CS121: Software Development

Today

- Overview
  - What is "software development?"
  - What is this course about?
- LAC computers/accounts

"Software Development"

Key Processes

- Requirement specification/analysis
- Design
- Implementation
- Testing

"Software Development"

- Requirement specification/analysis
- Design
- **Implementation** (focus of CS70)
- Testing

Why study software development?

- Society has become increasingly dependent on software systems.

Example 1

Expedia Maps:
I need to go to the airport

Output reported in *The Risks Digest* Oct. 1, 1999

Excerpts from **Expedia Maps** directions:
From: **Laurel, Maryland**
To: **Baltimore-Washington International Airport, Maryland**
Driving Distance: 5865.1 miles
Time: 9 day(s) 3 hour(s) 22 minute(s)

Time (**hour**:minute)	Instruction
0:00	Depart Laurel, Maryland
1:01	Entering Delaware
1:17	Entering New Jersey
3:24	Entering New York
3:51	Entering Connecticut
5:51	Entering Massachusetts
7:29	Entering New Hampshire
7:44	Entering Maine
12:20	Entering New Brunswick
20:20	Take the North Sydney-Argentia Ferry
34:32	Entering Newfoundland
36:35	Turn left onto Local road(s) (4543.1 mi)
219:22	Arrive Baltimore-Washington International Airport, Maryland

Why study software development?

- Society has become increasingly dependent on software systems.
- Failures in software systems can be dangerous and costly

Example 2

- 1981: FAA announced plans to modernize air-traffic control.
- 1985: IBM awarded contract. System estimate to have 1.5 million lines of code, cost **$2.5 billion**, and be deployed by 1991.
- 1987: Revised cost **$4.3 billion**, deployment slipped to 1995.
- 1994: FAA decided that the **project would never be completed**, and cancelled it. Net loss **$1.5 billion**

Example 3

- Linear accelerators create high- energy beams that can destroy tumors with minimal impact on the surrounding healthy tissue
- Therac 25 was the first linear accelerator with dosage controlled solely by software (as opposed to hardware)

1983: Pre-release Safety Analysis

- Programming errors have been reduced by extensive testing on a hardware simulator and under field conditions on teletherapy units. Any residual software errors are not included in the analysis.
- Program software does not degrade due to wear, fatigue, or reproduction process.
- Computer execution errors are caused by faulty hardware components and by "soft" (random) errors induced by alpha particles and electromagnetic noise.

Therac-25 History

- 1983: First Therac 25 installed
- 1985-1987: Six massive-overdose accidents due to "software error" are reported. Overdoses caused severe burns and death.
- 1987: Recalled for extensive design changes, including hardware to safeguard against software errors in dosage.

Therac-25 Software Errors

- Bugs in program modules UNIT TEST
- System errors due to misinterpretations of module interfaces SYSTEM TEST
- Errors in users' guide ACCEPTANCE TEST

Why study software development?

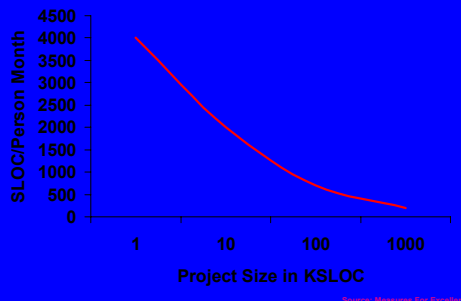
- Society has become increasingly dependent on software systems.
- Failures in software systems can be dangerous and costly
- Software design/development is a very hard problem

Stats on software projects

J. Johnson, "Creating Chaos," American Programmer, July 1995

- 31.1% are canceled before they are finished
- 52.7% overrun their cost estimates by at least 189%
- 33.3% overrun their time estimates by 100%-200%
- 94% of all projects do a "restart"

Large vs. Small Steps: Productivity



Source: Measures For Excellence, Putnam, 1992. Based on 1,600 systems.

How hard is software development?

What experts say:

[there are] no silver bullets" … that will do for software productivity, reliability, and simplicity what electronics, transistors, and large-scale integration did for computer hardware

– Frederick J. Brooks, Jr.
The Mythical Man-Month

What experts say:

"Wicked problems are problems that are fully understood only after they are solved the first time."

Rittell and Webber, Dilemmas in a general theory of planning, 1983

Software is a wicked problem…

DeGrace and Stahl, "Wicked Problems, Righteous Solutions, A Catalogue of Modern Software Engineering Paradigms.", Prentice-Hall 1980

Objectives of CS121

- Understand the problems
- Examine possible strategies
- Practice

What is this course about?

The design and development of large software projects
- "Theory"
- Practice

Software Design & Development "Theory"

"Theory" is not well-developed
- relatively new field
- paradigm shifts
- "wicked" problem
- no "silver bullets"
- there are techniques, strategies, and principles that can help

Major Topics

- Software Development
  - managing key processes
  - artifacts
- Software Design
  - principles
  - patterns
  - artifacts

Readings

- Weekly readings assigned on Tuesday
- Suggested questions
- Occasional quiz
- Material will be tested in exams

Reading

- Text: Software Design by David Budgen
  Buy it in the CS office for $60.88.
- Additional readings will be assigned

Software Design & Development Practice

You'll develop three software projects in this class

- arcade game
- miniature golf game
- computer game of your design

Why games?

They involve a range of problems that rarely show up in a single software project

- User interface design
- Computer graphics and sound
- Simulation and modeling
- Lots of mathematics
- Real-time
- Other possibilities: AI, networking, etc.

What about graphics

- You'll learn some basic OpenGL that will be more than adequate for your games.
- Yes … you can do a few all-nighters and figure out some cool effects … but that is not required … that is not even recommended.
- This is not a graphics course. If you want to do cool effects take the graphics course.

Major Topics

- Software Development
  - managing key processes
  - artifacts
- Software Design
  - principles
  - patterns
  - artifacts

Software Development Practice

You'll develop three software projects in this class

- arcade game
  - focus: Software development cycle, processes (particularly requirements elicitation/analysis), artifacts of development process
- miniature golf game
  - focus: Software design process, design principles & patterns, communication, artifacts
- computer game of your design
  - focus: Putting it all together

Misc.

- My email is z@cs.hmc.edu.
- My office is 2341 Olin.
- My office hours MTW 2:45-4:00

More Misc.

- The mailing list is cs-121-l. You need to subscribe if you are off-campus
- The course web page is www.cs.hmc.edu/~courses/2003/fall/cs121
- The tutor is Ed Heaney. You can find his email addresses on the web page. Tutoring hours will be arranged soon.

Grading

- Projects 75%
  - Breakdown
    - Arcade Game: 15%
    - Golf Game: 25%
    - Final Game: 35%
  - Evaluation
    - Game Grade * Your Contribution Factor
- Two midterms 10% each
- Individual assignments and quizzes 5%
- Class participation 0% (bump factor)

Today's reading assignment

- Budgeon: Chapter 1,2
- Brooks: "No Silver Bullet" available on course web page

The history of games

Which is borrowed from John Laird at the University of Michigan