Twitter Thread by Arien Malec



Arien Malec
@amalec



1/

Thinking about this tweetstorm, one of the issues I've run into as an engineering leader is what to call the software engineering stuff that's "agile" given that the Agile Community(tm) has killed the brand.

> I might do an \u201cagile\u201d tweet storm to the effect that all the attention is to the least leveraged portions of the value stream. Interest?
>
> — Arien Malec (@amalec) October 26, 2019

2/

And by & large, I've taken to call it "DevOps", because the DevOps community have taken up much of the mantle @KentBeck & the XP community started with. & Kent has independently focused on safe small changes deployed to production. Which is DevOps.

3/

Much of the art here is making changes safe enough to deploy to production continuously. And to do that, we need to design incrementally, test obsessively, take architecture seriously so we decompose dependencies. & we need to automate everything & do it all the time.

4/

It turns out that this is what Kent & @RonJeffries @GeePawHill & many other folks have been nattering on about & being broadly misunderstood. @KentBeck has some brilliant essays (scattered across FB & his site alas) & @GeePawHill has amazing twitter threads on the topic

5/

When you look at *what it takes* to get to the DORA measures that @nicolefv & team write about in Accelerate, the input metrics for the DORA outputs, it's making small changes safe.

6/

As an engineering leader, I provide training, tools, mentorship, leadership development, vision, etc. to help people learn the skills needed to achieve those output metrics. And most of those skills are what @GeePawHill might call the skills of making.

7/

Unfortunately many of those skills are deeply counterintuitive & much of the work is as much unlearning as learning. For example, there's an implicit definition of work as writing new code, or even writing code.

8/

Because that's what engineers love to do, and because there are emotional and sometimes financial incentives to make customer visible functionality, we need to overcorrect sometimes on focusing on the tools of making.

9/

Providing visibility & reward for the people who build the CI/CD tooling or build a deployment pipeline that automates acceptance testing, or figure out how to do AppMesh with Terraform as a module or automates linters & code coverage tools in the pipelines.

10/

Great teams end up spending most of their time building user facing functionality because they build the tools of making and sweat automation, IoT, design, architecture, code quality & test automation. Less successful teams try to write lots of code & get stuck.

11/

Accelerating teams towards that point where they've incorporated the habits of investing in the tools of making & designing architectures that decompose change into very small safe increments is a key area of software engineering management.

12/

Rallying these changes under the flag of DevOps has been the most successful way I've seen to describe these habits.

END