Software Disasters: What Have we Learned

Thomas Zangari

The article focuses on three cases: The Mars Polar Lander, The Patriot missile, and the Therac-25 in other to expose that software defects come in many forms quick can led to fatalities, los of investment with a wide range of consequences.

In the Mars Polar Lander case, a single line of code provoked the lost of the probe while it was approaching the Mars surface. It was established that it generated a miss calculation, and that because of it, the landing rockets did not followed threw until the legs of the probe had touched the surface causing it to smash in the ground. Further investigation revealed that the entire program was lacking experienced managers, a test and verification program and adequate safety margins and that proper testing was not done, do to budget constraints and because of a larger problem, that is that system are becoming so complex that is becoming impossible to manage.

In the patriot missile case: 28 people died and 97 were injured because a Scud missile was not intercepted by the automatic reviling system of the patriot ECS system that was responsible to launch the patriot missile to intercept the Scud. Investigation revealed that the cause was do to the fact that the patriot missile system was reconverted from his original role and that its software design was not build to perform under those extensive operation hours causing a miscalculation in the internal timing. The conclusion of the investigation was that robust testing is needed for safety-critical-software and that the product must be tested in the environment of operation.

In the Therac-25 case: several people died because of exposure to high levels of radiation while ongoing cancer treatment. Investigation revealed that the cause was to be traced to bug in the software and future research revealed that it was from a programmer with no formal training.

From all three cases, one common factor that is implicitly exposed, is that there are strong economic constrains in each case that influenced decisions and procedures. Unfortunately they led to los of live. I suppose that the cost benefit analysis that was carried out was strongly concentrated on the economic aspect leading to oversee what type of consequence could eve triggered. However, from the software aspect, I understood that complex systems and continuing implementation without rigorous testing can lead to oversee crucial aspect for the performance of the product, and, like in the other two cases adapting software has to be followed by rigorous testing .