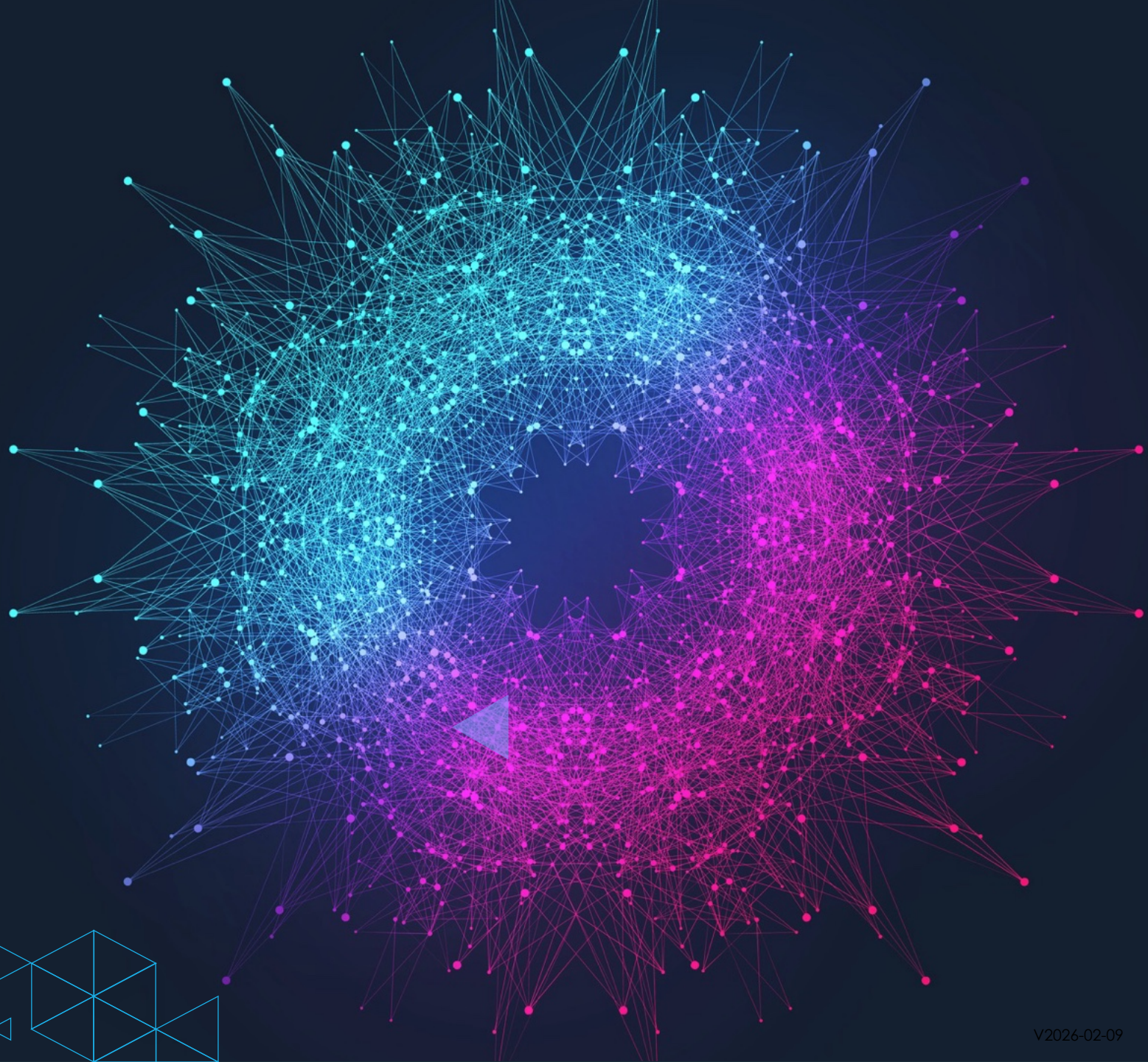
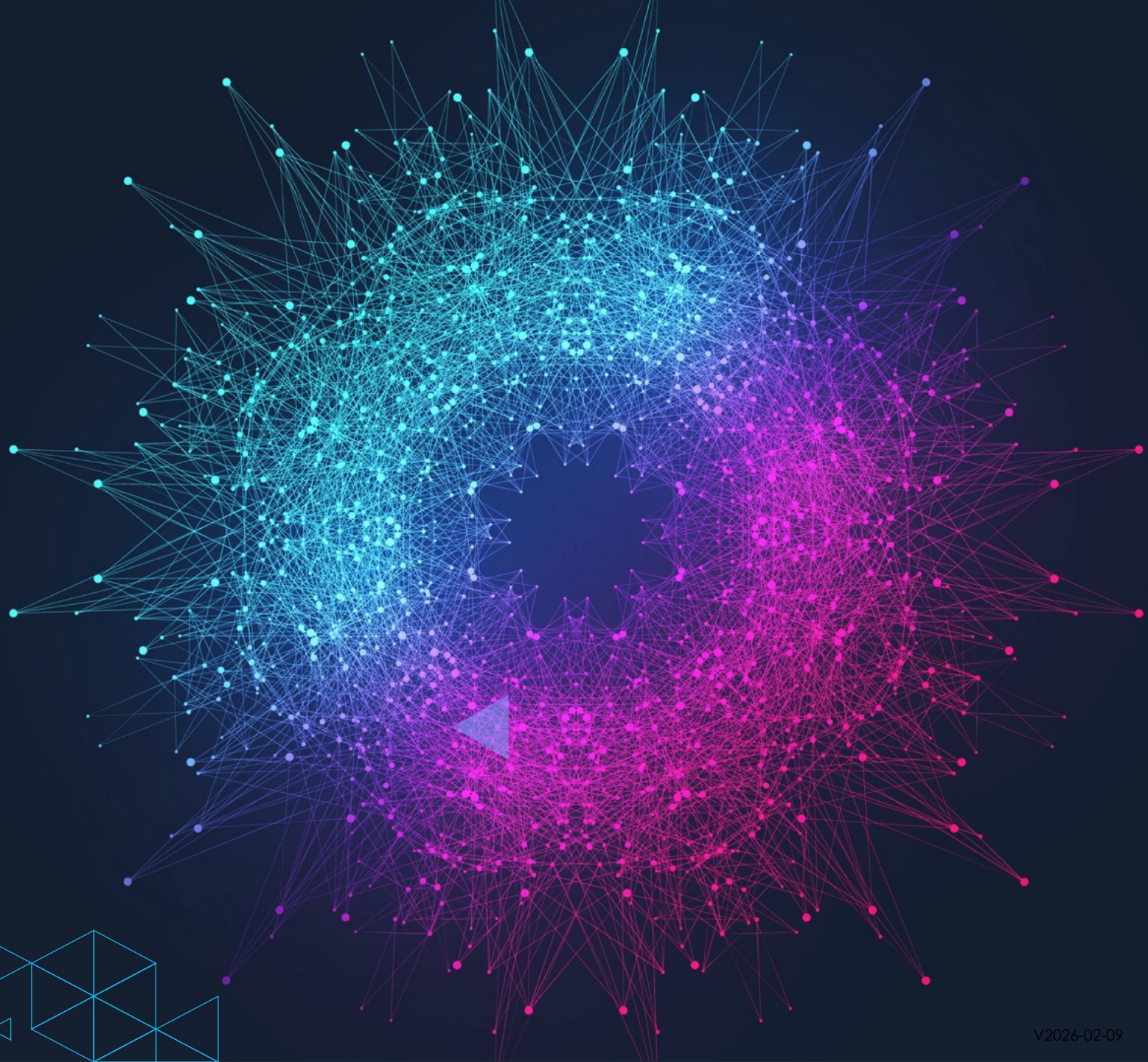
AISA

MELBOURNE



V2026-02-09

AISA
MELBOURNE
FEB 2026



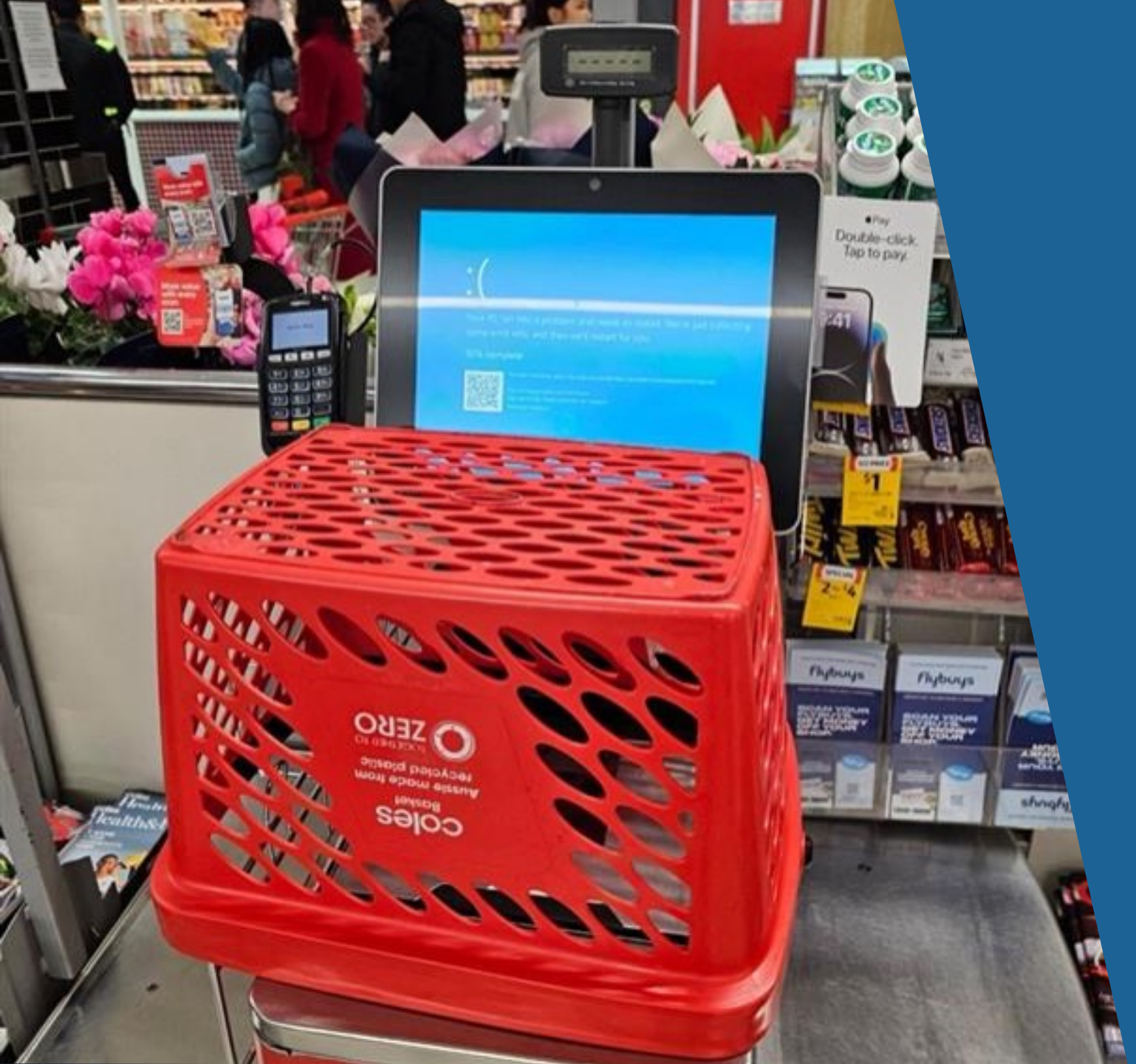
Better AppSec through better DevEx

Dan Ting

Principal Consultant, AppSec

CyberCX

V2026-02-09



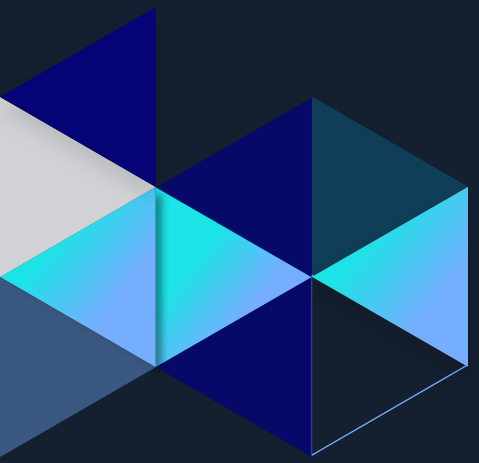
What if…
all software suddenly stopped working?

"Airports were left in chaos, supermarket check-outs started malfunctioning and journalists scrambled without the basic tools of the trade to report on an issue causing havoc worldwide."

https://www.abc.net.au/news/2024-07-20/what-happened-crowdstrike-global-outage-explainer/104122582

Organisations are software driven

AppSec failures, can have business continuity consequences.



abc.net.au/news/2019-12-04/opal-tower-builder-launches-new-3...

Incognito (5)

ABC NEWS iview listen

NEWS Search... Log in

Opal Tower builder blames engineer for cracks, $30m repair bill, court documents reveal

By Nick Sas

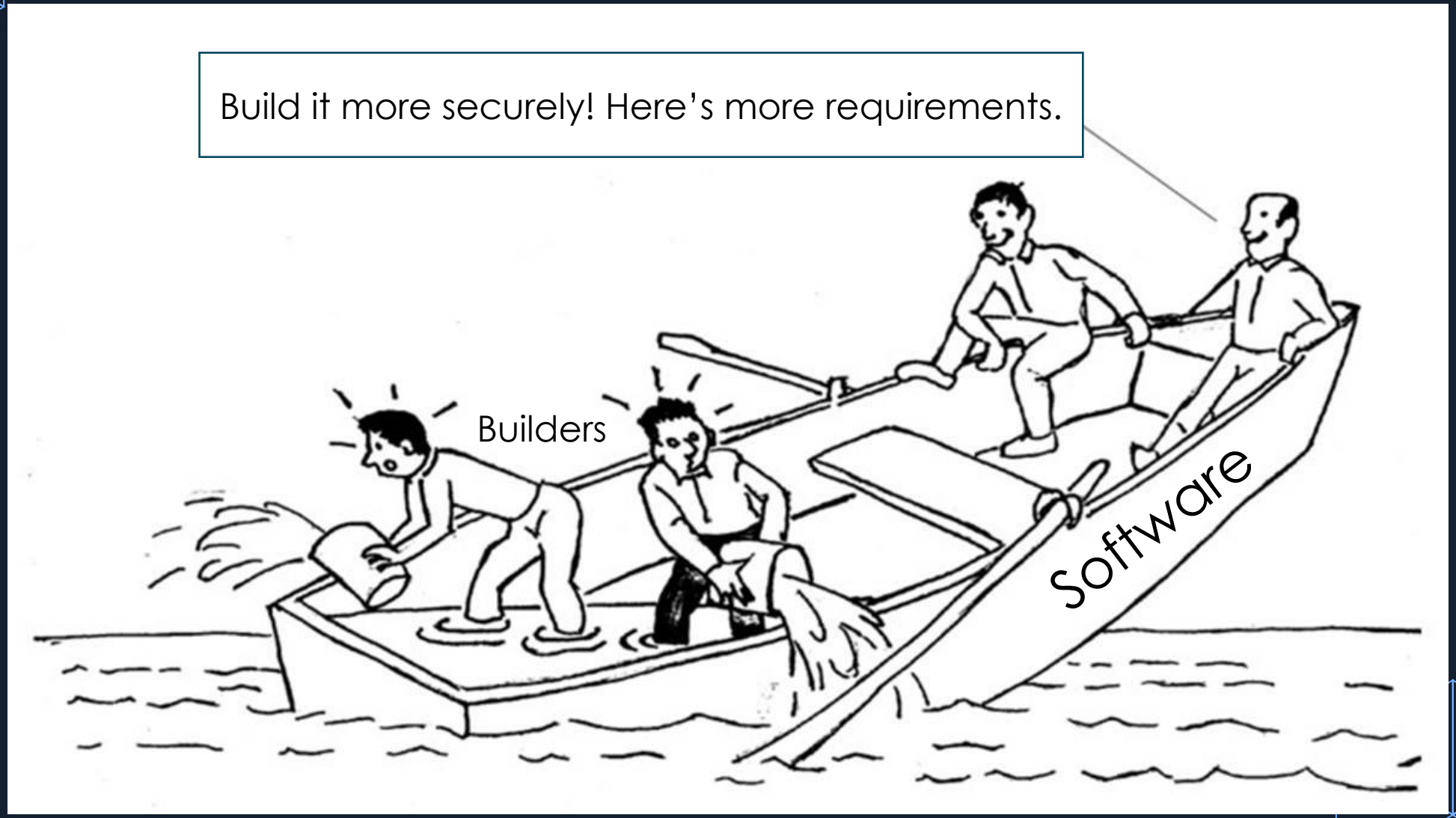
Construction and Real Estate Industry

Wed 4 Dec 2019

The Opal Tower saga has sparked questions about Sydney's construction industry. *(AAP: Mick Tsikas)*

abc.net.au/news/opal-tower-builder-launches-new-30m...

Share article


Build it more securely! Here's more requirements.
Builders
Software

“I want to be proud of the work that I do, and quality of systems that I build.”

Raise your hand if believe developers you know agree with this

It’s not a lack of will.

It’s often “other priorities”


DevEx


DevEx

Defining an Application Security Strategy

Challenges

- ! Siloed AppSec programs and teams
- ! Internal ‘PO’ based engagement models between Security and Technology
- ! Point in time assessments and reviews

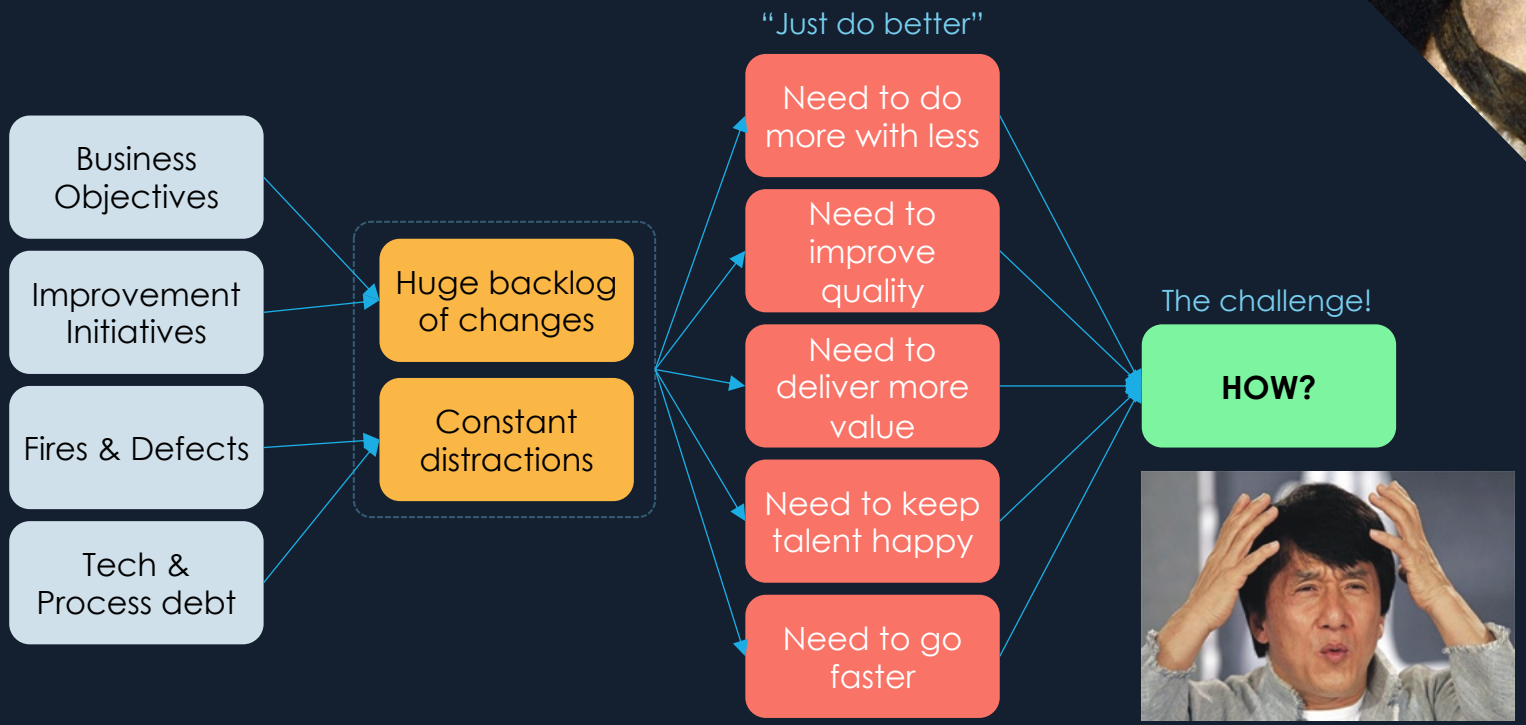
What’s working?

- ✓ Focus and integration with Developer Experience (DevEx)
- ✓ Re-defining the role of an AppSec Engineer
- ✓ Security team members embedded into Development Teams

Source: https://cybercx.com.au/resource/hack-report/

How do we improve AppSec through DevEx?

The Challenges



Source flowengineering.org

Sisyphus (1548–49) by Titian, Prado Museum, Madrid, Spain

Shift Perspective

Application Security

- The practice and capability of
- building safer, and more trustworthy* software, that is fit for purpose.

* reliable, available, resilient

~~Application~~
~~Security~~
Quality

- The practice and capability of
- building safer, and more trustworthy* ~~software~~ service/electronics, that is fit for purpose.

*reliable, available, resilient

A better Developer Experience (DevEx)

- Enabling higher performing teams = Better Application Security

IMPROVED STABILITY

DORA

IMPROVED THROUGHPUT

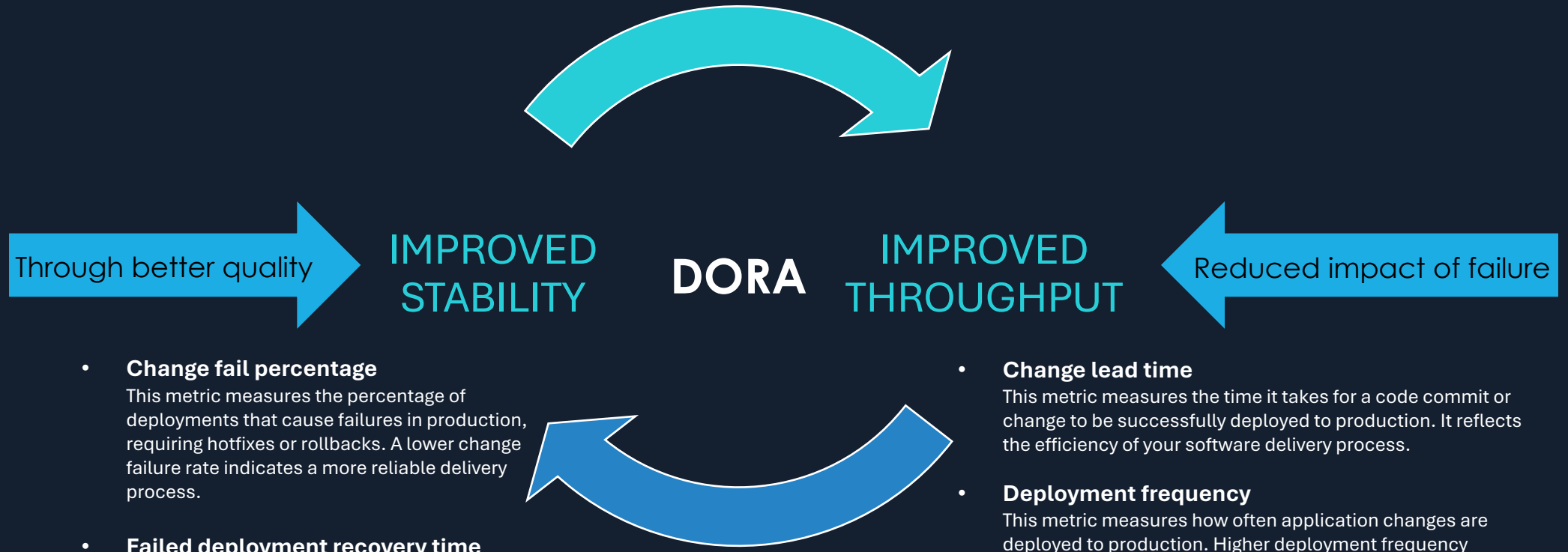
- **Change fail percentage**
  This metric measures the percentage of deployments that cause failures in production, requiring hotfixes or rollbacks. A lower change failure rate indicates a more reliable delivery process.
- **Failed deployment recovery time**
  This metric measures the time it takes to recover from a failed deployment. A lower recovery time indicates a more resilient and responsive system.

- **Change lead time**
  This metric measures the time it takes for a code commit or change to be successfully deployed to production. It reflects the efficiency of your software delivery process.
- **Deployment frequency**
  This metric measures how often application changes are deployed to production. Higher deployment frequency indicates a more efficient and responsive delivery process.

DORA | Get Better at Getting Better (https://dora.dev/)

A better Developer Experience (DevEx)

- Enabling higher performing teams = Better Application Security



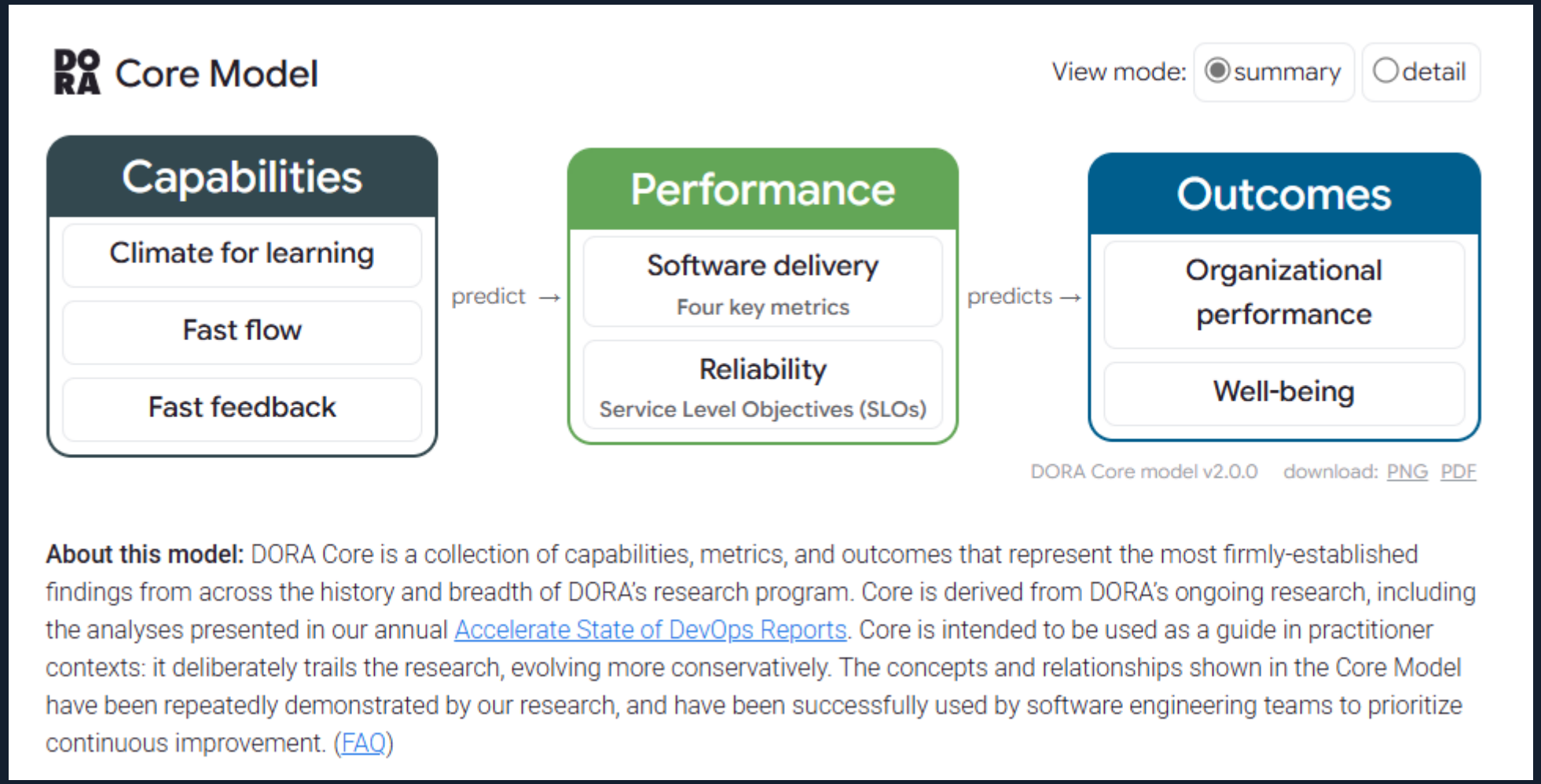
- **Change fail percentage**
  This metric measures the percentage of deployments that cause failures in production, requiring hotfixes or rollbacks. A lower change failure rate indicates a more reliable delivery process.
- **Failed deployment recovery time**
  This metric measures the time it takes to recover from a failed deployment. A lower recovery time indicates a more resilient and responsive system.

- **Change lead time**
  This metric measures the time it takes for a code commit or change to be successfully deployed to production. It reflects the efficiency of your software delivery process.
- **Deployment frequency**
  This metric measures how often application changes are deployed to production. Higher deployment frequency indicates a more efficient and responsive delivery process.

Better DevEx
= Better App Security

The terrain

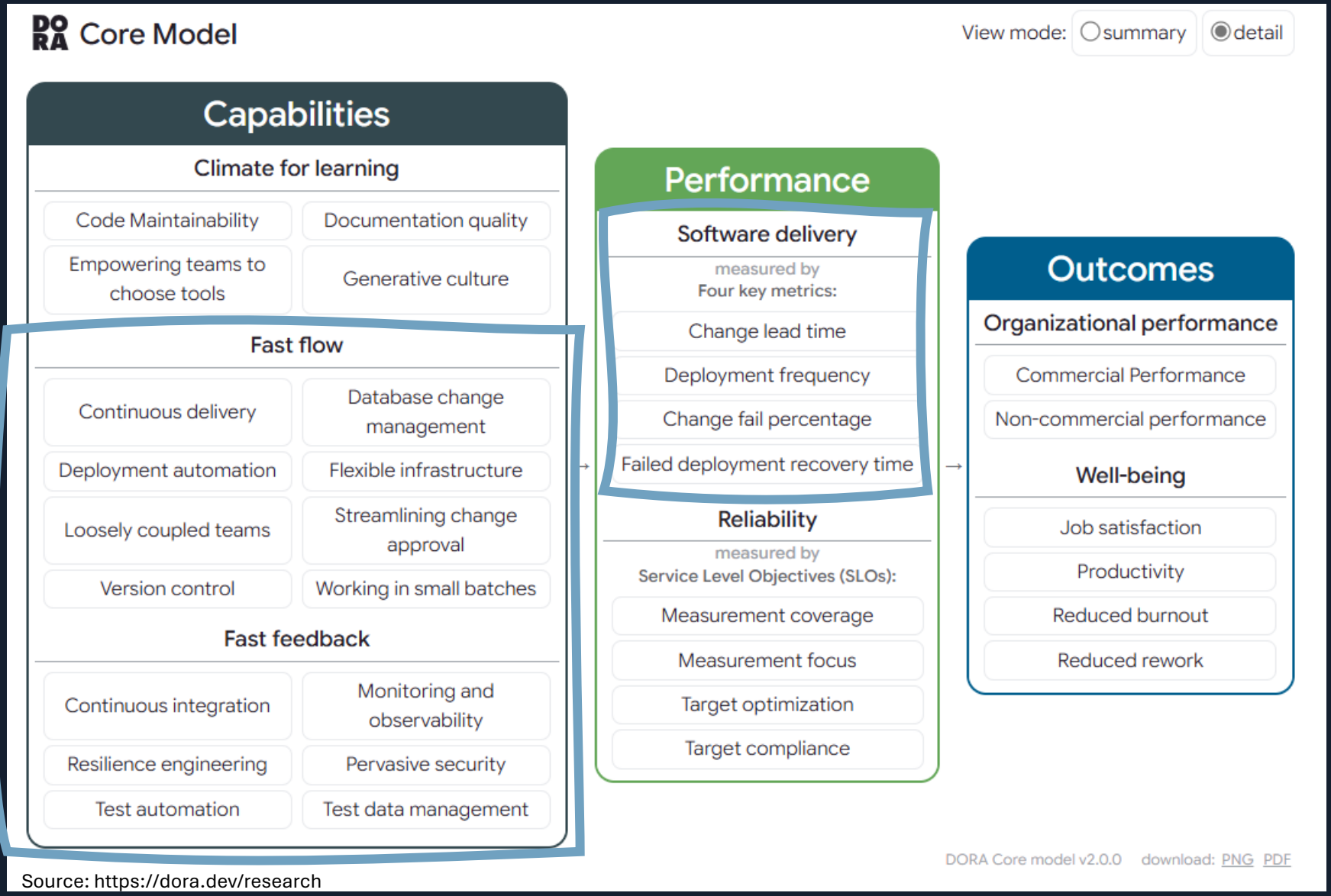
Developer Experience & DORA



*DORA stands for DevOps Research and Assessment. It was founded by Dr. Nicole Forsgren, Jez Humble, and Gene Kim

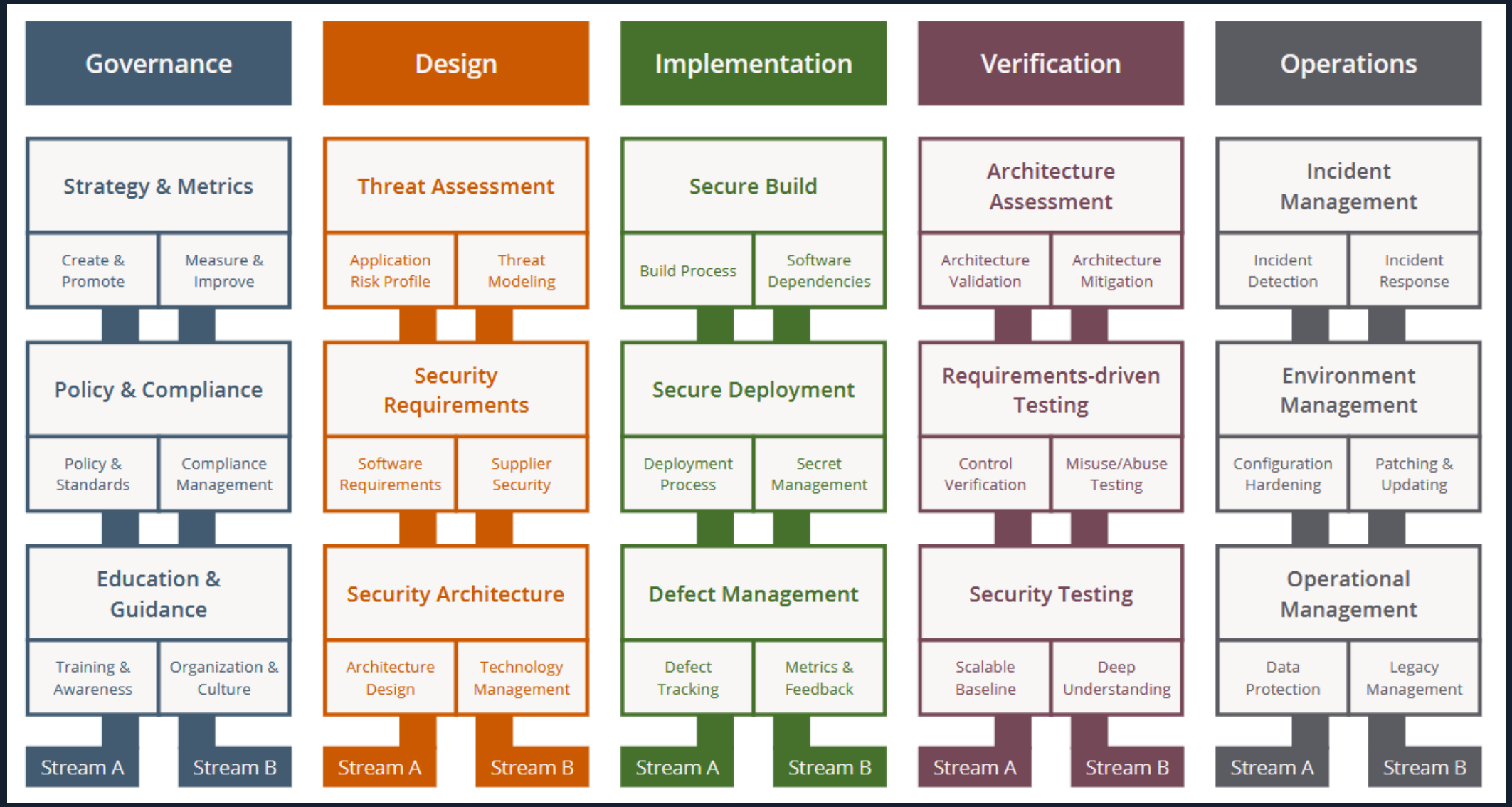
Source: https://dora.dev/research

Focused on the Software Development/Delivery Lifecycle



Source: https://dora.dev/research

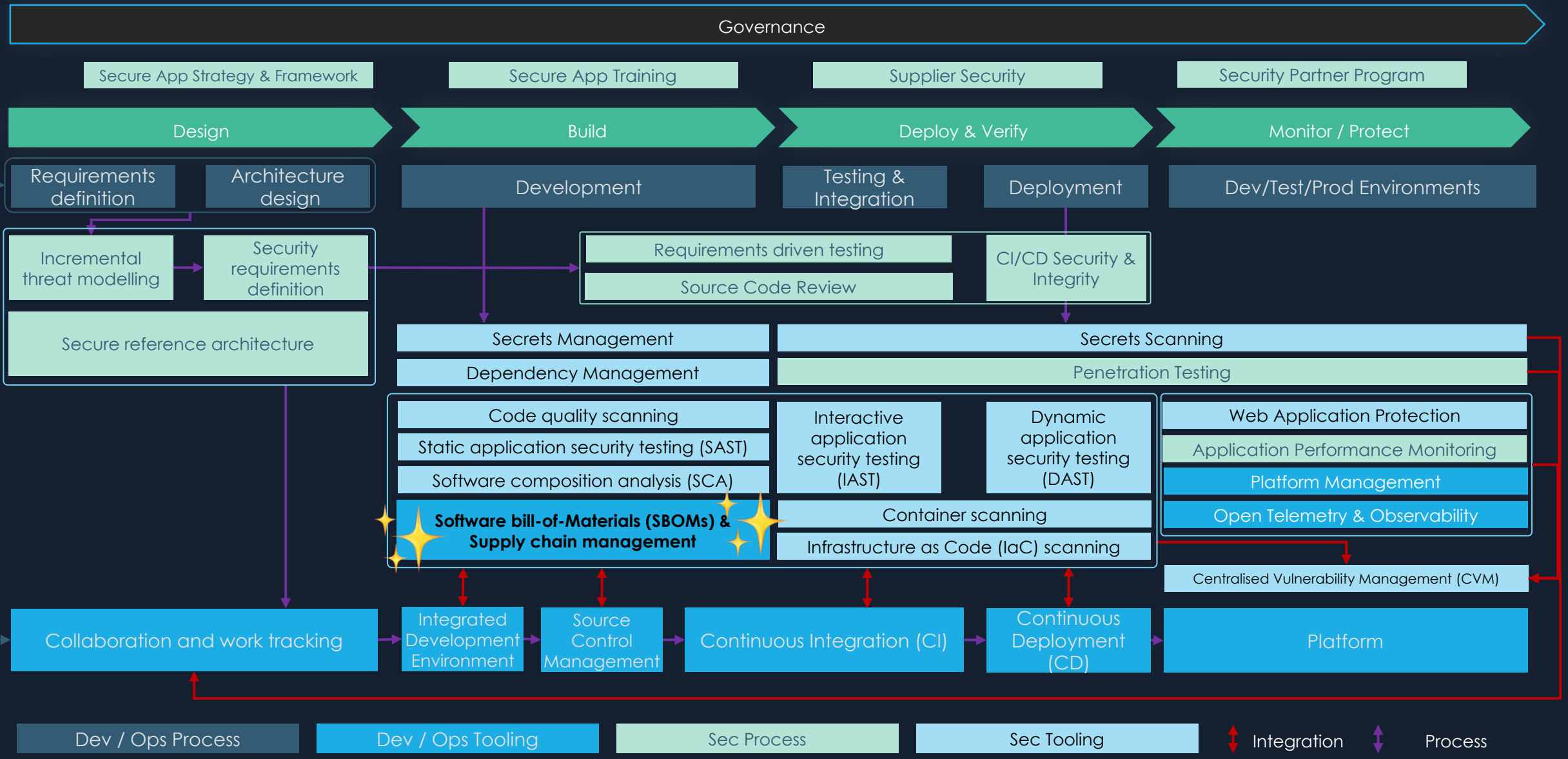
OWASP Software Assurance Maturity Model (SAMM)



https://owaspsamm.org/model/

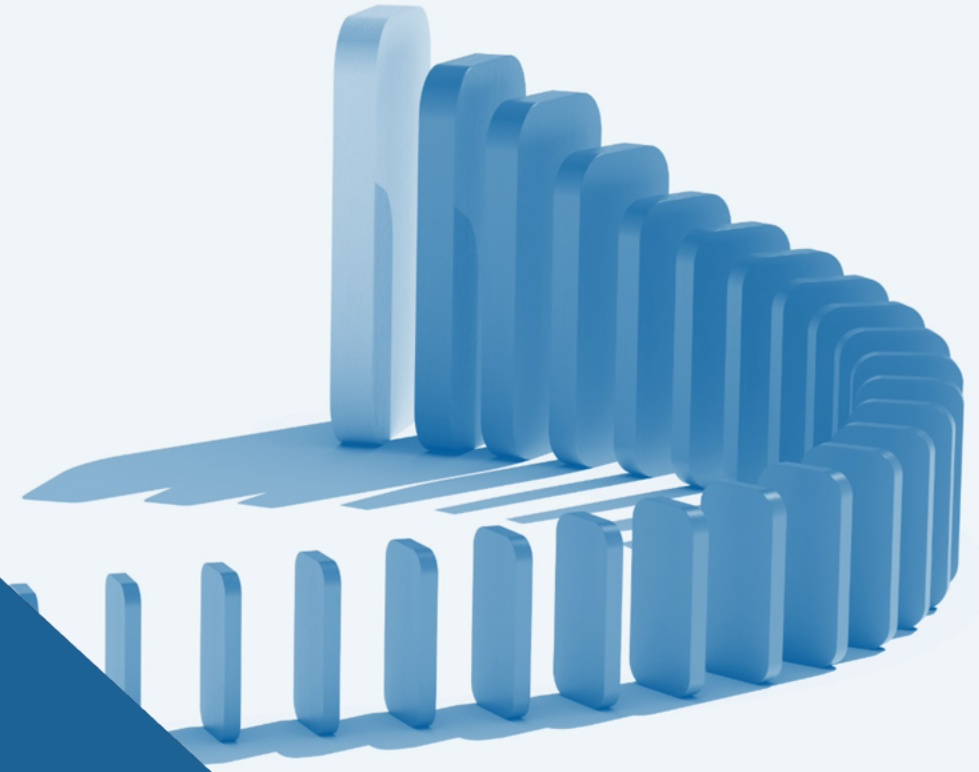
Note: I'm using SAMM as a model framework, but SSDF, DSOMM and others works too.

Example Secure Software Development Program



How do we start?

Efforts with OUTSIZED IMPACT



Drive better DevEx & AppSec

AppSec activities with outsized impact

01 **Abuse Stories (Threat Model)**
– Identify hazards & improve preparedness

02 **Prioritise with RICE**
– navigate tasks by scoring the backlog

03 **Build paved roads**
– build the flow you need & improve it.

[01]

Abuse Stories

Look ahead

What can go wrong?

So that we can plan & act accordingly

We already do this daily.

Crossing the road

As an _adversary_,
I want to ___ ,
so that ___ .

Write it down with your User Stories

Story-level Threat Modelling

- Focuses on a single feature or user story
- Generally, evil brainstorming = Abuse stories

For example, User Story → Abuse Story:

As a user, I want to update my personal information so that my account remains accurate and up-to-date.	As a malicious actor, I want to change someone's personal information so I can impersonate them or take over their account.
As a customer, I want to make a secure payment so I can complete my purchase with confidence and have my membership renewed.	As a malicious actor, I want to manipulate the payment process to avoid being charged but still have my membership renewed.
As a staff member, I want to access customer records so I can provide timely and personalised support.	As a malicious actor, I want to extract large volumes of customer data so I can sell it, leak it, or use it for other activities.

More Details : https://cheatsheetseries.owasp.org/cheatsheets/Abuse_Case_Cheat_Sheet.html

4 Key Questions

- What are we working on?
- What can go wrong?
- What are we going to do about it?
- Did we do a good enough job?

Source https://www.threatmodelingmanifesto.org

Agile (Story-Level) Threat Modelling

Sprint Planning

- Dev team plans for security work
- Dev team writes abuse stories
- Dev team asks ‘what could go wrong’
- Include threat modelling in Definition-of-Done

Daily Standup

- Include threat modelling progress as part of team updates

Backlog Grooming

- Review and decide if stories in backlog are security-related
- Flag security-related stories for sprint planning
- Prioritise (with RICE)

Sprint Review

- Present mitigations or changes based on threat modelling
- Gather feedback from security team

Retrospective

- Ask: Did we identify and mitigate threats early enough?
- What went well?
- What can be improved?

Agile (Story-Level) Threat Modelling

Sprint Planning

- Dev team plans for security work
- Dev team writes abuse stories
- Dev team asks ‘what could go wrong’
- Include threat modelling in Definition-of-Done

Daily Standup

- Include threat modelling progress as part of team updates

Backlog Grooming

- Review and decide if stories in backlog are security-related
- Flag security-related stories for sprint planning
- Prioritise with RICE

Sprint Review

- Present mitigations or changes based on threat modelling
- Gather feedback from security team

Retrospective

- Ask: Did we identify and mitigate threats early enough?
- What went well?
- What can be improved?

[02] Security Prioritisation using RICE

Infinite to do... what should we do?

Navigate

Competing Priorities

If everything is critical, where do we start?

What is RICE?

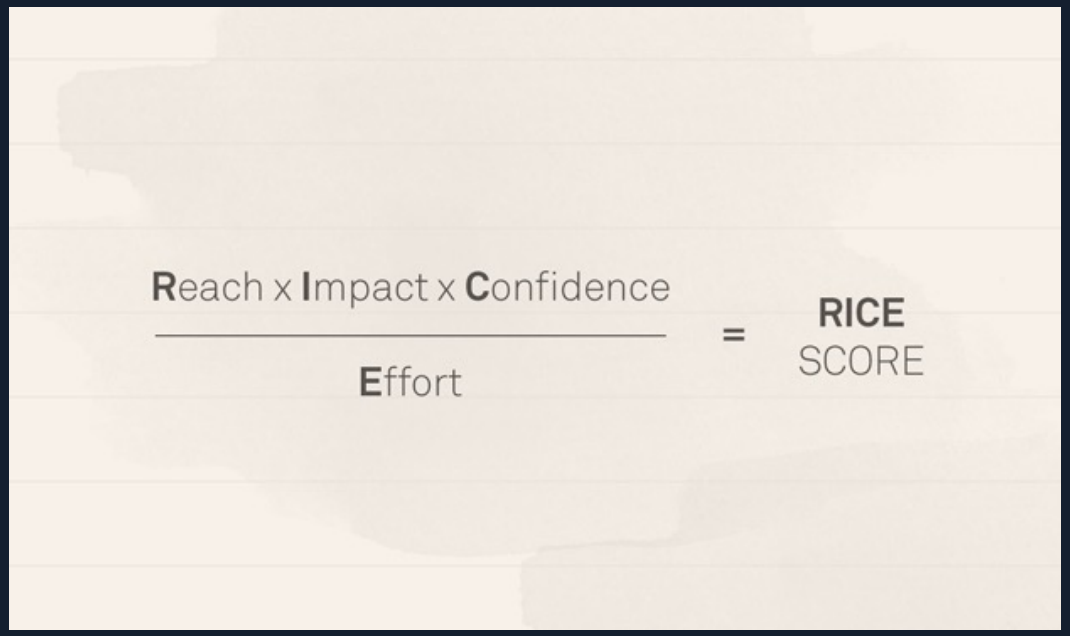
- A prioritisation scoring methodology by Intercom.
- Prioritisation is difficult, so, we use heuristics.
- **Reach** = number of people/events per time period
- **Impact** = a multiple-choice scale:
  3 for "massive impact",
  2 for "high", 1 for "medium",
  0.5 for "low", and finally 0.25 for "minimal"
- **Confidence** = level of confidence about your estimates
- 100% is "high confidence", 80% is "medium", 50% is "low"
- **Effort** = Estimate number of person-months/sprints

$$\frac{\text{Reach} \times \text{Impact} \times \text{Confidence}}{\text{Effort}} = \text{RICE SCORE}$$

score measures "total impact per time worked"

RICE = Risk / Effort

- Prioritise what to work on first using RICE Scoring Prioritisation Framework

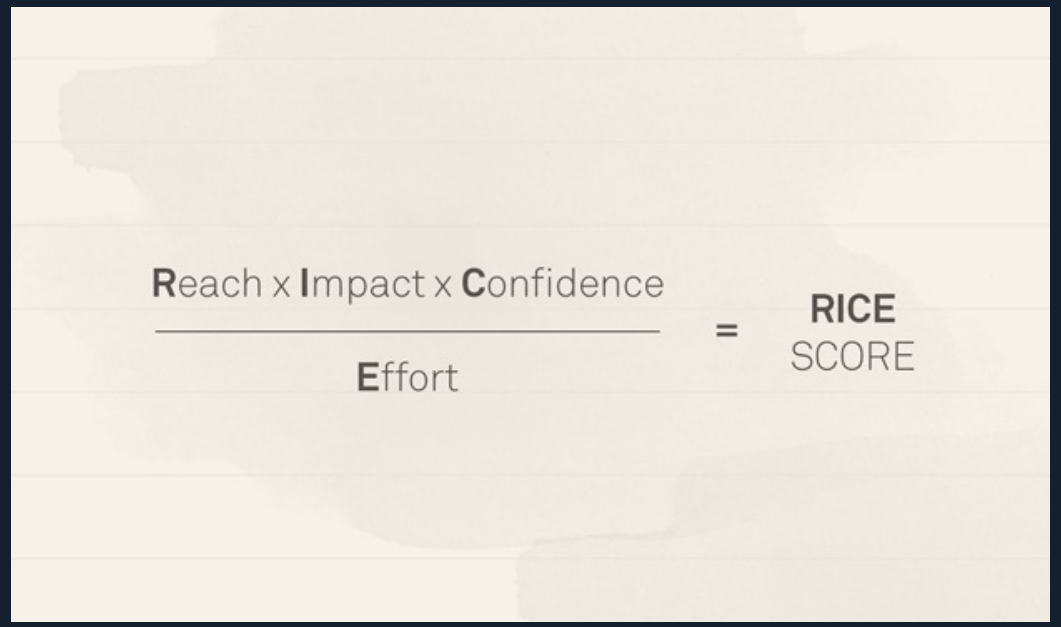


Likelihood		Consequence: Insignificant 1	Minor 2	Moderate 3	Major 4	Catastrophic 5
Is expected to occur in most circumstances	5 Almost certain	Medium	High	High	Extreme	Extreme
Will probably occur	4 Likely	Medium	Medium	High	High	Extreme
Might occur at some time in the future	3 Possible	Low	Medium	Medium	High	Extreme
Could occur but doubtful	2 Unlikely	Low	Medium	Medium	High	High
May occur but only in exceptional circumstance	1 Rare	Low	Medium	Medium	Medium	High

Risk Rating = Consequence **X** Likelihood

RICE = Risk / Effort

- Prioritise what to work on first using RICE Scoring Prioritisation Framework



We can use the same work prioritisation framework to manage and prioritise security risk

RICE Scoring Example

Task	Description	Reach (pax)	Impact	Confidence	Effort (Story Points)	RICE Score
Fix persistent XSS & CSRF in the card payment screen	Remediate penetration test findings. Our compliance team mentioned that if we don't fix it by EOM, our banks might block us from receiving payments until it's fixed.	1500	3	80%	5	720
Create send renewal invoice button	The finance & membership teams have asked us to create a "button" for them to send renewal invoices to one or many members.	1500	3	80%	10	360
Create membership self-service dashboard	The CEO wants members to be able to self-service manage their membership details & payments. The manual process is very painful, and we've received member complaints.	1500	3	50%	25	90
Fix bug in deployment bash script	Fix the bug in the deployment script that's causing a race condition, and sometimes failing the build. It's been a pain.	5	2	80%	2	4
Implement SAST & DAST into the CI/CD pipeline	Our security champion mentioned SAST & DAST tools to scan our code before it deploys to help catch potential security issues. Let's add these to our pull request checks, and as part of our staging, and prod pipelines.	5	2	100%	5	2
Backlog pruning & reorganisation	The backlog is a mess. Can we automate some processes & checklists?	5	2	80%	4	2
Implement dark mode for admin panel	The admin panel is very white and bright. We should have a dark mode option	5	1	100%	3	1.66666667

* For a fictitious generic membership management portal

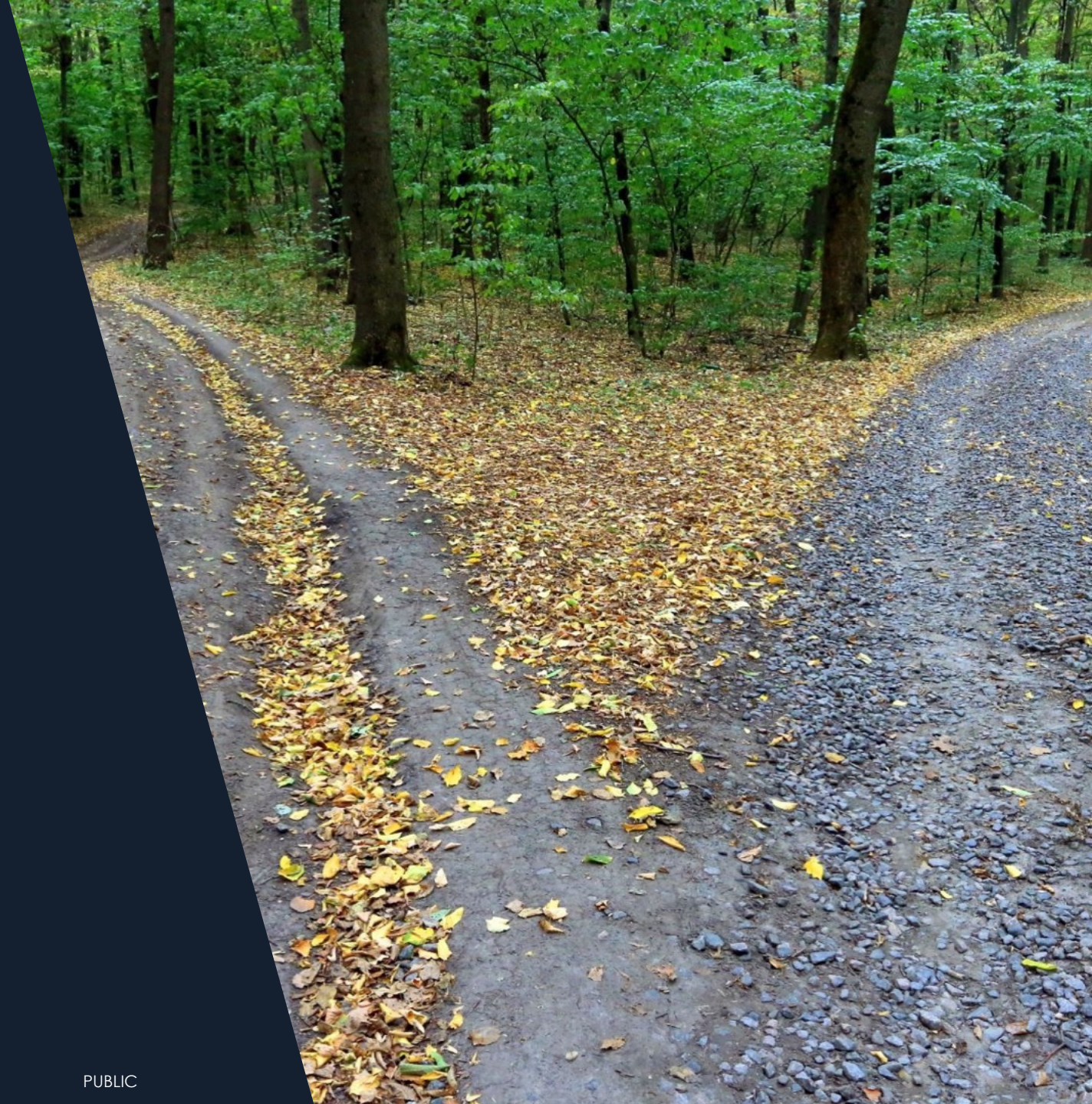
It penalises

- Niche initiatives – small reach, but large impact
- Innovation efforts – Lots of unknowns/uncertainty
- Strategic Alignment & Transformation
- Developer's Experience/Needs
- Technical Debt/Infrastructure/Housekeeping efforts

Fix bug in deployment bash script	Fix the bug in the deployment script that's causing a race condition, and sometimes failing the build. It's been a pain.	5	2	80%	2	4

RICE = choose between product features!

- **Not general prioritisation.**
- **But, we can adapt it.**



Adapted RICE

$$\frac{\text{Reach} \times \text{Impact} \times \text{Confidence}}{\text{Effort}} = \textbf{RICE Score}$$

RICE Score =

(Reach
× Impact
× Confidence
× Strategic
× Housekeeping
× Passion
) / Effort

- 1-Self, 10-team, 100X-population
- 5-level scale – 1 Very Low, 5 V. High
- 5-level scale – 1 Very Low, 5 Certain
- Each alignment = 3X
- Is housekeeping => 10
- Passion pet tasks => 8 (limit 1/pax)
- (however you measure effort. e.g. person-days)

*This is only an example. You will need to work with your team to experiment & build appropriate metrics.

Example Adapted RICE

Task	Description	Reach (pax)	Impact	Confidence	Effort (Story Points)	Strategic	House-keeping	Passion	RICE Score
Fix persistent XSS & CSRF in the card payment screen	Remediate penetration test findings. Our compliance team mentioned that if we don't fix it by EOM, our banks might block us from receiving payments until it's fixed.	200	5	4	5	6	10		48000
Fix bug in deployment bash script	Fix the bug in the deployment script that's causing a race condition, and sometimes failing the build. It's been a pain.	10	3	4	2		10		600
Create membership self-service dashboard	The CEO wants members to be able to self-service manage their membership details & payments. The manual process is very painful, and we've received member complaints.	200	4	3	25	6			576
Create send renewal invoice button	The finance & membership teams have asked us to create a "button" for them to send renewal invoices to one or many members.	100	3	4	10	3			360
Backlog pruning & reorganisation	The backlog is a mess. Can we automate some processes & checklists?	10	2	5	4		10		250
Implement SAST & DAST into the CI/CD pipeline	Our security champion mentioned SAST & DAST tools to scan our code before it deploys to help catch potential security issues. Let's add these to our pull request checks, and as part of our staging, and prod pipelines.	10	2	4	5	3			48
Implement dark mode for admin panel	The admin panel is very white and bright. We should have a dark mode option	1	3	5	3			8	40

[03]
Build
Paved Roads

Accelerate


How fast can you go?


Guard rails
& paved roads

The 3 ways

from The Phoenix Project book

#1 - Flow/Systems Thinking

- Optimize the flow of work from left to right—maximize the flow of value from development (Dev) to operations (Ops) to the customer.
- Key Practices:
  - Reduce lead time and cycle time.
  - Automate repetitive tasks (e.g., CI/CD pipelines).
  - Use tools to visualize work (Kanban boards, flow diagrams).
  - Abuse/Adversary Stories
  - Foster collaboration between all teams involved in the value stream.

#2 - Amplify Feedback Loops

- Enable and shorten feedback loops to detect problems as soon as they occur and prevent them from propagating or recurring.
- Key Practices:
  - Automate testing at all stages of the pipeline.
  - Build telemetry and monitoring into systems for real-time feedback.
  - Postmortems to analyze failures and improve systems/processes.

#3 - Culture of Continuous Learning & Experimentation

- Create a culture that values experimentation, learning from failures, and constant improvement.
- Key Practices:
  - Conduct "blameless" postmortems and retrospective.
  - Promote a mindset of hypothesis-driven development and experimentation.
  - Encourage innovation through hackathons, R&D days, and pilot projects.
  - Design systems that are resilient to failure.

Yes, The Unicorn Project, DevOps Handbook, and Accelerate are great reads too

Secure by Design & Secure by Default

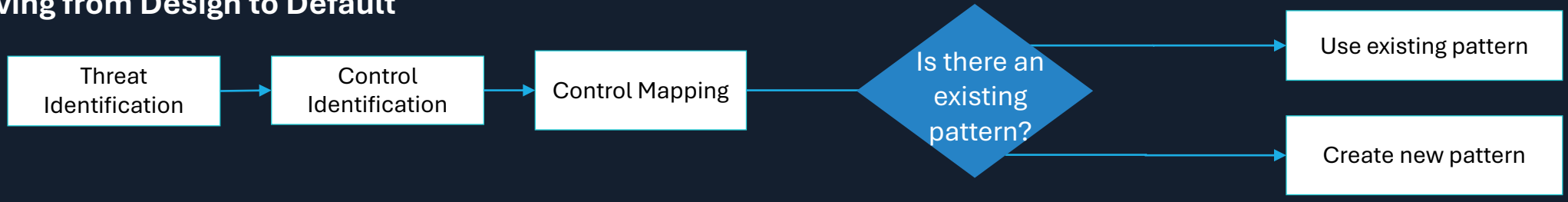
Challenges

- ! Repetitive spreadsheets detailing generic threats and controls
- ! Lack of documentation of architecture and security decision making
- ! Security controls “built from scratch” every time

What’s working?

- ✓ Culture of always asking “what can go wrong”
- ✓ Simplified threat modelling embedded into ceremonies
- ✓ Secure defaults and “paved roads”

Moving from Design to Default



Roadmap

Make secure development the easier default

Secure by Default

- A continually improving flywheel for Secure Software Development Lifecycle (SSDLC) that improves the developer experience.

Automate and Standardise

- Use automation for deployments, testing, and infrastructure.
- Standardize environments to reduce variability and errors.
- Build Paved-Roads with security & SSDLC built-in

Measure and Improve

- Track key metrics: lead time, deployment frequency, MTTR (mean time to recovery), change failure rate.
- Use postmortems and retrospectives to drive improvement.

Foster a Culture of Trust and Safety

- Encourage psychological safety and open communication.
- Support learning from failures rather than punishing mistakes.
- Ensure that lessons learnt and knowledge are collected, curated, well-managed, and accessible.
- Encourage co-creation, co-discovery & support over edicts.

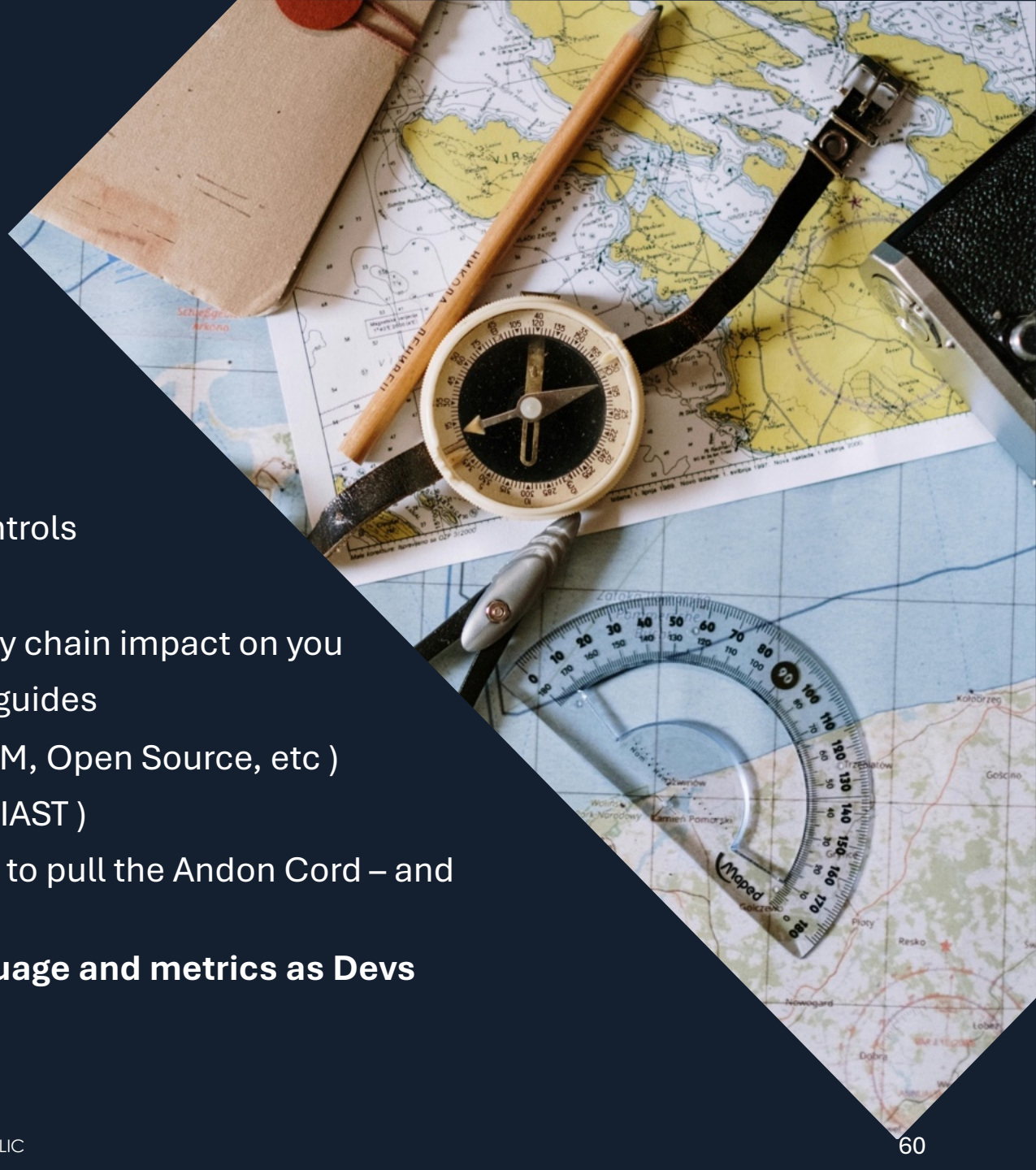
Build Cross-Functional Collaboration

- Break down silos between Dev, Ops, QA, and Security.
- Encourage shared ownership of outcomes.
- Cut through the complexity through a strike-force, focusing on uplifting 1 team at a time.

In Practice

Some examples

- Abuse/Adversary Stories with each User Story
- Centralise your CI/CD & build secure by default controls
- Know what you have (App Inventory)
- Know your ingredients (SCA & SBOM), and its supply chain impact on you
- Build a consolidated requirements & best practice guides
- Build, share and reuse patterns (e.g. Terraform, ORM, Open Source, etc)
- Quality Check (Test Suites, Unit Tests, SAST, DAST, IAST)
- Have a culture that empowers & supports the team to pull the Andon Cord – and remediate defects ASAP.
- **Security teams to use the common tooling, language and metrics as Devs**



Don’t forget

- Pay down technical debt & do housekeeping – incl backlog pruning
- Make work visible – that means good measures, and metrics.
- Document & Communicate

Key Theme

Develop a common language & cross functional collaboration – between the business, developers, & security.

Let them cook, and support them!

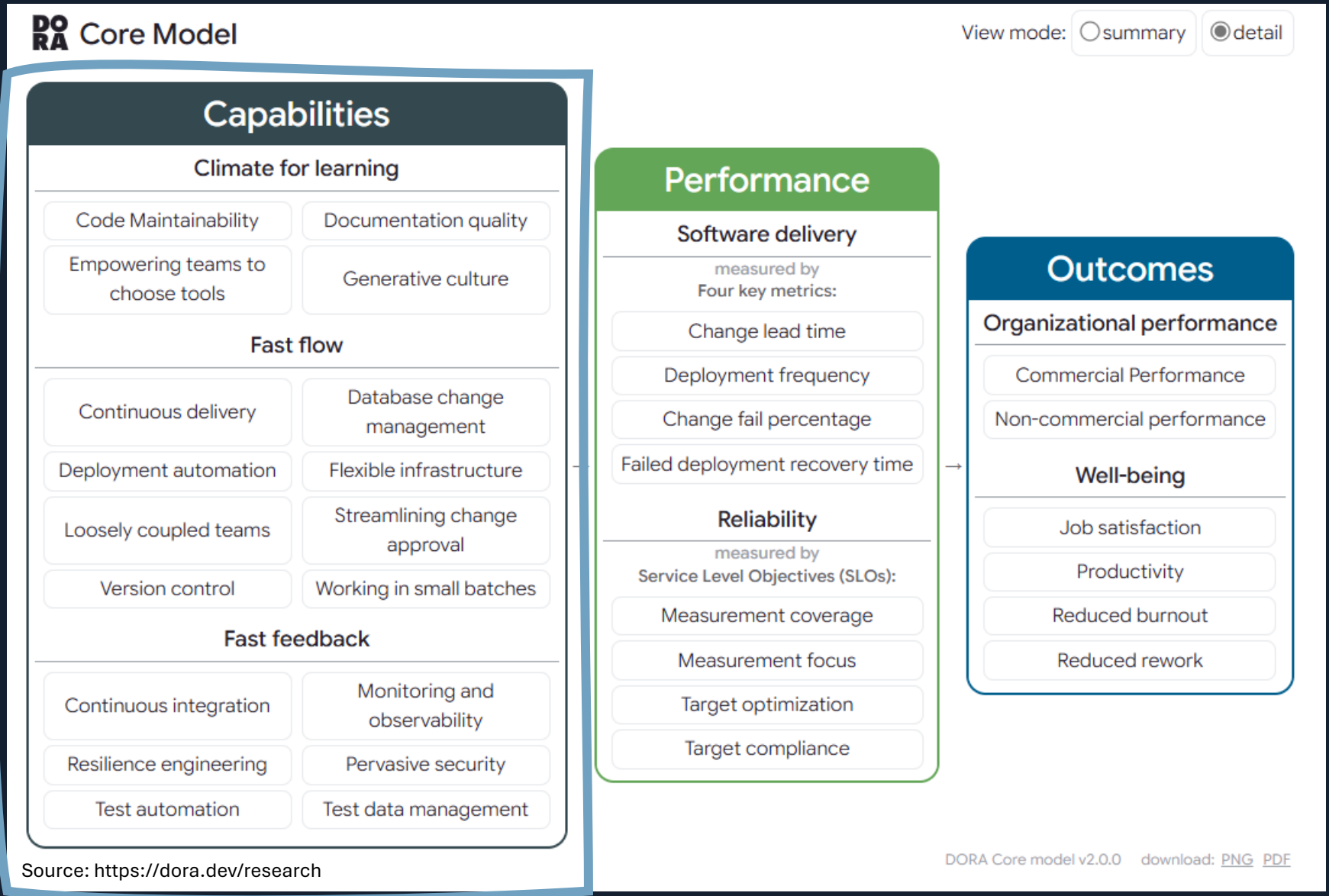
One at a time

Don't boil the ocean –
focus on teams that are already seeking change,
and uplift one workflow at a time, while
respecting team autonomy.

and improve as we iterate through the uplift...

Lessons Learnt from Manufacturing

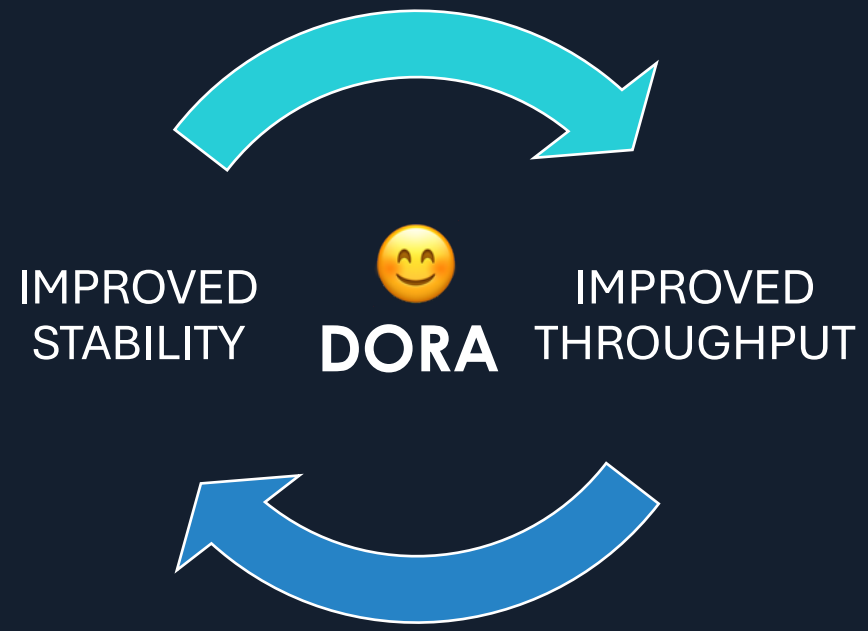
Focused on the Software Development/Delivery Lifecycle



Source: https://dora.dev/research

DevEx ❤️ AppSec

Enabling higher performing teams = Better Application Security



01 **Abuse Stories**
– Identify hazards & improve preparedness

02 **Prioritise (with RICE)**
– navigate tasks by scoring the backlog

03 **Build paved roads**
– build the flow you need & improve it.

Higher quality (and secure) apps, means lesser frustration.
A better developer experience 🥳

Better DevEx & AppSec 💪

Get started today

01 **Abuse Stories**
– Identify hazards & improve preparedness

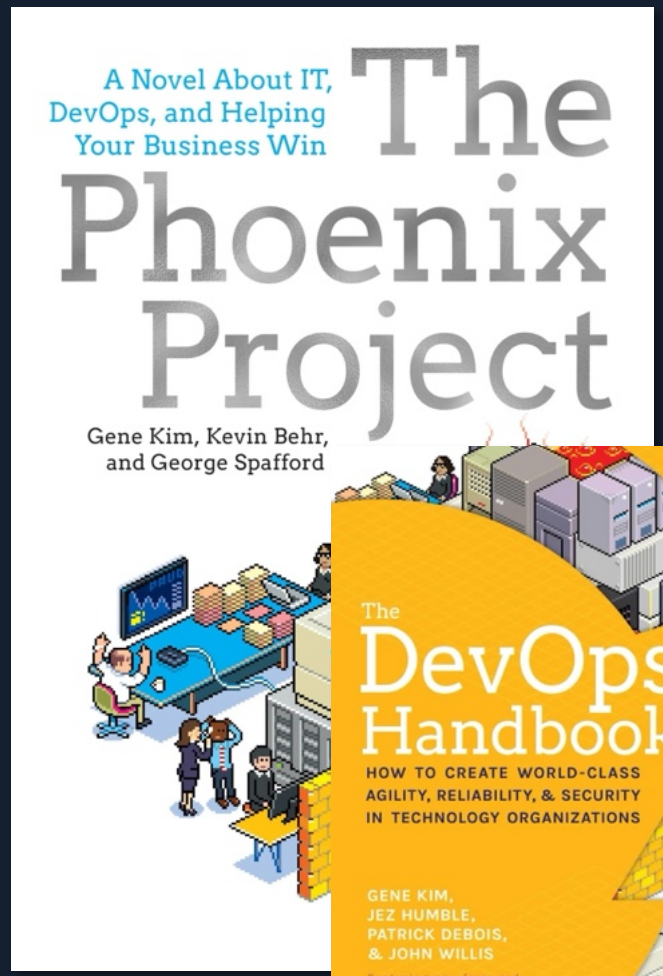
02 **Prioritise (with RICE)**
– navigate tasks by scoring the backlog

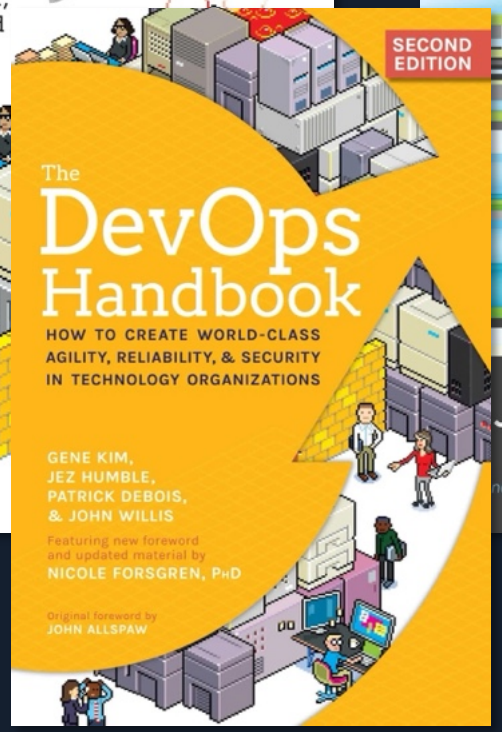
03 **Build paved roads**
– build the flow you need & improve it.

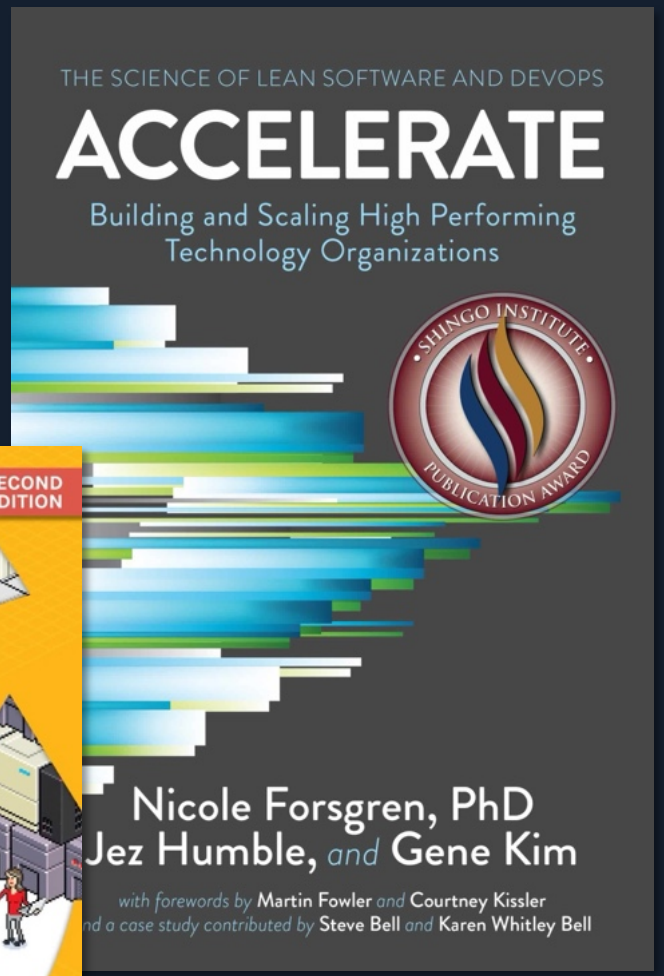
<href>

Further Reading

https://cybercx.com.au/resource/hack-report/







Q&A

Thank you

daniel.ting@cybercx.com.au

Where to find me on Socials?
hellodanielting.com

Get slides @
thisis.my/aisa2602

CyberCX | AISA

Thank you

daniel.ting@cybercx.com.au

Where to find me on Socials?
hellodanielting.com