Metrics and Measurement

17-313 Fall 2024

Foundations of Software Engineering

https://cmu-313.github.io

Michael Hilton and Rohan Padhye

Administrivia

- Project 1(b) due on Thursday (Sep 5th) at midnight
- Slack: Great usage of #technical-support-channel so far!

- Project 2 will be released tonight.
  - P2 onwards will be team projects! Teams at the end of class.
  - P2A due next Thursday, Sep 12th. Some topics (Kanban, user stories) will be covered next Tuesday, Sep 10th.

Smoking Section

- Last full row

Today's Learning Goals

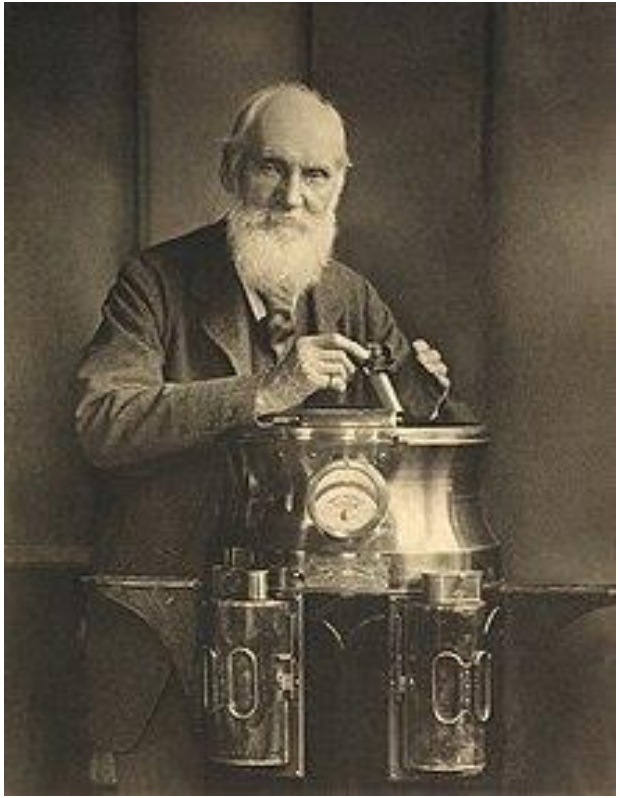
- Explain the importance of measurement and metrics in Software Engineering
- Provide examples of metrics for software qualities and process
- Apply goal-based frameworks for decision making using metrics
- Identify the limitations and dangers of decisions and incentives based on measurements

Measurement in everyday life

- Economics
  - price, inflation rate, stock price, volume
- Medicine
  - heart rate, blood pressure, body temperature, ECG
- Engineering
  - Force, torque, heat transfer coefficient, thermal efficiency
- Natural sciences
  - AQI, carbon footprint, Soil pH

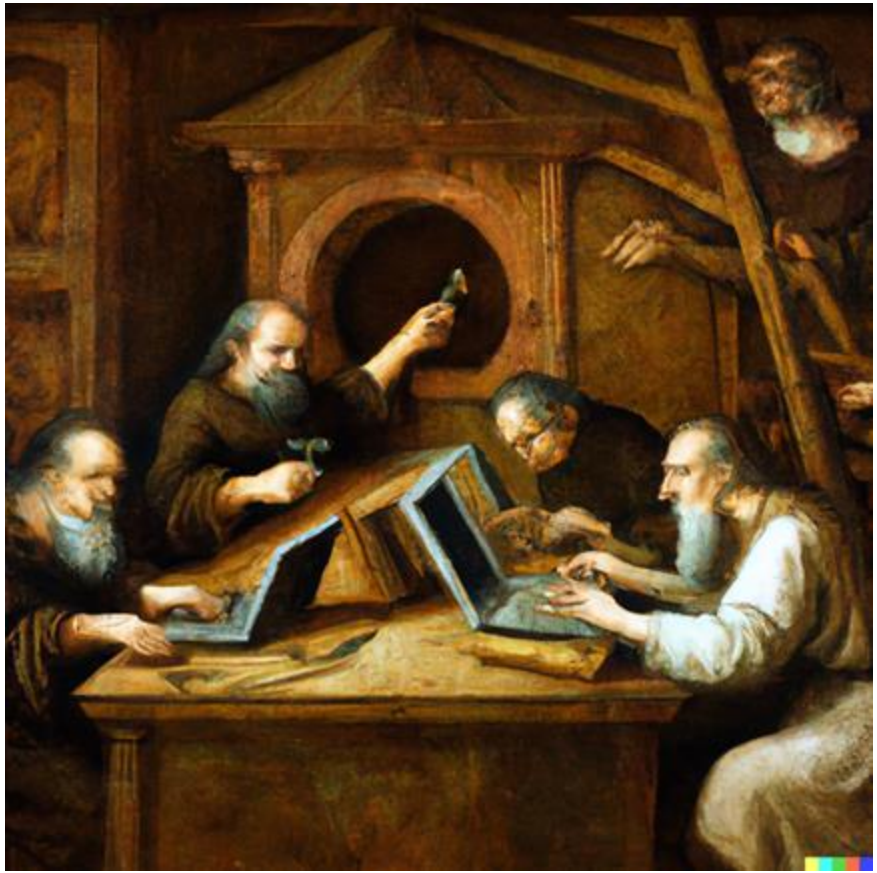
“To measure is to know;
if you can not measure it,
you can not improve it”

William Thomson, Lord Kelvin



$$K = \left(\frac{5}{9}(F - 32)\right) + 273.15$$

Software Development… before Software Engineering



by DALL-E

Software Engineering

Software Engineering: Principles, practices (technical and non-technical) for confidently building high-quality software.

What does this mean? How do we know?
Measurement and metrics are **key** concerns.

Outline

- Measurements and Metrics
- How to use measurements and metrics?
- Case study: Autonomous Vehicle Software
- Risks and challenges
- Metrics and incentives

Outline

- **Measurements and Metrics**
- How to use measurements and metrics?
- Case study: Autonomous Vehicle Software
- Risks and challenges
- Metrics and incentives

What is Measurement?

- **Measurement is the empirical, objective assignment of numbers, according to a rule derived from a model or theory, to attributes of objects or events with the intent of describing them.** – Craner, Bond, “Software Engineering Metrics: What Do They Measure and How Do We Know?”
- **A quantitatively expressed reduction of uncertainty based on one or more observations.** – Hubbard, “How to Measure Anything …”

Software Quality Metrics

- IEEE 1061 definition: **"A software quality metric is a function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given attribute that affects its quality."**

Entity	**Attribute**	**Measurement**
Object or Process	Quality of Interest	Method to obtain a number or a symbol

What entities to we care about? (examples)

- Software product
- Modules
- Software development process
- People

What software qualities do we care about? (examples)

- Functionality (e.g., data integrity)
- Scalability
- Security
- Extensibility
- Bugginess
- Documentation
- Performance
- Installability
- Availability
- Consistency
- Portability
- Regulatory compliance

What process qualities do we care about? (examples)

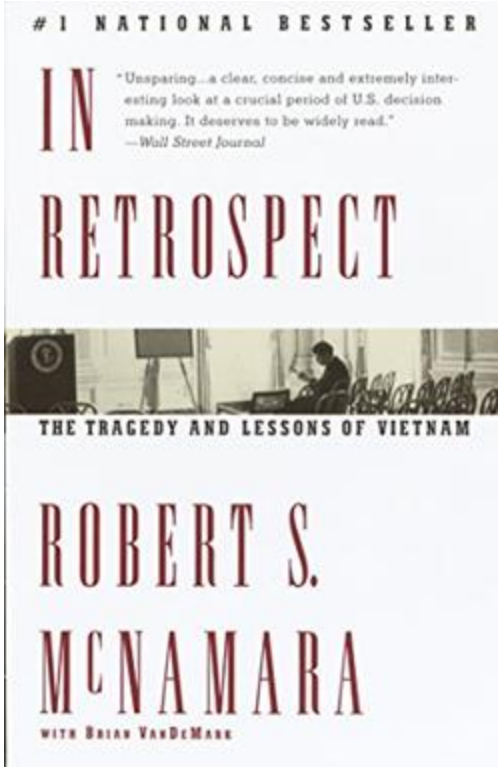
- Development efficiency
- Meeting efficiency
- Conformance to processes
- Reliability of predictions
- Fairness in decision making
- Regulatory compliance
- On-time release

What people qualities do we care about? (examples)

- Developers
  - Maintainability
  - Performance
  - Employee satisfaction and well-being
  - Communication and collaboration
  - Efficiency and flow
  - Satisfaction with engineering system
  - Regulatory compliance
- Customers
  - Satisfaction
  - Ease of use
  - Feature usage
  - Regulatory compliance

Non-trivial qualities

- Software
  - Code elegance
  - Code maintainability
- Process
  - Fairness in decision making
- Team
  - Team collaboration
  - Creativity


#1 NATIONAL BESTSELLER
IN RETROSPECT
"Unsparing...a clear, concise and extremely interesting look at a crucial period of U.S. decision making. It deserves to be widely read."
—Wall Street Journal
THE TRAGEDY AND LESSONS OF VIETNAM
ROBERT S. McNAMARA
WITH BRIAN VANDEMARK


FROM THE DIRECTOR OF "THE THIN BLUE LINE" AND "GATES OF HEAVEN"
"IF THERE IS ONE MOVIE THAT OUGHT TO BE STUDIED BY MILITARY AND CIVILIAN LEADERS AROUND THE WORLD AT THIS TREACHEROUS HISTORICAL MOMENT, IT IS 'THE FOG OF WAR.' ERROL MORRIS'S PORTRAIT OF FORMER UNITED STATES DEFENSE SECRETARY, ROBERT S. MCNAMARA."
"SMART AND COMPLETELY FASCINATING!"
"STUNNING! SUPERBLY MADE!"
"SPLENDID! A MASTERPIECE!"
THE FOG OF WAR
BEST DOCUMENTARY FEATURE
BEST ORIGINAL SCORE
BEST EDITING

McNamara fallacy

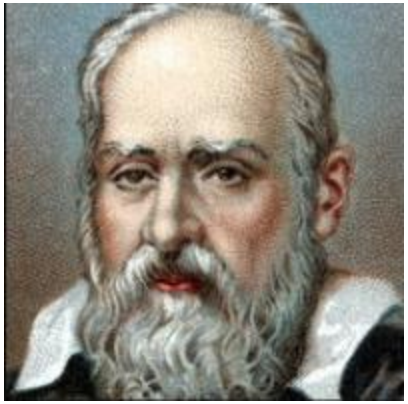
- Measure whatever can be easily measured.
- Disregard that which cannot be measured easily.
- Presume that which cannot be measured easily is not important.
- Presume that which cannot be measured easily does not exist.

https://chronotopeblog.com/2015/04/04/the-mcnamara-fallacy-and-the-problem-with-numbers-in-education/

Make it measurable

“Measure what is measurable, and make measurable what is not so.”

Galileo Galilei



Everything is measurable

- If X is something we care about, then X, by definition, must be detectable.
  - How could we care about things like “quality,” “risk,” “security,” or “public image” if these things were totally undetectable, directly or indirectly?
  - If we have reason to care about some unknown quantity, it is because we think it corresponds to desirable or undesirable results in some way.
- If X is detectable, then it must be detectable in some amount.
  - If you can observe a thing at all, you can observe more of it or less of it
- If we can observe it in some amount, then it must be measurable.

Douglas Hubbard, How to Measure Anything, 2010

Examples:
Code Complexity

Code Complexity: Lines of Code

- Easy to measure

```
> wc –l file1 file2…
```

LOC	projects
450	Expression Evaluator
2,000	Sudoku
100,000	Apache Maven
500,000	Git
3,000,000	MySQL
15,000,000	gcc
50,000.000	Windows 10
2,000,000,000	Google (MonoRepo)

Code Complexity: Halstead Volume

- Introduced by Maurice Howard Halstead in 1977
- Halstead Volume =
  number of operators/operands *
  $\log_2$(number of distinct operators/operands)
- Approximates size of elements and vocabulary

Code Complexity: Cyclomatic Complexity

- Proposed by McCabe 1976
- Based on control flow graph, measures linearly independent paths through a program
  - ~= number of decisions
  - Number of test cases needed to achieve branch coverage

```
if (c1) {
                f1();
} else {
                f2();
}
if (c2) {
                f3();
} else {
                f4();
}
```

M = edges of CFG – nodes of CFG + 2*connected components

"For each module, either limit cyclomatic complexity to [X] or provide a written explanation of why the limit was exceeded."
– NIST Structured Testing methodology

Code Complexity: Object-Oriented Metrics

- Number of Methods per Class
- Depth of Inheritance Tree
- Number of Child Classes
- Coupling between Object Classes
- Calls to Methods in Unrelated Classes
- …

Code Complexity: “Allowable mass” proxy

- Mentioned in a talk by Jinnah Hossein, Boeing (formerly SpaceX) at CMU
- More complex the software task ==> consumes bigger part of the mass budget (in grams)
- Keeps the software from growing unconstrained and ensuring maintainability and quality by existing staff

Outline

- Measurements and Metrics
- **How to use measurements and metrics?**
- Case study: Autonomous Vehicle Software
- Risks and challenges
- Metrics and incentives

A Goal-based Framework

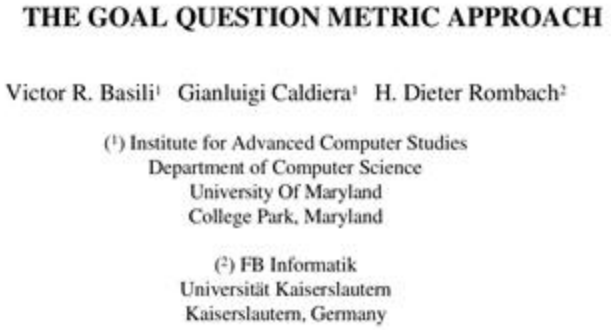
"Every measurement action must be motivated by a particular goal or need that is clearly defined and easily understandable."

Software Metrics: A Rigorous and Practical Approach. N.Fenton, J.Bieman

Goal: What do you want to achieve?

Questions: What do you need to answer to know whether your goal is met?

Metrics: What measurements do you need in order to answer each question?



THE GOAL QUESTION METRIC APPROACH

Victor R. Basili[1] Gianluigi Caldiera[1] H. Dieter Rombach[2]

([1]) Institute for Advanced Computer Studies
Department of Computer Science
University Of Maryland
College Park, Maryland

([2]) FB Informatik
Universität Kaiserslautern
Kaiserslautern, Germany

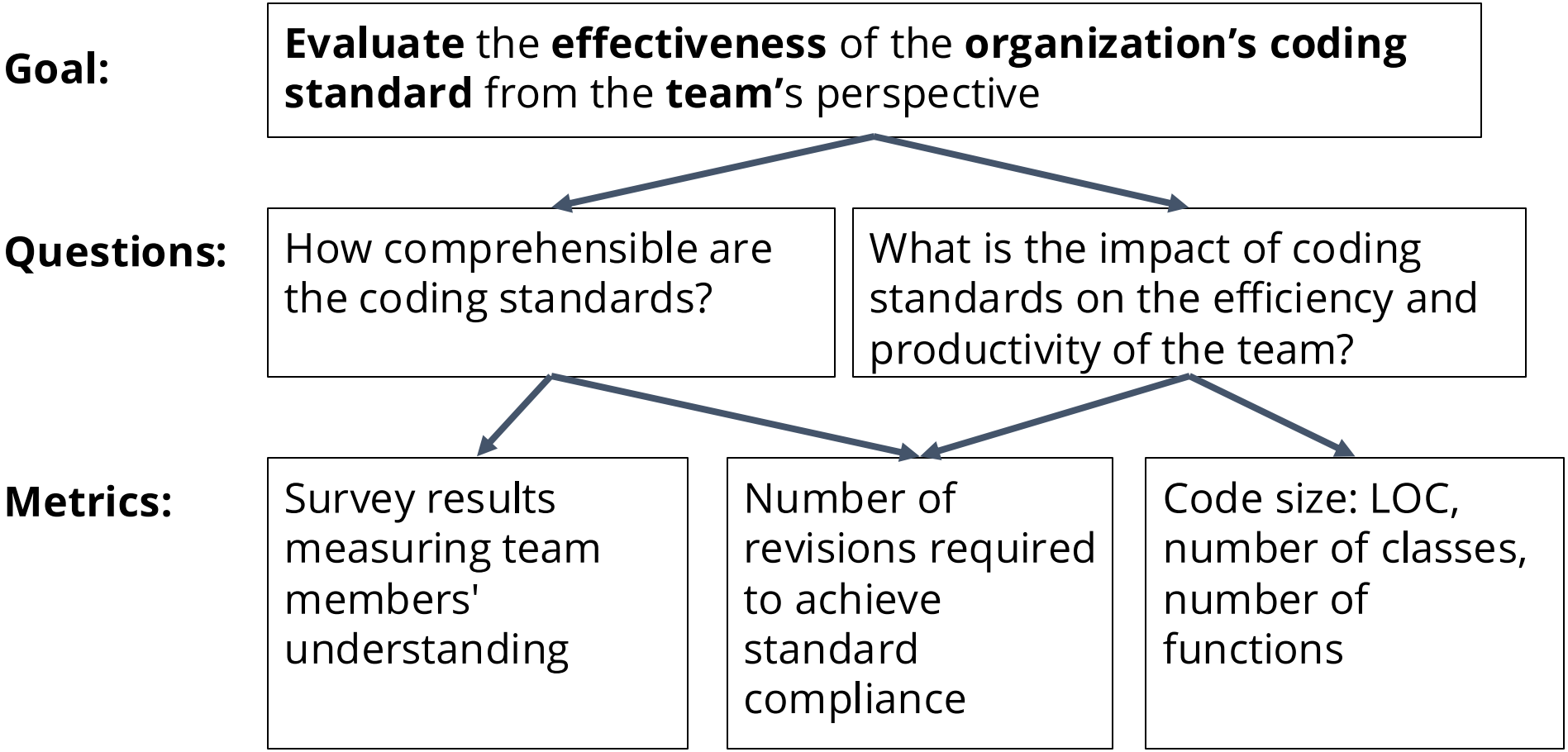
GQM: Defining Goals

P: **Purpose** (improve, evaluate, monitor, …)

I: **Issue** (reliability, usability, effectiveness, …)

O: **Object** (final product, component, process, activity)

V: **Viewpoint** (any stakeholder)


Goal:
Evaluate the effectiveness of the organization's coding standard from the team's perspective
Questions:
How comprehensible are the coding standards?
What is the impact of coding standards on the efficiency and productivity of the team?
Metrics:
Survey results measuring team members' understanding
Number of revisions required to achieve standard compliance
Code size: LOC, number of classes, number of functions

Goal:

Monitor the **performance** of the **web server** to enable the **ops team** to make decisions

Questions:

How quickly can users complete their tasks?

How many concurrent users can we support?

Metrics:

Average latency per request in milliseconds

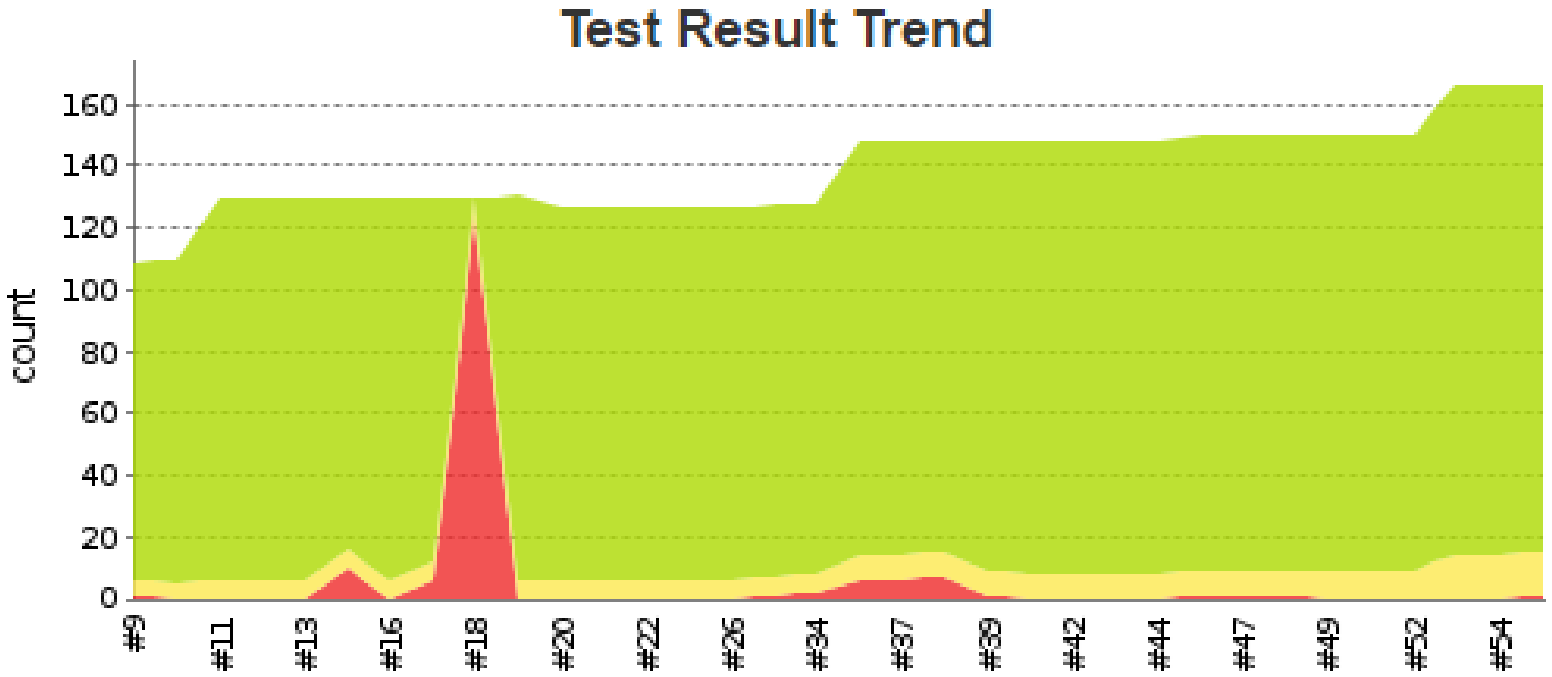
Throughput: Number of requests served per second

Peak memory consumption, as a % of max available

Measurement for Decision Making

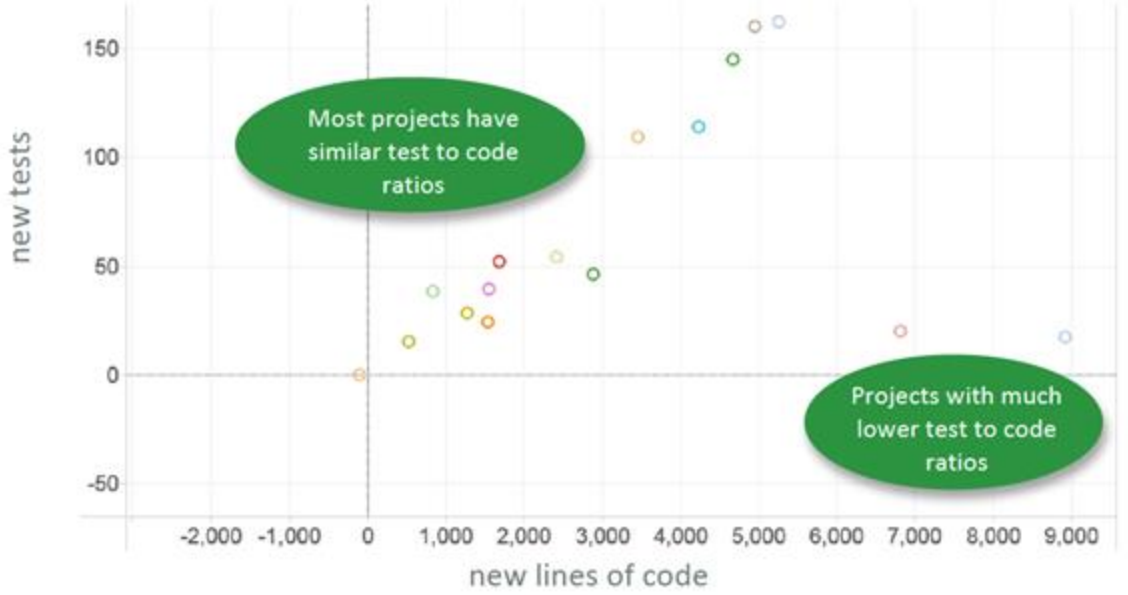
- Fund project?
- More testing?
- Fast enough? Secure enough?
- Code quality sufficient?
- Which feature to focus on?
- Developer bonus?
- Time and cost estimation? Predictions reliable?

Trend analyses



Benchmarking against standards

- Monitor many projects or many modules, get typical values for metrics
- Report deviations



Outline

- Measurements and Metrics
- How to use measurements and metrics?
- **Case study: Autonomous Vehicle Software**
- Risks and challenges
- Metrics and incentives

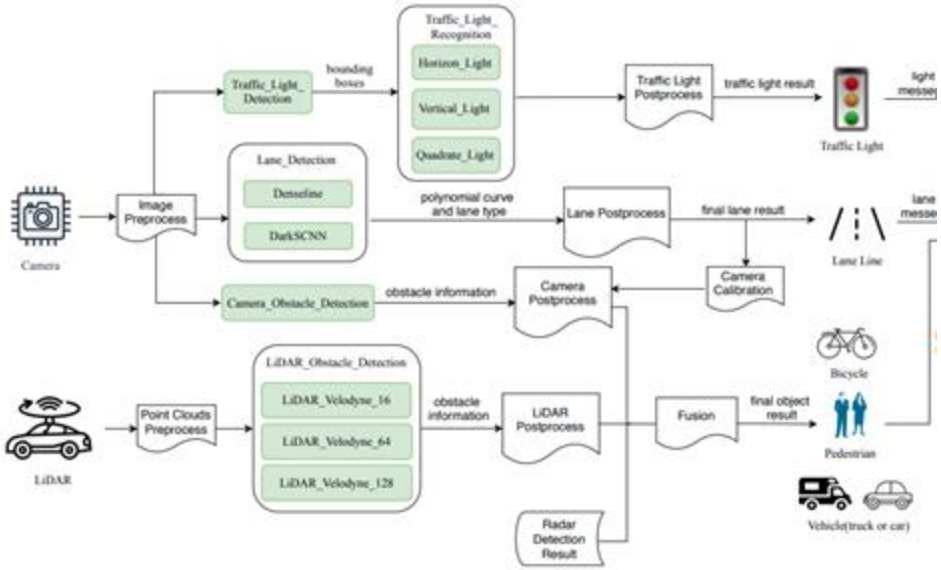
By what metrics can we judge AV software (e.g., safety)?

(1) Code coverage

- Amount of code executed during testing.
- Statement coverage, line coverage, branch coverage, etc.
- E.g., 75% branch coverage
  ⇒ 3/4 if-else outcomes have been executed

(2) Model Accuracy

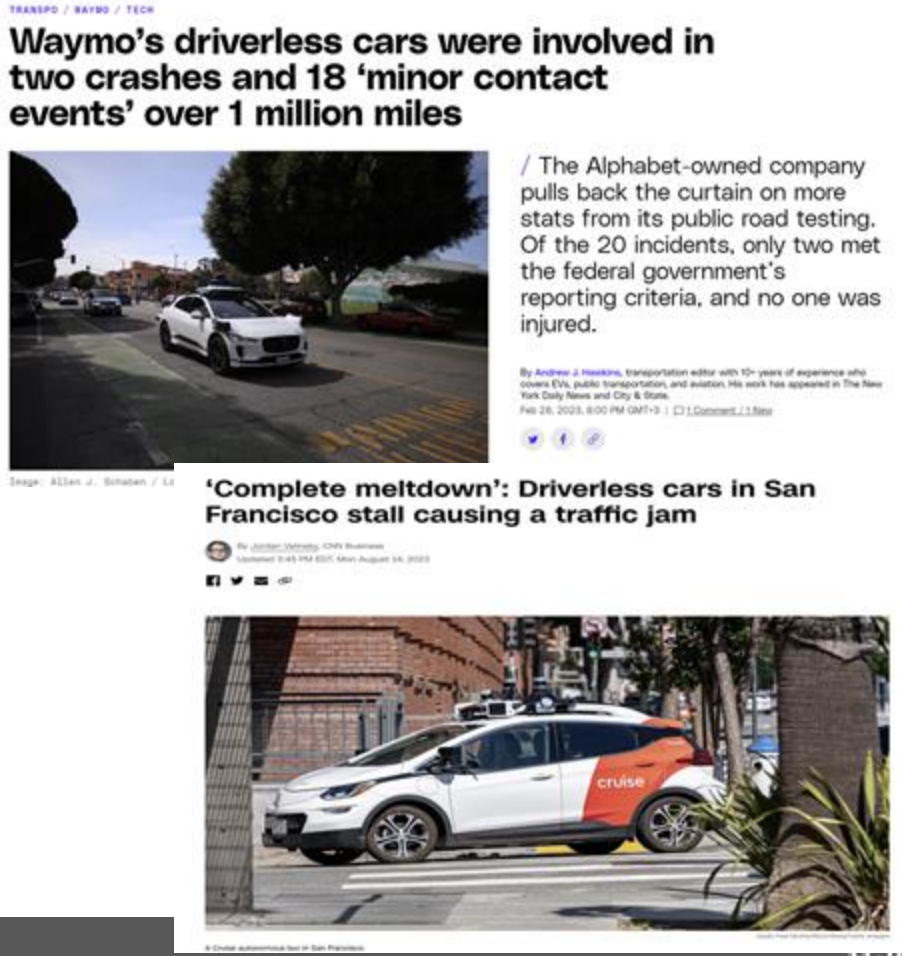
- Train machine-learning models on labelled data (sensor data + ground truth).
- Compute accuracy on a separate labelled test set.
- E.g., 90% accuracy implies that object recognition is right for 90% of the test inputs.



Source: Peng et al. ESEC/FSE'20

(3) Failure Rate

- Frequency of crashes / fatalities
- Per 1,000 rides, per million miles, per month (in the news)



TRANSPO / WAYMO / TECH

Waymo's driverless cars were involved in two crashes and 18 'minor contact events' over 1 million miles

/ The Alphabet-owned company pulls back the curtain on more stats from its public road testing. Of the 20 incidents, only two met the federal government's reporting criteria, and no one was injured.

By Andrew J. Hawkins, transportation editor with 10+ years of experience who covers EVs, public transportation, and aviation. His work has appeared in The New York Daily News and City & State.

Feb 28, 2023, 8:00 PM GMT+3 | 1 Comment / 1 New

Image: Allen J. Schaben / La

'Complete meltdown': Driverless cars in San Francisco stall causing a traffic jam

By Jordan Valinsky, CNN Business

Updated 3:45 PM EDT, Mon August 14, 2023

A Cruise autonomous taxi in San Francisco

(4) Mileage

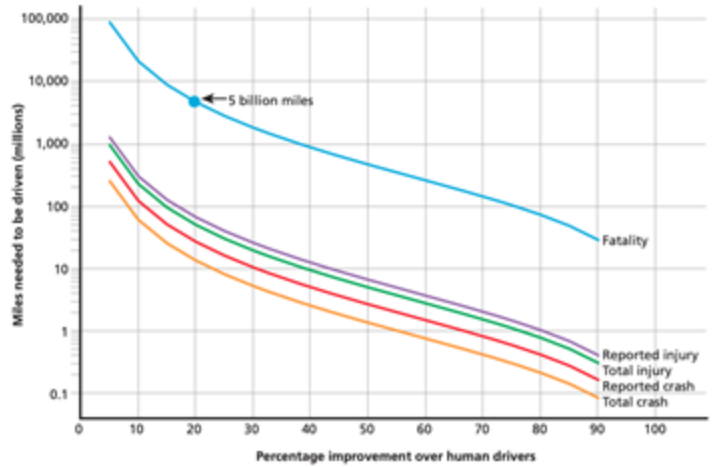
RAND CORPORATION

Driving to Safety

How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?

Nidhi Kalra, Susan M. Paddock

Figure 3. Miles Needed to Demonstrate with 95% Confidence that the Autonomous Vehicle Failure Rate Is Lower than the Human Driver Failure Rate



Source: waymo.com/safety (September 2021)

Participation Activity

- You can work in groups of 3.
- Apply the **Goal-Question-Metric** framework to explore various aspects of AV software
- Define one goal, two questions, and at least one metric per question
- Write it down on a piece of paper with your Andrew ID(s) on it.
- Share with the class!

- Software
  - Test coverage
  - Model accuracy
  - Size of codebase
  - Age of codebase
- Software Process
  - Time since the most recent change
  - Frequency of code releases
  - Number of emails sent during development
- Contributors
  - Number of contributors
  - Age of contributors
  - Employee satisfaction of contributors
- Documentation
  - Amount of code documentation
- Application
  - Customer satisfaction
  - Mileage
  - Crash/kill rate

Example

Goal: Ensure energy efficiency and sustainability from the point of view of the organization and environmental analysts

Question 1: What is the vehicle's energy consumption under different driving conditions?

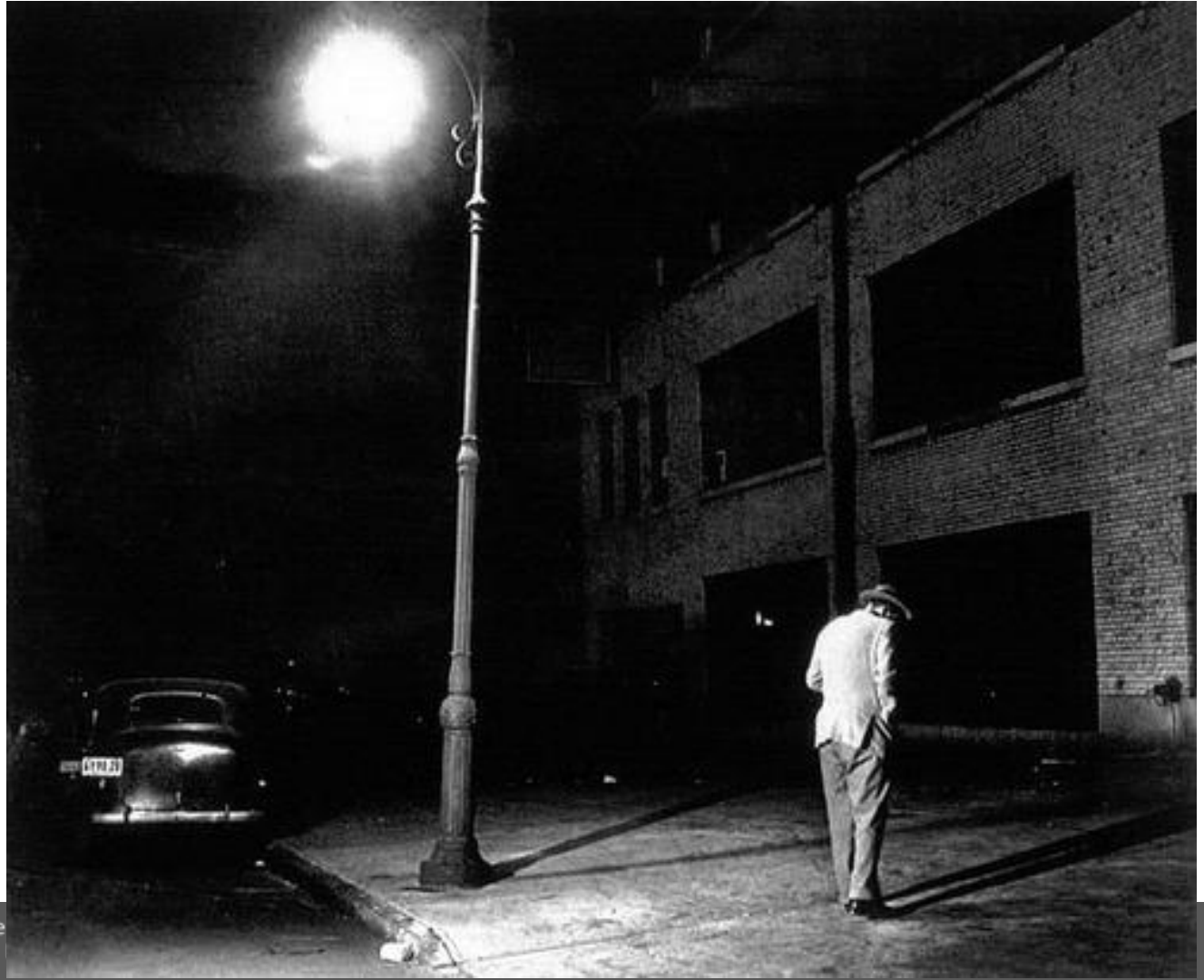
Metrics: Kilowatt-hours per 100 kilometers under city, highway, and mixed driving conditions.

Question 2: How efficient is the battery management system?

Metrics: Battery life in miles, number of charge cycles

Outline

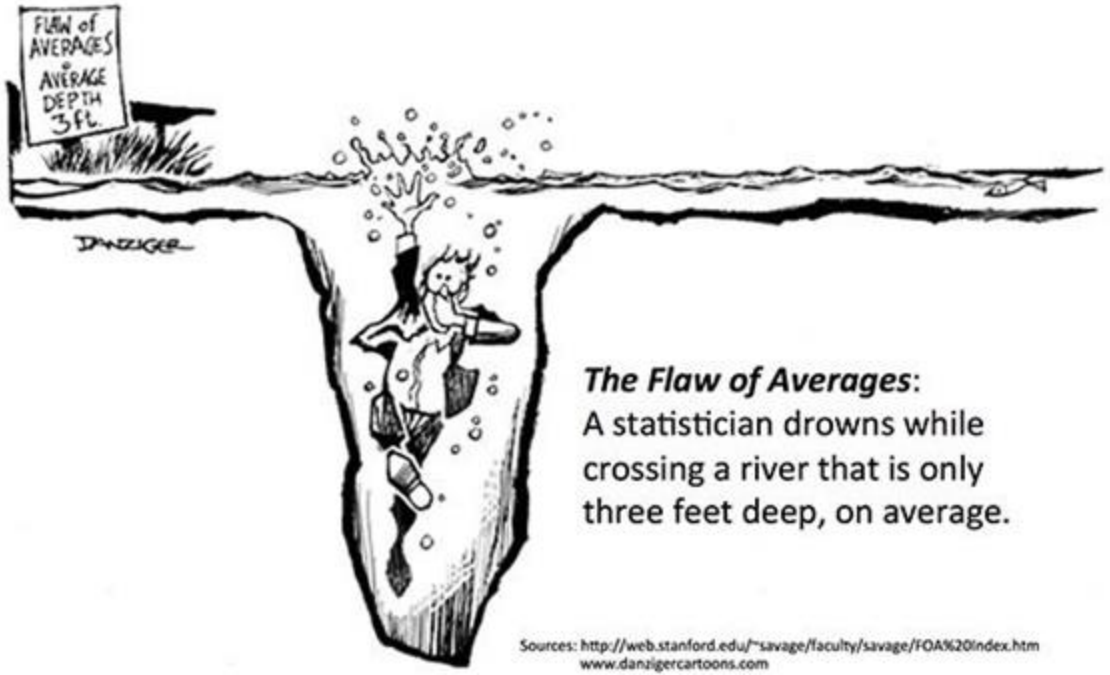
- Measurements and Metrics
- How to use measurements and metrics?
- Case study: Autonomous Vehicle Software
- **Risks and challenges**
- Metrics and incentives



The streetlight effect

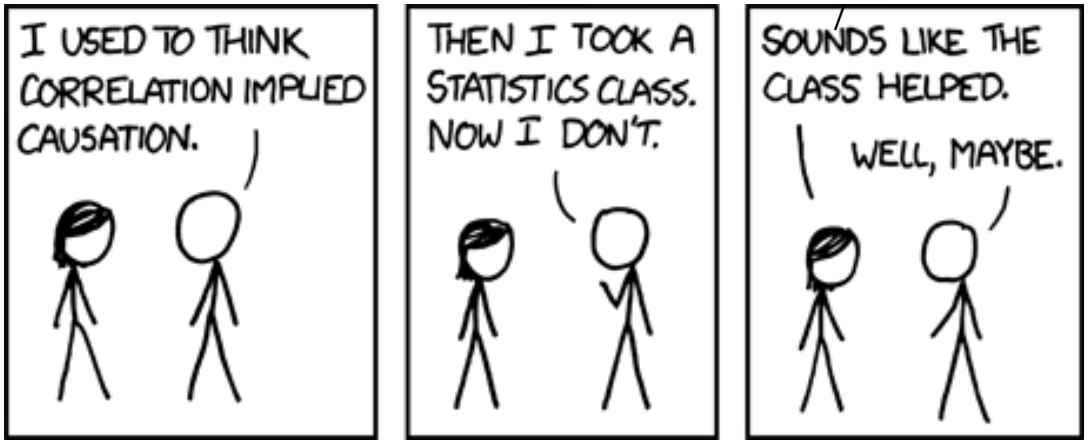
- A known observational bias.
- People tend to look for something only where it's easiest to do so.
  - If you drop your keys at night, you'll tend to look for it under streetlights.

Bad statistics: What could possibly go wrong?



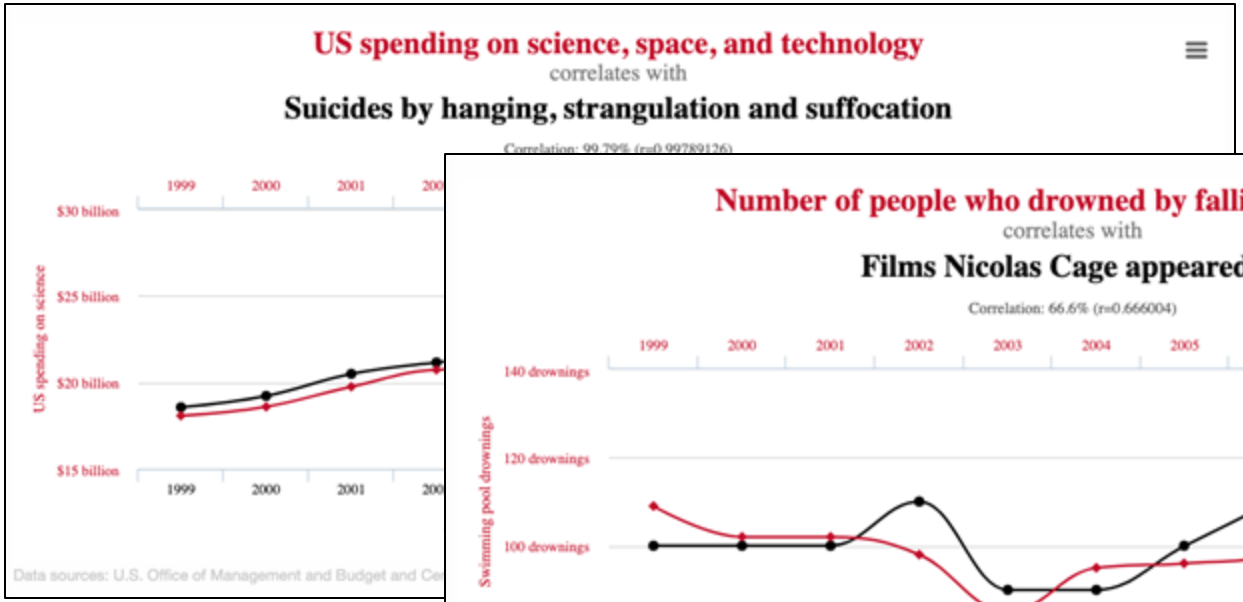
Making inferences

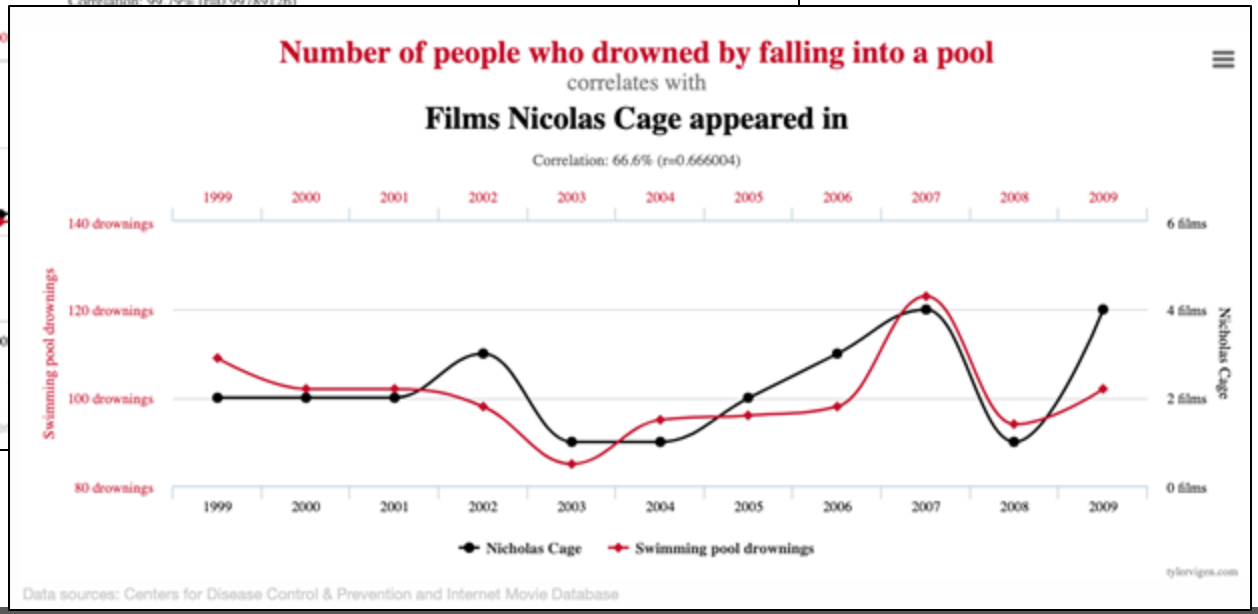
http://xkcd.com/552



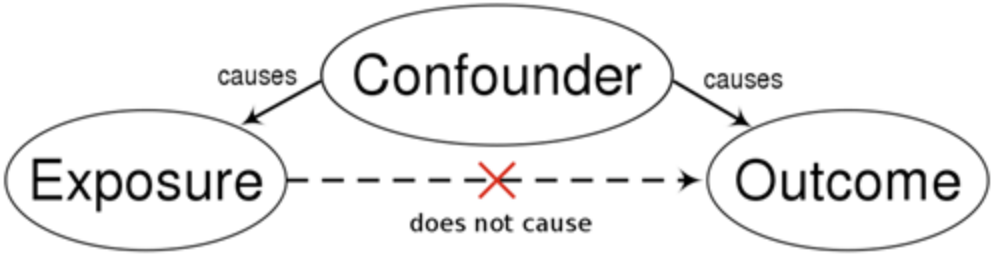
- To infer causation:
  - Provide a theory (from domain knowledge, independent of data)
  - Show correlation
  - Demonstrate ability to predict new cases (replicate/validate)

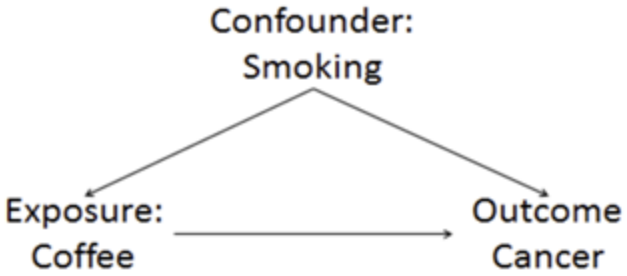
Spurious Correlations





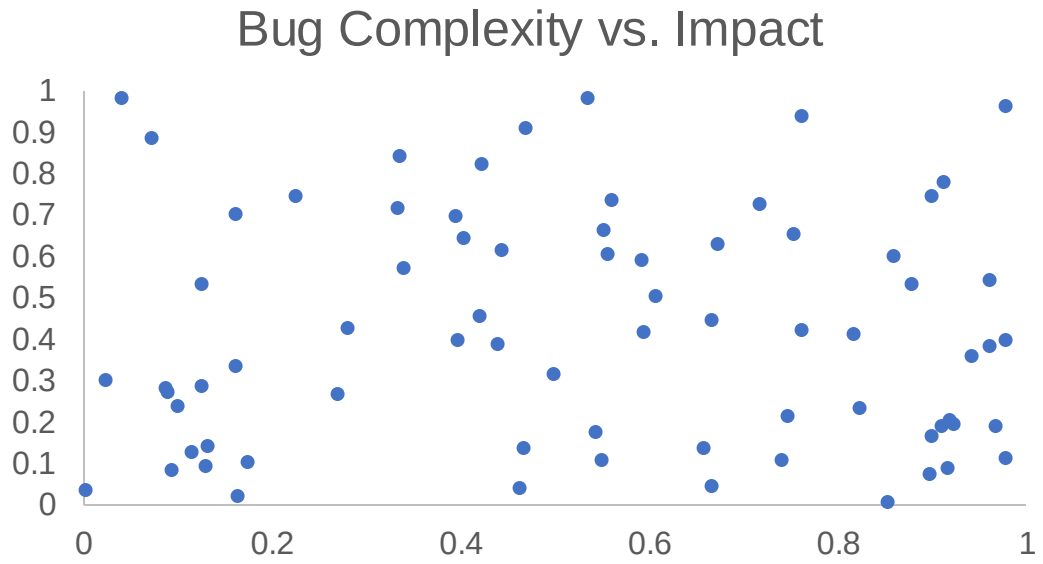
Spurious Correlations: Confounding variables



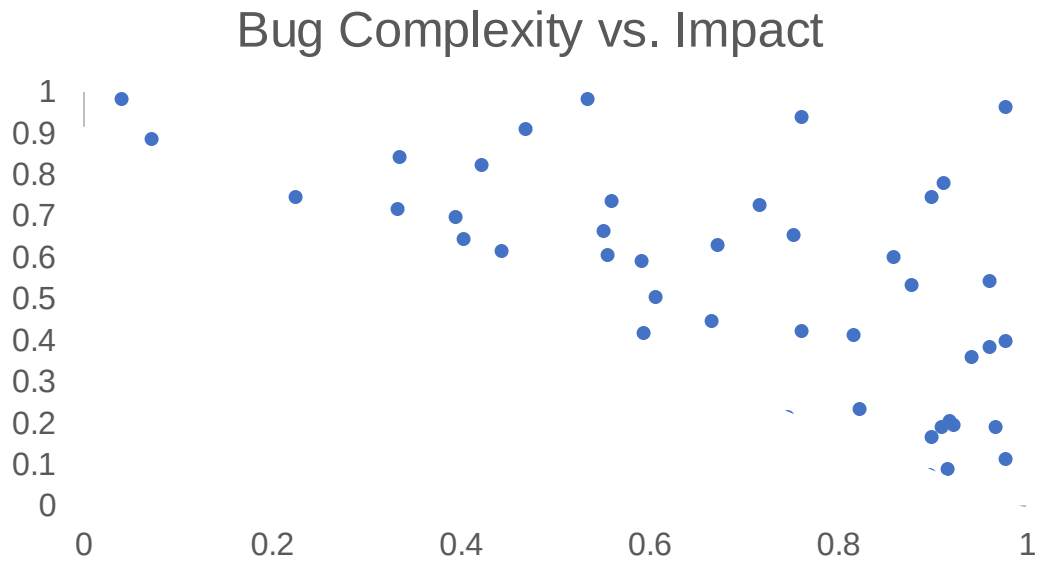


- If you look only at the coffee consumption → cancer relationship, you can get very misleading results
- Smoking is a confounder

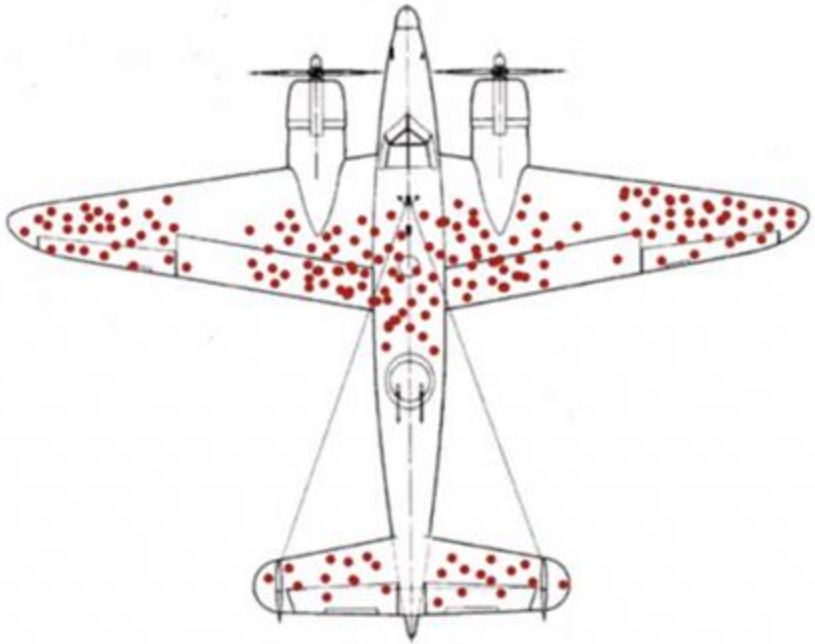
Spurious Correlations: Berkson's Paradox



Spurious Correlations: Berkson's Paradox



Survivorship bias



Measurement reliability

- Extent to which a measurement yields similar results when applied multiple times
- Goal is to reduce uncertainty, increase consistency
- Example: Performance
  - Time, memory usage
  - Cache misses, I/O operations, instruction execution count, etc.
- Law of large numbers
  - Taking multiple measurements to reduce error
  - Trade-off with cost

Outline

- Measurements and Metrics
- How to use measurements and metrics?
- Case study: Autonomous Vehicle Software
- Risks and challenges
- **Metrics and incentives**

Goodhart's law: "When a measure becomes a target, it ceases to be a good measure."

http://dilbert.com/strips/comic/1995-11-13/

The New York Times

The Price of Wells Fargo's Fake Account Scandal Grows by $3 Billion

The bank reached a settlement with federal prosecutors and the Securities and Exchange Commission after abusing customers.

Share full article 199

Wells Fargo used fraud to open up fake accounts and force customers into services

Incentivizing Productivity

- What happens when developer bonuses are based on
  - Lines of code per day?
  - Amount of documentation written?
  - Low number of reported bugs in their code?
  - Low number of open bugs in their code?
  - High number of fixed bugs?
  - Accuracy of time estimates?

What you need to know

Metrics are important in Software Engineering

Apply goal-oriented approaches to software metrics

Provide examples of metrics for software qualities and process

Understand limitations and dangers of decisions and incentives based on measurements