



# Technical Debt in Space

- Jesse Warden
- Software Developer at Capital One

# Why?

- New technology ideas
- Build your self-esteem
- Learn a tiiiiiiiiiiency piece of software in space history

# How?

- cherry picked 5 US and 1 European, successful and failed space missions
- Chronological
- Caveat: My interpretations of good or inaccurate documentation
- I'm not providing entire context, just picking the parts I loved.

# Technical Debt

- Managers who have never created technical debt don't understand it.
- Developers use an analogy, of “debt”, to communicate and negotiate.
- Ward Cunningham who coined it was a dev working in finance.

# Debt Example

- Dad's Boatlift Business
- Needs: trucks, tools, barges, pile drivers
- Profitable in 18 months if he can get it all
- Doesn't have \$400,000 in cash

# Technical Debt Example

- I'm an FP Developer who wants to leverage Suspense for lots of data
- I've never unit tested React Hooks
- I need to learn both on a real-project
- Red Green Refactor

# Good Technical Debt Example

- I'm an FP Developer who wants to leverage Suspense for lots of data
- I've never unit tested React Hooks
- I need to learn both on a real-project
- Red Green Refactor

# Red Green Refactor (TDD)

- Red: write failing test for your feature
- Green: make the test pass with as little code as possible; it's ok if the code is ghetto fab
- Refactor: Make the code better

# Bad Debt Example

- Don't pay off loan for awhile
- at some point you're paying nothing but interest

# Bad Technical Debt Example

- No unit tests
- No automated deployments
- Violating DRY because you're in a hurry
- No love

# Bad Technical Debt

- Unpaid Technical debt has a huge human and monetary cost and leads to various failures in software, productivity, and software developer retention
- Lots of technical debt that you're not allowed to pay down is not **fun**.

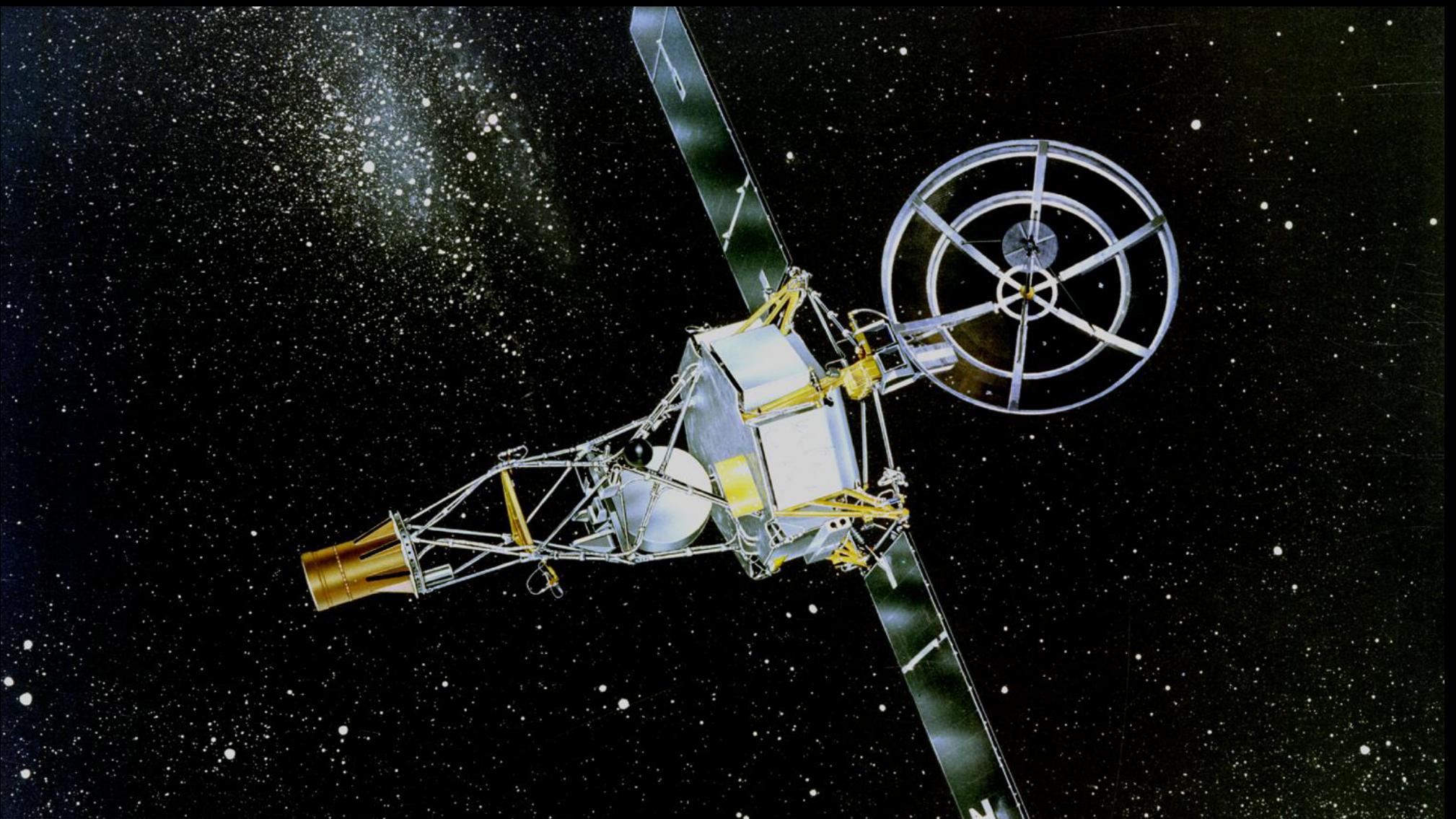


# Mariner 1

Planetary flyby of Venus, launched July 22, 1962

# Mariner 1

- First planetary mission
- Interplanetary probe
- Planetary orbiter
- gravity assist



Rocket blew up 5  
minutes after launch

\$18.5 million



# Mariner 1 Points

- No clue what went wrong
- Engineers & officials dumbed down explanations
- Contradicted each other
- Deadline for Mariner 2



# Hidden Figures

Good movie

# Mariner 1 Failure Cause Theories

Description	Sample	Unicode	CSS/HTML
Overline (markup)	$\overline{Xx}$	N/A	<code>text-decoration: overline;</code>

- Missing overbar
- Missing Hyphen
- Comma instead of dot in FORTRAN, loop went nuts

# Lessons

- Unit tests
- Property Tests (fuzz tests)



# John Hughes of Haskell fame

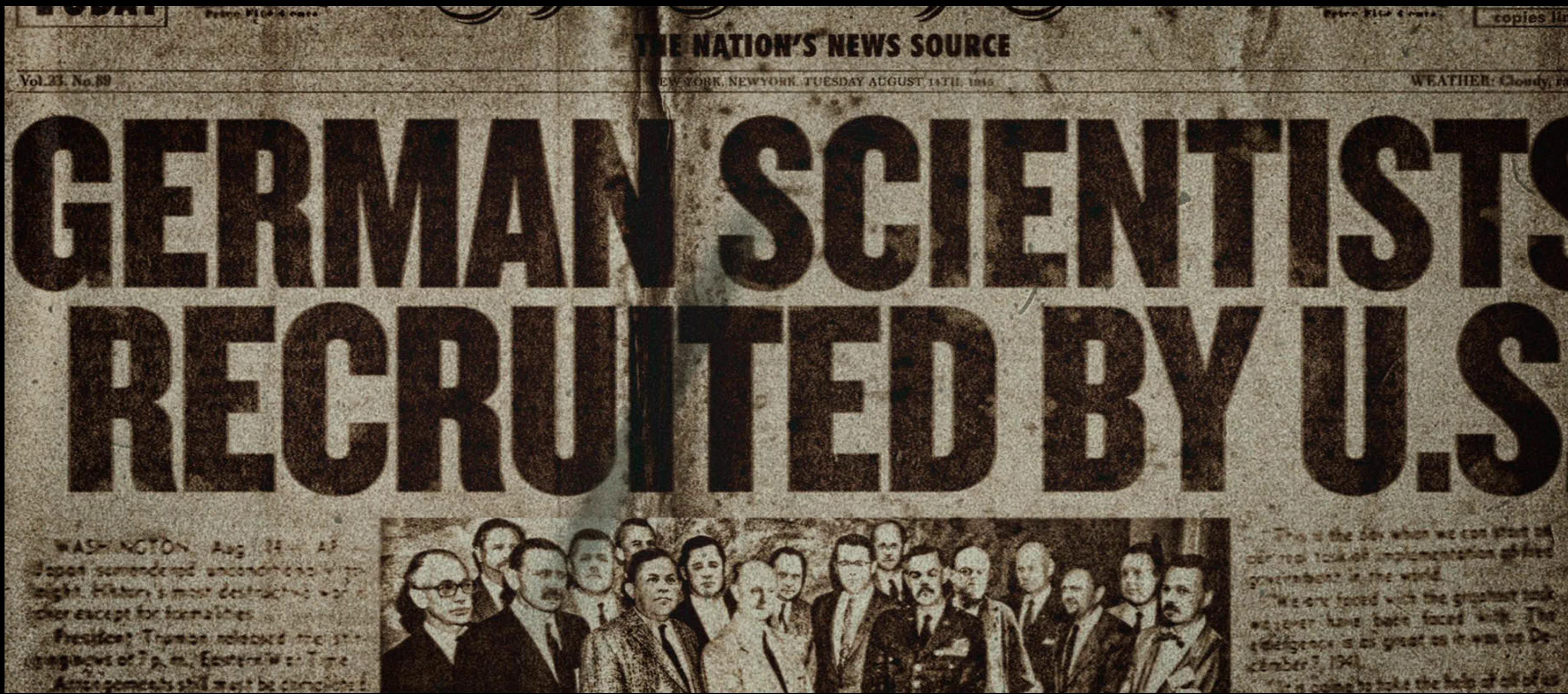
Quickcheck: Generate Unit Tests

[https://www.youtube.com/watch?v=hXnS\\_Xjwk2Y](https://www.youtube.com/watch?v=hXnS_Xjwk2Y)



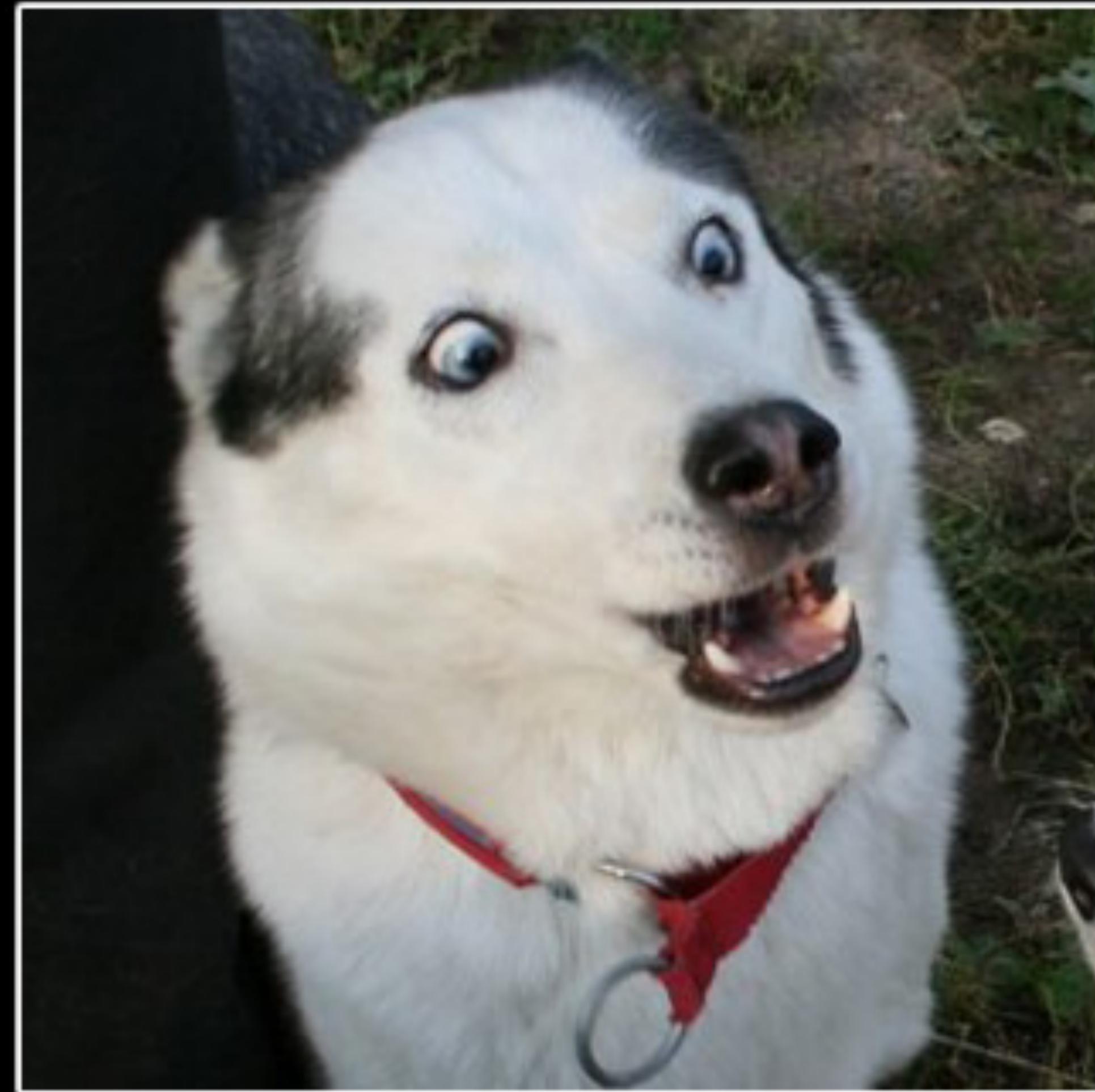
# Saturn 5

Rocket that took men to the moon.



# Operation Paperclip

1,600 German scientists  
V-2 rocket designer



**Moar  
Unit Tests!!!11oneone**

“you don't want to be testing piece-wise in space. You want to test the entire system because who knows which one's going to fail, and you'd better have it all together so that whatever fails, you have a reasonable chance of finding the real failure mode, not just the one you were looking for.””

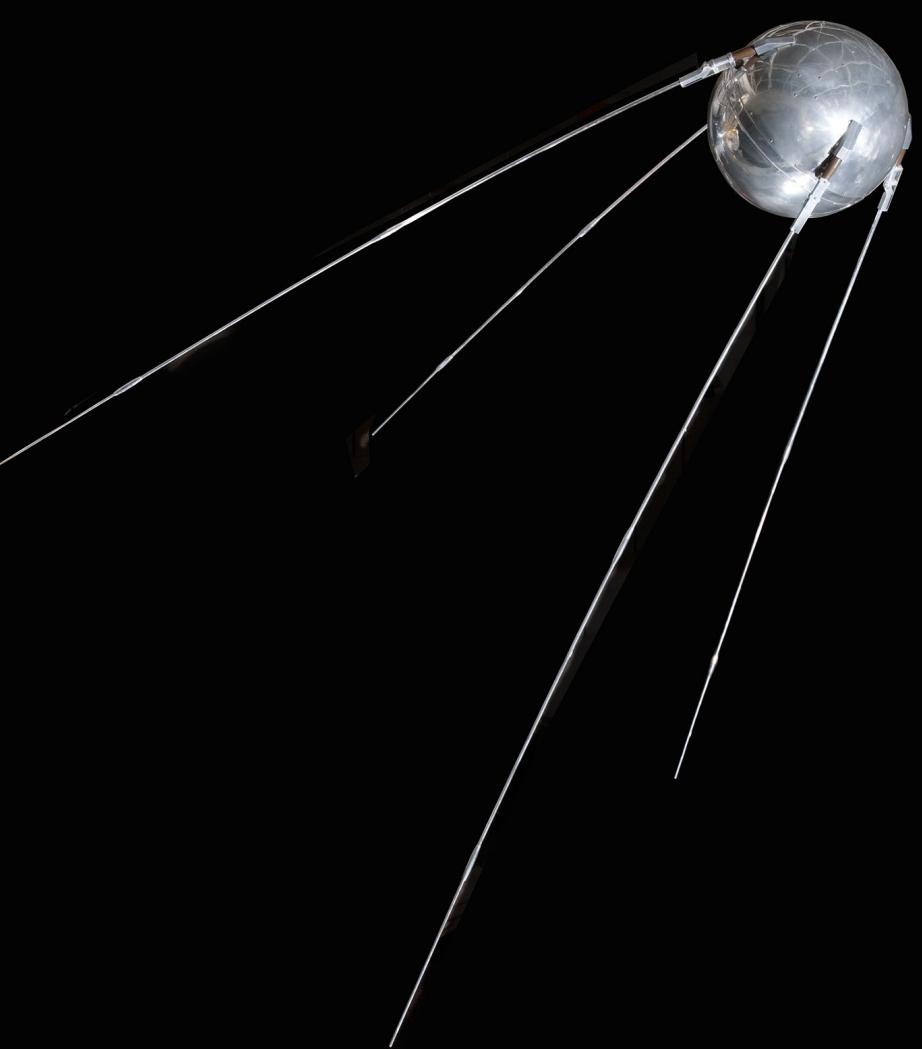
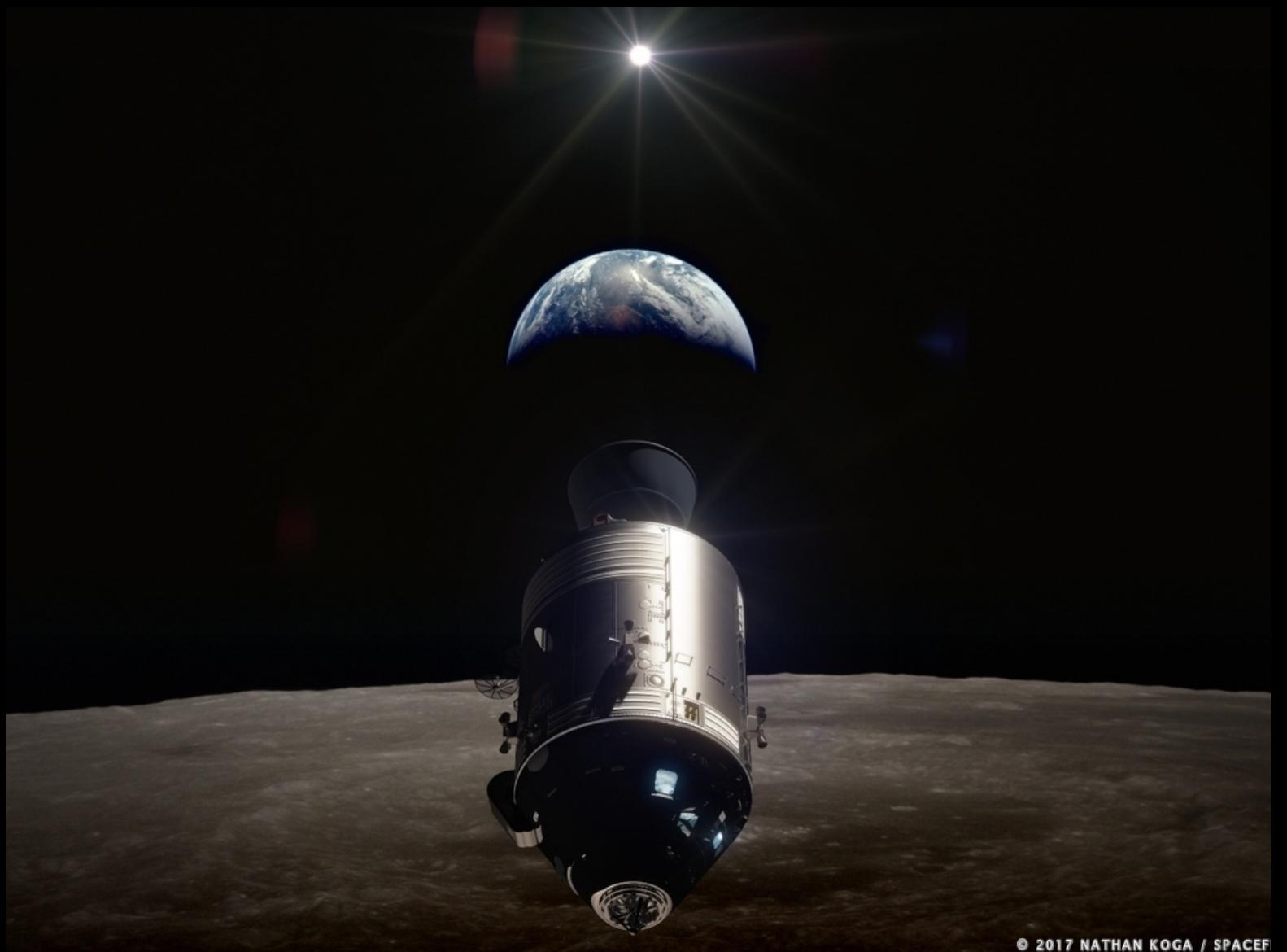
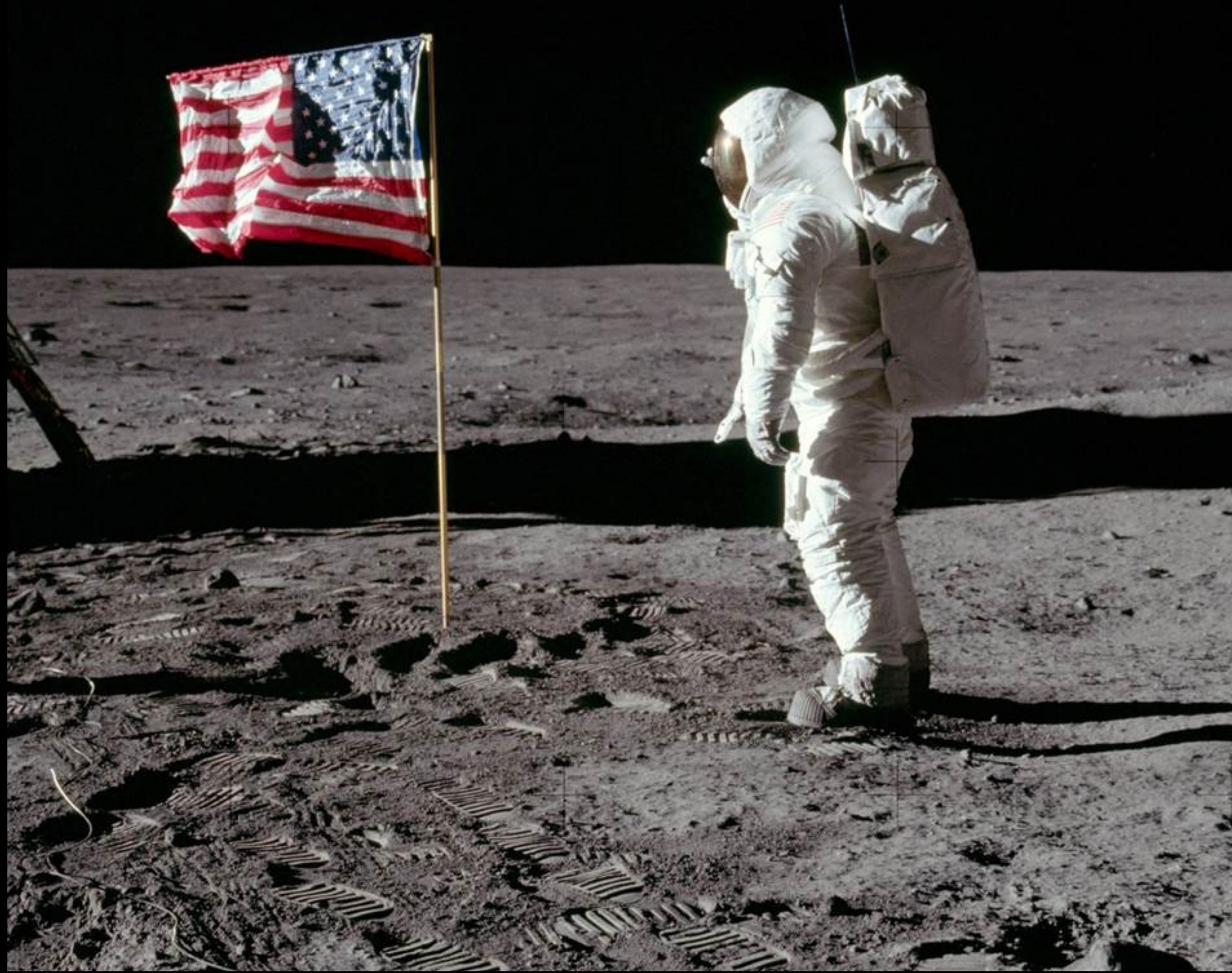
*—George Mueller*

# Full-stack, not just green field

- **Front-end:** works in particular browsers
- **Front-end API:** JSON API sends back HTTP 500 from webserver
- **API:** code's legit, but Database works, IAM Policies work, JWT security works
- **Orchestration API:** increased surface area, you get what they actually publish in the response and handle partial successes
- **Database:** Does it actually scale with perf tests, is the data good or not, does your API correctly do pooling, does your API take too long to combine the data.

# Staturn 5

- Carried Astronauts around the moon on its 3rd flight vs. 10th or later as originally planned.
- Saved millions of dollars, years of development, and probably enabled us to win the space race against the Soviets.



© 2017 NATHAN KOGA / SPACEF

# Apollo Missions

- **Sputnik 1**, first satellite launched by the Russians, freaked Americans out
- **Mercury**: Put man into space, safely return him, and learn
- **Gemini**: Put 2 men into space, safely return them, and learn about space maneuvers.
- **Apollo**: Put 3 men into space, land them on the moon, and return them safely.

# Apollo Guidance Computer

- NASA had no clue; in the contract they gave to MIT instead of IBM for the rockets, they basically put “Y’all figure out how to do... things.”
- Assembly Code
- Used Core Rope Memory (LOL Memory, Little Old Lady)
- Bad code you wrote was litterly weaved into it

# Margaret Heafield Hamilton

- Margaret Heafield Hamilton, lead the AGC program
- She jokingly called it “Software Engineering” as her title because... no one knew wtf software devs did. It stuck. Goal was sound, though: give legitimacy to it just like other engineering disciplines so those building it got respect.
- Career is preventing errors.





# Apollo 8

- Margaret's daughter hit pre-launch button, crashed the computer
- She tried to fix, NASA wouldn't let her
- “Test pilots are highly trained, they won’t press the wrong button”



# Apollo 8

- Men were so tired and couldn't sleep well, started making mistakes.
- Apollo 8 around the moon, pilot pressed the pre-launch button, crashed the computer, and it deleted some of the memory.
- ... she even had a sticker "Don't press P1 switch after you launch morons"



# Apollo 11

- Beeping errors when Lunar Module detaches and their 3 minutes from surface
- Houston knew it was bogus error
- Computer warning can't perform all tasks in real time
- parallel computing
- priority scheduling



# Apollo 11

- Near surface, Buzz Aldrin got irritated and grabbed the stick to fly manually
- parallel computing
- priority scheduling
- Concurrency
- AMAZE error handling



# Apollo 8 & 11 Lessons

- Good error handling
- Erlang/Elixir which run on the BEAM engine
- Akka for Scala or Java
- pm2/forever for Node process
- Webworkers in Browser



# Space Shuttle Challenger

10th mission on January 28, 1986

# Challenger's 10th Mission

- First teacher in space
- Observe Haley's Comet for 6 days
- 73 seconds into flight broke apart, killing all 7 crew members aboard
- A lot of kids, myself included, saw it on TV.
- A lot of times. Like 9/11, over and over and over.

# 2 Operating Systems

- Time to Release code
  - before challenger: 2 weeks
  - after challenger: 18 months



# 2 Operating Systems

- 1,000 unit tests for every line of code
- code reviews to place certain lines of code next to each other in case of lightning strikes
- constantly would run out of system resources running so many unit and end to end tests & complex test simulations

# 2 Operating Systems

- After mission(s) over, you have EPIC fixture data to test with.
- Take flight data from previous mission, load it into new software.

# Management Problem

- managers chose to cling to assumptions of "goodness"
- engineers were challenged to "prove it ISN'T safe"

**"For a successful technology," he concluded, "reality must take precedence over public relations, for nature cannot be fooled."**

*-Richard Feynman*



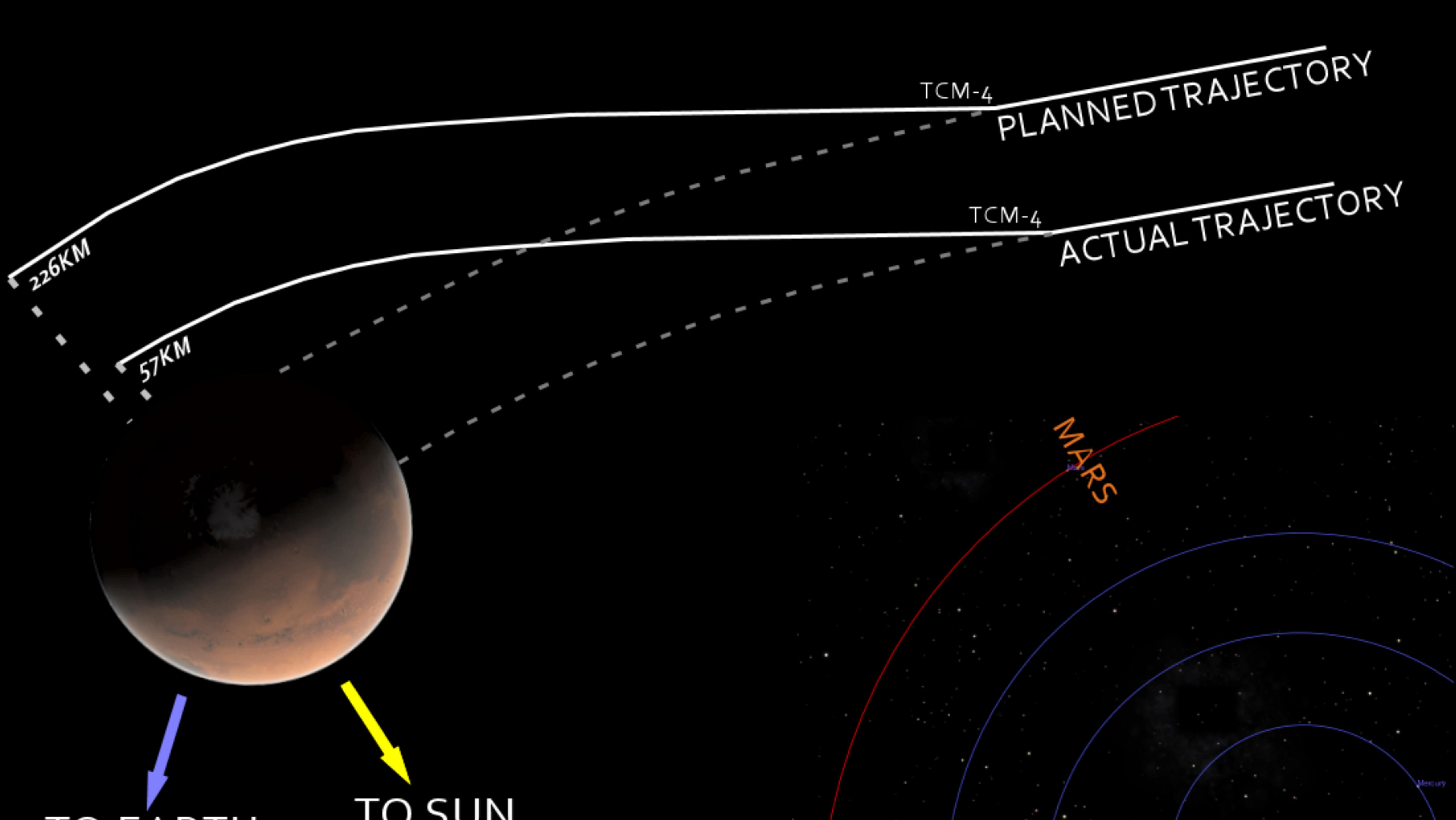
# Mars Climate Orbiter

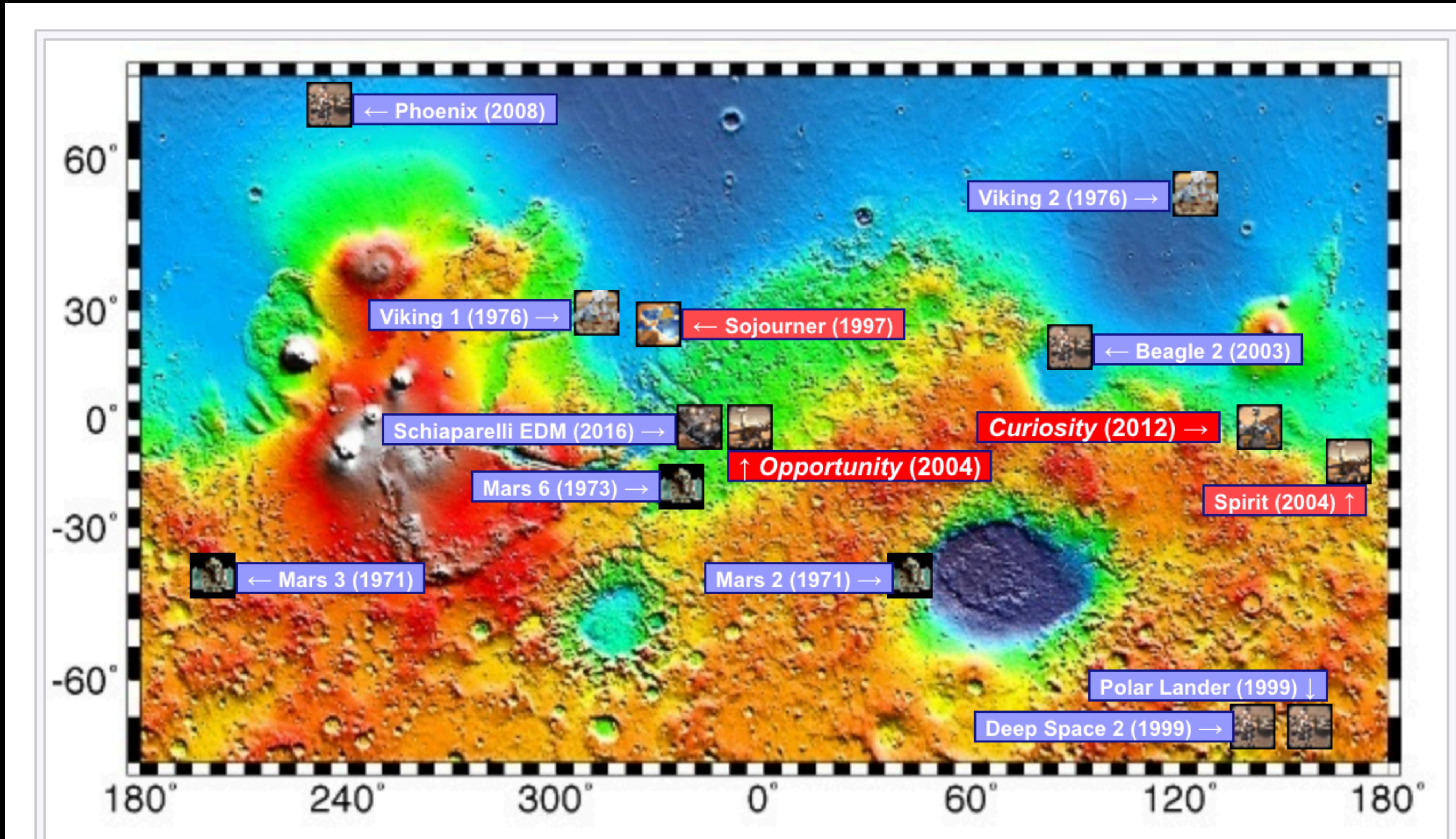
# Mars Climate Orbiter

- Orbit mars, act as a communication relay for various other lander/rovers
- 45 minutes to 2+ hours to communicate with Mars
- went too close to Mars and disintegrated in the atmosphere
- NASA specified newton-seconds (N s), Lockheed used pound-second (lb s)
- \$125 million == poof!

# Mars Climate Orbiter

- 2 navigators noticed earlier, but their concerns were dismissed by management
- even had a meeting to fix it, but didn't end up doing it
- **Mars Global Surveyor** had conversion correctly, but algorithm was too complex (or not commented enough) to read, so new code didn't include conversion





Interactive imagemap of the [global topography of Mars](#), overlaid with [locations of Mars landers and rovers](#)



Rover

Lander

**"prove something was wrong"**

**vs.**

**"prove all is right"**

**“Had we done end-to-end testing, we believe this error would have been caught. But the rushed and inadequate preparations left no time to do it right.”**

*–Developer On Mars Climate Orbiter*

**“Jet Propolsion Labratory’s process of 'cowboy' programming, and their insistence on using 30-year-old trajectory code that can neither be run, seen, or verified by anyone or anything external to JPL.**

**Sure, someone at Lockheed made a small error. If JPL did real software configuration and control, the error never would have gotten by the door.”**

*—Developer On Mars Climate Orbiter*

# Lessons from Mars Climate Orbiter

- Leadership invested in technology best practices
- Modern languages
- Unit & End to End Tests
- Code Comments



# Ariane 5

Maiden flight June 4th, 1996

# Ariane 5

- Launch Cluster 1 to study earth's magnetosphere
- Exploded 40 seconds after liftoff
- Decade of work, \$7 billion to build, rocket + cargo cost \$500 million



# Ariane 5

- 64 bit to 16 bit integer conversion because Ariane 5 engine was much more powerful than Ariane 4
- 32,767 is too big for 16 bit signed integer, threw exception... which had no try/catch.
- 3 others scenarios DID have exception handling.
- Copy pasted the code from Ariana 4

# Lessons From Ariane 5

- Proper error handling
- Manual number conversion and/or compiler help
- Property Tests

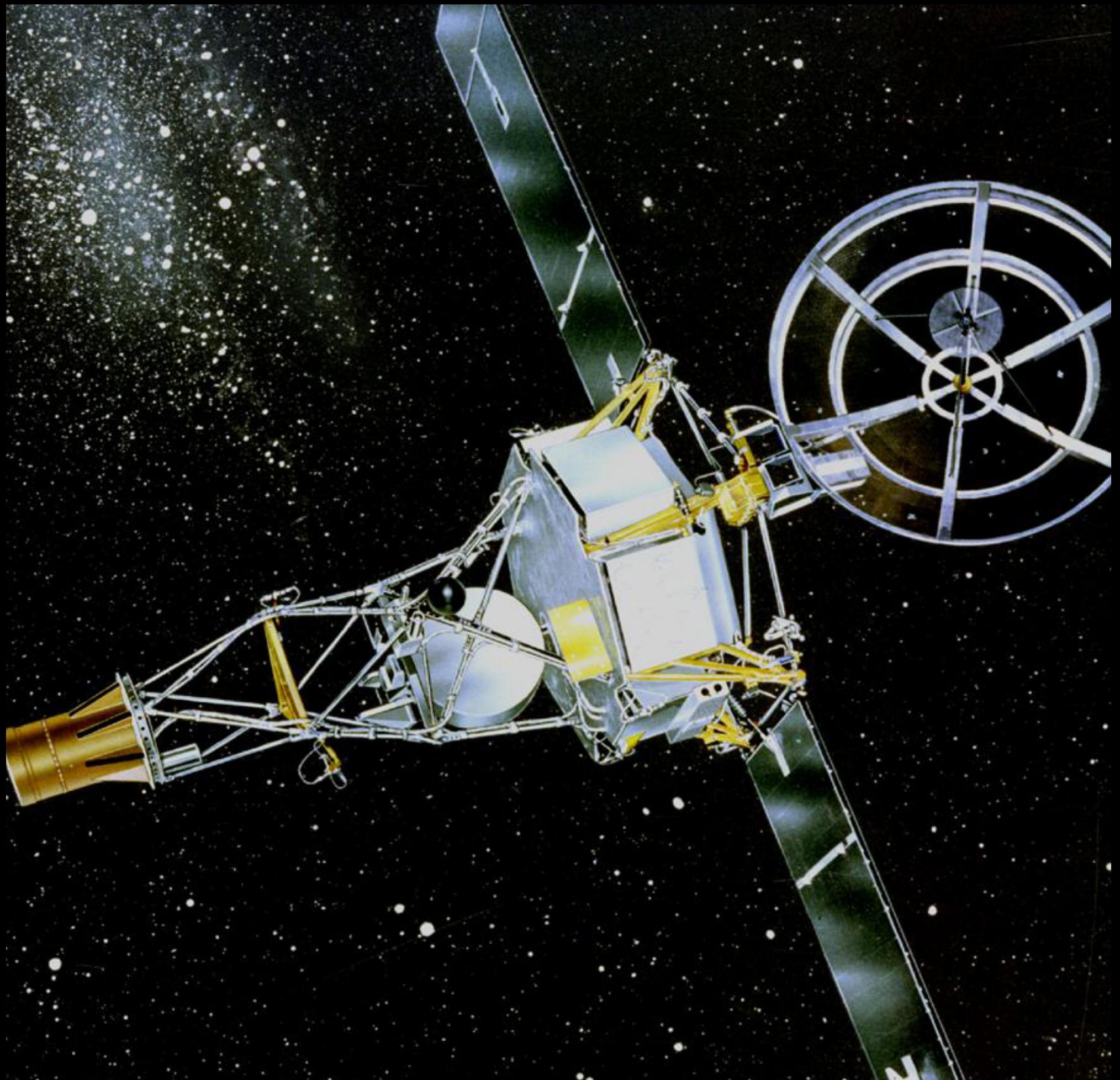
# Conclusions

- Technical Debt: Manage by Red Green Refactor
- ... tons of tests aren't the answer.
- Many different TYPES of tests help.
- Formal methods are best, but hard heh.
- <https://www.youtube.com/watch?v=EZ05e7EMOLM>



# Conclusions for Mariner 1

- Unit tests to ensure algorithms work as expected
- Unhappy and happy paths
- Property Tests to ensure whack data gives consistent results



# Conclusions for Saturn 5

- Test everything together as soon as possible.
- “All Up Testing”



# Conclusions for Apollo 8 and 11

- Concurrency with priorities is cool, but...
- testing via TLA+ helps ensure unknown scenarios (buy Hillel Wayne's book <http://is.gd/ptlaplus> )
- If you don't have kids, use chaos engineering.
- Netflix has Chaos Monkey



# Conclusions for Challenger

- Ethical management
- Safe whistleblowing for engineers



# Conclusions for Mars Climate Orbiter

- End to end testing (black box, functional)
- Code comments
- Don't ask permission, ask for forgiveness
- Leadership invested in quality engineering practices



# Conclusions for Ariane 5

- try/catch === no stack overflows
- Manual integer conversion + compiler help
- Property Tests
- Copy pasta is fine... just ensure the tests you copied are good too



# Thanks for your time!

- Learn more at my YouTube channel <https://www.youtube.com/user/jesterxl>
- my blog <http://jessewarden.com>
- Code examples <https://github.com/jesterxl>
- Questions? Contact me
- (I cuss here) Twitter: @jesterxl or Facebook
- (I don't cuss here) LinkedIn: <https://www.linkedin.com/in/jessewarden/>
- email: [jesterxl@jessewarden.com](mailto:jesterxl@jessewarden.com)