

# Self Driving Cars, Working in Japan, and SWE!

Henry Beckstein  
EECS481, F2025

# About Me

- Graduated 2024 w/ C.S. Degree
  - German and Asian Language Minor
- Columbus, IN 
- January: Moved to Japan!!! 
- Freetime?
  - Repairing cars, hiking, road trips, hanging out with friends
- Pikachu Enthusiast 



# About Me (But like professionally)

- 2019: Research Assistant at University of Michigan
  - [Importing real-life roads into an open source driving sim](#)
- 2021: Subaru MDP
  - Building a tool to benchmark competitor cars
- 2022: EECS 481 IA (*hi!*)
  - Every semester except Fall 2023
- 2023: EECS 485 IA (🕸️💻⌨️🖨️🖱️💿📀) 
  - Fall and Winter
- 2024: Subaru of Japan Research Intern
  - Developing AI @ [Subaru's Tokyo Satellite office.](#)
- 2025: AI Research @ Subaru AI Lab

- I've also built full-size airplanes(really!) and worked retail



# About me (along the way...)

5 Years to graduate....

Almost failed: Math 216 💀, EECS 376 😔

Switched Majors: MEng ➔ C.S.

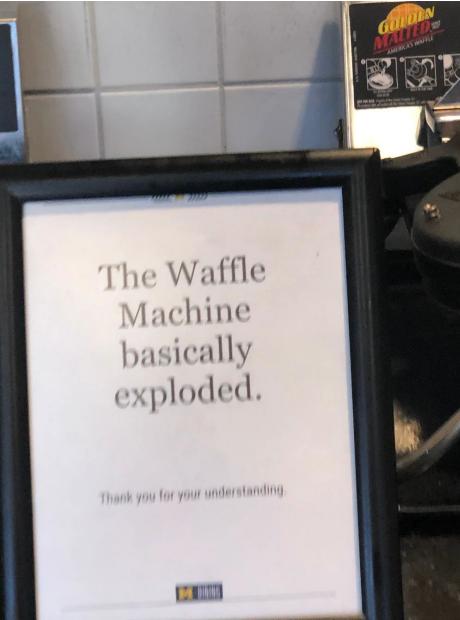
Lost father to cancer in W23

Fulbright Rejection (W24) 🎓🎓🎓

Bricked a car at work 💀💀💀💀



# Automotive Software Development



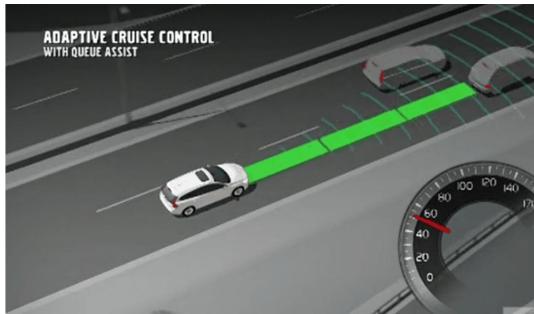
# A little bit of background...

- ‘Self Driving’ Cars aren’t exactly new...
  - ~1995: Highway capable prototype cars
  - ~2006: Automatic Emergency Braking in a production car
  - ~2008: Lane Keep Assist System
- But now they are in almost every car
  - Low cost to implement
  - Actually reduce ownership costs (Insurance)
  - Required by government regulations



# ADAS (Advanced Driver Assistance)

Adaptive Cruise Control



Car slows down

or speeds up with traffic

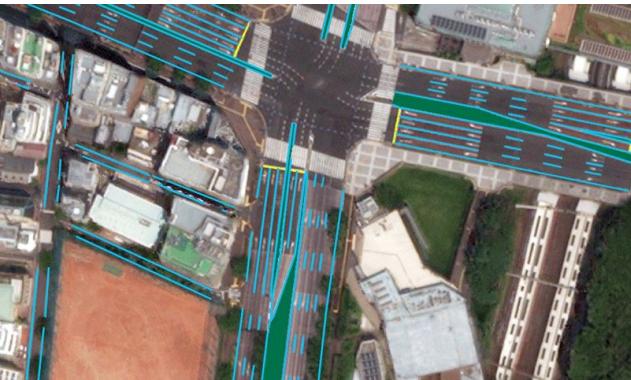
Steering Assist



Car steers in the lane

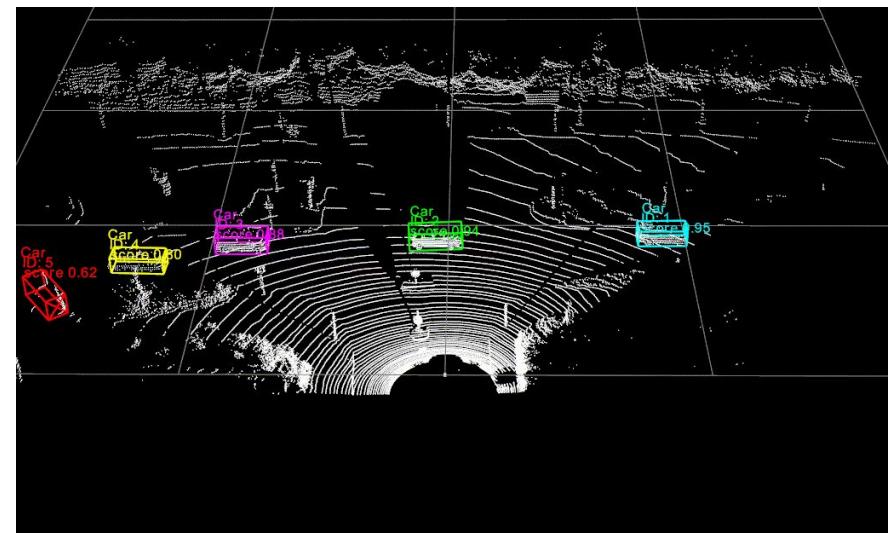
# Hardware for Self Driving

- Cameras
  - Generally, very low quality!
- Radar
- GPS
- Concept Sensors
  - LiDAR
  - Surround view radar



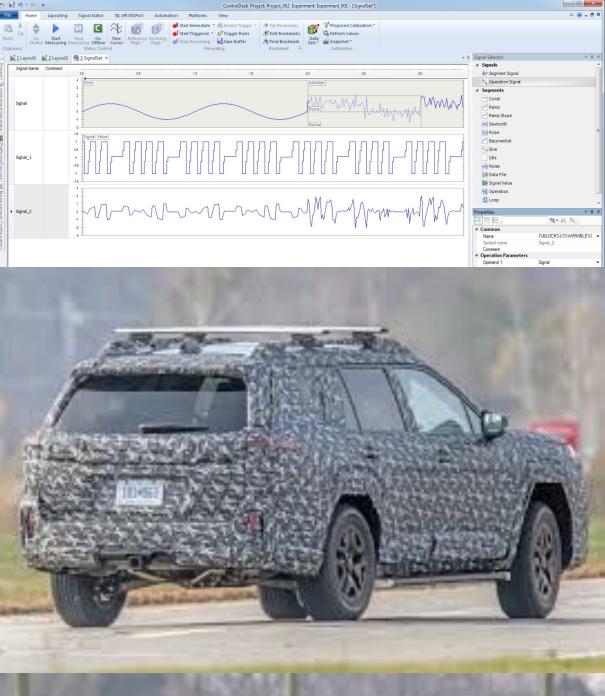
Black and White Cameras until 2013!

LiDAR is great but really expensive



# SWE in Automotive

- **EVERYTHING IS TESTING**
  - ~75% of Software Cost is maintenance and testing 
- Test for functional correctness
  - Does the car try to kill the driver? 
  - If there is a software fault, does it fail gracefully?
- Test for customer feeling 
  - Does the software 'feel good' to the driver? 
  - Is the driver able to understand what the car does?



# Requirements Elicitation and Design Decisions

Business drives **EVERYTHING**



# ✨ Branding ✨ and Legacy Costs control design decisions

- Rivian, Tesla, Lucid (New auto companies)
  - Customers expect high tech, premium products 🚗
  - Their brand is releasing futuristic software, even if it's buggy.
  - Hardware designs are all new, so the software is unconstrained but quality is hard to assure
  - Product focus is on features and capabilities
- Ford, GM, Subaru (Traditional / Legacy Automakers)
  - Customers expect reliable, low cost products.
  - Their brand is safety and dependability
  - Hardware designs are ancient, but costs are low and the hardware is proven
  - Product focus is on comfort and reliability

# How do we test?

- Driving!
  - Target routes (specific curves)
  - Duration Testing (coast to coast drives)
  - Scenario Testing (Government Regulation)
- Software Analysis
  - HILS testing (Simulate hardware inputs)
  - Automatic Software Analysis (Static or Dynamic)
- AI Testing
  - Driving Scenario Simulation
  - Hardware quality verification
  - Static Testing



**No one:**  
**Bursley Baits bus drivers at 2am:**



**Keiichi Tsuchiya**  
*Drift King & Racing Driver*

# Driving!

- Test routes focus on specific areas to tune performance
  - Test and tune at every speed (how hard does the car turn, etc)
  - These tests have the most influence over customer feel
- Duration Testing
  - Test in extreme conditions (Death Valley) to find the breaking point
  - Evaluate in real life conditions
- Most companies use professional drivers to gather data
  - Some companies (Tesla and others) gather customer data as well.
  - Conventional companies can't do this (why?)



## Reporting Safety Defects

### Contacting Tesla

For detailed information about your Model Y, go to <http://www.tesla.com> and log on to your Tesla account or sign up to get an account.

If you have any questions or concerns about your Model Y, in the United States, Canada or Puerto Rico, call 1-877-79TESLA (1-877-798-3752) and in Mexico, call 1-800-228-8145.

#### Note

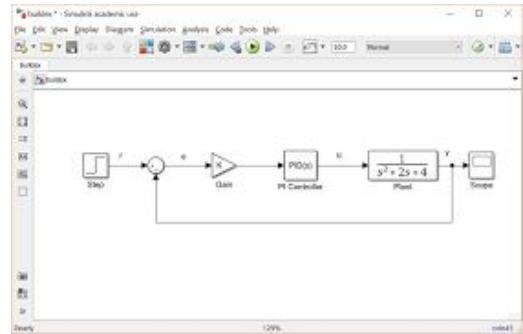
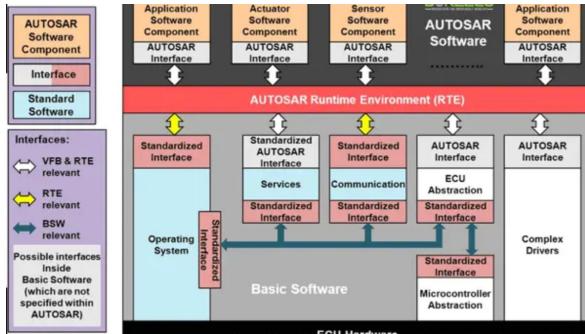
You can also use voice commands to provide feedback to Tesla. Say "Report", "Feedback", or "Bug report" followed by brief comments. Model Y takes a snapshot of its systems, including your current location, vehicle diagnostic data, and screen captures of the touchscreen. Tesla periodically reviews these notes and uses them to continue improving Model Y.

# Software Analysis

- HILS Testing is used for basic function testing
  - Bench test hardware (Power Steering, Control Module)
  - Test if supplier hardware meets specs
- Software Analysis tests for safety critical flaws
  - AUTOSAR Specification - Avoids memory leaks, etc
- Both methods fail to effectively test safety
  - HILS testing is too low level and test the entire vehicle
  - Software Analysis can test for memory leaks, but it can't verify if the program is correct

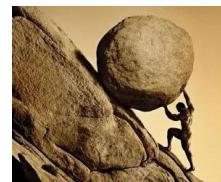


►U can eat this  
►@ BURSLEY!  
😊😊😊



# AI Development: A 5 step guide

1. Identify areas of poor performance by analyzing driving data
2. Investigate datasets, training code, and hardware for bugs
3. Triage found bugs 
  - a. Easy and High value... Fix it!
  - b. High difficulty, low value... ignore it!
  - c. High difficulty, High value... Do Something™
4. Retrain and evaluate the fix
  - a. Visual Evaluation
  - b. Driving scenarios
  - c. Static tests
5. Merge if the fix actually improves performance 



# Cross Language Projects (literally!)

Working in an almost entirely Japanese team



# Internship Experience at Subaru(2021-2023)

- When I joined Subaru, ~80% of our office was Japanese expats
  - We spent a lot of time just typing into Google Translate to explain ideas
- Most of the job was software maintenance
  - 'Fix this broken code from ~20 years ago and make it work'
- Sometimes, writing software to support research
  - 'Write code to scrape car reviews off this website'
  - 'Design software to analyze this GPS trace'
- People were really amazed at my (mid) software skills

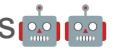


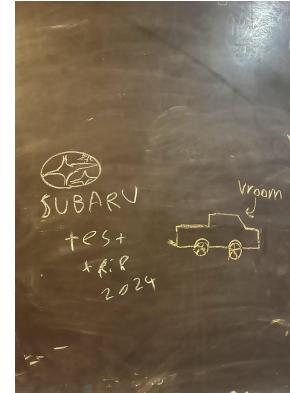
# Current Job experience at Subaru

- Run my own research projects
  - Lead a team of interns to take a far future research idea from concept to reality
- Design and Test improvements for Subaru's next generation AI
  - I'm in charge of the entire AI for a secret feature!
- Develop high definition map software for next generation Subarus
- Streamline production vehicle development 

  - Write my own apt repository to install development code onto test vehicles 

- Diagnose and repair broken embedded stuff 

  - Hand solder circuits and wiring harnesses 



# Impressions about Japanese Software

- In Japan, many developers aren't formally trained in C.S.
  - This makes developing large scale software really hard.
  - C.S. isn't paid any more then regular engineering,
    - There's little financial incentive to elect this major
- Japan's C.S. Education is really poor
  - Cutting edge there is what was cutting edge ~10 years ago.
  - A lot of the websites have security vulnerabilities.
- Japanese culture is ill suited to software development
  - Conflict is avoided, at the cost of shipping a better product
  - Ethical Hackers are often prosecuted
  - Change is VERY risky



# Impressions about Japanese Corporate life

- Almost everyone works for the same company their whole life
  - It's very hard to get hired mid-career
- You aren't incentivised to take risks
  - There's no reward for doing so, and it's socially unpopular
- Everyone must agree on the right way forward
  - Achieving consensus takes so long that opportunities are wasted.
- So many arbitrary decisions
  - **“We've already done it this way, so...”**
  - **“It's too risky to try it another way...”**
  - **“It will put too much pressure on HR”**
  - **“We need think about it...”**



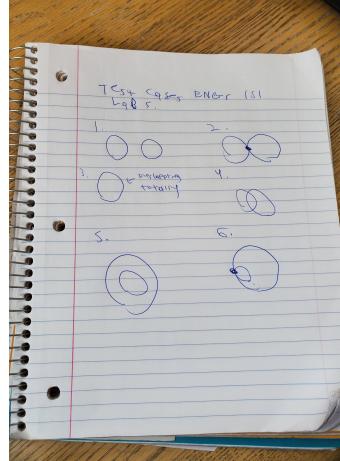
# That's not to say it's all bad...

- There's a lot of economic stability for workers. layoffs are unheard of
- Slow decision making leads to very reliable products
  - But these products don't incorporate the latest technology or match the market
- People are really nice. Everyone wants to make a better product
  - People really do want to help, everyone is just unsure of the big picture



# Working on projects in different languages

- Pictures are the best way to explain ANYTHING
  - Paint drawings and a few sentences are really great.
- We spend a lot of time double checking messages
  - There's a lot that gets lost in translation on both sides
- Comments and Documentation in Japanese
  - Occasionally we add english notes and do translations



# Classes at UM

Which ones are the most useful?



# EECS 481: Software Engineering

- I don't know if there's ever been a time I've explicitly used 481 concepts, but I explain a lot of the concepts to people new to SWE
  - "Why don't we use mocking to test this....."
  - "This design would be super hard to maintain..."
- 481 builds a lot of 'Soft Skills' that are really useful.
  - Michigan (and most schools) are really weak about teaching this.
  - Super useful on the job, and a nice flex to have 😎😎😎
- 481 is great at teaching how corporations actually work
  - It's not about writing the fastest code or the most lines.
  - It's all about talking to people and making decisions with little info..



# EECS 485, Web Development

- Even cars run on the web!!!
  - Automotive ethernet runs inside of the car
  - Remote start/unlock/updating all runs over the web
- P4: Distributed Systems is really useful
  - A car is literally a distributed system of computers
- This course develops great GIT skills 😊



# Other Advice

- Join a club!
  - The smaller clubs often have a lot of freedom
- Should I do MDP?
  - MDP is great because it oftentimes leads to a guaranteed job/internship offer
  - MDP is really painful because there's a lot of arbitrary paperwork
- Take classes that actually interest you
  - There's something to be said for just exploring, especially if you don't think it leads to a career

question @4212  

## cash

dr paolletti mentions cash a lot in lecture, what does that mean?

lecture

Edit good question 6

Updated 4 days ago by Anonymous Mous

 **S** the students' answer, where students collectively construct a single answer

Basically its when you have my kind of money, and people will say things like "wow.. look at him, he has a lot of cash"

(instructor's answer is also correct)

~ An instructor (  ) endorsed this answer ~

Edit thanks! 9

Updated 3 days ago by

 **i** the instructors' answer, where instructors collectively construct a single answer

Do you mean cache? The cache is something that holds recently used data inside the processor rather than in main memory to access them more quickly

# Thank You!

Questions??

[HenryBe@umich.edu](mailto:HenryBe@umich.edu)

<https://www.linkedin.com/in/henry-beckstein/>