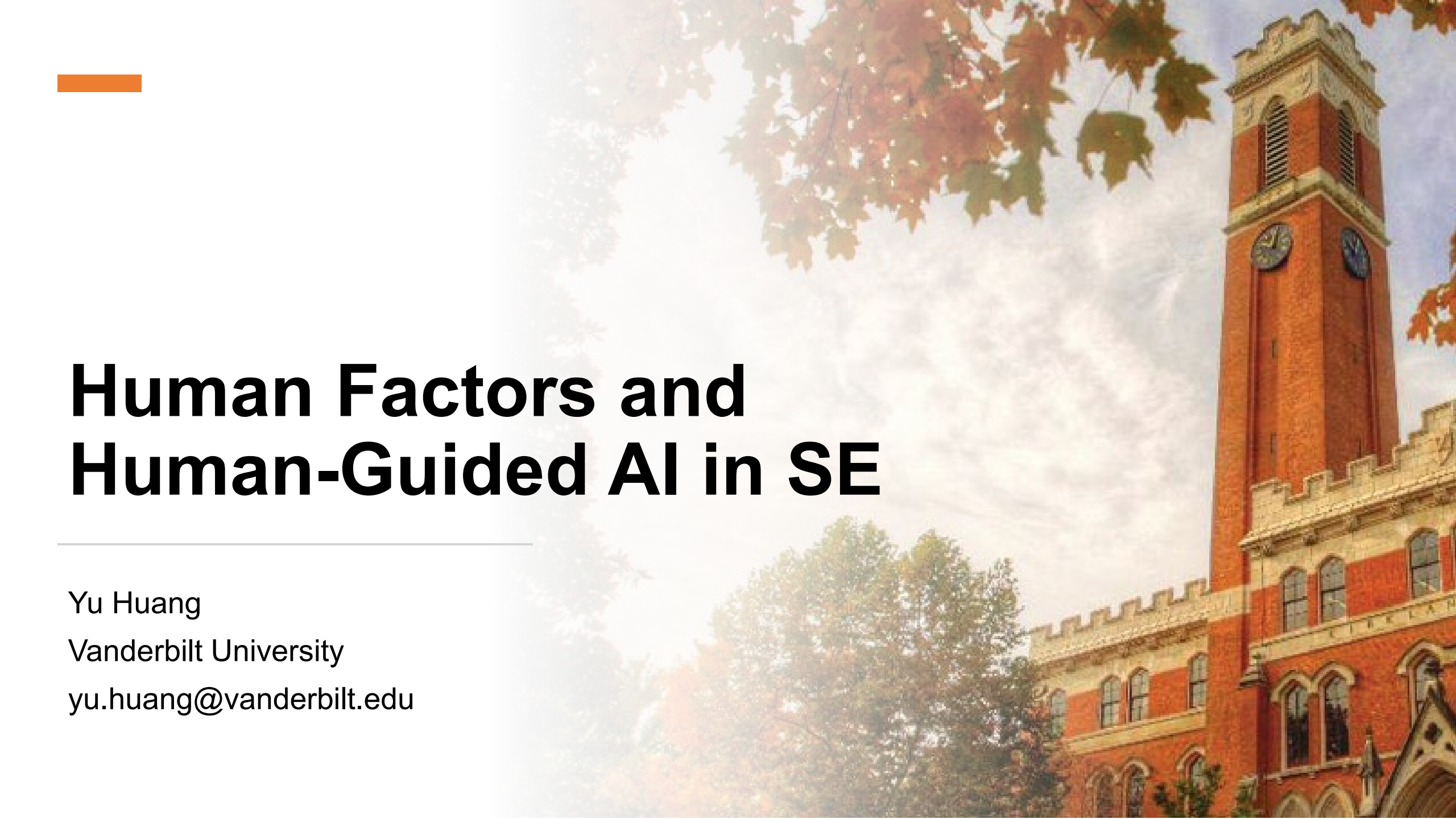
Human Factors and Human-Guided AI in SE

Yu Huang
Vanderbilt University
yu.huang@vanderbilt.edu



Final: April 18

- In-class exam
- Same format as midterm: Open book, open notes, open internet, No ChatGPT (gen AI).
- Everything included in the lectures is fair game, though we will focus a bit more on the second half of lectures
- HW6b: due on Apr 21, no extension

We want to improve productivity and reduce cost in software development and maintenance.

What is software engineering?

Programs

- Testing
- Fault localization
- Static analysis
- Dynamic analysis
- Debugging
- APR
- …

Programmers

- Will programmers use these tools? Why or why not?
- How do experts become experts?
- How to be productive?
- Biases?
- How to make a team function?
- How to estimate effort?
- …

The Human Aspect Matters

Captain Sully

Chesley (Sully) Sullenberger clarified vividly **the significance of the "human factor"** in our digital age. After saving 155 people by landing his disabled Airbus A320 in the Hudson River in January 2009, Sully became a national hero.

Sichuan Airlines Flight 8633

At the altitude of 9 km (30,000 ft; 9,000 m), the right front segment of the windshield separated from the aircraft followed by an uncontrolled decompression. The flight control unit was damaged, and the loud external noise made spoken communications impossible. Because the flight was within a mountainous region, the pilots were unable to descend to the required 8,000 ft (2,400 m) to compensate for the loss of cabin pressure. The sudden loss of pressure in the cockpit had caused multiple instruments to fail.

*The half-body of copilot was sucked out of the window and the pilot kept flown **by manual and sight**.* The three pilots were in short sleeves and suddenly it was -40°C in the cockpit. After 35 minutes, the crew made an emergency landing. 2 crew members were injured.

"Epic-level diversion".

The Human Aspect Matters

1. The Mariner 1 Spacecraft, 1962

The first entry in our rundown goes right back to the sixties.

Before the summer of love or the invention of the lava lamp, NASA launched a space mission to fly past Venus. It did not go to plan.

The Mariner 1 space probe barely made it out of Cape Canaveral before the rc course. Worried that the rocket was heading towards a crash-landing on earth destruct command and the craft was obliterated about 290 seconds after laun

2. The Morris Worm, 1988

Not all costly software errors are worn by big companies or government organizations. In fact, most costly software bugs ever was caused by a single student. A Cornell University student cre as part of an experiment, which ended up spreading like wildfire and crashing tens of thousanc computers due to a coding error.

The computers were all connected through a very early version of the internet, making the Mor essentially the first infectious computer virus. Graduate student Robert Tappan Morris charged and convicted of criminal hacking and fined $10,000, although the cost of the estimated to be as high as $10 million.

History has forgiven Morris though, with the incident now widely credited for exposing ital security. These days, Morris is a professor at MIT and the worm's sou eum piece on a floppy disc at the University of Boston.

3. Pentium FDIV Bug, 1994

The Pentium FDIV bug is a curious case of a minor problem that

Thomas Nicely, a math professor, discovered a flaw in the Pentiu response was to offer a replacement chip to anyone who could p

The original error was relatively simple, with a problem in the loc cause tiny inaccuracies in calculations, but only very rarely. In fac

4. Bitcoin Hack, Mt. Gox, 2011

Mt. Gox was the biggest bitcoin exchange in the world in the 2010s, until they were hit by a software error that ultimately proved fatal.

The glitch led to the exchange creating transactions that could never be fully redeemed, costing up to $1.5 million in lost bitcoins.

But Mt. Gox's woes didn't end there. In 2014, they lost more than 850,000 bitcoins (valued at roughly half a billion USD at the time) in a hacking incident. Around 200,000 bitcoins were recovered, but the financial loss was still overwhelming and the exchange ended up declaring bankruptcy.

5. EDS Child Support System, 2004

Back in 2004, the UK government introduced a new and complex system to manage the operations of the Child Support Agency (CSA). The contract was awarded to IT services company Electronic Data Systems (EDS). The system was called CS2, and there were problems as soon as it went live.

A leaked internal memo at the time revealed that the system was "badly designed, badly tested and badly implemented". The agency reported that CS2 "had over 1,000 reported problems, of which 400 had no known workaround", resulting in "around 3,000 IT incidents a week". The system was budgeted to cost around £450 million, but ended up costing an estimated £768 million altogether. EDS, a Texas-based contractor, also announced a $153 million loss in their subsequent financial results.

6. Heathrow Terminal 5 Opening, 2008

Imagine prepping to jet off on your eagerly-awaited vacation or important business trip, only to find that your flight is grounded or and your luggage is nowhere to be seen.

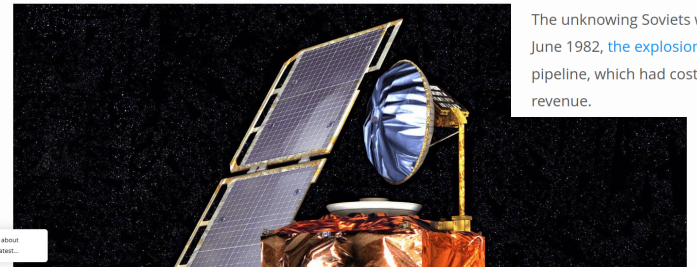
This was exactly what happened to thousands of travelers when Heathrow's Terminal 5 opened back in March 2008, and it was that performed well on malfunctioning luggage

British Airways also rev airport. Over the next 1 than £16 million.

7. NASA's Mars Climate Orbiter, 1998

Losing $20 from your wallet is probably enough to ruin your day — how would spacecraft? NASA engineers found out back in 1998 when the Mars Climate Orl too close to the surface of Mars.

It took engineers several months to work out what went wrong. It turned out tc mistake in converting imperial units to metric. According to the investigation re software produced by Lockheed Martin used imperial measurements, while the by NASA, was programmed with SI metric units. The overall cost of the failed m million.



8. Soviet Gas Pipeline Explosion, 1982

This error is a little bit different to the others, as it was deliberate (or so rumor has it). In fact, the Soviet gas pipeline explosion is alleged to be a cunning example of cyber-espionage, carried out by the CIA.

Back in 1982, at the height of the cold war tensions between the USA and USSR, the Soviet government built a gas pipeline that ran on advanced automated control software. The Soviets planned to steal from a Canadian company that specialized in this kind of programming.

According to accounts, the CIA the Canadians to place delibera Soviet pipeline.

The unknowing Soviets went ah June 1982, the explosion occurr pipeline, which had cost tens of revenue.

9. Knight's $440M in bad trades, 2012

Losing $440 million is a bad day at the office by anyone's standards. Even more so when it happens in just 30 minutes due to a software error that wipes 75% off the value of one the biggest capital groups in the world.

Knight Capital Group had invested in new trading software that was supposed to help them make a killing on the stock markets. Instead, it ended up killing their firm. Several software errors combined to send Knight on a crazy buying spree, spending more than $7 billion on 150 different stocks.

The unintended trades ended up costing the company $440 million, and Goldman Sachs had to step in to

10. ESA Ariane 5 Flight V88, 1996

Given the complexity and expense of space exploration, it's no wonde missions on our list of all-time software errors. However, the Europea even harsher cautionary tale than the rest, as it was caused by more t

Just 36 seconds after its maiden launch, the rocket engines failed due code from Ariane 4 and a conversion error from 64-bit to 16-bit data.

The failure resulted in a $370 million loss for the ESA, and a whole hos subsequent investigation, including calls for improved software analysis and evaluation.

11. The Millennium Bug, 2000

The Millennium Bug, AKA the notorious Y2K, was a massive concern in the lead-up to the year 2000. The concern was that computer systems around the world would not be able to cope with dates after December 31, 1999, due to the fact that most computers and operating systems only used two digits to represent the year, disregarding the 19 prefix for the twentieth century. Dire predictions were made about the implosion of banks, airlines, power suppliers and critical data storage. How would systems deal with the 00 digits?

The anticlimatic answer was "pretty well, actually". The millennium bug was a bit of a non-starter and didn't cause too many real-life problems, as most systems made adjustments in advance. However, the fear caused by the potential fallout throughout late 1999 cost thousands of considerable amounts of money in contingency planning and preparations, with institutions, businesses and even families expecting the worst. The USA spent vast quantities to address the issue, with some estimates putting the cost at $100 billion.

in a year

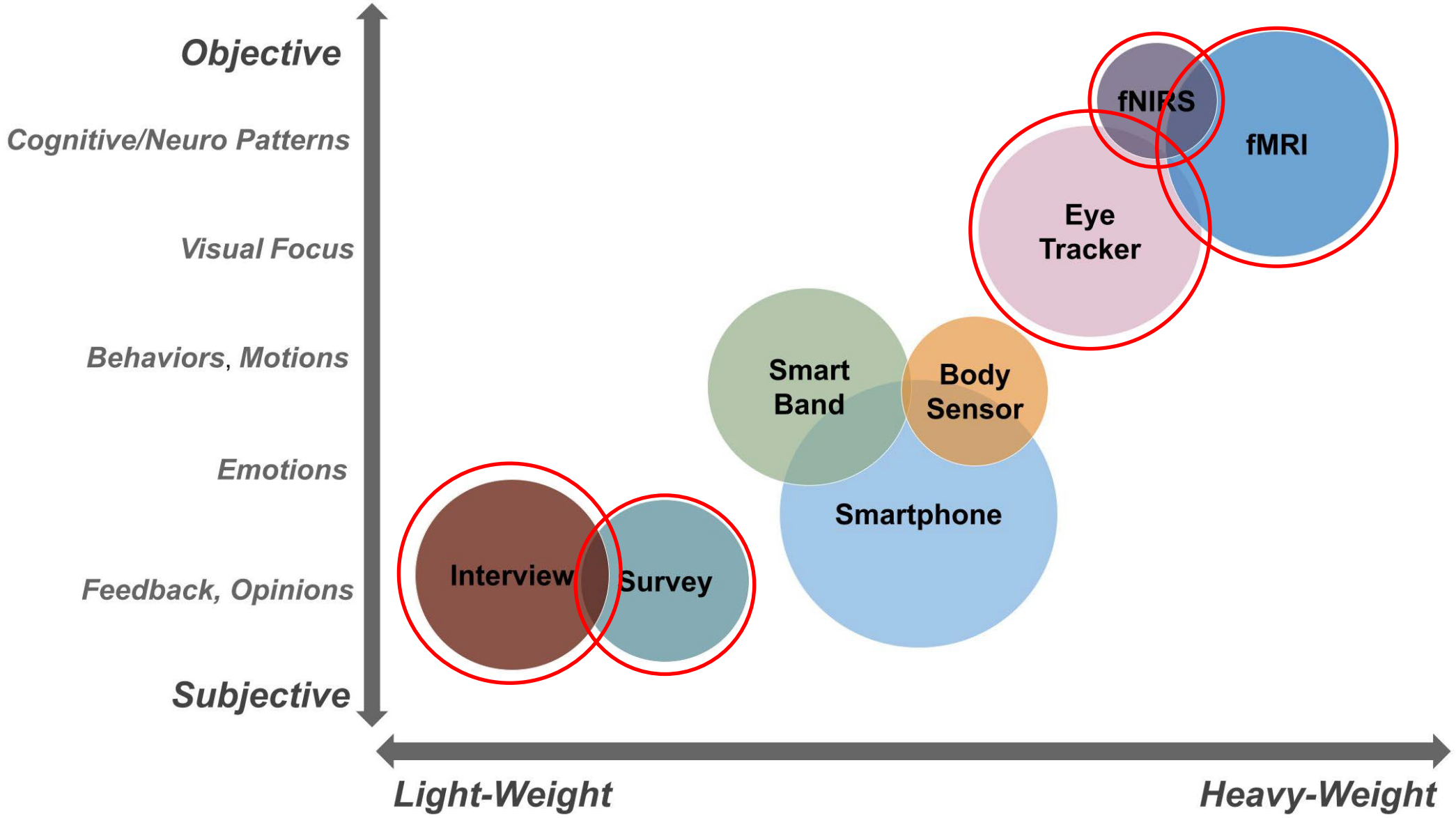
The Human Aspect Matters

- Early study of industrial developers found ***order-of-magnitude*** individual variations

Metric	Poorest	Best	Ratio
Debugging Hours Algebra	170	6	**28:1**
Debugging Hours Maze	26	1	**26:1**
CPU Seconds Algebra	3075	370	**8:1**
CPU Seconds Maze	541	50	**11:1**
Code Writing Hours Algebra	111	7	**16:1**
Code Writing Hours Maze	50	2	**25:1**
Program Size Algebra	6137	1050	**6:1**
Program Size Maze	3287	651	**5:1**
Run Time Algebra	7.9	1.6	**5:1**
Run Time Maze	8.0	0.6	**13:1**

H. Sackman, W. J. Erikson and E. E. Grant. *Exploratory Experimental Studies Comparing Online and Offline Programming Performance.* Communications of the ACM, 1968.

How to measure human aspects?



fMRI vs. fNIRS

Measure brain activities by calculating the **b**lood-**o**xygen **l**evel **d**ependent **(BOLD)** signal

- **F**unctional **M**agnetic **R**esonance **I**maging
  - ***Magnets***
  - **Strong** penetration power
  - Lying down in a magnetic tube:
    - Cannot move

- **F**unctional **N**ear-**I**nfra**R**ed **S**pectroscopy
  - ***Light***
  - **Weak** penetration power
  - Wearing a specially-designed cap:
    - More freedom of movement

fMRI vs. fNIRS

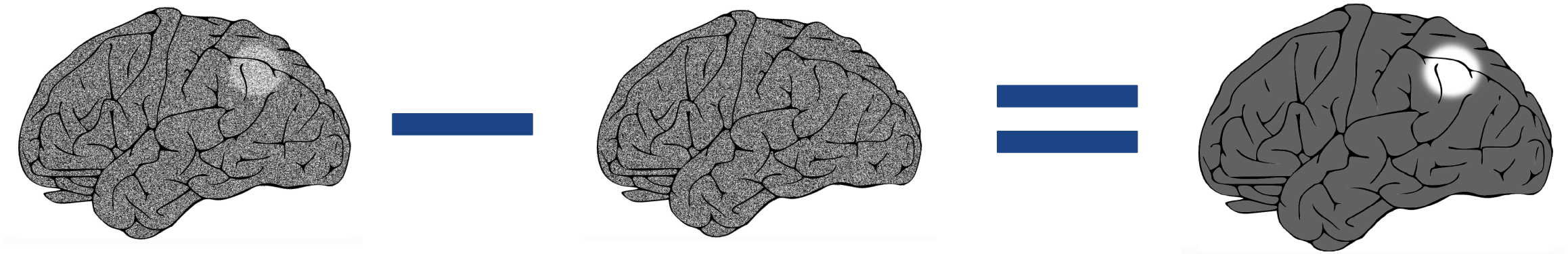
Measure brain activities by calculating the **b**lood-**o**xygen **l**evel **d**ependent **(BOLD)** signal

- **F**unctional **M**agnetic **R**esonance **I**maging
  - ***Magnets***
  - **Strong** penetration power
  - Lying down in a magnetic tube:
    - Cannot move

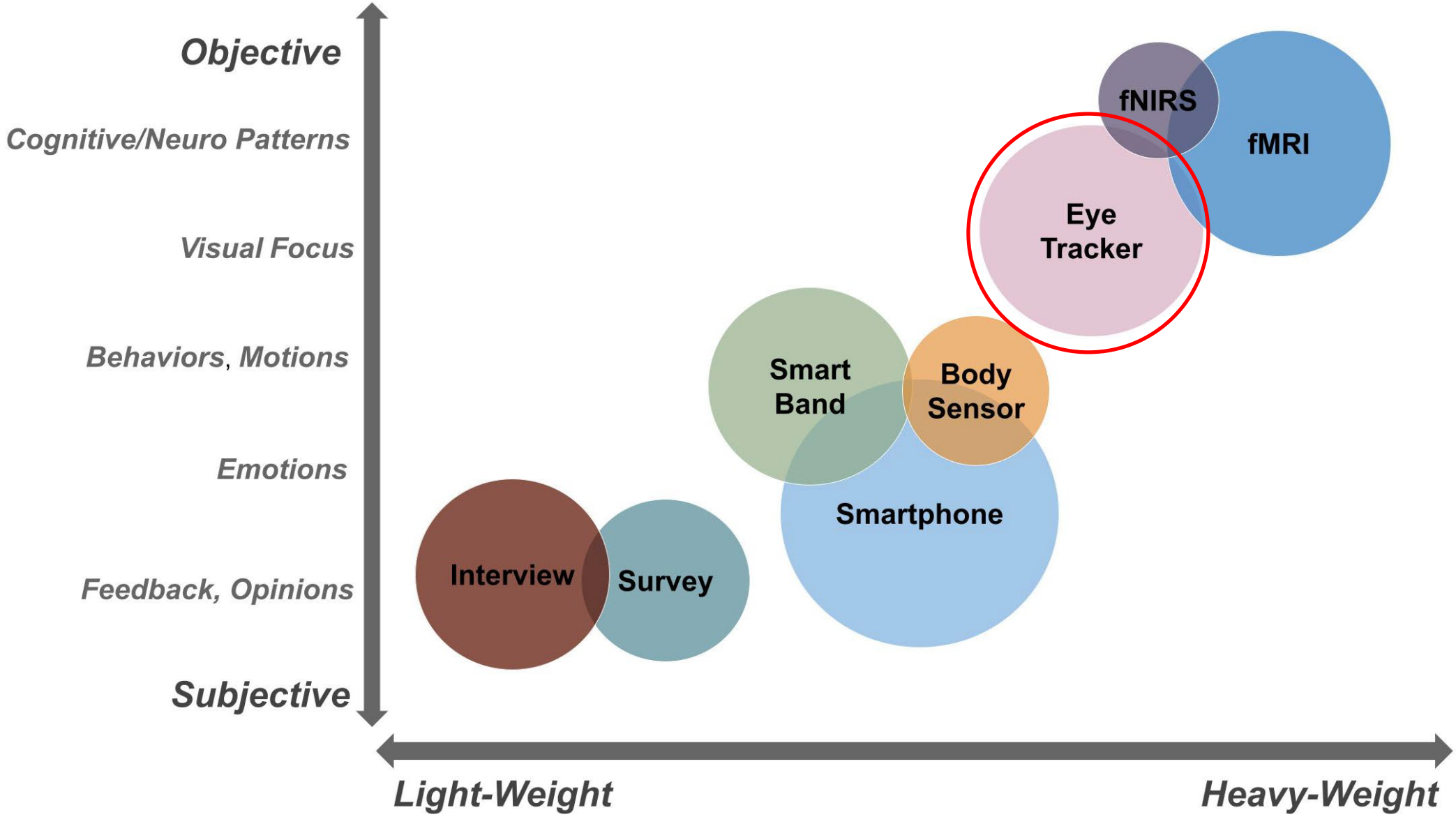
- **F**unctional **N**ear-**I**nfra**R**ed **S**pectroscopy
  - ***Light***
  - **Weak** penetration power
  - Wearing a specially-designed cap:
    - More freedom of movement

Think in Terms of Contrasts!

- Controlled experimental design
  - Task A = "**balancing trees** + nervous + …"
  - Task B = "**rotating 3D objects** + nervous + ..."
  - Contrast **A > B**: brain activations that vary between the tasks



How to measure human aspects?

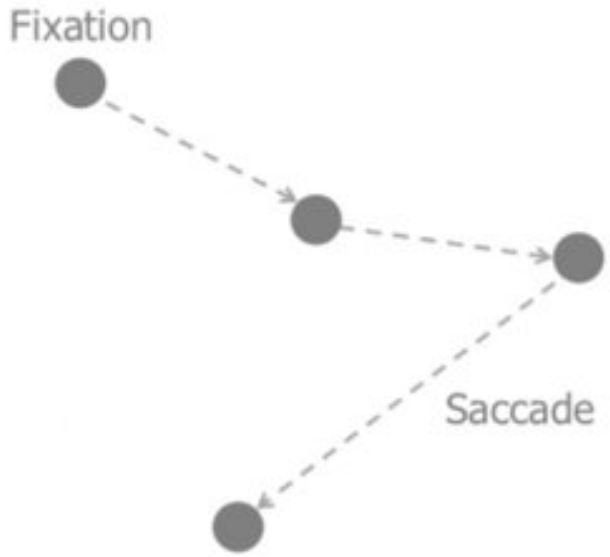


Eye-tracking

- Collect participants' visual attention by recording **eye-gaze** data: what are you looking at? How do you look at it?

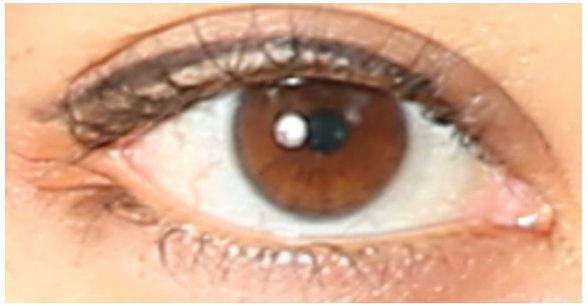
Eye-tracking: how we "look"

- Fixation: a spatially stable eye-gaze that lasts for approximately 100-300ms
  - Most of the information acquisition and processing occur during fixations
  - Only a small set of fixations is necessary to process a complex visual stimulus
- Saccade: continuous and extremely rapid eye movements, within 40-50ms, that occur between fixations
- Pupil size
  - Dilation is associated with cognitive work load

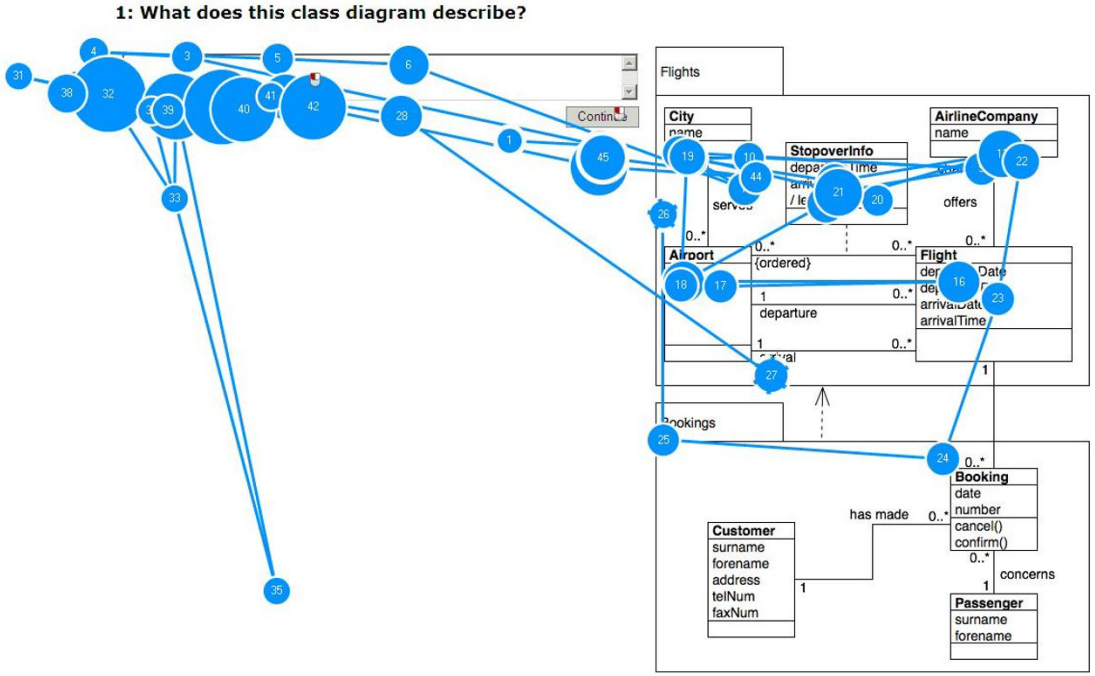


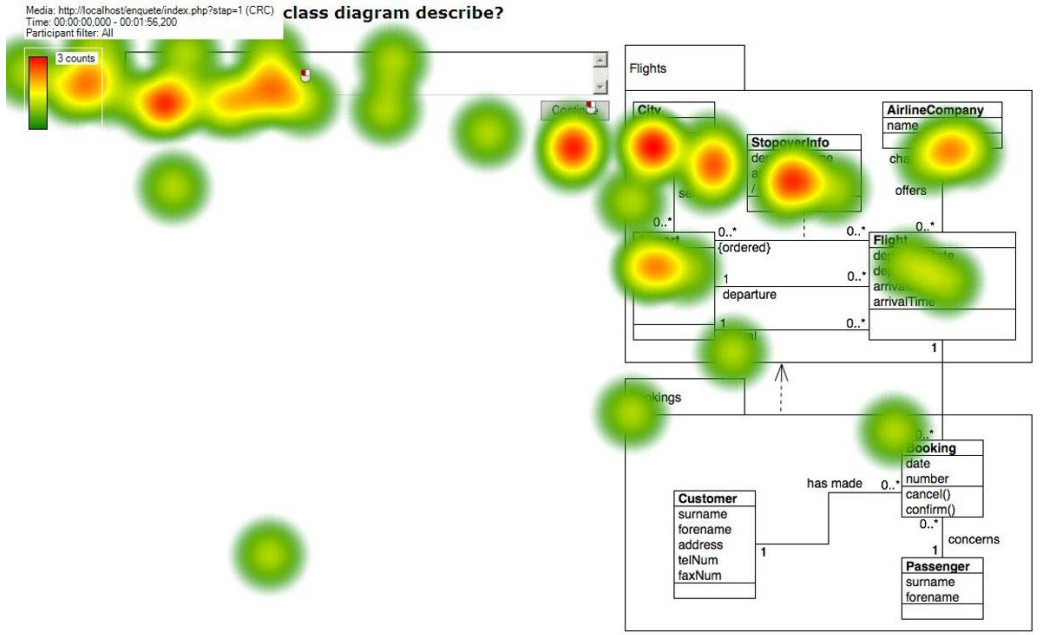
Eye-tracking: assumptions

- The immediacy assumption (Just and Carpenter, 1980):
  - The comprehension begins as soon as a participant sees a stimulus, e.g., as soon as a reader reads a word
- The eye-mind assumption:
  - The participant fixates her attention on a part of the stimulus until she understands that part



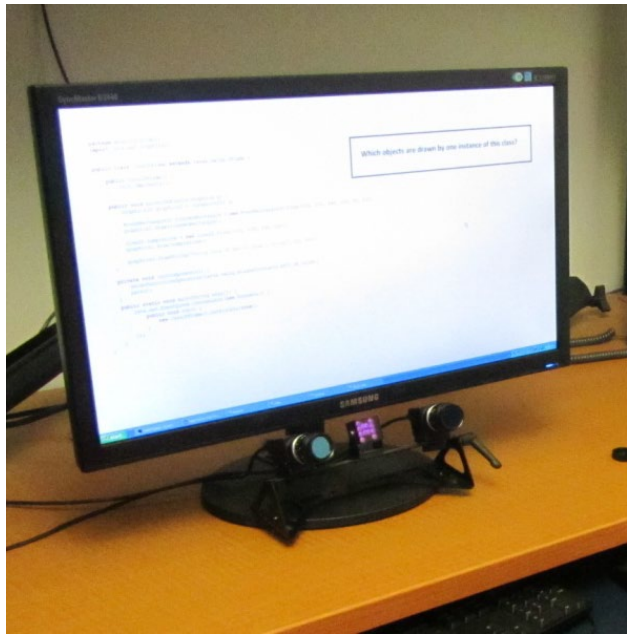
Eye-tracking: gaze plot, heat map, and raw data






```
Slide, Code, Number, x, y, Pupil X, Pupil Y, Time
Slide, Code, Number, x, y, Pupil X, Pupil Y, Start Time, End Time, Duration
C:\diagrams\1-M.png, K Space U, 3, 0, 0, 0.000000, 0.000000, 0.000, 0.000, 0.000
C:\diagrams\1-M.png, G, 1, 751, 1063, 39.000000, 38.000000, 0.297
C:\diagrams\1-M.png, G, 2, 688, 918, 39.000000, 38.000000, 0.314
C:\diagrams\1-M.png, G, 3, 688, 918, 39.000000, 38.000000, 0.331
C:\diagrams\1-M.png, G, 4, 688, 918, 39.000000, 38.000000, 0.347
C:\diagrams\1-M.png, G, 5, 684, 911, 39.000000, 38.000000, 0.364
C:\diagrams\1-M.png, G, 6, 683, 906, 39.000000, 38.000000, 0.381
C:\diagrams\1-M.png, G, 7, 683, 906, 39.000000, 38.000000, 0.397
C:\diagrams\1-M.png, G, 8, 683, 906, 39.000000, 38.000000, 0.414
C:\diagrams\1-M.png, G, 9, 681, 900, 39.000000, 38.000000, 0.431
C:\diagrams\1-M.png, G, 10, 678, 892, 38.000000, 38.000000, 0.447
```

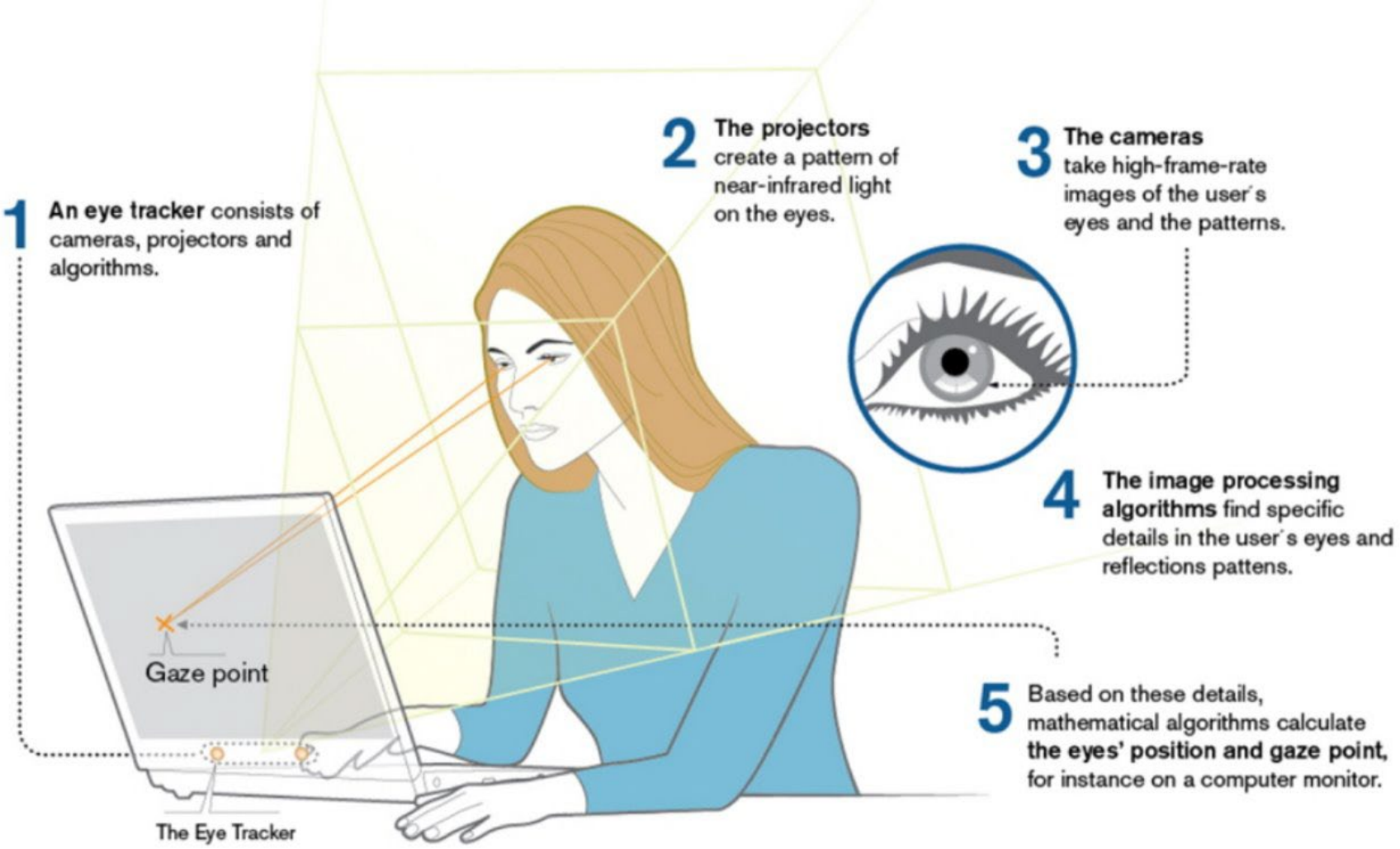
Eye-tracking: eye trackers



https://www.tobiipro.com/

https://www.tobiipro.com/

Eye-tracking: how does an eye tracker work?

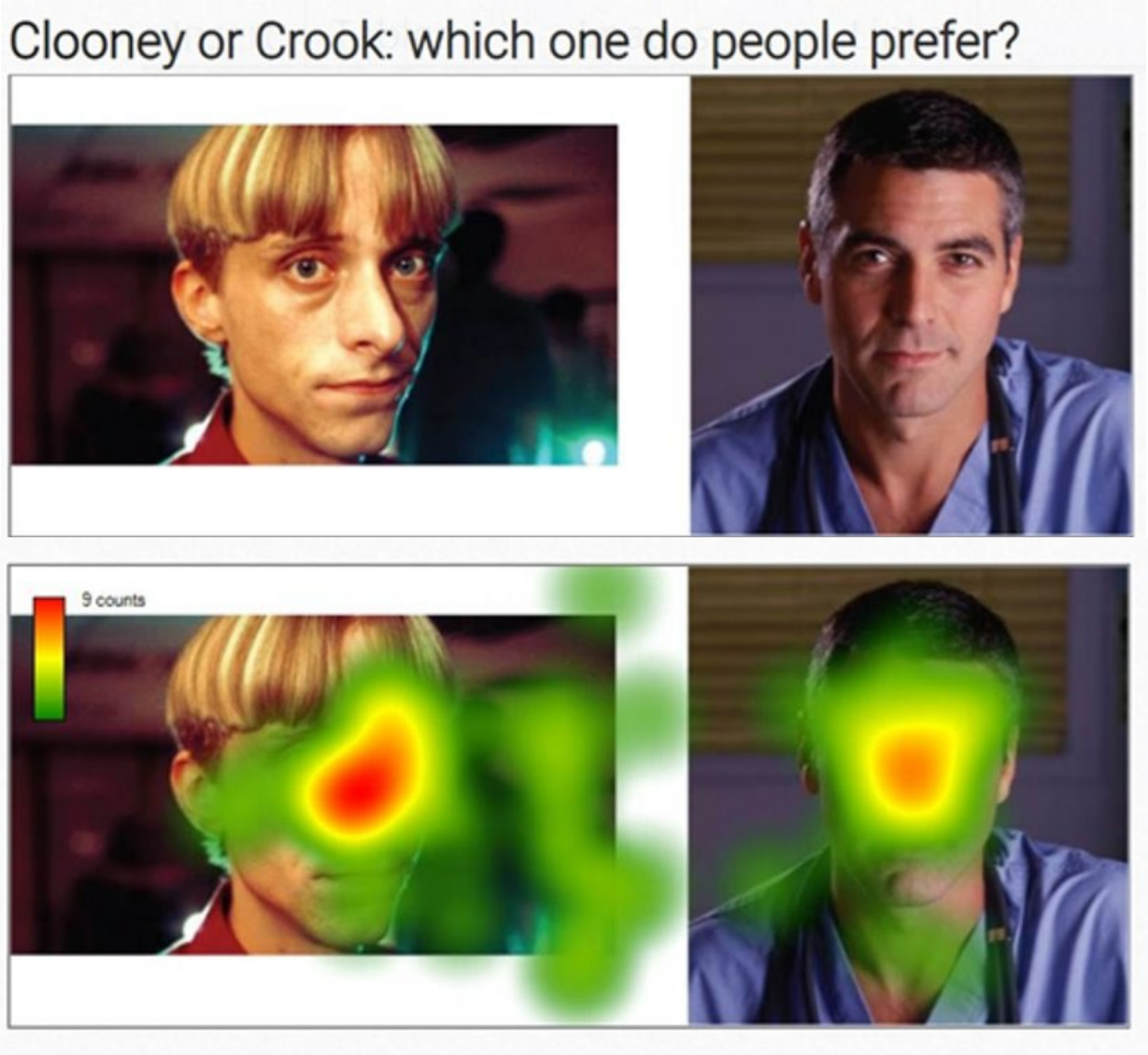


Eye-tracking: truth?

- Eye tracking allows you to know what people are thinking

Eye-tracking: truth?

- Eye tracking allows you to know what people are thinking



Eye-tracking: truth?

Misconception

~~Truth~~ about eye tracking

- Eye tracking allows you to know what people are thinking

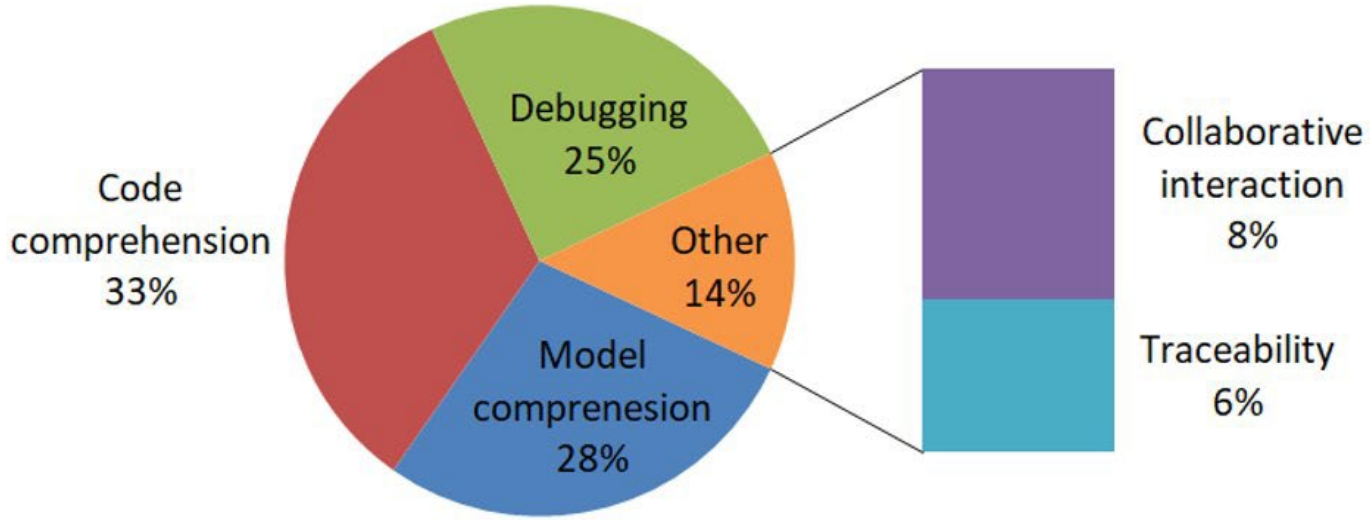
Fact: Eye tracking will give you evidence of

what people look at

Not what they **think, understand, or like**

Eye-tracking: for software engineering

Classification of SE eye tracking papers based on category (2015)



Code					Model				English text	Other
Pascal	C/C++	Java	C♯	Python	UML	ER	Tropos	BPMN		
2	3	16	1	1	7	1	1	1	2	3 applications

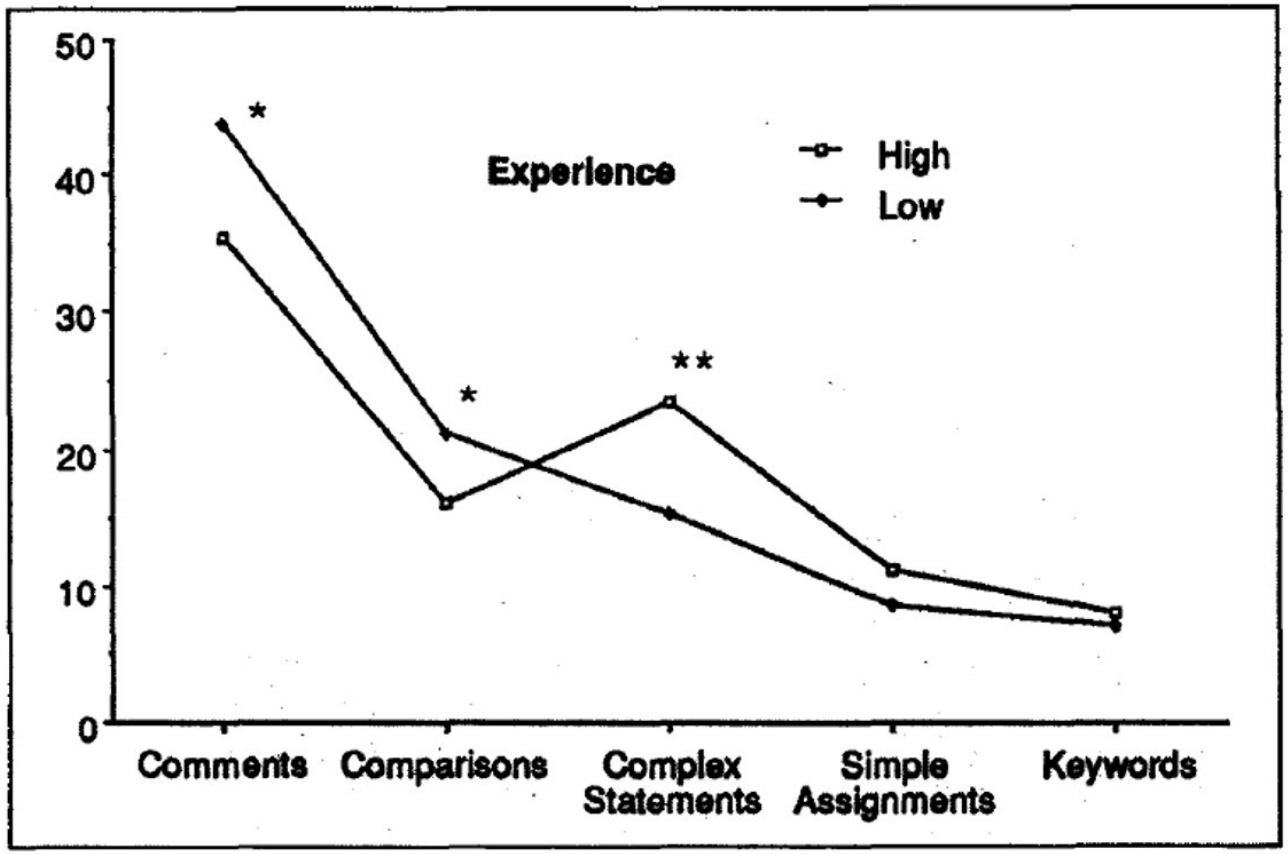
Eye-tracking: for software engineering

Types of SE questions in eye tracking experiments

Category	Type of Questions
Finding the Areas of Interest	What items or what parts of artifact (X), do participants view while performing task (Y)? Example: Does experience influence a participants focus on critical areas of the algorithm? (Crosby and Stelovsky, 1990)
Navigation Strategies	How do participants navigate through artifact/system (X) while performing task (Y)?
	Does the type of artifact (X) impact the participants' navigation strategies while they perform task (Y)?
	Do the participants' individual characteristics (Z) impact their strategies while they perform task (Y)? Example: Do the viewing patterns of experienced participants dier from those of novices?

Eye-tracking: for software engineering

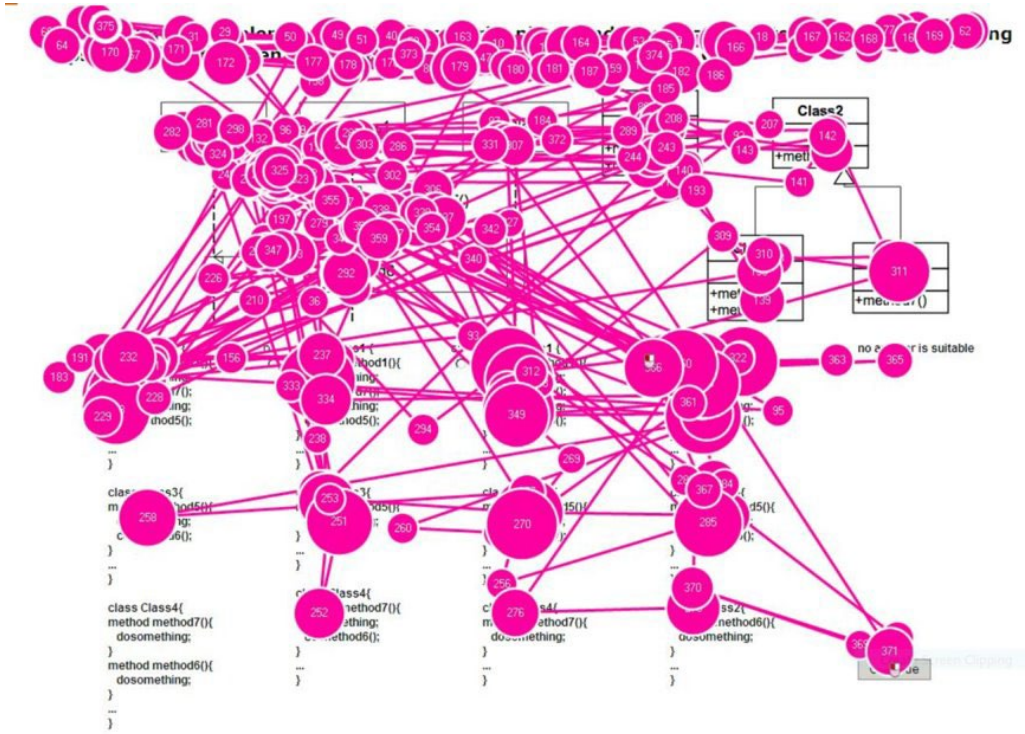
Martha Crosby 1990
Algorithm areas viewed: novices vs. experts

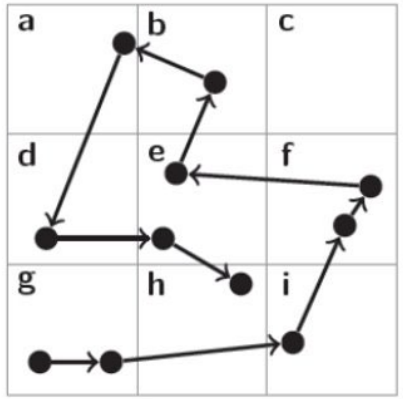


Eye-tracking: for software engineering

Scan path analysis

- A series of fixations or visited AOIs (Area of Interest) in chronological order.





Eye-tracking: for software engineering

Recent work:

- combined with other measures, e.g., medical imaging
- Investigate human biases in SE activities: e.g., gender, social info

Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines

Yu Huang
Univ. of Michigan
Ann Arbor, MI, USA
yhhy@umich.edu

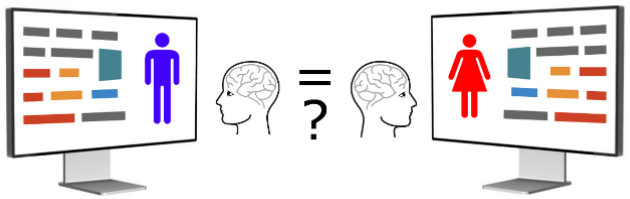
Kevin Leach
Univ. of Michigan
Ann Arbor, MI, USA
kjleach@umich.edu

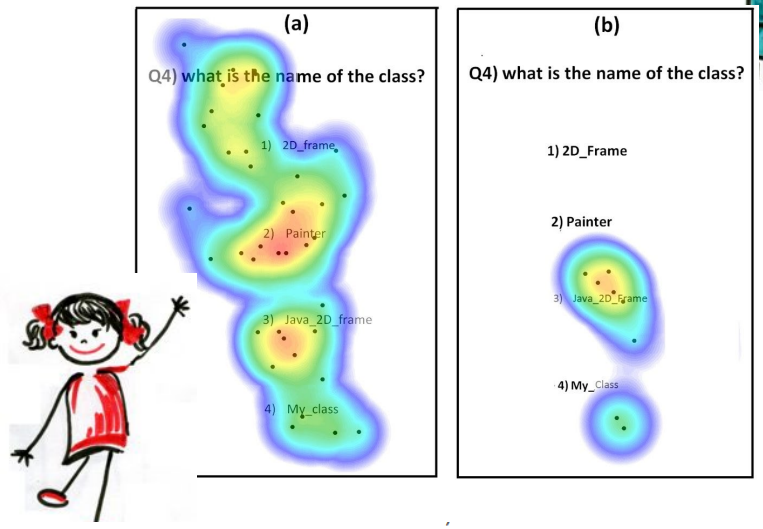
Zohreh Sharafi
Univ. of Michigan
Ann Arbor, MI, USA
zohrehsh@umich.edu

Nicholas McKay
Univ. of Michigan
Ann Arbor, MI, USA
njmckay@umich.edu

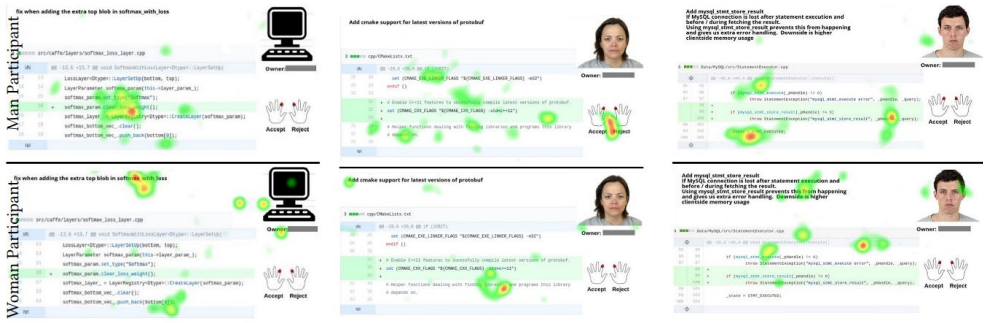
Tyler Santander
Univ. of California, Santa Barbara
Santa Barbara, CA, USA
t.santander@psych.ucsb.edu

Westley Weimer
Univ. of Michigan
Ann Arbor, MI, USA
weimerw@umich.edu





(a) Profile Page



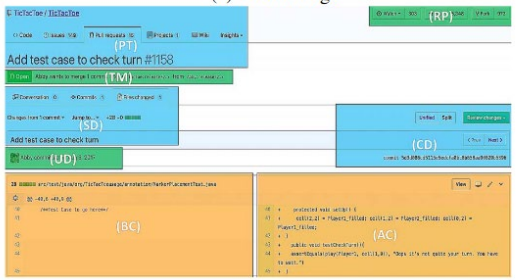
(a) A stimulus with a machine author (b) A stimulus with a woman author (c) A stimulus with a man author

Beyond the Code Itself: How Programmers *Really* Look at Pull Requests

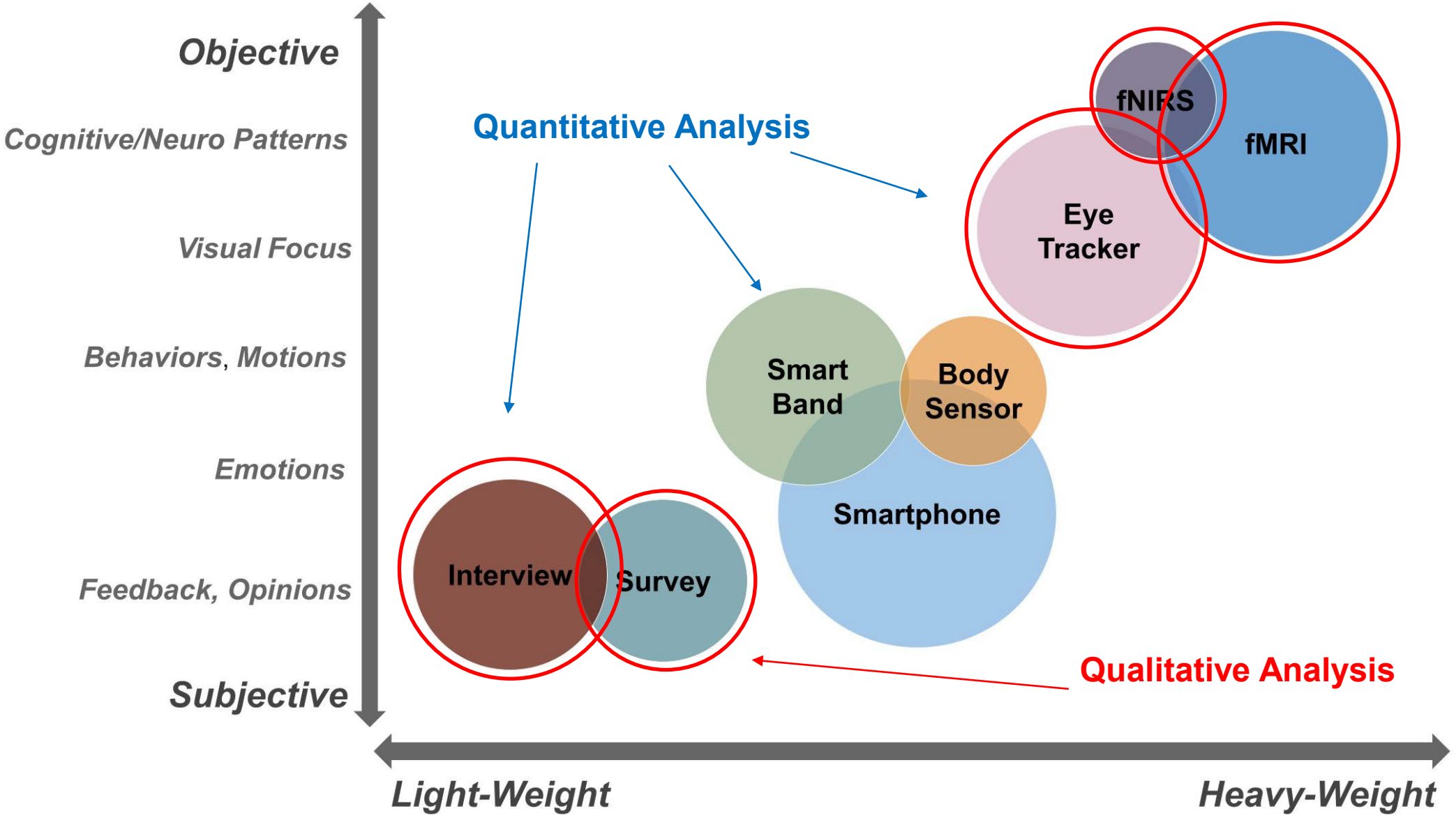
Denae Ford, Mahnaz Behroozi
North Carolina State University
Raleigh, NC, USA
{dford3, mbehroo}@ncsu.edu

Alexander Serebrenik
Eindhoven University of Technology
Eindhoven, The Netherlands
a.serebrenik@tue.nl

Chris Parnin
North Carolina State University
Raleigh, NC, USA
cjparnin@ncsu.edu



How to analyze human aspects?



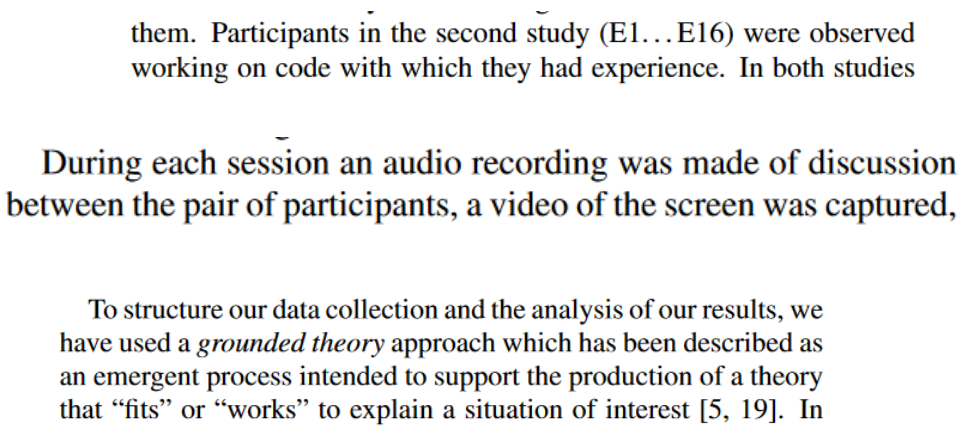
How to analyze human aspects: qualitative analysis

- Verbally-acquired data
  - Information that is gathered via speech, think-aloud protocol, oral retrospection, formal or informal interviews and surveys

With appropriate care in data gathering and analysis, verbal data *can* provide impactful insights in software engineering research.

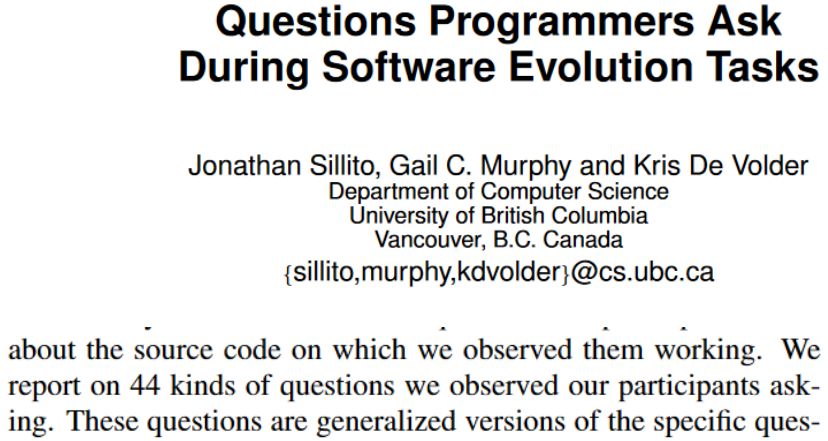
How to analyze human aspects: qualitative analysis

- Verbally-acquired data
  - Information that is gathered via speech, think-aloud protocol, oral retrospection, formal or informal interviews and surveys
- Classic example: the "Sillito et al." Questions, published in FSE ‘06, cited over 350 times


them. Participants in the second study (E1…E16) were observed working on code with which they had experience. In both studies

During each session an audio recording was made of discussion between the pair of participants, a video of the screen was captured,

To structure our data collection and the analysis of our results, we have used a *grounded theory* approach which has been described as an emergent process intended to support the production of a theory that “fits” or “works” to explain a situation of interest [5, 19]. In


Questions Programmers Ask During Software Evolution Tasks

Jonathan Sillito, Gail C. Murphy and Kris De Volder
Department of Computer Science
University of British Columbia
Vancouver, B.C. Canada
{sillito,murphy,kdvolder}@cs.ubc.ca

about the source code on which we observed them working. We report on 44 kinds of questions we observed our participants asking. These questions are generalized versions of the specific ques-

Results are useful directly (a structured answer to a fundamental question) and also as artifacts (re-used by later projects as indicative developer queries)

[Sillito, Murph, De Volder. Questions programmers ask during software evolution tasks. FSE 2006.]

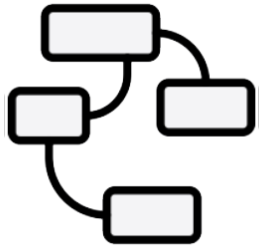
Qualitative Analysis: Metrics

- Establishing **validity** in qualitative research
  - Using multiple validity procedures
    - Member checking
    - Clarify bias
    - Spend prolonged time in the field
  - Using qualitative reliability
    - Document your procedures (scripts, codebook, etc.)
    - No drift in the definition of codes
    - Cross-check codes developed by different researchers

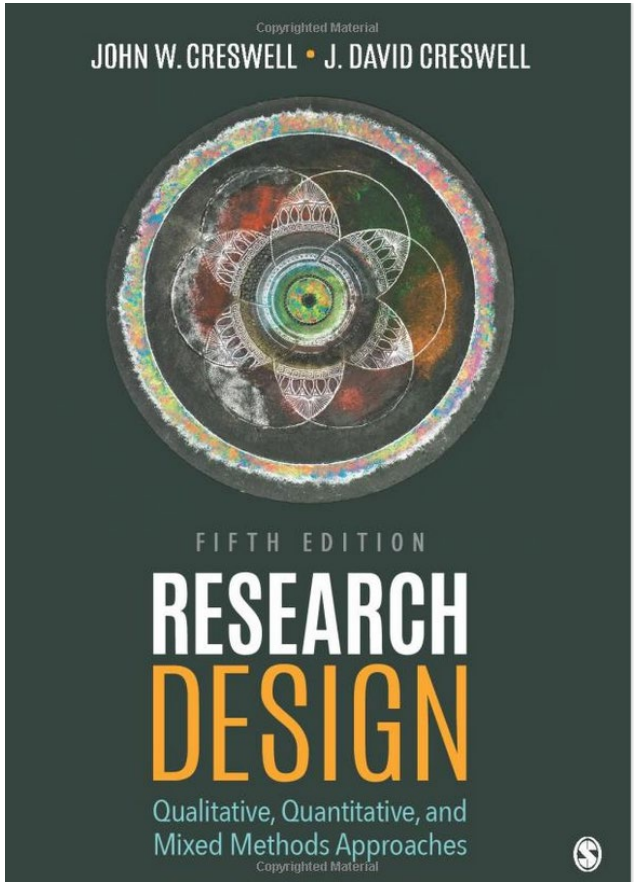
Showing Prompts

Audio Recording

Transcribing



Qualitative analysis



[Chi. Quantifying Qualitative Analyses of Verbal Data: A Practical Guide. J. Learning Sciences 1997.]

Qualitative Analysis: Useful Techniques

- Grounded theory in SE
- Similar to socio-technical studies, qualitative research can have a lot of variance
  - How can we mitigate that variance?
- Grounded Theory is a systematic methodology for qualitative research for constructing hypotheses via inductive (not deductive) reasoning
  - Method
    - Empirical/evidence based
  - Outcome
    - Key patterns of the data
    - Relationships between patterns

"It is not in your mind; it is in your data."

[Hoda. Socio-Technical Grounded Theory for Software Engineering. IEEE Trans. Software Engineering 2021.]

Qualitative Analysis: Useful Techniques

- Grounded theory in SE
- Inductive Thematic Analysis
  - Thematic exploration
    - Codes and the relationships
    - E.g. Tesch's Eight-Step Coding Process
  - Evaluation metrics
    - Saturation
    - Agreement
- Inter Rater Reliability (IRR) or Inter Rater Agreement (IRA)
  - Statistics as evidence
    - Cohen's kappa, Fleiss' kappa, etc.

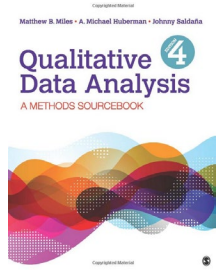
"It is not in your mind; it is in your data."

Leaving My Fingerprints: Motivations and Challenges of Contributing to OSS for Social Good

Yu Huang
University of Michigan
Ann Arbor, MI
yhhy@umich.edu

Denae Ford
Microsoft Research
Redmond, WA USA
denae@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, WA USA
tzimmer@microsoft.com



Category	Code	Description
motivation	motivation-helpuser	help end users
	motivation-helpdev	help developers
	motivation-longterm	how to keep yourself engaged in the project for a long time
	motivation-giveback	altruism
	motivation-impact	want to make impact
	motivation-better-programmer	want to look good in the community, improving skills, build up portofolio
	mitivation-hobby	I feel happy/fun, e.g., as a hobby.
	motivation-work	This is my job, or school projects, etc

Codebook Example

[Hoda. Socio-Technical Grounded Theory for Software Engineering. IEEE Trans. Software Engineering 2021.]

Qualitative Analysis: Combining Verbal and Nonverbal Data

- Strength of verbal data
  - Richness and holism
  - Discovery
    - New ideas, hypothesis
- Weakness of verbal data
  - Hard to evaluate the analysis (i.e., no “equations”)
  - Human biases
- Combining verbal and nonverbal data makes a strong and interesting case
  - Supplement, validate, or illuminate each other
  - Contrast: surprising knowledge!

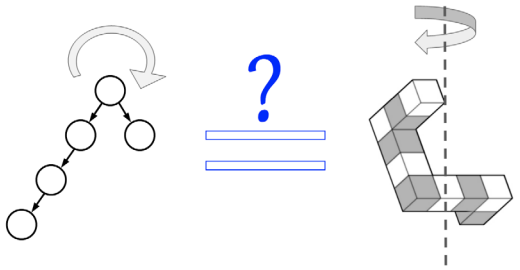
Qualitative Analysis: Combining Verbal and Nonverbal Data

- What do we learn from nonverbal data (medical imaging)?
  - Data structure manipulations do use the **same** parts of the brain as rotating 3D objects
- Nonverbal data can be powerful!
  - You cannot just ask humans: “what do your brain patterns look like?”
- What do we learn from verbal data (audio / interviews)?
  - 70% of participants report **no similarity** between data structure manipulation and 3D object rotation

Distilling Neural Representations of Data Structure Manipulation using fMRI and fNIRS

Yu Huang[1], Xinyu Liu[1], Ryan Krueger[1], Tyler Santander[2], Xiaosu Hu[1], Kevin Leach[1] and Westley Weimer[1]

[1]{yhhy, xinyuliu, ryankrue, xiaosuhu, kjleach, weimerw}@umich.edu, University of Michigan
[2]t.santander@psych.ucsb.edu, Univeristy of California, Santa Barbara



Qualitative Analysis: Combining Verbal and Nonverbal Data

- What do you think about pull requests generated by machines
  - "Machine generated code is worse on readability!"

  But all pull requests were written by humans! (We deceived you!)
- Do you think women and men write pull request differently
  - "There is no difference between pull requests written by men and women"

But there *is* a significant difference on your behavior! Both response time and final decisions are affected!

Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines

Yu Huang
Univ. of Michigan
Ann Arbor, MI, USA
yhhy@umich.edu

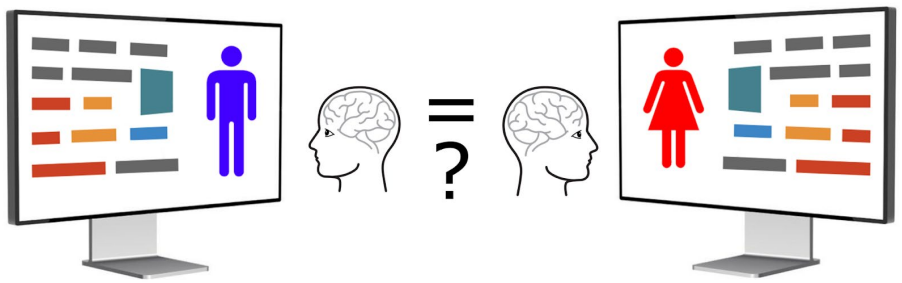
Kevin Leach
Univ. of Michigan
Ann Arbor, MI, USA
kjleach@umich.edu

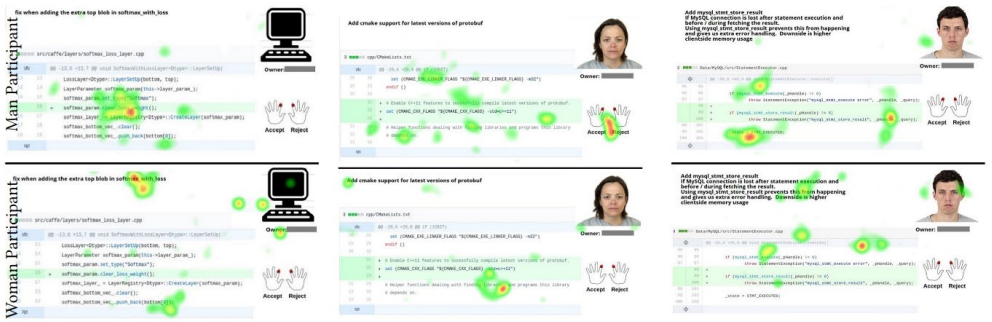
Zohreh Sharafi
Univ. of Michigan
Ann Arbor, MI, USA
zohrehsh@umich.edu

Nicholas McKay
Univ. of Michigan
Ann Arbor, MI, USA
njmckay@umich.edu

Tyler Santander
Univ. of California, Santa Barbara
Santa Barbara, CA, USA
t.santander@psych.ucsb.edu

Westley Weimer
Univ. of Michigan
Ann Arbor, MI, USA
weimerw@umich.edu





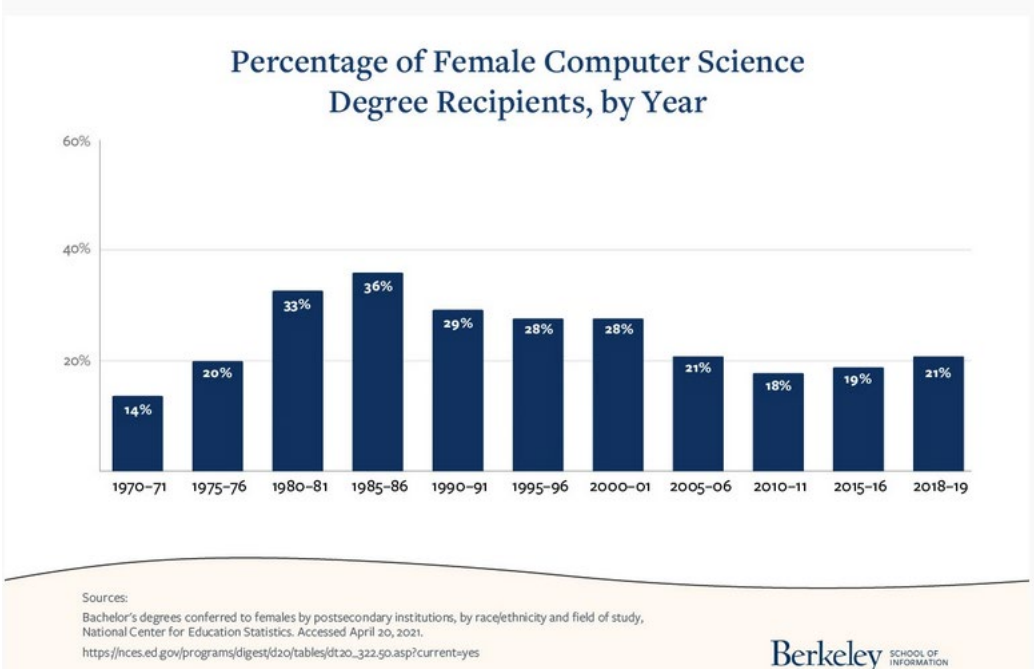
(a) A stimulus with a machine author (b) A stimulus with a woman author (c) A stimulus with a man author

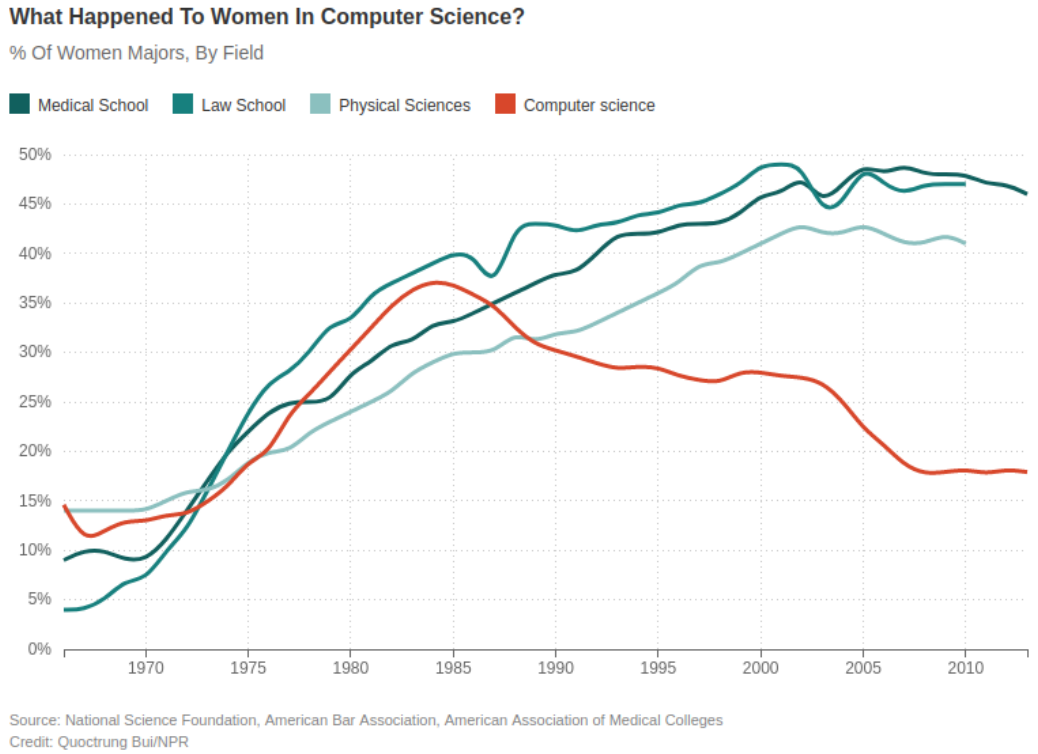
More on Biases and Diversity (endless)...

Gender differences and bias in open source: pull request acceptance of women versus men

Josh Terrell[1], Andrew Kofink[2], Justin Middleton[2], Clarissa Rainear[2], Emerson Murphy-Hill[2], Chris Parnin[2] and Jon Stallings[3]

Surprisingly, our results show that women's contributions tend to be accepted more often than men's. However, for contributors who are outsiders to a project and their gender is identifiable, men's acceptance rates are higher. Our results suggest that although women on GitHub may be more competent overall, bias against them exists nonetheless.

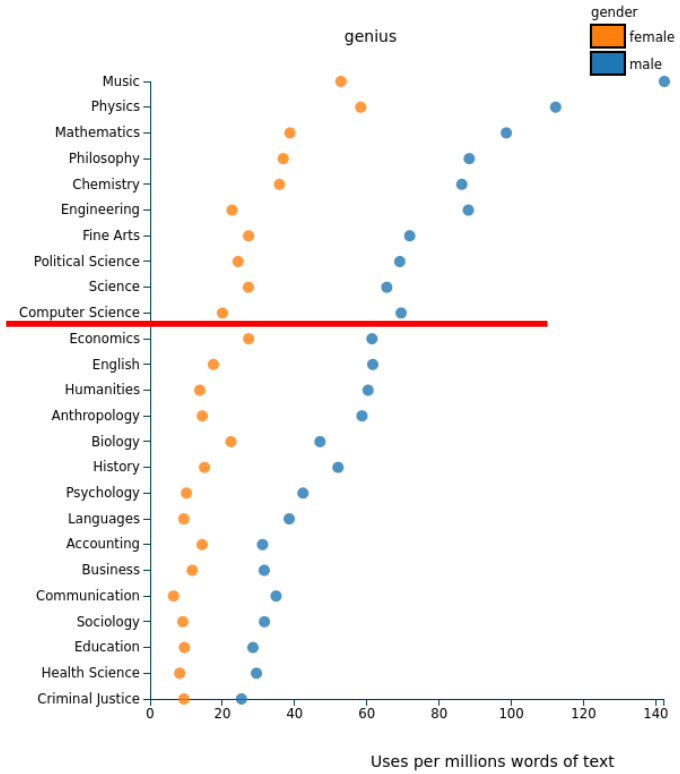


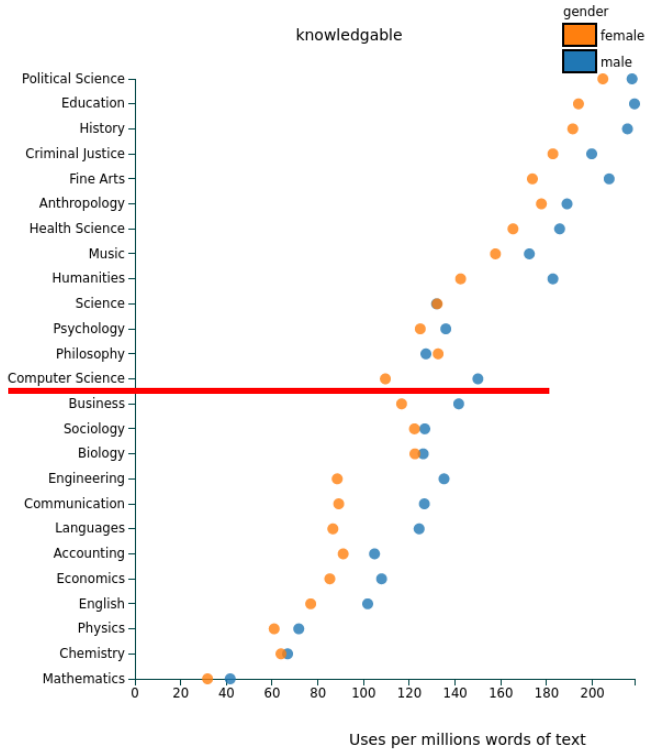


More on Biases and Diversity (endless)...

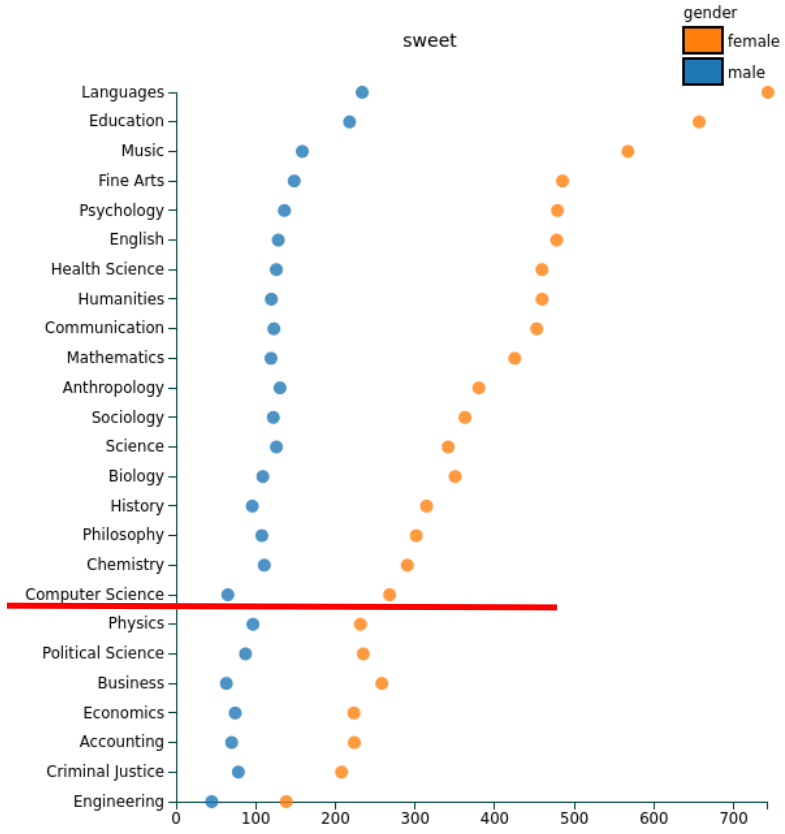
- Ratemyprofessors.com
- 14 million reviews
- A new tool allows those being rated (or anyone) to see the way students tend to use different words when rating male and female professors -- generally to the disadvantage of the latter.

More on Biases and Diversity (endless)...





More on Biases and Diversity (endless)...



More on Biases and Diversity (endless)...

Can salience of gender identity impair math performance among 7-8 years old girls? The moderating role of task difficulty

Emmanuelle Neuville
University Blaise Pascal, Clermont-Ferrand, CNRS, France

Jean-Claude Croizet
University of Poitiers, France

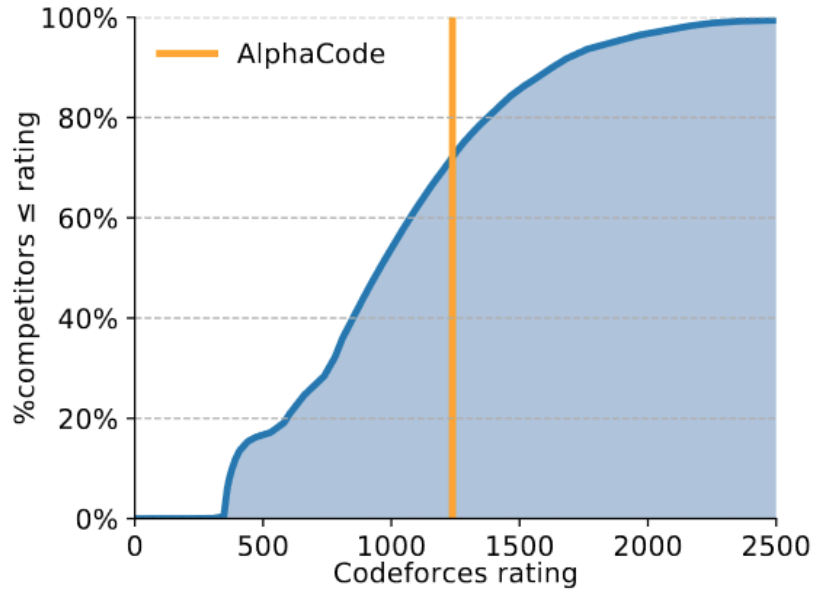
Can the salience of gender identity affect the math performance of 7–8 year old girls? Third-grade girls and boys were required to solve arthmetical problems of varied difficulty. Prior to the test, one half of the participants had their gender identity activated. Results showed that activation of gender identity affected girls' performance but not boys. When their gender was activated as opposed to when it was not, girls solved more problems when the material was less difficult but underperformed on the difficult problems. Results are discussed with regard to the stereotype threat literature.

Recall:

We want to improve productivity and reduce cost in software development and maintenance.

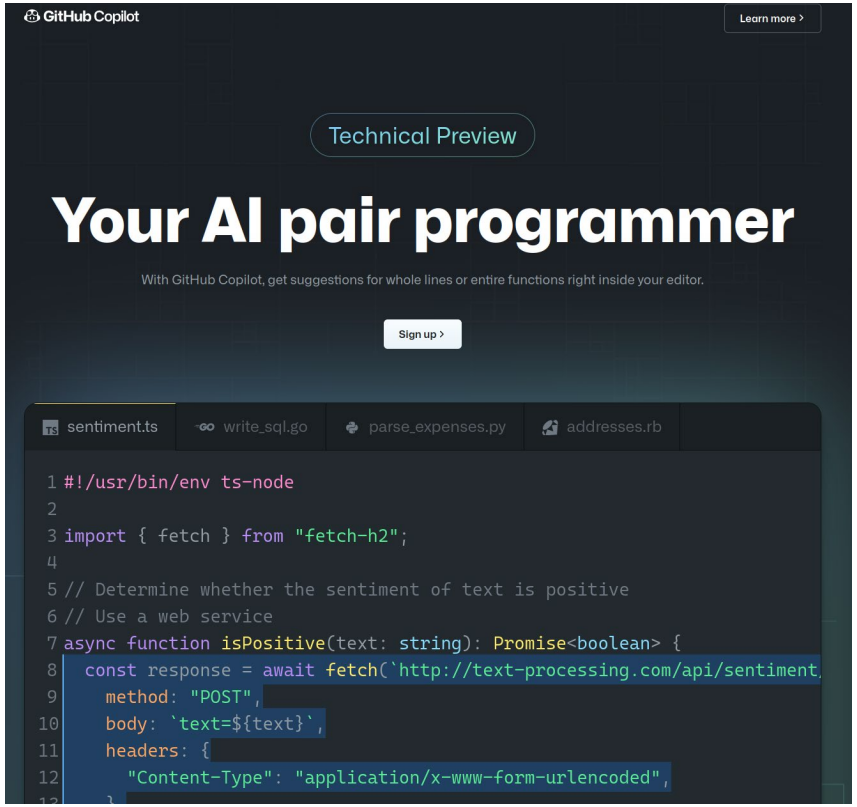
Can we design AI models to help with SE tasks?

programming problems, or else retrieving and copying existing solutions. As part of DeepMind's mission to solve intelligence, we created a system called AlphaCode that writes computer programs at a competitive level. AlphaCode achieved an estimated rank within the top 54% of participants in programming competitions by solving new problems that require a combination of critical thinking, logic, algorithms, coding, and natural language understanding.



(b) AlphaCode's estimated rating

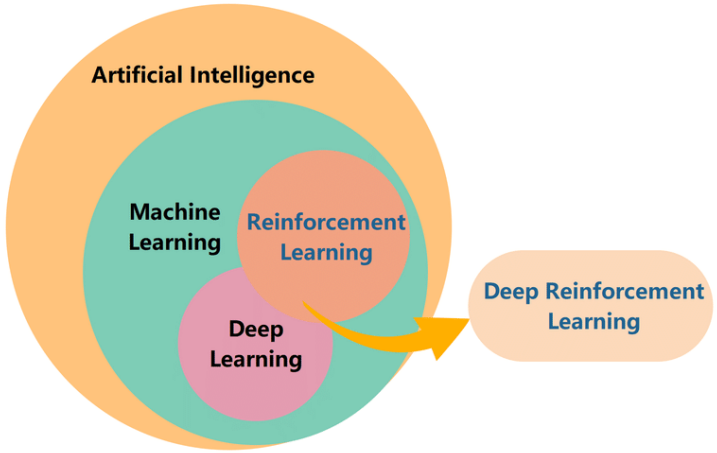
Can we design AI models to help with SE tasks?

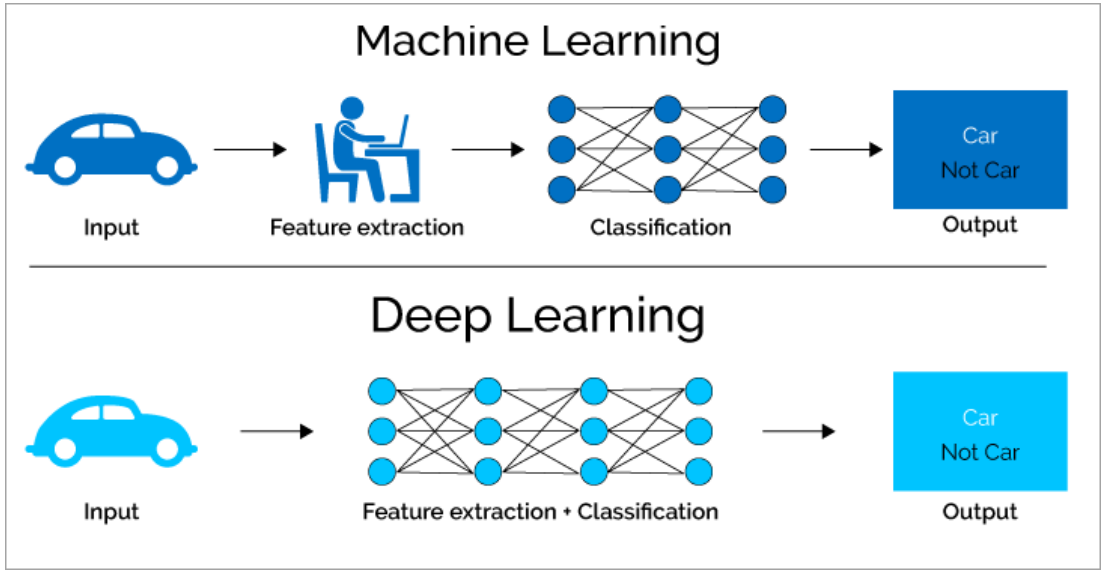


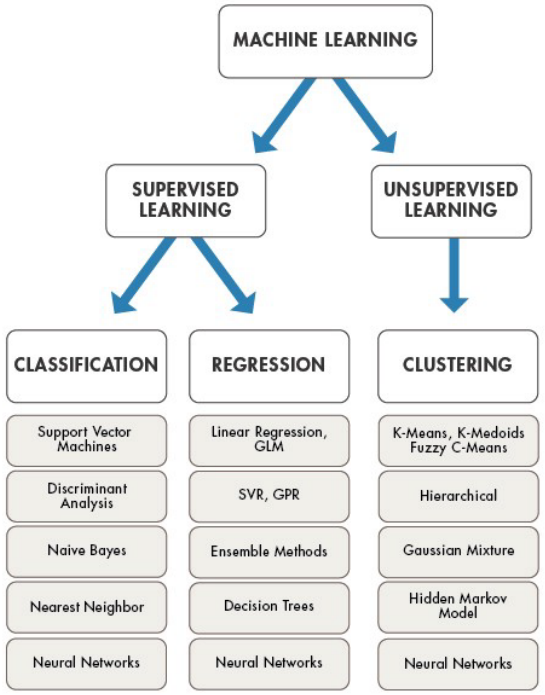
AI4SE: Downstream Tasks in SE

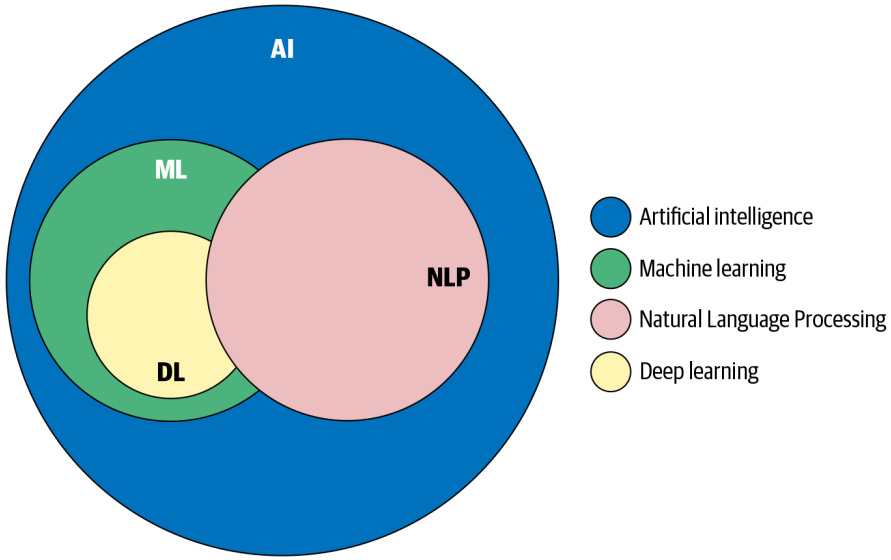
- "The task you actually want to solve" -- NLP
- Code generation
- Code summarization
- … …
- What can we talk about in 15 minutes?
    - Current work for AI4SE
    - Human-centered AI for SE
    - SE4AI

AI4SE: ML, AI and NLP



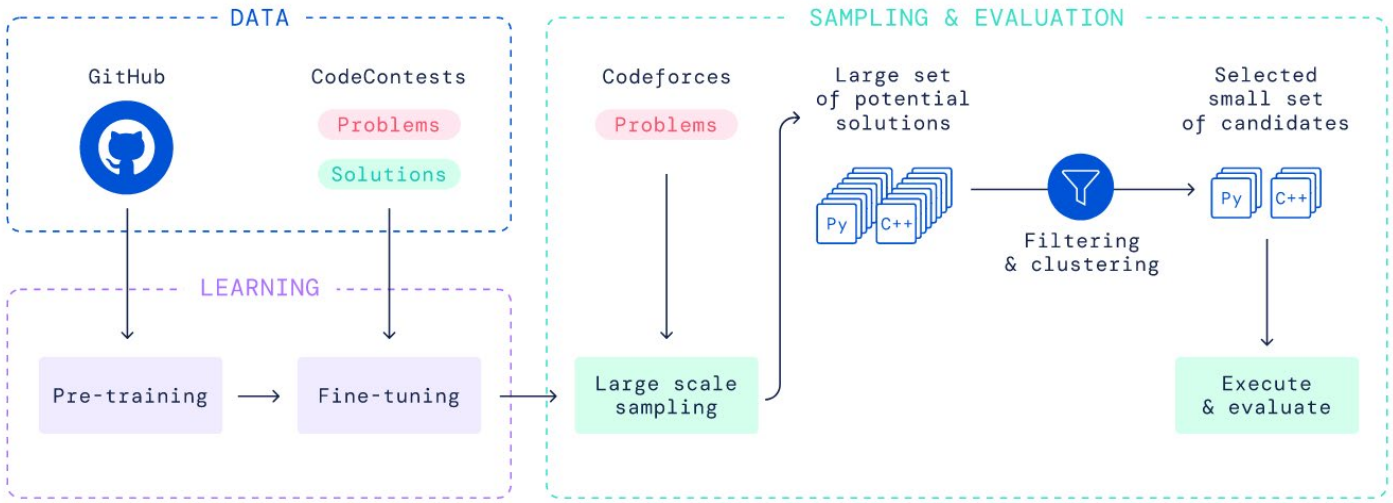




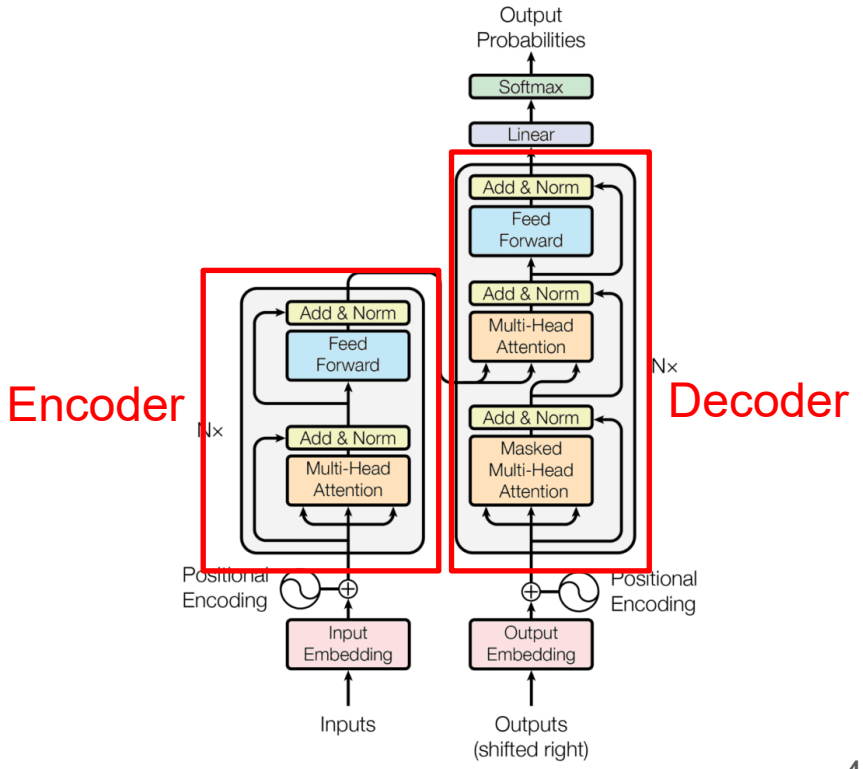


AI4SE: AlphaCode

AlphaCode: Transformer-based architecture
Transformer: handle sequential input data; a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV)



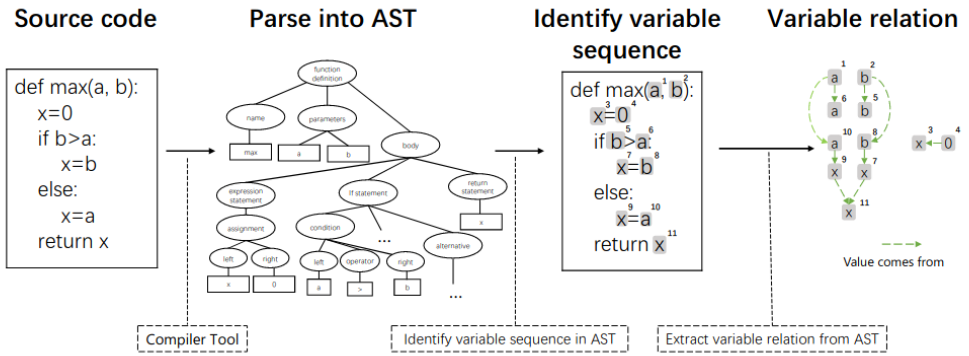
The Architecture of AlphaCode

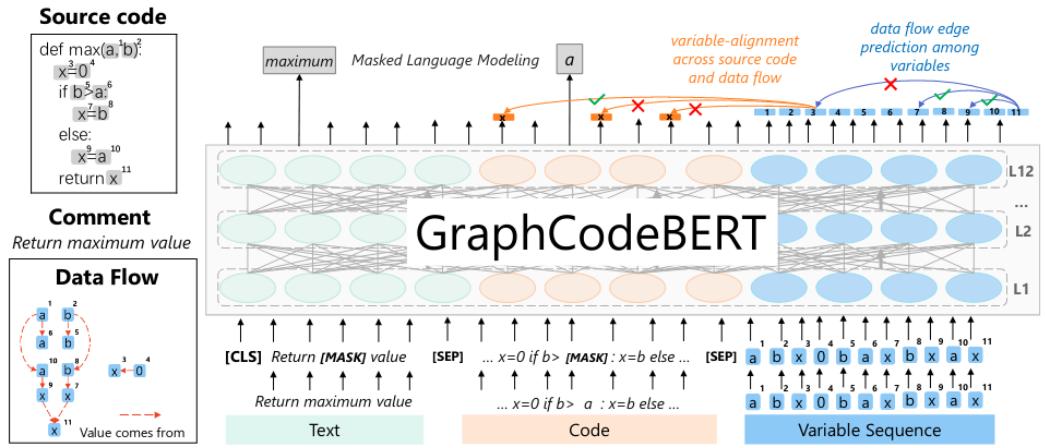


The Encoder-Decoder Architecture of Transformer

AI4SE: GraphCodeBERT

- BERT (**Bidirectional Encoder Representations from Transformers**) + "graph" representation of source code
- Focus on pre-training code representations with data flow
- Includes many topics you have learned in this course!





Pre-training embeddings of code are used in downstream tasks:
Code clone detection, code translation, natual language code search, etc.

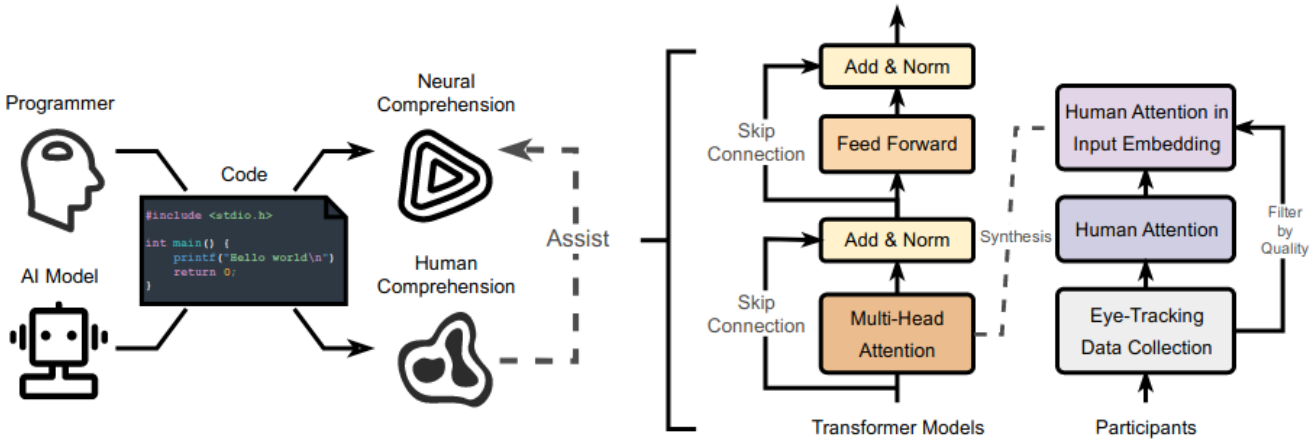
AI4SE: it is doable and happening!
Can we leverage developers' cognition to empower AI models for SE?
"So what" for human aspects research in SE?

Human-Centered AI for SE

EyeTrans: Merging Human and Machine Attention for Neural Code Summarization

YIFAN ZHANG, Vanderbilt University, USA
JILIANG LI, Vanderbilt University, USA
ZACHARY KARAS, Vanderbilt University, USA
AAKASH BANSAL, University of Notre Dame, USA
TOBY JIA-JUN LI, University of Notre Dame, USA
COLLIN MCMILLAN, University of Notre Dame, USA
KEVIN LEACH, Vanderbilt University, USA
YU HUANG, Vanderbilt University, USA

Accepted at FSE 2024



	MAF1@1				MAP@1				MAR@1			
Metrics	(R_1, N_1)	(R_2, N_1)	(R_1, N_2)	(R_2, N_2)	(R_1, N_1)	(R_2, N_1)	(R_1, N_2)	(R_2, N_2)	(R_1, N_1)	(R_2, N_1)	(R_1, N_2)	(R_2, N_2)
Transformer (Original)	96.90	64.62	90.47	49.70	96.53	61.90	88.46	46.85	97.74	71.29	92.90	57.74
EyeTrans (Original)	99.61	70.31	93.10	56.43	99.56	68.14	92.26	53.85	99.68	76.13	94.84	63.55
Improvement	+2.80%	+8.79%	+2.91%	**+13.52%**	+3.15%	+10.11%	+4.29%	**+14.95%**	+1.98%	+6.80%	+2.09%	**+10.06%**
Transformer (Filtered)	92.78	53.94	75.78	42.59	91.90	51.58	73.67	39.99	94.47	60.43	80.43	50.21
EyeTrans (Filtered)	96.09	58.44	89.74	54.40	95.61	56.01	88.74	51.95	97.02	65.11	91.92	61.28
Improvement	+3.56%	+8.35%	+18.42%	**+27.82%**	+4.03%	+8.59%	+20.51%	**+29.91%**	+2.71%	+7.73%	+14.33%	**+22.03%**
Transformer (Strict)	82.92	52.76	76.54	46.70	81.36	49.21	74.11	42.95	86.45	61.29	81.94	55.48
EyeTrans (Strict)	95.68	55.58	83.87	49.48	95.15	52.15	82.32	45.71	96.77	63.87	87.10	58.71
Improvement	**+15.38%**	+5.33%	+9.58%	+5.96%	**+16.94%**	+5.97%	+11.05%	+6.43%	**+11.95%**	+4.21%	+6.30%	+5.80%

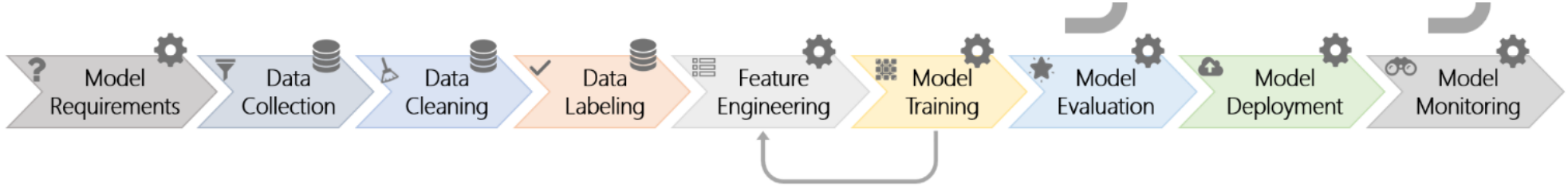
Software Engineering for AI

- All the ML/AI/NLP/DM tasks are software engineering tasks!
- What can SE do to assist them?

Software Engineering for Machine Learning: A Case Study

Saleema Amershi, *Microsoft Research*, Redmond, WA USA, samershi@microsoft.com
Andrew Begel, *Microsoft Research*, Redmond, WA USA, andrew.begel@microsoft.com
Christian Bird, *Microsoft Research*, Redmond, WA USA, cbird@microsoft.com
Robert DeLine, *Microsoft Research*, Redmond, WA USA, rdeline@microsoft.com
Harald Gall, *University of Zurich*, Zurich, Switzerland, gall@ifi.uzh.ch
Ece Kamar, *Microsoft Research*, Redmond, WA USA, eckamar@microsoft.com
Nachiappan Nagappan, *Microsoft Research*, Redmond, WA USA, nachin@microsoft.com
Besmira Nushi, *Microsoft Research*, Redmond, WA USA, besmira.nushi@microsoft.com
Thomas Zimmermann, *Microsoft Research*, Redmond, WA USA, tzimmer@microsoft.com

Challenge	Frequency: Medium vs. Low	Frequency: High vs. Low	Frequency: Trend	Rank: Experience Low	Rank: Experience Medium	Rank: Experience High
Data Availability, Collection, Cleaning, and Management	-2%	60%		1	1	1
Education and Training	-69%	-78%		1	5	9
Hardware Resources	-32%	13%		3	8	6
End-to-end pipeline support	65%	41%		4	2	4
Collaboration and working culture	19%	69%		5	6	6
Specification	2%	50%		5	8	8
Integrating AI into larger systems	-49%	-62%		5	16	13
Education: Guidance and Mentoring	-83%	-81%		5	21	18
AI Tools	144%	193%		9	3	2
Scale	154%	210%		10	4	3
Model Evolution, Evaluation, and Deployment	137%	276%		15	6	4



The nine stages of the ML workflow

SE4AI: there is so much to do!

CS 4278/5278 Principles of Software Engineering
SP 2024

Please take the course evaluation

Thank you and good luck!