Course and Instructor Evaluation of Educational Quality - Fall 2014 End of Term

Instructor Standard Report

Joel Coffman

Foundations of Software Engineering
Computer Science - EN.605.401.31
This Summary of Results provides an overview of the results of your course's evaluation by the students who enrolled in your course and completed a Course and Instructor Evaluation of Educational Quality. It supplements the detailed results presented in the Instructor Standard Report that follows. Three indices covering students’ overall assessment of courses and instructors have been developed and are described below. You will also see similar scores for your program in the same modality (i.e., online, on-site, or partnership courses), and for the whole Engineering for Professionals (EP) program in the same modality. Program and EP results are based on the prior three semesters.

- "Instructor Subscore" = A score on the 0%-100% percentage scale that represents the results of the instructor-oriented questions on the evaluation form. For each possible response (strongly disagree; disagree; neutral; agree; strongly agree), a value is assigned. Every response is tallied for the instructor-oriented questions, divided by the total possible score (i.e., if all students select “strongly agree”), and then expressed as a percentage. Instances where the student does not respond or selects “not applicable” are not counted in this calculation.

- "Course Subscore" = A score on the 0%-100% percentage scale that represents the results of the course-oriented questions on the evaluation form. For each possible response (strongly disagree; disagree; neutral; agree; strongly agree), a value is assigned. Every response is tallied for the course-oriented questions, divided by the total possible score (i.e., if all students select “strongly agree”), and then expressed as a percentage. Instances where the student does not respond or selects “not applicable” are not counted in this calculation.

- "Course and Instructor Score" = A score on the 0%-100% percentage scale that represents the results of all questions on the evaluation form (both Instructor and Course items). For each possible response (strongly disagree; disagree; neutral; agree; strongly agree), a value is assigned. Every response is tallied for all questions, divided by the total possible score (i.e., if all students select “strongly agree”), and then expressed as a percentage. Instances where the student does not respond or selects “not applicable” are not counted in this calculation.

<table>
<thead>
<tr>
<th></th>
<th>Your Course’s Total</th>
<th>Program Total for On-Site Courses</th>
<th>EP Total for On-Site Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor Subscore</td>
<td>90%</td>
<td>81%</td>
<td>84%</td>
</tr>
<tr>
<td>Course Subscore</td>
<td>84%</td>
<td>81%</td>
<td>81%</td>
</tr>
<tr>
<td>Instructor and Course Score</td>
<td>88%</td>
<td>81%</td>
<td>84%</td>
</tr>
</tbody>
</table>
1. Instructor Related Questions

THE INSTRUCTOR...

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Number Responding</th>
<th>Response Percents</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Appeared knowledgeable about course content</td>
<td>18</td>
<td>88%</td>
<td>1%</td>
</tr>
<tr>
<td>B Was prepared for class</td>
<td>18</td>
<td>83%</td>
<td>11%</td>
</tr>
<tr>
<td>C Displayed enthusiasm for teaching</td>
<td>18</td>
<td>78%</td>
<td>17%</td>
</tr>
<tr>
<td>D Used appropriate examples to clarify points</td>
<td>18</td>
<td>67%</td>
<td>28%</td>
</tr>
<tr>
<td>E Demonstrated relationships between theory and practice</td>
<td>18</td>
<td>67%</td>
<td>22%</td>
</tr>
<tr>
<td>F Motivated me to learn</td>
<td>17</td>
<td>47%</td>
<td>47%</td>
</tr>
<tr>
<td>G Gave understandable presentations</td>
<td>18</td>
<td>78%</td>
<td>22%</td>
</tr>
</tbody>
</table>

Program and EP results are based on the prior three semesters.
### THE INSTRUCTOR...

#### Provided useful responses to questions
- Number Responding: 18
- Response Percents: `72%` Strongly Agree, `22%` Agree, `8%` Neutral
- Instructor Mean: 4.7
- Standard Deviation: 0.6
- Program Mean (F2F): 4.1
- EP Mean: 4.3

#### Respected me as a student
- Number Responding: 18
- Response Percents: `83%` Strongly Agree, `17%` Agree
- Instructor Mean: 4.8
- Standard Deviation: 0.4
- Program Mean (F2F): 4.5
- EP Mean: 4.6

#### Responded to my inquiries in a timely manner
- Number Responding: 18
- Response Percents: `83%` Strongly Agree, `5%` Agree, `2%` Neutral, `6%` Strongly Disagree
- Instructor Mean: 4.6
- Standard Deviation: 1.0
- Program Mean (F2F): 4.4
- EP Mean: 4.4

#### Helped me develop new skills
- Number Responding: 18
- Response Percents: `67%` Strongly Agree, `22%` Agree, `11%` Neutral
- Instructor Mean: 4.6
- Standard Deviation: 0.7
- Program Mean (F2F): 4.3
- EP Mean: 4.3

#### Graded my performance fairly
- Number Responding: 18
- Response Percents: `76%` Strongly Agree, `11%` Agree, `8%` Neutral, `5%` Strongly Disagree
- Instructor Mean: 4.6
- Standard Deviation: 0.8
- Program Mean (F2F): 4.4
- EP Mean: 4.5

#### Provided grades in a timely manner
- Number Responding: 18
- Response Percents: `61%` Strongly Agree, `28%` Agree, `11%` Neutral
- Instructor Mean: 4.4
- Standard Deviation: 1.0
- Program Mean (F2F): 4.2
- EP Mean: 4.3

#### Provided useful feedback on assignments (e.g., homework)
- Number Responding: 18
- Response Percents: `56%` Strongly Agree, `28%` Agree, `28%` Neutral
- Instructor Mean: 4.2
- Standard Deviation: 1.2
- Program Mean (F2F): 4.1
- EP Mean: 4.2

#### Provided useful feedback on assessments (e.g., exams, papers, projects)
- Number Responding: 18
- Response Percents: `56%` Strongly Agree, `28%` Agree, `28%` Neutral
- Instructor Mean: 4.2
- Standard Deviation: 1.2
- Program Mean (F2F): 4.1
- EP Mean: 4.2

---

*Program and EP results are based on the prior three semesters.*
### 1. Course Related Questions

<table>
<thead>
<tr>
<th></th>
<th>Response</th>
<th>Number Responding</th>
<th>Response Percents</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The required reading materials contributed to my learning</td>
<td>18</td>
<td>50%</td>
<td>33%</td>
</tr>
<tr>
<td>B</td>
<td>The course assignments (e.g., homework) contributed to my learning</td>
<td>18</td>
<td>56%</td>
<td>39%</td>
</tr>
<tr>
<td>C</td>
<td>The course assessments (e.g., exams, papers, projects) measured what I learned in the course</td>
<td>18</td>
<td>61%</td>
<td>33%</td>
</tr>
<tr>
<td>D</td>
<td>The course content was relevant to my career</td>
<td>18</td>
<td>61%</td>
<td>33%</td>
</tr>
<tr>
<td>E</td>
<td>The course content challenged me</td>
<td>18</td>
<td>39%</td>
<td>44%</td>
</tr>
<tr>
<td>F</td>
<td>The course workload requirements were reasonable</td>
<td>18</td>
<td>39%</td>
<td>39%</td>
</tr>
</tbody>
</table>

*Program and EP results are based on the prior three semesters.*
I loved the format of the course, weekly exams and the project making up the bulk of the course grades. It forced a deeper understanding of concepts and forced us to demonstrate our understanding by solving a technical problem.

The quizzes helped to reinforce what was learned along the way, but they required an arduous amount of writing. Additionally, there was far too much memorization of terms and definitions required.

Aside from issues I mentioned in the professor comments, the assignments served as a well-structured incremental approach to completing the course project.

The course project had its good and bad moments. I did not like how the game that were supposed to be implementing required major and implied changes in order to even be feasible. That is, being instructed to implement Bananagrams when the only plausible implementation for such a small team in such a small time-frame was the Solitaire variation, and that are were supposed to discover that on our own. A more complete list of requirements would have helped guide the project. I know we were supposed to be learning how to deal with confusing or incomplete software requirements, but bad requirements is completely different than no requirements. We were simply told “Build Bananagrams” and that we could stray from the standard set of rules as we saw fit. We could have built Monopoly (or planned to, at least) and we would arguably have remained within those constraints.

Other than the above, the project flowed much more smoothly than I had expected. This is especially with respect to the "twist" of each team having to finish another team's project. I expected every team, including ours, to crash and burn from that point forward, but the exact opposite happened. We and every other team were able to complete the project we were given and do it well!

Great class all around. Weekly quizzes are an excellent way to force you to do the readings and review previous class material every week.

Course applies well to my current career.

I felt that the readings from The Mythical Man Month were interesting but outdated. I think I would have enjoyed them more reading it as leisure and not for a class. The beat readings for me were web or print articles like those from Joel Spolsky.

I also feel that the project organization could have benefited more by pairing teams together for the entirely of the class. We had design reviews, code reviews, and a twist which made us complete another teams project but I wish those had all been with the same team.

I like the course material and it showed me a new perspective to software engineering. I learned a ton of historical information, but at the same time learned how that relates to the field today. I was able to relate it to my workplace directly. I believe it was also clear on what we were expected to know from each lesson, which helped me learn tremendously. I didn't feel overwhelmed and I could focus on important points.

The semester long group project was a great way to help put into practice what we were learning. While the idea of having to complete another team's assignment halfway through the semester was a little frustrating at first, the idea was brilliant. In the real world you seldom get the privilege of working on projects where you have a say on the requirements and design of the software or you may have to maintain a horribly documented confusing mess of code someone else created. This project gave us great insight how important careful planning and forward thinking for change is to a project.

The Bananagrams project was a bit boring, I would have preferred to not spend so much time wrestling with Swing.

Project was just OK, though I understand it's difficult to give an appropriately scaled project. Java Swing is not fun. The project switch was interesting, and I think gave a good feel for the idea that requirements change. I appreciated how the course was focused more on the doing rather than the theory.

The choice of material was excellent. The only issue is that it can be hard to juggle weekly material with the project. There can be a lot going on.

I liked that this course focused on certain aspects of software engineering that classes I'd had in the past on the subject overlooked to an extent. Most classes in software engineering focus on documentation. This one touched on that, but also delved into some subjects other software engineering classes barely touch, like reviewing code, refactoring, testing, etc.
I think the Bananagrams project and the twist on the project (where teams swapped code) were good examples of how the real software world works. However, I think there are two places where the project could be improved.

1. In order to scope the project to a reasonable amount of work, we had to modify the rules of Bananagrams a lot to the point where the game was totally different. This led to final projects that fulfilled the specifications, but were really difficult to play.

2. The project twist was well executed the way it was done, but left some groups with broken projects. The code we received passed the demo, but had design gaps missed in code reviews that showed the previous group hadn't really thought things through. In the real world, it is common to inherit a project that wasn't originally yours, but usually it at least works (even if the code is a mess or buggy). Thus, I propose that the groups should have to complete a project (as in totally done) and then the twist is that the groups switch projects and have to add new functionality. This way, the groups get code that at least works.

For the course content, I would have liked to see a tie in to "what can I do now?" Many concepts (management style, project planning, project structure, resources for developers, etc) I completely agree with, but have little say in until I am another 15 years into my career.

Dr. Coffman is a very effective teacher that utilizes teaching techniques to reinforce the most important information. His usage of weekly refreshers and quizzes ensures that students retain knowledge of the principles taught throughout the course. Additionally, his project reinforces some of the abstract concepts taught throughout the course. His lessons on software engineering design patterns greatly piqued my interest, forcing me to go back and analyze software projects I've done at my job and critically think of how to apply them and make the software better. I hope to see a course offered through EP that focuses on this concept.

First, if I can just say something about this survey tool. It's awful. I had written a whole bunch of stuff in this box and hit "Next" only to be told that my session had timed out. I lost everything I wrote. I don't want to write it all over again, so I'll try to hit on the major topics below:

Though the professor appeared knowledgeable and enthusiastic, he often complained about how difficult it was to teach a class, especially with regard to grading.

The professor did not distribute assignments timely or coherently. Homework was assigned during class, but the necessary materials or instructions to get started on it were usually not distributed until one or two days later. This effectively gave us significantly less than a week to complete assignments. This was especially stressful because most assignments involved working with a team, which is already difficult to coordinate. Also, assignment instructions were usually confusing, incomplete or misleading.

The professor actively solicited feedback from the students in order to remedy issues such as the above. This was probably the best aspect of the course. This helped greatly in making the course less hostile for the students.

Prof Coffman was an excellent instructor. He was enthusiastic and knowledgeable and really inspired me to get excited about software development. What he does is probably what other instructors should emulate when teaching in school for professionals environment, he was flexible, fair, and not afraid to make adjustments to his teaching style if it was not working. I really enjoyed his class, learned a lot, and would take it again.

The course was very easy to understand and follow. I was aware of what I was expected to know to complete the quizzes and I'm confident that I know what material I will need to study for the final. I enjoyed having articles separate from the main course textbook to read for homework. I could tie almost everything being taught to an example of an event that has happened at my current software engineering job. Though I didn't really like the project twist towards the end of the course, I understood why it was done and felt it was a good example of what may happen in the real world when programming a product.

The only complaint I could think of is that he seemed too open to put class decisions to a vote. I would prefer that schedule and assignments not change too much because the majority thinks they are too hard or too easy. I think an instructor should take input from students but I don't think a plurality of opinion from students should be the decider. Still, this is minor and overall I do think the class was run fairly.

Joel Coffman was a great teacher. I like how the class helped us with memory retention of all in-class material over time. It greatly helped me learn and figure out how I was doing in the class. The amount of work needed for the class varied from week to week, and the reading at the end seemed like a lot to be mixed with the final. It was still very manageable though. I'm not sure how effective switching the team projects was, but it was an interesting idea. Some things that were stated as historical are still being used by my company, but maybe our older concepts are what work for us. I thought that was interesting. The material we learned was great and informative. I would recommend this professor to friends and would probably take him again if available for a future class.
I really enjoyed taking this class.

The weekly quizzes were a great way to keep on top of the reading and worked as a knowledge check for topics you may need to focus more on. I definitely prefer weekly quizzes to a midterm as it is much less stressful and daunting.

The quick reviews at the beginning of each class were also a great help.

Overall I would have to say that Professor Coffman did an excellent job teaching this course.

There is too much information covered in the lectures. I felt like we went over material just for the sake of saying we went over it. More examples will be very helpful instead of just memorizing definitions. Most of the assessments were how well you could memorize the material. May be that how its suppose to be but more real life examples would be nice.

Professor Coffman is a new teacher with a lot of new ideas for assignments, but is willing to admit when they do not all fit into the schedule and adjusts the schedule accordingly.

Listens to his students and puts a lot of weight behind their opinions to be fair and better himself as a professor, I greatly respect him for this and wish more teachers did this as well.

He's infectiously enthusiastic about the course.

Explains the concepts well and in an easy to understand manner. Presents well.

Instructor is very good at adjusting the schedule to make sure students are comfortable with it.

Your method of teaching was good, you were interesting to listen to and seemed to know what you're talking about and you answered questions well.

One way I think you could be more effective is by re-working some of the slide decks for the class. Many had slide after slide of blocks of text and became very boring and ineffective in teaching the material. Breaking it up with more examples instead of text would help.

The instructor was clearly knowledgeable and enthusiastic about the course.

Interesting lecturer. Very respectful of student's work obligations and went above and beyond to accommodate them. Answered questions well and made the lecture interactive rather than pure instructor talking.