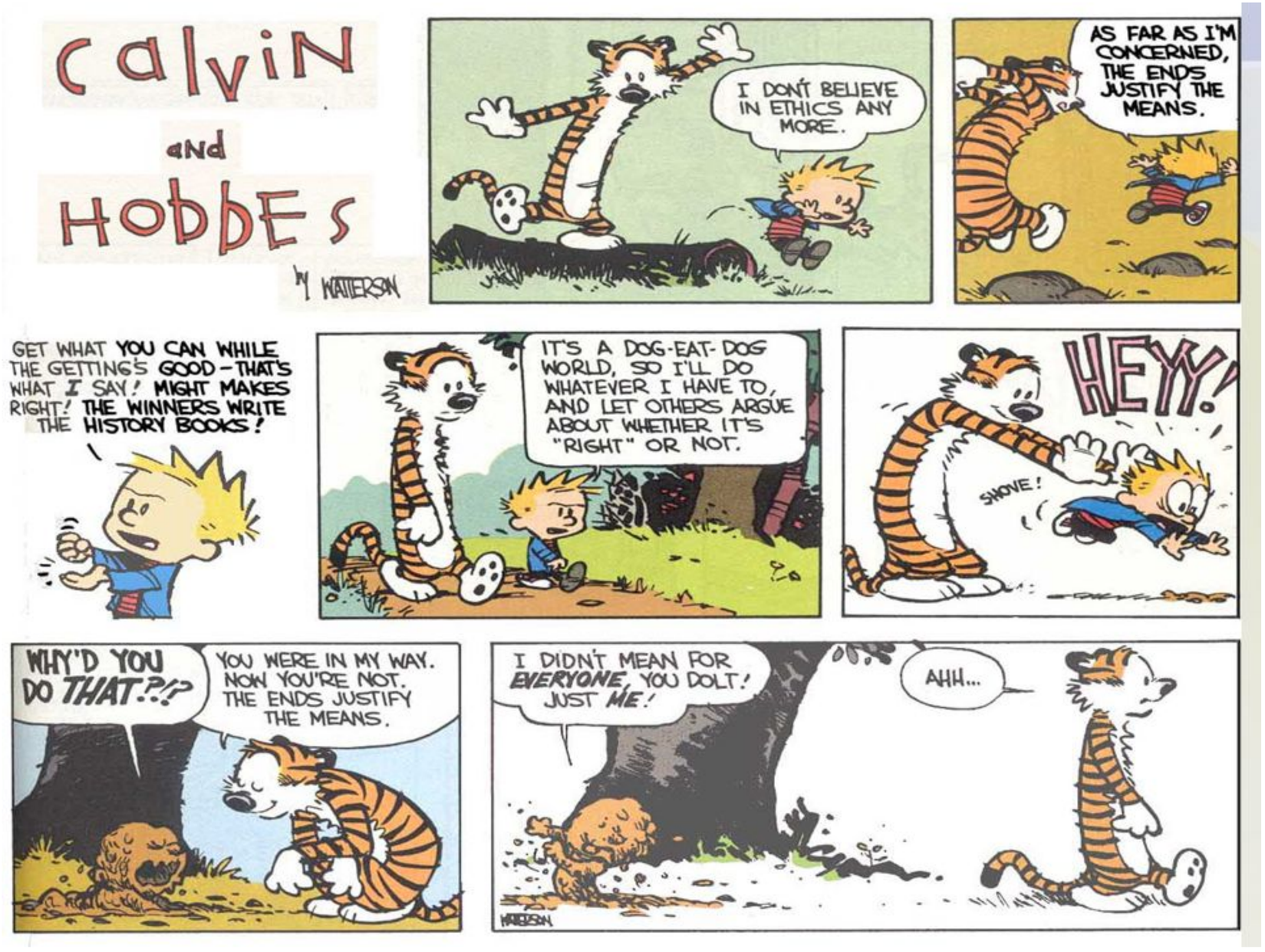
CS 208e
Computers and Ethics

Monday, November 15th, 2021

Chris Gregg

What are ethics?



What do we mean when we say “ethics”?

Ethics in Computer Science and Engineering

https://imgs.xkcd.com/comics/research_ethics.png

What are some of the issues that computer scientists / engineers / scientists need to think about with regard to ethics in the field?

Alt text: I mean, it's not like we could just demand to see the code that's governing our lives. What right do we have to poke around in Facebook's private affairs like that?

Your data…

Who owns your data?

https://www.youtube.com/watch?v=y1txYjoSQQc

Big Companies

Where does responsibility lie with big companies?

Software and Hardware Bugs: Ethics?

Maurice Wilkes, 1913-

> "As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."

Maurice Wilkes, lecture on "The Design and Use of the EDSAC," September 23, 1979

The Inevitability of Bugs

> Simply because of its inevitable large size, the software capable of performing the battle management task for strategic defense will contain errors. All systems of useful complexity contain software errors.

— Department of Defense, The Eastport report on Computing in Support of Battle Management, December 1985

> Although programming techniques have improved immensely since the early days, the process of finding and correcting errors in programming known graphically—if inelegantly—as debugging still remains a most difficult, confused and unsatisfactory operation. . . . Although we are happy to pay lip-service to the adage that to err is human, most of us like to make a small private reservation about our own performance on special occasions when we really try. It is somewhat deflating to be shown publicly and incontrovertibly by a machine that even when we do try, we in fact make just as many mistakes as other people. If your pride cannot recover from this blow, you will never make a programmer.

— Christopher Strachey, Scientific American, 1966

What Makes Computing Different?

- Computers are used to solve hard problems, which means that the difficulty is typically intrinsic to the application.
- Software has high “system complexity” and is therefore difficult to distribute among members of a large team.
- Bugs are omnipresent and inevitable.
- Software systems are discrete rather than continuous: there is no way to “overengineer” such systems to ensure correctness.
- Software systems are inherently chaotic: small changes in initial conditions generate massive changes in the results.
- The economics of software systems means that even flawed products can have tremendous economic value.

The Importance of Economics

- Economics has more impact on directions in modern computing than technology does. The most significant factors are:

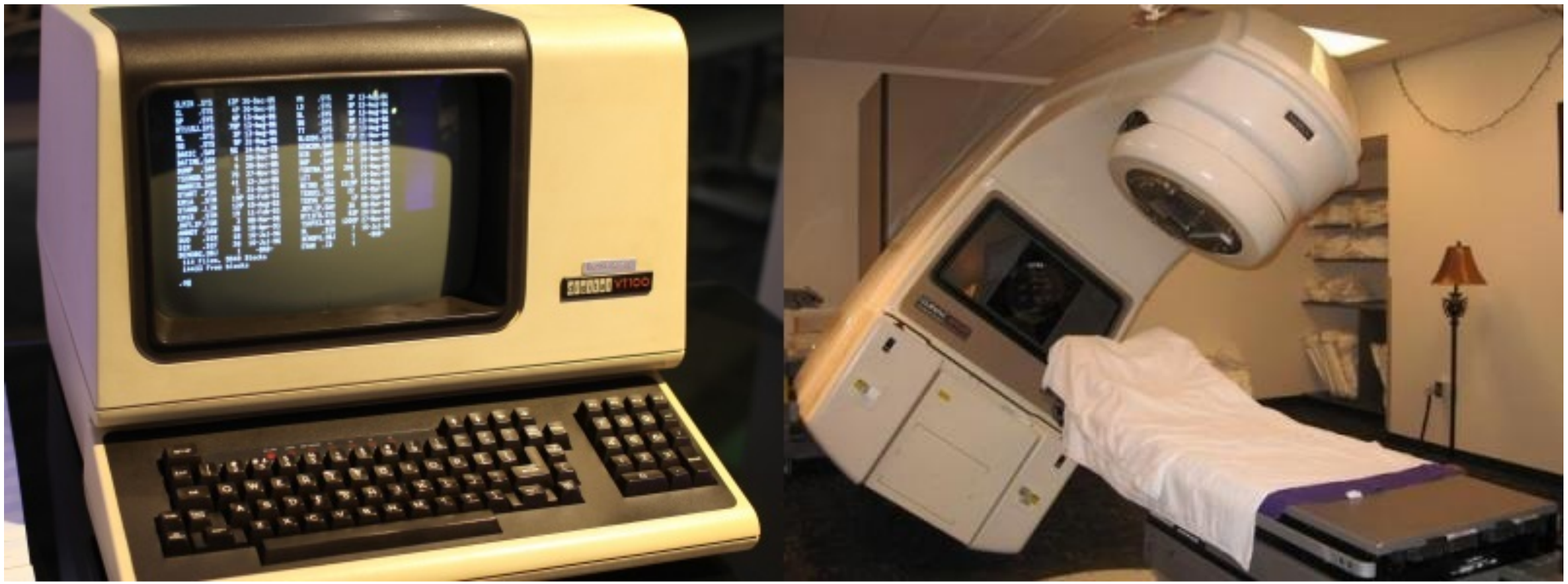
  - Low distribution costs. Software is expensive to produce, but essentially free to duplicate and distribute. Because development costs can be distributed across a larger base, big players have a distinct advantage.
  - Network externalities. The value of software increases with the number of people using that software.
  - High cost-effectiveness. Software tends to be remarkably useful, even when bugs exist.
  - Shortage of highly skilled labor. Individual Productivity varies widely among software developers. Demand is high for the most productive programmers, but the supply is extremely limited.

Variations in Programmer Productivity

- In 1968, a study by Sackman, Erikson, and Grant[1] revealed that programmers with the same level of experience exhibit variations of more than 20 to 1 in the time required to solve particular programming problems.
- More recent studies[2, 3, 4] confirm this high variability.
- Most industry insiders believe that the productivity variance is even higher today. In 2005, Google's VP for Engineering, Alan Eustace, told The Wall Street Journal that one top-notch engineer is worth 300 times or more than the average.[5]

1. H. Sackman, W. J. Erikson, and E. E. Grant. Exploratory experimental studies comparing on-line and off-line programming performance. Communications of the ACM, January 1968.
2. W. Curtis. Substantiating programmer variability. Proceedings of the IEEE, July 1981.
3. T. DeMarco and T. Lister. Programmer performance and the effects of the workplace. Proceedings of the 8th International Conference on Software Engineering. IEEE Computing Society Press, August 1985.
4. G. E. Bryan. Not all programmers are created equal. In Richard Thayer, Software Engineering Project Management (second edition), IEEE Computer Society, 1997.
5. T. Pui-Wing and K. Delaney. Google's growth helps ignite Silicon Valley hiring frenzy. Wall Street Journal, November 23, 2005.

Therac-25: The Killer Machine



https://blog.bugsnag.com/bug-day-race-condition-therac-25/

https://en.wikipedia.org/wiki/Therac-25

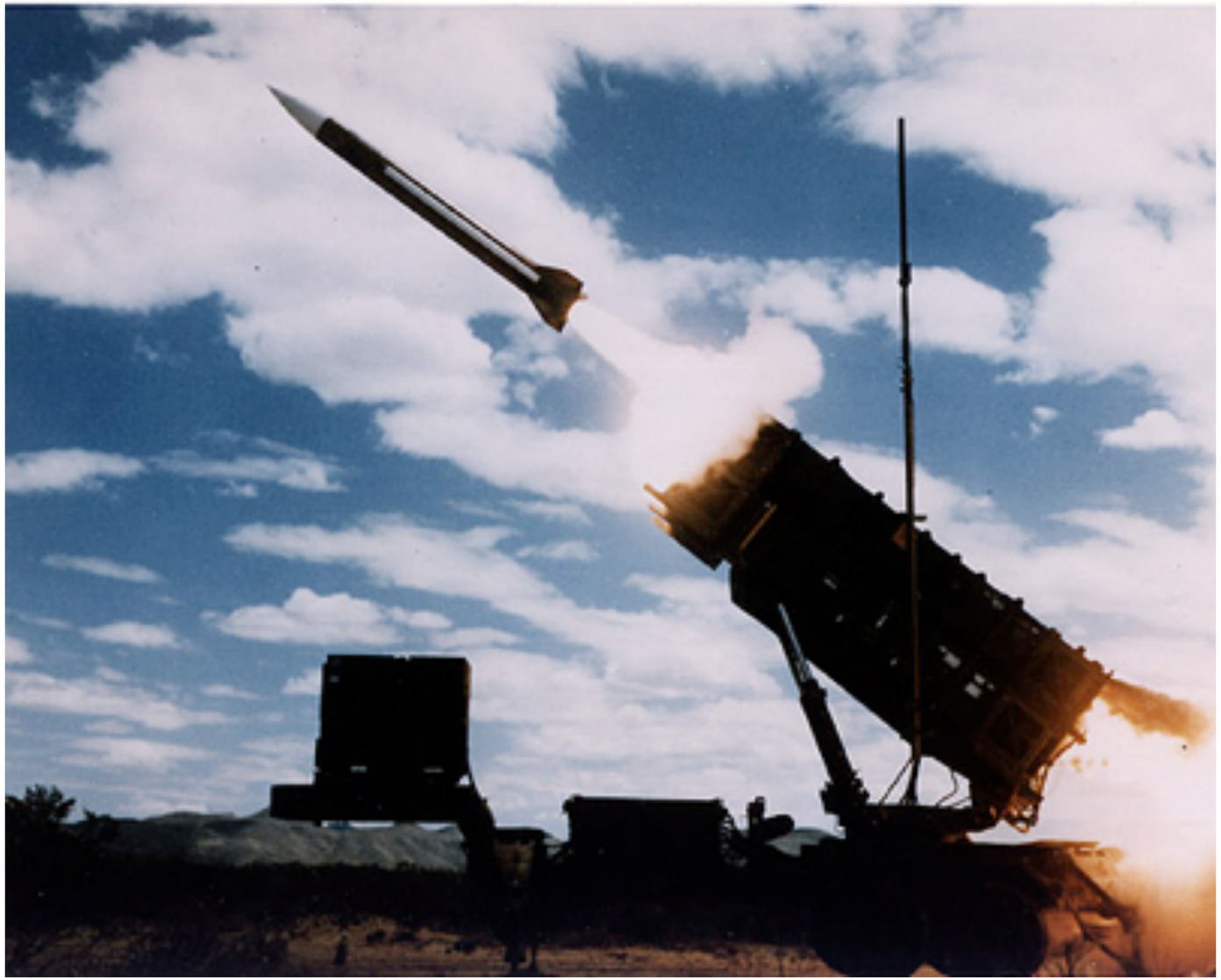
Therac-25: The Killer Machine

Race Condition example:

```
if (x == 5) // the “check”
{
  y = x * 2; // the “act”

  // if another thread changed x in between “if (x==5)”
  // and “y = x * 2” above, y will not be equal to 10.
}
```

Patriot Missle



Patriot Missle: Binary Calculation Error

- "Fixed point" numbers (instead of floating point numbers, which would have had different, but potentially bad problems, too)
- The binary expansion of 1/10 is 0.0001100110011001100110011001100....
- 24 bit register in the Patriot stored instead 0.00011001100110011001100
- An error 0.000000000000000000000011001100... binary, or about 0.000000095 decimal

Cost: 28 soldiers died, 100 injured

Patriot Missle: The Range Gate

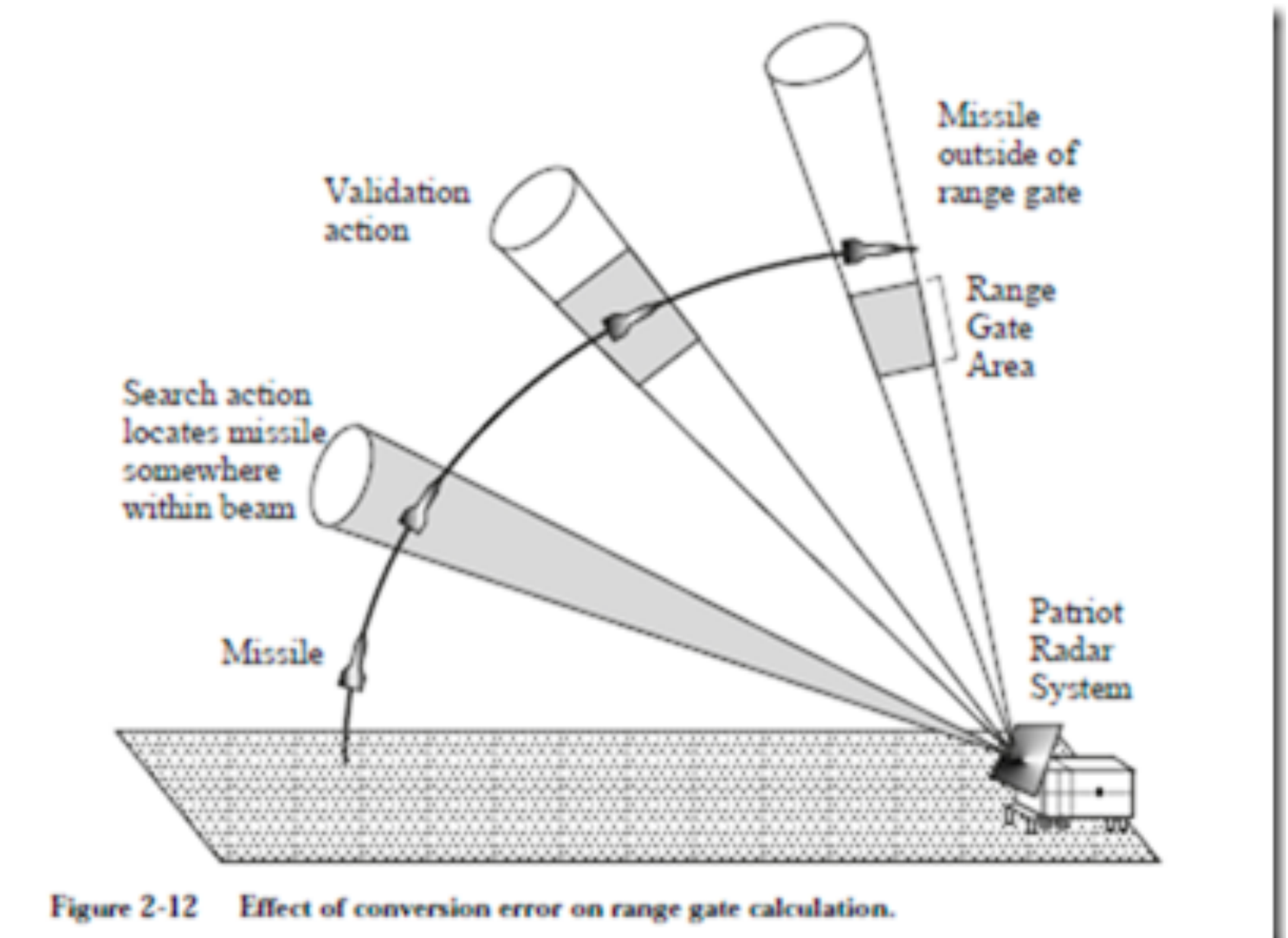


Figure 2-12 Effect of conversion error on range gate calculation.

Viruses / Trojan Horses / Thompson Hack

What can / should we do about malicious programmers?

The Government

What should we expect / demand from our government?

The Government: Whistleblowers

Edward Snowden

American computer professional

edwardsnowden.com

Edward Joseph Snowden is an American computer professional, former Central Intelligence Agency employee, and former contractor for the United States government who copied and leaked classified information from the National Security Agency in 2013. Wikipedia

Is whistleblowing the answer?

The Government: Police and Tech

National Security

Apple vows to resist FBI demand to crack iPhone linked to San Bernardino attacks

See why Tim Cook and Apple are refusing to hack into the San Bernardino shooter's iPhone. (Jhaan Elker/The Washington Post)

What are the ethical issues around tech and privacy?

The Government: Safety and Surveillance

Germany surveillance: Security trumps privacy as video bill passed

10 March 2017 Share

Videos are being increasingly deployed by the German state and private firms - but critics question their impact on crime rates

Public video surveillance appears set to become more commonplace in Germany after the lower house of parliament approved a package of new measures.

What are the ethical issues around tech and privacy?

Ethics and AI

Stanford and Ethics

Silicon Valley + Add to myFT

Stanford to step-up teaching of ethics in technology

University that helped spawn Silicon Valley responds to criticism of tech industry

Marc Tessier-Lavigne: 'We are thinking through the ethics and impact of technological advances.'

Andrew Jack and **Hannah Kuchler** JUNE 3, 2018 5

The university at the heart of Silicon Valley is to inject ethics into its technology teaching and research amid growing criticism of the excesses of the industry it helped spawn.

The board of Stanford University, one of the world's richest higher education institutions with an endowment of $27bn, will meet this month to agree funding and a plan to implement the findings of an internal review that recommends a new initiative focused on "ethics, society and technology" and improved access to those on lower incomes.

https://www.ft.com/content/a374fdac-6589-11e8-90c2-9563a0613e56

Stanford and Ethics

GOOD INTENTIONS OCTOBER 25, 2021 ISSUE

STANFORD TAKES ON THE TECHLASH

With more and more students becoming dorm-room C.E.O.s, three professors cooked up an ethics class for the coding set.

By Andrew Marantz
October 18, 2021

Last month, Reich, Sahami, and Weinstein published a book titled "System Error: Where Big Tech Went Wrong and How We Can Reboot." The introduction recounts the story of Joshua Browder, who "entered Stanford as a young, brilliant undergraduate in 2015." After three months at Stanford, he invented a chatbot to help people get out of paying their parking tickets; within a year he was the C.E.O. of DoNotPay, an "online robot lawyer" startup now valued at more than two hundred million dollars. "He is not a bad person," the professors write. "He just lives in a world where it is normal not to think twice about how new technology companies could create harmful effects"--such as encouraging citizens to stop funding public roads, which tend to crumble when people DoNotPay for them. Browder wrote on Twitter that Sahami and Reich were "my two favorite Stanford professors," and that he was "surprised to learn they spent the entire first chapter bashing DoNotPay."

"Thanks for engaging with the book," Sahami responded.

"I think we should discuss it in person during one of the classes," Browder wrote back. "I will bring the data."