Influencing Change in the Secure Software Development Lifecycle

OFFICIAL: Sensitive – NSW Government

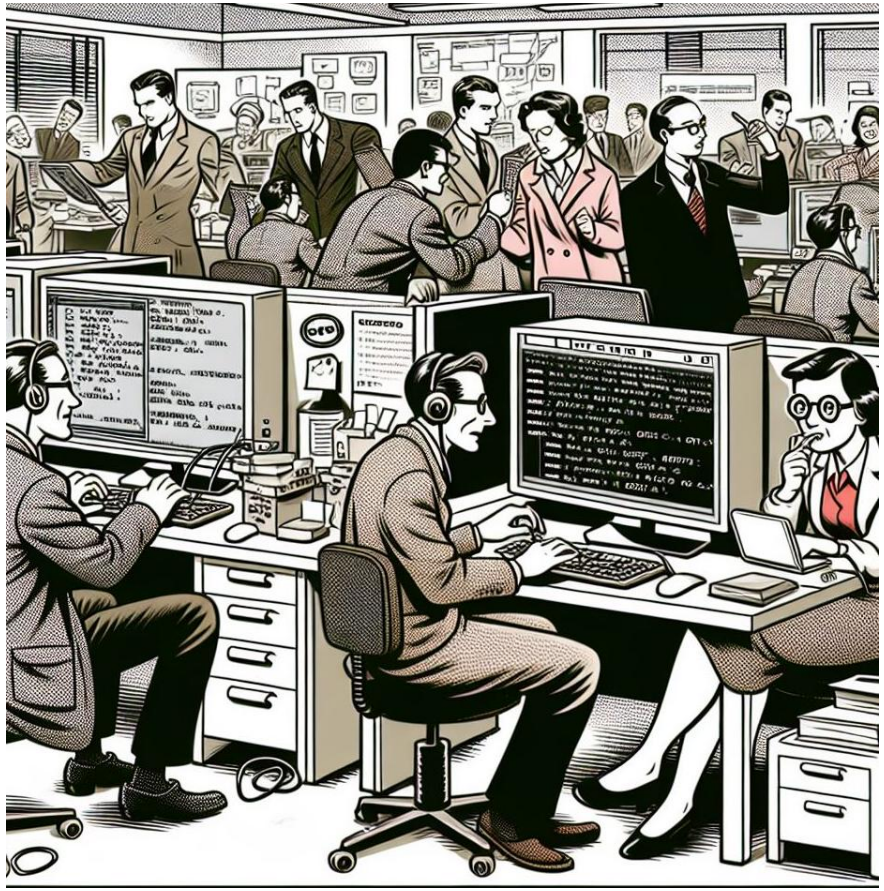
Alistair de B Clarkson – Associate Director of DevSecOps

22 Nov 2024

NSW GOVERNMENT

Service NSW

Outline



- Context
- -> Shifting Left
- The Problem Space
- -> Shifting too far left
- -> Cyber, the bad cop
- -> Softly softly
- -> Too tech to direct
- Solutions
- -> A trusted partner
- -> Data driven insights
- -> Tailoring Insights
- Results

1

Context

The DevSecOps Space

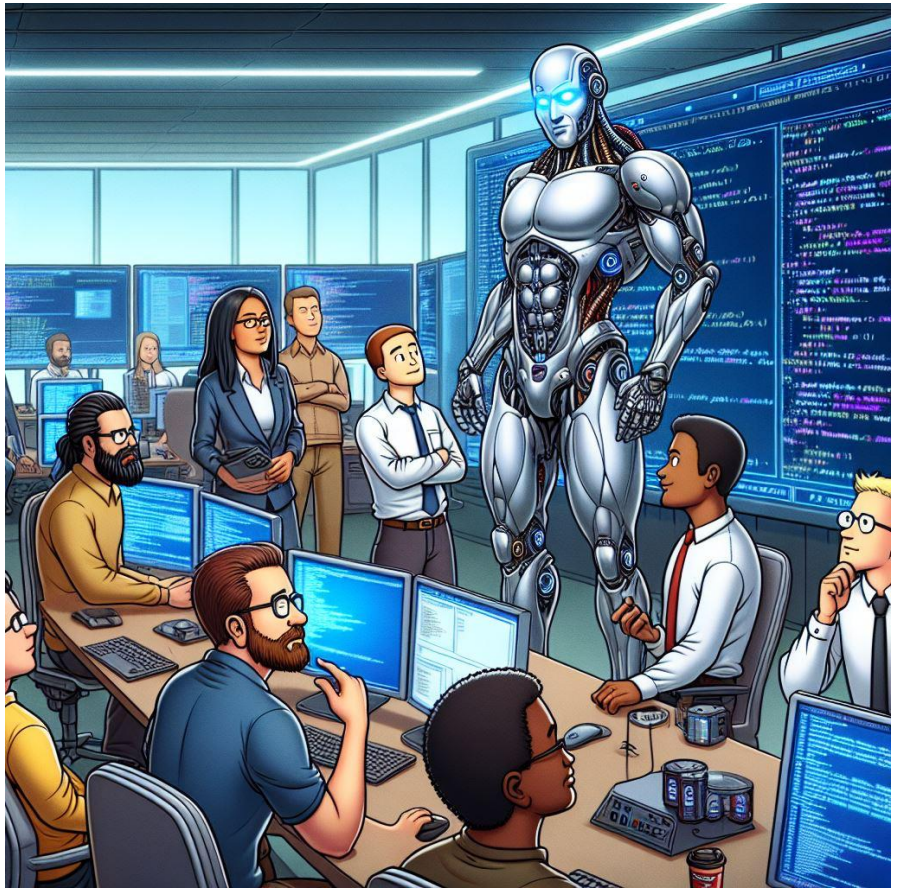
Shifting left & DevSecOps

In 2001 Larry Smith wrote an article titled ‘Shift-Left Testing’ which encouraged developers to get testing done earlier in the product lifecycle.

In 2009 the DevOps days conference helped shape the term DevOps to foster collaboration between developers and operations teams.

In around 2012 the DevOps movement inspired integration between cyber security and development to create DevSecOps.

Shift left became a familiar call in DevSecOps over the mid 2010’s to integrate security activities earlier in the development lifecycle. Developers had the context to solve problems earlier and became the people to do security earlier.



1

The Problem Space

An overview of some of the problems faced in the DevSecOps space

NSW GOVERNMENT

Service NSW

Shifting too far left

Developers have the context so they are best placed to:

- Develop the code
- Create Architecture Diagrams
- Maintain the documentation
- Write all the tests
- Add scanners to CI/CD pipelines
- Secure CI/CD pipelines
- Do vulnerability remediation
- Perform threat modelling
- Remediate risks
- ….
- ….
- Do everything

Cyber, The Bad Cop

Competing demands on developers come from:

- Product & Project Managers
- Developers themselves
- Cyber Security

To compete with such demands then there is the temptation to:

- Harden compliance requirements -> you must!
- Increasingly point out other people have a problem
- Label the other person as bad because they aren't doing enough

Softly softly

Soft approach:

- It’s just guidance
- It is a should do, not a must
- It’s best practice not compliance

Does not compete with demands, danger of:

- Getting overlooked
- Being at the back of the backlog without resourcing for it
- What evidence is there of it being completed / considered ?

Too tech to direct

DevSecOps is one of the most technical areas of product development and is not directly correlated with functional requirements or finance.

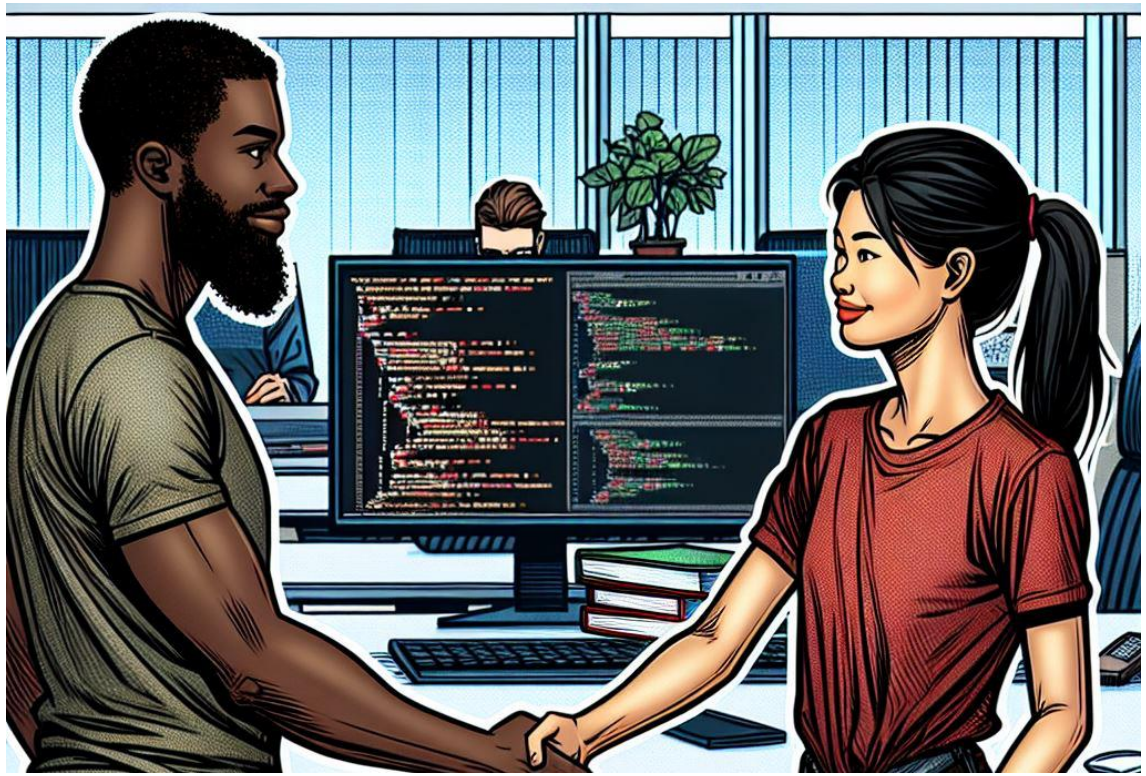
For top-down buyin on DevSecOps initiatives:

- Jargon may not be understood
- Impact may not appear immediately relevant
- Projects may have complex dependencies
- Projects may need to pivot for different targets

3

Solutions

A Trusted Partner



Rather than assigning a security champion and giving them more to do, could we have somebody embedded with a team who can help them with their challenges, identifying, tracking and prioritizing competing demands?

This Security Partner:

- Takes engineers on the journey of why cyber issues are a priority
- Has the context of both products and cyber issues
- Is able to provide technical assistance not just add tasks

Data Driven Insights

By creating dashboards of insightful information from vulnerability management tools, source code management, spreadsheets and other information sources…

Rather than dictate what to do, a partner can take group leadership on the journey of the why behind the what:

- Make a collaborative decision on priorities based on data
- Impact of issues is contextualized by organizational baselines
- Partners working with group are able to be responsive to current conditions for different teams

Tailoring Insights

Too much information is overwhelming. We have found that each insight needs to be tailored to be actionable by the group leadership. To be actionable it needs:

- To be understandable
- To facilitate a decision by group leadership
- Be attached to a clear impact
- Facilitate an action that can be performed by group leads

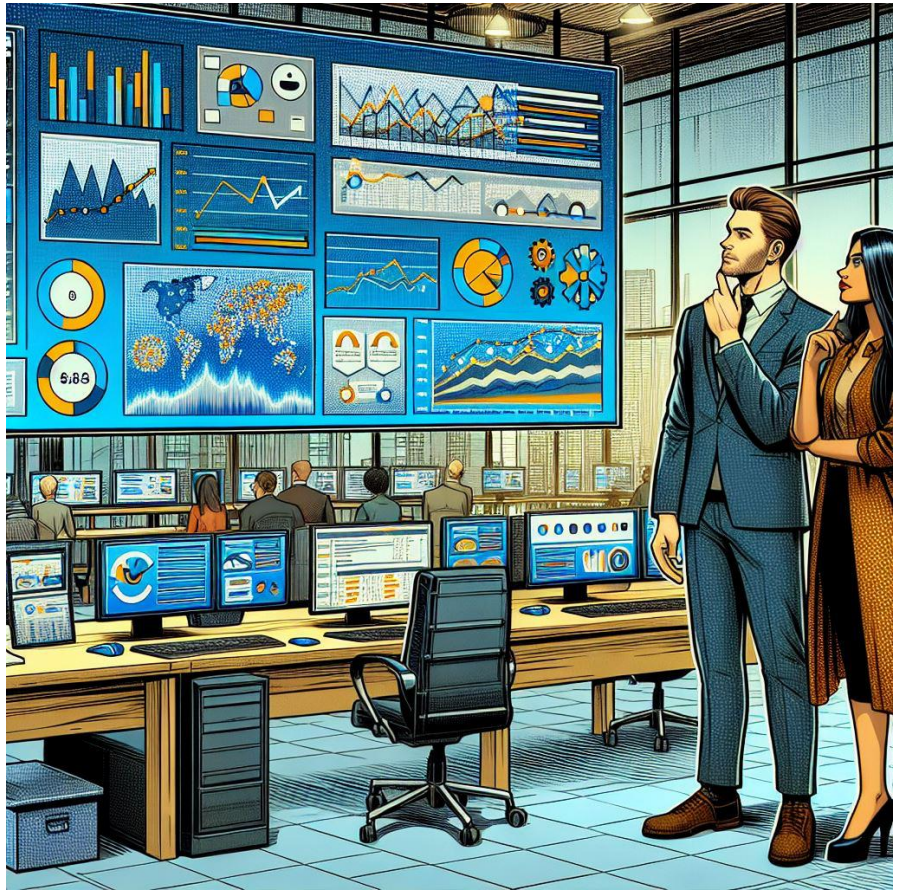
4

Results

What has happened here?

Outcomes

- Multiple teams have outgrown our DSOMM inspired DevSecOps maturity assessment and need more advanced measurements.
- Implemented automation and a wide range of capabilities that have reduced the time to fix vulnerabilities.
- Massively expanded our diagrammatic threat model coverage to have more than 90 threat models over ~70 product teams -> attributed with reducing the number of findings from penetration testing.
- > 1000 views on our DevSecOps guidance pages with training documentation / videos.
- DevSecOps department expanded to encompass all security advisory and partnership functions.



5

Any Questions