Department of Computer Science

CS411 Course Intro

Perry Donham
@perrydBUCS
perryd@bu.edu

What's the 4-1-1?

- CS411 has many sides
- On the face its a course about the *process* of creating software
- It's also a course about developing software in teams
- Along the way we'll look at techniques and terminology being used in industry
- You'll also learn about several important tools by using them
- You might also pick up a new language or framework

The goals

- The big goal is to understand how we can use process to create high-quality software
- The big question is: *What is quality?*
- Some things we're interested in and will examine include
  - Security
  - Reusability
  - Abstraction
  - Performance
  - Testability
- What else do you think leads to quality software?

Process

- Every software venture involving more than one or two people uses some sort of process
- Especially companies doing ‘real’ software — they live and breathe process
- We’ll spend a few weeks looking at a handful of common process models, and then you’ll apply a model to a semester-long project
- *Most students I’ve talked with after their job interviews tell me that the companies weren’t interested in their skills as much as they were their understanding of process*

Architectures

- Another ongoing topic in CS411 is software architecture
  - By this we mean both application architecture and underlying system architecture
- Why? **Every design is a compromise** … we never build exactly the thing we originally wanted
- It's important to understand the tradeoffs and the constraints of building software in different ways so that we can make intelligent architecture decisions

Approaches

- I often distinguish between architectures and approaches
- An *architecture* is the overall design
- An *approach* is the specific implementation
- For example, the architecture might call for a cache
- The approach is dictated by the constraints we have and an understanding of the tradeoffs of each…in-memory vs table-based vs document-based etc

Toolchains

- *Toolchain* is just a buzzword meaning the set of specific tools that are used for a project
- Each company has a preferred toolchain, some commercial products, some open-source, some a mix
- For example, I tend to use JetBrains products as my IDE, StarUML for design, github as my repository, Mocha and Chai for testing Javascript, Omniplan for project management, and so on
- You'll come up with your own toolchain for your team projects

The Project

- A lot of people take CS411 just for the project work!
- I'll give you details in a few weeks, but the high-level is
  - You'll work together in small teams
  - The teams will propose, design, build, and test a web application
  - The application is non-trivial and must consume two or more public data streams and correlate them in some way to benefit the user
  - It's a great chance to learn new languages and frameworks
  - You'll find it incredibly hard to work as a team!

- The flow is that we'll learn about a process piece, for example gathering business requirements, and you'll then apply that piece to your project
- Teams present their application at the end of the course
- Some teams deploy to a live server such as Amazon EC2
- If you aren't a coding whiz, you still have plenty to contribute… coding is just a small part of the work!

Exams and quizzes and such

- Per the syllabus, there are five components to your grade
  - 25% Midterm exam
  - 30% Quizzes (typically one per week)
  - 15% Case studies
  - 20% Team project score (shared by team)
  - 10% Team peer review (360° review)

Blackboard and Piazza and bears

- Course-related materials (overheads and handouts) are posted on Piazza
- Piazza is also for Q&A — please feel free to answer each others' questions (and I'll chime in, too)
- Project-related work will be turned in as a link to a github repo
- Gradescope is where you'll turn things in

Tools

- Quite a bit of work will be done in various tools; you are welcome to use whatever you prefer; try not to get caught up in tool porn, though!
- Here are some that I'l use in class:
  - IDE - JetBrains (www.jetbrains.com) has robust, free (for students) IDEs for most major languages and frameworks; I use WebStorm for Javascript…pyCharm is nice for Python, and so on
  - StarUML - staruml.io is the tool I use in class to work on UML diagrams (which we'll be learning about)
  - Github - Sample code in class, and your project work, is hosted on Github; if you aren't familiar with it, you might want to start to take a look (github.com)
  - MongoChef - nice GUI for working with mongoDB collections (mySQL Workbench is good for relational dbs)
  - Postman - Outstanding tool for testing RESTful APIs

Languages

- Your project team will decide what language and framework to use for the semester
- My class examples will be in Javascript using Node and Angular (and possibly Typescript)
- My advice is to use the class and project as an opportunity to learn a language and framework you aren't familiar with
- While not a language you'll also be working with databases in your project…I use MongoDB in class and you might want to play with that if you aren't familiar with it already

Contacting me and hours

- My office is in 64 Cummington, PSY228C (in the same little hall as Dave Sullivan)
- Hours are posted on Piazza
- Best way to get in touch is email: perryd@bu.edu
- You can also IM me (but see next slide)
- Phone is 202-567-7362 (Google will intercept it)
- Blog is sites.bu.edu/perryd
- Twitter is @perrydBUCS
- I'm also on Reddit

Privacy

- I'm a strong advocate of personal privacy and I try to use tools that limit the ability of others to eavesdrop
- To that end, if you'd like to practice using privacy tools
  - Use Signal to IM me or call me (https://whispersystems.org)
  - Send and receive email with me using PGP encryption; my public key fingerprint is C894 B69B 6576 C394 1452 2E9E 7C38 F315 BCC1 ADDF
    - gpgtools.org or Thunderbird + Enigmail for OS/X, gpg4win.de for Windows
- I am happy to help you set up either or both of these

Let's get started!