

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

Fall 2018 – Spring 2019 Assessment Cycle

Analysis of the Results of the Evaluations of the Assessment Data
of
the Program Student Outcome

Program Student Outcome:

S8: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

Curriculum Committee Subgroup: Software Analysis

Members: Cyril S. Ku (chair), John Najarian, Gilbert Ndjatou

Date: May 29, 2019

Updated On: _____

A. Analysis of the Results of the Evaluations of the Assessment Data

For the assessment period of Fall 2018 to Spring 2019, this program student outcome was assessed in the software engineering course (CS 3500) from two semesters (one section in Fall 2018 and two sections in Spring 2019).

For this course, 25, 26, and 24 students respectively took CS 3500 and 16 of them (which represents 21%) had less than adequate ability (D performance level). The majority of the students (nearly 80%) were in the A to C level of performance. As mentioned in the observation sections of the Faculty Course Assessment Reports and past experiences of the instructor, the preparedness of the students reflected in the performance of the students. In general, CS students are more prepared than CIT students.

B. Suggestions for Improvement

Since CIT students need CS 2550 (Foundations of Information Systems) as prerequisite and CS students need CS 3420 (Data Structures) as prerequisite, the background required for this course should be even for CS and CIT students. This course uses object-oriented concepts to design a software system. It is critical that students have object-oriented programming experience. CS students have CS 2400 (Computer Science II) before they can take CS 3420. Therefore, CS students are well-prepared for this course and performed much better than most of the CIT students. In the current pre-requisite structure for CIT students, all they need is CS 2550 then they can take CS 3500. CS 2400 should be included as a pre-requisite for them. Software development experience is important to do software design. When CS 2400 is included as a pre-requisite, which means students will have at least CS 2300 and CS 2400, plenty of programming practices.

C. Improvement Implemented

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

- a. Faculty Course Assessment Report: CS 3500-60, Fall 2018

Data Collected: Course grade for the entire course.

Method of Collection: There were four major activities for the team project (35%), exam #1 (15%), exam #2 (15%), exam #3 (15%) and 1 comprehensive final exam (20%).

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	5	20%
Adequate Ability (Level of performance of C)	9	36%
More than Adequate Ability (Level of performance of B)	7	28%
High Ability (Level of performance of A)	4	16%

Observations: The project and presentation scores were team-based (all the team members in a team got the same score if there was no dispute among team members) and the examinations were of course individual work. Starting this semester, I eliminated homework for this course because it did not help the students. I put more percentage on the project and I added one more exam to let students to have one more chance to improve their grade. The overall grades for this class did improve compared with last year’s grades. A general observation: the grades for the CS students were generally better than the grades for the CIT students.

- b. Faculty Course Assessment Report: CS 3500-01, Spring 2019

Data Collected: Course grade for the entire course.

Method of Collection: There were four major activities for the team project (35%), exam #1 (15%), exam #2 (15%), exam #3 (15%) and 1 comprehensive final exam (20%).

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	6	23%
Adequate Ability (Level of performance of C)	12	46%
More than Adequate Ability (Level of performance of B)	6	23%
High Ability (Level of performance of A)	2	8%

Observations: The project and presentation scores were team-based (all the team members in a team got the same score if there was no dispute among team members) and the examinations were of course individual work. This is the first section of the software engineering class and I had another section in the evening. Only two “A” students in this class while the evening section had seven “A”s. However, the course materials were the same and the project assignment was comparable. The grade distribution reflected more on the quality of the students than the way the class was conducted, the course materials, and the project assignment.

- c. Faculty Course Assessment Report: CS 3500-60, Spring 2019

Data Collected: Course grade for the entire course.

Method of Collection: There were four major activities for the team project (35%), exam #1 (15%), exam #2 (15%), exam #3 (15%) and 1 comprehensive final exam (20%).

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	0	0%
Some Ability (Level of performance of D)	5	21%
Adequate Ability (Level of performance of C)	8	33%
More than Adequate Ability (Level of performance of B)	4	17%
High Ability (Level of performance of A)	7	29%

Observations: The project and presentation scores were team-based (all the team members in a team got the same score if there was no dispute among team members) and the examinations were of course individual work. This is the second section of the software engineering class in the same semester. The overall performance of this section was better than the first section.

In the upcoming semester, I planned to change the percentage of the grades as follows: project (35%), exam #1 (16%), exam #2 (16%), exam #3 (16%), and final exam (17%) since the final exam was not really accumulative and concentrated on the materials after exam #3.
