Peter Kirkpatrick

September 7, 2009

CS 230 Lab

Section A

The story that I decided to write about is the intended malfunction of some software that in 1982 sparked the largest non-nuclear explosion in the planet’s history. During this time the Soviet’s were in the process of obtaining sensitive United States technology. Somehow the Central Intelligence Agency became knowledgeable of the situation and decided to help with the process and get back at the Soviets. Allegedly, the CIA had some operatives impregnate some Canadian computer systems with bugs. These computer systems were in charge of controlling the trans-Siberian gas pipeline. The intent of the bug was to make the system backfire on the Soviets. The tainted equipment was intended to pass initial inspections by the Soviets and then fail miserably during execution. It was only meant to interfere with the project, but both parties never thought it would end like it did.

The result was equivalent to that of three kilotons of TNT exploding in a single area. There were no casualties reported luckily, but during this ordeal the United States was in direct defiance of the first and ninth rules of the Ten Commandments of Computer Ethics, not to mention the countless rules ignored by the Soviets.

The first rule of the Ten Commandments of Computer Ethics clearly states that one should never use a computer to harm other people. Even though no casualties were reported during this incident, it doesn’t mean that none were accrued.

The ninth rule states that one should always think about the social consequences of the program you are writing or the system you are designing. Clearly the United States was not thinking about the consequences of their actions and even though the explosion didn’t harm anybody the countless tons of pollution and waste that now lay strung around the world because of the explosion could have vast repercussions on the world later on.

Peter Kirkpatrick

September 7, 2009

CS 230 Lab

Section B

If I were the Senior Software Engineer of a particular project that was going to revolutionize the way nurses keep track of their hospitalized patients and my decision alone could save or endanger millions of lives, knowing that my product was not up to par in a field that demands excellence would force me to delay the launch of the product; whether it costs investors millions of dollars or not. The ninth rule of the Ten Commandments of Computer Ethics plainly states to think about the social consequences of the program you are writing or the system you are designing. Knowingly putting out a product that could endanger so many innocent lives would be both unethical and immoral.

Even though the investors and the company I would be working for would benefit greatly due to the earlier release of the product with regards to the competition, the amount of lives that would be needlessly be placed in harm’s way is simply unnecessary.

As the Senior Software Engineer, it is my job to guarantee that our product is of the utmost quality, and if that is not the case then it is once again my job to either motivate the engineers in my command or delay the launch of said product. There is no substitute for quality in the software engineering field, and when innocent peoples’ lives are at stake then quality is the most important thing.

I have a strong obligation to the people that this product will be interacting with on a daily basis. I cannot allow something I have created that was knowingly unsound be released into the active duty. Not only would I be held accountable but just the sheer burden of knowing that I have harmed so many people with a product that I could have denied would hurt me just as badly. Our role on this planet is to prolong life and enjoy it to the fullest extent of the law. What kind of person could allow tainted software into a workplace knowing that casualties are almost certain and that the repercussions of such a cataclysmic event could harm so many people.

Yes, getting the product on the market before the competition is ideal. Moreover, if this deadline is not met then the company could go under. This is all true, but what happens if the product did hit the market and as predicted failed miserably. What would happen then? Then the software is used in conjunction with the hardware killing millions and the company is later sued for everything. What happens to me? I am then sentenced to prison because I didn’t do my job to the best of my ability. In my mind, a company going under is worth saving the lives of hundreds of people.