

# Introduction

17-313 Fall 2024

Foundations of Software Engineering

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

Michael Hilton and Rohan Padhye

# 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



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**



# Good News/Bad news



## 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



Hello  
my name is

<Name>

What year are you in and what is your major?

# 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,

relativ

121

121

135

92

121

92

**BOEING**

Probabilistic Consequence Gra



# REDLINE

The many human errors that brought down the Boeing 737 Max

Catastrophic Accident

9 IN (5.72 M)

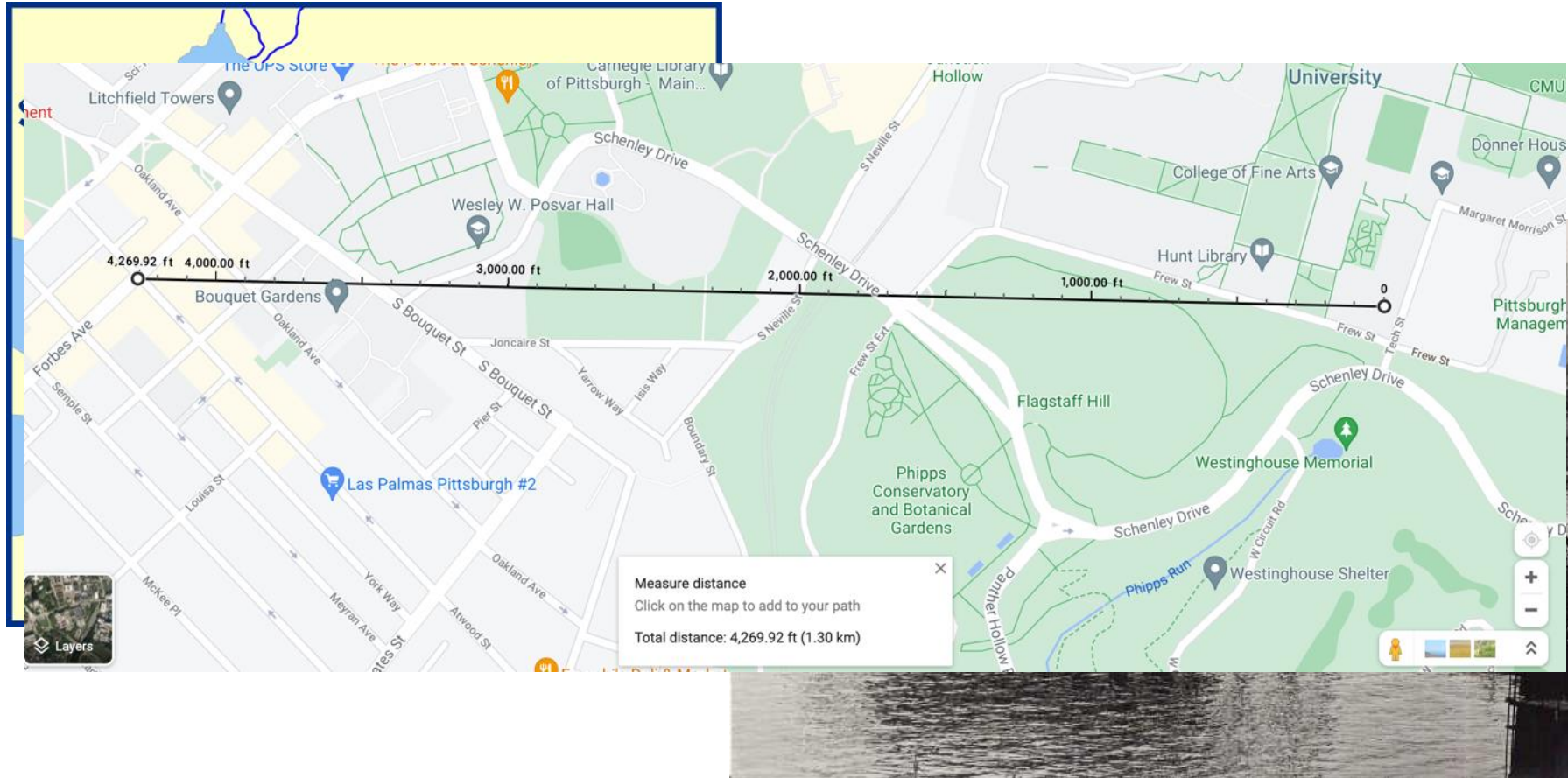
112 IN (12.55 M)

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

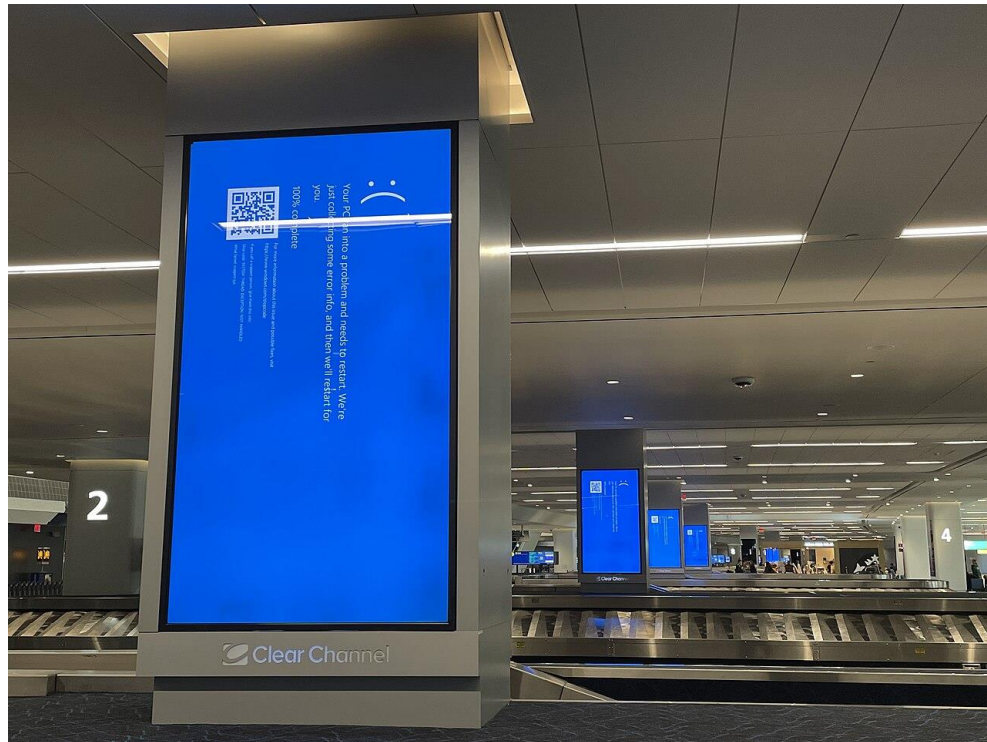
- No way to calculate stability, stiffness, or sailing characteristics

Metrics

- Failed pre-launch stability tests

QA

# The CrowdStrike Incident (2024)



LaGuardia Airport, New York City

Photo by Smishra1 - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=150535443>

The New York Times

[See more from our live coverage](#)

## Chaos and Confusion: Tech Outage Causes Disruptions Worldwide

Airlines, hospitals and people's computers were affected after CrowdStrike, a cybersecurity company, sent out a flawed software update.

[Share full article](#) [Share](#) [Bookmark](#) [Comments 951](#)



Travelers waiting to check in at the airport in Hamburg, Germany, on Friday. Bodo Marks/DPA, via Associated Press

# Complex software engineering issues involved

- Software running with kernel-level privilege can crash the system (BSOD) and prevent booting.
- Content updates more continuous than software updates

Requirements  
(Architecture &  
Design)

- Insufficient testing of content \*and\* parser

QA

- No staged roll-outs.
- No way to fix issue remotely after impact.

DevOps

- Skewed incentives

Metrics

- Limited liability

Licenses

# 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



# 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. If you DM us individually, we might both assume the other instructor will reply to you first.

# Connect with us for the class

- All links on our course website: <https://cmu-313.github.io>
- Canvas: <https://canvas.cmu.edu/courses/42497>
- Gradescope: <https://www.gradescope.com/courses/835182>
- Slack: Check your emails (will send again if you just enroll)
  - Slack vs Piazza



# 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 in almost every lecture
    - Most will be done on physical paper and turned in at the end of class.
  - **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 asking permission)

# 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 17-313!

- Case study analyses
- Team assignments
  - Teamwork exercises
- Open-source engagement
- Hands-on experience is key!!!
  - “Learn by Doing”

# Evaluation

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

# “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 integrate with an AI service.
- P5: Open-source Excursion
  - Open-ended project: contribute to an OSS project using everything you have learned; get kudos for having PRs merged

# Recitations

- Practical tasks, preparation for homework, extra material, discussions
- Have your GitHub account at the ready.
  - Bring your laptop!
- This week only: Async/offline recitation for NodeBB (to prepare for P1 and P2)
- From P2 onward: Project teams will all be in the same recitations... Good (forced) opportunity to meet in person



# 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.*** You will lose points for missing final presentations if you fly out early.

# 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 be affected.
  - Inform us of extended absences (e.g., hospitalization). We can help you make up some of the lost points in such cases, in conjunction with your advisor.
- 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.
- P1 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.
  - Participation point
- Help us shape the course based on
  - your background knowledge
  - your interests
- Posting to Canvas. We will also post on slack after inviting you all
- <https://forms.gle/moZxWujQPK8LxxdP9>



# Project P1 posted online



- P1A: Checkpoint due this Friday (Aug 30th)
  - Only 5% of total P1 points – meant to ensure you start on time
  - Only need to be able to install and run NodeBB locally
- P1B: Due next week on Thursday, Sep 5<sup>th</sup>
  - Refactor code in a single file to address code quality warnings.
  - Validate change via test coverage and manual testing
  - Expect to be **technically challenging** for non-experts; purpose is to learn new things and engage with a large code base without much hand-holding.