



# CESE 3321, 2021 Fall

## M100 Simple Mistakes

# MOO

## *Simple Mistakes*



# Simple Mistakes

## *Aerospace*



### Mariner 1—1962 July 22

- Venus fly-by mission
- ~18.5M US\$ (1962)
  - ~160M US\$ (2020)



<https://nssdc.gsfc.nasa.gov/image/spacecraft/mariner02.gif>



## Mariner 1—1962 July 22

- On-board systems began over-correcting for slight variations in flight path.
- Destroyed 293 seconds into the mission.
- A *single character* had been omitted from the translation of hand-written information into program form.

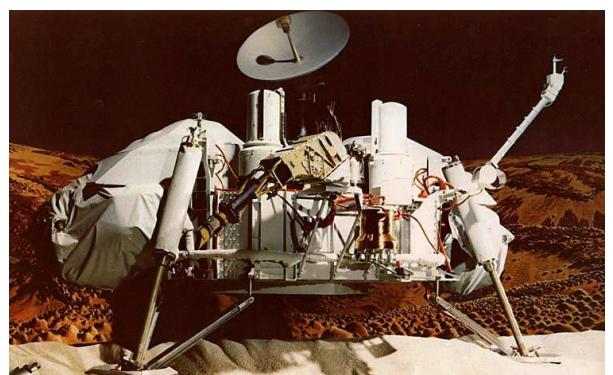


<https://nssdc.gsfc.nasa.gov/image/spacecraft/mariner02.gif>



## Viking 1 Lander—1982 November 19

- Mars landing mission
- Entire Viking project ~1B US\$ (1975)
  - ~4.8B US\$ (2020)

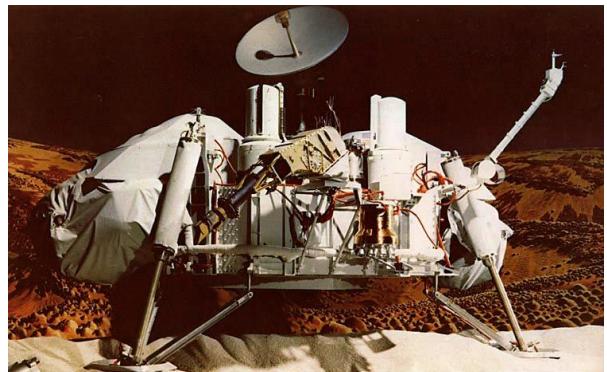


[https://nssdc.gsfc.nasa.gov/image/spacecraft/viking\\_lander\\_model.jpg](https://nssdc.gsfc.nasa.gov/image/spacecraft/viking_lander_model.jpg)



## Viking 1 Lander—1982 November 19

- Worked fine from 1972 July 20.
- A software update 1982 November 19 was written into the *wrong memory location* and broke the antenna aiming software.
- All contact was lost.

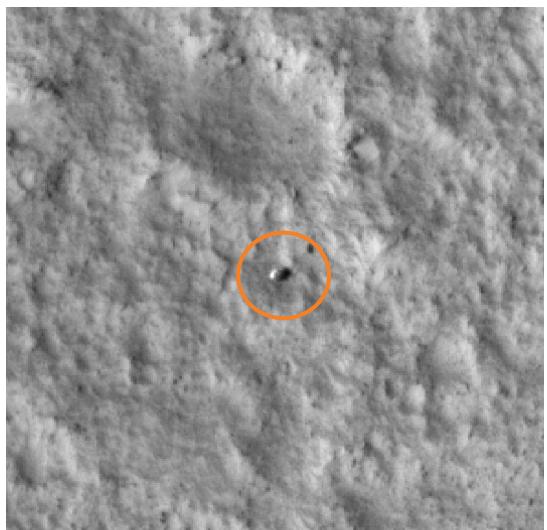


[https://nssdc.gsfc.nasa.gov/image/spacecraft/viking\\_lander\\_model.jpg](https://nssdc.gsfc.nasa.gov/image/spacecraft/viking_lander_model.jpg)



## Viking 1 Lander

- The Viking 1 lander is still there on Mars, of course. It might even still be trying to transmit, though *in the wrong direction*.
- The Mars Reconnaissance Orbiter took a picture of the Viking 1 lander in November 2006.



[https://www.nasa.gov/images/content/163925main\\_PSP\\_001521\\_2025\\_RED\\_VL-1\\_lander.tif](https://www.nasa.gov/images/content/163925main_PSP_001521_2025_RED_VL-1_lander.tif)



## Фобос 1 (Phobos 1)—1988 August 28

- Martian moon mission
- Entire Фобос project  
~300M US\$ (1988)
  - ~660M US\$ (2020)



[http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos\\_1.jpg](http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos_1.jpg)



## Фобос 1 (Phobos 1)—1988 August 28

- About 7 weeks into the mission, a ground operator omitted a *single hyphen character* from a command.
- The new command disabled the craft's attitude thrusters. Its solar array tilted *away* from the Sun. Eventually the batteries were depleted.
- All contact was lost.



[http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos\\_1.jpg](http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos_1.jpg)



## Фобос 1 (Phobos 1)—1988 August 28

- All commands were supposed to be proofed before sending, but the checking computer was not working and the operator forced the send.
- The code to turn off the thrusters was for testing purposes only and should have been deleted before launch.
- The software was in PROMs, though, and to save time they were not reprogrammed.

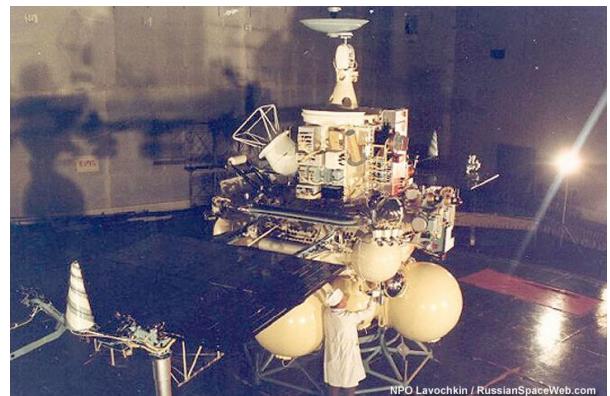


[http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos\\_1.jpg](http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/fobos_1.jpg)



## Фобос 2 (Phobos 2)—1989 March 27

- Companion craft to Фобос 1.
- By the time it reached Mars, *two of the three* onboard computers were malfunctioning.
- The working computer was helpless as the three computers “voted” on what to do and the failing computers outvoted it.
- All contact was lost.



[http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/prototype\\_assembly\\_1.jpg](http://www.russianspaceweb.com/images/spacecraft/planetary/mars/fobos/prototype_assembly_1.jpg)



## Cluster—1996 June 04

- Earth magnetosphere research mission.  
Four satellites that were to deploy in a tetrahedral formation.
- ~370M US\$ (1996)
  - ~610M US\$ (2020)

[https://commons.wikimedia.org/wiki/File:Ariane\\_501\\_Cluster.svg](https://commons.wikimedia.org/wiki/File:Ariane_501_Cluster.svg)



## Cluster—1996 June 04

- 36.7 seconds into the launch the conversion of a 64-bit FP number to a 16-bit integer caused an overflow.
- Garbage got sent from the Inertial Reference System to the On-Board Computer.
- The OBC mispositioned the engine nozzles which caused the disintegration of the craft.

[https://commons.wikimedia.org/wiki/File:Ariane\\_501\\_Cluster.svg](https://commons.wikimedia.org/wiki/File:Ariane_501_Cluster.svg)



## Milstar—1999 April 30

- Military communications
- ~800M US\$ (1999) for each satellite (doesn't include launch vehicle, etc.)
  - ~1.25B US\$ (2020)

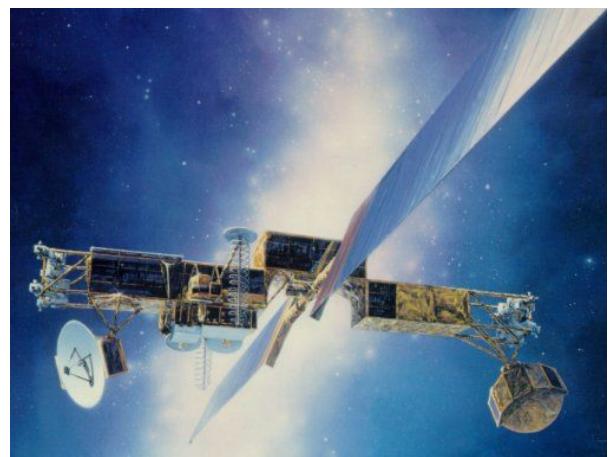


<https://media.defense.gov/2003/Mar/28/2000031174/-1/-1/0/030328-F-JZ000-059.JPG>



## Milstar—1999 April 30

- Five of six satellites were successfully launched. The third launch failed.
- Its Centaur/Titan IV-B launch vehicle fired its roll thrusters open-loop until all fuel was depleted.
- A parameter that should have been **-1.992476** had been entered as **-0.992476**.



<https://media.defense.gov/2003/Mar/28/2000031174/-1/-1/0/030328-F-JZ000-059.JPG>



## Mars Climate Orbiter—1999 September 23

- Martian climate observation from orbit
- ~330M US\$ for program (1998)
  - ~513M US\$ (2020)

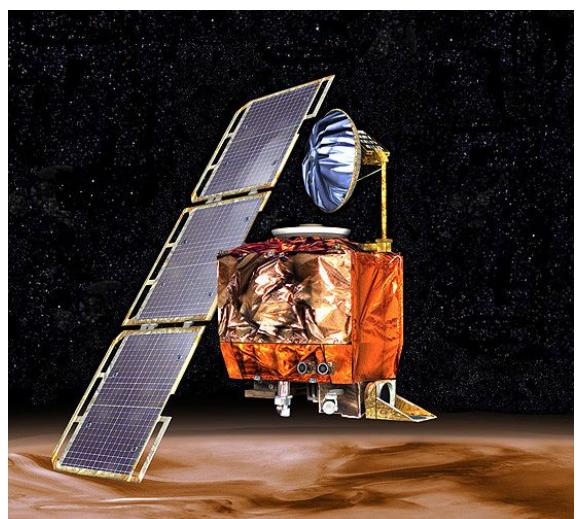


<https://nssdc.gsfc.nasa.gov/image/spacecraft/mars98orb.jpg>



## Mars Climate Orbiter—1999 September 23

- After 286 days in transit, the craft fired its engine to enter orbit.
- One piece of ground software generated results in “US Customary Units”, another expected input in “SI Units”.
- The orbiter was much closer to Mars than thought, skipped off Mars’ atmosphere, and (probably) entered solar orbit.

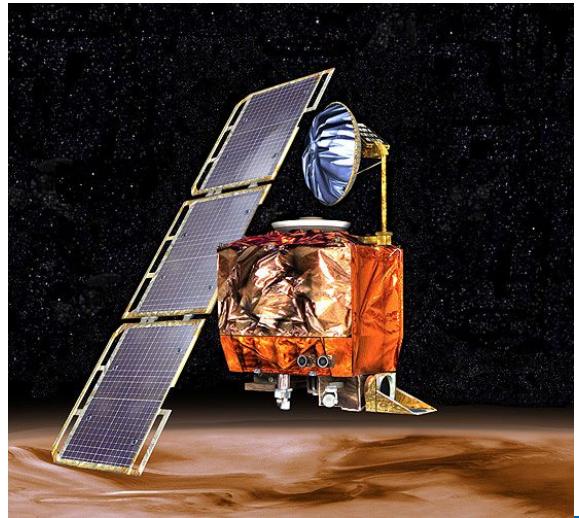


<https://nssdc.gsfc.nasa.gov/image/spacecraft/mars98orb.jpg>



## Mars Climate Orbiter—1999 September 23

- What makes this failure extra tragic is that the discrepancy between actual and expected position had been noticed at least *twice* before the failure.
- Concerns by personnel had been dismissed because they “did not follow the rules about filling out [the] form to document their concerns”.<sup>†</sup>



<https://spectrum.ieee.org/aerospace/robotic-exploration/why-the-mars-probe-went-off-course>  
<https://nssdc.gsfc.nasa.gov/image/spacecraft/mars98orb.jpg>



## Mars Climate Orbiter—1999 September 23

- The Mars Climate Orbiter captured exactly one image on its mission, on 1999 September 7 when it was about 4.5 million km from Mars.



<http://photojournal.jpl.nasa.gov/tiff/PIA02330.tif>



## Mars Polar Lander—1999 December 03

- Surface half of the Mars Climate Orbiter mission
- Deployment of the landing legs generated a false “landed” signal.
- The descent engine shut off at (probably) 40m altitude and the craft crashed.
- Was a “known” problem. A fix had been made, but the tests *were not rerun*.



[https://nssdc.gsfc.nasa.gov/image/spacecraft/mars\\_polar\\_lander.jpg](https://nssdc.gsfc.nasa.gov/image/spacecraft/mars_polar_lander.jpg)



## Mars Spirit Rover—2004 January 21

- Mars robot rover mission
- ~400M US\$ (2004)
  - ~548M US\$ (2020)

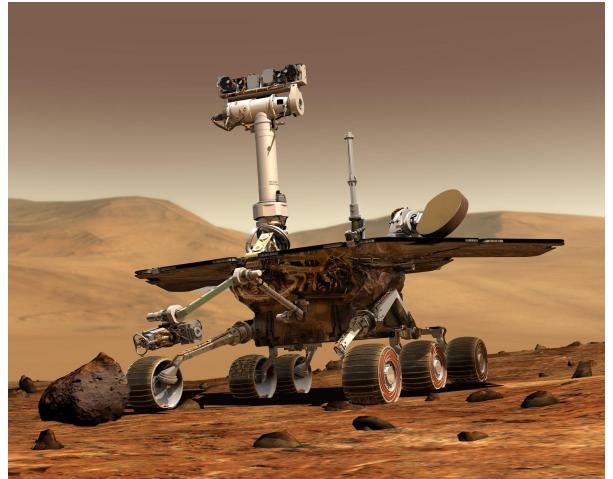


<https://www.jpl.nasa.gov/missions/web/mer.jpg>



## Mars Spirit Rover—2004 January 21

- Data was collected in Flash memory. On boot, this was copied to RAM.
- The Flash memory was bigger than the RAM, so eventually this copying overran the RAM, which caused an error.
- On error the unit rebooted, which caused an overrun error and a reboot, which caused an overrun error and a reboot, which ...
- Eventually control was restored.



<https://www.jpl.nasa.gov/missions/web/mer.jpg>



## Mars Global Surveyor—2006 June 02

- Mars global mapping mission
- ~220M US\$ (1996)
  - ~363M US\$ (2020)



[https://nssdc.gsfc.nasa.gov/planetary/image/mars\\_global\\_surveyor.jpg](https://nssdc.gsfc.nasa.gov/planetary/image/mars_global_surveyor.jpg)



## Mars Global Surveyor—2006 June 02

- After working fine since 1996 November, a software update in 2005 November incorrectly updated *a single parameter*.
- Supposedly fixed in 2006 June, a mistake in *that* update caused the *wrong memory addresses* to be used.



[https://nssdc.gsfc.nasa.gov/planetary/image/mars\\_global\\_surveyor.jpg](https://nssdc.gsfc.nasa.gov/planetary/image/mars_global_surveyor.jpg)



## Mars Global Surveyor—2006 June 02

- On 2006 November 06 a routine communication used these *wrong addresses* and turned the craft so that its *sole* working battery was *exposed to direct sunlight*.
- The battery overheated and failed in about 11 hours.



[https://nssdc.gsfc.nasa.gov/planetary/image/mars\\_global\\_surveyor.jpg](https://nssdc.gsfc.nasa.gov/planetary/image/mars_global_surveyor.jpg)



## **Simple Mistakes—Aerospace**

- There are a number of “simple mistakes” listed for various Aerospace endeavors.
- This is *not* because the Aerospace area is more susceptible to error.
- Quite the contrary: Aerospace software overall tends to be written by very smart persons who are very good at what they do.
- More items are listed simply because they tend to be *big* failures that tend to get noticed.
- Further, investigatory bodies tend to get to the bottom of what happened. We simply know more about these failures.

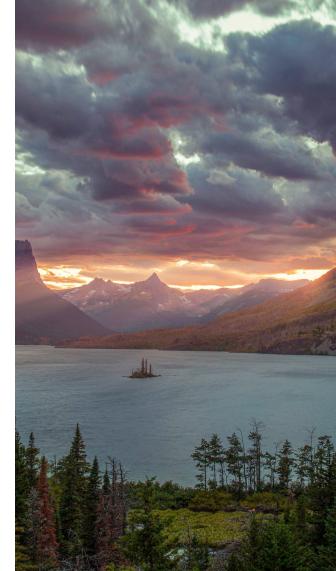


## **Simple Mistakes *Mobile Phones***



# Image Soft-Bricks Android—2020 May 30

- Trying to set this image as the background wallpaper “soft-bricks” Android phones.
- First discovered affecting the latest Samsung Galaxy, quickly shown to affect *all* standard Android  $\leq 10$  phones.
- (Android 11 preview seems OK.)



<https://twitter.com/UniverseIce/status/1266943909499826176>

# Image Soft-Bricks Android—2020 May 30

There's an uncaught exception when setting the wallpaper.

Crash.

Reboot.

There's an uncaught exception when setting the wallpaper.

Crash.

Reboot.

There's an uncaught exception when setting the wallpaper.

Crash.

Reboot.

There's an uncaught exception when setting the wallpaper.

Crash.

Reboot.

...

<https://pbs.twimg.com/media/EZU0R8eXkAEMm9l?format=jpg&name=large>

```
FATAL EXCEPTION: AsyncTask #1
process: com.android.systemui, PID: 20
java.lang.RuntimeException: An error occurred while executing doInBackground()
    at android.os.AsyncTask$4.done(AsyncTask.java:399)
    at java.util.concurrent.FutureTask.finishCompletion(FutureTask.java:383)
    at java.util.concurrent.FutureTask.setException(FutureTask.java:252)
    at java.util.concurrent.FutureTask.run(FutureTask.java:277)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1112)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:602)
    at java.lang.Thread.run(Thread.java:919)
Caused by: java.lang.ArrayIndexOutOfBoundsException: length=256; index=256
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold.getHistogram(I
        processHelper.java:165)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold.computeImageP
        processHelper.java:165)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold$ComputeTask.d
        BackgroundImageProcessHelper.java:83)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold$ComputeTask.d
        BackgroundImageProcessHelper.java:77)
    at android.os.AsyncTask$3.call(AsyncTask.java:278)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    ... 4 more
FATAL EXCEPTION: AsyncTask #1
process: com.android.systemui, PID: 298
java.lang.RuntimeException: An error occurred while executing doInBackground()
    at android.os.AsyncTask$4.done(AsyncTask.java:399)
    at java.util.concurrent.FutureTask.finishCompletion(FutureTask.java:383)
    at java.util.concurrent.FutureTask.setException(FutureTask.java:252)
    at java.util.concurrent.FutureTask.run(FutureTask.java:277)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1112)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:602)
    at java.lang.Thread.run(Thread.java:919)
Caused by: java.lang.ArrayIndexOutOfBoundsException: length=256; index=256
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold.getHistogram(I
        processHelper.java:165)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold.computeImageP
        processHelper.java:165)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold$ComputeTask.d
        BackgroundImageProcessHelper.java:83)
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold$ComputeTask.d
        BackgroundImageProcessHelper.java:77)
    at android.os.AsyncTask$3.call(AsyncTask.java:278)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    ... 4 more
FATAL EXCEPTION: AsyncTask #2
process: com.android.systemui, PID: 224
java.lang.RuntimeException: An error occurred while executing doInBackground()
    at android.os.AsyncTask$4.done(AsyncTask.java:399)
    at java.util.concurrent.FutureTask.finishCompletion(FutureTask.java:383)
    at java.util.concurrent.FutureTask.setException(FutureTask.java:252)
    at java.util.concurrent.FutureTask.run(FutureTask.java:277)
    at android.os.AsyncTask$3.call(AsyncTask.java:269)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1112)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:602)
    at java.lang.Thread.run(Thread.java:919)
Caused by: java.lang.ArrayIndexOutOfBoundsException: length=256; index=256
    at com.android.systemui.wallpaper.ImageProcessHelper$Threshold.getHistogram(I
        processHelper.java:165)
```

# Image Soft-Bricks Android—2020 May 30

- `getHistogram()` doesn't allow a pixel to have an intensity of  $\geq 256$ . (Should it?)

```
129     private int[] getHistogram(Bitmap grayscale) {
130         int width = grayscale.getWidth();
131         int height = grayscale.getHeight();
132
133         // TODO: Fine tune the performance here, tracking on b/123615079.
134         int[] histogram = new int[256]; ←
135         for (int row = 0; row < height; row++) {
136             for (int col = 0; col < width; col++) {
137                 int pixel = grayscale.getPixel(col, row);
138                 int y = Color.red(pixel) + Color.green(pixel) + Color.blue(pixel);
139                 histogram[y]++; ←
140             }
141         }
142
143         return histogram;
144     }
```

<https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/packages/SystemUI/src/com/android/systemui/glwallpaper/ImageProcessHelper.java>



# Image Soft-Bricks Android—2020 May 30

- It's a one-line fix. (But *why* does this happen?)

```
129     private int[] getHistogram(Bitmap grayscale) {
130         int width = grayscale.getWidth();
131         int height = grayscale.getHeight();
132
133         // TODO: Fine tune the performance here, tracking on b/123615079.
134         int[] histogram = new int[256];
135         for (int row = 0; row < height; row++) {
136             for (int col = 0; col < width; col++) {
137                 int pixel = grayscale.getPixel(col, row);
138                 int y = Color.red(pixel) + Color.green(pixel) + Color.blue(pixel);
139                 if (y > 255) y = 255; ←
140                 histogram[y]++;
141             }
142         }
143
144         return histogram;
145     }
```

<https://android-review.googlesource.com/c/platform/frameworks/base/+/1321016/1/packages/SystemUI/src/com/android/systemui/glwallpaper/ImageProcessHelper.java>



# Simple Mistakes

## *Insurance*



### **“Double Rounding” Litigation—1995 August 28**

- Premiums for automobile insurance were being “double rounded”.
  - E.g., 9.46 rounded to one decimal place is 9.5. Then rounded to zero decimal places, this becomes 10.
  - On the other hand, 9.46 directly rounded to zero decimal places is 9.
- This allowed the insurance companies to collect excess premiums from consumers.



## **“Double Rounding” Litigation—1995 August 28**

- Though this had been “common practice” for quite some time, the statute of limitations restricted the ensuing litigation to only two (2) years of alleged wrongdoing.
- Even so, the insurance companies involved agreed that *tens of millions of dollars* of excess premium had been collected during the litigated period.
- The result was a class action lawsuit on behalf of 4,401,817 policy holders.



## **“Double Rounding” Litigation—1995 August 28**

- The eventual settlement by the insurance companies was for 35,659,878 US\$ (1998), or about 57M US\$ (2020).
  - This does not include how much the insurance companies spent trying to defend themselves.
- Of that 35,659,878 US\$, the attorneys got 10,349,430 US\$.
- The remaining 25,310,448 US\$ was split amongst the injured parties, who got just about 5.75 US\$ each.



# Simple Mistakes

## *Medical*



### Therac-25—1985 June 03

- The Therac-25 was a radiation therapy device.
- In certain circumstances, it delivered *100x* the intended dose of radiation.

PATIENT NAME: John	BEAM TYPE: E	ENERGY (KeV):	10
TREATMENT MODE: FIX	ACTUAL	PRESCRIBED	
UNIT RATE/MINUTE	0.000000	0.000000	
MONITOR UNITS	200.000000	200.000000	
TIME (MIN)	0.270000	0.270000	
GANTTRY ROTATION (DEG)	0.000000	0.000000	VERIFIED
COLLIMATOR ROTATION (DEG)	359.200000	359.200000	VERIFIED
COLLIMATOR X (CM)	14.200000	14.200000	VERIFIED
COLLIMATOR Y (CM)	27.200000	27.200000	VERIFIED
WEDGE NUMBER	1.000000	1.000000	VERIFIED
ACCESSORY NUMBER	0.000000	0.000000	VERIFIED
DATE: 2012-04-16	SYSTEM: BEAM READY	OP.MODE: TREAT	AUTO
TIME: 11:48:58	TREAT: TREAT PAUSE	X-RAY	173777
OPR ID: 033-tfs3p	REASON: OPERATOR	COMMAND:	█



## Therac-25—1985 June 03

- Hardware interlocks which prevented this in previous models had been removed in favor of software checks.
- The software had been created by *a single individual who was also responsible for all testing*.
- Later review identified three major contributing factors,
  - Developer overconfidence
  - A lack of a rigorous testing methodology
  - Company history of *ignoring* operator / patient complaints.

<http://sunnyday.mit.edu/papers/therac.pdf>  
[https://commons.wikimedia.org/wiki/File:Therac25\\_Interface.png](https://commons.wikimedia.org/wiki/File:Therac25_Interface.png)



## Therac-25—1985 June 03

- Three persons died, one was crippled, one would have been crippled but died of something else, one recovered.
- Cost unclear—all lawsuits were settled out-of-court and the details are sealed.

PATIENT NAME:	John	TREATMENT MODE:	FIX	BEAM TYPE:	E	ENERGY (KEV):	10
UNIT RATE/MINUTE	0.000000	ACTUAL	0.000000	PRESCRIBED			
MONITOR UNITS	200.000000		200.000000				
TIME (MIN)	0.270000		0.270000				
GANTTRY ROTATION (DEG)	0.000000		0.000000				VERIFIED
COLLIMATOR ROTATION (DEG)	359.200000		359.200000				VERIFIED
COLLIMATOR X (CM)	14.200000		14.200000				VERIFIED
COLLIMATOR Y (CM)	27.200000		27.200000				VERIFIED
WEDGE NUMBER	1.000000		1.000000				VERIFIED
ACCESSORY NUMBER	0.000000		0.000000				VERIFIED
DATE: 2012-04-16		SYSTEM: BEAM READY		OP.MODE: TREAT		AUTO	
TIME: 11:48:58		TREAT: TREAT PAUSE		X-RAY			173777
OPR ID: 033-tfs3p		REASON: OPERATOR		COMMAND: █			

<http://sunnyday.mit.edu/papers/therac.pdf>  
[https://commons.wikimedia.org/wiki/File:Therac25\\_Interface.png](https://commons.wikimedia.org/wiki/File:Therac25_Interface.png)



# Simple Mistakes

## *General Software*



### Bugs ...

- The following is a summary of a bug release for a fairly widely-used piece of software.
  - It's been around for *years*.



## 2.3.0 release

The [REDACTED] team is proud to announce the 2.3.0 release of [REDACTED].

Sun 22 March 2020

Security related fixes:

- *Double-free vulnerability in [REDACTED].*
- *Null pointer reference at [REDACTED].*
- *Integer signedness error.*
- *Using uninitialized variables.*
- *Heap-based buffer overflow.*
- *Double-free in [REDACTED].*

By [REDACTED]

In [News](#).

For full list of changes, see [CHANGELOG.md](#).

This is a recommended update.

You can download the 2.3.0 version of [REDACTED] from the [REDACTED].

Check out the [full commits list](#) since the previous release.



# Simple Mistakes *Financial*



## Citibank — 2020 August 11

- Citibank was acting as an agent for Revlon, a company that had borrowed about 1.8 *billion* US\$ total from a syndicate of lenders.
- An interest payment of about 7.8 *million* US\$ was due.
  - The loan itself didn't mature until 2023 September 7.
- In order to pay the interest, Citibank had to “pretend” to be paying off the *entire* loan amount (principal *and* interest) but “redirect” the principal payment back to Citibank.



## Citibank — 2020 August 11

- The “Internal GL” field is the account number in Citibank’s internal general ledger.
- Looks OK, right? The principal payment is set to go to an internal account.
- (Ugly GUI, eh?)

BDLL	Borrower LIBOR Drawdown Prod	Drawdown
001BDLL201480094		001BDLL201480094
Facility Name	024462 REVOL CONSUMER PRODUCTS CORP	
REVLON TERM LOAN 2016		
GL Detail		
Component	Internal GL	Overwrite default settlement instruction
COLLAT		<input type="checkbox"/>
COMPINTSF		<input type="checkbox"/>
DEFUAL		<input type="checkbox"/>
DFLFTC		<input type="checkbox"/>
FRONT		<input type="checkbox"/>
FUND		<input type="checkbox"/>
INTEREST		<input type="checkbox"/>
PRINCIPAL	3003000023	<input checked="" type="checkbox"/>

<https://arstechnica.com/tech-policy/2021/02/citibank-just-got-a-500-million-lesson-in-the-importance-of-ui-design/>  
[https://www.courtlistener.com/recap/gov.uscourts.nysd.542310.gov.uscourts.nysd.542310.243.0\\_2.pdf](https://www.courtlistener.com/recap/gov.uscourts.nysd.542310.gov.uscourts.nysd.542310.243.0_2.pdf)



## Citibank — 2020 August 11

- Bzzt! It's *not* OK.
- For whatever bizarre reason, the GUI requires that not only the PRINCIPAL box should be checked, but also the FRONT and FUND boxes.
- Too bad, *all* of the money left Citibank. (893,944,008.52 US\$)
  - *And it came out of Citibank's wallet, not Revlon's.*
    - The only Revlon money that Citibank had was the 7.8 million US\$ for the required interest payment.
  - Oops!



## Citibank — 2020 August 11

- Citibank sued in an attempt to get the money back.
- Citibank claimed the recipients should have known it was a mistake and the recipients should return the money.
- The recipients laughed so hard they did themselves multiple internal injuries. They then refused to return the money.
  - Revlon debt was trading at about *40 cents on the dollar.* (!)
- The court case dragged on and eventually the judge ruled ...



## Citibank — 2020 August 11

The transfers matched — to the penny — the amount of principal and interest outstanding on the loan. The accompanying notices referred to interest being “due,” and the only way in which that would have been accurate was if Revlon was making a principal prepayment. And it appears that no mistake of the size or nature of Citibank’s had ever happened before. Faced with these circumstances, the Non-Returning Lenders believed, and were justified in believing, that the payments were intentional. Indeed, to believe otherwise — to believe that Citibank, one of the most sophisticated financial institutions in the world, had made a mistake that had never happened before, to the tune of nearly \$1 billion — would have been borderline irrational.

Accordingly, and for the reasons discussed above, the Court holds that the August 11th wire transfers at issue were “final and complete transaction[s], not subject to revocation.”

[https://www.courtlistener.com/recap/gov.uscourts.nysd.542310/gov.uscourts.nysd.542310.243.0\\_2.pdf](https://www.courtlistener.com/recap/gov.uscourts.nysd.542310/gov.uscourts.nysd.542310.243.0_2.pdf)



## Wells Fargo

- From 2010 April 13 through 2015 October 20, Wells Fargo screwed up whether or not home owners were eligible for mortgage modifications. (5 years, 6 months, 8 days!)
- About 625 families were denied modifications for which they were actually eligible.
- About 400 of those families *lost their homes*.
- Oops!



## Wells Fargo

- An expanded review revealed the problem actually occurred from 2010 March 15 through 2018 April 30. (8 years, 1 month, 16 days!)
- Now it's about 870 affected families.
- And about 545 of those families *lost their homes*.
- Oops! Oops! Oops!



## Wells Fargo

- So what happened?
- Wells Fargo won't say *exactly* but they do say it was a "calculation error" in a "mortgage loan modification underwriting tool".
- AKA ... a **bug** in the software!

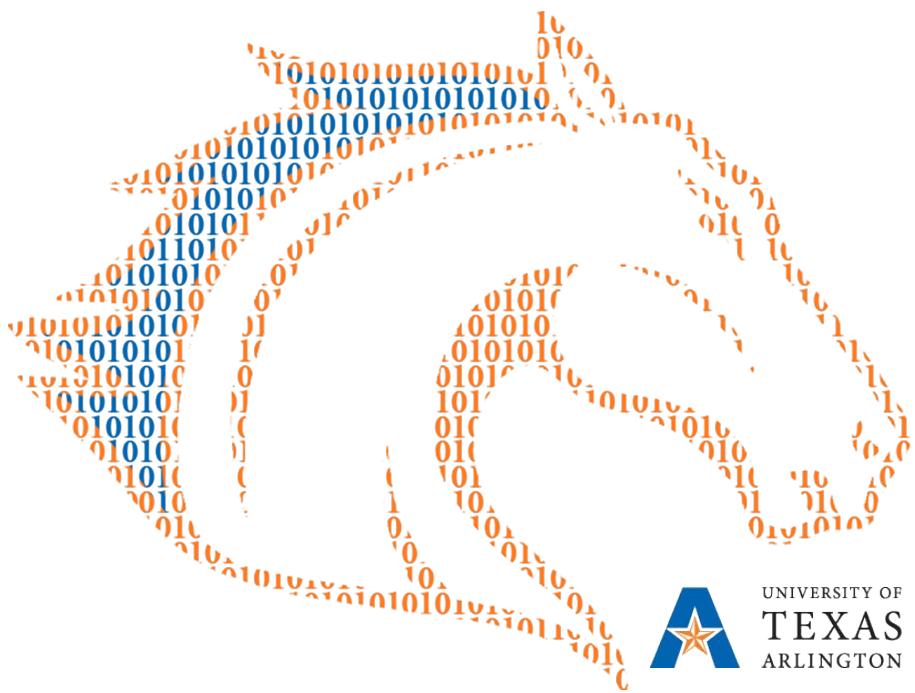


# Wells Fargo

- There are multiple lawsuits pending and at least one class action lawsuit in progress.
- Wells Fargo initially set aside 8 M\$ to take care of this but independent observers say it's going to cost plenty more than that.

Wells Fargo Form 10-Q for quarter ending 2018 Sep 30, page 6.

- **Mortgage Loan Modifications** An internal review of the Company's use of a mortgage loan modification underwriting tool identified a calculation error regarding foreclosure attorneys' fees affecting certain accounts that were in the foreclosure process between April 13, 2010, and October 2, 2015, when the error was corrected. A subsequent expanded review identified related errors regarding the maximum allowable foreclosure attorneys' fees permitted for certain accounts that were in the foreclosure process between March 15, 2010, and April 30, 2018, when new controls were implemented. Similar to the initial calculation error, these errors caused an overstatement of the attorneys' fees that were included for purposes of determining whether a customer qualified for a mortgage loan modification or repayment plan pursuant to the requirements of government-sponsored enterprises (such as Fannie Mae and Freddie Mac), the Federal Housing Administration (FHA), and the U.S. Department of Treasury's Home Affordable Modification Program (HAMP). Customers were not actually charged the incorrect attorneys' fees. As a result of these errors, taken together and subject to final validation, approximately 870 customers were incorrectly denied a loan modification or were not offered a loan modification or repayment plan in cases where they otherwise would have qualified. In approximately 545 of these instances, after the loan modification was denied or the customer was deemed ineligible to be offered a loan modification or repayment plan, a foreclosure was completed. The Company has contacted a substantial majority of the approximately 870 affected customers to provide remediation and the option also to pursue no-cost mediation with an independent mediator. Attempts to contact the remaining affected customers are ongoing. Also, the Company's review of these matters is ongoing, including a review of its mortgage loan modification tools.



UNIVERSITY OF  
TEXAS  
ARLINGTON

DEPARTMENT OF  
COMPUTER SCIENCE  
AND ENGINEERING

# Simple Mistakes

## *Quadriga*



### Quadriga

- As more of the Quadriga case unfolds it seems to be more of an out-and-out fraud rather than a “simple mistake”.
- Let’s let the dust settle for a bit and then reconsider just what happened with Quadriga.



## Quadrige

- Quadrige was a Canadian Cryptocurrency exchange, founded by Gerald Cotten.
- Mr. Cotten died on 2018 December 9 due to complications from Crohn's disease.
- So that's that, right?
  - After all, people die every day.



## Quadrige

- No one knows how to unlock the cryptocurrency wallets.
- Mr. Cotten took all the passwords with him to the grave.
- Various security experts have been trying to decrypt Mr. Cotten's laptop.
  - No success so far.
- About 190 M\$ worth of cryptocurrency is unable to be accessed by anyone.
- Oops!



## Quadriga

- On 2019 January 31, Quadriga's board of directors was forced to apply for creditor protection, saying ...

*"For the past weeks, we have worked extensively to address our liquidity issues, which include attempting to locate and secure our very significant cryptocurrency reserves held in cold wallets, and that are required to satisfy customer cryptocurrency balances on deposit, as well as sourcing a financial institution to accept the bank drafts that are to be transferred to us.*

*"Unfortunately, these efforts have not been successful."*

<https://news.sky.com/story/crypto-trader-boss-dies-with-password-to-clients-110m-11628106>



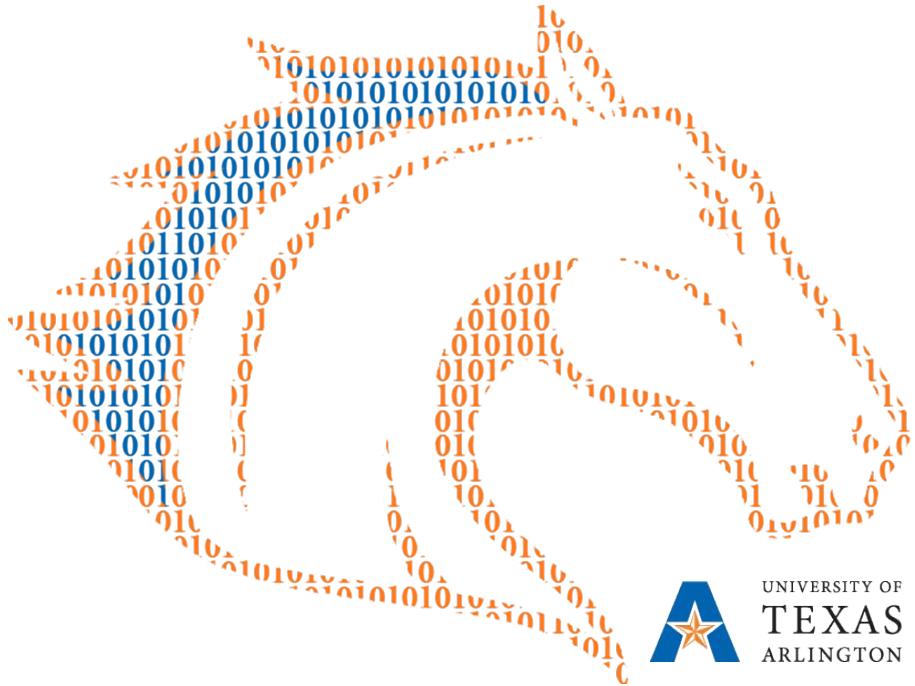
## Quadriga

- Some Quadriga clients are questioning whether or not Mr. Cotten is even dead.
  - They are openly speculating it was all an elaborate scam.
- Being out 190 M\$, I might speculate as well ...
- Lots more details at Amy Castor's summary timeline:

<https://amycastor.com/2019/02/12/how-the-hell-did-we-get-here-a-timeline-of-quadrigacx-events/>

<https://www.investinblockchain.com/is-quadriga-ceo-actually-dead-or-140-million-exit-scam/>





UNIVERSITY OF  
**TEXAS**  
ARLINGTON

DEPARTMENT OF  
COMPUTER SCIENCE  
AND ENGINEERING