

# Foundations of Software Engineering

17-313

Claire Le Goues  
@clegoues  
Associate Professor

Michael Hilton  
@michaelhilton  
Assistant Teaching Professor

# Claire Le Goues

2006: B.A. Harvard College

2006—2007: Software Engineer, IBM

2009, 2013: MS, PhD, University of Virginia

2009: Microsoft Research Intern

2013—: Assistant/Associate Professor, CMU



# Christopher Meiklejohn

- Third year PhD student in Software Engineering working at the intersection of distributed systems and programming languages.
- 15+ years of industry experience before starting PhD (Basho, Mesosphere, Adobe, Comcast, UK National Health Service, and others.)
- Google Summer of Code mentor, open source maintainer of many Erlang libraries on GitHub



# Michael Hilton



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 Teaching Professor at CMU - Fall 2017



“...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.*”

Faria Sana, Tina Weston, and Nicholas J. Cepeda. 2013. Laptop multitasking hinders classroom learning for both users and nearby peers. Computing Education

# Smoking Section



**Software is  
everywhere**

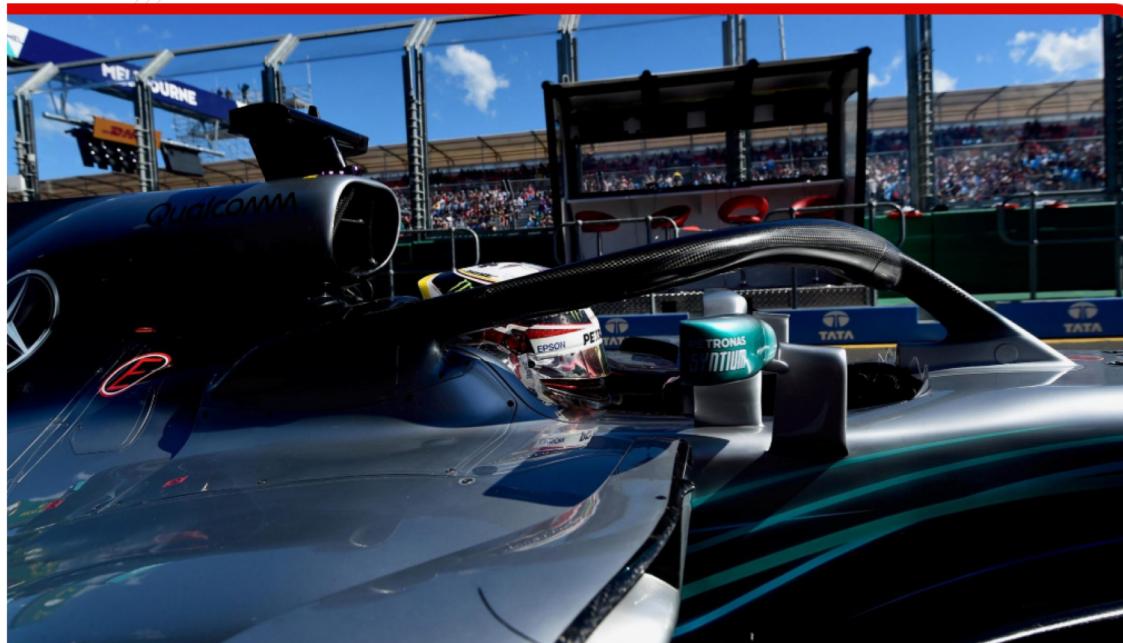
# Software glitch cost Hamilton victory - Mercedes

25 March 2018

MERCEDES

AUSTRALIA

HAMILTON



## Toyota Case: Single Bit Flip That Killed

Junko Yoshida

10/25/2013 03:35 PM EDT

During the trial, embedded systems experts who reviewed Toyota's electronic throttle source code testified that they found Toyota's source code defective, and that it contains bugs -- including bugs that can cause unintended acceleration.

"We did a few things that NASA apparently did not have time to do," Barr said. For one thing, by looking within the real-time operating system, the experts identified "unprotected critical variables." They obtained and reviewed the source code for the "sub-CPU," and they uncovered gaps and defects in the throttle fail safes."

The experts demonstrated that "the defects we found were linked to unintended acceleration through vehicle testing," Barr said. "We also obtained and reviewed the source code for the black box and found that it can record false information about the driver's actions in the final seconds before a crash."

Stack overflow and software bugs led to memory corruption, he said. And it turns out that the crux of the issue was these memory corruptions, which acted "like ricocheting bullets."

Barr also said more than half the dozens of tasks' deaths studied by the experts in their experiments "were not detected by any fail safe."

## Bookout Trial Reporting

[http://www.eetimes.com/document.asp?doc\\_id=1319903&page\\_number=1](http://www.eetimes.com/document.asp?doc_id=1319903&page_number=1)  
(excerpts)

"Task X death in combination with other task deaths"

# 8 Deaths Now Tied to E-Scooters

The reports come as a CR survey finds riders confused about where to use scooters, and some cities consider banning them

By Ryan Felton  
June 03, 2019





because the site is central to implementation of the most

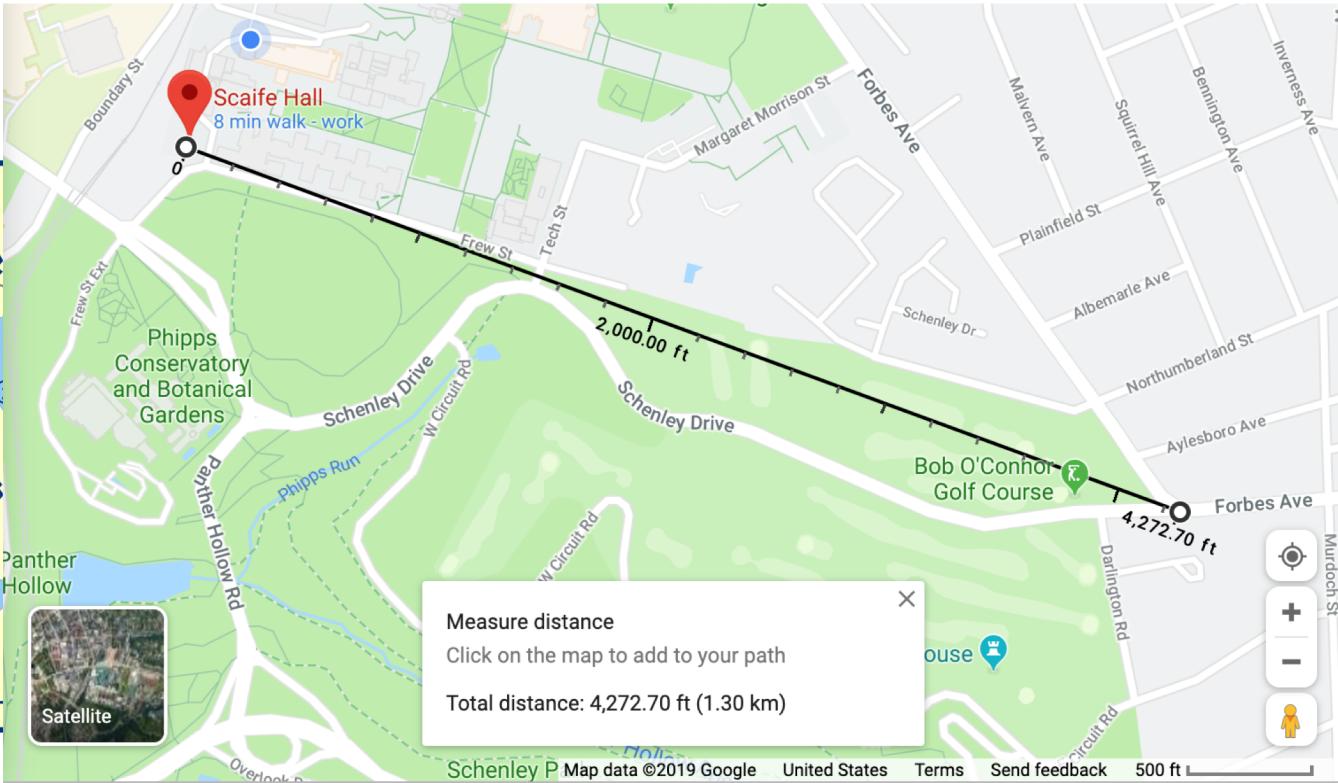
impression of the program. (Image credit: AFP/Getty Images via @ daylife)



# Vasa



# Vasa



# What happened?

- Changing shipbuilding orders
- No specifications for modified keel
- Shifting armaments requirements
- Shipwright's death
- No way to calculate stability, stiffness, or sailing characteristics
- Failed pre-launch stability tests

# What happened?

- 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





Hello  
my name is

Hello  
my name is

Name

Interesting software  
development experience?

# Syllabus and course mechanics

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

# Course Themes

- Software engineering as a human process
- Process
- Requirements
- Measurement
- Quality, incl. Security
- Time and team management
- Ethics
- Software Engineering for AI/ML
- Strategic thinking about software

# Prerequisites

- Assumes working knowledge of popular programming language(s)
- You will have the best experience if you have had an internship (ask us if you have any questions)
- vs 17-214
  - 17-313 largely focused on human issues and quality beyond functional correctness
  - 17-313 focused on larger scale

# Active Lecture

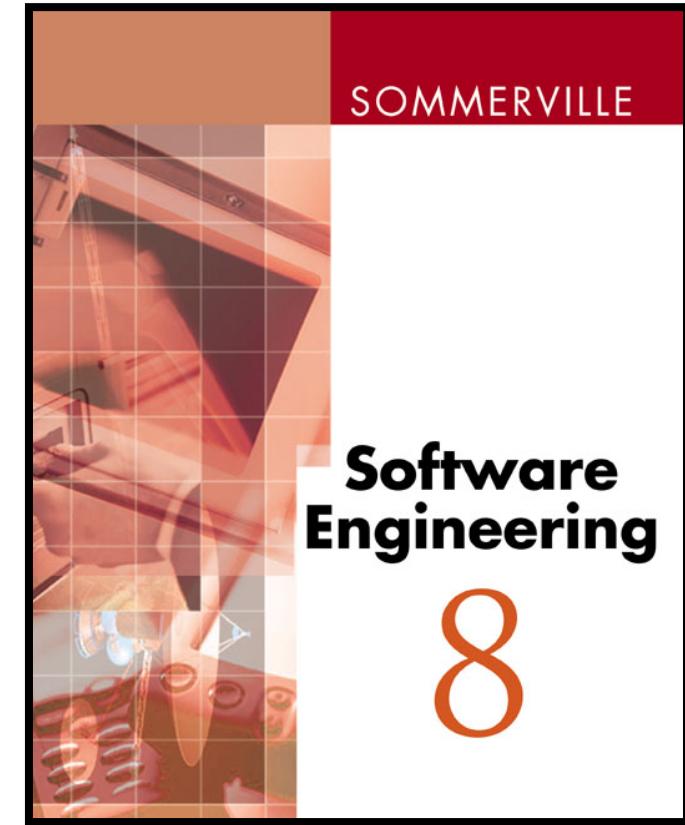
- Case study driven
- Discussion highly encouraged
- Contribute own experience
- Regular active in-class exercises
- In-class presentation
- Discussions over definitions

# Readings and Quizzes

- Reading assignments for most lectures
  - Preparing in-class discussions
  - Background material, case descriptions, possibly also podcast, video, wikipedia
  - Complement with own research
- Short and easy online quizzes on readings, due by start of lecture

# Textbook

- No single textbook
- Assigned readings from different sources
  - Book chapters (library)
  - News articles
  - Lecture notes
- Recommended supplementary reading: Sommerville, Software Engineering, edition 7 or 8
  - Aim for a used edition for <10\$



# Gaining Experience

- Case study analyses
- Team assignments
- Open source engagement
- 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 in lecture and recitation (10 %)
- Quizzes on reading assignments (10%)
- Read the learning goals!

# Participation

- Participation is important
  - Participation in in-class discussions
  - Active participation in recitations
  - Both quality and quantity are important, quality more than quantity
- Participation != Attendance

# Professionalism

- Being a professional means you should 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

# Recitations

- Practical tasks, preparation for homework, extra material, discussions
- Please bring laptop, have github account
- This week: Collaborating with Git and other tools

# Assignments

- Setup and Test an existing software product
- Come together as a team and decide on metrics
- Solicit requirements
- Develop a design doc, and implement an machine learning microservice
- Develop a plan for evaluating the quality of the software
- Contributing to an open source project of your choice

# Team Assignments

- Mirror realistic setting
- Assigned teams throughout the semester
  - Fill in team building survey before next lecture
- Peer evaluation and conflict resolution process as needed
- Most team assignments have individual components

# Late day policy

- No late days
  - (simply doesn't work with team assignments)
- Accommodations in case of health issues, travel for interviews, ... on case by case base
  - Inform us at least 2 days before deadline

# Academic Honesty

- Standard Collaboration Policy
  - University Policy on Academic Integrity
- +
- In group work, be honest about contribution of group members; do not cover for others

# Course Infrastructure

- Course website
  - schedule, slides, syllabus, office hours
- Canvas
  - homework, grades, discussions
- Git/Github for coding and collaboration
- Office hours on web page,
- open door policy
- [staff-17313@lists.andrew.cmu.edu](mailto:staff-17313@lists.andrew.cmu.edu)

# Two Sur



# Survey Goals

- Forming balanced groups
- Shaping the courses based on
  - your background knowledge
  - your interests
- Identifying experience in the room

# Reading Assignment Sep 1

- Sommerville Software Engineering, ed 7 or 8 – Chapter “Project Management”
- Complete quick quiz on Canvas before class

# Case Study 1: PeopleCars

- Scenario and question from prior final
  - Read scenario and question
  - Discuss answers with your neighbors
- 
- Keep answers until last lecture