

CEN 5016: Software Engineering

Fall 2024



University of
Central Florida

Dr. Kevin Moran

Week 2 - Class 1: Measurement & Metrics





- Course Schedule Posted
- Office Hours Decided (kind of)
 - Tuesday/Thursday 1:00pm-2:00pm (before class) Hybrid
 - Or by appointment
- Let me know if you are not on Ed Discussions
- Assignment 1, Getting started with Git, GitHub, and Typescript is due tonight at 11:59 pm
 - Use Megathread on Ed Discussions to ask questions
- Team-forming this week
 - Teams of 3 students
 - Look out for a post on Ed Discussions
- Assignment 2 out Thursday

Software Archeology & Anthropology



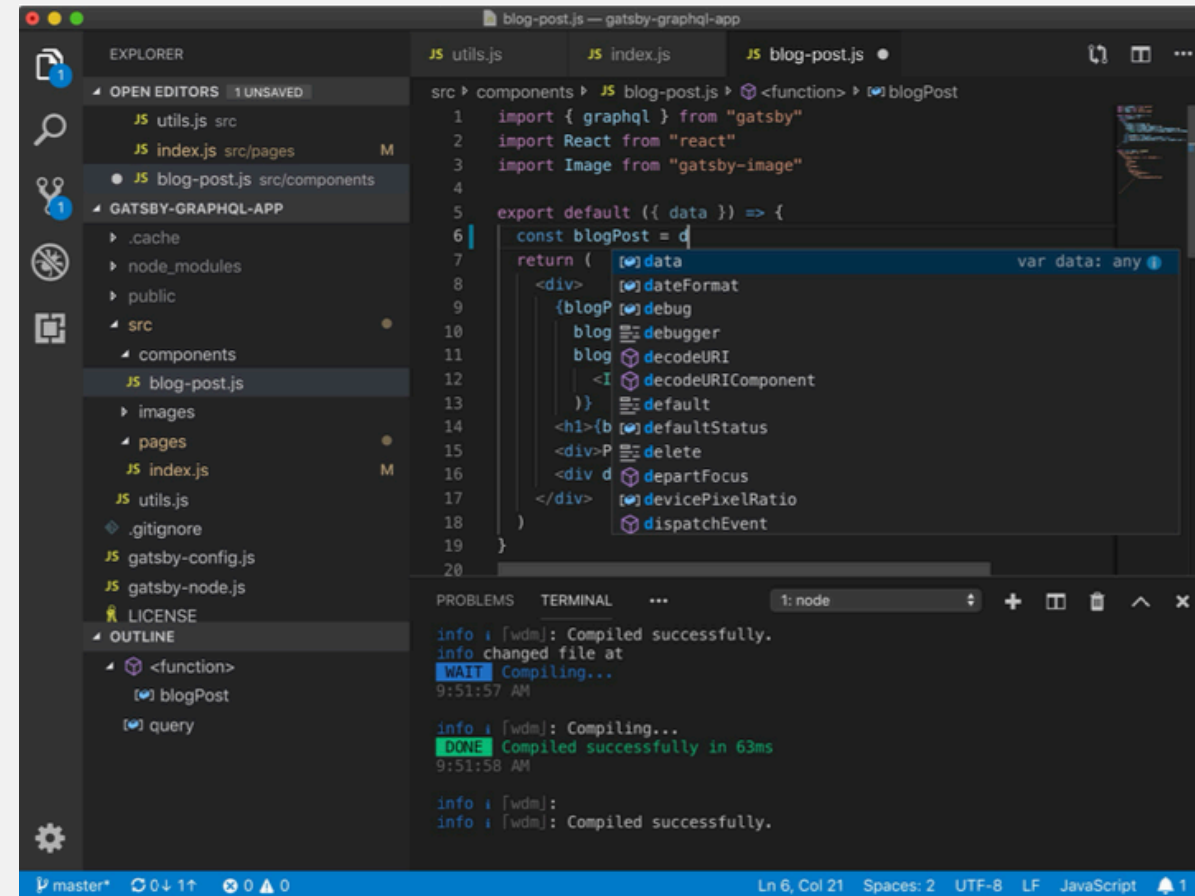
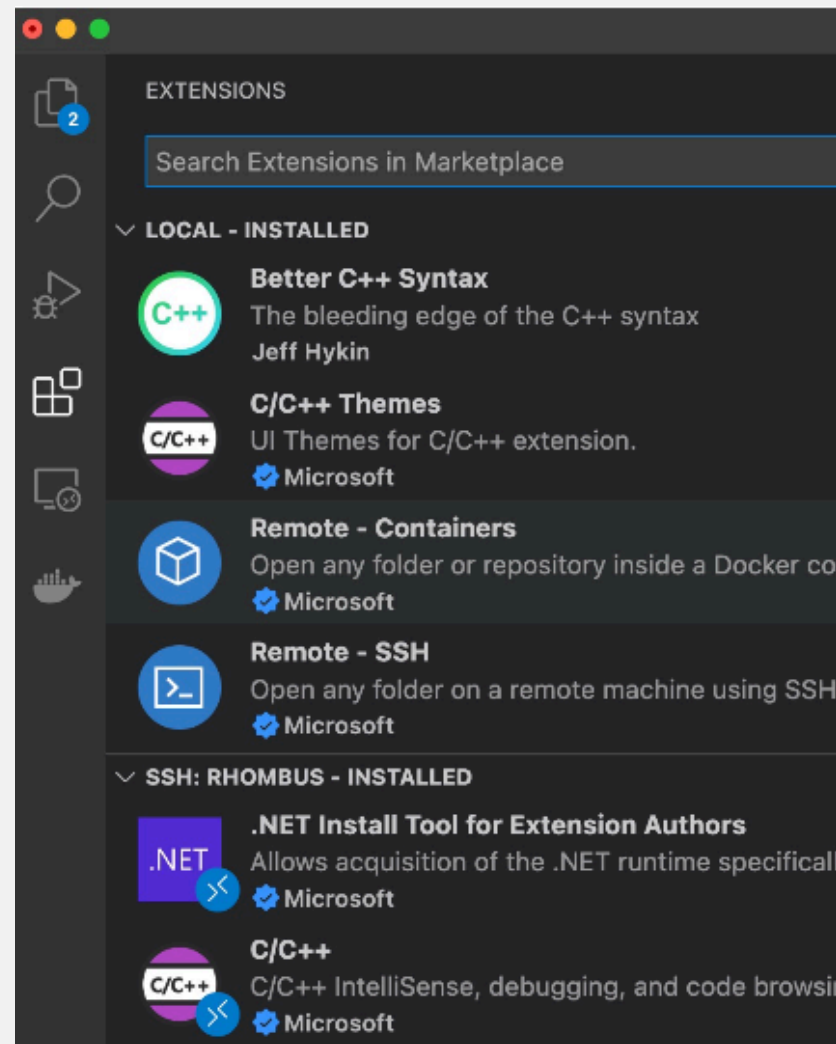
Creating a Model of Unfamiliar Code





- Basic needs:
 - Code/file search and navigation
 - Code editing (probes)
 - Execution of code, tests
 - Observation of output (observation)
- At the command line: grep and find! (Google for tutorials)
- Many choices here on tools! Depends on circumstance.
 - grep/find/etc.
 - Knowing Unix tools is invaluable
 - A decent IDE
 - Debugger
 - Test frameworks + coverage reports
 - Google (or your favorite web search engine)
 - ChatGPT or LaMA

Static Information Gathering: Use an IDE!



Consider Documentation and Tutorials Judiciously



- Great for discovering entry points!
- Can teach you about general structure, architecture (more on this later in the semester)
- Often out of date.
- As you gain experience, you will recognize more of these, and you will immediately know something about how the program works
- Also: discussion boards; issue trackers

The screenshot shows the TypeScript Documentation website. The header is dark blue with the TypeScript logo and navigation links: Download, Docs, Handbook, Community, Playground, Tools. A search bar is in the top right. The main content area is dark grey and titled "TypeScript Documentation". It is divided into three columns:

- Get Started**: Quick introductions based on your background or preference. Links include: [TS for the New Programmer](#), [TypeScript for JS Programmers](#), [TS for Java/C# Programmers](#), [TS for Functional Programmers](#), and [TypeScript Tooling in 5 minutes](#).
- Handbook**: A great first read for your daily TS work. Links include: [The TypeScript Handbook](#), [The Basics](#), [Everyday Types](#), [Narrowing](#), [More on Functions](#), [Object Types](#), **Type Manipulation**, [Creating Types from Types](#), [Generics](#), [Keyof Type Operator](#), [Typeof Type Operator](#), [Indexed Access Types](#), and [Conditional Types](#).
- Reference**: Deep dive reference materials. Links include: [Utility Types](#), [Cheat Sheets](#), [Decorators](#), [Declaration Merging](#), [Enums](#), [Iterators and Generators](#), [JSX](#), [Mixins](#), [Namespaces](#), [Namespaces and Modules](#), [Symbols](#), [Triple-Slash Directives](#), and [Type Compatibility](#).

Discussion Boards and Issue Trackers



- Software is written by people.
- How can we talk to them?
- Fortunately, they probably aren't dead.
- So, you can report problems on GitHub.
- Or, ask them questions on StackOverflow.

The screenshot shows the Stack Overflow search results page for the query "typescript on mac". The page features a search bar at the top with the query "typescript on mac" and buttons for "Log in" and "Sign up". The search results are displayed in a list format, with the top result being "Cannot use JSX unless the '--jsx' flag is provided" by Mahmoud, which has 858 votes and was answered on Nov 23, 2020. Other results include "Can't install Typescript on mac [duplicate]" and "npm install doesn't install typescript (on mac)". The page also includes a sidebar with navigation links (Home, Questions, Tags, Users, Companies, Collectives, Labs, Teams) and a "Hot Network Questions" section on the right.



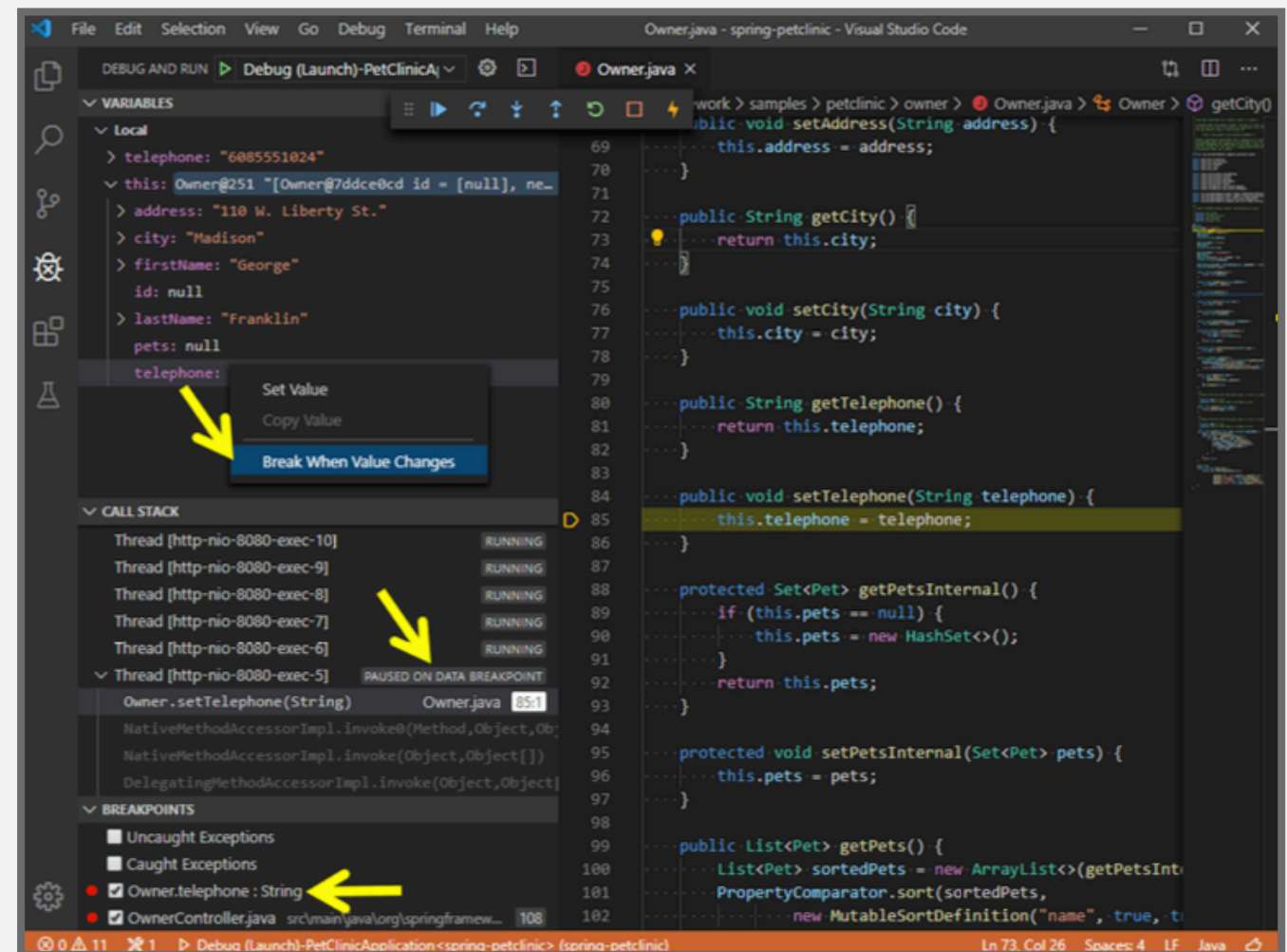
- Build it.
- Run it.
- Change it.
- Run it again.
- How did the behavior change?



Probes: Observe, Control, or “Lightly” Manipulate Execution



- print(“this code is running!”)
- Structured logging
- Debuggers
 - Breakpoint, eval, step through / step over
 - (Some tools even support remote debugging)
- Delete debugging
- Chrome Developer Tools



Step 0: Sanity Check Basic Model + Hypotheses

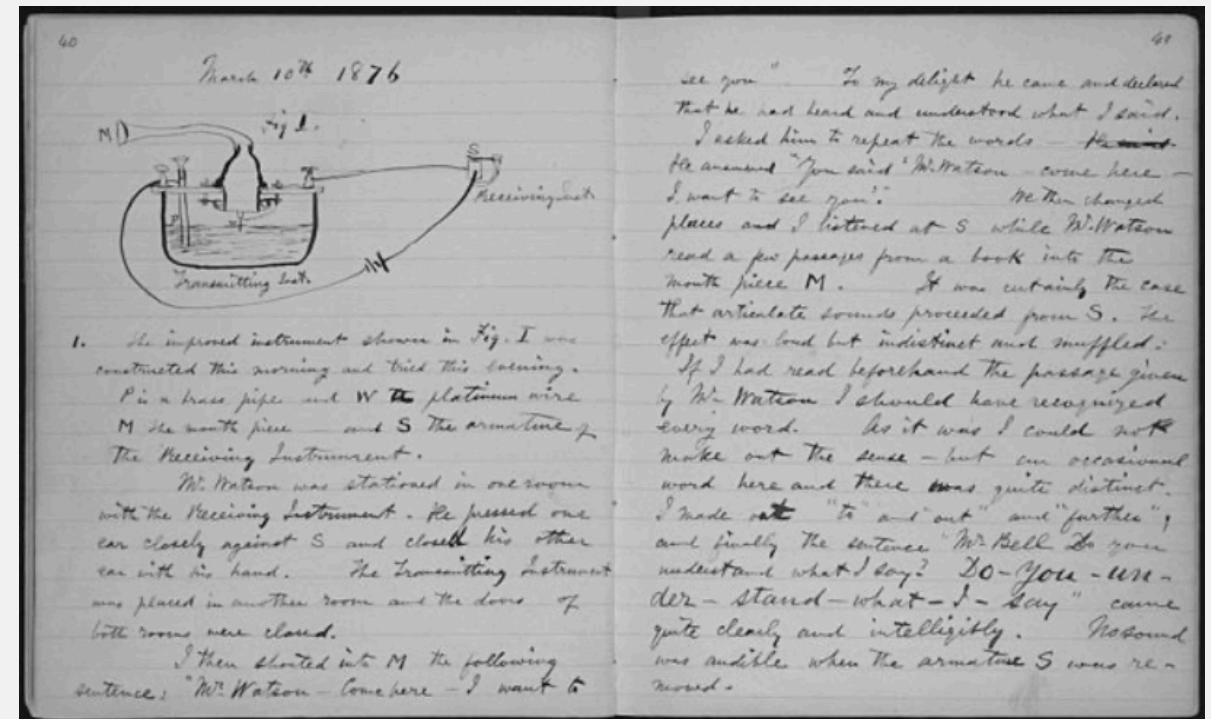


- *Confirm that you can build and run the code.*
 - Ideally both using the tests provided, and by hand.
- *Confirm that the code you are running is the code you built*
- *Confirm that you can make an externally visible change*
- *How? Where? Starting points:*
 - Run an existing test, change it
 - Write a new test
 - Change the code, write or rerun a test that should notice the change
- *Ask someone for help*

Document and Share Your Findings!



- Update README and docs
 - Or better: use a Developer Wiki
 - Use Mermaid for diagrams
- Screencast on Twitch
- Collaborate with others
- Include negative results, too!



Metrics & Measurement



Goals for Today



- Use measurements as a decision tool to reduce uncertainty
- Understand difficulty of measurement; discuss validity of measurements
- Provide examples of metrics for software qualities and process
- Understand limitations and dangers of decisions and incentives based on measurements



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

What does this mean?
How do we know?
-> Measurement & Metrics
are key concerns

Case Study: Autonomous Vehicles



Case Study: Autonomous Vehicles



- By what methods can we judge AV software quality (e.g., safety)?



Test Coverage

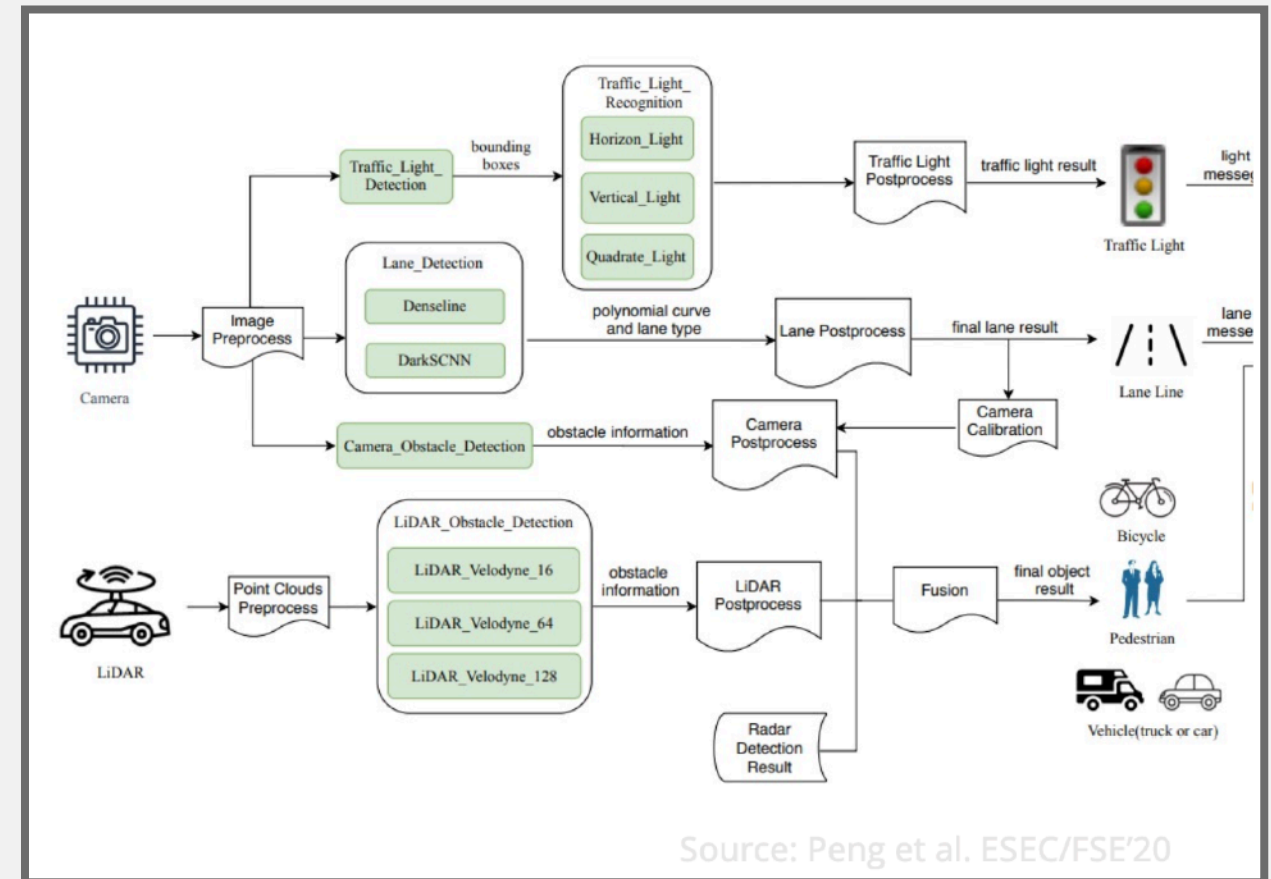


- Amount of code executed during testing.
- Statement coverage, line coverage, branch coverage, etc.
- E.g., 75% branch coverage -> 3/4 if-else outcomes have been executed

```
:  
: 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::SetSTDrivableBoundary(  
:     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_drivable_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]));  
:         }  
:     }  
:  
:     }
```



- 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.





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



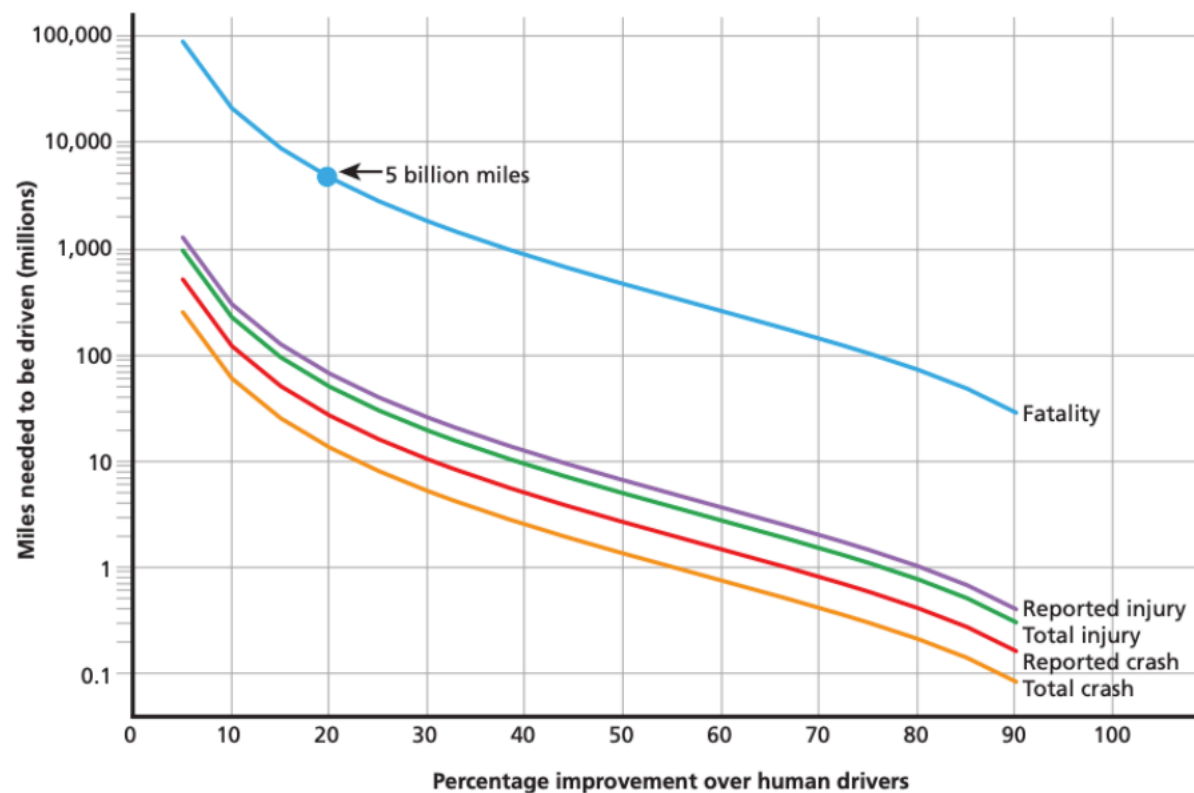


Driving to Safety

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

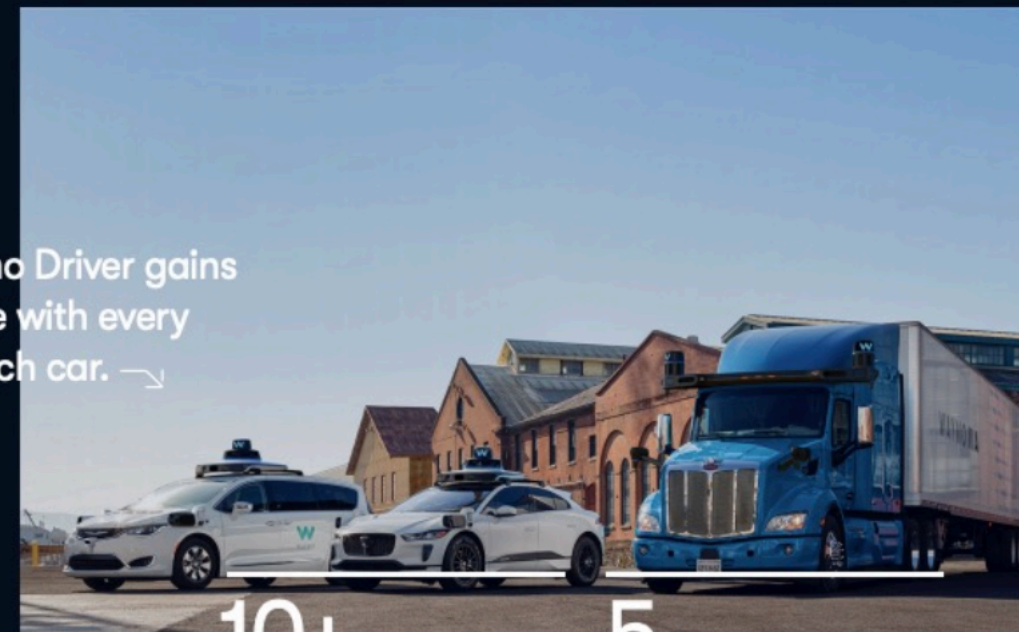
Nidhi Kalra, Susan M. Paddock

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



Building the World's Most Experienced Driver™

The Waymo Driver gains experience with every mile, in each car. ↗



10+

More than a Decade of Autonomous Driving in More than 10 States

5

Generations of Autonomously Driven Vehicles

15+

Billion Autonomously Driven Miles in Simulation

20+

Million Real-World Miles on Public Roads

Source: waymo.com/safety (September 2021)

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 ...”



- 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.”
- Metrics have been proposed for many quality attributes; may define own metrics

What Software Qualities Do We Care About?



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

What Process Qualities Do We Care About?



- On-time release
- Development speed
- Meeting efficiency
- Conformance to processes
- Time spent on rework
- Reliability of predictions
- Fairness in decision making
- Number of builds
- Code review acceptance rate
- Regulatory compliance
- Measure time, costs, actions, resources, and quality of work packages; compare with predictions
- Use information from issue trackers, communication networks, team structures, etc...

What People Qualities Do We Care About?



- **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



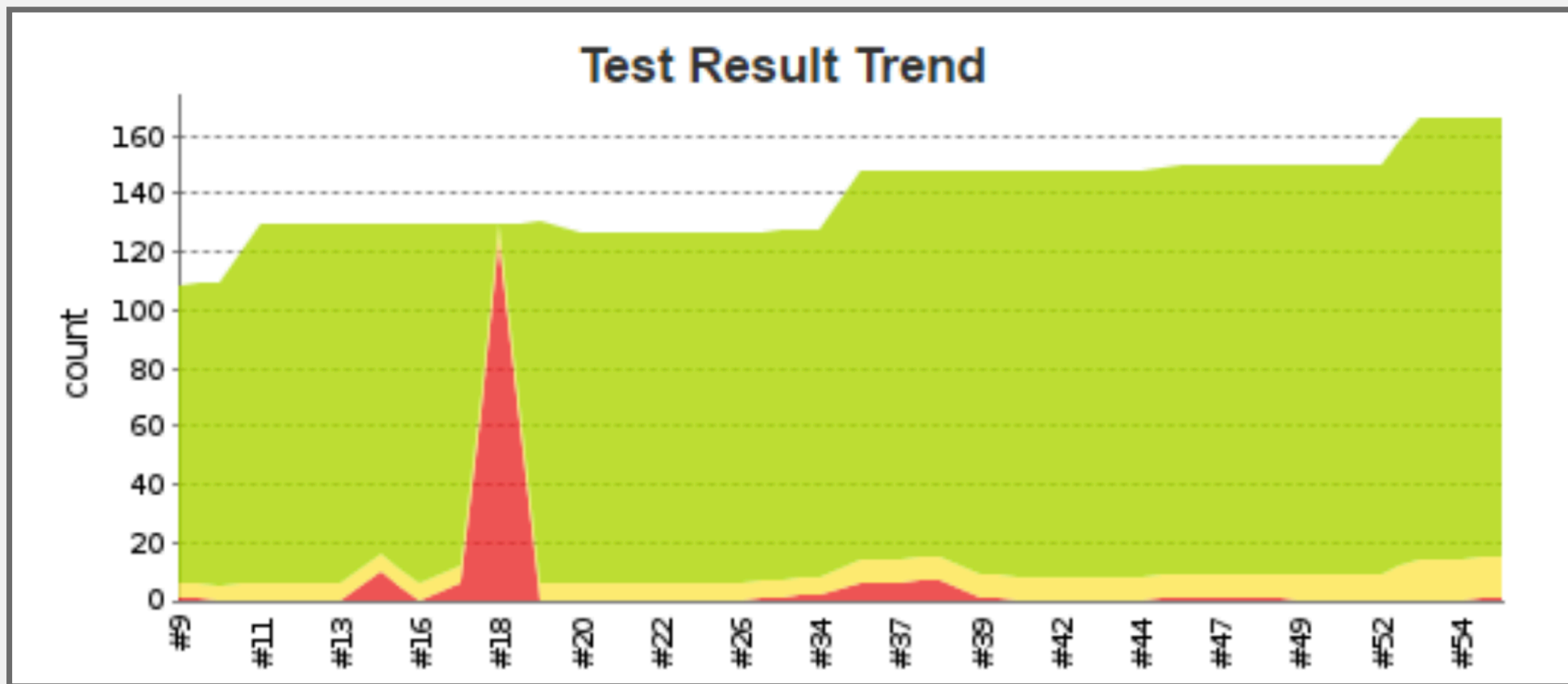
- If X is something we care about, then X, by definition, must be detectable.
 - How could we care about things like “quality,” “risk,” “security,” or “public image” if these things were totally undetectable, directly or indirectly?
 - If we have reason to care about some unknown quantity, it is because 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 21
- If we can observe it in some amount, then it must be measurable.

Why Measure?





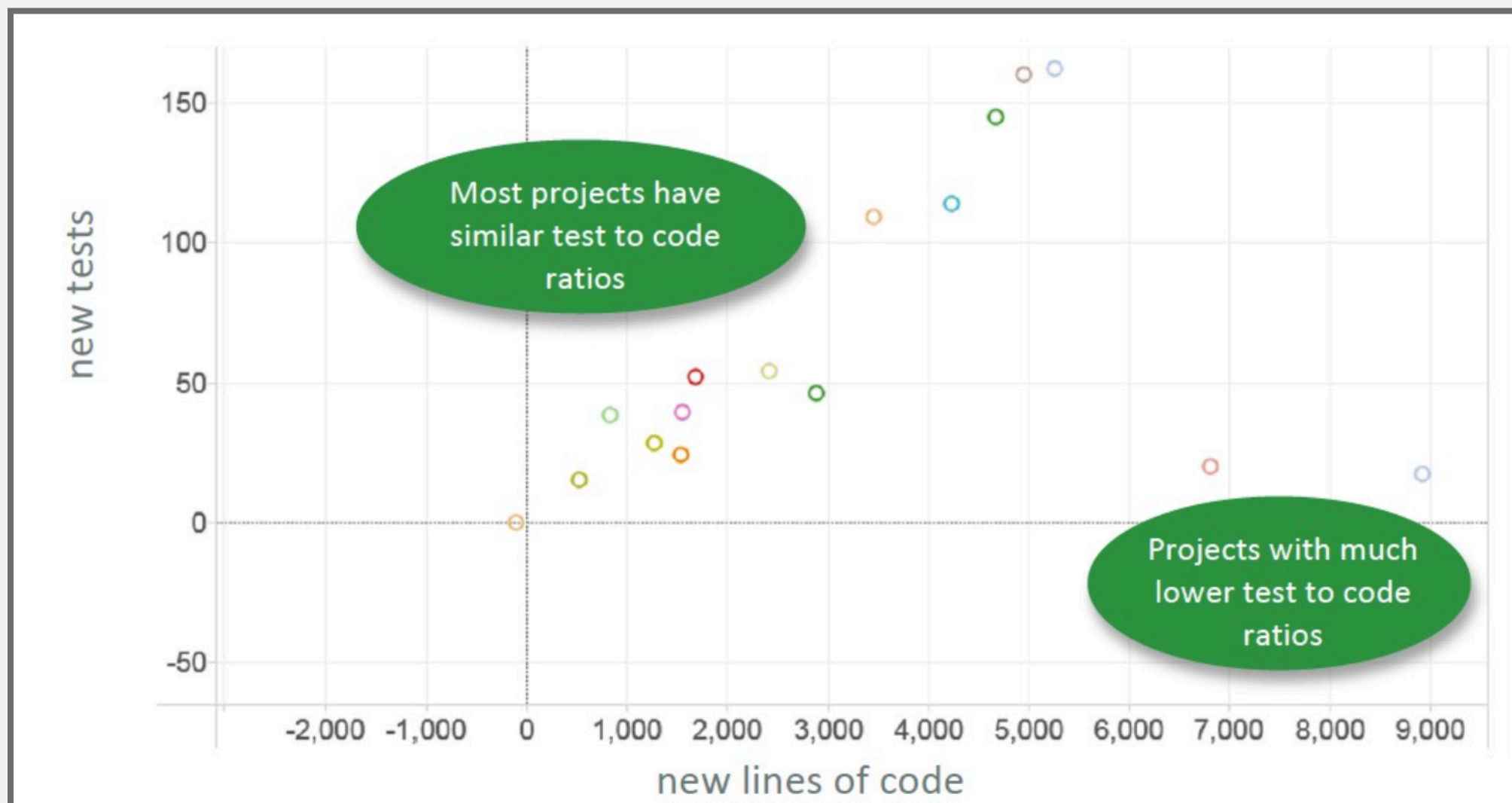
- Fund project?
- More testing?
- Fast enough? Secure enough?
- Code quality sufficient?
- Which feature to focus on?
- Developer bonus?
- Time and cost estimation? Predictions reliable?



Benchmarking Against Standards



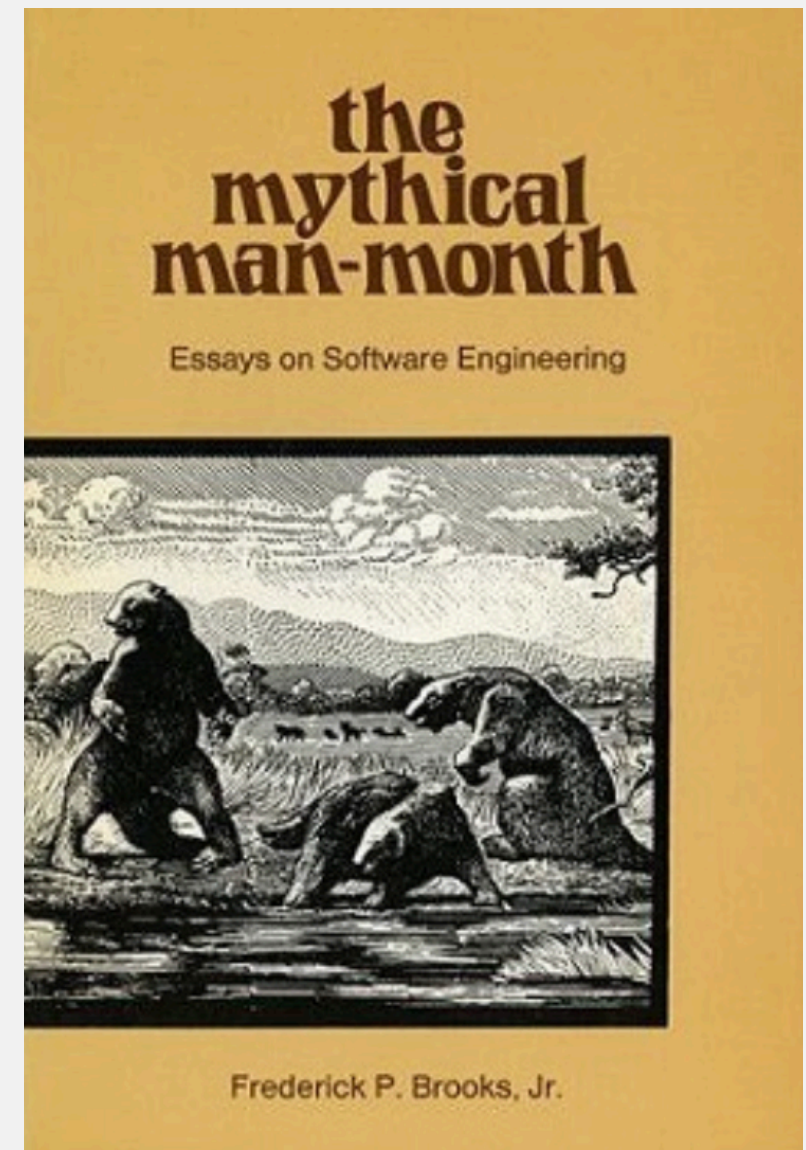
- Monitor many projects or many modules, get typical values for metrics
- Report deviations



Antipatterns in Effort Estimation



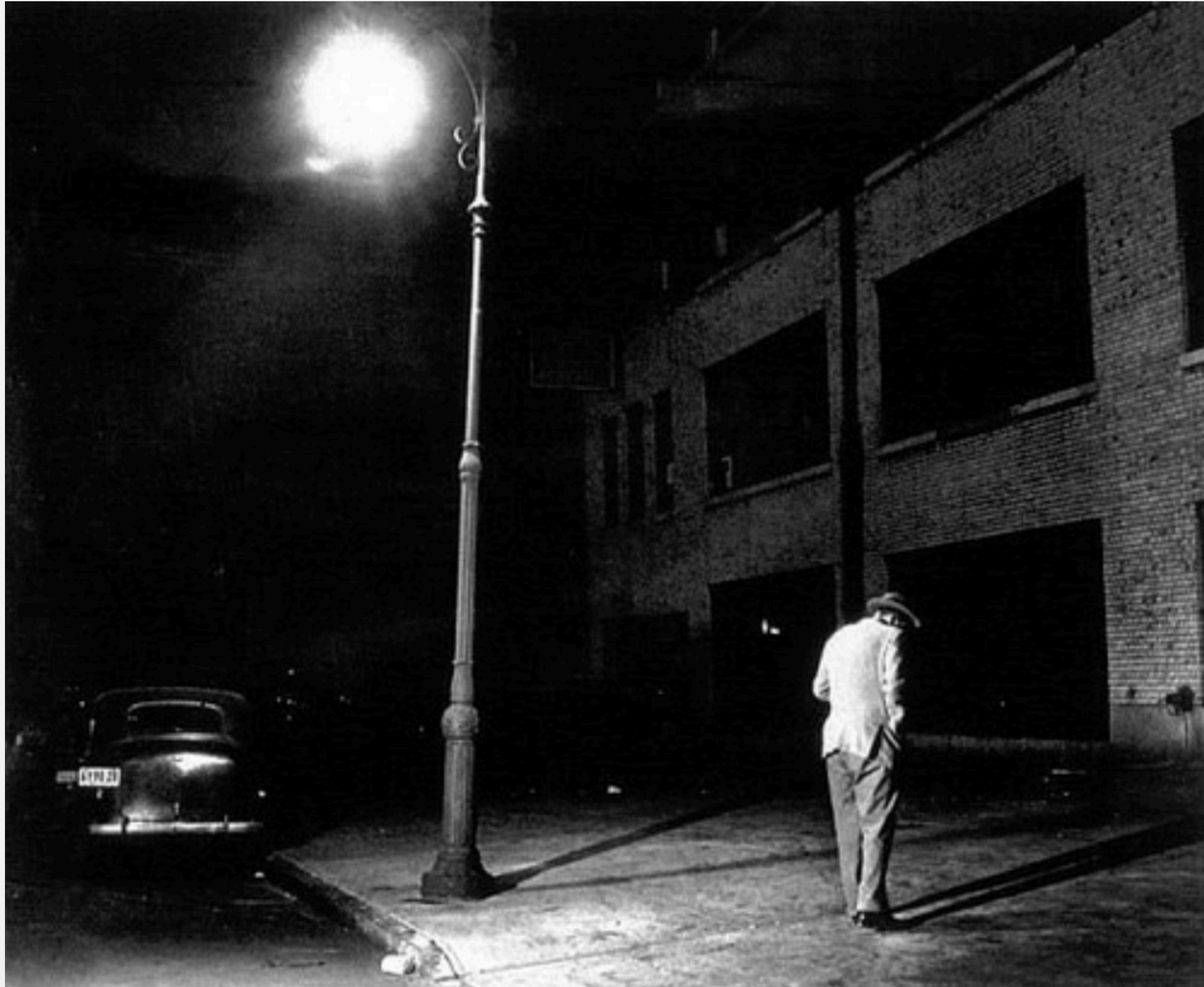
- IBM in the 60s: Would account in “person-months”
e.g. Team of 2 working 3 months = 6 person-months
- LoC ~ Person-months ~ \$\$
\$
- Brooks: “Adding manpower to a late software project [just] makes it later.”



Measurement is Difficult



The Streetlight Effect



The Streetlight Effect

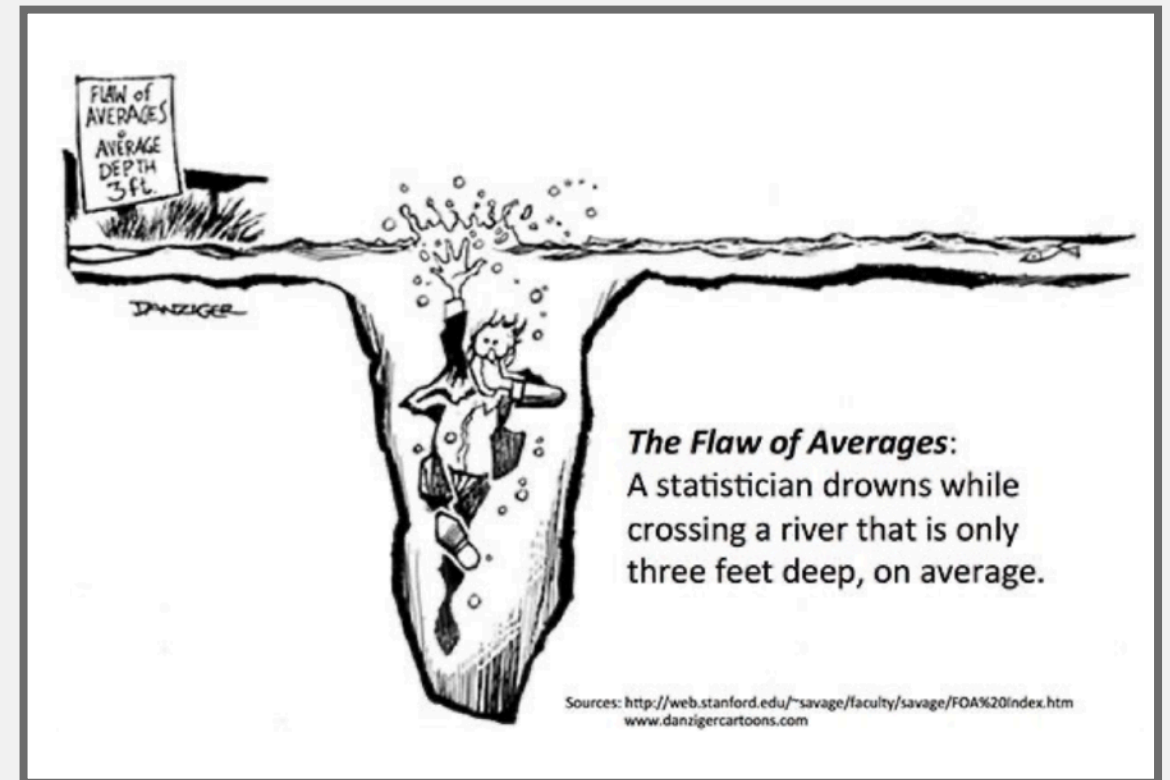


- A known observational bias.
- People tend to look for something only where it's easiest to do so.
- If you drop your keys at night, you'll tend to look for it under streetlights.

What could Possibly go Wrong?

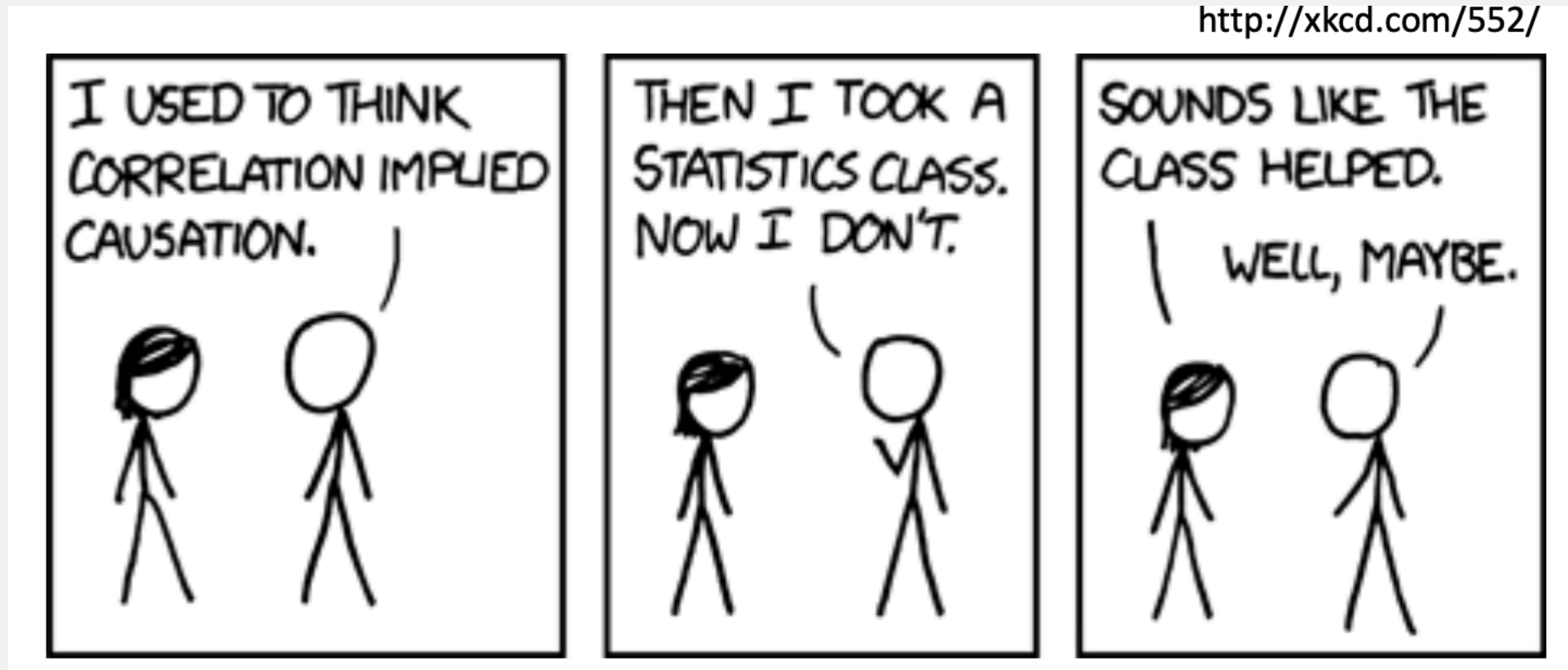


- Bad statistics: A basic misunderstanding of measurement theory and what is being measured.
- Bad decisions: The incorrect use of measurement data, leading to unintended side effects.
- Bad incentives: Disregard for the human factors, or how the cultural change of taking measurements will affect people.



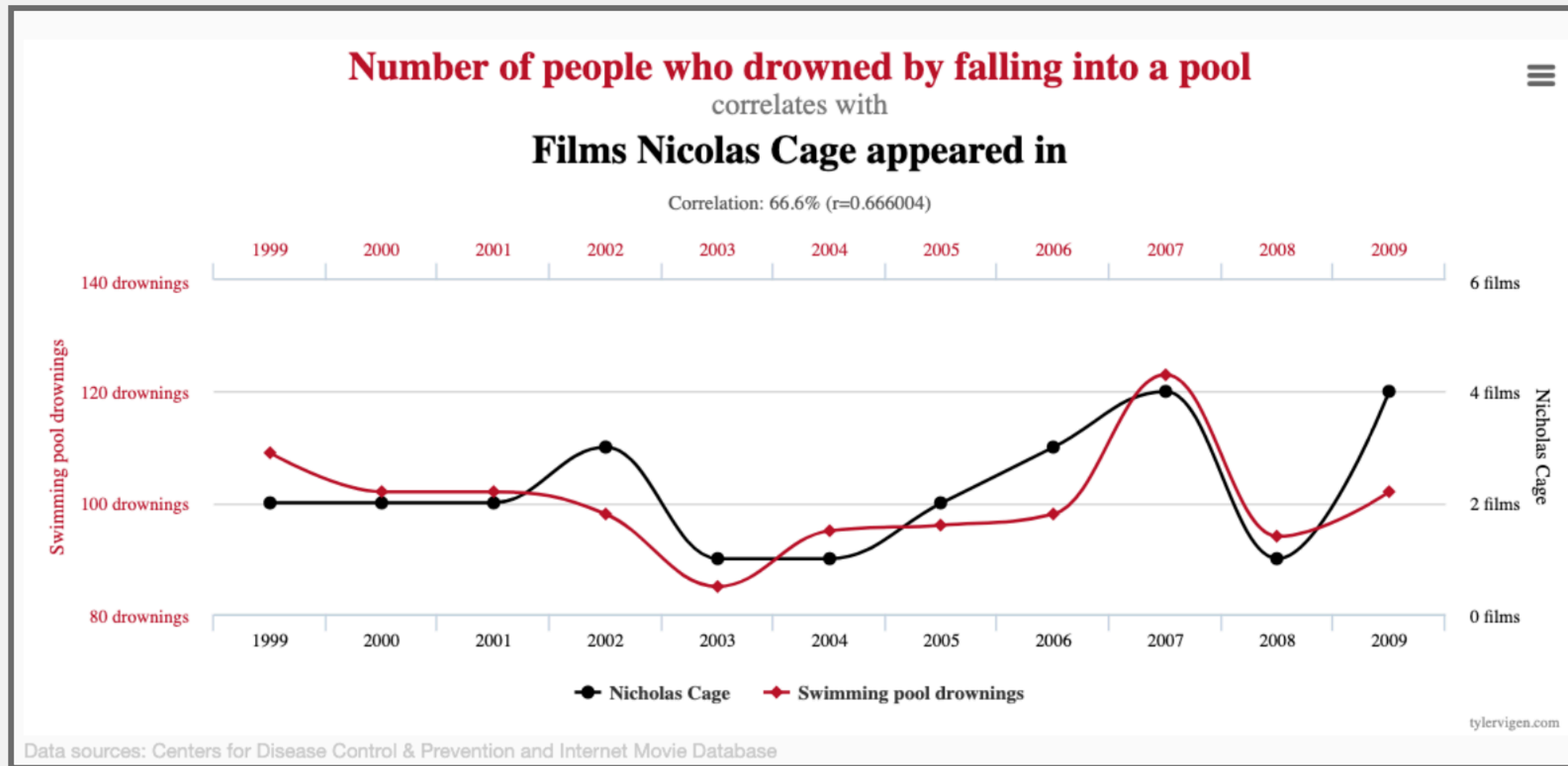


<http://xkcd.com/552/>

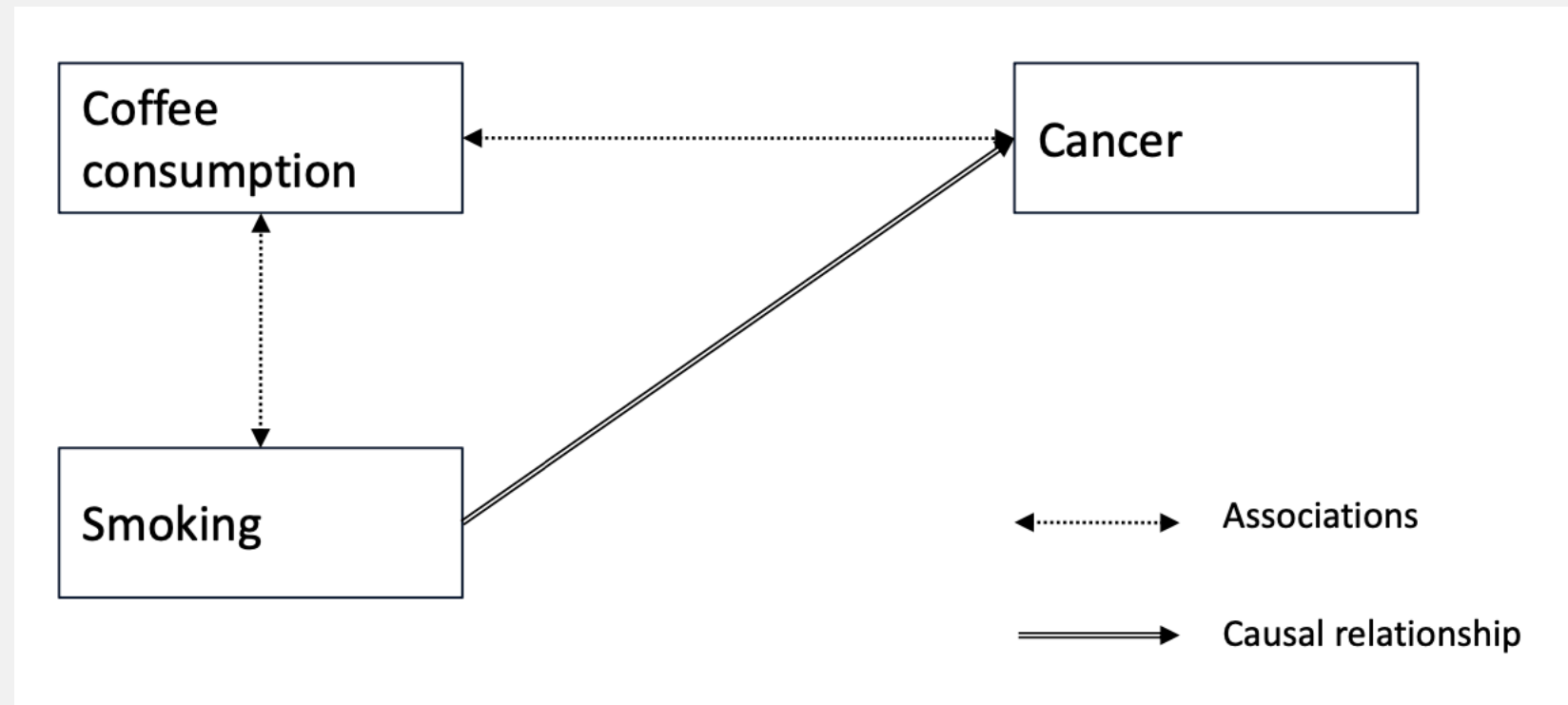


- To infer causation:
 - Provide a theory (from domain knowledge, independent of data)
 - Show correlation
 - Demonstrate ability to predict new cases (replicate/validate)

Spurious Correlations



Confounding Variables



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



RESEARCH-ARTICLE

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.”

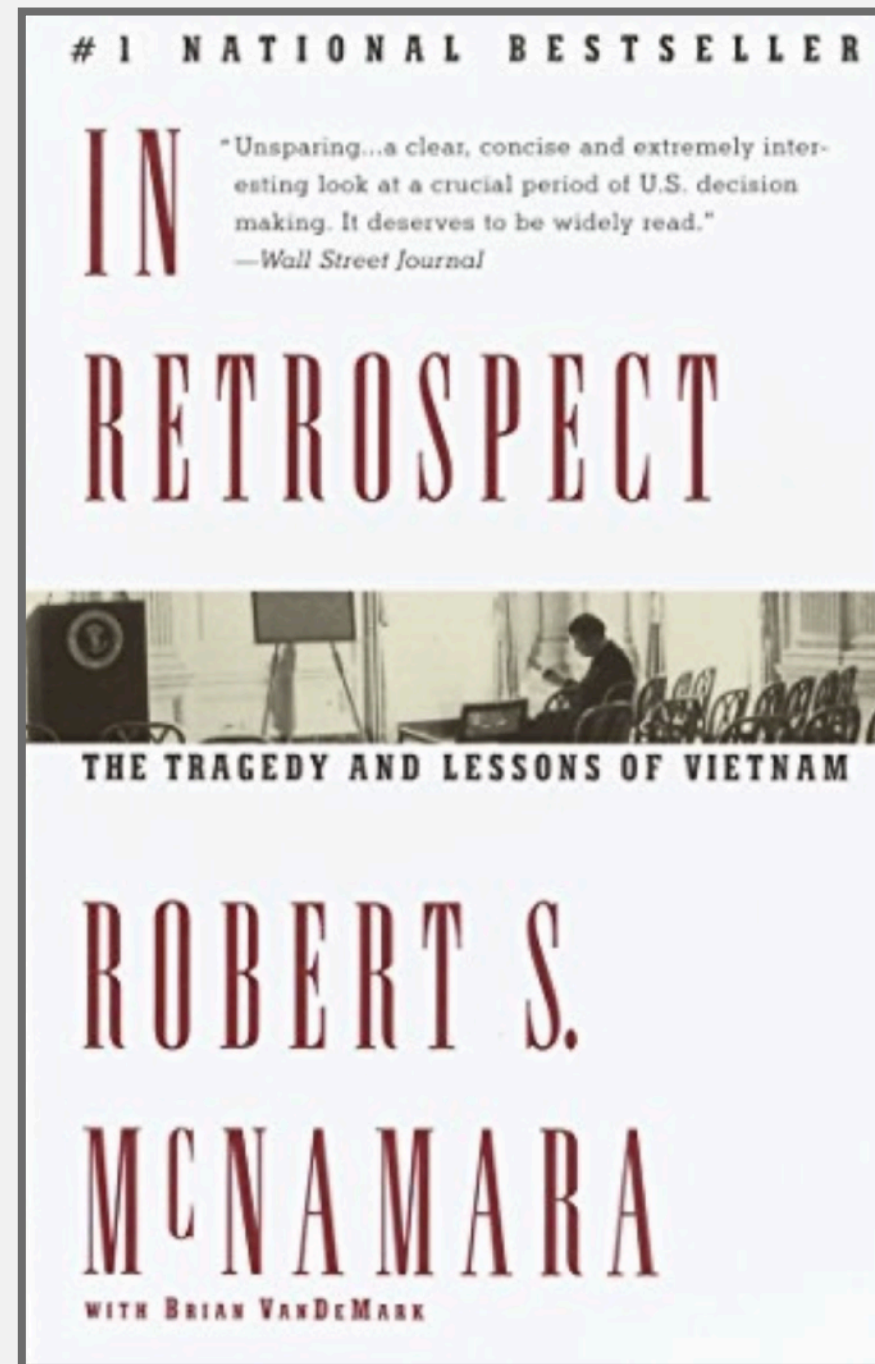
5



- **Construct validity** – Are we measuring what we intended to measure?
- **Internal validity** – The extent to which the measurement can be used to explain some other characteristic of the entity being measured
- **External validity** – Concerns the generalization of the findings to contexts and environments, other than the one studied



- 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





- 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.



- There seems to be a general misunderstanding to the effect that a mathematical model cannot be undertaken until every constant and functional relationship is known to high accuracy. This often leads to the omission of admittedly highly significant factors (most of the “intangibles” influences on decisions) because these are unmeasured or unmeasurable. To omit such variables is equivalent to saying that they have zero effect... Probably the only value known to be wrong...
- J. W. Forrester, *Industrial Dynamics*, The MIT Press, 1961



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



Simplistic Productivity Measures



- Lines of code per day?
 - Industry average 10-50 lines/day
 - Debugging + rework ca. 50% of time
- ● Function/object/application points per month ● Bugs fixed?
 - Milestones reached?



- 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?



- Productivity is all about developer activity
- Productivity is only about individual performance
- One productivity metric can tell us everything
- Productivity measures are useful only for managers
- Productivity is only about engineering systems and developer tools

WARNING!!



- Most software metrics are controversial
 - Usually only plausibility arguments, rarely rigorously validated
 - Cyclomatic complexity was repeatedly refuted, yet is still used
 - “Similar to the attempt of measuring the intelligence of a person in terms of the weight or circumference of the brain”
- Use carefully!
- Code size dominates many metrics
- Avoid claims about human factors (e.g., readability) and quality, unless validated
- Calibrate metrics in project history and other projects
- Metrics can be gamed; you get what you measure



- Measurement is difficult but important for decision making
- Software metrics are easy to measure but hard to interpret, validity often not established
- Many metrics exist, often composed; pick or design suitable metrics if needed
- Careful in use: monitoring vs incentives
- Strategies beyond metrics

Questions to Consider for Your Projects



- What properties do we care about and how do we measure them?
- What is being measured? Does it (to what degree) capture the thing you care about? What are its limitations?
- How should it be incorporated into process?
- What are potentially negative side effects or incentives?