

CS 4530 & CS 5500 **Software Engineering**

Lecture 12.2: Ethics in Software Engineering

Learning Objectives for this Lesson

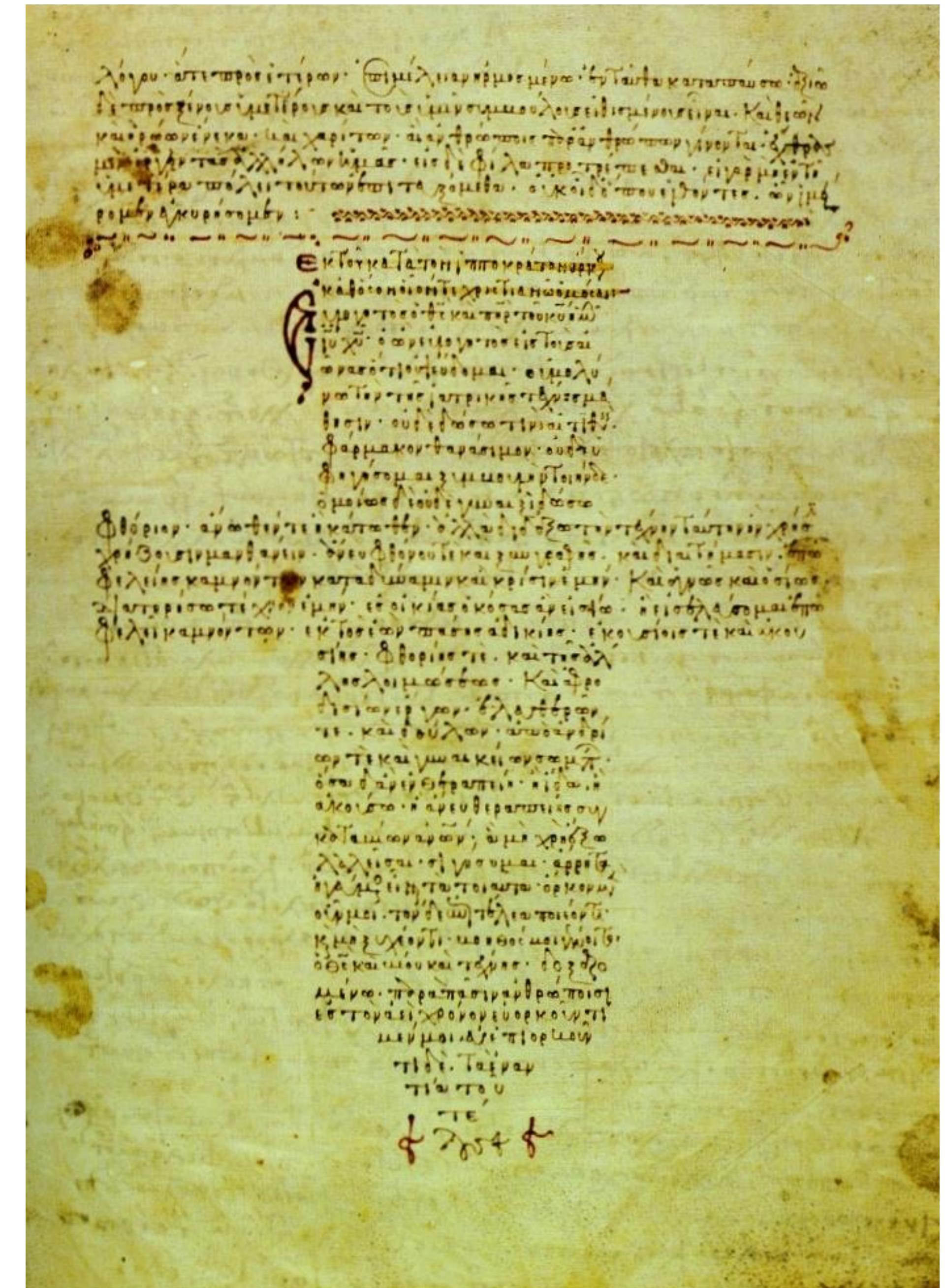
By the end of this lesson, you should be able to...

- Explain several of the meanings of “the public interest”.
- List some sources of ethical guidance for a software engineer.
- List several things that a software engineer can do to try to behave in an ethical manner.

Professional Ethics

Professional standards

- By 1675, standards established for: divinity, law, medicine
- Professionals exercise specialist knowledge or skill - professional ethics governs how this knowledge should be governed



12th-century Byzantine manuscript of the Hippocratic Oath

Code of Ethics

Professional Engineers

Engineers, in the fulfillment of their professional duties, shall:

1. Hold paramount the safety, health, and welfare of the public.
2. Perform services only in areas of their competence.
3. Issue public statements only in an objective and truthful manner.
4. Act for each employer or client as faithful agents or trustees.
5. Avoid deceptive acts.
6. Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.



Code of Ethics

Professional Engineers: Citigroup Center

- Design met building code, but did *not* account for all failure modes
- Last-minute changes to construction increased odds of failure
- Fixed before disaster could strike, but kept a secret for 20 years



Badly-engineered software can kill people

Therac-25 (1985-1987)

- Bug in software caused 100x greater exposure to radiation than intended
- At least 6 died
- Likely far more suffered: deaths occurred over a period of 2 years!
- Weak accountability in manufacturer's organization



“Therac-25” by Catalina Márquez, Wikimedia commons, CC BY-SA 4.0

Code of Ethics

ACM's Code of Ethics Software Engineers

1. PUBLIC – Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER – Software engineers shall act in a manner that is in the best interests of their client and employer.
3. PUBLIC – Software engineers shall act consistently with the public interest.
4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.
8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Unpacking “Public Interest”

1. PUBLIC – Software engineers shall act consistently with the public interest.

Do no harm: how can our software cause harm?

- How can my software fail? What are the implications of that failure?
- Who will use my software, and how might different users use it differently?
- How will my software impact those who do not use it directly?
- Will my software amplify negative behavior for users and society at large?

Unpacking “Public Interest”

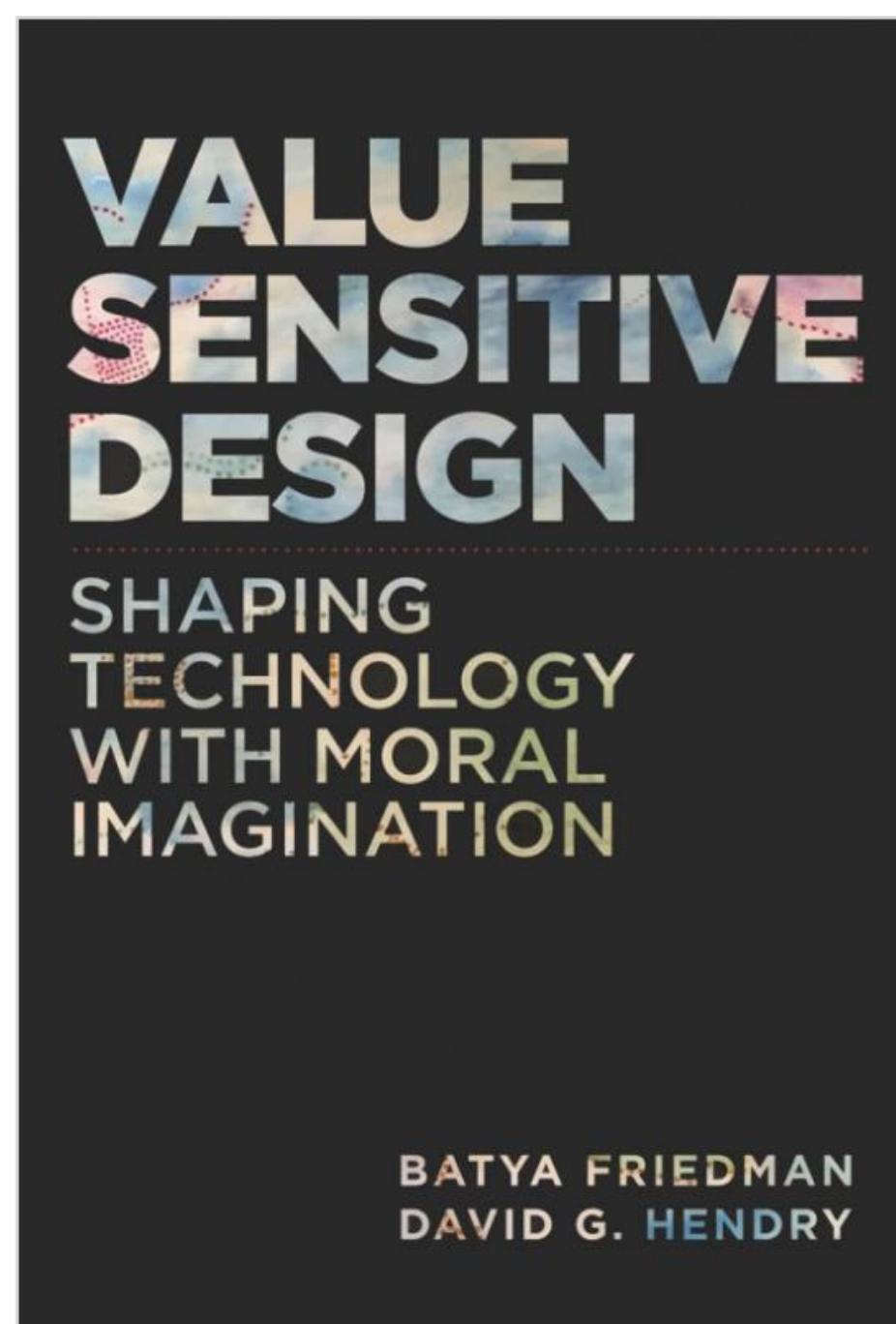
How can our software make a positive contribution?

- Can my software make people’s jobs easier?
- Can my software make people happier?
- Can my software amplify positive behavior for users and society at large?
- How can my software better achieve these goals?

Unpacking “Public Interest”

What values might our software promote or diminish?

- Human rights - Inalienable, fundamental rights to which all people are entitled
- Accessibility - Making all people successful users of the technology
- Justice - Procedural justice (process is fair) + distributive justice (outcomes are fair)
- Privacy - An individual's agency in determining what information about them is shared
- Human welfare - Physical, material and psychological well-being



Code ACM's

1. PUBLIC –
2. CLIENT A
- €

1. PUBLIC
- €
- 3.
4. JUDGMENT
5. MANAGEMENT
- management
6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.

8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Does ACM's Code of Ethics Change Ethical Decision Making in Software Development?

Andrew McNamara
North Carolina State University
Raleigh, North Carolina, USA
ajmcnama@ncsu.edu

TLDR: No

Emerson Murphy-Hill
North Carolina State University
Raleigh, North Carolina, USA
emerson@csc.ncsu.edu

ABSTRACT

Ethical decisions in software development can substantially impact end-users, organizations, and our environment, as is evidenced by recent ethics scandals in the news. Organizations, like the ACM, publish codes of ethics to guide software-related ethical decisions. In fact, the ACM has recently demonstrated renewed interest in its code of ethics and made updates for the first time since 1992. To better understand how the ACM code of ethics changes software-

The first example is the Uber versus Waymo dispute [26], in which a software engineer at Waymo took self-driving car code to his home. Shortly thereafter, the engineer left Waymo to work for a competing company with a self-driving car business, Uber. When Waymo realized that their own code had been taken by their former employee, Waymo sued Uber. Even though the code was not apparently used for Uber's competitive advantage, the two companies settled the lawsuit for \$245 million dollars.

Standards can give more concrete guidance.

- International bodies define standard processes that are designed to protect the public
- By (correctly) following such a standard, you can reduce the chance of harm to users, as well as your ethical (and legal) liability

INTERNATIONAL
STANDARD

IEC
62304

First edition
2006-05

Medical device software –
Software life cycle processes

This English-language version is derived from the original bilingual publication by leaving out all French-language pages. Missing page numbers correspond to the French-language pages.

Standards can give more concrete guidance.

The latest general information on the Coronavirus (COVID-19) is available on [Coronavirus.gov](#). For FAA-specific COVID-19 resources, please visit [faa.gov/coronavirus](#).

United States Department of Transportation

Federal Aviation Administration

About Jobs News Search

AIRCRAFT AIR TRAFFIC AIRPORTS PILOTS & AIRMEN DATA & RESEARCH REGULATIONS SPACE DRONES

FAA Home ▶ Aircraft ▶ Aircraft Certification ▶ Design Approvals

Aircraft Certification Software and Airborne Electronic Hardware

The Aircraft Certification Service is concerned with the approval of software and airborne electronic hardware for airborne systems (e.g., autopilots, flight controls, engine controls), as well as that used to produce, test, or manufacture equipment to be installed on airborne products. The FAA Aircraft Certification Service develops policy, guidance and training for software and airborne electronic hardware that has an effect on the airborne product (a "product" is an aircraft, an engine, or a propeller).

For a list of people you can contact for additional information regarding Aircraft Certification Software and Airborne Electronic Hardware activities, please visit the [Contacts](#) page.

Email List Update

We are updating the email list used for notification of activities relating to airborne digital systems developed using software and airborne electronic hardware. To be added to the list, send an email to

Top Tasks

- Get Form 337, Major Repair and Alteration
- Register an aircraft
- Look up an N-Number
- Review preliminary accident data
- Find aircraft safety alerts
- Search for SAIBs

FAA CONTINUED OPERATIONAL SAFETY
Continued operational safety ensures the integrity of a product throughout its service life, and includes mandatory requirements for modification, maintenance, inspection and corrective actions.

Standards can give more concrete guidance.

Example: Domino's + ADA

Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind

Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT



w3.org

Table of Contents

W3C Recommendation

- Introduction
 - [WCAG 2.0 Layers of Guidance](#)
 - [WCAG 2.0 Supporting Documents](#)
 - [Important Terms in WCAG 2.0](#)
- [WCAG 2.0 Guidelines](#)
 - [1 Perceivable](#)
 - [1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.](#)
 - [1.2 Provide alternatives for time-based media.](#)
 - [1.3 Create content that can be presented in different ways \(for example simpler layout\) without losing information or structure.](#)
 - [1.4 Make it easier for users to see and hear content including separating foreground from background.](#)
 - [2 Operable](#)
 - [2.1 Make all functionality available from a keyboard.](#)
 - [2.2 Provide users enough time to read and use content.](#)
 - [2.3 Do not design content in a way that is known to cause seizures.](#)
 - [2.4 Provide ways to help users navigate, find content, and determine where they are.](#)
 - [3 Understandable](#)
 - [3.1 Make text content readable and understandable.](#)
 - [3.2 Make Web pages appear and operate in predictable ways.](#)
 - [3.3 Help users avoid and correct mistakes.](#)
 - [4 Robust](#)
 - [4.1 Maximize compatibility with current and future user agents, including assistive technologies.](#)
- [Conformance](#)
 - [Conformance Requirements](#)
 - [Conformance Claims \(Optional\)](#)
 - [Statement of Partial Conformance - Third Party Content](#)
 - [Statement of Partial Conformance - Language](#)
- [Appendices](#)
 - [Appendix A: Glossary \(Normative\)](#)
 - [Appendix B: Acknowledgments](#)
 - [Appendix C: References](#)

WCAG 2.0 Specification

But many ethical decisions are still hard

Example: Social Media Platforms

- Social media platforms like Facebook, Twitter, or Tiktok can
 - Build communities across distance
 - Spread information for social action
 - Spread misinformation and disinformation
 - Amplify hate speech
- How to balance freedom vs regulation?



Where does this leave us?

So that we can sleep at night

- Consider the different ways that our software may impact others
- Consider the ways in which our software interacts with the political, social, and economic systems in which we and our users live
- Follow best practices, and actively push to improve them
- Encourage diversity in our development teams
- Engage in honest conversations with our co-workers and supervisors to explore possible ethical issues and their implications.

Learning Objectives for this Lesson

You should now be able to:

- Explain several of the meanings of “the public interest”.
- List some sources of ethical guidance for a software engineer.
- List several things that a software engineer can do to try to behave in an ethical manner.

CS 4530 & CS 5500 **Software Engineering**

Lecture 12.3: Acceptance and Inclusivity Testing

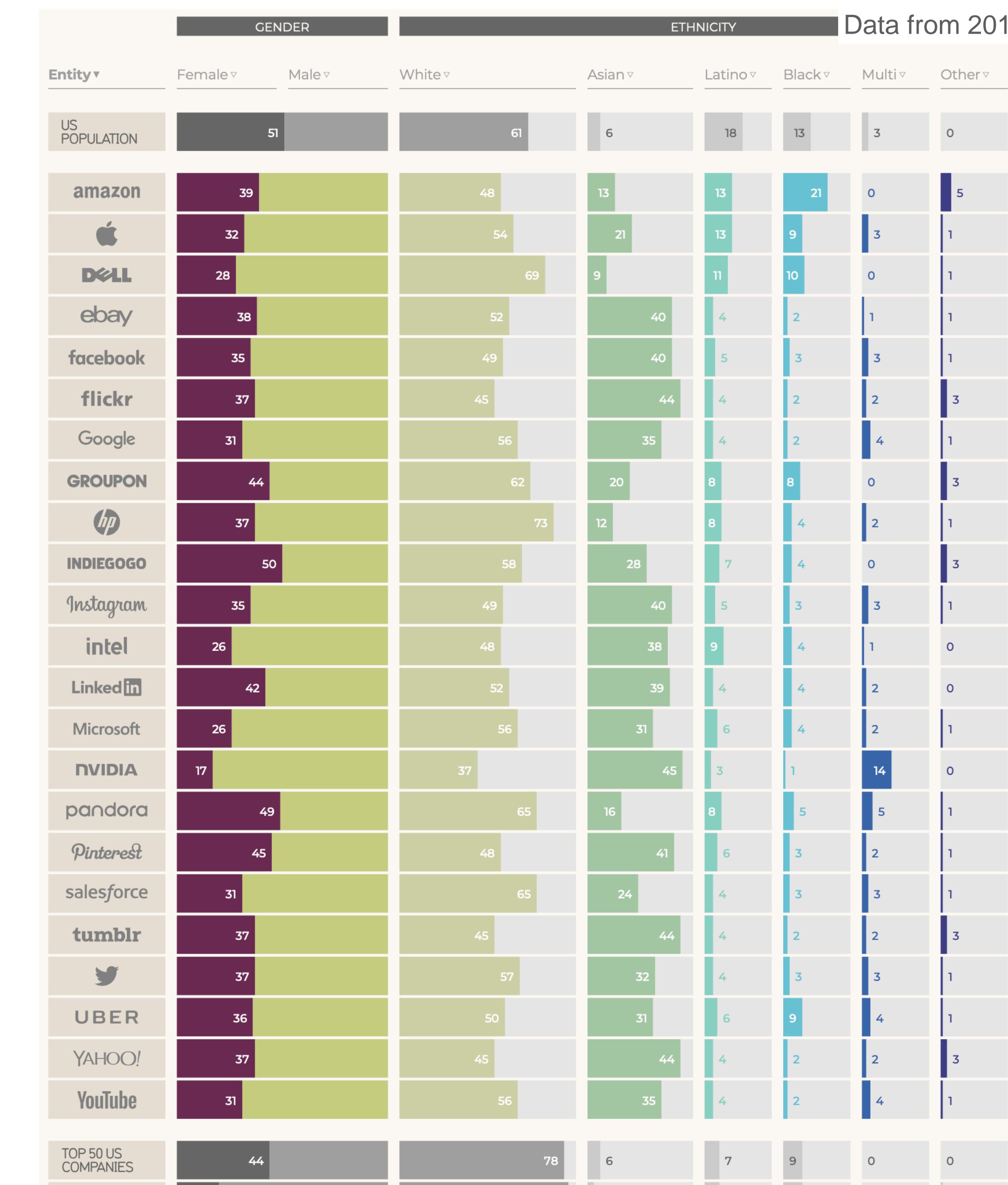
Learning Objectives for this Lesson

By the end of this lesson, you should be able to...

- Explain ways in which developers of software often differ from the users of that software, introducing potential inclusivity bugs
- Recognize persona-based cognitive walk-throughs as an approach to help put yourself in someone else's shoes

We are not our users

- Creating inclusive software requires us to acknowledge that we differ from our users
- Our quality assurance is only as good as we can understand our users



Bias is the Default

Example: Google Photos auto-tagging (2015)



DIGITS

Google Mistakenly Tags Black People as ‘Gorillas,’ Showing Limits of Algorithms

By [Alistair Barr](#)

Updated July 1, 2015 3:41 pm ET

SHARE TEXT

Google is a leader in artificial intelligence and machine learning. But the company's computers still have a lot to learn, judging by a major blunder by its Photos app this week.

The app tagged two black people as “Gorillas,” according to Jacky Alciné, a Web developer who spotted the error and tweeted a photo of it.

“Google Photos, y'all f**ked up. My friend's not a gorilla,” [he wrote on Twitter](#).

Google apologized and said it's tweaking its algorithms to fix the problem.

“We're appalled and genuinely sorry that this happened,” a company

<https://www.wsj.com/articles/BL-DGB-42522>

<https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/>



BACKCHANNEL BUSINESS CULTURE GEAR IDEAS MORE ▾

SIGN IN



TOM SIMONITE BUSINESS 01.11.2018 07:00 AM

When It Comes to Gorillas, Google Photos Remains Blind

Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn't found one.



Unconscious Bias in Software

Does your software support a variety of users?

- Aside from gender, race and ethnicity, *how* people interact with software varies, research has shown key inclusiveness *facets*:
 - Motivations
 - Information processing style
 - Computer self-efficacy
 - Risk averseness
 - Tech learning style
- Idea: Perform *cognitive walkthrough* of our software, through the lens of someone else



GenderMag: Gender Inclusiveness Magnifier

Persona-based evaluation

Abby Jones¹



scanning all her emails first to get
an overall picture before answering any of them.

Background and skills

their software systems are new to her.

“numbers person”
She likes Math and knows how to think with numbers

she also enjoys working with numbers and logic.

Motivations and Attitudes

▪ **Motivations:** Abby uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.

▪ **Computer Self-Efficacy:** Abby has low confidence about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

▪ **Attitude toward Risk:** Abby’s life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they’re more predictable about what she will get from them and how much time they will take.

How Abby Works with Information and Learns:

▪ **Information Processing Style:** Abby tends towards a comprehensive information processing style when she needs to more information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is “burst-y”; first she reads a lot, then she acts on it in a batch of activity.

▪ **Learning: by Process vs. by Tinkering:** When learning new technology, Abby leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn’t particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

¹ Abby represents users with motivations/attitudes and information/learning styles similar to hers. For data on females and males similar to and different from Abby, see <http://eusesconsortium.org/gender/gender.php>

GenderMag: Gender Inclusiveness Magnifier

Persona-based evaluation

- Step through a use case for your tool, acting as the persona
- Avoid jumping to conclusions - work in a group with multiple evaluators, take notes of issues as they occur
- Compare to heuristic evaluation (week 6)

<ul style="list-style-type: none">• 1. Pick a persona. eg: Abby• 2. Pick a use case/scenario in your tool, eg:<ul style="list-style-type: none">– in Book Store Navigator app...– “Find science fiction books”	 	<ul style="list-style-type: none">• 3a-b. Pick a Subgoal for that scenario. eg: Subgoal #1: “See bookstore map”. Q: Will Abby have formed this sub-goal...?<ul style="list-style-type: none">• Yes/no/maybe.Why? Consider Abby's Motivations...
<ul style="list-style-type: none">• 3c-d. Pick an Action for that subgoal. Action #1: “Tap ‘Browse Off’”:<ul style="list-style-type: none">– Q1. Will Abby know what to do?<ul style="list-style-type: none">• Yes/no/maybe.Why? Consider Abby's ... Tinkering	 <p>→ First answer Q1. After answering it, then perform the action.</p>	<ul style="list-style-type: none">– 3e. Q2. If she performs the action, producing will Abby see progress toward the subgoal?<ul style="list-style-type: none">• Yes/no/maybe. Why? Consider Abby's Self-Efficacy & ...

The Curb Cut Effect



[“Curb Cuts” by Mike Gifford, CC BY-NC 2.0](#)

Usability Testing

Directly measuring the usability and inclusivity of our software

- Observe real users interacting with your software - provide each user with a task, monitor their progress towards completing that task
- Consider a diverse set of users that represent those who will use your software
- Validate problems (and fixes) that you identify in cognitive walkthroughs
- Example: applying GenderMag + usability testing for [Microsoft Academic](#)

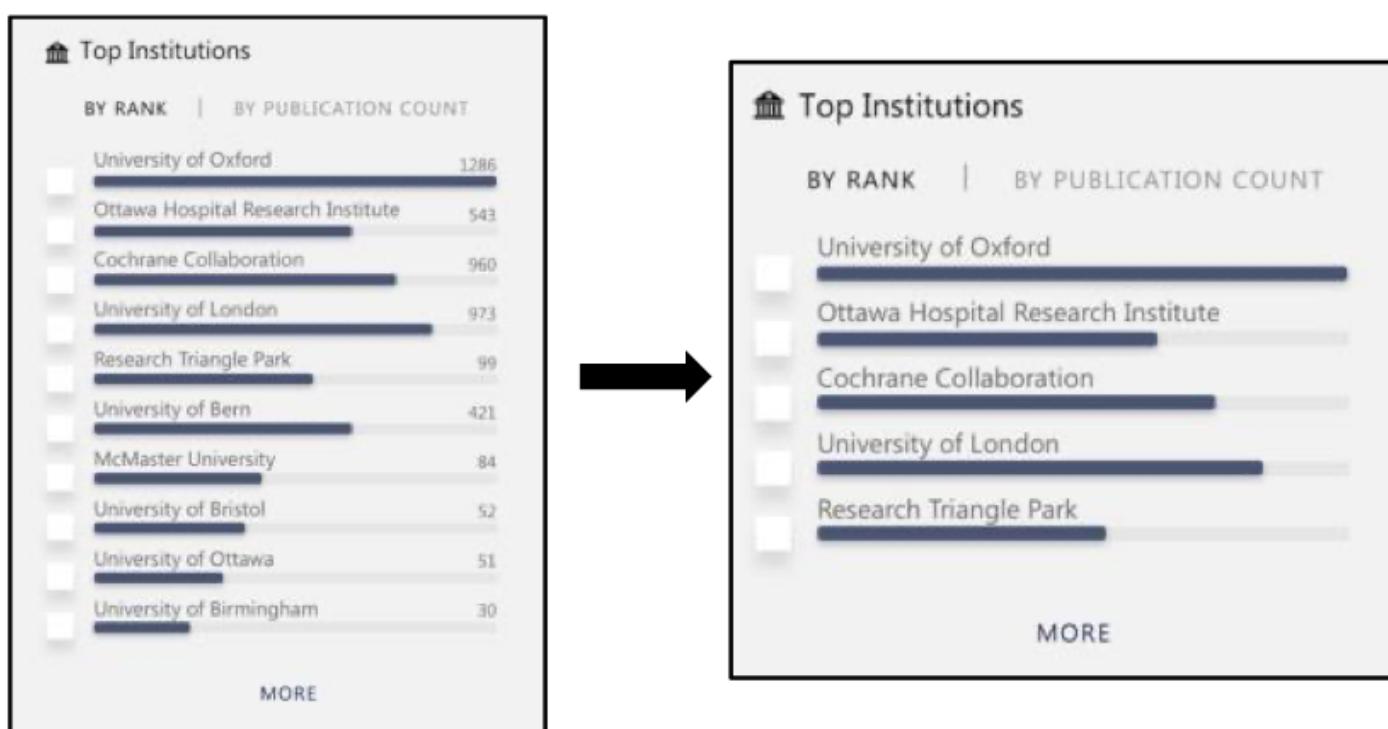


Figure 5. Issues 1 & 2 filtering redesign. (Left) Original: List of institutions with publication counts for each. (Right) post-GenderMag: Shorter list of institutions and removed the publication counts that drew attention away from the checkbox actionability.

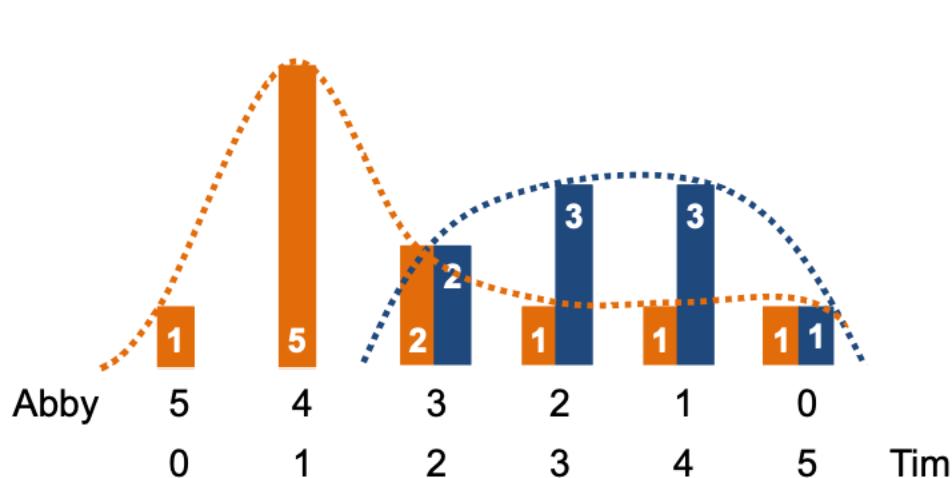


Figure 13. Y-axis: Counts of the 20 men and women participants by their facet values. (Same as Figure 2 but broken out by gender.) Orange: women, blue: men. X-axis: Abby=Abby Facets, Tim=Tim Facets. Example: the left bar says that the only participant with 5 Abby facets (0 Tim facets) was a woman; the right pair of bars says that one man and one woman had 5 Tim facets (0 Abby facets).

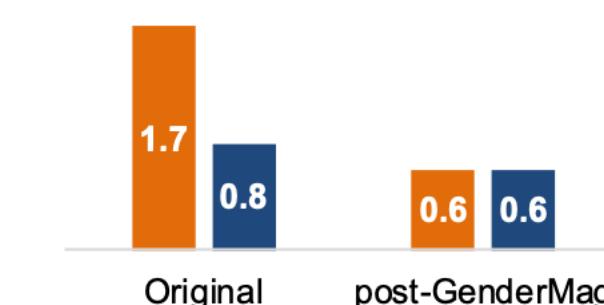


Figure 14. Average number of action failures per person by gender identification (orange: women, blue: men). In the Original version, women's action failure rates were over twice as high as men's; with the post-GenderMag redesign, all failure rates went down, and the gender gap disappeared.

Usability Testing

Evaluating Accessibility

- Check for conformance with requirements of standards
- Involve users in your evaluation - simply “meeting a standard” does not guarantee accessibility

<https://www.w3.org/WAI/test-evaluate/>

The screenshot shows the W3C WAI website with the URL <https://www.w3.org/WAI/test-evaluate/> in the address bar. The page title is "Evaluating Web Accessibility Overview". The main content area includes sections for "Summary" (describing the page as links to resources for evaluating web accessibility), "Page Contents" (listing topics like Introduction, Initial Checks, Tools, Conformance Evaluation and Reports, People, Standards, and Alternatives for Video Introduction - Resources Overview), and "Introduction" (explaining the importance of evaluating accessibility early and involving users). A video player at the bottom right shows a YouTube video titled "Evaluating Web Accessibility - Overview" with a magnifying glass icon over the word "accessibility".

Usability Testing

For some software, we are *nothing* like our users

The image displays three screenshots of the ENERGY 14 SYSTEM software, illustrating the process of creating and executing a commodity swap.

Screenshot 1: Commodity Swap Configuration

This screenshot shows the initial configuration of a swap. The "Currency Pair" is set to "USD/CAD". The "Settlement" option is selected as "Physical NDF". The "GENERAL" section includes fields for "Book" (Search Books by code or name...) and "Counterparty" (Search Counterparties by code or name...). The "UNDERLYING" section specifies the swap details: Swap "PJABF00-CLc1", Instrument "PJABF00", Weight "1", and Unit Conversion "~".

Screenshot 2: Underlying Details

This screenshot provides more detailed settings for the underlying instrument. It shows the swap period from "01-Jan-2016" to "31-Dec-2016" with a monthly frequency. The volume is set to "Buy 10,000 BBL per Day 2,520,000". The price is listed as "0.86 USD per BBL 3DP Standard". Observation dates are set for "PLD & NYMEX Yes Avg at End". Payment details include "US 5bd" with a business day type, "From After period end NY MFEQ".

Screenshot 3: Commodity Swap Execution

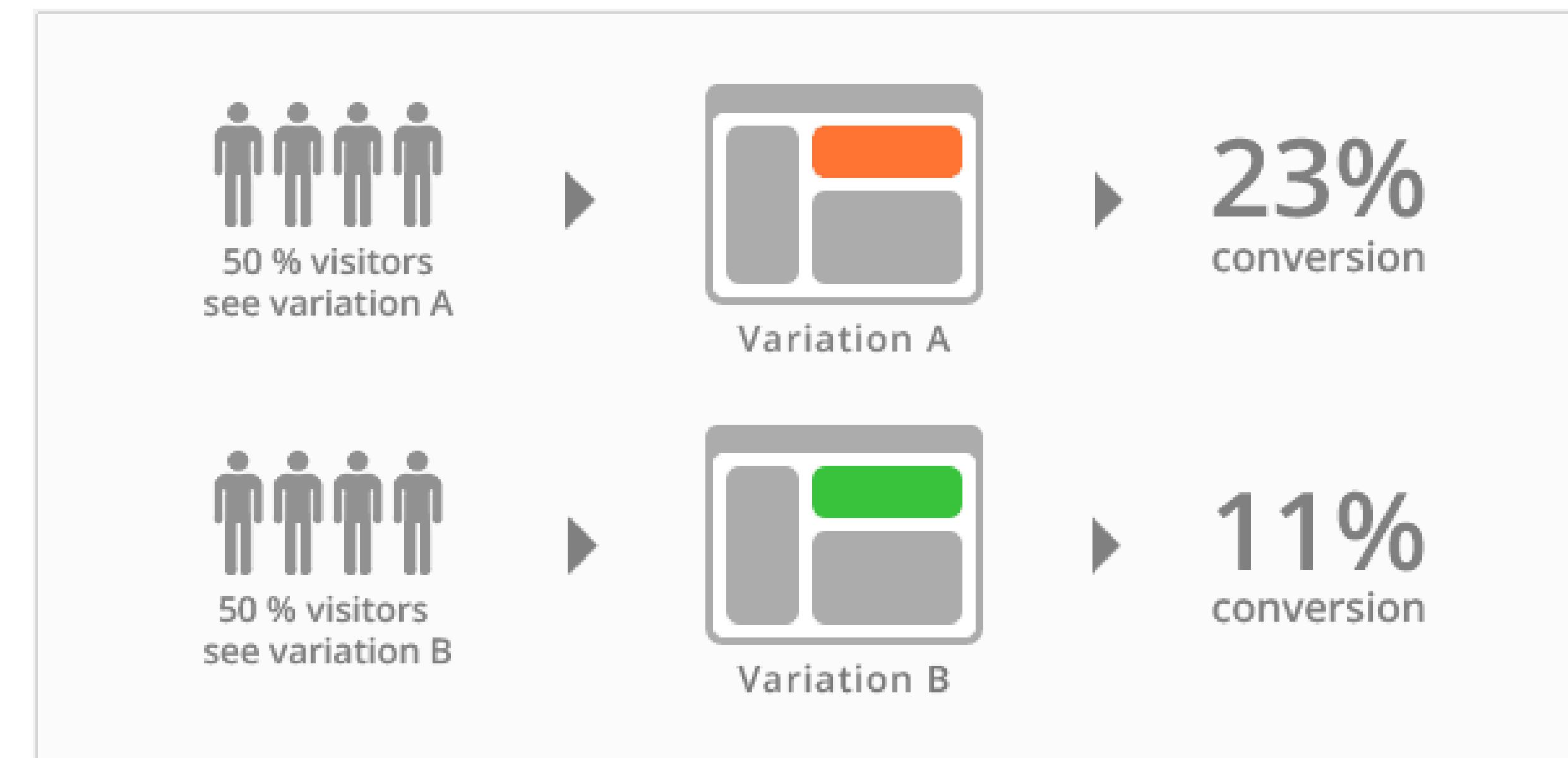
This screenshot shows the final executed swap details. The swap is identified as "BPI_1200667-0 cal'16 10k BBL/day (2.52M total) USD 0.86/BBL". The "Commodity Swap" section lists the underlying instrument "PJABF00 - CLc1" with a weight of "1 ~" and a unit conversion of "0.274". The swap period is "cal'16" from "01-Jan-2016" to "31-Dec-2016" with a monthly frequency. The volume is "Buy 10,000 BBL per Day 2,520,000". The price is "0.86 USD per BBL 3DP Standard". The observation dates are "PLD & NYMEX Yes Avg at End". The payment is "US 5bd" with a business day type, "From After period end NY MFEQ". The total value is "1,000,000 * 0:10.4".

Commodity swap screens & story © 2016, Brad Paley
<http://didi.co/consulting/>

Usability Testing in Continuous Development

A/B Testing

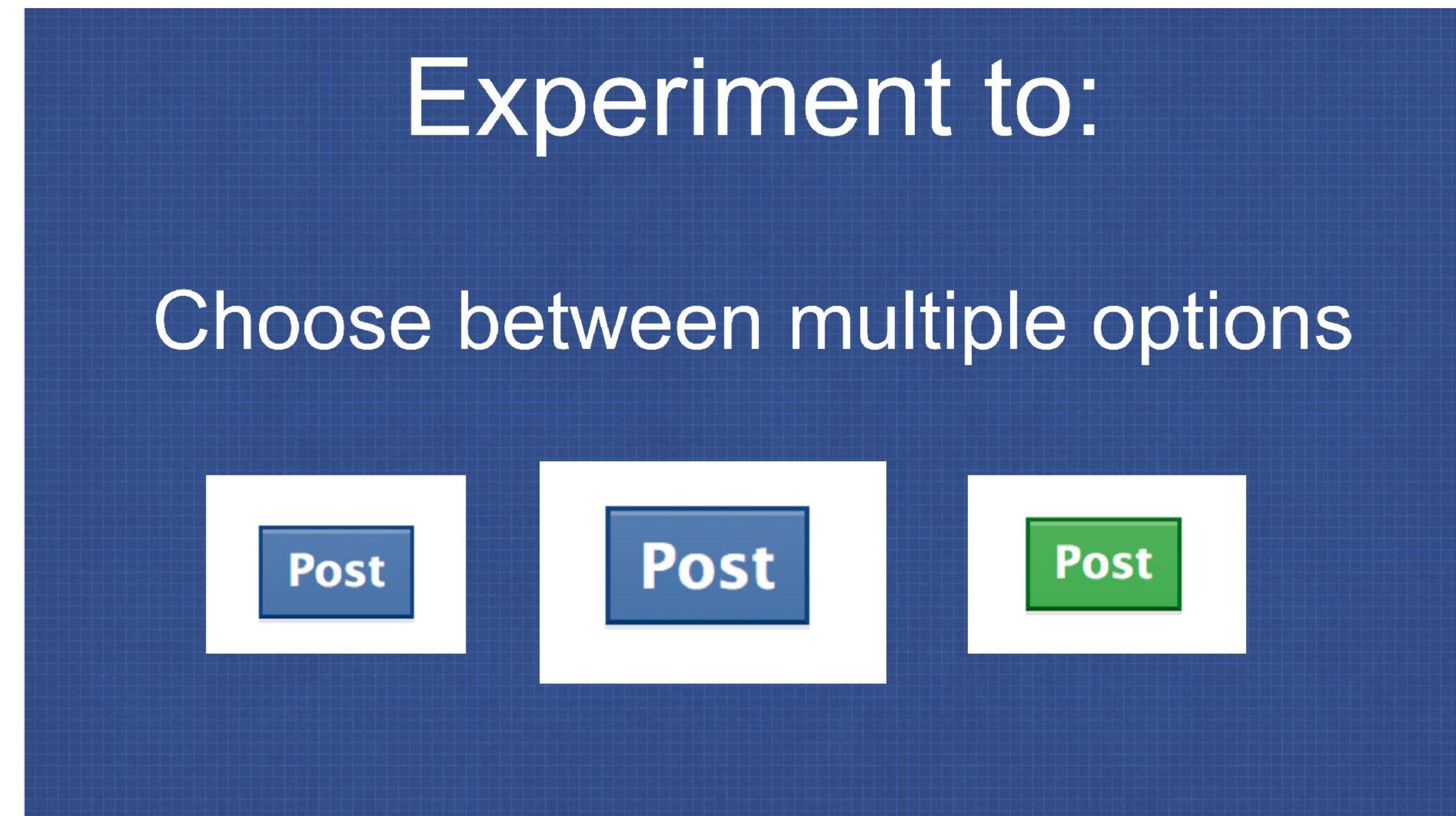
- Ways to test new features for usability, popularity, performance without a focus group
- Show 50% of your site visitors version A, 50% version B, collect metrics on each, decide which is better



Usability Testing in Continuous Development

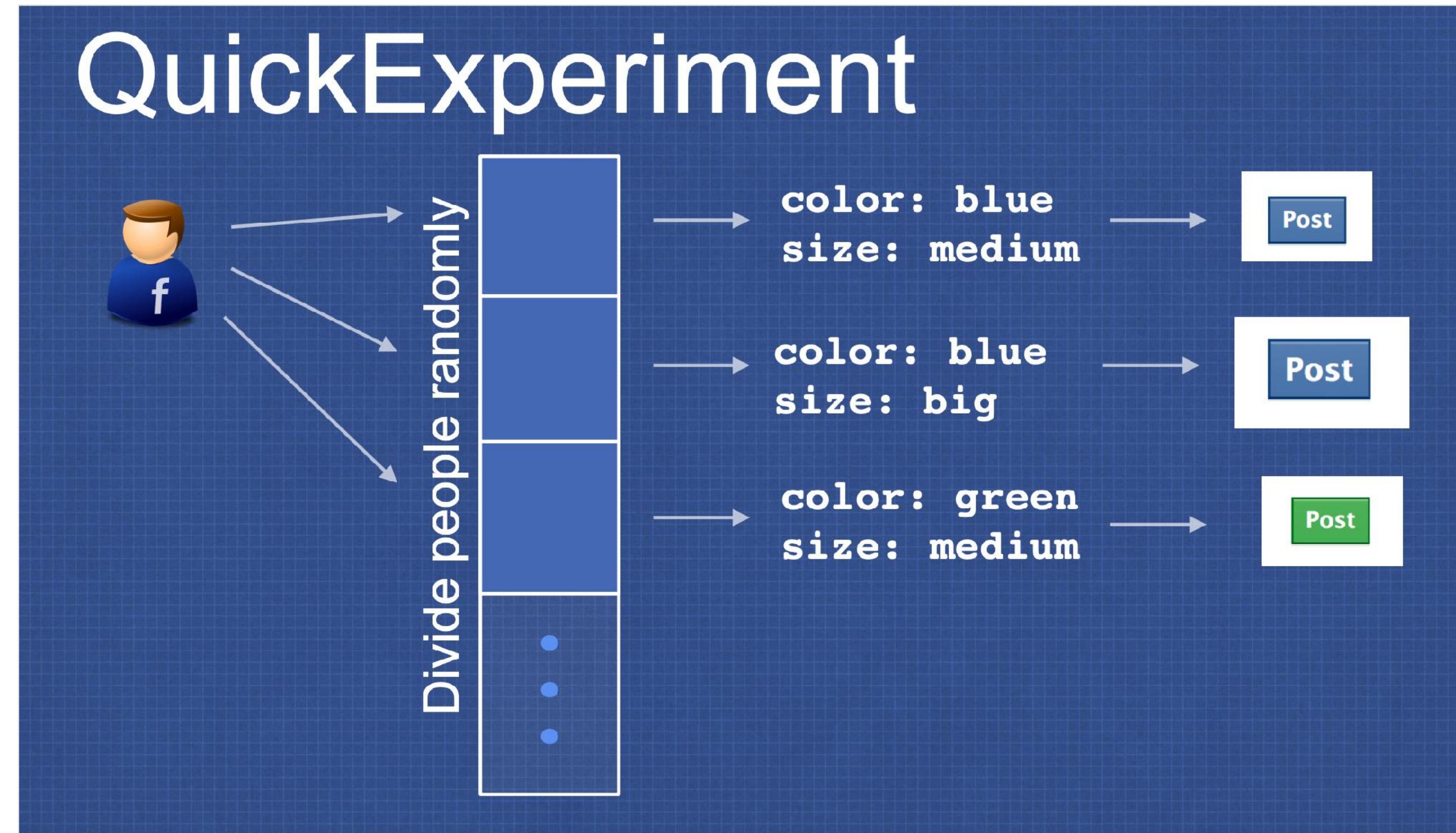
A/B Testing: PlanOut from Facebook (“N=10⁹ user study”)

- Used to test advertising strategies (and Facebook functionality)
- Segment audience and define KPIs, collect results



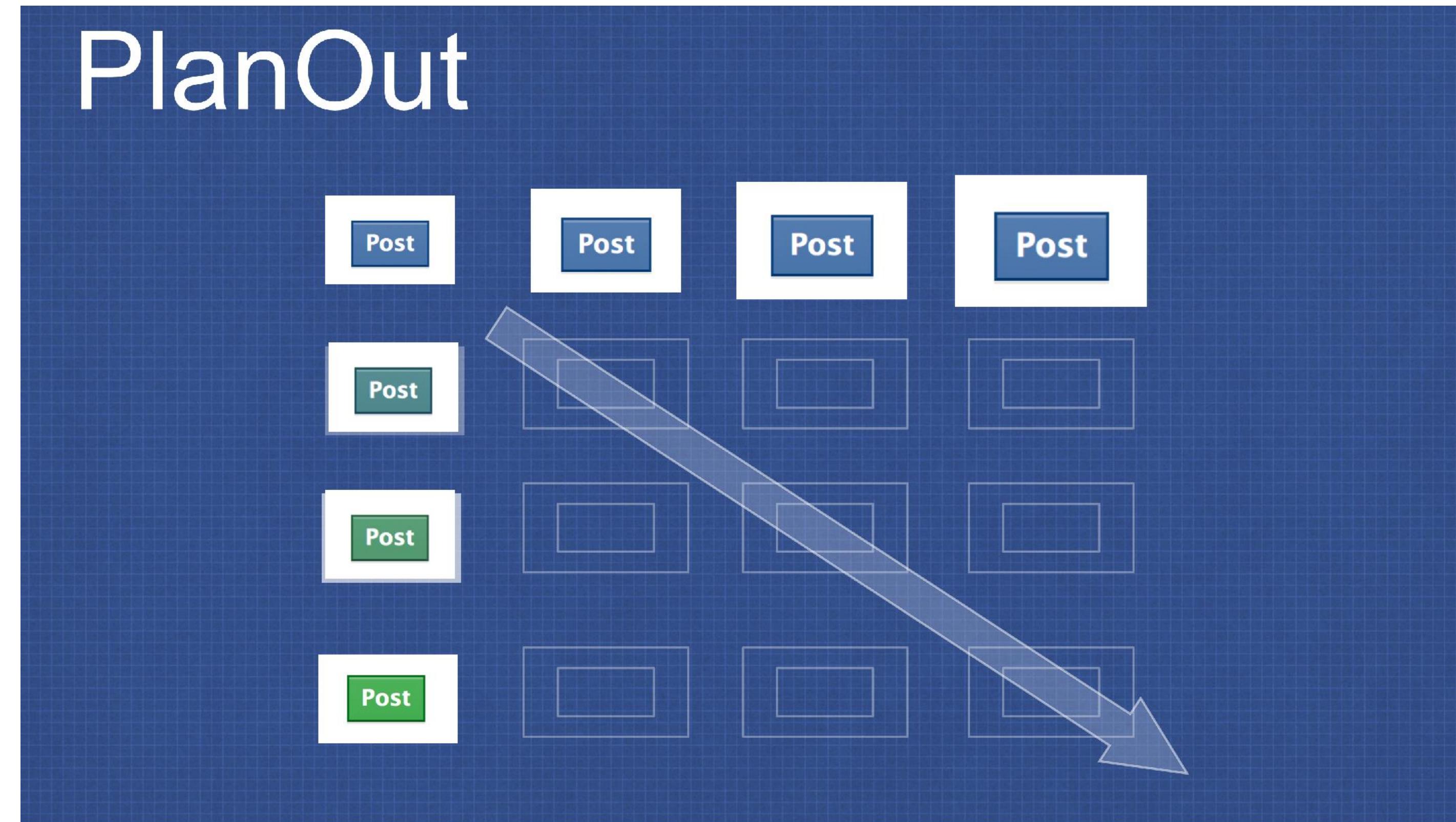
Usability Testing in Continuous Development

A/B Testing: PlanOut from Facebook (“N=10⁹ user study”)



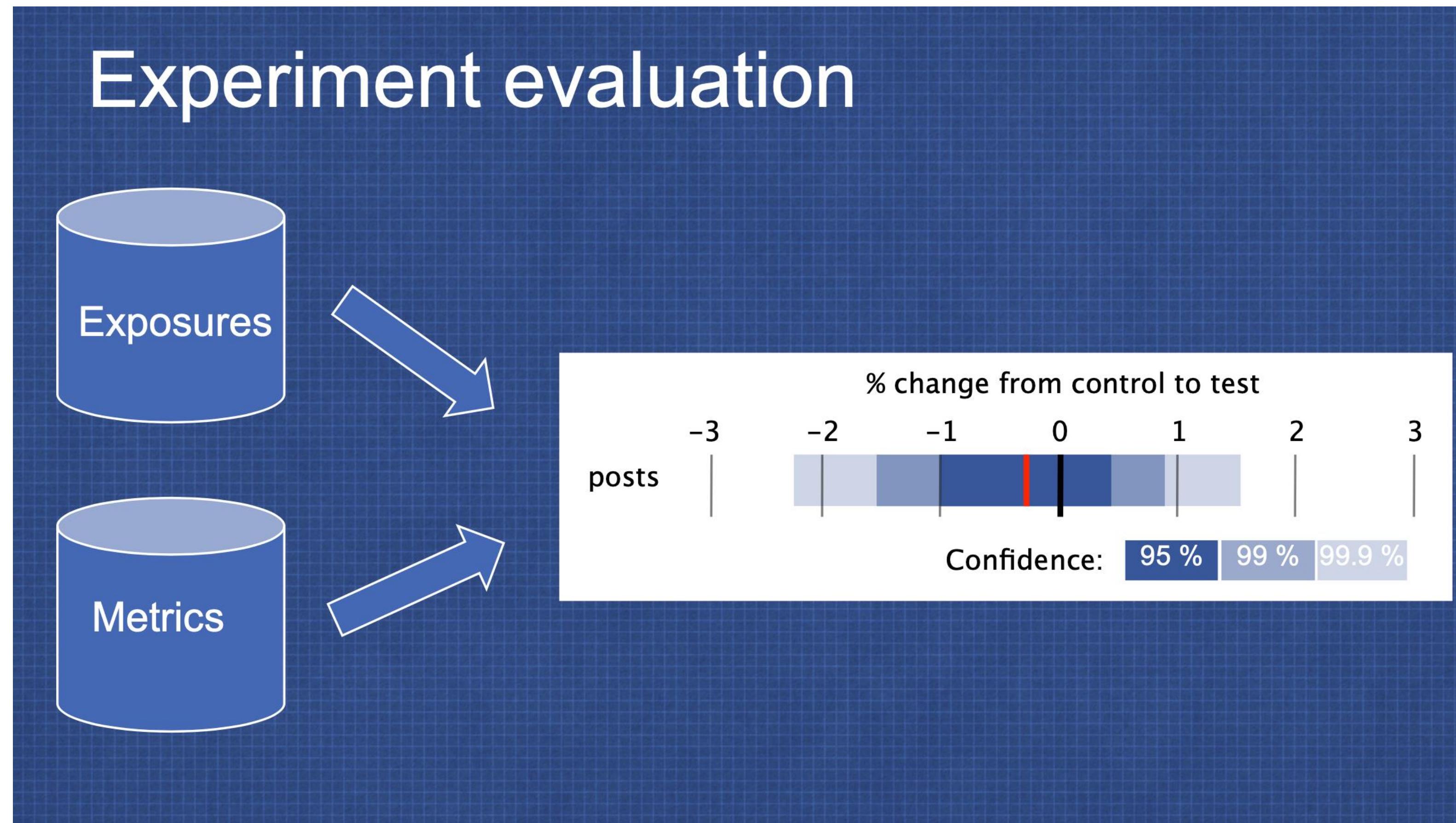
Usability Testing in Continuous Development

A/B Testing: PlanOut from Facebook (“N=10⁹ user study”)



Usability Testing in Continuous Development

A/B Testing: PlanOut from Facebook (“N=10⁹ user study”)



Engineering Equitable Software

Key takeaways

- With great power comes great responsibility
- Anticipate the unanticipated consequences of your software
- Limiting the accessibility of software might save money in the short term, but cost much more in the long term
- Form a diverse development team, and involve a diverse group of users to validate your software