Software Development (201830: Spring 2018)

Instructor: **Derbinsky, Nathaniel**
Subject: **CS**
Catalog & Section: **4500 03**
Course ID: **37083**
Objectives:

Enrollment: **31**
Responses Incl Declines: **27**
Declines: **0**

Course Related Questions

Question	Number of Responses	Response Rate	Course Mean	Dept. Mean	Univ. Mean	Course Median	Dept. Median	Univ. Median
The syllabus helped me to learn.	27	87%	2.9	4.0	4.0	3.0	4.0	4.0
The textbook(s) helped me to learn.	27	87%	2.8	3.7	3.8	3.0	4.0	4.0
The materials posted online, including Blackboard, helped me to learn.	27	87%	3.3	4.1	4.2	4.0	4.0	4.0
The out-of-class assignments and fieldwork helped me to learn.	27	87%	3.1	4.2	4.2	3.0	4.0	4.0
The lectures helped me to learn.	27	87%	2.8	4.0	4.1	3.0	4.0	4.0
The in-class discussions and activities helped me to learn.	27	87%	2.7	3.9	4.2	3.0	4.0	4.0
The classroom technology helped me to learn.	27	87%	3.0	3.9	4.0	3.0	4.0	4.0
The required textbook(s) were important for my success in this course.	27	87%	2.5	3.6	3.7	2.5	4.0	4.0
I would recommend that the instructor continue requiring the textbook(s).	27	87%	2.5	3.7	3.8	2.5	4.0	4.0
The required digital and/or other supplemental materials were important for my success in this course.	27	87%	3.1	4.0	4.1	3.0	4.0	4.0
I would recommend the instructor continue requiring the digital and/or supplemental materials.	27	87%	3.3	4.1	4.1	4.0	4.0	4.0
I found this course intellectually challenging.	25	80%	2.7	4.2	4.1	3.0	4.0	4.0

Note: 5:Strongly Agree; 4:Agree; 3:Neutral; 2:Disagree; 1:Strongly Disagree; -1:Not applicable;

Learning Related Questions

Question	Number of Responses	Response Rate	Course Mean	Dept. Mean	Univ. Mean	Course Median	Dept. Median	Univ. Median
I learned a lot in this course.	27	87%	2.8	4.2	4.3	3.0	4.0	4.0
I learned to apply course concepts and principles.	27	87%	3.1	4.2	4.3	4.0	4.0	4.0
I developed additional skills in expressing myself orally and in writing.	27	87%	2.3	3.8	4.0	2.0	4.0	4.0
I learned to analyze and evaluate ideas, arguments, and points of view.	26	83%	2.3	4.0	4.1	2.0	4.0	4.0

Note: 5:Strongly Agree; 4:Agree; 3:Neutral; 2:Disagree; 1:Strongly Disagree; -1:Not applicable;

Instructor Related Questions: Nathaniel Derbinsky

Question	Number of Responses	Response Rate	Course Mean	Dept. Mean	Univ. Mean	Course Median	Dept. Median	Univ. Median
The instructor possessed the basic communication skills necessary to teach the course.	27	87%	4.5	4.4	4.4	5.0	5.0	5.0
The instructor clearly communicated ideas and information.	27	87%	4.4	4.2	4.3	5.0	5.0	5.0
The instructor clearly stated the objectives of the course.	27	87%	4.2	4.3	4.4	4.0	5.0	5.0
The instructor covered what was stated in the course objectives and syllabus.	27	87%	4.3	4.4	4.5	4.0	5.0	5.0
The instructor came to class prepared to teach.	27	87%	4.4	4.4	4.5	5.0	5.0	5.0
The instructor used class time effectively.	27	87%	4.0	4.2	4.3	4.0	4.0	5.0
The instructor provided sufficient feedback.	27	87%	3.9	4.0	4.2	4.0	4.0	4.0
The instructor fairly evaluated my performance.	27	87%	4.0	4.2	4.3	4.0	4.0	5.0
The instructor is someone I would recommend to other students.	27	87%	4.4	4.1	4.3	4.0	4.0	5.0
The instructor treated students with respect.	27	87%	4.6	4.5	4.5	5.0	5.0	5.0
The instructor acknowledged and took effective action when students did not understand the material.	27	87%	4.3	4.2	4.3	4.0	4.0	5.0
The instructor was available to assist students outside of class.	27	87%	4.1	4.3	4.4	4.0	4.0	5.0
The instructor displayed enthusiasm for the course.	27	87%	4.0	4.4	4.5	4.0	5.0	5.0

Note: 5:Strongly Agree; 4:Agree; 3:Neutral; 2:Disagree; 1:Strongly Disagree; -1:Not applicable;

Instructor Effectiveness: Nathaniel Derbinsky

Question	Number of Responses	Response Rate	Course Mean	Dept. Mean	Univ. Mean	Course Median	Dept. Median	Univ. Median
What is your overall rating of this instructor's teaching effectiveness?	27	87%	4.3	4.3	4.4	4.0	5.0	5.0

Note: 5:Almost Always Effective; 4:Usually Effective; 3:Sometimes Effective; 2:Rarely Effective; 1:Never Effective;

Course Related Questions (7 comments)

Q: Please comment on the strength and/or weakness of the required textbook/course materials.

1. I would have appreciated more resources for developing the website. We weren't really taught any of the technology we were using, although knowing the technology was essential to completing the project which is a vital part of our grade. I would have appreciated a more structured approach of learning.
2. Course material was too much and all over the place. The materials for lectures, homework and exam did not connect to the course project in a clear manner. The course materials got in the way of working on the class project, and the lecture notes were often littered with "fun" comics that distracted from the purpose of the notes
3. Meh... This is very much a read the power point class.
4. Did we have a textbook?
5. The course materials were a disaster. The exams consisted of vague T/F questions such as "Scrum is usually 2 weeks long," by which "scrum" signified stand-up meetings. Terminology used in my co-op (aka real industry) was not matching up to what the lecture slides specified. These slides were mainly taken from other universities but still managed to add in unnecessary information so I ended up studying with the original slides from Stanford.
6. The slides were not always clear or consistant, particularly the slides with testing. While I tried to take notes based on the lectures, if I wanted to checked something on the slides I could not. I'm also a visual person, so it helps to have things written down...

7 Slides posted on the website were high quality.

Learning Related Questions (15 comments)

Q: Please comment on the strengths of this course and/or ways to improve this course.

1	This course was a mess. There were three sections with three different professors and everything always felt very chaotic and there were countless inconsistencies and confusions between what each of the professors were telling us. The hardest parts of this class were figuring out what the professors wanted from us, I wish there was only one professor or the three professors communicated their expectations amongst each other better.
2	The instructors/TAs were not always coordinated on requirements or expectations for assignments which made things confusing at times. Course teaches general information through lectures but most of the skills learned from this course were through independent research or peer teaching while working on the semester-long group project. Semester-long project forces you to learn a lot outside of the classroom and also teaches many skills helpful for transitioning to a co-op or job.
3	The grading for this class seems to be focused far more on nit picking instead of letting us focus on building the project. I really don't think you should require Jenkins but allow people to use different stacks considering that jenkins is not the go to tool for several other stacks. There is a level of hypocrisy in regards to the thought process, "you should treat this as you would for a job with us as clients but you have to do it this way." Because of that we started development and then found out it required 50 hours to get jenkins to not be on fire. That is one example. I think in general you should make sure that tests, and assignments are designed from the perspective of making sure that students are actually learning material and not just regurgitating crap from slides, or performing the menial dev task instead of finding a creative solution that works. This kind of class makes me extremely frustrated because I feel like you aren't rewarding creative solutions that would be applicable in the real world and are again, regurgitating the material as dogma, as you have presented it.
4	Needed to enforce the process in sprint 1, such as Jenkins and SonarQube (make sure it throw error if code doesn't have test 100% coverage in sprint 1). Needed to be more result driven. Give more detail feedback on graded assignments and earlier (within 1-2 weeks after the deadline).
5	This course was a mess. The project was way too big considering that it was entirely web development, which is not covered by this course. However, the worst part was the grading. Requirements were often vague, and when they weren't it was only because there were multiple posts on piazza asking for clarification so the assignments would be rewritten sometimes less that 24 hours before the due date. In addition, the rubrics often felt completely against the spirit of the course. For example on homework 3, which covered material on how to write functional tests in general, over half the points were allocated to testing for nulls and integer over/underflow, which are Java-specific features. To make matters worse, the original rubric got the correct answers wrong! In multiple places! If you even have the gall to pull a bait-and-switch like that on an assignment, at least get the rubric correct for god's sake. Now I know (well I don't but it's reasonable to assume) that this wasn't done on purpose with malicious intent, but this incident and others gave the impression that Prof. [redacted] wanted to both rule over the course with an Iron Fist, while simultaneously not care about its success whatsoever. This was particularly evident in his answers to questions on piazza, where he would repeatedly answer reasonable, opened questions about clarifications with a yes or no answer and next to no explanation. It was honestly extremely frustrating to watch my peers be ignored in this way, and to feel like I had no other option to than to accept an incorrect grade because [redacted] changed the rubric so many times that no other Professor or TA could explain his reasoning. Prof. Derbinsky did everything he could to make this class enjoyable and useful to us, but [redacted] ultimately prevented that. I honestly think he should be fired, or banned from leading another course at the very least.
6	Focus more on operations and process. Most of second half of class felt pointless. Project felt kind of like a waste of time.
7	better TAs
8	Assignments were not well written. Grading feedback was unhelpful, and goals were not clear for the semester wide project. Make expectations clear.
9	The course was either too heavy on dev ops or not heavy enough. If it is a dev ops class, please replace some other work with more dev ops. If not, the class required far too much dev ops learning, making work inefficient towards the actual goal. Please consider setting up the course with a dev ops setup and skeleton for work, rather than requiring students figure that out on their own.
10	This course was not particularly useful, especially after being on co-op. The most learning intensive part of the course was the project setup. The project did not help to educate me as a student, because I'd already been on a co-op doing actual agile development. instead of teaching me about the process, it put an undue stress on my semester. Additionally, the project, homework, and exams didn't line up well and did nothing but get in the way of each other. The exams and project deadlines were on consecutive days, which meant doing well on the project meant sacrificing the time before the exam we could study, and we could only really study right before the exams due to the homework grades not being released until about a week before the midterm. Overall, having all 3 did nothing but over complicate the coursework. My biggest issue with the course was [redacted] It felt like my professor had no power (see professor comments) even though he had a lot of good and reasonable ideas. Instead, everything had to go through [redacted] He would reply on the course piazza with either a yes/no reply to a not yes or no question, reply with some form of non-answer, copy and paste the already linked confused wording

extremely abrasive and unprofessional to the point that piazza became a useless tool for the course, since all the answers seemed to be from [redacted] and were generally unhelpful.

Additionally, the TAs for the course were extremely inconsistent and grading criteria was unclear or unreasonable. One such example was a problem on the 3rd homework where a problem had a way to lose more points that the problem was worth (you couldn't lose more than the point value of the question, but there were ways to deduct more points than the problem was worth with penalties).

11 This course was definitely the worst CS class I have taken at Northeastern.

The contents of the course felt like they were haphazardly thrown together just days before each lecture. The grading was inconsistent, and requirements for homeworks and the projects were consistently confusing and self-contradictory.

Moreover, the classes and homeworks did not prepare me for the project very well. The long project in this course was a web development based group project, and most of the students in the class did not have web development experience. This was a good opportunity to learn some web development technologies, but that is not the focus of the course, so you are mostly on your own with the development side of things.

I think the main focus of this course, the project, should remain the focus, but I would recommend that it not be a web development focused project, or that perhaps options for types of projects could be presented to the students.

Additionally, too much of the first month of the class was spent on setting up development environments and other tools. The tutorials to set up Jenkins and other technologies did not really teach me how they work, just how to get them working by following the directions. That being said, even in those cases, the directions were usually incomplete, resulting in having to do setup up to a certain point or struggling with setup until a TA could help you or another student figured out the problem and posted on Piazza.

Overall, this course is an important course to take as a budding software developer, but in its current state, I would steer anyone away from taking it if possible due to all of these problems with the course's current organization.

I did learn a lot in this class, such as Git, how Jenkins works

12 This course is a complete waste of time for anyone who has gone on co-op. The lectures are too theoretical and historical to be of much practical use, and the project was mostly not supported by the lecture material.

13 This course shouldn't be required for students who have completed a co-op. I am actively trying to ignore the material taught in this course since it goes against what my professional employment has encouraged. Project groups were randomly chosen in lab on our own with no course consideration for skills (front end vs back end) in the group. An improvement would be doing away completely with this course but if that's not an option at least balance the teams since there is no instruction on how to design a web application.

14 A lot of time was spent doing set up tasks, and a lot of difficulty came from following specific instructions that didn't always work. I'd say the first two and a half months of this class was spent trying to set up our project, and I was already familiar with react and SQL. I wasn't learning new things or skills related to software development, unless you count practicing patience... Also the homeworks didn't always have clear questions, and a lot of the questions felt relatively useless. I already know how to write tests, and I had even more practice while working on the main project. The homework just seemed to waste time and take time away from the main project.

15 This course has a bit of an identity crisis. It seems like a grab bag of a bunch of different topics you were supposed to learn on co-op, just to make sure that no one at Northeastern graduates without this experience. Some of the content was useful, such as how to think about test coverage and a little bit about to set up software dev workflow. For the other content, git was something I already knew, and I had experience working in a formal software dev environment. Also, all the UML and requirements content is basically a high level review of IS3500, making this course even less useful for combined CS/IS majors like myself.

The project was not as useful as it could have been. The actual development work was tedious and dull, and the framework selected for the class was way too heavy. Practicing the process was kind of good. The technical side of things, with git, sonarqube, and Jenkins was not ideal. They made us set up Jenkins, but did not really go in depth on what was going on. It would have been better if we were taught more about CI pipeline/devops instead of just being given poorly written instructions.

Furthermore, Jenkins is just bad, really bad. We had it eat up massive amounts of memory and kill the VM, not to mention that the UI is incomprehensible. Getting SonarQube to talk to Jenkins was a nightmare, with the Jenkins task just randomly hang and need to be killed manually. GitHub actually worked. Jira was mis-configured so that smart commits never really worked (I did bring this up to the course staff), and we had so little control over it that it made it hard for us to adapt it to our project.

Overall, the project felt like a lot of busy work and seemed to focus more on learning how to develop a web app with some random process stuff mixed in. The project could have really been great. I think if it was oriented more towards learning the deveops side (actually explain to us what a good CI pipeline looks like, what choices need to be made, etc.), that would be good. Also, there was very little client interaction and requirements management practice in the project. I think this is the other place where the project could excel: teaching CS students how to talk to clients and manage the resulting requirements and design work. This aspect felt like it was missing.

Oh, and the home works felt like busy work. The home works were poorly defined and didn't feel like they were actually helping us learn. It would be better if the home work was just integrated into the project.

One more thing, we ran out of content 2/3rds through the semester. Our Professor did a good job lecturing on supplemental topics, but

Instructor Related Questions: Nathaniel Derbinsky (12 comments)

Q: Describe instructor's strengths, areas for improvement, and any additional comments.

1	Great lecturer!
2	Professor Derbinsky was great. He's personable, spoke from experience and related that experience well to the subjects we were discussing. It didn't seem like he was in charge of the course load/expectations (this course was taught by three professors and Professor Weintrub seemed to be the ringleader), so I can't speak to that.
3	Derbinsky is a great, no BS professor. Listens to concerns and understands students and tries to tailor classes to our interests. Gets through required material quickly and clearly to make time for project work time or will give mini lectures as requested by students, such as database or machine learning. Would definitely recommend to students.
4	Derbinsky was a very good professor and his respect for the students was good. He made sure to cut out a lot of the excess comics/pointless videos that littered the lecture notes and would let us go early if the lecture was over instead of dragging out the content to fill the whole time block. This was a great respect for the students and our time which was already stretched thin as we could use this time to meet with our groups and work. Additionally, he was willing to listen to groups who needed help or feedback. However, it felt like he had no power. Instead, when people had serious questions, he'd give one answer, but then say he needed to clear it with the other professors, and come back to class telling us the opposite of what he had said (which was a very reasonable response). Instead, it was clear that though Derbinsky was my professor, [redacted] was the actual professor in charge and my professor I saw in class had almost no power in the course. We were forced to dance to [redacted]'s tune meaning feedback from our professor in our lectures could be totally changed in the course of under a day.
5	Prof. Derbinsky was a great professor! Although I did not like Software Development for a number of reasons, Professor Derbinsky was not one of them. He used the class time efficiently, and his lectures were interesting, especially his stories about working as a freelance software developer.
6	Nate did an awesome job of recognizing existing faults in the class and working to mitigate them with the students.
7	Prof. Derbinsky did everything he could to make this class a success, but Prof. [redacted] ultimately held him back. I truly appreciate his openness and honesty, and really enjoyed the lectures that he gave on his own research (aka, an actually interesting topic, as opposed to the rest of the course)
8	A fantastic teacher bound by a terrible curriculum. Let him or somebody else rewrite the course
9	Prof Derbinsky tried his best with the material he was given. The slides are ripped off of other universities and contain a lot of dry and bloated material so lectures weren't stellar but it wasn't his fault. He would be better suited for a class in his specialty like Machine Learning, which he shared interesting stories about.
10	an earnest fellow
11	Our instructor was really great, but the problem was he had to communicate with the other professors, who were less clear and organized it seemed. I feel like our professor was pretty fair, but then the other professors would set these weird arbitrary standards. I would've been a lot happier with this course, I think, if it was just Nate Derbinsky teaching it.
12	Professor Derbinsky is great. He is a good lecturer and very fair to students. He has a passion for software development and machine learning, which he loves to share with the class. Even when we ran out of content for the class, he kept lecturing on topics around his research that he though we'd be interested in. I really enjoyed his lectures.