

PEER-LED TEAM LEARNING THE EXPERIENCE OF LEADING

FROM PEER-LED TEAM LEARNING TO PROFESSIONAL WORK EXPERIENCES

BEILI WANG

When Professor Janet Liou-Mark at New York City College of Technology, CUNY, recommended to me that I take the Peer Leader training course in 2001, I had unspoken questions about it: “What are the benefits for taking the time out from my already very busy schedule to take this one-credit course that will not count toward my degree, not to mention all the extra homework that needs to be done?” I told myself: “To give the Professor a good impression. To make friends with the students from different majors and schools. To learn some skills that can help me tutor other students.” These were my answers to my own questions at that time. Little did I know the skills, such as problem-solving and communication that I learned from the course, actually benefited me a great deal as I pursued my A.S., B.S, and M.S. degrees in Computer Science (CS) and obtained employment as a Tech Analyst at JPMorgan Chase. In a recent conversation with Professor A.E. Dreyfuss and Professor Liou-Mark, I am grateful to them for helping me hear myself out and to realize how those skills helped me in real-life situations. I hope by reading about my experiences below fellow students will be helped to realize the relationship between classroom work and professional work.

When a CS student recently asked me about how to study in CS because he found himself stuck in his programming homework, I understood him completely, for the same frustration and helplessness haunted me for many sleepless nights when I was a CS student. To my own surprise, the general advice I gave to him on why and how to fix his bug was actually the same steps in any problem-solving situation.

First, understand the problem and objectives fully. Read the problem carefully and analyze what has been asked, what is the input and what should be the output, what is the final result needed to be achieved. As a student, if you have any questions about what problem needs to be solved, you can always ask professors to clarify it. As a tech professional, I work on projects with a group of people at work. For any small or big project, before I start any work, I have to understand all the details in a requirement document which is written by a business team on what clients want and what the final result should be. If I have any question about the requirement document, I will contact the business team or the clients to clarify their expectations.

Second, think and design. Really take time to think, to break a big problem into small problems, to design all the steps and details. This step should take the most time and energy during the problem-solving process. In CS studies, there are a lot of tools to utilize to help us transfer the abstract ideas into executable steps, such as flow charts, UML diagrams, Pseudo code, etc. At work, when the tech

team receives the requirement document that the business team generated, the tech team will review it and have many meetings to discuss the design. All the tech team members put their expertise together in the design meetings. During the meetings, they break the huge requirement document into small sub-requirements to small problems, draw and write on the blackboard in the conference rooms, or use software tools in the virtual conference rooms to help them to think. The result of all the design meetings is a detailed technical specification document which states all steps of an optimal solution to achieve the final results that clients demand.

Then, implement. Sometimes, students will jump into this step because of time constraints. In fact, this step should never be the first or the second step. Students always make mistakes by not taking the understanding of the problem and thinking through each step seriously, turning to the programming too soon. The usual mindset is at least I start coding and eventually the result will present itself. It is rarely true. And, almost always, the students will find themselves stuck with a bug in the program and cannot go further. And, when stuck, it actually requires more time to think, to fix, and to change other parts of the program. Sometimes, it even requires redoing everything to fix one bug. At work, this kind of mistake will waste a lot of manpower, money, postpone the whole project schedule, and worst of all, damage the reputation of the company, and affect future business decisions. If the first and second steps are done right, the implementation step will be much easier because it is just to follow all the details in the design.

Finally, test and review. To make sure the solution is correct, sample data can be used to test and see if the expected result is reached. In the workplace, when the tech team finishes implementation and testing, the application will be delivered to the business or the clients to have user testing before releasing the final product to the real market. Always, there are bugs found during testing because human errors are inevitable; so the testing is a critical step and should not be ignored. After each project is finally delivered, the work is still not completely done until a “lessons learned” meeting is over. In the final review meeting, everyone who participated in the project will give their thoughts on what went well and what went wrong. The result of the review meeting will be put into a formal document. And, the review document will help the team on other similar projects to come.

It is also not the first time that students asked me if CS students really need to know a lot about Math, or if Math education in the classroom is going to help them in “real life.” From my experiences, a person can do fairly well in programming and not necessarily do very well in Math. However, my suggestion to students is you need to learn Math well to be successful in the profession you choose. Math is a fundamental subject that teaches us how to think logically and how to practice our problem-solving skills. Problem-solving skills are definitely a must in the job market. In many job interview sessions, the interviewer will ask the applicant to solve problems in order to evaluate the job seeker’s logical thinking process and problem-solving skills. Some of the problems can be really hard and have no definite answer. The key is not to get the right answer but to think logically and to think it out loud. When you need to solve a problem during a job interview, the important tip is to think and speak at the same time. Let the interviewer know your thinking process. If you just think for a long time with no words coming out of your mouth, the interviewer might suspect you have poor problem-solving skills and do not know how to attack a problem.

Another key skill in the professional work environment is communication. In the workplace, people almost always work in a team environment. Sometimes, it seems that you work alone on some

problems or research, but at the end, you need to do a presentation to let other people know the result of your work and to answer their questions. You need good communication skills to present your ideas clearly without misunderstanding. A group of people needs to work efficiently to achieve a common goal, and good communication skills are necessary. These include how to relate to other people in the team, how to listen patiently, how to restate the conversation to show your understanding, how to communicate your ideas in a clear way, and so on. In CS, on many occasions, we need to look at the codes written by other people to fix bugs or to make necessary up-to-date changes. When we code, commenting on our own code in a clear understandable way can save a lot of trouble later on. Good commenting practice is also a part of communication skills. In the workplace, you are expected to use “people skills” and communication skills daily to get the work done effectively and efficiently.

Besides problem-solving and communication skills that have benefited me, is always keeping an open mind. An open mind - to take the Peer-Led Team Learning course, to learn different things, to learn as much as possible, and to participate in different activities - really opened so many opportunities for me. Keeping an open mind enables me to see there are a lot of people who eagerly want to help me to become successful in my life. I sincerely hope by reading about my experiences and learning from them other students will be helped to be successful in their future endeavors as well.

Beili Wang
Former Peer Leader
New York City College of Technology, CUNY

Cite This Article as: Wang, B. (2012). From Peer-Led Team Learning to professional work experiences. Peer-Led Team Learning: The experience of leading. Online at <http://www.pltlis.org>. Originally published in *Progressions: The Peer-Led Team Learning Project Newsletter*, Volume 12, Number 1, Fall 2010.