

Team-Based Learning in: "Applications Programming"

1. Course Situation

- Department: Information Technology
- Subject: Java Programming
- Level: Upper division
- Students: N = 14, Male, IT majors, juniors and seniors, and taking this course as part of a programming concentration
- Time Structure: 2 sessions per week, 2 hours per session, 10-week course
- Course and classroom factors: The students had to create programs using various applications such as NetBeans, ant, JUnit, JavaBeans, and CVS. Every student had a computer available to them in the classroom.

2. Learning Goals for the Course

- To be able to integrate all topics of the course into a stable, well tested cohesive program
- Be able to work together on a team to create a complex integrated program
- To be able to efficiently work together in a team
- Recognize the power of team learning
- Become a self-directed learner
- Be able to share ideas and knowledge with other students
- To be able to critically analyze programs and code
- To be able to create a project that meets its stated goals
- Create a project that is useful

3. Reasons for Changing to Team-Based Learning

- Previous way of teaching and problems encountered:

This course used a power point presentation to teach the topics and was followed-up by a brief in-class assignment. A weekly programming assignment and three take-home tests were assigned. The problem with this method of instruction was that the students were bored, all the work was individual, and there wasn't a group project.

- What made me think that team-based learning might be an alternative worth trying:

For several years, I had successfully infused cooperative learning techniques into introductory programming courses, so I understood the importance of group work which was missing in this course. Since this was an upper level

course I also wanted to go beyond cooperative learning to further promote self-direction and deep learning. I had also taught courses that contained group projects, but there were always dysfunctional groups and students with bad group experiences. The TBL book convinced me that team-based learning was the technique that I could employ to overcome these problems and to transform the groups into teams. TBL's claim that it creates an environment where the team can outperform the ability of their best member also deeply resonated with me.

4. Changes Made

- Changes made:

RAP: Weekly reading assignments were given from a custom textbook that incorporated all of the applications covered in this course. One RAT was given each week and contained ten multiple choice questions. I only lectured on topics when they had questions after taking the RAT.

In-class Activities: New activities were created that attempted to keep students switching between teamwork and individual work. I wanted them to start working together and helping each other while still coding on their own. This mixture was important to me because I felt that it simulated the workplace where programmers spend a lot of time coding alone while relying on their co-workers for help, advice, and guidance.

I also wanted them to feel comfortable asking questions from their teammates, so they had to follow these steps whenever they had a question:

- 1) Ask one team member for help, if it remains unanswered then
- 2) Ask the team for help and if it still remains unanswered then
- 3) As a team, ask the instructor for help

Project: One of the biggest deficiencies in the previous approach of instruction for this course was the lack of a group project. In this version of the course, group projects were assigned where they had to integrate every application discussed in the course into a cohesive program. In an attempt to keep the teams from just dividing up the work and programming independently on the project, additional weekly in-class activities were provided to make them work on various aspects of the project as a team. These activities included generating program requirements, project design, and usability testing. In addition to submitting their final projects for a graded evaluation, the teams also had to conduct a poster session where they discussed and defended their final project.

External Evaluation: I invited one of our department's industrial advisory board members to attend the poster session and perform a qualitative analysis of each project. This evaluation was also part of the grade for their projects. The purpose of the external evaluation was multi-fold, they could get a feel of the type of assessments that are used in the "real world", it gave them added incentive to think that a potential employer would be evaluating their work, and it also gave our board member first hand experience to see the quality of our students' work. The board member was a formal VP of a major software corporation and currently runs his own consulting business.

- Results: (in terms of the general operation of the course).

On the first day of class I told the students that this was a new teaching method for me and that they would be given several anonymous summative evaluations throughout the course. These surveys elicited the students' opinions on every aspect of the course, had them rate all the exercises and techniques, and asked for general comments. We discussed their comments and suggestions together and I adjusted the exercises and assignments several times throughout the quarter. I believe that this was instrumental to the success of the course.

Some of the meticulous, well planned out exercises turned out to be disasters, but based on the students' critical feedback, the subsequent activities were a lot more successful at increasing the students' mastery of the topics. The mid-course adjustments were also appreciated by the students which made them more tolerant of mistakes. The students confirmed this feeling on their end of the year comments when they stated that the course started slow with several problems but it ended up great.

5. Examples of Team Assignments

The in-class exercises varied depending on the subject. Some started with a team brain-storming activity where they answered detailed questions as a team or as a short overview of the assignment. These questions or overviews were specifically designed to help them get started on the assignment. Next, they worked on the code individually (sometimes on portions of the same program and sometimes on separate programs). Once everyone finished, the team members got together and discussed their results. At this point they combined their programs or turned in one program. The group members were graded based on their team's combined program.

For the project, each team had to create a program that incorporated all 10 topics. One of these topics was 2D Graphics where each team member had to create an animation and integrate it into the project. Every animation needed to have the same look-and-feel as the other member's contributions. This required

a strong design that Java Beans (another application discussed in the course) enforces. Similarly, an attempt was made to have all the applications closely integrated so the students would naturally work together on the project.

6. Impact

A. On Student Learning and Performance:

Comparing final grades, the non-TBL course had: 11 A's, 3 B's, 1 C, 0 D's, and 1 F. The TBL course had: 12 A's, 2 B's, and 0 C's, 0 D's and 0 F's. So grade-wise, the TBL students outperformed the non-TBL students.

But perhaps the best indication of the impact on student learning is the comments provided by the Industrial Advisory Board Member.

First of all, here are his overall impressions:

- Each of the teams seemed to genuinely have good chemistry...it seemed like the students liked each other and respected the idea of respective strengths and leveraging skills
- They were competitive in a healthy good natured way...and i think they were interested in each teams interpretation of the task
- All ended up approaching the problem from very different points of view...eg Team 1 thought that extensibility was key...Team 2 was into online code confirmation....and Team 3 was into modularity and animation....all valid, all good stuff
- I liked the enthusiasm of the teams, they were proud of their work and that is very healthy...I think they invested a lot preparing for the presentations...even the story boards were good...
- I saw nothing negative....and frankly am quite impressed by the level of teamwork demonstrated in the class...projects in the real world are built by work teams who learn to rely on one another...these guys demonstrated that in spades...if they walk out with a grasp of how teams need to be productive and interoperate then they will have learned a great deal more than simple programming...congratulations

And his individual team impressions:

- About Team 1...very adult view of the need for design flexibility, you could tell that getting that right might have caused them to restart a couple of times...proud of their work and a fun group to interact with

- About Team 2...most professional, all spoke, all were confident, seemed rehearsed...i was impressed with how they demonstrated enthusiasm without appearing disingenuous
- About Team 3...they were the most casual, the project was well done, especially the animation...I think there was a greater disparity between the level of contribution/perhaps enthusiasm of the team members but...great respect for respective strengths...I admired how they talked about individual contribution

He was so impressed with the student's work that he convinced a fellow board member to also attend the next poster session.

B. On Student Attitudes

A survey was given to the students at the end the course. They were asked to rate how effective the overall course had been in achieving various student learning goals. The following lists the percentage of students that felt the course was effective in reaching the stated goal (the survey contained a 5-point Likert scale):

- 84% - Working together as a team
- 92% - Self-Directed Learner
- 100% - To be able to share ideas with members of your team
- 84% - Team-based Learning method of learning
- 92% - Favorable experience with the team

The response to the course was overwhelmingly positive and only one person stated that they strongly disliked working in teams. Typical comments included:

- Good team work and communication
- The team was great
- My team was better than any other team in the history of teams not counting the 2004 Red Sox
- Good group of guys to work with, all wanted to succeed, everyone communicated well
- The team was a lot of fun and worked very well together
- Group was fun to work with and entertaining
- Really funny ambiance, nice guys, i really had some pleasure exchanging and talking with them

C. On the Teacher

I loved the experience and it was a complete change over the previous method of instruction. One of the students had attempted to take the course the previous year and stated that the TBL method was far superior to the way it was

previously taught. The experience between the two methods of instruction was like night and day. The students were highly engaged and seemed to genuinely enjoy the class. The students embraced the TBL concept right from the beginning. On the first day of class while they worked on the grade-weighting exercise, the majority of students strongly supported the need to maximize the teamwork portion. They were already convinced that teams can outperform their best member. They chose the maximum allowable percentages for the team portions of the grading.

I will definitely use TBL for this course the next time I teach it and I am looking forward to seeing how the students perform now that I have experience using TBL.

7. [Related Publications](#)

- None...yet.

8. [Contact Information](#)

Keith Whittington
102 Lomb Memorial Drive
GCCIS, IT Dept. 70-2527
Rochester, NY 14623
Phone: (585)475-5363
Email: kjw@it.rit.edu
FAX: (585)475-2181

9. [APPENDIX: More Detailed Information about Team Assignments](#)

Here is one of the in-class exercises:

JUnit In-Class Exercise

Background

In the conference, you will find a program named Toy.java that lets the user choose a toy. You will create JUnit tests for this program to ensure that it works. Do NOT do this exercise in NetBeans, use the installed version of JUnit.

Overview

You will be performing the following steps:

- 1) Team discussion and planning
- 2) Individual work
- 3) Team conference and consolidation of work
- 4) Class discussion and comparison of each team's work

NOTE: The final half-hour of the class will be dedicated to going over each team's solution. You must submit a team solution described below before the end of class.

Procedure

1) Entire Team – Preliminary Work

To ensure you don't waste a lot of time at the beginning of the exercise, answer the following questions.

Where is the junit.jar file located?

What is the classpath on your machine?

What is the name of the graphical JUnit testing program?

How do you create a test case class?

What is the command to run JUnit tests with the JUnit graphical interface?

2) Entire Team - Planning

Do the following before starting to code so you will be able to efficiently complete the assignment and involve all members of the team.

- Download the Toy.java file from the FirstClass conference.
- Examine and discuss with your team what the class does.
- List ALL the tests that should to be done.
- Everybody choose one test to code.

NOTE: If you have more tests than members, let the member who finishes first grab the next test case and code it

3) Coding - Individual

Do the following steps:

- Every person code your designated test case.
- Test your code and ensure that it works
- If you have questions, following this procedure:
 - Ask help from a teammate
 - Both of you ask for help from your entire team
 - Entire team ask the instructor
- Do the following if you finish before your teammates
 - Take one of the remaining test cases, code it, and test it
 - If all test cases are being worked on, check on your teammates and see if they need help

4) Consolidation – Team

NOTE: The following steps HAVE to be completed before 5:20

- When all test cases are completed, combine the tests into one file
- Run the new file and verify that all test cases work.
- Hand in one copy of your written documentation from the first 2 steps
- Download your file to the conference

5) Group Comparisons

In class, we will compare the results of each team and you will be given instructions at that time.