

# Introduction

17-313 Fall 2023

Foundations of Software Engineering

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

Andrew Begel and Rohan Padhye

# Introductions

# Andrew Begel

Associate Professor at CMU  
16.5 years at Microsoft Research  
PhD from UC Berkeley – 2005

Web: [andrewbegel.com](http://andrewbegel.com)  
X/Twitter: @abegel  
Google: "Andrew Begel"



# Rohan Padhye

Assistant Professor at CMU  
Visiting Academic, Amazon  
PhD from UC Berkeley – 2020

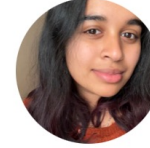
Web: [rohan.padhye.org](http://rohan.padhye.org)  
X/Twitter: @moarbugs  
Google: "Rohan Padhye"



# Teaching Assistants



**Anuda Weerasinghe**



**Antara Pal**



**Alexis Axon**



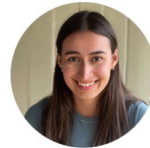
**Eyob Dagnachew**



**Grace Xin**



**Michael Zhou**



**Sophia Witt**



**Vasu Vikram**



# Software is everywhere



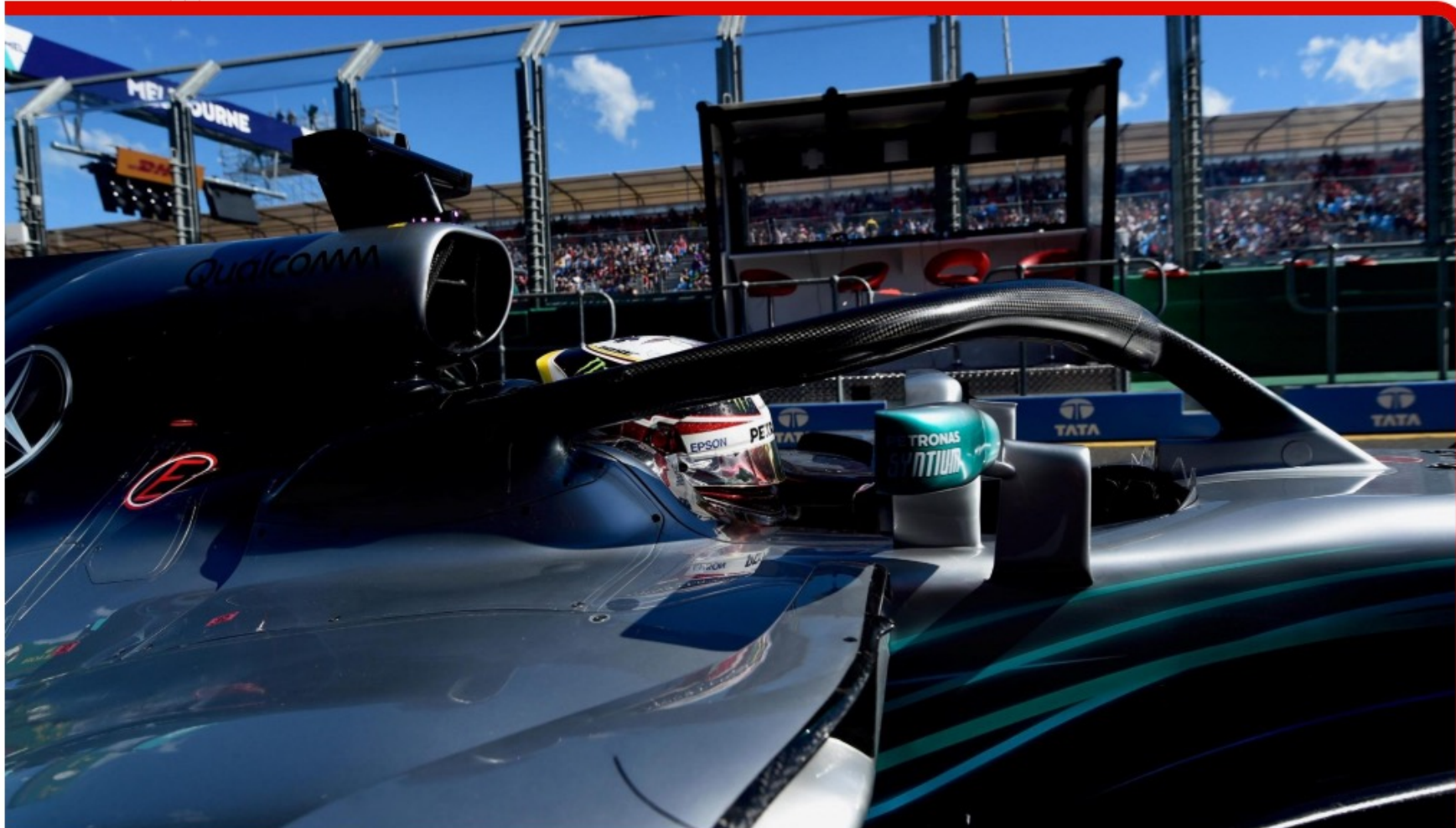
# Software glitch cost Hamilton victory - Mercedes

25 March 2018

MERCEDES

AUSTRALIA

HAMILTON



# The System is down at the moment.

We're working to resolve the issue as soon as possible. Please try again later.

## Forbes

# HealthCare.gov Diagnosis: The Government Broke Every Rule Of Project Management



**Loren Thompson** Senior Contributor

Aerospace & Defense

*I write about national security, especially its business dimensions.*

f After 400 software fixes and major hardware upgrades, the Obama Administration is claiming to have achieved its goal of transforming HealthCare.gov into a web-site that



The Patient Protection and Affordable Care Act, better known as Obamacare, will probably be remembered as President Obama's most important domestic policy initiative. However, inept federal management of the HealthCare.gov



Obama Speech Today on Tech Problems of HealthCare.gov | The New York Times

The New York Times 4.34M subscribers

Subscribe

41



Share

Download



Figure 1, Probabilistic Consequence Gra

121	relativ
121	
135	
92	
121	
92	



# REDLINE

The many human errors that brought down the Boeing 737 Max

Catastrophic Accident

9 IN (5.72 M)

112 IN (12.55 M)

Effects of Occupants

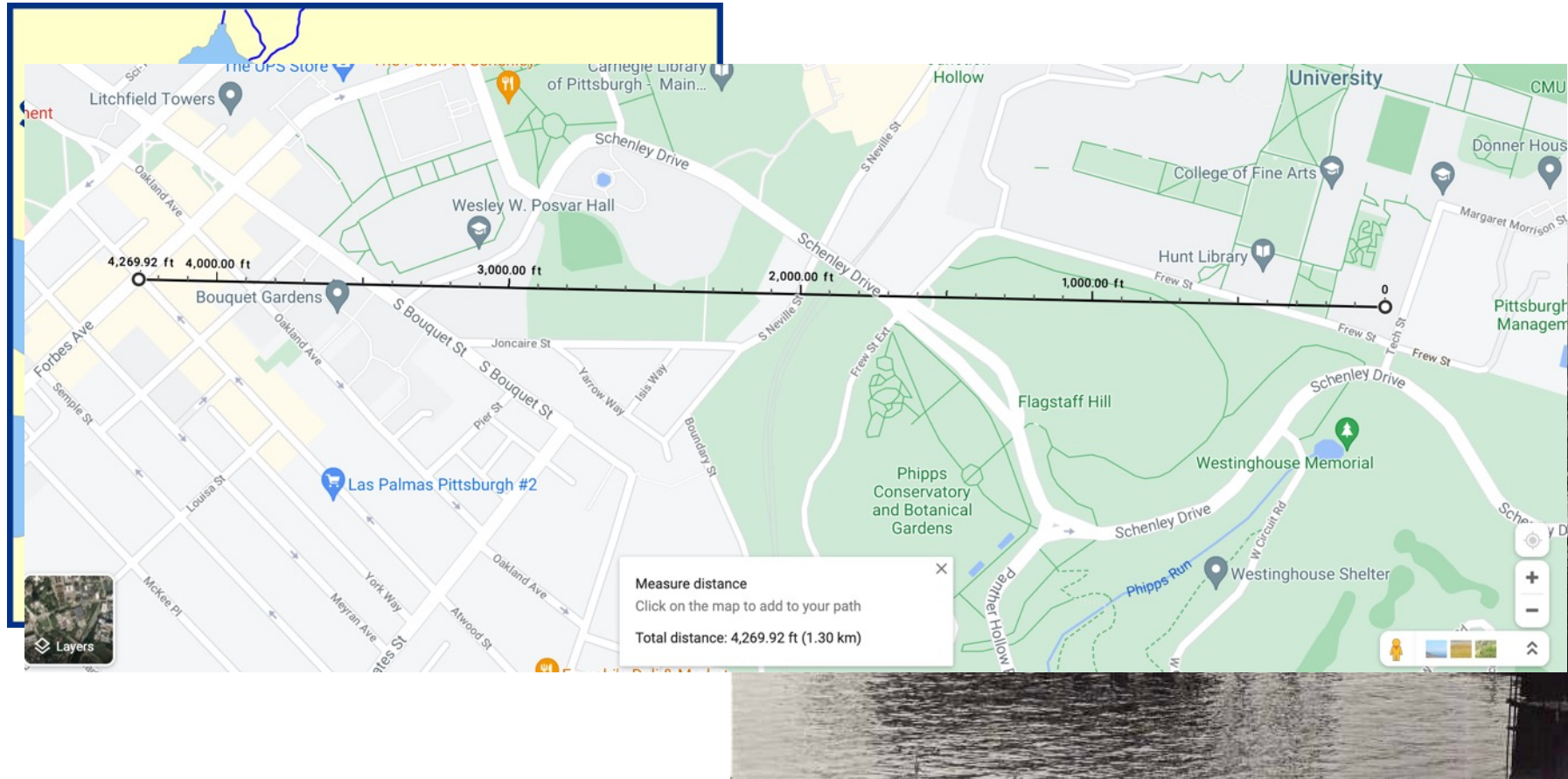


# Vasa





# Vasa



# What happened is now called “Vasa syndrome”

- Changing shipbuilding orders
- No specifications for modified keel
- Shifting armaments requirements

Requirements

- Shipwright's death

Teams

- No way to calculate stability, stiffness, or sailing characteristics

Metrics

- Failed pre-launch stability tests

QA



# Software Engineering?

- What is engineering?
- And how is it different from hacking/programming?

# 1968 NATO Conference on Software Engineering

- Provocative Title
- Call for Action
- “Software crisis”



# Margaret Hamilton



# This Course





## Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana<sup>a</sup>, Tina Weston<sup>b,c</sup>, Nicholas J. Cepeda<sup>b,c,\*</sup><sup>a</sup>McMaster University, Department of Psychology, Neuroscience, & Behaviour, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada<sup>b</sup>York University, Department of Psychology, 4700 Keele Street, Toronto, ON M3J 1P3, Canada<sup>c</sup>York University, LaMarsh Centre for Child and Youth Research, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

## ARTICLE INFO

## ABSTRACT

- “...participants who multitasked on a laptop during a lecture scored lower on a test compared to those who did not multitask, and participants who were in direct view of a multitasking peer scored lower on a test compared to those who were not. The results demonstrate that multitasking on a laptop poses a significant distraction to both users and fellow students and can be detrimental to comprehension of lecture content.”



## Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana<sup>a</sup>, Tina Weston<sup>b,c</sup>, Nicholas J. Cepeda<sup>b,c,\*</sup><sup>a</sup>McMaster University, Department of Psychology, Neuroscience, & Behaviour, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada<sup>b</sup>York University, Department of Psychology, 4700 Keele Street, Toronto, ON M3J 1P3, Canada<sup>c</sup>York University, LaMarsh Centre for Child and Youth Research, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

## ARTICLE INFO

## ABSTRACT

- "...participants who multitasked on a laptop during a lecture scored lower on a test compared to those who did not multitask, and participants who were in direct view of a multitasking peer scored lower on a test compared to those who were not. The results demonstrate that multitasking on a laptop poses a significant distraction to both users and fellow students and can be detrimental to comprehension of lecture content."

# Smoking Section

- Last full row



# Course infrastructure and logistics

- Infrastructure/source of truth
  - Course website: schedule, slides, syllabus, office hours
  - Canvas (and Gradescope) homework, grades, other material
  - Slack for communication and collaboration.
  - Git/Github for coding and collaboration
- Logistics
  - Lecture in-person only
  - All recitations are in-person
- Office Hours are flexible.
- If you want to talk to us, DM/email ALL INSTRUCTORS at once.  
Trust us, it's faster.



# Connect with us for the class

- All links on our course website: <https://cmu-313.github.io>
- Canvas: <https://canvas.cmu.edu/courses/36599>
- We will send you an invite for Slack, please be on the lookout for it.

Hello  
my name is

<Name>

What is the most recent software team project you've worked on, and how was your experience?

# Course Themes

- Software engineering as a human process
- Requirements and Specifications
- Metrics and Measurement
- Software Quality: Testing + CI + Security
- Continuous Deployment and DevOps
- Software Project Teams
- Managing Time, Teams, and Risks
- Software Architecture and Design Docs
- Scaling and Performance, Trade-offs
- AI/ML in Software Engineering
- Open-Source Software
- Strategic Thinking about Software

# Prerequisites

- Assumes working knowledge of popular programming languages (e.g., 15-121, 15-122)
- You will have the best experience if you have already had an internship or been involved in a large-ish software development project (ask us if you have any questions)
- How is it different from 17-214?
  - 17-313 largely focused on human issues and quality beyond functional correctness
  - 17-313 focused on larger scale



# Readings, Quizzes, and Participation Activities

- Reading assignments for some lectures
  - Preparing in-class discussions: background material, case descriptions, possibly also podcast, video, Wikipedia
- In-person activities
  - **Lecture:** Active learning exercises every lecture (except this one)
  - **Recitation:** Working sessions, submission on Canvas/Gradescope
- All of the above count as graded “participation activities”
  - You may miss up to 4 participation activities with no grade penalty (No need to send emails ahead-of-time)

# Textbook

- No single textbook
- Assigned readings from different sources
  - Book chapters (library)
  - News articles
  - Lecture notes
- Recommended supplementary reading: Software Engineering at Google
  - Available for free online (legally!):  
<https://abseil.io/resources/swe-book>

O'REILLY\*

## Software Engineering at Google

Lessons Learned  
from Programming  
Over Time



Curated by Titus Winters,  
Tom Manshreck & Hyrum Wright

# Gaining Experience: Central to 313!

- Case study analyses
  - Team assignments
  - Open-source engagement
  - Hands-on experience is key!!!
- 
- No “survivor”-style projects – wait till 17-413 (Capstone)

# Evaluation

- Assignments (60 %)
  - Regular homework, mostly in teams with individual component
  - Open-source engagement
- Midterm (20 %)
- Participation activities (20 %)
  - In-class exercises
  - Pre-class reading assignments
  - Recitation exercises



# Recitations

- Practical tasks, preparation for homework, extra material, discussions
- Have your GitHub account at the ready.
  - Bring your laptop!
- This week: GitHub (helpful for recitation 1)
- Teams will all go to the same recitations

# “Homework” Assignments / Projects

- P1: Setup and test a large existing software product
  - Get up-to-speed with new technologies quickly and on your own
- P2: Collaborative development on a large software project
  - Add features and follow SE process
- P3: Continuous Integration + Deployment
- P4: Develop a design doc, and deploy an ML-based microservice
- P5: Open-source Excursion
  - Open-ended project: contribute to an OSS project using everything you have learned; get kudos for having PRs merged

# Warning! Course & HW structure may be different than what you are used to...

- Lecture topics are on high-level ideas about software engineering; case studies and experiences
- Projects require applying these ideas to technical artifacts
- Projects simulate “real-world” professional SE experience
- Technical aspects of project will not be taught in class
  - Explicit learning goal: learn new tools, languages, etc. on your own
  - Ask for help when needed; recitations provide demos and resources
- Project requirements are often vague or under-specified (intentionally)
  - Feel free to ask for clarifications, but expect subjective responses
  - Focus for assessment is engagement, not absolute correctness

# Team Assignments

- Mirror realistic setting
- Assigned teams throughout the semester
  - Fill in team building survey before next lecture
- Teamwork surveys every week
- Conflict resolution process as needed
- Most team assignments have individual components



# Professionalism

- Being a professional means, you must work well with others
- The best professionals are those who make those around them better
- If you feel someone is not treating you or someone else in a professional manner, you have two options:
  - If you feel you have the standing to do so, speak up!
  - Reach out to the course staff, and we will meet with you privately to discuss it, as well as preserve your anonymity

# Final Projects

- Open-source excursion is the most fun part of the course!
- Very open-ended project. 24% of overall grade.
- Brings together everything you will have learned from lecture and prior assignments
- Teamwork and communication is very important
- In-person presentation in finals week (no exam)
- ***Do NOT book flight tickets for end-of-semester holidays until finals are scheduled.***

# Late day policy

- **Assignments:** No late days
  - Simply doesn't work with team assignments
  - Plan for unexpected delays ahead of time (not just before deadline).
- **Participation activities (lecture + recitation):** Accommodations in case of health issues, travel for interviews, university sports, etc.
  - Everyone gets 4 free absences. No need to inform us beforehand.
  - Beyond 4 absences, participation grade can get affected.
  - Inform us of extended absences (e.g., hospitalization). We can help you make up some of the lost points in such cases.
- If you have an assignment due after a trip, turn it in *before* you leave.
  - You may not have Internet where you're going.
  - Your return travel may be delayed beyond the assignment deadline!

# Academic Honesty

- Standard Collaboration Policy
- In group work, be honest about contribution of group members; do not cover for others
- Unless explicitly prohibited, you may use generative AI (e.g. ChatGPT) to help you write your prose and code. You are responsible for its correctness. Be sure to attribute the content to the service you used.
- HW1 will be done in one public repo. PLEASE reach out if you have concerns.

# For next class: survey, scheduling



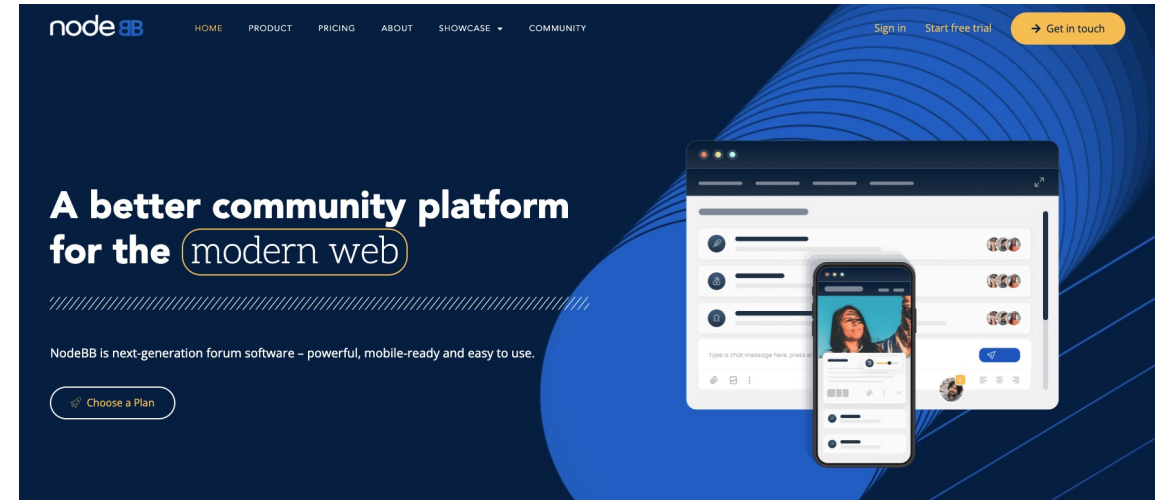


# First-week Survey due Thursday

- Form groups based on schedule availability.
  - This is ridiculously important.
  - Identify experience and working styles.
- Help us shape the course based on
  - your background knowledge
  - your interests



# Project P1 released



- P1A: Checkpoint due this Friday (Sep 1<sup>st</sup>)
  - Only 5% of total P1 points – meant to ensure you start on time
- P1B: Due Sep 7<sup>th</sup>
  - Translate a file to TypeScript (**technically challenging** for non-experts; purpose is to learn new things and engage with a large code base)