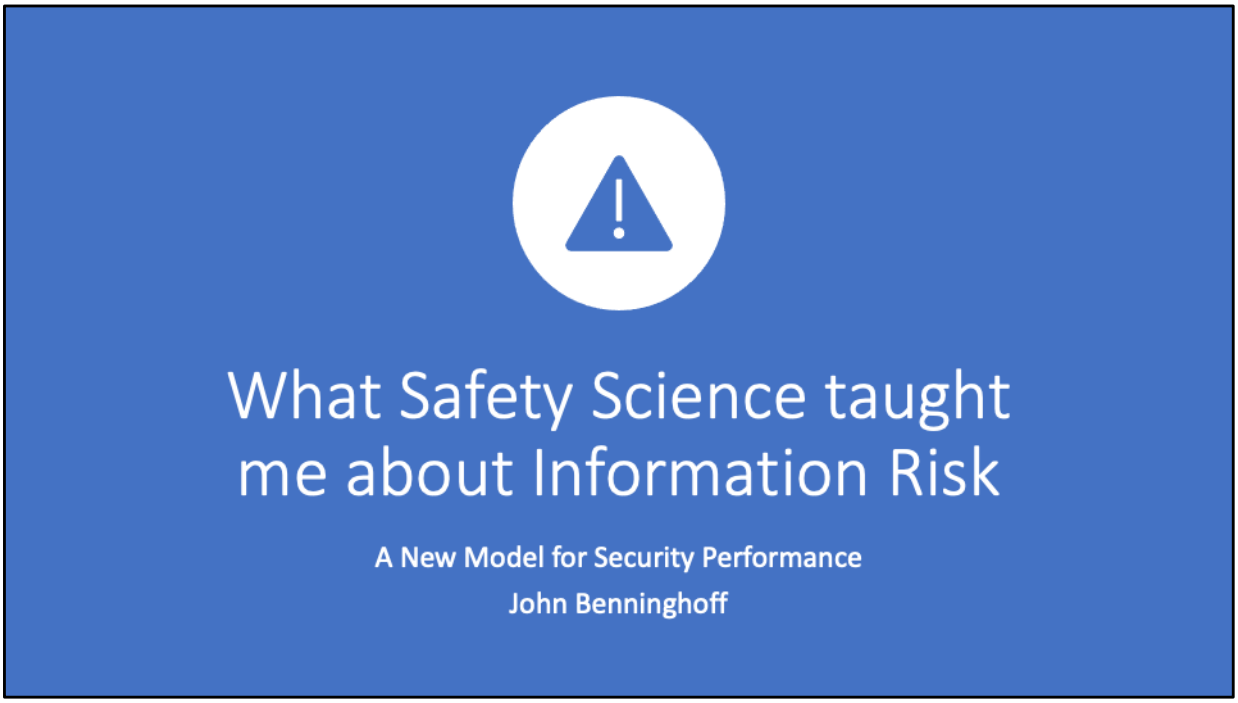


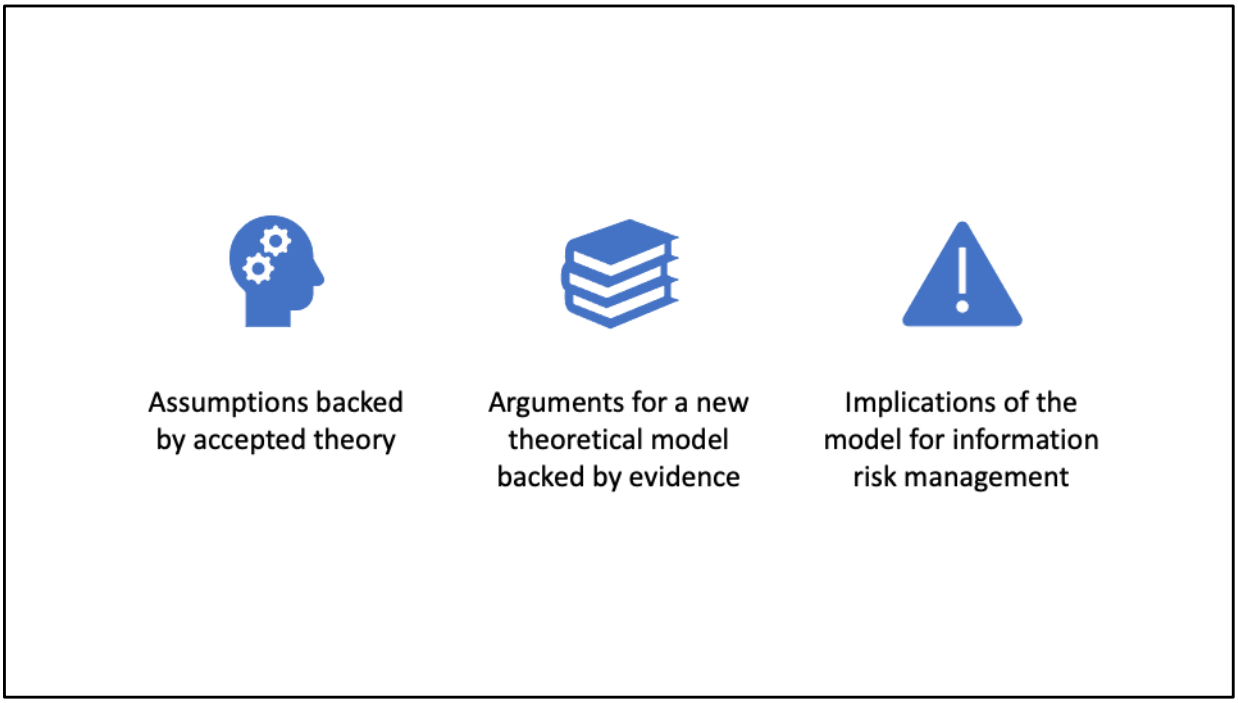
When I set out to write this talk, I had an idea in mind, but I wasn't expecting to propose a new model of security performance. Hopefully this will reflect the intersection of risk and security.
This is an updated version of a talk I gave in Aug 2021; to prepare for this, I consulted an expert… "past me!" I found him both insightful and sometimes wrong.
Notes on references included in slides, will be posted!

This is a continuation of a journey that began around the time of the first SIRACon in 2012 (at Secure360)…summary of talk, comparing 'protection' (secret service, ECSP, public assassination attempts, predicting violence) and 'safety' (aviation) approaches to risk management.

References:
https://vimeo.com/44519848

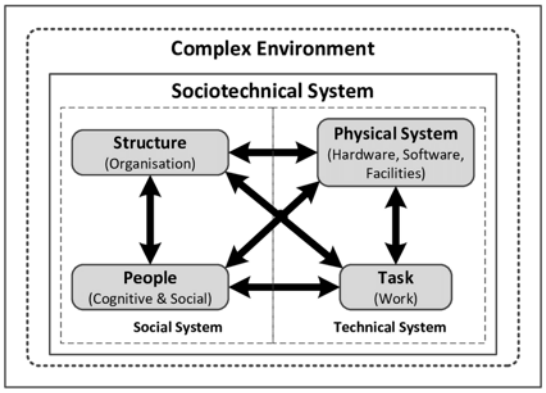
2018: TCD. A view of security shaped by my 2+ years of safety science research while pursuing my MSc in Psychology, Managing Risk and Systems Change, and ongoing. Shaped my views of safety, risk, security and performance.

References:
https://psychology.tcd.ie/postgraduate/msc-riskandchange/
Image: https://commons.wikimedia.org/wiki/File:Trinity_college_library.jpg



Talk outline; accepted theory (within safety science), evidence from security for a new model, what that means for us as security practitioners. I was surprised by some of the implications of the theory!

Assumption 1: organizations are sociotechnical systems



Sociotechnical Systems Theory is a generally accepted part of modern safety science, and foundational to my master's program; so foundational (especially in the UK/IE) that we didn't really study its origin; According to Wikipedia, "The term sociotechnical systems was coined by Eric Trist, Ken Bamforth and Fred Emery, in the World War II era, based on their work with workers in English coal mines at the Tavistock Institute in London."

References:
https://en.wikipedia.org/wiki/Sociotechnical_system
Image:
https://www.researchgate.net/publication/306242078_Assessing_the_impact_of_new_technology_on_complex_sociotechnical_systems

Assumption 2: all failures are systems failures


How Complex Systems Fail

(Being a Short Treatise on the Nature of Failure; How Failure is Evaluated; How Failure is Attributed to Proximate Cause; and the Resulting New Understanding of Patient Safety)

Richard I. Cook, MD
Cognitive Technologies Laboratory
University of Chicago

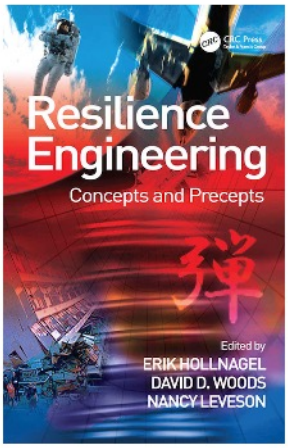
1) Complex systems are intrinsically hazardous systems.

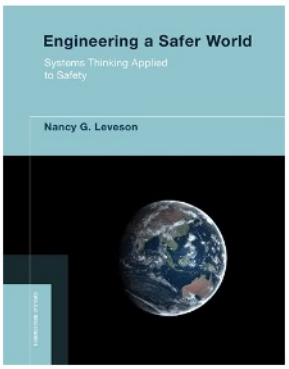
2) Complex systems are heavily and successfully defended against failure.

3) Catastrophe requires multiple failures – single point failures are not enough.

4) Complex systems contain changing mixtures of failures latent within them.

5) Complex systems run in degraded mode.





How complex systems fail (not directly stated in the paper, but implied), Resilience Engineering, Leveson, Others. ”Root Cause”, “Component Failure” vs System Failures, unexpected component interactions.

References:

Cook, R. I. (1998). *How complex systems fail*. Cognitive Technologies Laboratory, University of Chicago. https://www.researchgate.net/publication/228797158_How_complex_systems_fail

Leveson, N. (2011). *Engineering a safer world : systems thinking applied to safety*. MIT Press. https://mitpress.mit.edu/books/engineering-safer-world

Hollnagel, E., Woods, D. D., & Leveson, N. (2006). *Resilience engineering : concepts and precepts*. Ashgate.

Argument 1: resilience improves through performance

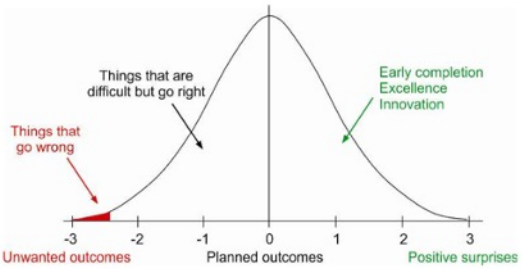


Figure 9: Event probability and safety focus

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,c,d]	0-15%[b,c,d]	46-60%

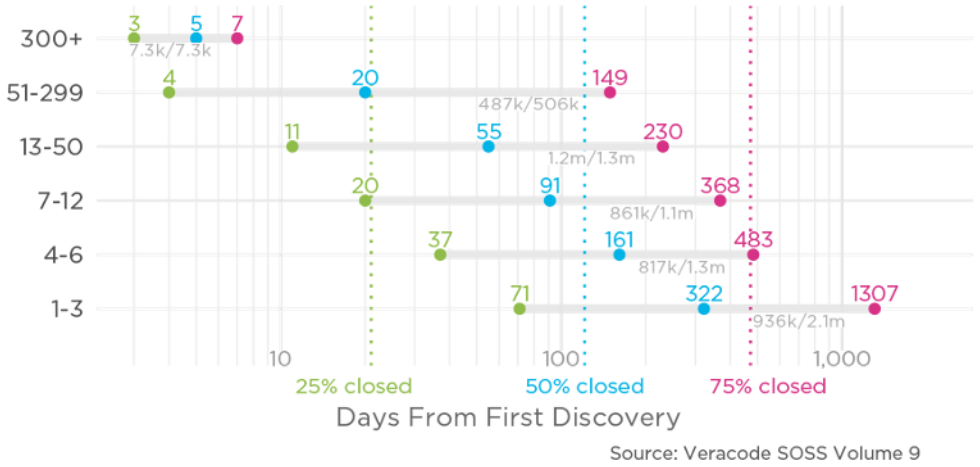
Resilient systems fail less often and recover faster
Safety-II; we can't have a science of non-events, and must instead study the full range of performance, 'working safely (or securely')
We don't care about how many breaches, only that the system resists threats and recovers faster (since we don't control the environment), "how do we defend better?"
Thus, we need to improve the security performance of the system (we also don't care about component performance, only performance of the system as a whole; stopping a phishing email from installing malware vs stopping a person from clicking the link
Shifts from managing risk to managing performance
Impact of Forsgren, Google DORA research – shows how performance in productivity, reliability, availability and security all move *together.* I've studied the research and conducted my own; the best teams do *everything* right (there is no trade-off)!

References:
Hollnagel, E. (2014). Is safety a subject for science? Safety Science, 67, 21-24. https://doi.org/10.1016/j.ssci.2013.07.025
Hollnagel, E., Wears, R. L., & Braithwaite, J. (2015). From Safety-I to Safety-II: a white paper. The resilient health care net: published simultaneously by the University of

Southern Denmark, University of Florida, USA, Macquarie University, Australia.
Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate : the science behind DevOps : building and scaling high performing technology organizations (First edition. ed.). IT Revolution.
Forsgren, N., Smith, D., Humble, J., & Frazelle, J. (2019). 2019 Accelerate State of DevOps Report. DORA & Google Cloud. https://research.google/pubs/pub48455/

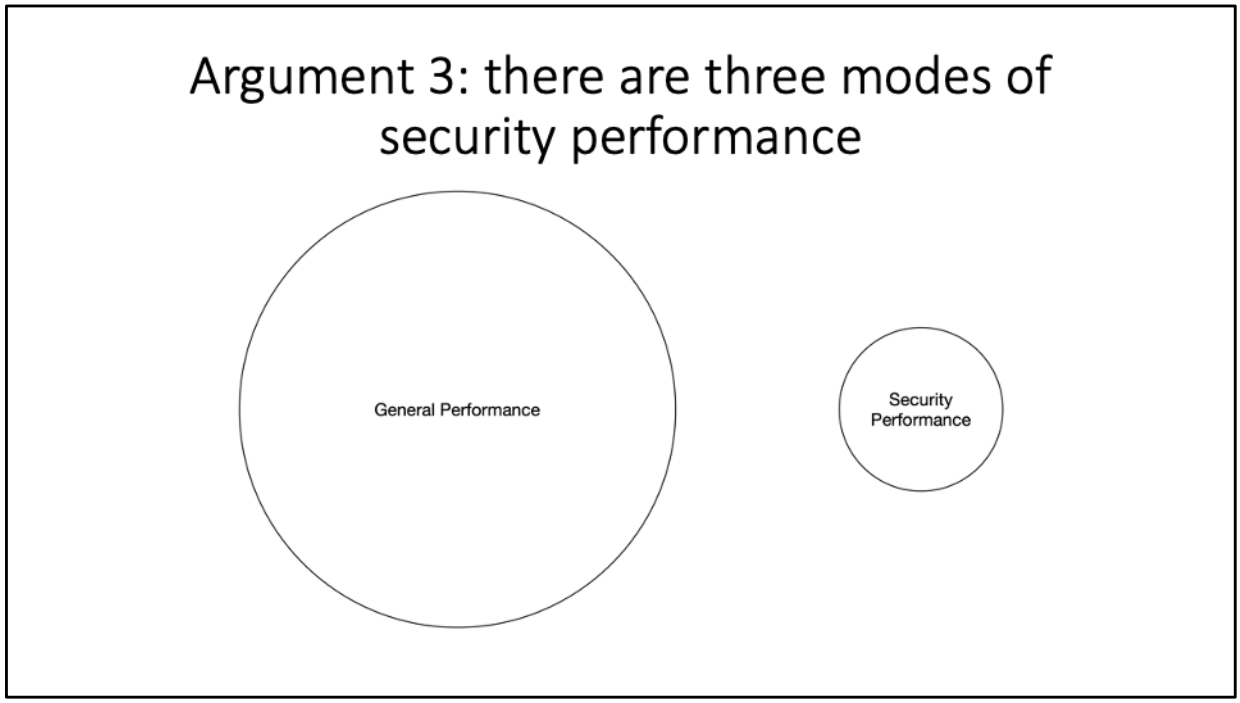
Images from Hollnegal, et al. (2015) and Forsgren, et al. (2019)

Argument 2: security performance is correlated with general performance



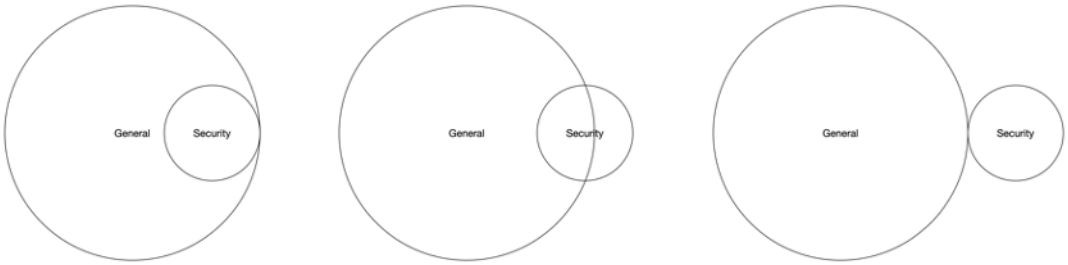
The “wow!” diagram. Veracode SOSS graphic shows how frequent testing (general performance) is correlated with security performance (faster closure)
Thinking about the relationship between security and performance, I realized that there were … (3 modes)

References:
Veracode. (2019). State of Software Security Volume 9. https://www.veracode.com/sites/default/files/pdf/resources/ipapers/state-of-software-security-volume-9/index.html
Veracode. (2019). State of Software Security Volume 10. https://info.veracode.com/report-state-of-software-security-volume-10.html
Image: Veracode (2019)

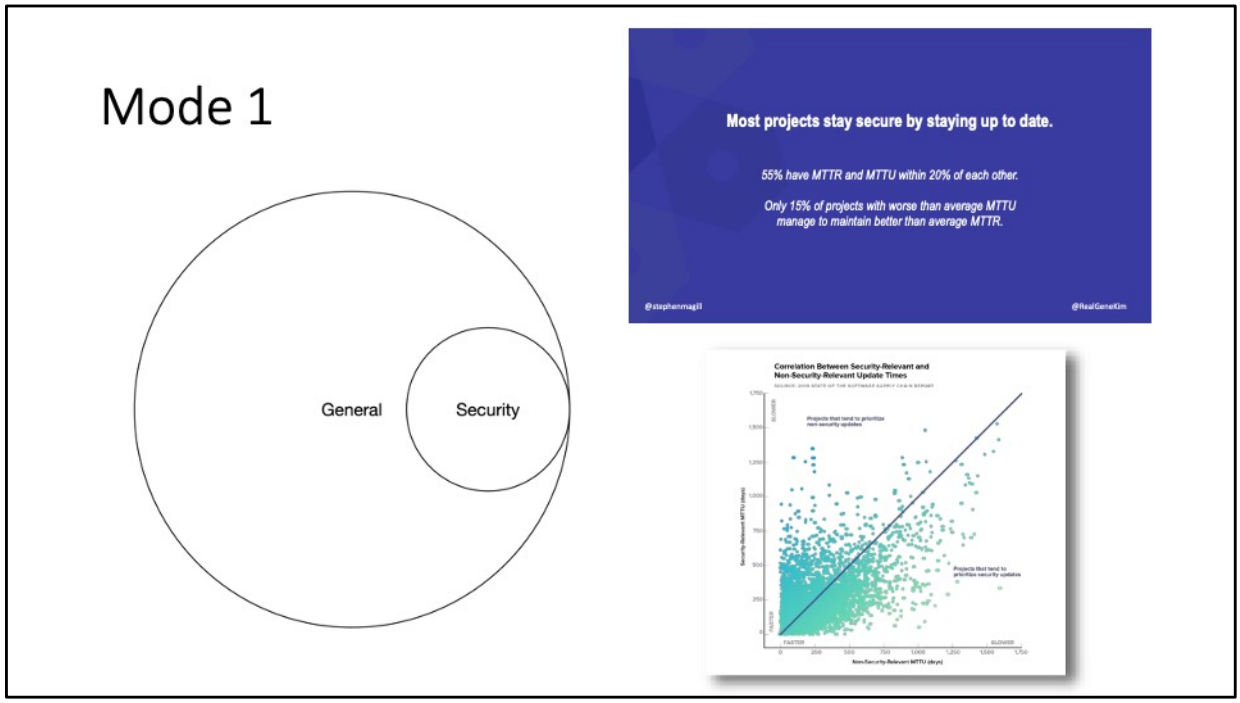


The model is an attempt to explain the relationship between general performance on technology activities, and provide insights to improving performance (and thus working securely)
The size of the circles are deliberate; security activities are small by comparison to everything else
How do we fit in with the larger picture? We are a small part of a larger team.

Argument 3: there are three modes of security performance



Mode 1: Security is entirely contained within general performance
Mode 2: Security is partly outside of general performance
Mode 3: Security is entirely outside of general performance



This is a model, which is by nature an oversimplification, but is helpful in understanding. Mode 1 – contained within security.

Gene Kim work with Stephen Magill: Java dependencies in Maven ecosystem, security is achieved through staying up to date – not a separate or security specific activity!

Compare to Ben & Jay's talk yesterday on Vulnerabilities; excellent talk, I agree with everything they said, but reject the premise: that our job is to get better at fixing the vulns as a separate task

2021 Security Outcomes (Cyentia/Cisco): the biggest factor in reported security program success: proactive refresh of technology.

As does Jay Jacobs' work on the correlation between SSL/TLS vulnerabilities (which reflected **maintenance**) and likelihood of breach.

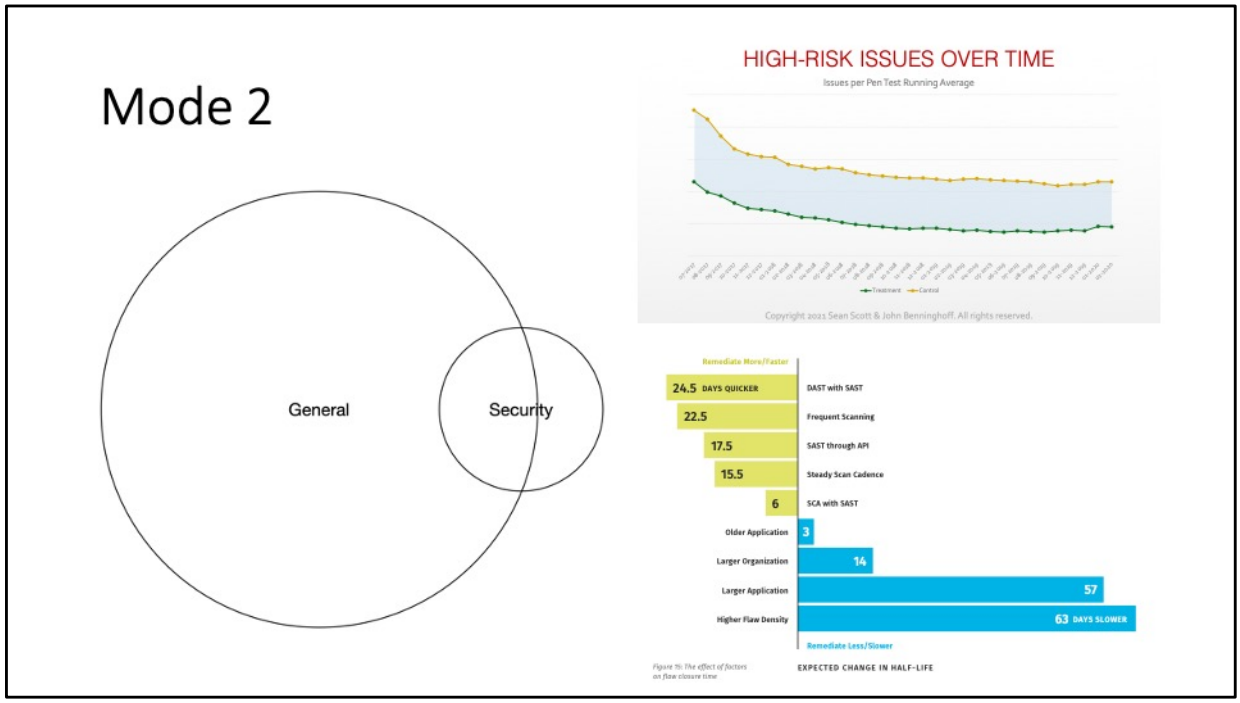
References:

The 2021 Security Outcomes Study. (2020). Cisco, YouGov, Cyentia. https://www.cisco.com/c/en/us/products/security/security-outcomes-study.html

Magill, S., & Kim, G. (2019). A data-driven look at practices behind exemplar open source projects. https://www.youtube.com/watch?v=YoWkuFzEYFs

sonatype, galois, & IT Revolution. (2019). 2019 State of the Software Supply

Chain. https://www.sonatype.com/en-us/2019ssc
Images: Magill, S., & Kim, G. (2019)



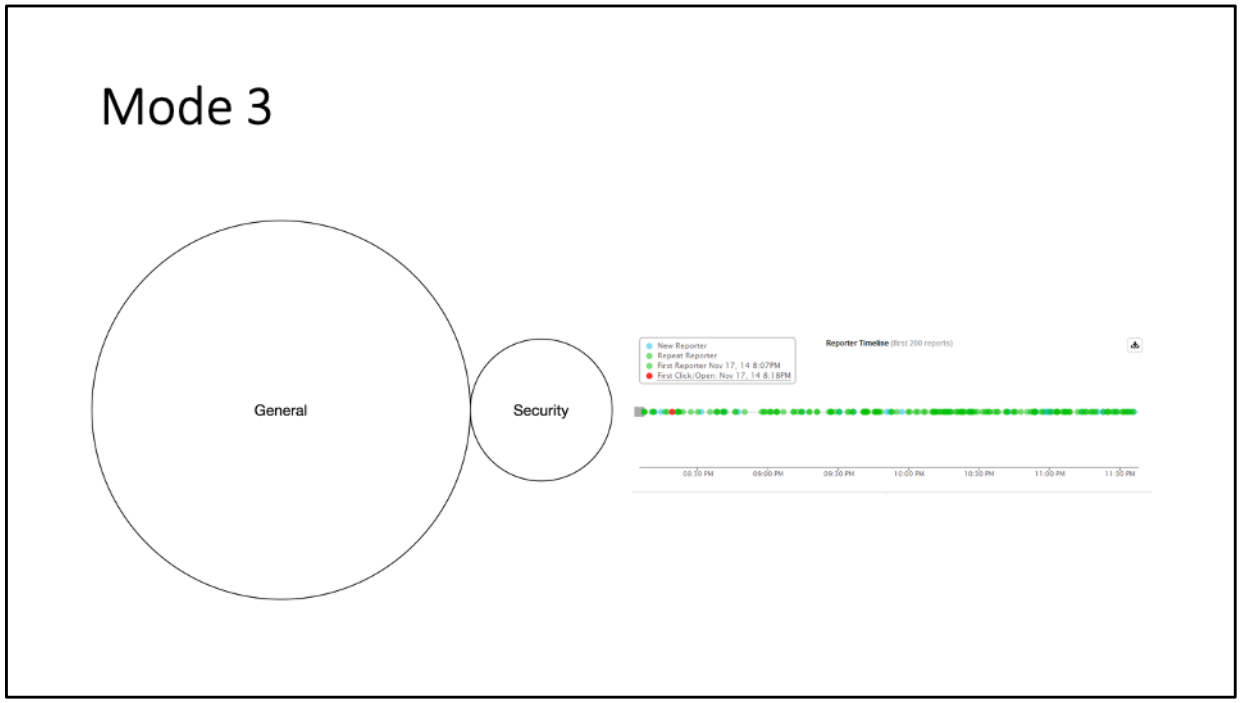
In our AppSec program, we found that teams we worked with performed better on pen-tests than teams we did not; this is an example where security performance overlaps but is not contained within general performance (writing software vs writing secure software) – spoke about this at Secure360 last year (2021)! “Does our AppSec program work?” 50% reduction in new high pen-test findings, reduction in fix time. The Cyentia/Veracode findings not only reduce time to fix, they also *reduce the number of vulnerabilities*. Our practices made breaking builds on “high” static code analysis security testing the norm.

References:

Scott, S. (2021). Secure Coding in Large Enterprises: Does Application Security Coaching, Training, and Consulting Increase a Development Team’s Ability to Deliver Secure Code. University of Missouri-St Louis.

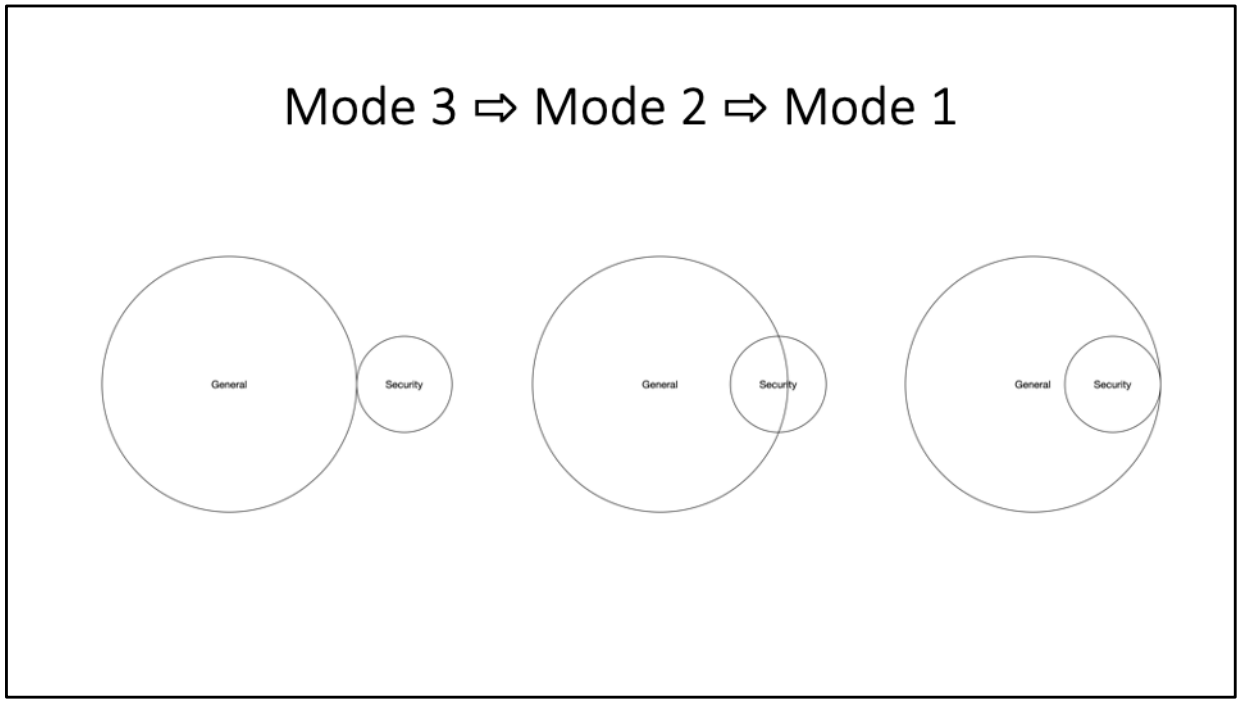
Veracode. (2020). State of Software Security Volume 11. https://info.veracode.com/report-state-of-software-security-volume-11.html

Images: Scott (2021) (top), Veracode (2020) (bottom)



PhishMe/Cofense presentation from Secure360 2015: phishing is (or was) a novel attack that our general performance is not equipped to deal with. My experience at a large Canadian bank in the early days of phishing: response team was busy every night taking down phishing domains until we hired a firm that had quickly stood up an outsourced takedown service.
System performance exceeding individual performance.

Image: Cofense, Secure360 2015

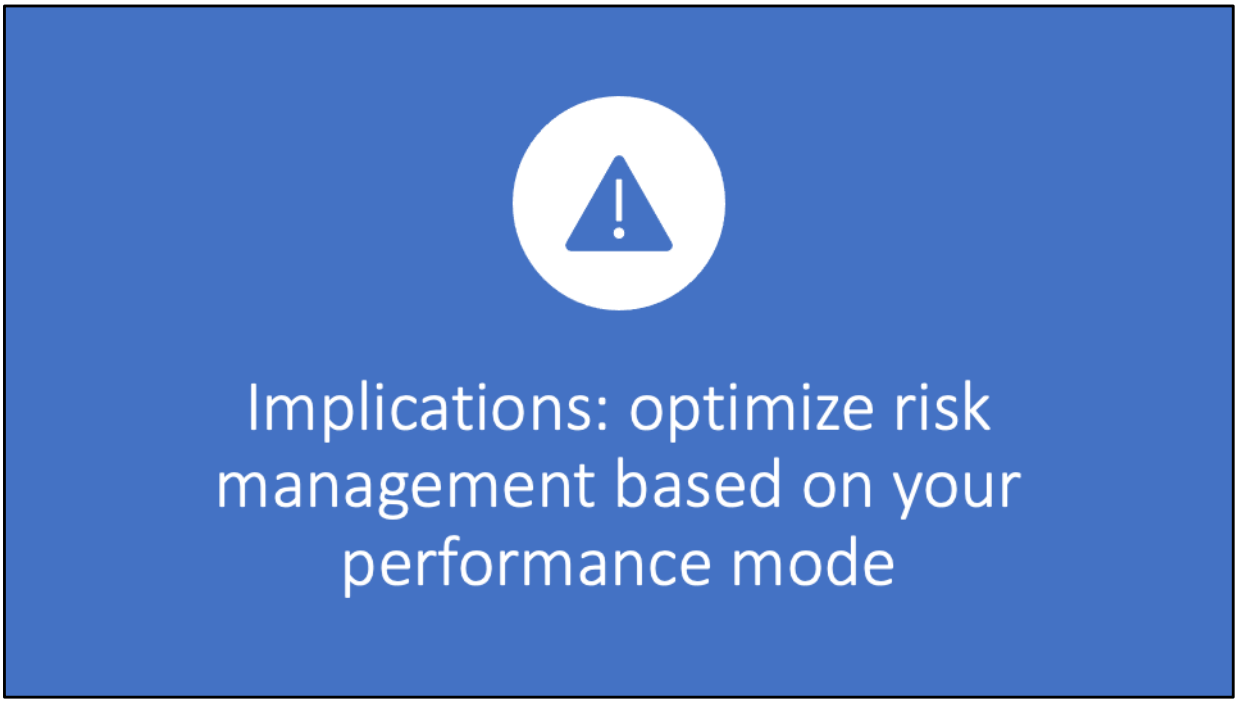


Over time, performance transitions from mode 3 to mode 2 to mode 1 (really, general performance grows and absorbs security)

Transition of vulnerability management to automated upgrades over time

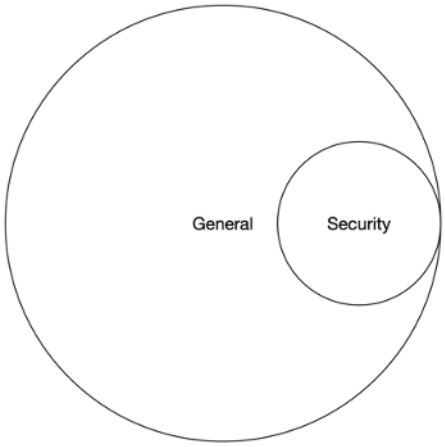
Conversation with Doug Crockford, “father of json”, 2008, “I don’t believe in Security as a separate profession”: AppSec was entirely new when L0pht testified before congress in 1998 (mode 3) and has shifted to mode 2, will shift to mode 1 as Doug predicted.

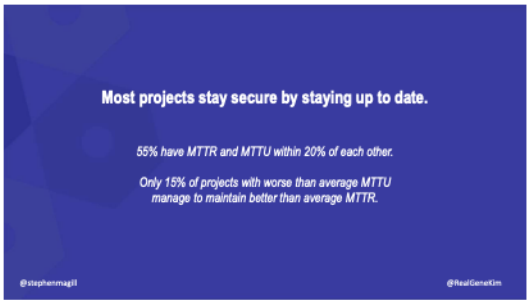
Doug: our job as engineers is to make security the norm, part of our work.



Understanding which mode you are operating in informs you on how to improve performance, and also guides risk analysis

Mode 1: improve general performance

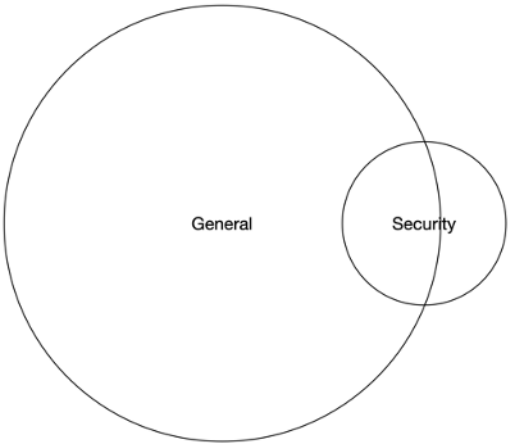


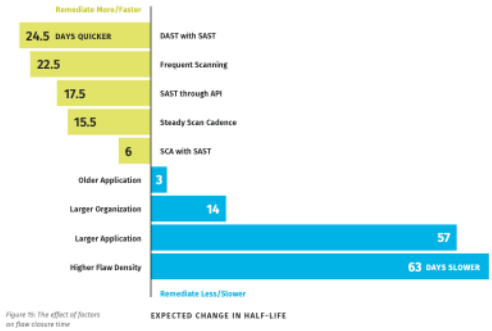


Example: focus on improving installing all updates, not just security updates (get rid of your vuln management program – SPC, web hosting examples)
Security team supports the system, “be a cheerleader”, support the CIO
Analysis includes costs and benefits of improving performance, including inherent risk reduction
Challenge to ending VM: “what about solarwinds?”

Image: Magill, S., & Kim, G. (2019)

Mode 2: add security enhancements to general performance

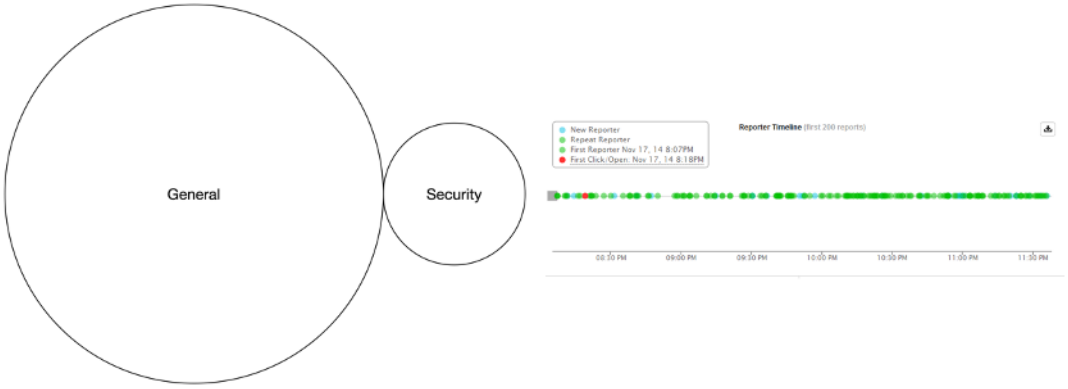




Example: train developers on how to prioritize, test, and fix security bugs (enhances their bug management performance)
Security team improves the system by adding security expertise
Analysis includes costs and benefits of improving performance, including risk reduction (don't measure security risk in isolation)

Image: Veracode (2020)

Mode 3: create security-specific systems

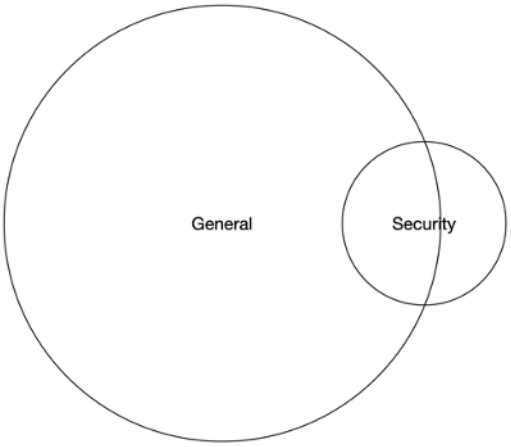


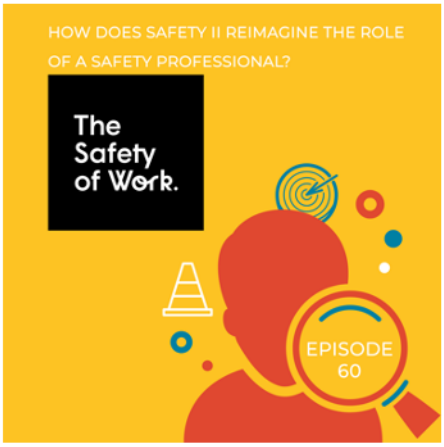
Cofense is a security system that withstands phishing attacks.
Example: stopping phishing emails (was/is a novel attack that general performance doesn't address)
Create a system (Cofense example) that defends against attacks (don't just train users and stop)
Security team does the work; designs and builds the system using their security expertise
Analysis includes costs and benefits of improving security performance, mainly risk reduction

Example 2: (brag) early vulnerability management work in ~2001, targeting a specific threat (worms; Nimda) – built a sociotechnical system that included support and reporting tailored to management, engineering, and a simple prioritization model targeting preauth RCEs: reached full effect in 3 years (2004 internal pen-test by Cofense founder; Foundstone, Mandiant, FireEye)

Image: Cofense, Secure360 2015

Guided Adaptability



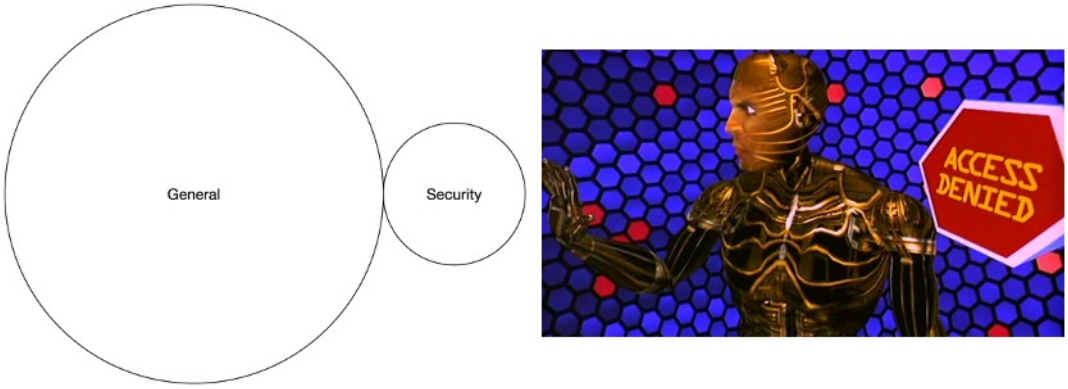


Mode 2 example. (Mode 1 examples are boring).
Capacities to manage safety: “analyse hazards, implement controls, monitor conformance, delegate authorities, and standardize safety culture”
Challenges: work-as-imagined vs work-as-done; understand how work is done and support safe variation from the rules.
“The mode we present here, ‘guided adaptability’, is not a new idea, but clarifies the principle that safety comes neither from preventing or encouraging variation, but from recognising that variation is inevitable. The goal of safety management is to facilitate safe variation.”
“to create foresight about the changing shape of risk, and facilitate action, before people are harmed “

https://safetyofwork.com/episodes/ep60-how-does-safety-ii-reimagine-the-role-of-a-safety-professional

Provan, D. J., Woods, D. D., Dekker, S. W. A., & Rae, A. J. (2020). Safety II professionals: How resilience engineering can transform safety practice. *Reliability Engineering & System Safety*, *195*, 106740. https://doi.org/10.1016/j.ress.2019.106740

Work against the adversary



Mode 3 example. Work against the adversary without affecting the work; degrade the performance of the attacker. This is one of the ways security is different than safety.

Example from Wolfgang yesterday; the pen-tester that was defeated by making it harder to attack (block IPs with high 404 rates for 15 min)

Wolfgang Goerlich, Secure360 2022, “Between the Chair and Keyboard”
Most of Marcus Ranum’s work.

- Assumption 1: organizations are sociotechnical systems
- Assumption 2: all failures are systems failures
- Argument 1: resilience improves through performance
- Argument 2: security performance is correlated with general performance
- Argument 3: there are three modes of security performance
- Implications: optimize risk management based on your performance mode

Recap

Questions?
Challenges?

https://www.information-safety.org
https://www.linkedin.com/in/jbenninghoff/
@jbenninghoff
jbenninghoff@mac.com

Ask me about the model, or any of my other work!

- Dossier 1: A socio-technical case study of an IT major incident management team
- Dossier 2: A review of an Agile Transformation change initiative using Structured Enquiry
- Dossier 3: A comparison of NIST and STPA risk assessment methods applied to an informational website
- Dossier 4: Development of an Agile CONOPS for an automated software delivery system using Activity Theory
- Dossier 6: A cross-domain review of cybersecurity and general competency frameworks
- Thesis: A cross-team study of factors contributing to software systems resilience at a large health care company

Summary of master's research