Recitation 1: We Did Nothing Wrong

The big takeaways from this paper aren't the technical details about how these systems work, but rather the understanding that many, many things went wrong; that adhering to some larger design principles could've helped; and that there are serious consequences to our design choices.

- **What went wrong here?** Many things. Some examples:
  - Multidata manual was unclear
  - Person read the manual incorrectly, or assumed too much from the manual
  - The code to enter dimensions did not throw an error
  - The code to calculate the radiation did not throw an error
  - The error itself was very obscure
  - Multidata did not react quickly to reports of over-radiation
  - Multidata had no quality-assurance program
  - The machine lacked regular maintenance (Multidata didn't purchase a maintenance contract)
  - Etc.

- **Would fixing any one of these things have solved the problem?** You can debate this, but note that we have a huge collection of problems, not just a single software bug.

- **Is modularity relevant? Where would modularity help?**
  - Software: It's hard to tell about the Multidata machine, but the Therac-25's design — also mentioned in the paper — was not modular and that partly resulted in the bug (we know this because there is a detailed report about the Therac-25 machine).
  - A modular system would've been easier to test

- **Do you agree with the following?:** "Many specifications and designs aren't thought out well enough. Programmers, no matter how good, make logical mistakes. In addition, testing procedures often aren't rigorous enough, he says. And today, with so many software programs interacting with other software programs, there's no way to predict what will happen when two pieces of code come in contact with each other for the first time."