# Disaster #1 – Therac-25

## Describe the background behind the incident.

The Therac-25 was a new radiation therapy machine which was produced after the Therac-6 and built off of the Therac-20. Radiation therapy through this type of machine is a commonly used technique for people who have cancer. It works by killing off the rogue cancer cells by damaging their DNA. Generally uses X-rays, gamma rays, and charged particles (such as beta particles) for treatment. (Garfinkel, 2005) This treatment technique damages normal cells as well as cancer cells so the setup and use of a radiation therapy machine must be carefully as to minimize side effects.

## Describe the problem.

Given certain circumstances, the Therac-25 medical radiation therapy machine is prone a malfunction where the machine changes the radiation target somewhere slightly off from the desired location and then delivers high-power doses of radiation. Since healthy uninfected cells are also destroyed using this procedure (National Cancer Institute, 2010), it ended up being fatal to many patients at several different medical facilities. Also, as a design consideration when building this solution, the company decided that the software to run on this machine was so stable that it did not need any kind of emergency kill switch.

## Describe the cause or causes behind the problem

The cause of this product malfunctioning was later found out to be a bug in the operating system where, if the operator types in the prescription too fast, the machine could not accept it properly and would configure for high power mode at an incorrect location. This type of error is commonly known as the “race condition” (FreeBSD Documentation Project), simply meaning a flaw which is determinant on two signals racing each other to influence the output first. This issue is predominant in software doing asynchronous I/O and using shared memory on multiple threads or processes. This issue is also prevalent in hardware under certain conditions. (Netzer & Miller, 1992)

## What would have prevented the problem?

This issue could have been prevented if the engineers had tested operating system of the machine more closely and had looked further into the control of asynchronous threads. Also, this issue was not helped by the design choice of not including an emergency stop button to turn off the machine quickly if a malfunction occurs. This was left out however because the company decided that their software was going to be so stable that it wouldn’t need one.

## Did this occur because a requirement was changed inappropriately? Justify your answer.

## Did this occur because the technology was rushed? Justify your answer.

## Did this occur because the problem should have been caught by normal testing but was not? Justify your answer.

Since this error sounds like it would have been fairly difficult to reproduce unless a developer tests their code on a running machine themselves, I’m not entirely sure the problem would have been caught in normal testing circumstances. Despite this however, the original software engineers should have put more time into testing the operating system by itself before claiming that it was, software wise, bulletproof.

## Did people pre-warn against the possibility of such a problem occurring? If so, what role did they have and why were their warnings ineffective. Justify your answer.

## What role did software or electronic systems play in causing the disaster? Justify your answer.

In this disaster, the software and electronic systems teamed up to make something which is extremely destructive. This is due to the reliance on the inconsistency that is software and software development paired with the power that is “does something in real life” hardware. As an example, if the software was Carlo and the hardware was a gun, Carlo would have no trouble taking the gun and shooting someone in the head.

## Provide your references for the above information. Use an accepted bibliographic format (e.g. APA format).

FreeBSD Documentation Project. (n.d.). *3.7 Race Conditions*. Retrieved November 22, 2012, from FreeBSD Handbook.

Garfinkel, S. (2005, November 8). *History's Worst Software Bugs*. Retrieved 11 22, 2012, from WIRED: http://www.wired.com/software/coolapps/news/2005/11/69355

Netzer, R. H., & Miller, B. P. (1992). What are race conditions?: Some issues and formalizations. *ACM Letters on Programming Languages and Systems (LOPLAS)*, 74-88.

<http://www.dcs.gla.ac.uk/~muffy/papers/HIS1.pdf>

http://www.ingentaconnect.com/content/jcaho/jcjqs/2004/00000030/00000012/art00007

# Disaster #2 – Mars Climate Orbiter

## Describe the background behind the incident.

## Describe the problem.

After a misunderstanding between a control machine and its’ operator, the mars climate orbiter was accidentally put into orbit around Mars. The spacecraft was then deemed as to have crashed into the planet’s surface sometime around December later that year.

## Describe the cause or causes behind the problem

The cause to this incident was later found to be a miscommunication between the control software and the user. These misunderstandings on units lead to an unreasonable overcorrection in control

For the Mars Climate Orbiter project, NASA outsourced its development to a third company, Lockheed-Martin.

## What would have prevented the problem?

## Did this occur because a requirement was changed inappropriately? Justify your answer.

## Did this occur because the technology was rushed? Justify your answer.

## Did this occur because the problem should have been caught by normal testing but was not? Justify your answer.

## Did people pre-warn against the possibility of such a problem occurring? If so, what role did they have and why were their warnings ineffective. Justify your answer.

## What role did software or electronic systems play in causing the disaster? Justify your answer.

## Provide your references for the above information. Use an accepted bibliographic format (e.g. APA format).

FreeBSD Documentation Project. (n.d.). *3.7 Race Conditions*. Retrieved November 22, 2012, from FreeBSD Handbook.

Garfinkel, S. (2005, November 8). *History's Worst Software Bugs*. Retrieved 11 22, 2012, from WIRED: http://www.wired.com/software/coolapps/news/2005/11/69355

Lloyd, R. (1999, September 30). CNN. *Metric Mishap Caused Loss of Nasa Orbiter*.

National Cancer Institute. (2010, June 30). *Radiation Therapy for Cancer*. Retrieved 11 22, 2012, from National Cancer Institute: http://www.cancer.gov/cancertopics/factsheet/Therapy/radiation

Netzer, R. H., & Miller, B. P. (1992). What are race conditions?: Some issues and formalizations. *ACM Letters on Programming Languages and Systems (LOPLAS)*, 74-88.

ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO\_report.pdf

# Disaster #3 – NORAD

## Describe the background behind the incident.

## Describe the problem.

## Describe the cause or causes behind the problem

## What would have prevented the problem?

## Did this occur because a requirement was changed inappropriately? Justify your answer.

## Did this occur because the technology was rushed? Justify your answer.

## Did this occur because the problem should have been caught by normal testing but was not? Justify your answer.

## Did people pre-warn against the possibility of such a problem occurring? If so, what role did they have and why were their warnings ineffective. Justify your answer.

## What role did software or electronic systems play in causing the disaster? Justify your answer.

## Provide your references for the above information. Use an accepted bibliographic format (e.g. APA format).

# Disaster #4 –

## Describe the background behind the incident.

## Describe the problem.

## Describe the cause or causes behind the problem

## What would have prevented the problem?

## Did this occur because a requirement was changed inappropriately? Justify your answer.

## Did this occur because the technology was rushed? Justify your answer.

## Did this occur because the problem should have been caught by normal testing but was not? Justify your answer.

## Did people pre-warn against the possibility of such a problem occurring? If so, what role did they have and why were their warnings ineffective. Justify your answer.

## What role did software or electronic systems play in causing the disaster? Justify your answer.

## Provide your references for the above information. Use an accepted bibliographic format (e.g. APA format).