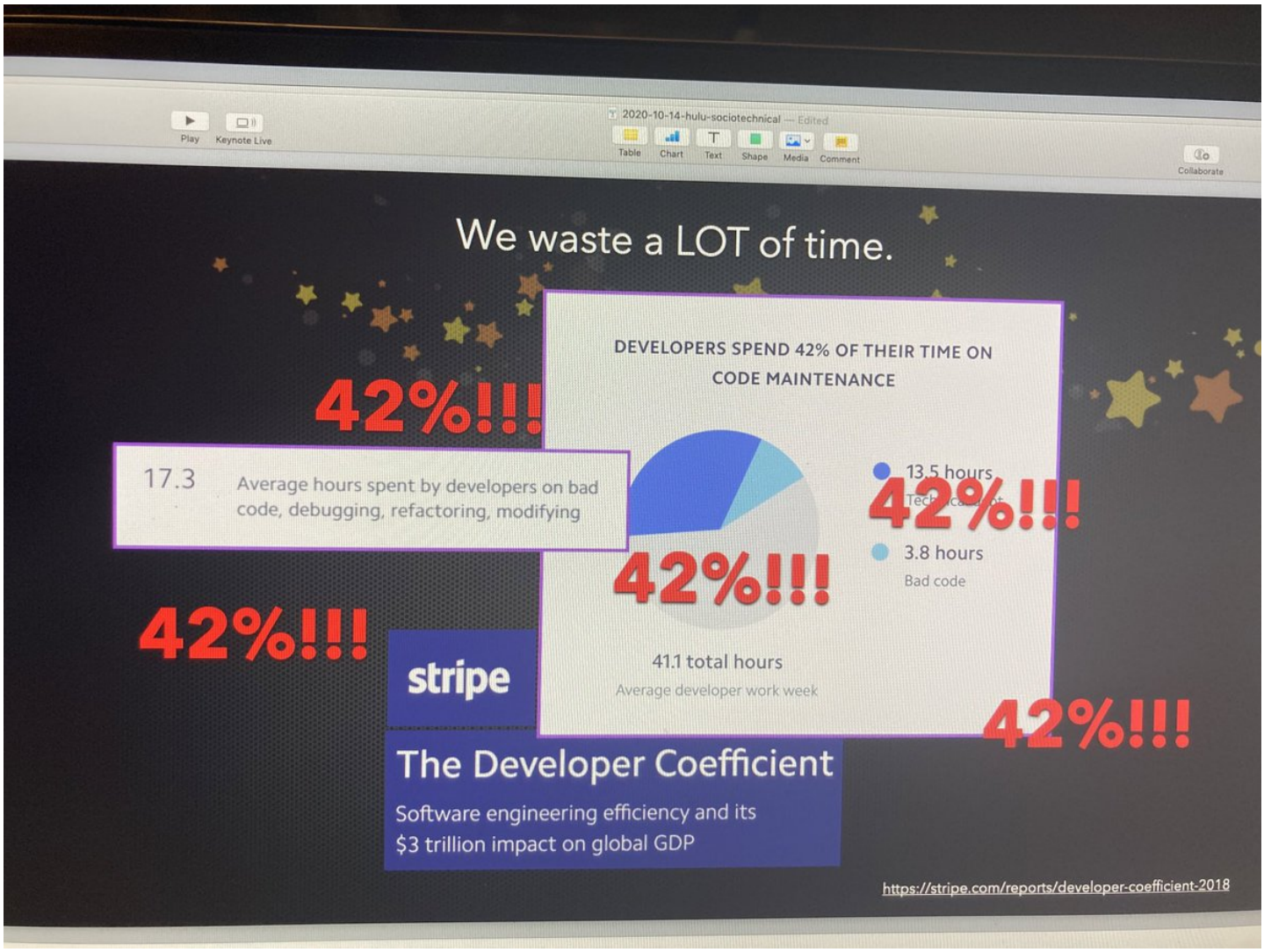
Twitter Thread by Charity Majors



Charity Majors
@mipsytipsy

Developer productivity, y'all. It is a three TRILLION dollar opportunity, per the stripe report.

Eng managers and directors, we have got to stop asking for "more headcount" and start treating this like the systems problem that it is. https://t.co/XJ0CkFdgiO



When people often have to spend weeks just to get a local development environment up, there is a lot to improve. \U0001f641

— Daniel Schildt (@autiomaa) December 20, 2020

If you are getting barely more than 50% productivity out of your very expensive engineers, I can pretty much guarantee you cannot hire your way out of this resourcing issue. ■

(the stripe report is here: https://t.co/SbyZDarWr9)

Say you've got a strategic initiative that 3 engineers to build and support it. Well, they're going to be swimming in the same muddy pipeline as everyone else at ~50%, so you're actually gotta source, hire and train 6, er make that 7 (gonna need another manager too now)...

...which actually understates the problem, because each person you add also adds friction and overhead to the system. Communication, coordination all get harder and processes get more complex and elaborate, etc.

So we could hire 7 people, or we could patch up our sociotechnical system to lose say only 25% productivity to tech debt, instead of 42%? ■

By my calculations, that would reclaim 3 engineers worth of capacity given a team of just 17-18 people.

And yes I KNOW these numbers are utter bullshit and people and teams don't work quite like that. I do.

But also, we are wasting a fuck load of people's valuable time and energy, all because we refuse to improve our pipeline and fulfill the real promise of Continuous Delivery.

It would be one thing if that 42% of wasted productivity was the fun stuff, like beach vacations and slacking off when the boss is out.

But it's not. It's the brutal grind of detangling stupid problems that should never have existed, repeatedly, with your eyes taped closed.

It's the hours your teams spend debugging the same problem over and over, because your tools aren't modern enough to let you verify it is the same problem til the bitter end. ■

It's getting woken up in the middle of the night. It's flappy pages. For deprecated code.

It is the shit that grinds people down and makes them say they hate graphs, and computers, and life.

And let us be clear: this is a fucking management problem. The fact that developer productivity is not high on every VP, director, and manager 2021 goals is a management failure.

I know it's not easy. Systems problems never are. But there's a staggering amount of upside to every inch of improvement you make, and a steep cost to the status quo. Your competitive existence is at stake.

You've seen this graph, right? (The yellow bubble is the "elite" teams.)



Those teams aren't "better" at engineering. They swim in better pipelines. They write the code -> it gets deployed, in the tightest, shortest, most automatic loop possible.

Then they tend to that shit, and don't let it drift.

I've been yelling at engineers about this for years, but maybe I've been doing it wrong. Engineers by and large already know this, are already on board, are trying to make a case.

I am sure there are exceptions, but increasingly I see this as a manager/director/VP eng problem.

I also think this is a place where non technical eng managers, or even managers who are told to focus exclusively on people coaching, can do a great deal of damage despite excellent intentions. ■

Hear me out. I don't have it out for y'all; I've known some stellar nontech EMs.

But I admit it's a hard call to make and defend. You're asking the leadership org to accept a slowdown on their roadmap and less stability upfront for an indeterminate length of time, in order to implement something that runs absolutely counter to our instincts as humans!

The "speed = safety" aspect of shipping software is counterintuitive at best. Yes, it's true, but it doesn't *feel* true.

That's a hard line for an EM to take with upper management (none of whom may have read or care about DORA reports or Accelerate).

Hard enough for an EM with an engineering background who knows for god damn certain that it's true because she's been there herself.

EMs who don't have that personal grounded knowledge? Nah. I can't imagine going out on this limb based on hearsay. I sure wouldn't.

These are complex sociotechnical systems. To understand, unpack, and improve them, you need a dual fluency -- experience in tech, experience with people, and curiosity about the intersection.

Anyone who walls you off with people OR tech isn't acting in your interest long term.

Annnyway. I gotta stop rambling and go to bed.

Moral of the story? (Morals?)

If you are an engineer, set yourself a goal that 2020 is the last year you leak time working for a place without observability and CD. Find one or help your job become one, but don't get left behind.

If you are an eng VP or CTO who knows what I'm talking about, begin working o11y and CD into your 2021 strategy. In a month or two, write a medium post about how and why you are doing this and use it as recruiting catnip.

(If you *don't* know...catch up I guess)

If you are an eng manager or director....oh, kitten, listen to me.

The power to change the lives of engineers and teams around the world *dramatically* for the better rests primarily with you.

You span worlds. You have the moral authority of the doer and the influence.

You probably don't push your own agenda much. That's good! ■

But what is the point of power and influence if you don't haul it out when a mountain needs moving?

This fucking mountain, lol. Burn it to the grond. ■

(p.s. if wondering what o11y and CD have to do with each other... It can be prohibitively hard to reach continuous deployment without real time visibility. why even care about o11y's real-time feedback loop if your code rots for days or weeks before it rolls out to users?)