

Analyzing Software Failure on the NASA Mars Climate Orbiter

16th December 2017 on Space, Software Development, Programming, Software Architecture by Christopher Demicoli

The [Mars Climate Orbiter](#) was a robotic space probe manufactured by [Lockheed Martin](#) and launched by [NASA's JPL](#) on December 11, 1998. The purpose of this probe was to study the Mars climate, atmosphere, and surface changes and to act as the communications relay in the Mars Surveyor '98 program for [Mars Polar Lander](#). **The total cost of this mission was \$327.6 million.**

Nine and a half months after launch, in September 1999, the Mars Climate Orbiter was programmed to fire its main engine to achieve an elliptical orbit around Mars. On September 23, 1999 at 09:00:46 UTC, the space probe began this planned orbital manoeuvre. After about 4 minutes, the spacecraft passed behind Mars, 49 seconds earlier than expected. Due to this, the Mars Climate Orbiter went out of radio contact and communication with the probe was never re-established.

On November 10, 1999, the Mars Climate Orbiter Mishap Investigation Board released [a report](#) describing the issues that lead to the loss of the spacecraft.

Because of software error, the spacecraft encountered Mars at a lower than anticipated altitude and disintegrated due to atmospheric stresses.

According to this report, the primary cause of this failure was that **one piece of ground software supplied by Lockheed Martin produced**

results in a United States customary unit, while a second system, supplied by NASA, expected those results to be in SI units.

In this example, we can clearly see the huge impact of developing software without good specifications, requirement and guidelines can have. In this case, NASA already had a Software Interface Specification (SIS) that specifically defines both the format and units of the software used by ground-based computers. For the Mars Climate Orbiter, the contractor (Lockheed Marting) did not follow the SIS, so the calculations were off by a factor of 4.45, which is the conversion ratio from force in pounds to Newtons. An erroneous trajectory was computed using this incorrect data.

What a blow to space exploration it was to have spent \$327.6 million on this mission and have the probe crash into the Martian atmosphere because two teams of software engineers, who both wrote great code, did not communicate together and did not follow specifications.

Rocket science is hard, but communication is harder. Tests are fundamental.

blog.cdemi.io