**United Airlines and Hawaiian Airlines service outages**

The FAA maintains a database called Notice to Air Missions (NOTAM). On January 11, NOTAM went down, [causing a nationwide “ground stop” that halted all takeoffs](https://www.computerworld.com/article/3685211/us-flights-resume-after-system-failure-causes-faa-to-halt-air-travel.html), though planes in the air were allowed to continue to their destinations. An honest mistake that cost the country millions.

**The NYSE’s brittle backup process**

The New York Stock Exchange faced a similar crisis in January as well. On January 24, when a Chicago employee [failed to turn the backup server off at the appropriate time](https://www.bloomberg.com/news/articles/2023-01-25/nyse-mayhem-traced-to-a-staffer-who-left-a-backup-system-running). As a result, when trading began in New York at 9:30 a.m., the NYSE computers thought they were continuing the previous day’s trading session and ignored the day’s opening auctions, which are supposed to set initial prices for many stocks. The outcome was a series of violent market swings and numerous transactions at incorrect prices that had to be cancelled at great expense.

**When a crash means a real crash**

in 2010, a “catastrophic” engine failure occurred when a pilot tried a so-called “hot start” — powering down and then restarting the engine mid-mission. This mechanical problem was [fixed in software](https://www.bitdefender.com/blog/hotforsecurity/army-helicopter-crash-blamed-on-skipped-software-patch/), with Australian Defence Force rolling out a software patch designed to prevent the helicopter from being hot started. Unfortunately, the first rule of software patches is that they work only if you actually roll them out, and despite the fact that this patch has been available for the better part of a decade, it wasn’t installed on all of Australia’s Taipans, resulting in a hot start that led to a [helicopter crash during a training mission this past April](https://www.theregister.com/2023/04/18/helicopter_crash_missing_software_patch/)

**Artificial intelligence, real failure**

AI failures also hit the tech journalism world, with CNET being forced to [retract more than 35 stories](https://www.cio.com/article/1253464/Responsible%20AI%20Machine%20Partner,%20or%20RAMP,) that were written with the help of a tool called the Responsible AI Machine Partner, or RAMP. The less-than-responsible results not only left the company with egg on its face but [drove a backlash from its workers as well](https://www.wired.com/story/cnet-published-ai-generated-stories-then-its-staff-pushed-back/).

**Cascading phone failures down under**

Australia was the site of another high-profile IT failure in November, when Optus, the country’s second-largest telecom provider, went down for 12 hours, leaving half of Australians without phone or Internet connectivity. The fault could be ultimately traced to [routing changes sent by Singtel](https://www.theregister.com/2023/11/21/optus_ceo_quits/), the Singapore-based company that owns Optus; this information was apparently such a large wave of data that it overwhelmed Optus’s routers, which then had to be physically restarted — something that took quite a long time, given Australia’s size.

In the industry of Software Engineering, the failures of software are inevitable, but as a software developer, I assume that to measure and predict in any possible situations which are really important. According to the Ariane 5 Flight 501 incident, because of negligence, when the Ariane 4 was inherited by Ariane 5, the defection happened since the data size of Ariane 4’s software cannot meet to the data size of Ariane 5. My perspective of the reduction software failures issue is that, beside of measurement carefully, we also need a source of humans high-quality in software fields, who is qualified and have high qualification. Reducing maximizes the negative effects caused by software failures will help people to achieve many achievements in the field of technology.