Software Risks

Wednesday, November 5

Therac-25

A radiation therapy machine: cancer is treated by irradiating it

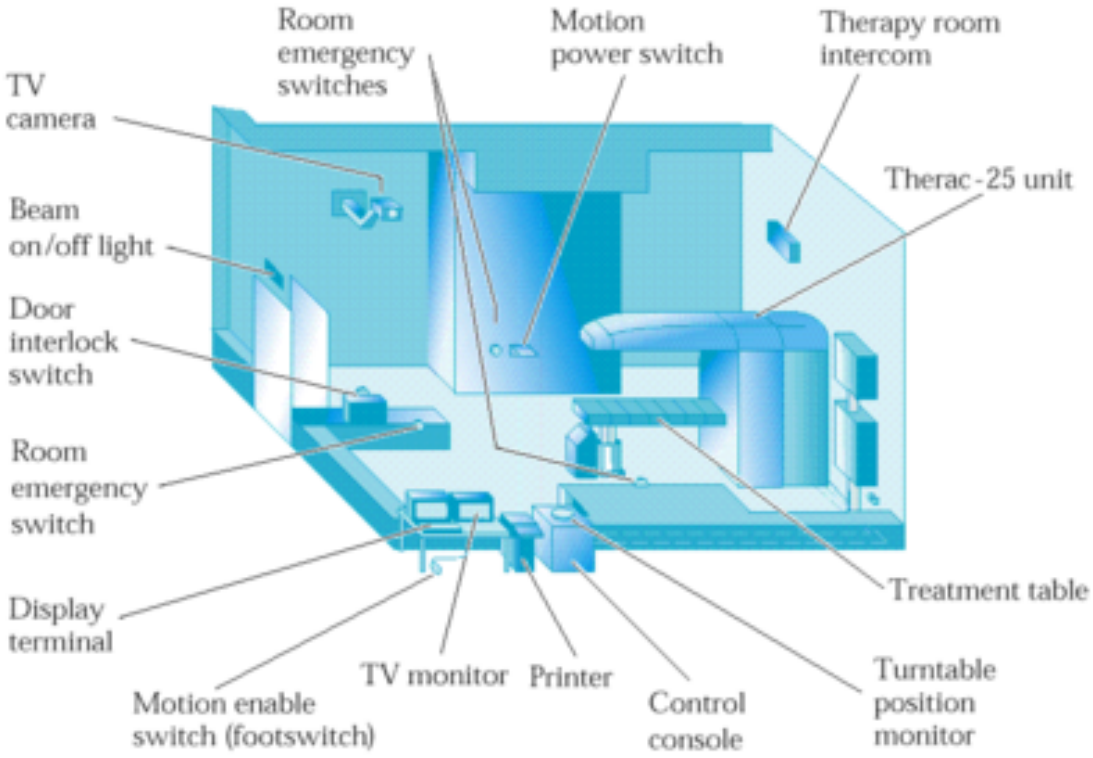
6 accidents, 4 deaths, but 100's of lives saved by the treatment delivered

Normal dose was 80–180 rads

A lethal whole-body dose is ~500 rads

(Read Page 10 Description)

The first time in history that software killed someone!




VARIAN
CLINAC

What Did They Do?

Therac-25 was an upgrade to Therac-20 that replaced hardware interlocks with software

The software to run the machine was a custom operating system for a PDP-11/23

- Custom scheduler that had race conditions (no atomic test & set operations)
- Different processes communicated through shared memory
- 8-bit counters used as signals that overflowed

The user interface was misleading

- Keyboard input accepted edit operations, but they didn't behave as expected
- Default values caused undesired behavior (convenience compromised safety)
- Error messages were inscrutable
- The system paused when something went wrong, but there were lots of errors

What Didn't They Do?

Documentation: Development and behavior should be documented

Concurrency: Use techniques that are easy to understand for difficult problems

Interface: Input–output relationship should not be surprising

Testing: A plan for how to test components and interactions should be part of design

Auditing: Should have an independent and robust way of tracking what actually happened

Modularity: For example, a clean interface between input and execution

Simplicity: Software designs should be simple

Redundancy: No single component failure should be able to cause an accident

What Did the Food and Drug Administration Do?

One of the first cases of the FDA handling a software problem

The machine wasn't removed from service until the manufacturer proved unable to fix it

Plans for correcting the software received reasonable feedback

The FDA involved user feedback to its decision process