

Final Lecture

17-313 Fall 2024

Foundations of Software Engineering

<https://cmu-313.github.io>

Michael Hilton and Rohan Padhye

Administrivia

- Project Final Report & Presentations – Sunday, December 8th, 11:59pm
- Final Presentations - Monday, December 9th, 1:00-4:00pm
 - 4 minutes long, with 2 minutes Q&A.
 - Participation from all team members during the presentation is **required**.
- Individual Reflections – Monday, December 9th, 11:59pm

AMA



Semester Review



S3D Software and Societal
Systems Department

Carnegie
Mellon
University

Introductions

Michael Hilton

Associate 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 Teaching Professor at CMU - Fall 2017 to current



Rohan Padhye



B.E. University of Mumbai – 2011



M.Tech. IIT Bombay – 2013

IBM Research Software Engineer @ IBM Research 2013—2015



Security Engineering Intern @ Samsung – Summer 2017



Research Intern @ Microsoft – Summer 2018



PhD, UC Berkeley – 2020



Assistant Professor, CMU – Since Fall 2020



Visiting Academic, Amazon Web Services 2022—2023



PASTALAB.org

Program Analysis, Software Testing,
and Applications Laboratory

Teaching Assistants



Jesse Ding



Courtney Miller



Yerim Song



Juan Ageitos



Vy Tran



Kareem Segizekov



Smoking Section

- Last full row



Vasa

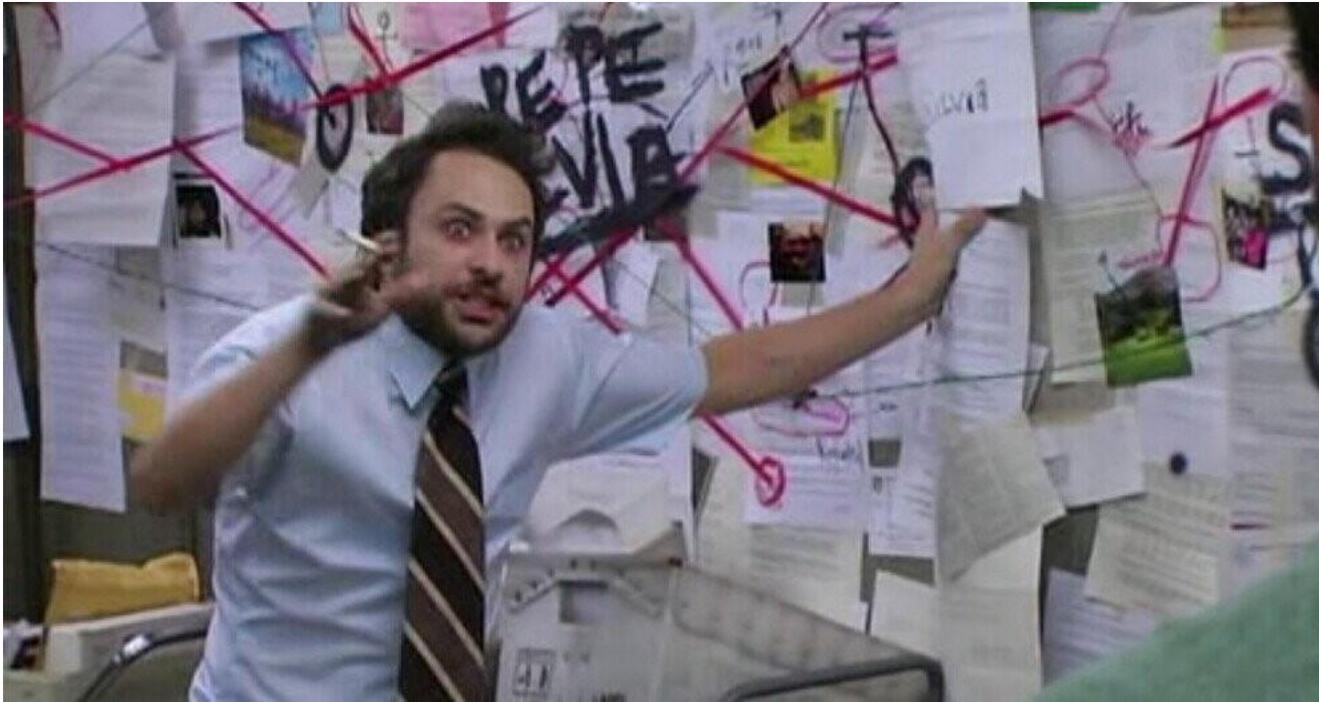


Project 1: Hello, NodeBB!

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

Challenge: How do I tackle this codebase?





1 NATIONAL BESTSELLER

IN RETROSPECT



THE TRAGEDY AND LESSONS OF VIETNAM

ROBERT S.
MCNAMARA

WITH BRIAN VANDEMARK

"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



(★★★★) (★★★★) (★★★★) (★★★★)

FROM THE DIRECTOR OF "THE TIDE BLUE LINE" AND "GIRLS ON HROWN"

"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."

—Michael Lewis, THE NEW YORK TIMES

"SMART AND COMPLETELY FASCINATING!"

"STUNNING! SUPERBLY MADE!"

"SPLendid! A Masterpiece!"

Errol Morris, *The Fog of War*

THE FOG OF WAR

BEST DOCUMENTARY FEATURE
Directed by Errol Morris

BEST ORIGINAL SCORE
Philip Glass

BEST EDITING
Karen Schmeer
Doug Aibel
David King

2003 Official Selection AFI Film Festival - Fri, Nov 14, 2003 7:30pm

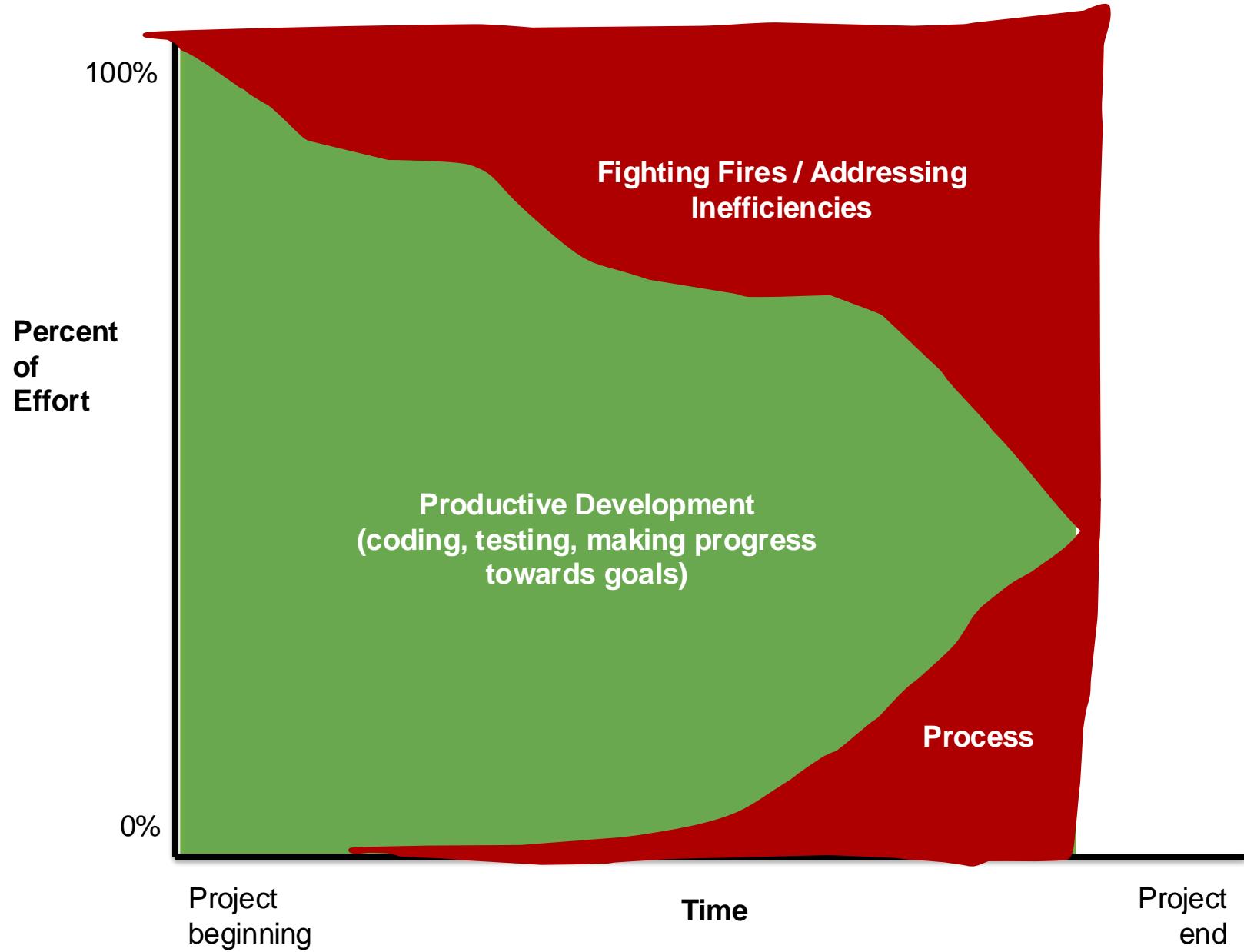


Project 2A: Collaborative Development

Deliverables

Team Process & Planning – 70 points – due Thursday, September 12th, 11:59pm

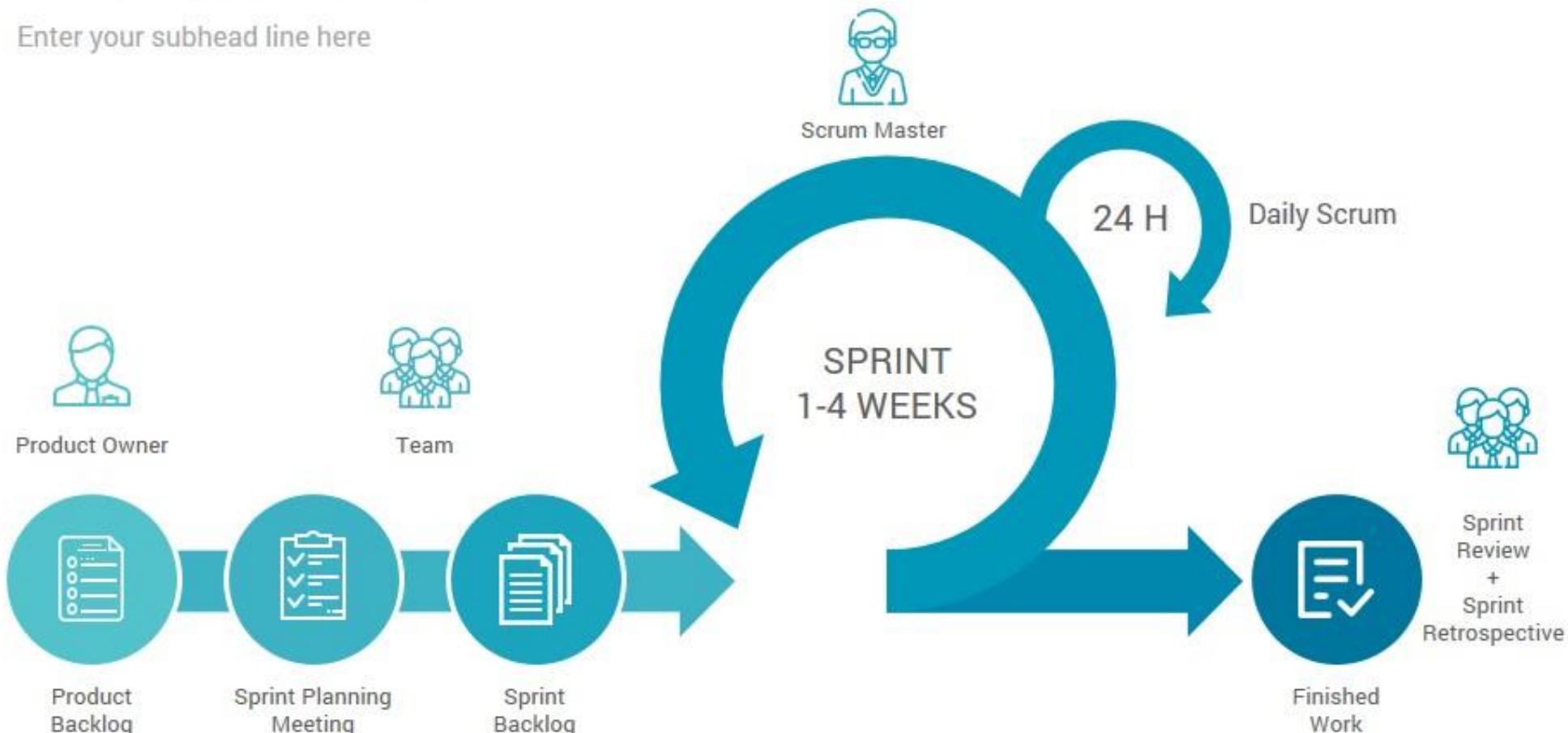
- **Prerequisite:** Team Setup
- Teamwork Contract (35 pts)
- Project Planning (35 pts)
- *Extra Credit* (7 pts)



Elements of Scrum

Scrum Process

Enter your subhead line here





XS



S



M



L



XL

made by :codica

codica.com

Working solo vs. as a team



Team survey

RESEARCH-ARTICLE



Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys

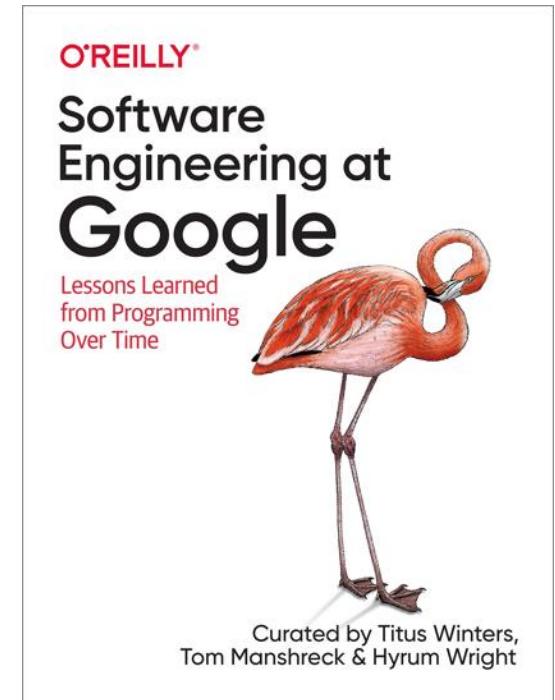
Authors:  Kai Presler-Marshall,  Sarah Heckman,  Kathryn T. Stolee [Authors Info & Claims](#)

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>

Design Documents

- The *best* design docs suggest design goals, and cover alternative designs, documenting the strengths and weaknesses of each.
- The *worst* design docs accidentally embed ambiguities, which cause implementors to develop contradictory solutions that the customer doesn't want.

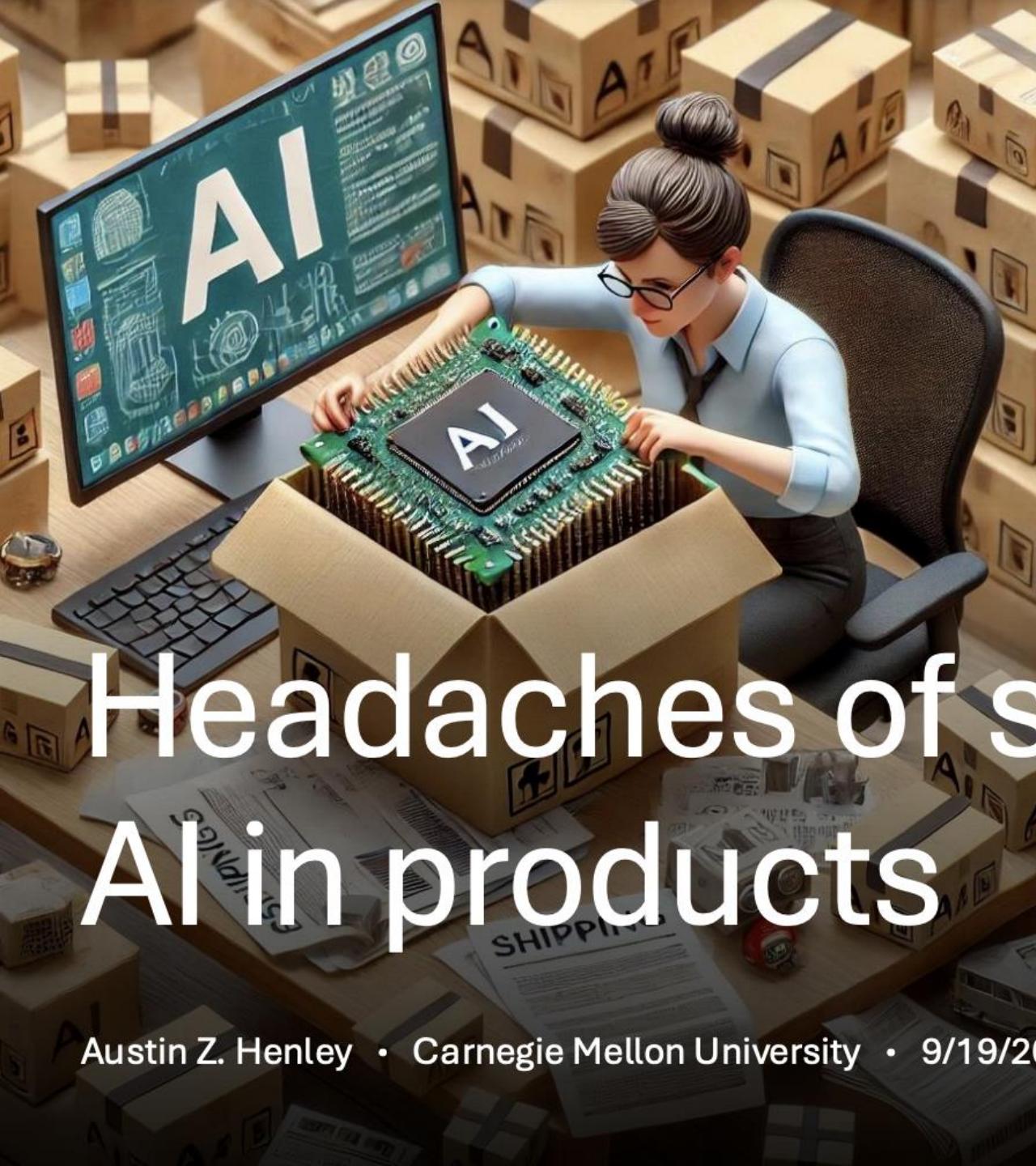


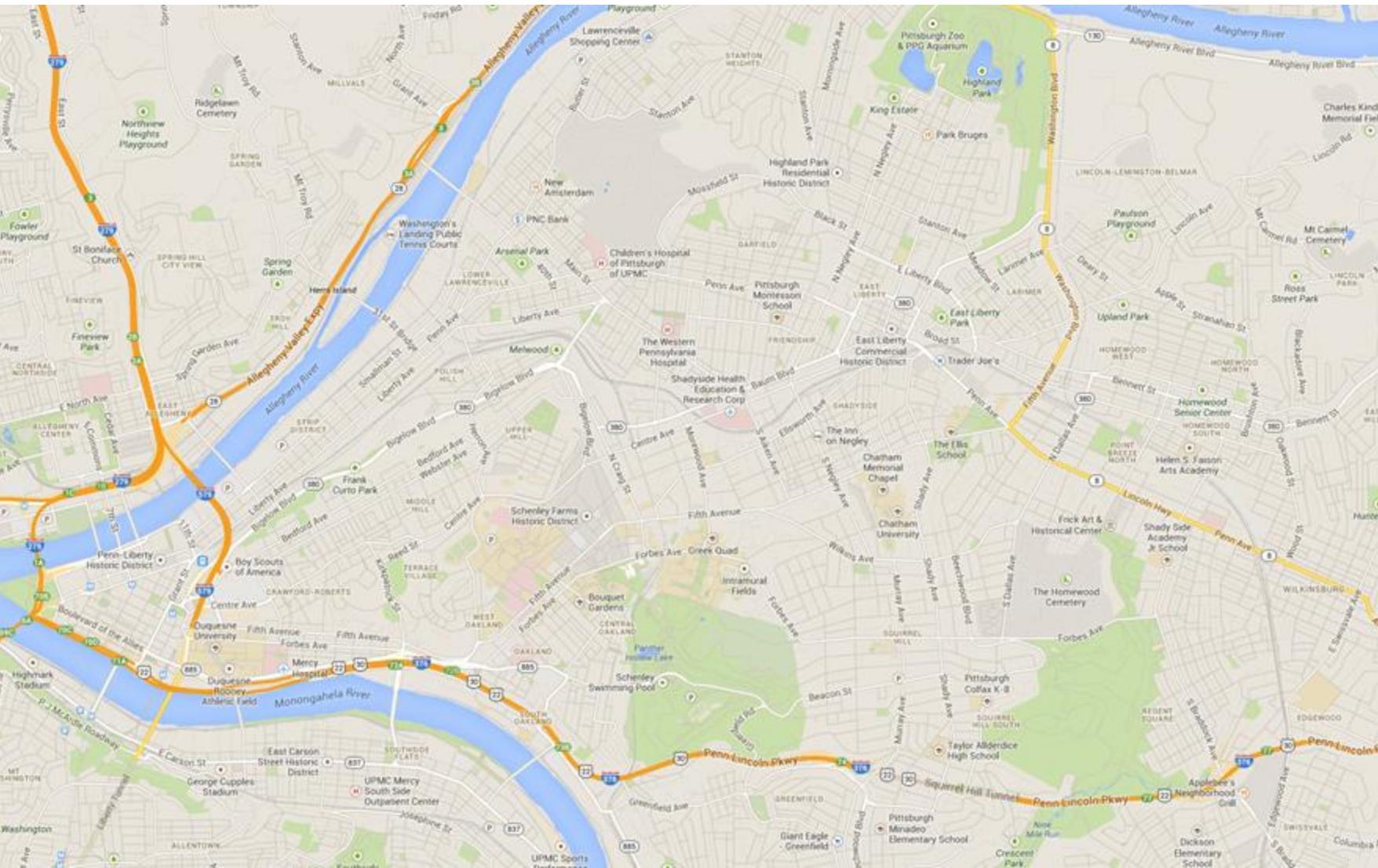
The screenshot shows a Google Docs interface with the following details:

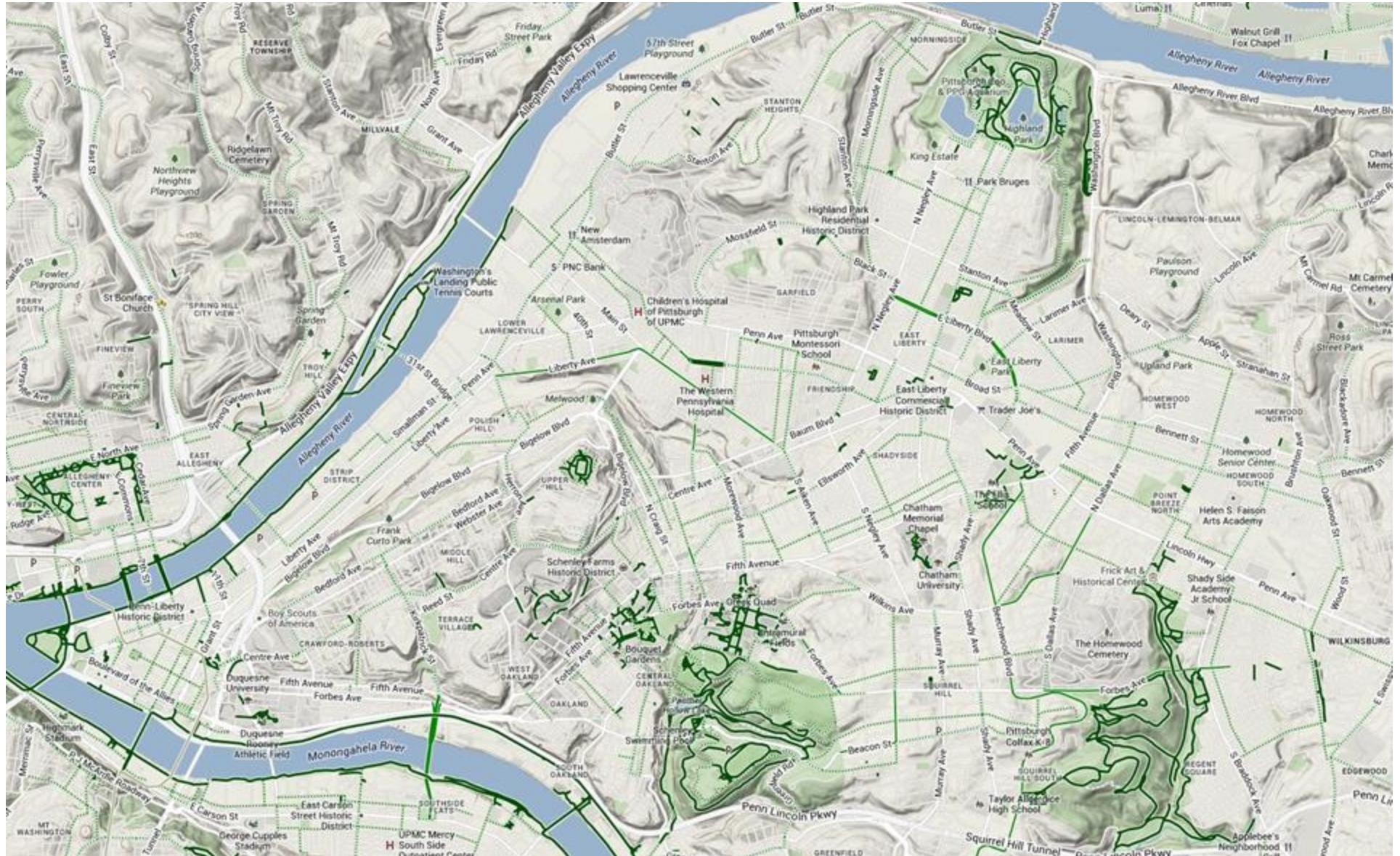
- Title:** Design Doc for Feature: More Sec...
- Toolbar:** Includes File, Edit, View, Insert, Format, Tools, Extensions, Help, and Share buttons.
- Text Area:** The main content area contains the following text:
 - Background:** We want to roll this app out on the app store, but we need to ensure that everything is secure and our liability is covered before we can do so. We want each user to be protected.
 - Problem:** There is not enough authentication in the scooter app which leaves users vulnerable to security breaches. It also causes liability issues if we cannot guarantee that a signed in user was the one on the scooter.
 - Proposed Solution:** Two factor authentication (ie Duo Mobile), Third Party Auth Provider using OAuth (Clerk, Auth0, etc), Okta, Phone/email verification code
- Section:** Considered Alternatives:
 - Building an in-house email verification step
 - Password requirements (at least 10 characters, 1 number, 1 special character)
 - Paired with additional biometric authentication
 - Using pre-set keys like GitHub for authentication
 - Adding authorized device trusted by the user / unauthorize devices
 - Limiting the number of devices for signing in
 - Additional biometric authentication (using iPhone's face ID or fingerprint)

Headaches of shipping AI products

Austin Z. Henley • Carnegie Mellon University • 9/19/2024

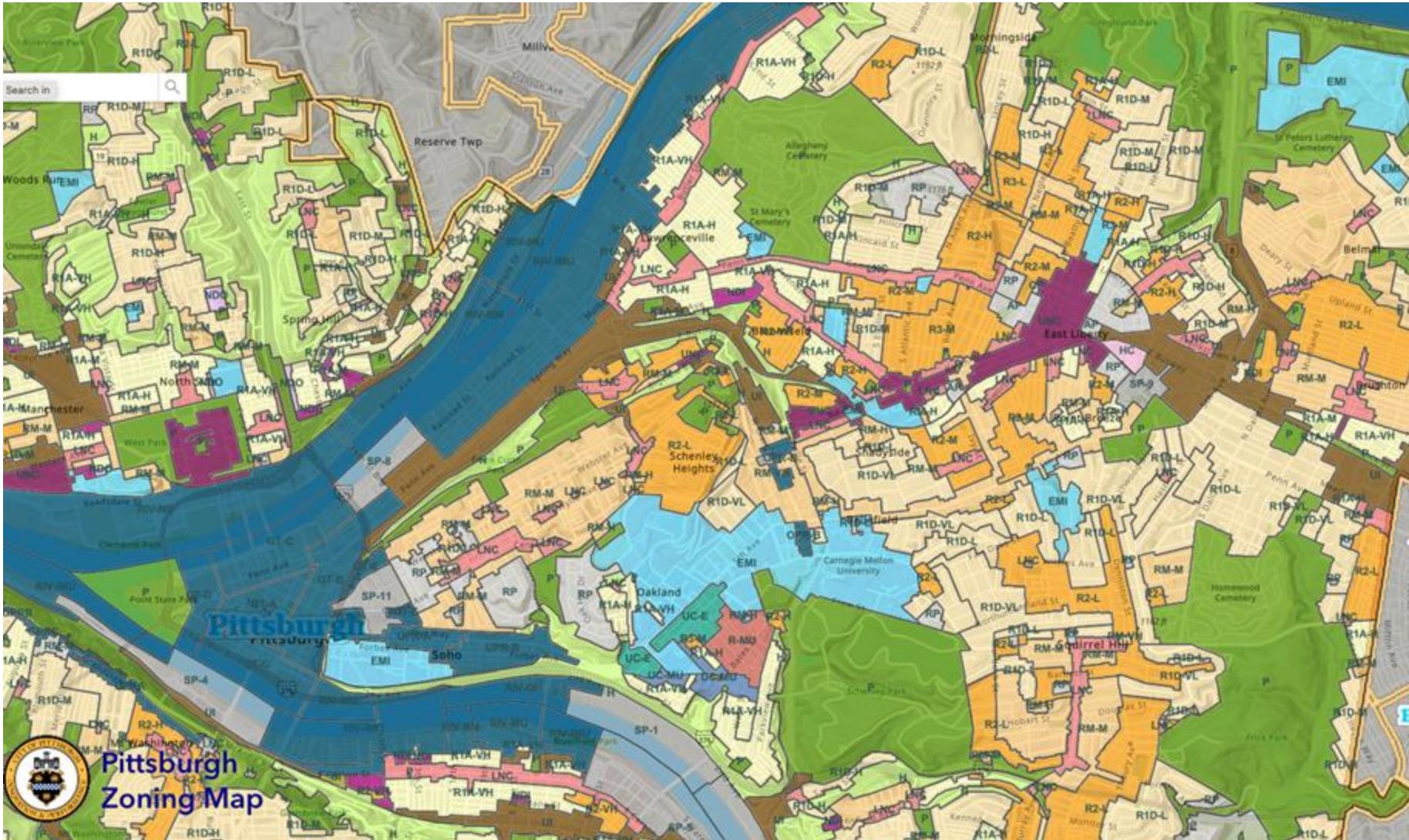


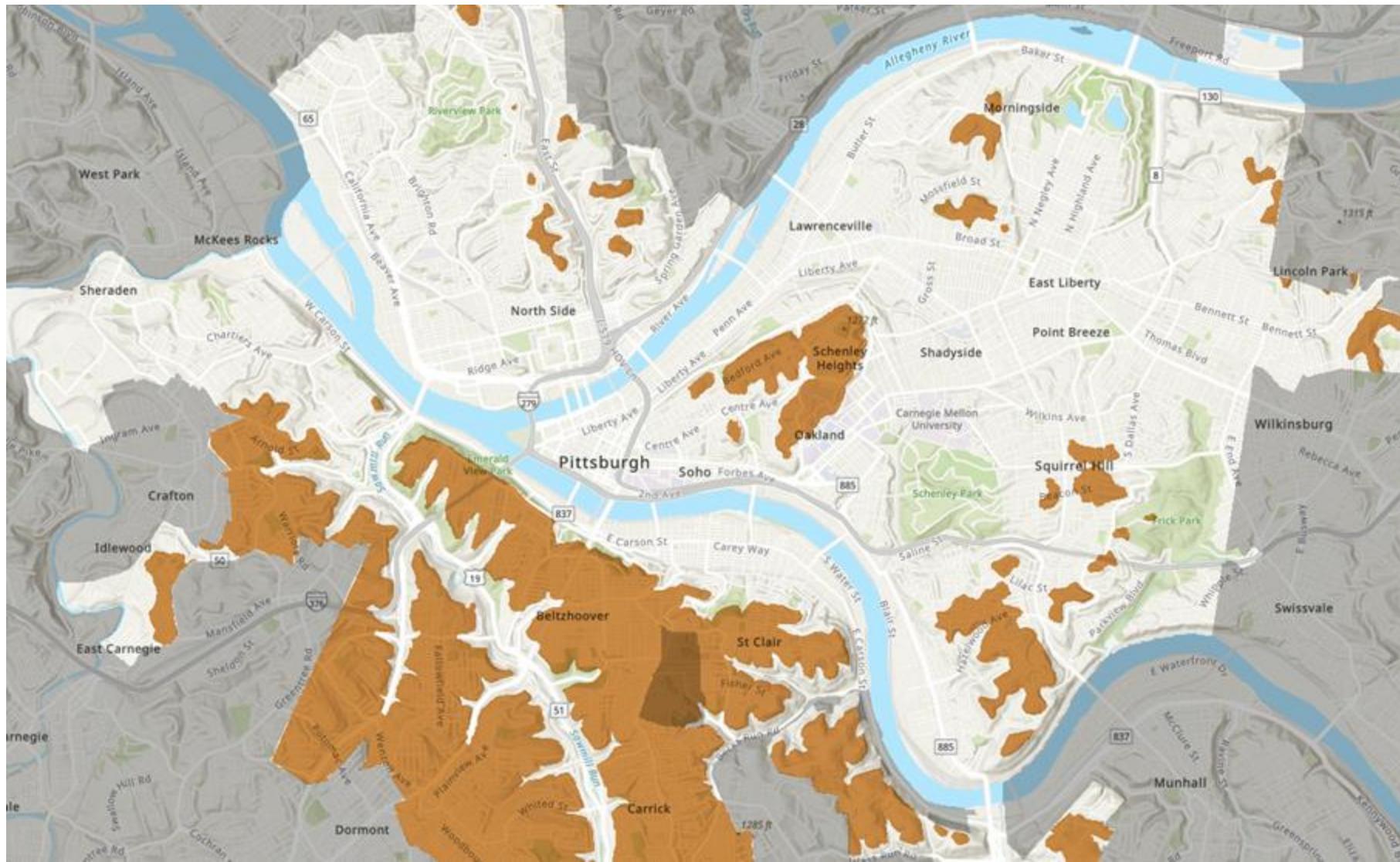




Fire Zones & Firehouses

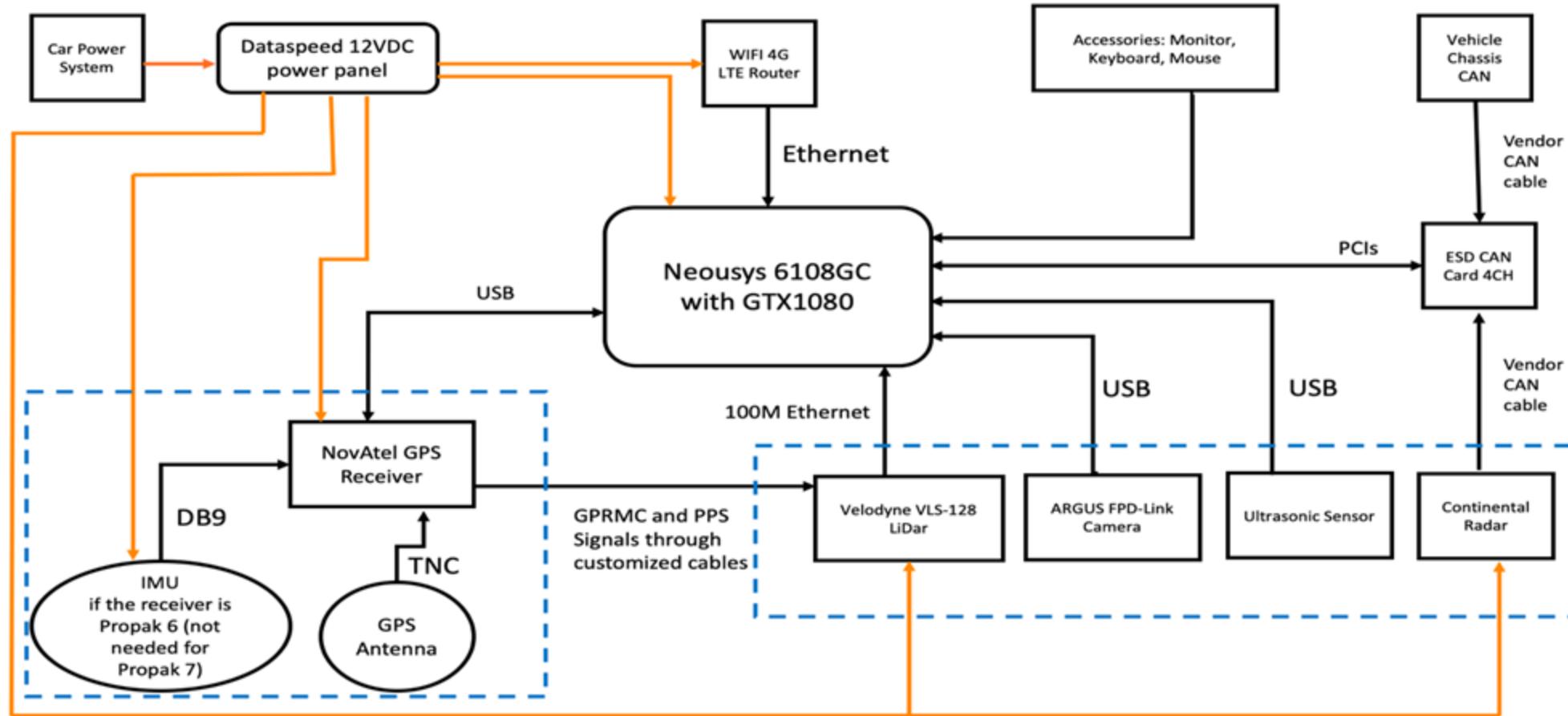




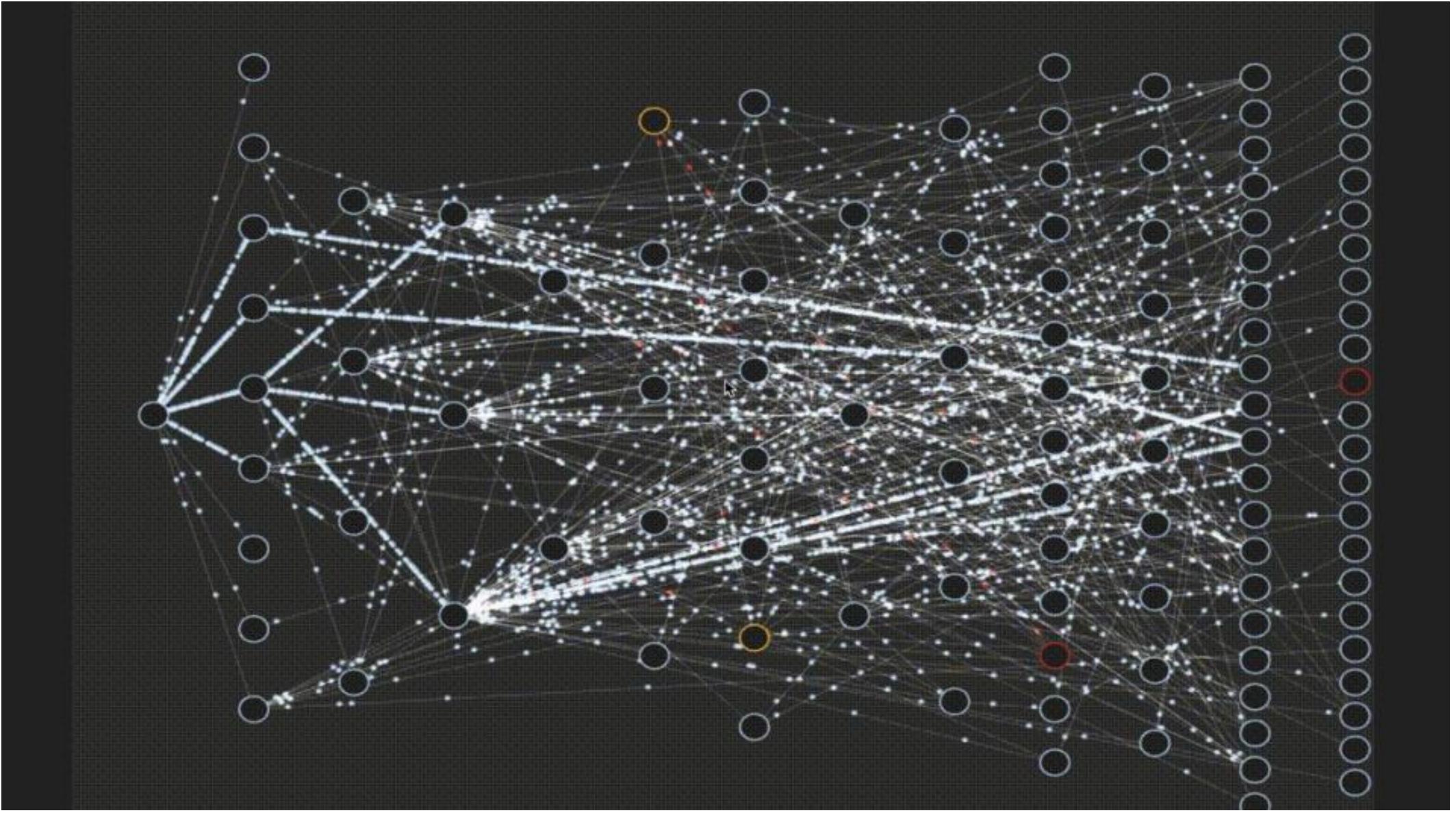


Source: Pittsburgh Zoning Map
<https://gis.pittsburghpa.gov/pghzoning/>

Apollo Hardware Architecture

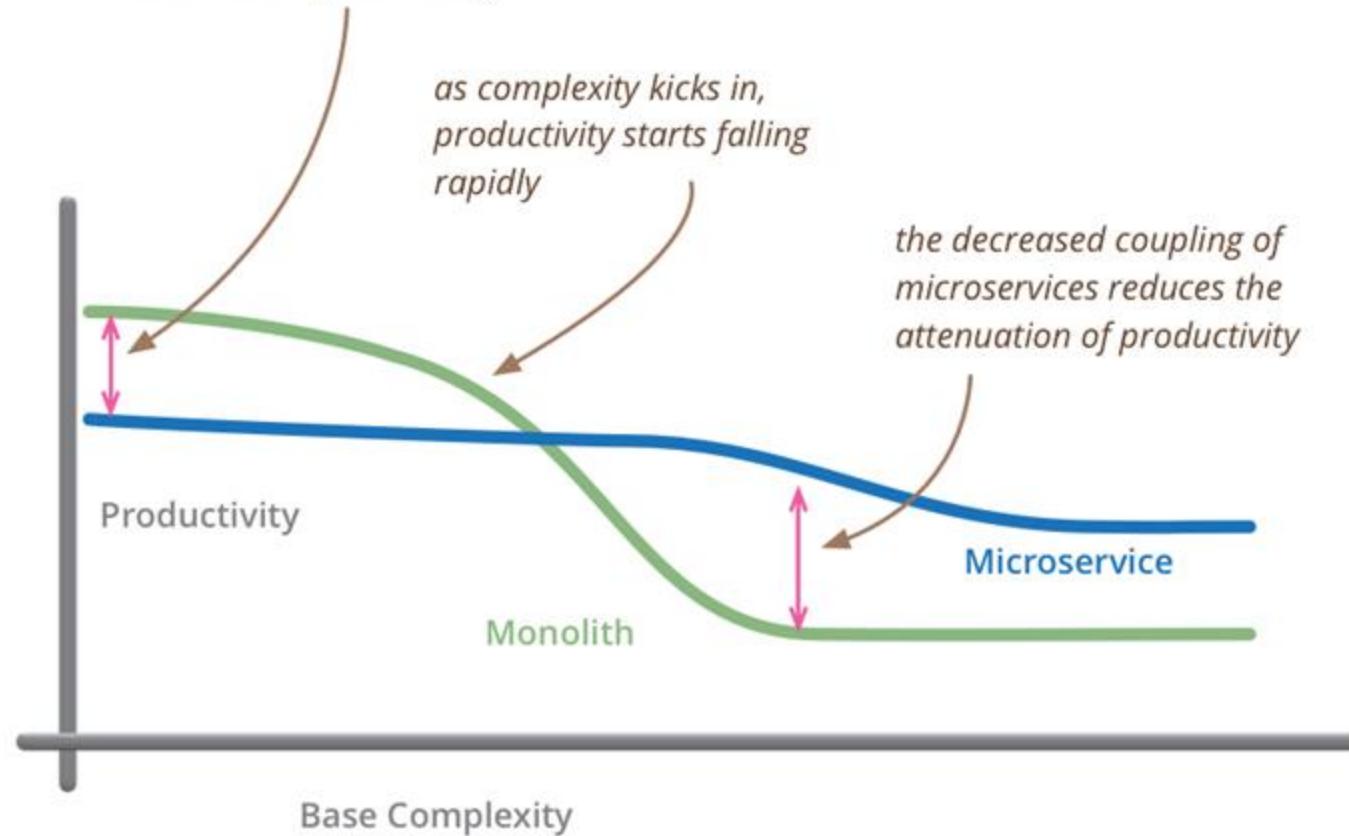


Source: <https://github.com/ApolloAuto/apollo/blob/v6.0.0/README.md>



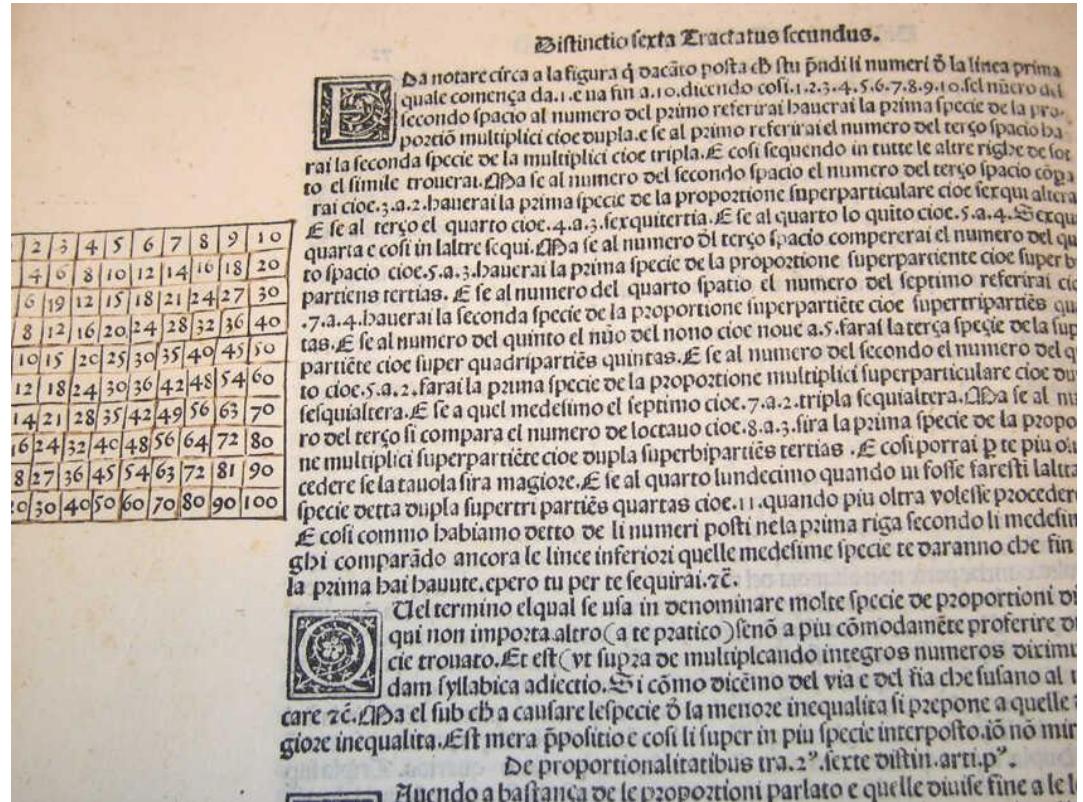
Microservices overhead

for less-complex systems, the extra baggage required to manage microservices reduces productivity



Risk





Checklists help manage complex processes



The Checklist: <https://www.newyorker.com/magazine/2007/12/10/the-checklist>

What are Program Analysis Tools?

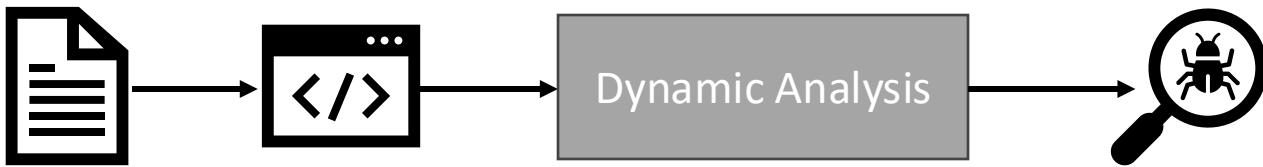


A screenshot of a static analysis tool interface. The code being analyzed is from a file named "src/controllers/accounts/posts.js". A specific line of code is highlighted with a red box and an error message: "This function expects 3 arguments, but 4 were provided." The code snippet includes lines 136 through 146.

```
src/controllers/accounts/posts.js
...
136 -     },
137 -     },
138 -     );
139 -
140 -     postsController.getBookmarks = async function (req, res, next) {
141 -         await getPostsFromUserSet('account/bookmarks', req, res, next);
142 -
143 -
144 -
145 -
146 -     postsController.getPosts = async function (req, res, next) {
147 -         await getPostsFromUserSet('account/posts', req, res, next);
148 -     };

```

This function expects 3 arguments, but 4 were provided.



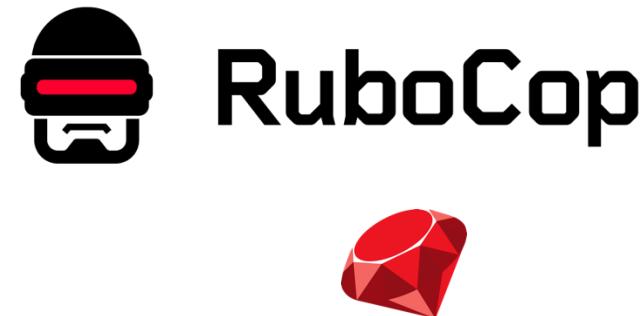
A screenshot of a dynamic analysis tool interface titled "COVERALLS". The interface shows a sidebar with navigation icons and a main area displaying a code coverage report. The code being analyzed is from a file named "Auth.js". Lines 65 through 88 are shown, with some lines highlighted in yellow and others in green, likely indicating different execution paths or coverage levels. The code snippet includes lines 65 through 88.

```
COVERALLS
...
65 Auth.reloadRoutes = async function (params) {
66     loginStrategies.length = 0;
67     const { router } = params;
68
69     // local Logins
70     if (plugins.hooks.hasListeners('action:auth.overrideLogin')) {
71         winston.warn('[authentication] Login override detected, skipping local login strategy');
72         plugins.hooks.fire('action:auth.overrideLogin');
73     } else {
74         passport.use(new passportLocal({ passReqToCallback: true },
75             controllers.authentication.localLogin));
76     }
77
78     // HTTP bearer authentication
79     passport.use('core.api', new BearerStrategy({}, Auth.verifyToken));
80
81     // Additional logins via SSO plugins
82     try {
83         loginStrategies = await plugins.hooks.fire('filter:auth.init',
84             loginStrategies);
85     } catch (err) {
86         winston.error(`[authentication] ${err.stack}`);
87     }
88     loginStrategies = loginStrategies || [];
89     loginStrategies.forEach((strategy) => {

```

Use linters to enforce style guidelines

Don't rely on manual inspection during code review!



17-313: Foundations of Software Engineering

Fall 2024 Midterm Exam

Michael Hilton and Rohan Padhye

Name: _____

Andrew ID: _____

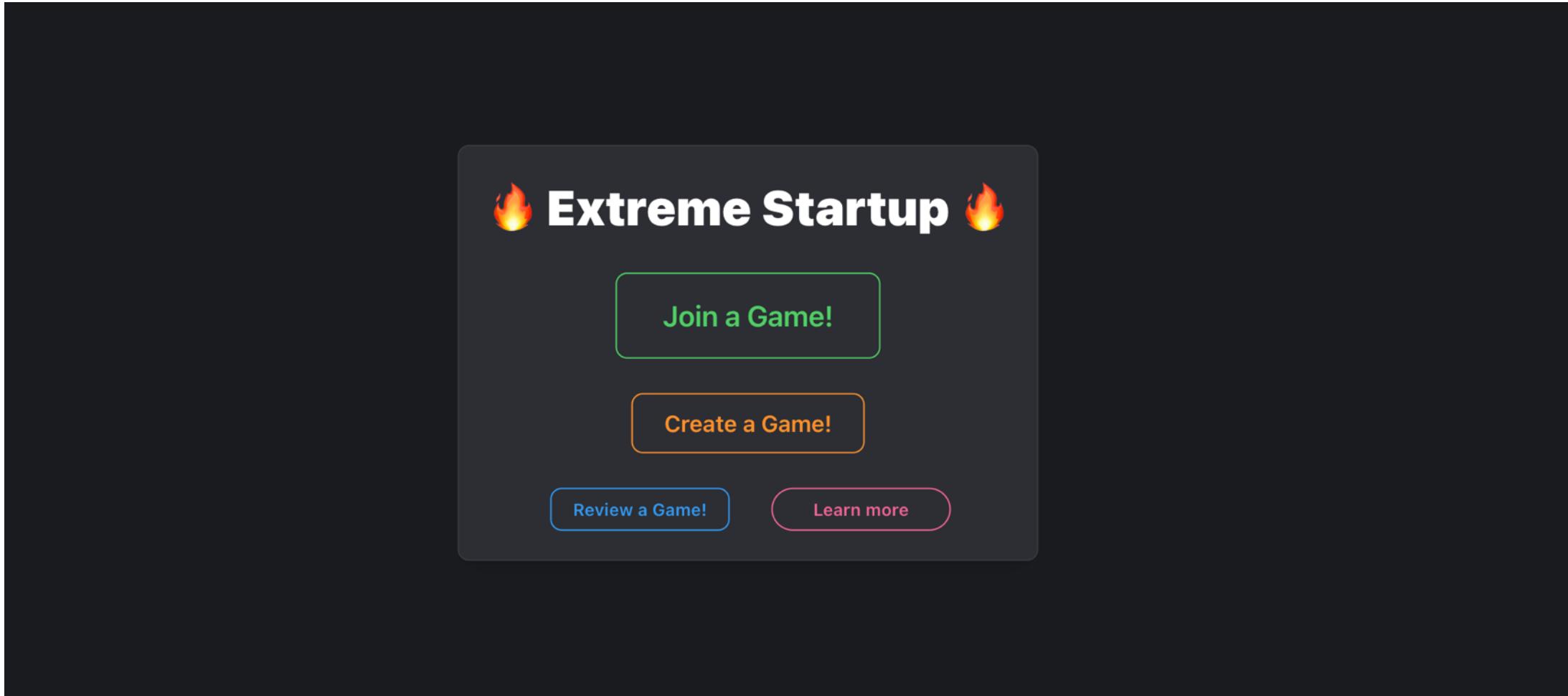
Scenario

You work as a lead software engineer at a local Pittsburgh startup called *AutoChair* (no relation to AutoDesk). The startup solves a key problem for people commuting to Downtown and Oakland by car; that of finding good parking spots near their workplace. The solution is to leverage the local custom of using a parking chair to reserve a spot even when not using it (see right).

AutoChair allows customers to pay for a spot in advance using a mobile app that provides a map-based view of the local neighborhoods. Users must choose a region of one or more blocks where they want to park, and the app estimates a cost based on selected region size, hourly parking costs in the area, and service fees. If the user accepts and pays, *AutoChair* dispatches an electric autonomous parking vehicle (APV), which is shaped like a chair, to drive around the user selected region of the neighborhood and find a parking spot to reserve. When the customer's car arrives at the location, the APV releases the parking spot and drives away in service of the next customer. If the APVs can't find an open parking spot before the customer's car arrives, then *AutoChair* must provide a full refund.

Your team is in charge of the backend system that must smoothly coordinate between customers, payment processors, the city's parking meters, and APVs. You are also developing internal software for use by the operations ("ops") team, which is responsible for managing the fleet of APVs.





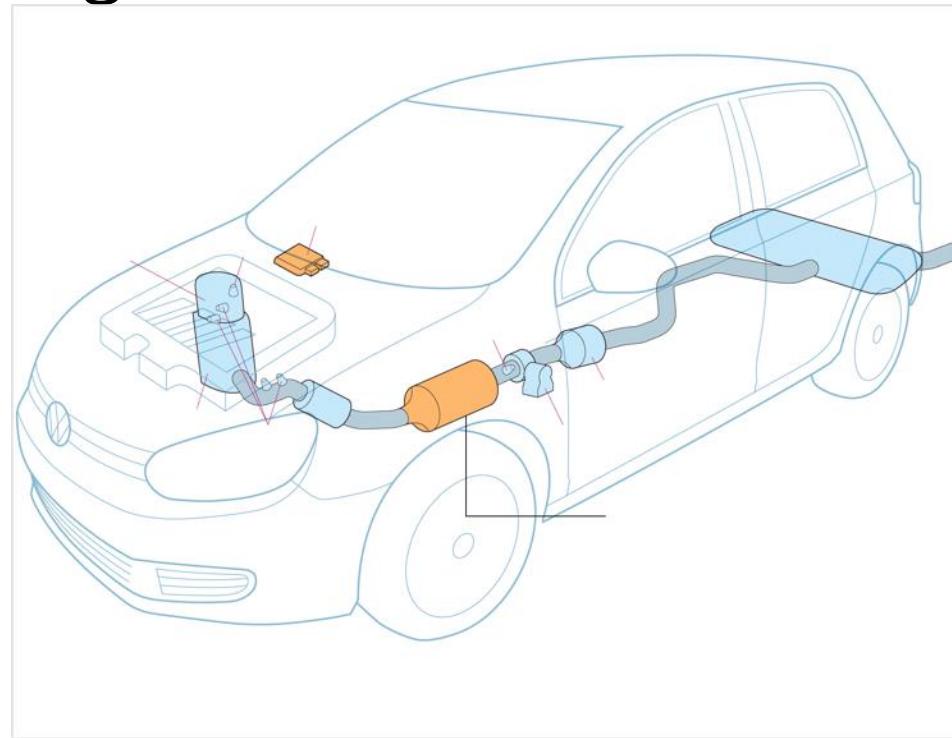
Project 3: Continuous Integration + Deployment

Learning Goals

- Learn how to deploy a full-stack application
- Gain hands-on experience with analysis tools, including setting up, customizing, and using them
- Practically assess and compare the costs and benefits of existing static and dynamic bug-finding tools
- Integrate CI/CD tools into development practice

Volkswagen Scandal

VW was caught cheating on emissions for Diesel engines

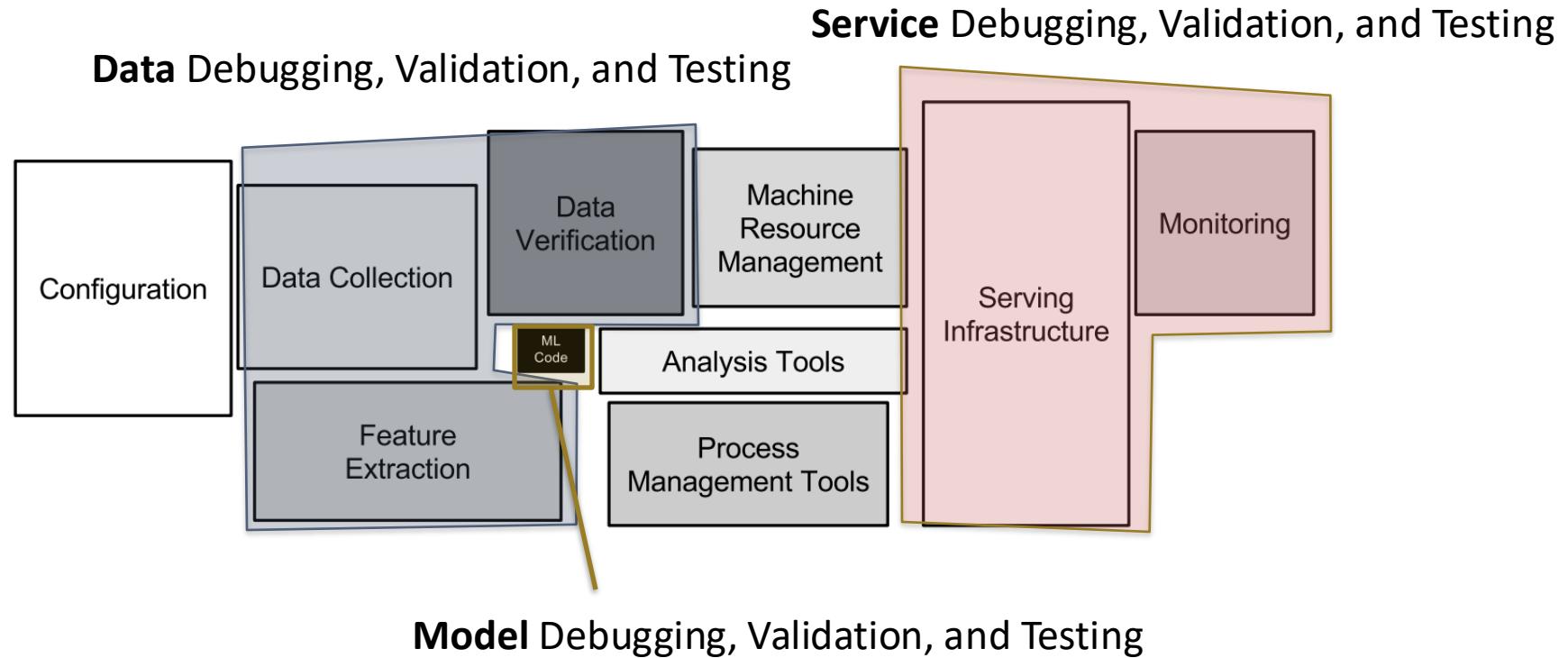


<https://www.nytimes.com/interactive/2015/business/international/vw-diesel-emissions-scandal-explained.html?mtrref=www.google.com&assetType=REGIWALL>

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?

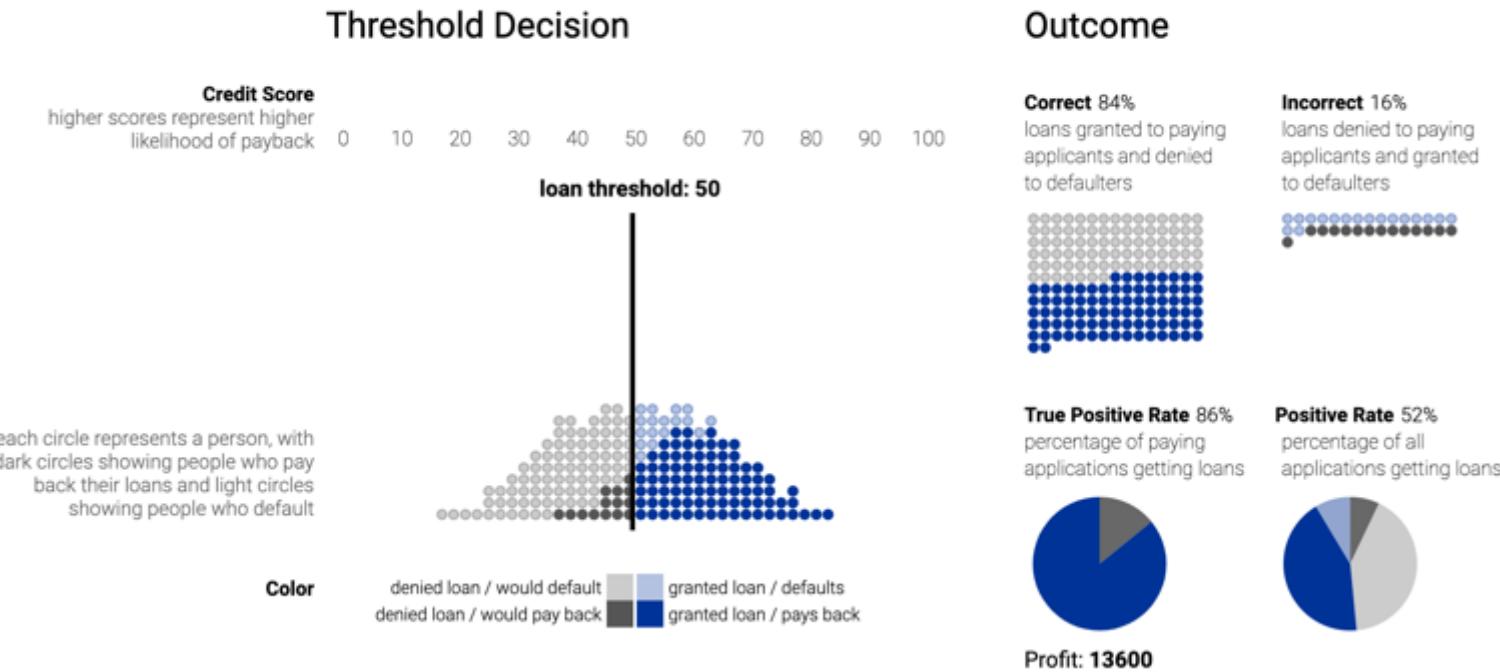
What does it mean to do QA for a ML System?



Explainability

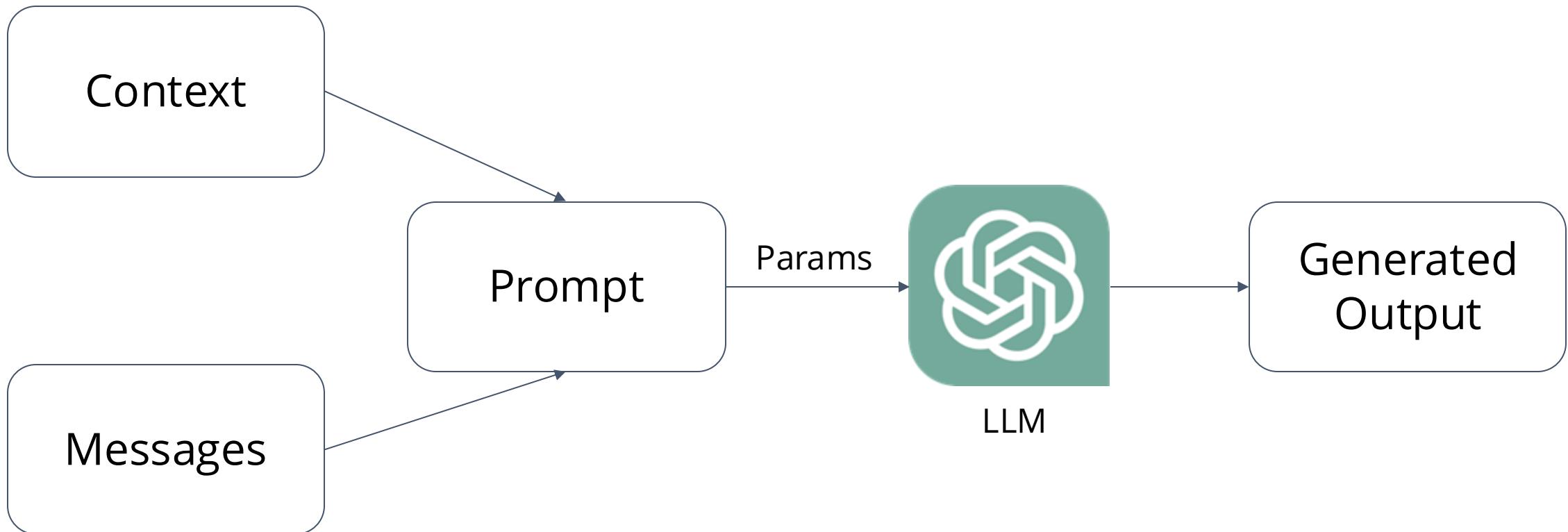
Simulating loan thresholds

Drag the black threshold bars left or right to change the cut-offs for loans.



<https://research.google.com/bigpicture/attacking-discrimination-in-ml/>

Basic LLM Integration



Barton P. Miller, Lars Fredriksen and Bryan So

Study of the Reliability of UNIX Utilities

COMMUNICATIONS OF THE ACM December 1990/Vol.33, No.12

33

“

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.

”

Communications of the ACM (1990)

How to identify these bugs?

Fuzz Testing



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



GIVING AND RECEIVING FEEDBACK

2024

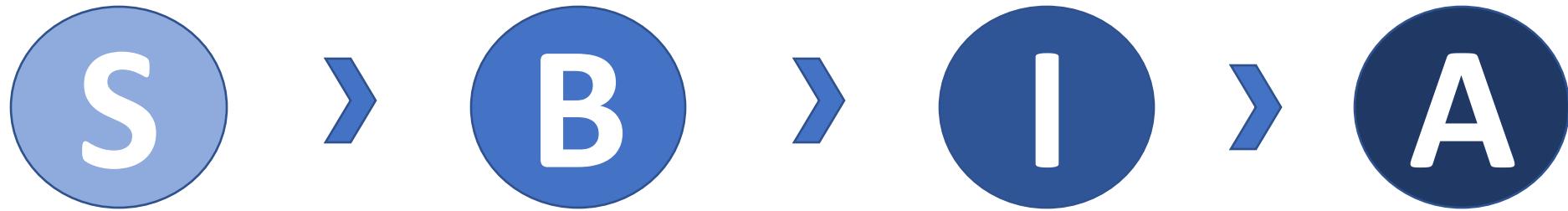
Taya R. Cohen, Ph.D.

Professor of Organizational Behavior and Business Ethics

Carnegie Mellon University
Tepper School of Business

THE INTELLIGENT FUTURE

DEVELOPMENTAL FEEDBACK (SBIA)



Situation

Set the context.
Help put the person
in the focus on what
you are referring to.

Behavior

Focus on the
objective behavior
to be repeated or
changed.

Impact

Share the direct
impact of the
behavior.

Alternative

Share an alternative
behavior to use next
time.

Adapted from Center for Creative Leadership

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
- 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

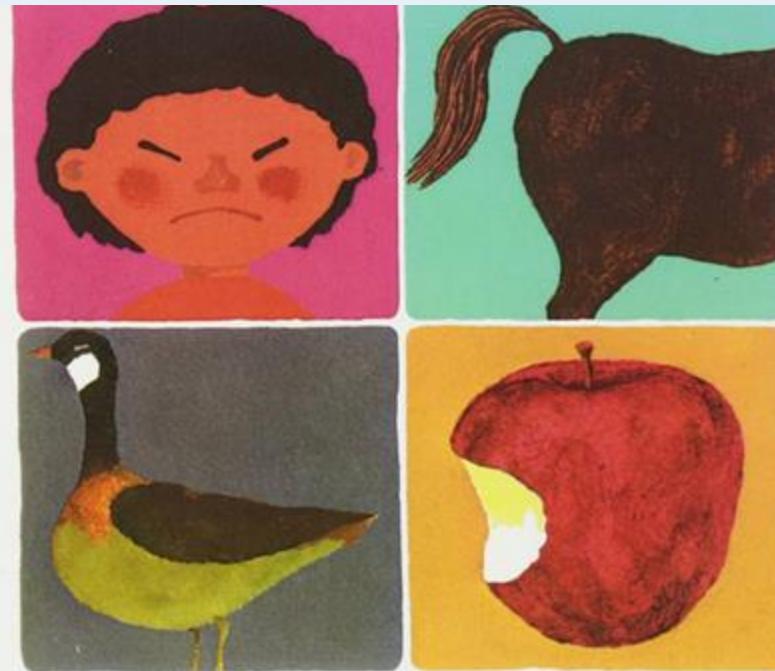
A better analogy?: Pollution

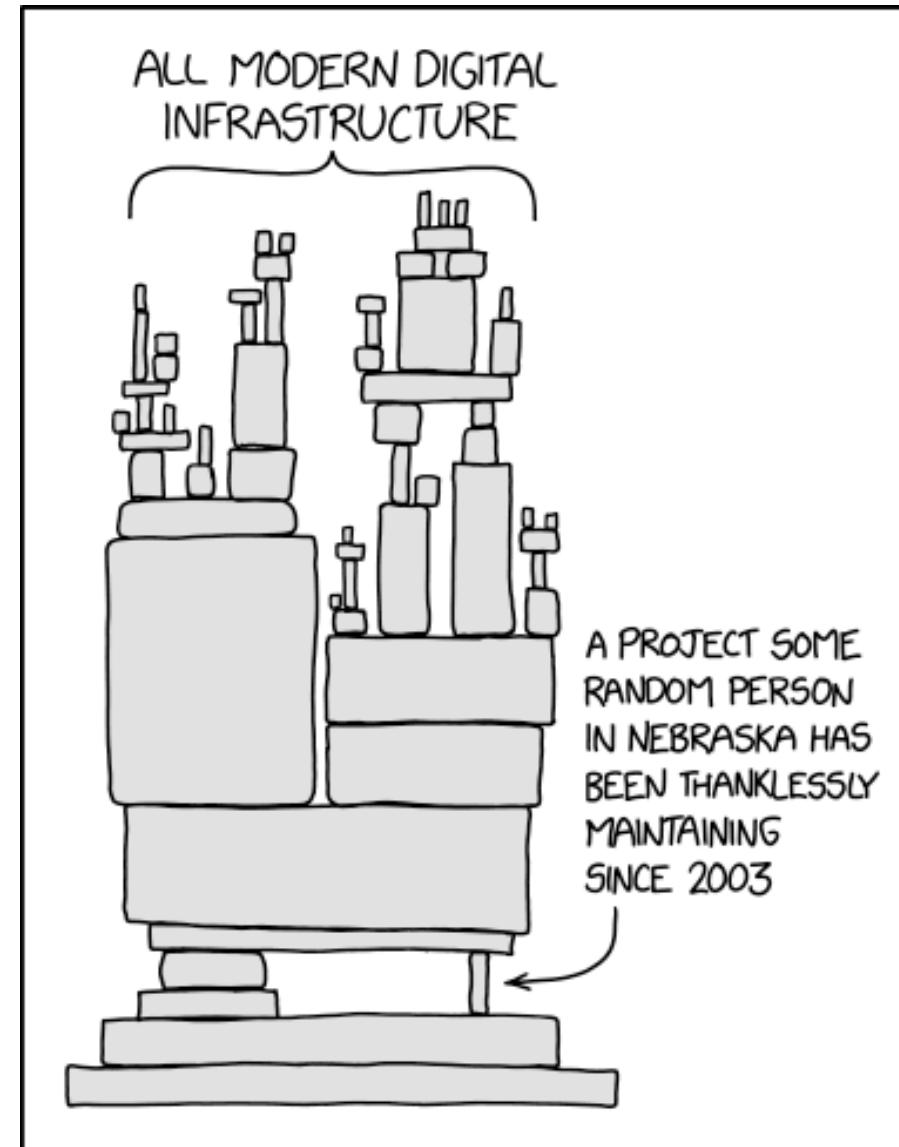


AQI Basics for Ozone and Particle Pollution			
Daily AQI Color	Levels of Concern	Values of Index	Description of Air Quality
Green	Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Yellow	Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Red	Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Purple	Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

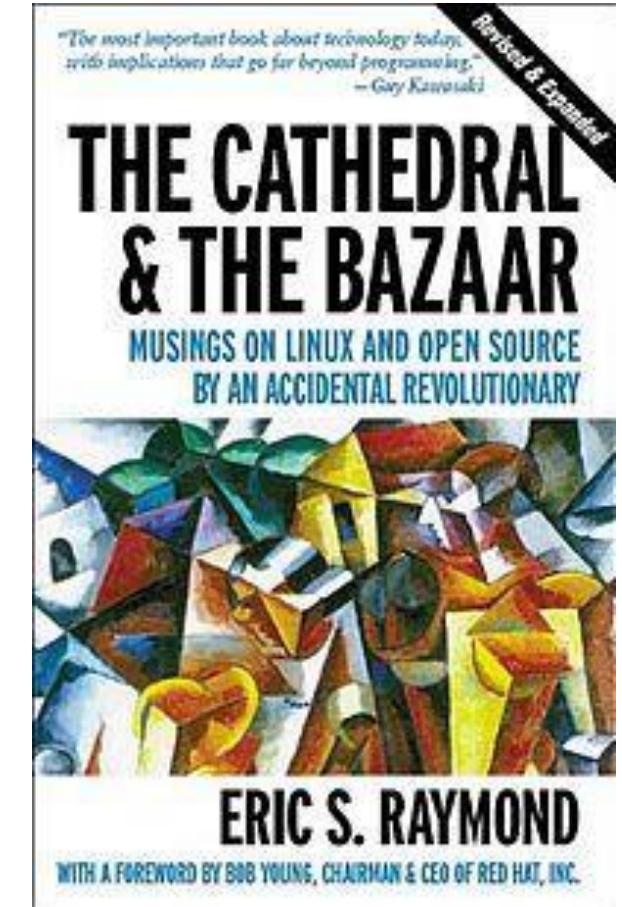
<https://www.airnow.gov/aqi/aqi-basics>

EVERYONE CREATES TECHNICAL DEBT





The Cathedral and the Bazaar



Left-pad (March 22, 2016)



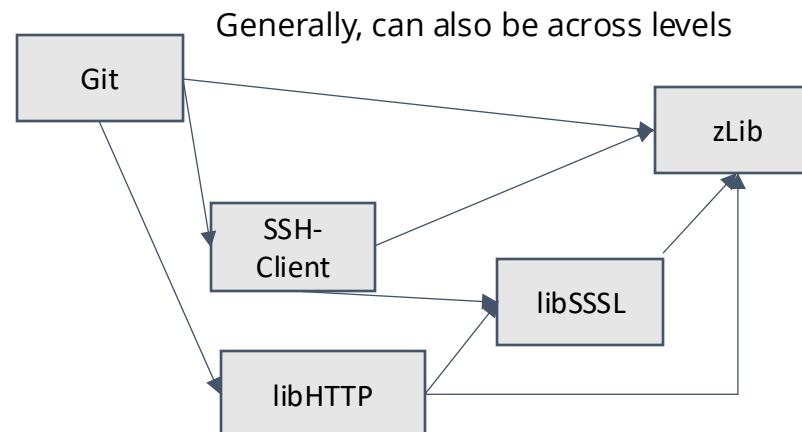
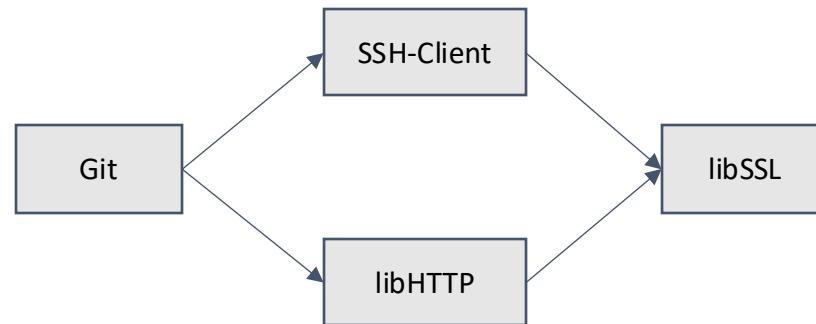
The screenshot shows a news article from The Verge. At the top left is the 'THE VERGE' logo with 'TECH' and 'REVIEWS' dropdown menus. To the right is the 'OBSSESSIONS' section with a 'QUARTZ' logo. Below the header, the text 'NPM ERR!' is visible. The main title is 'How one programmer broke the internet by deleting a tiny piece of code'. A sub-section below it reads 'How an irate developer briefly broke JavaScript'. The text 'Unpublishing 11 lines of code brought down an open source house of cards' is present. The author is listed as 'By Paul Miller | @futurepaul | Mar 24, 2016, 4:29pm EDT'. Social sharing icons for Facebook and Twitter are at the bottom left, and a 'SIGN IN' button is on the right. The Register logo is also visible.

How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using

Diamond Dependencies

What are some problems when multiple intermediate dependencies have the same transitive dependency?



Project 5: Open Source Excursion

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

Reliably Releasing Software

Foundations of Software Engineering

Christopher S. Meiklejohn
Software Engineer
DoorDash

Carnegie Mellon University



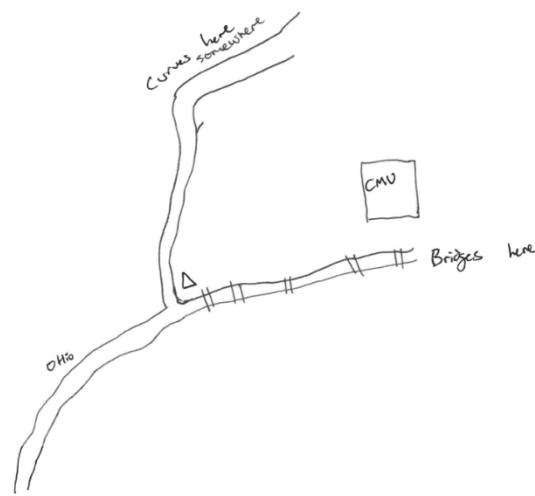
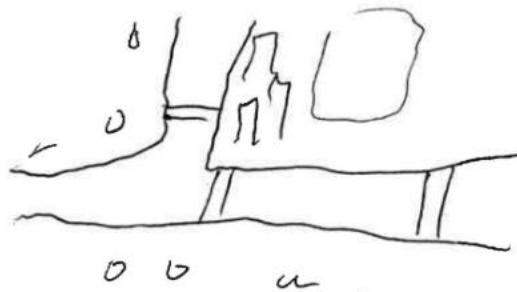
17-313: Foundations of Software Engineering

Fall 2024 Midterm II Exam

Michael Hilton and Rohan Padhye

Name: _____

Andrew ID: _____



10X Engineers

- Aka “rock-star”, “ninja”

ROCK STAR DEVELOPER



Congratulations! You were offered equity!: Stock Options

Running Example: Kerri

Kerri's offer tells her the number of shares of option grants she was offered, but not how much they're worth.

This is standard.

MEETLY
2525 CHARLESTON ROAD SUITE 104
MOUNTAIN VIEW, CA 94043

SEPTEMBER 3, 2015

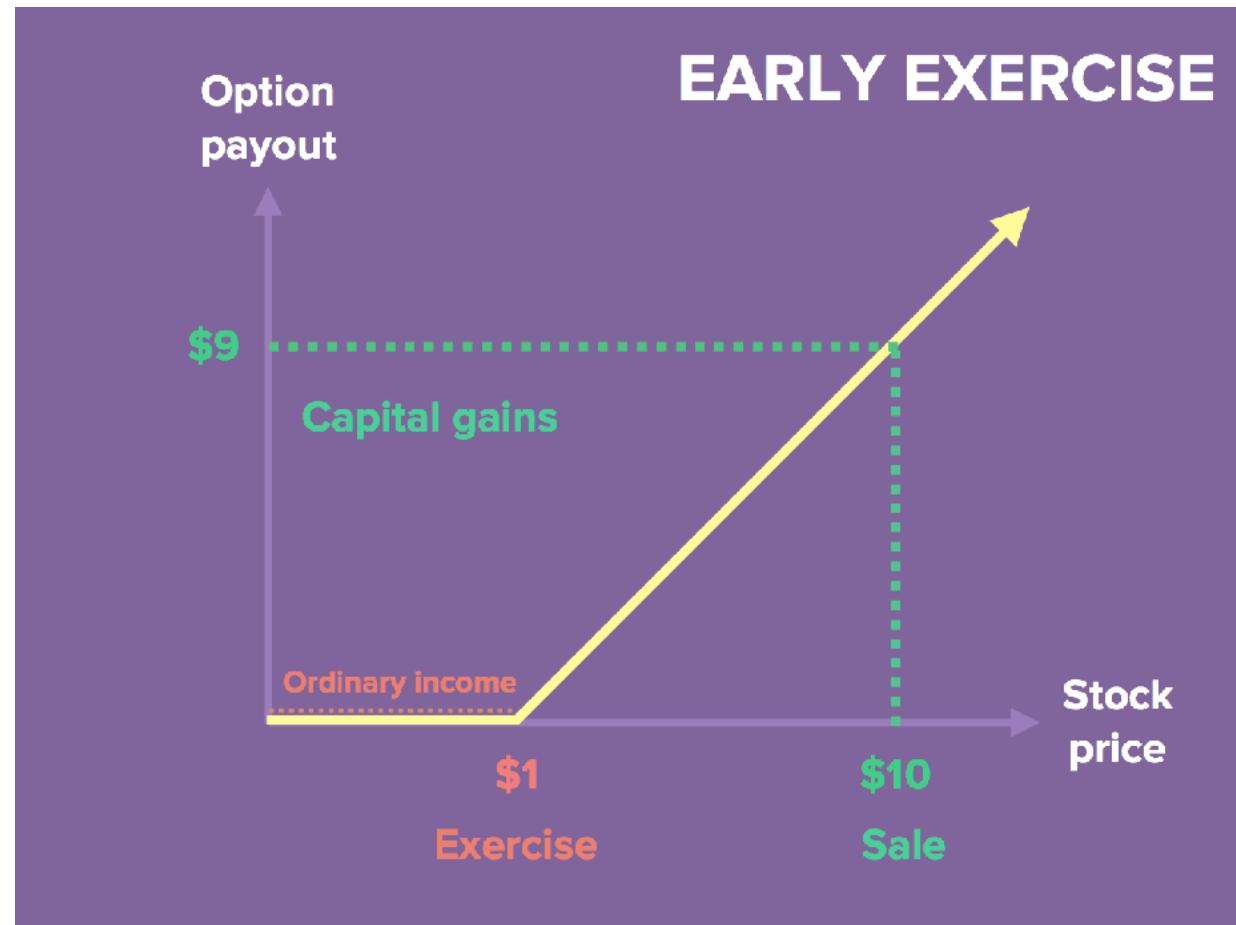
Kerri Stuart
Via email to: kerri@example.com

Dear Kerri:

I am pleased to offer you a position with **Meetly, Inc** (the "Company"), as Senior Engineer. If you decide to join us, you will receive an annual salary of \$100,000, which will be paid semi-monthly in accordance with the Company's normal payroll procedures. In addition, subject to the approval of the Company's Board of Directors, you will be granted 100 shares of option grants of the Company, which shares will vest in accordance with the following vesting schedule: 25% on your first year anniversary date, and the remainder for 1/36th for each month of continuous service thereafter. You should note that the Company may modify job titles, salaries and benefits from time to time as it deems necessary.

The Company is excited about your joining and looks forward to a beneficial and productive relationship. Nevertheless, you should be aware that your employment with the Company is for no specified period and constitutes at will employment. As a result, you are free to resign at any time, for any reason or for no reason. Similarly, the

Holding periods in Kerri's example



Risky because:

There is no guarantee that your stock will ever be liquid, so you are paying to buy stock that could one day be worthless.

AMA



