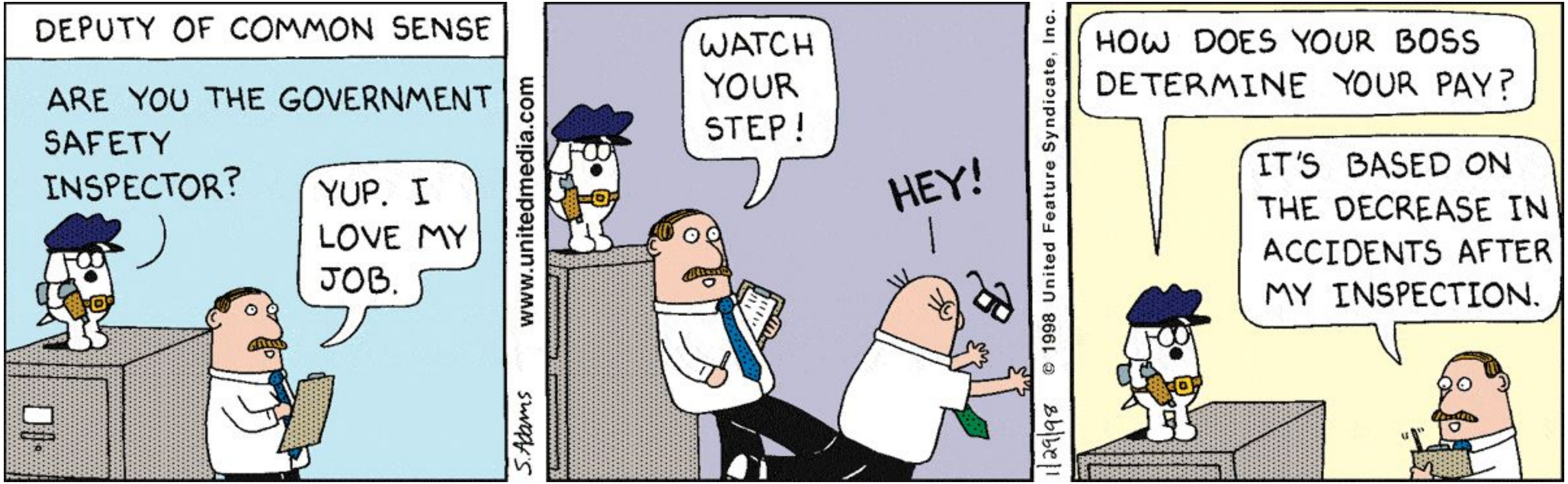
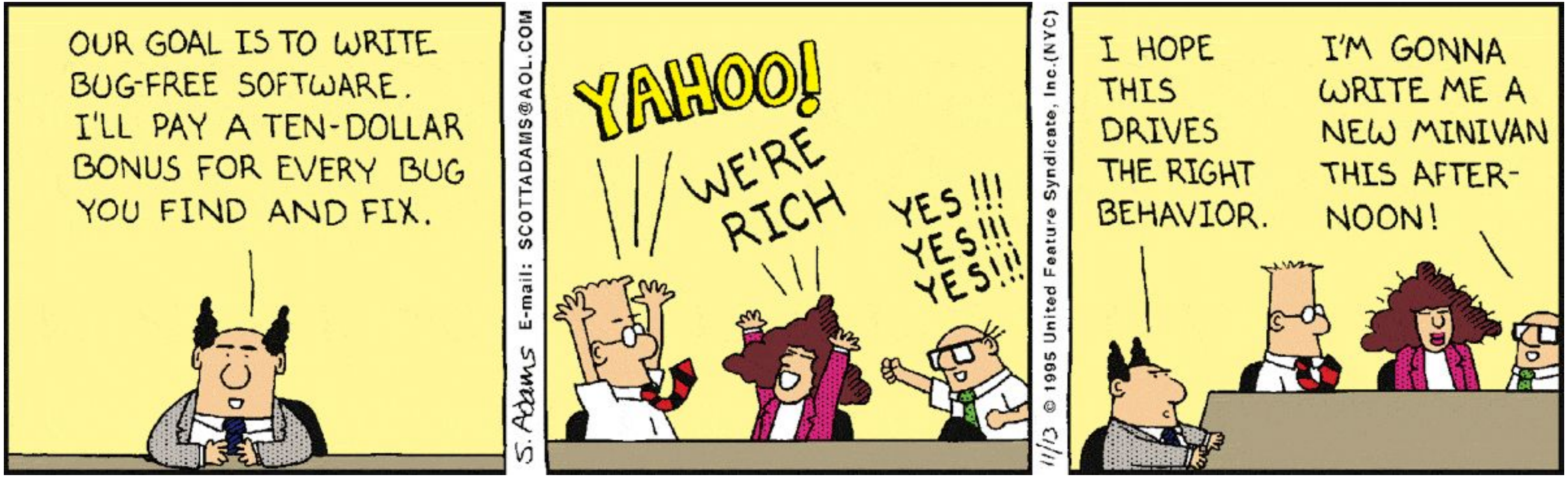
Useful Metrics and Reporting

(For Appsec and Vulnerability Management Programs)

Dilbert.

By
Scott Adams


DEPUTY OF COMMON SENSE
ARE YOU THE GOVERNMENT SAFETY INSPECTOR?
YUP. I LOVE MY JOB.
www.unitedmedia.com
S. Adams
WATCH YOUR STEP!
HEY!
11/29/98 © 1998 United Feature Syndicate, Inc.
HOW DOES YOUR BOSS DETERMINE YOUR PAY?
IT'S BASED ON THE DECREASE IN ACCIDENTS AFTER MY INSPECTION.


OUR GOAL IS TO WRITE BUG-FREE SOFTWARE. I'LL PAY A TEN-DOLLAR BONUS FOR EVERY BUG YOU FIND AND FIX.
S. Adams E-mail: SCOTTADAMS@AOL.COM
YAHOO!
WE'RE RICH
YES!!! YES!!! YES!!!
11/13 © 1995 United Feature Syndicate, Inc. (NYC)
I HOPE THIS DRIVES THE RIGHT BEHAVIOR.
I'M GONNA WRITE ME A NEW MINIVAN THIS AFTER-NOON!

Opheliar (Ophe) Chan

Director at Security Compass, Co-Lead of OWASP Toronto Chapter

10+ years in Security Consulting doing penetration testing, application/software security program consulting, vulnerability management, general security advisory , etc. More recently been helping to build a product implementation practice. Have also been: a dev, security researcher, technical writer.

Effectively: Security brain for hire with speciality in Appsec interpretive dance

Find me at: Opheliar.Chan@owasp.org

Standard Disclaimers

My opinions are my opinions. None of what I say represents the views of my employer or any groups I work with. They're just nice enough to let me speak in public without a gag order.

I'll credit anyone whose info/images I borrow, but I'm not making money off this, so please don't sue.

I show a couple products/projects because they're commonly used or an example I know of, not because I'm advertising for them or advising you specifically to use them (with one exception).

Scope and Objectives

What problem are we trying to solve?

What are we doing now?

How will we know we’re solving the problem?

How do we know if we’re not doing what we’re supposed to be doing?

What else do we need to solve the problem?

How will we know when we’re done?

Objectives

Characteristics of ‘Real Objectives’

- **S**pecific
- **M**easureable
- **A**ttainable
- **R**ealistic
- **T**imed

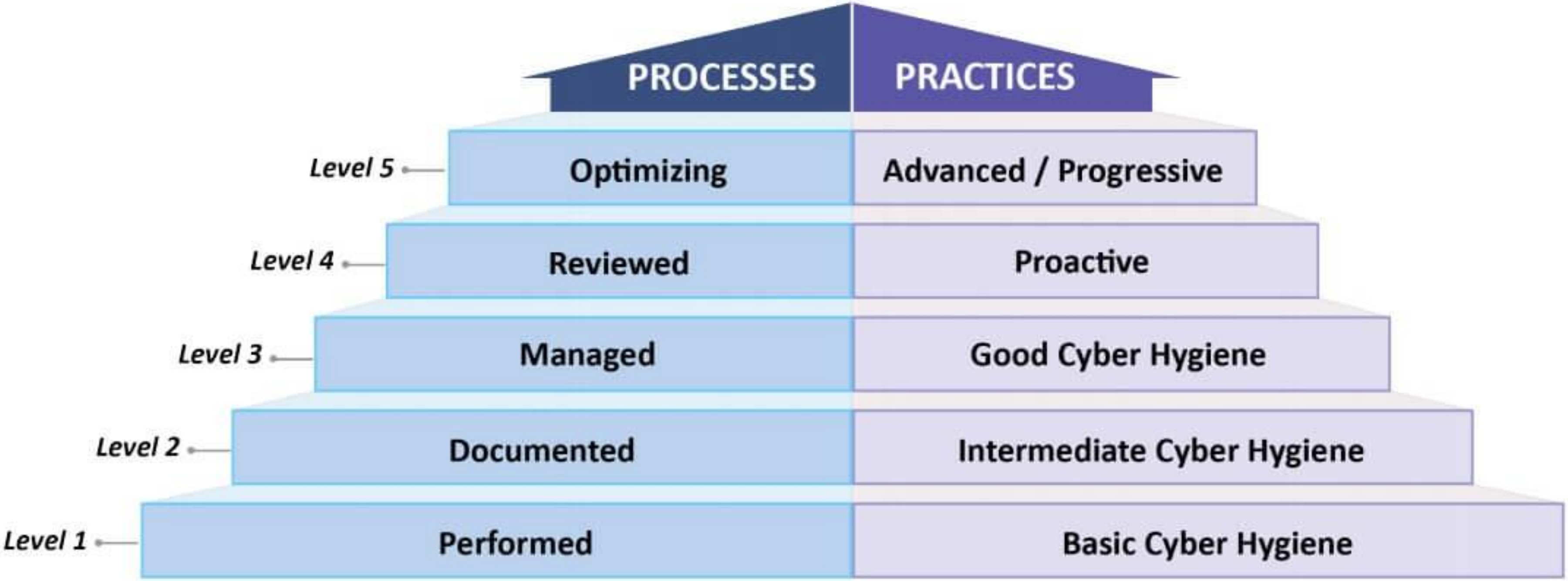
“Reduce risk by catching vulnerabilities before they make it into production.”

Program Guiding Principles

- **Empower the org to take easy and secure routes**
  - Provide clear explanation of controls and requirements
  - Focus on building easier paths to security and reusable elements

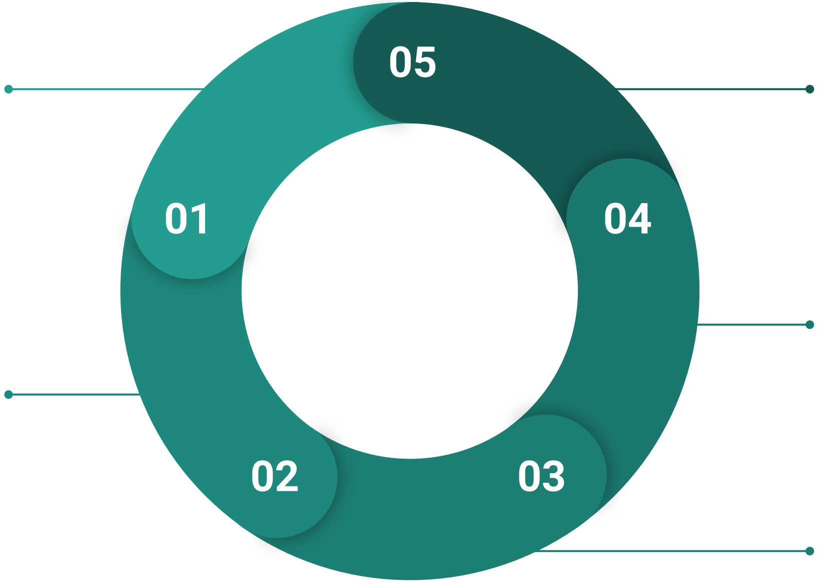
- **Help the business control their own destiny.**
  - What fundamental security controls are broken? then fix the root cause to prevent them from reoccurring

US Federal Cybersecurity Maturity Model



	PROCESSES	PRACTICES
Level 5	Optimizing	Advanced / Progressive
Level 4	Reviewed	Proactive
Level 3	Managed	Good Cyber Hygiene
Level 2	Documented	Intermediate Cyber Hygiene
Level 1	Performed	Basic Cyber Hygiene

https://www.acq.osd.mil/cmmc/draft.html



Watch

Agree on Problems, Objectives, Timelines

Teach & Fix

Test (Measurement)

Assess & Adjust (Reporting)

Two Realities

Calvin & hobbes, by Bill Watterson

Business Lines own risk

Reporting used for visibility & improvement

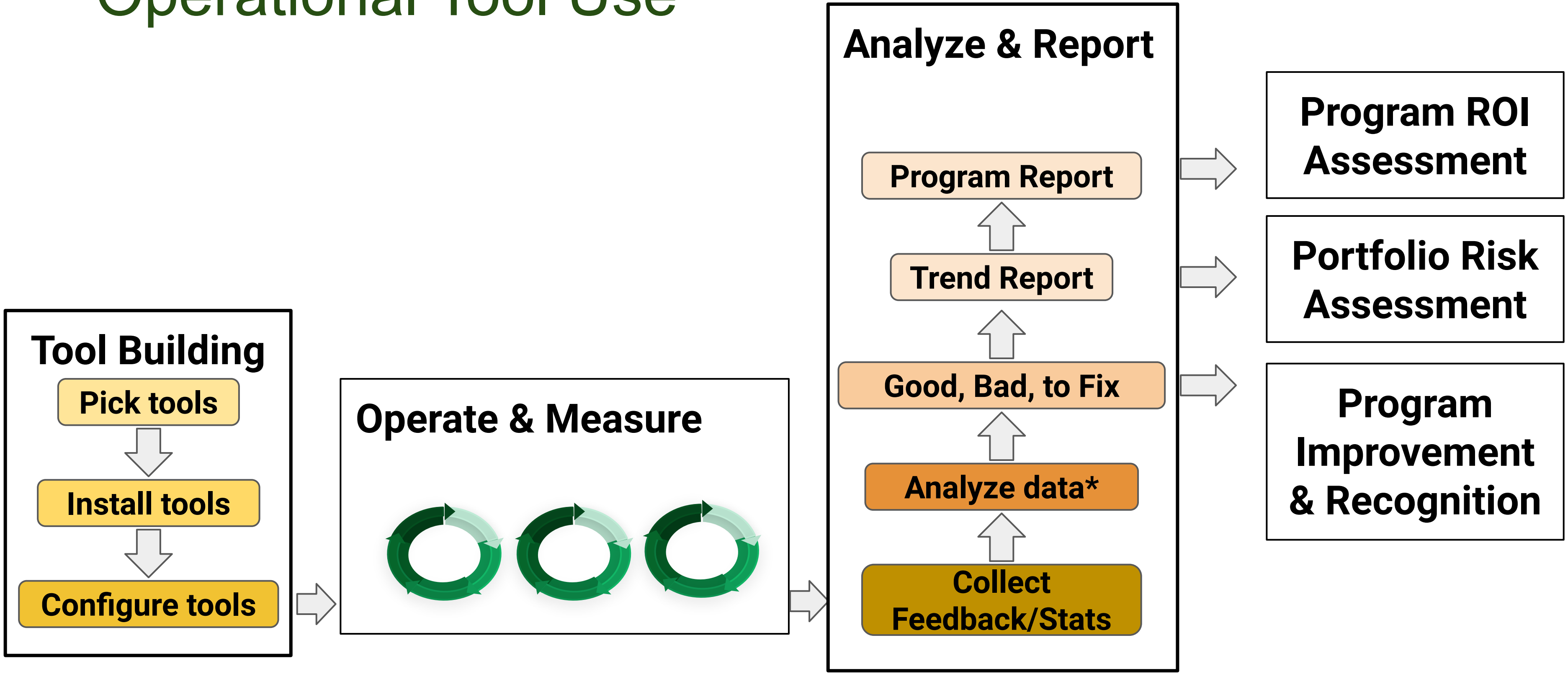
Infosec is accountable for risk

(even if business nods and say they own risk, they don't behave like it)

From Gunshow by KC Green

1. Metrics and reporting for visibility
2. Propose Business Case/Action Plan
3. Negotiate Details

Operational Tool Use



Tool Building

Pick tools

Install tools

Configure tools

Operate & Measure

Analyze & Report

Collect Feedback/Stats

Analyze data*

Good, Bad, to Fix

Trend Report

Program Report

Program ROI Assessment

Portfolio Risk Assessment

Program Improvement & Recognition

Metrics

Objectives:

- Situational awareness
- Measure progress towards objectives
- Describe what ‘better’ looks like
- Predict or plan for the future

Stakeholders: Your team, whoever is fixing the bugs, PMO, your bosses

Reporting

Objectives:

- Get other people to do things
- Show success/progress towards objectives
- Show what a course of action would do/not do (a.k.a build a business case)

Characteristics: accessible, targeted, action or information-based

Stakeholders: Your team, whoever is fixing the bugs, PMO, your bosses

Tactical Guiding Principles

Show your work!

No unexplained numbers

- All should come with a story and context that somehow indicates health, maturity, or informs action plans

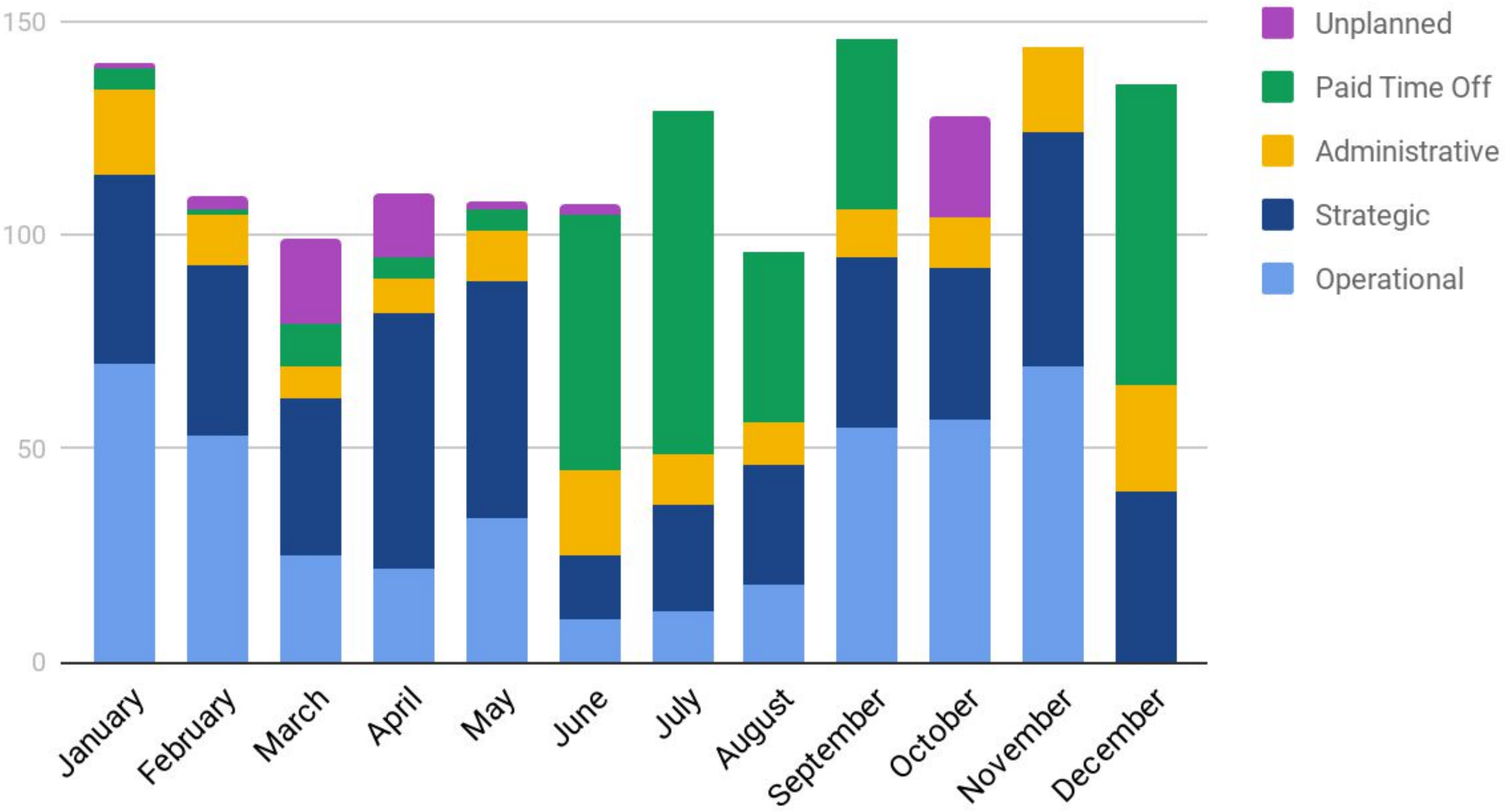
Sustained effort - Allocate consistent time, resources, and effort to measure/report consistently or it’s going to flop.

Focus on making progress!

Situational Awareness

What are we doing now?

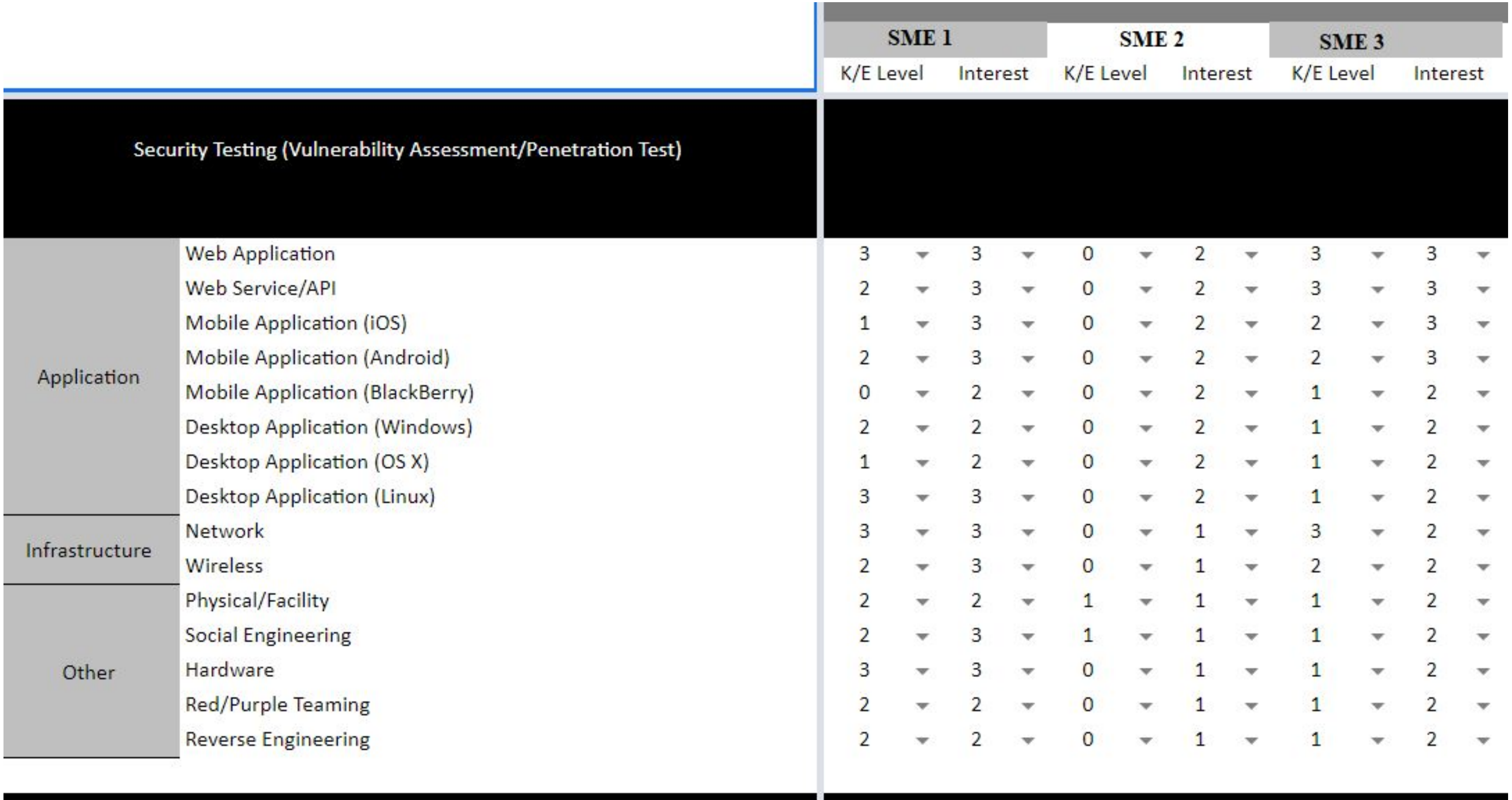
Work Cadence



Time & effort estimators

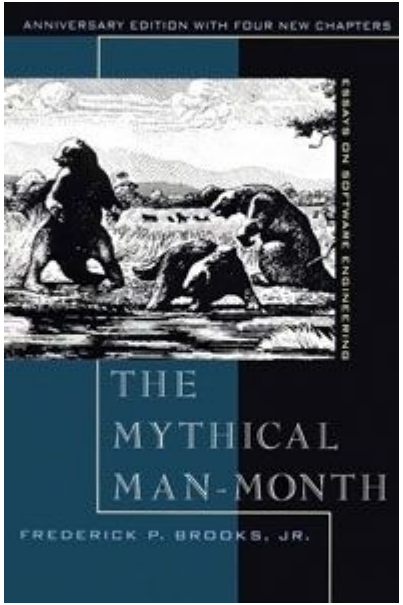
- Effort estimations
- Progress tracking
- Sanity checking

Primative Skills, Interest, Certification Matrix



		SME 1		SME 2		SME 3	
		K/E Level	Interest	K/E Level	Interest	K/E Level	Interest
Security Testing (Vulnerability Assessment/Penetration Test)							
Application	Web Application	3	3	0	2	3	3
	Web Service/API	2	3	0	2	3	3
	Mobile Application (iOS)	1	3	0	2	2	3
	Mobile Application (Android)	2	3	0	2	2	3
	Mobile Application (BlackBerry)	0	2	0	2	1	2
	Desktop Application (Windows)	2	2	0	2	1	2
	Desktop Application (OS X)	1	2	0	2	1	2
	Desktop Application (Linux)	3	3	0	2	1	2
Infrastructure	Network	3	3	0	1	3	2
	Wireless	2	3	0	1	2	2
Other	Physical/Facility	2	2	1	1	1	2
	Social Engineering	2	3	1	1	1	2
	Hardware	3	3	0	1	1	2
	Red/Purple Teaming	2	2	0	1	1	2
	Reverse Engineering	2	2	0	1	1	2

Warnings!


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

A few key take aways:

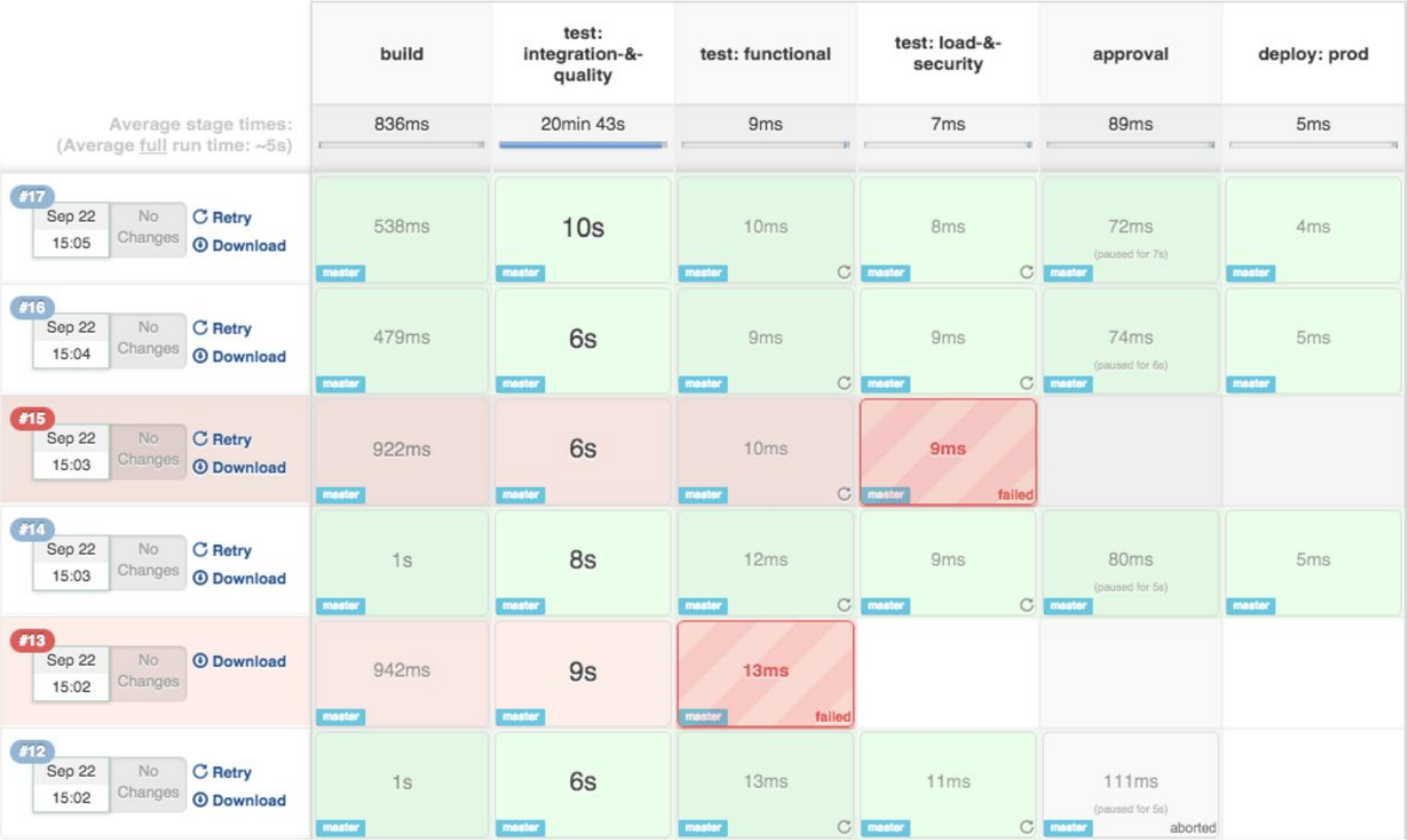
- Brooks's law: “*Adding manpower to a late software project makes it later*” - because of communication & learning overhead
- Some jobs take however long they take and managing them by deadlines just results in no results or reallllly undesirable shortcuts
- It matters who you assign to do the job
- “In a suitably complex system there is a certain irreducible number of errors. Any attempt to fix observed errors tends to result in the introduction of other errors.”

https://www.amazon.ca/Mythical-Man-Month-Anniversary-Software-Engineering-ebook/dp/B00B8USS14/

Portfolio/Code Coverage - The basics

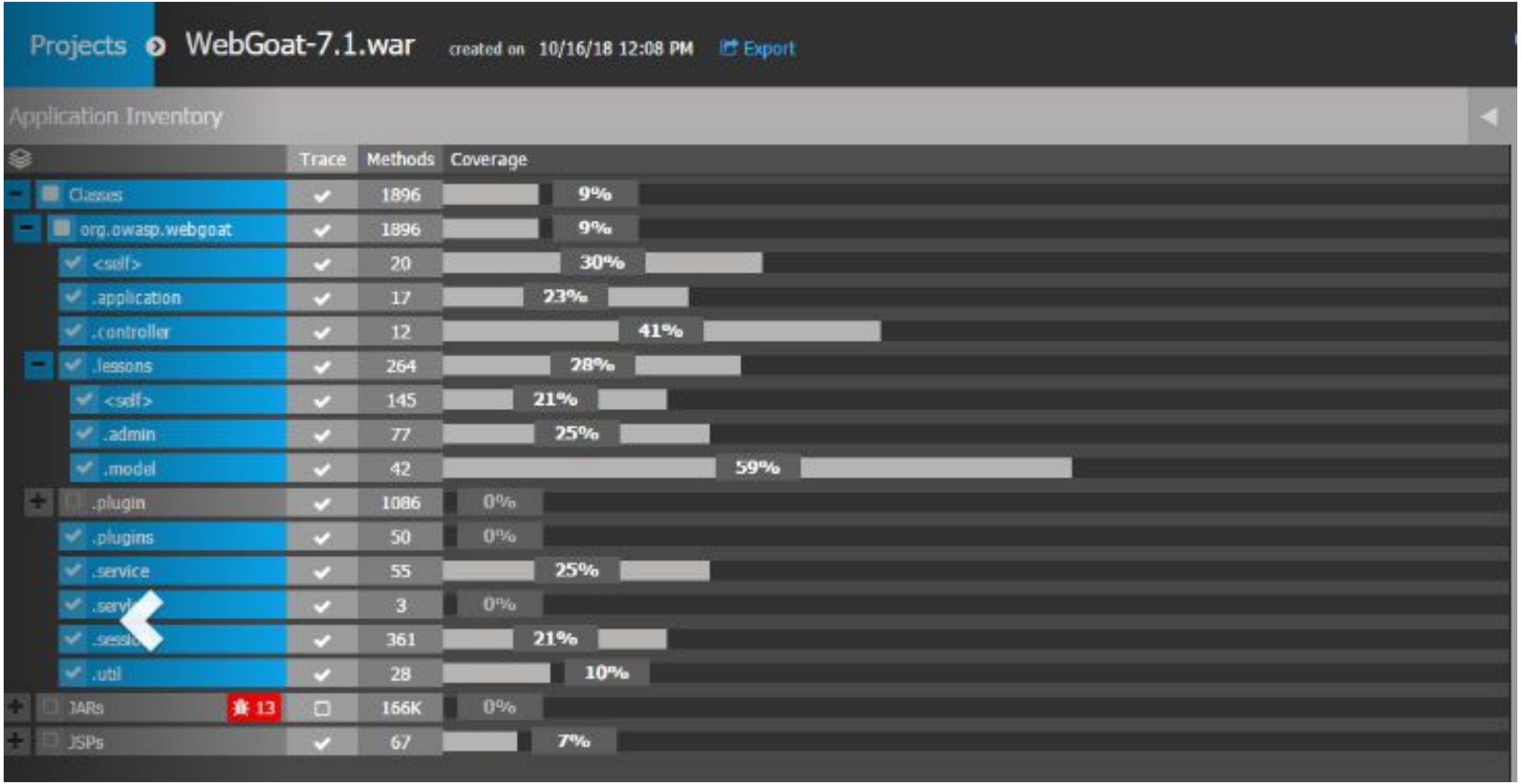
Classification	Covered	Out of Scope	Not Yet Addressed	Total
High	30	12	39	81
Medium	22	14	12	48
Low	12	15	14	41
N/A	1	2	15	18
TOTAL	**65**	**43**	**80**	**188**

Portfolio/Code Coverage - Managed



	build	test: integration-&-quality	test: functional	test: load-&-security	approval	deploy: prod
Average stage times: (Average full run time: ~5s)	836ms	20min 43s	9ms	7ms	89ms	5ms
#17 Sep 22 15:05 No Changes Retry Download	538ms master	10s master	10ms master	8ms master	72ms (paused for 7s) master	4ms master
#16 Sep 22 15:04 No Changes Retry Download	479ms master	6s master	9ms master	9ms master	74ms (paused for 6s) master	5ms master
#15 Sep 22 15:03 No Changes Retry Download	922ms master	6s master	10ms master	9ms master failed		
#14 Sep 22 15:03 No Changes Retry Download	1s master	8s master	12ms master	9ms master	80ms (paused for 5s) master	5ms master
#13 Sep 22 15:02 No Changes Download	942ms master	9s master	13ms master failed			
#12 Sep 22 15:02 No Changes Retry Download	1s master	6s master	13ms master	11ms master	111ms (paused for 5s) master aborted	

Code vs Test Coverage



Projects WebGoat-7.1.war created on 10/16/18 12:08 PM Export

Application Inventory

	Trace	Methods	Coverage
Classes	✓	1896	9%
org.owasp.webgoat	✓	1896	9%
<self>	✓	20	30%
.application	✓	17	23%
.controller	✓	12	41%
.lessons	✓	264	28%
<self>	✓	145	21%
.admin	✓	77	25%
.model	✓	42	59%
.plugin	✓	1086	0%
.plugins	✓	50	0%
.service	✓	55	25%
.servl	✓	3	0%
.sessio	✓	361	21%
.util	✓	28	10%
JARs 13	□	166K	0%
JSPs	✓	67	7%

http://code-pulse.com/

Getting Other People to do things

Return on Investment?
“Actionable plans”?

Classical tactics to get other people to do things (hopefully)

Tactics	Story	Measurement Examples
Competitions/ Comparisons	“This team is doing X approved activity more than you” “You’re falling behind your peers”	● Code Quality ● Time to remediate ● Types of findings ● Rate of recurrence ● % vulns overdue ● % vulns fixed before prod ● Checklist Compliance Requirements Met
Shame	“Your team is in the bottom 10 for this enterprise metric. ”	
Rewards & Recognition	“You guys are clearly more secure. You can have more fun projects, more autonomy, something else cool”	

Much better tactics to get other people to do things

Return on Investment, Actionable Remediation, and Usefulness

“Teams that took this training have fewer of this type of vulnerability”

“Your team keeps making these types of security mistakes. Please make these changes to your process and send all of them to training in this area so they stop repeating the same mistakes.”

“Since we implemented this toolset, we’ve found and fixed X number of design issues that might otherwise have been found by pen testers, up from 0% before”

Checklist Compliance Requirements Met

“What’s in it for me?”

“Look, if we don’t do this stuff we literally cannot do business because we will fail the next audit without question and can no longer take visa payments/X new big customers will not buy and pay for us, etc. Let’s get this done.”

You’ll need:

- A list of absolutely minimal requirements
- Assign someone(s) to negotiate/deliver/operate minimal viable implementation.
- Build this framework as the basic ‘paved path’ and monitor it

Peer Group Analysis - The Basics

Team	# apps	Total Findings	avg per app
Spaceship!	7	26666	3809.43
Cake	2	3513	1756.50
Mobile	6	5222	870.33
HoneyBadger	3	1643	547.67
Diversity	4	996	249.00
Rocket	4	335	83.75
Ninja	3	233	77.67
Assets	10	346	34.60
Detective	2	53	26.50

Peer Group Analysis

Team	System	High	Medium	Low	Total	LoC	avg high defects	avg defects
HoneyBadger	Kenya	145	290	320	755	2,299,100	0.0003%	0.0018%
	Ethiopia	450	278	146	874	1,134,562	0.0397%	0.0770%
	Speckled	0	3	3	6	523,333	0.0000%	0.0011%
Cake	Red Velvet	255	143	1676	2074	5,535,223	0.0046%	0.0375%
	Carrot	62	25	3	90	5,325,672,433	0.0000%	0.0000%
Ninja	Yazaemon	353	2	526	881	263,571,341	0.0001%	0.0003%
	Kirigakure	52	3	23	78	235,235	0.0221%	0.0332%
	Chiyome	536	236	100	872	6,864,788,888	0.0000%	0.0000%

Team Dashboard (basic snapshot)

System	Total Findings	avg high defects	defects/ loc	closed this month	open	new
Kenya	763	0.003%	0.018%	32	761	44
Ethiopia	874	0.040%	0.077%	23	851	25
Speckled	6	0.000%	0.001%	2	4	1

Training

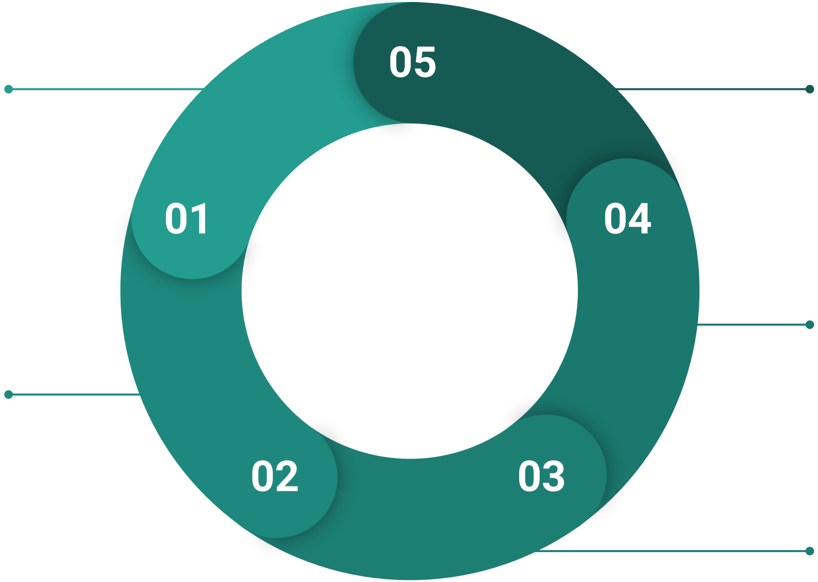
Unique Learners	2
Courses Taken	3

Top 3 issues by frequency

Vulnerability Name	CWE ID	OWASP (2017)	#
Cross-Site Scripting	79	A7	598
SSL Misconfiguration	310	A6	56
Java Deserialization	502	A8	120

Top 3 issues by severity

Vulnerability Name	CWE ID	OWASP (2017)	#
Java Deserialization	502	A8	120
Incorrect user management	286	A5	22
SQL injection	89	A1	15



01 Watch

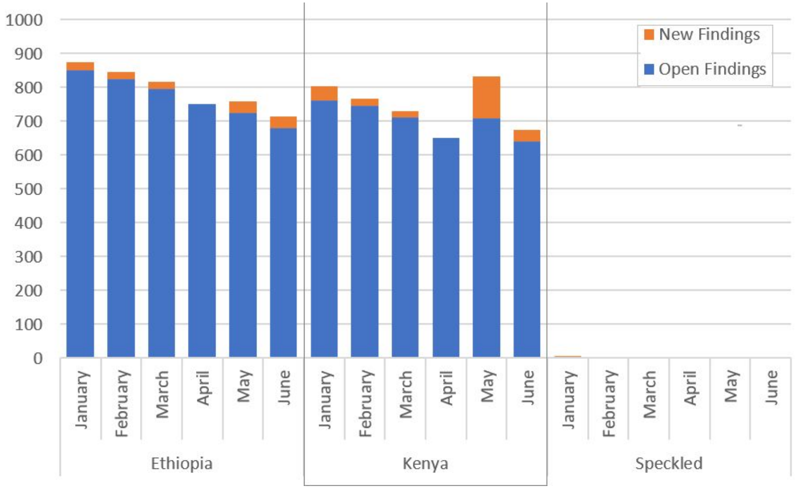
02 Agree on Problems, Objectives, Timelines

03 Teach & Fix

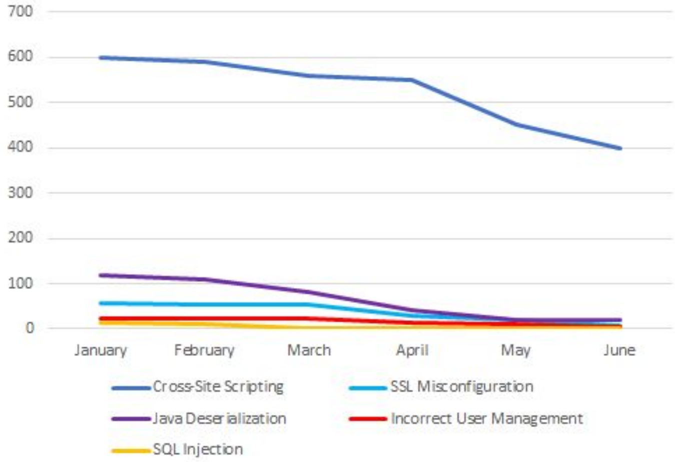
04 Test (Measurement)

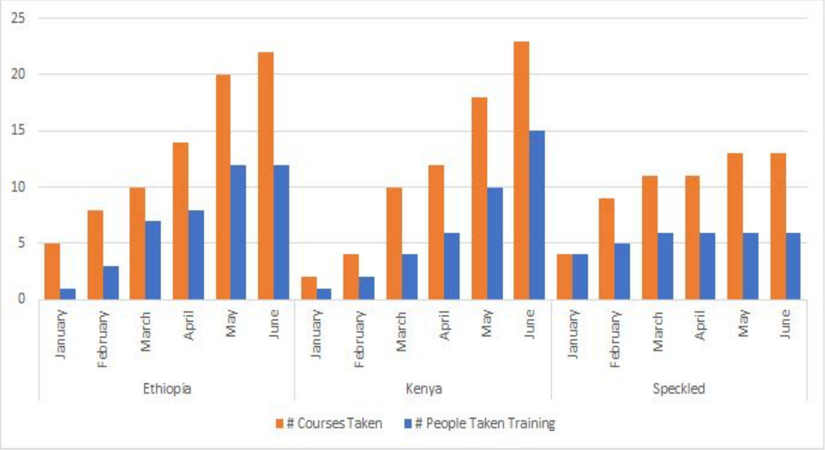
05 Assess & Adjust (Reporting)

Finding Trends



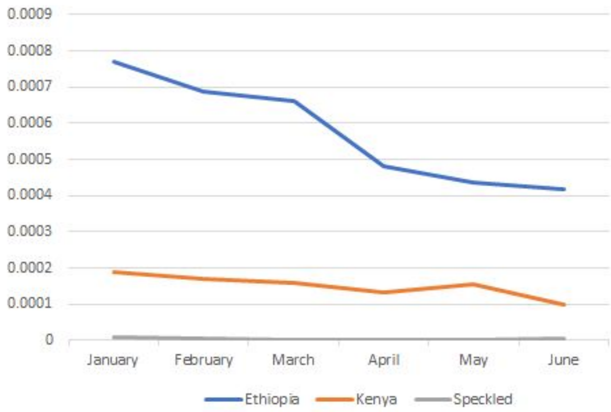
Top 3 Prevalent, Top 3 Severity Issues



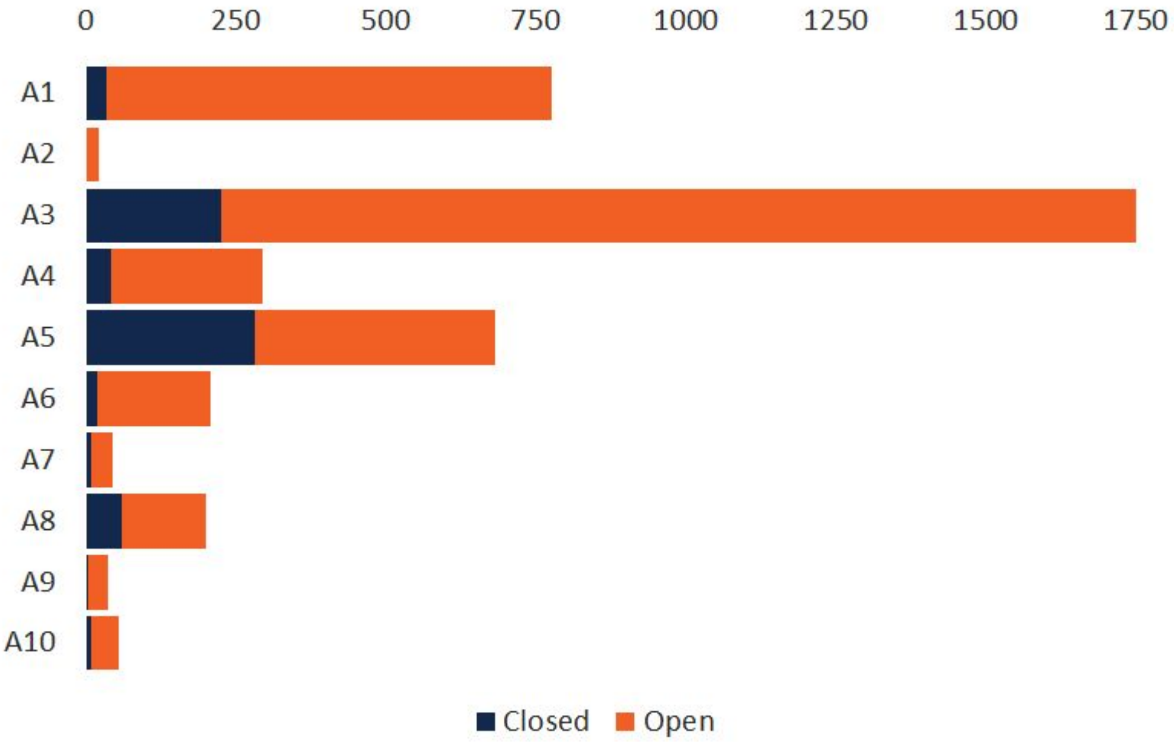


Training Trends

Code Quality Trends (Defect/LoC)



Vulnerability Trends by OWASP Top 10 2013 Categories



- Cross-site scripting (A3) is a clear standout
- Injection (A1) is primarily related to Java deserialization
- Security Misconfiguration (A5) lacks specificity, but is generally improved via stronger reference code or configuration standards

How are we doing?
What should we do next?

Metrics to tell you how your team is doing

(and to get them to do their current job better)

Are people using the services/tools voluntarily? Why/why not?

What are the bottlenecks/pain points?

Are we getting better at delivery?

Who is doing well and who is not?

- SLA’s met
- Adoption Rate
- Pull vs push interactions
- Hours spent on activities

Showing success and predicting the future

Do we have enough people to deliver?

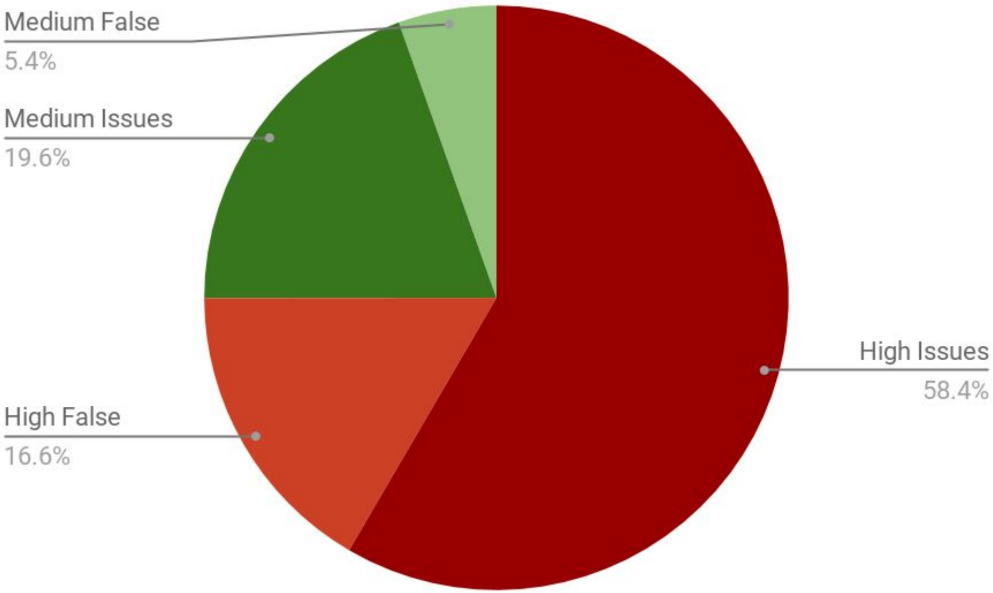
(Time to onboard and/or Time to triage **X** # things to onboard/triage) **/** # people

Return on investment & improvements over time

Real vs reported issues / time spent triaging

Code quality = # defects / # lines of code

Scan Tool Accuracy



Questions?

Guiding Principles

- Empower the org to take easy and secure routes
  - Provide clear explanation of controls and requirements
  - Focus on building easier paths to security and reusable elements

- Help the business control their own destiny.
  - what fundamental security controls are broken? Fix the root cause to prevent them from reoccurring