Therac-25 Accidents

CSE 403 - Spring 1997

Therac-25 Accidents

- Therac-25 is a medical linear accelerator.
- Between June 1985 and January 1987, it has been blamed for six known accidents involving massive radiation overdoses.
- What can we learn from this?
- What should we do, if we are ever to be involved in a project like this?

Genesis of the Therac-25

- In collaboration with CGR, AECL developed medical linear accelerators, Therac-6 and Therac-20.
  - Therac-6: 6MeV X-ray only.
  - Therac-20: 20MeV dual mode.
  - Both of them essentially the result of augmenting original system with additional computer control using DEC PDP-11.
  - I.e., hardware safety features and interlocks.
- CGR and AECL breaks up.
- AECL is to design Therac-25
  - Therac-25: 25MeV dual mode linacs.

Genesis of the Therac-25

- AECL designed the Therac-25 to take advantage of computer control from the outset.
  - Rely more on software for monitoring electron-beam scanning, plus interlocks to ensure safe operation.
- Started from the Therac-6 package to develop the software.
  - Some of the Therac-20 routines obviously slipped in.

Accidents

- Between June 1988 and January 1987, there were six known cases of radiation overdose involving Therac-25.
- Traced to two types of software defects, eventually.
  - There were some false starts too.

System Engineering Issues

- Management inadequacies and lack of procedures for following through on all reported incidents.
- Overconfidence in the software and removal of hardware interlocks (making the software into a single point of failure that could lead to an accident).
- Presumably less-than-acceptable software engineering practices, and
- Unrealistic risk assessments along with overconfidence in the results of these assessments.

Software Engineering Issues

- Documentation should not be an afterthought.
- Software quality assurance practices and standards should be established.
- Designs should be kept simple.
- Ways to get information about errors---for example, software audit trails---should be designed into the software from the beginning.
- The software should be subjected to extensive testing and formal analysis at the module and software level; system testing alone is not adequate.

Software Engineering Issues

- Safety must be built into software, and, in addition, safety must be assured at the system level despite software errors.
- Reuse and reliability.
- Usability vs. safety.