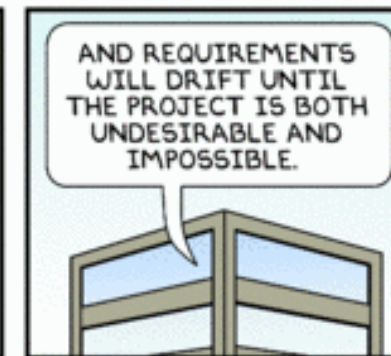
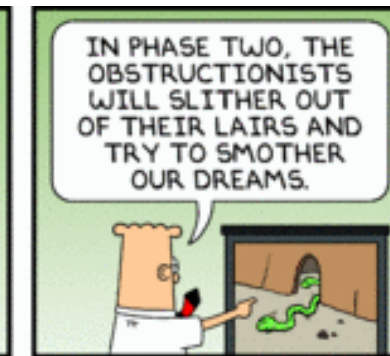
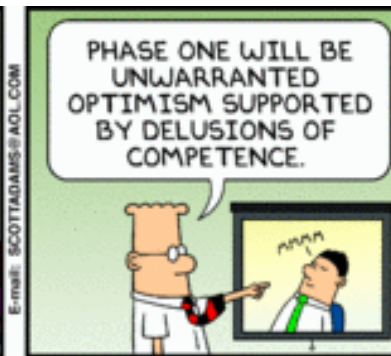


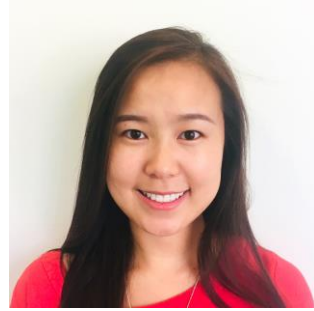
CS 4278/5278: Principles of Software Engineering

Dr. Yu Huang
Assistant Professor
CS, ISIS, LIVE
yu.huang@vanderbilt.edu



Course Information

- **Instructor:** Yu Huang



- **Teaching Assistants:**

- Zihan Fang



Anda Liang



Eric Li



- **Course meetings**

- FGH 134
- TR 1:15 – 2:30pm Central Time

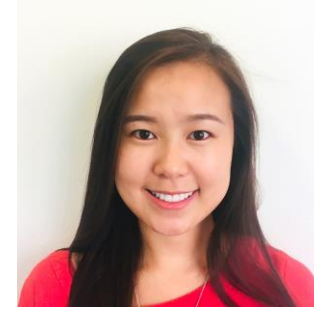
All course staff host OH: check the course website later (TBD at this moment)

If you need to email us, please include CS4278/5278 in the subject line

Course Information

- **Instructor:** Yu Huang

- yu.huang@vanderbilt.edu
- PhD in Computer Science, UMich
- MS in Computer Engineering, UVa
- BS in Aerospace Engineering, HIT, China
- Software Engineering: human factors, OSS, AI4SE
- <https://yuhuang-lab.github.io/>



Administrative Information

- **Course website:** <https://huang.isis.vanderbilt.edu/cs4278>
- Piazza: <https://piazza.com/class/lr5c8w0mskl2xk>
- Brightspace: <https://brightspace.vanderbilt.edu/d2l/home/475762>
- Autograder: <https://autograder.isis.vanderbilt.edu/>

What is the worst thing that can happen if you say something wrong in this class?



What is the worst thing that can happen if you say something wrong in this class?



Why should you study Software Engineering?

List of public corporations by market cap - 2023

Rank	First quarter		Second quarter		Third quarter		Fourth quarter	
1		Apple ▲2,609,000 ^[28]		Apple ▲3,050,000 ^[28]		Apple ▼2,677,000 ^[28]		Apple ▲2,994,000 ^[28]
2		Microsoft ▲2,146,000 ^[29]		Microsoft ▲2,532,000 ^[29]		Microsoft ▼2,346,000 ^[29]		Microsoft ▲2,795,000 ^[29]
3		Alphabet ▲1,332,000 ^[30]		Alphabet ▲1,530,000 ^[30]		Alphabet ▲1,662,000 ^[30]		Alphabet ▲1,764,000 ^[30]
4		Amazon ▲1,058,000 ^[31]		Amazon ▲1,337,000 ^[31]		Amazon ▼1,312,000 ^[31]		Amazon ▲1,570,000 ^[31]
5		Nvidia ▲686,090 ^[32]		Nvidia ▲1,044,000 ^[32]		Nvidia ▲1,074,000 ^[32]		Nvidia ▲1,223,000 ^[32]
6		Berkshire Hathaway ▼677,770 ^[33]		Tesla ▲829,670 ^[34]		Tesla ▼794,200 ^[34]		Meta ▲909,000 ^[35]
7		Tesla ▲656,420 ^[34]		Berkshire Hathaway ▲745,010 ^[33]		Meta ▲772,490 ^[35]		Tesla ▼789,930 ^[34]
8		Meta ▲549,480 ^[36]		Meta ▲735,450 ^[36]		Berkshire Hathaway ▲769,260 ^[33]		Berkshire Hathaway ▲783,550 ^[33]
9		TSMC ▲482,410 ^[37]		TSMC ▲523,410 ^[37]		Eli Lilly ▲509,890 ^[38]		Eli Lilly ▲553,370 ^[38]
10		Visa ▲473,870 ^[39]		Visa ▲497,370 ^[39]		Visa ▼480,990 ^[39]		TSMC ▲539,390 ^[37]

Source: https://en.wikipedia.org/wiki/List_of_public_corporations_by_market_capitalization (as of Dec 31, 2023)

List of public corporations by market cap-2022

Rank	First quarter		Second quarter		Third quarter		Fourth quarter	
1		Apple ▼2,850,000 ^[29]		Apple ▼2,212,000 ^[29]		Apple ▲2,221,000 ^[29]		Apple ▼2,066,000 ^[29]
2		Microsoft ▼2,311,000 ^[30]		Microsoft ▼1,920,000 ^[30]		Microsoft ▼1,737,000 ^[30]		Microsoft ▲1,787,000 ^[30]
3		Alphabet ▼1,846,000 ^[31]		Alphabet ▼1,435,000 ^[31]		Alphabet ▼1,254,000 ^[31]		Alphabet ▼1,145,000 ^[31]
4		Amazon ▼1,659,000 ^[32]		Amazon ▼1,080,000 ^[32]		Amazon ▲1,151,000 ^[32]		Amazon ▼856,940 ^[32]
5		Tesla ▲1,114,000 ^[35]		Tesla ▼697,660 ^[35]		Tesla ▲831,150 ^[35]		Berkshire Hathaway ▲681,770 ^[34]
6		Berkshire Hathaway ▲779,150 ^[34]		Berkshire Hathaway ▼602,450 ^[34]		Berkshire Hathaway ▼596,410 ^[34]		UnitedHealth ▲495,370 ^[39]
7		Nvidia ▼684,880 ^[33]		UnitedHealth ▲481,870 ^[39]		UnitedHealth ▼472,410 ^[39]		Johnson & Johnson ▲461,840 ^[40]
8		Meta ▼605,250 ^[36]		Johnson & Johnson ▼467,090 ^[40]		Johnson & Johnson ▼429,500 ^[40]		ExxonMobil ▲454,240 ^[41]
9		TSMC ▼540,670 ^[37]		Tencent ▼445,990 ^[42]		Visa ▲374,380 ^[43]		Visa ▲439,950 ^[43]
10		UnitedHealth ▲479,830 ^[39]		Meta ▼436,390 ^[36]		Meta ▼364,650 ^[44]		Tencent ▲405,090 ^[42]

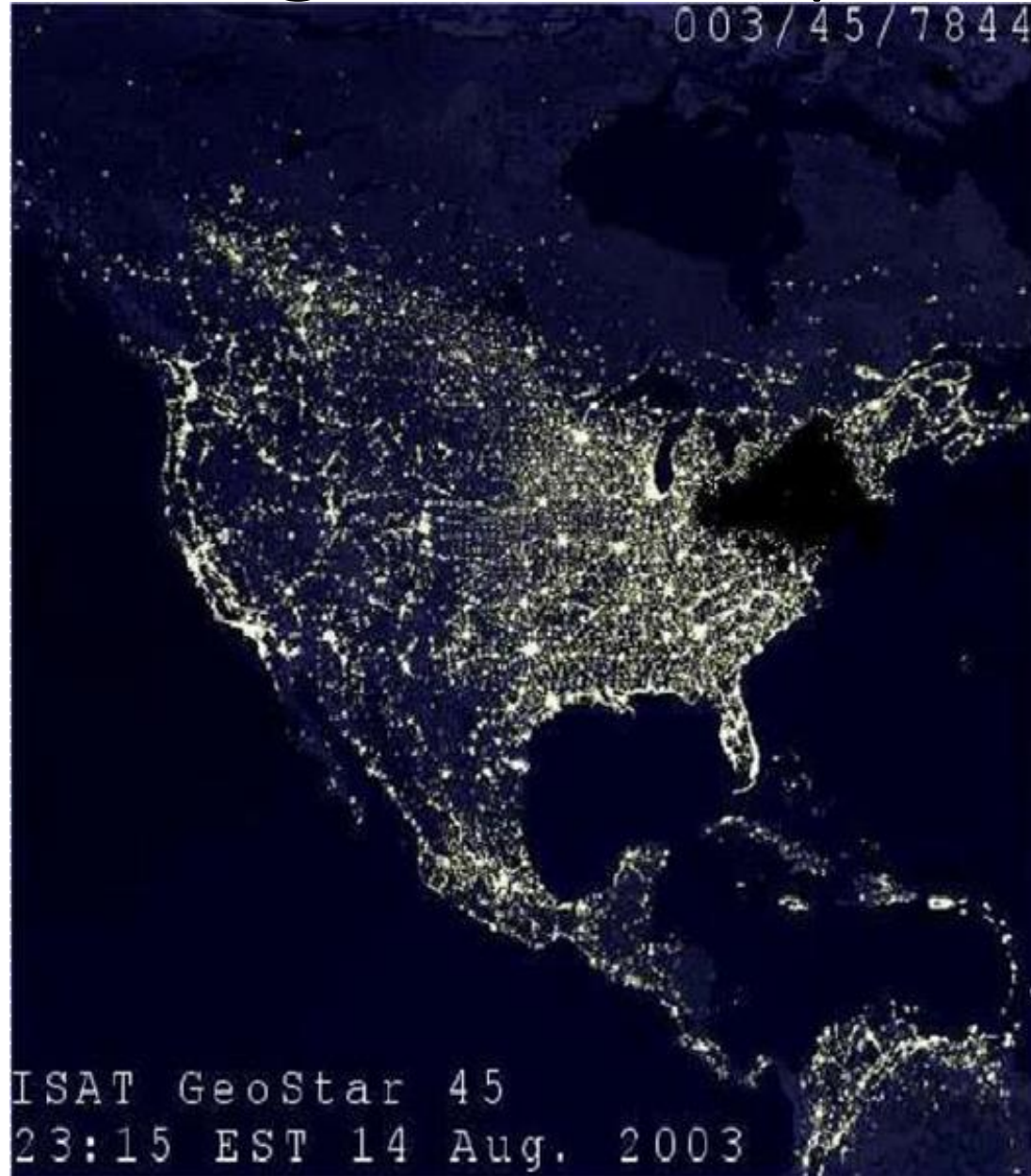
Source: https://en.wikipedia.org/wiki/List_of_public_corporations_by_market_capitalization

List of public corporations by market cap-2021

Rank	First quarter		Second quarter		Third quarter		Fourth quarter	
1		Apple ▼2,050,000 ^[20]		Apple ▲2,286,000 ^[20]		Apple ▲2,339,000 ^[20]		Apple ▲2,913,000 ^[20]
2		Microsoft ▲1,778,000 ^[21]		Microsoft ▲2,040,000 ^[21]		Microsoft ▲2,119,000 ^[21]		Microsoft ▲2,525,000 ^[21]
3		Amazon ▼1,558,000 ^[22]		Amazon ▲1,735,000 ^[22]		Alphabet ▲1,777,000 ^[23]		Alphabet ▲1,922,000 ^[23]
4		Alphabet ▲1,395,000 ^[23]		Alphabet ▲1,680,000 ^[23]		Amazon ▼1,664,000 ^[22]		Amazon ▲1,691,000 ^[22]
5		Meta ▲838,720 ^[24]		Meta ▲985,920 ^[24]		Meta ▼956,890 ^[24]		Tesla ▲1,061,000 ^[25]
6		Tencent ▲766,970 ^[26]		Tencent ▼721,460 ^[26]		Tesla ▲776,850 ^[25]		Meta ▼935,640 ^[24]
7		Tesla ▼641,110 ^[25]		Tesla ▲654,780 ^[25]		Berkshire Hathaway ▼619,950 ^[27]		Nvidia ▲732,920 ^[28]
8		Alibaba Group ▼615,010 ^[29]		Berkshire Hathaway ▲637,280 ^[27]		TSMC ▼579,030 ^[30]		Berkshire Hathaway ▲668,630 ^[27]
9		TSMC ▲613,410 ^[30]		TSMC ▲623,160 ^[30]		Tencent ▼574,460 ^[26]		TSMC ▲623,930 ^[30]
10		Berkshire Hathaway ▲590,050 ^[27]		Alibaba Group ▲615,140 ^[29]		Nvidia ▲517,900 ^[28]		Tencent ▼559,900 ^[26]

Source: https://en.wikipedia.org/wiki/List_of_public_corporations_by_market_capitalization

What is the wrong with this picture?



Software is Critical: Power

The **Northeast blackout of 2003** was a widespread [power outage](#) that occurred throughout parts of the [Northeastern](#) and [Midwestern United States](#) and the Canadian province of [Ontario](#) on Thursday, August 14, 2003, just after 4:10 p.m. EDT.^[1]

Some power was restored by 11 p.m. Most did not get their power back until two days later. In other areas it took nearly a week or two for power to be restored.^[2] At the time, it was the world's second [most widespread blackout in history](#), after the [1999 Southern Brazil blackout](#).^{[3][4]} The outage, which was much more widespread than the [Northeast Blackout of 1965](#), affected an estimated 10 million people in Ontario and 45 million people in eight U.S. states.

The blackout's primary cause was a programming error or "[bug](#)" in the alarm system at the control room of [FirstEnergy Corporation](#), an [Akron, Ohio](#)-based company. The lack of an alarm left operators unaware of the need to re-distribute power after overloaded transmission lines hit unpruned foliage, triggering a "[race condition](#)" in the [energy management system](#) software, a bug affecting the order of operations in the system. What would have been a manageable local blackout cascaded into massive widespread distress on the electric grid.



Software is Critical: Defense

- Quoting an Air Force lieutenant general, “The only thing you can do with an F-22 that does not require software is take a picture of it.”



[Crouching Dragon, Hidden Software: Software in DOD Weapon Systems (Ferguson, IEEE Software, 2001)]

Software is Critical: Privacy

Equifax security breach impacts 145.5 million

Name, SSN, DOB, Address. Also DL# and CC#

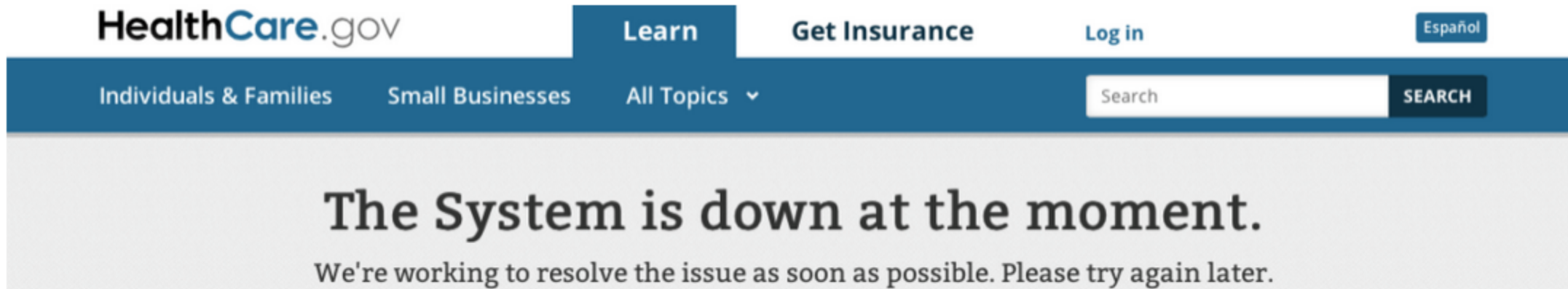
“I didn't have to do anything fancy,” the researcher told Motherboard, explaining that the site was vulnerable to a basic “forced browsing” bug. The researcher requested anonymity out of professional concerns. **“All you had to do was put in a search term** and get millions of results, just instantly—in cleartext, through a web app,” they said. In total, the researcher downloaded the data of hundreds of thousands of Americans in order to show Equifax the vulnerabilities within its systems. They said they could have downloaded the data of all of Equifax's customers in 10 minutes: “I've seen a lot of bad things, but not this bad.”

Software is Critical: Space

- The European Space Agency's Ariane 5 Flight 501 was **destroyed** 40 seconds after takeoff (June 4, 1996). The US \$1 billion prototype rocket **self-destructed** due to a bug in the on-board guidance software.
- (The bug? Bad conversion of `double` to `short`, leading to an overflow.)



Software is Critical: Healthcare



The US Government's failed launch of the Healthcare.gov website highlights issues with integrating technology into a large bureaucratic organization.

"I'm going to try and download every movie ever made, and you're going to try to sign up for Obamacare, and we'll see which happens first" – Jon Stewart challenging Kathleen Sebelius (former Secretary of Health and Human Services) to a race.

www.HealthCare.gov Overview

The affordable care act (increasing healthcare access to all Americans), signed by President Obama on 23 March 2010, required that the United States Department of Health and Human Services (HHS) launch HealthCare.gov on 1 October 2013. This website going to be the official health care exchange that would allow residents to compare prices of health care plans, identify if they qualify for federal subsidies, and enroll in a chosen plan. The Affordable Care Act gave states the right to create their own healthcare exchange or opt-in to the federal exchange (healthcare.gov).

Software is Critical: Healthcare



Healthcare.gov was officially launched on 1 October 2013 covering residents of 36 states that did not create and manage their own healthcare exchange. Problems with the website were apparent immediately. High website demand (250,000 users [5 times more than expected]) caused the website to go down within 2 hours of launch. While website capacity was initially cited as the main issue, additional problems arose mainly due to the website design not being complete. Users cited issues such as drop down menus not being complete and insurance companies cited issues with user data not being correct or complete when it reached them.

In addition, the website's login feature (which is the first step to accessing the website) could handle even less traffic than the main website which created a huge bottleneck. Due to poor planning, this same log in method was also used by website technicians, making it extremely difficult for them to log in and troubleshoot problems.

A total of 6 users completed and submitted their applications and selected a health insurance plan on the first day.

The key issues discussed above resulted in the rollout of the healthcare.gov website ballooning the initial \$93.7M budget to an ultimate cost of \$1.7B.

Software is Critical: Healthcare (!)

- Therac-25 radiation therapy machine
- At least six accidents in which patients were given massive overdoses of radiation
- Because of concurrent programming errors, it sometimes gave its patients radiation doses that were hundreds of times greater than normal, resulting in death or serious injury



Software is Critical: Coronavirus

Teleconference apps and new tech surge in demand amid coronavirus outbreak

Akanksha Rana, Arriana McLymore

3 MIN READ




(Reuters) - Global downloads of business apps including Tencent Conference, WeChat Work, Zoom, Microsoft Teams and Slack have risen nearly five fold since the start of the year, data showed, as the coronavirus outbreak changes how corporations work.

While such apps attracted 1.4 million new users across the App Store and Google Play in the first week of January, that figure jumped to a record 6.7 million in the first week of March, according to app analytics firm Sensor Tower.

The outbreak, which reached global pandemic status this week, has forced companies to rely heavily on business conferencing tools as workers stop commuting to offices.

Software is Critical: Education

Autograder - CS4278/5278 - Principles of Software Engineering Spring 2024 

HW0 - Setup



HW1a - Test Coverage - AVL



HW1b - Test Coverage - PNG



HW1c - Test Coverage - Java



Software is Critical: Education



黄羽

- HOME
- CONTACT
- ABOUT
- RESEARCH
- PEOPLE
- PUBLICATIONS

Teaching

CS 4278/5278

Spring24, TR, 1:15pm - 2:30pm, FGH 134.

Vanderbilt University

CS 4278/5278

Principles of Software Engineering - S23

Spring23, TR, 1:15pm - 2:30pm. University Club 117

Vanderbilt University

CS 8395

Advanced Topics in Software Engineering - F22

Fall22, TR, 11:00am - 12:15pm. Featheringill Hall 203

Vanderbilt University

CS 4278/5278

Principles of Software Engineering - S22

Spring22, MW, 2:30pm - 3:45pm. Engineering & Science 048

Vanderbilt University

CS 8395

Advanced Topics in Software Engineering - F23

Fall23, TR, 11:00pm - 12:15pm. ESB 001

Vanderbilt University

Advising and Research Opportunities

I welcome students to discuss their research interests and career plans. Unfortunately, faculty's schedule is not always open (in fact, most of the time we are busy with who knows what is going on). So please check my calendar below for potential meeting slots: I really prefer putting such meetings on Monday, Tuesday and Thursday so that I can focus on other work on Wednesday and Friday. Please only consider Wednesday/Friday if that is your only choice.

[My calendar can be viewed here.](#)

Software is Critical: Education



黄羽

- HOME
- CONTACT
- ABOUT
- RESEARCH
- PEOPLE
- PUBLICATIONS

Teaching

<p>CS 4278/5278</p> <p>Spring24, TR, 1:15pm - 2:30pm, FGH 134.</p> <p>Vanderbilt University</p>	<p>CS 4278/5278</p> <p>Principles of Software Engineering - S23</p> <p>Spring23, TR, 1:15pm - 2:30pm. University Club 117</p>	<p>CS 8395</p> <p>Advanced Topics in Software Engineering - F22</p> <p>Fall22, TR, 11:00am - 12:15pm. Featheringill Hall 203</p>	<p>CS 4278/5278</p> <p>Principles of Software Engineering - S22</p> <p>Spring22, MW, 2:30pm - 3:45pm. Engineering & Science 048</p>
---	---	--	---

<p>CS 8395</p> <p>Advanced Topics in Software Engineering - F23</p> <p>Fall23, TR, 11:00pm - 12:15pm. ESB 001</p> <p>Vanderbilt University</p>
--

Announcements

I've been putting such meetings on Monday

My calendar can be viewed here.

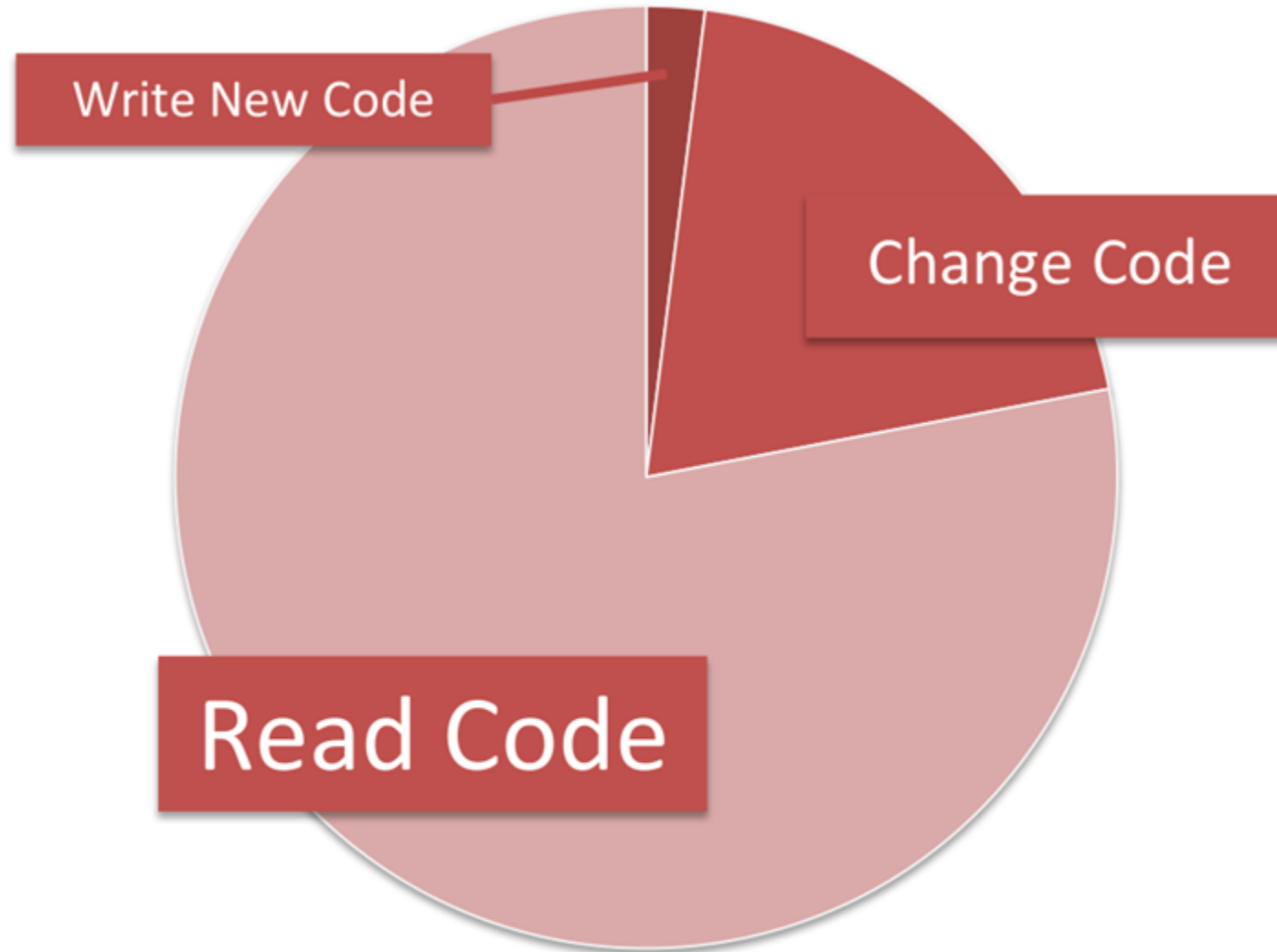
```
1470 <div class="col-xl-4 col-sm-6 mb-2">
1471   <div class="card h-100 border-primary shadow-sm">
1472     <div class="card-header">
1473       <h5 class="card-title">CS 4278/5278</h5>
1474       <h6 class="card-subtitle"><a
1475         href="https://huang.isis.vanderbilt.edu/cs4278s24>Principles
1476         of Software Engineering - S24</a></h6>
1477     </div>
1478     <ul class="list-group list-group-flush card-body p-0">
1479       <li class="list-group-item">
1480         Spring24, TR, 1:15pm - 2:30pm, FGH 134.
1481       </li>
1482     </ul>
1483     <div class="card-footer text-muted small">
1484       Vanderbilt University
1485     </div>
1486   </div>
1487 </div>
1488 </div>
1489 </div>
1490
1491
1492
```

What Is Software Engineering?

What Is Software Engineering?

- The majority of industrial software engineering is *not* writing code.
- The dominant activities in software engineering are **comprehension** and **maintenance**.





“Understanding code is by far the activity at which professional developers spend most of their time.”

[Peter Hallam. *What Do Programmers Really Do Anyway?* Microsoft.]

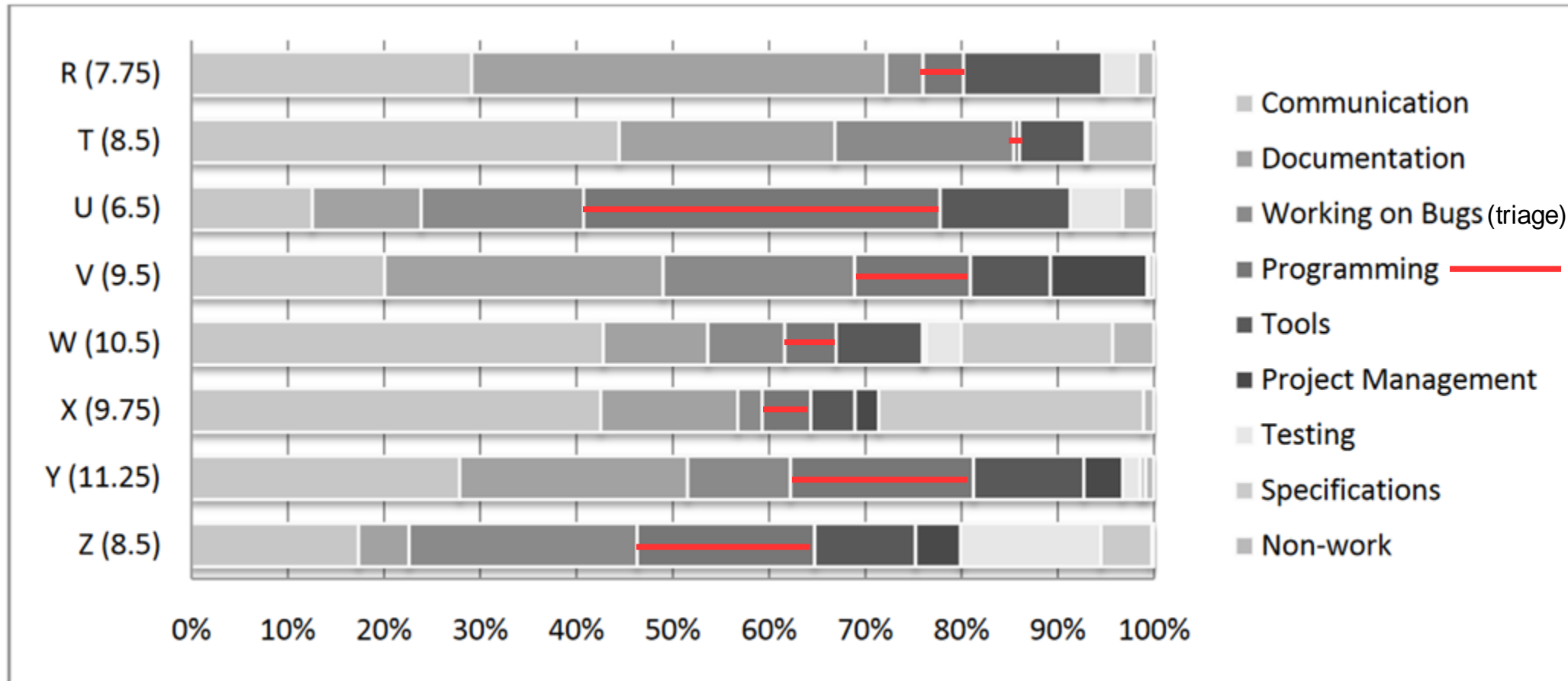
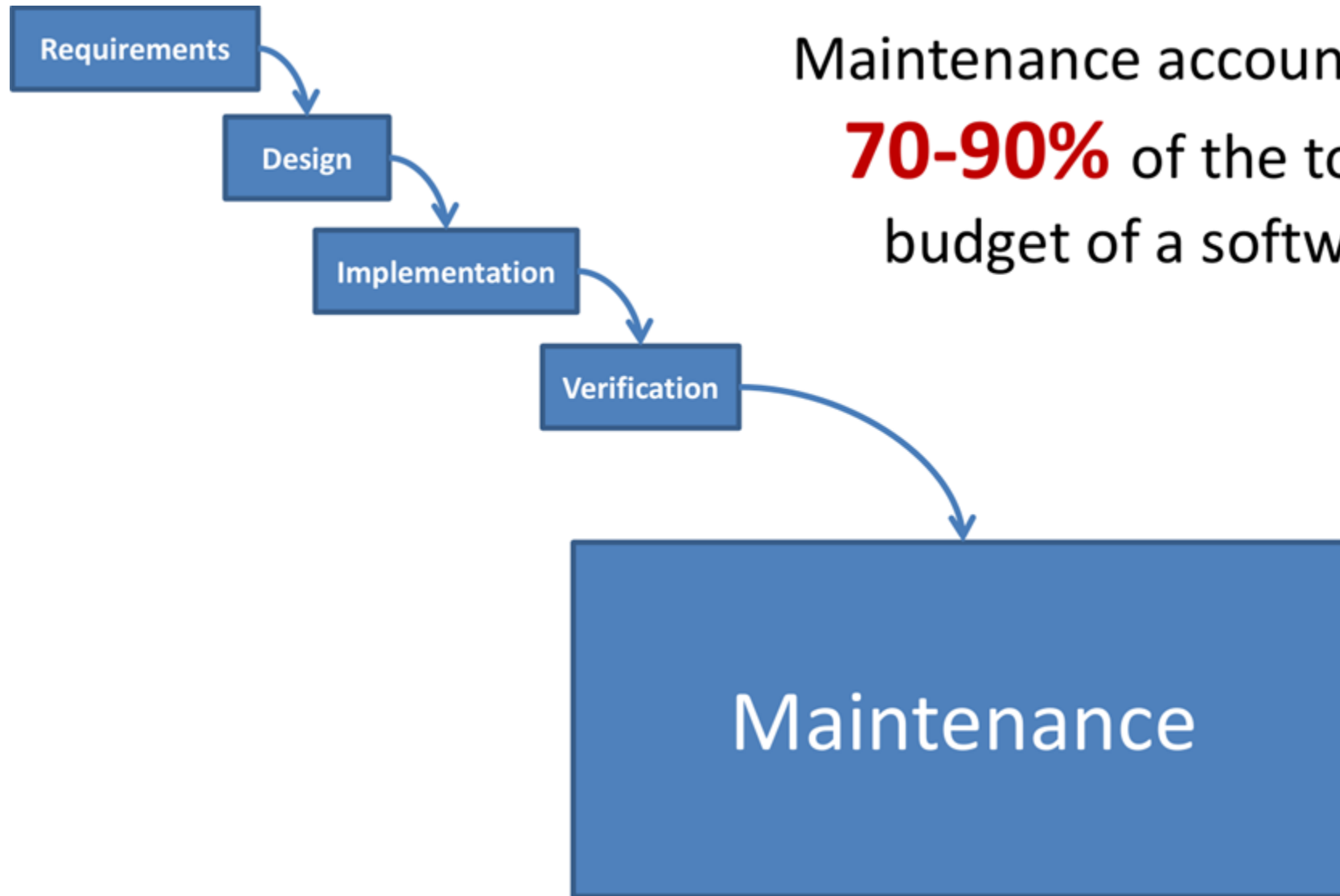


Figure 1. Tasks by time for *each* subject, normalized by the total time + time where events overlapped in each observation. Total observation time in hours is listed in parentheses after each subject's identification letter.

- Hour logging of new devs (1-7 months) at Microsoft: **programming** is 10-20% of the time.

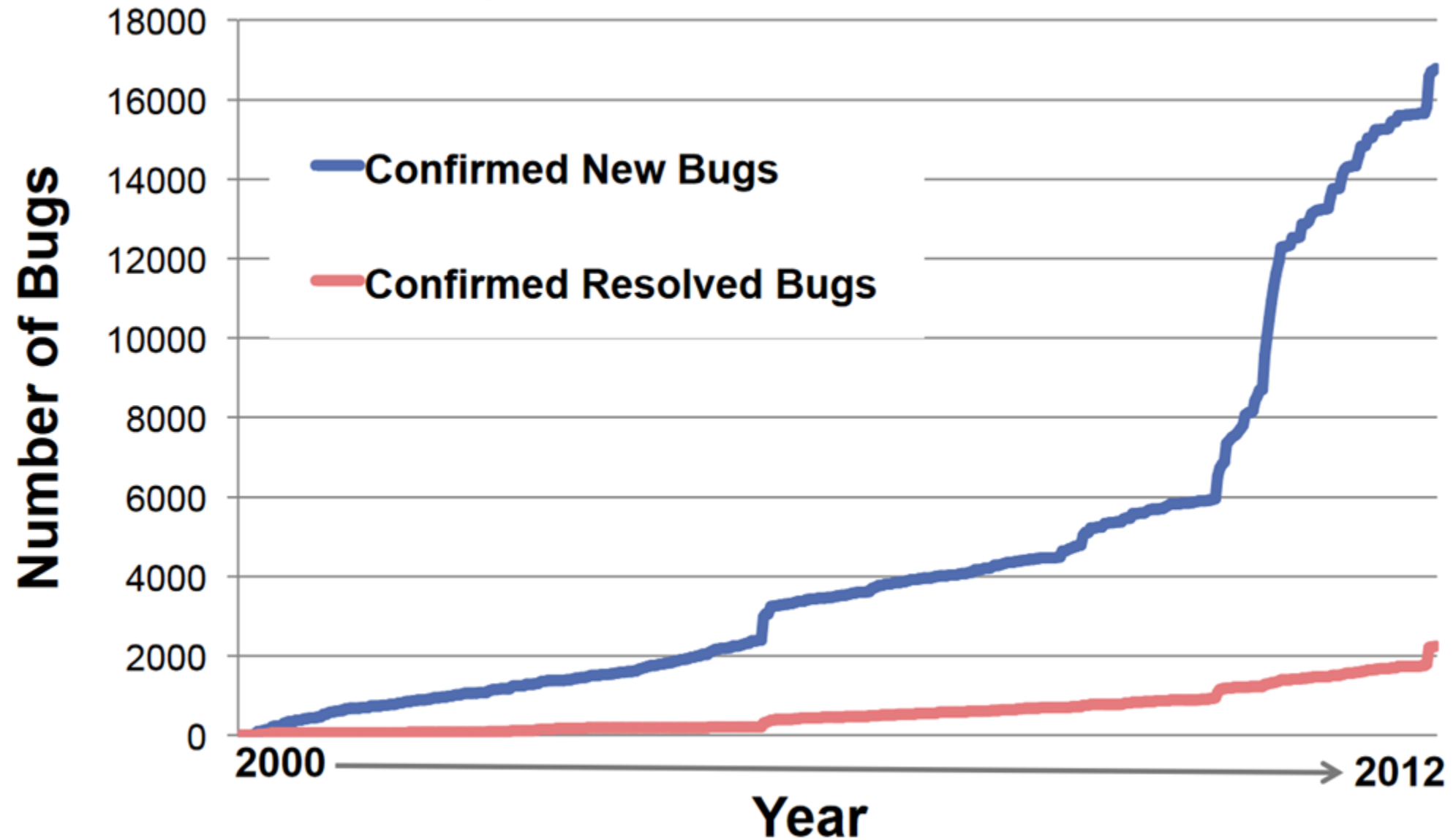
[Begel and Simon. *Novice Software Developers, All Over Again*. Computing Education Research, September 2008. Microsoft.]



Maintenance accounts for about **70-90%** of the total lifecycle budget of a software project.

[Pigoski. *Practical Software Maintenance: Best Practices for Managing Your Software Investment*. Seacord, Plakosh, and Lewis. *Modernizing Legacy Systems: Software Technologies*.]

OpenOffice bugs: 2000-2012



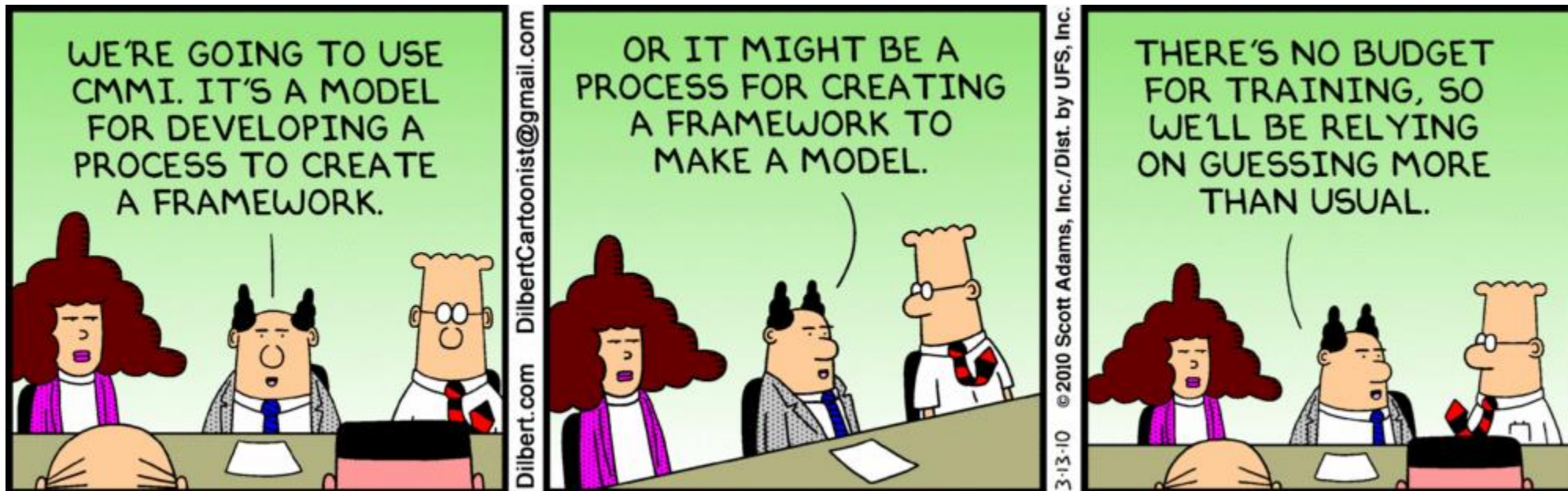
[Fry. *Leveraging Light-Weight Analyses to Aid Software Maintenance*. 2014]

A Key Issue

“Half of software engineering is crap.”

– Prof. Weimer

(Prof. Leach thinks it's more like two-thirds crap – Prof. Huang seconds that)



Class Philosophy

“Anyway, here's the '**good parts**' version. S. Morgenstern wrote it. And my father read it to me. And now I give it to you. What you do with it will be of more than passing interest to us all.”

– William Goldman, *The Princess Bride*



This Course


- <https://huang.isis.vanderbilt.edu/cs4278>
- Let's walk through the website together – it is so important!!
 - Administrivia
 - Assignments and Grading
 - Outline of Topics



How will this help me graduate?



- Upper-Level CS Course
- Prerequisite for CS4279
- Required for graduation?

- 
- Major Design Experience
 - Capstone

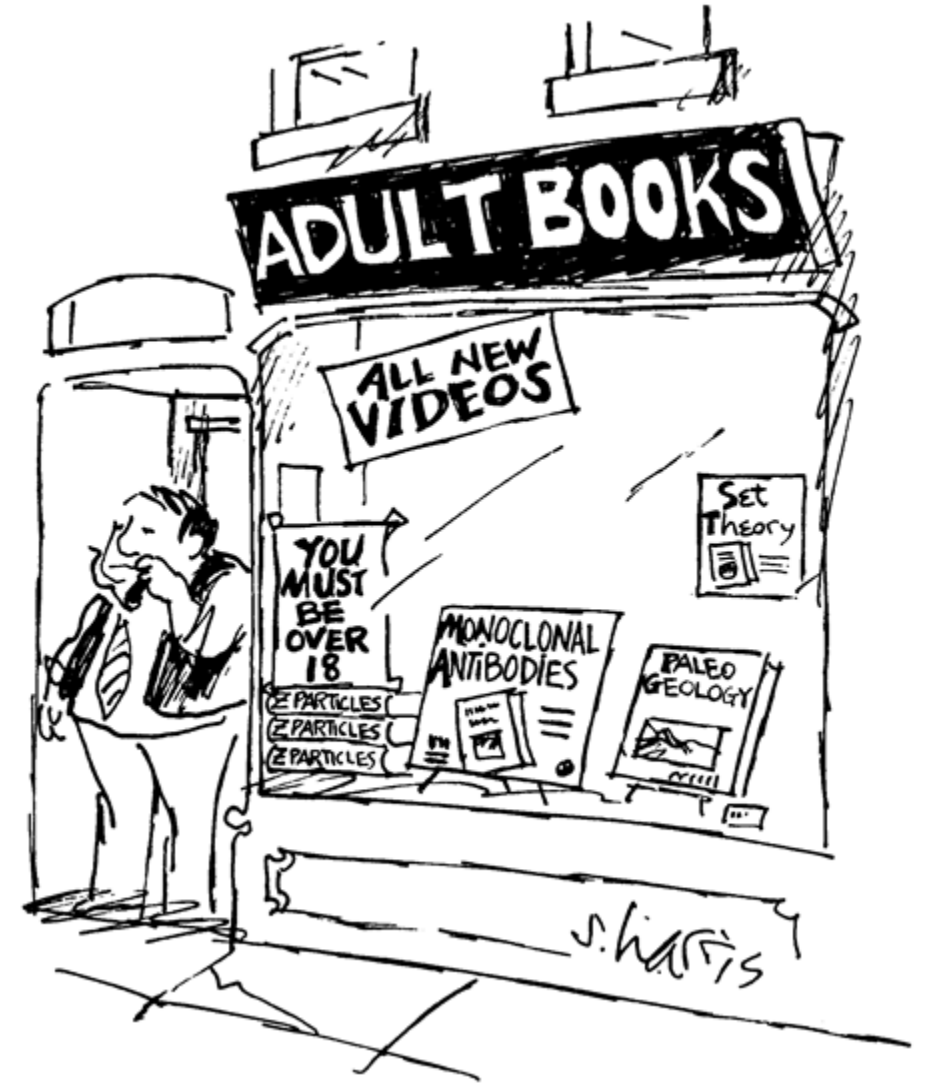
Who should take this class?

- Prepare yourself for a software development job
- Want to know what modern software engineering is like
- Want to learn about basic research topics in Software Engineering
 - Want to learn more about SE research?
 - Consider CS8395 – Advanced Topics in SE

How Hard Is This Course?

- Workload Survey is misleading!
 - Easier than the hardest class @ Vandy CS
- More “**time consuming**” than “difficult”
- A lot of practical skills
- See webpage quotes from former students!

Welcome to the real world!



Blah blah attendance

(ur old, etc...)

when the professor who banned phones and laptops in class now has to learn how to teach online 🤔

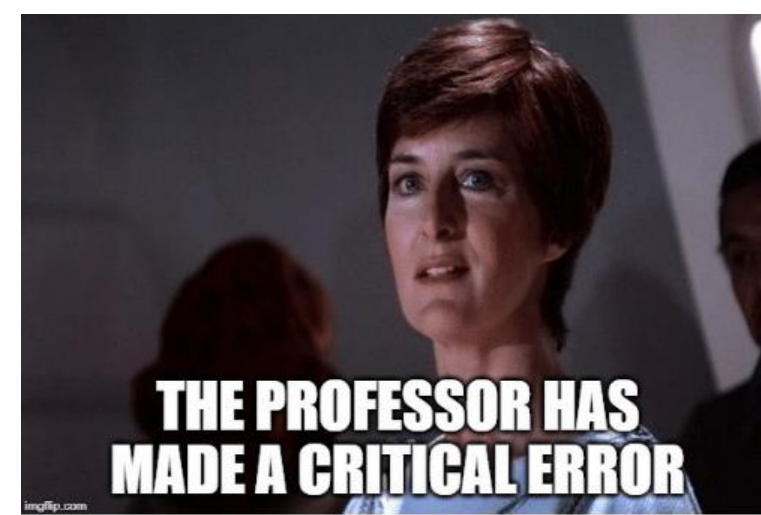


ing education technology, overwrought fears about the perils of technology have proven equally exaggerated. Those apprehensive about computer-assisted tutoring or online instruction would do well to keep in mind that such concerns have greeted almost any new learning tool. Dave Thornburg and David Dwyer, for instance, offer up a list of past complaints in their book *Rethinking Education in the Age of Technology: The Digital Revolution and Schooling in America*. From today's vantage point, some of the concerns make for amusing reading:

From a principal's publication, 1815: "Students today depend on paper too much. They don't know how to write on a slate without getting chalk dust all over themselves. They can't clean a slate properly. What will they do when they run out of paper?"

Attendance

- I would prefer you attend the lectures live as they are delivered
 - Pay more attention synchronously
 - Opportunity to ask questions before you forget
- We will have random quizzes on some lectures (evil!)



When the syllabus says they don't take attendance and participation doesn't matter

A meta-analysis of the relationship between class attendance in college and college grades reveals that attendance has strong relationships with both class grades ($k = 69$, $N = 21,195$, $\rho = .44$) and GPA ($k = 33$, $N = 9,243$, $\rho = .41$). These relationships make class attendance a better predictor of college grades than any other known predictor of academic performance, including scores on standardized admissions tests such as the SAT, high school GPA, study habits, and study skills. Results also show that class attendance explains large amounts of unique variance in college grades because of its relative independence from SAT scores and high school GPA and weak relationship with student characteristics such as conscientiousness and moti-

Crede et al. "Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance With Grades and Student Characteristics." *Review of Educational Research*, 2010. Vol. 80. DOI: 10.3102/0034654310362998

Attendance



Welcome to CS 4278/5278!

Hello class! Welcome to CS 4278/5278 -- Principles of Software Engineering!

<https://huang.isis.vanderbilt.edu/cs4278/>

This course is a "good parts version" of the topics covered in software engineering courses at top CS departments. It focuses on the dominant activities that modern industrial software engineers wrestle with: reading and modifying code (i.e., software maintenance). It also covers defects, measurement, uncertainty, productivity and even brains -- to a lesser degree.

Many of the lectures and homework assignments are already available on the webpage (I am still finalizing some details, such as the OH time of all course staff and one newer HW, should be finalized soon). You can take a look at them, as well as advice from former students, if you are curious.

Here I want to point out several things most of you might care about before the first lecture:

(1) Do I need to prepare a certain textbook?

-- Nope

(2) Why cannot I see any (useful) information about this class on Brightspace?

-- We will only use Brightspace for pdf submissions (i.e., homework reports) and grading release. All the (useful) information about this course is on the course website (listed above).

(3) Do I need to bring a laptop to the class?

-- You will need to bring a laptop to the class for only one of the lectures (Productivity; check the course website:Lectures). Other than that, you are welcome to bring your laptop if that is how you make notes. But a laptop is not required during lectures.

However, you may want to bring a pen and paper with you (why? I will explain it in our first lecture).

I hope to see you all in class tomorrow!

ge
ith
43,
ol-
ce,
gh
en-
use
rak
oti-

Crede et al. "Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance With Grades and Student Characteristics." *Review of Educational Research*, 2010. Vol. 80. DOI: 10.3102/0034654310362998

Quiz Example

- Not for credit, just an example!

1. Name:_____ VUID:_____ Email:_____ (1 point)

2. Who killed Dumbledore? (1 point)

3. Is CS4278/5278 (Yu's session) a project/capstone course (1 points)?

4. Where can you find all the course materials including lectures and all the deadlines (2 points)?

Communication Resources

- Use **Piazza** for questions
 - Also feel free to be informal
 - Help each other
- “Code of Conduct”

Assignments and Grading

- Assigned reading due **before each lecture**
 - Where can you find them?
- Normal due dates even if you add late!
- Attend lectures, take notes, visit forum



TR [Process, Risk and Scheduling](#)
01/11/24 [Scheduling](#)
[overview]

- [Wikipedia's Software Development Process](#)
- [Buse and Zimmermann's Information Needs for Software Development Analytics](#) (if you are low on time, read only Sections I-V)
[Microsoft]
- [HW1 Specification](#)
- Optional: "How well do real companies plan and estimate effort?" Find out in: [Anda et al.'s Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System](#)
[Simula]

Readings and Quizzes

- No expensive, outdated textbook
- Assigned reading to be done before lectures
 - High-level summaries (e.g., Wikipedia)
 - Industrial tech reports and academic research
 - Homework assignment instructions
 - Optional readings for further exploration (not included in quizzes)
- Higher standard than the CS usual

Assignments: HW 0-5

- Six Assignments
 - HW 0: Optional
 - HW 1-5: Test Coverage, Test Automation, Mutation Testing, Defect Detection and Unit Testing, Debugging Automation
 - [HW 6](#)
- Coding: **autograder**
- Multiple languages (C, C++, Java, Python, etc.), bash
- Writing: **Brightspace**
- Due dates posted in advance (now!)
- Materials available in advance (now!)

Homework 0 — Dev Setup (Optional)	1%
Homework 1 — Test Coverage	10%
Homework 2 — Test Automation	10%
Homework 3 — Mutation Testing	10%
Homework 4 — Defect Detection and Unit Testing (No Teams Allowed)	13%
Homework 5 — Debugging Automation	10%
Homework 6 — Contribution	15%
Comprehension Quizzes	5%
Participation and Professionalism	2%
Examination 1	12%
Examination 2	12%

Assignments: HW 6

- **HW6 – Undergrad**

- HW6a, b
- Open Source GitHub Contribution
- A lot of examples

- **HW6 - Grad**

- HW6a, b
- 7-page NSF style research proposal
- Less examples
- Schedule meetings
- Exceptions apply: Talk to me if you have taken CS8395 – Advanced Topics in Software Engineering

Warnings for the Assignments (everyone)

- It is the students' responsibility to set up the environment and make sure it works on your machine.
 - HWO
- “I feel the homework in this class is very **different** from other CS courses I have taken. I don't like it.”

Participation and Professionalism

- By default
- “Code of Conduct”
 - Piazza
 - Class
- Cheating

What kind of person do you want to be?

Check “why you shouldn’t cheat”:

<https://www.youtube.com/watch?v=hMloyp6NI4E>

Homework 0 — Dev Setup (Optional)	1%
Homework 1 — Test Coverage	10%
Homework 2 — Test Automation	10%
Homework 3 — Mutation Testing	10%
Homework 4 — Defect Detection and Unit Testing (No Teams Allowed)	13%
Homework 5 — Debugging Automation	10%
Homework 6 — Contribution	15%
Comprehension Quizzes	5%
Participation and Professionalism	2%
Examination 1	12%
Examination 2	12%

Exams

- Two-hour online exam? Maybe
 - You pick a two-hour block of time in a 24-hour window.

Homework 0 — Dev Setup (Optional)	1%
Homework 1 — Test Coverage	10%
Homework 2 — Test Automation	10%
Homework 3 — Mutation Testing	10%
Homework 4 — Defect Detection and Unit Testing (No Teams Allowed)	13%
Homework 5 — Debugging Automation	10%
Homework 6 — Contribution	15%
Comprehension Quizzes	5%
Participation and Professionalism	2%
Examination 1	12%
Examination 2	12%

Changeups and Trivia

- “[Professors who] deliberately and consistently interspersed their lectures with ... some other form of deliberate break ... usually commanded a better attention span from the class, and these deliberate variations had the effect of postponing or even eliminating the occurrence of an attention break”

[Johnstone and Percival. *Attention breaks in lectures*. *Education in Chemistry*, 13. 49-50, 1976.]

[Middendorf and Kalish. *The “Change-up” in Lectures*. *TRC Newsletter*, 8:1 (Fall 1996).]

Computer Science

- *This* English mathematician and writer published the first algorithm (~1842) to be carried out by a general-purpose computer and is often called the first computer programmer.

Computer Science

- *This* English mathematician and writer published the first algorithm (~1842) to be carried out by a general-purpose computer and is often called the first computer programmer.



Ada Lovelace

Computer Science

- *What* did that first program do?

Number of Operations.	Nature of Operations.	Variables acted upon.	Variables receiving results.	Indication of change in the value of any Variable.	Statement of Results.	Data.												Working Variables.			Result Variables.							
						V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉	V ₁₀	V ₁₁	V ₁₂	V ₁₃	V ₁₄	V ₁₅	V ₁₆	V ₁₇	V ₁₈	V ₁₉	V ₂₀	V ₂₁	V ₂₂	V ₂₃
1	X	V ₁ × V ₂	V ₃ , V ₄ , V ₅ , V ₆	V ₃ = V ₂ V ₄ = V ₁ V ₅ = V ₂ V ₆ = V ₁	= 2n	1	2	n	2n	2n	2n																	
2	-	V ₄ - V ₁	V ₇	V ₇ = V ₄ - V ₁	= 2n - 1	1			2n - 1																			
3	+	V ₇ + V ₁	V ₈	V ₈ = V ₇ + V ₁	= 2n + 1	1				2n + 1																		
4	+	V ₈ + V ₁	V ₉	V ₉ = V ₈ + V ₁	$\frac{2n-1}{2n+1}$				0	0																		
5	+	V ₉ + V ₁	V ₁₀	V ₁₀ = V ₉ + V ₁	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$		2																					
6	-	V ₁₀ - V ₁	V ₁₁	V ₁₁ = V ₁₀ - V ₁	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_1$																							
7	-	V ₁₁ - V ₁	V ₁₂	V ₁₂ = V ₁₁ - V ₁	= n - 1 (= 3)	1		n																				
8	+	V ₁₂ + V ₁	V ₁₃	V ₁₃ = V ₁₂ + V ₁	= 2 + 0 = 2		2																					
9	+	V ₁₃ + V ₁	V ₁₄	V ₁₄ = V ₁₃ + V ₁	$\frac{2n}{2} = A_1$																							
10	X	V ₁₄ × V ₁₁	V ₁₅	V ₁₅ = V ₁₄ × V ₁₁	$B_1 \cdot \frac{2n}{2} = B_1 A_1$																							
11	+	V ₁₅ + V ₁₁	V ₁₆	V ₁₆ = V ₁₅ + V ₁₁	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$																							
12	-	V ₁₆ - V ₁	V ₁₇	V ₁₇ = V ₁₆ - V ₁	= n - 2 (= 2)	1																						
13	-	V ₁₇ - V ₁	V ₁₈	V ₁₈ = V ₁₇ - V ₁	= 2n - 1	1																						
14	+	V ₁₈ + V ₁	V ₁₉	V ₁₉ = V ₁₈ + V ₁	= 2 + 1 = 3	1																						
15	+	V ₁₉ + V ₁	V ₂₀	V ₂₀ = V ₁₉ + V ₁	$\frac{2n-1}{3}$																							
16	X	V ₂₀ × V ₁₁	V ₂₁	V ₂₁ = V ₂₀ × V ₁₁	$\frac{2n-2n-1}{3}$																							
17	-	V ₂₁ - V ₁	V ₂₂	V ₂₂ = V ₂₁ - V ₁	= 2n - 2	1																						
18	+	V ₂₂ + V ₁	V ₂₃	V ₂₃ = V ₂₂ + V ₁	= 3 + 1 = 4	1																						
19	+	V ₂₃ + V ₁	V ₂₄	V ₂₄ = V ₂₃ + V ₁	$\frac{2n-2}{4}$																							
20	X	V ₂₄ × V ₁₁	V ₂₅	V ₂₅ = V ₂₄ × V ₁₁	$-\frac{2n-2n-1}{4} = A_2$																							
21	X	V ₂₅ × V ₁₁	V ₂₆	V ₂₆ = V ₂₅ × V ₁₁	$-\frac{2n-2n-1}{3} = B_2 A_2$																							
22	+	V ₂₆ + V ₁₁	V ₂₇	V ₂₇ = V ₂₆ + V ₁₁	$A_2 + B_2 A_2 + B_2 A_2$																							
23	-	V ₂₇ - V ₁	V ₂₈	V ₂₈ = V ₂₇ - V ₁	= n - 3 (= 1)	1																						
Here follows a repetition of Operations thirteen to twenty-three.																												
24	+	V ₂₈ + V ₁	V ₂₉	V ₂₉ = V ₂₈ + V ₁	= B ₂																							
25	+	V ₁ + V ₂₉	V ₃₀	V ₃₀ = V ₁ + V ₂₉ by a Variable-card. V ₁ = V ₂₉ by a Variable-card.	= n + 1 = 4 + 1 = 5	1		n + 1																				

Psychology:

The Fundamental Attribution Error

- The **fundamental attribution error** is that people emphasize internal characteristics when explaining the behavior of **others** but external factors when explaining their own behavior.
 - Example: cutting someone off in traffic.
- In an experiment (Jones and Harris, 1967), subjects read pro- and anti-Castro essays. Then they were asked to rate the pro-Castro attitudes of the writers. Conditions:
 - When subjects believed the writers choose the positions freely:
 - Expect: “pro-Castro” essays → positive attitude to Castro
 - When subjects believed the positions were determined by a coin toss:
 - Expect: neutral attitude on average (we don’t know)

Psychology:

The Fundamental Attribution Error

- Experimental findings:
 - Even when they knew the position came from a coin toss, subjects rated pro-Castro writers as having a positive Castro attitude.
 - “The subjects were unable to properly see the influence of the situational constraints placed upon the writers; they could not refrain from attributing sincere belief to the writers.”
- [Jones, E. E.; Harris, V. A. (1967). *"The attribution of attitudes"*. Journal of Experimental Social Psychology. 3 (1): 1–24.]
- SE Implication: **Teamwork**. Be careful when you see defects (mine just mean I made a typo, others mean they are stupid).

Extra Credits

- Extra credit questions in exams
 - Integrated in your exam score
- Extra credit in HW4
 - Unit testing performance
 - Make fun of your instructor – I haven't finalized the details and threshold yet!

Extra Credits: HW6*, +1

- **HW6 – Undergrad**

- HW6a, b
- Open Source GitHub Contribution
- A lot of examples

- **HW6 - Grad**

- HW6a, b
- 5-page NSF style research proposal
- Less examples
- Schedule meetings

Extra Credits: HW6*, +1

- **HW6 – Undergrad**

- HW6a, b
- Open Source GitHub Contribution
- A lot of examples

- **Your Pull Request gets merged (before the Apr 21 deadline)!**

- **HW6 - Grad**

- HW6a, b
- 5-page NSF style research proposal
- Less examples
- Schedule meetings

- **You finished the proposed work (before the Apr 21 deadline)! Ofc this means the project cannot be a project you have done before.**

Guest Lecture

- One guest lecture
 - Industrial software development
- Update the dates once confirmed with the guest speaker
- Zoom

Optional Teams

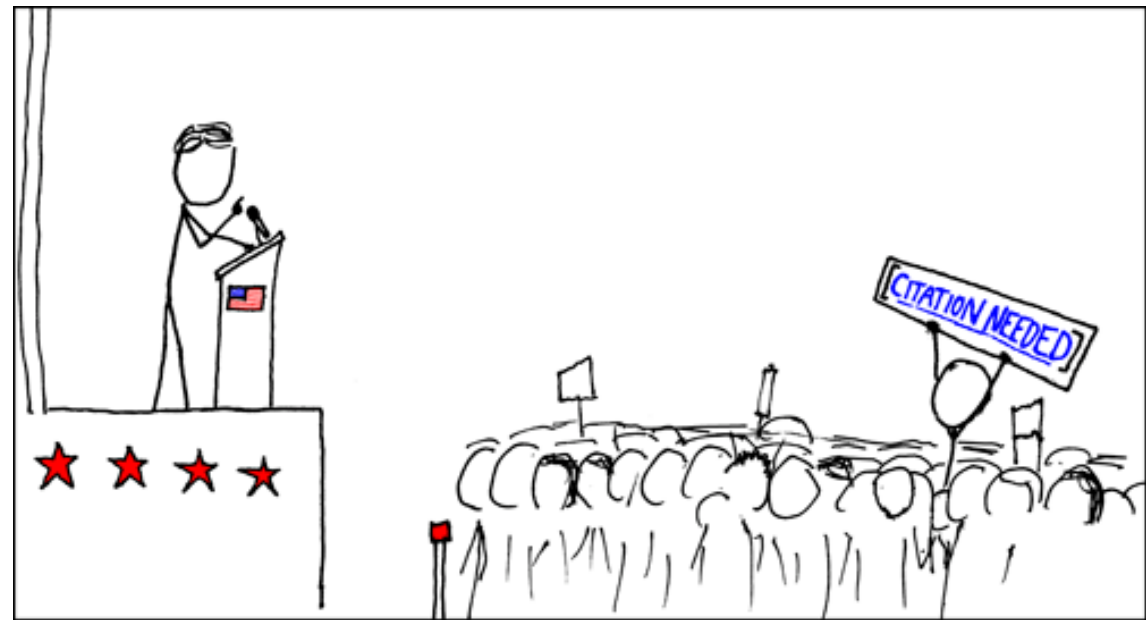
- *Modern industrial software engineering is almost exclusively team-based*
- But this is an ULCS, not a Capstone/MDE
 - You will be exposed to building a large project in a team elsewhere in the curriculum
- For most of the assignments, you may work **alone or in pairs** of your own choosing (**NOT for HW4; you have to work alone for HW4**)
 - HW6: no teaming between undergraduate and graduate students
 - **We are not responsible if your partner disappears**
 - Use piazza to find partners, etc.

Grace Period and Late Policy (read more on website)

- Two grace periods in total for each student (no matter in teams or individual)
 - 4-day extension without any penalty
 - Must email ALL course staff by the original deadline
 - Each homework can be applied with no more than one GP
 - Teams must use grace periods together (if one has no GP available, then the team cannot use it)
 - NOT applied to HW6b
 - HW4 does not allow teams
- Late policy: w/o GP, h% off – h is the number of hours late (no more submission allowed after 4 days – 96 hours)

Software Engineering You Can Believe In

- Citations for strong claims (or ask on forum)
- Guest Lectures
 - Large companies, startups, researchers, etc.
- Readings from Industry
- Material from
 - Westley Weimer @ UM
 - Kevin Leach @ Vanderbilt
 - Prem Devanbu @ UCSD
 - Christian Kästner @ CMU
 - Claire Le Goues @ CMU
 - Ciera Jaspen @ Google



Core Course Topics

- Measurement and Risk
 - Process, scheduling, and information
- **Quality Assurance**
 - Code review, **testing**, and **analysis**
- Software Defects
 - Reporting and localizing
- Software Design
 - Requirements, ~~patterns~~, and maintainability
- Productivity at Scale
 - People, teams, interviews, and brains

Course Themes

- Software engineering is a human process
- Software engineering deals with large scales
- Software engineering requires strategic thinking
- Software engineering is constrained by reality

Analogy: Engineering Envy

- **Producing** a car
 - Estimate costs, risks
 - Expected results
 - High quality
- Separate plan and production
- Simulate before constructing
- Quality assurance through measurement
- Potential for automation



Dangerous Analogy

- Producing a car
 - Estimate costs, risks
 - Expected results
 - High quality
 - Separate plan and production
 - Simulate before constructing
 - Quality assurance through measurement
 - Potential for automation
- Software = Design = Plan
 - Programming is design, **not production**
 - Production (copying/loading a program) is automated
 - Simulation is not necessary
 - Quality measurement?

Software Engineering

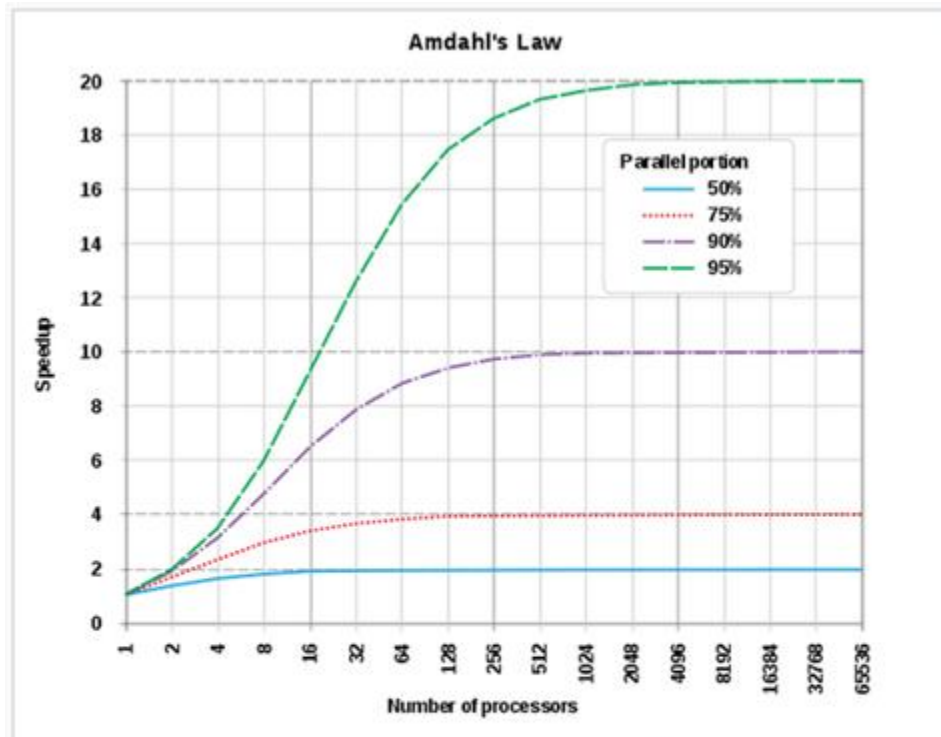
- “My favorite operational definition of engineering is '**design under constraint.**' Engineering is creating, designing what can be, but it is constrained by nature, by cost, by concerns of safety, reliability, environmental impact, manufacturability, maintainability, and many other such 'ilities.'”

[Bill Wulf, NAE President, The Urgency of Engineering Education Reform, 2008]

- “[Software Engineering is] The Establishment and use of sound **engineering principles** in order to obtain **economically** software that is **reliable** and works **efficiently on real** machines.”

[Fritz Bauer 1975, S. 524]

Measurement Teaser



Evolution according to Amdahl's law of the theoretical speedup in latency of the execution of a program in function of the number of processors executing it, for different values of p. The speedup is limited by the serial part of the program. For example, if 95% of the program can be parallelized, the theoretical maximum speedup using parallel computing would be 20 times.

- What is Amdahl's Law?
- Suppose you want a program to run faster
- Suppose you want software to be created and sold faster

Quality Assurance Teaser

- To assess quality, we can look at the source code or run the program
- Testing is the dominant approach here
- But not all test suites are created equal!
- Statement coverage, branch coverage

- Mutation testing
- Automated test generation

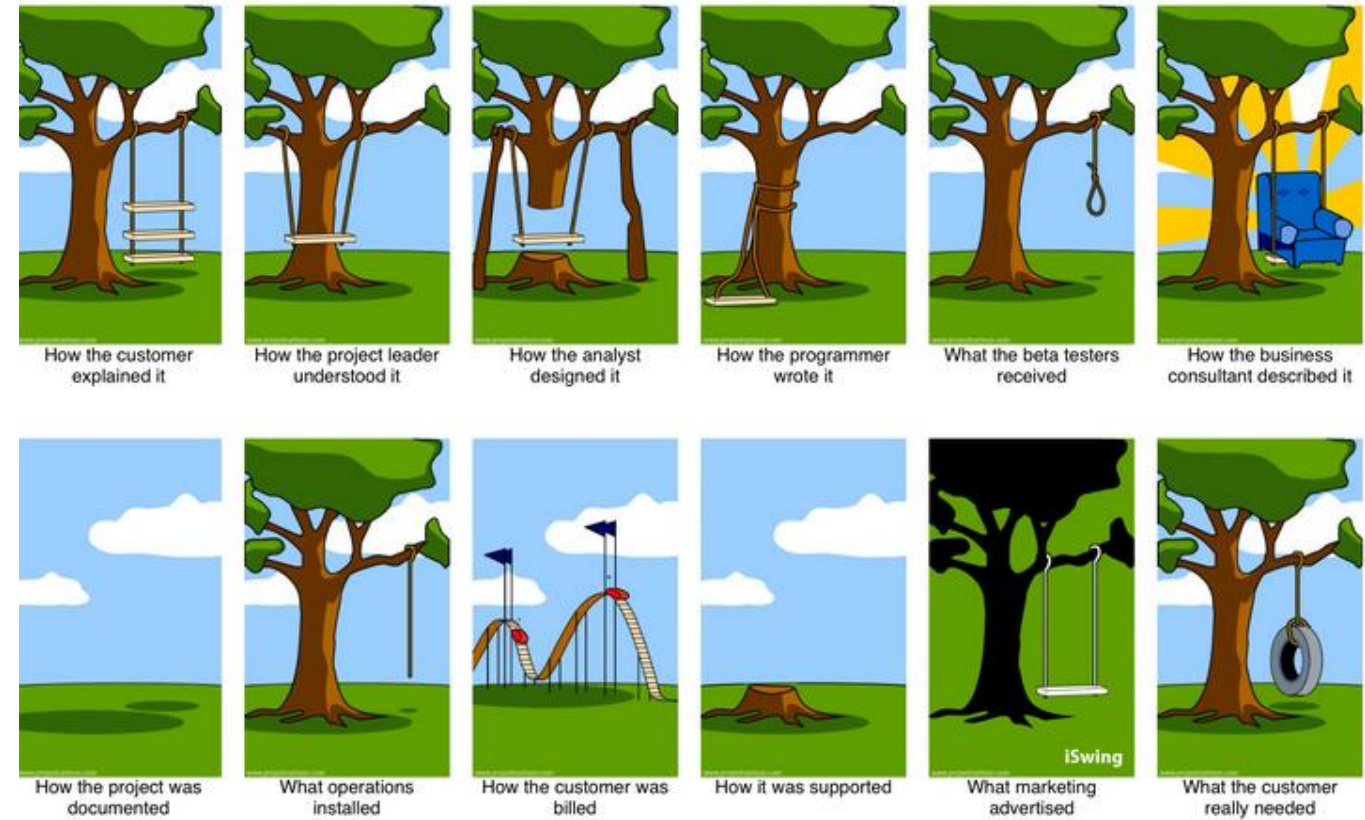
Defect Teaser

- Just put in print statements
- Find the line with the bug
- Flail around, resubmit until it passes

- Automatic fault localization
- Debugging as Hypothesis Testing

Design Teaser

- Requirements and Specifications
 - How can we elicit what people actually want?
- Validation and Risk
- Design for Maintainability



Productivity Teaser

- The ratio of programming time and program performance between novices and experts has been published at up to 28:1
 - Why?
- Pair Programming, Agile, etc.
- How do experts and novices think?
- Medical Imaging Studies
- More discussion on human factors in Software Engineering

Research

- Recruiting undergraduate and graduate researchers in my group
 - Email me your CV and research interests
 - Research credits, VUSRP, SUGRE, REU
- All of students' data in this class may be used in future research (**de-identified**). **You will need to notify the instructor by emails if you don't want your data to be included.**
 - For example, home assignments, project reports.
- Support research recruitment
 - Piazza advertisement
 - In-class advertisement
 - Will **NOT** affect your grades

Questions?

- You are responsible for all assignments at their listed times even if you add the course late.
 - Ask TA to add you to the Piazza, Autograder and Brightspace (session 02) if you aren't in yet