17-313: Foundations of Software Engineering

Welcome to Wonderland

Claire Le Goues
Associate Professor

Michael Hilton
Associate 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

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
Associate Teaching Professor at CMU – Fall 2020

Christopher Meiklejohn

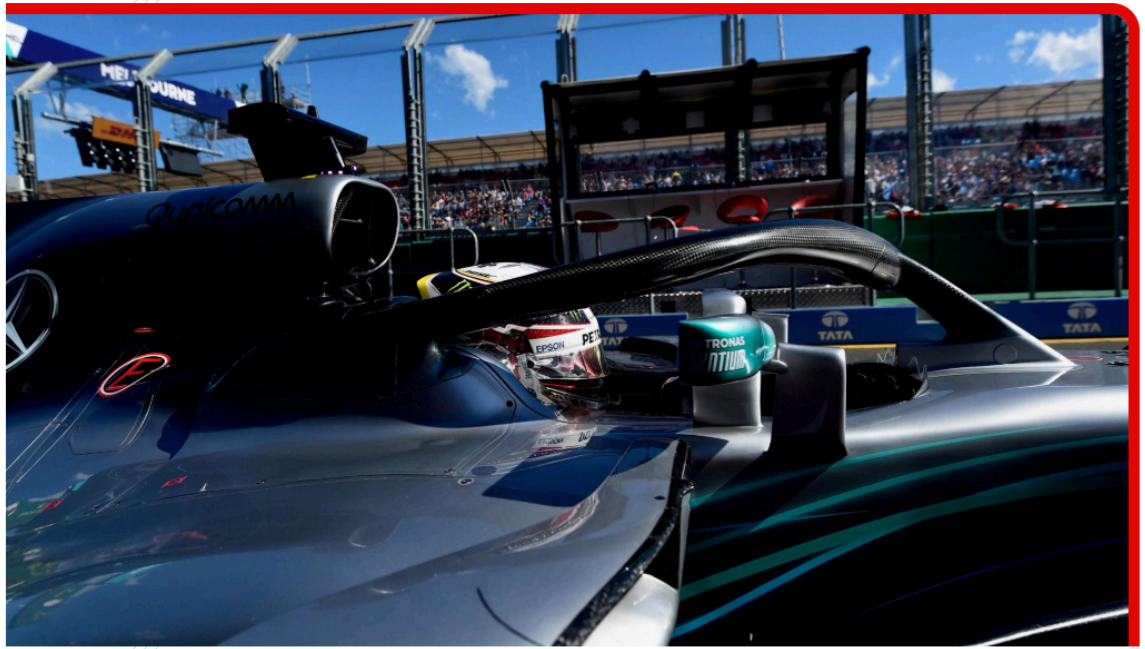
- PhD student in Software Engineering working at the intersection of distributed systems and programming languages.
- 15+ years of industry experience before starting PhD (Basho, Mesosphere, Adobe, Comcast, UK National Health Service, and others.)
- Google Summer of Code mentor, open source maintainer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

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 (excerpts)

"Task X death in combination with other task deaths"

14

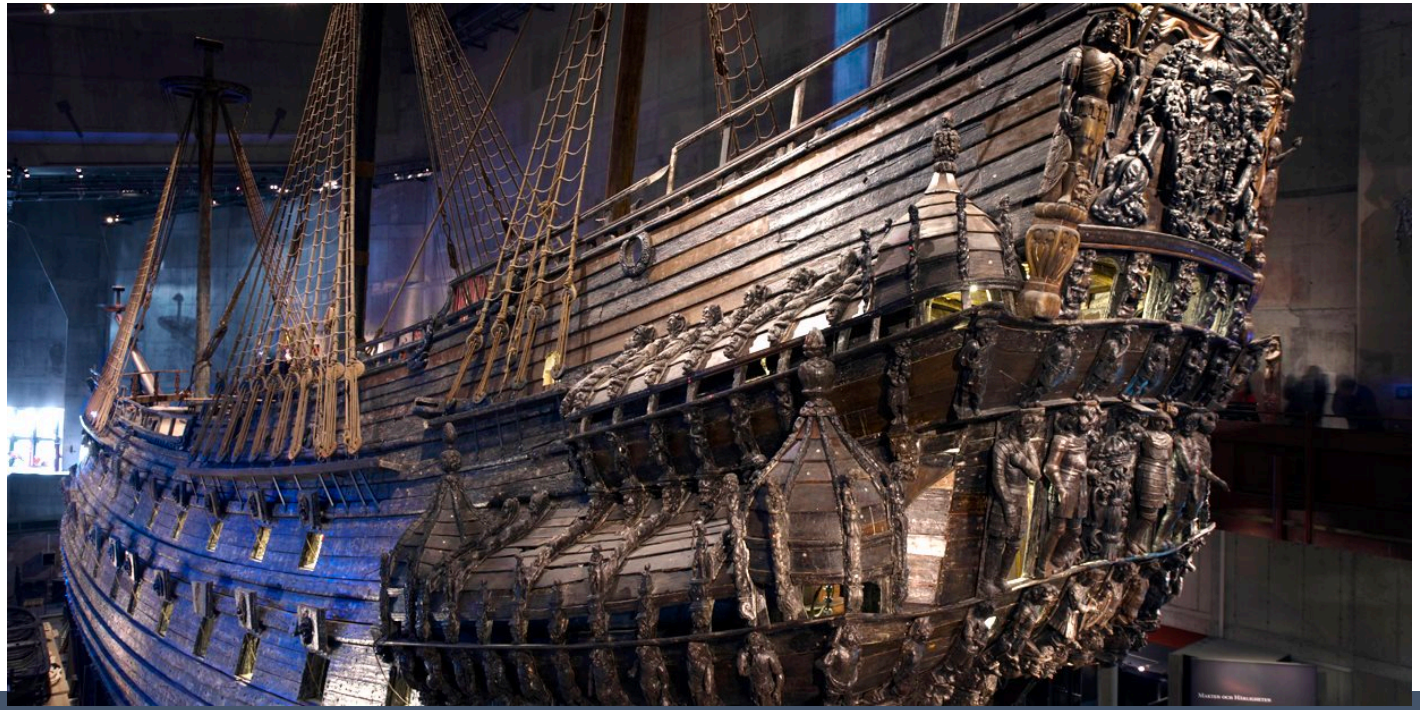
Forbes

because the site is central to implementation of the most

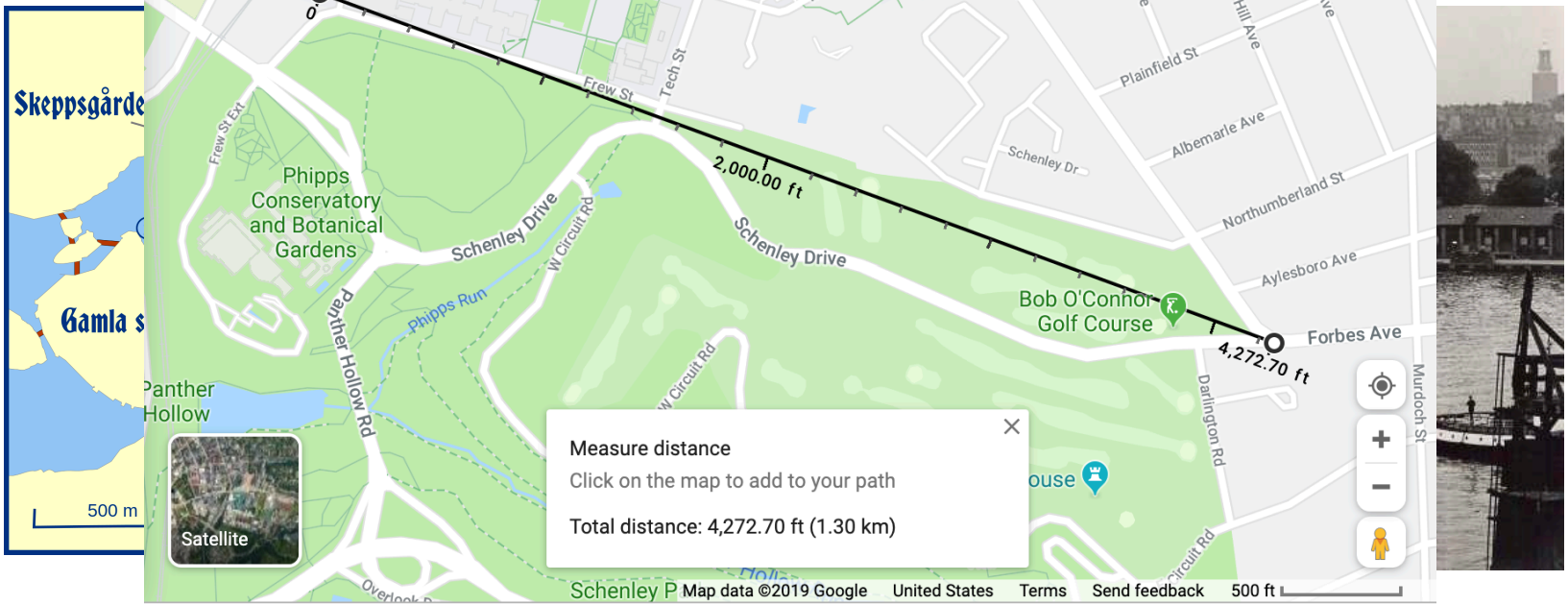
impression of the program. (Image credit: AFP/Getty Images via @ daylife)


THE VERGE
BOEING
REDLINE
The many human errors that brought down the Boeing 737 Max

Vasa



Vasa


Skeppsgårde
Gamla s
500 m
Scaife Hall
8 min walk - work
0'
Boundary St
Frew St
Frew St Ext
Tech St
Margaret Morrison St
Forbes Ave
Malvern Ave
Squirrel Hill Ave
Bennington Ave
Inverness Ave
Plainfield St
Albemarle Ave
Schenley Dr
Northumberland St
Aylesboro Ave
2,000.00 ft
Phipps Conservatory and Botanical Gardens
Schenley Drive
W Circuit Rd
Phipps Run
Panther Hollow Rd
Panther Hollow
Bob O'Connor Golf Course
4,272.70 ft
Forbes Ave
Darlington Rd
Murdoch St
Measure distance
Click on the map to add to your path
Total distance: 4,272.70 ft (1.30 km)
Satellite
Schenley P
Map data ©2019 Google
United States
Terms
Send feedback
500 ft

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”

Course infrastructure and logistics

- Infrastructure/source of truth:
  - Course website: schedule, slides, syllabus, office hours
  - Canvas (and Gradescope)
    - homework, grades, lecture videos, other material
  - Slack for communication and collaboration.
  - Git/Github for coding and collaboration
- Logistics:
  - Lecture over zoom.
  - Wednesday recitations are “hybrid” – but remote at least this week. Fridays are full-remote
  - Office Hours are over zoom.
- If you want to talk to us, DM/email all of us at once. Trust me, it’s faster.

Syllabus and course mechanics

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

https://canvas.cmu.edu/courses/19142

Sign up for Slack!!

Course infrastructure and logistics++

- More on recitation: If you are registered for a hybrid recitation but know for sure you don’t want to attend in person, please reach out!
- In-person sections will be taught mostly by Chris, maybe sometimes by Michael.
- Claire will be fully remote this semester.

Class over zoom is weird.

- My kid will probably interrupt, this time period intersects perfectly with when she wakes up from nap and comes to knock on my office door.
- On your end, one thing you can do is use this as an opportunity to *curate* your online presence:
  - Change your name on zoom to your name, not your Andrew ID.
  - Upload a profile picture.
  - Do the same on Slack.
- If it's possible to turn on your video in general, that's great!
  - But we're also realistic. Consider trying to turn on your webcam for (1) recitation, and/or (2) breakout sessions/rooms.
  - It really really helps.


Hello
my name is

Name

Previous software development experience?

Software development ambitions?

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

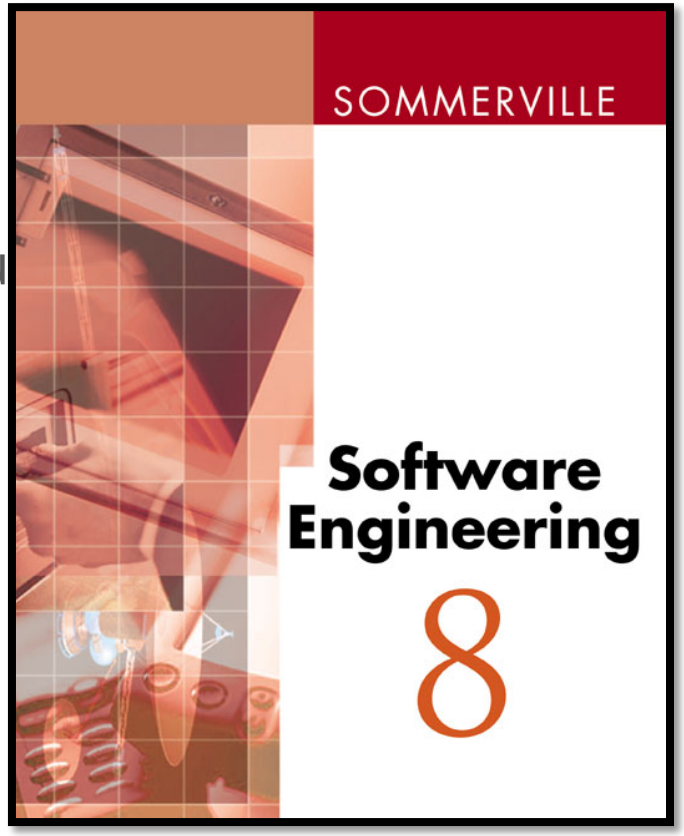
- Assumes working knowledge of popular programming language(s)
- You will have the best experience if you have had an internship (ask us if you have any questions)
- vs 17-214
  - 17-313 largely focused on human issues and quality beyond functional correctness
  - 17-313 focused on larger scale

Readings and Quizzes

- Reading assignments for most lectures
  - Preparing in-class discussions
  - Background material, case descriptions, possibly also podcast, video, Wikipedia
  - Complement with research
- Short and easy online quizzes on readings, due at noon on day of lecture.
  - Goal is to prep for lecture.
  - Uniform time for fairness with those watching asynchronously.

Textbook

- No single textbook
- Assigned readings from different sou
  - 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: Central to 313!

- 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 %)
  - Open everything, designed for an 80-minute class time but assigned for a 24-hour period, to deal with *waves hands at everything.*
  - I know you hate take-home tests but I swear, we'll take the same (type of) exam we use every year so it really will be designed to take 80 minutes.
- Class-associated activities (10 %)
- Quizzes on reading assignments (10%)

Class-associated activities

- Clearly, we can't do the same kinds of "I throw candy to inspire you to discuss actively" activities we normally can.
  - But still, discussion is highly encouraged, both verbally and in the Slack chat!
  - ***Have you signed up for Slack yet?***
- Regular in-class exercises to help with this.
  - Are you watching asynchronously? Wait a few slides, we have a plan for that.

Recitations

- Practical tasks, preparation for homework, extra material, discussions
- Have your github account at the ready.
  - If you are attending in-person, bring your laptop!
- 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 a machine learning microservice
- Develop a plan for evaluating the quality of the software
- Contribute 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

“But I’m not on Pittsburgh time!”

- Lectures will be recorded and posted.
  - We will provide instructions on how to submit the in-class exercises.
  - A lot of those exercises will be better if you can discuss with someone, however. We encourage watching with a buddy.
    - Need a buddy? DM us on Slack and we’ll help you find one.
- Recitation will *also* be recorded and posted but PLEASE PLEASE attend if you’re in, say, the continental US or a moderately related time zone.
  - If you are *not*, again, DM us on Slack, and we will work with you to develop a plan to help you get the support you need (i.e., dedicated office hours)
- We will account for time zone in assigning teams.

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

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

For Sept 3: survey, scheduling

Two ”quizzes” for Thursday: survey, scheduling

- Forming groups based on schedule availability.
  - This is ridiculously important.

- Shaping the courses based on
  - your background knowledge
  - your interests
- Identifying experience/interest.