

William Paterson University
College of Science and Health - Department of Computer Science

Fall 2015 – Spring 2017 Assessment Cycle
Analysis of the Program’s Student Outcome Assessment Data

Objective: S4:

Work effectively as part of a team in a software or hardware project.

ABET’s Related Student Outcomes: (d)

Members: Cyril S. Ku (Chair), John Najarian, Bogong Su

Date: November 21, 2017

Updated On: January 30, 2018

A. Analysis of the Assessment Data

For the assessment period of Fall 2015 to Spring 2017, this student learning outcome was assessed in the following courses: CS 3410, CS 3500, and CS 4800.

This analysis report is for S4 (Work Effectively as part of a team in a software or hardware project). All these course CS 3410, CS 3500, and CS 4800 involved team projects. For CS 3410, it was hardware-oriented while CS 3500 and CS 4800 were software-oriented. CS 3410 only had one team who failed the project in three semesters. Most of the students in CS 3410 and CS 3500 were in the “Adequate Ability” or higher level and so these two courses satisfied S4 objective quite well. For CS 4800, the comments for the two semesters in this assessment report were the same and indicated that some students did single person project. One student did a sloppy job on the project.

B. Suggestions for Improvement

CS 3410: As the instructor indicated, project guideline and evaluation criteria need to be modified to prevent plagiarism in the future although 2 out of 3 semesters, this plagiarism problem did not occur.

CS 3500: In general, although all the projects were in the “Adequate Ability” or higher level, none of the projects was in the “High Ability” level. Perhaps assign a simpler project (smaller scale) but ATM system is not a complicated one and still no team got the “A” level of performance. Maybe more UML examples can be given to demonstrate the concepts.

CS 4800: As indicated by the instructor, he was somewhat reluctant to teach CS 4800 again due to the nature (seminar type) of the course. The recommendation is for this instructor talk to other instructors of this course and share experience about how this course should be conducted.

C. Improvement Implemented

The improvement suggestions have been carried out in the Fall semester of 2017. For CS 3500, starting in the Spring 2018 semester, the instructor put more grade percentage to project (35%) instead of 20%. He put more emphasis on projects and eliminated homework in this course. The homework did not really help the students as most students copied each other anyway.

D. List all the “performance level/frequency/percentage” tables and their sources.

- a. Faculty Course Assessment Report: CS3410, Spring 2016

Data Collected: Each student’s level of performance on oral presentation and written report.

Method of Collection: Team work (two students) is required for a digital logic design project and to make an oral presentation and to produce a report on it. He/she then receives a numerical grade on the contents and knowledge of the project presented and a numerical grade on his/her report from the instructor of the course.

Performance Levels	Frequency	Percentage
No Ability	0	0%
Some Ability	4	19 %
Adequate Ability	5	24 %
More than Adequate Ability	0	0%
High Ability	12	57 %

Observations:

- 1) In general the quality of all projects were much worse than F15, half projects were OK however less creative. Two teams’ projects were unable to demo correctly during the oral presentation. Two teams’ design were copied from an incorrect design on web and their circuits do not work correctly. One team’s is copied from a design of F15. These three teams got low scores.
 - 2) Plagiarism problem in project has not occurred in recently years, unfortunately many students have this problem in S16. I will modify the project guideline and evaluation criteria to prevent it.
-

- b. Faculty Course Assessment Report: CS3410, Fall 2016

Data Collected: Each student’s level of performance on oral presentation and written report.

Method of Collection: Team work (two students) is required for a digital logic design project and to make an oral presentation and to produce a report on it. He/she then receives a numerical grade on the contents and knowledge of the project presented and a numerical grade on his/her report from the instructor of the course.

Performance Levels	Frequency	Percentage
No Ability	1	5%
Some Ability	0	0 %
Adequate Ability	0	0 %

More than Adequate Ability	7	35%
High Ability	12	65 %

Observations:

- 1) All teams work seriously. One student quitted early and didn't do the project.
 - 2) Five teams select same topic, 'digital clock' which is a relative easy project. Those teams work independently and no plagiarism problem.
-

c. Faculty Course Assessment Report: CS3410, Spring 2017

Data Collected: Each student's level of performance on oral presentation and written report.

Method of Collection: Team work (two students) is required for a digital logic design project and to make an oral presentation and to produce a report on it. He/she then receives a numerical grade on the contents and knowledge of the project presented and a numerical grade on his/her report from the instructor of the course.

Performance Levels	Frequency	Percentage
No Ability	0	10 %
Some Ability	4	10 %
Adequate Ability	8	38 %
More than Adequate Ability	2	10 %
High Ability	2	10 %

Observations:

In general the quality of all projects are much worse than F16, non-team has creative work. Four teams work hard. Two teams' design are too simple.

Plagiarism problem in project does not occur.

d. Faculty Course Assessment Report: CS3500, Spring 2016

Data Collected: Each student's level of performance on participating in the software team project.

Method of Collection: There were 3 interim project reports (requirements – Use Case Models; specifications and design – Class Diagrams and Communication Diagrams; implementation – C++ or Java codes for the classes) and 1 final report. The final report included the revised materials from the 3 interim project reports. The frequency listed below is at the team level.

Performance Levels	Frequency	Percentage
--------------------	-----------	------------

No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	1	25%
Adequate Ability (Level of performance of C)	1	25%
More than Adequate Ability (Level of performance of B)	2	50%
High Ability (Level of performance of A)	0	0%

Observations: The project was a bookstore administration system. Except for 1 team, the quality of the projects was quite good. CASE tool was not required for the project but 1 team use Visual Paradigm for some of the UML models.

e. Faculty Course Assessment Report: CS3500, Fall 2016

Data Collected: Each student's level of performance on participating in the software team project.

Method of Collection: There were 3 interim project reports (requirements – Use Case Models; specifications and design – Class Diagrams and Communication/Sequence Diagrams; implementation – C++ or Java codes for the classes) and 1 final report. The final report included the revised materials from the 3 interim project reports. The frequency listed below is at the team level.

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	2	25%
Adequate Ability (Level of performance of C)	1	12.5%
More than Adequate Ability (Level of performance of B)	5	62.6%
High Ability (Level of performance of A)	0	0%

Observations: There was one team which was very close to the High Ability Level. The project was an automated teller machine system (ATM). Except for 2 teams, the quality of the projects was pretty good. CASE tool was not required for the project but 1 team used Visual Paradigm and 1 team used ArgoUML for some of the UML models.

f. Faculty Course Assessment Report: CS3500, Spring 2017

Data Collected: Each student's level of performance on participating in the software team project.

Method of Collection: There were 3 interim project reports (requirements – Use Case Models; specifications and design – Class Diagrams and Communication/Sequence Diagrams; implementation – C++ or Java codes for the classes) and 1 final report. The final report included the revised materials from the 3 interim project reports. The frequency listed below is at the team level.

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	1	14%
Adequate Ability (Level of performance of C)	3	43%
More than Adequate Ability (Level of performance of B)	3	43%
High Ability (Level of performance of A)	0	0%

Observations: The team project was called Student Information and Administration System. There was one team that was very close to the No Ability Level. I discussed with this team extensively to make sure they understand the tasks they had to tackle. It just happened that the make-up of this team consisted of members who did not do well in the exams.

g. Faculty Course Assessment Report: CS4800, Spring 2016

Data Collected: The students were divided into 2 person teams to teach a chapter of an ethics book, not only to teach them ethics and getting them used to speaking in front of a group, but also to see how they worked together if they had a partner.

Method of Collection: The above assessment table included the teamwork as part of the measure of their research project and presentation. They gave periodic progress reports orally and elaborated then on their distribution of tasks and level of participation of each member of the team. Because they voluntarily chose their team membership and were cohesive teams, they shared efforts and coordinated well. No one was riding on the backs of others. That they in groups chose their topics and genuinely were actively engaged had much to do with this. I thought before the course that I would have to force load distribution but they did it spontaneously.

Observations: Most of the students took the path of you do that part and I do this part without interfering with each other. They only coordinated well and at several stages, not just putting the slides together. Among popular activities, some were making computer games and some were designing software based on PHP, GIS, SQL, or other web-scripting language, database, or API's. A tiny minority did single person projects, one because their work was research into Security Issues. Another 1 student (out of the 21) actually did a miserable job, with a sloppy code that did not work well (though the overall design showed promise) and a hastily pasted together presentation, showing deficient dedication of time, effort, and practice (tragic). In general, minimal prodding or supervision effort was required to induce the teamwork.

Personal Note:

I personally find it disconcerting to teach a course that lacks a specific subject, a focused specific domain of CS, with the associated reasoning, particular set of methodologies, problem sets, or some form of knowledge domain. Ethics and Scientific Method (even CS-specific) just lack the same feel as direct inquiry. Both are general and abstract, not as excitingly specific (like a group-theorist

having to study category theory.... just not the same). It is unnerving. I am still somewhat reluctant to teach CS4800 again. The unstructured nature is hard to come to terms with; this assessment may indicate a phobia about oblivion or lack of control. On the positive side, the students worked cohesively as integrated team members, as the course intended. However, with no lengthy final exam or dissemination of CS lectures (except for the ethics component), I feel something is amiss. The student experienced professional growth and developed confidence as senior CS students but I am left confused. Perhaps I need to learn more from what happened as there are lessons here I have not fully comprehended or incorporated into my axiomatics or mindset. The “Scientific Method applied to CS”-component is important, as it teaches students that CS actually uses it. It clearly inculcates that we are an inductive Science, as well as a deductive one.

h. Faculty Course Assessment Report: CS4800, Fall 2016

Data Collected: The students were divided into 2 person teams to teach a chapter of an ethics book, not only to teach them ethics and getting them used to speaking in front of a group, but also to see how they worked together if they had a partner.

Method of Collection: The above assessment table included the teamwork as part of the measure of their research project and presentation. They gave periodic progress reports orally and elaborated then on their distribution of tasks and level of participation of each member of the team. Because they voluntarily chose their team membership and were cohesive teams, they shared efforts and coordinated well. No one was riding on the backs of others. That they in groups chose their topics and genuinely were actively engaged had much to do with this. I thought before the course that I would have to force load distribution but they did it spontaneously.

Observations: Most of the students took the path of you do that part and I do this part without interfering with each other. They only coordinated well and at several stages, not just putting the slides together. Among popular activities, some were making computer games and some were designing software based on PHP, GIS, SQL, or other web-scripting language, database, or API's. A tiny minority did single person projects, one because their work was research into Security Issues. Another 1 student (out of the 21) actually did a miserable job, with a sloppy code that did not work well (though the overall design showed promise) and a hastily pasted together presentation, showing deficient dedication of time, effort, and practice (tragic). In general, minimal prodding or supervision effort was required to induce the teamwork.

Personal Note: I personally find it disconcerting to teach a course that lacks a specific subject, a focused specific domain of CS, with the associated reasoning, particular set of methodologies, problem sets, or some form of knowledge domain. Ethics and Scientific Method (even CS-specific) just lack the same feel as direct inquiry. Both are general and abstract, not as excitingly specific (like a group-theorist having to study category theory.... just not the same). It is unnerving. I am still somewhat reluctant to teach CS4800 again. The unstructured nature is hard to come to terms with; this assessment may indicate a phobia about oblivion or lack of control. On the positive side, the students worked cohesively as integrated team members, as the course intended. However, with

no lengthy final exam or dissemination of CS lectures (except for the ethics component), I feel something is amiss. The student experienced professional growth and developed confidence as senior CS students but I am left confused. Perhaps I need to learn more from what happened as there are lessons here I have not fully comprehended or incorporated into my axiomatics or mindset. The “Scientific Method applied to CS”-component is important, as it teaches students that CS actually uses it. It clearly inculcates that we are an inductive Science, as well as a deductive one.
