

To DevSecOps or not to DevSecOps: is that a question ?

Using an Archetype-based model of
Security in DevOps



Mario Platt
Strategy Director

Today's Agenda

- ✓ The 2 schools of thought
- ✓ Who is it for ?
- ✓ DevOps Security Archetype model
- ✓ Meet the Archetypes
- ✓ Grappling with Constraints and Bottlenecks
- ✓ Helping Archetypes - Governance and Maturity
- ✓ Helping Archetypes - Team Topologies
- ✓ Do we need DevSecOps or not, then ?

The 2 schools of thought

DevSecOps

 **Abhay Bhargav** @abhaybhargav · Feb 24, 2019

Replying to @dinodaizovi

For one, **devsecops** fosters (or at least aims to) more collaboration with cross functional teams than traditional enterprise security constructs. Working with cross functional teams leads to a codified knowledge base (automation) and the cycle continues. 1/

1 1 1 1 1

 **Dino A. Dai Zovi** @dinodaizovi · Jul 21, 2019

Replying to @bascule and @thephreck

Yeah, I feel like a lot of "**DevSecOps**" still doesn't quite get the most important aspect of having security *be* engineering teams to increase empathy. Working to secure the org using the same tools and same environment prevents "do as I say, not as I do."

1 1 1 6 1

 **Avi Douglen** @sec_tigger · Dec 5, 2019

Replying to @secfigo @EndlessMason and 2 others

Absolutely agree with this! For now, we still need to treat it like something on its own - because that's what it is, right now. But it shouldn't be, and that is what we need to be working towards, when **DevSecOps** is not a thing.

1 1 1 2 1

 **Kelly Shortridge @ RSAC** @swagitda_ · Jun 5, 2019

Replying to @swagitda_ @anton_chuvakin and @nicolefv

My definition of **DevSecOps** is "a marketing term invented for security professionals who don't understand how to work with DevOps but who want to preserve their relevancy without real effort, ideally by buying a solution they understand with a new shiny label"

7 7 16 1

What DevSecOps isn't



Mario Platt @madplatt · 22h

DevSecOps is NOT #Compliance until you have an effective mechanism of tracing back to the policy requirements which mandate your technical controls. Until then, it's good you're doing it but your Compliance team is probably not getting that much benefit for THEIR responsibilities

William Gregorian @WillGregorian · Sep 10

Replies to @swagitda_

Oh, you didn't know? DevSecOps is also compliance. All-in-one cornucopia of awesome winning sauce.

1 reply · 1 retweet · 6 likes

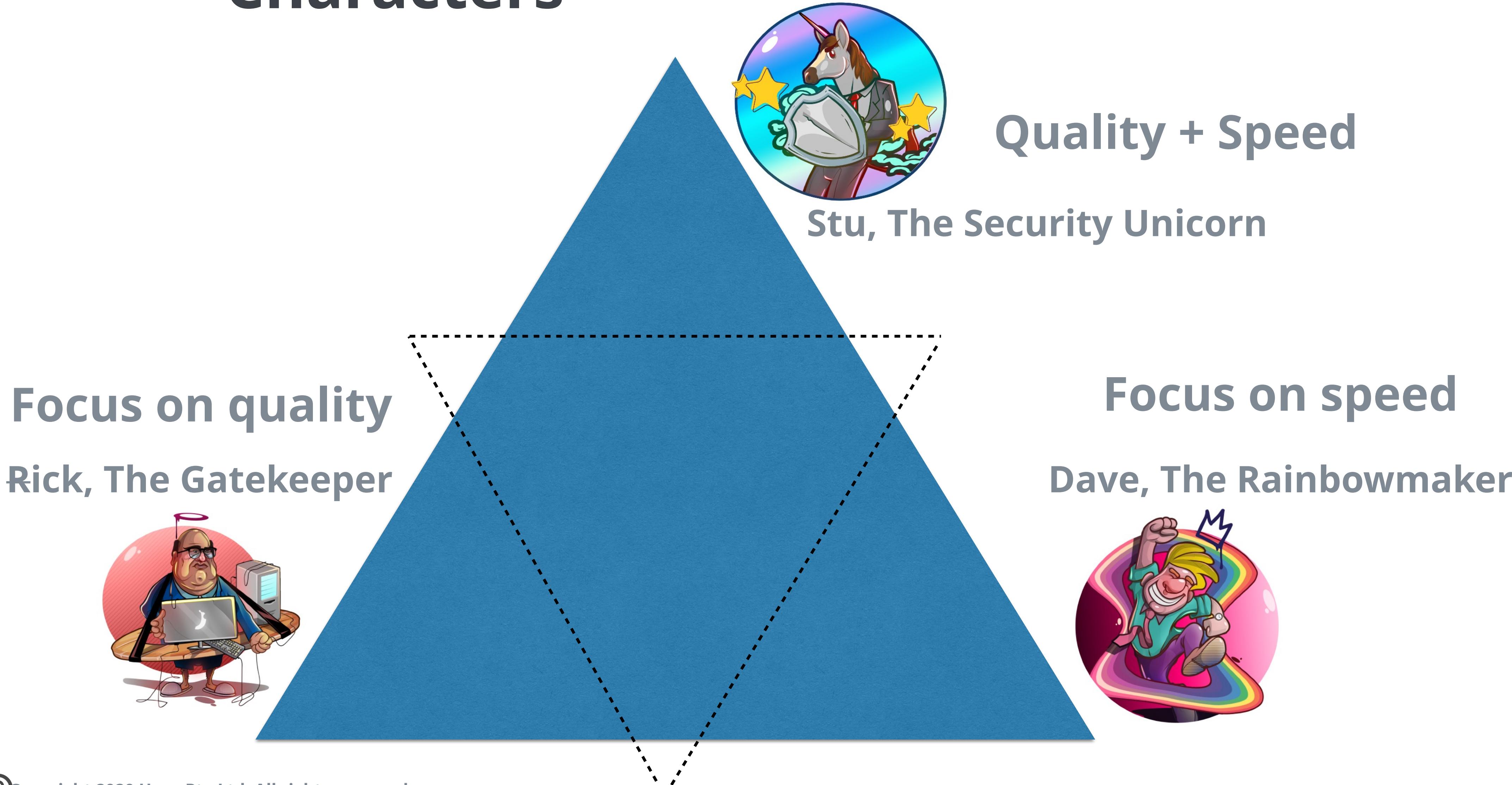
These are Security Pros.

But who is DevSecOps meant to serve ?

Control & Governance functions

Engineering organisation

DevOps Security Archetype model - Characters



DevOps Security Archetype model - Organisations / Teams



The Security Unicorns

The Gatekeepers



The Rainbowmakers



Meet Dave and his team of Rainbowmakers



Dave and the teams aren't negligent or stupid.

They lack **situational awareness** to do better.

Process, social practice and cognitive load considerations are failing THEM

Meet Dave and his team of Rainbowmakers



Essential characteristics:

- **No integrated security telemetry**
- **Avoids engagement with Compliance**
- **No secure baselines or modelling of threats**
- **Limited automated testing overall**
- **No product level security reporting**
- **Security is someone else's job, or output of pentests/audits**

Meet Rick and his team of Gatekeepers



Dick and the teams aren't thick or business averse

They **lack understanding of control reliance** in a DevOps world and have **inertia due to past success** of their current model

Team Topologies and poor traceability from technical checks to Compliance objectives are failing THEM

Meet Rick and his team of Gatekeepers



Essential characteristics:

- **Gated processes supported by committees and Review Boards**
- **Limited understanding and trust of modern development practices & engineers**
- **Policies and standards developed outside of Engineering context**
- **Lack of technical knowledge, or worse, outdated threat models**

Meet Stu and his team of Security Unicorns



Stu and the teams are.... OK :)

They have a great Engineering culture and are a learning organisation.

Compliance colleagues **don't work in silos**, but **collaborate to establish process and then trust** that the teams are focused in developing quality software

Meet Stu and his team of Security Unicorns



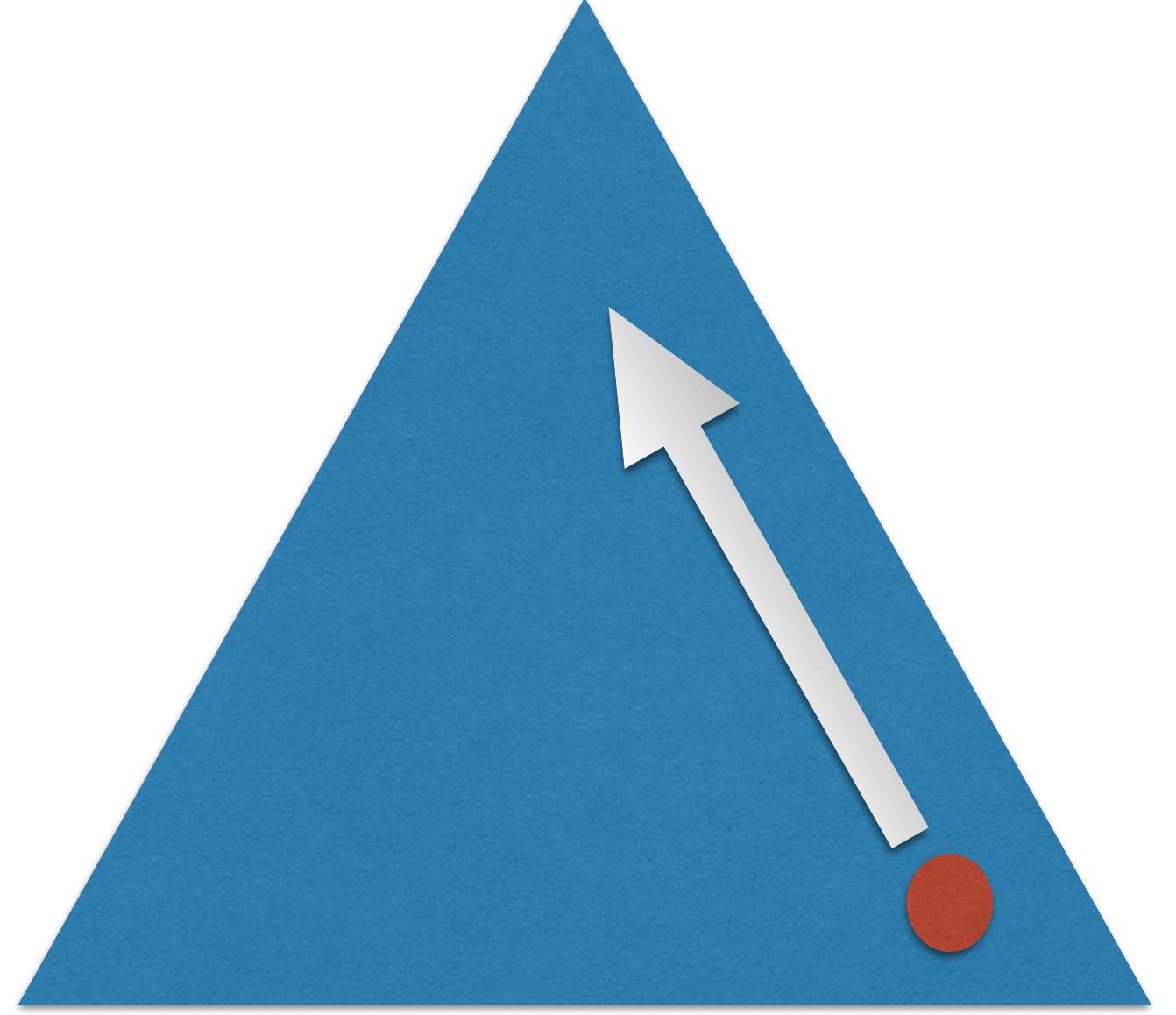
Essential characteristics:

- **Security is embedded into DevOps practices**
- **Security is an element of product/service quality**
- **Teams keep up to date threat models of what they build**
- **Their process has the right security at the right time.**
- **Compliance IS Code**
- **Less Command and Control from Compliance, more collaboration**

**But in their current practices, all are
affected by Bottlenecks**

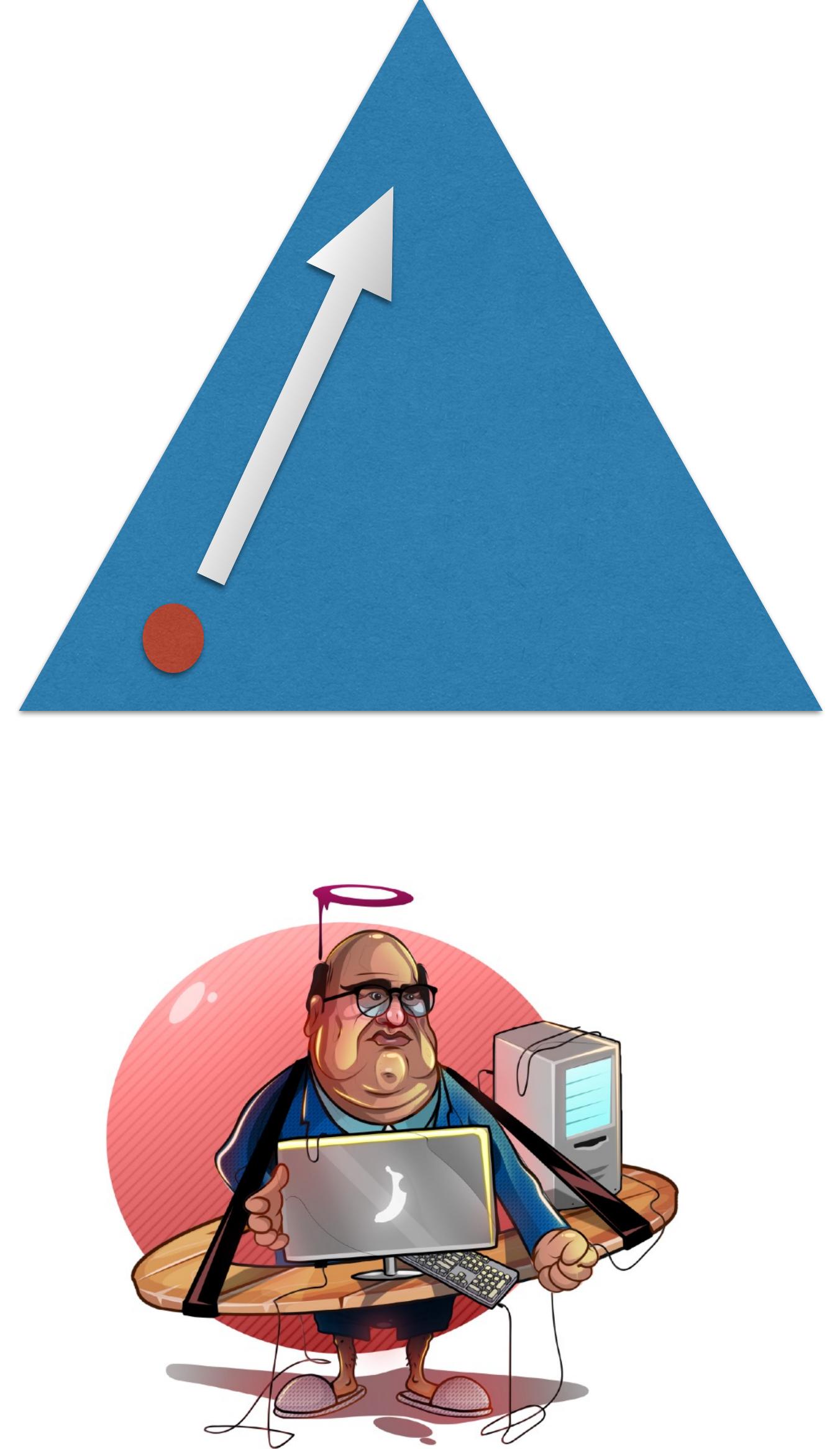
Bottlenecks to going up the triangle

- **Lack of Security Telemetry**
- **Product Management security prioritisation**
- **Security Expertise by DevOps and Software Engineers**
- **Poor Process assurance and practices**
- **Low sense of agency and ownership for security**



Bottlenecks to going up the triangle

- Assignment of security responsibilities
- Spending money on the wrong things
- Gated processes and out-of-band approvals
- Control reliance mismatches
- Policy ISN'T Code or Stories
- Command and control culture
- Inertia due to past success



Further Challenges

Communication

Traceability & Process

Learning journey and
Mental Models

Evolution of Practices

Team Topologies

The Communication Challenge

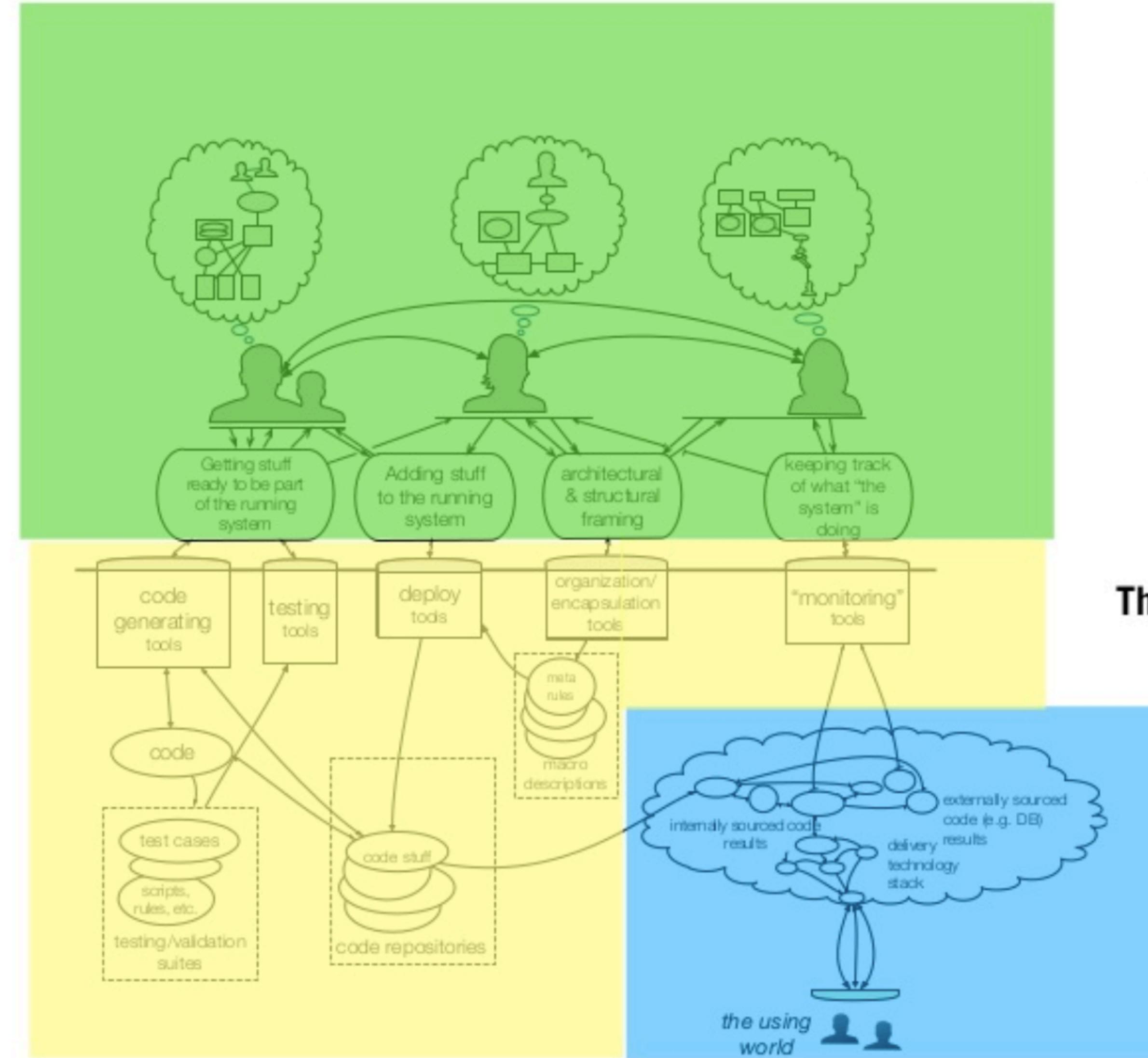
Their management systems

The Stories they tell and understand

The language used to communicate concerns

They see the system differently.

They say the same names, but they mean different things



***The Work Is Done
Here***

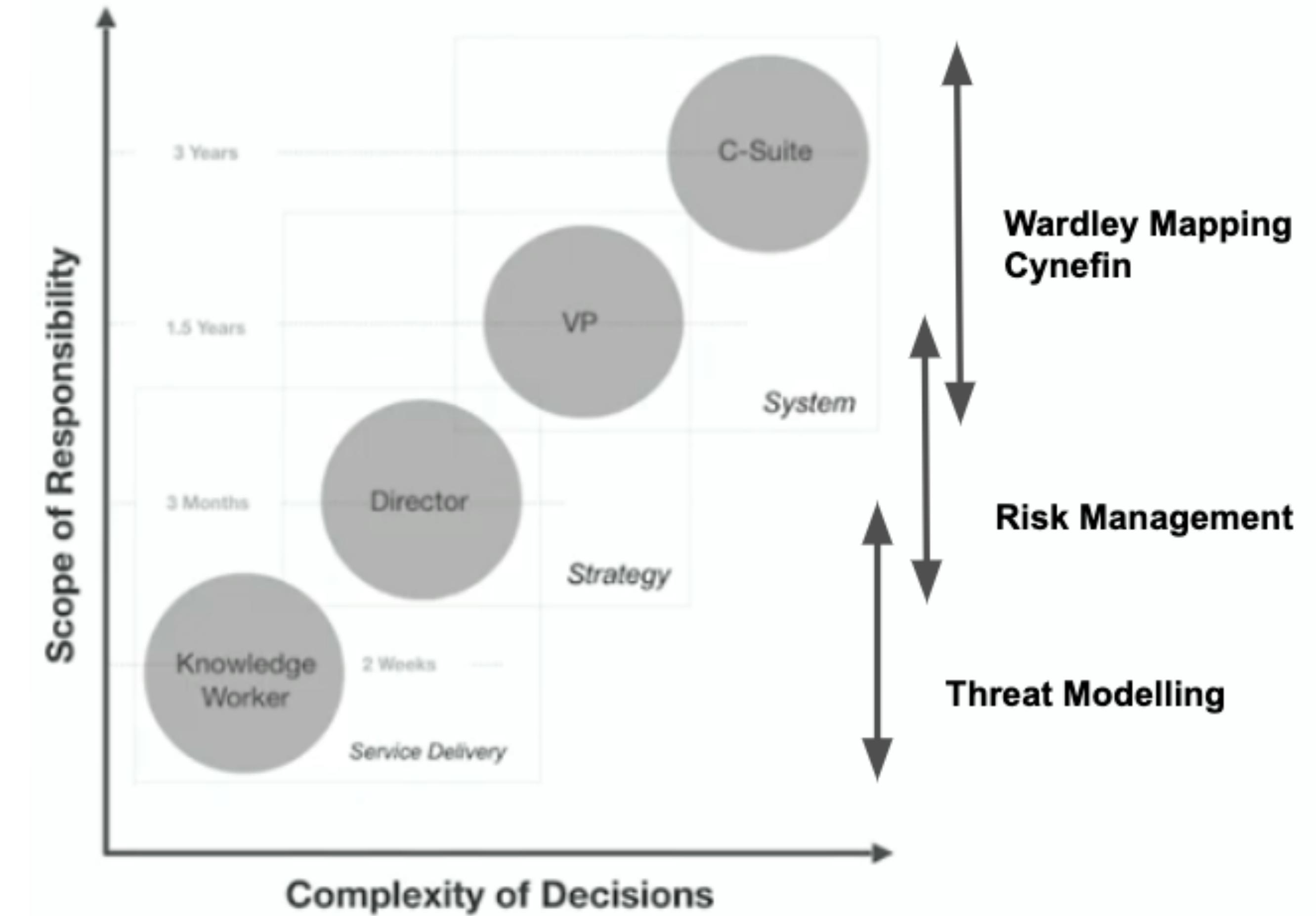
**The Stuff You Build and
Maintain With**

**Your Product Or
Service**

Complexity of decisions vs Scope of responsibility

The timespan of their narratives are different.

Sprints vs managing risks



And the language is different too



**Control testing
Testing procedures
Evidence review
Checklists and spreadsheets
Compliance to
Risk analysis and uncertainty**

Control... blah blah
control.... Blah....
Governance... blah... Risks....
Blah... Compliance....blah blah
Boogey man at the door

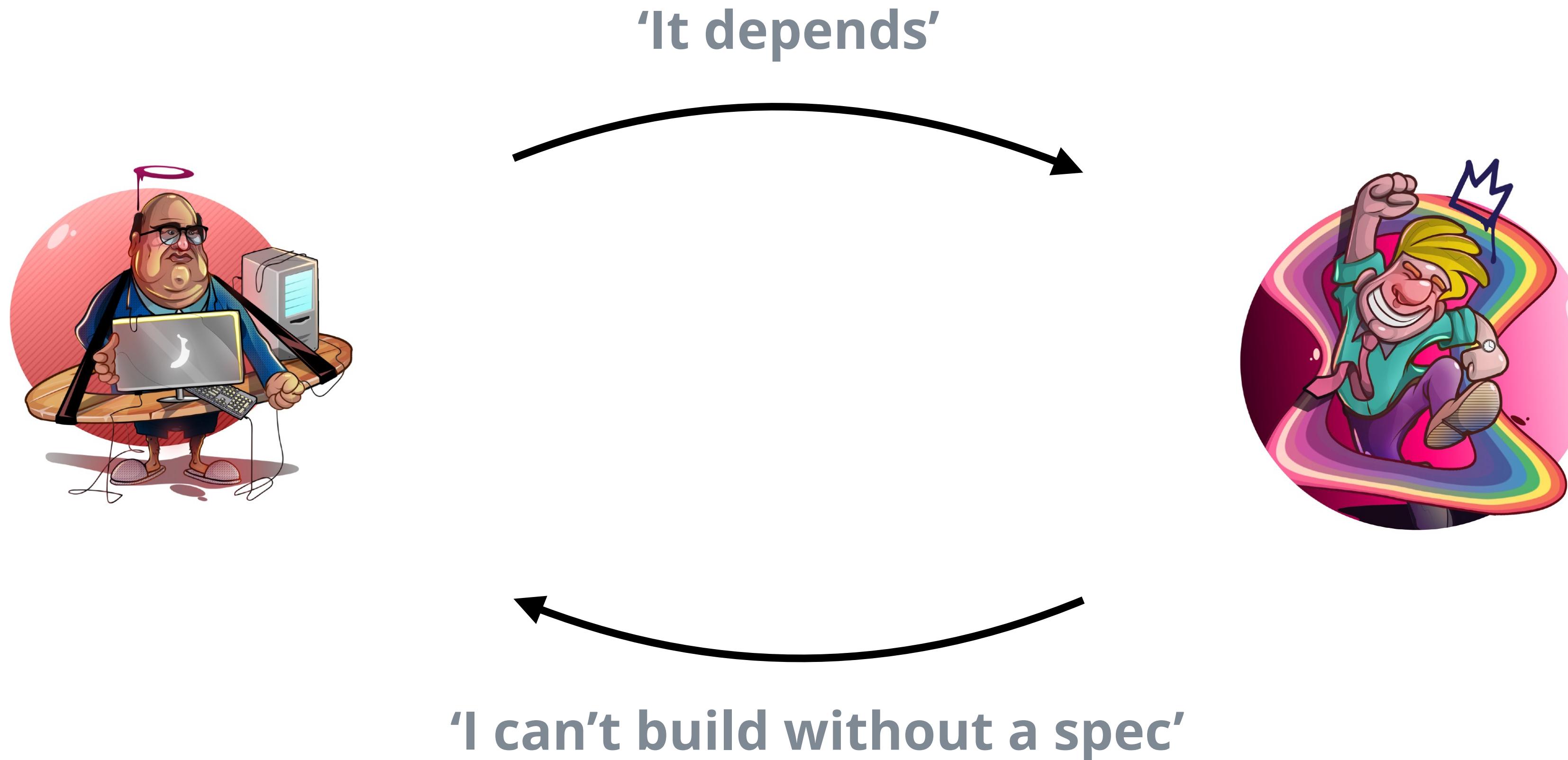
Wall of Confusion and Despair



**Code Tools
Processes and procedures
Delivery artefacts
Specifications
Sprints and Stories**

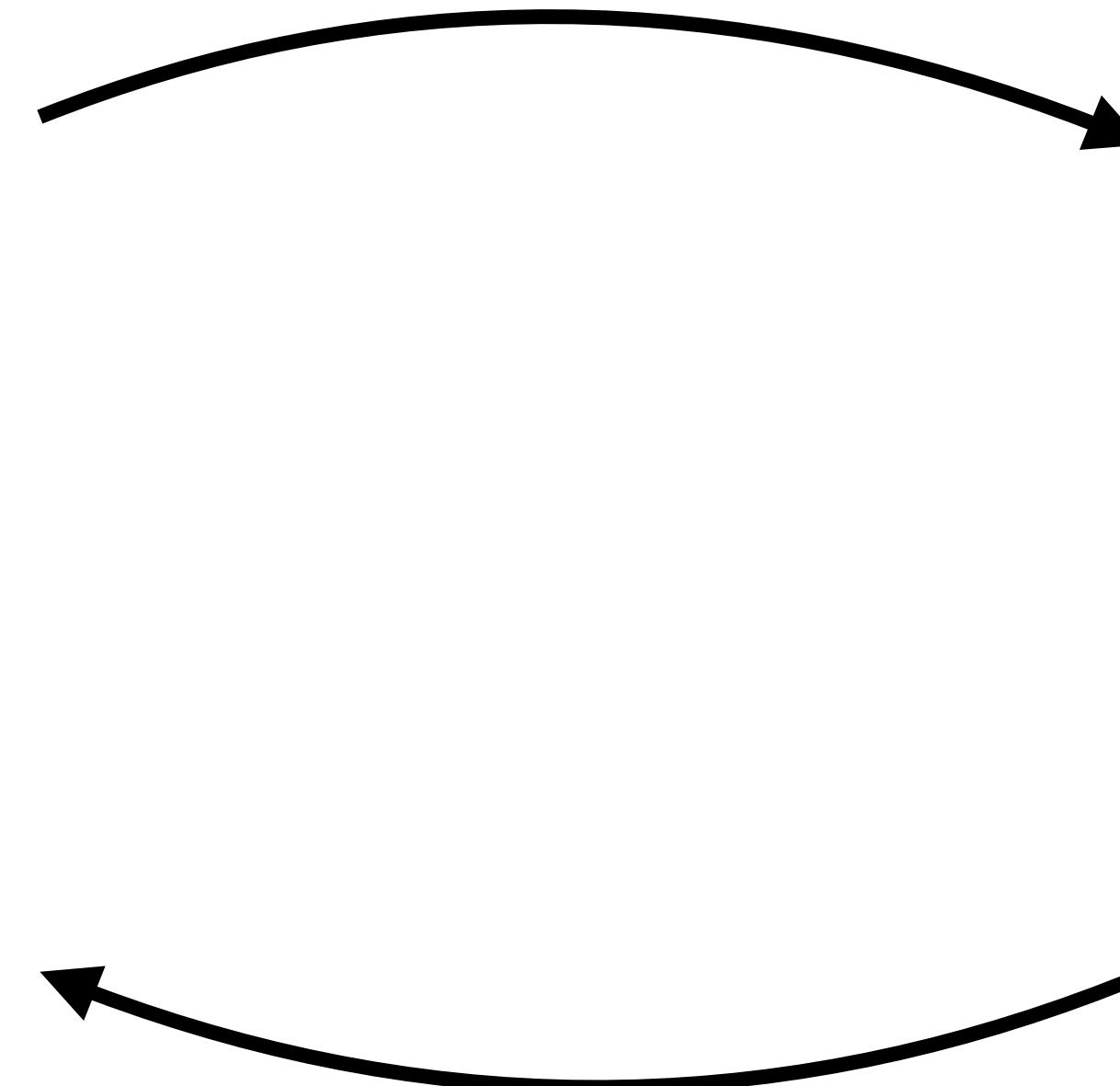
Features... bah blah... that
looks cool.... Blah blah...
Speed.... Argh can't get that
function to work blah blah
They won't get out of the
way...

New Regulation Syndrome



Uncontextualised Policies Syndrome

I've written policies
They're really good because I was technical a decade ago



Which policies and where ?
Those patterns don't make sense anymore. What are you on about ?

DoD Enterprise DevSecOps Reference

Design

Hysn DSOMM

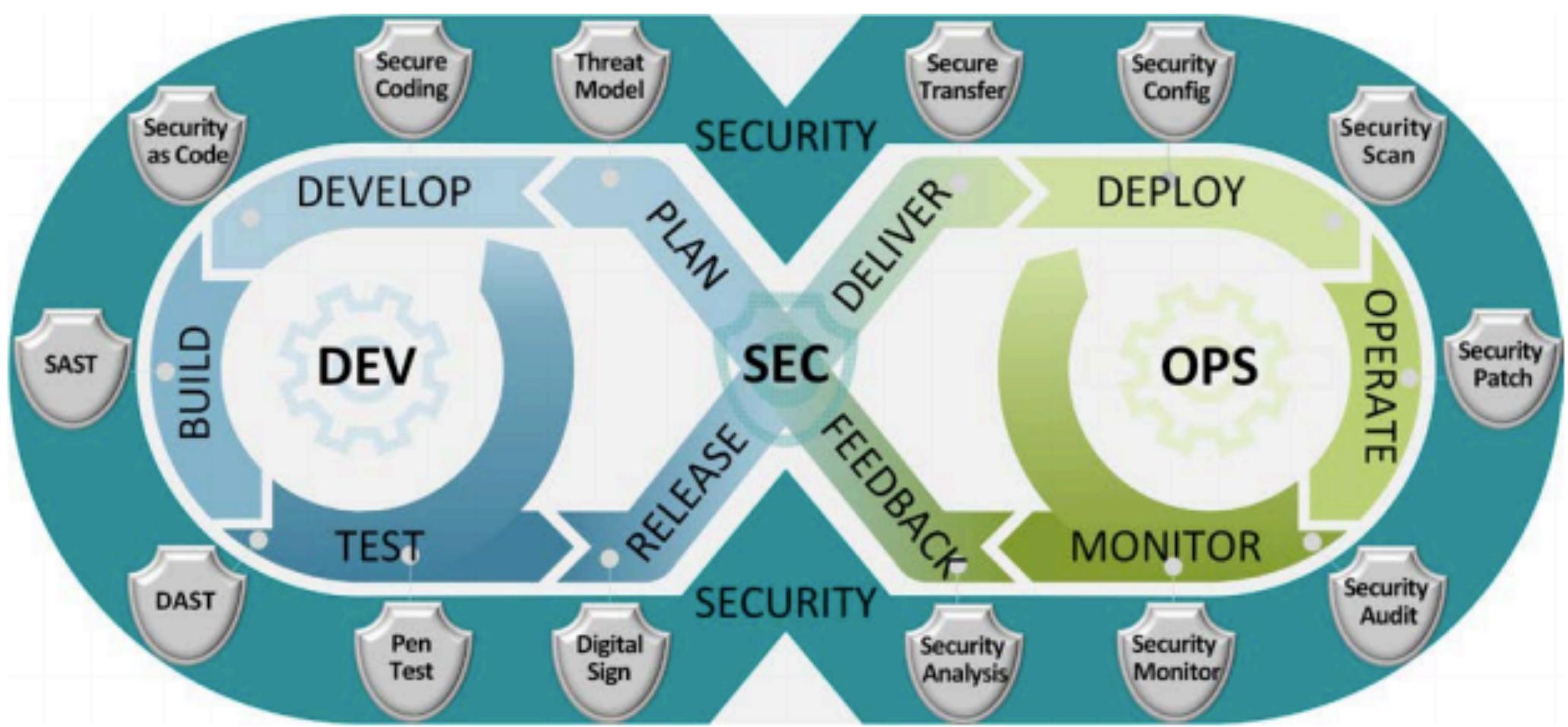
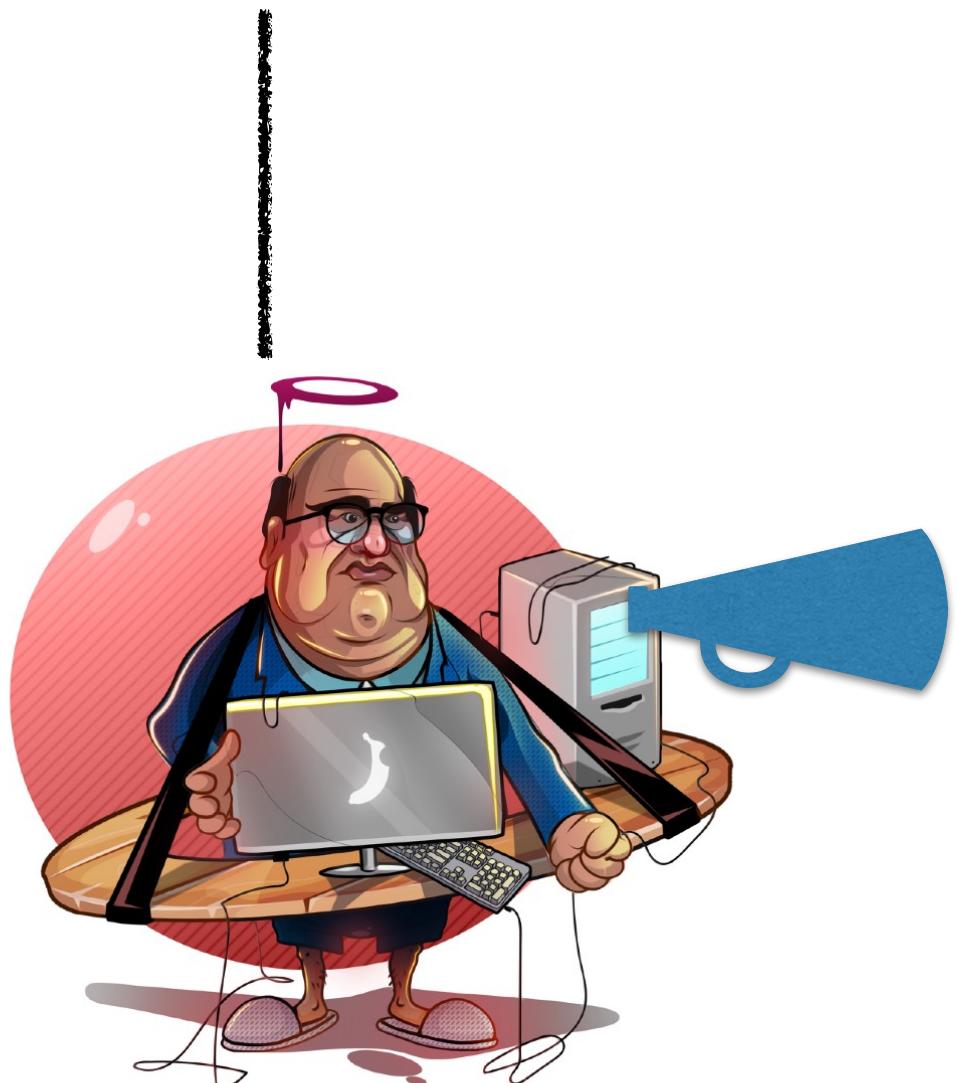
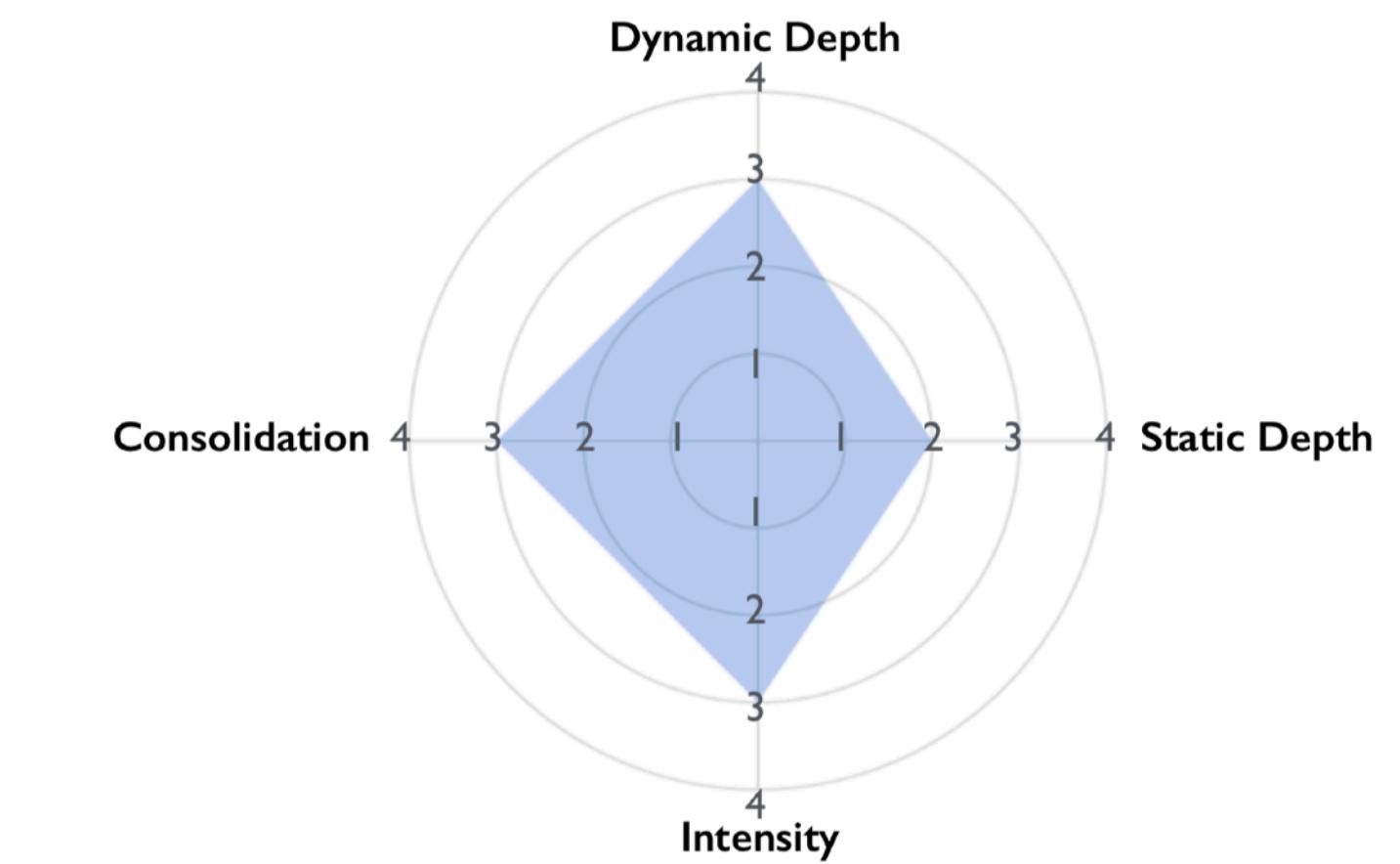


Figure 3: DevSecOps Software Lifecycle



Helping communicate
between different mental
models

Mapping and traceability

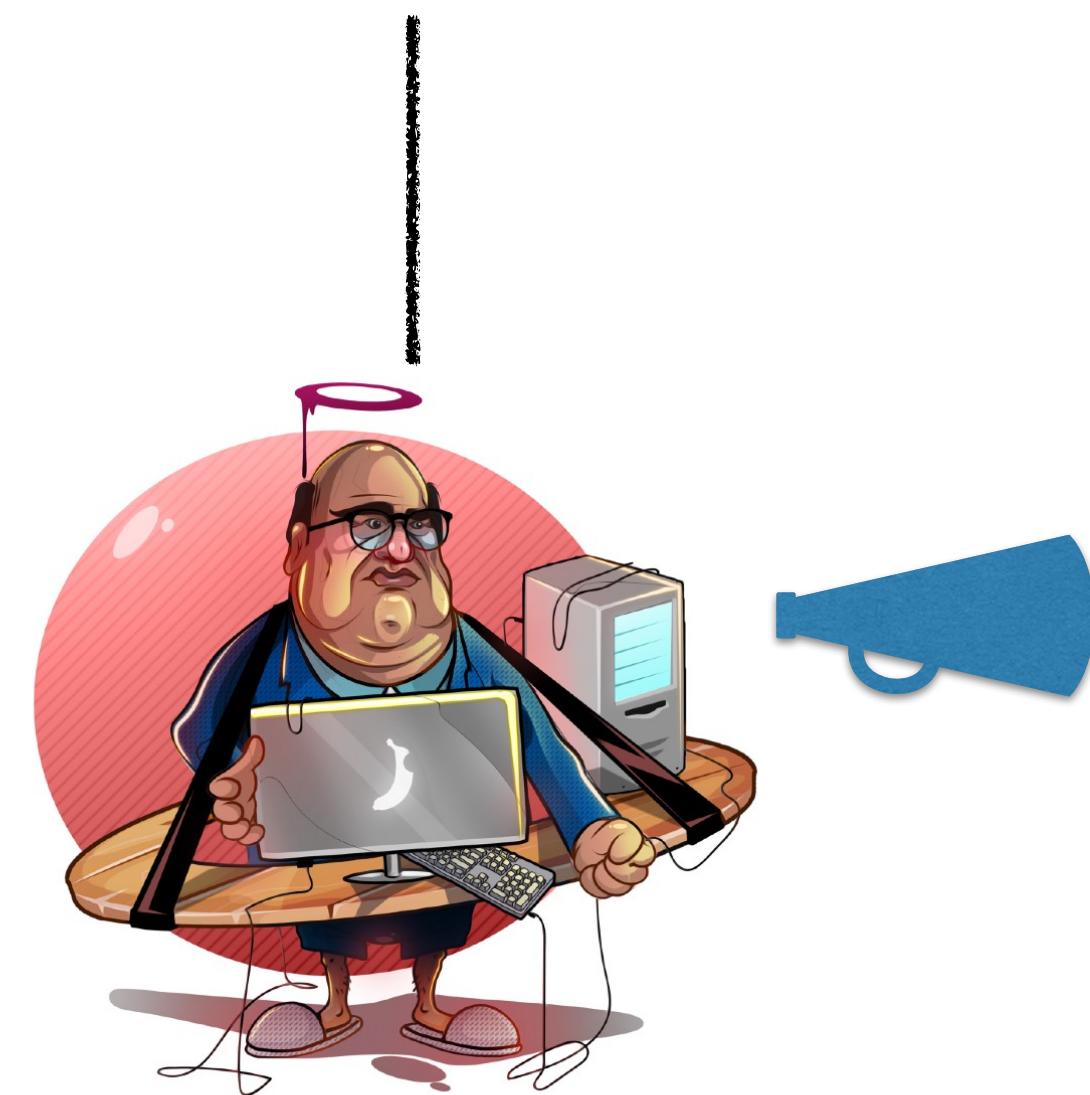


Requirements



OWASP

Application Security Verification Standard 4.0



Helping communicate
between different mental
models

Mapping and traceability

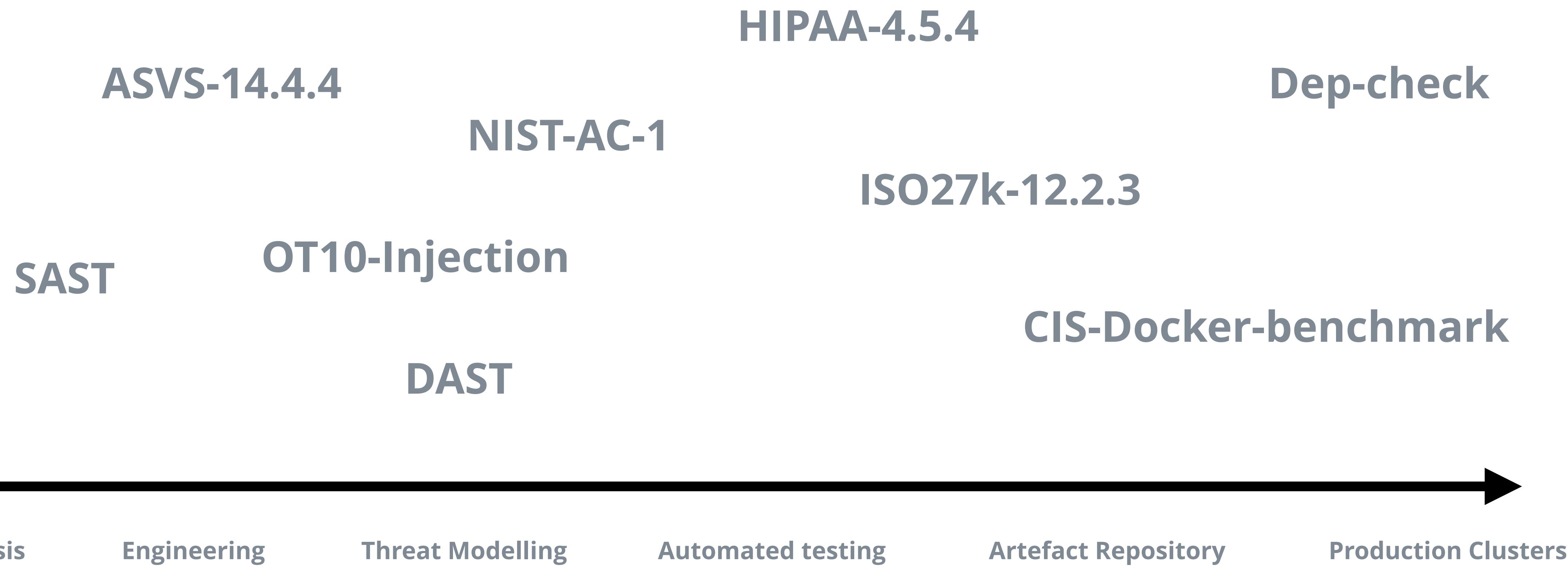
Implementation



Life is too short · AppSec is tough · Cheat!

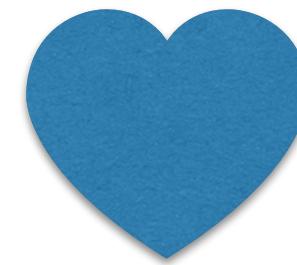


Use metadata for traceability glue



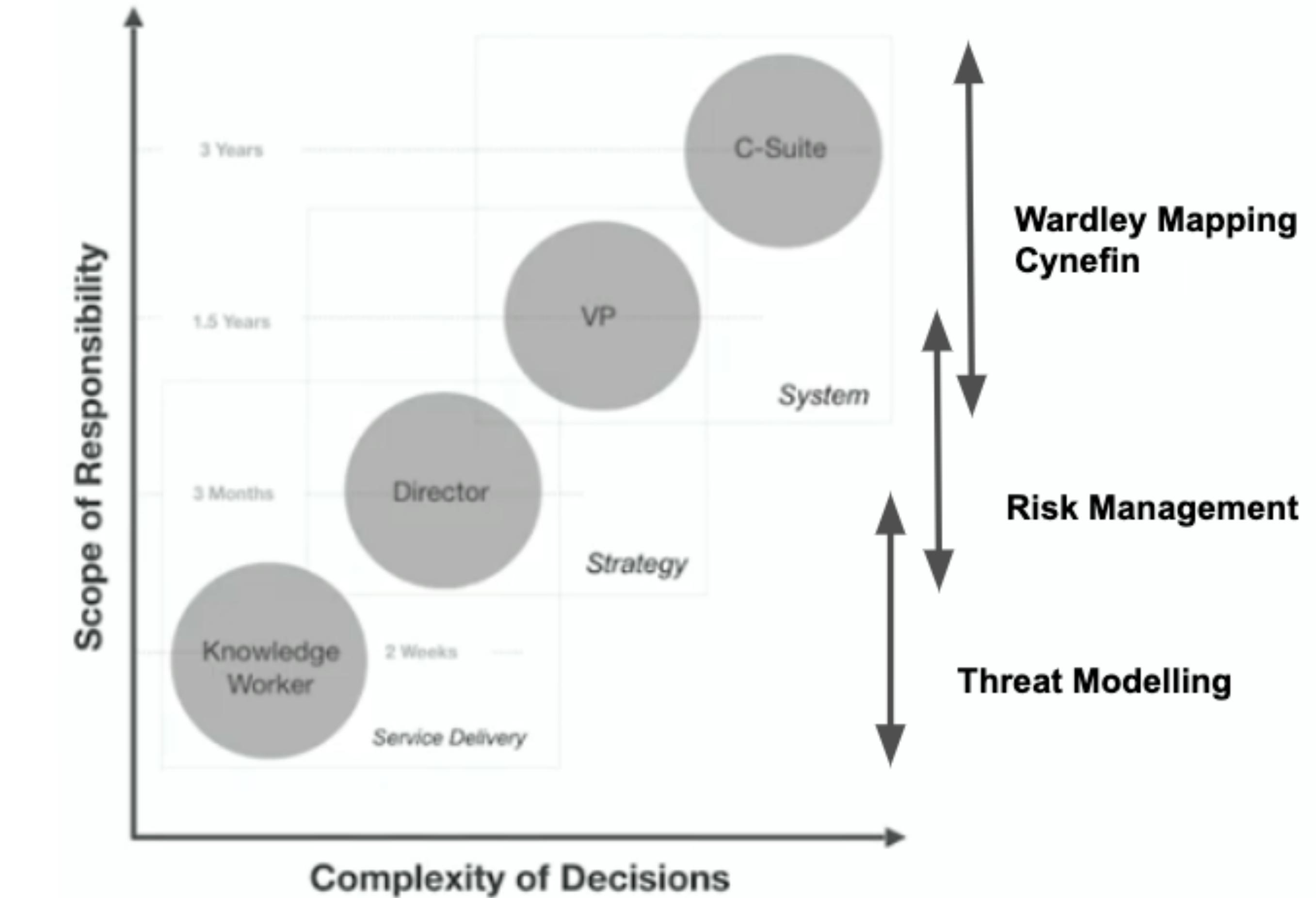
But how do we connect them ?

Threat Modelling



Risk Management

???



Strategies & Systemic Risk

C-Suite and VP

Risk Management

VP and Directors

Threat Modelling

Directors and Knowledge Workers

©Copyright 2020 Hysn Pte Ltd, All rights reserved.

Strategic Risk

Trusted

Reputable

Product / Service Risk

Hardened

Recoverable

Key Risk Indicators

Confidential system components > 65% baseline met

Confidential systems without tested recovery plan > 2

Threat

Security misconfiguration

Service recovery

Mitigation

Develop and apply baseline

Define and test recovery plan

Validation

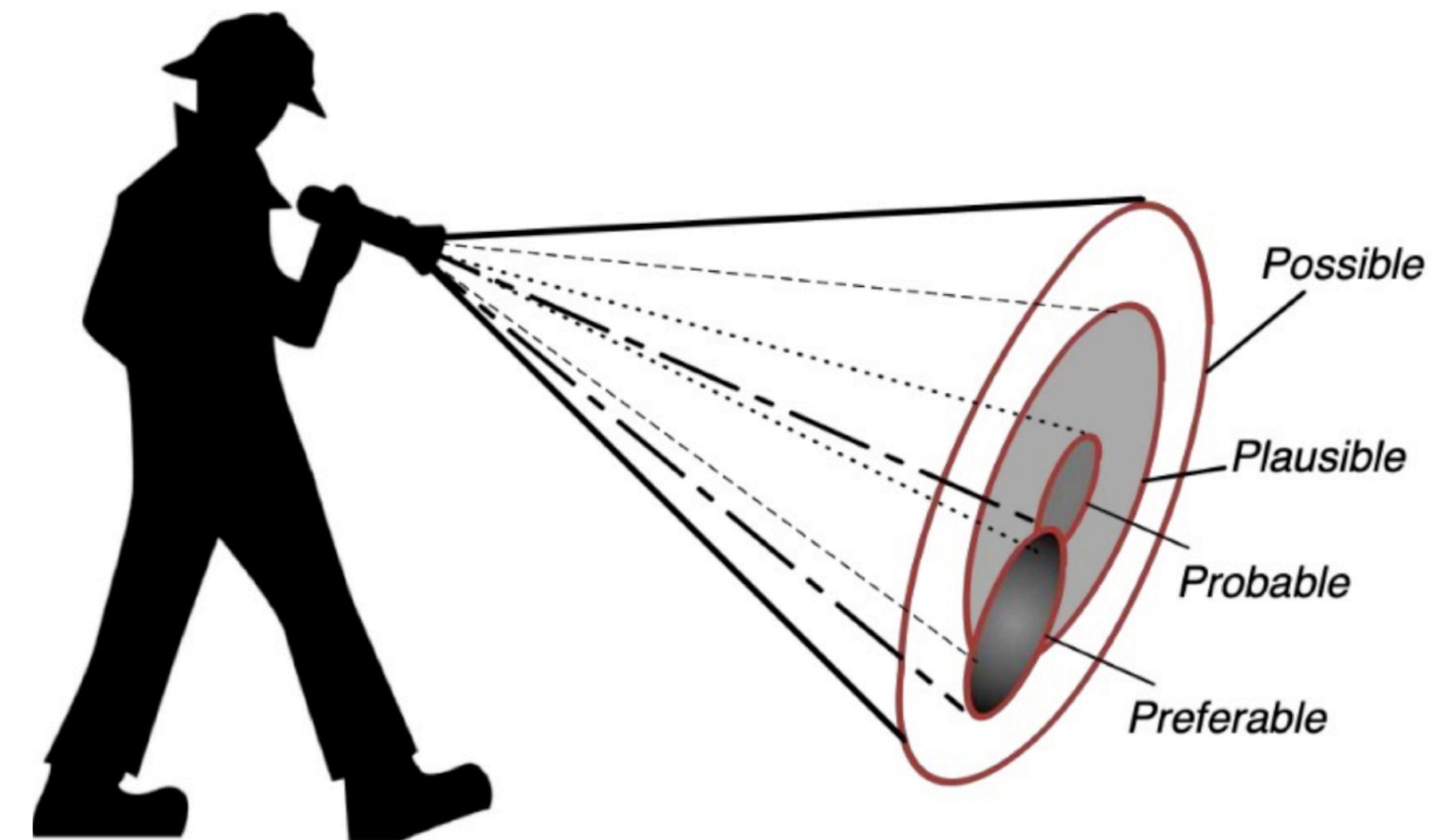
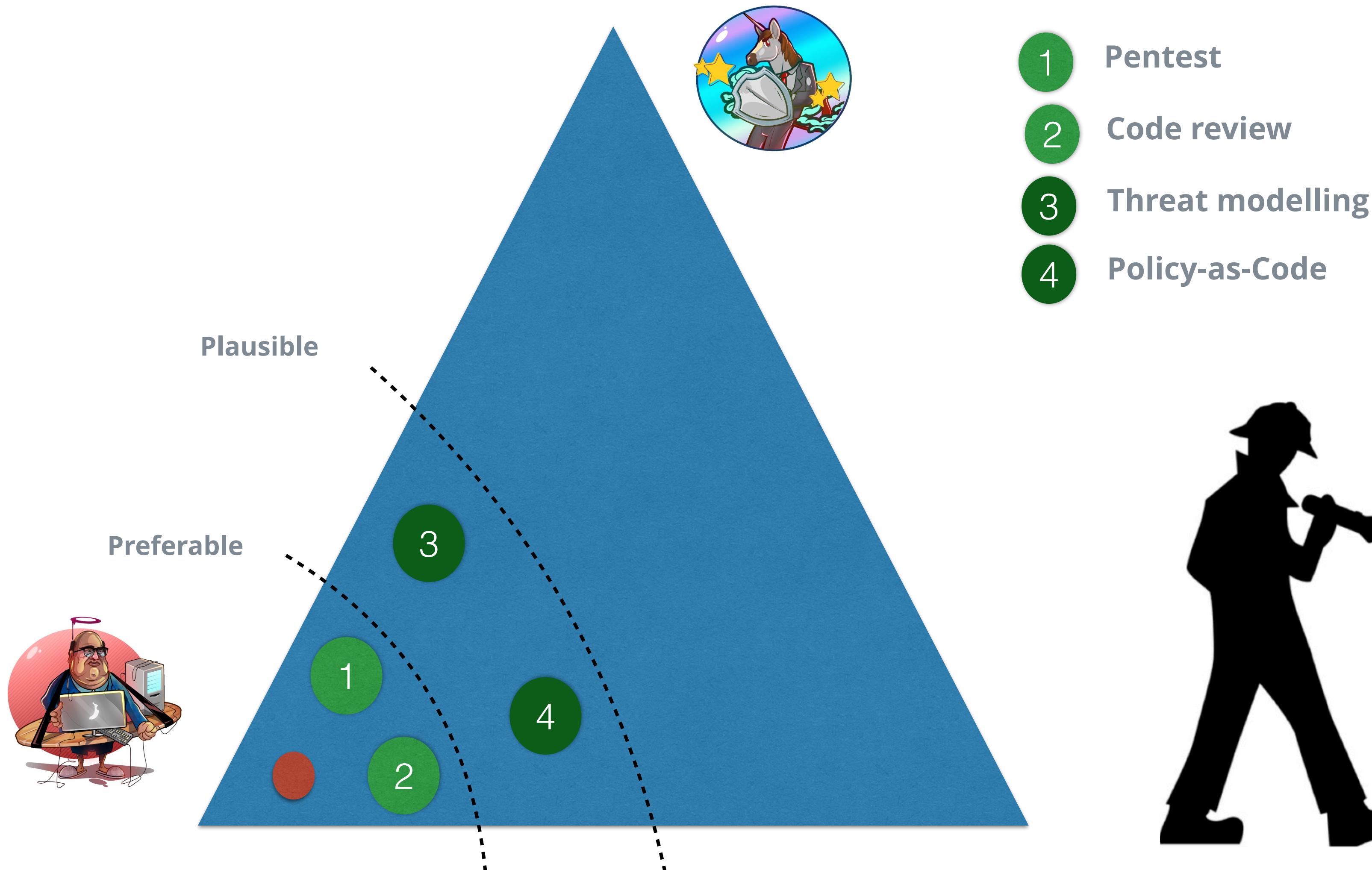
Compliance as Code

Scheduled testing with post-mortem

And process to connect them



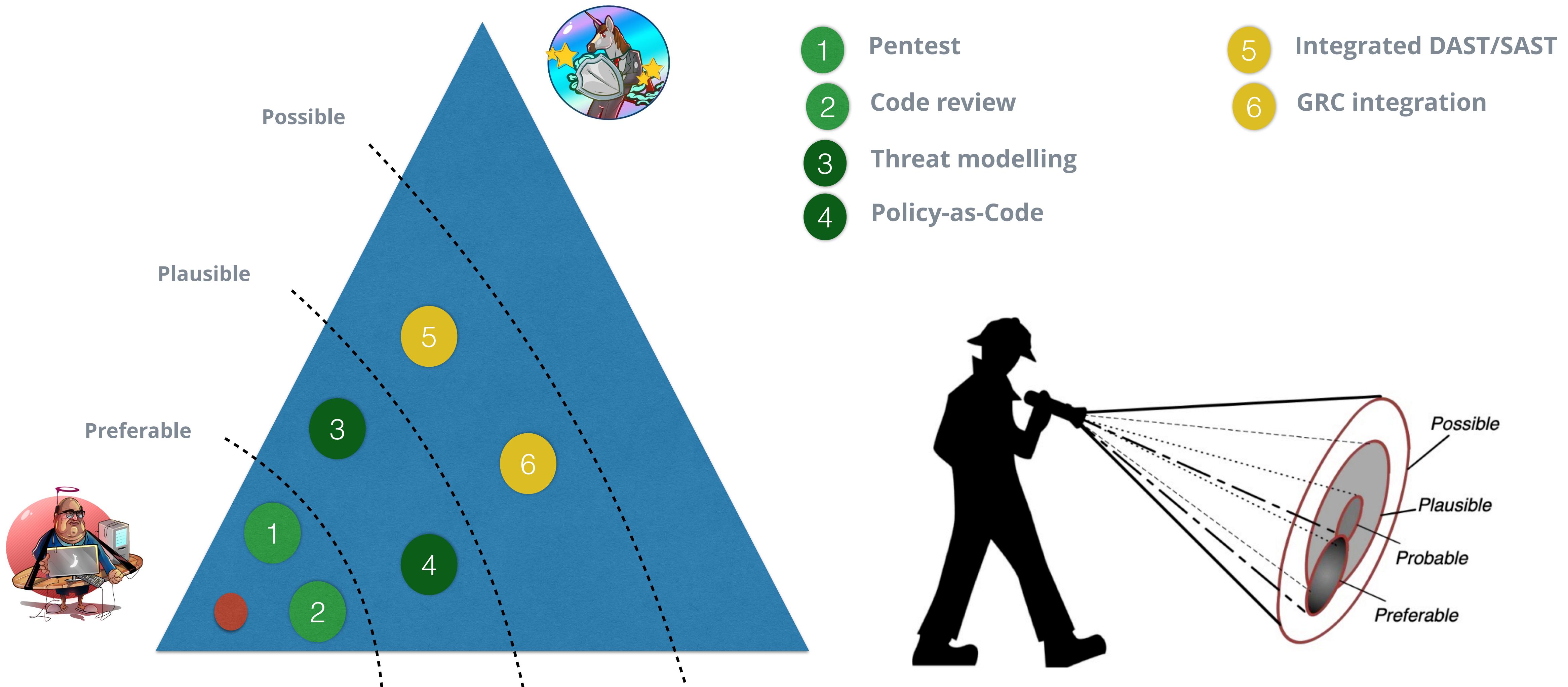
Meet them where they are



Constraint is TRUST in both teams and process/automation

Jabe Bloom @cyetain <https://www.slideshare.net/cyetain/three-frames-devopsdays-atl>

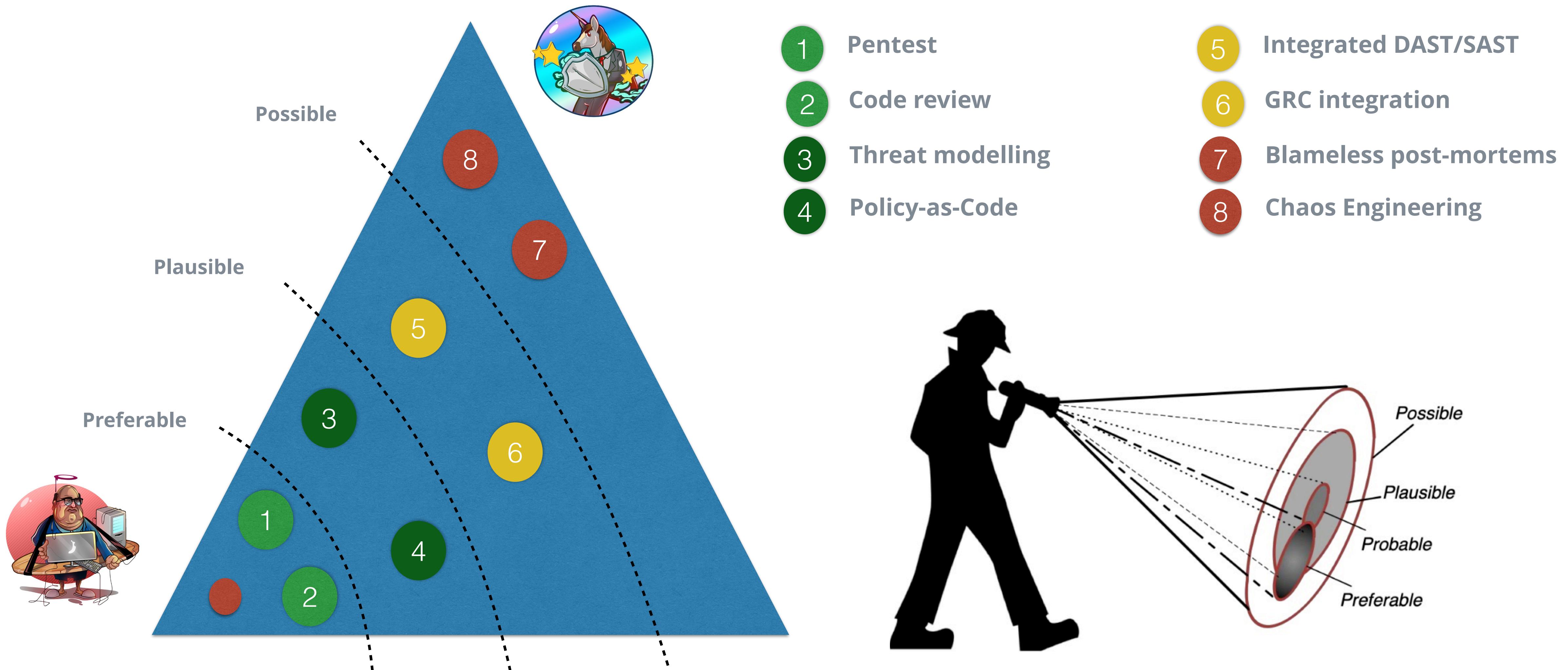
Meet them where they are



Constraint is TRUST in both teams and process

Jabe Bloom @cyetain <https://www.slideshare.net/cyetain/three-frames-devopsdays-atl>

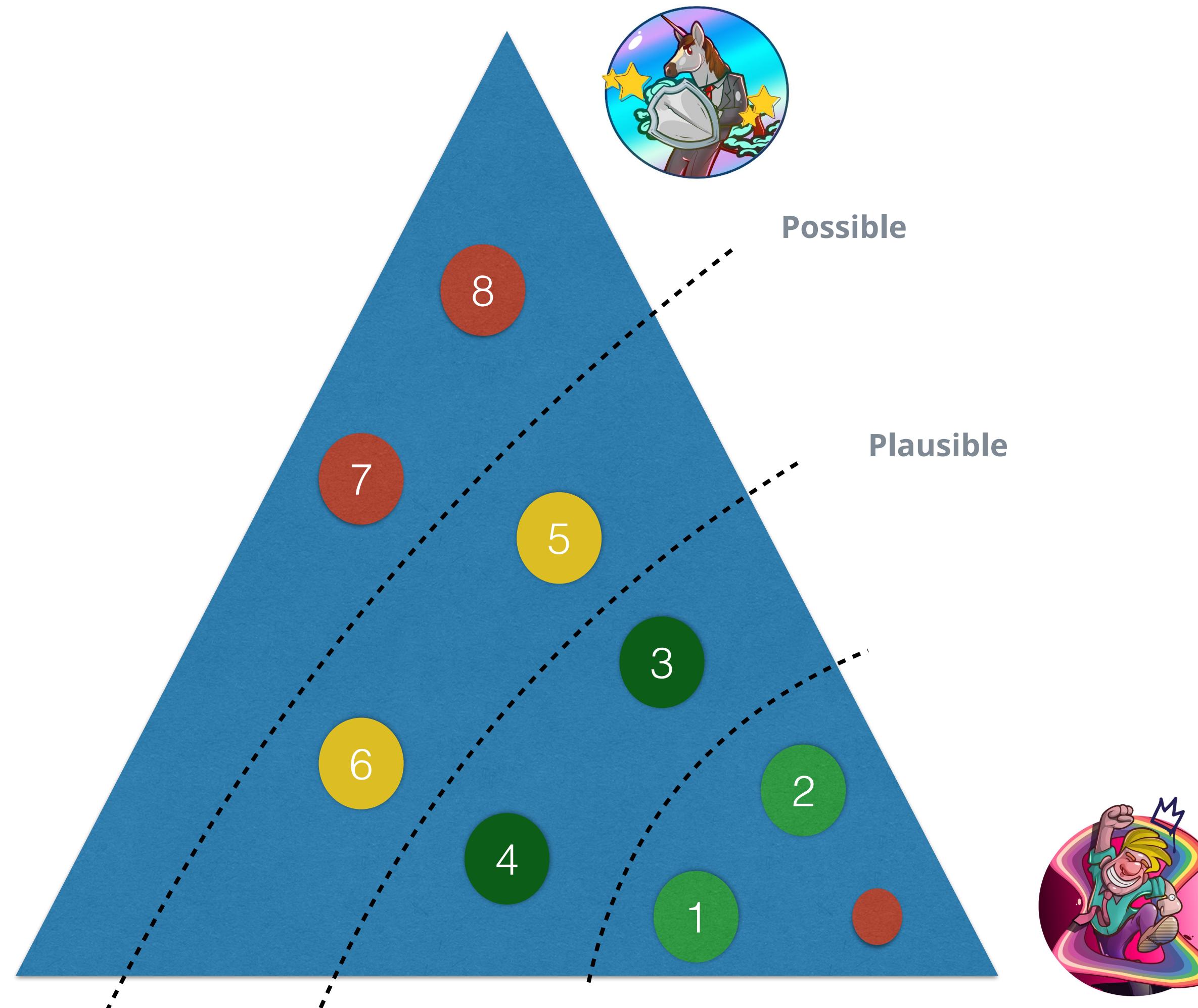
Meet them where they are



Constraint is TRUST in both teams and process

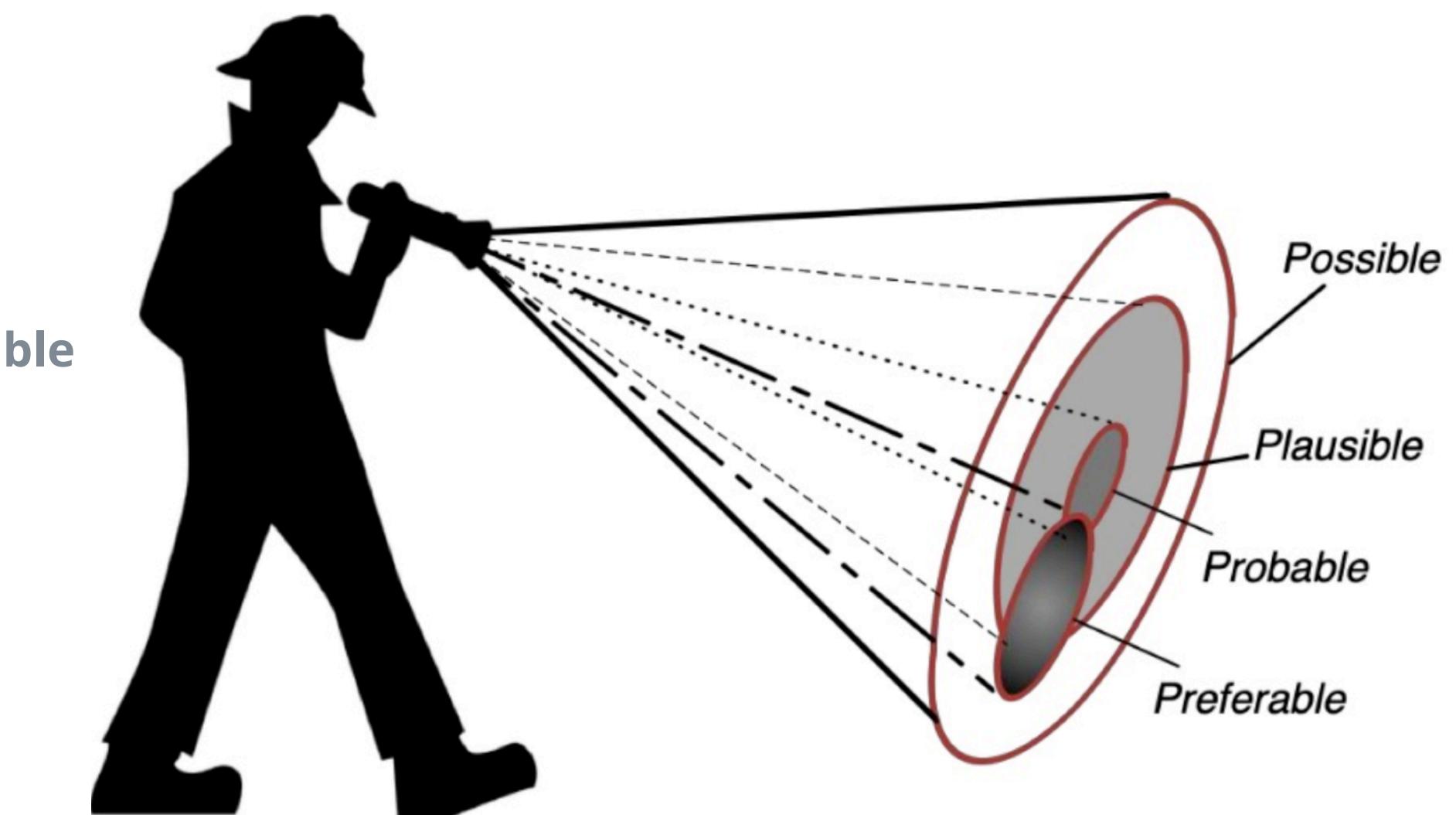
Jabe Bloom @cyetain <https://www.slideshare.net/cyetain/three-frames-devopsdays-atl>

Meet them where they are



- 1 TM with STRIDE
- 2 Process assurance
- 3 Baseline tooling
- 4 Policy-as-Code

- 5 TM w/ methodology
- 6 Scenario Testing
- 7 Risk analysis
- 8 Security backlog ownership

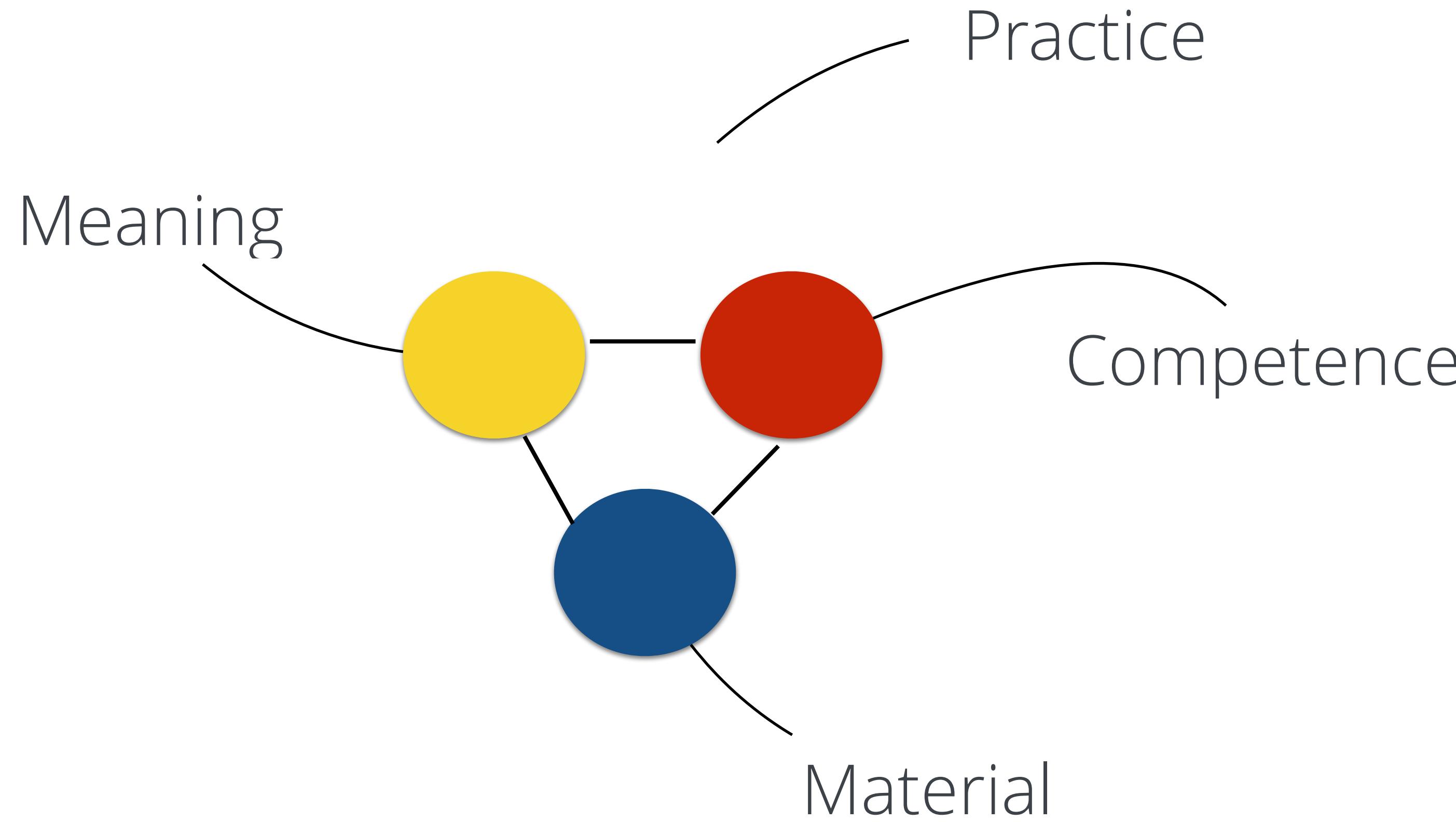


Constraints are Agency and Ownership

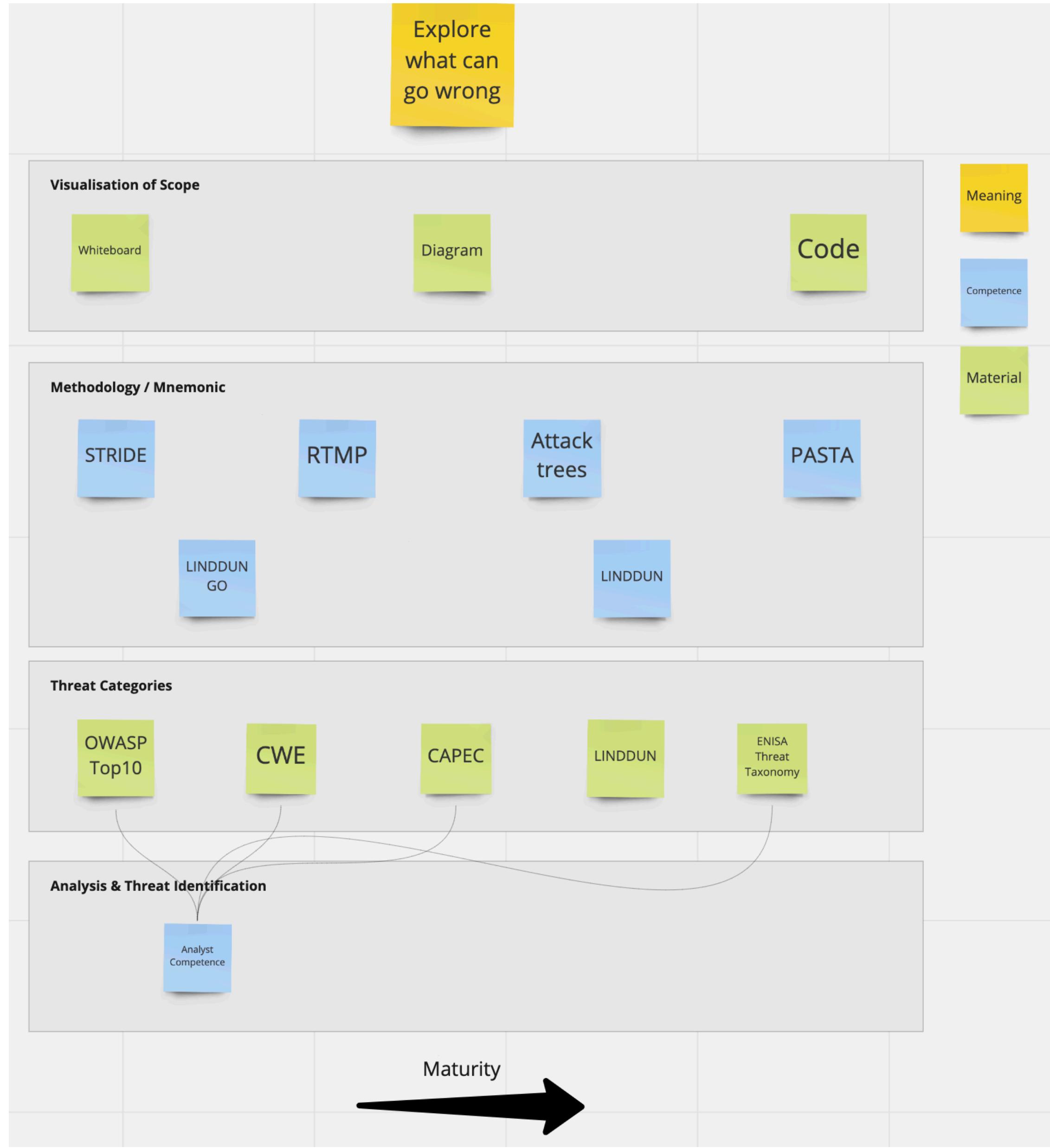
Jabe Bloom @cyetain <https://www.slideshare.net/cyetain/three-frames-devopsdays-atl>

**But looking at the artefacts is
not enough**

**We need to understand their
interactions**



“material, meaning and competence are **not just interdependent, they are also mutually shaping**” Elizabeth Shove



Evolve the Meaning of Practices

**“In a DevOps world, a Pentest
is not for finding security
issues.**

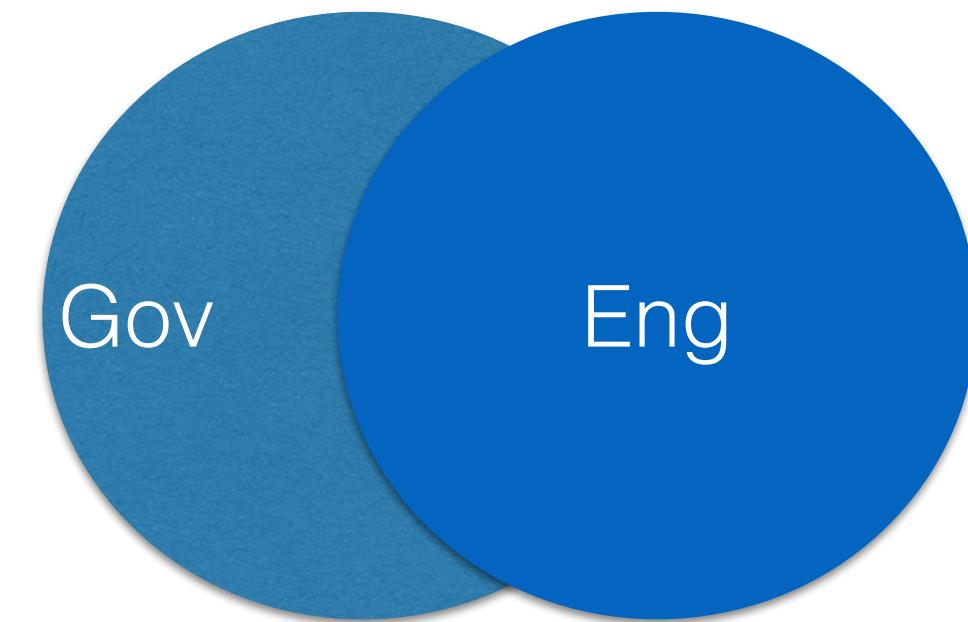
**It's to improve process”
Mohammed A. Imran**

Team interactions also evolve

**We need to change our “Team Topologies”
based on effectiveness and maturity of our
practices**

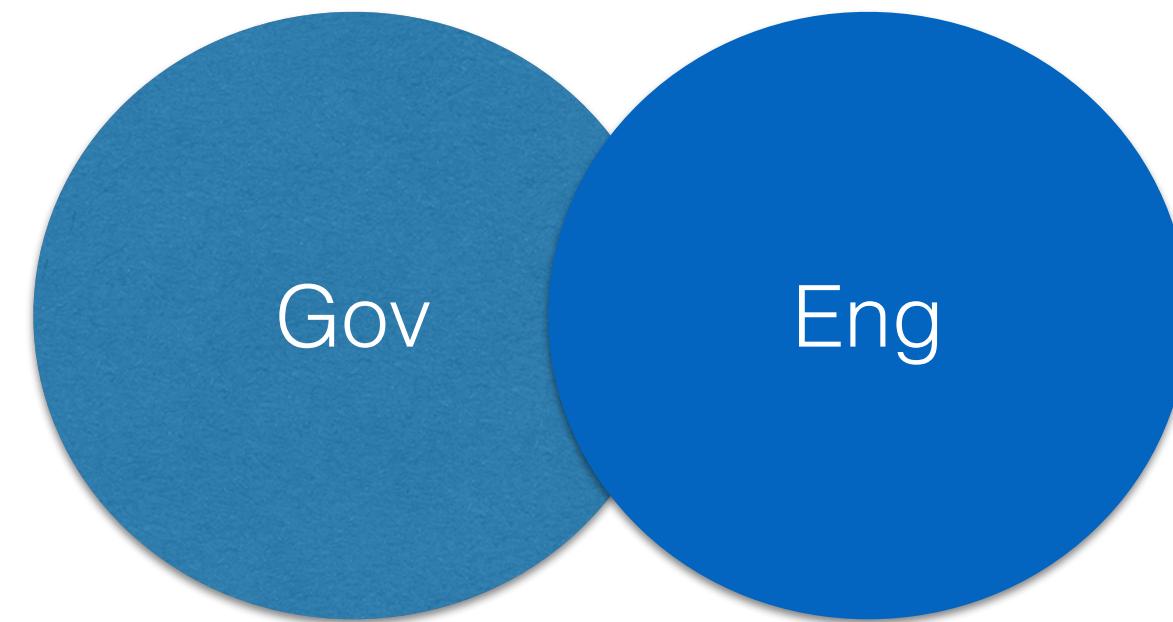
**Most fail as they assume shared mental
models where they don’t exist yet**

'Team Topologies' applied to Cyber Security



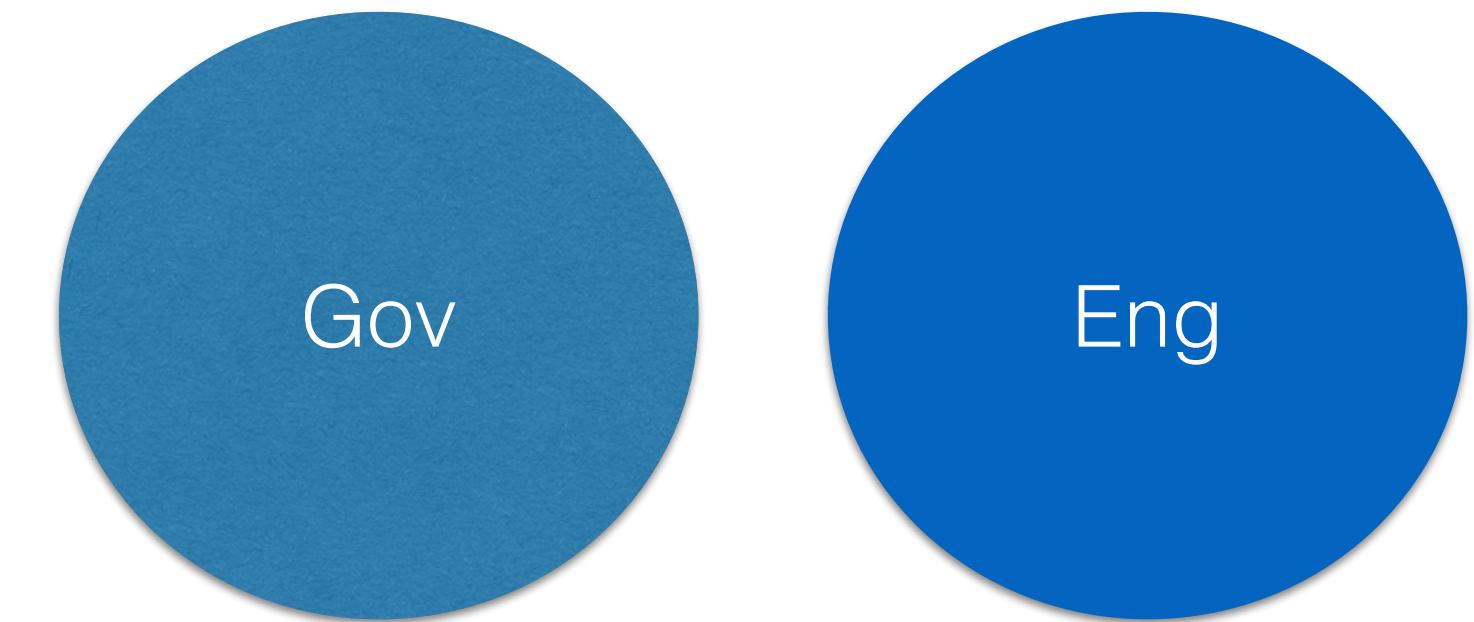
Collaboration

- Governance mapping (metadata management)
- Joint Threat modelling and Risk assessments
- Visibility and Reporting
- Management system updates



Facilitation

- Focus on cross-team training and contextual guidance
- Separate Threat modelling and Risk assessment sessions
- Boundary spanning more active between domains



X-as-a-Service

- Fully defined interfaces (Team or actual APIs)
- Informed/Supportive roadmaps
- Exception Management

Team Evolutionary Path

So, should YOU DevSecOps or not ?

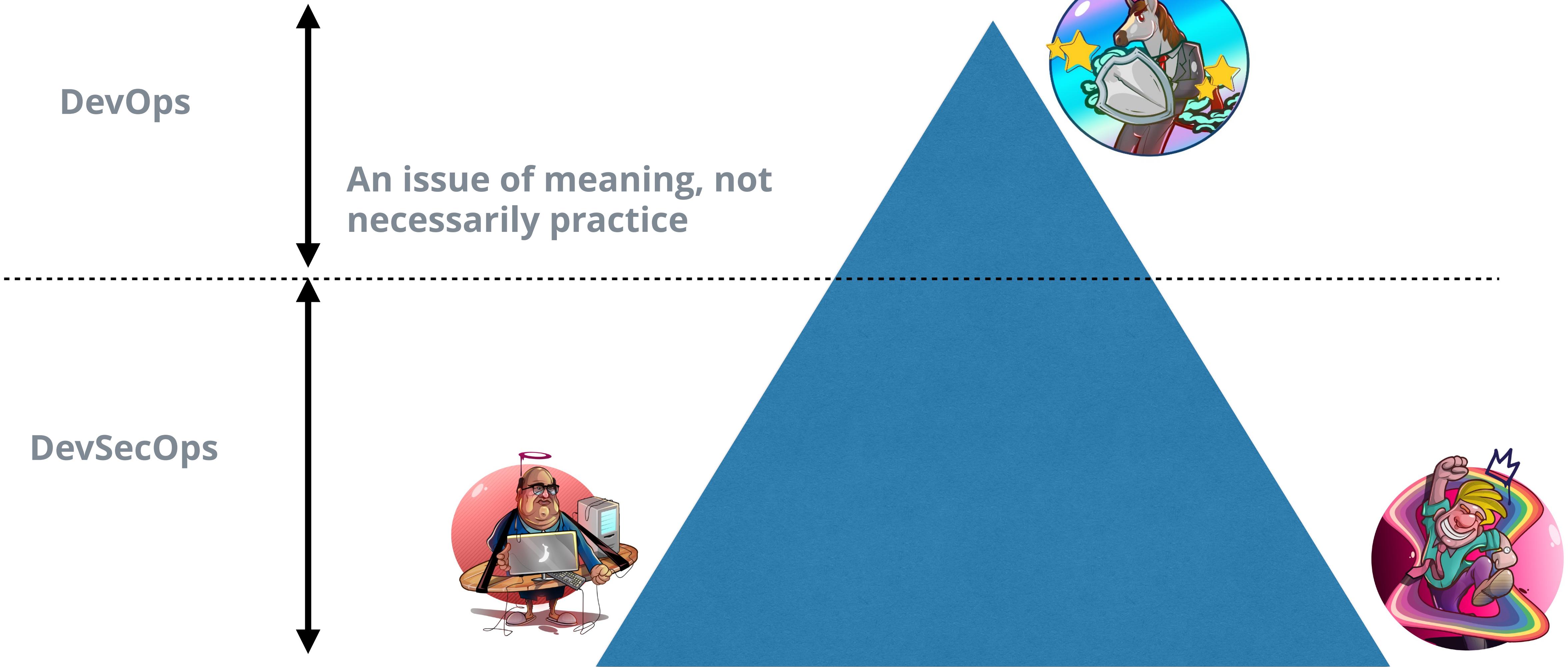
TRUST + AGENCY = (SECURE) DEVOPS



Naive to assume practices exist to make secure systems

But at some point, calling it something other than DevOps may be detrimental to culture and ownership

To DevSecOps or not ?



Use the 'label' strategically and whilst it provides value

Develop situational awareness to understand when it's right time to move on

If your strategy mentions the word 'DevSecOps' or Security in Devops and you're not

- helping your Governance teams benefit from short feedback loops and training them to understand DevOps
- Not increasing the agency and ownership of security across your Product or Project teams in THEIR language, not YOURS
- Not enabling the best possible Developer and Engineering Experience of Security you can afford
- Not actively trying to breakdown silo-ed barriers and connect governance systems

You're probably doing it wrong!

Q&A



Mario Platt

mario@practical-devsecops.com

Twitter: @madplatt

LinkedIn: [marioplatt](#)

Medium: @marioplatt



©Copyright 2020 Hysn Pte Ltd, All rights reserved

Pre-Summit training: <https://pre-summit-training-sessions.heysummit.com/>

Summit : <https://open-security-summit-2020.heysummit.com/>

DevSecOps: SAST, DAST, Compliance as Code

Strategy: Wardley Mapping, Cynefin, Team Topologies