

# Reliability

- Some systems that computers control:
  - Banking; finance; stock market; commerce; e-commerce
  - Medical systems (diagnostics; life support...)
  - Communications systems/networks
  - Buildings (HVAC, security, lights...)
  - Basic infrastructure
    - Energy (power plants; toxic chemical plants; oil & gas pipelines)
    - Water; sewer...
    - Traffic signals, transportation systems
  - Air traffic control, air craft, space craft
  - Military (Command & control; defense & weapons systems--missiles, ships, tanks, ...)
  - Personal and household items

- Many reasons for failure in computer systems:
  - Software “bug”...?
  - Poorly designed software
  - Poorly designed user interface...meaning?
  - Improper use:
    - using system for purpose unintended by creator
    - lack of user training
    - poor documentation
  - Data entry error --Incomplete data
- What might be a cause for *these* failures:
  1. The outrageous phone bill.
  2. Ninth grader's hopes dashed.
  3. Apartment living in L.A.
  4. Plane heads in wrong direction.
  5. Patriot missiles fail to launch.
  6. USS *Vincennes* shoots down civilian airbus.

- And then there's the **Denver Airport's \$193m baggage system** (mid 90s)
  - What did they promise about your luggage?
  - What happened during testing?
  - If you could select **one word** to describe Denver's troubles, what would it be? (*Why is this so hard?*)

## **Therac-25** (*landmark case of how things can go awry*)

- What was this device used for?
- How was its design *fundamentally* different from that of its predecessors (#6, #20) with regard to *safety* features? *...and so what?*

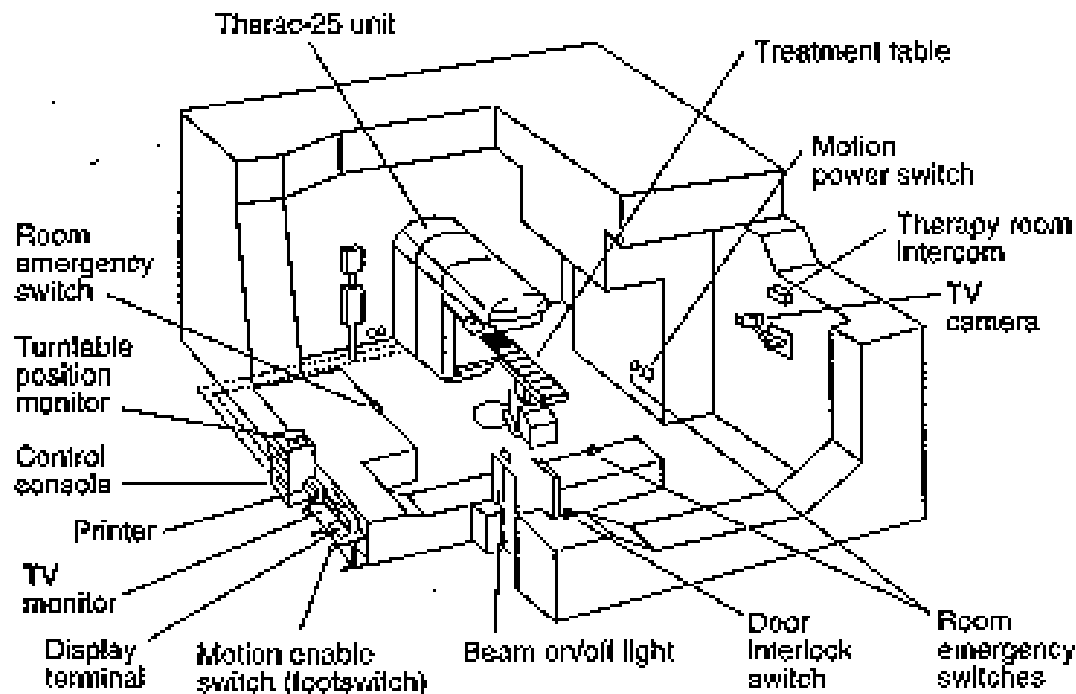
- Understanding Therac:

- How many operating modes did it have?
- Why  $> 1$  mode in *one* machine, do you think?
- *To create X-ray photons ...*
- **1985-1987: Six known accidents ...**

- Early March 1986,  
Tyler, Tx:

- Vernon Kidd  
receives dose  $> 100$   
times prescribed  
dose. *What  
happened that day...*

- What could have prevented  
that---*aside from the*  
operator choosing not to  
proceed?



- AECL engineer: could not reproduce error.
  - *It's not possible for Therac-25 to give an overdose ...*
- Tested by independent engineering firm.
  - *Machine does not appear capable of giving a patient an electrical shock...*
- Put back in use in late March.....then:
  - 2nd acc. in Tyler, Tx, late March (Ray Cox)
    - Same operator, 3 weeks later...
    - *This time, physicist replicated Malfunction 54.*
      - *Data entry speed during editing was  $\leq 8$  seconds ....*
      - *Interesting note here .....*
      - *So what was the crux of the problem in **both** cases?*
    - With *hardware* safety interlocks, instructions are *hardwired into* the hardware; might blow a fuse.

- A second known software design error (bug):
  - Why was “Set-Up” test done before each treatment?
  - What’s a *flag variable*? (in English!)
    - If device NOT ready, what did program do to ensure the variable was not equal to 0?
    - Theoretically, what could happen to a flag variable *value* during testing?
  - The variable was defined by programmer to be *how large*?
  - How large a *decimal value* can an **8-bit byte** represent before it *overflows* (left-most digit is truncated)? Let’s see:

8-bit code:

| <del>256</del> | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | <i>place value</i> |
|----------------|-----|----|----|----|---|---|---|---|--------------------|
| <del>1</del>   | 1   | 1  | 1  | 1  | 1 | 1 | 1 | 1 | = 255              |
| <del>1</del>   | 0   | 0  | 0  | 0  | 0 | 0 | 0 | 0 | = 0                |

- What happened when routine was done the **256th** time?
- So what caused that software “bug”?

- **Digression:**

- Analog systems: very small change in input produces very small change in response.

- EG: bimetallic strip to measure temperature. Won't change or fail catastrophically if there's a slight change in input.

- Digital: How did ONE BIT CHANGE make a difference?

- And can ripple throughout.

- So what would have been a *much better* way to handle that **flag variable**?

- Why so many incidents (six accidents!) before it was finally taken out of service?
  - Several reasons....

## So.... WHO WAS TO BLAME?

- Programmers? What did they do wrong?
- Vendor?
  - “The Titanic Effect”
- Customers (hospitals, clinic staff)?
- The FDA? (related problems here?)

- **AGAIN:** what **SINGLE** word describes why reliability here is so **HARD**?

*“The ethical dimensions of computer reliability are bound up with the nature of software, and the complexity of such systems.”*

- The development process is complex.
  - In a large system, *no one person understands the entire system.*

- Theoretically speaking, what would it take to create perfectly reliable software?
  - *In other words, **when*** would it have to work right?
  - Then *what* would the programmers, and especially the **testers**, need to know?
    - Is that ever possible?
    - **Illustration** of system that monitors performance of nuclear power plants ...
  - Testing: proves the presence of bugs, not the absence!
  - Fixing one bug can introduce others ....

- What BIG question should we ask before we “throw the baby out with the bath water”?
- A more realistic definition of reliable software:
  - “probability that it will not fail during a given period of operation under given conditions.”
  - GOAL: reduce risk *(more shortly)*.
- Another big problem for programmers:
  - Pressure to finish a product and get it to market.

**Why?**

# R I S K

- Is it reasonable to demand zero risk?
- Doesn't hardware ever fail?
- We trust our lives to risky “high tech” tools daily.
  - Any risks with other tools? Such as:
    - Things you **get into**.... Risks?
    - Things you **plug in** and use.... Risks?
  - Any risks with really *low-tech* tools?

- Digression:
  - What do technology critics say?
  - What do others say to dispute that?
    - Is our dependency on computers different from our dependency on other technologies, such as electricity? The plow?
    - Are mistakes in software the *same* as those that occur with, say, electricity? Why or why not?
- How can we avoid risk of a tool altogether?
  - Elevator?      Auto accident?

- When any tool breaks down, what does it remind us of?
- Why do we use RISKY tools?
- When should negative effects condemn a tool?
- Some tough questions ....
- What were some lessons learned here?

*"I'm happy to work on games...; **critical** systems are scary...But I **would** like to make a difference."*