**The Therac-25:

Software Fatal Failure**

***Presented to:***

***Prof. Lawrence Chung***

**In partial fulfillment for the requirements for**

**SYSM 6309: Advanced Requirements Engineering**

**Spring ‘2012**

***By: Aagbara Kpea***

**University of Texas at Dallas**

**ABSTRACT**

Therac-25 was a medical linear accelerator, a device used to treat cancer. Therac-25 was different from its predecessors, therac-6 and therac-20 at the time of its use because of it was software oriented. The software made the laborious operational set-up process much easier and helps to monitor the safety of the machine. In this case on safety critical software, some patients received much more radiation than prescribed despite the software safety programming. Some of these overdoses resulted is fatalities.

**INTRODUCTION**

The Therac-25, a computerized radiation therapy machine, massively overdosed patients at least six times between June 1985 and January 1987. Each overdose was several times the normal therapeutic dose and resulted in the patient's severe injury or even death. Although, the overdoses sometimes involved operator error, occurred primarily because of errors in the Therac-25's software and because the manufacturer did not follow proper software engineering practices.



Overconfidence in the ability of software to ensure the safety of the Therac-25 was another important factor which led to the accidents. The Therac-20, a predecessor of the Therac-25, employed independent protective circuits and mechanical interlocks to protect against overdose. The Therac-25 relied more heavily on software. Moreover, when the manufacturer started receiving accident reports, it was unable to reproduce the accidents, assumed hardware faults, implemented minor fixes, and then declared that the machine's safety had improved by several orders of magnitude.

The design of the software was itself unsafe. The Therac-25 supported a multitasking environment, and the software allowed concurrent access to shared data. This precarious implementation caused program failure under certain conditions.

Risk assessments were, from the start, unrealistic. A risk assessment performed by the manufacturer seems to consider only hardware failures as it lists the possibilities of the computer selecting the wrong energy or mode as 1e-11 and 4e-9 respectively. Justification never appears for these numbers, but, more surprisingly, the company accepted this low risk assessment easily.

Follow-through on accident reports was unacceptable. After one accident, the manufacturer tried to reproduce the condition which occurred at the treatment. When it could not, it concluded that a hardware error caused the accident, and implemented a solution based on that assumption. It declared that the system was several orders of magnitude safer, but accidents did not cease.

The Therac-25 incidents demonstrate that several misconceptions in the manufacturer's attitude led to the accidents. Poor software design, overconfidence in the software's abilities, unreasonably low risk assessments, and poor manufacturer response to complaints all contributed to the overdoses. Companies must understand that for safety-critical software design rigorous testing and failure analyses are essential and that trained software engineers, not simply any reasonably experienced engineers, should implement the software design.

**BACKGROUND OF THE THERAC-25**

In early1970’s, AECL (Atomic Energy of Canada Limited) and a French Company (CGR) collaborated to build Medical Linear Accelerators (linacs). It was a generational medical break-through for cancer patients. They developed two models- Therac-6, and Therac-20.

AECL and CGR end their working relationship in 1981. In 1976, AECL develops the revolutionary "double pass" accelerator which leads to the development of Therac-25. In March, 1983, AECL performs a safety analysis of Therac-25 which apparently excludes an analysis of software. July 29, 1983, the Canadian Consulate General announces the introduction of the new "Therac 25" Machine manufactured by AECL Medical, a division of Atomic Energy of Canada Limited. Medical linear accelerator (linacs) was known generally as “Therac-25”.

**HOW THERAC-25 WORKS**

**How to generate the beam:**

Early radiation therapy machines used a radioactive source like cobalt to produce the ionizing radiation needed to treat cancerous tissue. Some machines still use an active radiation source. But most radiation therapy today is done with a linear accelerator. In principle, a linear accelerator works just like the computer monitor. The electrons are accelerated by the gun in the back of the monitor and directed at the inside of the screen, where phosphors absorb the electrons and produce light. A medical linear accelerator produces a beam of electrons about 1,000 times more powerful than the standard computer monitor. The longer a linear accelerator is, the higher the energy of the beam it can produce. The innovation of Therac 25 was that the designers found a way to fold the beam back and forth so a very long accelerator could be fit into a smaller space. Thus powerful beams could be produced, but within a reasonable amount of space

**Getting beam into the body:**

Patients can be treated directly with the resulting electron beam, as long as the beam is spread out by scanning magnets to produce a safe level of radiation. The medical linear accelerator spreads and directs the beam at the appropriate place for treatment. The picture below shows a typical medical linear accelerator in operation.

To solve this problem, Therac-25 and many other machines can switch to a mode in which X-ray photons are used for treatment. These penetrate much more deeply without harming intervening tissue. To do this, the electron beam is greatly increased in intensity and a metal foil followed by a beam "flattener" is placed in the path of the electron beam. This transforms the electron beam into an X-ray (called photons in some literature). This process is inefficient and requires a high intensity electron beam to produce enough X-ray intensity for treatment. Therac-25 used a 25 MeV electron beam to produce an X-ray for treatment. 25 MeV is 25 million electron volts (eV -- an eV is the energy needed to move one electron through a potential of one volt).

****

**ACCIDENTS WITH THE THERAC-25**

**At Tyler, Texas facility:**

Two accidents occurring in rapid succession provided the first clues to what was happening. On March 21, 1986, a patient was being irradiated for the ninth time at the East Texas Cancer Center in Tyler, Texas, for a tumor that had been removed from his back. The centre's Therac-25 had already successfully treated more than 500 patients over a two-year period. The patient lay on his stomach on the table in the treatment room, which was connected to the computer console room next door by an intercom and video monitor. On this day the intercom was broken and the video monitor was unplugged. The technician left the treatment room and shut the door. At the computer console the operator typed in the prescription data for an electron beam of 180 rads, then noticed an error by typing in command x (for x-ray treatments) instead of e (for electron). The operator ran the cursor up the screen to change the command x to e, as the patient’s prescription required. Everything verified and beam turned on. The machine stopped and the computer screen flashed "Malfunction 54," a mysterious message not even mentioned in the Therac-25 manual.

**At Yakima Valley, Washington Hospital:**

Another accident occurred at the Yakima Valley Memorial Hospital. On January 17, 1987 an operator placed a patient on the turntable in the field-light position for small position verification doses. After attempting to administer the treatment dose, the machine shut down with a quick malfunction message and a treatment pause. The operator pushed the "P" button, and the machine paused again. The machine indicated that the patient had received his prescribed 7 rad of treatment. The patient, however, complained of a "burning sensation" and died three months later from complications related to the overdose.

**ROOT CAUSE ANALYSIS**

The primary reason was attributed to the bad software design and development practices, and not explicitly to several coding errors that were found. In particular, the software was designed so that it was realistically impossible to test it in a clean automated way.

Researchers who investigated the accidents found several contributing causes. These included the following institutional causes:

* AECL did not have the software code independently reviewed.
* AECL did not consider the design of the software during its assessment of how the machine might produce the desired results and what failure modes existed. These form parts of the general techniques known as [reliability modeling](http://en.wikipedia.org/wiki/Reliability_modeling) and [risk management](http://en.wikipedia.org/wiki/Risk_management).
* The system noticed that something was wrong and halted the X-ray beam, but merely displayed the word "MALFUNCTION" followed by a number from 1 to 64. The user manual did not explain or even address the error codes, so the operator pressed the P key to override the warning and proceed anyway.
* Due to overconfidence in the software, AECL personnel were having doubts believing operators of the equipment.
* AECL had never tested the Therac-25 with the combination of software and hardware until it was assembled at the hospital.

**REQUIREMENTS ISSUES**

Each bug contained in the Therac-25 software was also found in the software of the Therac-20. However, the hardware safety interfaces in the Therac-20 prevented any accidents from occurring in the other machine. Although, an overall poor engineering process was used, the main weakness of the Therac-25 software is in the lack of specifications and formal testing procedures. As a result, certain bugs remained in the software as the product was distributed to users. In this section, I will discuss the software bugs that contributed to the Therac-25 accidents.

The Therac-25 software errors that cause radiation overexposures can be reduced down to interface errors, a race condition. The first of these errors involved the entering of treatment data by the machine operator. Once an operator enters treatment information at the terminal outside of treatment room, the magnets used to filter and control radiation levels are set. There are several magnets, and the process takes about 8 seconds. If the operator makes a very, very quick change of the treatment information, within 1 second, the change is registered. Or, if the operator is rather slow about it, takes more than 8 seconds, the change is also registered. However, if the change occurs within the eight seconds it takes to set the magnets, the change is not detected and the magnets continue to be set up improperly, and thus the level of radiation is set up improperly. Once the magnets are set, there is no test performed to double check that the treatment information entered matches how the magnets are set. Another variable, which controls whether photon or electron mode is to be used, does detect the operator edit and sets the mode to the edited mode. As mentioned earlier, much higher levels of radiation are needed in photon mode to produce the same levels of output in electron mode. Therefore, if the beam is set for photon mode, but the turntable is set up for electron mode, a radiation overdose occurs.

The other software errors, causing the accidents, involved the nature of the real-time system. When the turntable is in test mode, a variable called Class3 is set to a non-zero value. As long as the operator is testing the position of the light beam, the variable increments. Once testing procedures are complete, the variable is set to zero and the radiation beam is allowed to pass. Class3, however, was stored in one byte of memory. As a result, every 265th increment results in the value of zero assigned to it. In one of the accident, the operator pushed the set button at the exact moment that Class3 rolled over to zero. As a result, a full prescription beam was released without any of the beam flatteners in place.

There was no proper traceability matrix in place. No established ways to get information about errors, i.e., 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.

**OVERVIEW OF THE ACCIDENTS RACE CONDITION**

****

****

**CORRECTIVE ACTION PLAN AND LESSON LEARNED**

Many factors contributed to the cause of the Therac-25 accidents. AECL was responsible for many of the factors. AECL used an overall, poor engineering method. For example, only one programmer was assigned to creating the machine's complex, real-time software. Below are the corrective action plan that were recommended:

* Documentation should not be an afterthought.
* Software quality assurance practices and standards should be established.
* Designs should be kept simple and ensure user-friendly interfaces
* Complacency
* Assumption that problem was understood without adequate evidence
* Sole reliance on software for safety
* Systems engineering practices need proper coordination

Formal software specifications and testing criteria were not written. In addition, very little software and system testing was performed. To achieve the levels of machine safety as proclaimed by AECL's engineers, the machine would have had to run and been tested properly. In addition, when old bugs are fixed in a program, new bugs tend to crop up. AECL's claims of safety improvement by "orders of magnitude" after machine fixes were completely unfounded. In addition, AECL's denial that the machine could malfunction, despite finding and fixing several problems, was groundless. As long as AECL was convinced that their machine could not cause a radiation overdose, they were not going to discover any machine deficiencies.

**REFERENCES**

* The Therac-25 Accidents (PDF), by [Nancy G. Leveson](http://en.wikipedia.org/wiki/Nancy_G._Leveson) (the 1995 update of the IEEE Computer article)
* http://en.wikipedia.org/wiki/Therac-25
* Jacky, Jonathan. "Programmed for Disaster." The Sciences 29 (1989): 22-27.
* Leveson, Nancy G., and Clark S. Turner. "An Investigation of the Therac-25 Accidents." Computer July 1993: 18-41.