# CS 4873: Computing, Society & Professionalism

Blair MacIntyre | Professor | School of Interactive Computing

## Week 2: Case Study: Therac-25

January 25, 2021

*Slides adapted from Sauvik Das, Munmun de Choudhury, and Amy Bruckman*

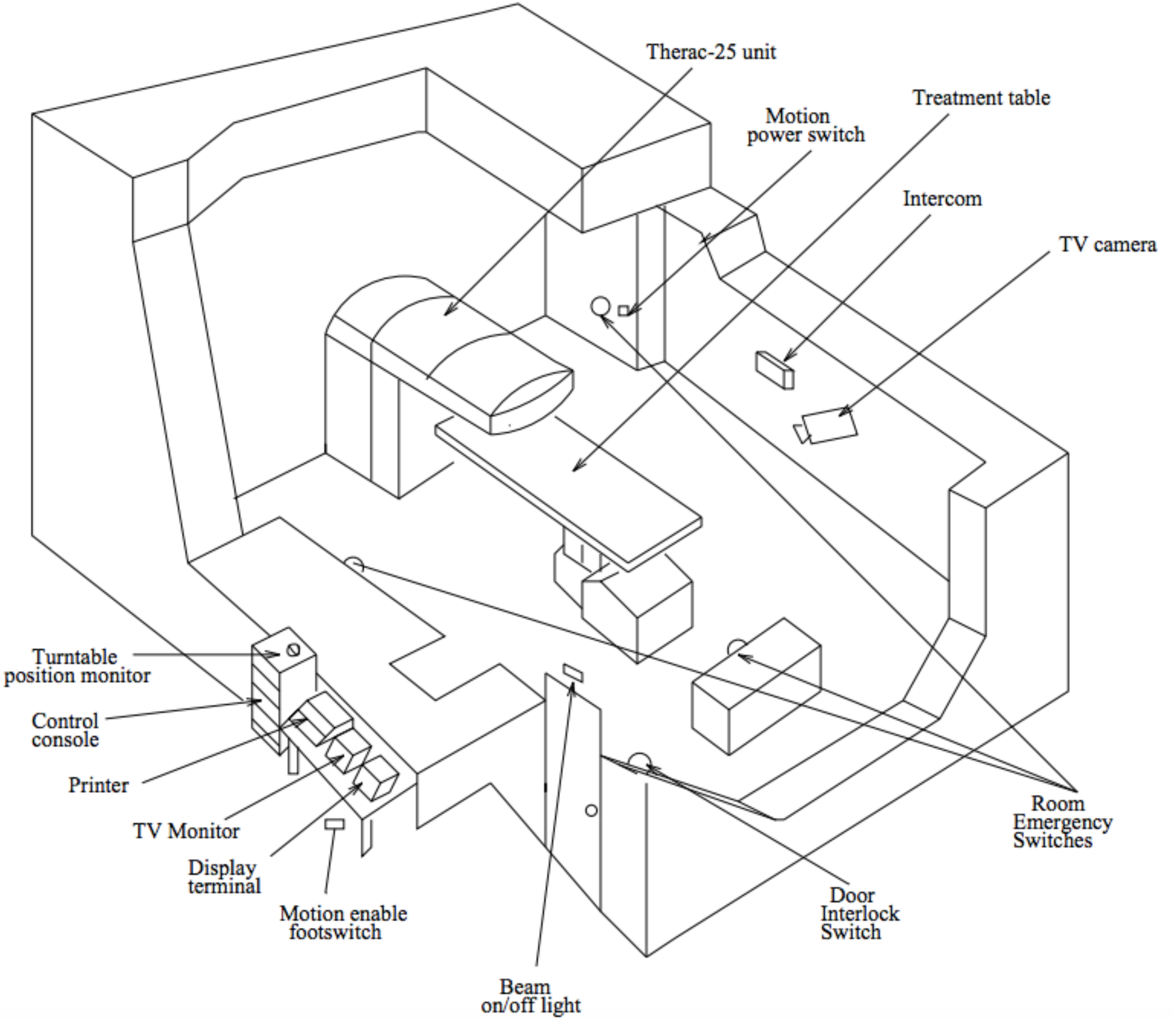
# Homework 1

- Available on Canvas
- Due: Feb 1st, 2020 (11:59pm Eastern Time)
- Submission on Canvas.

# Why you, as a CS major need to know about ethics…

# Genesis of the Therac-25

- Atomic Energy of Canada Ltd (AECL) and French company CGR built Therac-6 and Therac-20
  - Delivered 25 MeV photons or electrons of various energies
- History before Therac-20
  - Software to convenience hardware
- Therac-25 built by AECL
  - Tensions between the two orgs
  - PDP-11 an integral part of system
  - Hardware safety features replaced with software to cut costs
  - Reused code from Therac-6 and Therac-20
  - Compact, economic advantage
- First Therac-25 shipped in 1983
  - Patient in one room
  - Technician in adjoining room


Therac-25 unit
Treatment table
Motion power switch
Intercom
TV camera
Turntable position monitor
Control console
Printer
TV Monitor
Display terminal
Motion enable footswitch
Beam on/off light
Door Interlock Switch
Room Emergency Switches

# Operation

- The radiation software required that **three essential programming instructions** be saved in sequence:
  - first, the quantity or dose of radiation in the beam;
  - then a digital image of the treatment area; and
  - finally, instructions that guide the multileaf collimator.

- Electron mode
  - 5-25 MEV
  - Magnets spread beam
  - Ion chamber monitor

- X-ray mode
  - 25 MEV electrons hit target
  - “Beam flattener” attenuates
  - 100x beam current
  - Ion chamber monitor

- Field-light mode
  - No current
  - Mirror & light used to check alignment
  - No ion chamber (since not treating)

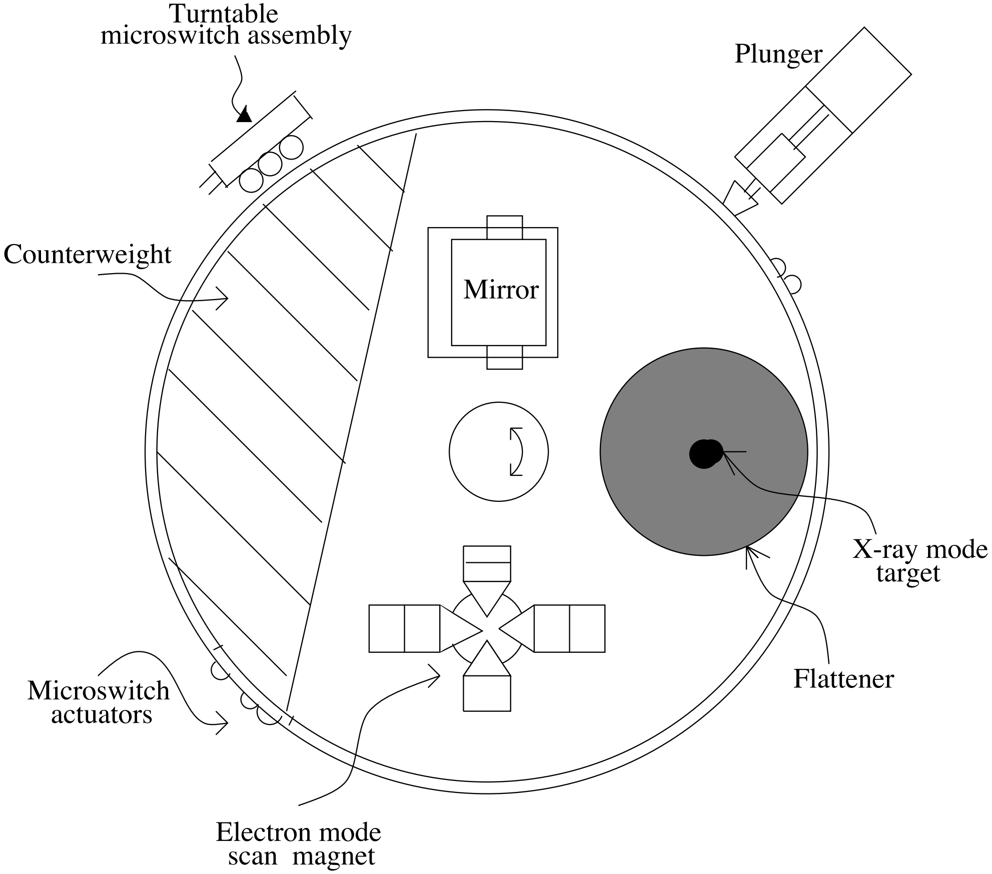
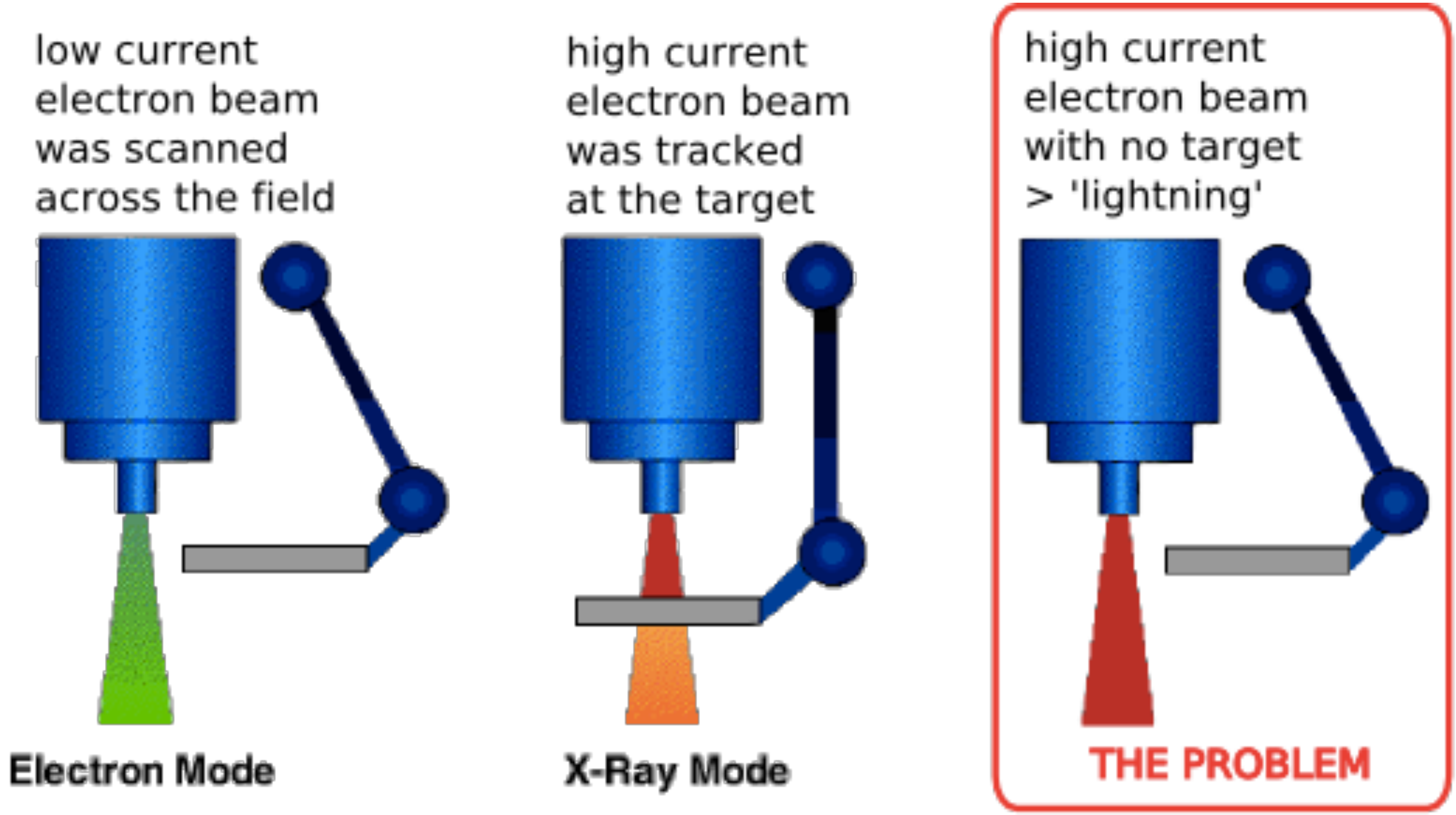


Figure 1: Upper turntable assembly.

# Operation


low current electron beam was scanned across the field
high current electron beam was tracked at the target
high current electron beam with no target > 'lightning'
Electron Mode
X-Ray Mode
THE PROBLEM

# Role of the Software

# The Context

- Radiation therapy
  - Many people with cancer were diagnosed and treated, but were also exposed more radiation than they needed

# The Context

- 11 installed machines (5 US; 6 Canada); 6 major accidents; 3 deaths
  - Improper scanning of the spread of the radiology beam, causing radiation burn and secondary cancer
- Machine recalled in 1987
- Denial – manufacturer and operation refused to believe that the system could make a mistake

# Case 1: Marietta, GA

- Breast cancer patient, receiving therapy on nearby lymph nodes
  - Felt a “tremendous force of heat” when the machine was turned on
  - Technician on site (Tim Still) contacts AECL about possible bug, but was told it was impossible
- Later found out that she received between 15,000 – 20,000 rads (typical dose is 200, 1000 can be lethal if delivered to whole body).
- Shoulder/arm was paralyzed, breast had to be removed

# Case 2: Ontario

- Patient came in for 24th treatment. Operator put in routine dosage
  - Therac shut down after 5 seconds an error message, saying No Dose had been administered. Operator hit “proceed” command to deliver dose.
  - Repeated process 4 times.
- Patient complained of a burning sensation around treatment area (hip)
  - Later hospitalized. Died because of cancer, but would have needed total hip replacement because of radiation overexposure

# Case 3: East Texas

- Experienced operator made a mistake in configuring the treatment
  - Entered “x” for x-ray, when she meant to enter “e” for electron
  - Realized her mistake after entering all the other parameters and fixed the mistake by using keyboard navigation shortcuts
- Audio / video facilities weren’t working that day, so operator couldn’t see patient
- Turned on beam, but the treatment stopped prematurely and reported an underdose. So she proceeded with the treatment.
- Unbeknownst to operator, patient felt strong pain after the first beam and attempted to get up when second beam hit. Was banging on the door to alert her to stop

# What Went Wrong: Gap in End Users' Understanding

- When the computer kept crashing, the operator did not realize that her instructions had not been noticed by system
- Software errors showing dose was not delivered, technician failed to verify

* Another therapist failed to catch the error; errors were cryptic numbers (Malfunction 1 – 64) that provided no indication that a patient might be at risk.

# What Went Wrong: Infrastructural Gaps

- (For the later accidents in the optional NYTimes Reading)

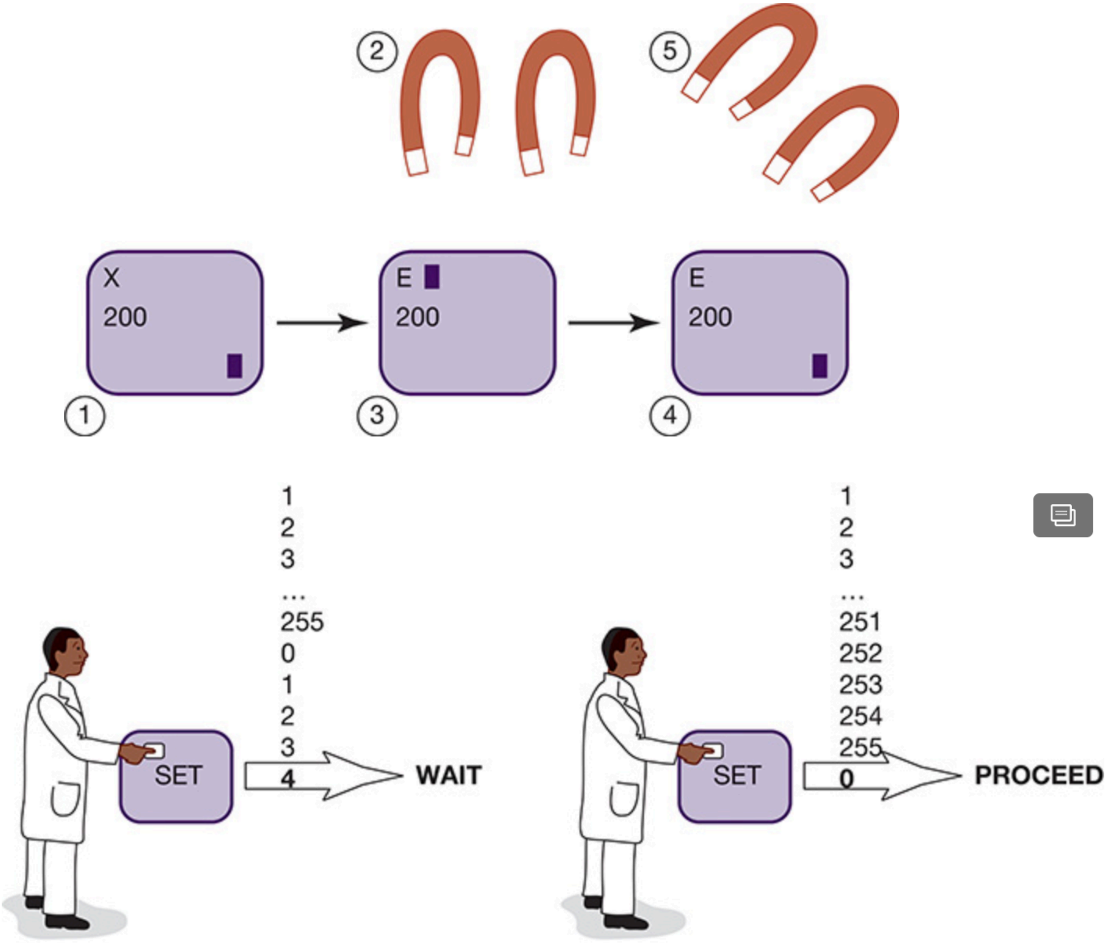
- It was customary — though not mandatory — that the physicist would run a test before the first treatment to make sure that the computer had been programmed correctly. But the hospital had a staffing shortage.

# What Went Wrong: Issues in the Design of Therac-25

- AECL focused on fixing individual bugs not testing the whole system
  - Manufacturer would not believe that machine could fail
- System not designed to be fail-safe
  - Industry standards not followed
  - No proper hardware was installed to catch safety glitches
- No devices to report overdoses
- AECL did not communicate fully with customers
- Lack of communication and organization between hospitals, government and manufacturer

# What Went Wrong: Software Errors

- Race conditions.



# What Went Wrong: A Lack of Fault Tolerance

- One therapist mistakenly programmed the computer for “wedge out” rather than “wedge in,” as the plan required.
- And the physics staff repeatedly failed to notice it during their weekly checks of treatment records.

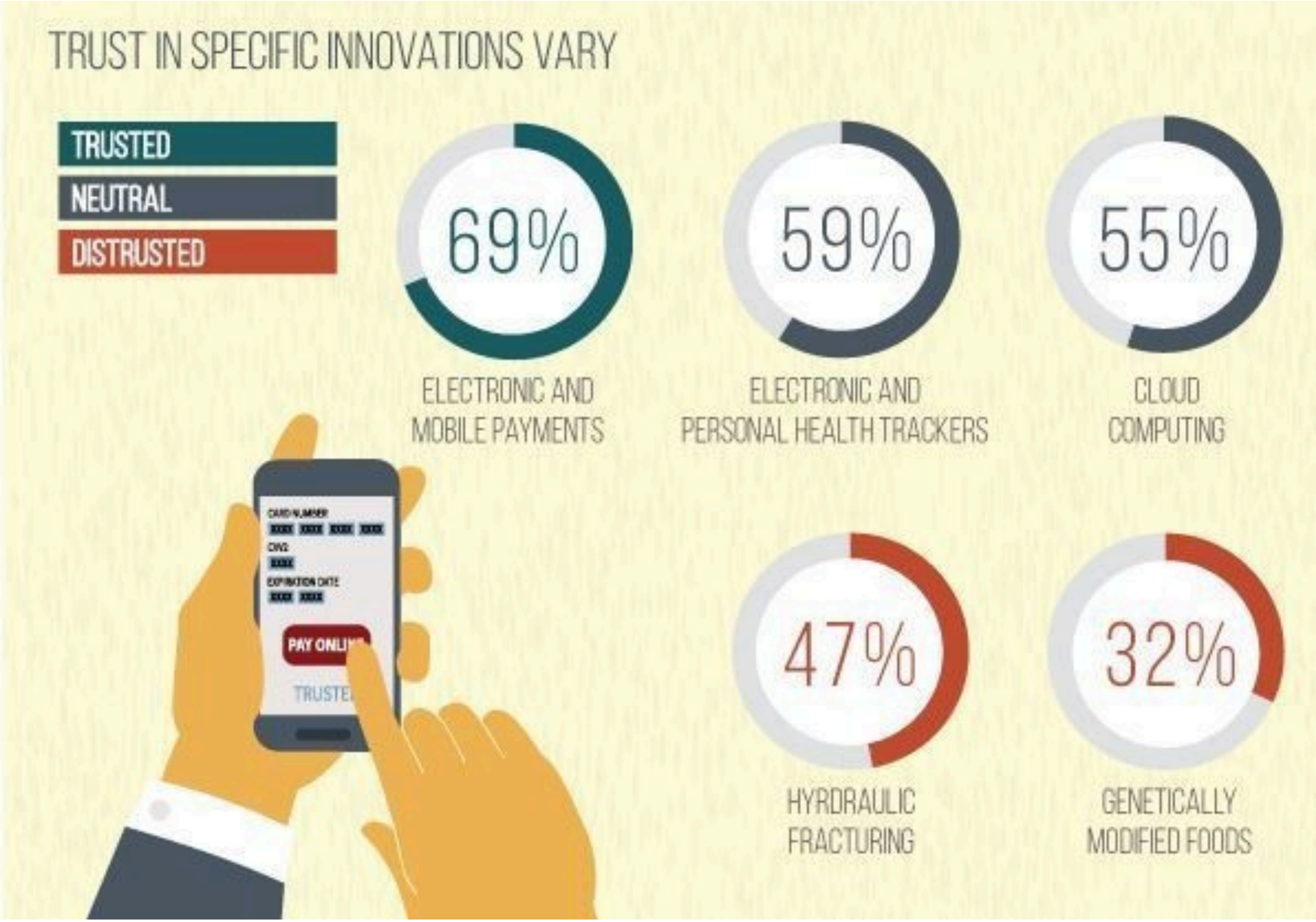
Dr. Howard I. Amols, chief of clinical physics at Memorial Sloan-Kettering Cancer Center in New York:

“Linear accelerators and treatment planning are enormously more complex than 20 years ago. But hospitals are often too trusting of the new computer systems and software, relying on them as if they had been tested over time, when in fact they have not.”

# What Went Wrong: Partial automation, forgetting the human

- Computerization reduced human time needed to calibrate machines and perform safety checks
- But human intervention was still needed to check whether the technology’s software came up with a good treatment solution for a patient

- Don’t Forget the Human!


TRUST IN SPECIFIC INNOVATIONS VARY
TRUSTED
NEUTRAL
DISTRUSTED
69%
ELECTRONIC AND MOBILE PAYMENTS
59%
ELECTRONIC AND PERSONAL HEALTH TRACKERS
55%
CLOUD COMPUTING
47%
HYRDRAULIC FRACTURING
32%
GENETICALLY MODIFIED FOODS
CARD NUMBER
XXXX XXXX XXXX XXXX
CVV2
XXXX
EXPIRATION DATE
XXXX XXXX
PAY ONLI
TRUSTE

# What Went Wrong: Human Factors

- People respond to their work environments. Things that affect judgment:
  - Staffing shortages
  - What is rewarded; what is punished
  - Prior experience
  - Hubris: “We’ve made this 5 orders of magnitude safer!”

- A Philadelphia hospital gave the wrong radiation dose to more than 90 patients with prostate cancer — and then kept quiet about it.
- A Florida hospital disclosed that 77 brain cancer patients had received 50 percent more radiation than prescribed because one of the most powerful — and supposedly precise — linear accelerators had been programmed incorrectly for nearly a year.

# What Went Wrong: Regulation loopholes

- The FDA does not approve each new medical device on the market
- Medical accelerators only required “pre-market notification” — i.e., “firm must establish it is equivalent in safety to a product already on the market.”
  - Because Therac-6 and 20 were already in the market, this was easy for the -25 because, in theory, it had the same safety features (only in software not hardware)
- Only changed to needing approval from FDA *after* first few safety incidents.

# Why Detection is Difficult

- Identifying radiation injuries can be difficult.
- Organ damage and radiation-induced cancer might not surface for years or decades, while underdosing is difficult to detect because there is no injury.
- For these reasons, radiation mishaps seldom result in lawsuits, a barometer of potential problems within an industry.
- Under-reporting of “accidents”

# People involved in the tragedies

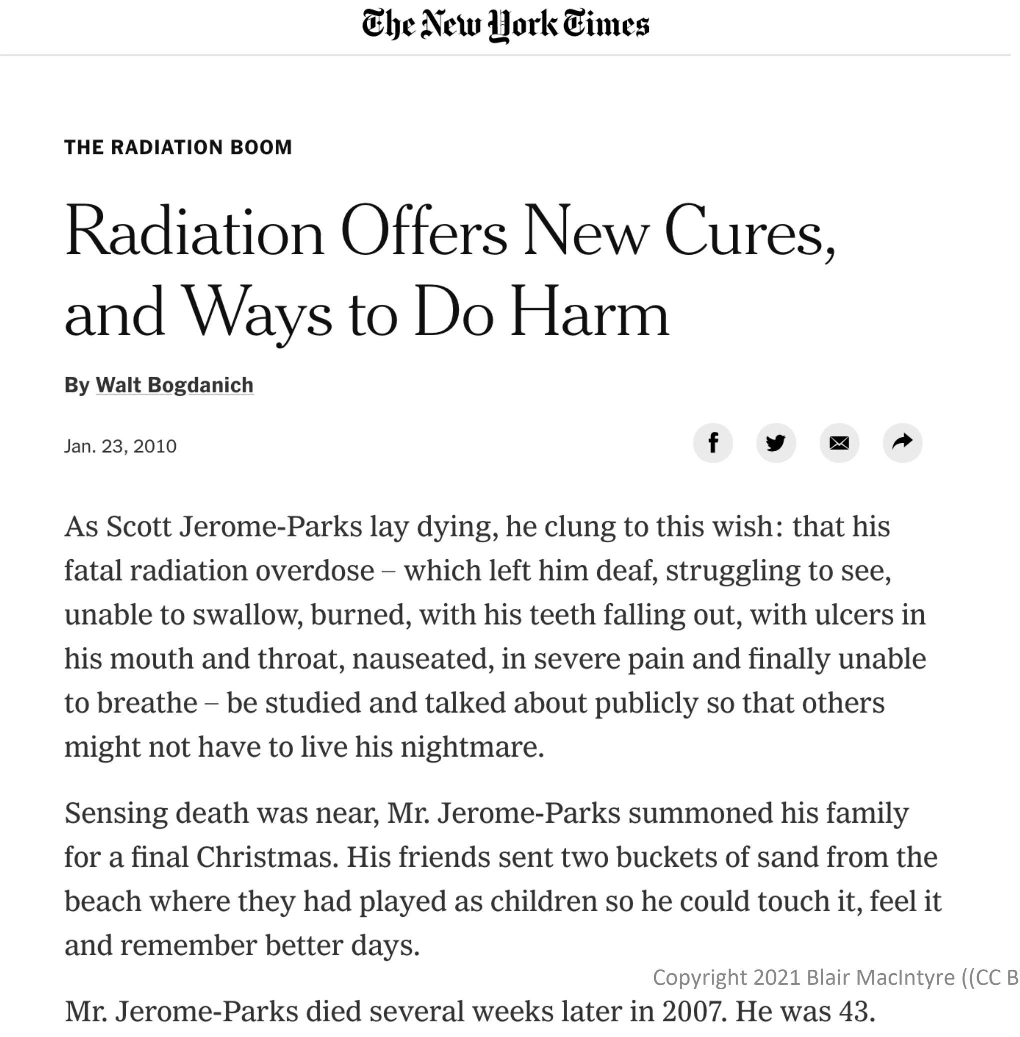
- Company who made the software for the accelerometers
- Programmers and testers behind the software
- Doctors who prescribed medication
- Staff and technicians who managed the accelerometers

** Think about it for your recitation section!

# Post Mortem

- Software lessons
  - Difficult to debug programs with concurrent tasks
  - Design must be as simple as possible
  - Documentation crucial
  - Code reuse does not always lead to higher quality

# That was like 30 years ago …



The New York Times

THE RADIATION BOOM

## Radiation Offers New Cures, and Ways to Do Harm

By Walt Bogdanich

Jan. 23, 2010

As Scott Jerome-Parks lay dying, he clung to this wish: that his fatal radiation overdose – which left him deaf, struggling to see, unable to swallow, burned, with his teeth falling out, with ulcers in his mouth and throat, nauseated, in severe pain and finally unable to breathe – be studied and talked about publicly so that others might not have to live his nightmare.

Sensing death was near, Mr. Jerome-Parks summoned his family for a final Christmas. His friends sent two buckets of sand from the beach where they had played as children so he could touch it, feel it and remember better days.

Mr. Jerome-Parks died several weeks later in 2007. He was 43.

# Solution?

*“Mistakes are a fact of life. It is the response to the error that counts.”*
- American writer and educator Nikki Giovanni

# Lesson: Adopt Systems Thinking

- Don’t just monitor, learn from mistakes
  - “Blame culture” should give way to “Learn culture!”
- A well-designed incident learning system in a radiotherapy program consists of several steps
  - occurrence of incidents,
  - their identification and response,
  - reporting,
  - investigation,
  - causal analysis,
  - corrective actions,
  - Learning
- All included in a cyclic feedback loop
- A nationwide public mandatory reporting system

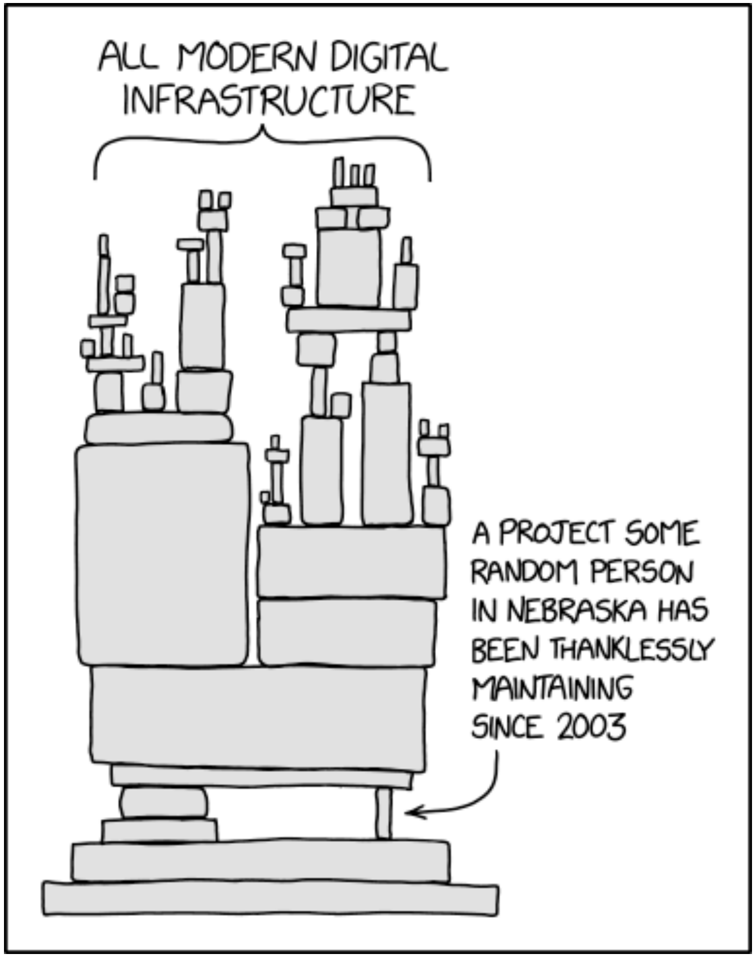
# Lesson: Use Defensive Design

- Designing for when things go wrong.
- Defensive design is the practice of anticipating ways that an end-user could misuse a device, and designing the device so as to make such misuse difficult or impossible (e.g., by eliminating race conditions).

# Lesson: Accounting for the human-in-the-loop

- Automating away parts that are tricky for humans can be great, if well-tested, but many systems will *always* require a human-in-the-loop
- For these situations, it is not enough to write good code—the interface *must* account for human variance and error.
- Human-centered design:
  - Safe defaults
  - Interpretable errors
  - Intuitive design

# Another Lesson Learned ….


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

- The software we produce can have significant upstream implications. We’re not just twiddling bits!
- While most of you may not be working directly on life-critical applications, who knows how your software will be used in the future.
  - Assumptions made for Therac-6 and 20 didn’t hold for 25.
- Check your assumptions, code defensively, and make the easiest path the safest path.