Use of JiTT in a Graduate Software Testing Course: An Experience Report

Alexandra Martinez
ECCI, Universidad de Costa Rica
San Jose, Costa Rica
alexandra.martinez@ecci.ucr.ac.cr

ABSTRACT

This paper describes our experience using Just-in-Time Teaching (JiTT) in a graduate Software Testing course during two semesters. JiTT is a pedagogical strategy that bridges in-class and out-of-class components through preparatory web-based assignments, known as warm-ups. Our JiTT design was as follows. The preparatory out-of-class component consisted of a reading test, which required students to read a chapter from the textbook and then answer a web-based test available in our virtual platform. Reading tests were due the day before class in order to give the teacher enough time to read over the student's responses and adjust the next lesson accordingly. The in-class component was organized around student common misconceptions or difficulties, extracted from the reading tests submitted by the students. Discussions and cooperative learning activities were among the teaching strategies used in class. Our approach was assessed from the students' and teacher's perspective. The students' perspective was obtained from a survey. The teacher's perspective consisted in an assessment of strengths and limitations. Results from our evaluation show that a vast majority of students believe their learning improves when they prepare for class by reading the material in advance. They also think that reading tests are an effective way of verifying that students did the assigned reading. Most of them also consider that JiTT is an appropriate teaching strategy for the course. From the teacher's perspective, a major strength found was that students were more engaged in class, asking interesting questions that enriched class discussion. Also, the use of open-ended (essay-type) questions in reading tests has the additional benefit of helping them become better writers (organize their ideas better and clarify their thinking through writing).

CCS CONCEPTS

• **Applied computing → Interactive learning environments**; Collaborative learning;

KEYWORDS

JiTT, Just-in-Time Teaching, Software testing, graduate course

ICSE-SEET'18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5660-2/18/05.
https://doi.org/10.1145/3183377.3183397

ACM Reference Format:
Alexandra Martinez. 2018. Use of JiTT in a Graduate Software Testing Course: An Experience Report. In *ICSE-SEET'18: 40th International Conference on Software Engineering: Software Engineering Education and Training Track, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3183377.3183397

1 INTRODUCTION

Just-in-Time Teaching (JiTT) is a pedagogical strategy developed by Novak et al. [17] to help nontraditional students improve their learning [19] . The essence of JiTT is to link the in-class and out-of-class activities through *warm-ups* [19]. JiTT warm-ups are preparatory web-based assignments, which often consists in assigning some reading material to the students and asking them to complete a short online assignment a few hours before class. Based on the student responses to the assignment, the teacher plans for the upcoming class. JiTT has been shown to help students develop a feeling of ownership, actively participate in the different class activities, have greater control of their learning, and feel more motivated and engaged in the course [2, 17]. With JiTT, class time is used more effectively because less time is spent on material students have learned from reading, and more time is spent on more difficult subjects [2]. Experiences on the use of JiTT in Computer Science courses have been reported in the literature [1, 6, 7, 9–11, 14, 18] .

JiTT is part of a broader approach for teaching and learning called Active Learning [3]. McConnell [16] describes Active Learning as the use of techniques that involve students in the learning process, rather than listening to a lecture in a passive way. Students can actively learn by reading, writing, discussing, solving a problem, or responding to challenging questions [15, 16]. Active Learning has been shown to improve student comprehension and retention of material [5], as well as to increase student motivation and higher order thinking [3, 16]. Active Learning techniques have been successfully used in Computer Science [4–6, 8, 13, 15].

The goal of this paper is to report on our experience using JiTT in a graduate-level Software Testing course, for two semesters. This course has 4 credit-hours and 64 hours of class time in a 16-week semester. It is offered by the Master of Science Program in Computer Science and Informatics at the University of Costa Rica.

The rest of the paper is organized as follows. Section 2 summarizes related work in the area. Section 3 describes the context of the course. Section 4 explains the design of the JiTT strategy for the course. Section 5 specifies when and how was the strategy implemented. Section 6 presents the assessment of the implemented strategy and shows our findings. Finally, Section 7 concludes the paper and outlines our plans for future work.

2 RELATED WORK

Several works describing the use of JiTT in the field of Computer Science have been published. We mention the more relevant ones next.

Bailey et al. [1] present an implementation of JiTT and Peer Instruction in an introductory computer science course. They emphasize on the feedback loop between out-of-class work (web component) and classroom component as a key element of the JiTT strategy. Their web component consists of warm-up exercises, course web pages, and assessment exercises (homework and lab assignments). The classroom component consists of lectures, laboratory sessions, and in-class student activities. Based on a qualitative assessment, they report that students were satisfied with the strategy but complain about the amount of out-of-class work. Our work, in contrast, uses an out-of-class component based solely on ready tests. Also, our students do not usually complain about the amount of work outside the class.

Davis [9] describes her experience with JiTT in courses that featured a large reading component, namely Operating Systems, Networking, Software Design, and Human-Computer Interaction. Two approaches are presented: the first one consists of discussion questions emailed by students about the assigned readings; the second consists of warm-up exercises prepared by the instructor. The author states that the second approach is more labor intensive on the instructor's side, but helps students tie reading and thinking out of class with work on design problems in class. In our case, the reading component of the course is not too large, but we share the view on warm-ups.

Carter [6] reports on an experiment that combines JiTT with Peer Instruction in an introductory computing course for engineering students. The experiment lasted just one week and was as follows. First, students were asked to review some online screencasts before class, then at the beginning of class they were assessed on that material, and finally, a lecture was provided by the teacher as needed. The author claims that student feedback was positive but recognizes that the impact on student learning could not be assessed due to the short time students experienced with this technique. More recently, the same author developed a complete set of online multimedia screencasts for an introductory programming course [7] . These screencasts are proposed as an alternative to reading assignments from a textbook. Unlike this work, our JiTT implementation uses a more traditional type of out-of-class component, consisting of readings from books or articles.

Li et al. [11] reports on their experience with JiTT in a Program Design course. They use a website to put up real-time content for students before each class (instead of assigning readings), which is similar to Carter's approach [7]. Authors conclude that the effect of the JiTT-based teaching mode is better than the traditional one. In our work, reading assignments are given to the students instead of website content.

Gurka [10] reports her experience using JiTT in CS1 and CS2 courses. In her study, participation in the JiTT questions was optional, but extra point were offered. Questions were either conceptual or code-specific. The author claims that class participation improved (since students had already worked on the questions at their own pace) and that students enjoyed the approach, studying more time outside the class. However, no formal assessment was made on the effect of JiTT. In contrast, we conducted a student survey to gather their perspective on the use of JiTT, and provide the results of this assessment.

Perez-Poch et al. [18] present their experience using JiTT in a Computer Science Fundamentals course from an Industrial Engineering Department. Authors found significant improvements in student satisfaction, final grades, and absenteeism reduction. Our approach differs from theirs in that their online assignments (out of class component) are not mandatory where ours is. Also, in their case, these assignments are practical exercises in preparation for laboratory sessions, whereas in our case, reading tests also cover theoretical concepts. Last, they used a formal multivariate analysis with two control groups, whereas we did not have control groups because only one group of this course is opened every year.

We reported in [14] a previous experience using JiTT in an undergraduate database course. A significant improvement on the students' interest in the course and learning of the material (based on exams and final course grades) was reported. This work and the current one differ in several aspects: (1) the course context - one is an undergraduate Database course, the other is a graduate Software Testing course (meaning the maturity and motivation of the students is different), (2) the weight of JiTT reading tests was lower (10%) in the Database course than in the Software Testing course (40% and 22% for the two instances), (3) the grading was performed by the teacher in the Software Testing course, rather than by the teaching assistant as in the Database course. Because the teacher was doing all the grading in the Software Testing course, she made a detailed assessment of the students responses to reading tests, including their writing, which is one the major contribution of the current work.

Unlike all the works presented in this section, our JiTT approach was designed for and implemented in a graduate-level course (all others have use JiTT in undergraduate courses). Another difference with previous works is the subject of the course, since no one else has studied JiTT in the context of a Software Testing course, to the best of our knowledge. Last, we differ from other works in the use of open-ended questions to both uncover weaknesses in student understanding, and help improve their writing. These are the main contributions of our work.

3 COURSE CONTEXT

The graduate-level Software Testing course, is part of a group of several software engineering courses regularly offered by the Master of Science Program in Computer and Information Science at our university. It is a 4-credit-hour course with 64 hours of class time in a 16-week semester, and a 2-credit-hour co-requisite lab course where students put theory into practice by developing a large practical project or an applied research project. The class meets once a week for 3.5 hours.

3.1 Course Objectives and Contents

The main goal of the course is to guide the students in the learning of the theoretical foundations and necessary skills for understanding and applying software testing processes, techniques, and activities within the context of software quality assurance.

Table 1: Course learning units.

Unit name	Unit content
1. Principles of Software Testing and Quality	Basics: quality assurance and control, software testing, verification and validation (V&V), test cases, software defects. The testing process and V&V activities during the software lifecycle. Quality factors.
2. Planning and Managing the Testing Process	Components of a Test Plan. Management of the testing process. Defect reporting and tracking. Test metrics.
3. Types of Testing	Static vs. dynamic tests, manual (technical reviews) vs. automated tests, black box vs. white box tests, functional vs. non-functional tests (load and performance, security, localization, usability, accessibility), and regression tests.
4. Test Design Techniques	Black-box techniques: equivalence partitioning, boundary value analysis, cause-effect graphing, intuition and experience. White-box techniques: control flow testing (statement coverage, decision cover-age, condition coverage, decision-condition coverage, multiple-condition coverage, path coverage, basis path coverage), data flow testing (all defs coverage, all uses coverage, all DU paths coverage). Combination Testing (all-pairs technique). Exploratory Testing.
5. Levels of Testing	Unit testing, integration testing, system testing, user acceptance testing. Alfa, beta, pre-release (RC) and final tests.
6. Advanced Topics	Student presentations on advanced topics in software testing. Candidate topics include: mutation testing, cloud testing, model-based testing, database testing, performance/load testing, security testing, and automated test generation tools.

After completing the course, students are expected to be able to:

- Understand and explain the fundamental concepts of software testing.
- Identify best practices for software testing and quality.
- Compare and apply different techniques, levels, and types of software testing.
- Plan, manage and implement a software testing process.

The course reading materials are based on several reference books and recent papers in the area. The course contents are divided in six units, as listed on Table 1.

3.2 Course Tools and Labs

Tools. The tools used in the course are Microsoft Visual Studio Ultimate (which includes the Test Manager) and Microsoft Team Foundation Server (TFS). Team Foundation Server is the collaboration platform at the core of Microsoft's application lifecycle management solution, and it delivers source control, work item tracking, automated builds, project web access and informative web sites and reporting. Visual Studio, on the other hand, is an integrated development environment, which includes support for testing tasks such as unit testing, performance testing, load testing, and UI automation. It also allows visualization and reporting of relevant work items during the testing process. These tools were chosen because they provide an integrated and comprehensive suite for application lifecycle management, including test planning, authoring, automation, execution, tracking, monitoring and managing. More details on the course toolware are given in [12].

Table 2: Course labs and associated course units.

Lab practice	Course unit
Lab1: User stories and task management Lab2: Source control Lab3: Test case and test plan management Lab4: Bug reporting and management	2. Planning and Managing the Testing Process
Lab5: User interface automation Lab6: Performance testing Lab7: Load testing	3. Types of Testing
Lab8: Code coverage	4. Test Design Techniques
Lab9: Unit testing	5. Levels of Testing
Lab10: Virtual environments	6. Advanced Topics

Labs. We used a total of ten labs in the course. These lab practices together with the course units they support are shown in Table 2. As can be observed from this table, all but one of the course units have supporting labs, which allow students to put into practice the concepts studied throughout the course. For most of the labs, the software under test (SUT) was the MVC Music Store application that Microsoft provides as tutorial application on ASP.NET MVC. This is a web application that sells music albums online, and implements site administration, user sign-in, and shopping cart functionality. The specifics of the labs can be found in [12].

3.3 The Students

The majority of our master's students work fulltime in different industries, from small and medium-size software organizations to large IT Departments of non-IT companies. Most of the students work on software development, some are software architects or project managers although there is a trend for students to start working as software testers. A small percentage of our students have other job profiles such as service management and support desk professionals.

3.4 The Teacher

The author was the instructor of the Software Testing courses. She is an Associate Professor at the Department of Computer and Information Science, with over seven years of teaching experience.

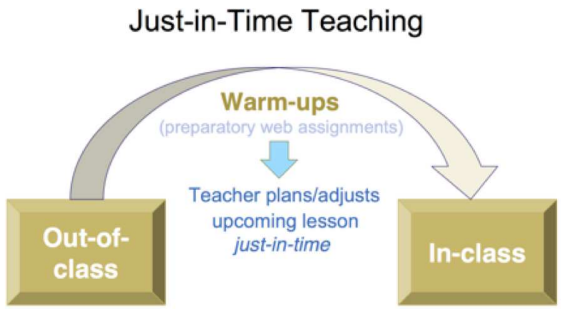

Figure 1: Just-in-Time Teaching components and workflow.

She has a Ph.D. in Computer Engineering, academic background in software testing, and three years of industry experience as a software tester.

4 DESIGN

As it was previously mentioned, the essence of JiTT lies in bridging the in-class and out-of-class components through preparatory web assignments, known as JiTT "warm-ups"[19]. Figure 1 shows a diagram of JiTT workflow. The following sections describe how we designed the two main components of the JiTT strategy.

4.1 Design of the Out-of-class Component

The out-of-class component consisted of a series of "reading tests" (warm-ups in JiTT terminology) which served two purposes: to prepare students for the activities to be developed in the following class, and to provide feedback to the teacher regarding difficulties students were having with the material. In each reading test, students were first required to study a book chapter or other reading material, and then asked to complete a web-based test available in the virtual course platform. Reading tests were due the day before class, usually at 8:30 p.m., in order to give the teacher just enough time to read over the student's responses and prepare the next lesson accordingly.

The structure of the reading tests was as follows. They contained three questions: two technical questions about the assigned reading and one last question for students to write their doubts or questions to the teacher. The technical questions were open-ended questions (see examples in Table 3, and the third question was always "What doubts or questions do you have about the reading?" In most cases, the grade of a reading test was calculated as 4 points for each technical question and 2 points for the third question, for a maximum grade of 10 per test.

Grading of the technical questions in the reading tests was based on evidence of reading rather than correctness. The latter would have been unfair since students had just read the material for the first time and they had not had a chance to discuss it, practice it, or clarify doubts in class with the teacher. By 'evidence of reading' we mean that the student shows enough evidence (details, explanations, arguments) that she actually studied the assigned reading, and is not just making up an answer. We emphasize this from the very first day of class in order to lower students anxiety (students fear that they will be evaluated and graded over subjects they don't master yet). Grading of the third question was based on effort, and no penalty was applied if students did not write any questions (but normally they did). Students were bought in the idea after explaining that writing their questions and doubts was actually beneficial to them because the teacher would tailor the class activities (exercises, discussions, etc.) accordingly, meaning that more in-class time was going to be dedicated to the parts of the subject that students found more difficult or challenging.

All the grading was made by the teacher since no teaching assistant was assigned to the course. To handle this workload, the teacher used the following strategy: the night before class she read over students' answers with the aim to detect misconceptions or unclear points that would drive the lesson next day (no grading was done at this point); then, when time permitted (typically within

Table 3: Technical questions examples.

Course unit	Sample technical question
1. Principles of Software Testing and Quality	Do you agree with the psychological approach to software testing proposed by Myers? If you agree, refer to the three aspects that seem most thoughtful and indicate how they are addressed in the organization where you work. If you disagree, indicate what aspects you disagree with, and offer an alternative approach.
2. Planning and Managing the Testing Process	Indicate whether your organization generates a Test Plan per software project. If the answer is yes: (a) estimate the percentage of similarity between the Test Plan generated by your organization and the one recommended by the IEEE standard, (b) mention two aspects recommended by the standard that are not taken into account in your organization's Test Plan, and (c) inquire how long, on average, does it take to generate the Test Plan. If the answer is no: (a) inquire why a Test Plan is not used and list the main reasons you found, (b) share your opinion on whether the existence of such a document would help your organization or not.
3. Types of Testing	Suppose you have been hired for system-level testing Microsoft Windows Media Player. What type of configuration, volume and performance tests would you do?
4. Test Design Techniques	Describe what are the advantages of each black box technique, both relative to other black box techniques and to the exhaustive approach.
5. Levels of Testing	Indicate which of the integration testing approaches is used in your organization and describe in broad terms how it is implemented. If integration tests are not performed within your organization, give a recommendation based on one of the approaches proposed in the reading and briefly explain why you believe that the chosen approach is most appropriate for your organization.

one or two weeks of submission), she graded the reading tests, giving feedback as necessary. Grading each student's reading test usually took between 10 and 15 minutes, demanding a great deal of time from the teacher. Part of the grading included checking the writing style, grammar, spelling of the answers, and providing feedback to help students improve their writing skills. Being able to write properly (in a clear, concise and correct manner) was deemed an important competence to build in a professional aspiring to get a master degree. Besides, we believe in the "thinking through writing" philosophy, which considers writing as a thinking tool, therefore we aim to help students become better writers as well as better thinkers.

Only open-ended (and sometimes essay-type) questions were used in the JiTT reading tests for the following reasons. First, we had previous bad experiences using closed-ended questions in the context of an undergraduate course where we found that those type of questions facilitated cheating among students and essentially failed to achieve its purpose (i.e., to get students prepared for the next class). Second, we have not yet detected any occurrence of cheating since we started using open-ended questions and students do come prepared to class. Third, we believe that open-ended questions help students exercise higher order thinking skills like analyzing, relating concepts, applying, evaluating or judging, thus achieving a deeper learning. Fourth, we believe that teachers should help students build up their writing skills so that when they graduate, they are able to write properly, and one way to do that is to have students practice writing and giving them feedback on how to improve.

4.2 Design of the In-class Component

The in-class component was partly organized around student common misconceptions or difficulties, extracted from the reading tests submitted by the students. Discussions and cooperative learning activities (including problem solving) were among the teaching strategies used in class. In-class activities heavily rely on the active participation of students. For example, rich discussions contrasting different viewpoints and work experiences were common since graduate students are mature and experienced professionals.

5 IMPLEMENTATION

We implemented JiTT during two semesters: the second semester of 2013 and the second semester of 2014. In the first instance, 16 students (5 women and 11 men) registered the course, while in the second instance, 18 students (4 women and 14 men) registered. In what follows, we offer additional details about course grading (particularly, the quantity and value of reading tests) in the two instances where JiTT was implemented.

5.1 Course Evaluation

The evaluation instruments used in the course were homework (including reading tests), quizzes, oral presentations on advanced topics, and lab reports. The weighting of each evaluation aspect towards the final course grade was similar but not identical across the two implementation instances, as shown in Table 4.

Quantity and weighting of reading tests. In the first instance, nine reading tests (warm-ups) were performed, accounting for 40% of the final course grade (this, plus a 5% quiz totalled 45% for homework and quizzes, as reported in Table 4). In the second instance, five reading tests were performed, accounting for 22% of the course grade (this, together with three quizzes of 4.3% each totalled 35% for homework and quizzes).

Table 4: Evaluation aspects and weighting.

Evaluation aspect	1st Instance	2nd Instance
Homework and quizzes	45%	35%
Advanced topic presentation	25%	20%
Lab reports	30%	45%

6 ASSESSMENT AND FINDINGS

The use of JiTT in the course was assessed from the students' and teacher's perspective. The students' perspective was obtained from a survey designed by the teacher. Participation in this survey was voluntary and anonymous. All of the students registered in the course completed the survey, in both implementation instances. The teacher's perspective was gathered from a qualitative assessment of strengths and limitations of the use of JiTT in the course.

6.1 The Students' Perspective

The survey contained 32 questions in the first implementation and 28 in the second implementation. We present here the results of the four most relevant questions related to JiTT. Other questions were too specific of the reading materials used in JiTT, hence were not deemed relevant to report. The rest of the questions were not related to JiTT (they asked students their perception on the course tools, educational platform, support materials, research of advanced topic, learning of course units and other general aspects of the course).

The first question was "Comment on what was on your initial reaction when the teacher announced that you had to read the material before class, and whether this first impression has changed or not (and how) now that the course has ended?" Table 5, shows sample student responses from both instantiations. Many students were initially worried because they thought that the readings were going to be too long, or that they would not have enough time to read and do the reading tests, or that the reading tests were going to be too hard. However, the majority of these students changed their mind as the course progressed, and ended up with a very positive attitude towards JiTT. A few students still complained that there was not enough time to do the readings and finish all the other course assignments. There were also some students who have had similar experiences in other courses, so JiTT was not new or surprising for them.

The second question was "When you read the material before class, does your learning improve, remains the same, or worsen compared to when you do not read the material in advance and are faced with it for the first time in class?" Table 6 summarizes student responses. We observe from this table that most students believe that their learning improves when they read the material prior to class.

Table 5: Sample student answers regarding their initial reaction to JiTT and how it evolved.

Instance	Comment
First	**Student 1**: I knew I had to be prepared before each class. I had to accommodate over time but finally in retrospect, I learned a lot and it was worth.
First	**Student 2**: In my case, I found reading material before class very successful to achieve an understanding of the topics to be studied. For many years I had the experience of reading topics before class, and I think it is advantageous and promotes study habits. With this form of study, we have to keep up with the readings, even bring doubts to the classroom.
First	**Student 3**: At first I thought that it was hard work, but then classes became enjoyable once the shock of the first class of this type passed, and I think it was an interesting way of working.
First	**Student 4**: It is a good idea since the student arrives with an idea of the topics that will be treated in class and can clarify his doubts. It's much easier to stay focused in class when one knows about the topic being treated than when one knows nothing. My impression is still the same now that the course has ended.
Second	**Student 5**: I worried a little because I thought it was going to be difficult due to my work... if the readings were long, it was going to take a long time. But this first impression changed because when I came to class, I understood the subject better.
Second	**Student 6**: My initial reaction was surprise and uncertainty since it was my first time with such activity. I worried about the reading tests being very complex and not having enough time to complete them. Now that the course has ended, that impression has changed since the readings were interesting and the reading tests very appropriate.
Second	**Student 7**: My first thought was that it would be a pretty heavy course and that its was going to be complicated due to the lack of time between college and work, however, readings were not very extensive and their quality made them easy to assimilate. As the course progressed, this impression changed and I actually found very helpful to read every week, I feel I learned more than in other courses.
Second	**Student 8**: I had already experienced it in other courses and I think it is helpful in better understanding the topic that will be studied in class. I think it should be kept in the course.

Table 6: Effect of reading before class on learning.

Effect on learning	First instance	Second instance
Improves	94%	89%
Remains the same	6%	11%
Worsens	0%	0%

Table 7: Suitability of JiTT for the course.

Suitable?	First instance	Second instance
Yes	94%	100%
No	6%	0%

The third question was "Do you think the educational strategy of reading the material before class, making reading tests, and using class time to clarify the most difficult points or to practice, is appropriate for this course?" Table 7 summarizes student responses. From this table, we see that a vast majority of students think JiTT was appropriate for the course. In this question, there was an optional space for students to write comments. Table 8 shows sample student comments to this question.

From Table 8 we can draw interesting points. First, some students (students 2 and 6) suggested to perform reading tests after class, once the teacher has clarified doubts. However, one of them actually acknowledged that doing it this way would detract students from reading in advance. Indeed, the main problem we see with this alternative approach is that students could stop reading before class thinking that it would enough to go to class and learn what is needed for the later test. But even if students really read before class, we still foresee problems: if we perform the reading tests at the end of the class period, they would take a away valuable time from other class activities; if we perform the tests outside the classroom, they would be like any other homework and would further discourage reading before class. The second interesting point from Table 8 is the opinion some students have about JiTT being useful only in certain scenarios, like theoretical courses (student 8) or subjects that cannot be easily misunderstood (student 4). We agree with the idea that JiTT is probably more useful in the theoretical parts of courses than in the practical ones. But we disagree with the idea that JiTT should not be used for complex topics that can be misunderstood, since it is precisely in these cases where reading in advance and clarifying doubts in class can be most helpful. The third and last interesting point from Table 8 is that even when students felt that the questions they wrote in the reading tests were not answered (student 7), they still saw value in the reading tests because they had to write and the exercise of writing helped them self-assess how well they had understood the material. This supports the "thinking through writing" approach.

The fourth question of the student survey was "Do you think reading tests and the type of questions used are an effective way to verify that the student did the assigned reading?" Table 9 summarizes student responses. We see that most students deem the reading tests (and its questions) effective at verifying that students actually read. In the first instantiation of JiTT, this survey question had an optional space for students to write comments. Table 10 shows sample student comments to this question (for the first instantiation only). We want to highlight three ideas from Table 10. First, it seems that the type of open-ended questions used really forced students to read (Students 1 and 2). Second, questions that related the readings to the organization where the student worked, seemed to have been well received (Student 3). Lastly, no matter how much the teacher tries to make questions hard to respond

Table 8: Sample student comments on the suitability of JiTT for the course.

Instance	Comment
First	**Student 1**: It really helps to make the lessons more pleasant (not boring) and in this way it captures the student's attention and one learns more.
First	**Student 2**: I agree with reading the material before class, but reading tests should take place after doubts have been clarified.
First	**Student 3**: We make better use of time and everything becomes clearer.
First	**Student 4**: I think it was very useful because it met the objective of clarifying difficult points. But it is not suitable for all subjects. For example, the student could be penalized if he misunderstands a complex topic and incorrectly answers the question in the reading test.
Second	**Student 5**: As we read in advance, we have an idea of what the class is going to be about, and therefore can take advantage of the class with comments, questions and deeper and specific criteria.
Second	**Student 6**: Yes, but I would change the time of evaluation until after class. Though I understand that in reality few of us would read the material without having the reading test before class.
Second	**Student 7**: I did not see value in the part of the reading tests where we had to write our doubts, as many of my questions were never answered. But I did find useful the part where we had to explain concepts in our own words, as it allowed me to check whether I had understood or not.
Second	**Student 8**: Yes, actually it is suitable for any theoretical course.

Table 9: Effectiveness of reading tests and type of questions.

Effective?	First instance	Second instance
Yes	88%	89%
No	12%	11%

from simply skimming, there will always be students who will attempt it (Student 6).

6.2 The Teacher's Perspective

Based on the teacher's prior experience with JiTT in undergraduate courses, an assessment of findings, strengths and limitations, was made on the use of JiTT at graduate level.

A major finding was that JiTT seems to be a good fit for graduate-level courses as well. Not only did students liked this teaching strategy, but also considered they learned more from reading before class. From the teacher's perspective, the same benefits (strengths) of using JiTT in undergraduate courses were observed here: students were more engaged in class, asked more interesting and deeper questions that enriched class discussion, and were ready to play an active role in class (e.g., by giving informed opinions, solving practice problems, etc.).

Table 10: Sample student comments on the effectiveness of reading tests and questions.

Instance	Comment
First	**Student 1**: Questions were mostly very specific, so that if one had not read the material, one could not answer the questions.
First	**Student 2**: Questions were not too specific or too general, forcing the student to read.
First	**Student 3**: Many of the questions asked in reading tests allowed to compare situations and aspects of the theory with the organization where the student works.
First	**Student 4**: I think it helps the student to learn the subject. Commenting on the topic lets you express your knowledge and this should be the spirit of the reading test.
First	**Student 5**: Not only do they prove that one studied the subject, but also help to confirm that what was read was properly understood.
First	**Student 6**: In some cases, due to time constraints, I had to read the questions first and then search the material looking for things that would allow me to answer the questions.

An additional benefit of using JiTT's reading tests (warm-ups) with graduate students was that they were better able to self-assess their understanding of the reading material. This can be deduced from the types of questions they wrote in the reading tests: they knew when they were not able to grasp a concept or technique, and they posed really good questions. Comparatively, undergraduate students are not able to formulate as good questions (they tend to write more general questions or comments like "I did not understand x concept" whereas graduate students tend to give more context, even examples). At the end of the day, the ability to self-assess helps the students because they get early feedback on things they should study more or pay more attention to, but also helps the teacher because she can nail down where the problem is and try to give more specific explanations or examples.

Last but not least, the use of open-ended questions in JiTT's reading tests had the extra benefit of helping students develop their writing skills since students practiced writing from 5 to 9 times throughout the course. Their writing was graded by the teacher and feedback on specific writing issues (grammar, spelling, style) was given to the students. With time, students' writing notably improved: ideas were better organized and clearer, answers were more focused, and there were in general fewer spelling errors and style issues. This benefit was not observable in previous JiTT experiences with undergraduate courses because the grading of reading tests was done by teaching assistants and they were not asked to check the writing of responses (only the contents).

On the down side, a major limitation of using JiTT in a graduate course was that all the grading had to be done by the teacher, and

the workload was too heavy. When JiTT was used in undergraduate courses, reading tests were graded by the teaching assistants, and yet a large amount of time was devoted to designing questions for warm-ups, reading students answers, and preparing for the next-day lesson. In the graduate-level experience, on top of designing questions, reading students answers and preparing classes, we had to grade and give feedback to the students. Grading an entire reading test (for all students) usually took about 3 or 4 hours. If you add the time required to grade other homework and lab reports, plus the time to actually prepare the class (or lab), you will soon realize that JiTT suffers from scalability.

7 CONCLUSIONS

We presented an experience report on the use of JiTT in a graduate-level Software Testing course, for two semesters. As part of the context, the course's objectives, contents, evaluation, students and teacher were described. The way we implemented JiTT was explained in detail, particularly the out-of-class and in-class components. The preparatory out-of-class component consisted of a web-based reading test due the day before class. The in-class component was partially organized around student common misconceptions or difficulties, extracted from answers to reading tests. Our experience with JiTT in the graduate course was assessed from the students' point of view by means of survey, and from the teacher's standpoint by means of an assessment of strengths and limitations.

Results from the students survey showed that (*i*) most students believe their learning improves when they read the material before class, (*ii*) a vast majority of students consider that JiTT is an appropriate teaching strategy for the course, (*iii*) many students think that reading tests are an effective way of verifying that students did the assigned reading.

Findings from the teacher assessment indicate that the use of JiTT had the following strengths: (*i*) students were more engaged in class, asking deep and interesting questions, (*ii*) students were able to self-assess their understanding of the subject and inform the teacher about it, and (*iii*) students were able to improve their writing skills. On the other hand, the major limitation of the use of JiTT in the graduate course was the heavy workload imposed on the teacher, especially with regard to grading.

In the future, we plan to explore ways to reduce the size of the readings in each reading test, since there was an underlying fear in students that readings were going to be too long and they would not have enough time to do the readings and complete the tests on time. We will also try to find a solution to the grading problem, and at first it appears that hiring a skilled teaching assistant (who could give feedback on the writing) might be a good solution.

REFERENCES

[1] T. Bailey and J. Forbes. 2005. Just-in-Time Teaching for CS0. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*. 366–370.

[2] R. Blake, K. Marrs, J. Watt, and A. Gavrin. 2003. Just In Time Teaching (Jitt): Using The Web To Enhance Classroom Learning. In *Proceedings of the American Society for Engineering Education Annual Conference and Exposition*.

[3] C.C. Bonwell and J.A. Edison. 1991. Active Learning: Creating Excitement in the Classroom. In *ASHE ERIC Higher Education Report No. 1. The George Washington University, School of Education and Human Development*.

[4] C.C. Bonwell and J.A. Edison. 2010. Promoting active learning through case driven approach: An empirical study on database course. In *StudentsâĂŹ Technology Symposium*. 191–195.

[5] T. Briggs. 2005. Techniques for Active Learning in CS Courses. 21 (December 2005), 156–165.

[6] P. Carter. 2009. An experiment with online instruction and active learning in an introductory computing course for engineers: JiTT meets CS1. In *Proceedings of the 14th Western Canadian Conference on Computing Education*. 103–108.

[7] Paul Carter. 2012. An Experience Report: On the Use of Multimedia Pre-instruction and Just-in-time Teaching in a CS1 Course. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 361–366.

[8] D. Cordes and A. Parrish. 2002. Active learning in computer science: impacting student behavior. In *Frontiers in Education*, Vol. 1. T2A1–T2A5.

[9] J. Davis. 2009. Experiences with Just-in-Time Teaching in Systems and Design Courses. In *Proceedings of the 40th ACM Technical Symposium on Computer science education*. 71–75.

[10] Judith S. Gurka. 2012. JiTT in CS 1 and CS 2. *J. Comput. Sci. Coll.* 28, 2 (Dec. 2012), 81–86.

[11] W. Li and G. Xin. 2012. Research and practice of program design teaching mode based on JiTT concept. In *7th International Conference on Computer Science Education*. 1897–1899.

[12] G. Lopez and A. Martinez. 2014. Use of Microsoft Testing Tools to Teach Software Testing: An Experience Report. In *Proceedings of the American Society for Engineering Education Annual Conference and Exposition*.

[13] S. Ludi, S. Natarajan, and T. Reichlmayr. 2005. An Introductory Software Engineering Course that Facilitates Active Learning. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, Vol. 1. 302–306.

[14] A. Martinez. 2012. Using JITT in a Database Course. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*.

[15] J.J. McConnell. 1996. Active learning and its use in Computer Science. In *Proceedings of the 1st conference on Integrating Technology into Computer Science Education*.

[16] J.J. McConnell. 2005. Active and Cooperative Learning: Tips and Tricks (Part I). In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, Vol. 37. 27–30.

[17] G. Novak and E. Patterson. 2009. An Introduction to Just-in-Time Teaching (JiTT). In *Just in Time Teaching Across the Disciplines*, S. Simkins and M. Maier (Eds.). Stylus Publishing, 3–24.

[18] A. Perez-Poch and D. LÃşpez. 2017. Just-in-time teaching improves engagement and academic results among students at risk of failure in computer science fundamentals. In *2017 IEEE Frontiers in Education Conference (FIE)*. 1–7.

[19] S. Simkins and M. Maier. 2009. *Just in Time Teaching Across the Disciplines*. Stylus Publishing.