Who's this guy?

William Pietri / @williampietri
william@williampietri.com

- I imagine you have two obvious questions: who's this guy?
- My dad started developing software in the mid-1960s
- I started in the early 1980s.
- I got involved in the Agile movement circa 2000
- I've done ops, development, coaching, and management, including the CTO of a venture-backed startup
- so that's me

What's your point?

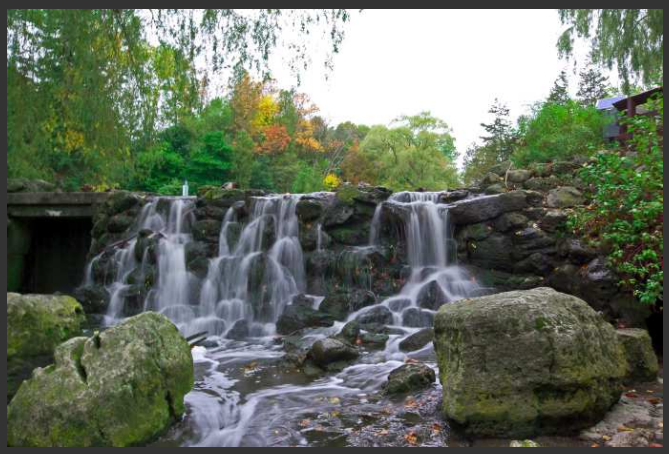
- I agree entirely with my fellow panelists on the power of Agile methods
- Unlike them, I'm asking you not to use them
- Early on, taught them to great effect
- In recent years, though, companies attempting to “do Agile” have caused a lot of suffering to little positive effect.
- Why? Because there's a mismatch between Agile values and company values.

segue: what would we do instead?

What if Agile didn't happen?

- Cast your mind back in time to the mid-90s
  - Dominant paradigm: waterfall
  - Normal iteration, 18-36 months
- If Agile hadn't come along, would we be doing that now?
- No. Nobody would put up with that for web sites and apps.
- We'd be releasing faster, doing a sort of mini-waterfall

We would be doing Mini-Waterfall



- What would that look like?
  - It would still be phasist in approach
  - It would still be top down and driven by requirements
  - It would still have strong roles (dev, QA, product, ops, mangement)
  - It would be shorter: 2-8 weeks
- This is what a lot of supposedly Agile teams are doing now. Why? Because that's what fits their company values.

Don't do Agile; do Mini-Waterfall

- Be honest
- Avoid the pain
- Choose to play the game you can win

How to tell? 10 points

The main determinant is culture, but culture is slippery.

Let's talk about some actual behaviors and implied values.

1: recent major pain

- expensive, embarrassing, or messy
- painful enough to cause a “moment of clarity”
- awful enough to power you through the work of major change
- provides clarity on what a process change should solve

2: autonomous teams

- broad business goals
- wide latitude in how to accomplish them
- teams picking work, not being assigned it

3: cross-functional teams sit together

- team: a group of people who win and lose together
- product + design + dev + qa + ops + ?

4: creativity or control

- bottom-up innovation or top-down specification?

5: performance or predictability

- both are valuable
- predictability is more comfortable for managers
- predictability is safer in low-trust environments

6: steer by customer value

- Do projects measure results?
- Are those numbers used to improve the product?

7: data vs HiPPOs

- What's the reaction when teams come back with data that a manager's pet project won't work?
- Does a team need permission to get data to evaluate a project's value?

8: reality over appearance

- Is there a significant history of underestimation and deadline slippage?
- is there a large and growing database of bugs?
- Is there a lot of technical debt?

9: maximum speed of iteration

- How frequently can you release?
- How frequently can you do user tests on new code?
- How frequently can stakeholders comment on new versions?

10: company-wide change

- not just engineers
- not just software
- organization-wide change to take advantage

Will you be changing how the whole company operates, or just software development teams?

In conclusion

I told you that Titanic needed more lifeboats

SAFETY SCANDAL REVEALED

Historic . . . Clarke's papers with Henry Wilde, right

TITANIC bosses were told to fit 50 per cent more lifeboats just days before the ship sank, it has emerged.

By JAMES MILLS

Flickr photo credits: oberazzi, williams_jt, manager_2000, thomwatson, tjdewey, tonygartshore

- Unless you have a burning desire to make major improvements, don't “do Agile”
- Instead, do mini-Waterfall