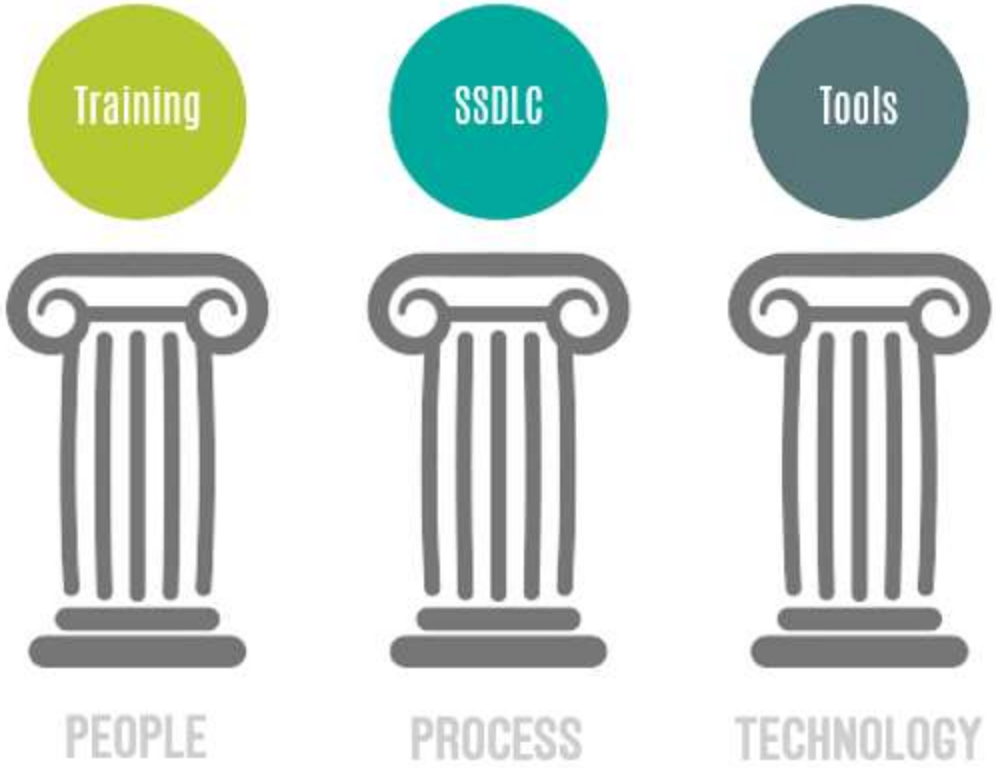
Application Security
On-Prem and in the Cloud

Din Cox, Ph.D., CISSP, ISSAP, CSSLP, CEH

Our Focus

Application Security Program



About Me - 17 yrs.

Application Security & Cloud - Now

Security Researcher/Bug Hunting

Security Architecture - Past

Past - Network Security Engineering

Software & Systems Engineering - Past

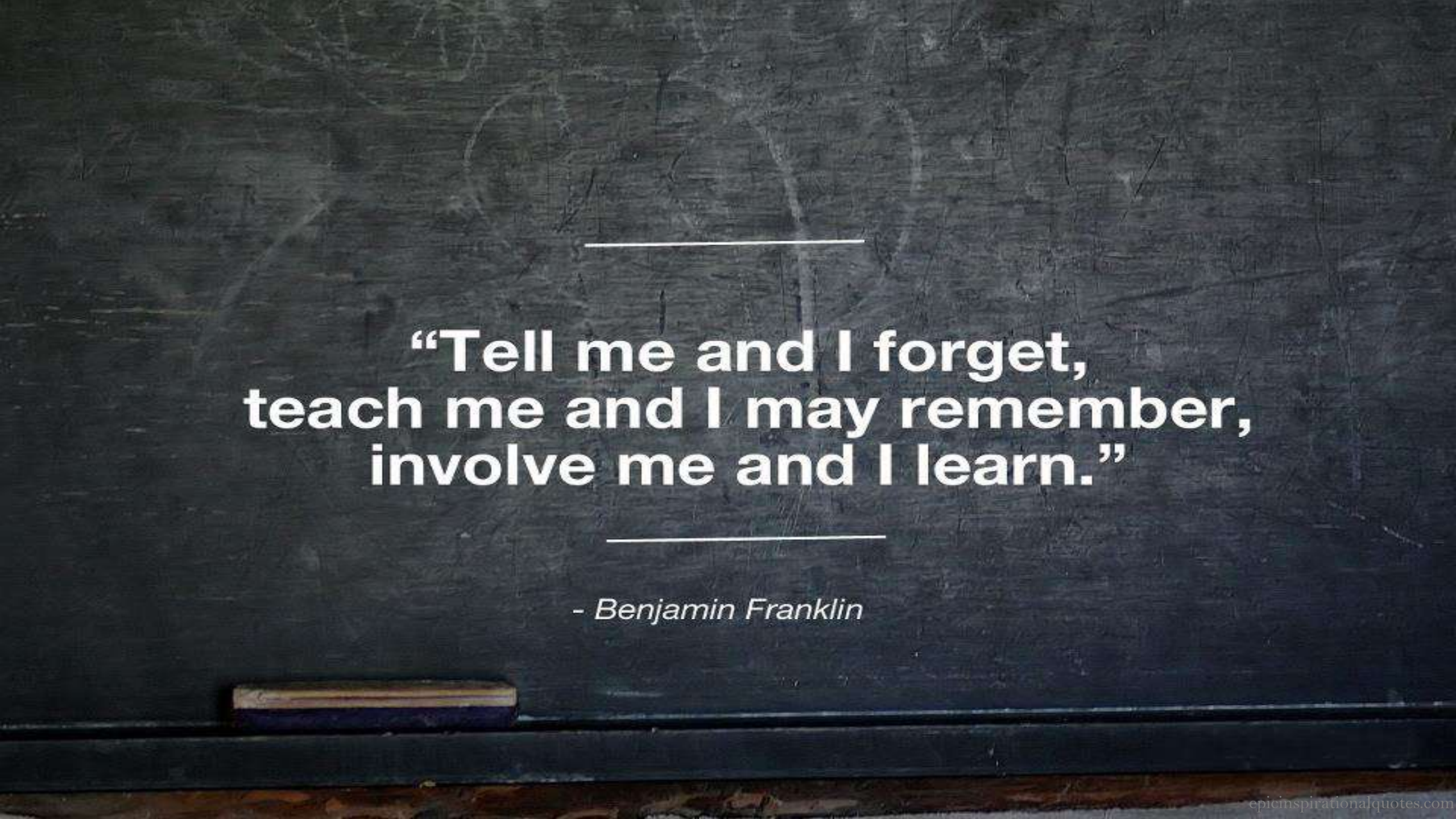
Leadership roles in between

First things First

Software defects with security ramifications fall into two distinct categories:

- Implementation Bugs (Cross-site scripting, SQL Injection, etc.)
- Design Flaws (Architecture, business logic, insecure fail state, etc.)

1



Key

- Developers are first line of defense
- Should be MANDATORY for anyone who writes code
- Tailored to the programming language / framework
- Ongoing education (brown bags, conferences, etc.)
- Secure code training throughout SDLC phases

Application Inventory

- Document each application within an inventory
- Document security related details (sensitive data, exposure, existing controls)
- Identify POCs to ensure continuity
- Leverage automation to help populate data. Always Verify.


SSDLC
PROCESS

SSDLC

Software Development Lifecycle without security

- Developer domain
- Silo’d approach
- Systematic and proven process
- Wasn’t design with security in mind

SSDLC

Secure Software Development Lifecycle

- Developer and Application Security team domain
- Unified approach to secure software
- Prescriptive, practical, proactive
- Eliminate security problems early

Approach

SSDLC for Agile development reorganizes security practices into:

- One-Time practices
- Every-Sprint practices
- Bucket practices

One-Time

Foundational security practices that must be established once at the start of every new project.

Every-Sprint

Essential security practices that should be performed in every release.

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	17. Execute Incident Response Plan
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modelling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

Bucket

Important security practices that must be completed on a regular basis but can be spread across multiple sprints during the project lifetime.

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	17. Execute Incident Response Plan
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modelling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

Bucket

Important security practices that must be completed on a regular basis but can be spread across multiple sprints during the project lifetime.

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	17. Execute Incident Response Plan
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modelling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

Federal SDLC Requirements

NIST SP 800-64 Security Considerations in the System Development Life Cycle:

"*Special attention should be placed upon code repositories with an emphasis on systems that support distributed code contribution with check-in/check-out functionality.*"

" *…System maintenance, security, and operational considerations…*"

NIST SP 800-65 I… …ecurity into the Capital Planni… …vestment Control Process

"*Project manager responsibilities include the following: Developing a project management plan that integrates security throughout the life cycle.*"

NIST SP 800-53, control SA-3 guidance:

"*The security e… …g principles in SA-8 cannot b… …ed if individuals that **d… …d test** information systems and …mponents (including information …nology products) do not understand security…. It is equally important that **developers include individuals on the development team** that possess the requisite security expertise and skills to ensure that needed security capabilities are effectively integrated into the information system.*"

*This control is selected for all system categorizations including LOW impact systems.

DON'T ATTEMPT TO READ


Tools
TECHNOLOGY

Common Approach

- Dynamic scanning – CDM (Appscan, Web Inspect etc.)
- Penetration testing / Vulnerability Assessment

//NOT Comprehensive enough//

Proactive Approach

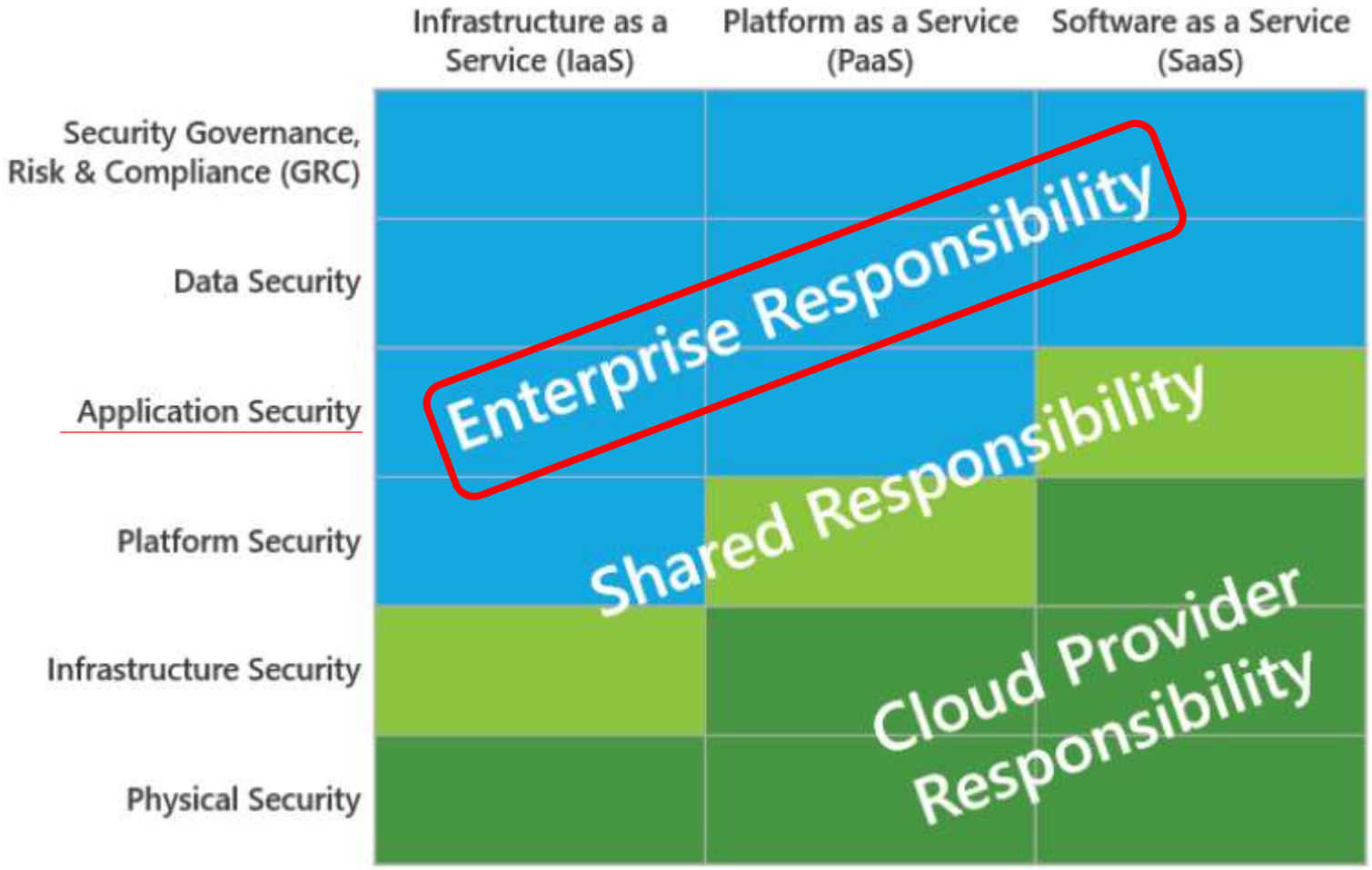
- Static Analysis
- Dynamic Analysis
- Vulnerability Assessment & Penetration Testing
- Activity Monitoring & Metrics
- Web Application & Database Firewalls

WHAT DOES THIS ALL MEAN

….in the Cloud?


I MIGRATED OUR NORTHERN DATA CENTER TO THE CLOUD.
Dilbert.com DilbertCartoonist@gmail.com
BUT THE CLOUD STOPPED WORKING AND I CAN'T FIND THE PHONE NUMBER FOR OUR CLOUD GUY.
SO . . . WHATEVER.
7-5-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick
YOU LOST OUR DATA CENTER?
THAT'S ONE WAY TO LOOK AT IT.

Application Security Responsibility?



Cloud Ready / Friendly Application?

Being able to determine impact if:

- Cloud provider / employee accessed the application
- Manipulation of function or process by unauthorized entity
- Inability to provide expected results / unexpected changes
- Loss of application availability

Application Risks

Two key risks to watch out for:

- Multi-tenancy
- Third-party administrators

https://www.pinterest.com/gettelsarasota/real-estate-jokes/

New Frontier

Crucial for developers to understand security requirements specific to:

- Deployment model (Public, Private, Community, Hybrid) where the application will operate / run
- Service model (IaaS, PaaS, SaaS)

Help to determine organization and provider security responsibilities

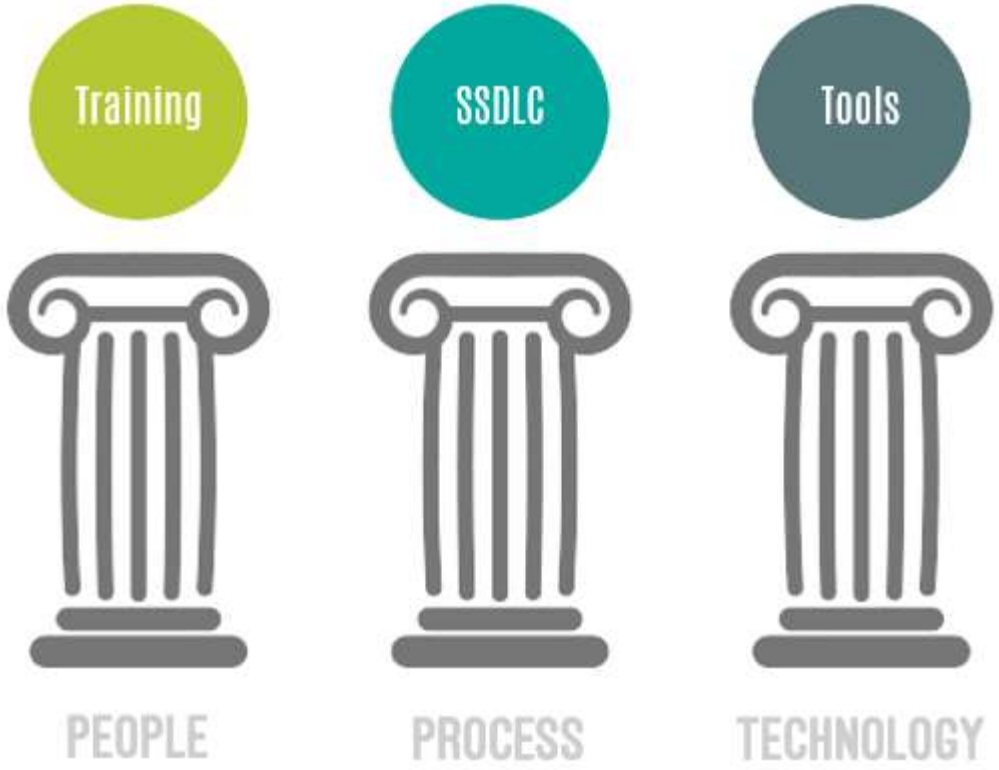
Configuration Management & Versioning

Having proper software configuration management and versioning will go a long way.

- CI/CD pipeline
- Tools (Chef, Puppet)
- Clearly define processes

Summary

Application Security
On-Prem and in the Cloud



Resources & References

- https://www.microsoft.com/en-us/sdl/
- https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet
- https://www.bsimm.com/
- https://www.owasp.org/index.php/Category:OWASP_CLASP_Project
- http://www.opensamm.org/
- http://www.isecom.org/research/
- https://www.owasp.org/images/6/67/OWASPApplicationSecurityVerificationStandard3.0.pdf

Thank
You!

Any Questions?