Computers and Society

Software Reliability

Notice: This set of slides is based on the notes by Professor Guattery of Bucknell and by the textbook author Michael Quinn

Computer Errors

Considered in the broadest context, computer errors occur when the use of a computer leads to performance that is unexpected or is outside the specified norm.

Such errors include design errors, implementation errors, and cases where incorrect data is entered into the system. Incorrect conclusions based on such data has led to incidents such as disenfranchised voters and false arrests.

Computer Errors (2)

User misuse of computers cannot be blamed on the system designers or programmers.

However, it is important for system designers and developers to understand how their systems will be used. They must consider how users will interact with the system and design it to minimize user errors.

They must also be aware of how their systems could be misused.

Software Errors

We will focus more on problems caused because of errors programmers make in software.

Many errors are relatively small and are obvious when they occur. While they are often quickly fixed, effects can spread. For example, a billing mistake may lead to a bill that a customer can't pay. Failure to pay is detected automatically by another computer system, that charges the customer additional fees.

Software Errors (2)

Some examples of software failures:

• In 2001 a bug in a US telephone company's billing software billed cell phone customers $600 per minute of cell phone use.
• In 1996 a bug in a US Postal Service system caused mail addressed to the Patent and Trademark Office to be returned to sender. Two weeks of mail (50,000 pieces) was affected.

Software Errors (3)

• The day the City of London started operating a new computerized ambulance dispatch system in 1996, people making emergency calls were put on hold for 2 hours, and ambulances took 3 hours to arrive. As many as 20 people died.
• In December 2004, Comair, a US airline, cancelled all flights on Christmas day. About 30,000 passengers were affected and 1100 flights were canceled. The computer system failed because it had been overloaded because of cancelations due to bad weather the two preceding days.

Some More Recent News

- December 7, 2013, a computer glitch caused airline disruption in England
  - 100,000 passengers affected, 1,300 flights disrupted, 228 canceled at Heathrow
- April 16, 2013, software problem caused American Airline cancellations and delays
  - 1,000 cancellations, 1,100 delays

http://www.dailymail.co.uk/news/article-2519774/Plane-delays-Flights-delayed-hours-airports-Britain-technical-problems.html

http://www.ainonline.com/aviation-news/ain-air-transport-perspective/2013-04-22/american-airlines-apologizes-disruptions-blames-software

Wired News's 10 Worst Bugs

In chronological order (most of the following text was taken directly from the *Wired* article):

July 28, 1962: A bug in the Mariner I space probe's flight software caused the rocket to go off course and be destroyed. The bug was a result of incorrect transcription of a mathematical formula into computer code.

Wired News's 10 Worst Bugs (2)

1982: A Soviet Union gas pipeline in Siberia explodes. Some reports claim that this was the largest non-nuclear explosion in world history.

Reports have claimed that the explosion was the result of a CIA bug planted in the system in retaliation for the Soviets buying the system from Canada to get around US export regulations.

Wired News's 10 Worst Bugs (3)

1985-1987: Therac-25

Software bugs and poor design lead to problems that kill three patients.

The cause of the problem was that the software didn't deal with *race condition* properly.

Wired News's 10 Worst Bugs (4)

1988: Unix system bugs allow the Morris Worm to infect between 2,000 and 6,000 computers in less than a day by taking advantage of a buffer overflow. The specific code is a function in the standard input/output library called *gets()* designed to get a line of text over the network.

Programmers respond by attempting to stamp out the *gets()* function in working code, but it is not removed from the C language's standard input/output library, where it remains.

Wired News's 10 Worst Bugs (5)

1988-1996: The authors of the Kerberos security system neglect to properly "seed" the program's random number generator with a truly random seed. As a result, for eight years it is possible to trivially break into any computer that relies on Kerberos for authentication.

It is unknown if this bug was ever actually exploited.

Wired News's 10 Worst Bugs (6)

January 15, 1990: A bug in a new release of the software that controls AT&T's #4ESS telephone switches caused them to crash. This happened because a bug produced a cascading failure in which 114 switches were crashing and rebooting every six seconds. This left roughly 60 thousand people without long distance service for nine hours.

Wired News 10 Worst Bugs (6.2)

These three machines sent out recovery messages. These additional recovery messages added to the network load, overloading more machines. These machines then crashed in response to the recovery messages.

AT&T Switch Problem Explanation

The New York switch sent a message to all the other 4ESS switches it is linked with that it was not accepting additional traffic. AT&T manager Larry Seese referred to that message as a "congestion signal." After the switch successfully completed the reintialization, the New York switch went back in service and began processing calls.

That is when the fault in the new software reared its ugly head. Under the previous system, switch A would send out a message that it was working again, and switch B would double-check that switch A was back in service.

AT&T Problem Explanation (2)

With the new software, switch A begins processing calls and sends out call routing signals. The reappearance of traffic from switch A is supposed to tell switch B that A is working again. This is supposed to be faster.

"The first common channel signaling system 7 initial address message (caused by a call attempt) that switch B receives from switch A alerts B that A is back in service. Switch B then resets its internal logic to indicate that A is back in service," said Seese.

AT&T Problem Explanation (3)

The problem occurred when switch B got a second call-attempt message from A while it was in the process of resetting its internal logic. "[The message] confused the software. it tried to execute an instruction that didn't make any sense…" so switch B shut itself down to avoid spreading the problem, Seese explained.

Unfortunately, switch B then sent a message to other switches that it was out of service. Once switch B reset itself, it sent out call processing messages. That caused identical failures around the nation."It was a chain reaction. Any switch that was connected to B was put into the same condition."

Wired News's 10 Worst Bugs (7)

1993: A silicon error causes Intel's new Pentium chip to make mistakes when dividing floating-point in a specific range. Although the bug affects few users, it becomes a public relations nightmare.

With 3 million to 5 million defective chips in circulation, Intel offered to replace Pentium chips only for consumers who could prove they needed high accuracy.

Wired News 10 Worst Bugs (7.2)

Eventually the company had to agree to replace the chips for anyone who complained.

The bug ultimately cost Intel $475 million.

Wired News's 10 Worst Bugs (8)

1995/1996: -- The Ping of Death. A lack of sanity checks and error handling in the IP fragmentation reassembly code makes it possible to crash a wide variety of operating systems by sending a malformed "ping" packet from anywhere on the internet.

Windows computers, which display the "blue screen of death" when they receive these packets, are most affected, but the attack also affects many Macintosh and Unix systems.

Wired News's 10 Worst Bugs (9)

June 4, 1996: The European Ariane 5 rocket reused code from the earlier Ariane 4 rocket. The Ariane 4 includes code that converts a 64-bit floating-point number to a 16-bit signed integer.

On its first flight, the Ariane 5's faster engines cause the 64-bit numbers to be larger than in the Ariane 4, triggering an overflow condition that results in the flight computer crashing, and the crash of the rocket.

Wired News 10 Worst Bugs (9.2)

First the backup computer crashed, followed 0.05 seconds later by a crash of the primary computer.

As a result of these crashed computers, the rocket's primary processor overpowered the rocket's engines and caused the rocket to disintegrate 40 seconds after launch.

An uninsured $500 million satellite was lost.

Wired News 10 Worst Bugs (10)

November 2000: National Cancer Institute, Panama City. In a series of accidents, therapy planning software created by Multidata Systems International, a US firm, miscalculated the proper dosage of radiation for patients.

Multidata's software allows a radiation therapist to draw on a computer screen the placement of metal shields called "blocks" designed to protect healthy tissue from the radiation.

Wired News 10 Worst Bugs (10.2)

The software will only allow technicians to use four shielding blocks, and the Panamanian doctors wish to use five.

The doctors discover that they can trick the software by drawing all five blocks as a single large block with a hole in the middle. What they didn't realize is that the Multidata software gives different answers depending on how the hole is drawn.

Wired News 10 Worst Bugs (10.3)

Draw the hole in one direction and the correct dose was calculated, draw in another direction and the software recommended twice the necessary exposure.

At least eight patients died, while another 20 received overdoses likely to cause significant health problems. The physicians, who were legally required to double-check the computer's calculations by hand, are indicted for murder.

Patriot Missile Failure

During the 1991 Gulf War, the US military used an anti-missile defense system called the Patriot missile system. While the army initially claimed that the system destroyed 95% of Iraq's SCUD missiles fired at the US positions, later analysis showed this was closer to 9%.

One missile got through the US defense and killed 28 soldiers.

Patriot Missile Failure (2)

An analysis of the system showed that the Patriot system lost track of the SCUD missile because of a miscalculation of the expected place where the SCUD would be.

The tracking system used the system clock signal in its computation. The system clock was a floating point with too little precision, which resulted in truncation errors. Over time the truncation errors accumulated into a significant error.

Patriot Missile Failure (3)

The system was designed with the assumption that it would never be active for more than a few hours. It was tested under those conditions, and the truncation error never got too large.

During the Gulf War, the systems ran for days at a time. The system that failed had been in operation for more than 100 hours.

Mars Robot Mission Failures

In 1999, two missions to Mars by NASA, the US's space agency, ended in serious failures because of software issues.

In the first, there was a miscommunication between two organizations. One wrote the software for the $125 million Mars Climate Orbiter using metric units (newtons). The other, which calculated the thrust requirements, used English units (foot-pounds). As a result the wrong thrust level was used and the Orbiter crashed.

Mars Robot Mission Failures (2)

In the second, engineers suspect that the computer in the $165 million Mars Polar Lander got a bad signal from the landing gear and shut down the engines too soon. The Lander landed too fast and crashed.

Tony Spear, a NASA project manager, observed, "It is just as hard to do Mars missions now as it was in the mid-70's. I'm a big believer that software hasn't gone anywhere. Software is the number-one problem."

Analysis: E-Retailer Posts Wrong Price, Refuses to Deliver

- Amazon.com in Britain offered iPaq computer for £7 instead of £275 in March 13, 2003
- Orders flooded in
- Amazon.com shut down site, refused to deliver unless customers paid true price
- Was Amazon.com wrong to refuse to fill the orders?

Rule Utilitarian Analysis

- Imagine rule: A company must always honor the advertised price
- Consequences
  - More time spent proofreading advertisements
  - Companies would take out insurance policies
  - Higher costs → higher prices
  - All consumers would pay higher prices
  - Few customers would benefit from errors
- Conclusion
  - Rule has more harms than benefits
  - Amazon.com did the right thing

Kantian Analysis

- Buyers knew 97.5% markdown was an error
- They attempted to take advantage of Amazon.com's stockholders
- They were not acting in "good faith"
- Buyers did something wrong