAT model in hybrid and mixed online/physical classrooms using TeachBack and live class streaming. Results from our experiments and our experiences demonstrate that the proposed CSAT methodology is both feasible and pedagogically effective, and has the potential to revolutionize teaching and learning in higher education.

1.4. Organization of the Thesis

In Chapter 2 we present the theories and literature that form the foundations for the CSAT model and as well as the design for TeachBack. The chapter provides an overview of the learning and teaching theories and methods that empirical and literature studies have shown to be most effective in the classroom. A review is given on the principles of backward design and learning objectives in curriculum and pedagogy design. We also explore methods of assessment and the emerging development and use of forms of learning analytics for educational uses and environments. We also give a detailed review of the literature on the agile software development methodologies and how they have been adopted in several ways by other researchers in education. A critique is given on the limitations of the traditional approaches to teaching, learning and adopting technologies in the classroom. The chapter ends by reviewing the current uses, merits and shortcomings of technology in the education space.

Chapter 3 describes the TeachBack application and how it was designed and developed. It also reviews the pedagogical motives underlying the design of the various tools and features. This chapter presents an extensive overview of the user interface, interactions, and functionality offered to students and teachers as the main types of TeachBack users inside and outside the classroom.

Chapter 4 describes how CSAT can be supported using TeachBack. We review how the design and features of TeachBack correspond with the design and requirements of implementing a CSAT-based pedagogy in the classroom. The chapter describes examples of use cases or general procedures of using TeachBack in facilitating various learning and teaching activities according to the CSAT model. The chapter ends with an overview of the research studies that were conducted to evaluate the feasibility and effectiveness of CSAT.

Chapters 5, 6, 7 and 8 are the four research studies that were conducted to evaluate the feasibility and effectiveness of the CSAT model using TeachBack. Each of the studies include the motivation, experimental designs, results and discussions of the findings in light of CSAT.

The thesis ends with conclusions in Chapter 9 where the research findings are summarized. The chapter also discusses the limitations of the thesis as well as suggests future work and improvements.

In the Appendix section of the thesis, Chapter A describes (Computer-Supported) Scrum-Based Agile Pedagogy (SBAP), an extension of the CSAT which is more tightly analogous to the Scrum methodology of software development. The SBAP methodology is presented as being more revolutionary and potentially a more effective approach in transforming pedagogy design to meet the demands of the work industry and the technology-competent class of the 21^{st} century. The chapter ends by describing how the SBAP pedagogy framework is implemented inside and outside of the classroom.

CHAPTER 2

Theoretical Foundations for CSAT

This chapter describes and critiques the selections of theories and literature results that form the foundation for Computer-Supported Agile Teaching (CSAT). This review provides the rationale for the designs of TeachBack and the research questions addressed in this thesis, as well as the foundations for the experiments pursued to answer those questions.

The CSAT pedagogy design proposed in this dissertation is founded on the concept of adaptive or dynamic pedagogy inside and outside of the classroom (over the duration of the course). The adaptive trait is achieved by establishing a pedagogy that is aware and responsive to both the learning needs of the students as well as the effectiveness of the teaching approaches used.

The design involves a rich selection of teaching approaches which are founded on the best learning and teaching theories in the fields of research and practice of education. In CSAT, instructional approaches are continuously assessed based on their effectiveness and efficiency in meeting the intended learning objectives. This is accomplished through various forms of assessment and feedback-based instructional activities which provides constant awareness of students' learning levels and needs. Throughout this design, we take advantage of the affordances and capabilities of modern information technology in achieving better efficiency and effectiveness of the supported learning and teaching activities.

2.1. Theories of How Students Learn

The philosophy guiding the design of learning and instructional activities in CSAT stems from student-centered and active learning theories. According to observations by Eberly et al [**113**], a significant number of institutions of higher education continue to treat information transfer (from teachers to students) as the most important method and value of teaching; this is referred to as teacher-centered teaching. However, learner-centered teaching is based on the principle that knowledge cannot be transmitted directly to learners in its complete form, but rather by guiding learners to discover and construct knowledge independently [**130**]. This change shifts the emphasis in learning and teaching from scope and nature of contents to be learned to the nature of the learning process.

Active Learning, as defined by Prince [**124**], refers to "any instructional method that engages students in the learning process". This definition is broad enough to suggest the inclusion of some traditional classroom activities such as lectures, but in contrast to the traditional teacher-centered lectures, active learning lectures would require students to take active roles such as taking notes, asking and answering questions, and reflecting. In short, active learning requires students to do meaningful learning activities and think about what they are doing [**24**]. There are numerous implementations of active learning, however, all of them employ the core of the theory which is requiring student activity and engagement in the learning process. Research shows that active learning methods result in better student attitudes and improvements in students' thinking and writing [**24**]. Moreover, the various avenues of discussions in the activities have been shown to surpass traditional lectures in students' retention of material, in motivating students for study, and in developing thinking skills [**103**].

Active learning is a super-set of the many peer-assisted and problem-based learning (PBL) approaches. **Peer-assisted Learning** refers to "the acquisition of knowledge and skills through active helping and supporting among status equals or matched companions" [**157**]. Peer-assisted learning techniques are further divided into collaborative and peer tutoring processes [**64**]. The relationships among the theories and methods that form active and student-centered learning are depicted in Figure 2.1. Most traditional pedagogy and curriculum design efforts are devoted towards con-

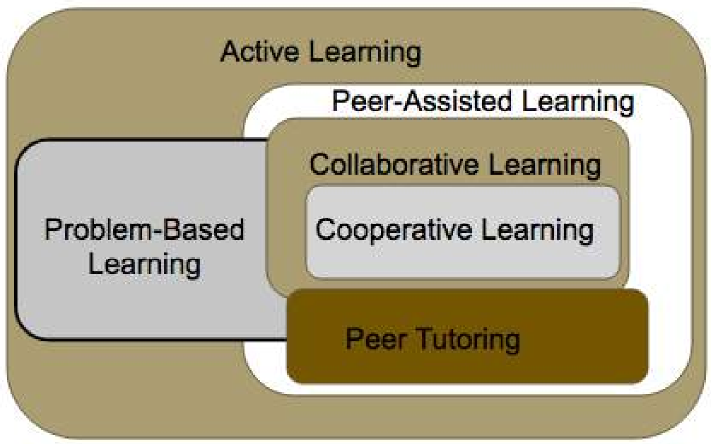

FIGURE 2.1. Active and student-centered learning theories and methods.

tent and how students interact with the materials, much less thought is put into instructor-student and student-student interactions in the learning and teaching processes. Studies by Roger and Johnson [**129**], however, claim that students' interaction patterns have a big impact on how students learn, their self-esteem, and their feelings about the instructors, course content, the school, and about each other.

Competition and individualism are still the most common modes of interaction among students during learning, especially when compared to cooperative or collaborative learning. In a meta-analysis of 124 studies that compared cooperative, competitive and individualistic learning approaches, Roger and Johnson [**129**] found the following results in favor of cooperative interactions: (1) students achieve more, (2) students are more positive about school, subject areas and instructors, (3) students are more positive about each other regardless of differences in ability, ethnic background or handicapped or not, and finally (4) students are more effective interpersonally — they are better at taking perspectives of others, taking part in controversy, interaction skills and positive expectations.

Collaborative Learning refers to any instructional method in which students work together in small groups toward a common goal [**127**]. The core of collaborative learning is the emphasis on student interactions rather than on learning as a solitary activity. Collaborative learning encompasses all group-based instructional and learning methods, such as cooperative learning and the Think-Pair-Share practices.

Cooperative Learning is the most structured component of the collaborative learning spectrum [**144**]. It is a structured form of group work where students pursue common goals while being assessed individually, and sometimes also as a group [**52**, **107**]. Its core is on the focus on cooperative incentives rather than competition or full individualism to promote learning. Cooperative learning includes three key components: (1) students work in teams toward the attainment of some superordinate goal, (2) labor is divided between team members such that each individual takes responsibility for a different sub-goal, and (3) individual contributions are pooled into a composite product to ensure that the goal is reached [**57**]. Most educational literature on cooperative learning suggests that its effectiveness depends on five factors: positive interdependence, (face-to-face) interaction, individual accountability, small groups and interpersonal skills, and group self-evaluation [**45**].

Think-Pair-Share (TPS) [**83**,**94**,**95**,**105**] is another implementation of classroom-based collaborative active learning. In a typical Think-Pair-Share activity, students work on a problem posed by the instructor, first individually, then in pairs or small groups, and finally all participate in a class-wide discussion. There are three phases to a typical TPS activity: Think — the instructor poses a question or problem to which students individually think and commit their answers, Pair — during this step students work on the problem with their neighbors or assigned small groups, and finally in the Share phase — students engage in class-wide discussions guided by the instructor where students share their individual (or small group answers), reasonings and debate alternative solutions. The TPS technique directly draws its benefits from several learning theories including active learning, cooperative learning, and the wait-time theory [**156**]. The idea of wait-time in TPS is based on providing students with sufficient processing time at the think phase, which enhances the depth and breadth of thinking [**12, 171**]. This is considered the knowledge construction stage during which students generate the valuable information of what they do know and what they do not know.

Benefits of the TPS strategy include promoting engagement, allowing students to express their reasoning, reflect on thinking, and obtain immediate feedback on their understanding and misconceptions [**88**]. Moreover, the Think-Pair-Share technique is recommended as an instructional activity that engages learners in higher-order thinking, and as a feedback mechanism both for students and teachers [**35**]. This is because TPS offers a mechanism of formative assessment in a large classroom where both instructors and students can get valuable feedback towards modifying their teaching and learning respectively. In this regard, TPS provides students with prompt and descriptive feedback on their understanding, both from peers and the instructor. It also provides instructors with with an immediate feedback on the quality of student understanding. TPS activities have also shown to enhance students' oral communication skills developed from the ample time that is devoted to discussions with one another, which results in responses that are often more intellectually concise.

Problem-based Learning (PBL) is "an instructional method where relevant problems are introduced at the beginning of the instruction cycle and used to provide the context and motivation for the learning that follows" [**124**]. According to Hmelo-Silver [**77**], problem-based learning has five goals which are geared towards helping students develop: flexible knowledge, effective problem-solving skills, self-directed learning skills, effective collaboration skills, and intrinsic motivation. Running somewhat parallel to these goals are the characteristics of PBL as described by Barrows [**13**], as learning is student-centered, learning occurs in small student groups, teachers are facilitators or guides, problems form the organizing focus and stimulus for learning, problems are a vehicle for the development of clinical problem-solving skills, and new information is acquired through self-directed learning. PBL learning is also an example of a cognitive apprenticeship [**32**] where students learn in the context of solving complex and meaningful tasks. The role of the instructor is guiding students on the learning process by pushing them to think deeply and modeling the kinds of questions that students need to be asking themselves. The evidence in support of PBL includes improving student attitudes and opinions about their programs of study [**162**] as well as improving the long-term retention of knowledge compared to traditional instructional approaches [**63**, **100**, **114**]. There is also numerous evidence that it improves student study habits in ways such as increasing library use, textbook reading, class attendance, and studying for meaning instead of simple recall [**3**, **63**, **96**, **162**]. In general, PBL is reported as being a more challenging, motivating and enjoyable approach to education [**115**] while providing a natural environment for developing problem-solving and life-long learning skills.

One implementation of active and student-centered learning that has become popular in educational research and institutions of higher education is the flipped classroom. By a simple definition, the **Flipped (or Inverted) Classroom** is a class structure where the events that have traditionally taken place inside the classroom now take place outside the classroom and vice versa [**89**]. Contrary to what this definition seems to suggest, the flipped classroom approach is more than a simple re-ordering of in-class and at-home activities. The practice employs student-centered and active learning theories to introduce (often group-based) learning activities inside the classroom and direct individual instruction (often some form of computer-based content coverage) outside the classroom. For example, a flipped course would include activities such as watching asynchronous web-based video lectures outside of the classroom and forms of open-ended problem solving in the class. According to the research, the flipped classroom approach represents an expansion and shift of the curriculum and pedagogy, rather than a mere re-arrangement of classroom and at-home activities. This expansion and shift is demonstrated in Figure 2.2.

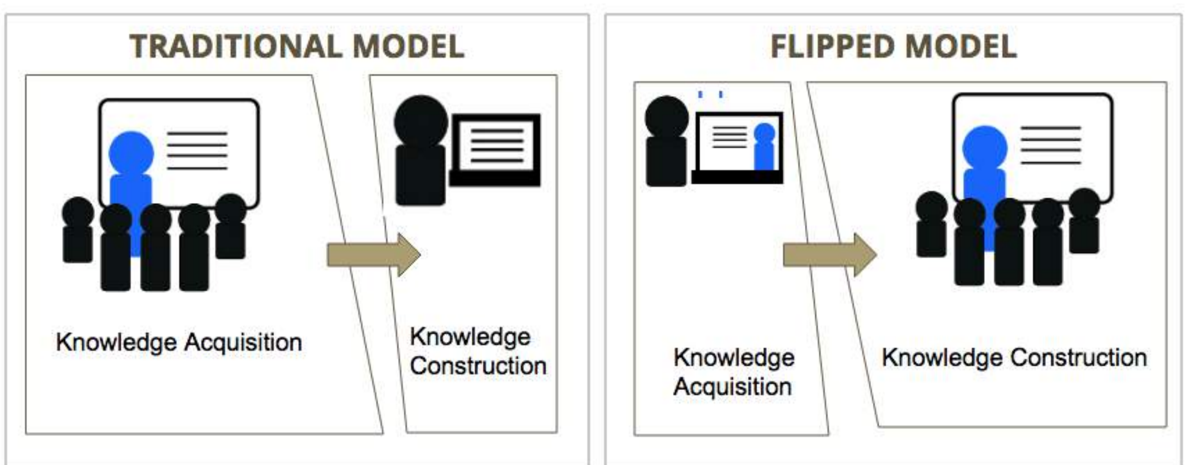

FIGURE 2.2. A comparison of the traditional and flipped classroom models.

Based on original works by Bergmann and Sams [**18**, **19**], who are credited with pioneering the flipped classroom approach, the flipped pedagogy is shown to help struggling students, and also increase student-teacher and student-student interactions in the classroom. In a flipped class, there are many more opportunities for one-on-one conversations among students and with the instructor as students work through hands-on activities and problem solving. These increased interactions have been shown to help certain types of learners to retain knowledge better [**81**]. In general, there is growing evidence that flipped classrooms enhance student participation, motivation and engagement in learning [**111**], especially when compared to traditional instructor-centered classrooms where students are passive recipients of lecture presentations. All of these factors have been shown to result in improved academic performance [**41**, **109**].

2.2. Backward Design and Learning Objectives

The consideration of the student-centered and active learning approaches in the design of CSAT serve to increase the potential learning outcomes compared to the traditional teacher-centered, passive transfer of knowledge. Moreover, these approaches force instructors and educational institutions to move the focus of curriculum and pedagogy design from content to the learning process and outcomes. This focus is best achieved by the principle of backward curriculum design where all instruction methods ought to begin with the identification of the learning outcomes of a course rather than with the course materials and contents.

Most improvement initiatives in education are said to focus on two areas, the first is the classroom where an emphasis is given to effective learning and instructional practices, and the second is institutional — focusing on creating results-oriented institutions that use analysis of achievement data to develop improvement plans [**106**].

According to the framework outlined in *Understanding by Design* by Wiggins and McTighe [**168**], schools can achieve these by first thinking carefully about the desired outcomes and then working backward to design and develop appropriate learning, instructional and assessment plans to achieve them. Their framework provides a three stage backward design process that guides the design of curriculum and assessment: (1) determine learning goals; (2) collect, analyze, and summarize evidence from multiple sources of data to determine how well students are learning; (3) consider the root causes of achievements (and short-comings) and then determine appropriate actions to promote enduring learning and increase performance outcomes [**168**]. This approach covers two types of outcomes, learning outcomes and program outcomes. Learning outcomes describe what students are expected to be able to demonstrate in terms of knowledge, skills and values upon completion of a course, a span of several courses, or a program. Program outcomes refer to what an educational program is expected to accomplish.

The proposed CSAT design focuses the backward design approach to the classroom and course levels. Furthermore, the tasks of designing the curriculum and program outcomes are usually done at the program or the institution levels, as opposed to being duties of course instructors. Therefore, our efforts will focus on the pedagogy design on the part of instructors and students in achieving the specified learning goals of a course. For instance, to effectively achieve the learning goals for a course, instructors must fully consider the desired learning outcomes when choosing the materials to be included in the teaching, the instructional strategies, the schedule, and the syllabus design [**39**].

A clear specification and understanding of the learning objectives provides the foundation for selecting effective instructional activities and materials, and choosing appropriate targets for assessment, and evaluating the effectiveness of the teaching and learning processes. Traditionally, courses and syllabi have been described using lists of topics without providing a sufficient breakdown of specific learning objectives or the levels of expertise to be achieved by students across each topic and its parts [**147**]. Moreover, in the study by Starr et al [**147**], non-expert faculty members are said to face the problems in bridging the gaps between a course description and what to teach in the classroom, as well as the gap between what has been taught in the classroom and how to assess what students have learned and not learned. In CSAT we propose a pedagogy design where learning objectives are explicitly defined at the various stages, starting at the course and descending through the units, classes and all the way down to the individual instructional/class activities used in the classroom. This approach transforms a course description or syllabus into clear components that need to be taught and assessed in the classroom.

2.3. Assessment and Learning Analytics

The second step outlined in *Understanding by Design* is the assessment of learning and teaching through continuous collection, analysis and summarization of evidence from multiple sources of data in order to determine how well students are learning. Osters and Tiu [**117**] define assessment in the general context of education as "a systematic and on-going process of collecting, interpreting, and acting on information relating to the goals and outcomes developed to support the institution's mission and purpose". Two popular forms of assessment are summative and formative assessments. Summative assessment generally takes place after a specified period of instruction (end of topic, mid-term, end of course - final exam) and requires making a judgement about the learning that has occurred towards a single grade representing overall competency of a learner. Formative assessment, on the other hand, refers to a diagnostic use of assessment that is intended to provide feedback to students and teachers about the learning and instruction as well as enabling instructors to adapt the teaching and learning to meet the learning needs of the students. Even though summative assessment results can be used at the program and institution levels to potentially monitor and improve 'long-term' program outcomes, their nature is to serve as 'after-the-fact' measures of competency, usually represented as a single variable. Moreover, Osters and Tiu [**117**] argue that grades alone do not provide adequate feedback to students' performance because grades represent overall competency of students and do not identify strengths and weaknesses on specific learning outcomes.

In contrast to summative assessments, formative assessments can be done through the various forms of activities that teachers and students undertake in order to assess the various learning outcomes at various points in the course of the semester. For example, a formative assessment can be used at the end of an instructional activity, sub-topic or class meeting in order to assess whether the intended learning objectives have been mastered by students. And since the goal is to inform on the learning relative to the learning goals, an instructor can use the information to adjust the teaching as necessary. Additionally, feedback is an important part of formative assessment where learners become aware of any gaps between the desired learning goals and their current knowledge, understanding, or skills; the feedback can also guide students through the necessary actions to reach the goals. [**125**, **134**]. This feedback usually comes from the instructor, but it can also be in the form of self-evaluation by learners themselves through the established understanding of the learning objectives and assessment criteria. Learners who are given opportunities to self-reflect have been shown to achieve greater performance improvements compared to those who do not [**56**, **59**]. Due to the various learning opportunities afforded in formative assessments, researchers and practitioners have come to acknowledge the shift in formative assessment from 'assessment of learning' to 'assessment for learning' and 'assessment as learning' [**158**].

The reach and efficiency of the second and third steps in the *Understanding by Design* framework can be improved beyond simply using formative assessments to evaluate learning and teaching to facilitate instructional decision-making and design of interventions that are intended to achieve outcomes. This is because learning outcomes depend on many factors, including student motivation, learning styles, previous background, teaching methods, grading scales, and assessment difficulty levels. [**1**]. A more holistic assessment ought to consider all such variables that may be playing critical roles in the learning and teaching outcomes. As education delivery is increasingly diversifying into online and blended learning environments, teaching and learning often include the use of technology in the forms of web-based environments and systems that facilitate and manage learning and teaching [**5**]. These environments capture an ever-increasing amount of data about students' learning behaviors, experiences, characteristics and assessment.

Siemens and Long [**143**] define the practice of **Learning Analytics** as the "measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and environments in which it occurs." Most literature relates Learning Analytics to Academic Analytics (AA) and Educational Data Mining (EDM), however, some researchers distinguish the three practices as having different stakeholders with distinct purposes at various levels of abstraction [**53**,**142**]. These various forms of learning analytics have been under-used in educational research in general [**51**], and higher education institutions have only recently started to take advantage of the potential of these analytical tools [**21**].

The new ability to use the predictive and analytical power of learning analytics together with traditional statistical analysis (such as classification, clustering, association rules, and visualization) to enable deeper and richer understanding of students' learning experiences and behaviors is a critical step forward. Recent empirical studies show the effectiveness of using learning analytics with students' data to streamline the teaching and learning processes, and to more accurately extract and analyze learning, navigation patterns and behaviors, as well as to provide feedback to students and faculty about the unfolding learning and teaching experiences [**65**,**80**]. Other studies have used learning analytics in predicting and monitoring at-risk students [**165**,**169**], dropouts from MOOCs (Massive Open Online Courses) [**72**], and academic risk in online teaching environments [**138**].

The power of learning analytics comes from its ability to discover hidden relationships, patterns and interdependencies among various measures of learning and performance outcomes with students' learning characteristics, behaviors, and patterns. For example, discovering, monitoring and predicting the key performance indicators (KPI) of students can help in designing, tailoring and targeting highly effective and efficient intervention approaches [**40**, **54**]. This goes beyond the affordances of formative assessment activities in assessing and adapting learning and teaching to the needs of the learners relative to learning objectives.

2.4. Agile Software Development and Pedagogy Design

The agile software development methodologies are derived from the Agile Manifesto [**6**], an outline of principles and values formulated to address the need for an alternative to the documentation-driven and heavy-weight software development processes of the past. The manifesto is the work of the Agile Alliance, a group of representatives from Extreme Programming, Scrum, Adaptive Software Development, Agile Organizing Framework, Pragmatic Programming and others who were originally sympathetic to the cause. The philosophy guiding agile software development came about as a result of the observation that self-organizing and managing teams working towards a goal are more effective than tightly controlled and directed approaches to software development.

Prior to the formation of the agile methodologies, software development was characterized as a documentation-driven and heavy-weight process. The projects followed what was known as the 'waterfall' methodology where developers would start by agreeing to the full list of customer requirements, then design and implement the software, test it and then deploy it to the customer, and making sure that things kept working for the customer. The traditional pedagogy design is similar to the waterfall methodology once seen in software development. Most current curriculum designs start with creating the syllabus (a list of things students should be taught), division of syllabus/topics into class schedule, implementation in the classroom, and using summative assessments for grades [**20**]. This approach results in the familiar instructor-centered teaching with students seen as passive recipients of knowledge. In addressing this, the academic community has been working on various efforts to develop techniques that increase student participation and engagement in the learning process, as well as to create opportunities for active and collaborative learning which have been shown to facilitate improved learning outcomes [**141**]. Some of these efforts have been inspired in different ways by the methods of agile software development.

Agile methods of development have also moved into the academic environment with a number of studies reporting on courses devoted to teaching agile software development methods [**4**, **17**, **23**, **76**]. Other studies report on incorporating agile methods in the instructional approaches for computer science and engineering courses as a way to achieve instructional and learning goals. Pope-Ruark and others [**123**] adopted scrum principles in redesigning the curriculum for two semester-long project-based courses. The authors observed that despite working in various group projects, students had not developed identities as effective collaborators. And therefore the courses were redesigned to create experiences that would help students learn the process of effective collaboration and build identities as engaged collaborators despite the individualistic and competitive environment of the grade-based academic system. The scrum approach helped students learn to collaborate in “the true sense of the word” — to plan, communicate and execute as a team [**123**]. It also enabled students to actively create, reflect on, and articulate their roles as collaborators rather than reinforce individualistic or competitive mentalities.

In a similar study, Marija Cubric [**38**] designed a blended learning process for a project management course based on the agile development principles. The author demonstrated that agile principles can be used in any subject area to successfully work with students in achieving learning objectives especially in developing non-cognitive skills such as teamwork, giving and receiving feedback, collaborative writing and incremental and iterative development.

Andy Chun [**28**] developed the Agile Teaching/Learning methodology (ATLM) for computer science courses as a way to develop effective knowledge sharing, nurturing of self-learning and life-long learning skills. Chun compared the teaching/learning processes and software development, noting similarities in that each involved stakeholders with differing (and sometime competitive) objectives, a tight schedule, and with considerable expected and unexpected changes along the way. Based on the agile manifesto of software development, the ATLM approach values students/teachers and their interactions rather than a particular approach to teaching/learning, working knowledge rather than rote-learning, communication rather than negotiation, and responding to changes rather than just following a schedule [**28**]. As such, ATLM emphasizes agility, communication and the learning process. The approach is iterative with parallel teaching and learning cycles, allowing self-adaptive variations to be applied to future iterations. The teaching cycle includes traditional lectures, monitoring of student progress (using weekly quizzes) and receiving feedback from students and then making necessary adjustments to the schedule, content and coursework based on the students' feedback and progress. The learning cycle includes the lecture component, practice through coursework, independent study opportunities and sharing of knowledge. Even though ATLM was inspired by agile software development principles, the author argues that it can be applied to a wide variety of courses that require agility in teaching and learning.

D'Souza and Rodrigues [**47**] proposed the Extreme Pedagogy framework based on the agile software methodology of Extreme Programming. This framework was intended to address the skill deficiency, especially in engineering graduates, as a result of the traditional teaching methods. As in the case of the waterfall model of software development, the traditional instructional designs are also criticized as being too linear, dogmatic, systematic and constraining [**47**]. In order to overcome the many limitations of the traditional pedagogy, Extreme Pedagogy adapts the core practices of Extreme Programming to the teaching and learning processes in engineering education. Similar to ATML, Extreme Pedagogy values students and teachers and their interactions over administrative processes, working knowledge over grades, collaboration with students over fixed syllabi, and responding to change over following a plan. This boils down to three characteristics that govern extreme pedagogy: learning by continuous doing, learning by continuous collaboration, and learning by continuous testing.

Learning by continuous doing addresses the inefficiencies [**20**] of the traditional instructional methods which are heavily lecture-based with students seen as passive participants. It uses active learning approaches to sustain interest and motivation where students learn for understanding (deep learning) rather than mere knowledge retention (surface learning) [**123**].

Learning by continuous collaboration promotes opportunities of collaboration and communication among students and with the instructor. Collaborative and cooperative learning are well researched in education, where they are shown to promote deep learning, promote self-esteem, and improve interpersonal skills. Moreover, collaborative learning increases students' interest in learning, teaches critical thinking skills [**12**, **171**], enables longer retention of knowledge, and makes students more satisfied with their classes [**129**].

In learning by continuous testing, frequent formative and summative assessments are used to assess learning and to provide feedback to guide future class activities [**22**]. The frequent use of assessments gives valuable feedback to instructors enabling them to revise their teaching and it also helps students to monitor their own learning and to take self responsibility of their learning progress.

2.5. Review of Educational Technology

The extensive development in computers and educational technologies over the past decade is slowly being integrated into the classroom [**150**]. This is reflected by the inevitable increase and adoption of online and hybrid course offerings and related computer-mediated technologies. This educational shift and the potential of technological advancements have been embraced as positive progress by largely all parties involved in higher education, from administrators and educators to students. Researchers too, argue that these new technologies have the potential to reshape all scholarly areas [**167**]. When harnessed intelligently, information technology has already shown the potential to improve productivity in teaching and learning. Current practices and research in online and hybrid learning technologies have shown desired improvements in collaborative and active learning, individualized and timely feedback, integration of globally diverse perspectives, establishment of equal, flexible and affordable access to education [**146**].

In an analysis of five large-scale studies that review the impact of educational technologies on student achievement in all education levels [**137**], access to learning and instructional technologies is shown to result in overwhelmingly positive outcomes in student performance, learning styles and attitudes. However, some of the evidence shows that learning and teaching technologies are less effective or ineffective when the learning goals and the focus of the technology are unclear. Moreover, despite the increasing interest and investment in educational technologies, studies show the persistent disconnect between the technologies, research results, design and practice [**166**]. Therefore, despite the potential for advancing teaching and learning, the existing issues of establishing good practice, reproducing research findings, and effective implementation designs have continued to impede the rate of successfully transforming the education system.

Empirical research studies highlight various teaching and learning theories or methods that are associated with improvements to both teaching and learning outcomes [**120**, **121**]. However, there is currently little evidence that these theories are employed in the design of the educational technologies and in the pedagogies supported by them [**15**, **29**, **93**, **116**]. Perhaps, as McNaught [**104**] points out, this is due to the fact that the designers and producers of the technologies are not themselves educational experts and thus "they find the diverse array of theoretical perspectives alien and overwhelming". These and related works are published under the communities of Computer-Supported Intentional Learning Environments (CSILE) [**136**] and Computer-Supported Cooperative Work (CSCW) [**25**, **112**].

Further review of literature shows more areas that affect the successful deployment of educational technologies; these are the institutions, resources and teachers. For example, institutions are said to give little time to teachers to manage and familiarize themselves with the technologies [**128**]. Moreover, some institutions are said to be slow in embracing educational technologies and some show active resistance to change [**37**, **61**], perhaps due to lack of understanding of why to change and the details involved in the process. In addition, some schools do not see the need to change as they are content with their familiar and established ways of teaching and learning, especially their outdated cultural beliefs on the relationships between students, teachers and computers.

In the case where technology is already moving into the classroom, some faculty have also been reluctant to adopt the educational technologies and revise their pedagogies accordingly due to a number of personal, behavioral, and environmental factors. Some of the personal and behavioral factors include attitude and anxiety, self-efficacy, willingness to make a time commitment and take personal risk, computer competency, beliefs, knowledge, and perceived relevance [**33**, **49**, **62**, **78**]. According to current research, some of the environmental factors include supportive administration, sharing of resources, availability of support staff, and effective training [**33**, **78**, **108**]. Other highlighted factors that influence teachers' decisions to successfully employ computer-mediated technologies in the classroom are access to resources, quality of software and hardware, ease of use, incentives to change, commitment to professional learning, and background in formal computer training [**110**]. These reviews particularly highlight the role of pedagogy in educational technologies, suggesting that teachers' beliefs about teaching and learning with technology are central to the integration.

In concluding, research argues that a more theoretically consistent approach to adopting educational technologies is to inter-relate theory with the desired features of learning and teaching and then to map the relevant technology tools and resources against these [**34**]. This will enable practice to reflect the underpinning theories for better outcomes. Moreover, computer-mediated instructional technologies can also be appropriated accordingly based on the different contexts of learning and instruction. While engaging in various forms of computer-supported education, teachers are expected to perform a variety of tasks and undertake numerous roles such as a designer, teacher, evaluator, administrator, and a 'pedagogy engineer' and therefore have the responsibility to employ a carefully calibrated mix of instructional and learning styles and technology based on relevant pedagogy.

CHAPTER 3

TeachBack

3.1. Design Goals

TeachBack is a web application (website) designed to support CSAT by facilitating various pedagogical interactions for instructors and students in the classroom. The application provides in-class interactions such as allowing students to give feedback to the instructor, allowing instructors to conduct individual and group/collaborative formative assessments, and providing avenues for students, instructors and TAs to engage in discussions and provide real-time assistance through back-channel forums. It also provides the instructor with statistics on student performance in formative assessments and participation in class activities. The primary goal in the design is to offer an application that can facilitate various evidence-based pedagogical activities and at the same time be a platform for continuous assessment of those activities as is required for agile teaching and learning.

The decision to design TeachBack as a website enables the application to be easily accessible across many devices without having to worry about device specific platforms and operating systems. This is achieved by relying on two facts, the ubiquity of internet access and the fact that all internet-enabled devices have access to internet browsers as native or third-part applications. The choice of deployment as a web based application also means that users do not have to deal with TeachBack specific software installations which would require more expertise or help from technology staff. It also means that users do not have to deal with subsequent installations during application updates as this is solely left to the development team to deploy updates to the hosting web server.

TeachBack is developed as a Ruby On Rails [**155**] application. Ruby On Rails is an open source software framework for developing web based applications using the Ruby [**55**] programming language. Rails applications are normally database-backed web applications and strictly adhere to the Model-View-Controller (MVC) [**92**] pattern.

The Affective Tutor is an earlier version of TeachBack, and was developed in JavaScript. This web application was used in several classes during the fall 2013 and spring 2014 semesters. The design goals, features and experiences of using The Affective Tutor were published in a conference paper [**75**].

3.1.1. Interactive Pedagogy. At the heart of the TeachBack design is a set of lightweight interactive tools intended for real-time pedagogy-based interactions in the classroom. The set of interactive tools was selected based on the observation that active learning and interactive teaching result in improved learning outcomes as well as deeper learning compared to traditional lecture based learning. These tools include functionalities for students to give real-time feedback to the instructor, and to ask questions in a supervised back-channel forum where students and get help from peers or teaching assistants. They allow instructors and TAs to conduct individual or group-based formative assessments, and collect and review students' data on various levels of participation and performance within the tools and activities in the class. [**75**].

3.1.2. A Classroom Management System. By design TeachBack is also a classroom management system because it retains all users' data for extended access, use and analysis. A dedicated backend database stores data such as user accounts, courses and their enrollments, lectures, activities and all interactions, participation and performance records. This enables TeachBack to serve as a repository of learning materials whereby students can always go back to access items such their notes, formative assessment questions as well as in-class questions and answers in the forum. Students' participation, interactions and performance records are grouped and accessible at the activity, lecture and course levels. This data enables students to monitor and review their own learning and participation, and more importantly it allows instructors to monitor, assess and evaluate the learning, teaching and pedagogy at various stages of the course by simply using evidence automatically collected by TeachBack.

3.2. User Interface

TeachBack is a web based application which means that its end-users interact with the application as a website using web browser applications. The front-end website is viewable as full website on regular desktop, laptop and widescreen tablet computers. The web interface is also designed to be mobile friendly and therefore adjusts and adapts accordingly on small screen sizes so that users can still use the application from mobile devices such as smartphones and small tablets.

In order to access the application, the user needs to launch a web browser and direct the browser to TeachBack's specific web address (URL), which is currently http://teachback.herokuapp.com.

3.2.1. User Accounts and Course Enrollments. When first visiting TeachBack a user is taken to the home (root/first) page which primarily displays forms for signing in or signing up to the application. TeachBack functionality and usage is tied to a user's account for privacy reasons and also to keep user-specific records such as account information, courses enrolled as well as specific activities and interactions within the application. Therefore, when using the application for the first time a user has to signup whereby creating an account, and during all future visits a user will only need to login to the existing account.

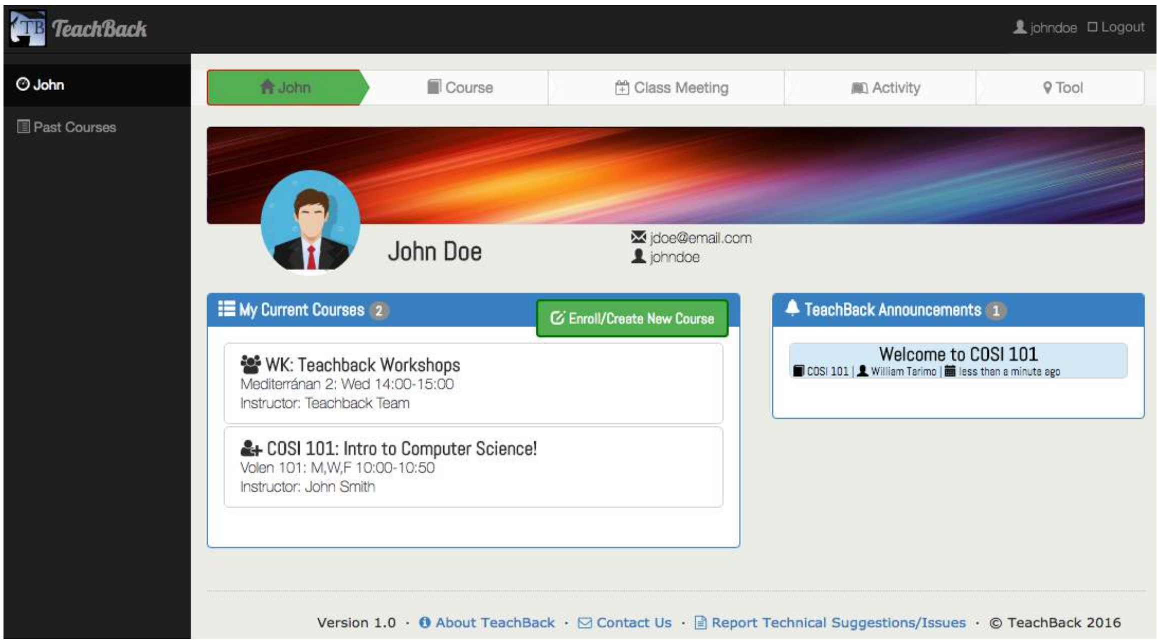

FIGURE 3.1. TeachBack: A user's profile page.

After signing up or logging in a user is taken to a profile page with a panel that lists all courses that a user is enrolled in during the current semester, see Figure 3.1. A second and smaller panel on the right side of the page lists 'current' announcements created in courses that a user is enrolled in. Courses enrolled in during all previous semesters are listed in a separate but similar page which can be accessed from the user's profile page. In the current page users can create a new course or enroll in an existing course. Clicking on the 'Enroll/Create New Course' button which opens up a dialogue with separate forms for enrolling or creating a course. During creation of a new course the application assigns to the course two randomly generated and unique 7-digit numbers to be used for enrollment into the course. One number is for enrolling instructors and teaching assistants (TAs) which offers an enrollment with admin privileges to the specific course and all its underlying components. The second number is for students' enrollment which offers privileges designed for students' interactions within the course. Ideally, an instructor would create a course at the beginning of the semester or course and then provide the appropriate enrollment numbers to additional instructors, TAs and students in order for them to enroll in the course.

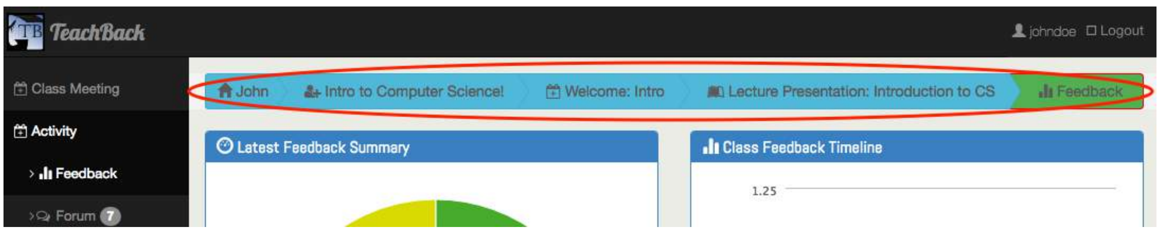

FIGURE 3.2. TeachBack: The navigation toolbar (or breadcrumb).

The user interface, interactions and features are roughly organized in the following hierarchy: User > Course > Class Meeting > Activity > interactive tool. The specific details and features from these components are explained in the following sub-sections. All pages feature a navigation toolbar (breadcrumb) with named navigation links to all the five levels of the application. As seen in Figure 3.2, this toolbar (outlined with a red oval) informs users about where they are in the application and also provides a quicker way to jump back up to any higher level in the application. On the left side of all pages (the dark region seen in Figure 3.2) is a permanent navigation bar which features dynamic navigation links based on the content of the current page. For example, at the course level for an instructor, this navigation bar will contain links for navigating to the user's profile/home page (for navigating one level up), initial course page (Course), enrollment list page (Enrollment), course editing page (Edit Course) and course specific statistics page (Course Stats).

3.2.2. Courses. A TeachBack course holds details for a specific class and once created, it maintains references to its enrolled users, their interactions, and a collection of course specific class meetings and activities as depicted in the hierarchy in in Figure 3.2. Starting at the course level users enrolled in the course as instructor or TA and those enrolled as students have slightly different interfaces and features at their exposure. For example, instructors and TAs, but not students, can edit course settings, delete the course as well as create and manipulate subsequent class meetings and activities within the course. The initial page for a course is a listing of all class meetings created under the course. In this page instructors and TAs can create new class meetings where as students can only select and interact with the meetings.

3.2.3. Class Meetings. A TeachBack Class Meeting represents a class meeting/lesson corresponding to a specific date and time. The main class meeting page lists one or more pedagogical activities for that class meeting. Activities are explained in detail in the next sub-section. TeachBack uses the course profile information to specify the explicit date and time for each class meeting; this allows the application to know when a meeting is current or active which enables further functionalities like recording students' detailed attendance. For example, if a meeting is active then the application can use a student's levels of participation to record that student's attendance as either Absent, Showed Up Only or Active Participation.

3.2.4. Activities. The TeachBack design is intended to allow the kind of pedagogies where class meetings can be divided into short pedagogical activities that have specific learning objectives. Activities may include short lecture presentations, formative assessments, group or class-wide discussions and instructor-led concept demonstrations. This model allows easy declaration and assessment of small sets of learning objectives per activity, especially though concept-focused formative assessments. Students are also able to assess the achievement reached on the specified learning objectives at the end of each activity by rating an activity in the scale 0–5. Having more than one activity per class is not mandatory and therefore it is okay for a TeachBack lecture to have a single activity.

Most of the interactive tools offered in TeachBack are accessible at the activity level. They are the **Feedback** for real-time in-class feedback, **Forum** for back-channel discussions and help on questions, **Questions** for individual formative assessment, and the **GroupWork** which is for group-based collaborative formative assessments. Based on the type of the activity or teaching preferences, an instructor can allow the class to use any selection of these tools as activity add-ons that can be toggled on or off.

3.3. Feedback

This feature provides a way for students to give affective and cognitive feedback to the instructor throughout an activity. By default, TeachBack offers three feedback sentiment choices, Engaged, Bored, and Confused; these sentiments and their chart colors can be customized by the instructor at the course or lecture settings pages. Students can give feedback at various points during an activity. Alternatively, an instructor can ask all students for feedback at any particular point especially at the end of an activity. In giving feedback, students choose any of the three sentiments along with an optional 50-character comment. This feedback and the comments are displayed in a real-time pie chart as well as a history time-line chart throughout the activity. A screenshot from the instructor feedback page appears in Figure 3.3.

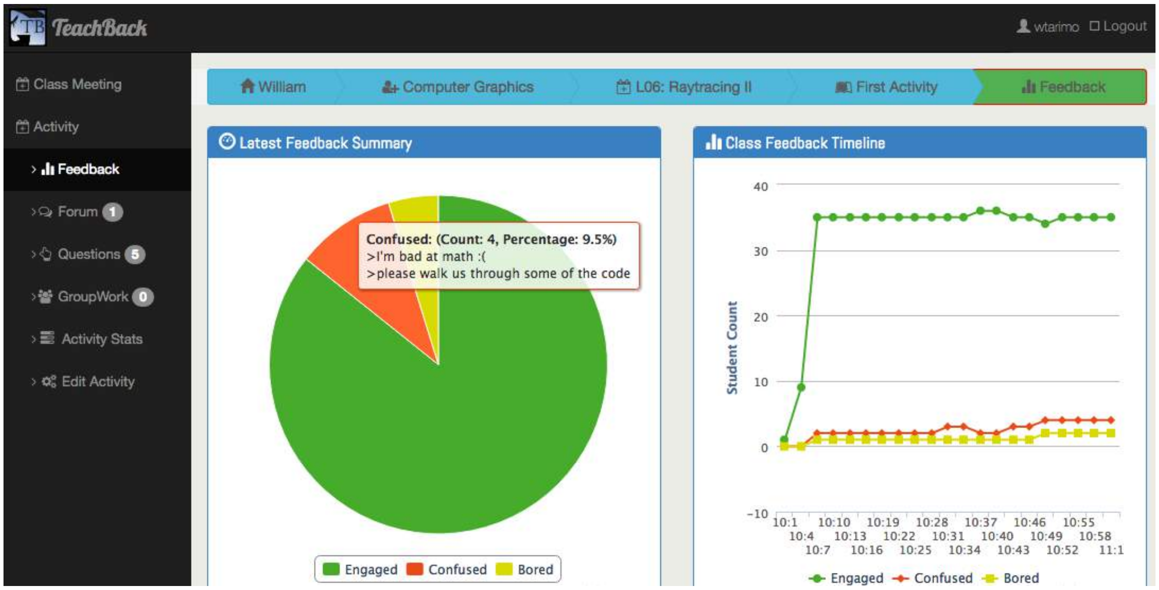

FIGURE 3.3. TeachBack: An instructor's view of student's real-time feedback.

3.4. Forum

This is a back-channel supervised discussion forum where students can post questions and ideas to the entire class or interact privately with the instructors and TAs. It is an avenue for students to get individualized help without having to ask the instructor and interrupt the class, which is especially useful in large classes with at least one TA dedicated to monitoring the forum. Students can also help each other, and an instructor can address these questions and posts where necessary. It was designed to allow TAs to provide real-time support to students in large classes during class sessions. A secondary goal was to encourage shy students in large classes to be able to ask questions. Typically, at the end of an activity, the instructor would review the feedback page as well as the forum page to address or comment on any relevant misunderstandings or points. The TAs and other students would have already provided answers to forum issues, so the main purpose is to make all students aware of these often subtle questions. A screenshot of the Forum tool can be seen in Figure 3.4.

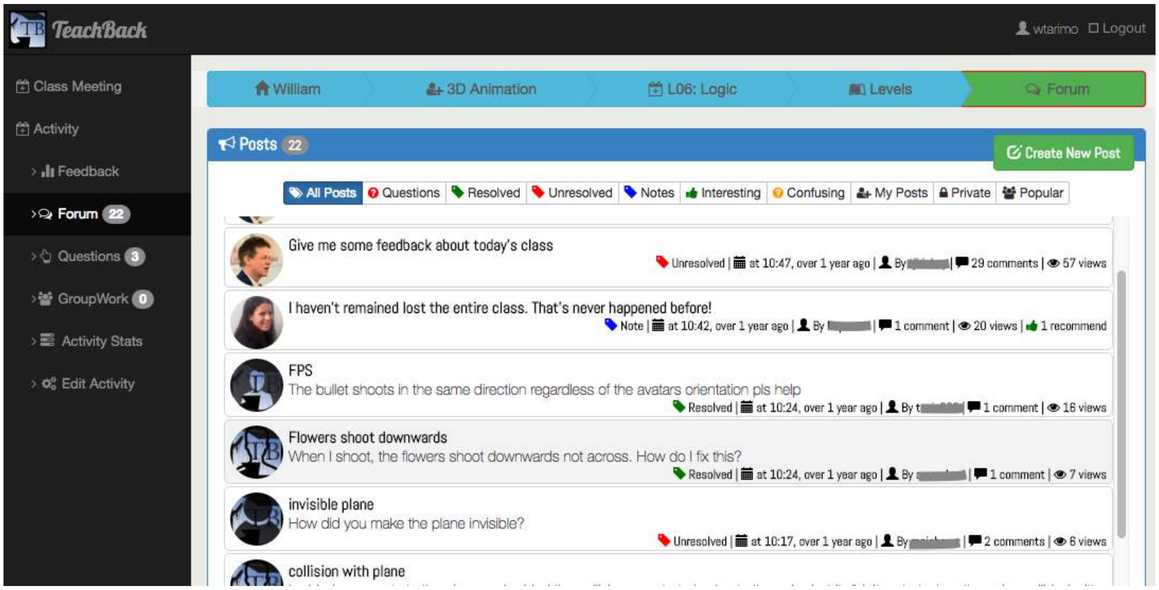

FIGURE 3.4. TeachBack: A screenshot of the Forum tool.

3.5. Questions

This is a light-weight 'clicker' or audience response system for conducting (formative) assessments. Questions allows the instructor to easily create questions of various answer types, like true/false, yes/no, numeric, text and multiple choice. Students respond using their laptops or smart phones and the instructor sees their grouped responses in real-time. Figure 3.5 shows a screenshot of the Questions page as seen by an instructor.

In the case where Questions indicates student understanding is too low, the instructor can review those topics that are confusing and easily create another query to test understanding. The Questions answers are grouped automatically based on similarity by the system and can be graded by TAs or the instructor in real time with partial credit and comments provided. The instructor can make these summaries visible to the class and these graded, anonymous responses become useful resources for students' self learning.

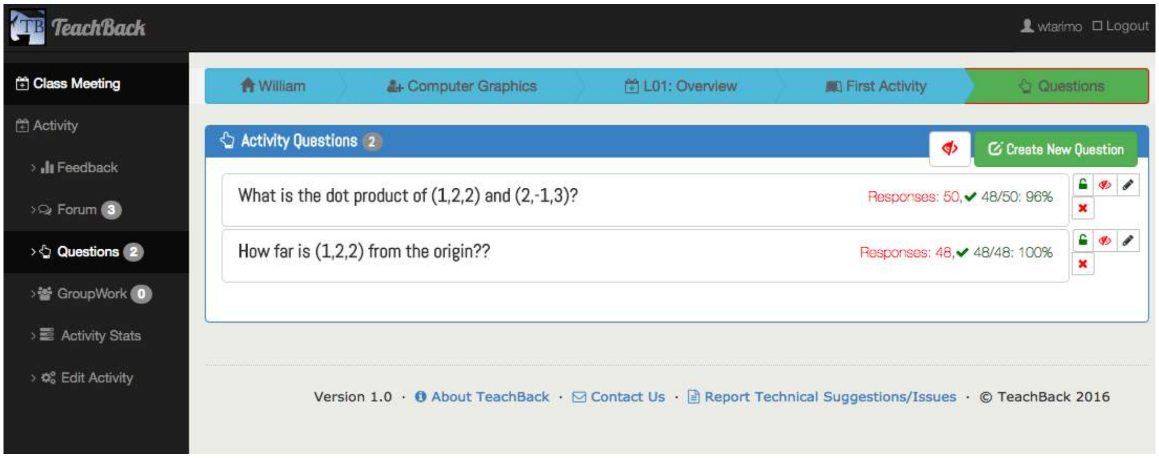

FIGURE 3.5. TeachBack: The Questions main page as seen by an instructor.

3.6. GroupWork

GroupWork is a 'clicker' functionality for conducting formative assessment similar to Questions except that GroupWork involves students working in small collaborative teams. The question structure and types are the same as in Questions but the answering process is different. In GroupWork students are automatically teamed up in groups of three based on attendance in a specific lecture. Each lecture will therefore have different group assignments for GroupWork where each student is assigned to a team based on the order that they signed into the TeachBack lecture.

The answering of GroupWork problems is loosely based on the Think/Pair/Share [**83**] methodology where each group member has to first attempt a problem individually and submit an individual answer and then afterwards have a group discussion and evaluation of the individual answers before each member has to submit a final answer. Students cannot change or submit multiple individual answers, so a student's first answer is permanently saved as an individual answer. Second or group answers, however, can be modified or changed because students submit their group-based answers by using a voting mechanism which is essentially selecting a particular answer as one's final answer. After group discussions, students will have to decide as a group (or individually in the case where is no agreement on a single response) for a final answer which can be done by entering a new answer then vote for it or by voting on any of the already existing answers from the group - including individual answers. Figure 3.6 shows a students' view of a question in GroupWork with access to answers

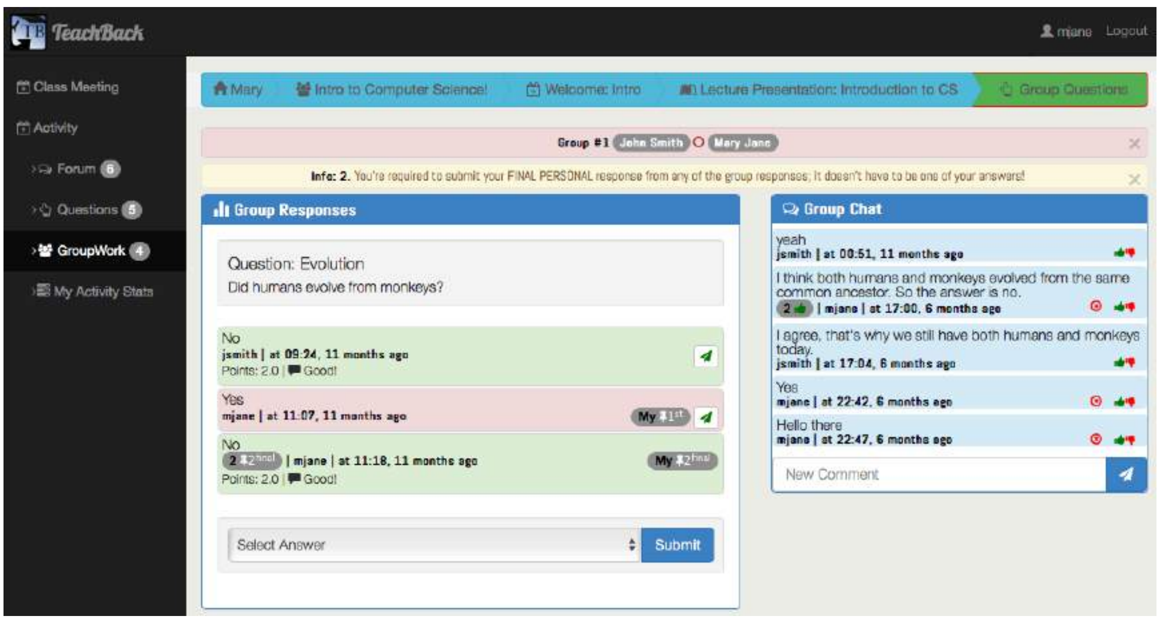

FIGURE 3.6. TeachBack: Students' question view in GroupWork.

panel and the group mini-forum. The interface is designed to include a mini-forum area for each group and question so that group members can still have a discussion in the case where members are not physically sitting next to each other. An important design feature in GroupWork is making sure that group members attempt to answer questions individually before engaging with other group members. In order to achieve this, the interface is designed such that a group member cannot see answers or messages from the rest of the group until that student has submitted his/her individual answer.

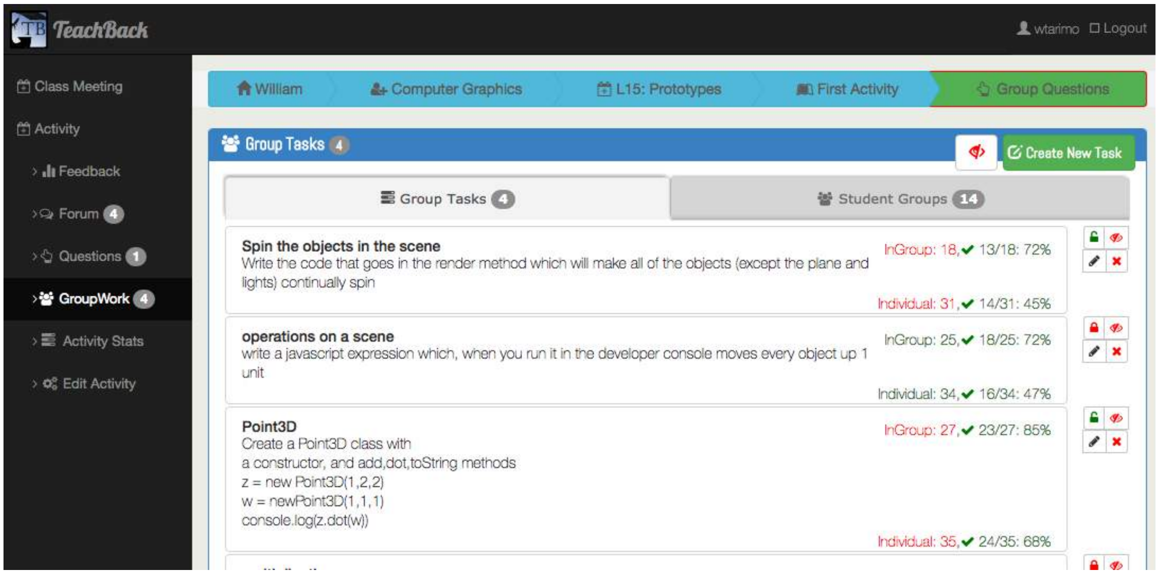

Figure 3.7. TeachBack: The main GroupWork page listing questions.

The interface is designed such that instructors can view and grade both types of answers submitted in GroupWork. In the main GroupWork page, Figure 3.7, each listed question displays the counts for individual and group responses. During class, these counts update live as students respond to questions, this allows instructors to more easily gauge and manage the progress of the assessment. After a question is graded, the percentage of correct answers is also displayed for the individual and group answers. This detail provides a convenient way to see and compare the students' performance when responding individually and in groups. When viewing and grading a GroupWork question, the individual and in-group answers are displayed in separate tabbed sub-pages. This makes it convenient to work with a particular set of answers at a time, with the flexibility to only review and grade one set or both.

3.7. Statistics

TeachBack has functionality dedicated to collecting, aggregating and displaying various measures of students' interactions, participation and performance in assessments. The links Course Stats, Meeting Stats and Activity Stats show these statistics data at the course, class meeting and activity levels respectively. When enrolled in a course as instructor/TA, these pages include data from all enrolled students, whereas for a user enrolled as a student, these pages only include data for that specific student. As seen under Course Stats in Figure 3.8, the data presented include participation and performance in Questions and GroupWork, participation and interactions in Forum and Feedback, various types of class participation under Attendance, and finally the At-Risk tab which displays students who are potentially at risk of doing poorly in the course. The At-Risk analysis is explained in detail in the following sub-section.

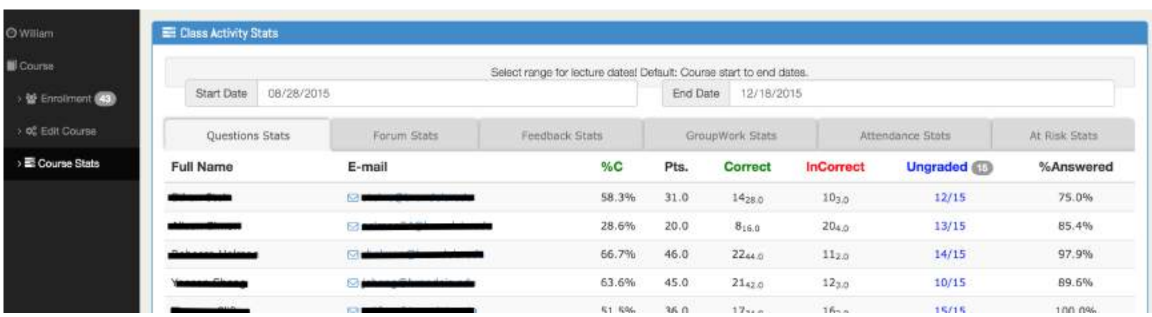

FIGURE 3.8. TeachBack: Course Statistics.

3.7.1. Students At-Risk.

The 'At Risk Stats' tab displays participation and performance records from a subset of students who might potentially be at risk of doing poorly in the course. This experimental feature was introduced as a result of the experiments reported in Chapter 6 where we studied the correlations between various cognitive and affective learning features with performance in the course. Results from those experiments demonstrated statistically significant positive correlation between each of engagement and learning speed with final performance in the course, giving evidence that analyzing these measures can be used to automatically predict which students are at risk of doing poorly in the course. Based on these findings, the 'At Risk Stats' evaluation criteria is based on both the percentage of formative assessments problems attempted and performance in those attempted questions (which is the percentage of answers that are graded as correct).

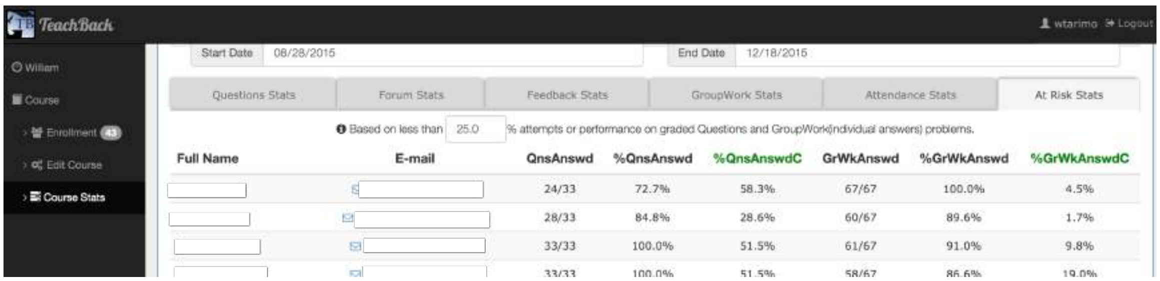

FIGURE 3.9. TeachBack: The At Risk Stats tab in Course Stats.

As seen in the screenshot of the Course Stats page in Figure 3.9, an instructor can change the threshold percentage value which is 25% by default. This data is taken from assessments conducted using Questions as well as GroupWork, however, in GroupWork, only individual answers are considered as this set represents a more accurate mastery level for individual students.

CHAPTER 4

Implementing CSAT Using TeachBack

4.1. TeachBack Design and Features

4.1.1. Activity-based Classes. In CSAT pedagogy design, class meetings are divided into one or several learning or teaching activities followed by assessment of learning objectives in the form of student feedback and/or formative assessments. The activities have explicit learning objectives which are assessed in the end to make sure that the goals are reached before moving on to new objectives. Dividing class meetings into multiple smaller chunks based on learning objectives has several advantages over class meetings without breaks. The division makes it easy and clear for the instructor to manage the communication, teaching and assessment of the many learning objectives by focusing on one subset at a time. This also makes it easier for students understand the goals of various activities and the corresponding learning expected from each. It is also easier for students to sustain engagement, motivation and attention in the classroom [**147**].

The design of class meetings in TeachBack is based on this model. In TeachBack, each *class meeting* object can have one or several *activities*, see Sub-sections 3.2.3 and 3.2.4. Each class meeting has, among other details, a title and a description; with the description intended to specify the learning objectives intended to be covered for the day. Each activity within a class meeting also has a title and a specific subset of the learning objectives. Additionally, TeachBack has a functionality that allows each student to give a rating (of 1 to 5) to an activity based on a personal assessment of how the activity achieved the intended learning objectives for them. This not only gives an opportunity to students to self-evaluate their learning outcomes, but also gives another form of student-generated feedback (average activity rating) to the instructor on the extent to which the learning goals were achieved and can be used to assess the effectiveness of the instructional methods and materials used.

Most of the interactive tools offered in TeachBack are accessible at the activity level, (see Sections 3.3 - 3.6) . They are the **Feedback** for real-time in-class affective feedback, **Forum** for back-channel discussions and help on questions, **Questions** for individual formative assessment, and the **GroupWork** which is for group-based collaborative formative assessments. All these tools can be used in a large variety of ways to facilitate the learning and teaching activities in the classroom.

4.1.2. Data Collection & Usage. All TeachBack users (including teachers & teaching assistants as well as students) must register or login to access TeachBack features. This allows TeachBack to keep track of all courses that a user has ever been associated with. Within each TeachBack course is a collection of class meetings, activities, and all data associated with activities and interactions from all users enrolled throughout the duration of the course. For students, the application collects all participation data from Forum posts and comments, Feedback records, to performance in assessments using Questions and GroupWork tools. More details on data collection and presentation are given in Section 3.7. This vast amount of data can be used in CSAT for various purposes as we will show in the following sections.

4.2. Implementing CSAT Using TeachBack - Example Use Cases

4.2.1. Pre-Class. The first scenario is in pre-class activities where students are given content to cover before a class meeting (such as readings, lecture videos or presentation slides) and then asked to interact with that content using TeachBack at home. A pre-class feedback, reflection or assessment can then be added in one of several ways.

A pre-class quiz, formative or summative, can be assigned by creating an activity with a list of questions using the Questions tool. The GroupWork tool can be used instead, where students would first attempt the quiz problems, first individually, and then in groups. The Forum can be used in several ways as well, for example, students can be asked to create one or more forum posts based on parts of the material that they found interesting or confusing. Other students, TAs and the instructor can collaborate in this activity by providing comments to the posts. Regardless of the method used, the goal of the reflection or quiz is to provide the instructor with a 'just-in-time' feedback on how many students covered the pre-class material and how to best prepare the in-class activities to address the demonstrated misunderstandings of the students. If a large number of students did not meet the expectations, then the instructor may have to radically change the lesson plan, this is analogous to *pivoting* in the agile software development process.

4.2.2. In The Classroom. Before or at the beginning of a class meeting, a corresponding class meeting object would be created in TeachBack, specifying relevant information and learning objectives to the students. Several learning and teaching activities can be used to accomplish the learning objectives of the day. A short lecture presentation activity can be used by the instructor to introduce new material or clarify concepts and misconceptions noted in the pre-class content reflection. This is also a convenient way to make sure that important concepts and materials are covered as intended, especially by students who might not have completed the pre-class activities. In TeachBack, a corresponding activity, titled 'Lecture Presentation on Chapter 1' - for example, would be created and with the appropriate learning objectives stated. The instructor can toggle on any selection of the Feedback, Forum, Questions or GroupWork tools for use during this activity.

The Feedback tool can be used to monitor students' affective and cognitive states throughout the lecture presentation. Students can be instructed to monitor their learning states and provide a feedback whenever there is a change between Engaged, Bored or Confused (these sentiment options can be customized by the instructor at the Class Meeting or Course settings). Alternatively, the instructor can ask students to provide feedback at specific break points throughout the presentation. The instructor can then respond accordingly based on the feedback levels and the short text comments provided along with the feedback. One major benefit of the Feedback tool is that students who are confused do not feel isolated. Typically, 25 - 50% of the class feel confused with new material.

The Forum tool can be used as an avenue for students to ask questions whenever they get confused during the presentation. This is especially convenient in the case of large classes where TAs can be assigned to continuously monitor the forum. It is also helpful for specific students who are uncomfortable speaking in person. In this scenario, TAs (remote or physically in the classroom), and other students (especially the ones who are momentarily Bored) can respond to these questions without interrupting the instructor. The instructor can also pause regularly in order to check the Forum and attend to students' questions. This allows classes to flow more smoothly.

At the end of the lecture presentation, the instructor should use a formative assessment to provide a series of questions to assess students' mastery of the covered learning objectives. The Questions tool can be used for this purpose. The Questions tool interface displays, in real-time, the incoming number of responses for each question. After a sufficiently large enough number have responded to a question, the instructor can (optionally lock a question and) review the responses. Responses are grouped by similarity, which is convenient for numeric and short text answers which are not automatically grouped as in the case of multiple-choice questions. The review process can also include actual grading using the one-click buttons providing Correct, Incorrect, or PartiallyCorrect quick grading options. If TAs are present, then they can assist with the grading task, including the provision of points and comments in real-time. The instructor can use the assessment results to decide whether enough students have met the learning objectives.

The GroupWork tool can be used in formative assessments in the place of the Questions tool. GroupWork activities are modeled after the Think-Pair-Share methodology. Students respond to a question, first individually, then in groups of three. The individual answers are handled in essentially the same process as if Questions was used, therefore these responses can reviewed for formative assessment purposes. Using GroupWork has more advantages over Questions as students get to engage in collaborative and cooperative learning while re-attempting problems in groups. This type of collaborative activity has been shown to result in increased learning gains, critical thinking skills, positive interpersonal skills and positive self-esteem [**45**, **88**, **115**].

4.2.3. After the Class. After a class meeting, the instructor can analyze multiple data sources from the class meeting in TeachBack and its associated activities to assess the effectiveness of the activities and identify students that need additional interventions. The instructor should also take a moment to look at the "At-Risk Stats" tab in the Course Stats page which highlights those students who might be struggling to master the concepts tested in formative assessment exercises.

A quick follow-up can be a review of class attendance records for the day presented in Meeting Stats section of a TeachBack class meeting. The corresponding Attendance Stats tab categorizes student attendance as either *Absent*, *Active Participation* or *Showed Up Only* based on the level of class participation. This is quick way to identify and reach out to students who didn't show up to class and those who didn't participate during class. The instructor should also take a moment to look at the "At-Risk Stats" page which highlights those students who performed poorly in the assessments.

If the instructor wants more detail about student performance, she can also take a closer look at participation levels by inspecting the Questions, GroupWork and Feedback Stats tabs, especially based on the extent to which these tools were used in class. The Feedback, Questions and GroupWork Stats pages at the activity and class meeting levels can reveal the students who particularly struggled to master the learning objectives. This information can be used for further intervention and support to individual students, and also in addressing misconceptions, in future classes.

More analysis of the pedagogy and student progress can be conducted after several classes or an iteration cycle based on the same pedagogy selections. A review of students' attendance, participation levels, performance in formative assessments, and feedback types can reveal more permanent attributes of student learning progress which have an impact on long-term learning outcomes. The Course Stats page shows students' records since the beginning of the course, records can also be filtered by specifying a date range. This can be used to identify and intervene with the struggling students, students at risk of poor performance in the course, and the students not well-served by the learning and teaching approaches used in the classroom.

Similarly, we can evaluate the effectiveness and efficiency of the various learning and teaching activities employed in a pedagogical sprint consisting of several classes. Activities that result in a significant number of students being consistently bored indicates a problem with the pedagogy. Similarly, an activity with consistently low performance on formative assessments might indicate that the approach is not effective or efficient in achieving the learning and teaching goals for the students or the instructor. Either way, CSAT requires an appropriate adjustment to the pedagogy in a way that leads to optimal learning outcomes.

4.3. Evaluating the Feasibility and Effectiveness of CSAT

In the last four years of this doctoral work, we reviewed research on education technology and pedagogy design in the classroom, developed technology to support dynamic teaching and learning, and implemented computer-supported pedagogy in several courses. Some of the results from these experiments were published in various related conferences. During this period we devised various methods to improve learning and teaching in the classroom. Over the four years, we designed and implemented three iterations of the TeachBack web-based application to facilitate interactive, active and dynamic teaching and learning in the classroom. This process involved continuous engagement with teachers and students in the classroom as we refined our pedagogy ideas and the TeachBack features to facilitate these ideas. We conducted several research studies to assess various aspects of the CSAT framework.

The next four chapters of the dissertation describe these studies in detail. We provide an overview of the studies in the remainder of this chapter.

4.3.1. Learning in Computer-Supported Pedagogy. In the experimental study reported in Experiment 1, we evaluated the feasibility and effects of computer use in the classroom as a formal facilitator of the learning and teaching activities. The experiment compared a computer-supported and a non-computer-supported approach to similarly flipping the CS1 class. The results of the study showed no significant differences in learning outcomes between the two approaches. We hypothesize that the computer supported sections of the class did not do better because the instructor did not use the data collected for the purposes of adjusting the pedagogy. More importantly, the use of computers in support of the pedagogy was not shown to harm students' learning outcomes. And contrary to current beliefs that computer access negatively distracts students from effective learning, most of students who participated in the study did not report distraction due to computer use in the classroom. A few students did, however, reported having better concentration in the classroom without computer use. Most of the students preferred the computer-supported pedagogy over the non-computer-mediated alternative. In addition, students approved the overall active learning pedagogy when it is computer-supported. These findings together with the additional use of TeachBack in various other classes, demonstrate that the CSAT methodology can be effectively supported by technology in the classroom. Some of these results were published in these referenced works, [**75**, **152**, **153**].

4.3.2. Monitoring and Predicting Learning Outcomes. In the study reported in Experiment 1 we did not use any of the data collected by TeachBack and other tools for purposes such as discovering the effectiveness of the learning and teaching processes and how the various learning behaviors are related to successful learning outcomes. This is because we attempted to maintain the same pedagogy with and without computer use. In the study reported in Experiment 2, we analyzed the data generated by students' interactions and participation in classroom activities that were facilitated by TeachBack in the previous semester. Student learning style features such as overall participation and engagement, as well as performance in formative assessments were shown to have a strong correlation with overall performance in the course. Interestingly, students' confidence during learning activities was, as measured by self-assessed cognitive and affective feedback, shown to be a poor predictor of learning outcomes. For example, students who often report to be confused are not the ones who are do poorly in the course. These findings show that we can use data generated from students interactions and use of online tools to discover learning characteristics in a more accurate and timely manner compared to other measures such as summative assessments alone. This opens possibilities such as better understanding of effective learning and teaching methods, feedback and assessment of learning outcomes, early detection of students at-risk, and ideas on how to best adapt pedagogy, interventions and support to struggling students. Some of these findings were published under [**154**].

4.3.3. Supporting CSAT in Mixed Online/Physical Classroom. In Experiment 3, we present an implementation of a synchronous classroom with face-to-face and remote students using TeachBack with live-streaming of class meetings. Both remote and face-to-face students would interact and participate in in-class activities using TeachBack. Echo360 was used to provide remote students with a live video-streaming of the classroom class meeting as a guide on the materials, lecture presentations, instructional activities and other interactions happening in the classroom. Results show that attending classes remotely when implemented this way has learning outcomes that are comparable to attending classes face-to-face. Moreover, it does not negatively impact students' results as is the case for absenteeism. Indeed, the combination of live class meeting streaming, TeachBack and access to lecture recordings has the potential to increase synchronous and engaged class attendance to nearly 100% which is essential for the CSAT framework. The approach allows students who would otherwise miss part or entire classes to still attend and participate remotely in a way that is technologically feasible, convenient to students, and pedagogically justifiable.

4.3.4. Supporting Research-based Learning Activities. In Experiment 4, we present and evaluate an implementation of a computer-mediated collaborative assessment and learning activity that's modeled on the Think-Pair-Share methodology. This is also an implementation of a problem-based leaning functionality. In this chapter we present the experience of using the GroupWork tool for collaborative formative assessment and analyzed the corresponding learning outcomes. Compared to individual formative assessment exercises, the results show that collaborative formative assessment has significantly higher learning gains. Students performed better individually after participating in group discussions compared to when they worked alone. The results demonstrated positive correlation between overall performance in the course and students' average performance in formative assessments when working individually and in groups. This demonstrates another computer-mediated learning activity which can be used in CSAT pedagogy for formative assessment, prediction of learning outcomes, active learning, and as a general cooperative, collaborative and problem-based learning activity.

CHAPTER 5

Experiment 1: Effects of Computer Use in the Flipped Classroom

5.1. Motivation

There is a growing body of evidence which demonstrates that active learning pedagogies improve learning outcomes in a wide variety of courses, including introductory programming courses [**8**, **14**, **148**]. It is very natural to allow students to use their laptops in class during active learning sessions of an introductory computer science course. Many faculty, however, are wary of requiring computer use during class sessions since they feel students might become distracted. This experiment sets out to establish the viability of fully using computers in the classroom.

In recent years we have seen many new developments in the way teaching and learning are accomplished inside and outside of the classroom. The last decade has seen research, development and adoption of new pedagogies, classroom technology and software applications. One such new pedagogy technique has been the 'inverted' or 'flipped' classroom in which static content is covered outside of class (through readings or videos) and class time is devoted to more interactive and engaging activities. Even though most approaches have leveraged the ubiquity of technology, flipping a classroom does not necessarily require the use of computers or other networked technology.

In this work we present our case study of partly-flipping a large CS1 class. The course was an Introduction to Programming in Java and C in which we used a partly-flipped pedagogy that combines both in-class lectures and in-class programming challenges often using a Think/Pair/Share technique [**83**]. Since the course was taught in two sections (of about 150 students each), we were able to design an experiment to evaluate the effect of two approaches to partly-flipping the classroom. The first approach is to require all students to bring a laptop or tablet to class and use their computers for various interactions, to answer questions and to solve coding challenges. The second approach is to ban computers from the classroom and to require students to solve problems with pen and paper and to be prepared to present and discuss their solution to the class if called upon.

In the computer-mediated sessions, students used two web-based applications, TeachBack (see Chapter 3) and Spinoza [**43**], to interact with the instructor and the other students while solving programming. Spinoza is a web-based Java IDE that allows students to solve simple programming problems online and provides the instructor with a real-time view of the progress of the class with similar solutions grouped together. In the non-computer sessions, we endeavored to replicate the same pedagogy using pen, paper, blackboards and the instructor's computer projected on a screen. Both sessions covered exactly the same material and used exactly the same pedagogy. Students received nearly identical lectures and were given the same programming challenges. The only difference is that in one section students were allowed to use their computers to solve the programming problems, while in the other section they had to use pen and paper only.

In the following sections we present the experimental design that was used to compare the computer and non-computer approaches to the pedagogy. We then proceed to describe the pedagogy used and we compare the way it was implemented using the computer-mediated and pen-and-paper based approaches. Finally, we present the results of the experiment and discuss its implications for computer use in the partly flipped introductory programming classroom.

5.2. Experimental Design

Introduction to Programming in Java and C is the first course in the Computer Science major in our department. Students who performed well in an equivalent CS1 course in high school may skip the course, but all other potential majors are required to take it. It was taught in two sections (self-selected by the students). One section had 136 students and the other had 148. Both sections had the same instructor, exams, homeworks, teaching assistants, and daily lesson plans. For both sections, we provided screen recordings of each class that students could review at their leisure.

The course was divided into 4 units, each lasting about 3 weeks. Each unit culminated in a 90-minute exam that provided a summative assessment of student mastery of the material for that unit. In the first two units students were required to bring their computers to class and to interact with the instructor using TeachBack and Spinoza. Ten percent of their final grade was based on the number of TeachBack formative assessment questions they answered (whether the answers were correct or not). During Units 1 and 2, students were required to bring computers to class and use them to interact with the instructor and their peers. During Unit 3, computers were banned from section 1 while still being required in section 2. During Unit 4, the protocol was reversed: computers were required in section 1 and banned in section 2. This provided us with two units of control in which both sections used computers, and two experimental units where one section required computer use and the other banned its use.

5.3. The Active Learning Pedagogy

Before each week of classes students were assigned topics or subtopics to read and as a weekly homework they were asked to submit a short reflection on what they learned and to discuss any confusing ideas in the reading. Each class had lectures intermixed with class-wide interactive activities. The lectures involved PowerPoint slides, notes from the class website, live coding demonstrations by the instructor, and visits to various websites. The interactive activities included short answer questions as well as programming challenges.

In this section we discuss the main pedagogical techniques used in the two versions of the class and along the way we introduced the TeachBack (see Chapter 3) and Spinoza tools.

5.3.1. PowerPoint Lecture Activity. Although the students were required to read the text before class, we often began a class with a PowerPoint overview of the main ideas presented in the readings. In the computer-based version of the class, students could view the PowerPoint slides on their computers and ask questions of the teaching assistants using the TeachBack Forum. In the pen-and-paper version they could print out the slides on paper before class and ask questions by raising their hands, which interrupted the class flow. In the computer-mediated version of the class, we used a web-based Integrated Development Environment.

5.3.2. Live-Coding Activity. Another lecture-style activity is when the instructor solves or demonstrates a programming problem using a Java IDE and the class watches (or in the computer-mediated version, follows along). This can be made interactive by asking students to provide suggestions for how to solve the problem. In the computer-mediated version when students are following along with the coding using Spinoza and they encounter syntax errors they can interact with the TAs using the TeachBack Forum without interrupting the class.

5.3.3. Answering Student Questions during Class. In both versions of the class, students were encouraged to ask questions if they were confused. In the pen-and-paper version, students would raise their hands and engage with the instructor while the class paused. In the computer-mediated version, students used the Forum feature of TeachBack to ask questions online, and have their questions answered by TAs assigned to the course, or sometimes by other students who were monitoring the forum. The instructor would briefly review the forum with the class at the end of most activities.

5.3.4. Posing Questions for Students to Discuss and Answer. After a lecture activity, we would usually pose a series of questions and ask the students to think for a minute about a solution, then to talk with their neighbors about their solution, and finally to share their solutions with the class. Typical examples would be predicting the result of evaluating a snippet of code, or finding a bug in a piece of code shown on the projector. In the computer-based version, we used the Questions feature of TeachBack. Figure 5.1 shows a typical activity in which the instructor projected a method on the screen and asked students to predict the return value for various calls. The Questions tool allows the instructor and TAs to not only see the solutions (grouped) but to grade them and assign points and comments. Once a sufficiently large number of students have submitted an answer, the instructor reviews the most common solutions and leads a short class discussion on the different approaches and the different kinds of errors. In the pen-and-paper version, it is difficult to determine how many students have completed the activity and it is hard to tell what the most common solutions and errors were. Students were motivated to solve problems in

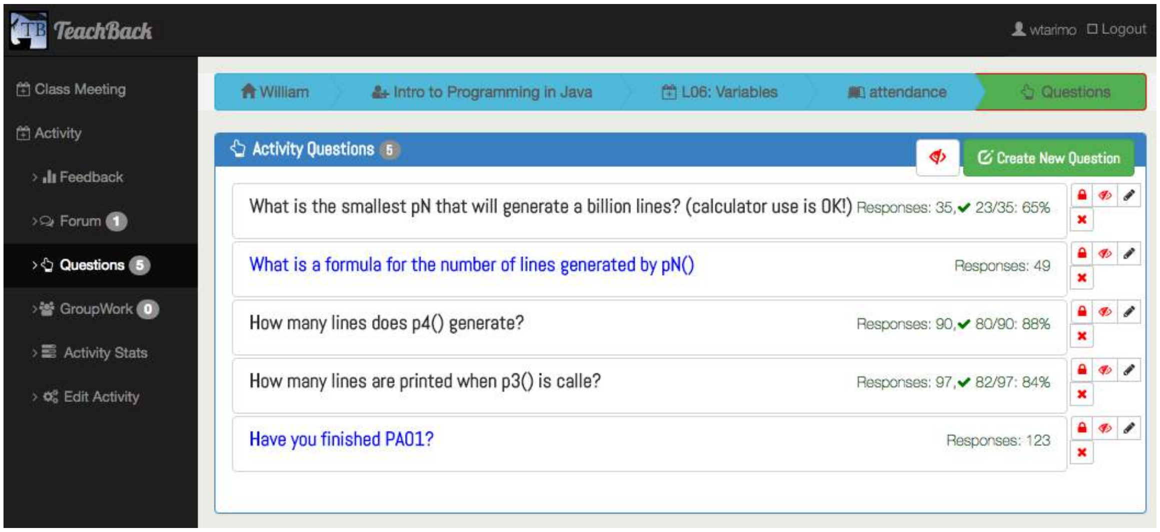

FIGURE 5.1. Classroom discussions and formative assessments using the Questions feature in TeachBack.

the pen-and-paper class by randomly selecting students to describe their solutions (possibly on the board or typing into the instructor's computer).

5.3.5. Programming Problems. In this activity, students are given a programming problem and asked to think about how they would solve it and then work with their neighbors to come up with a solution. For example, students could be asked to write a method with three integer parameters that returns true if the parameters all have different values.

In the computer-mediated version of the class, we used Spinoza to allow instructors to quickly create programming problems. Figure 5.2 shows the instructor view of a Spinoza programming problem which provides a description of the problem on the left and some initial scaffolding code in the center. Students see a very similar interface except for the instructor specific options on the far left column. Below the code there are tabbed views for Test Results (results of running instructor's unit tests on the coded solution), StdOut (code output),as well as views offering functionality to submit coded solution for grading and sharing code with other users. Students can then write, run, and debug the problem using the web-based IDE. Spinoza has an instructor's view which shows the number of students that have hit the 'Compile and Run' button and it groups the programs together based on a similarity function (ignoring white space, variables names, etc.). The instructor can see in real-time

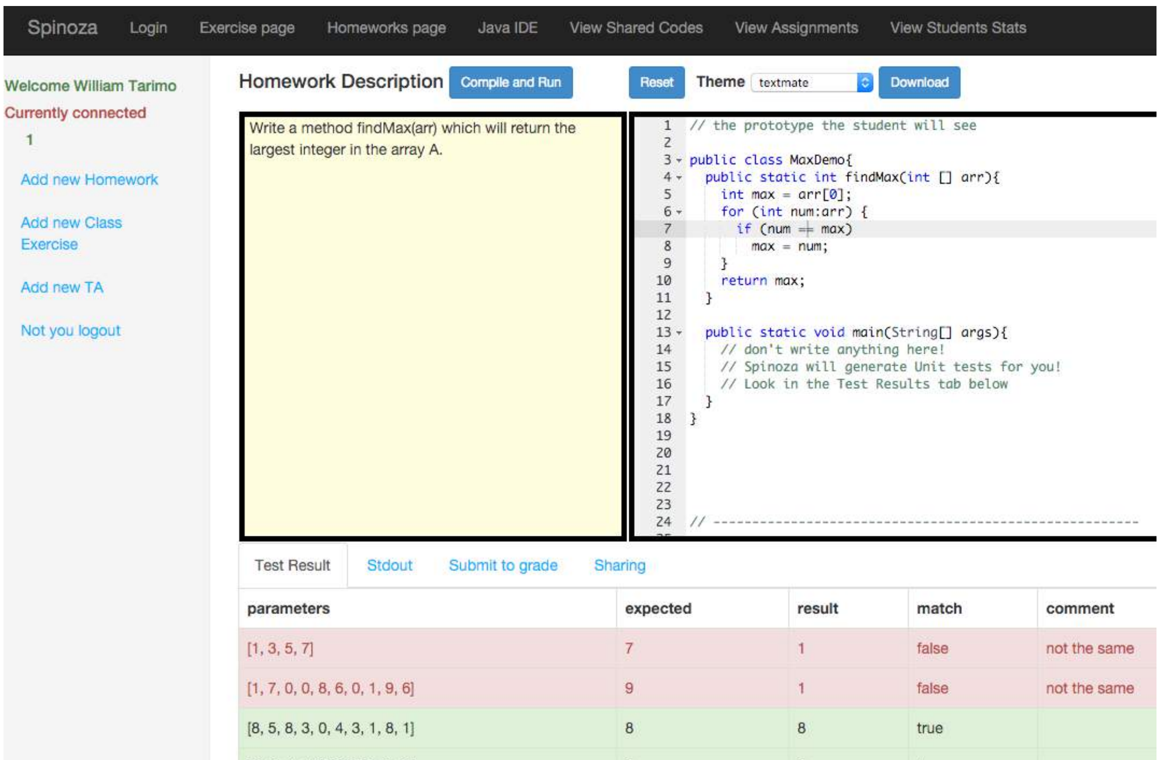

FIGURE 5.2. Spinoza: Instructor's view of a programming problem in Spinoza HomeWork.

the most popular proposed solutions to the problem and can view and debug those solutions in front of the class. The debugging process itself can be formulated as a Think/Pair/Share model [**83**], where students try to find and discuss the bugs (both syntactic and logical) in small groups before sharing with the class.

In the pen-and-paper version of the class, programming problems are displayed on the screen and students are asked to write their solutions on paper. The instructor then randomly selects students to share their solutions. The disadvantage of this approach is that the instructor does not know what the most common solutions or errors are and the process of sharing a solution with the class is more time consuming.

5.3.6. Feedback. After new material has been introduced we often ask the students for feedback, typically at the end of an activity or class. We ask whether they are confused, bored, or engaged by the material and also ask for a short comment. In the computer-mediated version, this is done using the TeachBack Feedback feature, which displays a pie chart showing the three responses. Hovering over one of the pie slices reveals a list of the comments students provided. We often find 20%–50% of students who report feeling confused when a class introduces new material (e.g. arrays or the for-each loop). This provides an excellent opportunity to reassure them that it is natural to feel confused when learning new material. The comments also show what confused them or expand on their affect. At this point the instructor also clarifies the various confusion issues. Since it is so easy to get and analyze feedback from students using TeachBack, we often get feedback after each activity in a single class. TeachBack also provides an instructor/TA view of the daily progress of individual students using performance and participation statistics at an activity, lecture and course levels. In the pen-and-paper version, we ask students to put this information on a small card or piece of paper, which is then reviewed by the instructor after the class. One disadvantage of this approach is that we cannot report the results until the following day and it can take 30 minutes to an hour to read through a few hundred separate comments.

5.4. Results

After each unit, students were asked to complete a survey where they self-assessed their level of understanding of the material in that unit as well as their level of enjoyment of the material in that unit. In units 3 and 4 they were also asked to rate each of the different styles of pedagogy employed in terms of its effectiveness for their own learning.

We kept track of the number of students from each section that visited TAs during each of the units and asked students to estimate how many hours they spent working on the course outside of class. We also kept track of each student's participation in various components of TeachBack during each class, each unit, and the semester. Finally, grades on the four unit quizzes as well as course grades were used to measure mastery of the material by unit and over the entire course.

We found four main results from our analysis of the data which we discuss below:

5.4.1. The Use of TeachBack/Spinoza in Class Does Not Harm Learning Outcomes. In Units 1 and 2, both sections used computers in class. In Unit 3, computers were banned in section 1 and required in section 2. In Unit 4, the reverse policy held, computers were required in section 1 and banned in section 2. We found that there were no statistical differences between the two sections during those units in terms of quiz scores, student satisfaction, student self-assessment of understanding, or student use of teaching assistants. From the surveys at the end of each unit, students self-reported their levels of learning and satisfaction in the range [1–5]. As seen in Figures 5.3 and 5.4, the averages on each section do not indicate any significant influence from the changes of pedagogies in units 3 and 4. Section 1 generally indicated a higher level of enjoyment, understanding, and mastery than section 2, for all units, but that increased level of understanding was not statistically

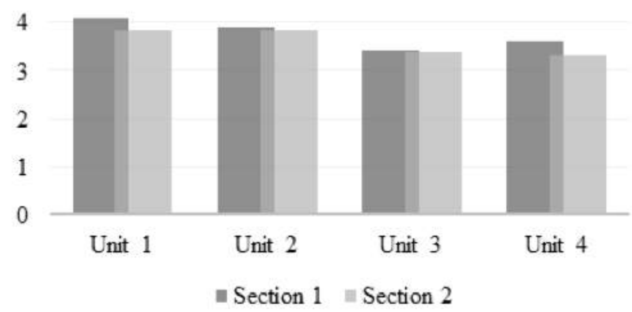

FIGURE 5.3. Average perceived enjoyment.

significant. For example, in Figure 5.4, the difference between the average under-

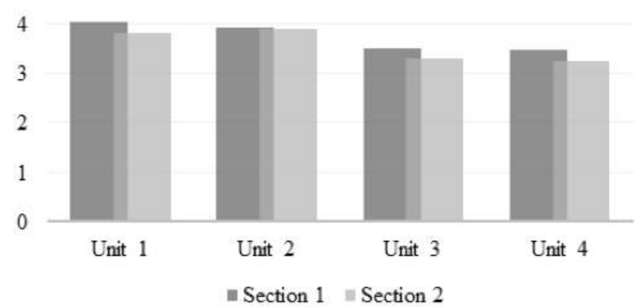

FIGURE 5.4. Average perceived understanding.

standing in unit 3 between sections 1 and section 2 was 0.17 but the p-value for the two-tailed unpaired T-test for those means was .20 which is not significant. Likewise, in Figure 5.3 the difference of average enjoyment for unit 4 between sections 1 and 2 was 0.23 but the p-value was .12, again indicating no significant difference. None of the apparent differences in section 1 and section 2 shown in these three figures was significant at the .10 level.

Looking at performance in quizzes at the end of units, if the use of computers was especially distracting, we would expect to see Section 1 outperform Section 2 in Unit 3, and the opposite occur in Unit 4. Likewise, if the use of computers was

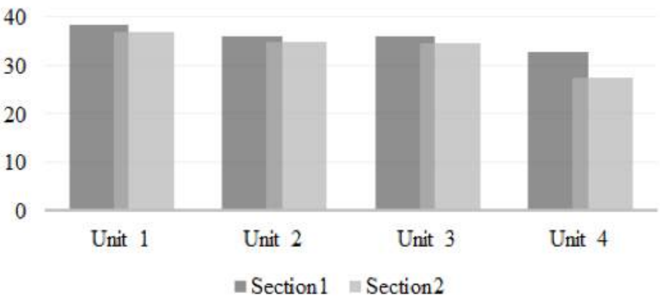

FIGURE 5.5. Average end-of-unit quiz grades.

highly beneficial we would expect Section 2 to outperform Section 1 in Unit 3, and the reverse in Unit 4. As demonstrated in Figure 5.5, no such effects were found.

5.4.2. Most Students Prefer using Computers in Class. When asked about the two different styles of active learning – writing programs with your neighbors on paper versus writing programs on your computer while talking with your neighbors, the use of computers was thought to be more effective and the results are statistically significant. Students used a five point scale to rank effectiveness of learning from 1 (not effective) to 5 (very effective). Solving programming problems with friends using pen-and-paper was ranked at 2.96/5 and solving programs using Spinoza with friends at 3.65/5 with a difference of 0.69. This is significant at the 0.001 level using a two-tailed paired T-test. The 95% confidence interval of the difference is 0.5 to 0.88.

Below are some typical comments from students after unit 4. Here is a section 1 student, happy to be able to use his computer again in class:

- "I really enjoyed when we got to live code in class. It was helpful to either follow along with what [the professor] was typing or work on building up the program with the people around us. It allowed me to see what thought process has to go into building up a program."

And here are comments from students in section 2 explaining why they were disappointed about not being able to use computers in class:

- "The lack of computers makes following along a lot less interesting and understanding class material becomes much more difficult."

- "Taking notes on paper and not being able to practice coding in class slowed down acquisition of the material greatly. It [took] much longer for this unit than others to master the material. I also disliked being asked to work in teams or to talk to people in class, but that's because I'm shy."

- "We can't use computer[s] to do real-time programing in class. To make it up, I have to go back home and watch the class recordings to brush my memory on what programing topics we went through in class that day. It is really time consuming."

5.4.3. Some Students Were Distracted by Computers in Class. A close examination of the student comments about each unit demonstrated that there was a group of students who did not feel they learned as well with computers as they did without. Indeed, a few students would attend the class meetings from the other section when the pedagogy was switched because they felt they could not learn well when required to interact with a computer in class. These were mostly students who reported being easily distracted in general. Below are some comments from students indicating what they liked about unit 4 when computers were not allowed.

- "Not using a computer, it lead me to better concentrate."

- "Not being allowed to use our computers helped for concentration and focus."

Most students, however, did not report being distracted by the use of computers in the class, contrary to the worries of many instructors. This observation is largely due to the nature of the pedagogy. The division of the class time into short interactive activities allowed students to always be engaged with the material, their peers or the instructor. There was no time for students to get side-tracked into distraction with non-class related endeavors.

5.4.4. Students Generally Approved of the Active Learning Approach.

In general, students appreciated the pedagogy used in the class, whether or not we were using computers. Here are some illustrative comments.

- "The class was very lenient towards our learning and it's a great feeling to know that the teaching staff is very forgiving for us 'newbies'. Learning is the number one goal."

- "I was forced to try to learn the material to the best of my ability beforehand to be as prepared as possible whether or not I was using my computer or notebook."

5.5. Discussion

One of the main goals of the experimental design was to be able to test the effects of the use/non-use of computers in a CS1 class in a way that removed as many possible confounding variables as possible. Therefore, both sections had the same instructor, the same course plans for every day, and each section had an exposure to a unit where computers were banned while computers were required in the other section.

We observed that the students in Section 1 did better in all four units than the students in Section 2 in terms of quiz grades, self-estimation of understanding, and level of enjoyment of the course. Section 1 was an earlier class (11:00) and Section 2 conflicted with some other large classes (e.g. Intro to Economics). Therefore, the two populations of students may have been qualitatively different, e.g. early-risers with a tendency to take Economics courses. If we had taught Section 1 without computers and Section 2 with computers, then the fact that Section 1 had "better" students would have made it more difficult to tease out any potential effect of the treatment (banning versus requiring computers in class).

Our experimental design did allow us to show that in a class in which computer-use was alternately required and banned, the student performance was not different (in a statistically significant way) in the two kinds of units. On the other hand, this experimental design also tended to hide some potential effects that might appear in an experiment where treatments don't alternate and multiple sections are assigned to computer-banned or computer-required treatments for the entire semester. In fact, we suspect that active-learning CS1 classrooms in which students are required to use TeachBack and Spinoza (or similar computer-based classroom technologies) for the entire semester may indeed have better learning outcomes than active-learning CS1 classroom that ban computer use. In this section, we provide some arguments in support of this claim.

The simplest critique of our experimental design is that although there was no statistically significant effect on the learning outcomes, it could be because the students in the computer-required (or the computer-banned) section may have had to work much harder to achieve the same performance as they did in the other section. We had one student, quoted above, who said they went back and reviewed the screen recordings every day during the computer-banned units so they could follow along with their computer and this took an enormous amount of time. It is hard however to accurately estimate how much time students are spending outside of class without some sort of automatic time-keeping protocol.

We hypothesize that the TeachBack Feedback mechanism in which students report if they are bored, confused, or engaged and then see the results for the entire class may have a beneficial effect on the students by easing their anxiety about being confused.

When they see that half of the class is regularly confused when new material is introduced they may feel less fearful and anxious and hence be able to learn more effectively. In our experimental design both sections started out with 2 units of computer-required classes and hence both benefited from this effect, if it was indeed present. Similarly, when students use Spinoza to solve programming problems in class and then are able to see that many other students in the class are making the same mistakes, we suspect that this can also reduce anxiety and improve learning effectiveness.

Another drawback of our experimental design is that it doesn't allows us to see if mandatory computer-use in class has any effects in future classes, e.g. does it improve retention in future CS courses (or not), do students in a computer-required course perform better (or worse) in the follow on courses. In our experiment both sections had three units in which computer-use was mandatory in class, and one unit in which it was banned. Any benefits either approach has over the other are averaged out over the semester and hence the experiment does not have any different impact on the two classes in years to come. This makes it impossible to study longer-term effects of computer use in the classroom.

Finally, since the protocol required that both courses use exactly the same pedagogy to the extent possible, we were not able to use the highly detailed and nuanced information that we had collected using TeachBack and Spinoza to find students who were at risk of doing poorly and target additional resources to them. In the computer-banned version of the course, the only way to assess student performance with our experimental design is through the homework and quiz grades. The quiz grades do indeed show who is doing poorly, but by then it is too late as quizzes are summative assessments. The homework grades are unreliable as they don't show how much time students spent on the homework, nor how much help they received from friends or teaching assistants, nor what kinds of mistakes they were making when completing the homework. If a struggling student works hard enough and gets enough help, his or her homework can be nearly as good as the student who is doing well. TeachBack and Spinoza on the other hand capture much more information and can give daily snapshots of student understanding at a much deeper level of detail.

5.6. Conclusions

Our findings demonstrate the critical part played by the pedagogy in a classroom. As we pursued the same interactive pedagogy with and without computers, we were able to support two main results. First, the class can be taught using essentially the same thought and learning processes with or without computers. Secondly, contrary to most beliefs, allowing computer-use in class on its own causes distractions and poor learning outcomes for only a small subset of students. Those students could perhaps be given accommodations to not use their computers in class. Finally, this study therefore demonstrates that the use of computers in this class does not affect learning outcomes in any significant way.

The key factor in student learning is the pedagogy itself. The highly interactive pedagogy we used encouraged students to maintain high levels of interaction, engagement and motivation with the material whether they used computers or not. We know from previous studies that active learning in flipped classes is a more effective pedagogy than straight PowerPoint lectures [**8**] and the results from this paper suggest that this pedagogy can be delivered either with or without a computer.

The various avenues of interaction offered by tools like TeachBack and Spinoza offer increased participation and involvement rates. But that is not all, like most computer-mediated communication tools, TeachBack and Spinoza allow content and conversations to be stored and accessed at later times. Both of these tools closely record students' participation and performances across the various interactive tools. For instance, in TeachBack, these statistics are both aggregated and viewable at the activity, class meeting and course levels, enabling close monitoring of students' progress in the class. Moreover, this information is individually provided for each student. Non-summative assessment data of this nature can be useful in early detection of students in trouble as well as provide new avenues to learn best how to carter to students' learning needs.

In addition, participants do not have to be in the same physical locations, and users can engage in multiple conversations at once. In a way, these tools liberate learning and teaching from constraints of time and distance [**75**], where barriers such as distance, disabilities, shyness and cultural difficulties are overcome [**126**]. Our proposed computer-mediated pedagogy features various interactive and engaging activities that do not give students the opportunities to get distracted. However, as we have discovered in this study, there are a few students who are ill equipped to handle computer-mediated interactions and online environments. Our results suggest that it might be worthwhile to offer two versions of the CS1 class, one which is fully computer-mediated providing the instructor with high quality and timely information about student performance, and one that is not computer-mediated to accommodate those students who are prone to distraction when given access to a computer in class. Another alternative approach is to teach a hybrid class, where computers are only allowed during certain in-class activities and are banned at other times, or to give accommodations to students who are highly distractible and have them work without computers in class.

CHAPTER 6

Experiment 2: Early Detection of At-Risk Students

6.1. Motivation

In this experiment, we provide evidence that analysis of student interaction with web-based pedagogical tools in a large partly-flipped CS1 class can be used for early detection of at-risk students. These tools enabled a wide variety of active learning pedagogies and provided a deep, real-time, and nuanced view of student performance in the class. We discovered, while analyzing the data after the course was complete, that there were several interesting correlations between the level of use of these tools and various measures of student performance.

TeachBack (see Chapter 3) and Spinoza [**43**] were developed to facilitate interactive classroom activities such as formative assessments, receiving cognitive and affective feedback from students, interactive coding and compiling of programming challenges, and facilitating collaborative learning and assessment tasks. In this analysis report, we will show that the use these tools provides an additional unforeseen advantage - the analysis of student interaction with these tools provides a deep insight into student learning styles and this knowledge can help us predict which students may be in danger of doing poorly in the class.

6.2. Experimental Design

6.2.1. The Partially Flipped Classroom. During the fall semester in 2014, we taught an Introduction to Programming in Java class which had a total of 284 students, divided into two roughly equal sections, each meeting three times a week for 50 minutes. In this class, we employed an active learning style of pedagogy. The two sections of the class were taught using the same interactive pedagogy where each class meeting started with pre-class assigned readings with a mandatory reading reflection or short quiz due before class. The 50 minute class sessions were composed of 3 – 5 interactive activities including short lectures, written feedback from students, problem solving challenges, and coding challenges, often using a Think/Pair/Share methodology [**83**]. TeachBack and Spinoza were used in almost every class to facilitate communication between the instructor and the students.

6.2.2. Data Collection. All student interaction with TeachBack and Spinoza was recorded in a database with timestamps and userIDs. This data was then combined with traditional grade book information including scores on homeworks, quizzes and responses to four surveys administered throughout the semester. For this study, we then looked for features of student interaction that were strongly correlated with overall performance in the course. Rather than use the letter grade, we measure overall course performance with an uncurved final course score that combines quiz scores, homework scores and class participation. The cutoff for a B- for the final course grade was an uncurved final score of 70%. We refer to the uncurved final course score as the course grade in this section. A score under 60% indicates a poor performance in the course and our goal was to seek student interaction data from the TeachBack and Spinoza tools that would help us identify students in danger of doing poorly in the class as early as possible so that some intervention could be taken.

6.3. Results

6.3.1. Engagement and Performance. The simplest measure of engagement in a course is whether students are attending class and participating. We calculated total engagement by adding three quantities: (1) the number of questions attempted in Questions, whether they were answered correctly or incorrectly, (2) the total number of feedback responses, and (3) the total number of forum posts and comments.

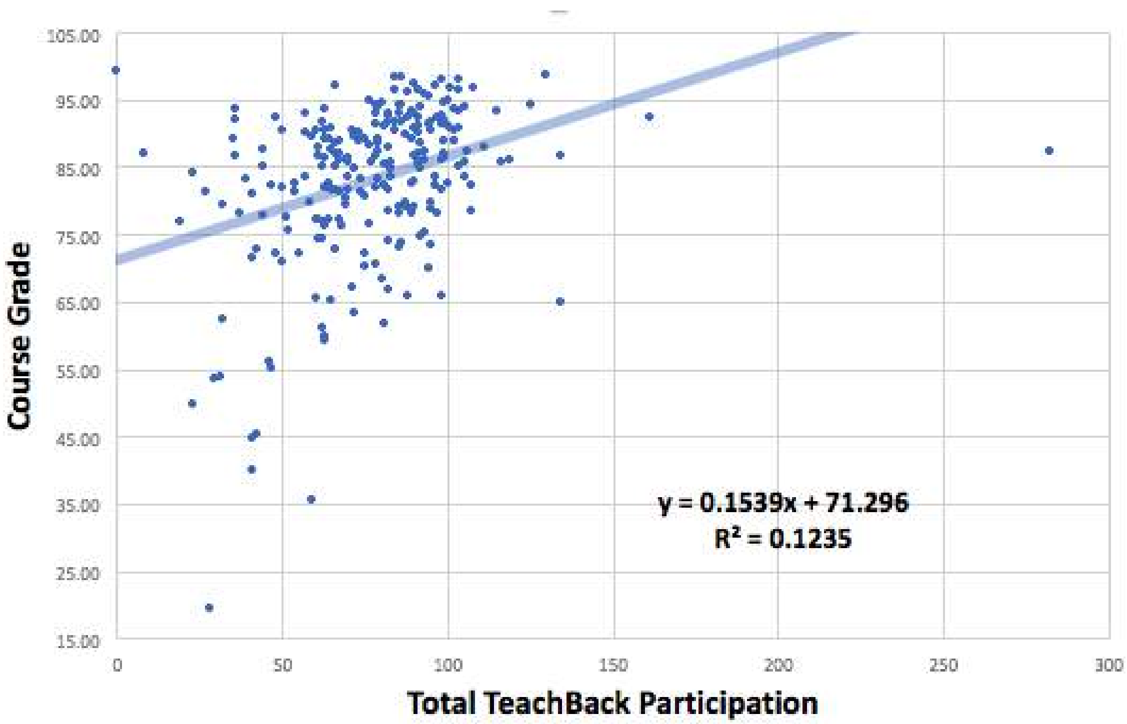

FIGURE 6.1. Engagement: Course grade vs. Overall participation in TeachBack

Our first result is that total participation in TeachBack is positively correlated with final course score. The data is plotted in Figure 6.1 together with a trend-line. This measure could be helpful in predicting which students might be at-risk, since we can observe that all students who performed poorly in the class (with a course score under 60%) had a participation level under 60. Students were required to answer at least half of all Questions problems and their participation accounted for 10% of their final course grade so we would expect at least a little correlation between participation and course grade.

6.3.2. Learning Speed and Performance. The next measure we studied was how quickly students could assimilate new knowledge. This initial understanding level was measured as the sum of all Questions points. The Questions data provides a formative assessment which is a rough measure of the initial level of student understanding. Since problems in Questions are graded (typically on a 0–2 scale), the sum of all of students' assessment points over the semester gives a measure that combines participation and formative assessment of initial levels of understanding. The correctness of their Questions responses was not part of their final course grade; it was entirely formative with no summative component. Since problems in Questions were often asked as soon as a new concept was introduced, one would not expect a perfect correspondence between this formative measure and the summative measures used to calculate the overall course grade. We found however, that total points, as a proxy for learning speed, is strongly correlated with overall performance in the course. Figure 6.2 shows a plot of total Questions points versus the uncurved final course score. This plot shows that performance in formative assessments is even more strongly correlated to the final course score than engagement and this indicates that fast learners tend to do better in the course. It could be that students who score high on total Questions points are those students who are being challenged in their Zone of Proximal Development [**164**].

From a prediction perspective, since there were 120 total possible Questions points, if we let F be the proportion (in the range 0.0-1.0) of total points earned, then the

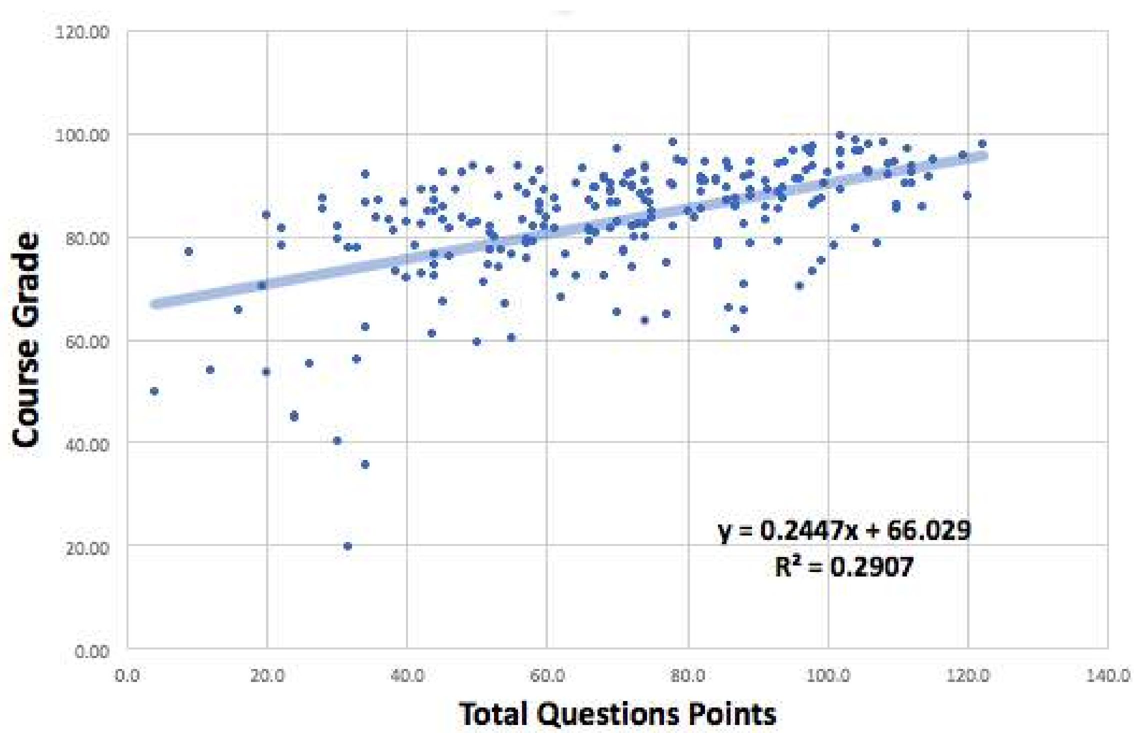

FIGURE 6.2. Learning Speed: Course grade vs. Total formative assessment points in TeachBack

best fit line for predicting the course grade G based on initial understanding F is

$$G = 66 + 30 * F$$

and the data fits this line fairly well (with $R^2 = 0.29$). Moreover, we see that all students who did poorly in the class (under 60% in the uncurved course grade) had a level of TeachBack participation with $F < 0.3$ in Figure 6.2. These findings demonstrate that student performance and initial understanding in class can roughly predict their overall learning and grades.

6.3.3. Confidence and Performance. A third measure that one might think could predict performance is student confidence which we measure as the student self-assessed level of confusion when first encountering new topics. One might think that students who feel they are often confused in class would be in danger of doing poorly. To get a measure of self-assessed initial confusion, we examined all student feedback responses and summed them with each confused response counting as -1 and each bored or engaged response counting as 1. Figure 6.3 contains a scatter plot of

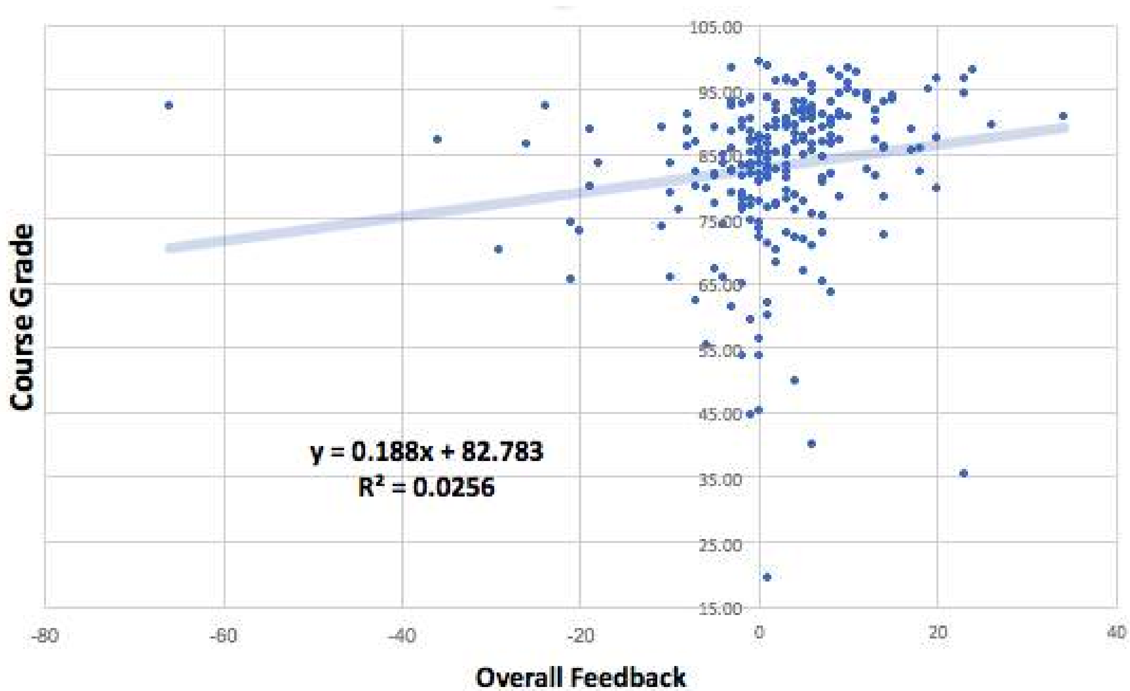

FIGURE 6.3. Confidence: Course grade vs. Overall self-assessed feedback in TeachBack

confidence vs course grade and we can see that the R^2 level is very low at 0.02 which indicates that the correlation is very weak. Indeed, we can see that the students with very low confidence levels are not the students who did poorly in the class. At-risk students seem to be over-confident in their level of understanding [**87**]. The students who were most confused, generally did quite well in the course and we suspect this might be due to imposter syndrome where students who do not fit the cultural stereotype underestimate their ability [**36**]. Moreover, the assessment of confusion can be seen as a measure of cognitive engagement and self-awareness. As seen, most confused students are the ones who are involved in learning and in constructing new knowledge by confronting what they know, do not know and personal misunderstandings.

6.4. Conclusions

This study has shown that there are several strong correlations between the way students interact with TeachBack and their overall performance in the course. Moreover, selected measurements provide proxies for student features that are difficult to measure through standard summative assessment tools, e.g. engagement, learning speed, and confidence. We suspect that by monitoring student interactions with these tools, one might be able to detect the students who are in danger of doing poorly as soon as they start to have trouble in the course, and well before the standard summative measures such as quiz and homework scores indicate a problem.

In a class where TeachBack is used everyday this would provide a sensitive, early-warning system to help direct TA resources towards students that would benefit from the extra support. Moreover, the monitoring of learning and assessment of risk can be done throughout the duration of a course. Risk assessments based on one to several classes can provide opportunities for immediate interventions at a more fine-grained level.

CHAPTER 7

Experiment 3: A synchronous class with remote and face-to-face students

7.1. Motivation

The traditional instructional model with an instructor and students meeting at the same time and same place in a classroom is still the dominant style of education, even though many institutions and researchers are actively exploring other models, such as distance education. In our view, the traditional model of instruction still provides the best teaching and learning environment due to the unbounded avenues of communication and interaction afforded by physical presence. However, the changing learning styles and needs of students and demanding schedules have forced the growth and adoption of remote instruction and distance education. The advancement of communication and information technology, as well, has contributed to the feasibility of distance education and enhancement of traditional classrooms by offering synchronous and asynchronous facilities for communication, instruction, and distribution of learning materials.

In this experiment we set out to study the effects and feasibility of adding two features onto a flipped course pedagogy: (1) real-time video streaming of the class meetings and (2) web-based class-wide interactions. In the first part of the semester, students were required to be physically present and to use an audience response system to interact in class. This created a baseline for the course as a traditional flipped classroom. About halfway through the semester, students were given the option to either attend classes remotely or face-to-face, on a class-by-class basis.

The experiment was designed to facilitate a synchronous experience for the remote students similar to being in the classroom in person. Remote students could follow the in-class activities in real-time by watching the live-stream of the ongoing class meeting in the classroom. The live video would guide the students on the content being covered, instructional activities being carried out and the various interactions happening between students and the instructor. We used the Echo360 Lecture Capture System to both archive a recording of the class for later on-demand viewing by students and to live-stream the class meeting. This system is available in half a dozen large lecture halls at the University.

We used the audience response system TeachBack (see Chapter 3) to facilitate the interactions and other instructional activities between the instructor and the students, both face-to-face and remote. TeachBack is a web-based application that provides tools for conducting individual and collaborative (formative) assessments, a way for instructors to collect and assess cognitive and affective feedback from students, and a way to more efficiently attend to student questions using the forum feature. The Echo-360 live-stream of the class provided a real-time access to the content and instructions happening in the classroom and TeachBack provided an interaction platform that allowed both remote and face-to-face students to participate equally well in all classroom activities.

As far as the existing practice of recording lectures for students' use, we do know that students are strongly in favor of having class meeting recordings and that they find them useful to their learning in various ways [**11**, **118**]. Some of the uses include catching up on missed classes, reviewing complex concepts, revision for exams and assessments, as well as a general learning tool [**84**, **99**, **102**]. Furthermore, various research studies have shown that access to lecture recordings has little to no effect on class attendance [**84**,**99**,**163**]. Some studies reported lower class attendance when students had access to lecture recordings [**46**, **69**, **82**], however, the differences disappeared when demographic differences among students where considered.

Similar to the mixed literature results on effects on attendance, access to lecture recordings has been shown to have little to no effect on student learning outcomes [**58**, **91**], whereas studies by Phillips et al [**122**] and Traphagan et al [**159**] suggest that frequent access to recorded lectures leads to more positive results and learning behaviors. Moreover, in a study where access to lecture recordings was reported to result in lower class attendance, the effects on attendance had a negligible effect on student attainment [**159**]. This review on the impact of lecture recordings sets a perspective on what to expect from the addition of the real-time synchronous and interactive lecture streaming as a remote alternative to face-to-face attendance.

7.2. Background and Related Work

Online instruction and learning are rapidly becoming alternatives and supplements to the traditional classroom instruction [**9**]. Students enroll in online courses and rely on online learning resources for a variety of reasons, including their desire for more flexibility. Asynchronous online environments offer anywhere and anytime learning; these include MOOCS, virtual classrooms, and learning from pre-recorded lecture videos. On the other hand, synchronous online environments offer anywhere learning but require students to tune in at specific class times. Studies show that students succeed in online learning environments when there are significant interactions and active participation [**42**, **85**] which also results in greater student satisfaction and engagement [**27**,**97**]. In this work we attempt to create an online learning environment with real-time interactions and active participation for remote students that is as close as possible to being physically present in the classroom.

Implementations of the real-time synchronous option for remote students similar to our approach are much less studied. Some researchers [**73**, **90**, **101**, **131**, **160**] have proposed or implemented lecture-streaming with real-time interaction among students and the instructor. However, most of the interactions are limited to the use of forums, text messaging and media sharing, which are still inferior to the interactions possible in a face-to-face classroom. Our implementation offers a more fully interactive and synchronous classroom for remote students though the use of TeachBack in enabling the important social, teaching and cognitive elements of presence [**68**, **151**] for online environments.

Most of the interactions and instructional activities in the class were mediated using TeachBack, and so there was very little difference in the classroom experience between remote and face-to-face students. This type of real-time hybrid class structure has the potential of significantly affecting learning and teaching experiences, learning outcomes, class attendance behaviors, and potentially reducing the need for large lecture halls. There is a clear need to study how the learning environment (physical classroom, virtual class, hybrid class, etc.) influences learning [**66**, **67**, **70**, **145**] and to understand why attendance in large lectures is decreasing and how to stop this trend [**44**, **86**, **102**].

7.3. Experimental Design

7.3.1. The Class. The experiment was carried out during a flipped advanced level Computer Graphics course with enrollment open to undergraduate and graduate students. The course exposed students, through a hands-on introduction, to the science and practice of rendering three dimensional (3D) images using both resource intensive ray-tracing methods and real-time techniques using the GPU. The course was also intensive in mathematics and programming content and exercises. Students programmed the implementations of ray-tracing algorithms in Java and used it to create their own 3D scenes. They also created interactive web-based applications handling 3D graphics using WebGL and ThreeJS and they learned how to code programmable shaders using GLSL. The experiment was carried out during the fall semester of 2015 where 41 students completed the course. The course consisted of 26 highly interactive class meetings.

It is important to clarify that the particular selections of the course, the semester, and the corresponding student group were independent of the experiment design, and only coincided with the availability of the instructor and the resources to conduct the experiment. In our opinion, the experiment could effectively be performed during a different semester, course and group of students and we would expect to find similar results.

Typical class meetings started with pre-assigned readings in Nota Bene [**60**] where, as a graded homework, students were required to collaboratively add five comments or questions or to answer questions already asked by others in the readings. Nota Bene is a web-based annotation tool that allows users to annotate arbitrary PDF files online in a collaborative fashion. The tool is a product of active research and multiple development cycles from Massachusetts Institute of Technology (MIT) and it has been used in real classes since 2009. The instructor used students' responses in the readings as guidance for the short lecture demonstrations during class. The short lectures included presentations from PowerPoint slides, websites and code samples. The goal was to make sure that students were familiar with all concepts from the readings and any raised misunderstandings were resolved before engaging students in active activities with the materials.

Since the class followed a flipped pedagogy, most of the class time was devoted to engaging students in hands-on tasks that re-enforced their understanding and mastery of the concepts covered. The course was programming intensive and thus a popular activity was coding demonstrations on implementing specific concepts where the instructor would write code snippets with students doing the same in their personal computers. Oftentimes, students would be asked to finish components of the code, and the instructor would select students to share and demonstrate their approaches. Another popular activity was formative assessment using the Questions and GroupWork tools offered in TeachBack. The Questions tool was devoted to simple concept or programming question where students would answer quickly and individually. GroupWork was used for more complex concepts and longer programming challenges where more time would be allocated for students to discuss their responses in small groups. During these formative assessments activities, the students who are remote and those in class would both participate equally. In GroupWork, student groups are generated randomly based on recorded attendance for the day and therefore the three students making a group could include students who are remote as well as those physically in class.

The Feedback tool in TeachBack was used to more quickly get an assessment of how students were doing during class with respect to the concepts just covered. For example, at various points during class the instructor would ask students to assess how they were doing and provide feedback. The instructor would then assess the percentage of students who reported being ‘Bored’, ‘Confused’, or ‘Engaged’ together with any accompanying comments. This was particularly important to the learning and teaching for the remote students. A particular emphasis was also given to the use and monitoring of the Forum where students would ask and answer questions about the concepts being covered in class. The use of the Forum was particularly important for the remote students as this was the only way to get personal attention from other students and the instructor.

7.3.2. Echo360 Lecture Capture System. Echo360 was set to automatically record each of the class meetings, from start to end. The recordings are accessible through a web-based component of Echo360 where students and instructors can browse and playback recordings of lecture meetings after logging in. This practice has been a success and an important component of many courses at the university. Students appreciate the value and usefulness of the recordings, both as a supplement to face-to-face class attendance as well as an occasional replacement for when classes are missed.

Out of the 26 class meetings in the semester, the last 16 were also live streamed in real-time. Starting at the 11th class meeting, students were informed about the new addition to the course and that they were freely allowed to attend the class remotely as long as they watch the live-streaming and participate fully in class activities and interactions using TeachBack.

7.3.3. Data Collection. By adding the remote attendance part-way in the semester, we hoped to be able to compare and discover various measures of the learning and teaching experiences before and after the change. Most importantly, we hoped to learn about the feasibility of the practice from the students' and instructor's perspectives and how it affected, if at all, students' performance in the course. We also expected to learn how remote attendance would affect students' motivation, face-to-face class attendance and the use of resources such as the lecture recordings. We hoped that allowing students the option of attending class remotely could increase attendance as students who were sick or traveling (e.g. with their sports teams or at interviews) could still attend.

We collected data for analysis from four sources. From TeachBack, we gathered data on student attendance and performance on formative assessments using the Questions and GroupWork tools. For each Echo360 lecture recording, the system recorded all student interactions with the recording. We therefore had access to data such as whether a recording was watched live and for how long, and the number of times and duration that a student watched a particular recording on demand. At the end of the course a survey was given to students in order to gather their perspectives and evaluation of the course style and their learning experiences, especially with respect to remote attendance and assessment of the technical tools used to implement it. For overall performance in the course, we used students' grades in a final cumulative written exam which was administered at the end of the course and covered most of the materials taught in the course. We also used the overall course grade which included students' overall participation (in TeachBack and in pre-class reading assignments), three programming projects throughout the semester, and the final written exam.

7.4. Results

7.4.1. Effects on Absenteeism. For this analysis we compared the numbers of students who attended classes (remote or face-to-face) and those who were absent (didn't show up in-person or remotely). Contrary to our expectations, allowing remote attendance did not increase students' overall class attendance, but it did not decrease it either. We had hoped for a significant reduction on the average number of students who would miss classes (not attending in-person or remotely). Considering the first 10 classes without live-streaming (Part 1), and the last 16 classes with live-streaming

(Part 2), the average number of absentees per day was approximately the same: there were 2.9 absentees during Part 1 and 2.875 absentees during Part 2. Since the class had 41 students, the average daily attendance was 93% which includes students attending remotely and those physically in the classroom. Anecdotally, attendance in classes tends to decrease toward the end of the semester [**69**], so the fact that there was no decrease in the second half, could be a positive outcome.

Students only attend class meetings if they perceive 'value' in doing so [**102**], especially in light of the availability of lecture recordings. Participation was 20% of the course grade, which may be part of the reason that attendance was so high throughout the semester. Table 7.1 shows the actual absenteeism during Parts 1 and 2 grouped by the number of classes missed by students. We also analyzed the usage

Missed Classes	**Students (Part 1)**	**Students (Part 2)**
0	23	17
1	11	12
2	5	8
3	0	2
4	2	0
6	0	2

TABLE 7.1. Absenteeism, grouped by number of classes missed by students during Parts 1 and 2 of the course.

of lecture recordings by the students who were absent during those recorded classes. Over the duration of the semester there were a total of 72 class absences. However, 24 (or 33.3%) of the time these specific students would later watch the lecture recordings for the specific missed classes. This rate was approximately the same during classes before and after the introduction of live-streaming, 35.7% and 31.8% respectively.

7.4.2. Student Use of Live-streaming. Attending classes remotely was embraced by students, with 75% of the class trying at least once and 18% attending remotely for at least 8 (or 50%) of the 16 live-streamed classes. Out of the 16 classes, Echo360 recorded a total of 121 significant live-streaming sessions from students. This averages to about 7.56 students live-streaming per day. This implies that, on average per class day, only about 75% of the registered students were physically present in the classroom, with 18% attending remotely and the remaining 7% being absent. Table 7.2 shows remote attendance counts grouped by the number of classes live-streamed by specific students.

Classes	Students	Percentage of Class
0	10	25%
1	10	25%
2 - 5	14	32%
7 or More	7	18%

TABLE 7.2. Remote attendance, grouped by number of classes live-streamed by specific students.

Echo360 data was used to analyze students' commitment patterns when watching the live-stream of the class remotely. The duration for each class was 80 minutes and from each live-viewing session that was recorded in Echo360 we were able to calculate the percentage of the class duration that was watched live during each viewing session. A total of 121 live streaming sessions were recorded, and as seen in Figure 7.1, 84 viewing sessions (about 69%) streamed at least 50% of the total class time, and 52 sessions (about 43%) streamed at least 90% of the total class time. The behavior demonstrated in Figure 7.1 shows that most remote students committed to streaming entire class sessions, which is satisfactory since remote attendance was supposed to replace face-to-face attendance which should normally last for entire class

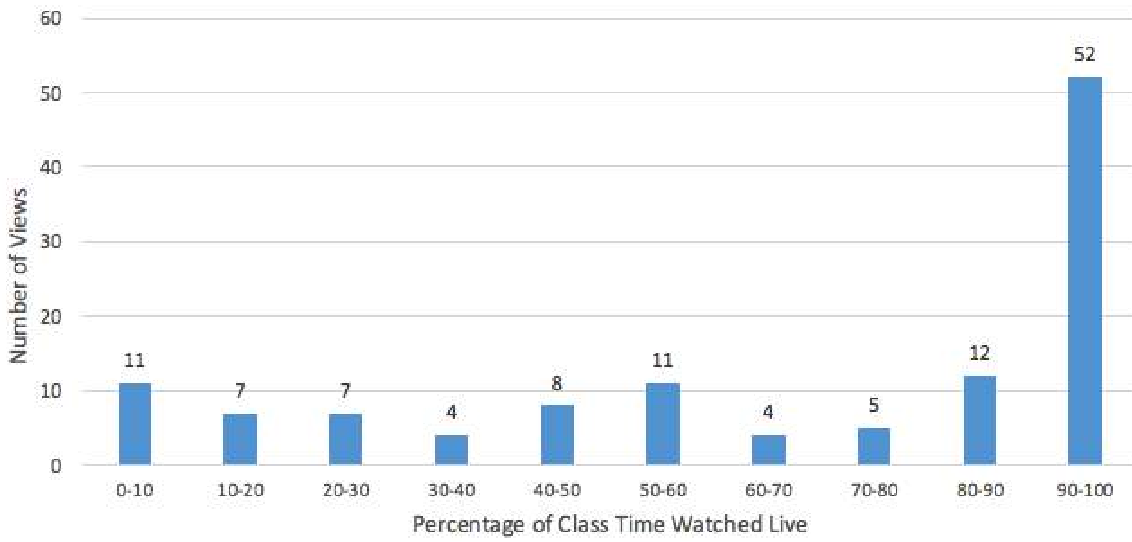

FIGURE 7.1. Distribution of percentage of class time watched in live-streaming sessions.

duration. However, there is a fair number of remote attendees who watched only part of the class sessions, perhaps due to oversleeping or having to leave early for another commitment.

7.4.3. Effects on Learning Outcomes. We looked for correlations between remote attendance and each of final exam and overall course grades in order to study whether remote attendance had any effect on the learning outcomes of the students, especially for the students who heavily used the option. The final exam was a cumulative exam that covered most of concepts taught in the semester, it represents the best estimate on students' mastery of concepts. The overall course grade included scores from final exam, class participation (engagement in class activities on TeachBack and in pre-class reading assignments), programming assignments, and three major programming and design projects. Course grade represents a more rounded mastery of material and excellence in the course.

Linear regression was used to study correlations between levels of remote attendance and each of final exam grade and overall course grade. As seen in Table 7.3, the regression analysis shows no statistically significant correlations with concepts mastery or overall performance in the course: statistical significance was measured at ($p < 0.05$) and the final exam and course grades have p-values of 0.459 and 0.343 respectively. There is, however, a very weak negative effect (as demonstrated by the negative coefficient values), but the R^2 values of 0.0145 and 0.0236 imply the regression models are indeed very poor fits for the data. The hint of negative effect could imply that attending classes in-person is slightly superior to attending remotely.

Item	R^2	**P-Value**	**Coefficient**
Course Grade	0.0145	0.459	-0.356
Final Exam	0.0236	0.343	-0.749

TABLE 7.3. Linear Regression Analysis: Course Grade vs. Remote Attendance and Final Exam Grade vs. Remote Attendance.

Another regression analysis showed the negative impact that absenteeism has on course performance. This is demonstrated in Table 7.4. There is a strong negative correlation ($R^2 = 0.3090$) between absenteeism and course grade, which is statistically significant with a p-value of 0.0002. It makes sense for absenteeism to have a more significant impact on course performance when a course is flipped with high levels of interactive learning activities. Even though absenteeism having a negative impact on course performance is an obvious result, the fact that there is no such negative impact for students who live-streamed demonstrates that remote attendance is an adequate substitute for physical presence in the classroom.

We performed further analysis by isolating the participation grade and it's role in the course grade in order to get a better comparison between remote attendance

Item	R^2	P-Value	Coefficient
Course Grade	0.309	0.0002	-2.995
Final Exam	0.0581	0.1338	-2.135

TABLE 7.4. Linear Regression Analysis: Course Grade vs. Absenteeism and Final Exam Grade vs. Absenteeism.

and absenteeism with respect to their impacts on the learning outcomes. As mentioned earlier, class participation (measured as overall engagement in class activities on TeachBack and in pre-class assignments), counted as 20% of the course grade. And thus you would expect absenteeism to have a significant impact on course grade due to the missed class participation points from the days missed. From each student's course grade, we subtracted the student's earned participation grade, resulting in a modified course grade that only reflects mastery of concepts and skills. The regression analysis results reported in Table 7.5, show the relationship between the modified course grade and each of remote attendance and absenteeism. The comparison shows

Item	R^2	P-Value	Coefficient
Remote Attendance	0.011	0.519	-0.261
Absenteeism	0.211	0.00283	-2.081

TABLE 7.5. Linear Regression Analysis: Modified Course Grade vs. Absenteeism and Remote Attendance.

a statistically ($p = 0.00283$) strong negative correlation between absenteeism and learning outcomes as measured with the modified course grade. According to the regression model, a single absenteeism can result in missing approximately 2 out of 80 grade points. Comparing the modified course grade and remote attendance, on the other hand, shows no statistical correlation (p-value of 0.519 with a very low

R^2 value). This result provides further evidence to suggest that attending classes remotely is comparable to attending them in-person.

Next, we looked at the impact of remote class attendance on class participation. Table 7.6 shows linear regression results comparing the relationship between participation grades with remote attendance, as well as with absenteeism. With a p-value of 0.441 the results show no statistical correlation between the participation grade and attending classes remotely. In other words, based on how this class was implemented, attending classes remotely did not result in poor class participation. And as expected, there is a statistically strong negative correlation between absenteeism and the class participation outcomes. Indeed, according to the regression model, missing one class can result in missing on average about 4.5 out of 100 class participation points.

Item	R^2	P-Value	Coefficient
Remote Attendance	0.0156	0.441	-0.476
Absenteeism	0.435	0.0000036	-4.568

Table 7.6. Linear Regression Analysis: Class Participation Grade vs. Absenteeism and Remote Attendance.

7.4.4. Use of Lecture Recordings. Lecture recordings were a popular learning resource to students. Indeed, the use of lecture recording was very significant to both students who attended classes face-to-face and those who attended remotely. A total of 273 viewing sessions were recorded in Echo360, accounting for live viewings, views on demand, or both. As seen in Figure 7.2, 162 sessions (about 59%) were views on demand where students watched lecture recordings at their own times for catching up on missed classes, personal studying or reviewing for exams and quizzes. 94 sessions (35%) were live-viewings only, when students attended class remotely. An interesting observation is from the remaining 17 sessions (6%) when students first watched the

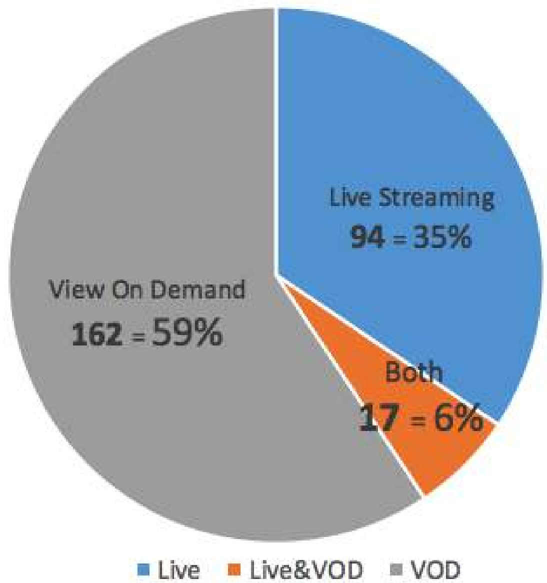

FIGURE 7.2. Distribution between live streaming and view-on-demand in Echo360.

recording live and continued on to watch on demand. This could count as the times when remote students were late and missed an earlier part of a class meeting and they had to watch an earlier section of the class meeting from the completed recording. If this is the case, the percentages of class time that remote students viewed live, as discussed in previous subsection, would be higher, albeit while still missing the class time participation in interactions and activities.

Figure 7.3, shows the distribution, by the number of recording minutes watched, of students viewing patterns of lecture recordings. As seen in the figure, most of the time students would watch short sections of lecture recordings. Echo360 allows students to start viewing at any point in the class meeting and provides the instructor with a heat map showing which parts were most watched. These tended to be the short-lecture sections of the class. Interestingly, there are some students who would watch more than 80 minutes (a recording is 80 minutes long, this means re-watching sections multiple times) of a recording in a single session.

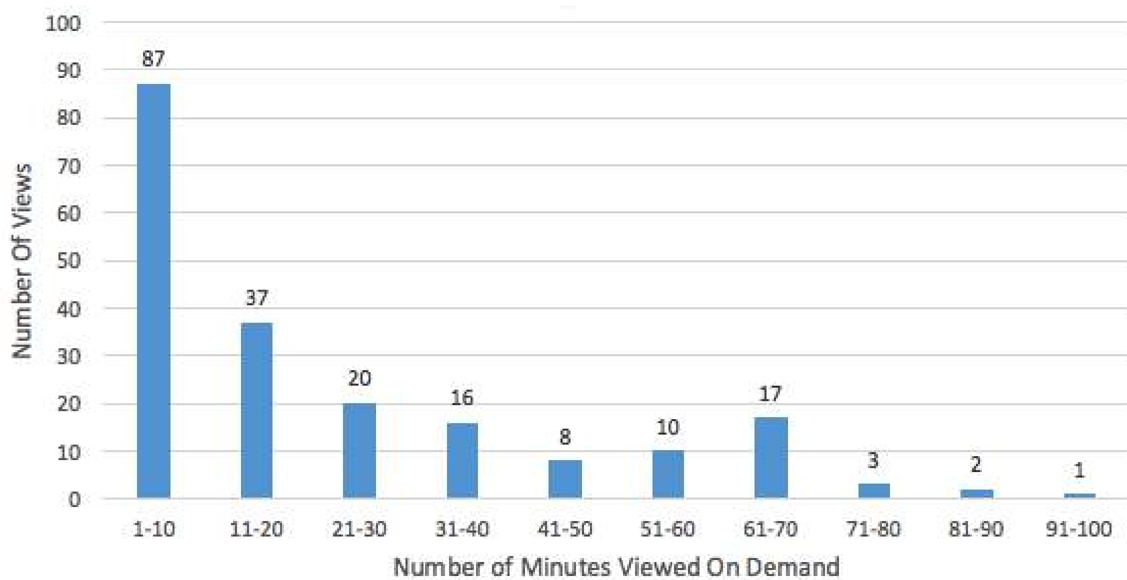

FIGURE 7.3. Distribution of class recording minutes watched during view-on-demand sessions. Each class recording is 80 minutes long.

7.4.5. Students' Opinions. This subsection summarizes students' survey responses on remote attendance using synchronous class streaming with participatory interactions facilitated with TeachBack. Most of the students liked the approach; with 51.5% of the class feeling that most or all of the other classes at university should implement this approach as an option for students. Moreover, 36.4% of the class felt that only some classes should implement it, and 3% felt that the approach was not helpful. When asked to compare the effectiveness of the remote attendance option to physically attending classes, 50% of the class said that the remote attendance was not as effective or less effective. However, 30.6% of the class reported that the option was just as effective or more effective. 19.4% of the classes responded that they did not try remote attendance and that they didn't have an opinion on the comparison.

Students were particularly satisfied with how we implemented remote attendance. Research literature shows that students prefer face-to-face classroom learning over the asynchronous non-interactive online modes, and our work attempted to recreate the classroom experience though synchronous interactions and real-time participation in activities. According to findings by Farrington et al [**50**], most students prefer the classroom as the primary learning space with online modes viewed primarily as a supplement to, or occasional replacement for, the face-to-face experience. In our experiment, however, 18% of our students chose the online mode over face-to-face more than half of the time.

On an effectiveness scale of 1–5 with 1 being 'not effective' and 5 being 'very effective', 87.2% of the students rated the effectiveness of TeachBack at 3 or above. On the effectiveness in using Echo360 to live-stream class meetings in the implementation, 79.4% of the class rated the system at 3 or above. These ratings show that students believed that the technical and pedagogical aspects of the implementation were practical and feasible in creating an effective alternative to attending classes face-to-face.

Looking at students' free response comments, the most popular advantage of the remote attendance option is convenience. Students liked the fact that they could attend the class from anywhere, including from home, during late commutes to school, and from out of town when they had to be away for various personal reasons. Attending class remotely was described by students as a valuable alternative when they have to miss class during sports trips or personal reasons like sickness, interviews or weddings.

Students were not fully satisfied with the video and audio quality of the lecture streaming, especially at times when it was hard to see projected screen texts and programming demonstrations. They thought that there was an inconvenient delay in getting responses to questions and group conversations when working online compared to face-to-face conversations in the classroom. Additionally, students thought that attending class remotely was prone to inferior concentration compared to being in class in-person. They also reported that video streaming depended on reliable internet connection; however, no one reported having an unreliable internet connection. These are all technical issues that can mostly be solved by better tuning on the Echo360 live-streaming parameters and reflect our relative inexperience in live-streaming a class for the first time.

In terms of suggestions for improvements of the implementation, some students suggested that remote attendance is good when it is used as a backup option when it's absolutely necessary for students to physically miss classes, like when sick and during bad weather. Remote attendance would be less well-suited to seminar-style classes that are heavy on discussions. In GroupWork, many students felt that groups should be made up of students who are entirely remote or entirely in the classroom. Some students thought that remote attendance would be bad when students abuse it or get encouraged to miss classes, have poor class participation and distractions - factors that would put the students at the risk of poor performance, but we didn't see any of these phenomena.

7.5. Conclusions

In this chapter, we described an experiment exploring the effects of allowing remote attendance through a combination of live streaming and mandatory web-based individual and group activities during the class. The students were given the option to attend remotely in the second half of the semester. Our main research question was to determine the effects of allowing remote attendance on absenteeism, course grade, and in class participation.

We therefore collected data on which students attended remotely each day, how much each student used the web-based interaction tool each day, and the final course grades of the students, with and without the class participation component. We discovered that the remote attendance option was very popular. Three quarters of the students attended class remotely at least once, and 18% attended remotely at least half the time it was an option. We also discovered that remote attendance had very little impact on course grade, on absenteeism, or on class participation. Finally, the results of a survey at the end of the semester indicate that over half of the students felt this option should be available for most or all of their classes, and only 3% felt that it was not a helpful option.

This experiment has demonstrated that it is technologically feasible and pedagogically justifiable to add an alternative to class attendance using live lecture streaming with a web application facilitating class interaction and activities. Data analysis has shown that this approach does not negatively impact students' learning outcomes or increase class absenteeism. Indeed, the combination of live lecture streaming, lecture recording and the use of TeachBack has the potential to improve class attendance to nearly 100% with class involvement and participation in a synchronous or timely fashion. This approach provides more students with the convenience and flexibility of attending classes from anywhere while guaranteeing synchronous engagement with the rest of the class. It also allows students who would otherwise miss part or entire classes to still attend and participate remotely.

It is possible that the positive results obtained from this study may not generalize to other classes, other instructors, other educational levels, or other disciplines. This particular class made heavy use of active learning and it is not clear if remote attendance would work as well in a traditional lecture style class as students could easily become distracted if they are not constantly required to interact with each other and respond to instructor's questions. The students in the class were highly motivated graduate students and upper-level undergraduates majoring in Computer Science at an elite university. It is possible that students in other majors or at less competitive universities might be less engaged in remote settings.

In the future, we plan on encouraging other faculty (at our institution and others) to try this approach, including the experimental design where the option is only available for the second half of the class. This should provide more data on the effectiveness of allowing remote attendance with mandatory web-based interaction. We expect that the remote-attendance option will be less effective in traditional lecture course, as compared to active-learning flipped-classrooms.

We also plan on making this option available to students for the entire semester and working on improving the technical aspects (sound and video quality) as well as adding more options to the mandatory web-based interaction tool. We expect that with these improvements, the remote-attendance option will be even more popular with students. We also suspect that many students will always prefer the in-class option because it provides the opportunity for face-to-face interaction with the instructor and their fellow students, and these interactions can be more socially rewarding.

CHAPTER 8

Experiment 4: Learning During Collaborative Assessment Activities

8.1. Motivation

This chapter presents a case study and the results of an experiment where students collaborated in small groups during formative assessment activities. The experiment studies whether collaboration during formative assessment activities has better immediate and long term learning outcomes of the participants compared to when they participate individually. In assessing the immediate effects on the learning outcomes, we compared students' overall performance when responding individually to formative assessment problems and their corresponding performance when working in collaborative groups. In assessing the impact of collaboration on overall learning outcomes on the course, we looked for evidence of correlations between performance in these assessment activities and final performance in the course. We also surveyed students at the end of the course to get their personal perspectives on their learning and experiences during group-based formative assessment. All the assessment activities were done using the GroupWork tool in TeachBack and the majority of the activities were formative assessments during class time. A few of the activities were assigned and completed outside of the class time as warmup exercises in preparation for quizzes or exams.

Over the last three years of empirical studies using TeachBack, the Questions feature has been used to facilitate formative assessment activities in the classroom where students respond individually. For example, our own prior studies (see Chapter 6) have shown statistically significant correlation between levels of participation and performance in in-class formative assessments and mastery of the material and concepts in a course. Until the fall semester of 2015, in-class formative assessment was done using the Questions tool in TeachBack where students answered questions individually followed by class discussions. In this experiment we wanted to see whether adding collaboration to these formative assessment activities would lead to better learning outcomes and experiences for the students.

The outcomes of this experiment would also inform us on the feasibility and practicality of using web based interactive tools like GroupWork in facilitating collaborative formative assessment activities in the classroom settings. The findings would provide the evidence necessary in proposing whether pedagogy design should include collaborative formative assessment activities and in a possible way that is practical and feasible.

The proposed implementation of collaborative formative assessment using GroupWork is based on various instructional activities and learning theories that empirical research and educational literature have shown to support improved learning and teaching outcomes. First, is the opportunity for students to work collaboratively and cooperatively during learning. Presently, competitive and individualistic are the more dominant modes of student-student interactions during learning [**129**]. However, compared to competition and individualism, cooperation and collaboration have shown to result in students achieving better learning outcomes, more effectiveness interpersonally and students develop positive attitudes towards learning, each other, instructors and subject areas. Second, having activities that allow a group of students to work together stresses the idea of co-construction of knowledge and mutual engagement of participants which establishes a constructivist classroom. A constructivist classroom focuses on student-centered discourse, over the typical teacher centered classroom, where students drive discussions and the teacher serves as a guide on the side [**119**]. Third, our extended use of formative assessment activities is a form of problem-based-learning based on cognitive apprenticeship [**32**] where students learn in the context of solving complex and meaningful problems. The role of instructor is to create appropriate questions, and then guide students on the learning process by encouraging students to think deeply through discussions in evaluating responses to the problems. And finally, GroupWork is closely modeled on the Think-Pair-Share [**83**, **94**, **95**, **105**] methodology of classroom-based collaborative active learning. In a typical Think-Pair-Share activity, students work on a posed problem by the instructor, first individually, then in pairs or small groups, and finally all participate in a class-wide discussion. Some of the benefits of this strategy include promoting engagement, allowing students to express their reasoning, reflect on thinking, and obtain immediate feedback on their understanding and misconceptions [**88**]. Moreover, the Think-Pair-Share technique is recommended as an instructional activity that engages learners in higher-order thinking, and as a feedback mechanism both for students and teachers [**35**].

8.2. Experimental Design

8.2.1. The Class. The experiment was carried out during the fall semester of 2015 in a flipped advanced level Computer Graphics course with enrollment open to undergraduate and graduate students. The course exposes students through a hands-on introduction to the science and practice of rendering three dimensional (3D) images using both resource intensive ray-tracing methods and real-time techniques using the

GPU. The course was also intensive in mathematics and programming content and exercises in Java and Javascript.

During that semester 41 students completed the course which consisted of 26 interactive class meetings, with each class lasting for 80 minutes and meeting twice in a week. TeachBack was used in the class to facilitate some of the instructional activities and interactions in the classroom. This includes formative assessments, students giving feedback to the instructor, the back-channel assistance forum for asking and answering questions, and taking notes by students. Echo360 was used to record all class meetings, and the recording allowed students to review missed classes and review lectures for self studying and preparation for exams and quizzes. During the last 16 class meetings, Echo360 was also used to live-stream class meetings and students were given the option to attend classes remotely, provided that they watched the live-streaming and fully participated in class activities using TeachBack.

Typical class meetings started with an assignment of pre-class readings using an online collaborative annotation tool. From the readings students were supposed to collaboratively add five new comments, questions or answers to questions already asked by others in the reading. The work completed in these readings was graded towards the participation grade in the course. Since the class followed a partially flipped pedagogy, most of the class time was devoted to engaging students in hands-on tasks that reenforced their understanding and mastery of the concepts being covered. The course was programming intensive and thus a popular activity was coding demonstrations on implementing specific concepts where the instructor would write code snippets with students doing the same in their personal computers. Oftentimes, students would be tasked with finishing the coding of a problem or concept, as formative assessments, and the instructor would select students to share and demonstrate their approaches.

Instructional activities that are of particular interest to this experiment were formative assessments using the Questions and GroupWork tools in TeachBack. The already familiar Questions tool was only used for in-class assessment during the beginning of the semester while the new GroupWork feature was being introduced to the students. Afterwards, most of the assessment activities were done using GroupWork as it was the focus of the experiment and the tool actually requires students to initially work individually (similar to using Questions) and then collaboratively in groups. When the Questions tool was used, it was devoted to assessing simple concepts or programming questions where students would answer quickly and individually. GroupWork, on the other hand, was more suited for more complex concepts and longer programming challenges where more time would be allocated for students to discuss their responses in small groups.

8.2.2. Collaborative Assessment using GroupWork. A detailed presentation of the GroupWork feature of TeachBack is reported in Section 3.6. Throughout the majority of the semester, formative assessment exercises where conducted using GroupWork. The majority of the exercises were given one-by-one in real-time during classes as a way to assess students' understanding and to inform the instructor on whether to reiterate the just covered concepts or move on to new material. The exercises also counted as an active activity during classes in getting students to actively engage with the material, gauge their own learning, reveal misconceptions, and more importantly engage in discussions and share the learning with others in the class.

A typical class meeting would be divided into alternating activities that include short lectures, presentations, whiteboard and coding demonstrations, gathering feedback from students, and answering students' questions in-person and using the Forum feature in TeachBack. In-between these activities, especially after covering new material, an instructor would give one or more formative assessment questions using the Questions or GroupWork tools. Each class meeting would include between four to eight collaborative formative assessment exercises in the form of GroupWork problems.

Whenever students login into their TeachBack accounts during a particular class meeting, the application automatically assigns the students into groups of three. These are the group assignments used in GroupWork where essentially each class meeting would have different randomly generated groups. The answering of GroupWork problems follows an adaptation of the Think-Pair-Share methodology which is built into the implementation of the GroupWork tool. During the first step, each student takes the time to think and attempt the problem individually and then submit a personal answer which cannot be changed. During the second step, group members share their personal answers and have a collaborative discussion where the team attempts to improve on their initial answers and perhaps reach an agreement on better responses. During the group discussions, the team can create one or more new answers based on the level of agreement in the group. Afterwards, each group member is required to submit a final answer by voting on any of the answers from the team. Students are not forced to agree on a single group answer, and that's why a voting system is used requiring each team member to vote/submit a final answer. This strategy is intended to encourage a sense individual accountability and autonomy as shown in cooperative learning theories. Figure 8.1 shows a students' view of a

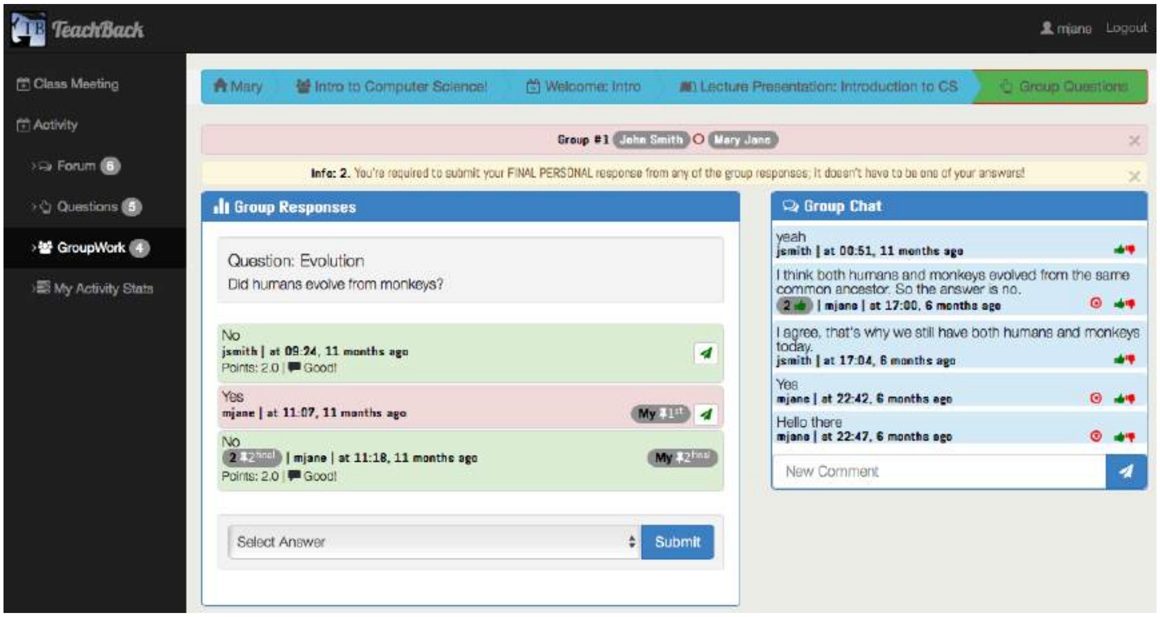

FIGURE 8.1. TeachBack: Students' question view in GroupWork.

question in GroupWork with access to answers panel and the group mini-forum. The interface is designed to include a group discussion forum area for each question so that group members can still have a discussion in the case where group members are not physically seating next to each other or some members are attending class remotely. An important design feature in GroupWork is making sure that group members attempt to answer questions individually before engaging with other group members. In order to achieve this, the interface is designed such that a group member cannot see answers or messages from the rest of the group until that student has submitted his/her individual answer.

The final stage in a GroupWork exercise is a class-wide discussion that is led by the instructor to go over responses to questions. The interface is designed such that instructors can view and grade both types of answers submitted in GroupWork. In the main GroupWork page, shown in Figure 8.2, each listed question displays the counts for individual and group responses. During class, these counts update in real-time as students respond to questions, this allows instructors to more easily gauge and manage the progress of the assessment. While reviewing student answers, an instructor has the freedom to grade the group and individual answers. This depends on the type of questions asked and answers provides as well as the size of the class. During the course reported in this experiment, an instructor would review the answers and a teaching assistant (TA) would do the actual grading at the same time.

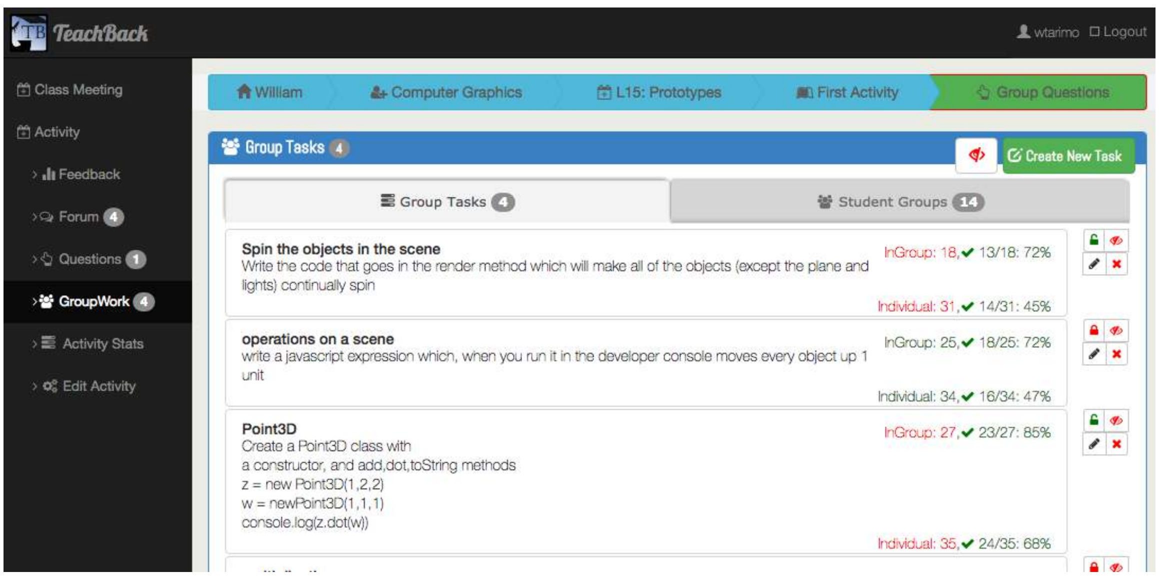

FIGURE 8.2. The main GroupWork page listing questions.

When viewing and grading a GroupWork question, the individual and in-group answers are displayed in separate tabbed sub-pages. This makes it convenient to work with a particular set of answers at a time, with the flexibility to only review and grade one set or both. Figure 8.3 shows a screenshot of a question page in GroupWork with options to display individual responses, group responses and a view to edit/update the question. In the responses view, each response has quick grading buttons where an answer can be graded as correct, partially correct, or incorrect (the green, yellow and red buttons seen in Figure 8.3). By default, correct and partially correct answers are awarded 2 and 1 points respectively. The button to the left of the green button will display a grading form for advanced grading where the grader can award a different number of points plus optional comments. After a question is graded, the percentage

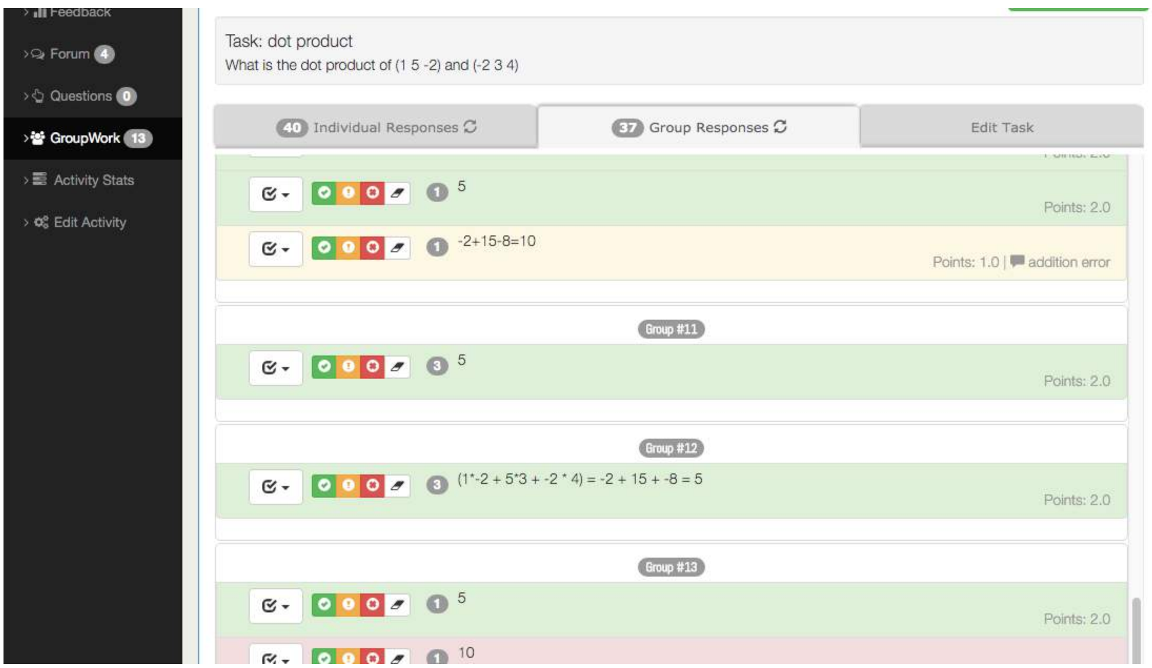

FIGURE 8.3. GroupWork: A GroupWork page showing group responses to a question.

of correct answers is also displayed for the individual and group answers. As seen in Figure 8.2, this detail provides a convenient way to see and compare the students' performance when responding individually and in groups.

8.2.3. Data Collection. In this experiment we set out to study the impact of adding small group collaboration in formative assessment activities on the learning outcomes of the students. The first analysis involved looking for the evidence to support whether students perform better in formative assessment activities when they work collaboratively in small groups compared to when they work individually.

For this, we used the grading data from GroupWork for all graded problems that were given throughout the semester. Each individual and in-group problem response was graded using the same rubric, based on the default grading behavior in GroupWork. The correctness of each answer was assessed on the scale of either being fully correct (2.0 points), partially correct (1.0 points) or fully incorrect (0 points); with partially correct answers earning some amount of partial points in the range of (0, 2).

The second analysis looked for evidence of correlation between performance in collaborative formative assessments and overall mastery of concepts covered in the course. Overall learning was measured as the course grade earned at the end of the semester which included cumulative grades from weekly or bi-weekly small programming assignments, three major programming projects, and the final written exam. The final exam was a cumulative exam which was administered at the end of the course and covered most of the materials taught in the course. 20% of the official course grade accounted for class participation (based on participation in TeachBack activities and pre-class readings), however, we omitted this participation component in the course grade that was used in our analysis as it it didn't exactly reflect mastery of concepts.

At the end of the course a survey was given to students in order to gather their perspectives and evaluation of their learning experiences from the use of collaborative formative assessment and how it was implemented using the GroupWork feature of TeachBack.

8.3. Results

8.3.1. Learning during collaborative formative assessment. For each student, we looked at the average performance when the student responded individually and when the student responded after the group collaboration stage. Average performance was calculated as the average number of points earned; that is, the sum of points earned divided by the number of questions attempted.

Figure 8.4 shows a scatter plot of average points earned by each student when responding individually versus when working in groups. The plot demonstrates that, on average, most students tend to earn more points as a result of the group collaboration. To be exact, only two students showed no average improvement while working in groups. We used a two-tailed t test to compare the mean of the average number

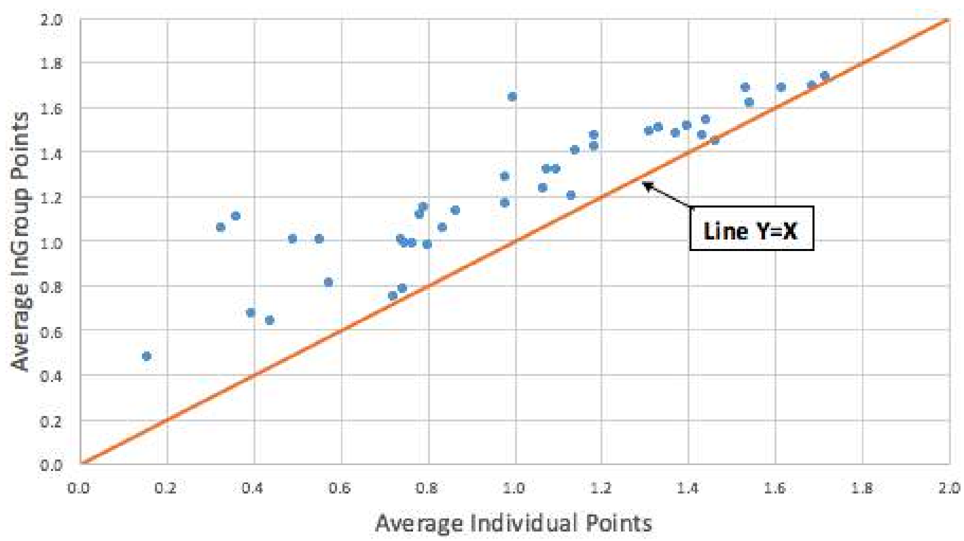

FIGURE 8.4. A scatter plot of average individual points vs average in-group points for all students.

of points from the individual and in-group populations. The class average in points when working individually is 0.995 points per question, and the average while working in collaboration is 1.223. The improvement in the mean of 0.228 is statistically significant with a P value of less than 0.0001, with a 95% confidence interval of 0.170 to 0.286. This improvement represents an effect size of 0.618, which is calculated as a Cohen's d Effect Size [**30**]. According to Cohen, an effect size of 0.618 is just above the 'medium' range. And according to a survey of academic research on instructional interventions, an effect size of 0.5 or higher would represent significant gains as it is higher than the results found in most instructional interventions [**2**, **48**].

Looking at the improvement in performance resulting from collaboration, Figure 8.5 shows that the students who tend to do poorly when working individually are the ones who take the most advantage of group collaboration. As seen in the plot, students with low average performance when working individually have the most improvement when working in groups. This is contrary to, for instance, the hypothesis that collaboration would affect all students in the same way.

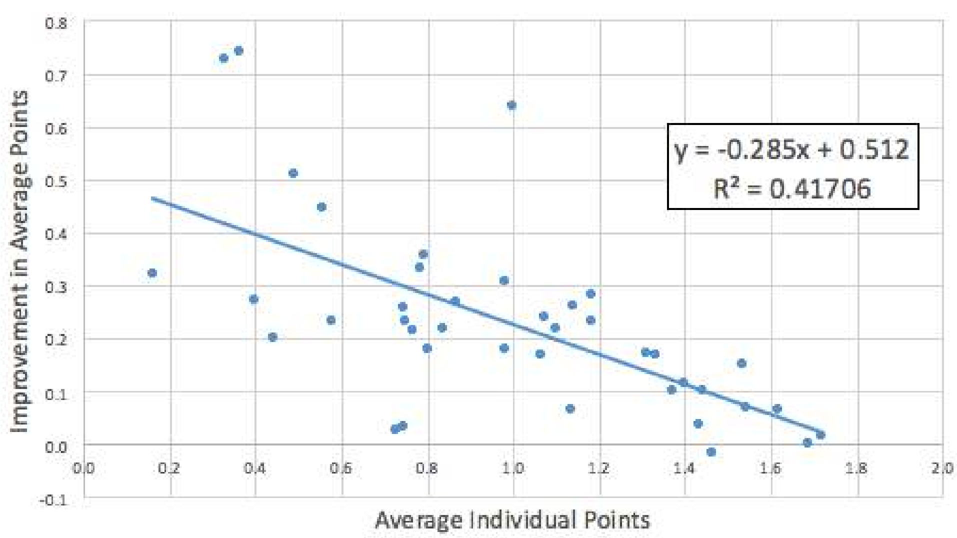

FIGURE 8.5. Average improvement in points from individual to in-group responses.

Secondly, for each GroupWork problem, we compared the average class performance from individual responses and in-group responses. Figure 8.6 shows a scatter plot of average points earned in each GroupWork question when students responded individually versus when they responded within groups. In only one of the 60 questions asked we see no average improvement in students' performance when they worked in groups. Therefore, regardless of various question types that were asked, students' performance for each question improved as a result of collaboration compared to working individually. Again, a two-tailed t test was used to compare the means of the average number of points for the two populations representing individual and in-group performance. The mean of average number of points earned increased by 0.233 from 1.003 points per question to 1.236, and the t-test test showed this difference to be statistically significant with a P value of less than 0.0001. The 95% confidence interval is 0.190 to 0.276. The corresponding effect size of the improvement is calculated as 0.609, which is again considered an impressive gain from an instructional intervention.

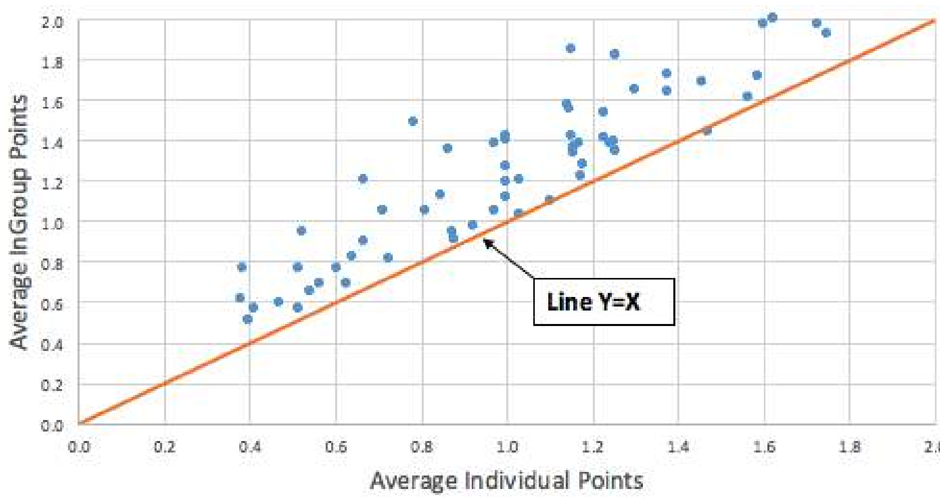

FIGURE 8.6. A scatter plot of average individual points vs average in-group points for all GroupWork questions.

8.3.2. Effects on Overall Learning in the Course. In this analysis we looked for evidence of correlation between performance in collaborative formative assessments and overall mastery of concepts covered in the course. Mastery of the material was measured as the course grade earned at the end of the semester which included cumulative grades from weekly or bi-weekly small programming assignments, three major design and programming projects, and the final written exam. Linear regression was used to test the hypothesis that performance in collaborative formative assessment could predict the learning outcomes in the course. Separate regression analyses where used to study the correlation between the final course grade and each of the individual and in-group average points in GroupWork problems. The results of the analyses are shown in Table 8.1. Both regression models indicate statistically significant,(P < 0.0001), positive correlation between course grade and each of the individual and collaborative performances in formative assessment. Indeed, the correlation coefficients are 0.647 and 0.568 between course grade & average individual points, and course grade & average in-group points, respectively.

Average Points	$\boldsymbol{R^2}$	**P-Value**	**Coefficient**
Individually	0.4188	<0.0001	13.074
In Groups	0.3232	0.0001	14.429

Table 8.1. Linear Regression Analysis: Course Grade vs. Average Individual Points and Course Grade vs. Average In-Group Points.

Figure 8.7 shows a visual representation of the regression analysis using scatter plots. On the left is a scatter plot of course grade versus average individual points, and on the right is a scatter plot of the course grade versus average in-group points. And even though both models are statistically significant, performance while working individually is a sightly stronger predictor of course performance compared to performance when working in groups. Individual performance has an R^2 value of 0.4188 which is higher than 0.3232 for in-group performance, implying a prediction model with a better fit for the data used. This observation is still consistent with an earlier study which found significant positive correlation between performance in a course and performance in formative assessment activities where students worked individually. We referred to performance in formative assessment activities as a measure of learning speed as the assessment is done immediately after new concepts are introduced. For this class, final course grade is comprised of summative assessments in which students worked individually, it therefore makes sense as to why there is a stronger correlation between course grade and individual performance in the GroupWork exercises compared to in-group performance. Moreover, despite the fact that each group member is required to submit a personal answer after group discussions, the group discussions and collaboration has a significant influence on subsequent group answers which in turn averages down the differences in mastery levels among group members.

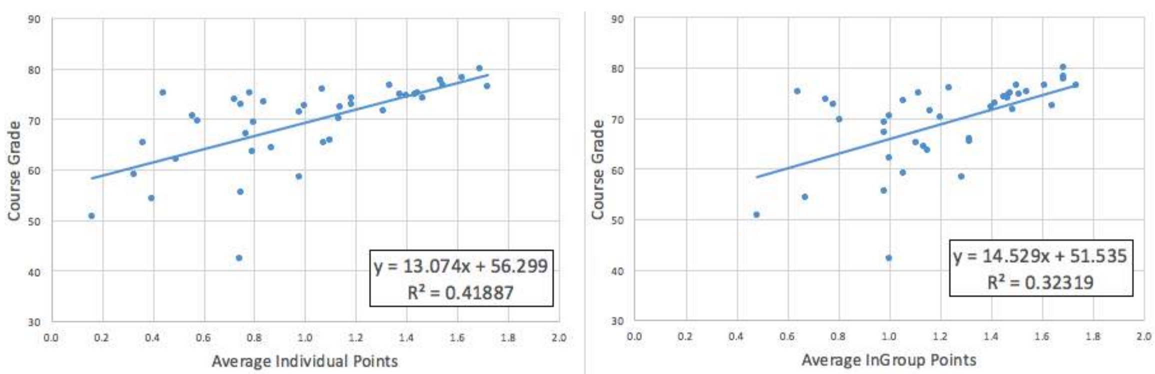

Figure 8.7. A scatter plots of course grade vs. average individual points, and course grade vs. average in-group points for all students.

8.3.3. Students' Opinions. The GroupWork assessment feature of TeachBack was introduced for the first time during the semester when this experiment was carried out. As such, the first few weeks of introducing GroupWork involved occasional fixing of technical issues and updates to the tool based on feedback from students and the instructor. Nonetheless, the overall deployment and usage of GroupWork was considered a success in facilitating collaborative formative assessment activities for the class.

Students were given a survey at the end of the course in order to get their learning experiences and evaluation of GroupWork and its use in formative assessment exercises. 37 of the 41 students participated in the survey. Using a range of 1 (not useful) to 5 (very useful), students were asked to assess the extent to which the use of GroupWork based collaborative assessment activities was useful towards their learning. 80.5% of the respondents responded with a rating of 3 or above, with 55.5% believing that it was useful with a rating of 4–5. Moreover, using a scale of 1(not satisfied) to 5 (very satisfied), 75% of the respondents reported being satisfied with how GroupWork tool was used in the course with a rating of 3–5. 44.4% of the respondents reported a satisfaction rating of 4–5.

Among the aspects of the implementation that students liked were how it forces them to start by attempting problems individually before engaging with a group in discussion and collaboration. The requirement to attempt problems individually was commented to be especially useful in tackling complex concepts as it revealed various personal misconceptions and confusions that would then be corrected during group and class discussions. Moreover, enforcing individual attempts results in many alternative answers to problems which would then be refined to fewer in-group answers. Students pointed out that having access to various alternative answers to a problem opens a new opportunity to learn from seeing how others think and approach answering questions.

Students also appreciated the introduction of group-based formative assessment using GroupWork as it “provided the opportunity for collaborative activities in class”. GroupWork activities were part of most of the class meetings during the semester, and as a result, student collaboration and formal discussions was part of the class routine. This introduced the component of collaboration to the pedagogy that would otherwise not be there as in traditional classrooms. Furthermore, the implementation of group-based formative assessment using GroupWork involves students working in groups of 3 where groups are randomly generated for each class meeting. Working with ever changing groups of classmates was pointed out by students in the survey as enabling them to “get to know others” in the class. However, when they were asked to rate how helpful GroupWork was in getting to know and meet other students in the class, in the scale of 1 (not helpful) to 5 (very helpful), only about half of the participants (58.3%) thought that it was helpful in this aspect. This survey question referred to the GroupWork feature of TeachBack, not the whole collaborative formative assessment activity. This is the case because students pointed out that conducting group discussions using the built-in GroupWork forum was not as effective, especially when most of the group members are participating remotely and not responding timely to the conversations.

Lack of online synchrony was also mentioned as issue when remote group members would not submit their individual answers in a timely manner resulting in delaying group discussions. In order to resolve this issues, students suggested assigning students into groups based on whether students are physically in the classroom or attending remotely online. In this way, students who are physically in the class can be grouped together, and can take advantage of being in the classroom, such as choosing to sit together and have face-to-face discussions. Similarly, students who are attending the class remotely should be grouped with other remote students, and they can all rely on online interaction using TeachBack. As a class during GroupWork exercises, there was no clear preference between having to physically sit together with group members or completely relying on GroupWork and the forum for group communication. From the survey responses, 44.4% of the respondents preferred moving and sitting together as a group, 41.7% preferred sitting wherever they are and use the GroupWork forum for interactions.

8.4. Conclusions

The findings of this experiment demonstrate the benefits of collaborative and problem-based learning in group-based formative assessment activities. The data analysis compared students' performance in formative assessment activities when they worked entirely individually and when they worked in small groups using the Think-Pair-Share methodology. Results showed that students reached a significantly higher level of achievement when formative assessment activities included collaboration. In general, all students performed better when they worked in groups than when they worked individually alone. Similarly, students' performance was higher for each assessment problem when students worked in collaboration.

Furthermore, the results show that students benefit in various amounts from collaboration based on individual student's mastery levels. Even though all students gather benefits from collaboration, it is the students who would usually do poorly when working individually who benefit the most. This sounds like a trivial outcome but in this case the benefit comes from having heterogeneous groups where groups are randomly generated for each class meeting. When it comes to choosing students assignment into groups based individual capabilities, heterogeneous groups tend to be more powerful than homogeneous groups [**129**]. This is because in collaboration learning comes from discussion (presentation and arguing of different perspectives and alternatives), explanations (to other group members, especially those in need of help), justification of answers, and shared resolutions.

The results demonstrated positive correlation between overall performance in the course and students' average performance in formative assessments when working individually and in groups. Performance working individually has a slightly stronger correlation compared to in-group performance. And since we were interested in studying the impact of collaboration on overall learning, a better study approach would have been comparing course performances between experimental and control populations of students, one with collaborative formative assessment and the other with only the traditional individualistic formative assessment.

In this work we have shown an implementation of collaborative formative assessment that is practical and feasible. Students think the implementation of was beneficial and useful towards their learning. Students were generally satisfied with the technical implementation of GroupWork even though they had recommendations for improvement. The demonstrated benefits to the teaching and learning outcomes imply that this instructional approach is worthy of further adoption, investigation and testing in various subject areas and curriculum designs.

CHAPTER 9

Conclusions

This doctoral work proposed the new CSAT model of pedagogy for agile teaching and learning in the college classroom. CSAT is a computer-mediated pedagogy framework based on the principles of agile software development where the pedagogy evolves and decisions are made based on continuous assessment of learners' strengths and needs. As in agile development, CSAT teachers and students work together as an agile team through open collaboration in continuous testing and evaluation of learning and teaching methods towards achieving a pedagogy that is effective and efficient for the specific curriculum goals. CSAT shifts the focus of pedagogy design from teaching, passive learning and content coverage to focusing on the individual learners, the learning processes and in discovering teaching and learning approaches that work best for the specific learners, content, and instructor.

We also designed and developed TeachBack — a web-based application to support the CSAT pedagogy model in the classroom. The application facilitates pedagogical activities such as individual and collaborative formative assessments, problem-based learning, assessment of affective and cognitive feedback, classroom communication and assistance with questions through a supervised back-channel forum, students' note-taking, and collection as well as analysis of students' data from class participation and performance in the various active learning activities. TeachBack has been successfully used in several classes and by different instructors over the last three years to support flipped classrooms and the CSAT methodology.

During the four years of this doctoral work, we conducted research studies to evaluate the feasibility and effectiveness of the CSAT model using TeachBack. We discovered that computers can be successfully used in the classroom to support learning and teaching without having a negative impact on students. Effective adoption of technology in the classroom requires a well-defined pedagogy model with the goal of augmenting the learning and teaching efforts in the class. For instance, we learned that data generated in computer-mediated classrooms can be used to more accurately discover learning characteristics of individual students, to better evaluate the effectiveness of learning and teaching methods, to rapidly detect students at-risk, and to adapt pedagogy to the needs and potentials of the learners using an evidence-based approach.

We also showed that we can successfully implement and reproduce results of various research-based learning and teaching approaches from the literature within the CSAT framework. Moreover, the CSAT framework can provide a platform for experimenting with various learning and teaching methods, and allow us to make evidence-based decisions on which methods work best for the particular instructor, students and subject matter. For example, through the design, implementation and experimentation with the GroupWork feature in TeachBack, we were able to show the positive benefits of the Think-Pair-Share methodology which provides opportunities such as cooperative, collaborative and problem-based learning in the classroom.

In another study we implemented a method to demonstrate how the CSAT model can be supported in hybrid and mixed online/physical classrooms using TeachBack and live class streaming. With remote students following the class proceedings using a live lecture stream, and actively interacting and participating in in-class activities using TeachBack, we discovered learning outcomes for remote attendees that are comparable to attending classes face-to-face. Moreover, the combination of TeachBack, live lecture streaming and access to lecture recordings can increase synchronous and engaged class attendance to nearly 100% which is not only essential for the CSAT practice but also a practical way to combat absenteeism and its impact on learning outcomes.

In concluding, based on the results and experience of using TeachBack in several implementations of CSAT, we believe that the proposed CSAT methodology is both feasible and pedagogically effective, and has the potential to revolutionize pedagogy design as well as learning and teaching in higher education.

9.1. Limitations and Future Work

TeachBack and several aspects of the CSAT pedagogy have been adopted in several classes and disciplines at Brandeis University and at one other institution. This includes different courses and instructors from the Mathematics, Biology, Anthropology, Economics and Computer Science fields. These instructors used TeachBack in supporting their pedagogies or in transitioning into active and flipped classrooms while using several aspects of the CSAT model.

Even though we used and reported on the experiences and feedback from these use cases, the experimental studies conducted to evaluate the effectiveness and feasibility of the CSAT were primarily based on Computer Science courses and were taught by the same instructor. This poses potential limitations on the presented results as being restricted by a narrow field of study. Moreover, we only worked with the college level of education. We believe that the CSAT pedagogy model can be adapted to other educational levels such as at the junior and senior high school years. It will be a worthy endeavor to more formally experiment with the application of the CSAT methodology in many other educational disciplines and levels.

We used TeachBack to successfully implement and reproduce results from the literature. These included the benefits of collaborative and problem-based learning using the Questions and GroupWork tools, as well as continuous monitoring and assessment of learning using the Stats and At-Risk features. In the future, we plan to add more features to TeachBack to increase options for learning and teaching activities in the classroom in support of the flexible CSAT pedagogy design.

We also plan to implement additional TeachBack features to more efficiently streamline the CSAT practice. The prediction and discovery of students' progress and learning characteristics can both be improved by the adding of statistical machine learning features. This will enable the instructor to more effectively monitor the pedagogy and make better-informed decisions.

Appendices

CHAPTER A

(Computer-Supported) Scrum-Based Agile Pedagogy (SBAP)

One of the central ideas of this dissertation is that the agile software development methodology can be adopted to create a more effective teaching methodology. This chapter describes (Computer-Supported) Scrum-Based Agile Pedagogy (SBAP), an extension of the main CSAT methodology which is more tightly analogous to the Scrum methodology of software development. The SBAP methodology is presented as being more revolutionary and potentially a more effective approach in transforming pedagogy design. Section A.1 describes Scrum and the general field of agile software development. Section A.2 starts with the motivational vision and characteristics of the SBAP framework, and followed by an analogy between the Scrum framework and the SBAP pedagogy design. The section ends with a detailed presentation of the SBAP framework.

A.1. Agile Software Development and Scrum

Scrum is one of the most popular and widely adopted agile software development methodologies. It is defined as an iterative design framework for complex software development projects that uses cross-functional teams, open collaboration, and well-managed chunks of time designed by the teams to achieve specific goals and actual deliverables toward an overall project [**7**]. It is not a process or a technique, but rather,"a process framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value" [**140**].

The scrum framework consists of **Scrum Teams** and their associated roles, events, artifacts, and rules; each component serves a specific purpose towards scrum success and usage. As seen throughout the description of the scrum framework, the rules bind together the events, roles, and artifacts, by governing the relationships and interaction between them. At the core of scrum is the empirical process control theory (or empiricism), which asserts that knowledge comes from experience and making decisions based on what is known. Every scrum-based implementation of this process is upheld by three pillars: transparency (of significant aspects of the process to all those responsible for the outcomes), inspection (of the artifacts and progress toward a desired goal in order to detect undesirable variances), and adaptation (timely adjustments of the process or material being processed in the case where one or more of its aspects deviate outside acceptable limits or where the resulting product would be unacceptable) [**139**, **140**].

In the next three subsections, (A.1.1, A.1.2, A.1.3), is an overview and summary of the Scrum framework from the Scrum Guide [**140**] and the book *Agile Software Development with Scrum* by Schwaber and Beedle [**139**]. More details can be found in these two sources.

A.1.1. The Scrum Team. The **Scrum Team** consists of a **Product Owner**, the **Development Team**, and a **Scrum Master**. The team is self-organizing (chooses how best to accomplish their work, rather than being directed by others outside the team), and cross-functional (has all competencies needed to accomplish the work without depending on others not part of the team). This model optimizes flexibility, creativity, and productivity.

The **Product Owner** is responsible for maximizing the value of the product and the work of the Development Team. The owner is the sole person responsible for managing the Product Backlog (the list of tasks to be done), even though she can be assisted by the Development Team, she remains accountable. She is one person, not a committee, and may represent the desires of a committee or organization through the Product Backlog.

The **Development Team** consists of professionals who solely do the work of delivering a potentially releasable Increment of a 'Done' product at the end of each Sprint. These teams are structured and empowered by the organization to organize and manage their own work which optimizes their overall efficiency and effectiveness. Some of the teams' characteristics include self-organization, cross-functionality, recognition of no titles for members other than Developer, recognition of no sub-teams, and accountability belongs to the team as a whole regardless of specialized skills and areas of focus of individual members.

The **Scrum Master** is a servant-leader of the team who is responsible for ensuring that Scrum is understood and enacted by making sure that the Scrum Team adheres to the theory, practices and rules. The master manages the interactions between those outside the team and Scrum Team while maximizing the value created by the team. The master's various roles include services to the Product Owner, the Development Team, and the organization.

A.1.2. Scrum Events. All scrum events are time-boxed such that every event has a maximum duration. Events are prescribed so as to create regularity and to minimize the need for meetings not defined in Scrum. A **Sprint** is a container for all other Scrum events, with each serving as a formal opportunity to inspect and adapt something. The sprint event makes the heart of Scrum, it is a time-box of one month or less during which a 'Done', useable, or potentially releasable product Increment is created. Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective. During the Sprint no changes are made that would endanger the Sprint Goal, quality goals do not decrease, and scope may be clarified and re-negotiated between the Product Owner and the Development Team as more is learned. Each Sprint has a definition of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product. Sprints are limited to one calendar month, which enables predictability through this regularity of inspection and adaptation of the progress. A Sprint can be cancelled by Product Owner before its duration is over — for example when its goal becomes obsolete. This may also happen under the influence of the stakeholders, the Development Team, or the Scrum Master.

The work to be performed during a Sprint is planned at the **Sprint Planning** event as a collaborative work of the entire Scrum Team. This event is typically time-boxed to a maximum of eight hours, for a one-month Sprint, where the Scrum Master ensures that the event takes place and that attendants understand its purpose. In summary, a Sprint Planning event answers the following questions: what can be delivered in the Increment resulting from the upcoming Sprint?, and how will the work needed to deliver the Increment be achieved? Created during the Sprint Planning event, the **Sprint Goal** is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment.

The **Daily Scrum** is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is accomplished by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one. During the meeting, the Development Team members explain the following questions: what did I do yesterday that helped the team meet the Sprint Goal?, what will I do today to help the team meet the Sprint Goal?, and do I see any impediment that prevents me or the team from meeting the Sprint Goal? The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. On a daily basis, the team should understand how it intends to work together as a self-organizing team to accomplish the goals and create the anticipated Increment by the end of the Sprint. These meetings improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge; all of these making it the key inspect and adapt meeting.

A **Sprint Review** is a collaborative event by the Scrum Team and stakeholders which is held at the end of the sprint to inspect the Increment and adapt the Product Backlog as necessary. This is an informal meeting which is also intended to elicit feedback and foster collaboration. Attendees to the event collaborate about what was done in the sprint, and changes to the Product Backlog, and finally on the next things that could be done to optimize value. The overall result is a revised Product Backlog that defines the possible items for the next sprint, with possible overall adjustments intended to meet new opportunities. This event is usually a four-hour time-boxed meeting for one-month sprints.

A **Sprint Retrospective**, on the other hand, is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint. It occurs after the Sprint Review but before the next Sprint Planning. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process. The purpose of the Sprint Retrospective is to:

inspect how the last Sprint went with regards to people, relationships, process, and tools; identify and order the major items that went well and potential improvements; and, create a plan for implementing improvements to the way the Scrum Team does its work. The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. Other improvements are in ways to increase product quality by adapting the definition of “Done” as appropriate. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

A.1.3. Other Scrum Artifacts. Scrum’s artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

The **Product Backlog** is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering. A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful.

The **Sprint Backlog** is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.

The **Increment** is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be "Done", which means it must be in useable condition and meet the Scrum Team's definition of "Done". Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis. For example, when a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency.

To summarize, Scrum teams make decisions themselves with an advising facilitator, they have short time-boxed self-assigned tasks, and they have a product owner monitoring and reviewing the workflow to create opportunities for the team to adapt, reflect on performance and change the product as work progresses.

All the agile software methodologies follow the same principles that are defined in the agile manifesto [**6**], which basically states that for a project to be successful we should value individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan. By following these principles, agile software development methodologies have become very successful in the software industry in developing high quality software in response to the changing requirements of the customer. [**31, 79, 98, 135, 149**]

A.2. (Computer-Supported) Scrum-Based Agile Pedagogy (SBAP)

Adopting agile principles into the process of curriculum design can be argued to result in a more learner-centered flexible approach to teaching where, according to Berry [**20**], instructors focus on teaching the particular students in their classes, rather than completing the syllabi they are given. The various agile teaching and learning methods inspired by scrum and agile principles demonstrate an effective framework for shifting the focus from instructor-centered instruction, the teaching process, and content coverage to a learner-centered pedagogy where the focus is given to students, the learning process, interaction and collaboration with students, and flexibility and adaptability of the teaching to the learning needs of the students. This analogy is demonstrated in Figure A.1.

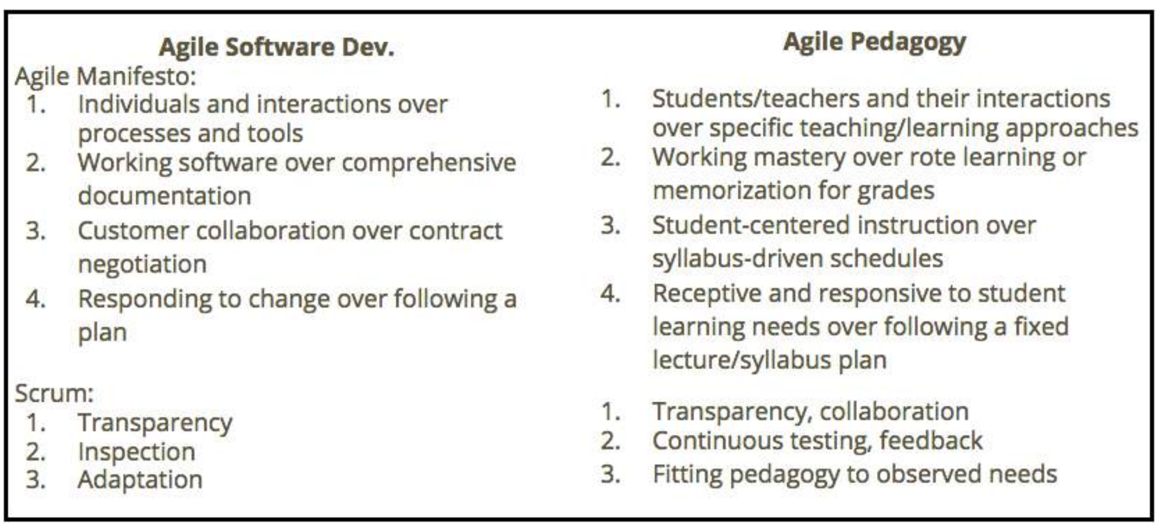

FIGURE A.1. The comparison between agile software development and agile pedagogy.

Our vision for (Computer-Supported) Scrum-Based Agile Pedagogy(SBAP) adopts this framework of agile teaching and learning methodologies, and builds on it. It extends agile teaching and learning by providing greater opportunities for improvement potential in curriculum and pedagogy design, more effective use of educational technology and caters more directly to the needs of the 21st century classroom. An example is sumarrized in Figure A.2.

- **Spontaneity**: each class is a sequence of contingent activities, selected by the instructor in real-time to best meet the needs of the students
- **Explicit Learning Goals:** each activity has explicit learning objectives shared with the students and evaluated by in-class formative assessments
- **Active Learning:** most activities involve students interacting directly with the content: collaborative, cooperative, problem-based, projects, etc
- **Agency**: students choose their own projects (with faculty approval) rather than having all students work on the same homework assignment
- **Belonging:** students work together as a class and in groups that change daily

FIGURE A.2. Some of the ways SBAP might manifest in the classroom.

First, the constantly changing nature of higher education worldwide calls for evolution and experimentation in curriculum development and in pedagogy design. There is a clear need for the education system to evolve and develop new ways to deal with current changes such as the unprecedented international, mobile and digital-native student population, large class sizes, increased demand for distance learning and teaching, and the dominance of social media and online technologies that offer access to high quality knowledge and interactions [**26**].

Second, if teaching is to be agile, then assessment, curriculum content, and instruction must work together in a flexible manner in the service of learning. According to Eleanor and Edmund [**10**], pedagogy design constitutes various elements from curriculum (content), assessment and instructional methods that teachers orchestrate and use to promote student learning. Moreover, the idea of agile teaching must therefore require dynamic pedagogy - a form of teaching with constant adaptation of assessment, curriculum and instruction in response to the potential and demonstrated learner behaviors [**10**].

In this construct, assessment determines what students have learned and not yet learned with respect to the curriculum objectives and the instruction methods used. This feedback gets fed forward in modifying subsequent instructions — the teacher's techniques and strategies used in the classroom to engage students in the learning process. We see that the instruction component of pedagogy design must be variable to change based on assessment. Furthermore, according to Eleanor and Edmund, the knowledge, skills, understanding, higher order thinking and problem solving skills which represent learning are determined by the opportunities given to learners to develop these competencies within a discipline that is organized around interrelated concepts and principles. This discipline is the curriculum content, and if teaching is to be learner-centered then curriculum content must also vary in service to effective learning. Therefore the SBAP instructor must continuously adapt the curriculum objectives and the instruction methods based on the current assessment of students' skills and knowledge.

According to Royle and Nikolic [**133**], the necessary changes must involve shifting towards increased agency for learners and rethinking of learning and teaching within a framework of digital competency. The anytime/anywhere access to information, for example, undermines teacher expertise, the myth that knowledge is fixed, and traditional notions of teacher controlled instruction and teacher-centered linear pedagogy. This also suggests a different role for the teacher in the learning process. Recent studies [**71**, **132**, **161**] illustrate that the integration and use of information technologies in education tends towards the enhancement of existing practices which accounts for the lack of real transformation and innovation in pedagogy and learning outcomes. In regard to the use of technology and curriculum design, Royle and Nikolic [**133**] argue that the educational system must be examined to ensure that it does not constrain those engaged with it but instead provide opportunities for diversity in both teaching and learning.

Based on the above goals and observations, we propose a SBAP methodology of dynamic pedagogy based on agile principles and dynamic interaction between curriculum content, assessment and instruction methods. The adaptive or dynamic trait is achieved by establishing a pedagogy that is aware and responsive to both the learning needs of the students as well as the effectiveness of the teaching approaches used.

The design involves a rich selection of teaching approaches which are founded on the best learning and teaching theories in the fields of research and practice of education. In SBAP, instructional approaches are continuously assessed based on their effectiveness and efficiency in meeting the intended learning objectives. This is accomplished through various forms of assessment and feedback-based instructional activities which provide constant awareness of students' learning levels and needs. In this construct, assessment is used to inquire about the nature and character of the learner, what has been learned and not learned, what learning and mediating processes are associated with effective learning and teaching for the learner. This requires more than the use of summative and formative assessments. In order to get more reliable insights into effective learning and teaching approaches for the learners, we use learning analytics to analyze the abundance of rich data generated during the learning and teaching processes. Throughout this design, we take advantage of the affordances and capabilities of modern information technology in achieving better efficiency and effectiveness of the supported learning and teaching activities.

A.2.1. The Analogy Between Scrum and SBAP. Instead of producing software as a product, SBAP focuses on discovering an optimal pedagogy that produces students' mastery of the skills specified in the learning objectives of the course. The optimal pedagogy brings about the best possible learning and teaching from the given teaching staff, group of students, resources and subject material. The students therefore serve as the customers or clients, with parents, the school and other stakeholders serving as secondary clients. The Scrum Team in SBAP consists of the teachers as both the Product Owner and Scrum Master, and teaching assistants and other staff as the Development Team. In SBAP students play active roles in the teaching-learning cycles of the pedagogy which include providing feedback and evaluations and testing or using the various teaching and learning activities employed in the classroom.

The Product Backlog is equivalent to the syllabus with learning objectives being analogous to 'User Stories'. A Sprint is analogous to a unit in the semester, let's call it a Pedagogical Sprint — which can be about a one-month period devoted to covering a specified set of learning objectives from the syllabus or the Pedagogical Sprint Backlog. The equivalent of an Increment or "Done" state would be students' satisfactory mastery of the unit objectives (the Sprint Goal) as measured by summative or set of formative exams. All these details are formulated during a Pedagogical Sprint Planning event where the teaching staff decides on what to be covered in the next sprint (Sprint Goals), and the corresponding set of learning and teaching activities, as well as resources (Sprint Backlog) needed in covering the specific learning objectives. The analogy is summarized in Figure A.3.

The strength of SBAP comes from the corresponding Daily Scrum, Sprint Review and Sprint Retrospective events. During Daily Scrums, the teaching staff meets before each class meeting to review progress and class activities. This is the event to inspect

Team:	
1. Product Owner	1. Instructor
2. Scrum Master	2. Instructor, TA, Staff
3. Development Team	3. Instructor, TAs, Staff, + Students
4. Stakeholders	4. Students, parents, institution, etc
Events:	
1. Sprint	1. Unit in a semester course
2. Daily Scrum	2. Pre/Post-class feedback review
3. Sprint Review	3. Unit/Midterm summative or formative exams
4. Sprint Retrospective	4. Instructor, TAs, Staff, + Students collectively review: feedback, results, resources, methods, enjoyment
Artifacts:	
1. Software Product	1. Optimal students' mastery of learning objectives
2. Product Backlog	2. Syllabus
3. Sprint BackLog	3. Unit learning objectives
4. Increment or "Done"	4. Students' satisfactory mastery of unit objectives

FIGURE A.3. The analogy between the scrum framework and SBAP.

what was covered during the previous class meeting, and the obstacles encountered and how they were addressed or perhaps how they will be addressed during the upcoming class meeting, and what will be covered (learning and teaching activities and resources) during the upcoming class towards the Pedagogical Sprint Goal. This includes the just-in-time planning, review and assessment of the pre-class, in-class and post-class activities and events. This is covered in the next subsection A.2.2 (SBAP Framework in the Classroom) and is depicted in Figure A.4.

The Pedagogical Sprint Review is held at the end of the unit where the teaching staff reviews students' progress and makes any necessary adjustments to the syllabus (the next Sprint and Product Backlogs). The inspection of the sprint Increment can be done by using summative (unit/mid-term exams) or formative assessments in assessing the mastery of the unit learning objectives. The Pedagogical Sprint Retrospective follows the sprint review, and evaluates the effectiveness of the pedagogy selections as well as the whole team. Observations and feedback from the students and the teaching staff are used to evaluate the effectiveness and efficiency of the teaching and learning activities and resources as used in the concluding unit. Strengths, weaknesses and improvements are identified and formally incorporated into the next sprint. This is covered in subsection A.2.3 (SBAP Framework Outside the Classroom) and is depicted in Figure A.5.

A.2.2. SBAP Framework in the Classroom. SBAP methodology follows the flipped classroom model in specifying both pre-class and in-class activities. Before each class, students go over materials that cover the intended learning objectives for the class. The materials can be in the form of readings, recorded lecture videos, or slides. The pre-class component is followed by mandatory form of reflection from students, which can be in the form of comments, short quizzes or simple listing of interesting or confusing topics encountered in the materials. This reflection is intended to provide the instructor with a just-in-time feedback in order to better select and direct in-class activities, enabling the instructor to more appropriately address students' strengths and weaknesses. This process is demonstrated in the pre-class component of Figure A.4.

The class time is devoted to active learning activities where students are given the opportunities to actively engage with the material and in the learning process. One or more activities can be dedicated to covering an explicit set of learning objectives. These activities can be any instructional approaches such as short lectures, group-based discussions or problem-solving. At the end of these activities, and before moving on the next set of learning objectives, formative assessments and other forms of feedback are used to assess students mastery of the intended learning objectives. If a sufficient number of students demonstrate mastery of the objectives then instruction can proceed onto the next learning objectives, otherwise, the instructor

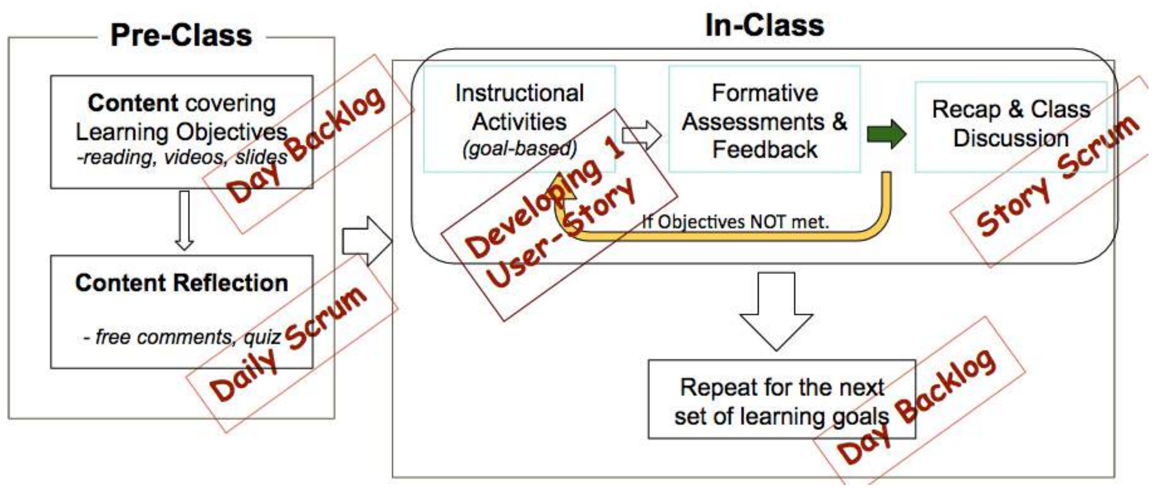

FIGURE A.4. A depiction of the students experience based on the SBAP pre-class and in-class activities.

must address students' short-comings by repeating the instructional activities with an appropriately modified approach until students master the intended goals.

A.2.3. SBAP Framework Outside the Classroom. Outside of the classroom, the SBAP methodology involves assessment of teaching and learning, interventions and adjustments to the pedagogy. This is an iterative process that can be conducted after every few classes. This relies heavily on data collection (assessment and feedback) related to specific instructional activities used in the classroom. During each cycle, sufficient evidence will have been collected and observed that can be used to assess the effectiveness and efficiency of the pedagogy selections. Evaluation is based on how effective specific instructional activities were in meeting the learning objectives (for the particular students and instructor) when used in the classroom.

This evaluation also takes into account students' feedback and opinions about how effective and efficient each instructional activity or material was towards their learning. This evaluation reveals which pedagogy selections are effective and not effective for the students and the teacher in achieving successful outcomes for the

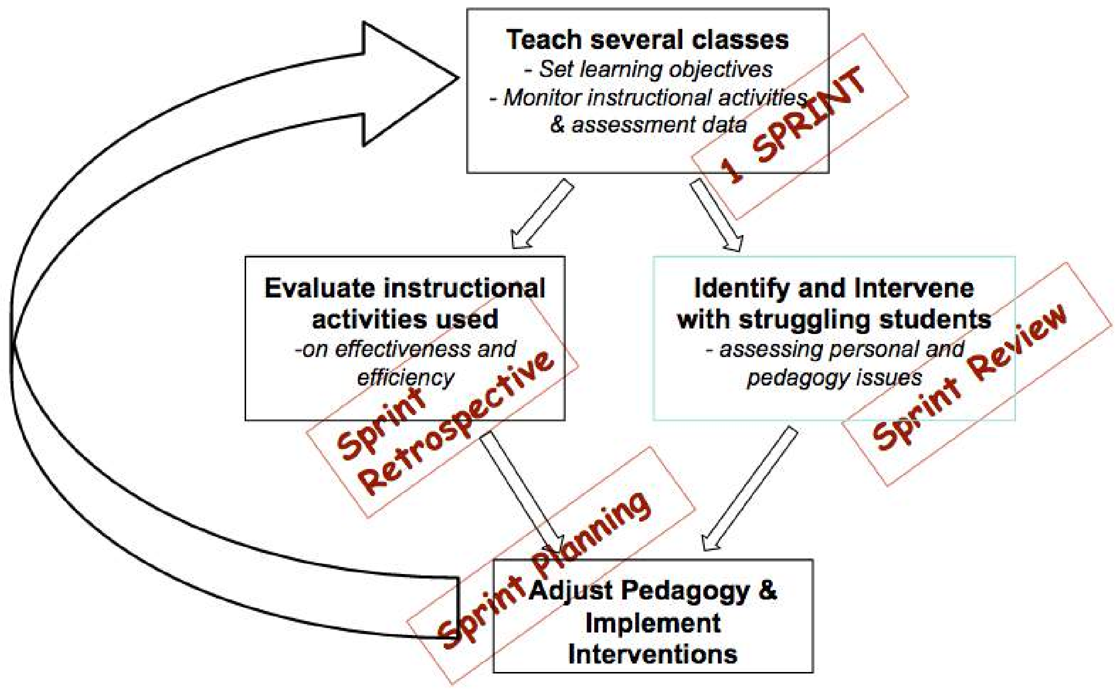

FIGURE A.5. A depiction of the instructor experience of a pedagogical sprint in SBAP

class. Instructional activities and materials that are deemed ineffective are removed from the pedagogy for the class or get appropriately adjusted based on the feedback and observed experiences. New instructional methods and learning activities can also be introduced into the class. This framework is demonstrated in Figure A.5.

Despite the best effort to adjust and modify instructional activities based on the learning needs of the class, there will always be a few students who will struggle to master the material compared to the majority of the class. Therefore, modifying the pedagogy involves implementing temporary interventions and support resources for the few students not well-served by the pedagogy selections for each iteration. With the help of technology, deeper analysis of the students' assessment, interactions and feedback data can be used to learn how to more effectively design support and interventions of these particular struggling students. Various forms of learning analytics and educational data mining can be used for this purpose. Support and intervention may include individual or group-based assistance or mentorship from the instructor and teaching assistants, assignment of additional learning activities and problem solving, and use of mandatory recitation sessions.

Bibliography

1. M'hammed Abdous, Wu He, and Cherng-Jyh Yen, *Using data mining for predicting relationships between online question theme and final grade*, Educational Technology & Society **15** (2012), no. 3, 77–88.
2. Mark Albanese, *Problem-based learning: why curricula are likely to show little effect on knowledge and clinical skills*, Medical education **34** (2000), no. 9, 729–738.
3. Mark A Albanese and Susan Mitchell, *Problem-based learning: a review of literature on its outcomes and implementation issues.*, Academic medicine **68** (1993), no. 1, 52–81.
4. Maria Isabel Alfonso and Antonio Botia, *An iterative and agile process model for teaching software engineering*, Software Engineering Education &# x00026; Training, 18th Conference on, IEEE, 2005, pp. 9–16.
5. I Elaine Allen and Jeff Seaman, *Grade change*, Tracking Online Education in the United States. Babson Survey Research Group and Quahog Research Group, LLC (2014).
6. Agile Alliance, *Agile manifesto*, Online at http://www. agilemanifesto. org **6** (2001), no. 6.1.
7. Scrum Alliance, *Scrum alliance*, URL: http://www. scrumalliance. org (2013).
8. Ashish Amresh, Adam R Carberry, and John Femiani, *Evaluating the effectiveness of flipped classrooms for teaching cs1*, Frontiers in Education Conference, 2013 IEEE, IEEE, 2013, pp. 733–735.
9. J Ben Arbaugh, *Virtual classroom characteristics and student satisfaction with internet-based mba courses*, Journal of management education **24** (2000), no. 1, 32–54.
10. Eleanor Armour-Thomas and Edmund W Gordon, *Toward an understanding of assessment as a dynamic component of pedagogy*, Unpublished manuscript (2012).
11. Thierry RH Bacro, Mulugeta Gebregziabher, and Timothy P Fitzharris, *Evaluation of a lecture recording system in a medical curriculum*, Anatomical sciences education **3** (2010), no. 6, 300–308.
12. Alison K Banikowski and Teresa A Mehring, *Strategies to enhance memory based on brain-research*, Focus on Exceptional Children **32** (1999), no. 2, 1.
13. Howard S Barrows, *Problem-based learning in medicine and beyond: A brief overview*, New directions for teaching and learning **1996** (1996), no. 68, 3–12.
14. Simon Bates and Ross Galloway, *The inverted classroom in a large enrollment introductory physics course: a case study*, The Higher Education Academy (2012).
15. H Beetham, S Jones, and L Gornall, *Career development of learning technology staff: Scoping study, a final report for the jisc jcalt*, Bristol: University of Bristol and University of Plymouth **72** (2001).
16. Mordechai Ben-Ari, *Constructivism in computer science education*, Acm sigcse bulletin, vol. 30, ACM, 1998, pp. 257–261.
17. Joseph Bergin, Clifton Kussmaul, Thomas Reichlmayr, James Caristi, and Gary Pollice, *Agile development in computer science education: practices and prognosis*, ACM SIGCSE Bulletin **37** (2005), no. 1, 130–131.

18. Jonathan Bergmann and Aaron Sams, *Remixing chemistry class*, Learning and Leading with Technology **36** (2008), no. 4, 24–27.
19. ______, *Flip your classroom: Reach every student in every class every day*, International Society for Technology in Education, 2012.
20. M Berry, *The case for agile pedagogy*, The Guardian. http://www. guardian. co. uk/teacher-network/teacher-blog/2012/may/16/agile-pedagogy-computer-programming-learning (2012).
21. Erik W Black, Kara Dawson, and Jason Priem, *Data for free: Using lms activity logs to measure community in online courses*, The Internet and Higher Education **11** (2008), no. 2, 65–70.
22. Paul Black and Dylan Wiliam, *Assessment and classroom learning*, Assessment in education **5** (1998), no. 1, 7–74.
23. Barry Boehm, Dan Port, and A Winsor Brown, *Balancing plan-driven and agile methods in software engineering project courses*, Computer Science Education **12** (2002), no. 3, 187–195.
24. Charles C Bonwell and James A Eison, *Active learning: Creating excitement in the classroom. 1991 ashe-eric higher education reports.*, ERIC, 1991.
25. Uwe M Borghoff and Johann H Schlichter, *Computer-supported cooperative work*, Computer-Supported Cooperative Work, Springer, 2000, pp. 87–141.
26. V Cable and D Willetts, *Higher education: Students at the heart of the system*, HM Treasury, London (2011).
27. Qidong Cao, Thomas E Griffin, and Xue Bai, *The importance of synchronous interaction for student satisfaction with course web sites*, Journal of Information Systems Education **20** (2009), no. 3, 331.
28. Andy Hon Wai Chun, *The agile teaching/learning methodology and its e-learning platform*, Advances in Web-Based Learning–ICWL 2004, Springer, 2004, pp. 11–18.
29. Sue Clegg, Alison Hudson, and John Steel, *The emperor's new clothes: globalisation and e-learning in higher education*, British journal of sociology of education **24** (2003), no. 1, 39–53.
30. Jacob Cohen, *Statistical power analysis for the behavioral sciences. 2nd edn. hillsdale, new jersey: L*, 1988.
31. Mike Cohn, *Succeeding with agile: software development using scrum*, Pearson Education, 2010.
32. Allan Collins, John Seely Brown, and Susan E Newman, *Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics*, Knowing, learning, and instruction: Essays in honor of Robert Glaser **18** (1989), 32–42.
33. US Congress, *Office of technology assessment.(1995). teachers and technology: Making the connection*, Washington, DC: Government Printing Office. I (1995).
34. Grainne Conole, Martin Dyke, Martin Oliver, and Jane Seale, *Mapping pedagogy and tools for effective learning design*, Computers & Education **43** (2004), no. 1, 17–33.
35. James L Cooper and Pamela Robinson, *Getting started: Informal small-group strategies in large classes*, New directions for teaching and learning **2000** (2000), no. 81, 17–24.
36. Catherine Cozzarelli and Brenda Major, *Exploring the validity of the impostor phenomenon*, Journal of Social and Clinical Psychology **9** (1990), no. 4, 401.
37. Larry Cuban, *Computers meet classroom: Classroom wins*, Teachers College Record **95** (1993), no. 2, 185.
38. Marija Cubric, *Agile learning & teaching with wikis: building a pattern*, Proceedings of the 4th International Symposium on Wikis, ACM, 2008, p. 28.
39. Nitza Davidovitch, *Learning-centered teaching and backward course design-from transferring knowledge to teaching skills*, Journal of International Education Research **9** (2013), no. 4, 329.
40. Shane Dawson, Liz Heathcote, and Gary Poole, *Harnessing ict potential: The adoption and analysis of ict systems for enhancing the student learning experience*, International Journal of Educational Management **24** (2010), no. 2, 116–128.

41. Jason A Day and James D Foley, *Evaluating a web lecture intervention in a human–computer interaction course*, Education, IEEE Transactions on **49** (2006), no. 4, 420–431.
42. Marie de Verneil and Zane L Berge, *Going online: Guidelines for faculty in higher education*, Educational Technology Review (2000), 13–18.
43. Fatima Abu Deeb and Timothy Hickey, *Spinoza: The code tutor*, Proceedings of the International Conference on Computer and Information Science and Technology. Ottawa, Ontario, Canada (2015), no. 132.
44. Sara Dolnicar, Sebastian Kaiser, Katrina Matus, and Wilma Vialle, *Can australian universities take measures to increase the lecture attendance of marketing students?*, Journal of Marketing Education **31** (2009), no. 3, 203–211.
45. Peter E Doolittle, *Understanding cooperative learning through vygotsky's zone of proximal development.*, (1995).
46. Michelle A Drouin, *If you record it, some won?t come using lecture capture in introductory psychology*, Teaching of psychology (2013), 0098628313514172.
47. Manoj Joseph D'Souza and Paul Rodrigues, *Extreme pedagogy: An agile teaching-learning methodology for engineering education*, Indian Journal of Science and Technology **8** (2015), no. 9, 828.
48. Robert Dubin and Thomas C Taveggia, *The teaching-learning paradox: A comparative analysis of college teaching methods.*, (1968).
49. Diane M Dusick and Soner Yildirim, *Faculty computer use and training: Identifying distinct needs for different populations*, Community college review **27** (2000), no. 4, 33–47.
50. Sally Farrington, Janine Tarr, Tracy Douglas, Marie-Louise Bird, Jane Pittaway, Kylie Hoffman, and Chin-Liang Beh, *This place or any place: student preferences for lecture ?places? in a blended learning environment in bioscience*, Teaching Matters 2014 Conference, 2014.
51. Robert Faulkner, Jane W Davidson, and Gary E McPherson, *The value of data mining in music education research and some findings from its application to a study of instrumental learning during childhood*, International Journal of Music Education **28** (2010), no. 3, 212–230.
52. Preston D Feden and Robert Mark Vogel, *Methods of teaching: Applying cognitive science to promote student learning*, McGraw-Hill Humanities, Social Sciences & World Languages, 2003.
53. Rebecca Ferguson, *Learning analytics: drivers, developments and challenges*, International Journal of Technology Enhanced Learning **4** (2012), no. 5-6, 304–317.
54. Catherine Finnegan, Libby V Morris, and Kangjoo Lee, *Differences by course discipline on student behavior, persistence, and achievement in online courses of undergraduate general education*, Journal of College Student Retention: Research, Theory & Practice **10** (2008), no. 1, 39–54.
55. David Flanagan and Yukihiro Matsumoto, *The ruby programming language*, " O'Reilly Media, Inc.", 2008.
56. David Fontana and Margarida Fernandes, *Improvements in mathematics performance as a consequence of self-assessment in portuguese primary school pupils*, British Journal of Educational Psychology **64** (1994), no. 3, 407–417.
57. Hugh Foot and Christine Howe, *The psyohoeduoational basis of peer-assisted learning*, Peer-assisted learning (1998), 27.
58. David S Franklin, Jennifer W Gibson, Jeanne C Samuel, William A Teeter, and Craig W Clarkson, *Use of lecture recordings in medical education*, Medical Science Educator **21** (2011), no. 1, 21–28.
59. JR Frederiksen and BJ White, *Reflective assessment of students? research within an inquiry-based middle school science curriculum*, annual meeting of the American Educational Research Association, Chicago, IL, 1997.

60. HayStack Group from MIT, *Nota Bene (NB) about nb*, `http://nb.mit.edu/about/`, 2009, Accessed: 2016-02-18.
61. M Fullan, *with stiegelbauer, s.(1991). the new meaning of educational change*, London: Cassell (1991).
62. Kathleen Fulton, *Learning in a digital age: Insights into the issues*, THE Journal (Technological Horizons in Education) **25** (1998), no. 7, 60.
63. Shelagh A Gallagher, *Problem-based learning: Where did it comes from, what does it do and where is it going?*, Journal for Education of the Gifted **20** (1997), no. 4, 332–362.
64. Gerald C Gannod, Janet E Burge, and Michael T Helmick, *Using the inverted classroom to teach software engineering*, Proceedings of the 30th international conference on Software engineering, ACM, 2008, pp. 777–786.
65. Enrique García, Cristóbal Romero, Sebastián Ventura, and Carlos de Castro, *A collaborative educational association rule mining tool*, The Internet and Higher Education **14** (2011), no. 2, 77–88.
66. D Randy Garrison, *Self-directed learning: Toward a comprehensive model*, Adult education quarterly **48** (1997), no. 1, 18–33.
67. ______, *Self-directed learning and distance education*, Handbook of distance education (2003), 161–168.
68. D Randy Garrison and J Ben Arbaugh, *Researching the community of inquiry framework: Review, issues, and future directions*, The Internet and Higher Education **10** (2007), no. 3, 157–172.
69. Pierre Gorissen, Jan Van Bruggen, and Wim Jochems, *Students and recorded lectures: survey on current use and demands for higher education*, Research in Learning Technology **20** (2012).
70. Charlotte Nirmalani Gunawardena and Marina Stock McIsaac, *Distance education*, Handbook of research on educational communications and technology **2** (2004), 355–395.
71. Mark Hadfield, Michael Jopling, Karl Royle, and Liz Southern, *Evaluation of the training and development agency for schools? funding for ict in itt projects*, TDA: London (2009).
72. Sherif Halawa, Daniel Greene, and John Mitchell, *Dropout prediction in moocs using learner activity features*, Experiences and best practices in and around MOOCs **7** (2014).
73. Wu He et al., *Using text mining to uncover students' technology-related problems in live video streaming*, British Journal of Educational Technology **42** (2011), no. 1, 40–49.
74. George Hein, *Constructivist learning theory*, Institute for Inquiry. Available at:/http://www. exploratorium. edu/ifi/resources/constructivistlearning. htmlS (1991).
75. Timothy J Hickey and William T Tarimo, *The affective tutor*, Journal of Computing Sciences in Colleges **29** (2014), no. 6, 50–56.
76. Gregory W Hislop, Michael J Lutz, J Fernando Naveda, W Michael McCracken, Nancy R Mead, and Laurie A Williams, *Integrating agile practices into software engineering courses*, Computer science education **12** (2002), no. 3, 169–185.
77. Cindy E Hmelo-Silver, *Problem-based learning: What and how do students learn?*, Educational psychology review **16** (2004), no. 3, 235–266.
78. Bob Hoffman, *Managing the information revolution: Planning the integration of school technology.*, Nassp Bulletin **80** (1996), no. 582, 89–98.
79. Emam Hossain, Muhammad Ali Babar, and Hye-young Paik, *Using scrum in global software development: a systematic literature review*, 2009 Fourth IEEE International Conference on Global Software Engineering, Ieee, 2009, pp. 175–184.
80. Jui-Long Hung and Steven M Crooks, *Examining online learning patterns with data mining techniques in peer-moderated and teacher-moderated courses*, Journal of Educational Computing Research **40** (2009), no. 2, 183–210.

81. David W Johnson, Roger T Johnson, and Karl A Smith, *Cooperative learning returns to college what evidence is there that it works?*, Change: the magazine of higher learning **30** (1998), no. 4, 26–35.
82. Amy NB Johnston, Helen Massa, and Thomas HJ Burne, *Digital lecture recording: A cautionary tale*, Nurse education in practice **13** (2013), no. 1, 40–47.
83. Spencer Kagan, *The structural approach to cooperative learning*, Educational leadership **47** (1989), no. 4, 12–15.
84. Arun Karnad, *Student use of recorded lectures: a report reviewing recent research into the use of lecture capture technology in higher education, and its impact on teaching methods and attendance*, (2013).
85. Greg Kearsley, *The nature and value of interaction in distance learning*, 1995.
86. Gabrielle E Kelly, *Lecture attendance rates at university and related factors*, Journal of Further and Higher Education **36** (2012), no. 1, 17–40.
87. Asher Koriat and Robert A Bjork, *Illusions of competence in monitoring one's knowledge during study.*, Journal of Experimental Psychology: Learning, Memory, and Cognition **31** (2005), no. 2, 187.
88. Aditi Kothiyal, Rwitajit Majumdar, Sahana Murthy, and Sridhar Iyer, *Effect of think-pair-share in a large cs1 class: 83% sustained engagement*, Proceedings of the ninth annual international ACM conference on International computing education research, ACM, 2013, pp. 137–144.
89. Maureen J Lage, Glenn J Platt, and Michael Treglia, *Inverting the classroom: A gateway to creating an inclusive learning environment*, The Journal of Economic Education **31** (2000), no. 1, 30–43.
90. HA Latchman, Ch Salzmann, Denis Gillet, and Hicham Bouzekri, *Information technology enhanced learning in distance and conventional education*, Education, IEEE Transactions on **42** (1999), no. 4, 247–254.
91. Wendy Leadbeater, Tom Shuttleworth, John Couperthwaite, and Karl P Nightingale, *Evaluating the use and impact of lecture recording in undergraduates: Evidence for distinct approaches by different groups of students*, Computers & Education **61** (2013), 185–192.
92. Avraham Leff and James T Rayfield, *Web-application development using the model/view/controller design pattern*, Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International, IEEE, 2001, pp. 118–127.
93. Bernard Lisewski and Paul Joyce, *Examining the five-stage e-moderating model: designed and emergent practice in the learning technology profession*, Research in Learning Technology **11** (2003), no. 1.
94. Frank Lyman, *Think-pair-share: An expanding teaching technique*, MAA-CIE Cooperative News **1** (1987), no. 1, 1–2.
95. Frank T Lyman, *The responsive classroom discussion: The inclusion of all students*, Mainstreaming digest (1981), 109–113.
96. Claire H Major and Betsy Palmer, *Assessing the effectiveness of problem-based learning in higher education: Lessons from the literature*, Academic exchange quarterly **5** (2001), no. 1, 4–9.
97. B Jean Mandernach, Emily Donnelli, Amber Dailey, and Marthann Schulte, *A faculty evaluation model for online instructors: Mentoring and evaluation in the online classroom*, Online Journal of Distance Learning Administration **8** (2005), no. 3.
98. Chris Mann and Frank Maurer, *A case study on the impact of scrum on overtime and customer satisfaction.*, Agile, 2005, pp. 70–79.

99. Jon-Paul Marchand, Marion L Pearson, and Simon P Albon, *Student and faculty member perspectives on lecture capture in pharmacy education*, American journal of pharmaceutical education **78** (2014), no. 4.
100. D Mårtenson, H Eriksson, and M INGELMAN-SUNDBERG, *Medical chemistry: evaluation of active and problem-oriented teaching methods*, Medical education **19** (1985), no. 1, 34–42.
101. Florence Martin, Michele A Parker, and Deborah F Deale, *Examining interactivity in synchronous virtual classrooms*, The International Review of Research in Open and Distributed Learning **13** (2012), no. 3, 228–261.
102. Peter Massingham and Tony Herrington, *Does attendance matter? an examination of student attitudes, participation, performance and attendance*, Journal of university teaching & learning practice **3** (2006), no. 2, 3.
103. WJ McKeachie, *Research on college teaching.*, Educational Perspectives **11** (1972), no. 2, 3–10.
104. Carmel McNaught, *Identifying the complexity of factors in the sharing and reuse of resources*, Reusing online resources: A sustainable approach to elearning (2003), 199–211.
105. Jay McTighe and Frank T Lyman, *Cueing thinking in the classroom: The promise of theory-embedded tools*, Educational Leadership **45** (1988), no. 7, 18–24.
106. Jay McTighe and Ronald S Thomas, *Backward design for forward action*, Educational Leadership **60** (2003), no. 5, 52–55.
107. Barbara J Millis and Philip G Cottell Jr, *Cooperative learning for higher education faculty. series on higher education.*, ERIC, 1997.
108. Stephen K Mittelstet, *The role of the administrator in foreign language education.*, (1992).
109. Marin Moravec, Adrienne Williams, Nancy Aguilar-Roca, and Diane K O'Dowd, *Learn before lecture: a strategy that improves learning outcomes in a large introductory biology class*, CBE-Life Sciences Education **9** (2010), no. 4, 473–481.
110. Shazia Mumtaz, *Factors affecting teachers' use of information and communications technology: a review of the literature*, Journal of information technology for teacher education **9** (2000), no. 3, 319–342.
111. Mrinal Kanti Musib, *Student perceptions of the impact of using the flipped classroom approach for an introductory-level multidisciplinary module*, CDTL Brief **17** (2014), no. 2, 15–20.
112. Dennis C Neale, John M Carroll, and Mary Beth Rosson, *Evaluating computer-supported cooperative work: models and frameworks*, Proceedings of the 2004 ACM conference on Computer supported cooperative work, ACM, 2004, pp. 112–121.
113. Sarah E Newton, Mary B Eberly, and Robert A Wiggins, *The syllabus as a tool for student-centered learning*, The Journal of General Education **50** (2001), no. 1, 56–74.
114. Geoffrey R Norman and Henk G Schmidt, *The psychological basis of problem-based learning: A review of the evidence.*, Academic medicine **67** (1992), no. 9, 557–65.
115. ——— , *Effectiveness of problem-based learning curricula: theory, practice and paper darts*, Medical education **34** (2000), no. 9, 721–728.
116. Ron Oliver, *Winning the toss and electing to bat: maximising the opportunities of online learning*, Proceedings of the 9th improving student learning conference, vol. 1, Citeseer, 2002, pp. 35–44.
117. Sandi Osters and F Simone Tiu, *Writing measurable learning outcomes*, 3rd Annual Texas A&M Assessment Conf, 2008.
118. Predrag Pale, Juraj Petrović, and Branko Jeren, *Assessing the learning potential and students' perception of rich lecture captures*, Journal of computer assisted learning **30** (2014), no. 2, 187–195.
119. A Sullivan Palincsar, *12 social constructivist perspectives on teaching and learning*, An introduction to Vygotsky (2005), 285.

120. Joy Palmer, *Fifty modern thinkers on education: From piaget to the present*, Psychology Press, 2001.
121. Joy Palmer, Liora Bresler, and David Edward Cooper, *Fifty major thinkers on education: From confucius to dewey*, Psychology Press, 2001.
122. Rob Phillips, Dorit Maor, Wendy Cumming-Potvin, Pauline Roberts, Jan Herrington, Greg Preston, Elizabeth Moore, and Laura Perry, *Learning analytics and study behaviour: A pilot study*, (2011).
123. Rebecca Pope-Ruark, Michelle Eichel, Sarah Talbott, and Kasey Thornton, *Let's scrum: How scrum methodology encourages students to view themselves as collaborators*, Teaching and Learning Together in Higher Education **1** (2016), no. 3, 5.
124. Michael Prince, *Does active learning work? a review of the research*, Journal of engineering education **93** (2004), no. 3, 223–231.
125. Arkalgud Ramaprasad, *On the definition of feedback*, Behavioral Science **28** (1983), no. 1, 4–13.
126. April Reed, *Computer-mediated communication (cmc) and the traditional classroom*, Teaching with Technology Today **5** (2000), no. 6.
127. TS Roberts, L McNamee, and S Williams, *Online collaborative learning in higher education*, Hershey, PA: Information Science (2004).
128. SI Robertson, J Calder, P Fungi, A Jonest, T O'Shea, and G Lambrechtst, *Pupils, teachers & palmtop computers*, Journal of Computer Assisted Learning **12** (1996), no. 4, 194–204.
129. T Roger and David W Johnson, *Cooperative learning: two heads learn better than one*, Context **18** (1988).
130. Barbara Rogoff, *Developing understanding of the idea of communities of learners*, Mind, culture, and activity **1** (1994), no. 4, 209–229.
131. Lawrence A Rowe, Diane Harley, Peter Pletcher, and Shannon Lawrence, *Bibs: A lecture webcasting system*, Center for Studies in Higher Education (2001).
132. Karl Royle and Mark Hadfield, *From ?posh pen and pad?to participatory pedagogies: One story of a netbook implementation project with 108 pupils in two primary schools*, International Journal of Mobile and Blended Learning (IJMBL) **4** (2012), no. 1, 1–17.
133. Karl Royle and Jasmina Nikolic, *Agile digital age pedagogy for teachers: Adapt.*
134. D Royce Sadler, *Formative assessment and the design of instructional systems*, Instructional science **18** (1989), no. 2, 119–144.
135. Outi Salo and Pekka Abrahamsson, *Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum*, IET software **2** (2008), no. 1, 58–64.
136. Marlene Scardamalia, Carl Bereiter, Robert S McLean, Jonathan Swallow, and Earl Woodruff, *Computer-supported intentional learning environments*, Journal of educational computing research **5** (1989), no. 1, 51–68.
137. John Schacter, *The impact of education technology on student achievement: What the most current research has to say.*, (1999).
138. Julie Schell, Brian Lukoff, and Cassandre Alvarado, *Using early warning signs to predict academic risk in interactive, blended teaching environments*, Internet Learning **3** (2015), no. 2, 6.
139. K Schwaber and Mike Beedle, *Agile software development with scrum*, (2002).
140. Ken Schwaber and Jeff Sutherland, *The scrum guide*, Scrum Alliance (2011).
141. Yael Sharan and Shlomo Sharan, *Group investigation in the cooperative classroom*, Handbook of cooperative learning methods (1994), 97–114.
142. George Siemens and Ryan SJ d Baker, *Learning analytics and educational data mining: towards communication and collaboration*, Proceedings of the 2nd international conference on learning analytics and knowledge, ACM, 2012, pp. 252–254.

143. George Siemens and Phil Long, *Penetrating the fog: Analytics in learning and education.*, EDUCAUSE review **46** (2011), no. 5, 30.
144. Barbara Leigh Smith and Jean T MacGregor, *What is collaborative learning*, 1992.
145. Liyan Song and Janette R Hill, *A conceptual model for understanding self-directed learning in online environments*, Journal of Interactive Online Learning **6** (2007), no. 1, 27–42.
146. Kathleen Spada, *Higher education in the digital age [book review]*, Professional Communication, IEEE Transactions on **57** (2014), no. 2, 150–153.
147. Christopher W Starr, Bill Manaris, and RoxAnn H Stalvey, *Bloom's taxonomy revisited: specifying assessable learning objectives in computer science*, ACM SIGCSE Bulletin **40** (2008), no. 1, 261–265.
148. Bethany B Stone, *Flip your classroom to increase active learning and student engagement*, Proceedings from 28th Annual Conference on Distance Teaching & Learning, Madison, Wisconsin, USA, 2012.
149. Jeff Sutherland, Anton Viktorov, Jack Blount, and Nikolai Puntikov, *Distributed scrum: Agile project management with outsourced development teams*, System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, IEEE, 2007, pp. 274a–274a.
150. Karen Swan and Marco Mitrani, *The changing nature of teaching and learning in computer-based classrooms*, Journal of Research on Computing in Education **26** (1993), no. 1, 40–54.
151. Karen Swan, P Shea, J Richardson, P Ice, DR Garrison, M Cleveland-Innes, and JB Arbaugh, *Validating a measurement tool of presence in online communities of inquiry*, E-mentor **2** (2008), no. 24, 1–12.
152. William T. Tarimo, Fatima Abu Deeb, and Timothy J. Hickey, *Computers in the cs1 classroom*, Proceedings of the 7th International Conference on Computer Supported Education, 2015, pp. 67–74.
153. William T Tarimo, Fatima Abu Deeb, and Timothy J Hickey, *A flipped classroom with and without computers*, International Conference on Computer Supported Education, Springer International Publishing, 2015, pp. 333–347.
154. ———, *Early detection of at-risk students in cs1 using teachback/spinoza*, Journal of Computing Sciences in Colleges **31** (2016), no. 6, 105–111.
155. Bruce Tate and Curt Hibbs, *Ruby on rails: Up and running: Up and running*, " O'Reilly Media, Inc.", 2006.
156. Kenneth Tobin, *The role of wait time in higher cognitive level learning*, Review of educational research **57** (1987), no. 1, 69–95.
157. Keith Topping and Stewart Ehly, *Peer-assisted learning*, Routledge, 1998.
158. Harry Torrance, *Assessment as learning? how the use of explicit learning objectives, assessment criteria and feedback in post-secondary education and training can come to dominate learning. 1*, Assessment in Education **14** (2007), no. 3, 281–294.
159. Tomoko Traphagan, John V Kucsera, and Kyoko Kishi, *Impact of class lecture webcasting on attendance and learning*, Educational Technology Research and Development **58** (2010), no. 1, 19–37.
160. Carsten Ullrich, Ruimin Shen, Ren Tong, and Xiaohong Tan, *A mobile live video learning system for large-scale learning?system design and evaluation*, Learning Technologies, IEEE Transactions on **3** (2010), no. 1, 6–17.
161. Jean Underwood, Thomas Baguley, Philip Banyard, Gayle Dillon, Lee Farrington-Flint, Mary Hayes, Gabrielle Le Geyt, Jamie Murphy, and Ian Selwood, *Understanding the impact of technology: Learner and school level factors*, (2010).
162. David T Vernon and Robert L Blake, *Does problem-based learning work? a meta-analysis of evaluative research.*, Academic medicine **68** (1993), no. 7, 550–63.

163. Brian R Von Konsky, Jim Ivins, and Susan J Gribble, *Lecture attendance and web based lecture technologies: A comparison of student perceptions and usage patterns*, Australasian Journal of Educational Technology **25** (2009), no. 4, 581–595.
164. Lev Vygotsky, *Interaction between learning and development*, Readings on the development of children **23** (1978), no. 3, 34–41.
165. Eugene P Wagner, Howell Sasser, and Warren J DiBiase, *Predicting students at risk in general chemistry using pre-semester assessments and demographic information*, Journal of Chemical Education **79** (2002), no. 6, 749.
166. Feng Wang and Michael J Hannafin, *Design-based research and technology-enhanced learning environments*, Educational technology research and development **53** (2005), no. 4, 5–23.
167. Martin Weller, *A pedagogy of abundance*, Revista española de pedagogía (2011), 223–235.
168. Grant P Wiggins and Jay McTighe, *Understanding by design*, Ascd, 2005.
169. Annika Wolff, Zdenek Zdrahal, Andriy Nikolov, and Michal Pantucek, *Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment*, Proceedings of the third international conference on learning analytics and knowledge, ACM, 2013, pp. 145–149.
170. Robert E Yager, *The constructivist learning model*, The science teacher **58** (1991), no. 6, 52.
171. Tanya Yerigan, *Getting active in the classroom*, Journal of College Teaching & Learning (TLC) **5** (2008), no. 6.