

# An Agile Boot Camp: Using a LEGO®-Based Active Game to Ground Agile Development Principles.

Thomas D. Lynch, Michael Herold, Joe Bolinger, Shweta Deshpande, Thomas Bihari, Jayashree Ramanathan and Rajiv Ramnath

The Ohio State University, lynch.268@osu.edu, herold@cse.ohio-state.edu, bolinger@cse.ohio-state.edu, deshpande.36@osu.edu, bihari@cse.ohio-state.edu, jayram@cse.ohio-state.edu, ramnath@cse.ohio-state.edu

**Abstract** - Industry-practiced agile methods must become an integral part of a software engineering curriculum. It is essential that graduates of such programs seeking careers in industry understand and have positive attitudes toward agile principles. With this knowledge they can participate in agile teams and apply these methods with minimal additional training. However, learning these methods takes experience and practice, both of which are difficult to achieve in a direct manner within the constraints of an academic program. This paper presents a novel, immersive boot camp approach to learning agile software engineering concepts with LEGO® bricks as the medium. Students construct a physical product while inductively learning the basic principles of agile methods. The LEGO®-based approach allows for multiple iterations in an active learning environment. In each iteration, students inductively learn agile concepts through their experiences and mistakes. Subsequent iterations then ground these concepts, visibly leading to an effective process. We assessed this approach using a combination of quantitative and qualitative methods. Our assessment shows that the students demonstrated positive attitudes toward the boot-camp approach compared to lecture-based instruction. However, the agile boot camp did not have an effect on the students' recall on class tests when compared to their recall of concepts taught in lecture-based instruction.

*Index Terms* – Active learning, Agile concepts, Software engineering education, Workshops

## INTRODUCTION

Many software development groups have started using agile methods to provide more flexibility in their development process [1]. Because of this increasing use of the techniques in industry, it is essential that computer science graduates are aware of and understand the essential components of these methods [2]. These skills, along with an understanding of their strengths and weaknesses, enable them to participate in agile teams, and apply these methods in professional practice with minimal additional training.

Agile methods provide the flexibility to change the system during the development cycle as the requirements naturally change and evolve. However, teaching agile methods is challenging because students have preconceptions of the software development process that do not mesh with agile principles [3]. In addition, the classroom setting does not allow enough time to cover all aspects of agile development and does not allow constant contact with a customer to drive home the benefits of an on-site customer [4]. A lecture session on agile development principles leaves the students unpracticed in the concepts and unlikely to internalize the practices.

Our software engineering capstone course for undergraduate and graduate students dedicates only a single class to teach agile concepts. The amount of material that makes up the agile philosophy necessitates an innovative teaching approach. The approach must convey the information, and, through practice and experience, ingrain the concepts. Given these constraints, having the student develop a piece of software over multiple iterations is impractical. Thus, we need a surrogate experience that approximates the agile development process, but does not require as large of a time investment.

Based upon a multi-day workshop from one of our industry partners, we developed a short, single lecture, or “boot camp”, that uses LEGO® bricks to teach agile development principles. By using these toy bricks as a construction medium, we remove the complexity of writing code and replace it with a higher-level idea of system components. The students then design and build an object to meet specifications through multiple iterations. Each of the iterations showcases new agile concepts for the students to focus on and internalize.

There are many different agile development methods and each has unique practices [5]. However, there are several core concepts [6]. The boot camp focuses on the core terminology and concepts that are common across the various agile methods. These concepts include relative estimation, stories, business value, retrospective, refactoring, co-location, velocity, continuous communications and the customer as a collaborator. Other common concepts such as stand up meetings were not covered due to the time constraints and the impracticality given the setting.

**BACKGROUND**

Previous research has shown that active learning is an effective method for teaching complex concepts [7]. Simple concepts can be taught in a prescriptive method: step A, then step B, etc. A complex concept like agile development cannot be taught satisfactorily in the same fashion. There is only a conceptual set of methods and no prescription for them. Through this active learning process, the students learn these concepts, take ownership of their learning and become more involved in the process.

There have been efforts to transition software development courses from lecture-style courses into the active learning format [8]. The results of this study show that student grades and satisfaction were measurably increased through merely changing the presentation style. Through our incorporation of active learning in the form of this workshop, we hope to show the same improvements.

Papers written about teaching agile development processes tend to examine courses whose only subject is agile development processes taught over a period of days, weeks or quarters/semesters [9]-[10]. In contrast, this paper addresses teaching the process in a single lecture in a course on software development methodologies. The subject of this paper is the effectiveness of the methods described herein. If this method proves effective, it can be used to effectively teach other concepts in a similar way.

**BOOT CAMP DETAILS**

Prior to attending the boot camp, the students watch a video lecture on the agile process. They then explore the concepts during the workshop with the instructor. The students explore the concepts by meeting design requirements through the construction of an animal using LEGO® bricks. These arrangements and preparation allow three iterations in a 90-minute class. The students are divided into groups of five or six. Graduate students familiar with the concepts fill the dual role of coach and customer for each group. In the coach role, the graduate students help the students understand the concept in focus. As the customer, they are in charge of acceptance testing and feedback. We will refer to this role simply as “the customer” for the rest of this paper.

Each of the iterations teaches specific agile concepts using two approaches. Before each iteration, new concepts are explained by the instructor in a brief presentation. The students then use those concepts in the next iteration through the coaching and encouragement of their customer. At the end of each iteration, the students conduct a retrospective with their customer. After the individual retrospectives, the instructor directs a whole-class discussion and has each team highlight their successes and the problems they encountered. The problems allow the instructor to present new agile concepts for the next iteration.

Prior to the first iteration the students are introduced to relative estimation. They are told to take their story cards for the first iteration and find the easiest one, with “easy” being

defined as the least amount of time to accomplish the task. This card is given an estimate of one. Then, for each of the remaining cards, the students assign an estimate to the card. If the card takes twice as long as the easiest card, then it is assigned a two. Stories are assigned a three if the team estimates that it will take three times as long as the easiest card. If the story is estimated to take more than three times as long as the easiest card, it is deemed impossible for the current iteration. The instructor explains that the impossible cards will be broken into multiple, simpler stories with an estimate between one and three, and will be reissued in a future iteration.

Finally, the students select a set of stories they can complete in ten minutes. They then commit to their customer that they will complete that set of cards in this iteration. They put the remaining story cards to the side. The instructor tells the students that they can implement those stories too, after they finish their committed stories, time permitting.

*First Iteration*

The first ten minute iteration begins and the students furiously begin to construct their animals using the story cards to guide them. Even though there is no prize or announced scoring, the students work hard to complete as many stories as possible.

During the first iteration, the customers play a passive role by physically and socially distancing themselves from their teams. This was accomplished by not interacting with their team unless the team specifically asks them a question. The customers also reject one or more cards. The reason for the rejection must be something that the students could have addressed, had they asked questions of the customer, e.g., the color of the animal, the size of a specific feature, etc. The idea is to encourage the students to actively engage their customer in the development process during the future iterations.

During the retrospective after the first iteration, the instructor guides the discussion toward communications through co-location and confirmation of requirements through discussions with the customer. The rejected cards drive the discussion. The rejection of the completed work due to unknown requirements emphasizes the need to communicate with the customer during the iteration to avoid having stories rejected.

During the retrospective, the instructor teaches the teams how to graph their velocity. Velocity is defined as the sum of the difficulty estimates of the cards completed during the iteration. The discussion turns to how velocity over time provides a measurable progress indicator of the amount of work a team can accomplish in an iteration. The managers in charge of the development team would use this information for project planning and project completion estimates.

Before this first retrospective, the business value of the stories is not mentioned to the student, even though it is present on the cards (unlabeled). Thus the students picked stories in the first iteration based only on the difficulty

estimates. At this point in the retrospective, the instructor explains that the business value is given on each story card and is a reflection of the importance of the story to the customer.

The instructor has the students graph the business value they completed during the first iteration. The completed business value is suboptimal due to the lack of focus on this metric when the students picked their stories. The instructor encourages them to take business value into account in the subsequent iterations, explaining that customers are interested in the business value returned, not the amount of labor or difficulty of the work.

Also, the instructor tells the teams to use the velocity from the previous iteration – an agile concept known as “yesterday’s weather” – to gauge the amount of work they can accomplish in the next iteration. This is an important concept, as the story points provide a measurement of work relative to team performance, which increases the accuracy of time-to-completion estimates of the total project.

### *Second Iteration*

After these concepts are taught, the students estimate the cards for the next iteration and make their commitments to the customer. The instructor begins the second iteration. During this iteration, the customer directly engages with the team, though the customer does not direct the team. Instead the customer is simply an active participant in the process and stays co-located with the team by staying with the team throughout the iteration.

At the end of the ten minute construction period, the customers and teams conduct acceptance testing for each story. During this retrospective, the groups discuss the difference in the number of stories accepted compared to the previous iteration. Ideally, the groups gained for each story early approval from the customer and thus enjoy 100% acceptance rate of their implemented stories.

The stories for the second iteration present requirement conflicts that the team must resolve with the customer. The instructor guides the group retrospective to consider that this often happens on real projects. Often, customers unknowingly introduce requirements that are in direct opposition to previously implemented features. Thus, the discussion addresses the rework that the teams experienced and how they can mitigate this problem in the future.

Finally, the teams adds to their previous plot their velocity and business values for this iteration. After the second iteration the teams should have returned more business value compared to the first iteration, due to the emphasis on returning business value when they were selecting their stories.

### *Third Iteration*

Prior to the start of the third iteration, a customer-identified leader from each team is transferred to another team. This forces a reorganization of the teams and can

significantly change each team’s dynamics. This is the primary concept that the teams must consider for this iteration.

At the end of the third iteration, the retrospective discussion centers on the change to the team dynamics due to the transfer of their group leader. The concept emphasized is common in industry. Teams often lose key players. If the team has information silos, that is, a single person who possesses all of the knowledge for a specific component or system, the team has difficulty recovering from the loss of that person. To fight silos, agile teams engage in pair programming where two people work together on a single computer with one person doing the typing. By programming in pairs, two people inherently are very familiar with each part of the product, so the silo effect is mitigated [11].

### STUDY PROCEDURE

The instructors for this software engineering course use an inverted classroom concept for teaching the class, as described by [12]. The students watch a video lecture before each class. During the class, the students and instructor discuss the concepts taught in the recorded lecture. For the agile development processes class, instead of discussing the concepts, the instructor holds the previously defined boot camp workshop.

After the exercise, the students voluntarily completed a questionnaire. The students were told that the answers were anonymous and would not affect their grade for the course. They were also told that the purpose of the questionnaire was to get their input on the Agile LEGO® Workshop teaching method and to measure how many students learned the targeted concepts.

The questionnaire has three types of questions. Three of the questions ask the students their opinion on the teaching method and whether they think it is better than a lecture on the topic. Two of the remaining five questions deal with two concepts taught only in the recorded lecture. These questions measure whether the recording alone taught the information. The final three questions are drawn from a pool of five questions on concepts taught both in the recorded lecture and in the workshop.

The process and questionnaire will be continued with future classes to gather more data to further confirm or disprove the effectiveness of this teaching method. Our current sample consists of 78 students, split between two sections of the course, both of which were offered in the Winter quarter of 2011.

### MEASURING THE EFFECTIVENESS

We used both qualitative and quantitative measures to assess the effectiveness of this workshop technique. Our qualitative measures consist of interviews with students and instructors. Quantitatively, we analyze the results of a questionnaire and aggregated final exam results.

*Qualitative Measures*

An anecdotal qualitative measure of the boot camp's effectiveness is the direct observation of the students during the lab. We inferred that they enjoyed the boot camp since they were extremely engrossed in the design and development of the brick animal during the class. All students appeared to participate in the groups. After the class, several students requested that a boot camp be held every week in place of the normal lectures.

Another observation by the customers was that the students used the concepts taught on both the current iteration and the subsequent iterations. Their application of the agile concepts demonstrated some understanding of the material presented.

We also conducted interviews with students after the end of the course. These interviews were with three graduate students and one undergraduate student, who were all recruited via an email sent to all members of the classes.

Of the four students three rated the workshop a 5 (very effective) in the effectiveness of teaching agile development principles. These three students all noted that the simplicity of the LEGO® bricks allowed them to focus entirely on the concepts. They all said that it helped to actually practice the concepts that they had only previously heard about.

One student noted that while he felt that the potential effectiveness was a 5, the actual execution was a 2 (ineffective). This was because his class schedule was out-of-sync with the other class and he had not watched the recorded lecture on agile practices yet. He also suggested that the tasks on the story cards were too simple, which eliminated a learning opportunity in deferring work to the next iteration.

Through this small initial test of student satisfaction, we see encouraging results, though we are hesitant to generalize the results to all students. Due to the nature of the recruitment, we only had students who did well in the class volunteer for an interview. We feel that the results may be positively biased because of our sample. Further investigation is necessary to check for this possible bias.

Also, the graduate students who acted as customers for the group gained valuable practice with agile techniques and gained utility by being able to interact with the students. We observed that the graduate students were acting in a reciprocal learning capacity while acting as customers. They often had to explain the introduced concepts to their student teams and, thus, had to solidify their understanding in order to effectively communicate the information.

*Quantitative Measures*

We also used quantitative measures. The effectiveness of the process was quantified by summarizing the results of questionnaires given to the students at the end of the boot camp. The summarization measures three things:

1. How much did the students like the boot camp for teaching the development process?
2. How effective was the workshop in teaching the process as measured immediately after the class? This is a measurement of short term recall.
3. Did the boot camp do a better job of teaching concepts than just the video lecture alone?

The questionnaire contained three opinion questions. These questions measured how the students felt about the use of the workshop to teach the agile development principles. Two of the questions were at the beginning of the questionnaire and the last was at the end. The first two questions were stated such that an affirmative meant the student liked the process, while the last question was stated such that an affirmative indicated that the student did not like the process. This was incorporated to verify that the student truly liked or disliked the process and were not just marking answers without forethought.

The questionnaire also contained five multiple choice questions on agile concepts. Of the five multiple choice questions, two were concepts taught only in the video lecture. The other questions covered concepts taught in the video and in the boot camp. Each question had three to five choices and had only one correct answer. The other answers were structured to appear just as reasonable as the correct answer. The correct answers were not necessarily the longest or the shortest answer. The location of the correct answer varied. This was to thwart the "Longest is the strongest, when in doubt pick 'C'" strategy of multiple choice tests. The choices required the students to know the answer. Guessing made all the answers equally likely outcomes.

Finally, to measure long term effectiveness of the process, the students' final exams were summarized by calculating the student's grades on the questions dealing with agile development processes and comparing it to the student's overall grade on the exam. If the teaching method is effective, then it is expected that the student will score higher on the agile development questions than on the overall exam.

**RESULTS**

The first type of questions, the opinion questions measuring whether the students liked the teaching method and if they thought it was an effective method, is shown in Table I. As shown, an overwhelming majority of the students preferred the boot camp method for teaching these concepts and felt that this was an effective method for teaching this information.

The results of the second type of question, testing the students' knowledge of the agile concepts and testing if the boot camp increased this knowledge, are shown in Table II. The students did significantly better on four of the five questions that were presented in both the boot camp and the lecture compared to the information presented only in the lecture. There was one exception to this overall result. That question and its answers are shown in Table III. The correct

## Session F1H

answer is 'C'. Why the students missed this question requires further examination. Either the boot camp failed to teach this concept or another factor influenced the students to choose an incorrect answer to the question. Perhaps the correct answer was unintentionally deceptive: all other options are much more complex so perhaps they felt that the simple answer could not possibly be correct. This observation is purely speculative and requires deeper analysis.

TABLE I  
OPINION QUESTIONS: 1 IS DISAGREE AND 5 IS AGREE.

Question	Disagree .. Agree				
	1	2	3	4	5
Using Legos to teach Agile programming concepts is an effective method to teach these concepts compared to a lecture presentation.	0	3	3	25	47
Legos made learning this subject more interesting.	1	1	3	11	62
A lecture would have been more effective in teaching the Agile concepts.	30	29	11	5	3

TABLE II  
PERCENT OF STUDENTS GETTING EACH QUESTION CORRECT

Lecture Only		Lecture and Boot Camp				
39.7%	23.1%	81.6%	97.5%	24.4%	81.6%	85.0%

TABLE III  
THE CORRECT ANSWER IS 'C'.

Project velocity refers to ... (Mark the box corresponding to your answer.)				
A) How fast the project is getting done and is calculated by summing the number of stories completed divided by the number of sprints.				
B) The change in the number of stories a team gets finished over time.				
C) Is the sum of the estimates completed in a sprint.				
D) Is the change in the sum of the business value completed sprint to sprint.				
Answer	A	B	C	D
Number of Students	42	7	19	9

To test the long term effectiveness of this teaching method, we evaluated the answers to questions on the agile methodology on the final exams. The students' results on the agile question was compared to their results on the other questions as well as their overall results on the exam. On the final exam each student was required to answer questions 1,

2 and 3 plus answer any two of the remaining five questions. The agile question was the seventh question and in Figure 1 is labeled Agile Question. 27 out of 43 students answered the agile question. Their answers were given 0 to 10 points, with no fractional points awarded.

As shown in Figure 1, for the final exam, the agile boot camp did not have an effect on the students' knowledge of agile development concepts compared to their knowledge of the other information presented in the course. Their average score was 7.33 points out of 10 points with a standard deviation of 1.66 points. The median was 7 points. The average score on the exam was 73.86% with a median grade of 76.00% which includes students that did not answer the agile question. Of the students that did answer the agile question, their average and median grades without the agile question were slightly higher than with the agile question. They scored an average of 76.10% when the agile question is excluded from their score with a median of 77.50%.

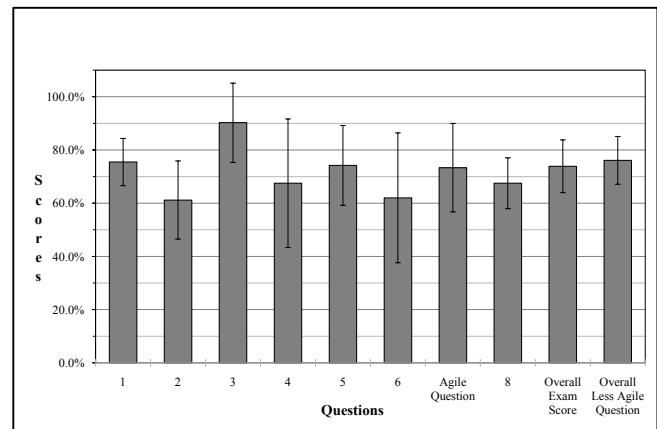


FIGURE 1  
COMPARISON OF AVERAGE AGILE AND FINAL EXAM SCORES INCLUDING STANDARD DEVIATIONS.

Through this preliminary analysis, we find these results are inconclusive. Figure 1 shows that student performance on the agile question appears to be worse than question 3 and on par with questions 1 and 5. We have not checked for statistical significance in the results. Perhaps statistical analysis would show some significance in our results, but practical significance is not evident.

Our instrument was flawed in that the agile question was not tuned to the content of the workshop. We feel that a direct assessment of the workshop content on the final exam may show an improvement over the questions that are focused on the content of the lectures. In the future, we will develop a better measure of student performance to check the efficacy of this technique.

While the academic significance of the workshop was not apparent, we do show an increase in student satisfaction with the workshop. This is valuable, as positive student attitude helps encourage them to put more effort into the course.

## CONCLUSIONS

The results of this study show that, while most of the students enjoyed active learning as incorporated in the boot camp, their apparent retention of the information was not significantly better than any of the other course information. The students did just as well on the final exam on concepts not taught using the boot camp workshop as they did on the agile development concepts taught in the workshop. Though the impact on learning outcomes appears to be negligible, we find that student enjoyment of the course is higher than in a strictly lecture-based class.

We note that our instruments for checking performance are flawed. The final exam did not accurately represent the workshop content and was more focused on the course content from a higher level. However, we did see value in the experiences we can provide our graduate students by using them as customers for the class.

In the future, we will improve our instruments to better check the efficacy of the technique. We feel that the technique has merit and we would like to investigate how we can use this technique in other courses and areas within this software engineering program.

One final anecdote: One student in an interview was asked if he knew about agile programming concepts. He told the interviewer about the boot camp. He noticed the interviewer had written in big letters "Lego Story" and circled it. He got a good job offer that evening from the company.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under NSF CCLI Grant No. 0837555 and the CERCS IUCRC Center for Enterprise Transformation and Innovation (CETI), supported by the NSF-IUCRC Program, Grant No. 0630188.

We thank ThoughtWorks, Inc. for the use of their LEGO® game.

## REFERENCES

- [1] Boehm, B., Turner, R., "Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods", *Proc. Of ICSE 2004*, pp. 718-719.
- [2] Hazzan, O., Dubinsky, Y., "Why Software Engineering Programs Should Teach Agile Software Development", *ACM SIGSOFT Software Engineering Notes*, Vol. 32, No. 2, 2007, pp. 1-3.
- [3] Mugridge, R., "Challenges in Teaching Test-Driven Development", *Lecture Notes in Computer Science*, Vol. 2675, 2003, pp. 1015-1018.
- [4] Mugridge, R., MacDonald, B., Roop, P., Tempero, E., "Five Challenges in Teaching XP", *Lecture Notes in Computer Science*, Vol. 2675, 2003, pp. 1011-1014.

- [5] Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., "Agile Software Development Methods: Review and Analysis", *VTT Publications*, Issue 478, 2002.
- [6] Beck, K., et al. "Manifesto for Agile Software Development", Agile Alliance. Retrieved 2011 Mar 29.
- [7] Briggs, T., "Techniques for Active Learning in CS Courses", *Journal of Computing Sciences in Colleges*, Vol. 21, No. 2, 2005, pp. 156-165.
- [8] Sowell, R., Gill, C., Chamberlain, R. D., Grimm, C., Goldman, K. J., "The Active-Learning Transformation: A Case Study in Software Development and Systems Software Courses", *Journal of Computing Sciences in Colleges*, Vol. 25, No. 5, 2010, pp. 165-172.
- [9] Boehm, B., Turner, R., *Balancing Agility and Discipline*, Boston: Addison-Wesley, 2003.
- [10] Stapel, K., Lübke, D., Knauss, E., "Best Practices in eXtreme Programming Course Design", *Proc. of ICSE 2008*, pp769-776.
- [11] Lindstrom, L., Jeffries, R., "Extreme Programming and Agile Software Development Methodologies", *Information Systems Management*, Vol. 21, No. 3, 2004, pp. 41-52.
- [12] Gannod, G. C., Burge, J. E., Helmet, M. T., "Using the Inverted Classroom to Teach Software Engineering", *Proc. of ICSE 2008*, pp. 777-786.

## AUTHOR INFORMATION

**Thomas D. Lynch**, Ph.D. Student, Department of Computer Science and Engineering, The Ohio State University, lynch.268@osu.edu.

**Michael Herold**, Ph.D. Student, Department of Computer Science and Engineering, The Ohio State University, herold@cse.ohio-state.edu.

**Joe Bolinger**, Ph.D. Candidate, Department of Computer Science and Engineering, The Ohio State University, bolinger@cse.ohio-state.edu.

**Shweta Deshpande**, Graduate Student, Department of Computer Science and Engineering, The Ohio State University, deshpande.34@osu.edu

**Thomas Bihari**, Senior Lecturer, Department of Computer Science and Engineering, The Ohio State University, Senior Consultant, Nationwide Insurance, bihari@cse.ohio-state.edu.

**Jayashree Ramanathan**, Director, C.E.T.I, Department of Computer Science and Engineering, The Ohio State University, jayram@cse.ohio-state.edu

**Rajiv Ramnath**, Director, C.E.T.I, Associate Professor of Practice, Department of Computer Science and Engineering, The Ohio State University, ramnath@cse.ohio-state.edu