History's Worst Software Bugs:

Summary and Considerations

The article thesis is that software defect, most notably known as bugs, are still present, and seems that while we are advancing in technology, they don’t seem to disappear. To support is thesis, the author exposes the case of the Prius car from the Toyota car company, and with software spreading out and touching different industries where it’s use is becoming essential, if not, a relevant component in the development, design, and functionality of the final product, it is a problem. The article continues by illustrating and describing the aftermath, with the subsequent damages of the most relevant, present and historical cases, where software defects or bugs had a major role for the catastrophic consequences that has brought. On all the cases that the article illustrates, the ones who caught my attention and made me aware of what a bug can seriously generate are: [Therac-25](http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_1.html) in 1980 and the **National Cancer Institute in Panama City case. In both of these cases the consequence of the defect** led to the loss of life. However, from all the cases illustrated in the article, the one that led to a loss of life are only two, and with a 20 year life span from one another. The gap from these two, in my opinion, relevant cases, suggests to me that a lot was done not to repeat the same mistakes, actually in the Therac-25 it was found that the operating system was developed by a programmer with no formal training. So, I suspect that a lot of legislation has been passed, or at least a lot of requirements have to be meet in order not to repeat those mistakes. However, while in all the cases illustrated there was in some way an economic lost, and in the Intel case, with the defect of the Pentium chip, a loss of 475 million dollars, the Therac-25 and the **National Cancer Institute in Panama City case** are the one that really a computer science student like me should be aware, and the responsibility’s, implicitly or explicitly, that code or test performed by me will and could have.