Software errors cost the U.S. economy [$60 billion](http://www.nist.gov/public_affairs/releases/n02-10.htm) annually in rework, lost productivity and actual damages.  We all know software bugs can be annoying, but faulty software can also be expensive, embarrassing, destructive and deadly.  Following are 20 famous software “disasters” in chronological order:  
============================

### 1.  Mariner Bugs Out (1962)

**Disaster:**  The Mariner 1 rocket with a space probe headed for Venus diverted from its intended flight path shortly after launch.  Mission Control destroyed the rocket 293 seconds after liftoff.



**Cost:**  $18.5 million

**Cause:**  A programmer incorrectly transcribed a handwritten formula into computer code, missing a single superscript bar.  Without the smoothing function indicated by the bar, the software treated normal variations of velocity as if they were serious, causing faulty corrections that sent the rocket off course

“The booster had performed satisfactorily until an unscheduled yaw-lift (northeast) maneuver was detected by the range safety officer. Faulty application of the guidance commands made steering impossible and were directing the spacecraft towards a crash, possibly in the North Atlantic [shipping lanes](http://en.wikipedia.org/wiki/Shipping_lanes) or in an inhabited area. The destruct command was sent 6 seconds before separation, after which the launch vehicle could not have been destroyed. The radio transponder continued to transmit signals for 64 seconds after the destruct command had been sent.[[2]](http://en.wikipedia.org/wiki/Mariner_1#cite_note-1)”

The error had occurred when a symbol was being transcribed by hand in the specification for the guidance program. The writer missed the superscript bar (or [overline](http://en.wikipedia.org/wiki/Overline)) in



by which was meant "the *n*th smoothed value of the time derivative of a radius R". Without the smoothing function indicated by the bar, the program treated normal minor variations of velocity as if they were serious, causing spurious corrections that sent the rocket off course.[[14]](http://en.wikipedia.org/wiki/Mariner_1#cite_note-13) It was then destroyed by the [Range Safety Officer](http://en.wikipedia.org/wiki/Range_Safety_Officer).[[15]](http://en.wikipedia.org/wiki/Mariner_1#cite_note-14)

### 2.  Hartford Coliseum Collapse (1978)



**Cost:**  $70 million, plus another $20 million damage to the local economy

**Disaster:**  Just hours after thousands of fans had left the Hartford Coliseum, the steel-latticed roof collapsed under the weight of wet snow.

**Cause:**  The programmer of the CAD software used to design the coliseum incorrectly assumed the steel roof supports would only face pure compression.  But when one of the supports unexpectedly buckled from the snow, it set off a chain reaction that brought down the other roof sections like dominoes.

* The top layer's exterior compression members on the east and the west faces were overloaded by 852%.
* The top layer's exterior compression members on the north and the south faces were overloaded by 213%.
* The top layer's interior compression members in the east-west direction were overloaded by 72%.

### 3.  CIA Gives the Soviets Gas (1982)

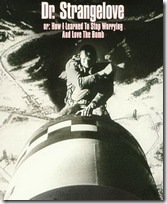


**Cost:**  Millions of dollars, significant damage to Soviet economy

**Disaster:**  Control software went haywire and produced intense pressure in the Trans-Siberian gas pipeline, resulting in the largest man-made non-nuclear explosion in Earth’s history.

**Cause:**  CIA operatives allegedly planted a bug in a Canadian computer system purchased by the Soviets to control their gas pipelines.  The purchase was part of a strategic Soviet plan to steal or covertly obtain sensitive U.S. technology.  When the CIA discovered the purchase, they sabotaged the software so that it would pass Soviet inspection but fail in operation.

### 4.  World War III… Almost (1983)



**Cost:**  Nearly all of humanity

**Disaster:**  The Soviet early warning system falsely indicated the United States had launched five ballistic missiles.  Fortunately the Soviet duty officer had a “funny feeling in my gut” and reasoned if the U.S. was really attacking they would launch more than five missiles, so he reported the apparent attack as a false alarm.

**Cause:**  A bug in the Soviet software failed to filter out false missile detections caused by sunlight reflecting off cloud-tops.

Petrov was initially praised for his cool head but later came under criticism and was, for a while, made the scapegoat for the false alarm. Further investigation, however, found that the satellite in question had picked up the sun's reflection off the cloud tops and somehow interpreted that as a missile launch.

### 5.  Medical Machine Kills (1985)

**Cost:**  Three people dead, three people critically injured

**Disaster:**  Canada’s Therac-25 radiation therapy machine malfunctioned and delivered lethal radiation doses to patients.

**Cause:**  Because of a subtle bug called a [race condition](http://en.wikipedia.org/wiki/Race_condition), a technician could accidentally configure Therac-25 so the electron beam would fire in high-power mode without the proper patient shielding.

### 7.  AT&T Lines Go Dead (1990)

**Cost:**  75 million phone calls missed, 200 thousand airline reservations lost

**Disaster:**  A single switch at one of AT&T’s 114 switching centers suffered a minor mechanical problem and shut down the center.  When the center came back up, it sent a message to other switching centers, which in turn caused them to shut down and brought down the entire AT&T network for 9 hours.

**Cause:**  A single line of buggy code in a complex software upgrade implemented to speed up calling caused a ripple effect that shut down the network.

### 10.  Ariane Rocket Goes Boom (1996)



**Cost:**  $500 million

**Disaster:**  Ariane 5, Europe’s newest unmanned rocket, was intentionally destroyed seconds after launch on its maiden flight.  Also destroyed was its cargo of four scientific satellites to study how the Earth’s magnetic field interacts with solar winds.

**Cause:**  Shutdown occurred when the guidance computer tried to convert the sideways rocket velocity from 64-bits to a 16-bit format.  The number was too big, and an overflow error resulted.  When the guidance system shut down, control passed to an identical redundant unit, which also failed because it was running the same algorithm.

### 12.  Mars Climate Crasher (1998)

### 12.  Mars Climate Crasher (1998)



**Cost:**  $125 million

**Disaster:**  After a 286-day journey from Earth, the Mars Climate Orbiter fired its engines to push into orbit around Mars.  The engines fired, but the spacecraft fell too far into the planet’s atmosphere, likely causing it to crash on Mars.

**Cause:**  The  software that controlled the Orbiter thrusters used imperial units (pounds of force), rather than metric units (Newtons) as specified by NASA.

### 15.  Y2K (1999)

**Cost:**  $500 billion

**Disaster:**  One man’s disaster is another man’s fortune, as demonstrated by the infamous Y2K bug.  Businesses spent billions on programmers to fix a glitch in legacy software.  While no significant computer failures occurred, preparation for the Y2K bug had a significant cost and time impact on all industries that use computer technology.

**Cause:**  To save computer storage space, legacy software often stored the year for dates as two digit numbers, such as “99″ for 1999.  The software also interpreted “00″ to mean 1900 rather than 2000, so when the year 2000 came along, bugs would result.

**18.  Cancer Treatment to Die For (2000)**

**Cost:**  Eight people dead, 20 critically injured

**Disaster:**  Radiation therapy software by Multidata Systems International miscalculated the proper dosage, exposing patients to harmful and in some cases fatal levels of radiation.  The physicians, who were legally required to double-check the software’s calculations, were indicted for murder.

**Cause:**  The software calculated radiation dosage based on the order in which data was entered, sometimes delivering a double dose of radiation.

Reference

<http://www.devtopics.com/20-famous-software-disasters-part-4/>