UCI Certificate in Teaching Excellence / CIRTL Associate Level Attainment Assignment
Aftab Hussain
June 26, 2020

TAR (Teaching as Research) Project Planning Page

After completing this worksheet, the participant will be able to:

1. Find background knowledge and evidence of effectiveness about a teaching practice of interest in their discipline appropriate to the project they choose (below)
2. Utilize their knowledge of the full-inquiry cycle and the foundational knowledge they have gathered to generate a simple proposal for a teaching as research project.

Part 1: What would you change to improve your class?

In this section, choose a course that you have experience with as an instructor or TA. An introductory course will provide you the most flexibility in discussing this project with colleagues.

What course will you modify? ____ICS53 Principles of System Design__________________

Quick course description:
Circle: Majors / non-majors   Circle: Intro (freshman) / lower division / upper division
# of students = 50   Circle: Discussion sections / labs / none

What is a skill that students have difficulty mastering in this course?

Skills students have difficulty mastering in this course are 1. debugging code and 2. collaborative programming (dividing tasks and merging their partial coding contributions into a single working program.)

Part 2: What is known about effective teaching related to this subject and skill?

In this section, use your library database of choice to explore education research or teaching scholarship writing in your discipline. Examples might include "reading quiz" "chemistry" "active learning" "deep reading" "writing skills" "history." If you aren't finding anything using this method, search this list of pedagogy journals for ones in your discipline: http://cetl.kennesaw.edu/teaching-journals-directory

In the space below, list the references for two articles that are at least somewhat relevant to your possible project, and describe the findings of the articles.

The articles

[1] David L. Largent. 2016. Measuring and understanding team development by capturing self-assessed enthusiasm and skill levels. ACM Trans. Comput. Educ. 16, 2, Article 6 (February 2016), 27 pages. http://dx.doi.org/10.1145/2791394

[2] Marina Prvan and Julije Ožegović. 2020. Methods in Teaching Computer Networks: A Literature Review. ACM, Trans. Comput. Educ. 20, 3, Article 19 (June 2020), 35 pages.
https://doi.org/10.1145/3394963

Findings in [1]

Largent in this work gives extended empirical results of his previous work that explored how coding teams form and interact in a college level computer science course. He compared his findings with Bruce Tuckman's theory of small group development that involves forming, storming, norming, performing, and adjourning. In a nutshell, a team's stage in their development progress can be determined by a combination of their enthusiasm and skill levels. In Largent's work, students from two Ball State University departments, in selected courses, were periodically given questionnaires relating to their level of skills and enthusiasm (a five-point Likert scale was used to obtain the responses). They found that the Tuckman model can be a useful tool for teaching and monitoring team development, especially towards computer science courses. Interesting specific findings include the observation that students self reported skills decline as the projects move forward. This could be attributed to the fact that many were overestimating their skills and did not entirely understand the requirements of their projects.

Findings in [2]

In this article, the authors provide a survey of methods used for teaching Computer Networks (CN). They used a content based analysis that involves categorization. They tracked the different categories of teaching CN such as using visualization techniques, using virtualization techniques, practice hands-on lab, and more sub-categories. One of their interesting classes of findings was the one in which they evaluated how these different teaching techniques compared with respect to their educational effectiveness. In this regard remote labs have been found to be particularly effective in two studies because of the task-focused nature of virtual labs and the minimal distractions involved in them (This would need to be further investigated in my opinion as student involvement in tasks may vary if they are unsupervised). Also, based on their survey, they concluded that one cannot compare the learning/emotional parameters enhanced by every classified method or decide which class is better, and further studies would need to be carried out in this direction.

Part 3: Creating a TAR Project

The purpose of a "teaching as research" project is to give a future faculty member practice in thinking about teaching as something that can be regularly improved and assessed. Your practice project does not need to be carried out, but it should be a sensible, interesting project that *could* be carried out by a normal graduate student or postdoc with a normal workload.
Keeping this in mind, answer the questions below, implementing what you have learned from your research in Part 2.

What assessment does the class have to measure student ability at this particular skill? Do you need to add an assessment that measures this ability?

The skills I wish to improve is 1. debugging code and 2. collaborative programming.

Towards skill (1), we have in-class quizzes conducted during discussion sessions, where students are presented with multiple-choice questions, and other short questions where they have to identify the correct output, point out the error in a code segment, or write short code snippets that fulfills a given function.
Towards skill (2), we have weekly lab sessions, and to improve their collaborative skills, we ask students to work in different teams every week.

Although we do not need to add assessments to measure these abilities, I think we would need to modify them in order to help them further improve skills 1 and 2. I discuss this in the next segment.

What activity will you change or add to attempt to improve this skill?

The first activity I like to change are the quizzes. Such tests can easily be prepared using Google Forms, which allows random question shuffling (which automatically discourages any cheating while taking the quiz). The students can take the tests online, and each student could get tailored feedback automatically, since we can prepare our answers and explanation beforehand.

The second activity that needs changing in my opinion is how we conduct the lab sessions to improve their collaborative programming skills. Too often, in my experience, I have seen teams where only one person does most of the work, and the other(s) watch(es). I thus think there is a need to capture the contributions of each student working in a team towards reaching the same goal. We may do this by dividing a lab task into different parts, and ask teammates to split the tasks among themselves. However, even if a teammate fails to do a task he/she picked, the task would need to be completed, in which case the other teammate(s) would need to complete it. In order for the instructor to capture who did what at the end of the day in each team, we may ask the students to fill out a form stating what tasks their teammates *completed successfully,* where any successfully completed task is one that can be used towards fulfilling the overall goal of their assignment.
We can carry out this activity by randomizing the teammates each week. From the student feedback forms we can easily discover which students are doing less in weekly assignments. That way, we would be able to pair up students with those whose performance were similar, which could inherently force lower performers to take more initiative in their work and encourage them to rely more on their own abilities, and consequently improve their collaborative programming skills.

Is there a way to create a control group that doesn't get this particular activity?

For the first activity (the quizzes) we may use a control group of students who do the quiz on paper as we usually do. Thereafter, we may compare their debugging skills with those who took the online version. We may assess these skills in one of the midterm exams.

For the 2nd activity (the group tasks) we may use a control group of 3 to 4 teams, whom we do not ask to fill up a peer assessment form, for the first 2-3 weeks. Then we dissolve this control group, and allow them to work with other students as randomly assigned. From then onwards, we may compare the quality of peer reviews of these control students with the rest. In my opinion, there may be some in the control group who may eventually exhibit slightly lower performances in their peer assignments.

How much time do you estimate this experiment would take to complete?

Prep before class	Implementation during class	Grading / scoring after class has ended	Analysis and writeup
Preparation of the online quiz would take at least 1 to 2 hours for each lab discussion session: it would include time taken to prepare the questions and the solutions as well. The lab assignment prep should take at least 1 hour, designing a programming problem with correct output. The questionnaire that would take into consideration the splitting of the assignment into subtasks would require around 15 minutes. For convenience we may also use Google Form to create this questionnaire.	The online quiz may be as long as 15 minutes. The lab task may be as long as 35 to 40 minutes.	The online quiz grading would be automatic (instantaneous). The lab task would involve demonstrations of the programs by the teams, and comparing their results with the desired output. For 20 groups, this should take between 20-25 minutes for each lab session.	Assessments from the online quiz could be accumulated in an online spreadsheet, as this is automatically done by Google Forms. Toward analyzing the student peer reviews for their levels of collaborations, we could again easily have all the information in a spreadsheet if we distribute online Google Form questionnaires. Merging results from multiple forms and analyzing the progress of each student should be very easy to do (1 hour at most). Finishing write-ups from the analyses should take around 1 hour.