Final Thoughts

17-313: Foundations of Software Engineering

https://cmu-313.github.io

Michael Hilton and **Josh Sunshine**

Spring 2026

Administrivia

Final presentations:

-Thursday, April 30, 2026
-08:30am - 11:30am
-TEP 1403

Snacks survey: https://forms.gle/T8ZoDh3XHQ4YHDiJ7

Fall 26 TA interest form: https://forms.gle/Zv3Lq4AWAAncrx7K8

AMA

Ask the professors any question you want

Semester Recap

Michael Hilton

Full Teaching Professor at CMU

A.S. Grossmont Community College 1999

B.S. San Diego State University - 2002

Software Engineer at DoD - 2002 to 2011

M.S. Cal Poly San Luis Obispo - 2013

PhD at Oregon State - 2017

Internship at Microsoft Research - Summer 2017

Assistant/Associate/Full Teaching Professor at CMU - Fall 2017 to current

Josh Sunshine

Associate Professor

B.A. Brandeis University, 2004.

Ph.D., Carnegie Mellon University, 2013

Faculty @ CMU since 2014

Teaching Assistants

Autumn Qiu

Juan Ageitos

Bing Bhakdibhumi

Alejandro Estrada

Kaia Newman

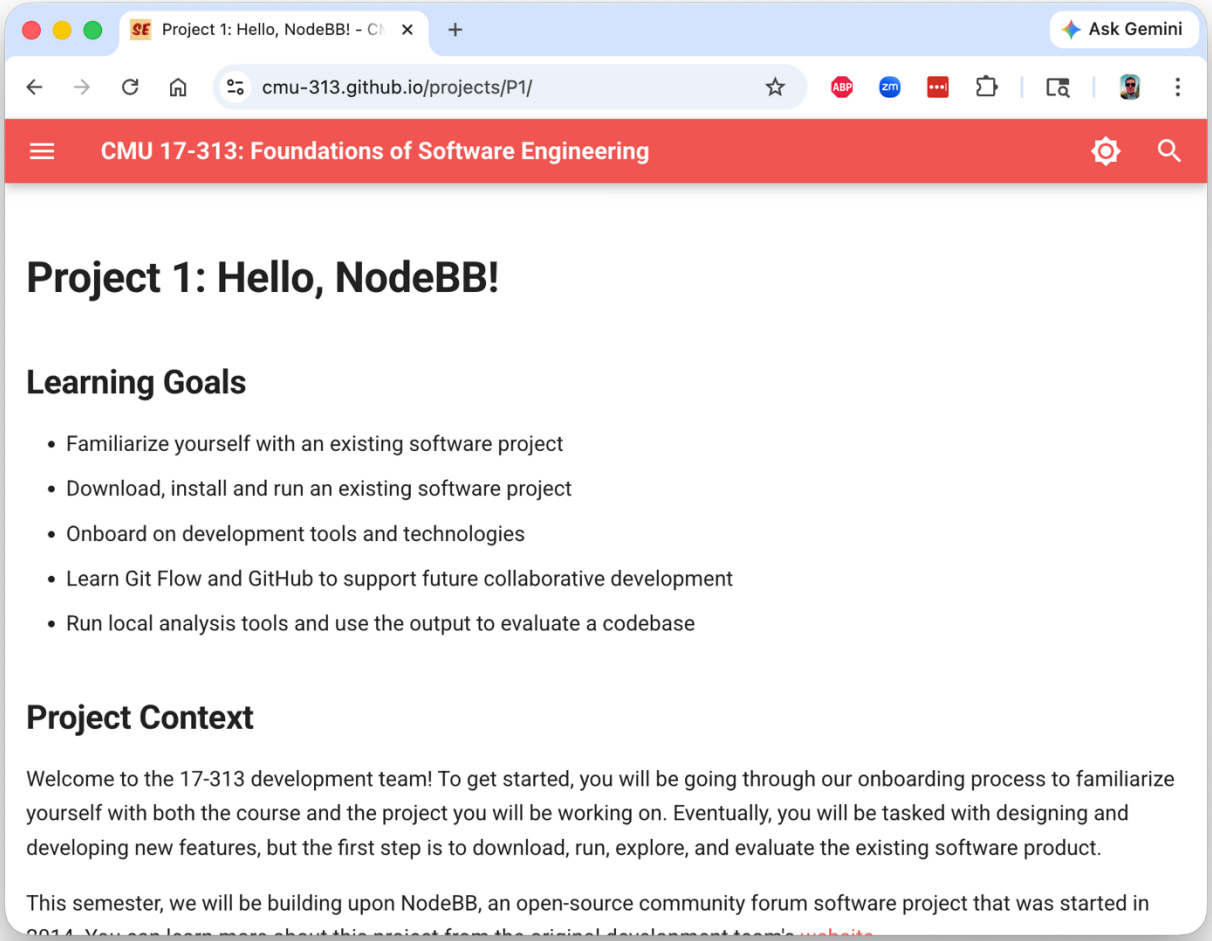
Hwei-Shin Harriman

Lisa Huang

Vasa

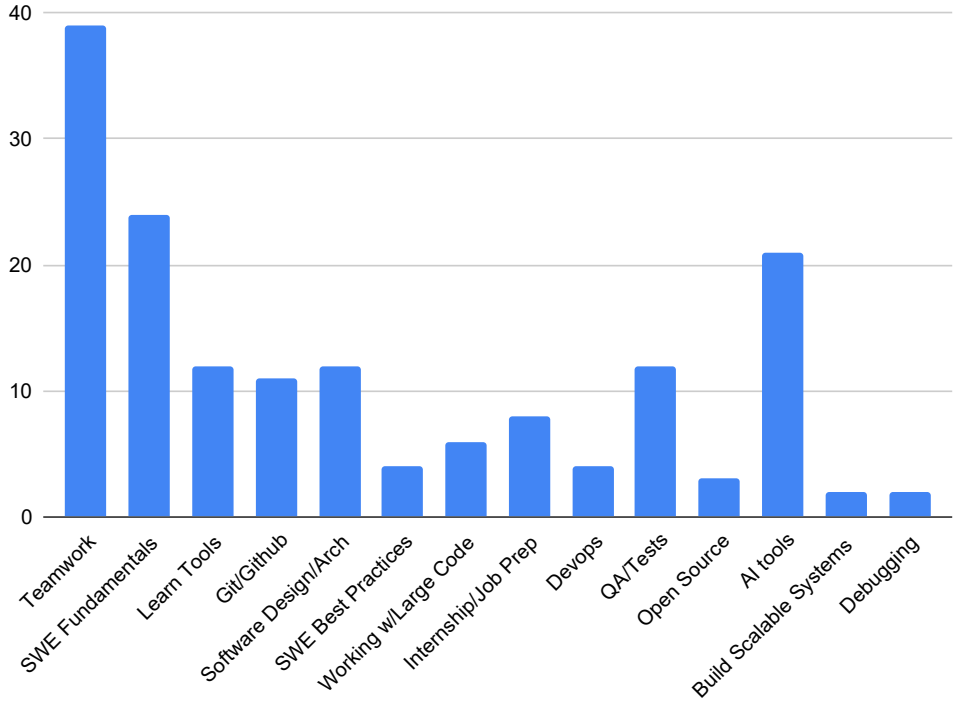
Smoking Section

- Last full row


Project 1: Hello, NodeBB! - Cl
Ask Gemini
cmu-313.github.io/projects/P1/
CMU 17-313: Foundations of Software Engineering
Project 1: Hello, NodeBB!
Learning Goals
Familiarize yourself with an existing software project
Download, install and run an existing software project
Onboard on development tools and technologies
Learn Git Flow and GitHub to support future collaborative development
Run local analysis tools and use the output to evaluate a codebase
Project Context
Welcome to the 17-313 development team! To get started, you will be going through our onboarding process to familiarize yourself with both the course and the project you will be working on. Eventually, you will be tasked with designing and developing new features, but the first step is to download, run, explore, and evaluate the existing software product.
This semester, we will be building upon NodeBB, an open-source community forum software project that was started in

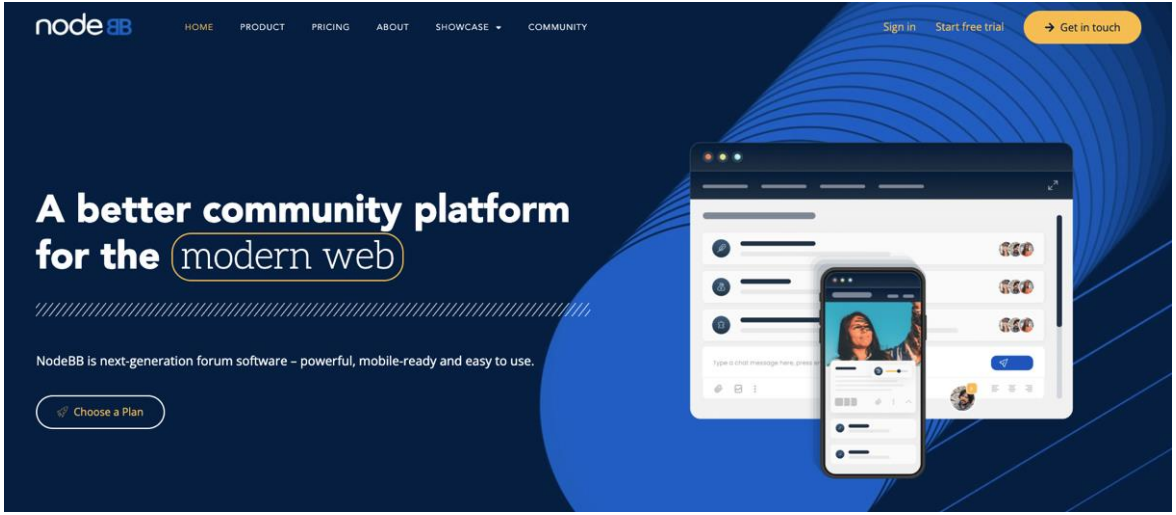
If you automate your thinking, you automate away the learning.

Hope to learn (from last class)

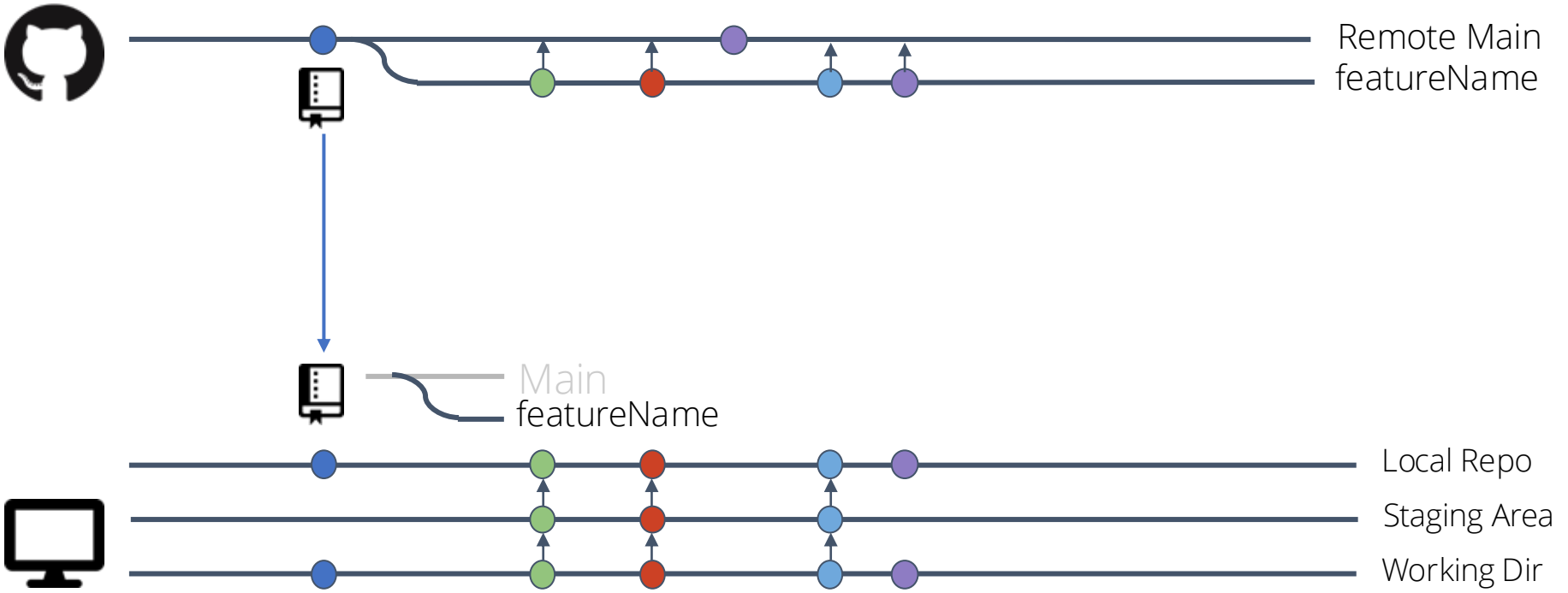


Context: big ole pile of code

- … do something with it!



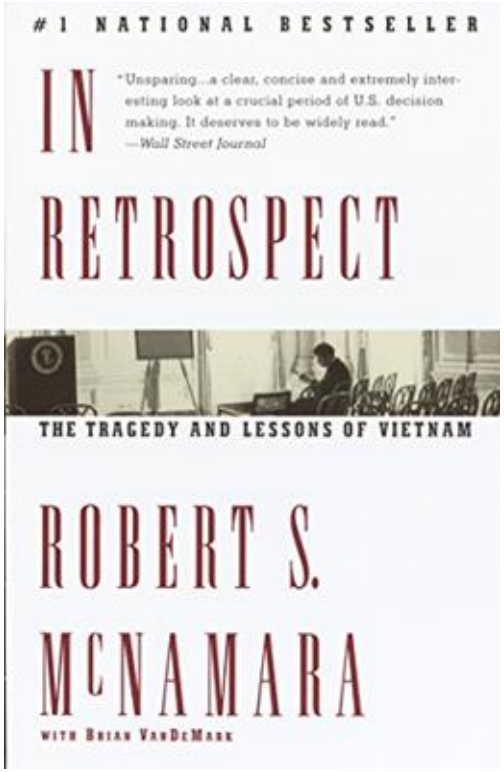
Open Pull Request on Github

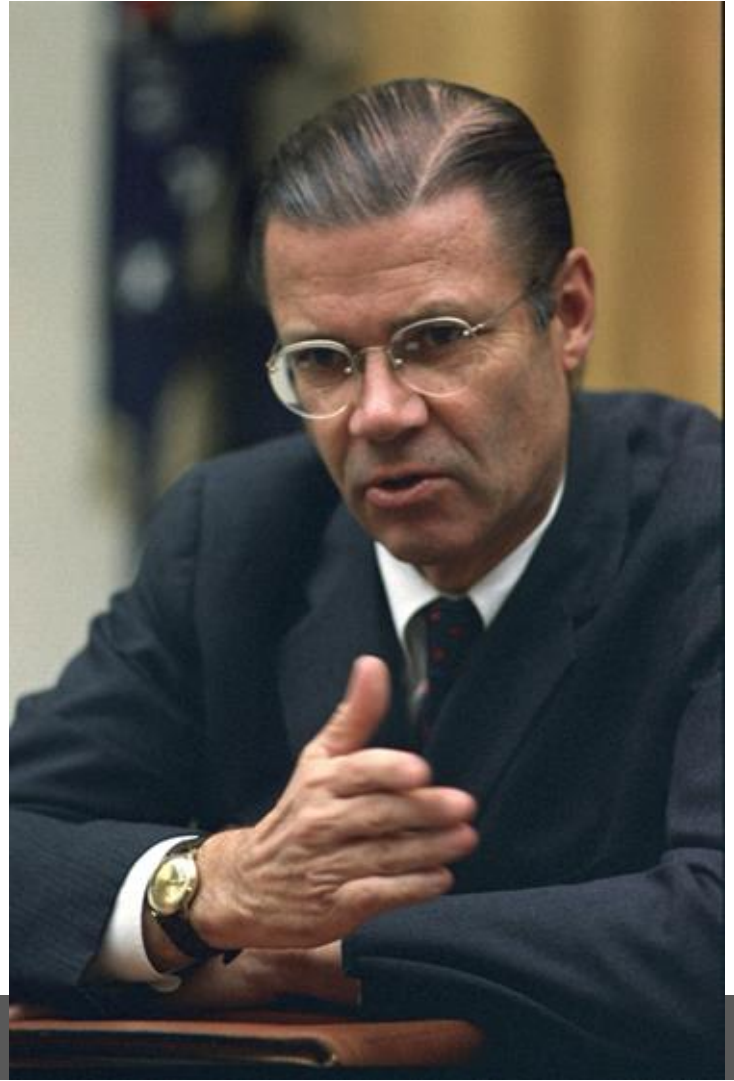


⚠ WARNING ⚠
AVOID SLOT MACHINE PROGRAMMING

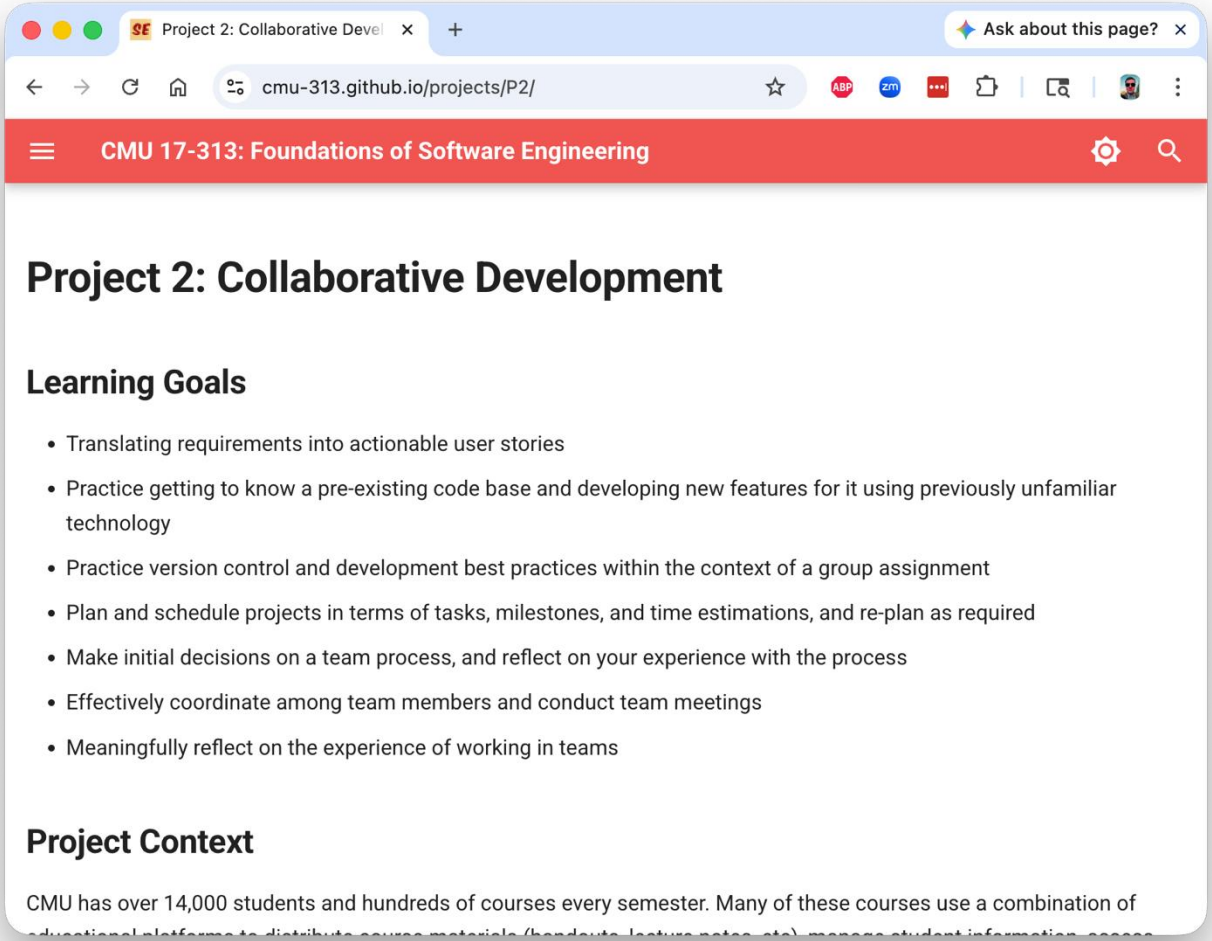
737-MAX Case Study

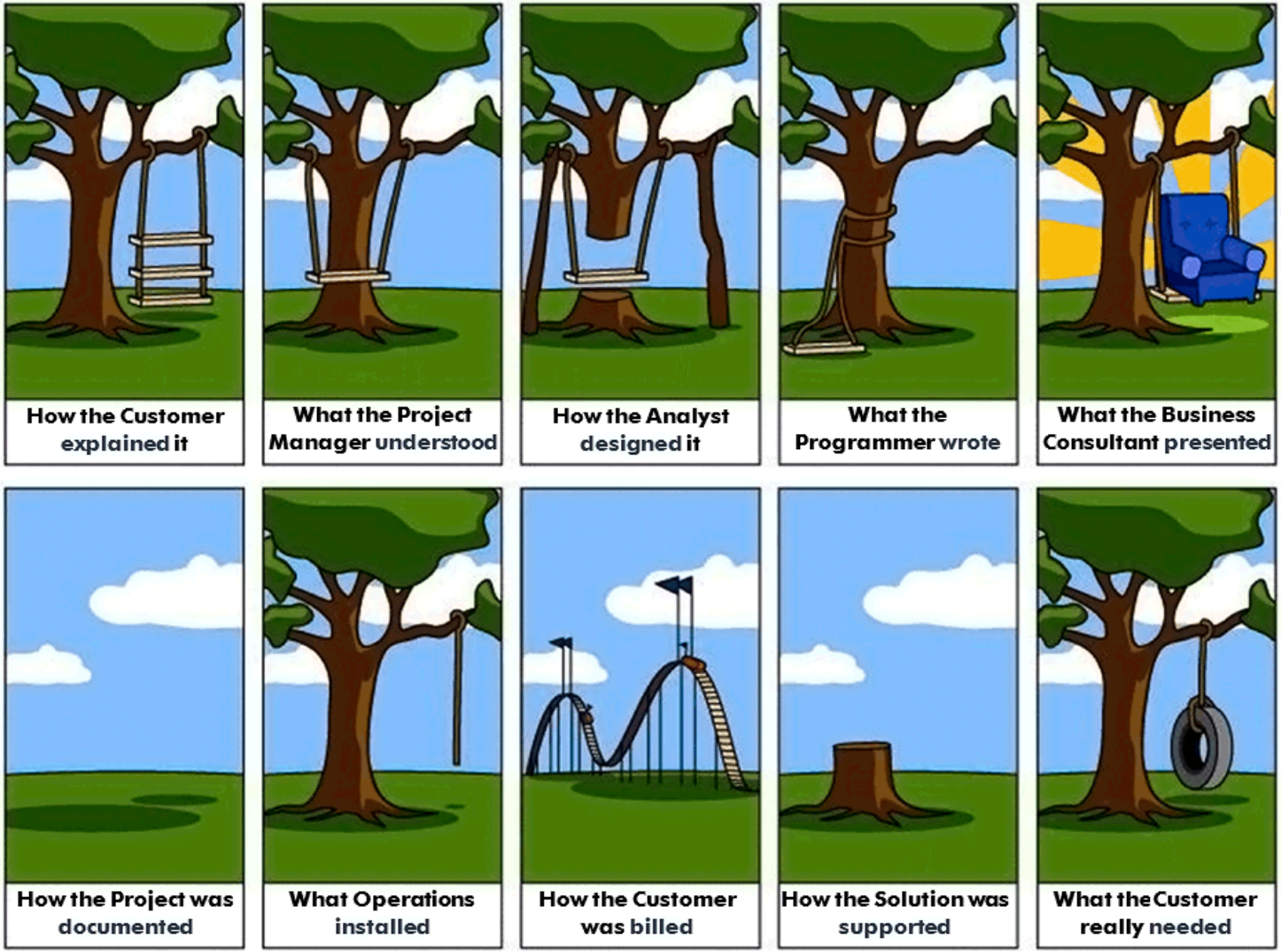
By what Metrics can we judge AV software (e.g., safety)?

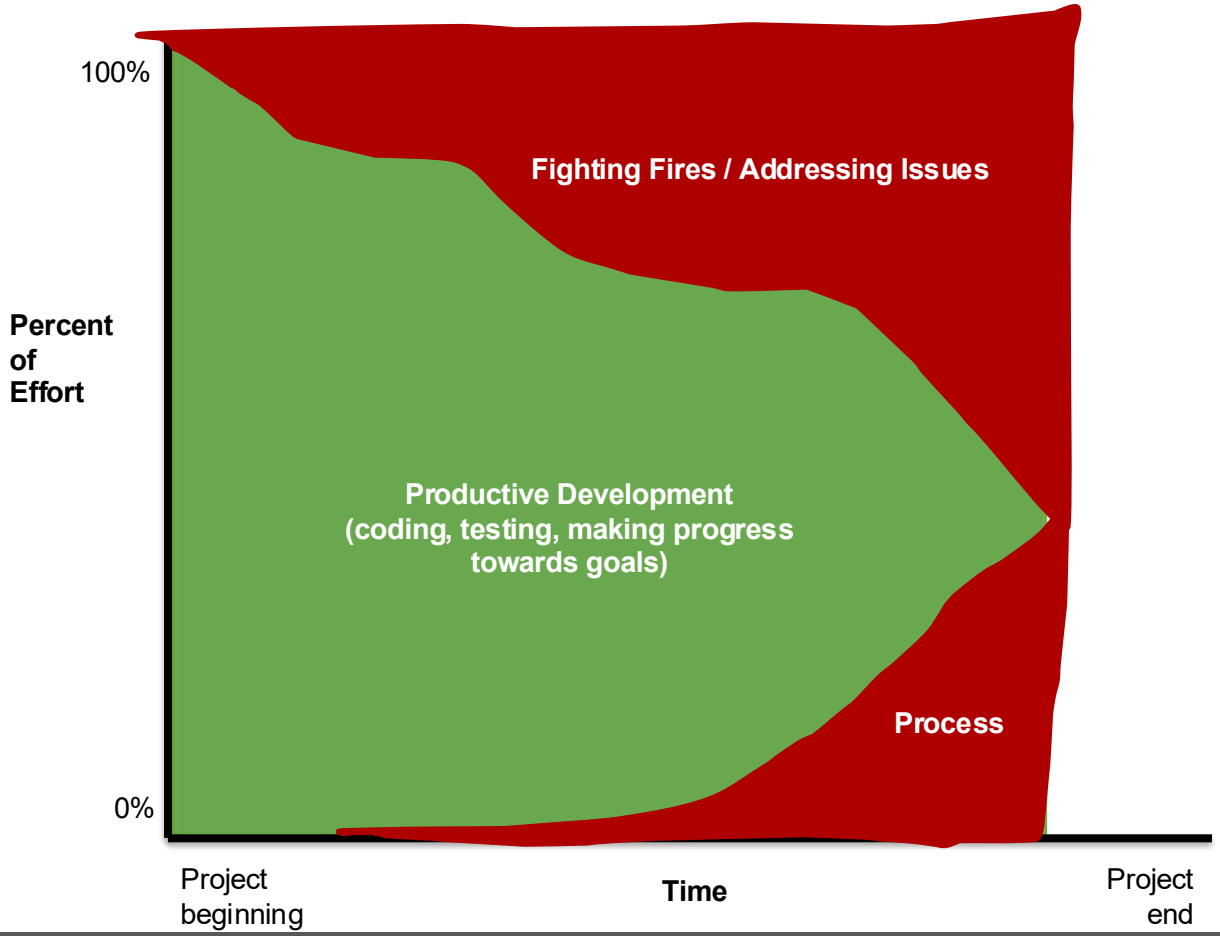

#1 NATIONAL BESTSELLER
IN RETROSPECT
"Unsparing...a clear, concise and extremely interesting look at a crucial period of U.S. decision making. It deserves to be widely read."
—Wall Street Journal
THE TRAGEDY AND LESSONS OF VIETNAM
ROBERT S. McNAMARA
WITH BRIAN VANDEMARK

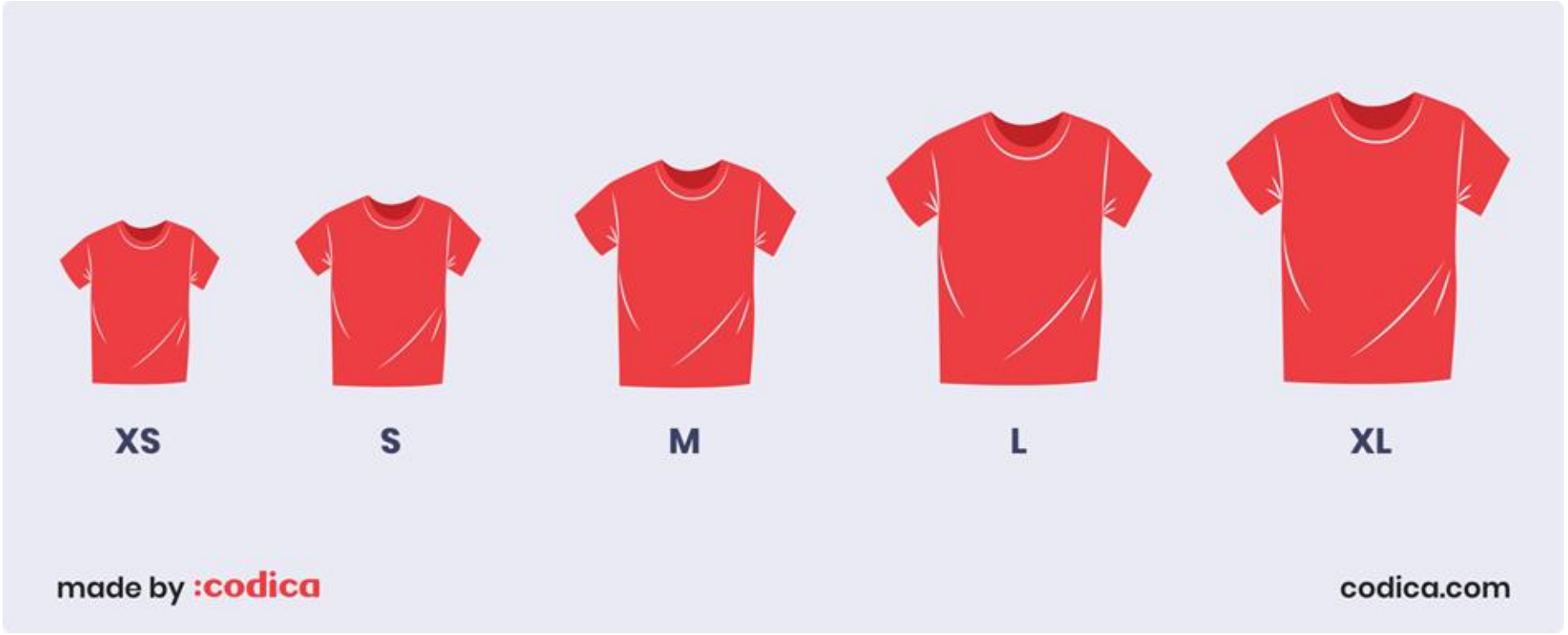



"IF THERE IS ONE MOVIE THAT OUGHT TO BE STUDIED BY MILITARY AND CIVILIAN LEADERS AROUND THE WORLD AT THIS TREACHEROUS HISTORICAL MOMENT, IT IS 'THE FOG OF WAR'. ERROL MORRIS'S PORTRAIT OF FORMER UNITED STATES DEFENSE SECRETARY, ROBERT S. MCNAMARA."
"SMART AND COMPLETELY FASCINATING!"
"STUNNING! SUPERBLY MADE!"
"SPLENDID! A MASTERPIECE!"
THE FOG OF WAR
BEST DOCUMENTARY FEATURE
BEST ORIGINAL SCORE
BEST EDITING


Project 2: Collaborative Deve
Ask about this page?
cmu-313.github.io/projects/P2/
CMU 17-313: Foundations of Software Engineering
Project 2: Collaborative Development
Learning Goals
Translating requirements into actionable user stories
Practice getting to know a pre-existing code base and developing new features for it using previously unfamiliar technology
Practice version control and development best practices within the context of a group assignment
Plan and schedule projects in terms of tasks, milestones, and time estimations, and re-plan as required
Make initial decisions on a team process, and reflect on your experience with the process
Effectively coordinate among team members and conduct team meetings
Meaningfully reflect on the experience of working in teams
Project Context
CMU has over 14,000 students and hundreds of courses every semester. Many of these courses use a combination of

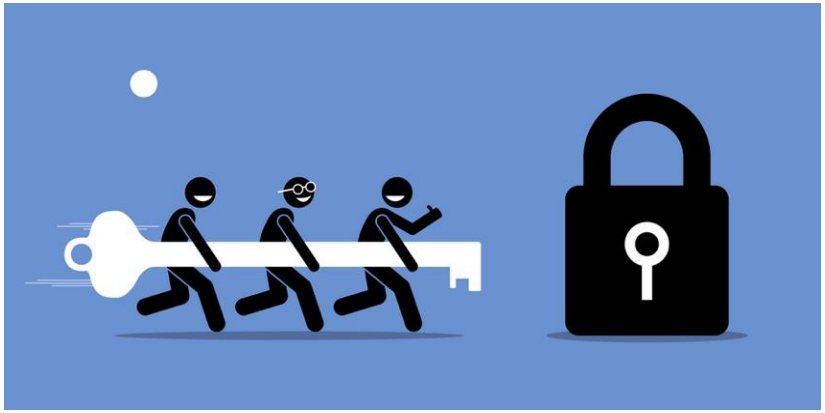

How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed


100%
Fighting Fires / Addressing Issues
Percent of Effort
Productive Development (coding, testing, making progress towards goals)
Process
0%
Project beginning
Time
Project end


XS
S
M
L
XL
made by :codica
codica.com

vectorstock.com/23988884

Working solo vs. as a team (human/AI?)



(11) *General Interference with Organizations an Production*

(a) Organizations and Conferences

(1) Insist on doing everything throug "channels." Never permit short-cuts to be take in order to expedite decisions.

(2) Make "speeches." Talk as frequently possible and at great length. Illustrate yo "points" by long anecdotes and accounts of pe sonal experiences. Never hesitate to make a fe appropriate "patriotic" comments.

(3) When possible, refer all matters committees, for "further study and consider tion." Attempt to make the committees as lar as possible — never less than five.

(4) Bring up irrelevant issues as frequent as possible.

(5) Haggle over precise wordings of cor munications, minutes, resolutions.

(6) Refer back to matters decided upon the last meeting and attempt to re-open t question of the advisability of that decision.

(7) Advocate "caution." Be "reasonabl and urge your fellow-conferees to be "reaso able" and avoid haste which might result embarrassments or difficulties later on.

(8) Be worried about the propriety of a decision — raise the question of whether su action as is contemplated lies within the jur diction of the group or whether it might confl with the policy of some higher echelon.

(b) Managers and Supervisors

(1) Demand written orders.

(2) "Misunderstand" orders. Ask endles questions or engage in long correspondenc about such orders. Quibble over them when yo can.

(3) Do everything possible to delay the delivery of orders. Even though parts of an orde may be ready beforehand, don't deliver it unti it is completely ready.

(4) Don't order new working materials until your current stocks have been virtually ex hausted, so that the slightest delay in filling your order will mean a shutdown.

(5) Order high-quality materials which are hard to get. If you don't get them argue about it. Warn that inferior materials will mean in ferior work.

(6) In making work assignments, always sign out the unimportant jobs first. See that the important jobs are assigned to inefficient workers of poor machines.

(7) Insist on perfect work in relatively unimportant products; send back for refinishing those which have the least flaw. Approve other defective parts whose flaws are not visible to the naked eye.

(8) Make mistakes in routing so that parts and materials will be sent to the wrong place in the plant.

(9) When training new workers, give in complete or misleading instructions.

(10) To lower morale and with it, produc tion, be pleasant to inefficient workers; give them undeserved promotions. Discriminate against efficient workers; complain unjustly about their work.

(11) Hold conferences when there is more critical work to be done.

(12) Multiply paper work in plausible ways. Start duplicate files.

(13) Multiply the procedures and clearances involved in issuing instructions, pay checks, and so on. See that three people have to approve everything where one would do.

(14) Apply all regulations to the last letter.

(c) Office Workers

(1) Make mistakes in quantities of material when you are copying orders. Confuse similar names. Use wrong addresses.

(2) Prolong correspondence with government bureaus.

(3) Misfile essential documents.

(4) In making carbon copies, make one too few, so that an extra copying job will have to be done.

(5) Tell important callers the boss is busy or talking on another telephone.

(6) Hold up mail until the next collection.

(7) Spread disturbing rumors that sound like inside dope.

(d) Employees

(1) *Work slowly.* Think out ways to increase the number of movements necessary on your job: use a light hammer instead of a heavy one, try to make a small wrench do when a big one is necessary, use little force where considerable force is needed, and so on.

(2) Contrive as many interruptions to your work as you can: when changing the material on which you are working, as you would on a lathe or punch, take needless time to do it. If you are cutting, shaping or doing other measured work, measure dimensions twice as often as you need to. When you go to the lavatory, spend a longer time there than is necessary. Forget tools so that you will have to go back after them.

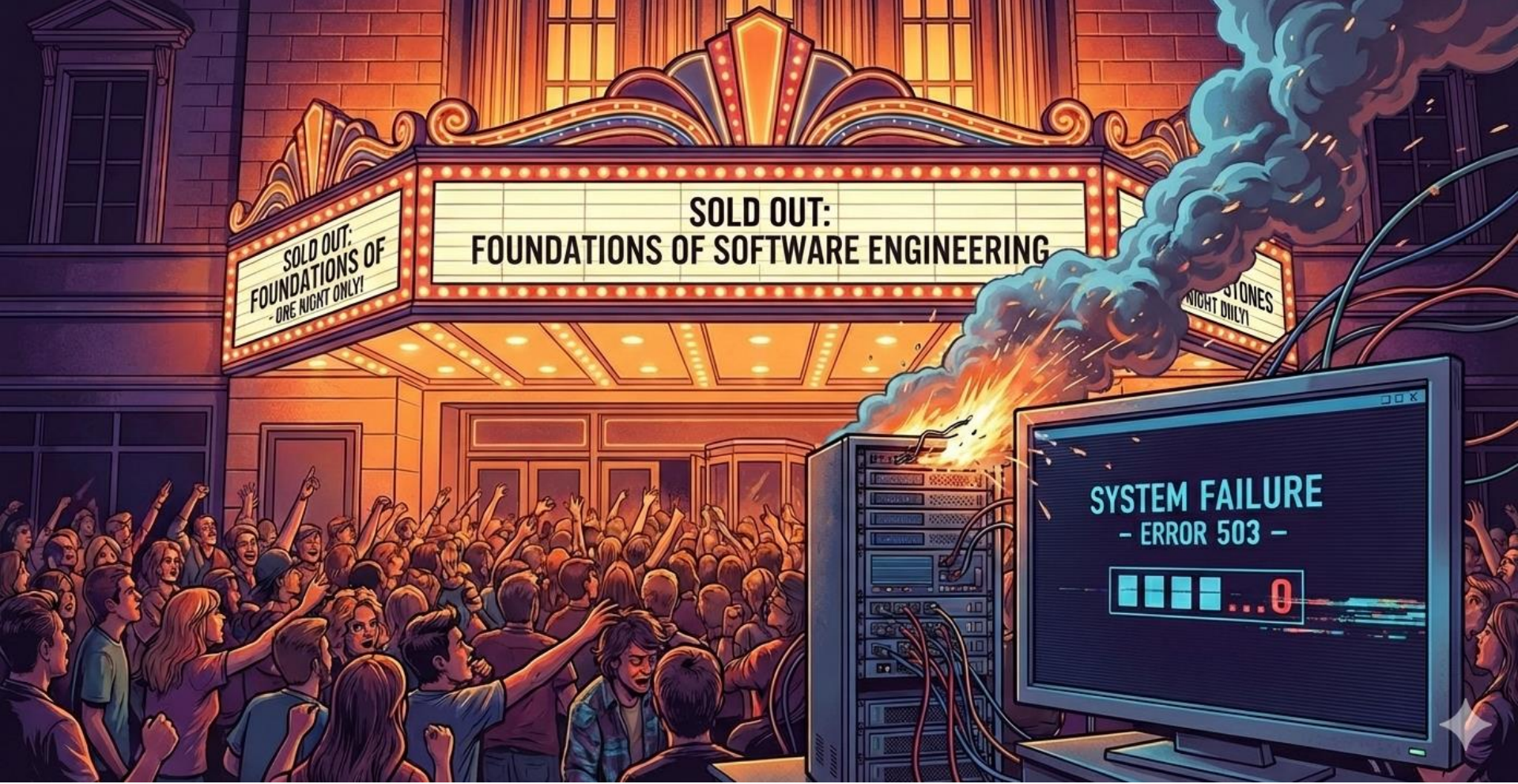
Team survey

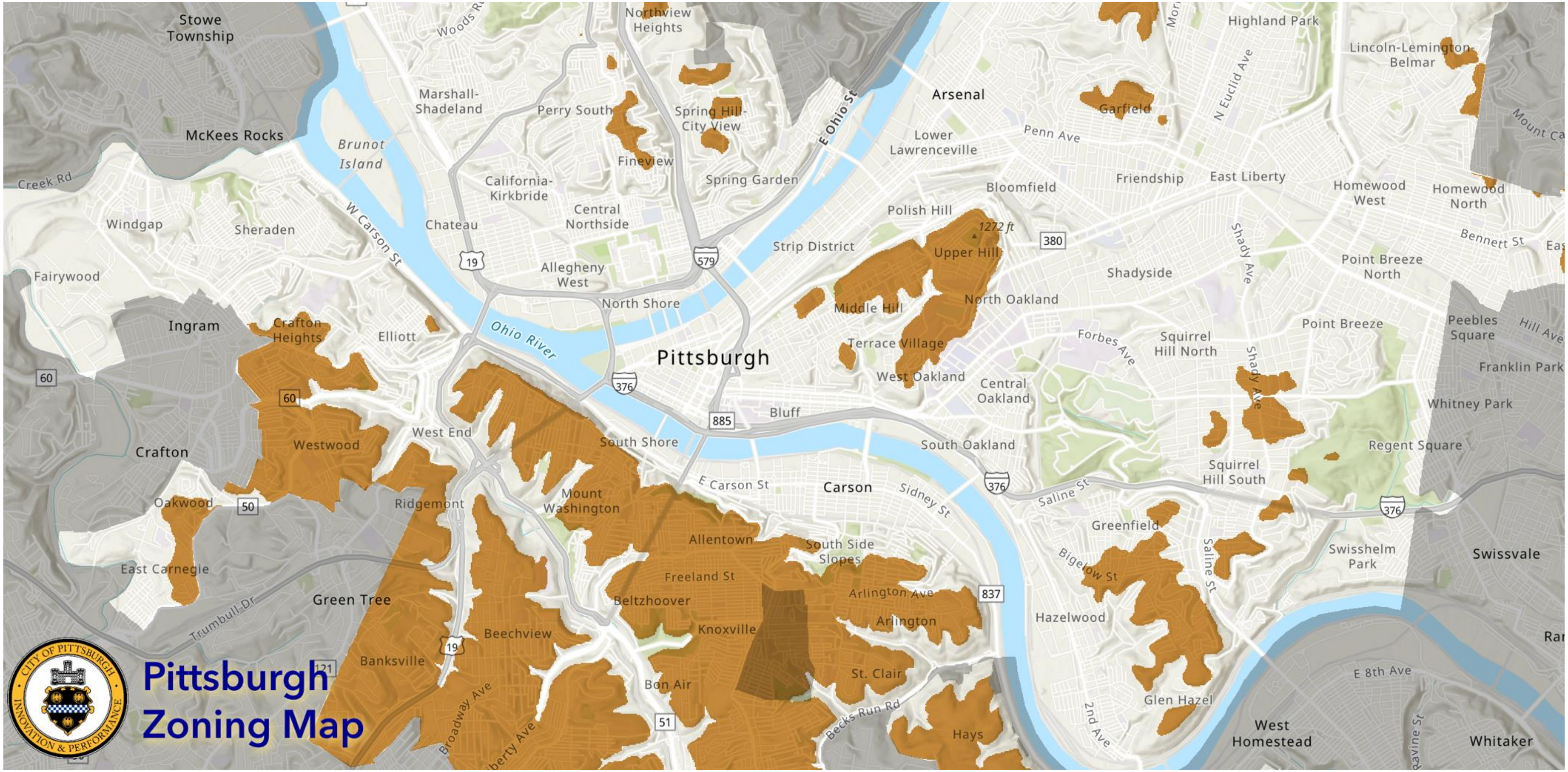
RESEARCH-ARTICLE

Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys

Authors: Kai Presler-Marshall, Sarah Heckman, Kathryn T. Stolee Authors Info & Claims

SIGCSE 2022: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 • February 2022 • Pages 126–132 • https://doi.org/10.1145/3478431.3499367


SOLD OUT:
FOUNDATIONS OF SOFTWARE ENGINEERING
SOLD OUT:
FOUNDATIONS OF
- ONE NIGHT ONLY!
NIGHT ONLY!
SYSTEM FAILURE
- ERROR 503 -
...0

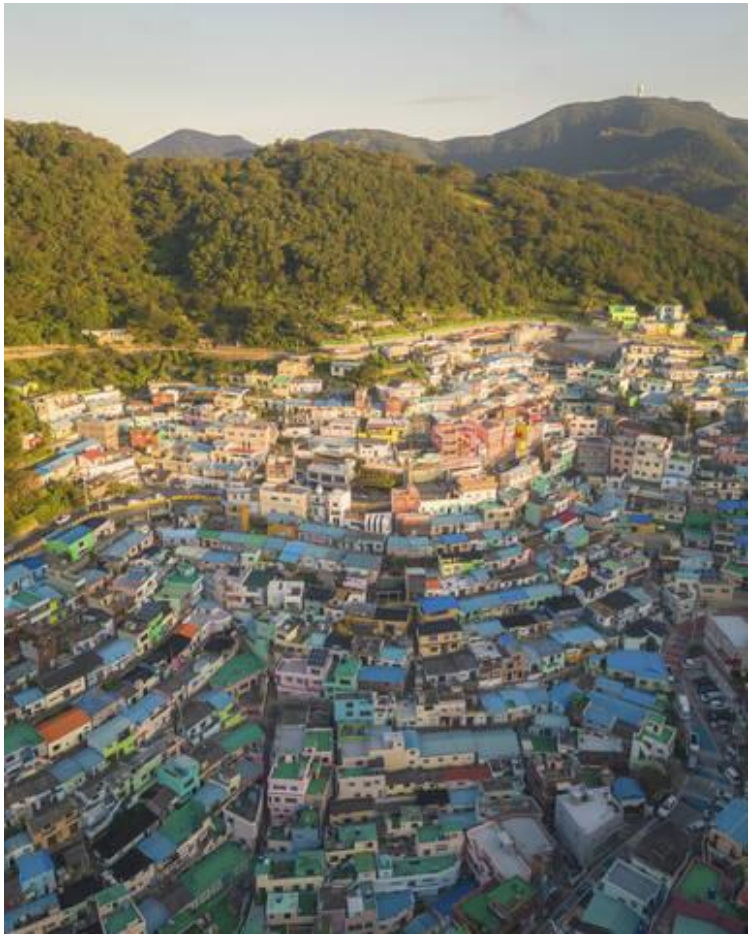


Pittsburgh Zoning Map
CITY OF PITTSBURGH
INNOVATION & PERFORMANCE
Stowe Township
McKees Rocks
Brunot Island
Marshall-Shadeland
Perry South
Northview Heights
Spring Hill-City View
Fineview
Spring Garden
California-Kirkbride
Central Northside
Chateau
Allegheny West
North Shore
Arsenal
Lower Lawrenceville
Garfield
Highland Park
Lincoln-Lemington-Belmar
Bloomfield
Friendship
East Liberty
Homewood West
Homewood North
Polish Hill
Strip District
Upper Hill
Middle Hill
North Oakland
Shadyside
Point Breeze North
Point Breeze
Peebles Square
Franklin Park
Whitney Park
Regent Square
Terrace Village
West Oakland
Central Oakland
Squirrel Hill North
Squirrel Hill South
Pittsburgh
Bluff
South Oakland
Ohio River
Windgap
Sheraden
Fairywood
Ingram
Crafton Heights
Elliott
West End
Westwood
Crafton
Oakwood
East Carnegie
Green Tree
Ridgemont
Mount Washington
South Shore
Carson
South Side Slopes
Allentown
Beltzhoover
Knoxville
Arlington
St. Clair
Beechview
Banksville
Bon Air
Hays
Hazelwood
Greenfield
Glen Hazel
Swisshelm Park
Swissvale
West Homestead
Whitaker
Creek Rd
W Carson St
E Carson St
Sidney St
Penn Ave
E Ohio St
Woods Run
Forbes Ave
Shady Ave
N Euclid Ave
Bennett St
Hill Ave
Saline St
Bigelow St
Freeland St
Arlington Ave
Becks Run Rd
Broadway Ave
Trumbull Dr
2nd Ave
E 8th Ave
Mount Carmel
1272 ft
19
60
50
51
121
279
376
380
579
837
885

https://www.archdaily.com/

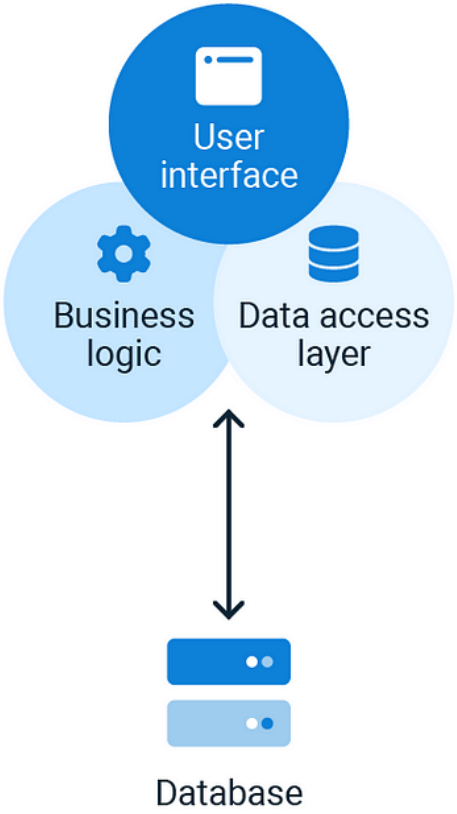
https://www.instagram.com/architectanddesign

https://www.mykonosceramica.com/

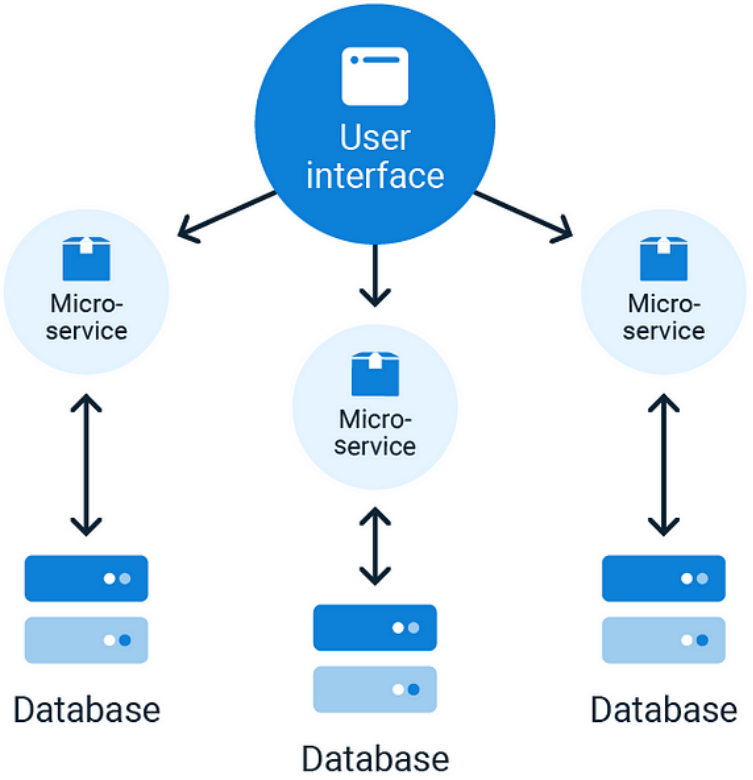


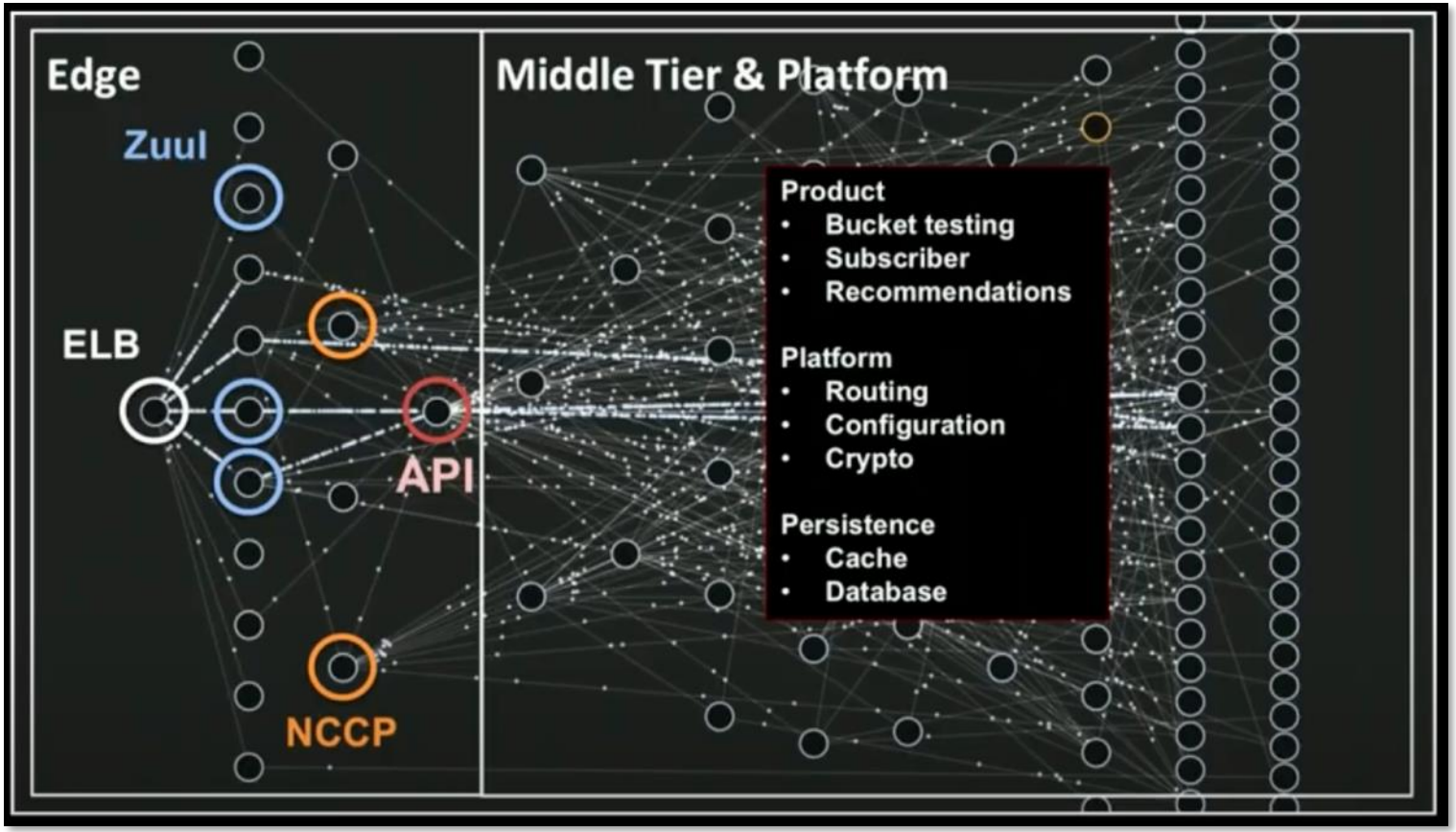
www.qr-view.com

Monolithic Architecture



Microservice Architecture



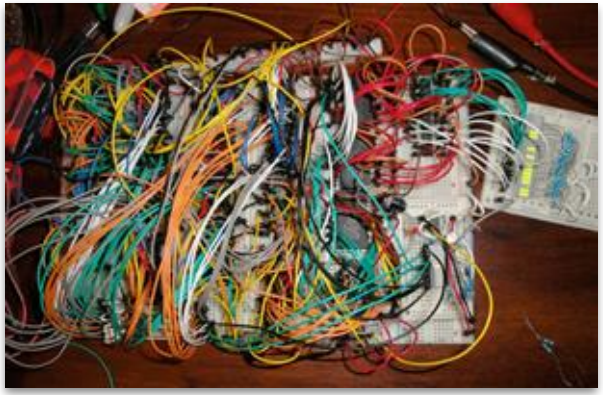


https://www.youtube.com/watch?v=V_oxbj-a1wQ

Risk


Tony Webster
@webster
Follow
I appreciate the honesty.
Pick a password
Don't reuse your bank password, we didn't spend a lot on security for this app.
At least 6 characters
your password
Continue
8:20 PM - 15 Sep 2018
5,868 Retweets 15,672 Likes
58
5.9K
16K

Internal Quality



- Is the code well structured?
- Is the code understandable?
- How well documented?

External Quality

- Does the software crash?
- Does it meet the requirements?
- Is the UI well designed?

Program Under Test: Wordle

- Guessing game
- User guesses a 5-letter English word
- After each guess, the tiles change color to show how close guesser is to secret word

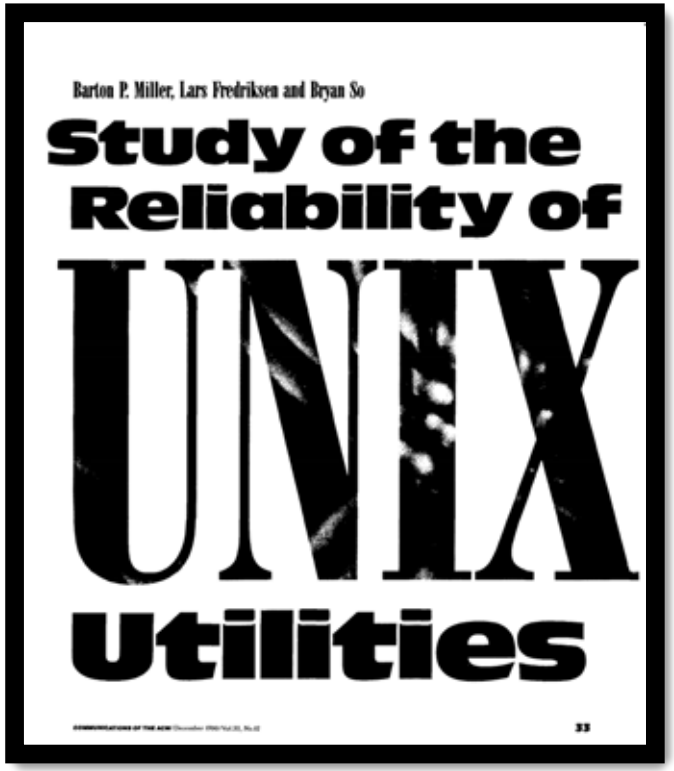
Static Analysis vs. Dynamic Analysis

- Requires only source code
- Conservatively reasons about all possible inputs and program paths
- Reported warnings may contain false positives
- Can report all warnings of a particular class of problems
- Advanced techniques like formal verification can prove certain complex properties, but rarely run in CI due to cost

- Requires successful build + test inputs
- Observes individual executions
- Reported problems are real, as observed by a witness input
- Can only report problems that are seen. Highly dependent on test inputs. Subject to false negatives
- Advanced techniques like symbolic execution can prove certain complex properties, but rarely run in CI due to cost

There are lots of static analysis tools!

Communications of the ACM (1990)

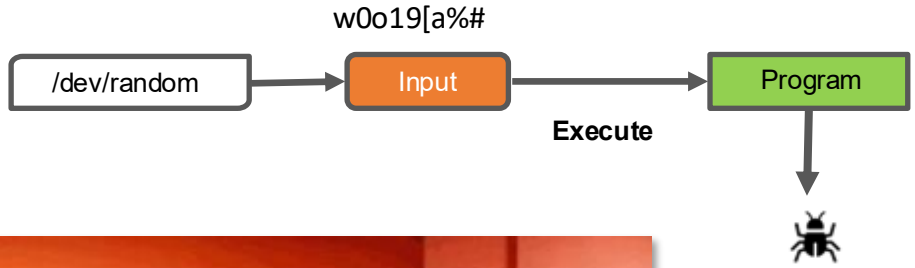

Barton P. Miller, Lars Fredriksen and Bryan So

Study of the Reliability of UNIX Utilities

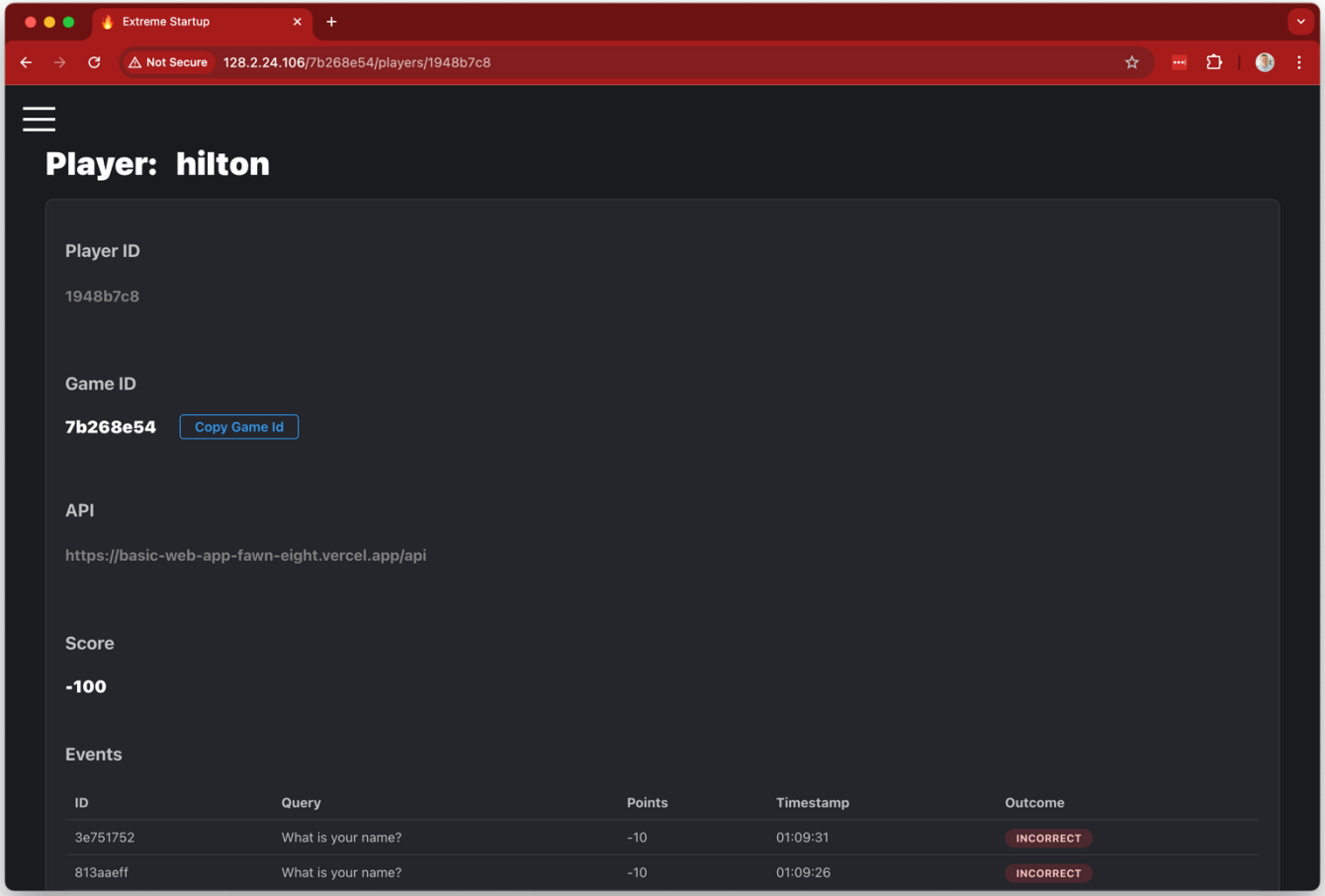
> On a dark and stormy night one of the authors was logged on to his workstation on a dial-up line from home and the rain had affected the phone lines; there were frequent spurious characters on the line. The author had to race to see if he could type a sensible sequence of characters before the noise scrambled the command. This line noise was not surprising; but we were surprised that these spurious characters were causing programs to crash.

How can we identify these bugs?

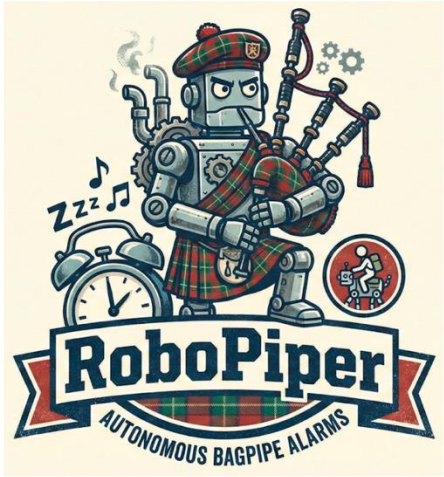
Fuzz Testing randomly generates inputs and checks for program crashes



A 1990 study found crashes in:
adb, as, bc, cb, col, diction, emacs, eqn, ftp, indent, lex, look, m4, make, nroff, plot, prolog, ptx, refer!, spell, style, tsort, uniq, vgrind, vi

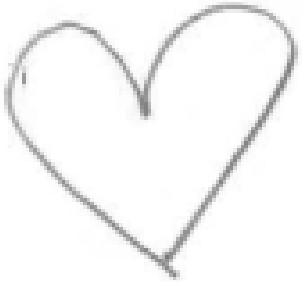

Extreme Startup
Not Secure
128.2.24.106/7b268e54/players/1948b7c8
Player: hilton
Player ID
1948b7c8
Game ID
7b268e54
Copy Game Id
API
https://basic-web-app-fawn-eight.vercel.app/api
Score
-100
Events
ID
Query
Points
Timestamp
Outcome
3e751752
What is your name?
-10
01:09:31
INCORRECT
813aaeff
What is your name?
-10
01:09:26
INCORRECT

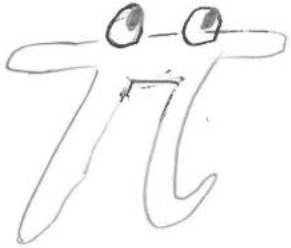
Scenario



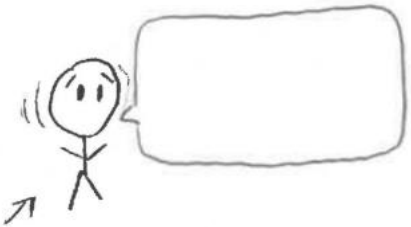
You have just joined the engineering team at RoboPiper, a disruptive Pittsburgh startup dedicated to solving the chronic oversleeping epidemic among Carnegie Mellon students. As a junior software engineer, you are responsible for the "Tactical Wake-Up" system, a complex software-hardware stack designed to ensure no 8:00 AM lecture is ever missed again. The system monitors student alarm clocks in real-time; when it detects a student has hit the snooze button for the third consecutive time, it triggers the immediate dispatch of an autonomous, kilt-clad robotic piper to the student's GPS coordinates.

Upon arrival, the RoboPiper initiates its primary enforcement protocol: a 120-decibel, high-fidelity rendition of "Fight for the Glory of Carnegie" that will not cease until the student physically exits their building to perform a manual shut-off. For those with a larger budget and shorter timeline, the "Highland Express" premium tier allows the student to mount the robot and ride the bagpiping automaton directly to their classroom.



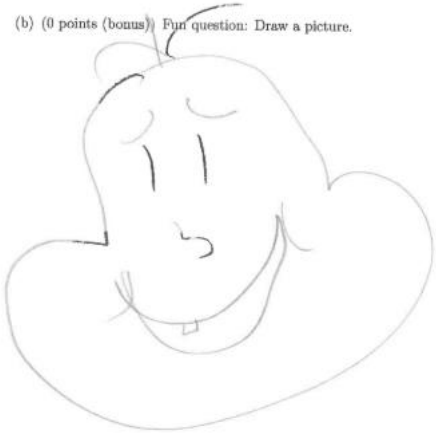


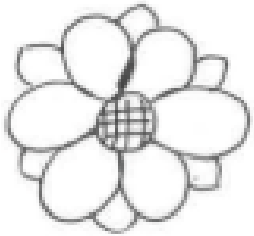
a picture.



has no mouth and
thus cannot scream

(b) (0 points (bonus)) Fun question: Draw a picture.

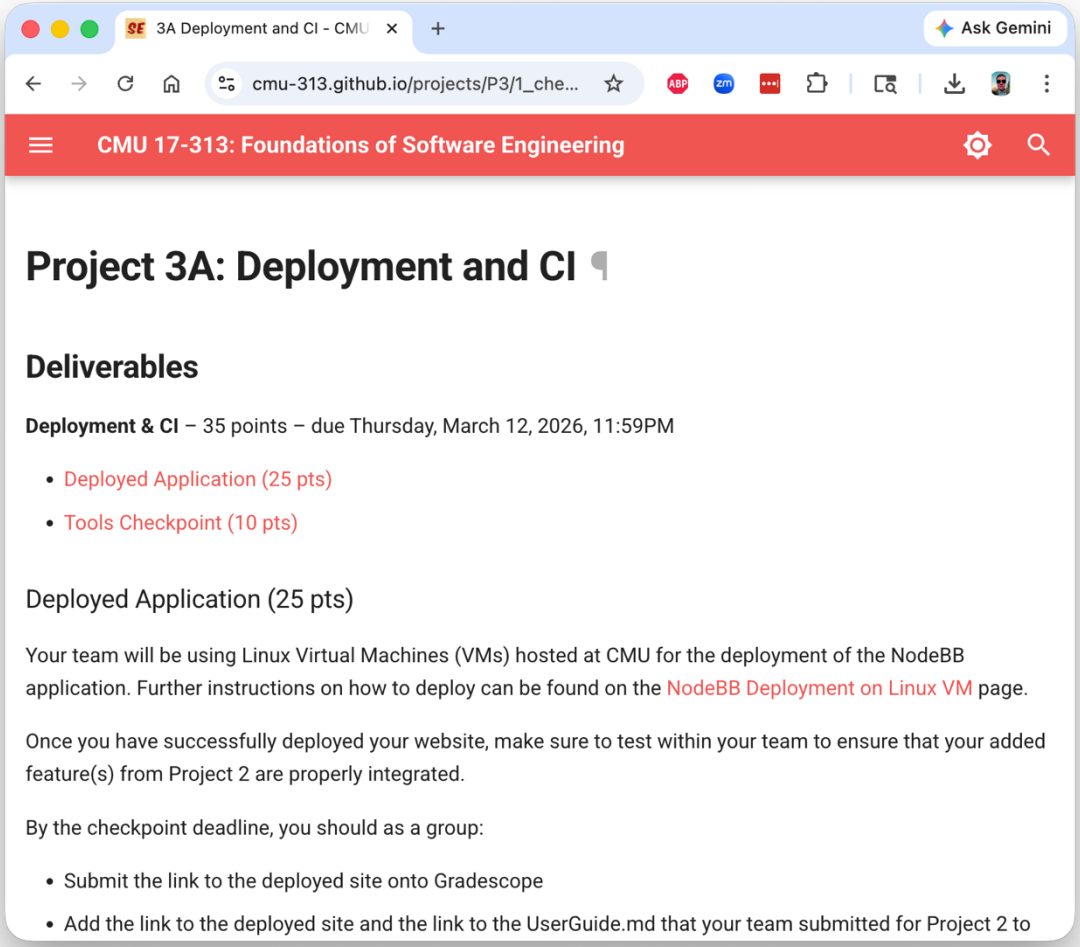




Time to write our own design docs! (No Gen AI for this task)

- Divide up into 4 sections –NOTE: you should be signed in w/Andrew to google
- Your mission:
  - Brainstorm a feature to add to a scooter app and write a design spec, together, in real time!
  - Review the design doc, collaborate around text
  - Review another team's design doc, ask questions/leave comments

bit.ly/4s17IdW | bit.ly/3PcVBMp | bit.ly/47zbQJU | bit.ly/40oWaFl


3A Deployment and CI - CMU
Ask Gemini
cmu-313.github.io/projects/P3/1_che...
CMU 17-313: Foundations of Software Engineering
Project 3A: Deployment and CI ¶
Deliverables
Deployment & CI – 35 points – due Thursday, March 12, 2026, 11:59PM
Deployed Application (25 pts)
Tools Checkpoint (10 pts)
Deployed Application (25 pts)
Your team will be using Linux Virtual Machines (VMs) hosted at CMU for the deployment of the NodeBB application. Further instructions on how to deploy can be found on the NodeBB Deployment on Linux VM page.
Once you have successfully deployed your website, make sure to test within your team to ensure that your added feature(s) from Project 2 are properly integrated.
By the checkpoint deadline, you should as a group:
Submit the link to the deployed site onto Gradescope
Add the link to the deployed site and the link to the UserGuide.md that your team submitted for Project 2 to

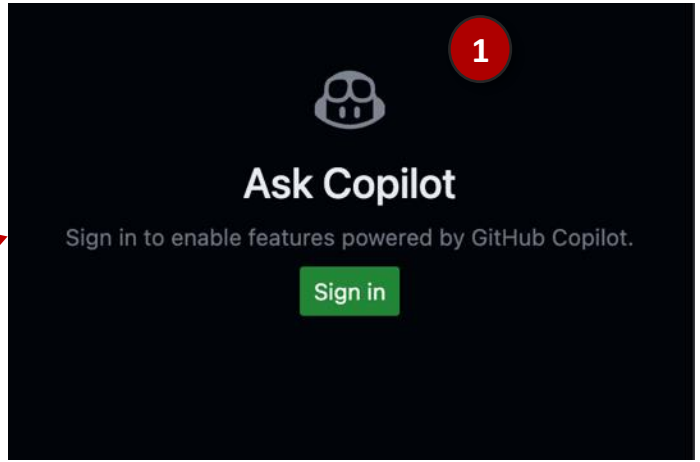
Amazon.com + Add to myFT

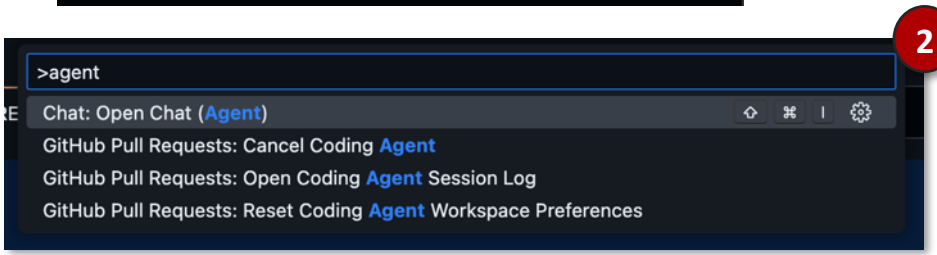
Amazon holds engineering meeting following AI-related outages

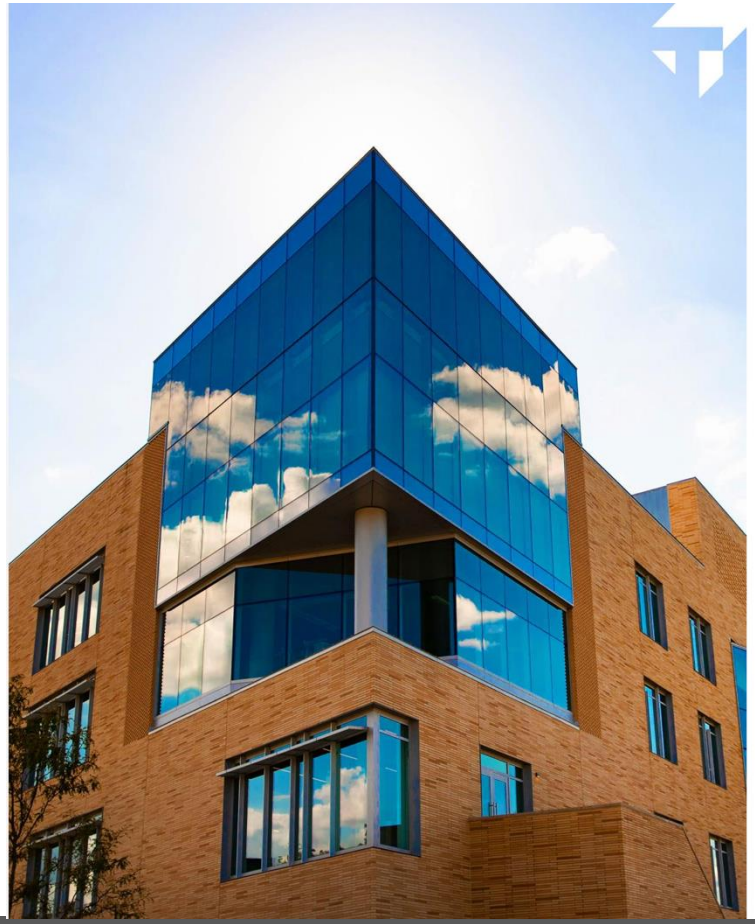
Ecommerce giant says there has been a ‘trend of incidents’ linked to ‘Gen-AI assisted changes’

- The online retail giant said there had been a “trend of incidents” in recent months, characterised by a “high blast radius” and “Gen-AI assisted changes” among other factors, according to a briefing note for the meeting seen by the FT.

Let’s use Copilot Agent







Difficult Conversations & Feedback

MARCH 2026

Taya R. Cohen, PhD

Professor of Organizational Behavior and Business Ethics

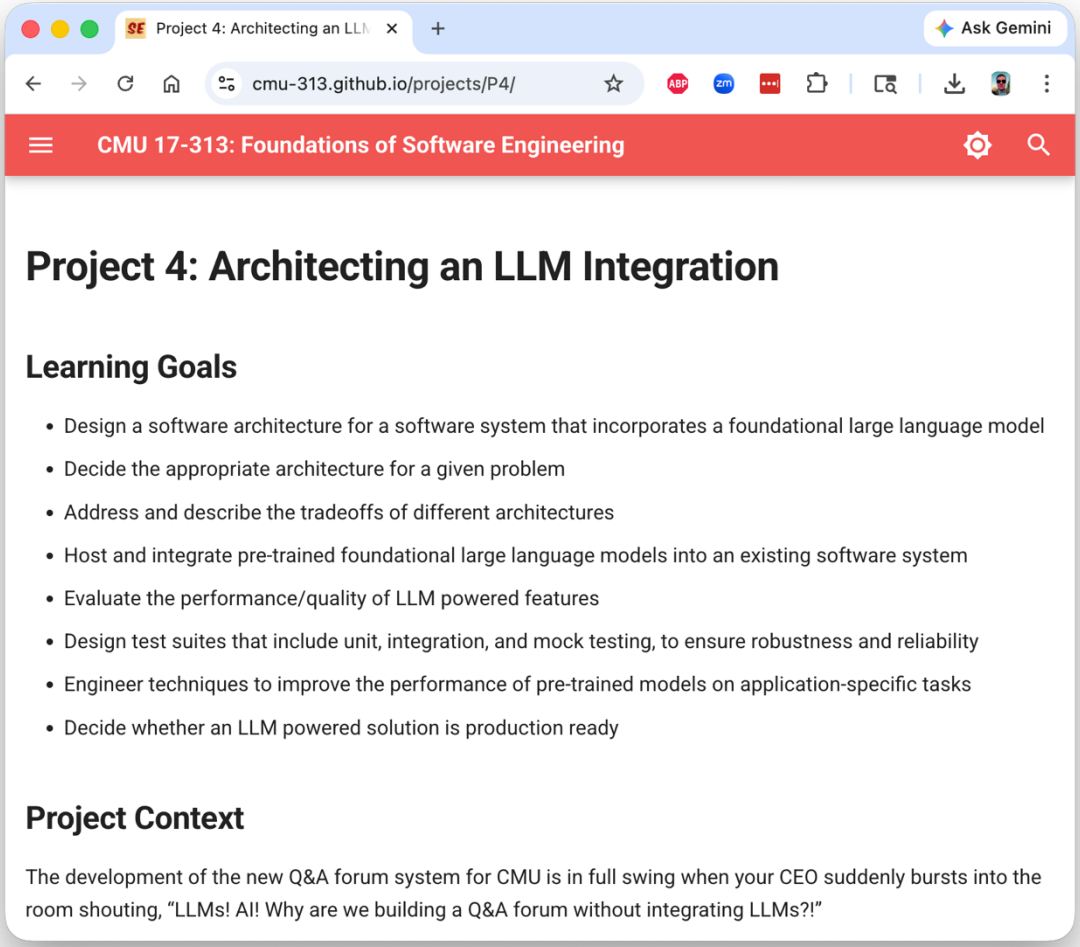
Carnegie Mellon University
Tepper School of Business

Carnegie Mellon University
Tepper School of Business

Property-based testing to the Rescue?

Conditional Properties

- **If** condition A is met, **then** property B must hold.
  Mathematical logic: $A \Rightarrow B$
- Examples:
  - **Wordle**: If the guess word matches the secret word the output should be (G,G,G,G,G).
  - **Compilers**: If the program includes an invalid keyword, the parser should throw a parse error.
  - **NodeBB** converts post titles into URL-safe "slugs" (e.g., "Hello World!" becomes hello-world). If the input string contains at least one alphanumeric character then the generated slug must never be empty and must only contain a-z, 0-9, and -.

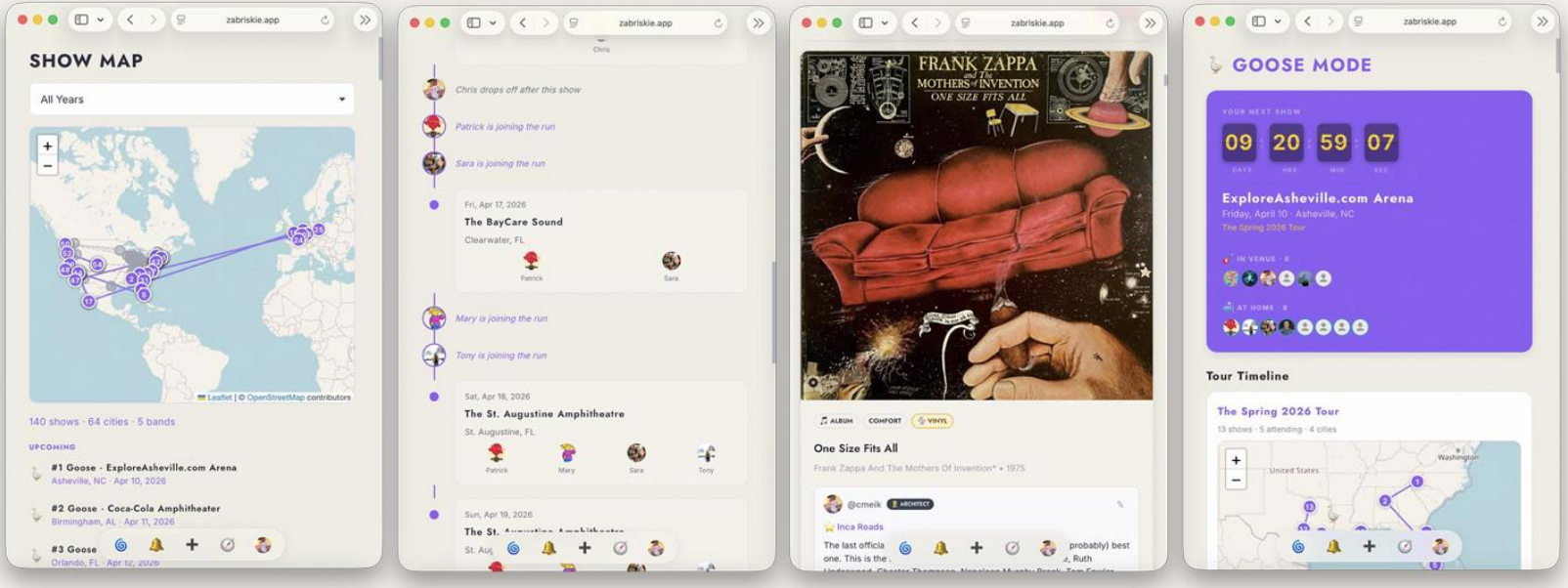

Project 4: Architecting an LLM
Ask Gemini
cmu-313.github.io/projects/P4/
CMU 17-313: Foundations of Software Engineering
Project 4: Architecting an LLM Integration
Learning Goals
Design a software architecture for a software system that incorporates a foundational large language model
Decide the appropriate architecture for a given problem
Address and describe the tradeoffs of different architectures
Host and integrate pre-trained foundational large language models into an existing software system
Evaluate the performance/quality of LLM powered features
Design test suites that include unit, integration, and mock testing, to ensure robustness and reliability
Engineer techniques to improve the performance of pre-trained models on application-specific tasks
Decide whether an LLM powered solution is production ready
Project Context
The development of the new Q&A forum system for CMU is in full swing when your CEO suddenly bursts into the room shouting, “LLMs! AI! Why are we building a Q&A forum without integrating LLMs?!”

Goals

Identify the core challenges with modifying, testing, and deploying applications **safely.**

Describe and **differentiate** the possible techniques for ensuring **reliable** and **safe delivery of software at scale.**

Practice identifying problematic changes and how to go about **making changes safely.**


zabriskie.app
SHOW MAP
All Years
Leaflet | © OpenStreetMap contributors
140 shows · 64 cities · 5 bands
UPCOMING
#1 Goose · ExploreAsheville.com Arena
Asheville, NC · Apr 10, 2026
#2 Goose · Coca-Cola Amphitheater
Birmingham, AL · Apr 11, 2026
#3 Goose
Orlando, FL · Apr 12, 2026
zabriskie.app
Chris drops off after this show
Patrick is joining the run
Sara is joining the run
Fri, Apr 17, 2026
The BayCare Sound
Clearwater, FL
Patrick
Sara
Mary is joining the run
Tony is joining the run
Sat, Apr 18, 2026
The St. Augustine Amphitheatre
St. Augustine, FL
Patrick
Mary
Sara
Tony
Sun, Apr 19, 2026
zabriskie.app
FRANK ZAPPA and the MOTHERS of INVENTION
ONE SIZE FITS ALL
ALBUM
COMFORT
VINYL
One Size Fits All
Frank Zappa And The Mothers Of Invention* • 1975
@cmeik
ARCHITECT
Inca Roads
The last officia
one. This is the .
probably) best
, Ruth
zabriskie.app
GOOSE MODE
YOUR NEXT SHOW
09
20
59
07
DAYS
HRS
MIN
SEC
ExploreAsheville.com Arena
Friday, April 10 · Asheville, NC
The Spring 2026 Tour
IN VENUE · 6
AT HOME · 8
Tour Timeline
The Spring 2026 Tour
13 shows · 5 attending · 4 cities
United States
Washington

Free software as a Philosophy

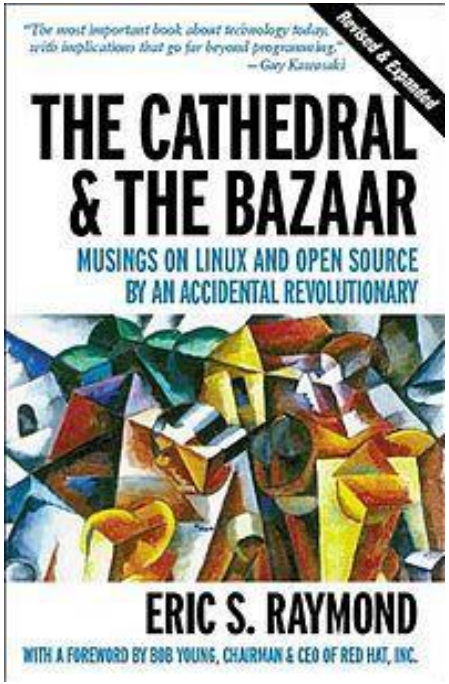
- “Free as in Speech, not as in beer”

Richard Stallman’s Free Software Foundation — free as in liberties

- Freedom 0: run code as you wish, for any purpose
- Freedom 1: study how code works, and change it as you wish
- Freedom 2: redistributed copies (of original) so you can help others
- Freedom 3: distribute copies of your modified version to others

Richard M Stallman (Licensed under GFDL)

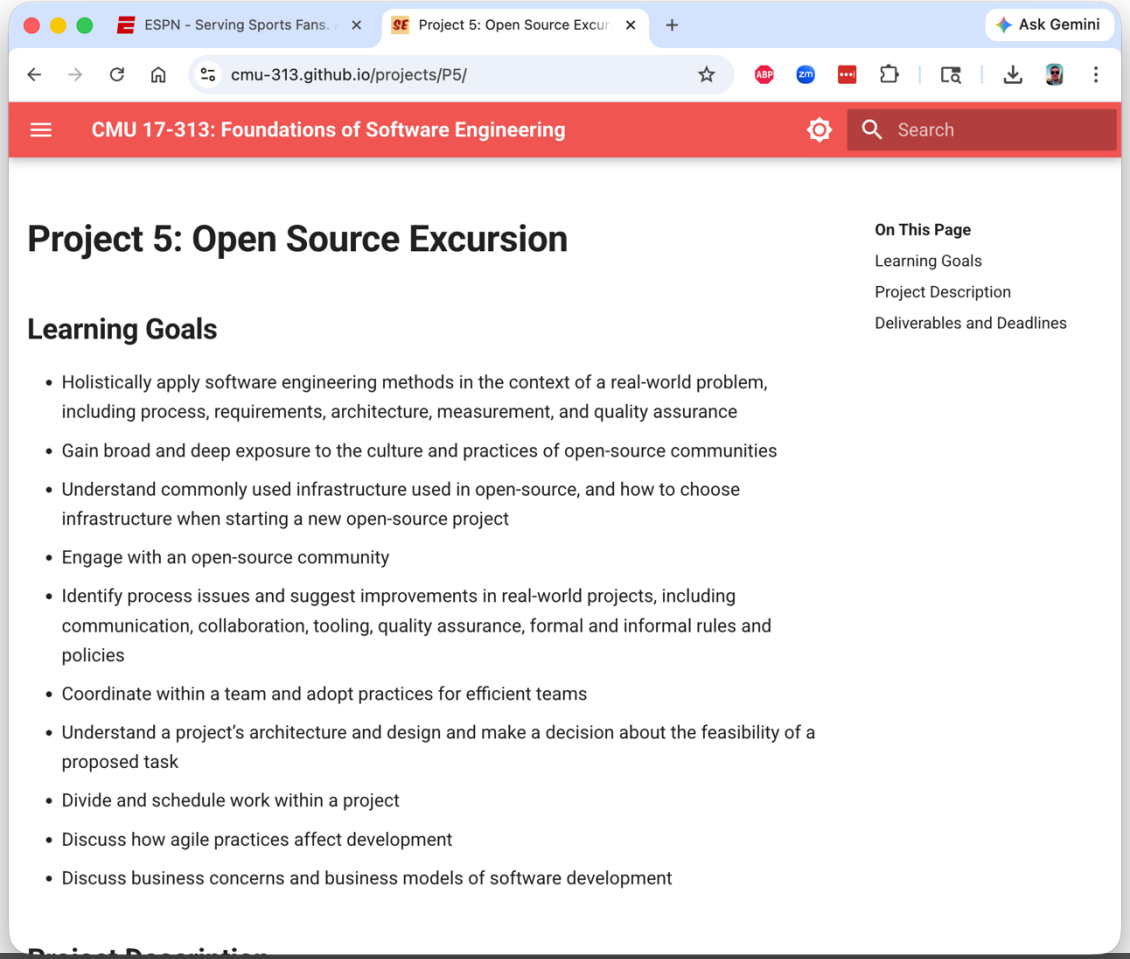
The Cathedral and the Bazaar



Which license to choose?

Three questions to promote human flourishing

1. Does my software respect the humanity of the users?
2. Does my software amplify positive behavior, or negative behavior for users and society at large?
3. Will my software's quality impact the humanity of others?


ESPN - Serving Sports Fans.
Project 5: Open Source Excur
Ask Gemini
cmu-313.github.io/projects/P5/
CMU 17-313: Foundations of Software Engineering
Search
Project 5: Open Source Excursion
On This Page
Learning Goals
Project Description
Deliverables and Deadlines
Learning Goals
Holistically apply software engineering methods in the context of a real-world problem, including process, requirements, architecture, measurement, and quality assurance
Gain broad and deep exposure to the culture and practices of open-source communities
Understand commonly used infrastructure used in open-source, and how to choose infrastructure when starting a new open-source project
Engage with an open-source community
Identify process issues and suggest improvements in real-world projects, including communication, collaboration, tooling, quality assurance, formal and informal rules and policies
Coordinate within a team and adopt practices for efficient teams
Understand a project's architecture and design and make a decision about the feasibility of a proposed task
Divide and schedule work within a project
Discuss how agile practices affect development
Discuss business concerns and business models of software development

Technical debt definition #2

Any software system has a certain amount of essential complexity required to do its job...

... but most systems contain cruft that makes it harder to understand.

*Cruft causes changes to take **more effort***

The technical debt metaphor treats the cruft as a debt, whose interest payments are the extra effort these changes require.

https://martinfowler.com/bliki/TechnicalDebt.html

10X Engineers

- Aka “rock-star”, “ninja”

Holding periods in Kerri's example

If either of these holding periods aren't met, your options are treated as NSOs (=more tax)

AMA

Ask the professors any question you want


That's all Folks!