

CSci 780  
Advanced Software Engineering

---

“An Investigation of the Therac-25  
Accidents”

By  
Nancy G. Leveson  
Clark S. Turner

9/10/04

1

Outline

---

- Deadly Consequences
- Background
- Causes
  - Bad Design
  - Bad Decisions
- Lessons
- Discussion

9/10/04

2

Deadly Consequences

---

- Between June 1985 and January 1987, six people were overdosed with radiation by the Therac-25 radiation treatment system
- Three died
- Three suffered serious and permanent injury

9/10/04

3

Quotes

---

- “tremendous force of heat... this red-hot sensation”
- “electric tingling shock”
- “developed erythema... in a parallel striped pattern”
- “felt like his arm was being shocked by electricity”
- “‘fire’ on the side of his face”
- “‘burning sensation’ in the chest”

9/10/04

4

Manufacturer Response: Denial

---

- Engineers explain that improper scanning was impossible
- $10^5$  improvement in safety
- Letter explains that overdose is impossible
- AECL knew of no other accidents
- Could not reproduce the error
- “Machine is working perfectly”

9/10/04

5

Background

---

- rad: **R**adiation **A**bsorbed **D**ose
  - Single doses in the 200 rad range
  - Patients suffered an estimated 15000-20000 rads
- MeV: **M**illion **E**lectron **V**olts
- Medical linear accelerators (linacs) accelerate electrons to create high-energy beams to destroy tumors

9/10/04

6

## Background (cont)

---

- Use accelerated electrons for shallow tissue
- Use X-ray photons for deeper tissue
- Minimal impact from the treatment on surrounding tissue
- Health care professionals not required to report incidents to manufacturers (until 1990)

9/10/04

7

## History of Theracs

---

- Therac-6
  - 6 MeV developed by AECL & CGR
  - PDP 11, Standalone without a computer
  - Hardware safety features
- Therac-20
  - 20 MeV developed by AECL & CGR
  - Hardware similar to Therac-6

9/10/04

8

## History of Theracs (cont)

---

- Therac-25
  - 25 MeV developed by AECL
  - Dual electron and photon capabilities
  - Designed to be controlled by computer
  - Many hardware safety features removed
  - Double pass electron acceleration
- Software
  - Monitoring machine status
  - Accepting and processing treatment input
  - Turning on and off the beam

9/10/04

9

## Software Development Process

---

- Most testing done as system testing
- Minimal unit and software testing
- More trust in software than hardware
- No accident protocol

9/10/04

10

## Faults

---

- Race conditions resulting in inputs not being correctly modified in the program
- Editing keys causing unintended results
- The turntable's actual position does not always match what the console indicates

9/10/04

11

## Software Design

---

- Little documentation on software:
  - Development
  - Specifications
  - Test Plan
- One person designed the software
  - PDP 11 assembly language
  - Several years to develop

9/10/04

12

## User Interface Design

- Operator Interface: ease over safety
  - The 'P' Key: too easy to proceed after malfunction
  - Editing keys: too easy to edit and set-up treatment
- Cryptic malfunction codes and no manual

9/10/04

13

## System Design

- No check to ensure turntable in correct position
- Missing safety system to check to saturated ion chambers
- No hardware safety systems

9/10/04

14

## Inadequate/Late Response

- Fail to ensure adequate software QA
- Failed to check compatibility of software modifications
- Focused on hardware issues (microswitch)
- Expanded test plan to hardware and software
- Visual inspection of turntable position
- Finally a recall and hardware interlocks added

9/10/04

15

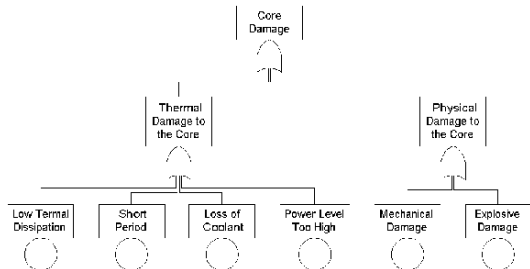
## Failure/Reliability Analysis

- Failure mode and effect analysis
  - Describe system response to all failure modes of individual system components
- Fault-tree analysis
  - Identify component failures leading to system failure
- Software inspection
  - Examination of functions' implementations
  - Search for coding errors
  - Qualitative assessment of reliability

9/10/04

16

## Example Fault Tree



9/10/04

17

## Lessons

- Complex systems have complex faults
- Software cannot be trusted
- Assigning blame is not helpful
- Safety-critical systems require exceptional software development methods
- Full system analysis is necessary

9/10/04

18

## Lessons (cont)

---

- Documentation is critical
  - Test software, and hardware+software
  - Need quality assurance guidelines and policies
  - Traceability of faults, development, etc.
- 
- Response all too familiar

9/10/04

19

## Where to go from here?

---

- Columbia: "Crater" used to determine that falling foam does not pose a significant risk
- Hatton discovers substantial differences in "identical" geological analysis software
- NRC warns operators of untrustworthiness of software analysis tools
- Dugan discovers faults in DFT tools:
  - PAND when really AND
  - Shared inputs treated as non-shared

9/10/04

20

## Discussion

---

- Responsibilities? Manufacturer, FDA, Hospitals, Operator
- What would you have done during development? During the failures?

9/10/04

21