

Metrics and Measurement

17-313 Fall 2026

Foundations of Software Engineering

<https://cmu-17313q.github.io>

Eduardo Feo-Flushing

Administrivia

- Project 1B due Sunday, Sep. 6, 11:59 PM
- Reading for Tuesday
 - Posted on the course website (as usual)
 - Reading Quiz due before the lecture (check Gradescope)
 - Optional viewing: “Flight/Risk” on Amazon Prime, “Downfall: the case against Boeing” on Netflix
- Slack
 - Join the **#p1** channel
 - Much better than email

Today's Learning Goals

- Explain the importance of measurement and metrics in Software Engineering
- Provide examples of metrics for software qualities and processes
- Apply a goal-based framework 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
 -

NS New Scientist

Ants use pedometers to find home

An experiment that involves attaching stilts to ants' legs reveals that the insects somehow keep a record of how many steps they take.

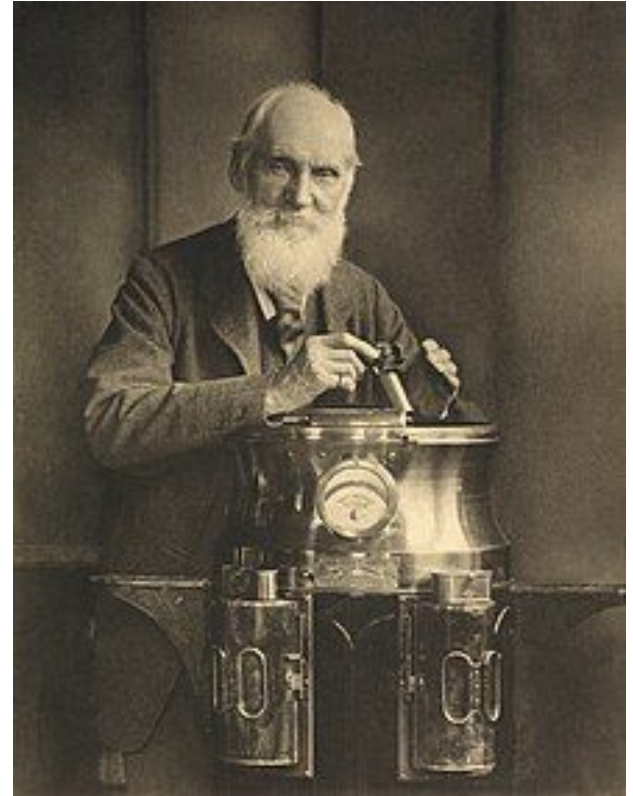
Jun 29, 2006



*“To measure is to know;
if you cannot measure it,
you cannot improve it”*

William Thomson, Lord Kelvin

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



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

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

Outline

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

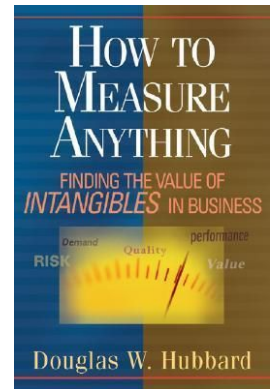
What is Measurement?

- **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?”

What is Measurement?

A more realistic definition

- A quantitatively expressed reduction of uncertainty based on one or more observations. – Hubbard, “How to Measure Anything ...”



Entity

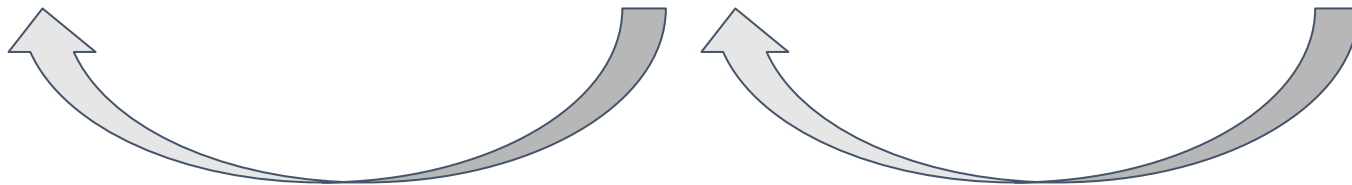
Object or
Process

**Quality
Attribute**

Quality of
Interest

Measurement

Method to obtain a
number or a symbol



What entities do 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
 - Performance
 - Employee satisfaction
 - Communication and collaboration
 - Efficiency and flow
 - Satisfaction with engineering system
- Customers
 - Satisfaction
 - Ease of use
 - Feature usage

Non-trivial qualities

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

Everything is measurable

- If X is something we care about, then X, by definition, must be detectable.
- If X is detectable, then it must be detectable in some amount.
- If we can observe it in some amount, then it must be measurable.

Douglas Hubbard, How to Measure Anything, 2010

Examples: Code Complexity

Measuring Code Complexity

- Lines of Code
 - Normalizing Lines of Code
 - Normalization by Language
- Halstead Volume
- Cyclomatic Complexity
- Object-Oriented Metrics

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)

Normalizing Lines of Code

```
/* How many lines of code is this? */
```

```
for (  
    i = 0;  
    i < 100;  
    i += 1  
    ) {  
    printf("hello");  
}
```

```
for (i = 0; i < 100; i += 1) printf("hello"); /* How many lines of code is this? */
```

- Ignore comments and empty lines
- Ignore lines < 2 characters
- Pretty print source code first
- Count statements (logical lines of code)
- See also: cloc

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();  
}
```

$M = \text{edges of CFG} - \text{nodes of CFG} + 2 \times \text{connected components}$

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

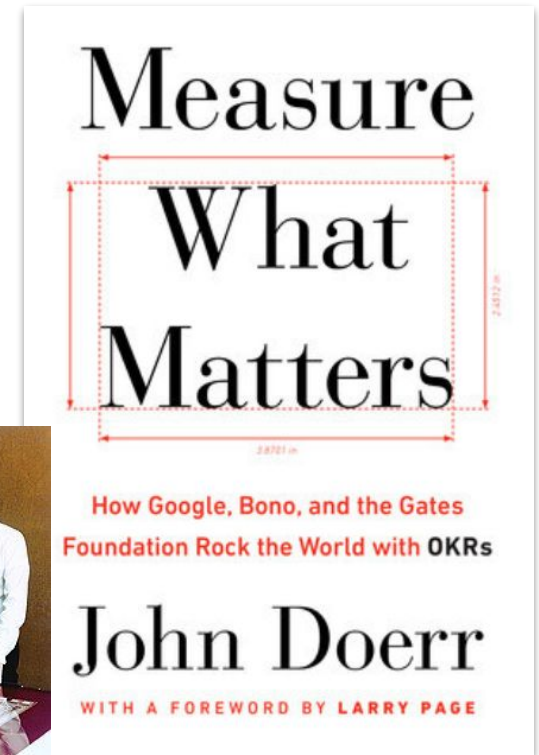
Outline

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

Goal-based frameworks

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

Measurement must be defined in a top-down fashion.



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

The GQM framework

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. Basili¹ Gianluigi Caldiera¹ H. Dieter Rombach²

⁽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:

(PIOV)

Evaluate the **effectiveness** of the **organization's coding standard** from the **team's** perspective

Questions:

How comprehensible is the coding standard?

What is the impact of the coding standard on the efficiency and productivity of the team?

Metrics:

Survey results measuring team members' understanding

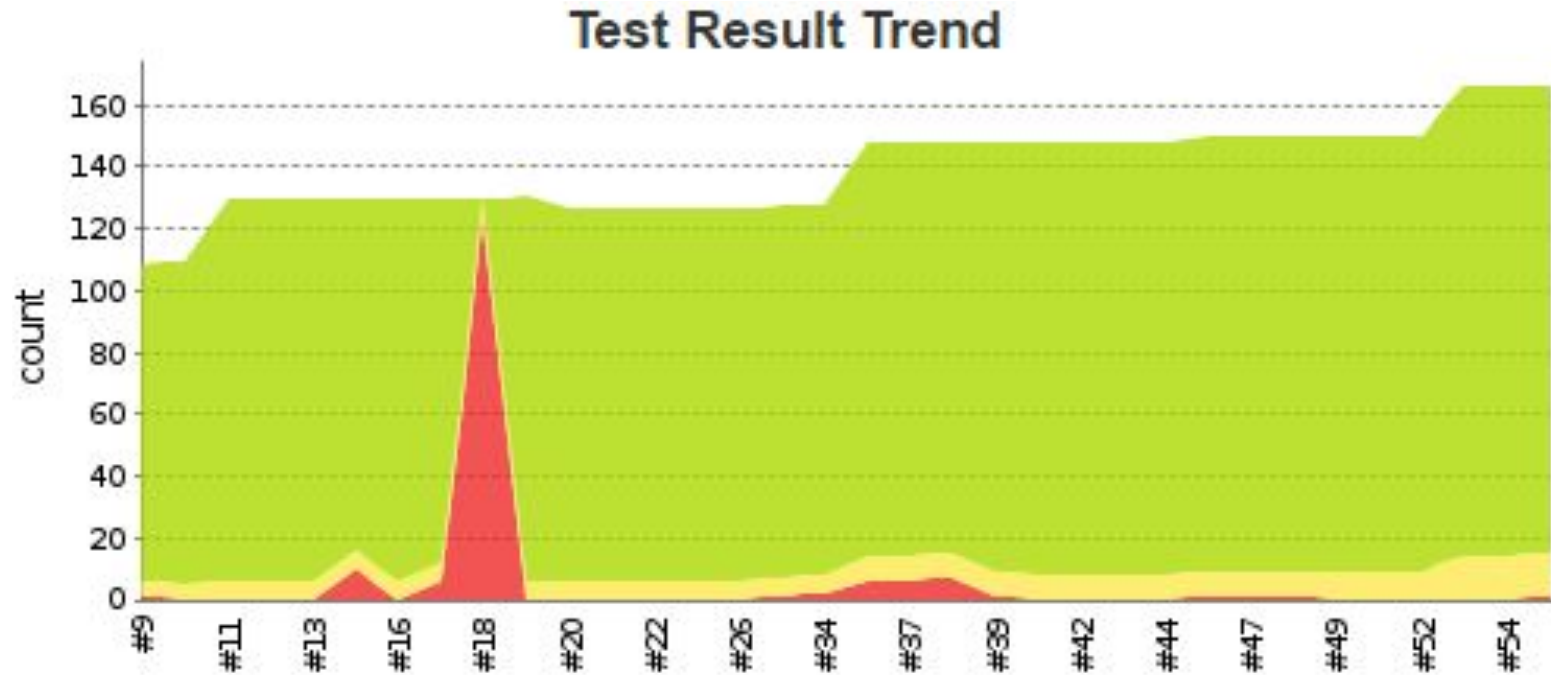
Number of revisions required to achieve standard compliance

Code size: LOC, number of classes, number of functions

Measurement for Decision Making

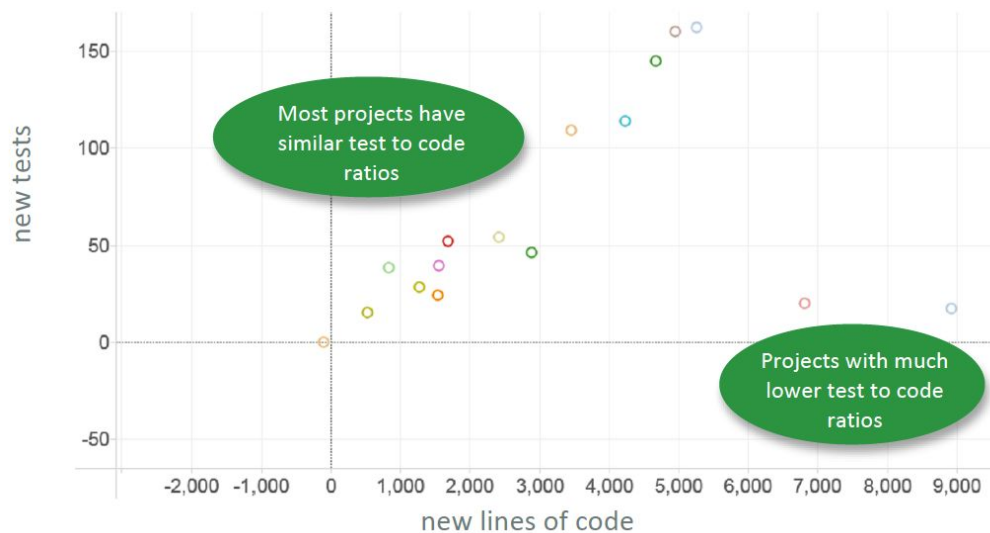
- Fund project?
- More testing?
- Fast enough? Secure enough?
- Code quality sufficient?
- Which feature to focus on?
- Developer bonus?
- Time and cost estimation? Predictions reliable?

Trend analysis



Benchmarking against baselines or comparable projects

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



Outline

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

Case study: Autonomous Vehicles

AV Software is _____



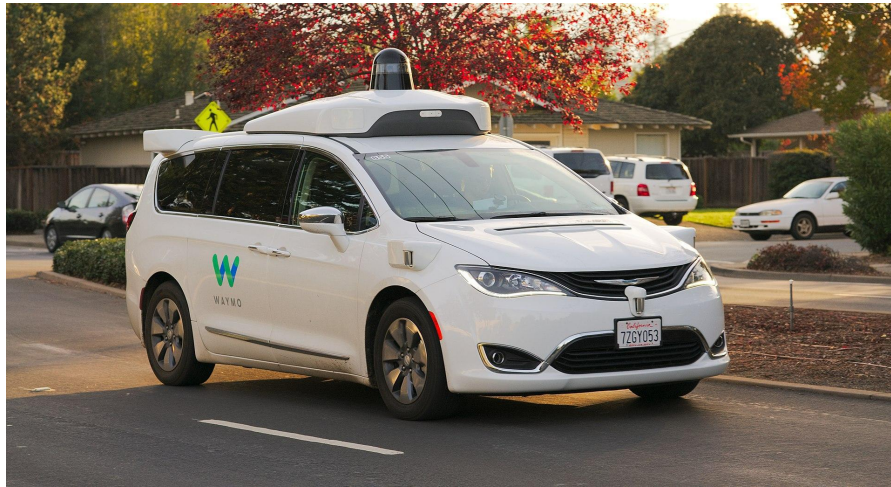
Let's apply the Goal-Question-Metric framework to explore various aspects of AV software



AV Software is safe



Goal: **Evaluate and enhance** the **safety** of the **autonomous vehicle system** from the perspectives of **safety engineers**.



Note that
we followed
PIOV to
define our
goal

Goal: **Evaluate and enhance** the **safety** of the **autonomous vehicle system** from the perspectives of **safety engineers**.

- How extensively are the critical safety functions of the vehicle's software covered by automated tests?
- How accurate are the vehicle's machine learning models in detecting and classifying road objects and obstacles?
- How consistently does the vehicle maintain safe and effective operation across different driving scenarios?



what we need to answer to know
whether our goal is met

(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

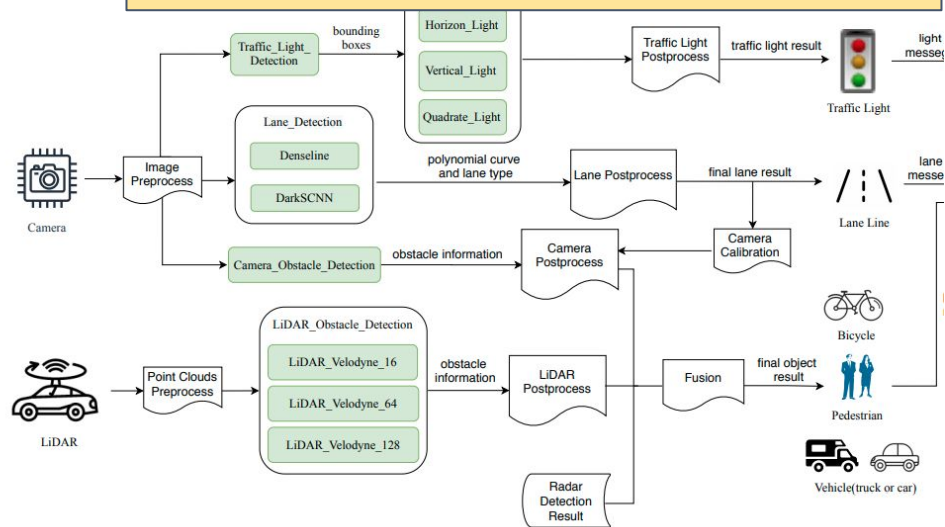
Possible measurement
How extensively are the critical safety functions of the vehicle's software covered by automated tests?

```
:
: 1698 : const TrajectoryPoint& StGraphData::init_point() const { return init_point; }
:
: 2264 : const SpeedLimit& StGraphData::speed_limit() const { return speed_limit; }
:
: 212736 : double StGraphData::cruise_speed() const {
[- +]: 212736 : return cruise_speed_ > 0.0 ? cruise_speed : FLAGS_default_cruise_speed;
:
:
: 1698 : double StGraphData::path_length() const { return path_data_length; }
:
: 1698 : double StGraphData::total_time_by_conf() const { return total_time_by_conf; }
:
: 1698 : planning_internal::STGraphDebug* StGraphData::mutable_st_graph_debug() {
: 1698 : return st_graph_debug;
:
:
: 566 : bool StGraphData::SetSTDivableBoundary(
: const std::vector<std::tuple<double, double, double>>& s_boundary,
: const std::vector<std::tuple<double, double, double>>& v_obs_info) {
[ + -: 566 : if (s_boundary.size() != v_obs_info.size()) {
: return false;
[ + +]:
: 40752 : for (size_t i = 0; i < s_boundary.size(); ++i) {
: 80372 : auto st_bound_instance = st_divable_boundary_.add_st_boundary();
: 160744 : st_bound_instance->set_t(std::get<0>(s_boundary[i]));
: 120558 : st_bound_instance->set_s_lower(std::get<1>(s_boundary[i]));
: 120558 : st_bound_instance->set_s_upper(std::get<2>(s_boundary[i]));
[- +]: 40186 : if (std::get<1>(v_obs_info[i]) > -kObsSpeedIgnoreThreshold) {
: 0 : st_bound_instance->set_v_obs_lower(std::get<1>(v_obs_info[i]));
:
:
[ + +]: 40186 : if (std::get<2>(v_obs_info[i]) < kObsSpeedIgnoreThreshold) {
: 50254 : st_bound_instance->set_v_obs_upper(std::get<2>(v_obs_info[i]));
:
:
: }
```

(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.

Possible measurement
How accurate are the vehicle's machine learning models in detecting and classifying road objects and obstacles?



Source: Peng et al. ESEC/FSE'20

(3) Failure Rate

- Crash and fatality rates
- Per 1,000 rides, per million miles, per month (in the news)

Possible measurement
How consistently does the vehicle maintain safe and effective operation across different driving scenarios?

TRANSP / WAYMO / TECH

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



Image: Allen J. Schaben / Lo

/ 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](#)



'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

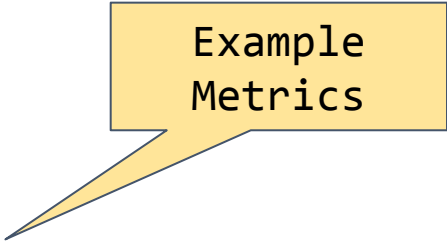


A Cruise autonomous taxi in San Francisco.

Participation Activity

- Apply the **Goal-Question-Metric** framework to **explore one aspect 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 on it.
- **Each person submits individually**
- You can work in groups of 2-3.
- 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
Metrics

Example

Goal: Ensure energy efficiency and sustainability from the perspectives of 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.

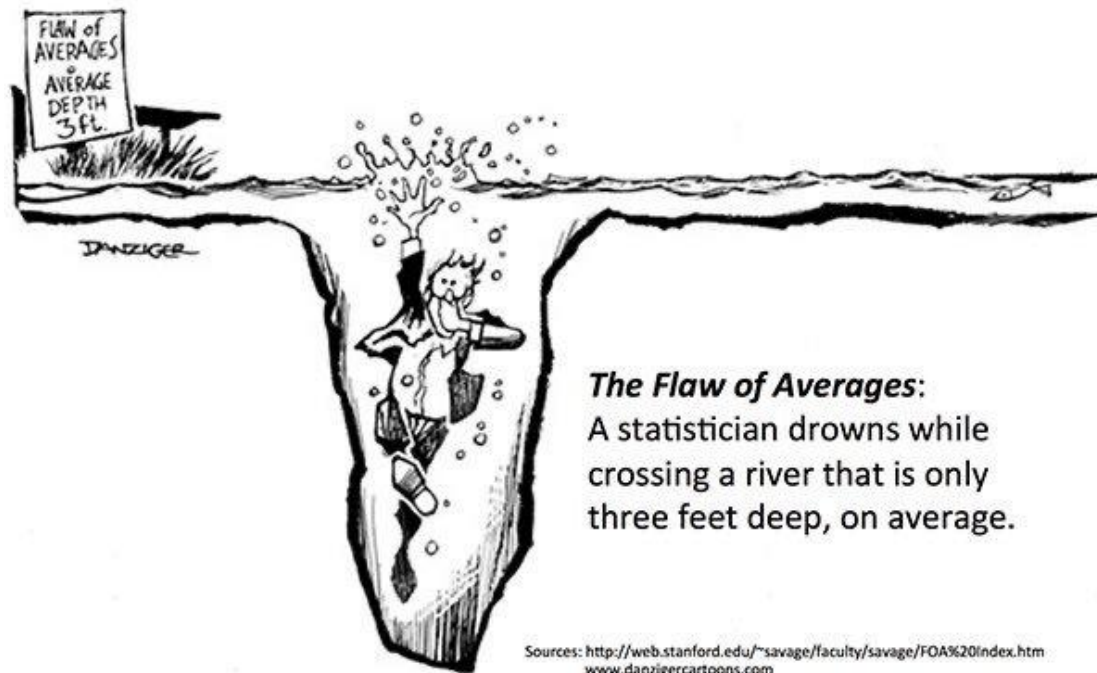
Question 2: How efficient is the battery management system?

Metrics: Battery life, number of charge cycles

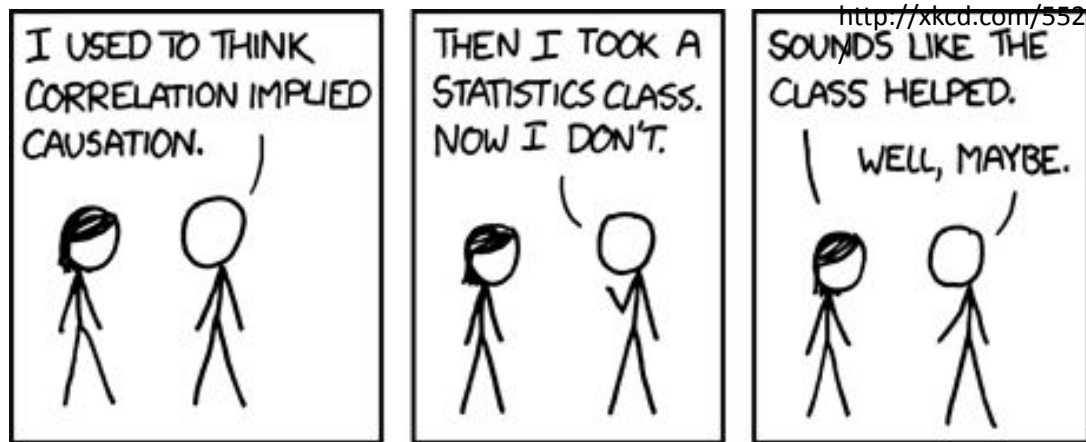
Outline

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

Bad statistics: What could possibly go wrong?



Making inferences



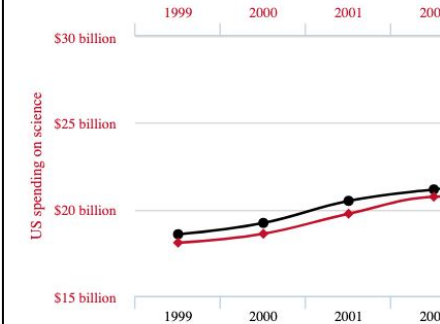
To support a causal claim:

- Establish a plausible **causal mechanism** using **domain knowledge**
- Show that the variables are associated and that the **cause precedes the effect**
- **Rule out alternative explanations** and **confounding variables** through study design or statistical controls
- **Replicate** the finding across data, studies, or contexts

Spurious Correlations

US spending on science, space, and technology
correlates with
Suicides by hanging, strangulation and suffocation

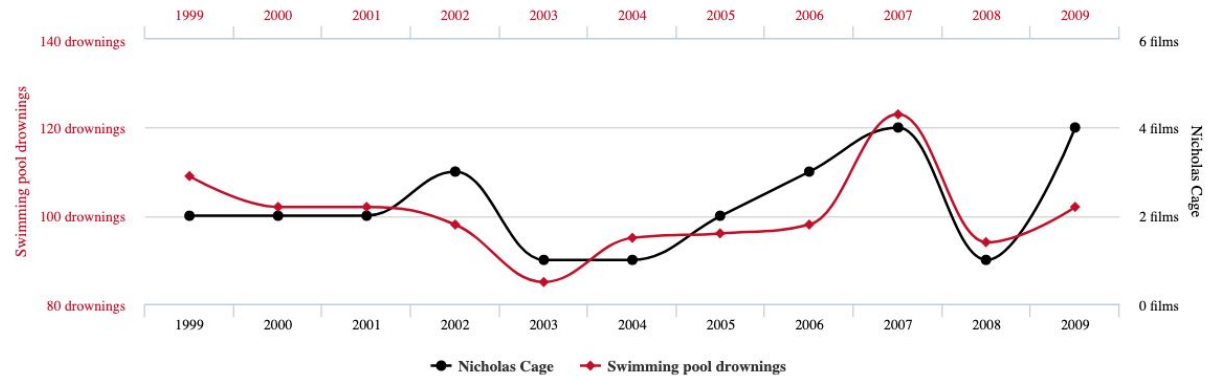
Correlation: 99.79% ($r=0.99789126$)



Data sources: U.S. Office of Management and Budget and Centers for Disease Control & Prevention

Number of people who drowned by falling into a pool
correlates with
Films Nicolas Cage appeared in

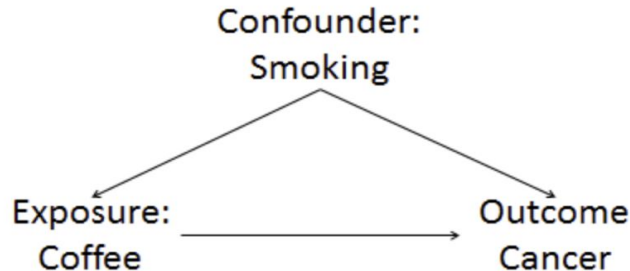
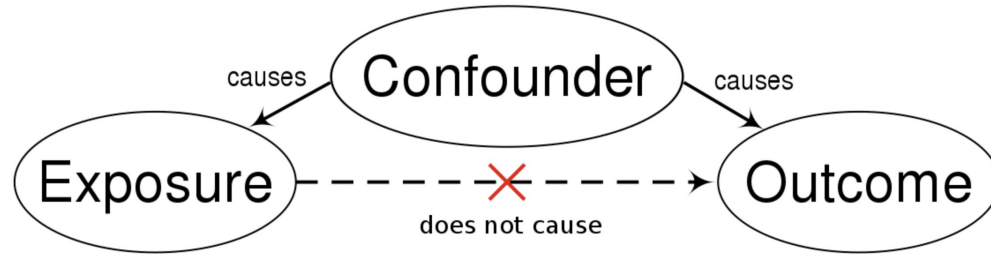
Correlation: 66.6% ($r=0.666004$)



Data sources: Centers for Disease Control & Prevention and Internet Movie Database

tylervigen.com

Confounding variables



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

Coverage is not strongly correlated with test suite effectiveness

Authors:  [Laura Inozemtseva](#),  [Reid Holmes](#) [Authors Info & Affiliations](#)

ICSE 2014: Proceedings of the 36th International Conference on Software Engineering • May 2014 • Pages 435–445 • <https://doi.org/10.1145/2568225.2568271>

“We found that there is a low to moderate correlation between coverage and effectiveness when the number of test cases in the suite is controlled for.”

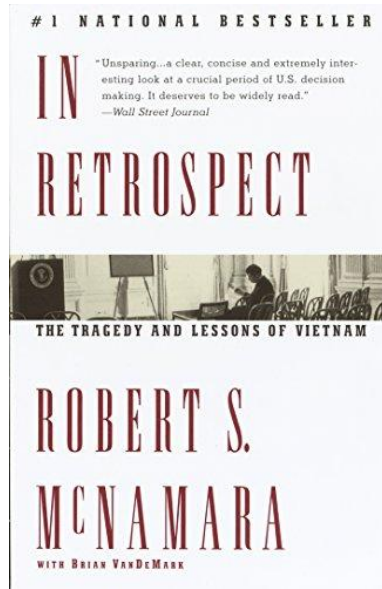
61

Most studies did not account for the confounding influence of test suite size

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
- Repeated measurements can reduce random error, but not systematic bias

McNamara fallacy



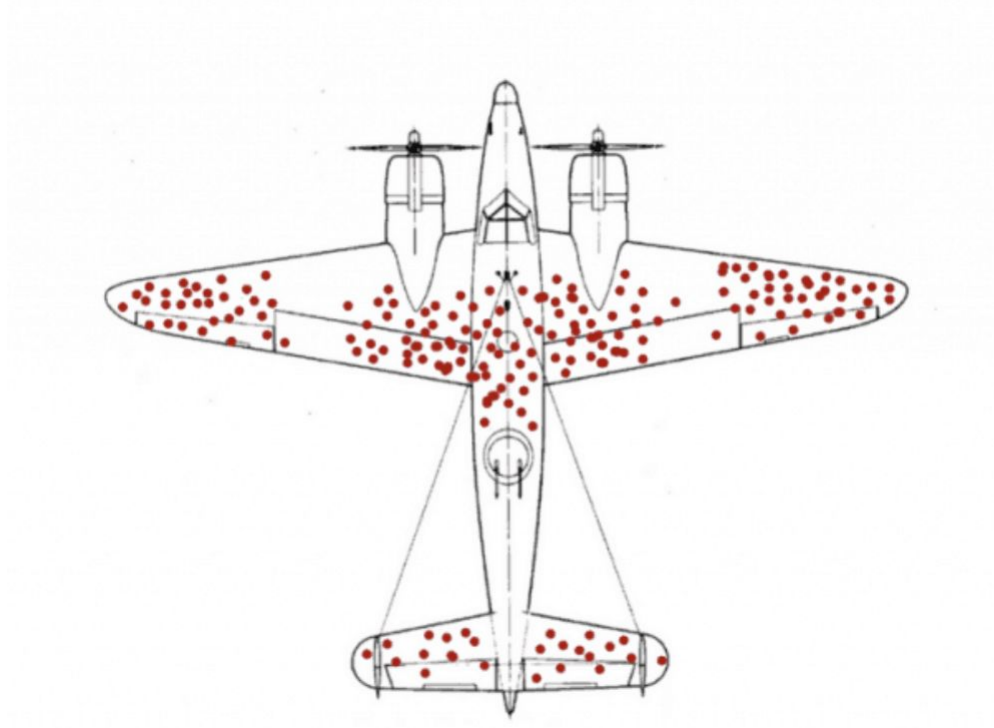
McNamara fallacy

- 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/>

Survivorship bias



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 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

CIO JOURNAL

Why Some Companies Say AI ‘Tokenmaxxing’ Is Key to Survival

Leaderboards that rank individual employees’ AI token usage have garnered recent criticism. But some say they’re vital for adapting to the AI age.

By **Isabelle Bousquette**

April 14, 2026 8:00 am ET

   Listen (1 min) 



by **Grant Gross**
Senior Writer

Tokenmaxxing: When AI adoption metrics go bad

Feature

Jun 5, 2026 • 6 mins

Without other metrics, token use leaderboards can create bad incentives, encouraging workers to maximize token use without worrying about cost, true productivity, or outcomes.

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 a goal-oriented approach to software metrics



Provide examples of metrics for software qualities and processes



Understand limitations and dangers of decisions and incentives based on measurements