Twitter Thread by Mario “I have a model for that” Platt

Mario “I have a model for that” Platt
@madplatt

2 different frames or metaphors for #CyberSecurity

Security as a Quality Management issue thus a problem of robustness

Security as a Safety issue thus a problem of resilience

They’re fundamentally different, may even be at odds but how ?

■

A Quality Management metaphor leads us down elements of testing, predictability and reliability in that we aim to be effective at dealing with predicted threats which represent a finite configuration of what our industries expect. We aim for “maintaining process integrity inside

Of known design, training and procedural parameters” (Dekker) so I prepare for a number of scenarios, ways I can foresee and threat model for and (not just operationally, but also reqs gathering, testing, training,sdlc and even threat intel) increasing the robustness of my system

All well and good, but from Complexity science we know that “robust constraints tend to fail catastrophically when design conditions are exceeded” @snowded
But this is also a cautionary tale. If you don’t have the capacity to deal with expected adverse conditions,

You have no business thinking or prioritising adaptive capacity because the “knows” can get you down and keep you there. So until you get to the point where your robustness is sound, this is the metaphor that is most appropriate and which can be argued for economically

If and when the robustness of your system is high, meaning you have appropriate mitigations in place to the outcomes of the threat models you build for your apps and platform, then your risk profile starts to change.

The very measure you've put in place, which got you this far to achieve a "secure system" by industry and accepted standards, may now be exact things which will contribute to its downfall because "in safe systems the very processes that normally guarantee safety and generate

Org success can also be responsible for org demise" (Dekker) This means that the challenges you're now likely to faced relate to the relationships & interconnections of your sociotechnical system (scarcity competition, patterns of info to decision makers, incrementalist nature

Of decisions taken over time which can cause drifts into failure, drifts into setting conditions for data breaches.
When you get here, a metaphor of Security as Safety is likely more appropriate. You've now dealt with the robustness of components and put in the measures which are

Expected to make you withstand the scenarios you can envision on your threat model. So now a focus on resilience becomes beneficial as you're building systems which "are effective at meeting threats that represent infinite reconfigurations of - or ones that may lie entirely

Beyond - what the industry could anticipate" and which are "capable of maintaining process integrity well outside the design base or outside training or procedural provisions"

You're now building for resilience and sustained adaptive capacity

"But can I do both?" You ask

The building of robustness benefits from processes of analysis (breaking things out to smaller parts and secure them to build them back up) while resilience benefits from processes of synthesis (role on the wider system and function within it)

So while you perform both types of assessments, a focus on resilience where there's limited robustness is of little value. A small punch can knock you down and keep you there. So I would advocate building robustness first as the focus of transition/transformation, potentially

Being attentive to affordances and constraints to identify opportunities for improvement.

Remember that sources of resilience in orgs are in our people. They're the ones dealing with the variability our systems and automation can't handle, so applying principles of learning

From incidents, and understanding what our teams were surprised by, and namely any "fundamental surprises" (where their models the system were or what's possible were challenged or took apart) are learning opportunities that should never go to waste

@threadreaderapp unroll