



**Jason Martin**

Vulnerability Governance, Manager  
Information Security & Risk Management

October 2022



# Minimum Viable Security - How to Get Started

The contents of this communication are the property of Highmark Health and should only be used for informational purposes. This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution. This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use.



# Common Obstacles with Secure Software Development

---

- Writing code is hard
  - Lack of security skills
  - Legacy software
  - Best practices are insufficient
  - Lack of risk focus, lack of audit and control points
  - Wrong automated tools
- Unsupervised collaboration
  - Emphasis on speed
  - Vulnerabilities in deployment pipeline
  - Unprotected production environment
  - Lack of security requirements traceability

# Dev<sup>^</sup>Ops Journey - Program Goals

---

## Sec

1. Add a robust platform and product security playbook to the application and product development services

---

2. Ensure a robust training and communications plan for secure development practices and application security best practices (i.e. OWASP)

---

3. Achieve realization of playbook and training through measuring frequency and density of security testing



# What is the challenge?

---

- Is every piece of software and subcomponent known? **NO**
- Do we understand what vulnerabilities are present in ALL our apps? **NO**
- Are our app teams staying on top of resolving problematic software? **NO**



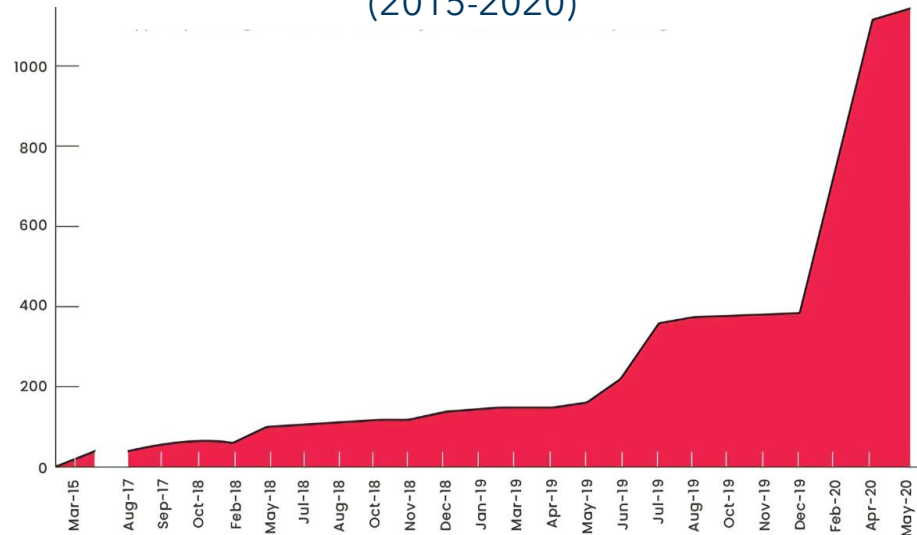
# Why does it matter?

**97%** of commercial code contains at least some open source codes<sup>1</sup>

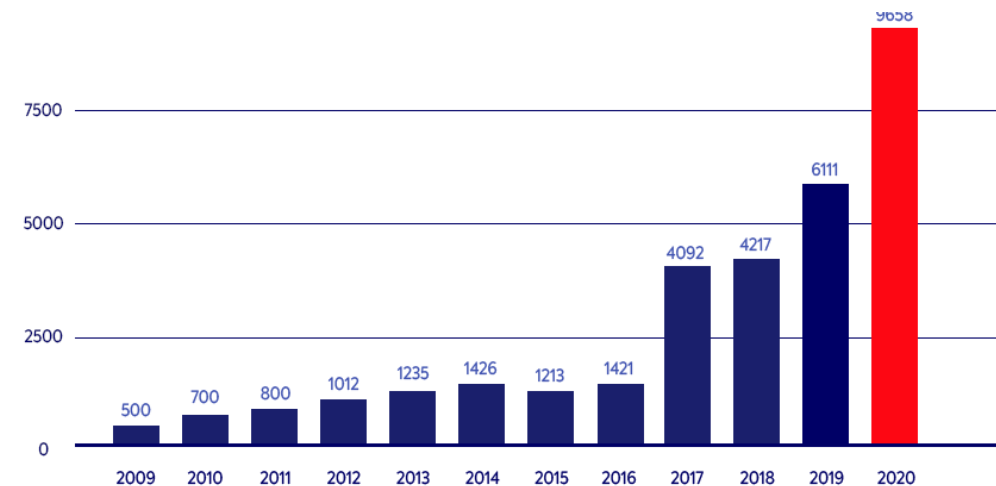
**81%** of codebases contain an outdated version of open source<sup>2</sup>

**62%** of breaches originated from a compromised software component<sup>3</sup>

Software Supply Chain Attacks<sup>4</sup>  
(2015-2020)



Open Source Vulns per Year<sup>5</sup>  
(2009-2020)



<sup>4</sup>Sonatype Software Supply Chain Attack report 2020

<sup>1+2</sup>Synopsys OSSRA report 2022

<sup>5</sup>Mend Annual Report, Open Source Vulnerabilities 2021

<sup>3</sup>Verizon Data Breach Investigations Report 2022

# How do we solve it?

A list of low hanging fruit that can be implemented early on to prevent most major, obvious software security issues.



## Objective

- Can it be Secured?
- Is it Secure?
- Is it staying Secure?



## Task

- Document key control objectives based on CI/CD framework, regulatory actions, and best practice.
- Identify Inter-dependencies and vital records.
- Establish authoritative source(s) for app data



## Deliverable

- App dev security posture measures compared against the various lines of business on a reoccurring basis.
- Consolidated quantitative impacts and risk due to inefficient secure coding.
- List of critical portals and web apps

# Minimum Viable Security Scorecard

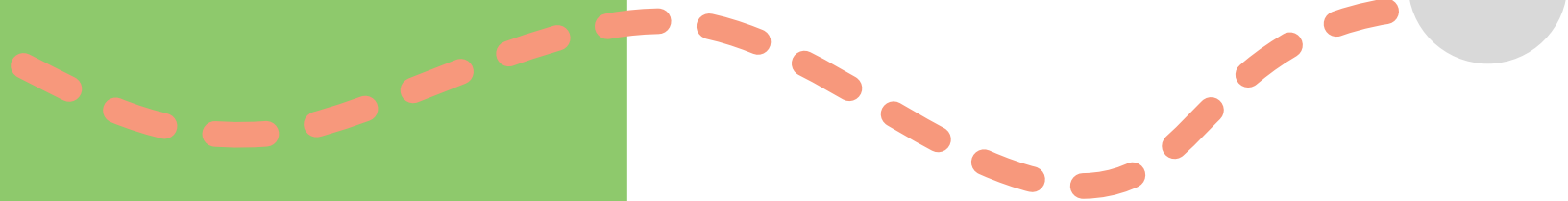
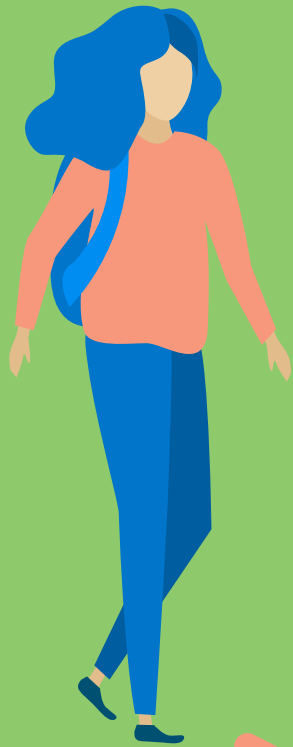
Step	Step 1: Discovery and Scope				Step 2: Security Requirements Checklist						Step 3: Validate / Reporting	Notes
Source	RSA Archer				Archer	Dev Team	SonarQube	GRC	Security	IAM	MVS	
App Name	Domain	Type	Internet Facing	Risk / Priority	Asset Data Quality Verified? (Y / N)	URL verified (Y/N)	Static App Sec Testing tool	SSP Completed (Y/N)	% of Open Blockers	MFA (Y/N)	Health Status	
App A	Z	Portal	No	1 - Critical	Yes	Yes	YES	YES	0%	Yes	A - Great	
App B	X	Portal	Yes	1 - Critical	Yes	Yes	YES	YES	0%	No	B - Good	
App C	Y	Portal	Yes	1 - Critical	Yes	Yes	YES	YES	0%	No	B - Good	
App D	U	App	No	2 - Significant	No	Yes	YES	YES	5%	No	C - Fair	Needs information validated
App E	X	App	No	2 - Significant	No	No	YES	No	75%	No	D - poor	Need URL to be verified/provided
App F	X	App	Yes	5 - Low	Yes	No	YES	No	3%	No	C - Fair	
App G	R	App	Yes	5 - Low	Yes	Yes	YES	YES	0%	No	B - Good	
App H	X	App	Yes	5 - Low	Yes	No	YES	YES	10%	No	C - Fair	Need URL to be verified/provided
App I	X	App	No	2 - Significant	Yes	No	unknown	No	n/a	No	D - poor	Need URL to be verified/provided
App J	Z	App	No	2 - Significant	Yes	n/a	YES	YES	6%	No	C - Fair	
App K	S	App	No	5 - Low	Yes	No	YES	YES	5%	No	C - Fair	Need URL to be verified/provided
App L	X	App	No	5 - Low	No	No	YES	YES	50%	No	D - poor	Check in Archer
App M	W	App	No	5 - Low	Yes	n/a	unknown	YES	5%	No	C - Fair	Needs information validated

Legend: A - Great B - Good C - Fair D - Poor

# Next Steps

---

Where do we go from here?

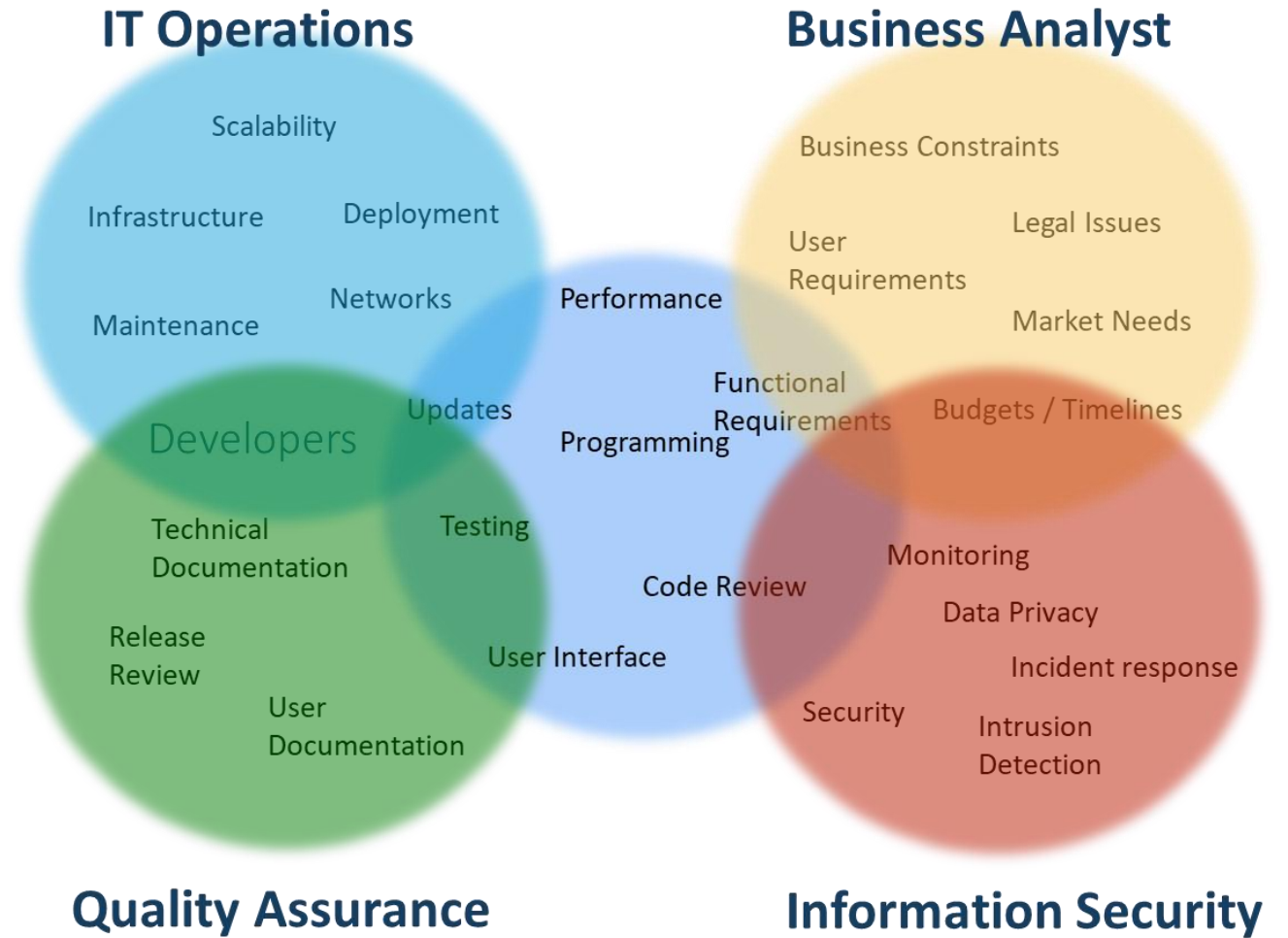




# Responsibility Assignment - RACI





## MVS Checklist:

- Perform quality / security testing – Level 1 ✓
- Meet the data quality conditions – Level 1 ✓
- System Security Plans (SSP) – Level 1 ✓
- Use a code repository / artifact mgmt – Level 2 ✓
- % of Open Findings Blocker and/or Critical – Level 2 ✓
- Multi-factor authentication (MFA) – Level 3 ✓
- Interactive Application Security Testing (IAST) – Level 4 ✓
- Continuous Threat Modeling – Level 5 ✓



# Responsibility Assignment - RACI

## MVS Checklist:

- Perform quality / security testing – Level 1 
- Meet the data quality conditions – Level 1 
- System Security Plans (SSP) – Level 1 
- Use a code repository / artifact mgmt – Level 2 
- % of Open Findings Blocker and/or Critical – Level 2 
- Multi-factor authentication (MFA) – Level 3 
- Interactive Application Security Testing (IAST) – Level 4 
- Continuous Threat Modeling – Level 5 



**RESPONSIBLE**

**DevOps &  
IT Operations**  
**(Service Owner)**



**ACCOUNTABLE**

**Business Analyst /  
Line of Business**  
**(Product Owner)**



**CONSULTED**

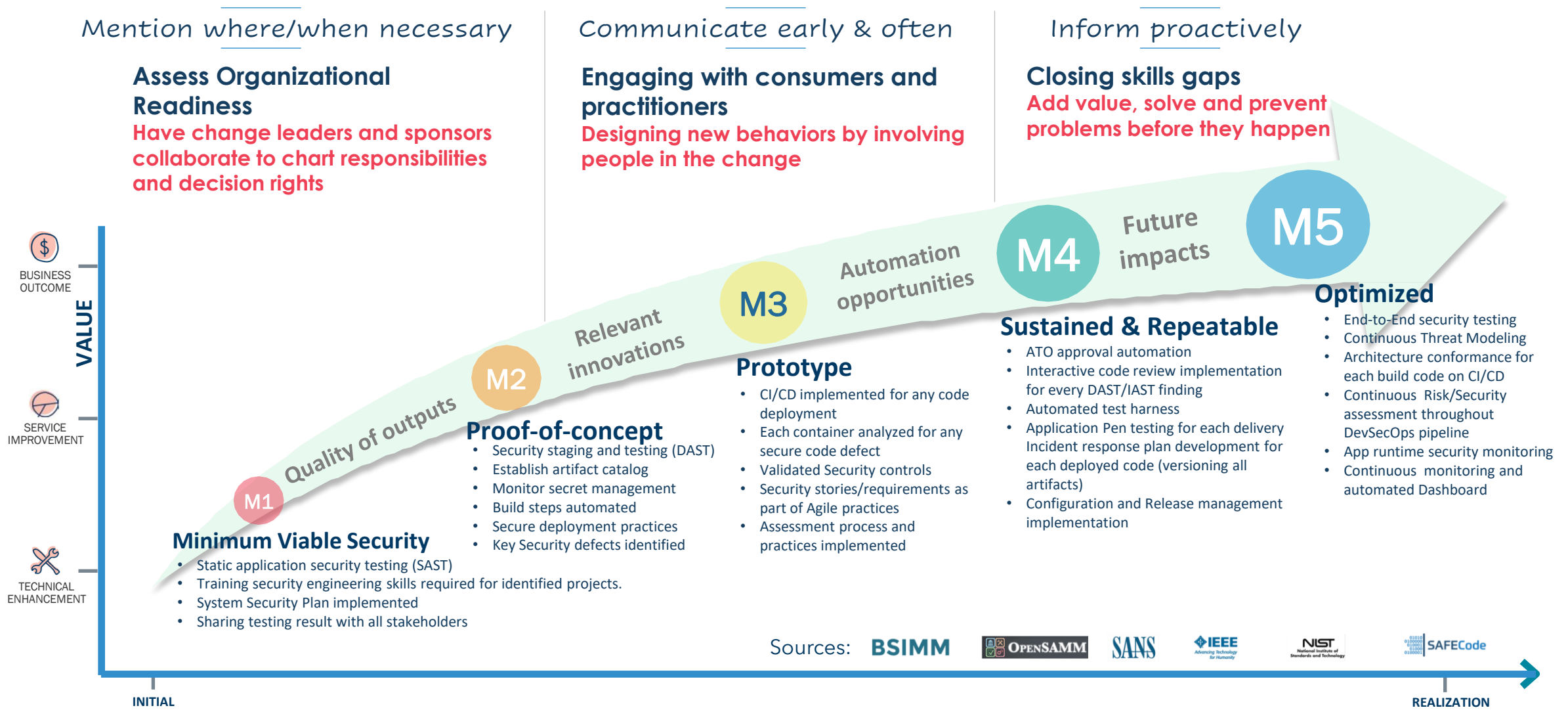
**Quality Assurance /  
Performance Testing**  
**(Process Analyst)**



**INFORMED**

**Information Security**  
**(Service Manager)**

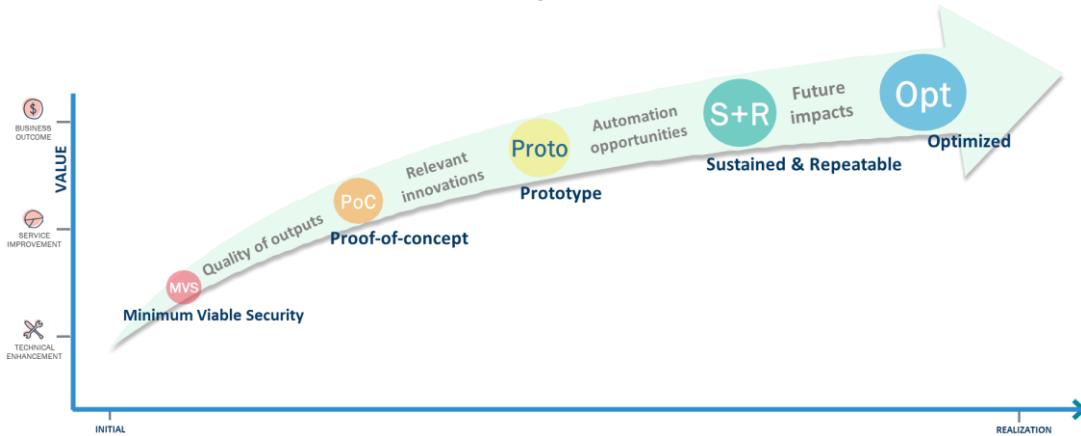
# DevSecOps Journey - Incremental Maturation / Implementation



# Maturity Score Performance Measure Rubric

Metric	Description	Unacceptable Range	Below Threshold Range	Threshold Range	Target Range	Exceeding
	<b>Maturity Curve</b>	-	M 1 + M 2		M 3 + M 4	M 5
<b>Eliminate Vulnerabilities*</b>	Application Security Defect Density (per 1,000 lines of code)*	> 5	2 - 5	1 - 2	1 - 0	0
	% of security testing coverage (SAST)	< 80%	80.1% - 84.9%	85% - 94.9%	95% - 98.9%	> 99%
	# of Average days to remediate (i.e., lead time)	>120	90 - 119	60 - 89	45 - 59	<30
	% of Open High/Critical Vulnerabilities aged < 30 Days	< 85%	85.1% - 90.9%	91% - 95.9%	96% - 97.9%	>98%

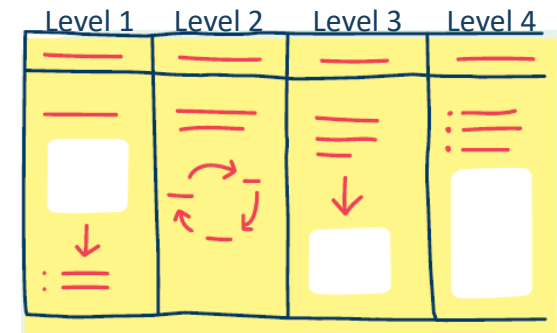
## Maturity Curve



Guidelines for the product teams



## Developer Playbook



# DevOps Journey - MVS Preliminary Tasks

Sec

M1

## Minimum Viable Security

### Monitoring detections

- Establish monitoring goals to visualize basic security metrics
- Monitor SAST through manual or automated means
- Review/share SAST results with all stakeholders

### SAST

- Setup Common Security quality gates
- Integrated SAST into the commit pipeline to identify security code defects each time the software is built or packaged
- SAST results remediated findings as prioritized security code defects

### System Security Plan

- Identify and list all apps including owner and artifact locations
- List all app build and environment components(libraries)

### Dependencies List and Open-Source Libraries Governance

- Create and maintain software dependency for each build
- Track ALL open-source libraries used in code development and build process

### Mandatory Training

- Train key team members on how to analyze and remediate SAST results
- Identify top security training objectives
- Drive effectiveness of security trainings:

### Standards & Practices

- Identify and develop risk-based threat modeling
- Identify and gather metrics from current SW delivery/deployment pipeline
- Develop MVP of common dashboard including Security findings
- Establish monitoring goals to visualize basic security metrics

New

Application Security Health		M 1	M 2	M 3	M 4	M 5
	Business App Owner Domain Z	4	Under Review			
Main	Business App Owner Domain Y	2	2	Not Started	Not Started	Not Started
	Business App Owner Domain X	2	Under Review	Not Started	Not Started	Not Started
	Business App Owner Domain W	2	Under Review	Not Started	Not Started	Not Started
	Business App Owner Domain V	1	Under Review	Not Started	Not Started	Not Started
	Business App Owner Domain U	3	Under Review	Not Started	Not Started	Not Started
	Business App Owner Domain T	N/A	N/A	N/A	N/A	N/A
	Business App Owner Domain S	2	5	Not Started	Not Started	Not Started
	Supporting	Business App Owner Domain R	1	Under Review	Not Started	Not Started
Business App Owner Domain Q		2	Under Review	Not Started	Not Started	Not Started

# Minimum Viable Security Scorecard

Enables application owners and business leaders to monitor their performance in order to make informed decisions that alleviates security vulnerabilities more rapidly.



# For more information

---

- SEI – Carnegie Mellon University
  - DevOps Blog:  
<https://insights.sei.cmu.edu/devops>
  - Webinar :  
<https://www.sei.cmu.edu/publications/webinars/index.cfm>
  - Podcast :  
<https://www.sei.cmu.edu/publications/podcasts/index.cfm>
- DevSecOps:  
<http://www.devsecops.org>
- Rugged Software:  
<https://www.ruggedsoftware.org>
- Once Click DevOps deployment
  - <https://github.com/SLS-ALL/devops-microcosm>