CS 320
Introduction to Software Engineering
Spring 2017

January 23, 2017
CS320

Instructor
● Prof. René Just
● Office: CS 358
● Office hours: Wednesdays 11am -- 1pm or by appointment
● rjust@cs.umass.edu

Teaching assistant
● Justin Purcell
● Office hours: Thursdays 2:30pm -- 4:30pm, CS 207
● jepurcel@umass.edu
Today

- My background
- What is Software Engineering?
- Why is Software Engineering important?
- Your expectations
- Course overview
- My expectations
- Logistics
My background

My research interests
- Software testing and debugging
- Static program analysis
- Software security
- Mining software repositories
- Empirical software engineering
What is Software Engineering?
What is Software Engineering?
What is Software Engineering?
What is Software Engineering?
What is Software Engineering?
What is Software Engineering?

All of the above and much more!

It’s more than just programming.
What is Software Engineering?

More than just programming

- The complete process of specifying, designing, developing, analyzing, and maintaining a software system.

What are common tasks in Software Engineering?
What is Software Engineering?

More than just programming

- The complete process of specifying, designing, developing, analyzing, and maintaining a software system.
- Common Software Engineering tasks:
  - Requirements engineering
  - Specification writing
  - Software architecture and design
  - Programming
  - Software testing and debugging
What is Software Engineering?

More than just programming

- The complete process of specifying, designing, developing, analyzing, and maintaining a software system.

- Common Software Engineering tasks:
  - Requirements engineering
  - Specification writing
  - Software architecture and design
  - Programming
  - Software testing and debugging

Why is Software Engineering important?
Why is Software Engineering important?

Software is everywhere...
Why is Software Engineering important?

Software is everywhere...and buggy!
Why is Software Engineering important?

Software is complex!

- ~15 million lines of code
Why is Software Engineering important?

Software is complex!

- ~15 million lines of code

Let’s say 50 lines per page (0.05 mm)
  - 300000 pages
  - 15 m (49 ft)
Why is Software Engineering important?

Infrastructure is software, too!

Example: Design/configuration space exploration

- 150 configurations
- 85 hours per execution
- 25,000+ CPU hours (~3 CPU years)
- $10k in elastic computing credits
Software development: the high-level problem

Specification → ??? → Source code
Software development: the high-level problem

One solution: “Here happens a miracle”
Software development: the high-level problem

One solution: “Here happens a miracle”
Software development: the high-level problem

Ad-hoc or systematic?

Pros: Ad-hoc

- No formal process. “Brain to keyboard”
- Easy, quick, and flexible.
Software development: the high-level problem

Ad-hoc or systematic?

Pros: Ad-hoc

- No formal process.  “Brain to keyboard”
- Easy, quick, and flexible.

Can you think of any drawbacks?
Software development: the high-level problem

Ad-hoc or systematic?

Pros: Ad-hoc
● No formal process. “Brain to keyboard”
● Easy, quick, and flexible.

Cons: Ad-hoc
● Might lack important tasks such as design or testing.
● Doesn’t scale to multiple developers.
● Impossible to measure effort and progress.
Summary: Software Engineering

What is Software Engineering?
- The complete process of specifying, designing, developing, analyzing, and maintaining a software system.

Why is it important?
- Decomposes a complex engineering problem.
- Organizes processes and effort.
- Improves software reliability.
- Improves developer productivity.
Your expectations

Introduction and a brief (5 minute) survey

- **Position**: What type of job are you looking for?
- **Top-3 tasks**: What do you think your tasks related to SE will be?
- **Top-3 expectations**: What do you expect from this course?
Course overview: the big picture

- **Software processes, requirements, and specification**
  - Learn about different software development processes.
  - Learn how to write a requirements document and a specification.

- **Software development**
  - Learn how to decompose a complex problem and build abstractions.
  - Improve your coding skills.

- **Software testing and debugging**
  - Learn how to write (unit) tests.
  - Hands-on experience, using testing and debugging techniques.

- **Class project**
  - Apply all of the above in a semester-long project, guided by CS529 students who have previously taken CS320.
Course overview: grading

Overall grading

- **50%** Semester-long class project *(in groups)*
- **20%** In-class exercises *(4 lab sessions in groups)*
- **20%** Midterm exam *(individual exam)*
- **10%** Participation
Expectations

● Programming experience and familiarity with one programming language (Java, C++, ...).

● Active participation in discussions.

● Teamwork and communication.

● Reflecting on submitted assignments and improving/resubmitting the work.

You must already know how to program!
Logistics

- Lectures: Mo/We, 2:30pm – 3:45pm
  Discussions: We 1:25pm – 2:15pm

- Lectures, discussions, lab session, and presentations in room CS 142.

- Course material, policies, and schedule on web site: http://people.cs.umass.edu/~rjust/courses/2017Spring/CS320

- Submission of assignments via Moodle: https://moodle.umass.edu

No discussion session on 01/25!