EC400 - Software Engineering in Practice Fall 2024

MW: 12:20 - 2:05 PHO 207

Instructor: Ed Solovey, PHO 531 Email: esolovey@bu.edu

- Drop-in office hours:
  - Tuesdays 1:00 pm - 3:00 pm PHO 531
  - Wednesdays 10:00 am - 12:00 pm PHO 531
- On-demand: https://calendly.com/esolovey-bu/30min (https://calendly.com/esolovey-bu/30min)

Course Objectives:

This class builds on students' software design skills by introducing them to an array of industry relevant concepts such as test driven development, effective understanding and debugging of existing codebases, use of IDEs, re-factoring, source control management, microservice architecture, intraservice communication, continuous deployment, data storage, fault tolerance and isolation, and system scaling.

Each week, students will get hands-on experience with putting these concepts to practice. We will start with an existing, monolithic application that reads data from a public API and computes analytics based on this infinite data stream and then exposes the results via a web interface. We will then use the concepts above to gradually tease apart and improve this application by separating it into smaller, single purpose microservices, all the while ensuring that functionality is maintained, and scalability and maintainability are improved.

Finally, we will cover product requirements and team collaboration as we add functionality to our microservices. Culminating in the deployment of the microservices to a cloud environment. The programming assignments for this course will be primarily in Java.

Prerequisites:

- EC 327
- EC 330 ideal
- Familiarity with Java is not required, but comfort in coding in any language is

Textbooks & Readings:

Because software engineering best practices evolve so quickly, there is no single textbook that we will be using. Rather, readings from a variety of sources will be required prior to each lecture.

- Common Reading Sources:
    - Software Engineering at Google, O'Reilly - free online (https://abseil.io/resources/swe-book/html/toc.html)
        - note: not everything in this book is relevant to our class or necessarily a best practice. We'll focus on selective reading from this book
    - Martin Fowler (https://martinfowler.com/)
        - blog with a collection of great posts on software engineering topics over the last 20+ years
- Reference Materials:
    - Java
        - CodeAcademy Java intro class (free) (https://www.bu.edu/link/bin/uiscgi_faculty.pl/1720550537/1720550537)
        - Oracle Java Tutorials (https://docs.oracle.com/javase/tutorial/)
    - IntelliJ IDE:
        - JetBrains Getting Started (https://www.jetbrains.com/help/idea/feature-trainer.html)
    - Gradle Build System:
        - Gradle Tutorial (https://docs.gradle.org/current/userguide/getting_started_eng.html?_gl=1*chclft*_gcl_au*NzAyNTEyMzY1LjE3MTk5MjY0MzI.*_ga*MTMxMTczOTQyNS4xNzE5OTI2NDMy*_ga_7W7NC6YNPT*MTcyMDYzMzM5Ny4yLjAuMTcyMDYzMzM5Ny42MC4wLjA.#getting_started)
        - Online Course (https://dpeuniversity.gradle.com/app/courses/012de84f-fcd3-45d4-9c4c-284382eb3f3f/?_gl=1*ol3kip*_gcl_au*NzAyNTEyMzY1LjE3MTk5MjY0MzI.*_ga*MTMxMTczOTQyNS4xNzE5OTI2NDMy*_ga_7W7NC6YNPT*MTcyMDYzMzM5Ny4yLjEuMTcyMDYzMzQwMS41Ni4wLjA.&_ga=2.202229394.2048166831.1720633397-1311739425.1719926432)

Assignments, announcements, course material, updated schedule, and other useful links

- posted on Blackboard (http://learn.bu.edu).

Course Outcomes:

As an outcome of completing this course, students will:

- be comfortable in navigating, understanding, and modifying existing, large code bases as well as building new ones from scratch.
- be able to re-factor code with confidence due to their appreciation for testing philosophies and code coverage metrics
- appreciate the benefits of Continuous Deployment and the need for Observability and Monitoring, and will be able to implement these for large systems.

- understand software engineering processes and learn how to best collaborate within software teams.
- be comfortable in a distributed systems environment and learn to build & deploy micro services.
- be able to articulate trade offs around software architecture choices and guide others towards decisions.

Topics:

1. Testing Philosophies & Methodologies
2. Code Probing, Debugging, and Execution Environments
3. Source Control & Static Analysis
4. Continuous Integration
5. Writing "Good" Code
6. Modularity, Composition, & Mutability
7. Observability & Monitoring
8. Software Lifecycle Processes
9. Microservices Architecture
10. Continuous Deployment
11. Software Architecture
12. Scalability
13. Software Performance
14. User Centric UX Design
15. Building UX with React
16. Technical Debt & Refactoring
17. Distributed Systems
18. Concurrency & Async Models
19. Cloud Computing
20. Software Engineering Interviews

Evaluation:

- **Grading**
  - 10 HW Assignments **50%**
  - 2 Midterm Exams **30%**
  - 1 Final Exam **20%**
- **Homework**
  - Will be posted in Blackboard on Wednesday mornings. They will be due at 11:59 pm on the following Wednesday.
  - No late assignments are accepted, but single lowest grade (or un-submitted) assignment will be automatically dropped.
  - There will be a very brief (2 minute survey) posted every week. Each student who completes the survey will receive 3 bonus points on that week's HW assignment (no going over 100 :)).
- **Midterms**

  - Will be closed book, no laptop, and will cover material from class lectures and homework assignments. They will occur in our classroom, PHO 207
  - Midterm One - Wednesday, October 16th, 12:20 - 2:05 PHO 207
  - Midterm Two - Wednesday, November 13th, 12:20 - 2:05 PHP 207
- **Final Exam**
  - Will be closed book, no laptop, and will cove material from class lectures and homework assignments.
  - Will be scheduled during standard finals week, December 16 - December 20

Course Policies:

The most successful software engineers are also the most collaborative ones. No one innovates by themselves in a vacuum. We are not going to sacrifice collaboration in this class. However, it is important to evaluate your individual progress in this class. To that end, students are encouraged to collaborate with all of their classmates as much as possible, brainstorm ideas, help each other with set up, and getting around bugs. Additionally, certain assignments will encourage/require peer code reviews.

Students are allowed to consult LLM's for ideas, but never for copying and pasting code

The 10 HW assignments that you submit **must be your own code and your own words - a copy of someone else's code or words is considered plagiarism**. If asked, you need to be able to explain every line of code that you submit. And I will plan on asking.

The closed book exams will cover the same material as the class lectures and assignments, and ensuring that students can reason through the assignments on their own will prepare them for the exams.

- **Academic integrity**
  - The homework assignments must be the result of your individual work. You may discuss the content, general approach, and challenges of the assignments with your classmates, but not the detailed solution. You are expected to formulate your approach and write the solutions of HW problems by yourself. Copying the solution and/or answer from another student or source is considered cheating. **You may not submit ANY code not written by you.**
  - **You may not collaborate in any way on the exams.**
  - Clearly reference any sources you used in your work: books, internet, and your collaborators! This includes websites (e.g. Stack Overflow) and AI assistants (e.g. ChatGPT) in which case you are required to include details of the prompts that you use. **Note that copying code or answers from such sources, or from another student is considered plagiarism.**
  - Boston University's academic code of conduct, https://www.bu.edu/academics/policies/academic-conduct-code/ (https://www.bu.edu/academics/policies/academic-conduct-code/) will be strictly applied.
- **Copyright**
  - All class material is copyrighted, and may not be shared publicly online by any means. This includes your own solutions to assignments.

If you feel like there is any ambiguity around this collaboration policy, please ask me questions to get clarifications before proceeding.

Inclusion:

I consider this classroom to be a safe and supportive place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the class.

Accommodations for Students with Documented Disabilities:

If you believe you might have a disability that requires accommodations, requests for accommodations must be made in a timely fashion to Disability & Access Services, 25 Buick St, Suite 300, Boston, MA 02215; 617-353-3658 (Voice/TTY). See established policies and procedures: http://www.bu.edu/disability/accommodations/ (http://www.bu.edu/disability/accommodations/)

⊕

Auto-generate question bank (https://learn.bu.edu/ultra/courses/_149934_1/outline/edit/document/_14156633_1/generate-document-questions?courseId=_149934_1&view=content)