Medical Devices: Therac-25

Author: Nancy Leveson

Therac-25

- Therac-25 was a radiation therapy machine.
- AECL and CGR built Therac-6.
- Therac-20 with dual mode.
  - –Could be operated without computer.

- AECL developed Therac-25.
  - – Could be operated only through computer.
  - – Some hardware safety mechanisms were replaced with software.
  - – First working prototype: 1976
  - – First commercial product: 1982

Errors

- Treatment suspend
  - Requires complete machine restart
- Treatment pause
  - Operator type “P” to proceed
- HTILT, VTILT, etc.
- MALFUNCTION <n>
- No documentation
- No indication of severity

- Occurred on average 40 times a day!

The Problem

- Between June 1985 and January 1987,
  - 3 died
  - 3 experienced permanent health/physical effects
- Because of overdose.

Lessons Learned

- 11 lessons learned
- Based on government records and material obtained form FDA.
- We will cover 6 of them.

Overconfidence in Software

- The first safety analysis did not include software, even though it was responsible for safety of the system
- When problems did occur, it was assumed to be a hardware failure

Reliability Vs. Safety

- Ran for 3 years without a problem.
- Reliability led to complacency.
- Reliability != safety

Lack of Defensive Design

- No self-check, error-detection or error-handling.
- Limited audit trails.

Unrealistic Risk Assessments

- First risk Assessment did not include software.
- claimed 5 orders of magnitude improvement from changing one micro switch.
- Software is harder to assess for failures than hardware

Inadequate Software Engineering Practices

- Dangerous design/coding practices could have been avoided.
- Software should be tested at the unit, module and software level.
- Specification and documentation should not be an afterthought.
- Regression testing on all changes

Software Reuse

- Therac-25 used modules from Therac-20.
- Safety is the quality of the system, not software.
- Sometimes, better to rewrite from scratch.

Other Factors

- Failure to eliminate the root cause.
- Inadequate investigation.
- Safety Vs. user friendly interface.
- User and government oversight and standards.

Conclusion

- Safety is part of the design and development process.
- Reliability != Safety.
- Software failures are harder to detect.
- Fault tolerance plays an important role.
- Software can fail !!


Question?