Reclaiming Your Inner Geek: Lessons for Systems Engineering from Safety Culture and Computer Science

Steven Jenkins
Principal Engineer

Jet Propulsion Laboratory
California Institute of Technology

Origins of this Presentation

- I recently heard yet another in a seemingly endless series of talks about “innovation”
- It seems we are constantly being implored to
  - Innovate!
  - Be Disruptive!
- This seems like useless advice to me
- Let me show you why….

Quiz

What US Company was Named Most Innovative by Fortune Magazine for size consecutive years (1995-2000)?

[1]

How'd That Work Out?

[2]

Since We're In Glasgow

- We must address the "No True Scotsman" defense here, namely that Enron wasn't truly innovative; they were fraudulent
- "Innovative" and "fraudulent" are not mutually exclusive
- In fact, Enron was engaged in a great many genuinely innovative activities, some of which
  - were fraudulent, and
  - despite being fraudulent, lost a great deal of money
- Innovative really just means *new*
- It doesn't mean *good*

More Innovation!

[3]

[4]

[5]

What About Being Disruptive?

[6]

[7]

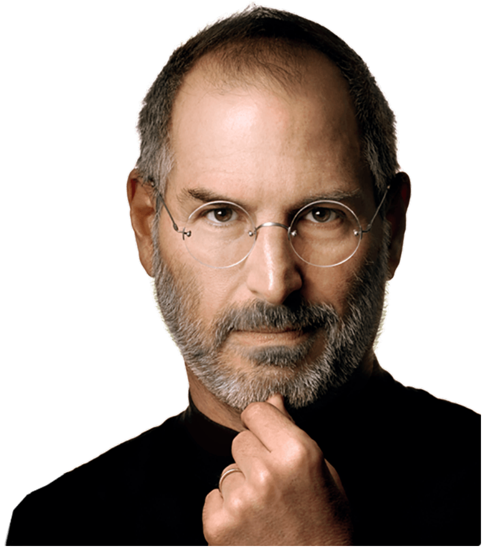
[8]

[9]

What About Being Disruptive?

- Sometimes good things are disruptive
  - Smallpox vaccine
  - Shannon's theory of communication
  - Internet
  - Brexit?
- But so are bad things
  - War
  - Famine
  - WannaCry
  - Brexit?
- Any fool can be disruptive
- It's important to be *good* too

Modern Geek Role Models



[10]

[11]

[12]

[13]

Modern Geek Role Models

- Visionary
- Ambitious
- Glib
- Arrogant
- Under indictment (sometimes)

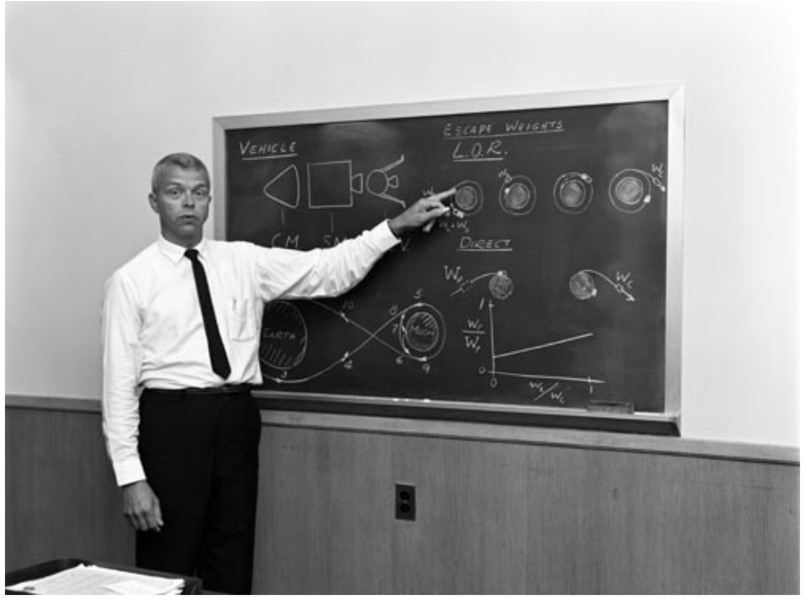
But is that all there is?

It's Not Enough to be Innovative or Disruptive

- We have to be good, too.
  - smart
  - educated
  - open-minded
  - diligent
- We need "proper paranoia" about how things can go wrong
  - knowing what or how to do
  - executing correctly
- Engineers are good at the knowing part
- What about execution?

Old-School Geek: John Houbolt

- Architect and advocate for Lunar Orbit Rendezvous mission concept for Apollo
- Overcame near-unanimous opposition by sheer skill and persistence



[14]

- *Houbolt was a brilliant engineering analyst and an energetic, persistent, and often eloquent advocate of the causes he espoused but he was not an overly shrewd behind-the-scenes player of institutional politics. Faced with the impasse of early 1961, his first instinct was simply to find more sound and logical retorts to the criticisms he had been hearing.* [Hansen 1999]
- *Thank you, John.* [von Braun 1969]

Old-School Geek: Katherine Johnson

- Navigation analyst for Mercury, Gemini, and Apollo programs

[15]

- *We needed to be assertive as women in those days – assertive and aggressive – and the degree to which we had to be that way depended on where you were. I had to be.* [Johnson 1999]
- *Her calculations proved as critical to the success of the Apollo Moon landing program and the start of the Space Shuttle program, as they did to those first steps on the country's journey into space.* [NASA 2016]

Old-School Geek: William Merrill

- Electrical Engineer, assessed fire risks associated with the World's Columbian Exposition in Chicago in 1893
- Founded [now] Underwriters Laboratories in 1894

[15]

- *Know by test, and state the facts.* [Merrill]
- *Our Mission: Working for a Safer World* [UL]

Old-School Geeks

- Educated
- Precise
- Rigid
- Principled
- Detail-Oriented
- Brutally Honest
- Safety-Conscious

Let’s pursue this safety angle

Some Notable Accidents

Freighter El Faro

[16]

- US cargo carrier lost off Bahamas with loss of 33 crew during Hurricane Joaquin in 2015
- NTSB investigation found that the probable cause of the sinking was poor judgement on the part of the captain

Freighter El Faro: Selected Findings

- Vessel encountered sustained winds causing list to starboard
- Water on exposed deck 2 pooled around and opened an unsecured hatch to cargo hold 3
- Vehicles in cargo hold 3 were not properly secured and came loose on the wet deck as the vessel rolled
- An adrift vehicle impacted and ruptured the 15 cm main firefighting inlet
- The resulting flooding overwhelmed the cargo hold 3 bilge pumps
- When the captained turned the ship to force a list to port, the main propulsion plant failed due to lack of lube oil pressure
- The oil level in the main engine sump was not maintained in accordance with the operations manual.
- There was no Damage Control Plan

Alaska Airlines 261

[17]

- US MD-83 jetliner crashed off California coast in 2000 with loss of 88 passengers and crew
- NTSB investigation found that the probable cause of the crash was loss of airplane pitch control due to failure of horizontal trim assembly jackscrew acme nut threads

Alaska 261: Selected Findings

- The acme nut threads on the airplane's horizontal stabilizer jackscrew assembly wore at an excessive rate
- There was no effective lubrication on the acme screw and nut interface at the time of the Alaska Airlines flight 261 accident.
- The excessive and accelerated wear of the accident jackscrew assembly acme nut threads was the result of insufficient lubrication, which was directly causal to the Alaska Airlines flight 261 accident.
- The on-wing end play check procedure [which assesses thread wear], as currently practiced, has not been validated and has low reliability.
- At the time of the Flight 261 accident, Alaska Airlines' maintenance program had widespread systemic deficiencies.

Observations

- In these (any many other) accidents, established standards of performance for routine operations were breached
- In both cases, proper attention would have prevented the accident
  - Securing hatches, properly lashing vehicles
  - Lubricating the jackscrew threads
- It likely never occurred to anyone that failing to secure a small hatch on an upper deck could sink a 240 m ship
- Nor that insufficient lubrication of a threaded rod with a low duty cycle could bring down a jetliner in stable cruise
- Attention to detail and standards of rigor are critical even when it’s not clear why
- The end of good procedure is good outcomes, not compliance
- This applies to the practice of engineering as well as flying airplanes or sailing ships

The Challenge and Risks of Complexity

- In the United States, we have a saying: Everyone talks about the weather but no one does anything about it
- Similarly, systems engineers talk about complexity, but I don’t think we’ve figured out how to deal with it
- There are effective ways to address complexity
- But first, let’s talk about complexity and how to be precise about it

Kolmogorov Complexity

- A. N. Kolmogorov was a Russian 20th Century mathematician who (in my opinion) formulated the simplest and yet most profound measure of complexity
- The Complexity C_S of some system S is the shortest description of S in some language
- Kolmogorov proved some interesting results about complexity
  - It depends only weakly on the choice of description language
  - There can be no algorithm for computing C_S—the best we can do is bound it from above
- An interesting corollary is that practical measures of system complexity reflect both system properties *and our understanding of the system*
  - "The shortest description we've been able to come up so far"

Example: Reed-Solomon (223, 255) Code

- The Reed-Solomon code is an error-correcting code for digital communication
- This code assigns to each 223-byte message a 225-byte codeword that contains the message and check bytes against error
- The codewords are transmitted and decoded to make messages
- If we didn't know the theory, we could nevertheless (in principle) describe the encoding and decoding algorithms by enumerating all (message, codeword) pairs
- Each message is 1784 (8 x 223) bits
- Each codeword is 1800 (8 x 255) bits
- There are 2^{1784} possible messages
- So the enumeration requires 2^{1784} x (1784 + 1800) bits
- That's a lot of bits....

Example: Reed-Solomon (223, 255) Code

- On the other hand….
- If we understand the mathematical abstraction (i.e., Galois Field) behind the algorithm, we can describe it much more concisely
- For example, in less than 2^{24} bits of Python
- The two descriptions describe the *exact same* encoder (system)
- Our estimates of algorithm complexity (i.e., shortest description length) differ wildly, not because the system changed, but because our understanding of it changed
- And, in particular, we conceived of the encoder using a powerful mathematical abstraction that unleashed 400 years of algebraic theory on the problem
- Finding the right abstraction is key to understanding

Questions?

sjenkins@jpl.nasa.gov

Image Credits

1. https://upload.wikimedia.org/wikipedia/commons/3/3f/Logo_de_Enron.svg
2. https://www.begintoinvest.com/enron-stock-chart/
3. https://www.autocar.co.uk/car-news/new-cars/delorean-dmc-12-go-back-production
4. http://content.time.com/time/specials/packages/article/0,28804,1913612_1913610_1913608,00.html
5. https://www.thehowtozone.com/at-work/startup-juicero-shutters-operations-and-seeks-a-buyer/
6. https://upload.wikimedia.org/wikipedia/commons/d/dc/MtGox.png
7. https://www.scmp.com/news/world/article/1849910/flypocalypse-now-hundreds-flights-delayed-cancelled-us-because-air
8. https://www.express.co.uk/pictures/sport/10980/Sutton-United-pitch-invader-Arsenal-sportgalleries

Image Credits

9. https://manchesterinklink.com/technology-thwart-wrong-way-drivers-expanded/
10. https://www.informationweek.com/mobile/medical-community-pays-tribute-to-steve-jobs-/d/d-id/1099834
11. https://in.reuters.com/article/us-tesla-ceo/proxy-firm-glass-lewis-opposes-teslas-massive-stock-award-to-musk-idINKBN1GH396
12. https://www.extremetech.com/computing/262647-linus-torvalds-says-intels-spectre-fix-complete-utter-garbage
13. https://www.vanityfair.com/news/2016/09/elizabeth-holmes-theranos-exclusive
14. https://crgis.ndc.nasa.gov/historic/File:1962-L-05849.jpg
15. https://commons.wikimedia.org/wiki/File:Katherine_Johnson_at_NASA,_in_1966.png

Image Credits

16. https://www.washingtonpost.com/news/post-nation/wp/2016/04/26/authorities-find-missing-data-recorder-for-sunken-cargo-ship-el-faro/?utm_term=.c2077c7219c1
17. https://en.wikipedia.org/wiki/Alaska_Airlines_Flight_261