

## **Software Failure Case Study.**

**Title:** The Therac-25 Radiation Therapy Machine

### **Introduction:**

The Therac-25 was a cutting-edge radiation therapy machine in the 1980s. It had a powerful computer, but software flaws led to six cases of patients receiving dangerously high radiation doses, resulting in deaths. The machine was recalled in 1987 for significant safety enhancements, including software and safety features. Insufficient accident reporting and investigations compounded the issue, highlighting the importance of rigorous safety measures and monitoring in medical technology.

### **Failure and Consequences:**

1. **Software Control Flaws:** The software controlling the Therac-25 was susceptible to programming errors, specifically data entry issues. In some cases, users inadvertently input incorrect treatment parameters, leading to disastrous outcomes.
2. **Inadequate Error Handling:** One of the most troubling aspects was the software's inability to provide adequate error messages. When incorrect data or operating conditions were detected, the machine simply displayed a "MALFUNCTION" message, which did not specify the nature of the problem. As a result, operators often could not identify the precise issue or take corrective action.
3. **Overexposure Incidents:** The lack of safeguards led to the machine delivering massive radiation overdoses to patients. Several patients experienced severe injuries, including burns and radiation poisoning, while some lost their lives.
4. **Delayed Response:** The manufacturer's response to these incidents was notably delayed. It took considerable time for the severity of the

software issues to be recognized and for remedial actions to be taken.

**Causes:**

1. **Software Bugs:** The Therac-25 software contained critical bugs that were not identified during the development and testing phase. These bugs manifested in various ways, such as incorrect data entry handling.
2. **Inadequate Hardware Safeguards:** Unlike its predecessors, the Therac-25 had limited hardware interlocks to prevent dangerous operating conditions. The absence of these physical safety measures made it heavily reliant on the software to prevent errors.
3. **Operator Training:** The operators were not sufficiently trained to use the machine or to recognize and respond to software-related issues. This lack of training compounded the problem.

**Lessons Learned:**

1. **Safety First:** In safety-critical systems, such as medical devices, safety must always be the paramount concern. Adequate hardware and software safeguards are essential to prevent life-threatening errors.
2. **Thorough Testing:** Rigorous and comprehensive testing is crucial in the development of software for such systems. This should include a focus on potential failure scenarios and how the software responds to them.
3. **Clear Error Messages:** Software should provide clear, specific error messages to alert operators to problems. Ambiguity in error messages can lead to confusion and incorrect actions.

4. **Regulation and Oversight:** This incident significantly contributed to increased regulatory scrutiny of medical device software. It led to the development of stricter standards and regulations for such systems.

The Therac-25 case is a stark reminder of the devastating consequences that can result from software failures in critical systems. It emphasizes the need for comprehensive testing, robust safety mechanisms, clear communication, and effective regulatory oversight to prevent catastrophic software errors, particularly in life-critical domains such as healthcare.

**References:**

1. **Article from ethics unwrapped(University of Texas):**  
<https://ethicsunwrapped.utexas.edu/case-study/therac-25>