RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **CXO-W09**

The Impact of Software Security Practice Adoption Quantified

Larry Maccherone

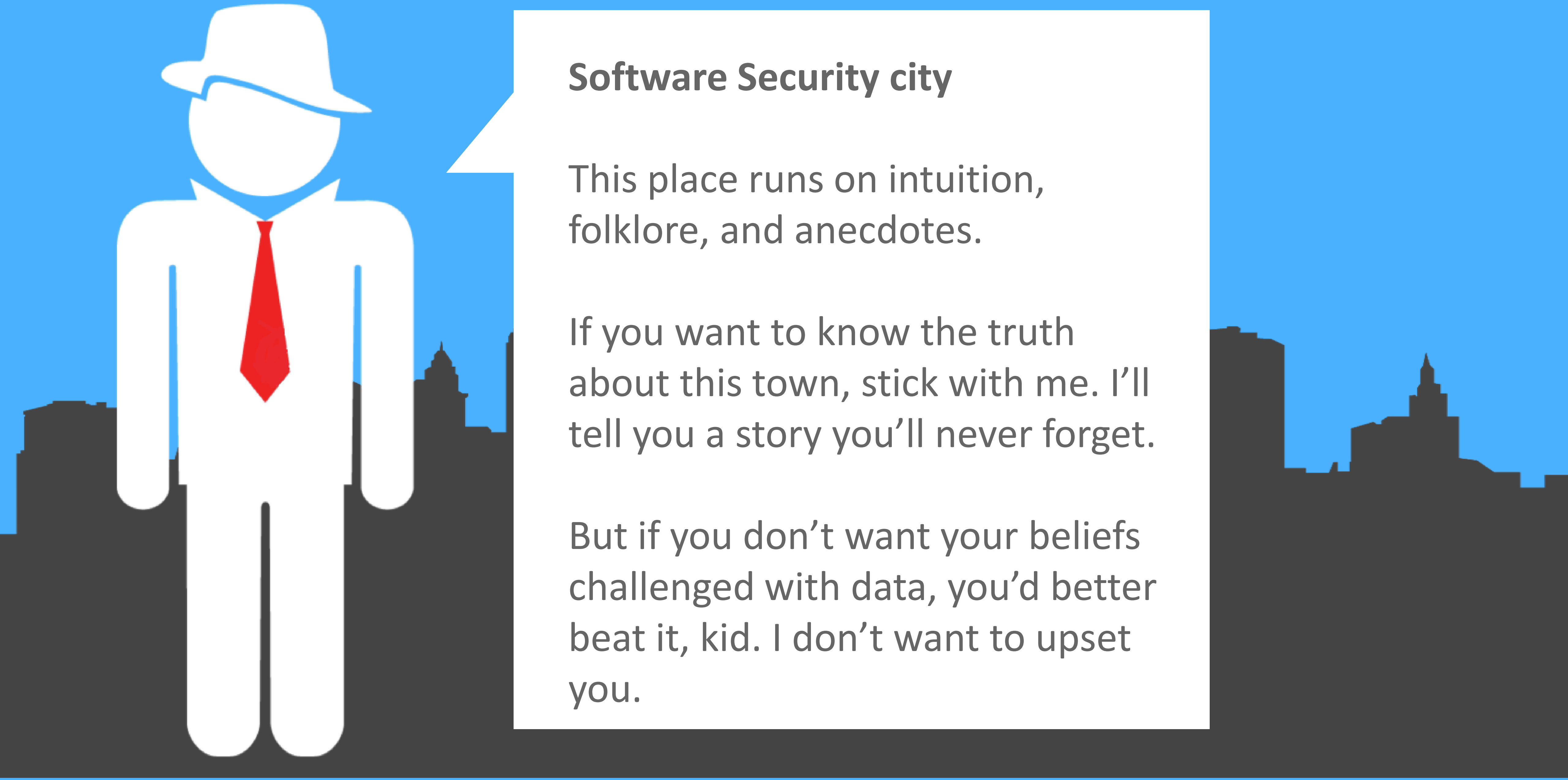
Distinguished Engineer, Comcast
@LMaccherone
LinkedIn.com/in/LarryMaccherone

A De-mystery Thriller

The Impact of DevSecOps Quantified

Larry Maccherone
@LMaccherone
LinkedIn.com/in/LarryMaccherone

RSAConference2020



Software Security city

This place runs on intuition, folklore, and anecdotes.

If you want to know the truth about this town, stick with me. I’ll tell you a story you’ll never forget.

But if you don’t want your beliefs challenged with data, you’d better beat it, kid. I don’t want to upset you.

CAUTION:

Correlation does not necessarily mean causation

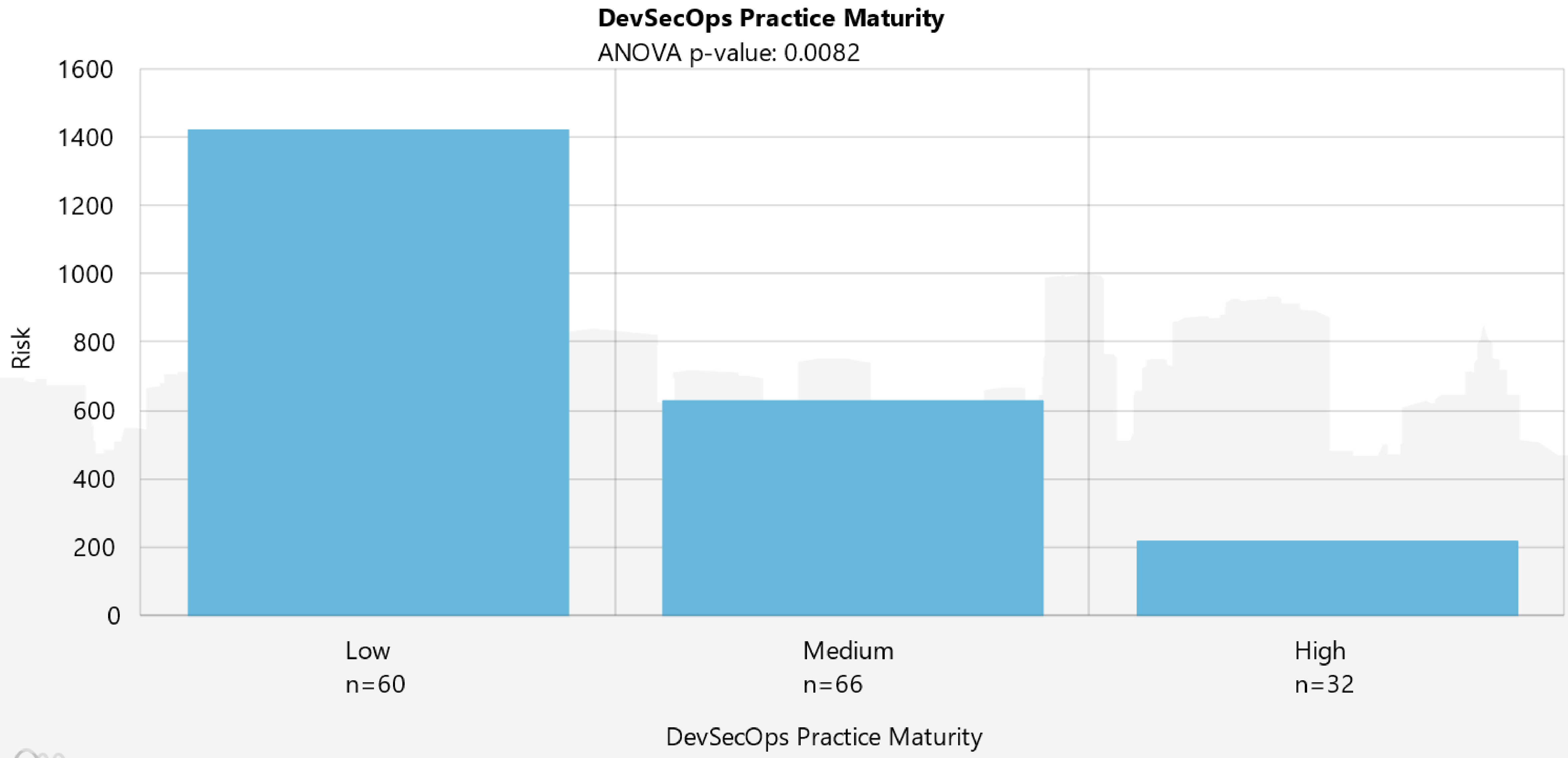
CAUTION:

There are no best practices

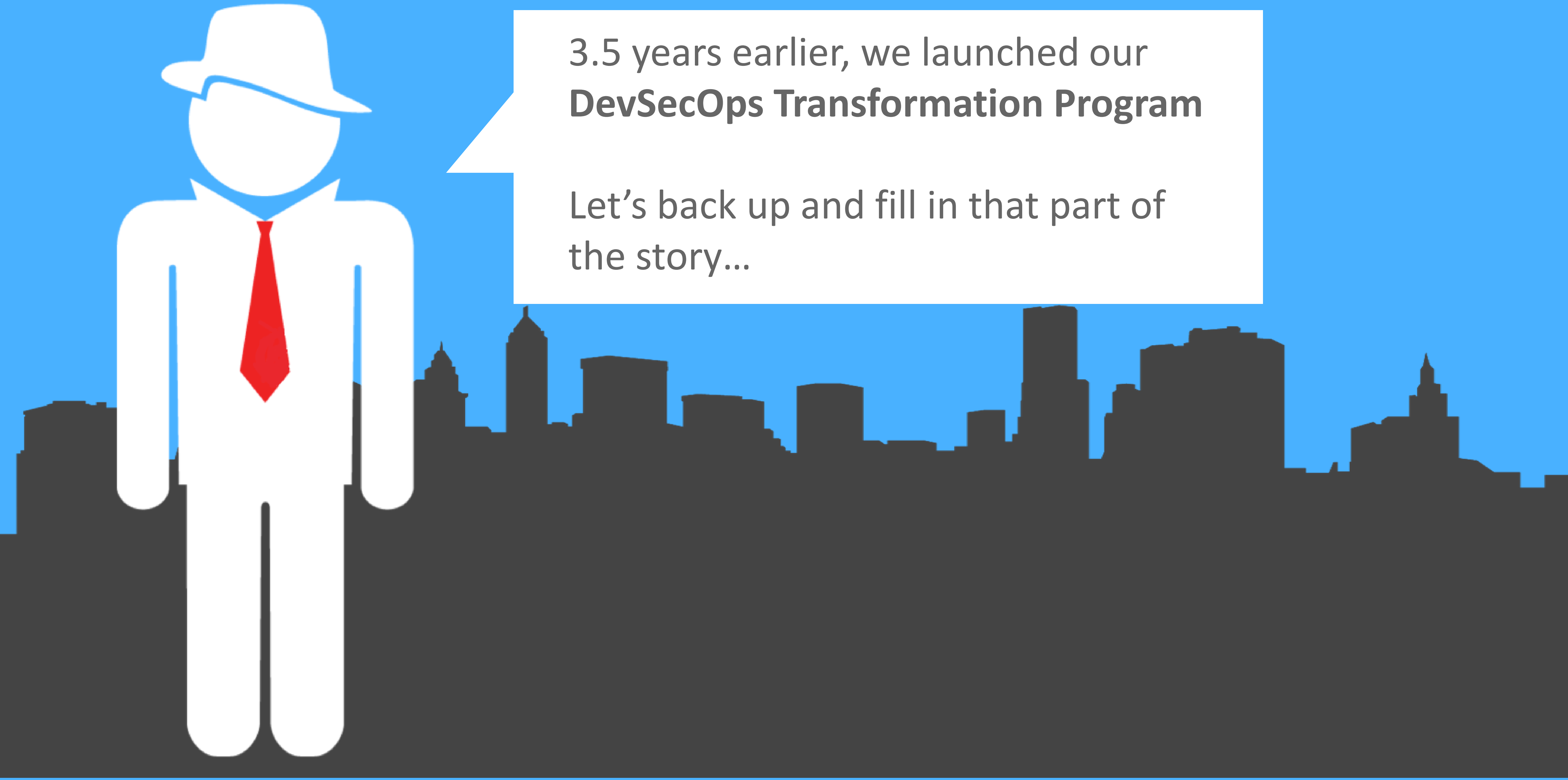
Only good practices in context

#NoAbsolutes

~Troy Magennis



DevSecOps Practice Maturity
ANOVA p-value: 0.0082
Risk
1600
1400
1200
1000
800
600
400
200
0
Low
n=60
Medium
n=66
High
n=32
DevSecOps Practice Maturity



3.5 years earlier, we launched our **DevSecOps Transformation Program**

Let’s back up and fill in that part of the story…

Security practices on DevOps continuum ➔ Dev[Sec]Ops

- Pen testing (Vuls found → Test scripts)
- Compliance validation (PCI, etc.)
- Fuzzing

- Analysis → Learning
- Defect/Incident 3-step
- New attack surface?
  Plan to update threat model

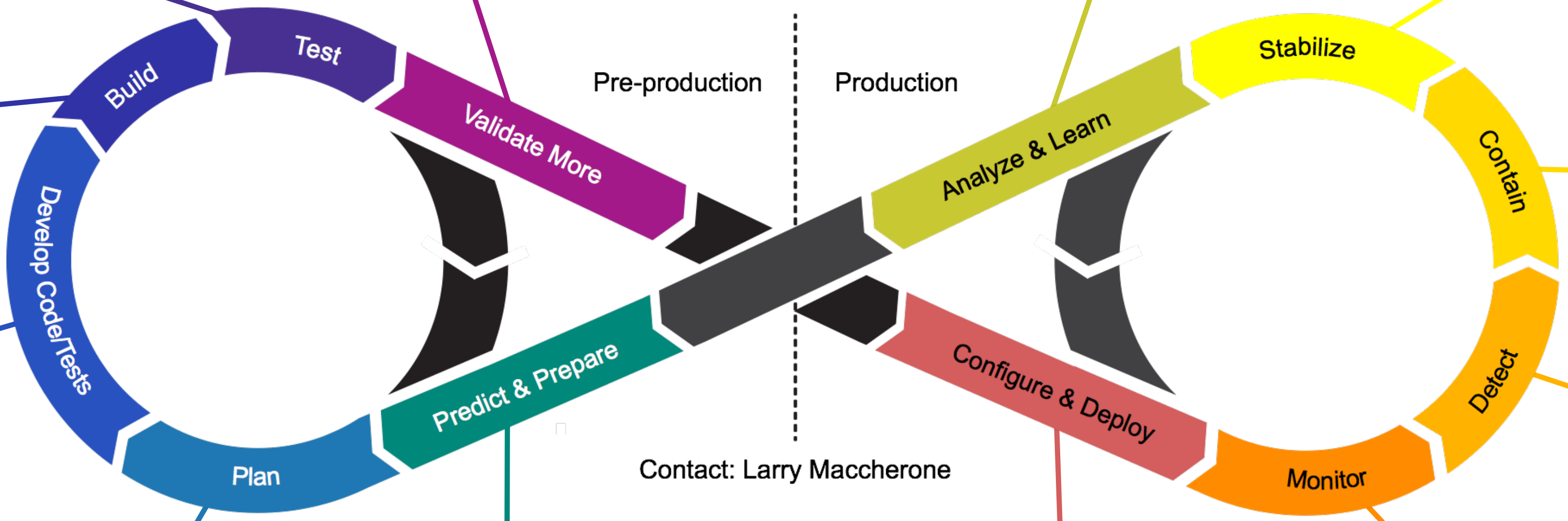
- Test security features
- Common abuse cases

- Restore/maintain service for non-attack usage

- Break the build
  code analysis

- RASP auto respond
- Roll-back or toggle off
- Block attacker
- Shut down services

- Static/IAST analysis
- Abuse case tests
- Code review



Contact: Larry Maccherone

- Intrusion detection
- App attack detection

- Threat modeling → backlog items
- Analyze/Predict → backlog items
- Design complies with policy?

- If we do X will it mitigate Y?
- Capacity forecasting
- Learning → Update playbooks and Training

- Configuration validation
- Feature toggles/Traffic shaping configuration
- Secrets management

- Log information for after-incident analysis

That’s a lot of stuff!

How do we get development teams to adopt?

3-Part Framework
Agile/DevOps/Security Transformation Framework

1. Hearts and minds of developers

2. Shallow team-level improvement ramp

3. Management visibility and goal setting

We, the Security Team…

COMCAST

Trust that Engineering Teams…

- Want to do the right thing
- Are closer to the business context and will make trade-off decisions between security and other risks

Pledge to…

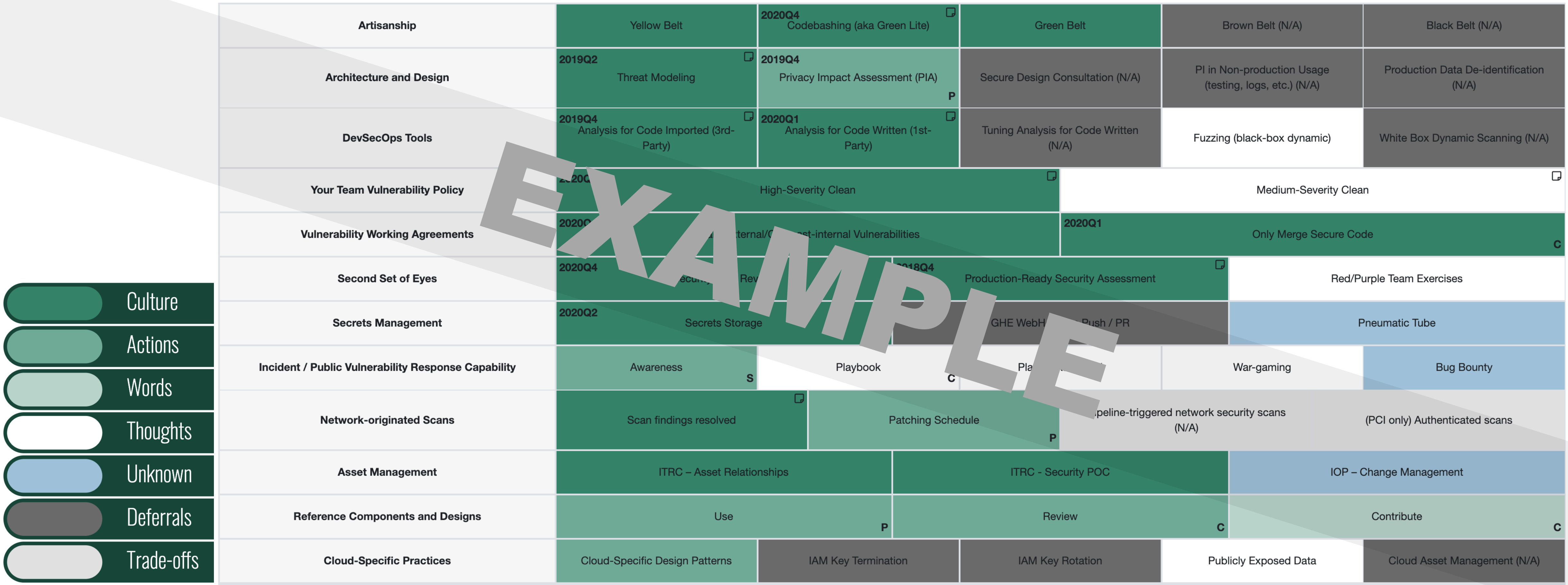
- Provide information and advice so those trade-off decisions are more informed
- Lower the cost/effort side of any investment in developer security tools or practices

Understand that…

- We are no longer gate keepers but rather tool-smiths and advisors

Live Demo

Get each team on the path to improvement
Thoughts → Words → Action → Culture (ThWAC)



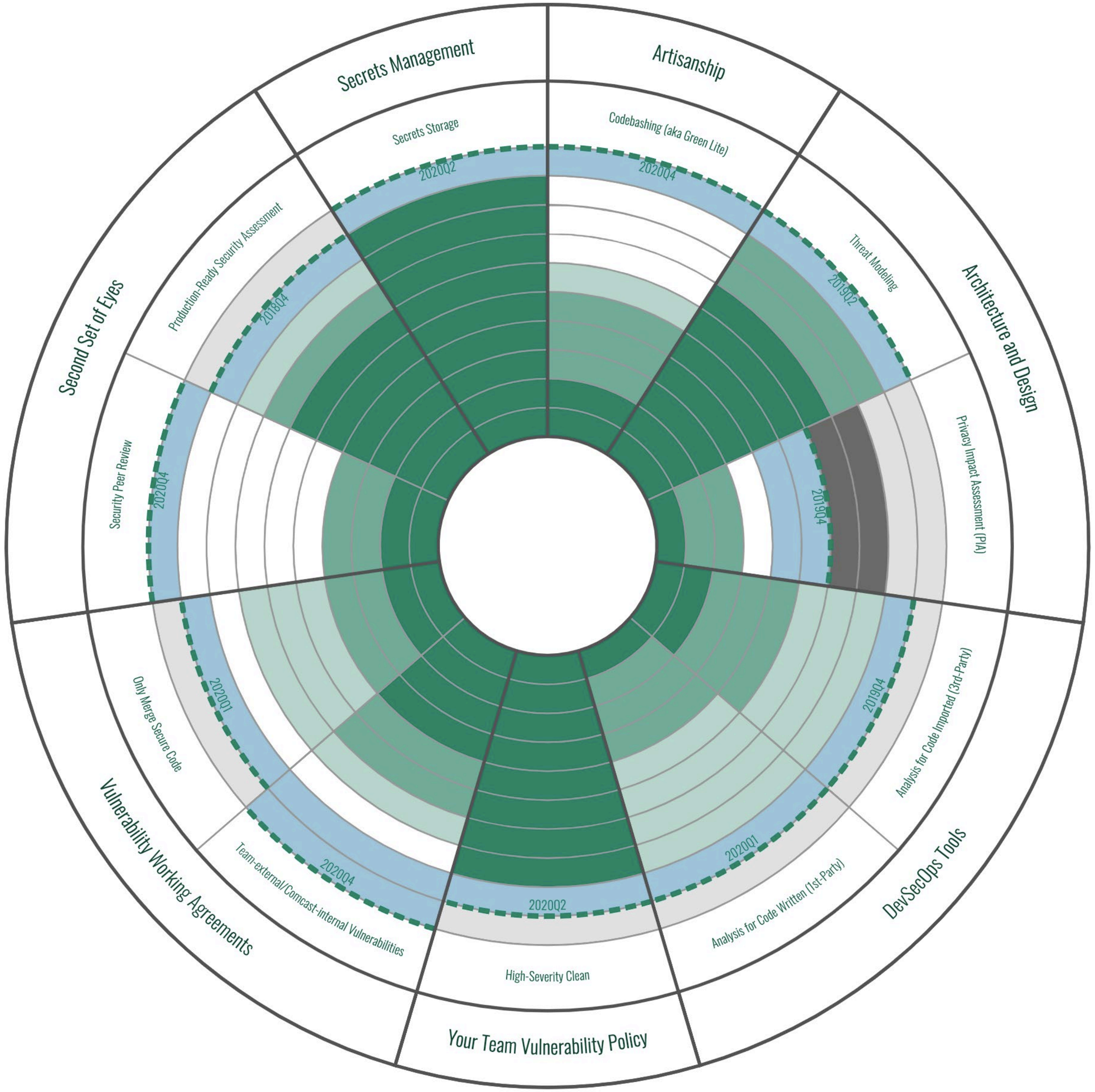
EXAMPLE

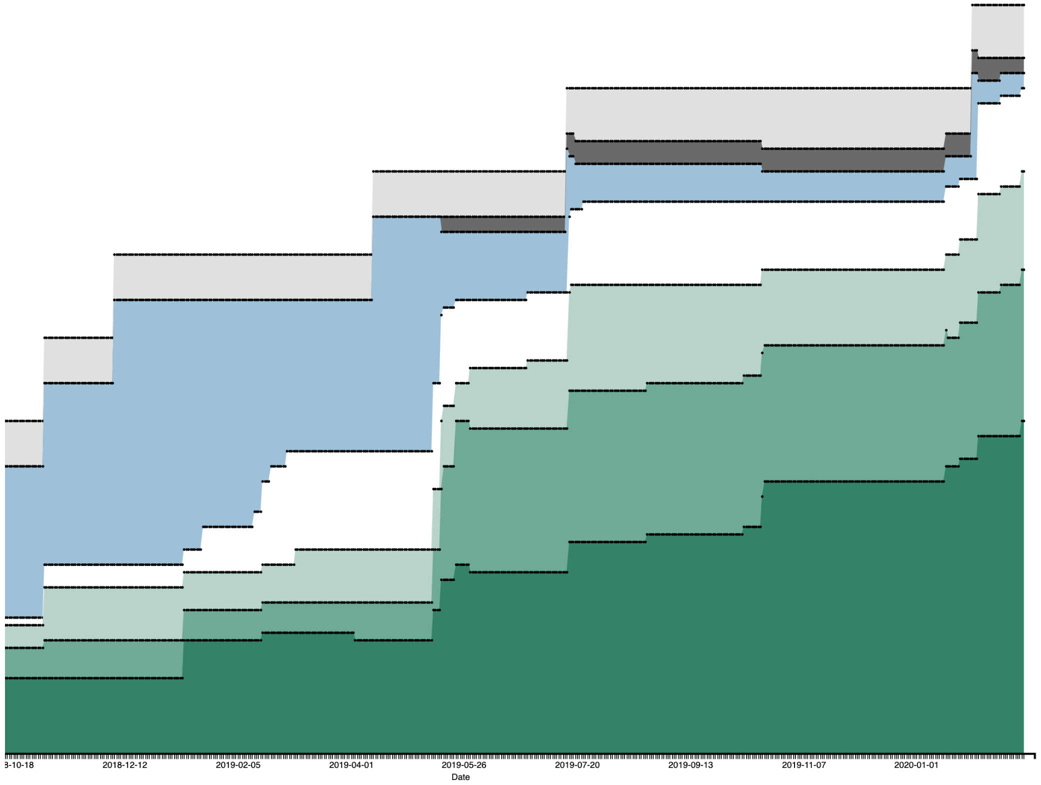
Artisanship	Yellow Belt	2020Q4 Codebashing (aka Green Lite)	Green Belt	Brown Belt (N/A)	Black Belt (N/A)
Architecture and Design	2019Q2 Threat Modeling	2019Q4 Privacy Impact Assessment (PIA) P	Secure Design Consultation (N/A)	PI in Non-production Usage (testing, logs, etc.) (N/A)	Production Data De-identification (N/A)
DevSecOps Tools	2019Q4 Analysis for Code Imported (3rd-Party)	2020Q1 Analysis for Code Written (1st-Party)	Tuning Analysis for Code Written (N/A)	Fuzzing (black-box dynamic)	White Box Dynamic Scanning (N/A)
Your Team Vulnerability Policy	20Q High-Severity Clean			Medium-Severity Clean	
Vulnerability Working Agreements	2020Q ternal/C st-internal Vulnerabilities			2020Q1 Only Merge Secure Code C	
Second Set of Eyes	2020Q4 ecur Rev		18Q4 Production-Ready Security Assessment		Red/Purple Team Exercises
Secrets Management	2020Q2 Secrets Storage		GHE WebH Push / PR		Pneumatic Tube
Incident / Public Vulnerability Response Capability	Awareness S	Playbook C	Pla	War-gaming	Bug Bounty
Network-originated Scans	Scan findings resolved	Patching Schedule P	peline-triggered network security scans (N/A)	(PCI only) Authenticated scans	
Asset Management	ITRC – Asset Relationships		ITRC - Security POC		IOP – Change Management
Reference Components and Designs	Use P		Review C		Contribute C
Cloud-Specific Practices	Cloud-Specific Design Patterns	IAM Key Termination	IAM Key Rotation	Publicly Exposed Data	Cloud Asset Management (N/A)

- Culture
- Actions
- Words
- Thoughts
- Unknown
- Deferrals
- Trade-offs

Larry Maccherone
@LMaccherone | LinkedIn.com/in/LarryMaccherone

Management Visibility into Progress on Essential 11 Dev[Sec]Ops Practices





The Program is working

"That was awesome!", "Loved it!", "Wow!", "Very valuable and engaging. Much more than I expected"

"Very different approach than we expect from security", "Dev team empowerment (teams own their own security)", "Process driven by dev team priorities, not policy-driven", "Collaborative effort to improve security"

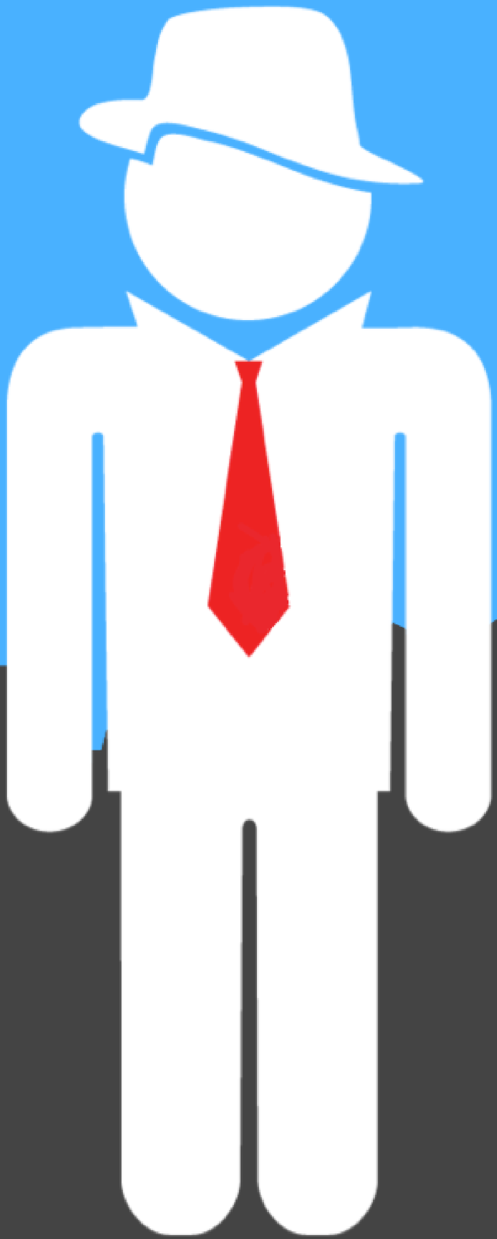
Most valuable was… "Learning about all the different practices", "Understanding the global view", "Quantifying what needs to be done"

"Loved the bang-for-the-buck ordering as opposed to a book of policies"

- Teams sign up for 2.46 practice adoption goals for each 90-day window. We ask for 1 or 2, maybe 3 or 4.
- 93% of team 90-day practice adoption goals are fully or mostly achieved
- Conducted these facilitated self-assessments with 200 different teams. 200-300 remain.

The investigation proceeds with ...

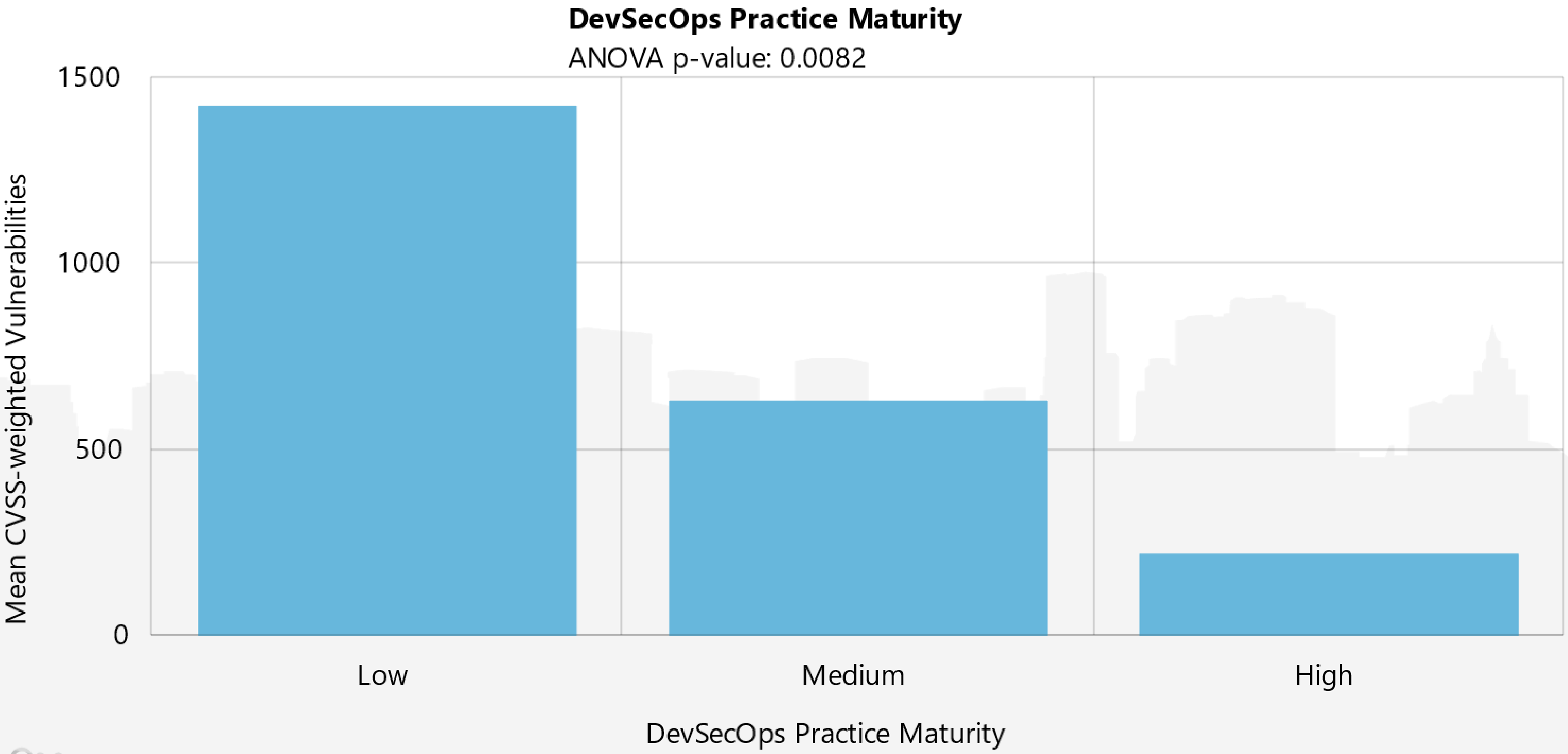
Finding Clues



- 195 teams onboarded to DevSecOps Transformation Program via facilitated self-assessment and workshop which gives us the practice data for our x-axis
- Able to map data from 158 of those teams to in-production network originated scan findings from Nessus, Qualys, etc. which gives us our outcome data for our y-axis
- Other sources of risk outcomes under consideration (like security incidents) for later study

The investigation proceeds with ...

DevSecOps Practice Maturity


DevSecOps Practice Maturity
ANOVA p-value: 0.0082
Mean CVSS-weighted Vulnerabilities
1500
1000
500
0
Low
Medium
High
DevSecOps Practice Maturity

CVSS-weighted Vulnerability Risk Score
(the Y-axis)

- The count of vulnerabilities found by network originated scans (Nessus, Qualys, etc.) weighted by their CVSS severity score
- Formula

$$Risk = \sum 2^{CVSS}$$

- So, a CVSS score of…
  - 10 adds 1024 to the risk score
  - 9 adds 512 to the risk score
  - …

DevSecOps Practice Maturity
(the X-axis)

Count of 7 key practices that the team has gotten all the way to "Culture"

- **Low**: 0
- **Medium**: 1-2
- **High**: 3 or more

DevSecOps Practice Maturity Analysis

- When Maturity is…
  - **High**, average CVSS-weighted vulnerability risk score is 219 equivalent to ~ 1 CVSS score 7.8 vuln
  - **Low**, average CVSS-weighted vulnerability risk score is 1423 equivalent to ~ 2 CVSS score 9.5 vulns or 6.4 CVSS score 7.8 vulns
- The ANOVA p-value is 0.008 indicating that there is a less than 1% chance this correlation is due to chance

In other words…

High DevSecOps maturity correlates with 85% lower security risk

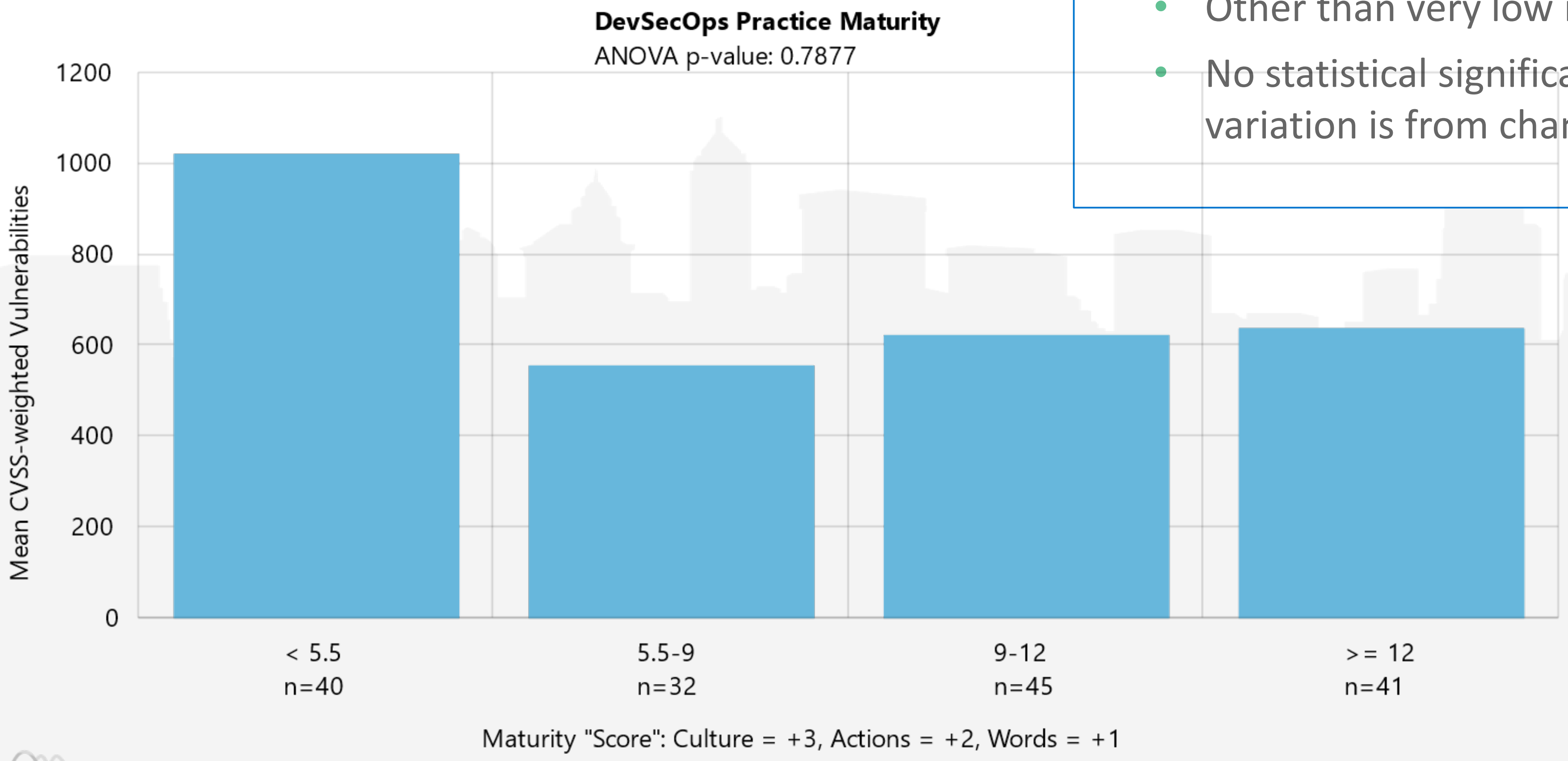
Evidence indicates…

- High DevSecOps maturity correlates with 85% lower security risk, so…
- **You should launch (or accelerate) your DevSecOps Transformation Program**

The investigation proceeds with ...

Should we give “partial credit”?

Evidence that Only “Culture” Counts

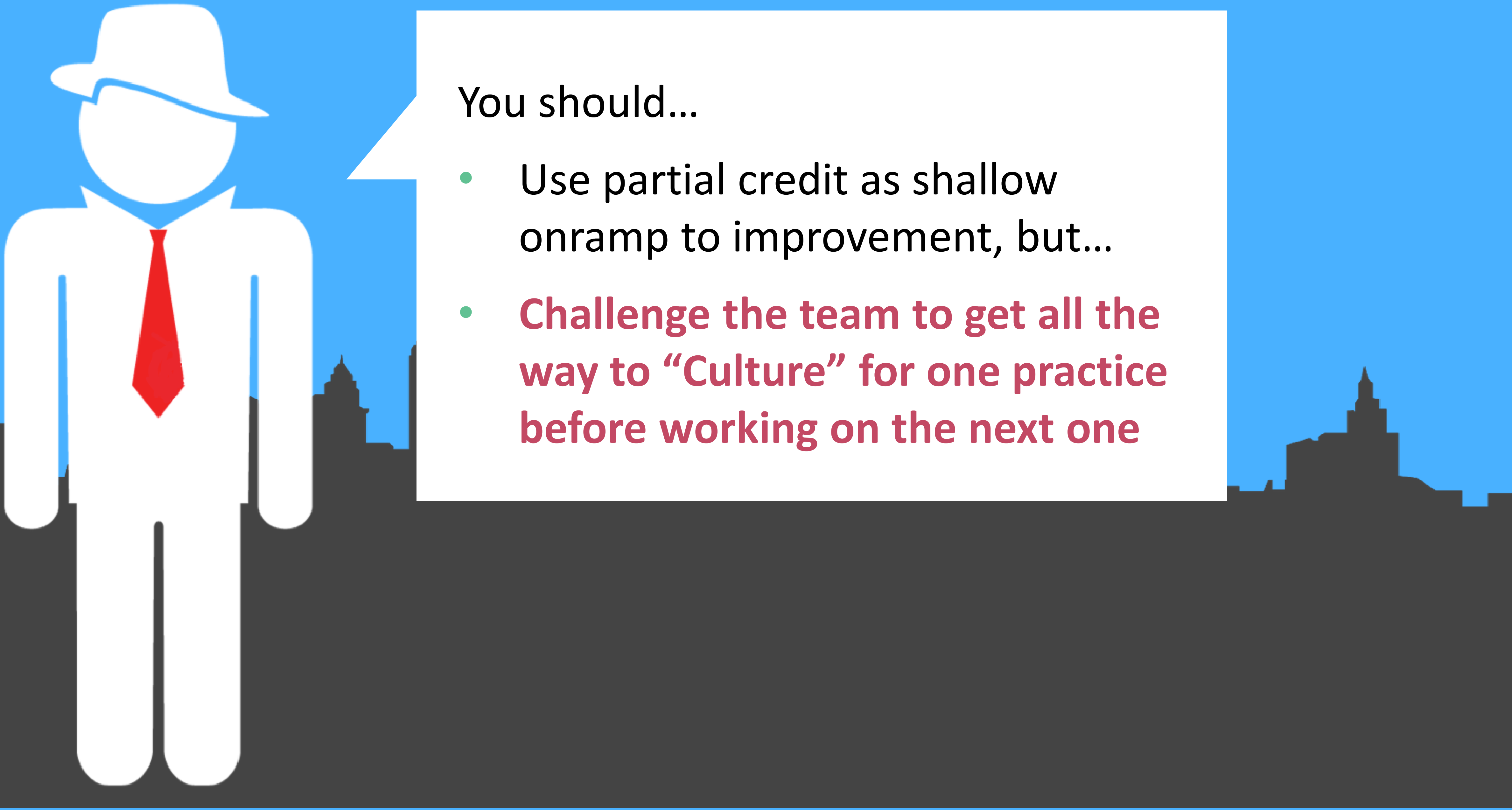


Same data but with partial credit for Actions (2 ”points”) and Words (1 “point”) in addition to Culture (3 “points”):

- Other than very low maturity, the trend disappears
- No statistical significance: almost 80% probability the variation is from chance

Key Insight: Must get all the way to *CULTURE*

- Maturity progression (ThWAC)
  - Thoughts
  - Words
  - Actions
  - Culture
- When we only "give credit" for Culture, the correlation is strong
- When we "give partial credit" for Words and Actions the correlation essentially disappears



You should…

- Use partial credit as shallow onramp to improvement, but…
- **Challenge the team to get all the way to “Culture” for one practice before working on the next one**

The investigation proceeds with ...

Quantifying the impact of each practice

Positively correlated practices

Practice	Risk reduction
Team-external/Org-internal vulnerability process (Note: probable bias with our risk score) *Process for assuring that team-external/org-internal vulnerabilities get resolved in the SLA*	73%
High-severity clean *Resolving the initial set of in pipeline scanning findings to zero*	65%
Security peer review *… as part of pull request*	61%
Secure coding training *Checkmarx Codebashing (2-3 hours) required for all team members who regularly write production code*	49%
Only merge secure code *Pull request branch protection status check on scan results to stay at zero*	49%
Threat modeling *4-8 hour workshop with security architect facilitating an evaluation of your product design*	46%
Production-ready security assessment (PRSA) *Periodically submitting your application(s) to internal white-box pen testing++ assessment*	44%
Secrets management *Assuring that passwords, certificates, API keys, etc. are securely stored & not in source code repositories*	20%

“But my gal, The Truth, she ain’t always kind…”

Practice	Risk reduction
Security yellow belt training Basic security awareness training and introduction to other training and programs	0%

The investigation proceeds with …

Why does basic security training seem to provide no risk reduction?

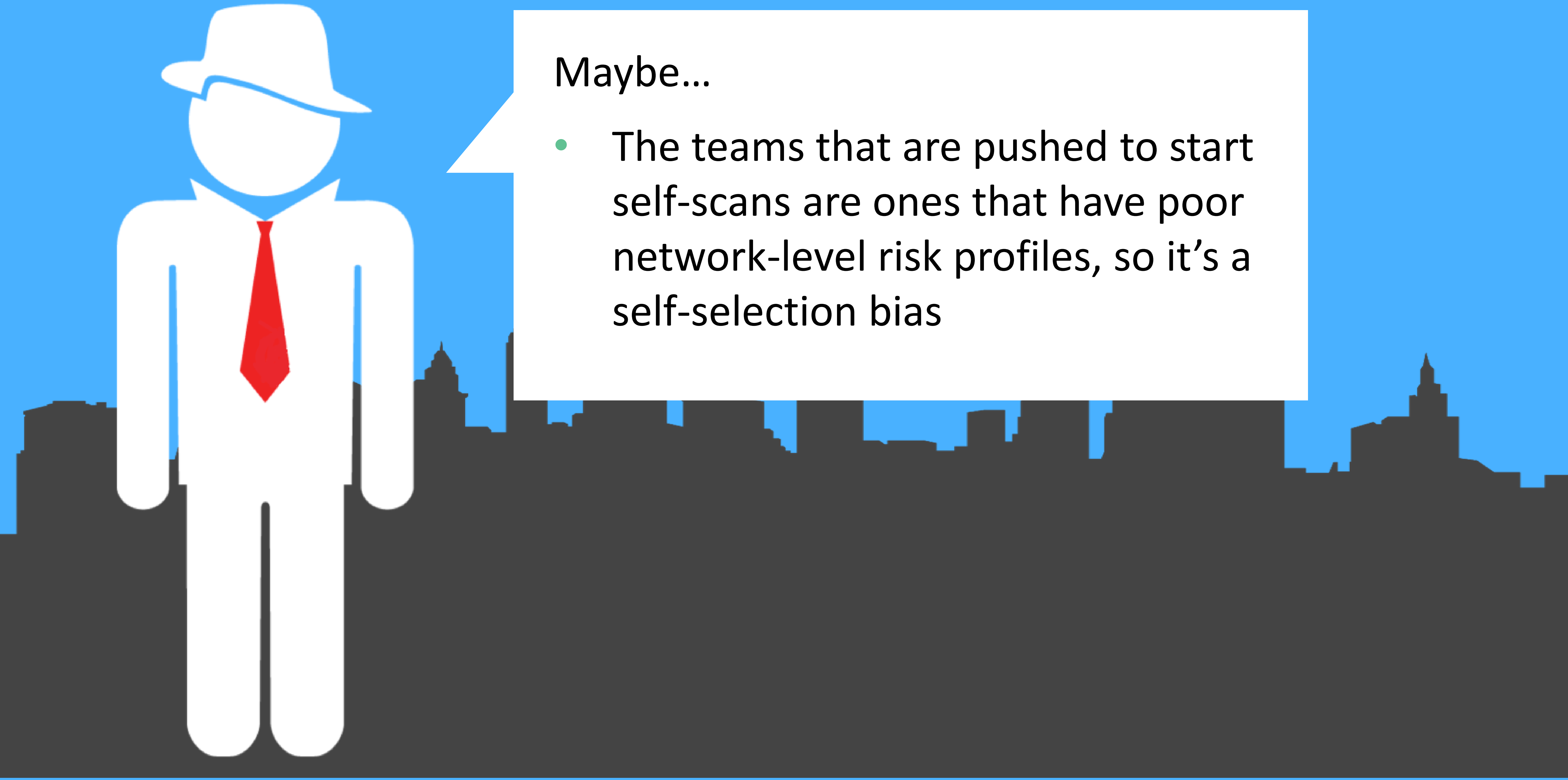
Because…

- Yellow belt is at saturation meaning almost everyone has it
- So, if it was anything but zero that would indicate that our math was wrong

Practice	Risk reduction
Analysis for code written scanning (1st-party, SAST/IAST) … integrated with CI/CD pipeline	-10%
Analysis for code imported scanning (3rd-party, SCA) … integrated with CI/CD pipeline	-39%

The investigation proceeds with …

Why does scanning alone negatively correlate with risk reduction?



Maybe…

- The teams that are pushed to start self-scans are ones that have poor network-level risk profiles, so it’s a self-selection bias

But keep in mind…

- These two practices…

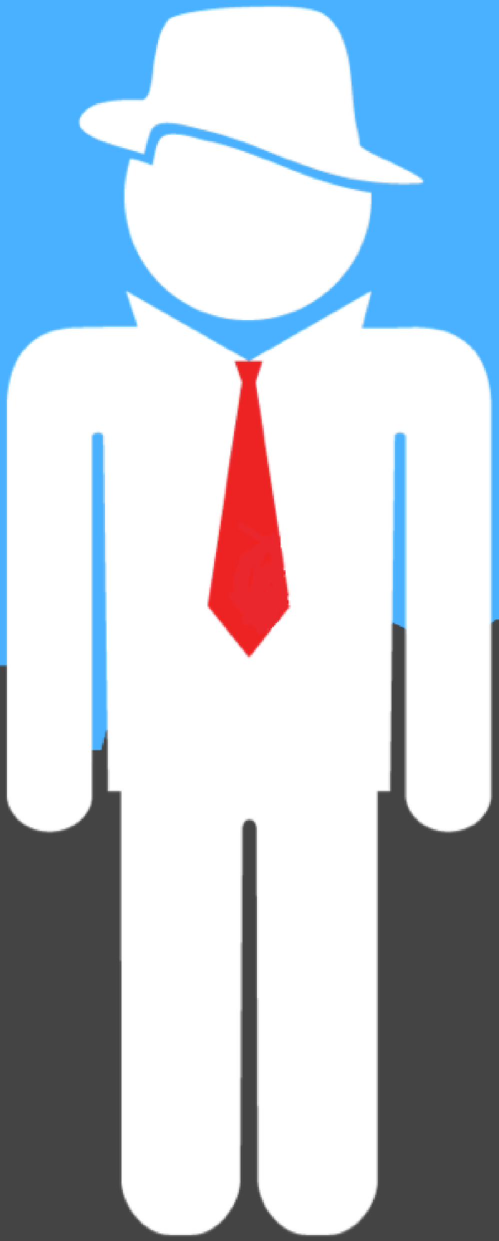
Practice	Risk reduction
Analysis for code written scanning (1st-party, SAST/IAST) … integrated with CI/CD pipeline	-10%
Analysis for code imported scanning (3rd-party, SCA) … integrated with CI/CD pipeline	-39%

- … are prerequisites for these two practices

Practice	Risk reduction
High-severity clean *Resolving the initial set of in pipeline scanning findings to zero*	65%
Only merge secure code *Pull request branch protection status check on scan results to stay at zero*	49%

Seems obvious but…

- We’re always looking for quick wins
- I constantly run into folks that put together a project plan to start scanning across their part of the organization leaving the effort to get to healthy resolution curves as a later exercise
- It’s much better to achieve value in each small part of the organization before starting to scan other parts



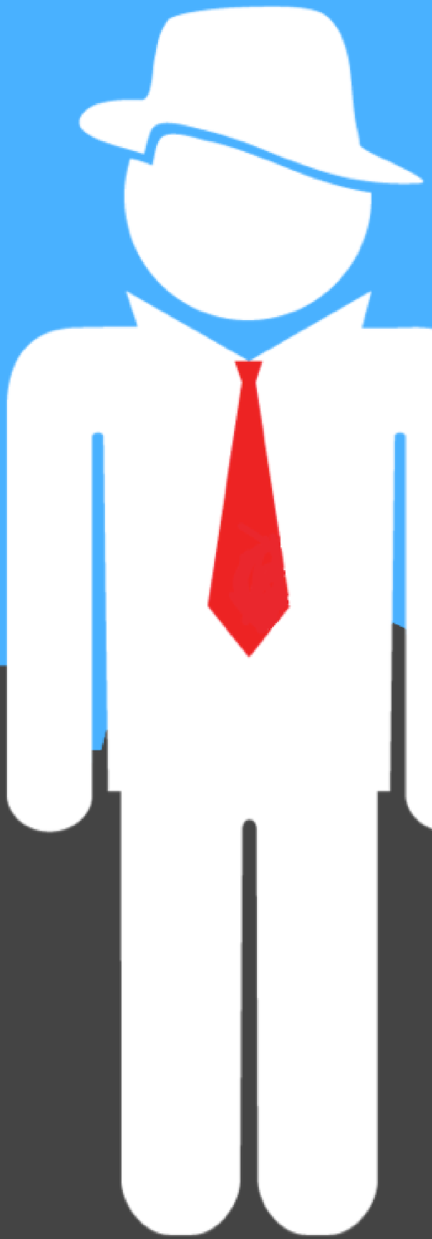
Evidence indicates…

- Scanning alone provides no risk reduction
- **Spend limited resources on driving healthy resolution curves before expanding scanning footprint**

Practice	Risk reduction
Security green belt training 1 person on team has been through 40+ hours of training and project	-76%

The investigation proceeds with …

Why is 40+ hours of security training highly negatively correlated with risk?



Since…

- Teams get to “Culture” for this practice by having only one member get Green Belt

Maybe…

- One individual is not able to change the outcome for an entire team
- Team members who sign up for Green Belt may be doing it to change roles or escape from a low performing team (self-selection bias)
- Teams that already have advanced security knowledge will have low risk but not value novice-level training (the opposing self-selection bias)

So we are now…

- Rethinking our approach to novice-level security training
- Improving selection criteria
- Shifting focus to getting all coders to take 2-3 hour secure coding training which is associated with a 49% risk reduction

Bottom line…

- Without this quantitative insight, we would have gone blindly forward with our training strategy

Apply What You Have Learned Today

- Starting now:
  - Invest in DevSecOps transformation
  - Using the data in this presentation and your own context, identify the key practices for your DevSecOps Transformation
- In the first three months following this presentation you should:
  - Gap analysis: Above list VS "easy button" for development teams to adopt
  - Put in place plans to create the "easy button" to close this gap
  - Start to hire talent to act as coaches and pipeline engineers for your DevSecOps transformation program
- Within six months you should:
  - Conduct 5-10 facilitated DevSecOps self-assessments and coaching workshops
  - Coach your first team to culture on the first few key "easy button" practices

A fact without a theory
is like a ship without a sail,
is like a boat without a rudder,
is like a kite without a tail.

A fact without a figure
is a tragic final act.

But one thing worse
in this universe
is a theory without a fact.

~ George Schultz

The Impact of DevSecOps Quantified

Upgrade **Intuition** to ***Insight***

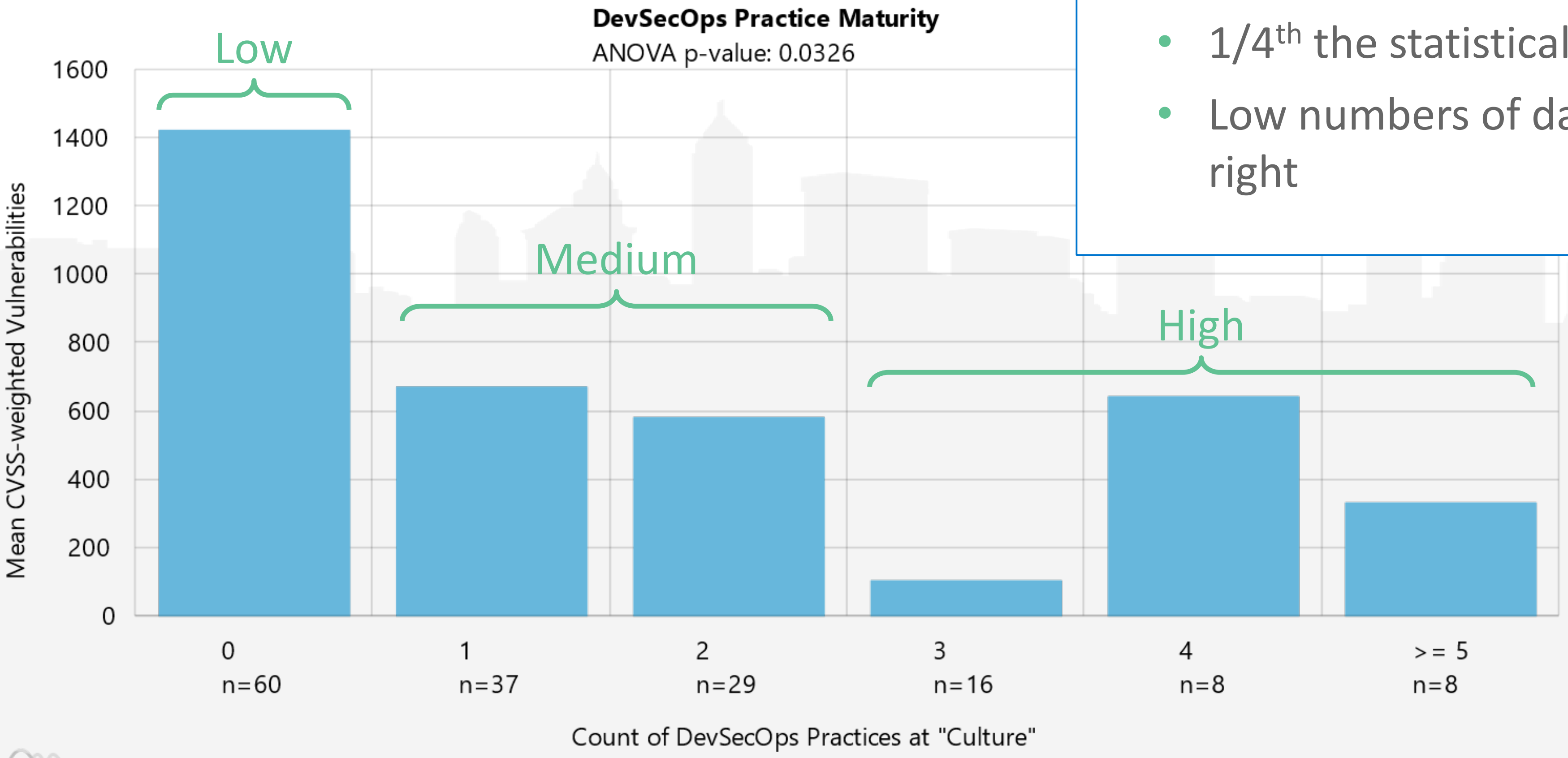
Swap **Anecdotes** for ***Evidence***

Replace **Folklore** with ***Facts***

Larry Maccherone
@LMaccherone
LinkedIn.com/in/LarryMaccherone

Extra Slides

Investigative loose end...
Why Just Low/Medium/High Maturity?



Same data, more fine-grained x-axis:

- Same general trend, but…
- 1/4th the statistical significance, because…
- Low numbers of data points in buckets to the right

Dev*[Sec]*Ops is…
empowered engineering teams
taking ownership
of how their product
performs in production
[including security]

Dev[Sec]Ops Manifesto

Build security in
more than bolt it on

Rely on empowered engineering teams
more than security specialists

Implement features securely
more than security features

Rely on continuous learning
more than end-of-phase gates

Adopt a few key practices deeply and universally
more than a comprehensive set poorly and sporadically

Build on culture change
more than policy enforcement

Larry Maccherone
@LMaccherone | LinkedIn.com/in/LarryMaccherone

We, the Security Team…

COMCAST

Trust that Engineering Teams…

- Want to do the right thing
- Are closer to the business context and will make trade-off decisions between security and other risks

Pledge to…

- Provide information and advice so those trade-off decisions are more informed
- Lower the cost/effort side of any investment in developer security tools or practices

Understand that…

- We are no longer gate keepers but rather tool-smiths and advisors

$$\text{trust} = \frac{\text{credibility} + \text{reliability} + \text{empathy}}{\text{self-interest}}$$

The TRUST Algorithm