# CSP 587 - Software Quality Management

# Homework #1


## Name: Abhiram Ravipati

## Student ID: A20539084


## Therac-25 Case:

Therac-25 is a radiation therapy machine developed by AECL (Atomic Energy of Canada Limited) that resulted in multiple radiation overdose deaths and major injuries between 1985 and 1987. Multiple system failures and inadequate testing, error-handling techniques in the system's architecture are the root cause for this problem. Here we are analysing several important factors which could have helped in designing a better and reliable system.

## 1. UI/UX Design:

### Failures:

The user interface of the Therac-25 made it easy for the operators to easily edit treatment parameters without proper verification. The error messages were ambiguous and did not clearly indicate the severity of problems. For Example, Malfunction 54. The system did not have clear confirmation steps for critical actions. In most of the cases, the operators were not aware that the machine was malfunctioning which led the operators to give repeated inputs to the system and errors which produced dangerous doses of heavy radiation.

### Best SQM Response:

A good SQM approach would be focusing on user-centric design of the system. During the development of user interface, it must be ensured that the error messages displayed are clear and important and critical information cannot be easily overridden by the operator. The user interface must be designed in a way that the operator can easily understand which can help minimize the input errors. System design should be in a way that the operators can easily interpret the system's behaviour when an error occurs. The SQM should also focus on creating user-friendly guidelines of the system usage. Critical actions should have multiple checks and confirmation before proceeding.

## 2. Regression testing

### Failures:

One of the main causes of the failures of Therac-25 is due to lack of proper regression testing of the system due to which the bugs in older versions reappeared and as the system is updated to next version new bugs appeared. For example, an older bug “Cursor up” function was believed to have been fixed but later played a major role in the accidents. Therac-25 used the software of Therac-20 without proper testing of the new system. The security checks in the hardware of the Therac-20 system are removed in the Therac-25 by relying completely on the software.

### Best SQM Response:

Whenever a software is reused from the previous versions of the system it must be thoroughly checked for errors and issues that might occur in the environment of new system. The SQM should make it compulsory that any system should undergo through a rigorous regression testing keeping in the mind of possible and unplanned issues in the new system. The SQM should make sure both functional, non-functional and other critical safety issues are properly tested before the new system is released for use. Newer test cases must be written in order to make sure that the system works as expected in the new environment. Frequent and dynamic testing must be implemented so that the bugs could be easily solved at an early stage.

## 3. Confidence due to previous success

### Failures:

The major drawbacks of the system are over confidence due to the success of the previous versions of the systems Therac-20 and Therac-6. Due to this, AECL (Atomic Energy of Canada Limited) stated that the machine could not malfunction in a way that would harm the patients. One of the major things which the AECL did due to the overconfidence is they removed the safeguards which were present in the hardware of Therac-20 and thought that the software of Therac-25 could handle them, eventually which failed to handle the errors. They also assured that safety improvements are made to the system after the early incidents by giving a false hope to the users. Because of these flaws, the machine malfunctioned and led to large doses of radiation to the patients.

### Best SQM Response:

Whenever a software is updated from one version to another it should be tested against all the odds in new environment. Even though the older version of the software is reliable when new changes are made to the system new issues might occur according to the new environment. Sometimes when a system is being updated to a newer version some minor issues might occur which should not be overseen because those minor issues in older system could show a huge impact in the newer system causing major outages. Most importantly the

SQM should ensure that the new system's safety shouldn't rely on previous version's success. Always the software should be tested with all possible test cases (both new and old).

## 4. Defensive design

### Failures:

The Therac-25 system depended completely on software for safety measures in contrast to the older system which had interlocks in its hardware. For example, The Therac-25 allowed the beam to be automatically activate without proper verification whether all the components of the system are present in correct position. The machine used to fail often without prior notification and clarity on the issue. Due to lack of defensive design, system ignored many issues which arose during the operation of system. This led to serious injuries and death of several patients.

### Best SQM Response:

In order to build a reliable and safety critical software systems, defensive design principles play a key role. Which would include adding multiple layers of safety checks and interlocks in both hardware and software. A system should be built in a way that even in the case of single point of failure it shouldn't cause a huge impact on the other components of the system. Implementing defensive design principles during the construction of the system would help us to identify issues early which can help us completely resolve the issue or in case of serious failure it can help us reduce the impact on the system.

## 5. Automated assessment of user awareness

### Failures:

In case of Therac-25, the operators are not fully aware of the potential risks and malfunctions of the system. Operators doesn't have a clear sense of the issue or system's state during a malfunction. Whenever an issue occurred the system allowed operators to override the issue and proceed easily without any restrictions. So, the issues are repeated again and again with no interruption from the system side. The system did not have any track of the issues and the warning messages shown to the operator during a critical issue can be easily bypassed.

### Best SQM Response:

The best SQM approach would be making an automated system incorporated in the software which can notify the operator in case of a potential risk. Also, an acknowledgement message showing the current state of the system or whenever the state of the system changes. In case of a critical risk, implementing an in-built system which checks automatically and prevents from moving forward to the next steps can help reduce the risk. Operator training modules

must be developed so that operators know how to act when an error message is displayed on the screen or during a possible potential critical risk.

## Conclusion:

This case study, Therac-25 explains us the importance of critical safety measures to be followed during the design of a software system. During the design of a software system each and every component of the system is critical which if compromised can lead to serious outages as seen in the case of Therac-25. A strong SQM approach would have addressed all these issues.