What Safety Science taught me about Information Risk

A New Model for Security Performance

John Benninghoff

SIRACon 2012!

Organizing Risk Management Programs

Or, What I learned from the Aviation Industry and the US Secret Service

TRANSVASIVE

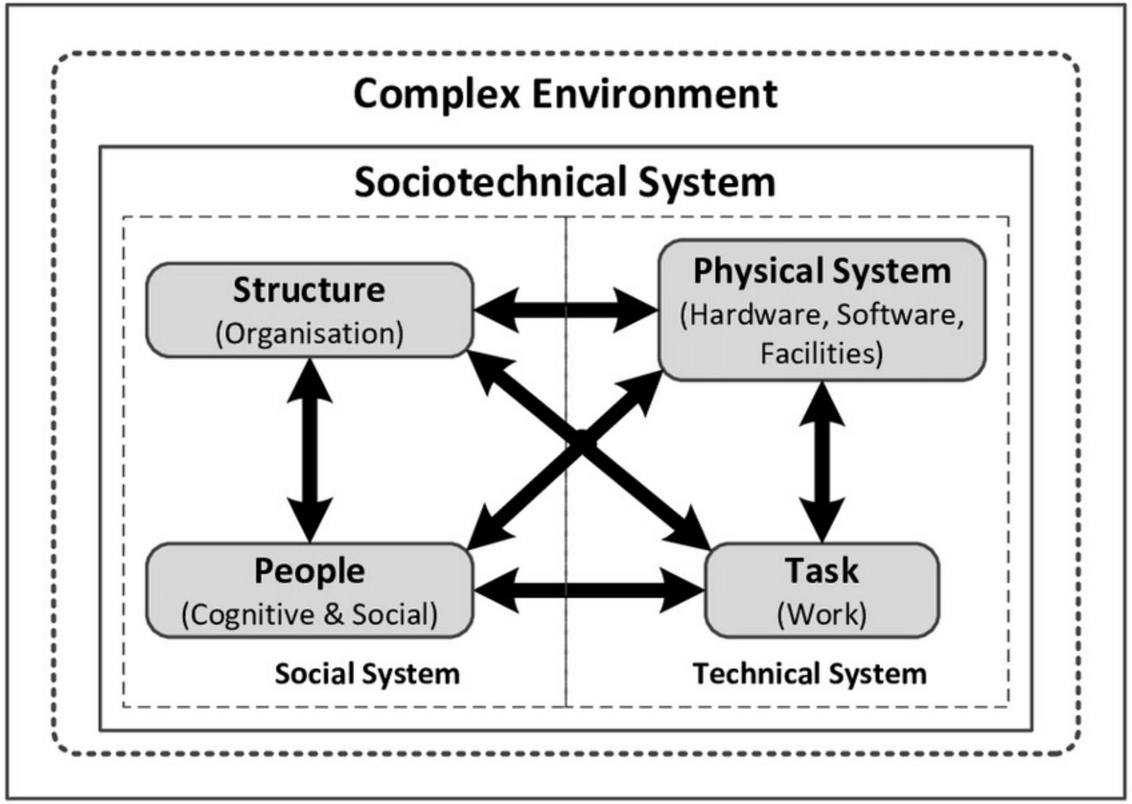
Transparent and Pervasive Security

Assumptions backed by accepted theory

Arguments for a new theoretical model backed by evidence

Implications of the model for information risk management

Assumption 1: organizations are sociotechnical systems



Assumption 2: all failures are systems failures

How Systems Fail

Cognitive
Technologies
Laboratory

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.**
All of the interesting systems (e.g. transportation, healthcare, power generation) are inherently and unavoidably hazardous by the own nature. The frequency of hazard exposure can sometimes be changed but the processes involved in the system are themselves intrinsically and irreducibly hazardous. It is the presence of these hazards that drives the creation of defenses against hazard that characterize these systems.

2) **Complex systems are heavily and successfully defended against failure.**
The high consequences of failure lead over time to the construction of multiple layers of defense against failure. These defenses include obvious technical components (e.g. backup systems, 'safety' features of equipment) and human components (e.g. training, knowledge) but also a variety of organizational, institutional, and regulatory defenses (e.g. policies and procedures, certification, work rules, team training). The effect of these measures is to provide a series of shields that normally divert operations away from

Argument 1: resilience improves through performance

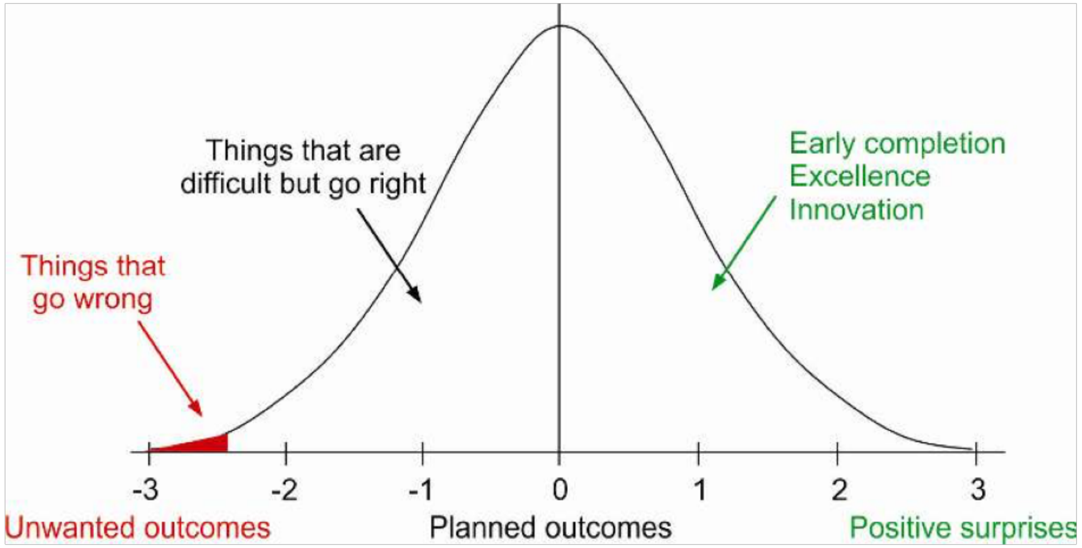
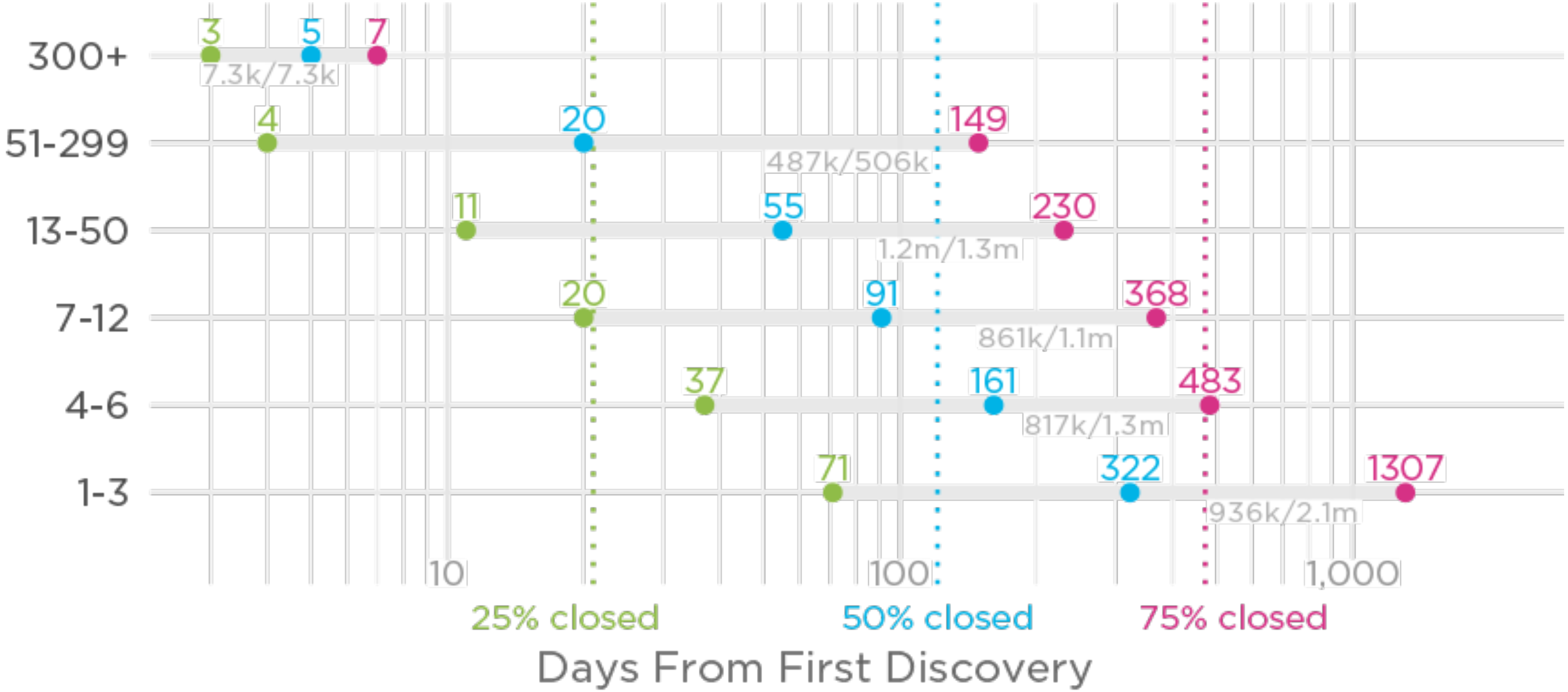


Figure 9: Event probability and safety focus

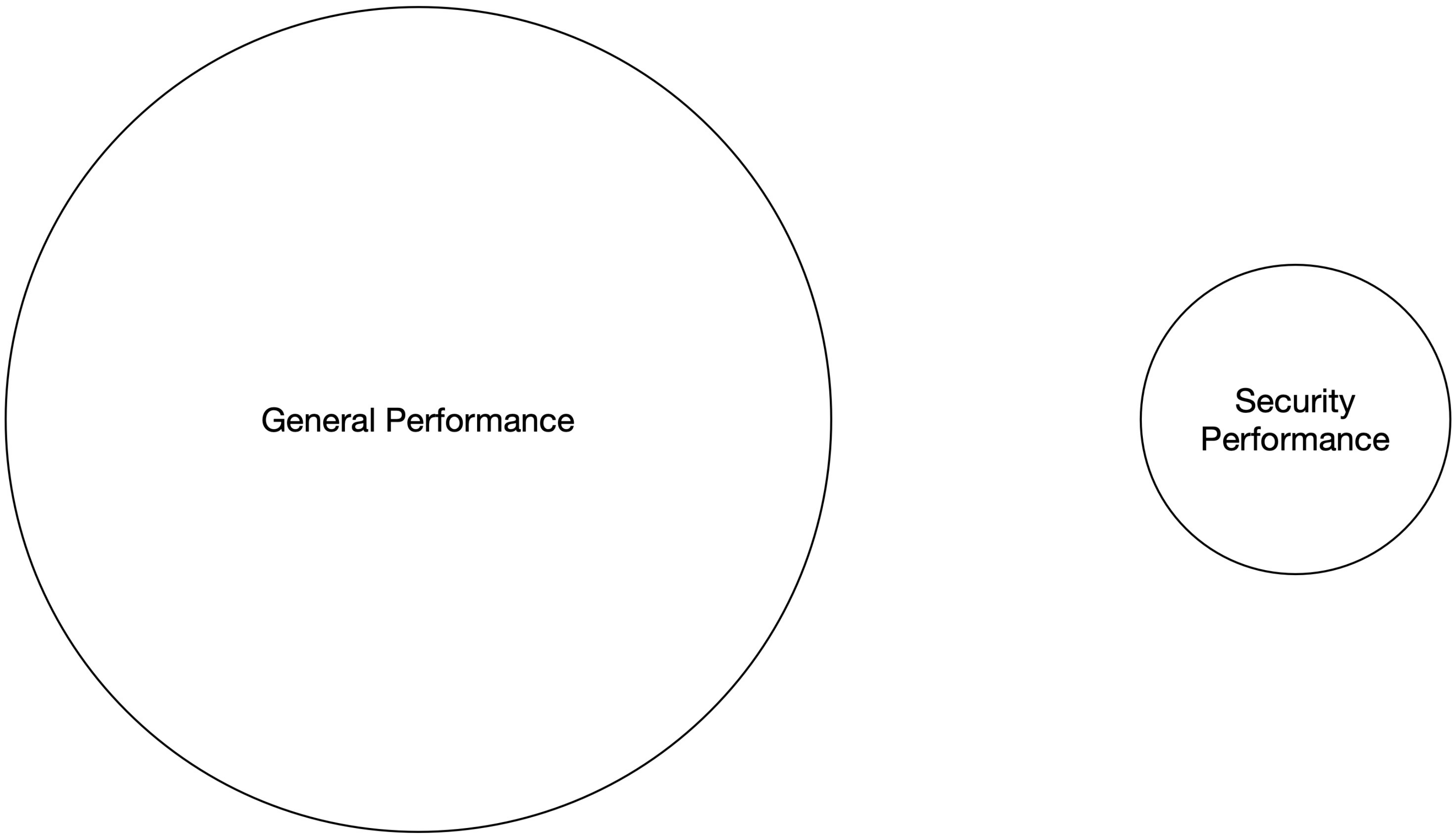
Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between 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,d]	0-15%[c,d]	46-60%

Argument 2: security performance is correlated with general performance

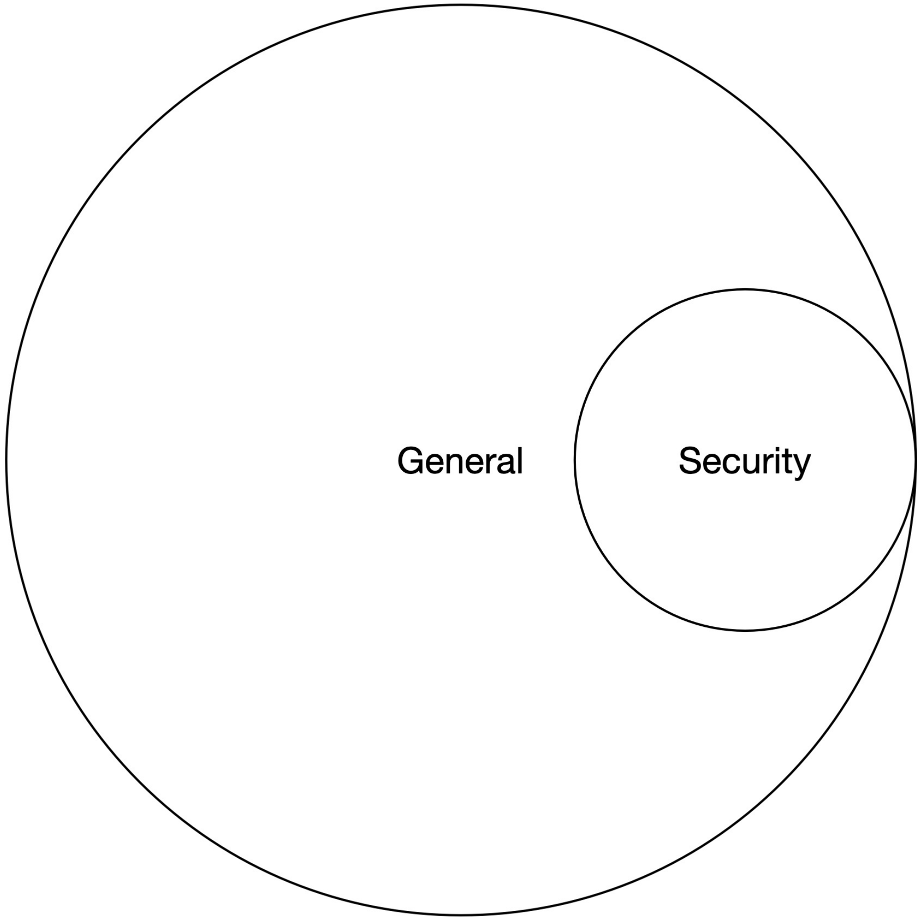


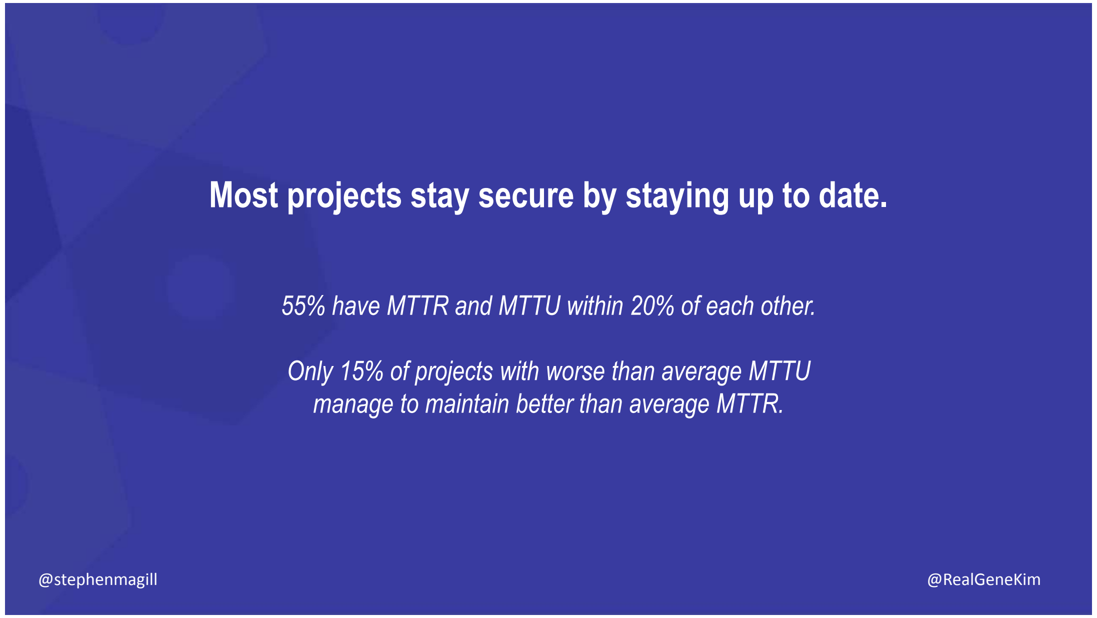
Source: Veracode SOSS Volume 9

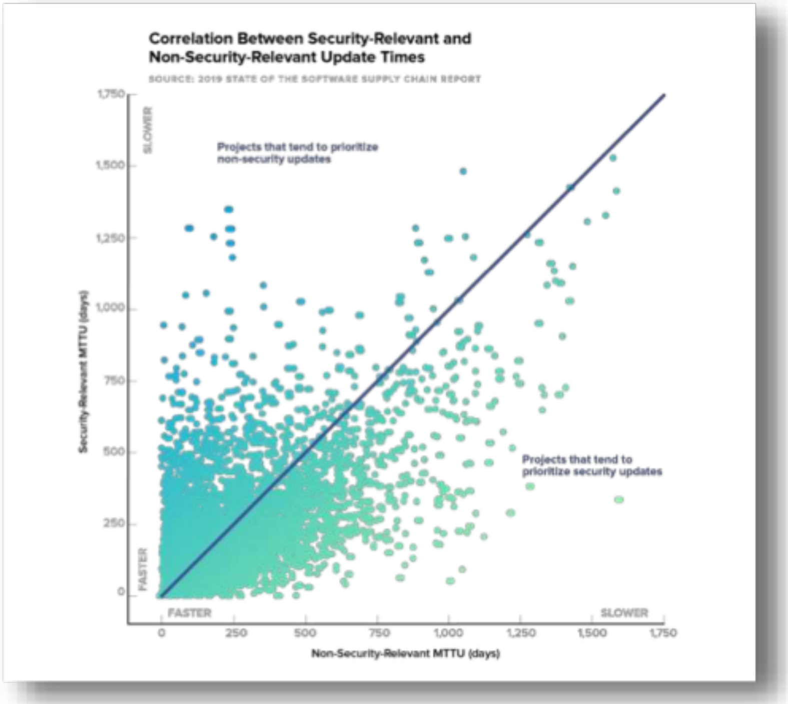
Argument 3: there are three modes of security performance



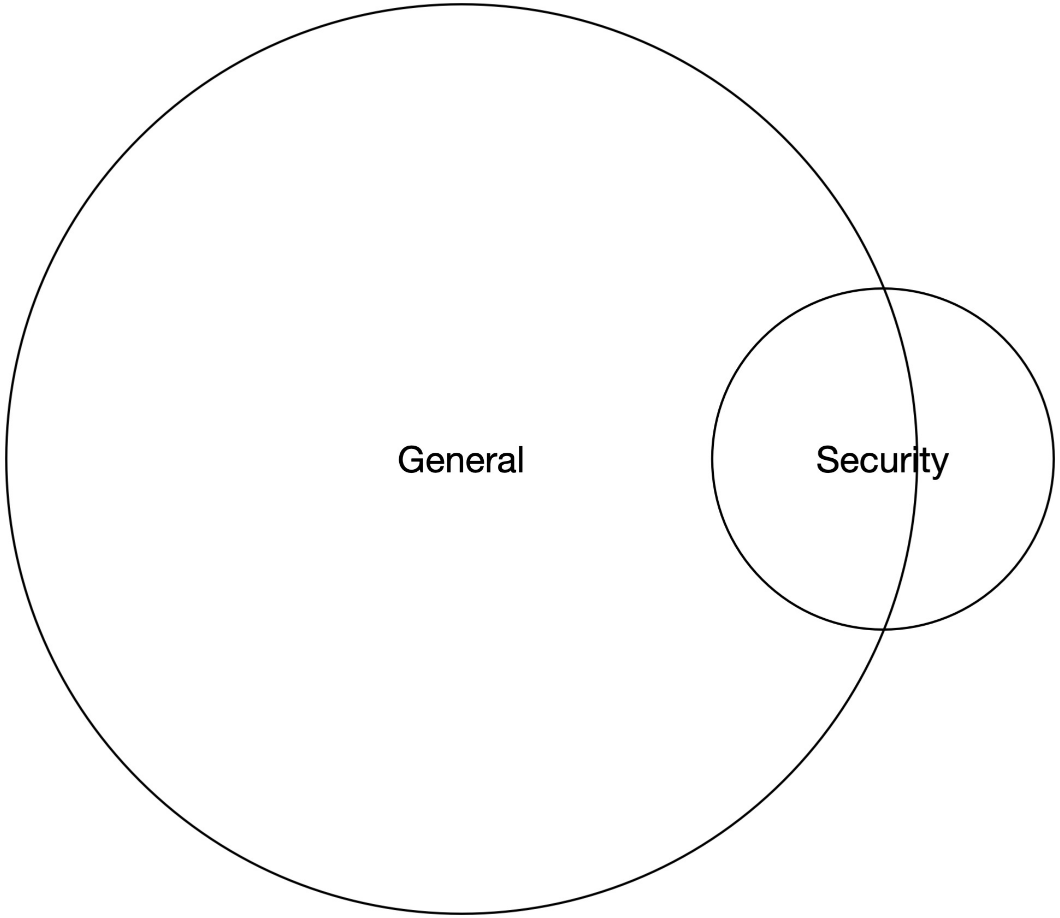
Mode 1

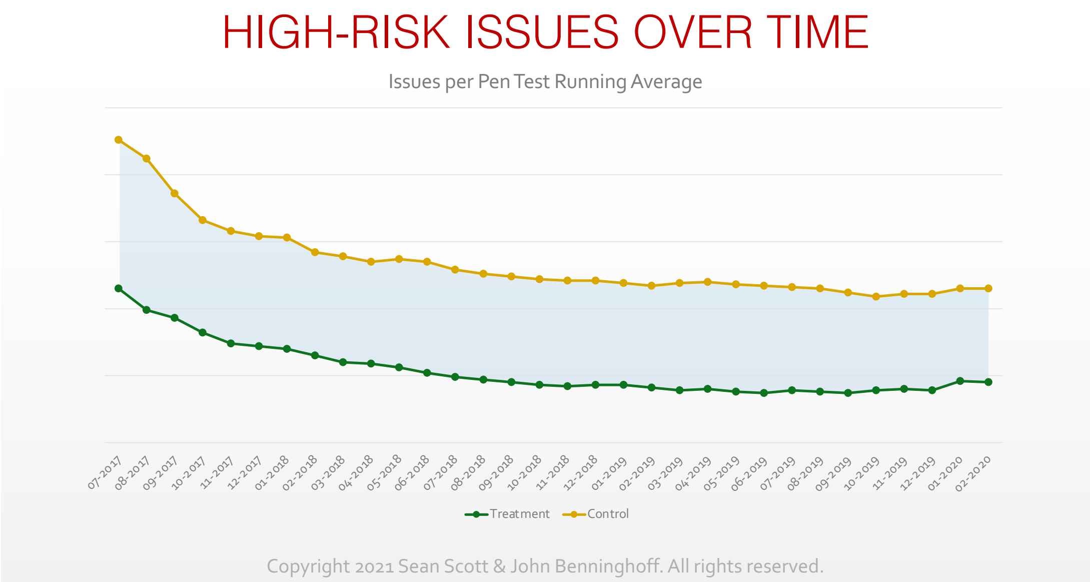

General
Security


Most projects stay secure by staying up to date.
55% have MTTR and MTTU within 20% of each other.
Only 15% of projects with worse than average MTTU manage to maintain better than average MTTR.
@stephenmagill
@RealGeneKim


Correlation Between Security-Relevant and Non-Security-Relevant Update Times
SOURCE: 2019 STATE OF THE SOFTWARE SUPPLY CHAIN REPORT
Projects that tend to prioritize non-security updates
Projects that tend to prioritize security updates
Security-Relevant MTTU (days)
Non-Security-Relevant MTTU (days)
SLOWER
FASTER
0
250
500
750
1,000
1,250
1,500
1,750

Mode 2





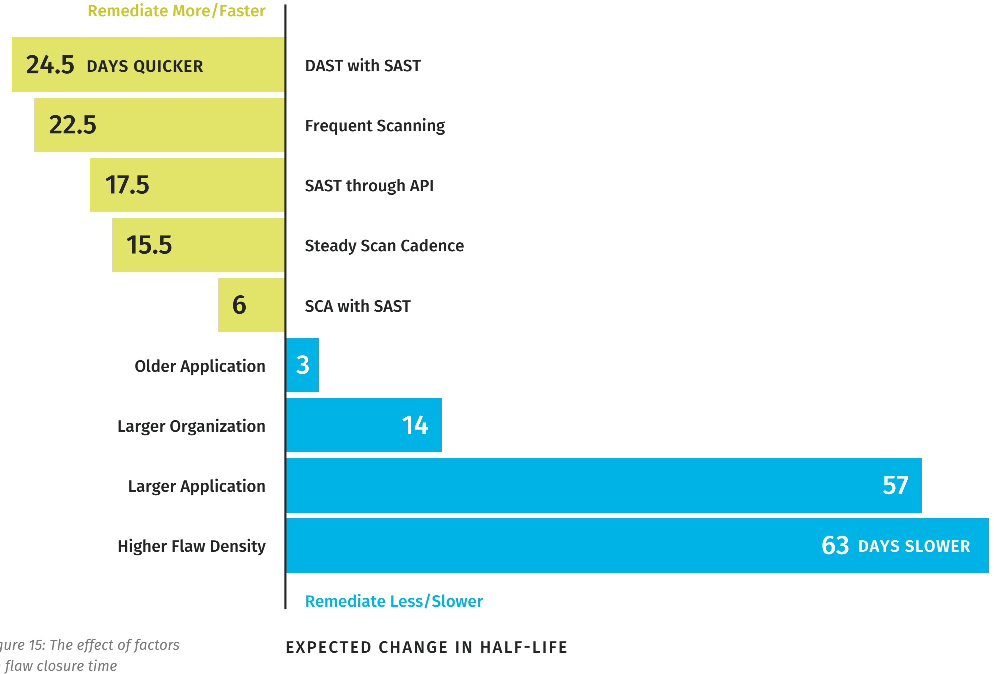
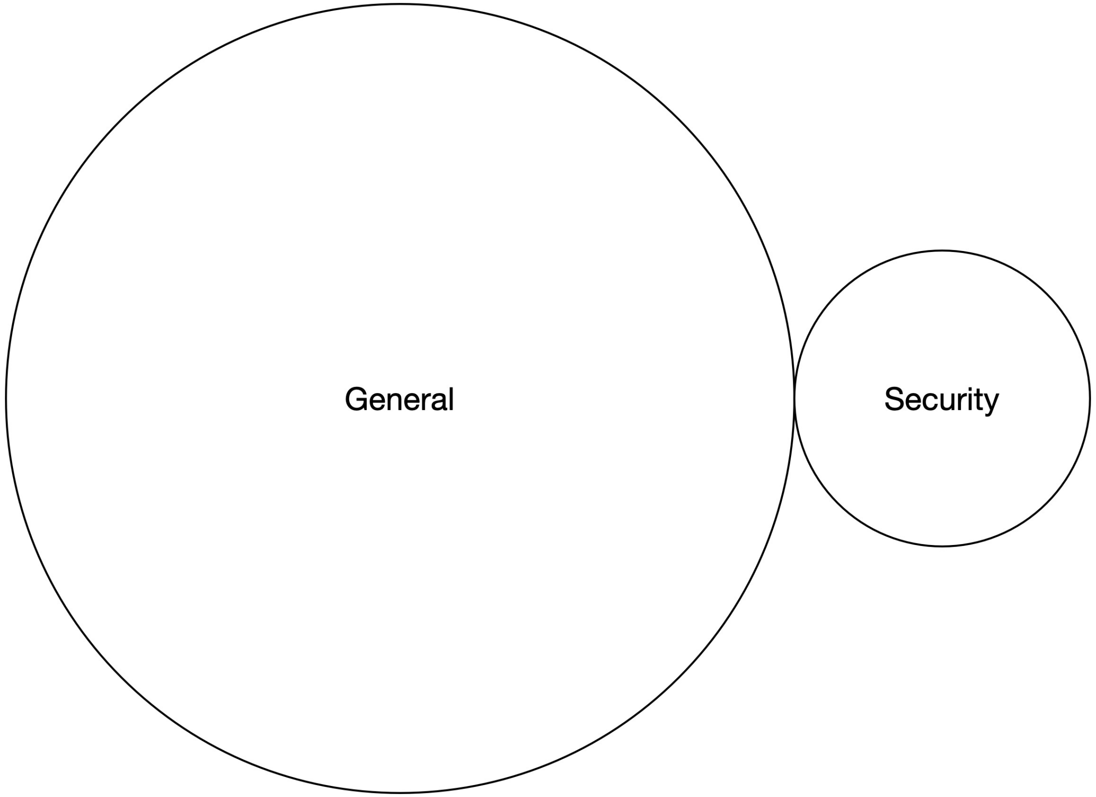
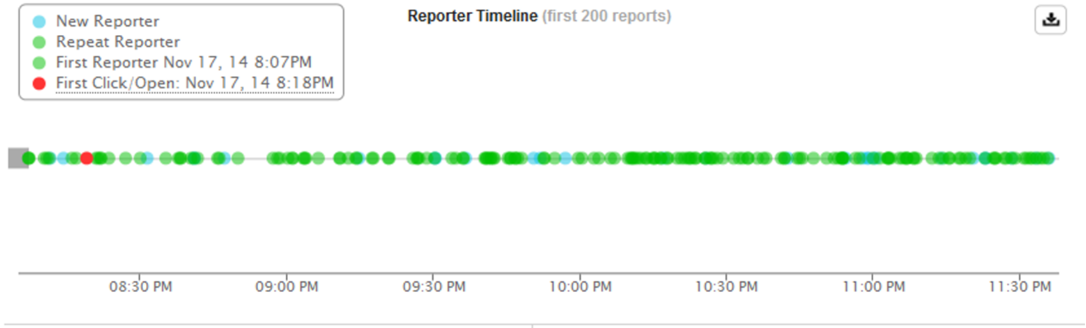


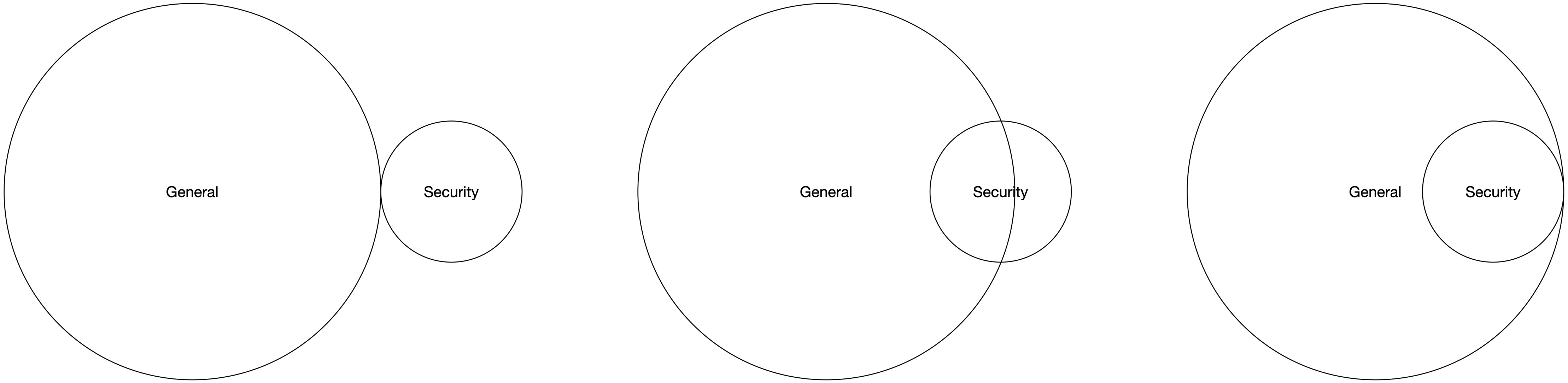
Figure 15: The effect of factors on flaw closure time

Mode 3


General
Security

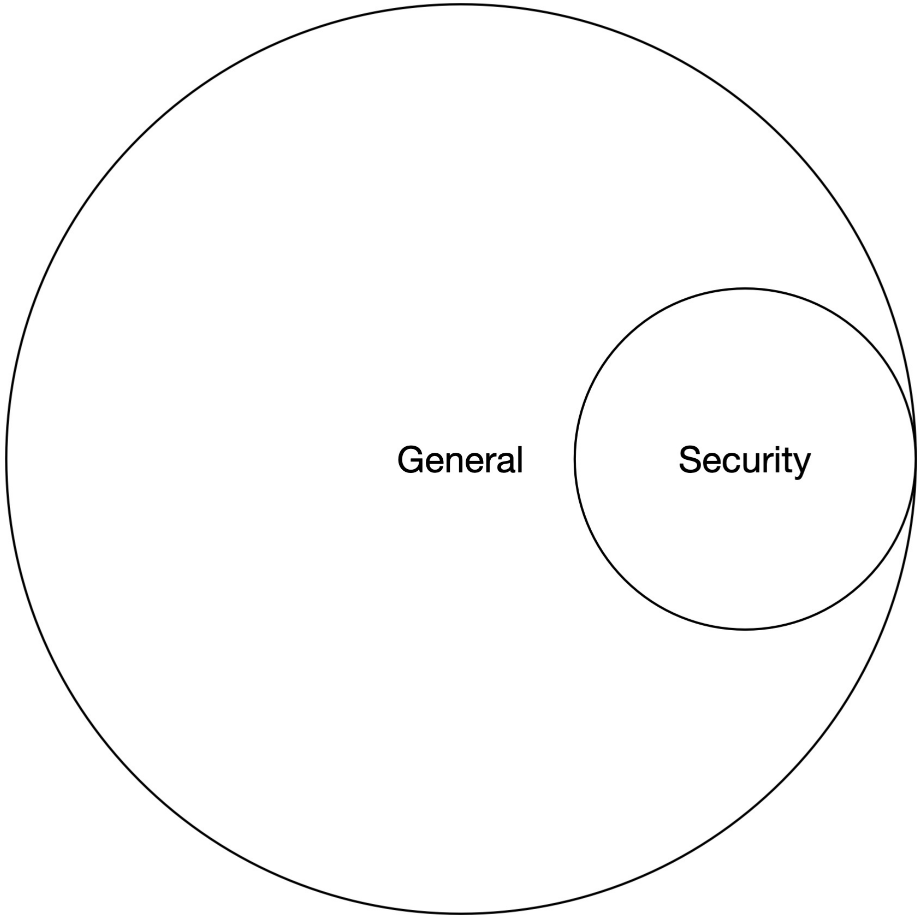

New Reporter
Repeat Reporter
First Reporter Nov 17, 14 8:07PM
First Click/Open: Nov 17, 14 8:18PM
Reporter Timeline (first 200 reports)
08:30 PM
09:00 PM
09:30 PM
10:00 PM
10:30 PM
11:00 PM
11:30 PM

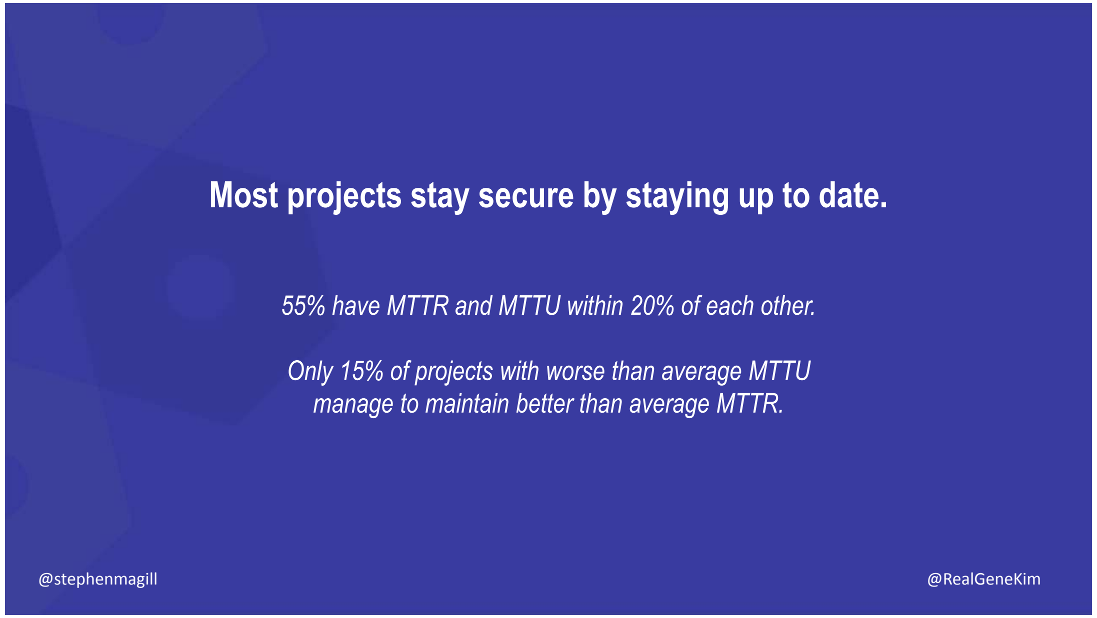
Mode 3 ⇨ Mode 2 ⇨ Mode 1



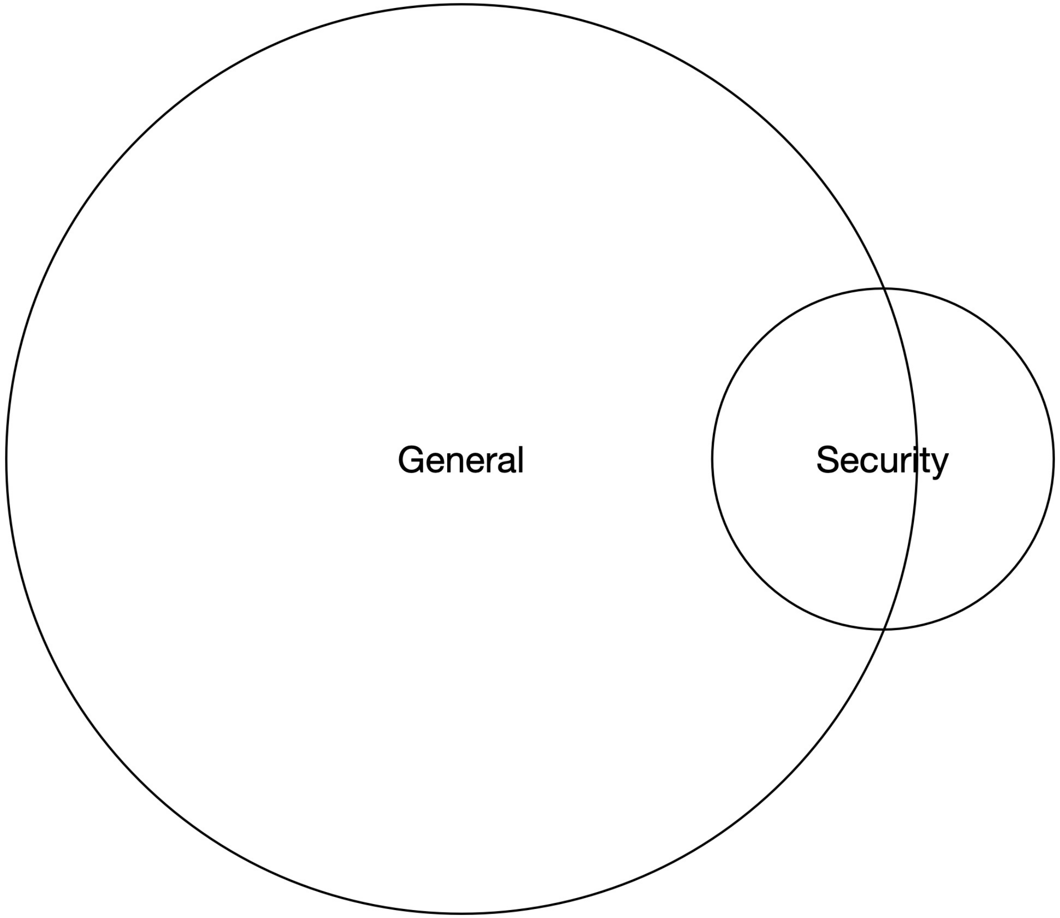
Implications: optimize risk management based on your performance mode

Mode 1: improve general performance





Mode 2: add security enhancements to general performance



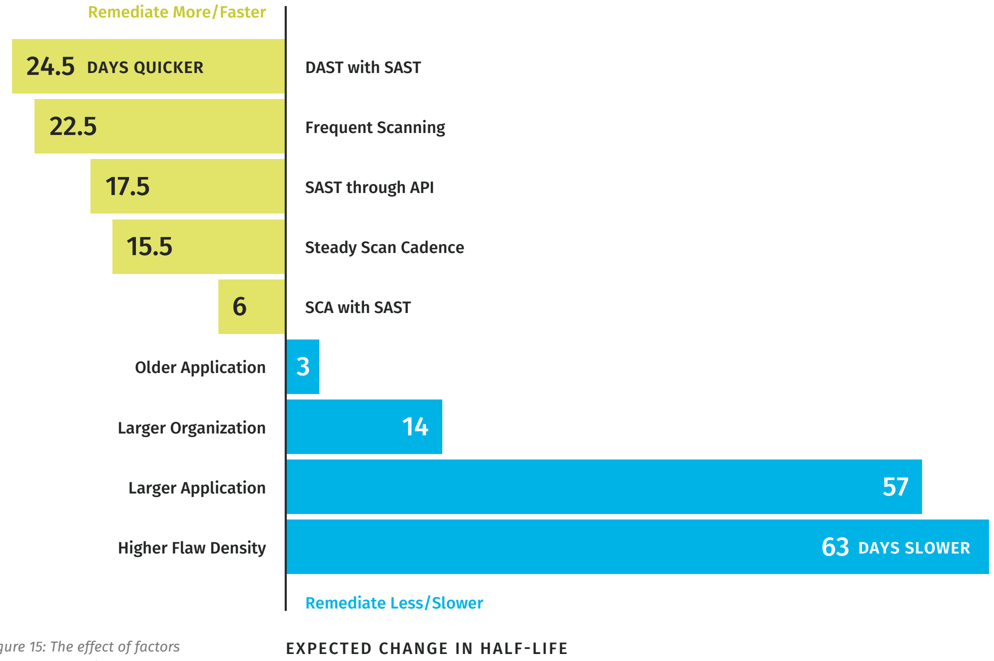
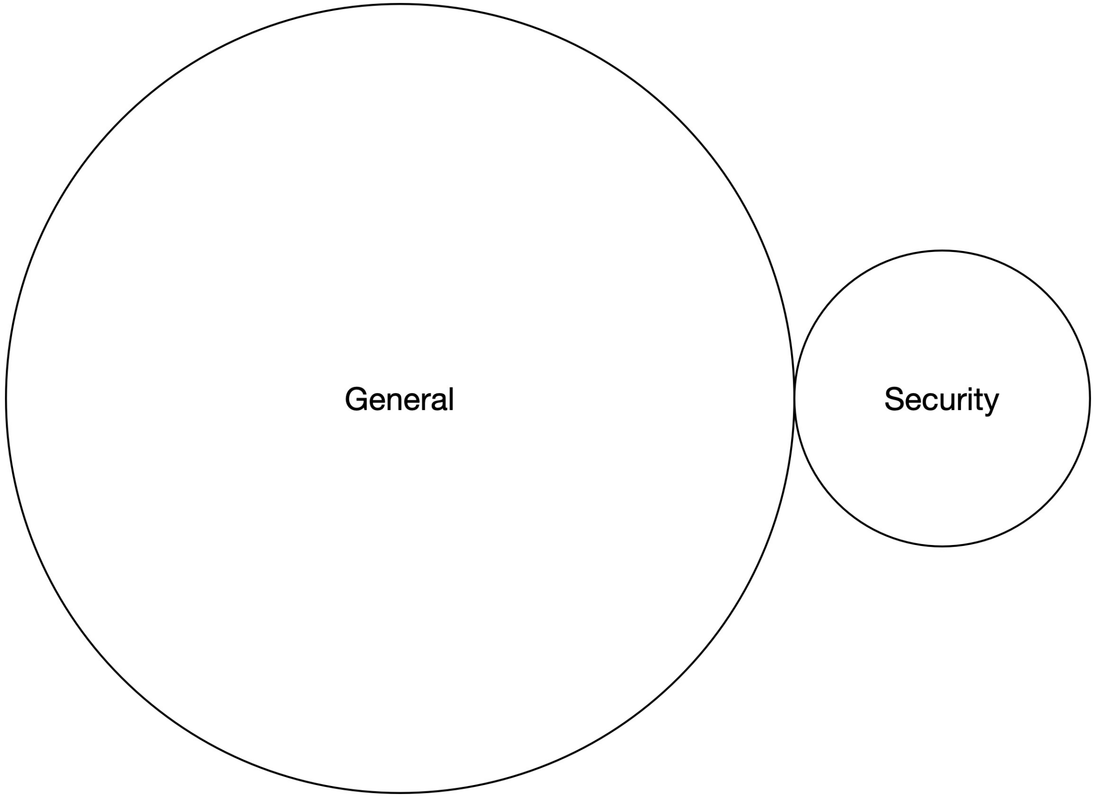
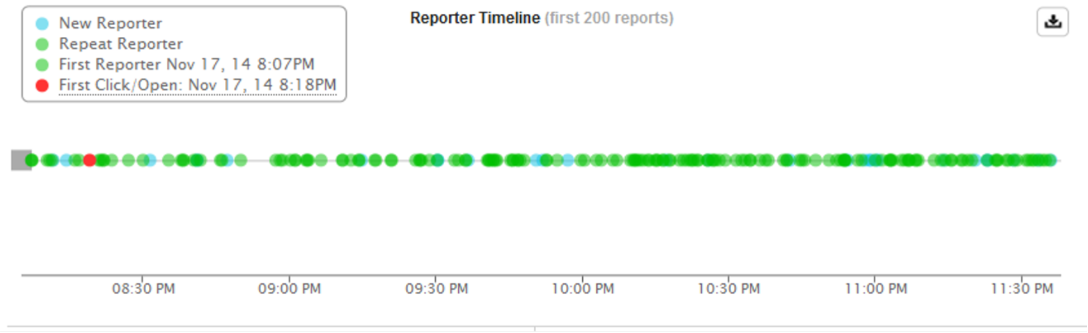


Figure 15: The effect of factors on flaw closure time

Mode 3: create security-specific systems





- 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

Questions?
Challenges?

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

- 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