stanza

<niallm@stanza.systems>
<niallm@sre.is>
@niallm
[long tail of stuff]

The Value of Reliability

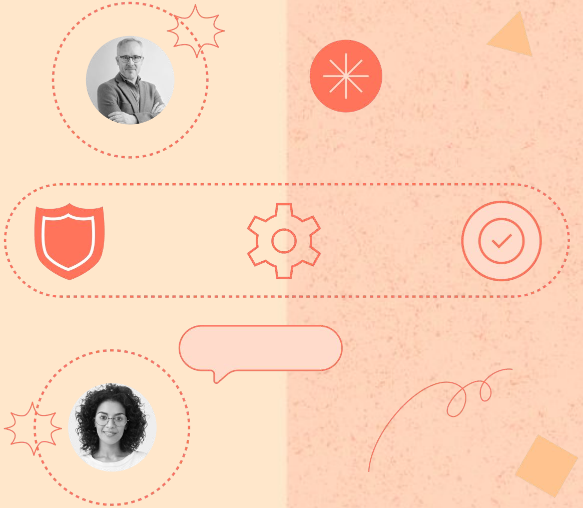
(Can it be valued? C.f. SRECon Keynote 2021)

How do we evaluate down-time?
What are the highest value parts of the stack?
How do we prioritise engineering effort?
How do we communicate the value effectively?

stanza

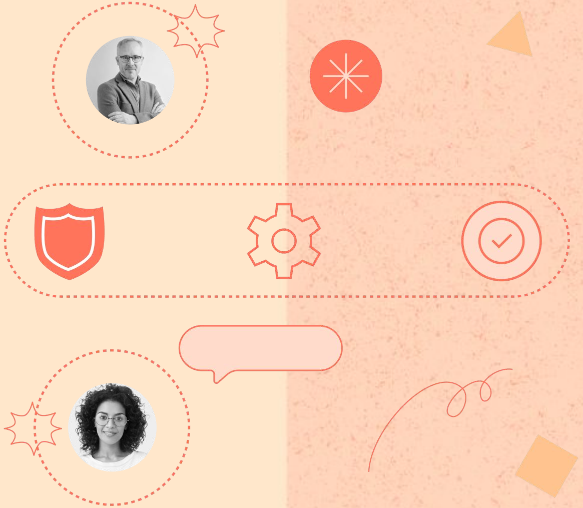
The Conventional Answer(s)

- "The Standard Model" (circa 2010-2020):
  - Assert "Reliability is the fundamental feature"
  - Assume e-commerce or otherwise financially involved website
  - Time spent has attributable $
  - Missing a request/second, assume total $ loss
    - Sometimes use averages or smoosh requests together to not waste too much time in calculation
    - == AuC loss
  - E.g. Amazon retail website, circa 2005



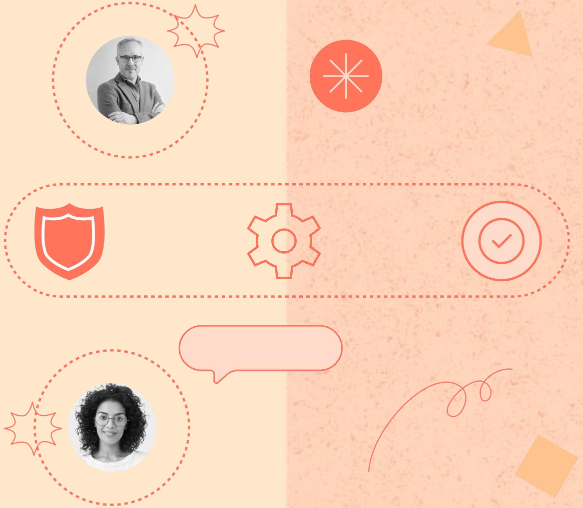
The Conventional Answer(s)

- Critiques of “The Standard Model”:
    - Not every action is equal; not every request matters the same
    - Is it an upper bound or lower bound?
    - Users do come back
        - (How many of them? On what schedule? Industry-wide dearth of info here. Maybe churn stats?)
        - Weird outage-seeking behaviours
    - Websites often don’t have attributable $/t
    - Horse-sized ducks and every-ten-years auction sites
    - Not everything is an e-commerce website
        - Pipelines!
        - ML!
    - User trust; intangibles
    - Doesn’t challenge mechanistic models



The Conventional Answer(s)

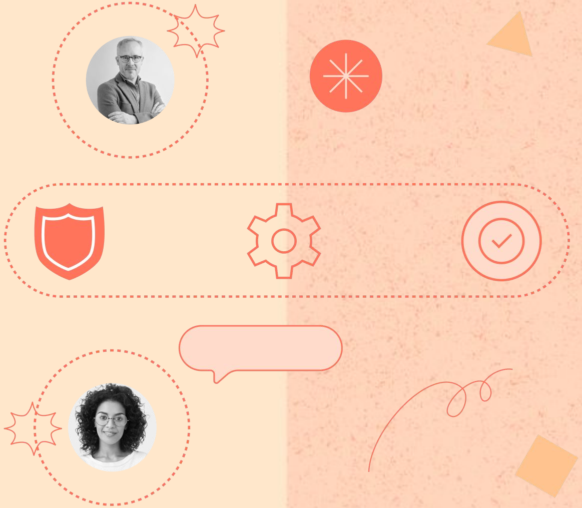
- “The Emerging Model” (2020+):
  - CUJ-focused
    - Not service-focused – user JTBD-focused
    - (Why do backend teams own reliability?)
  - User behaviour aware
    - Estimate user “backwash” based on previous trends and CUJ weightings
  - SLO-mature
    - They’re extremely important, and they have extremely important weaknesses
    - (Particularly one big loss vs lots of little ones)
  - Cost-aware as well as revenue-aware
- Primarily driven by AuC accuracy improvements



New Approaches

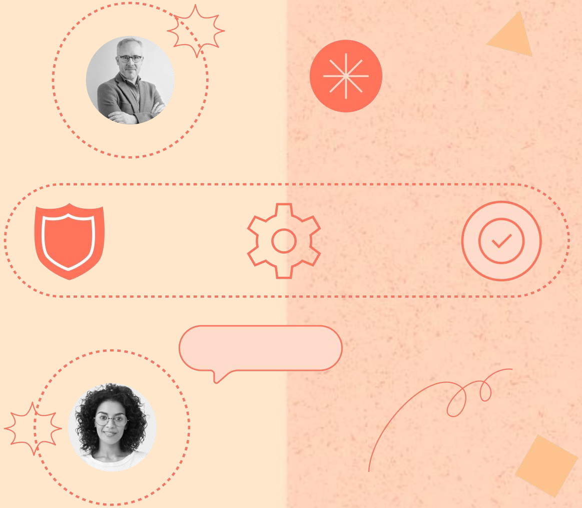
- Call-stack labelling
    - Akin to pprof
    - “Label” the call-stack according to revenue/cost etc
- Extending SLO reporting to encompass concentration measures
    - Analogous to

      ```
      round-trip min/avg/max/stddev =
      10.390/16.200/25.988/6.961 ms
      ```
    - What matters is whether the failures are concentrated in some way, and if so, how



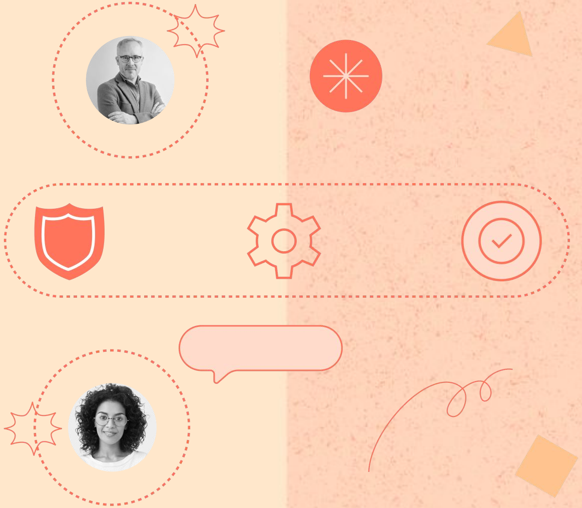
Prioritising engineering effort

- Usual approach
  - Beg for non-functional requirements eng time
  - Sigh and do it yourself
  - Blended stack-ranking of non-functional fixes, etc, based on previous impact, likelihood of reoffending (“total footprint”)
    - Most of which are intellectually dodgy, but better than random. Probably.



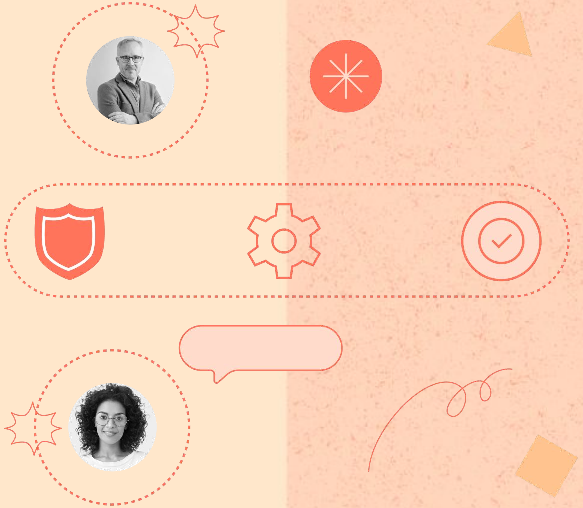
Prioritising engineering effort

- Different approach
  - A/B testing experiments paper from Microsoft – Ronny Kohavi et al
  - https://ai.stanford.edu/~ronnyk/ExPThinkWeek2009Public.pdf
  - “Our experience at Microsoft is no different: only about 1/3 of ideas improve the metrics they were designed to improve.”
  - “A team that simply launches 10 ideas without measuring their impact may have about 1/3 be good, 1/3 flat, and 1/3 negative (matching our current estimates on the ExP team).”



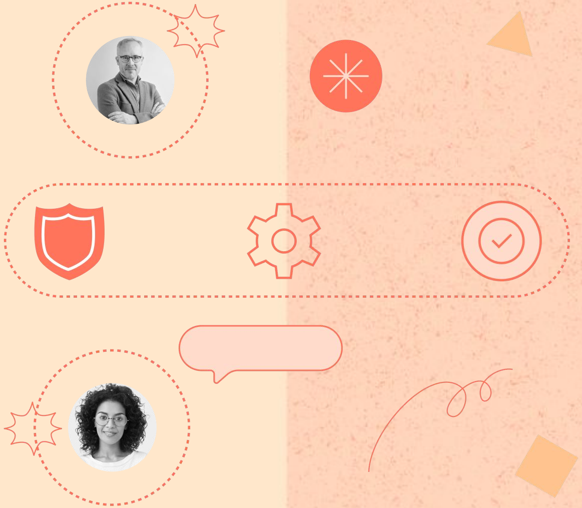
Prioritising engineering effort

- Different approach
    - Not saying product will *like* this argument, but:
        - if a randomly selected feature has ⅔ chance of being neutral/negative; and
        - your fix will prevent 'significant enough' loss; then
        - fix wins
    - In fact we could obviously extend this to a "rational stack-ranking scheme"
        - Though we run into the problem of valuing reliability
    - (Online experiments framework primarily benefits by allowing you to back out of bad things quickly)



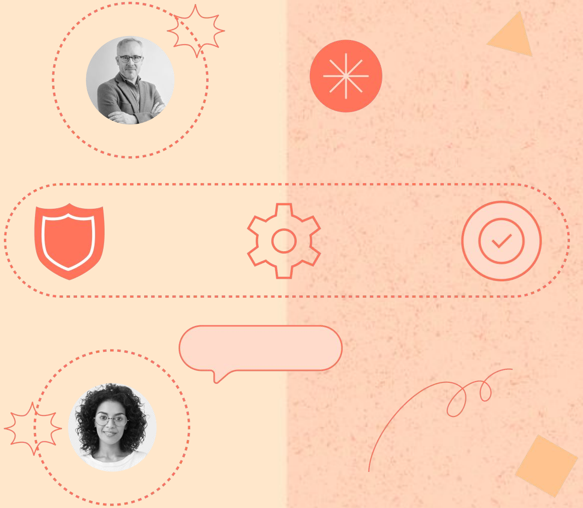
Communicating the value of reliability

- Old approach
  - “Reliability is the fundamental feature”
    - ZIRP argument
    - Less cynically, difference between established and newly created
- New approach (bad)
  - When there is no prospect of serious growth, then making the future customer experience better is worth less than extracting value from existing customers
  - No econorational argument for reliability other than chasing the minimum non-abandon rate
    - (the R…’s B……k of SLOs)
    - Twitter; cultural pressures to ignore the obviously bad things happening
- New approach (better)
  - ”The backlog is dark and full of terrors”



Communicating the value of reliability

- Environmental features
  - Execs model things by numbers and relationships, and they often need/have to make consequential decisions in the time they have walking between meeting rooms
  - Numerical arguments are cleanest, but only beneath a variable complexity threshold and in business domain/terms
  - If a decision can’t be made relatively cleanly on numbers, it brings in {power} relationships, and that can get complex
  - Net net: arguing for reliability, try to keep it numeric and connected to customer experience or COGS/CAC/etc
  - If you have to put it in relationship terms, try to build coalition of support
    - SRE horizontal approaches can be surprisingly helpful here



stanza

Parting thought

- Reliability is in retreat
  - Layoffs
  - Widespread belief growth is over
  - Various catastrophes across every segment of society
  - But those who remain are still behind the scenes, trying to keep it all going
  - Understanding and representing our value is an ongoing challenge
  - “Keeping things going” is being defunded
  - Users conditioned to expect/accept gradual disintegration - normalisation of deviance
  - The nature of (perceived) reliability is changing
  - “Reliability is a luxury good”
- How to help? *Give us your numbers*

Photo by Matheus Bardemaker on Unsplash