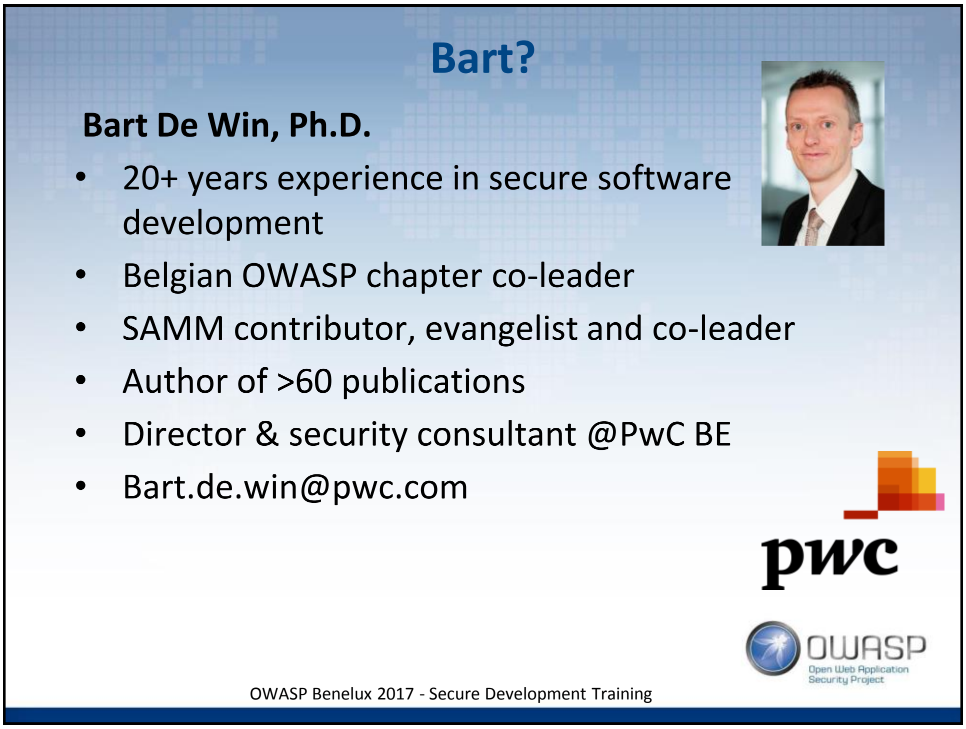
Secure Development: Models and Best Practices

Bart De Win

Bart.DeWin@owasp.org

OWASP Benelux 2017 - Secure Development Training

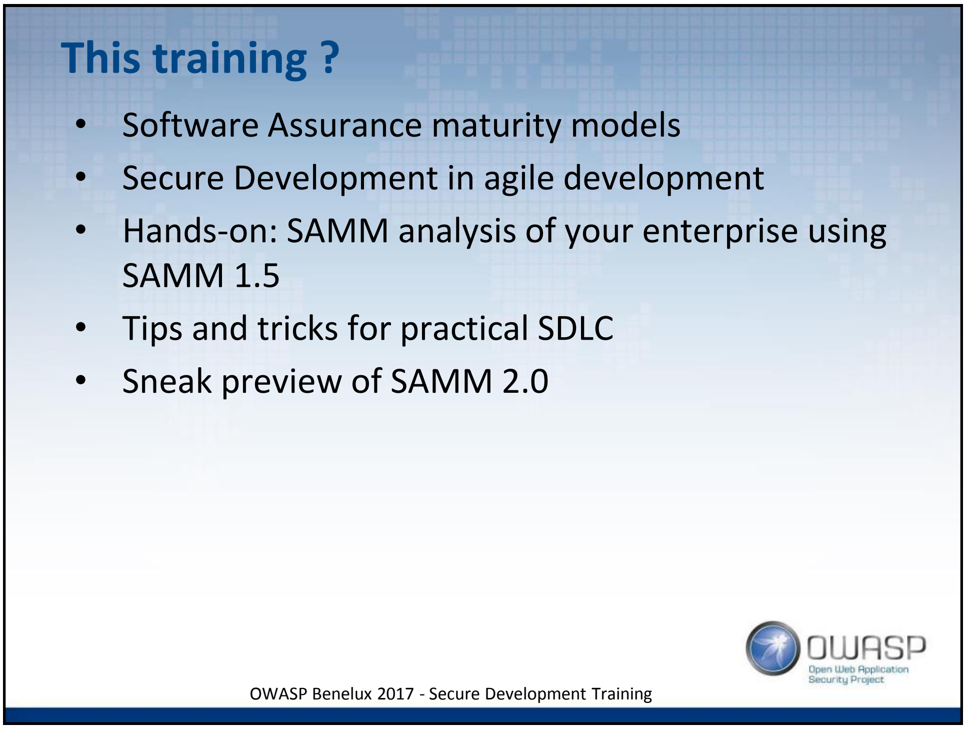


Bart?

Bart De Win, Ph.D.

- 20+ years experience in secure software development
- Belgian OWASP chapter co-leader
- SAMM contributor, evangelist and co-leader
- Author of >60 publications
- Director & security consultant @PwC BE
- Bart.de.win@pwc.com

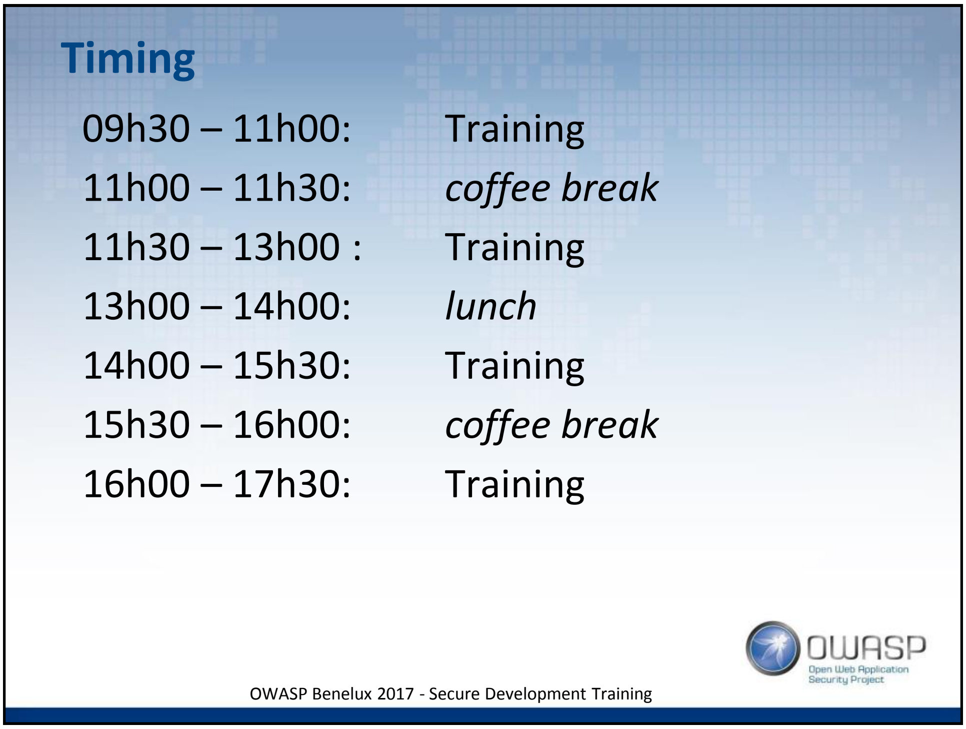
OWASP Benelux 2017 - Secure Development Training



This training ?

- Software Assurance maturity models
- Secure Development in agile development
- Hands-on: SAMM analysis of your enterprise using SAMM 1.5
- Tips and tricks for practical SDLC
- Sneak preview of SAMM 2.0

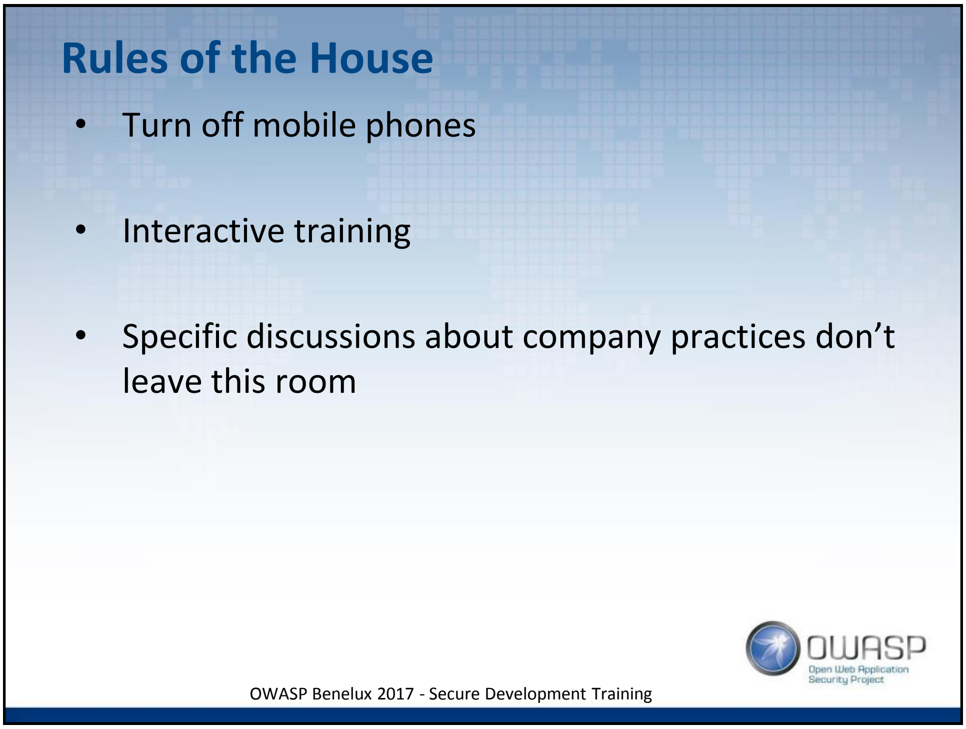
OWASP Benelux 2017 - Secure Development Training



Timing

09h30 – 11h00:	Training
11h00 – 11h30:	*coffee break*
11h30 – 13h00 :	Training
13h00 – 14h00:	*lunch*
14h00 – 15h30:	Training
15h30 – 16h00:	*coffee break*
16h00 – 17h30:	Training

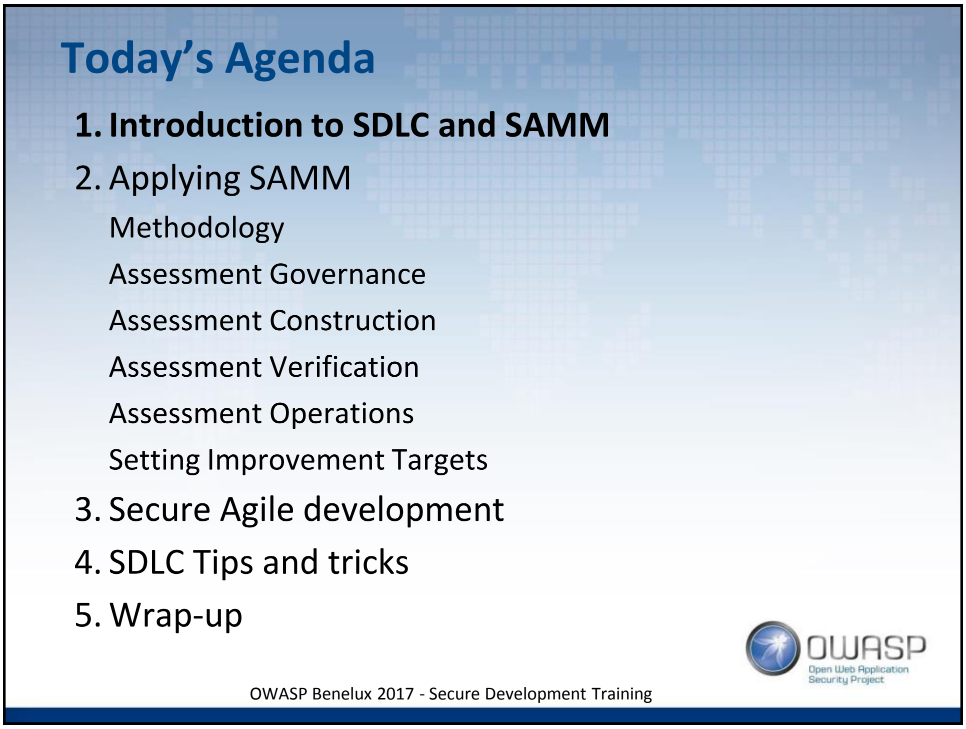
OWASP Benelux 2017 - Secure Development Training



Rules of the House

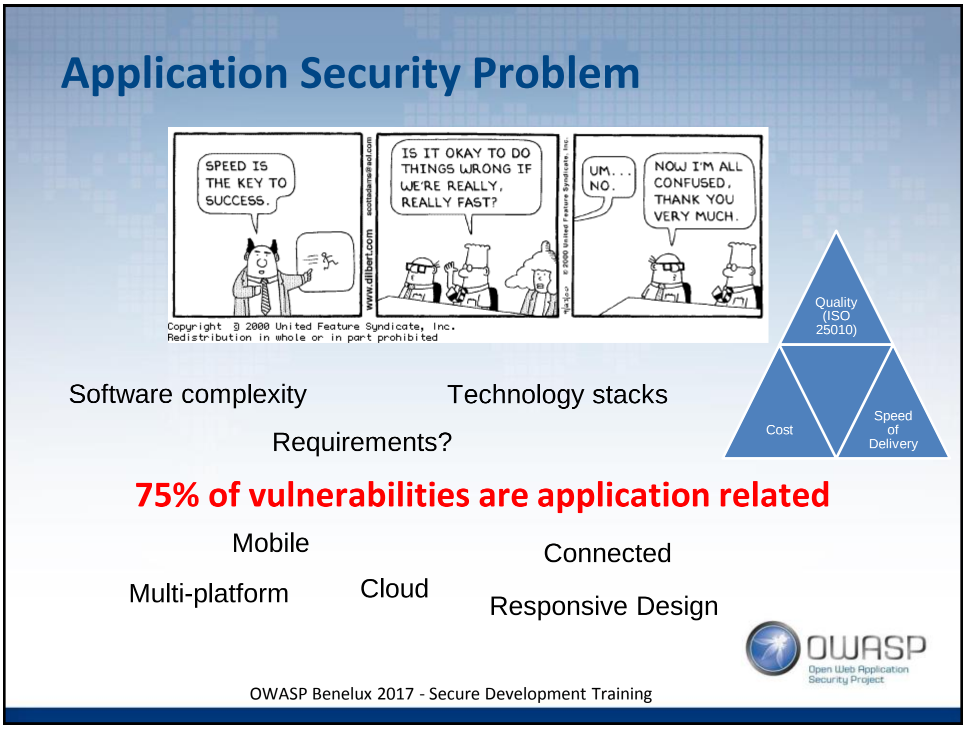
- Turn off mobile phones
- Interactive training
- Specific discussions about company practices don’t leave this room

OWASP Benelux 2017 - Secure Development Training



Today's Agenda

1. **Introduction to SDLC and SAMM**
2. Applying SAMM
   - Methodology
   - Assessment Governance
   - Assessment Construction
   - Assessment Verification
   - Assessment Operations
   - Setting Improvement Targets
3. Secure Agile development
4. SDLC Tips and tricks
5. Wrap-up



Application Security Problem

SPEED IS THE KEY TO SUCCESS.
IS IT OKAY TO DO THINGS WRONG IF WE'RE REALLY, REALLY FAST?
UM... NO.
NOW I'M ALL CONFUSED. THANK YOU VERY MUCH.
www.dilbert.com
scottadams@aol.com
© 2000 United Feature Syndicate, Inc.
Quality (ISO 25010)
Cost
Speed of Delivery

Copyright © 2000 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Software complexity

Technology stacks

Requirements?

75% of vulnerabilities are application related

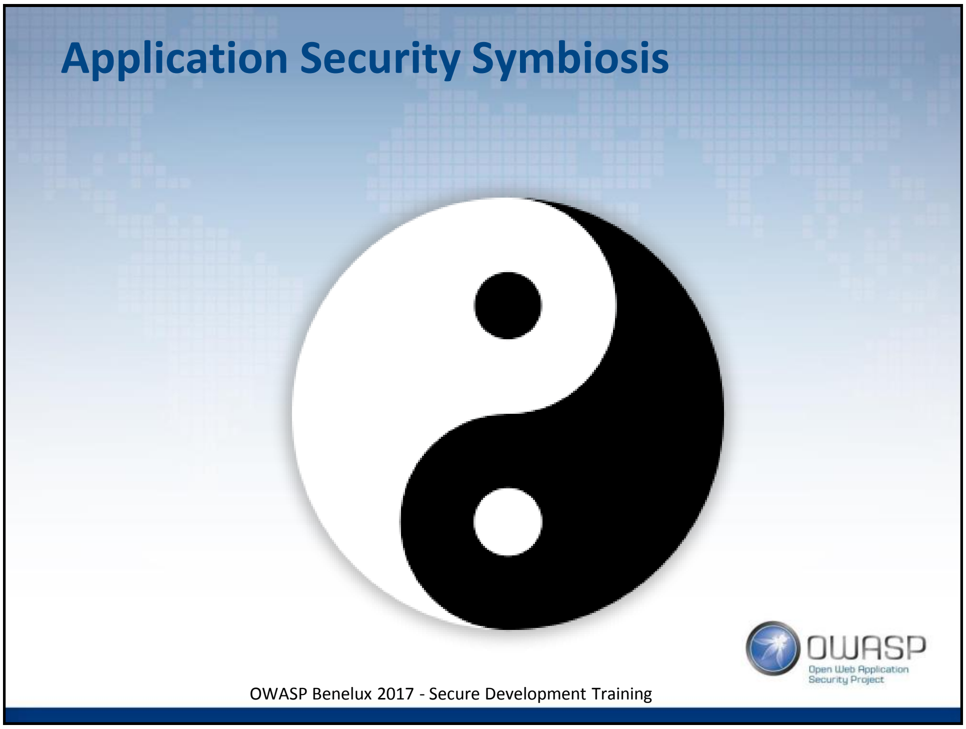
Mobile

Connected

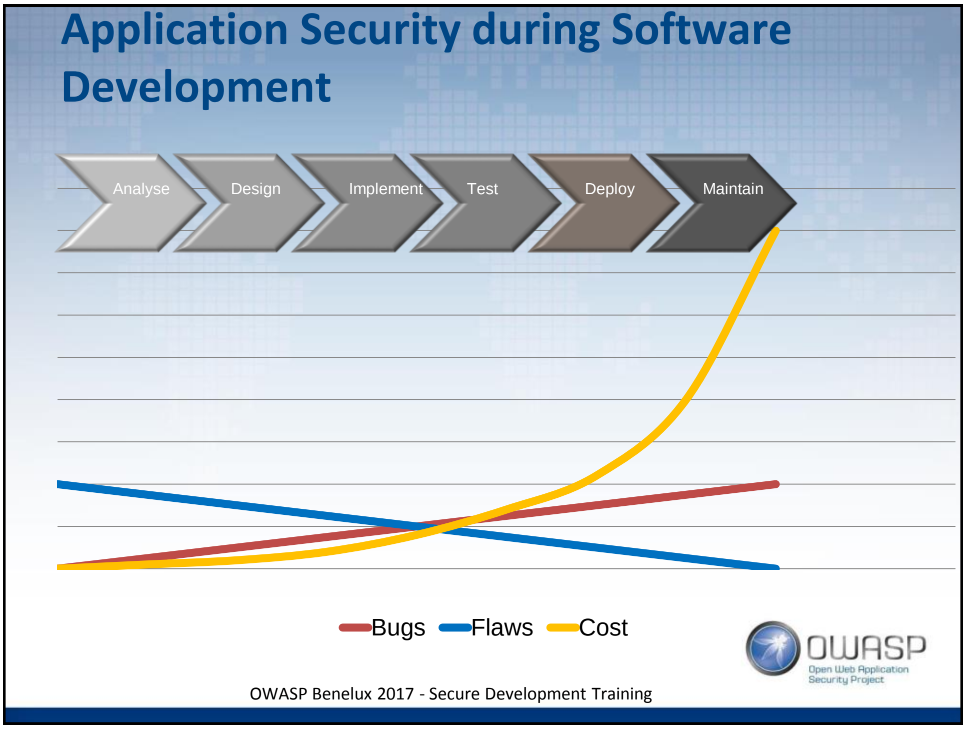
Multi-platform

Cloud

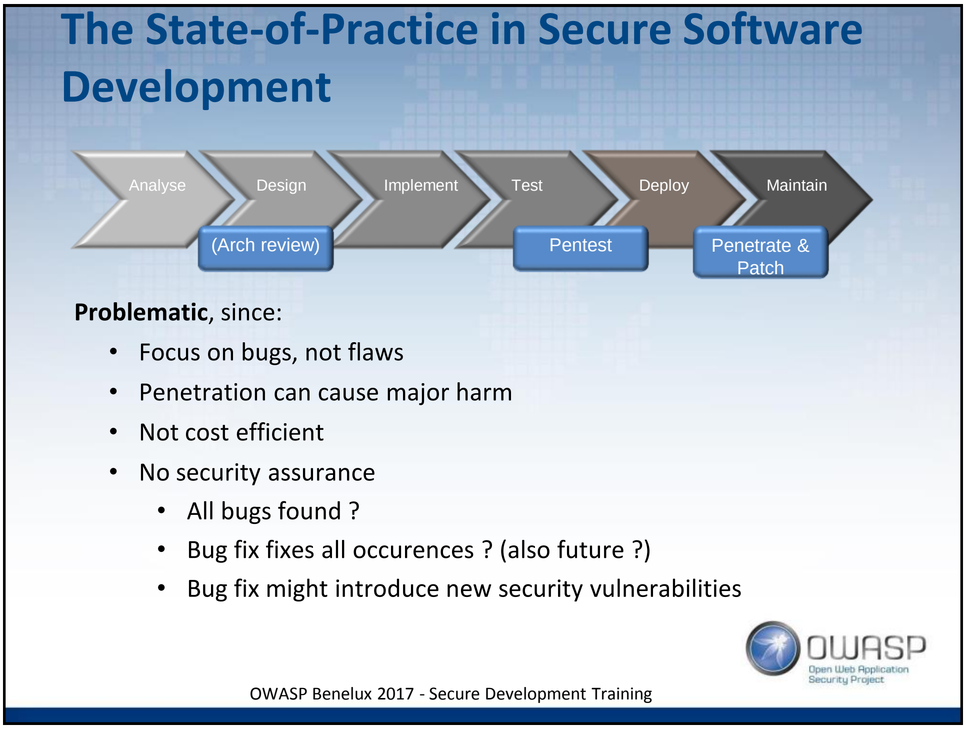
Responsive Design



Application Security Symbiosis



Application Security during Software Development



The State-of-Practice in Secure Software Development

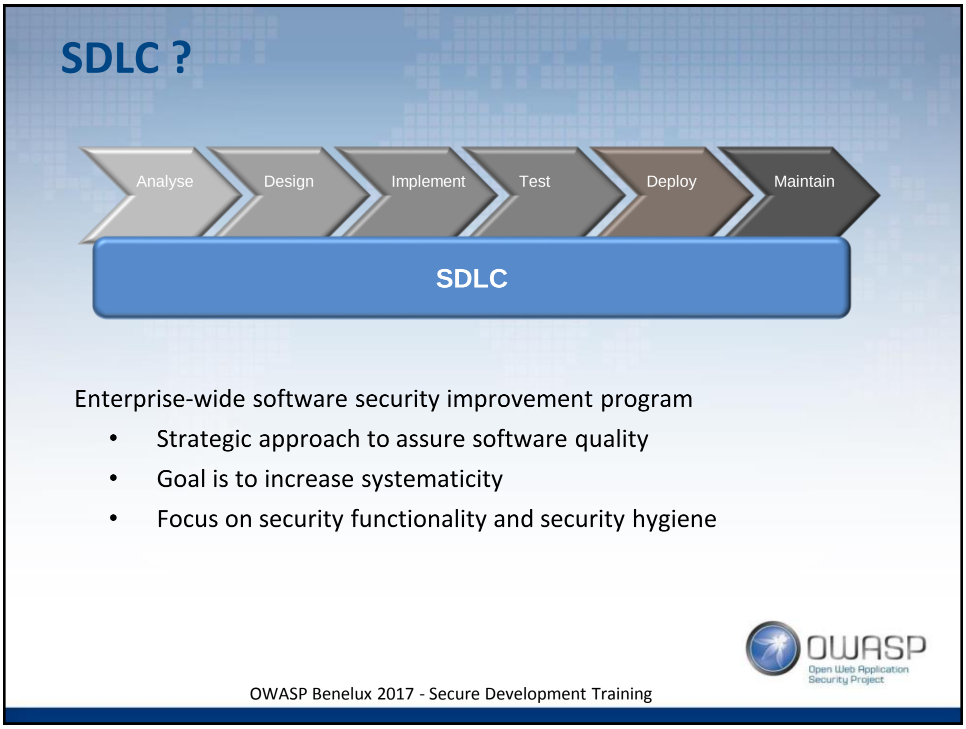
Analyse
Design
Implement
Test
Deploy
Maintain

(Arch review)
Pentest
Penetrate & Patch

Problematic, since:

- Focus on bugs, not flaws
- Penetration can cause major harm
- Not cost efficient
- No security assurance
  - All bugs found ?
  - Bug fix fixes all occurences ? (also future ?)
  - Bug fix might introduce new security vulnerabilities

OWASP Benelux 2017 - Secure Development Training



SDLC ?

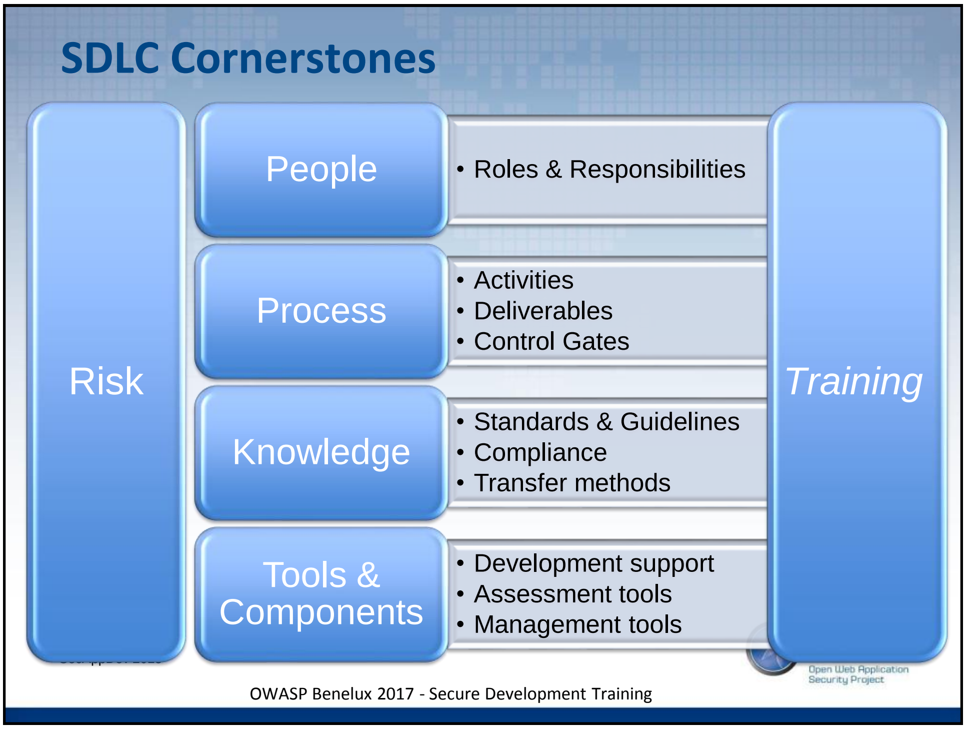
Analyse
Design
Implement
Test
Deploy
Maintain

SDLC

Enterprise-wide software security improvement program

- Strategic approach to assure software quality
- Goal is to increase systematicity
- Focus on security functionality and security hygiene

OWASP Benelux 2017 - Secure Development Training



SDLC Cornerstones

Risk

People
- Roles & Responsibilities

Process
- Activities
- Deliverables
- Control Gates

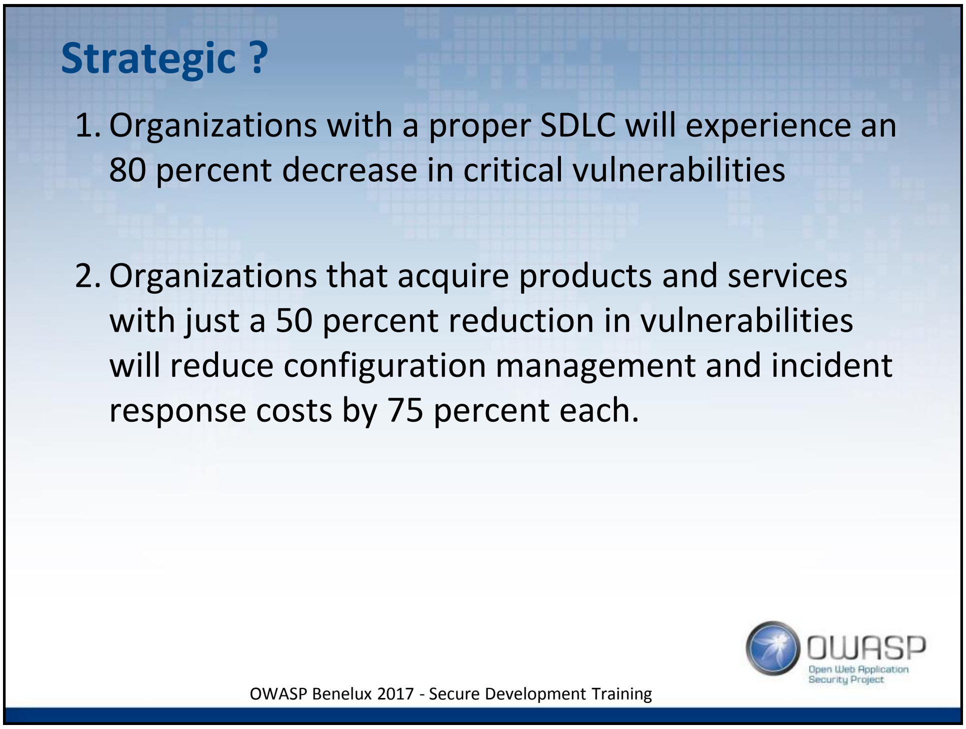
Knowledge
- Standards & Guidelines
- Compliance
- Transfer methods

Tools & Components
- Development support
- Assessment tools
- Management tools

Training

Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



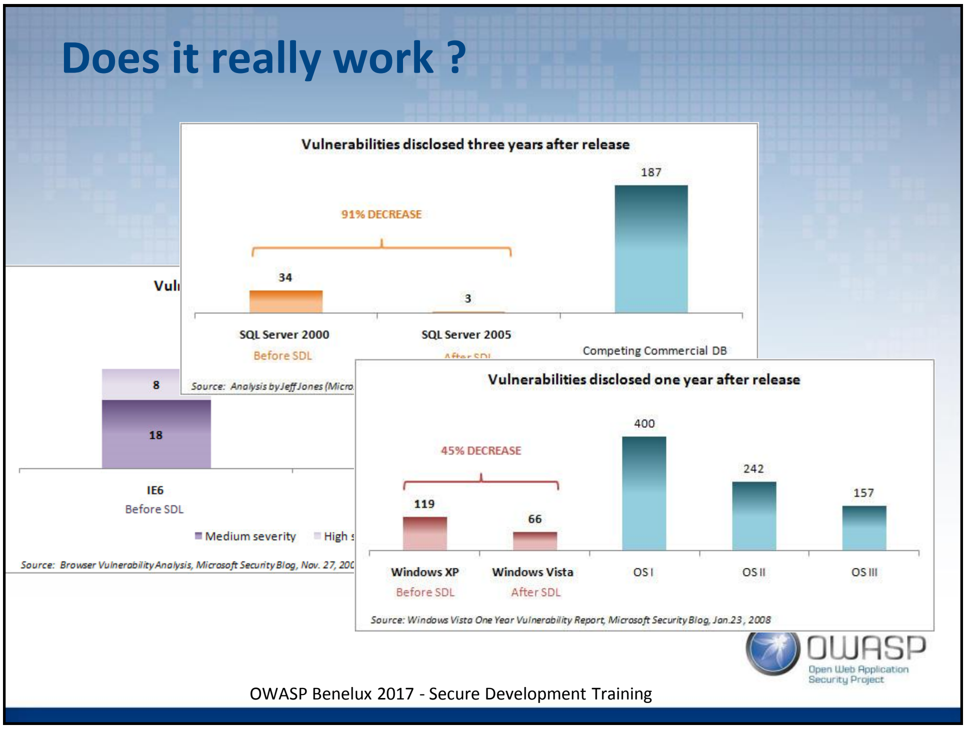
Strategic ?

1. Organizations with a proper SDLC will experience an 80 percent decrease in critical vulnerabilities

2. Organizations that acquire products and services with just a 50 percent reduction in vulnerabilities will reduce configuration management and incident response costs by 75 percent each.

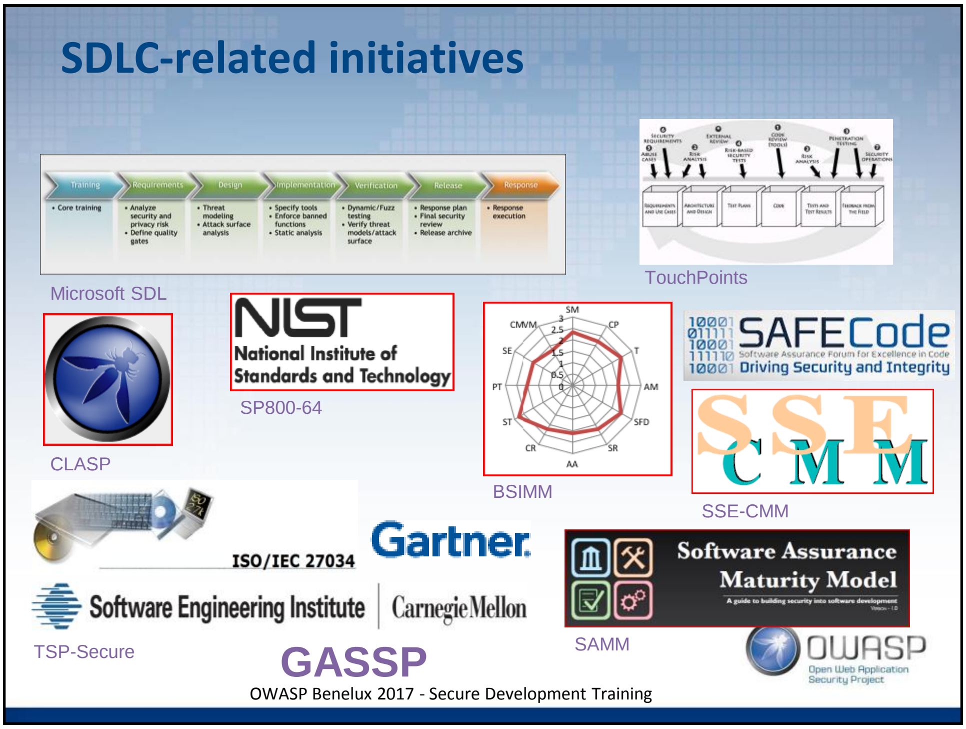
OWASP Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



Does it really work ?

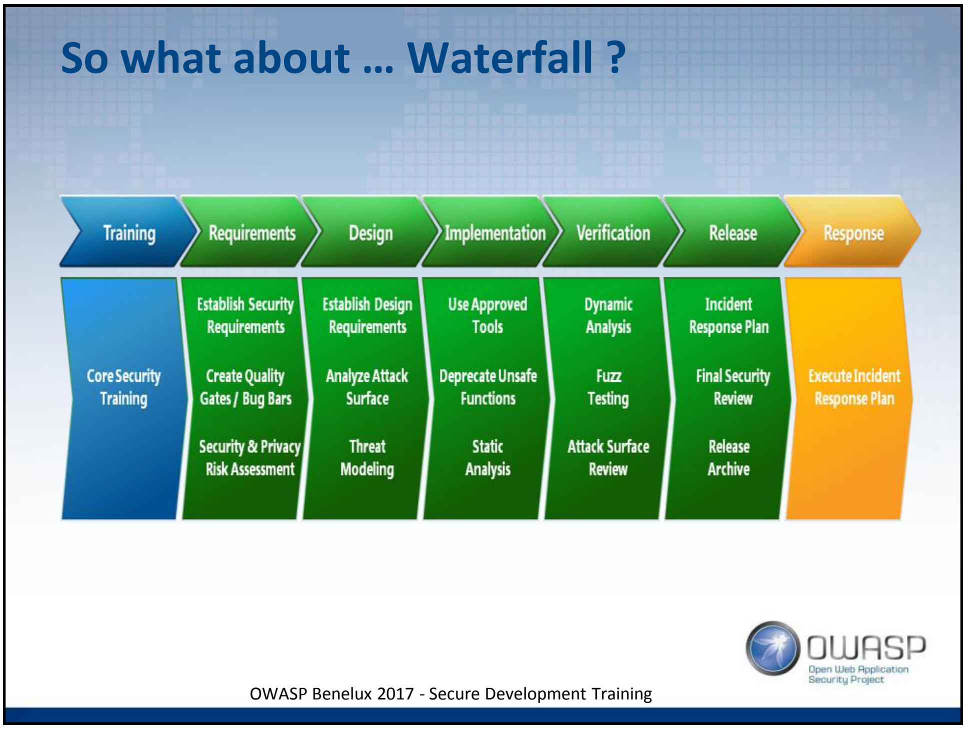
OWASP Benelux 2017 - Secure Development Training



SDLC-related initiatives

Training
Requirements
Design
Implementation
Verification
Release
Response
• Core training
• Analyze security and privacy risk
• Define quality gates
• Threat modeling
• Attack surface analysis
• Specify tools
• Enforce banned functions
• Static analysis
• Dynamic/Fuzz testing
• Verify threat models/attack surface
• Response plan
• Final security review
• Release archive
• Response execution
Microsoft SDL
TouchPoints
CLASP
NIST
National Institute of Standards and Technology
SP800-64
SM
CP
T
AM
SFD
SR
AA
CR
ST
PT
SE
CMVM
BSIMM
SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity
SSE CMM
SSE-CMM
ISO/IEC 27034
Gartner.
Software Assurance Maturity Model
A guide to building security into software development
SAMM
Software Engineering Institute | Carnegie Mellon
TSP-Secure
GASSP
OWASP
Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



So what about … Waterfall ?

Training

- Core Security Training

Requirements

- Establish Security Requirements
- Create Quality Gates / Bug Bars
- Security & Privacy Risk Assessment

Design

- Establish Design Requirements
- Analyze Attack Surface
- Threat Modeling

Implementation

- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

Verification

- Dynamic Analysis
- Fuzz Testing
- Attack Surface Review

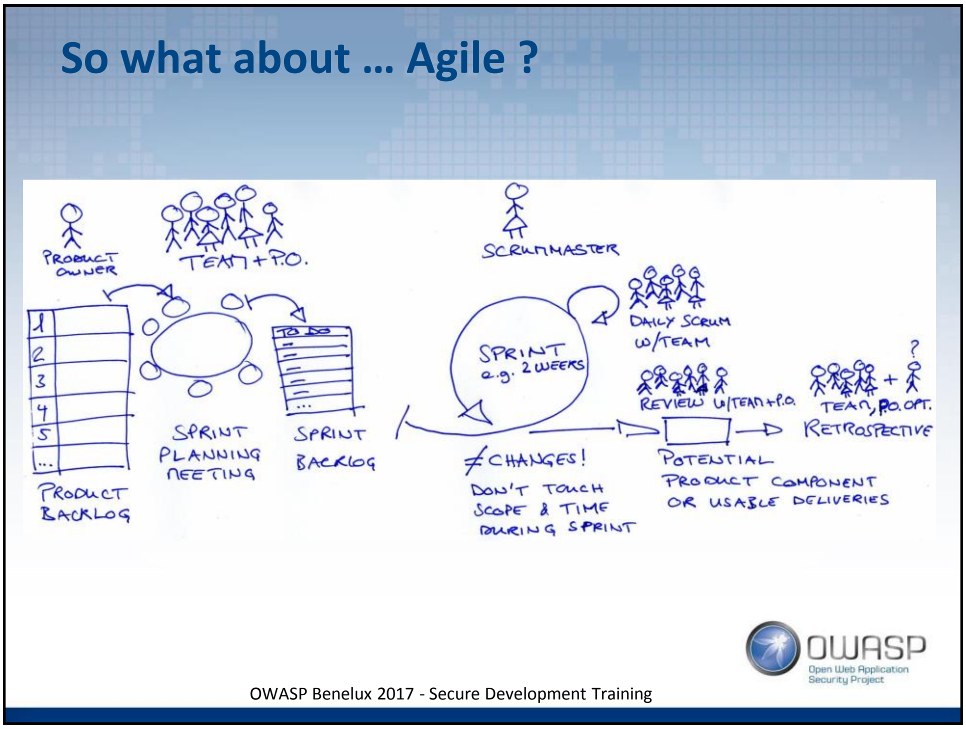
Release

- Incident Response Plan
- Final Security Review
- Release Archive

Response

- Execute Incident Response Plan

OWASP Benelux 2017 - Secure Development Training



So what about … Agile ?

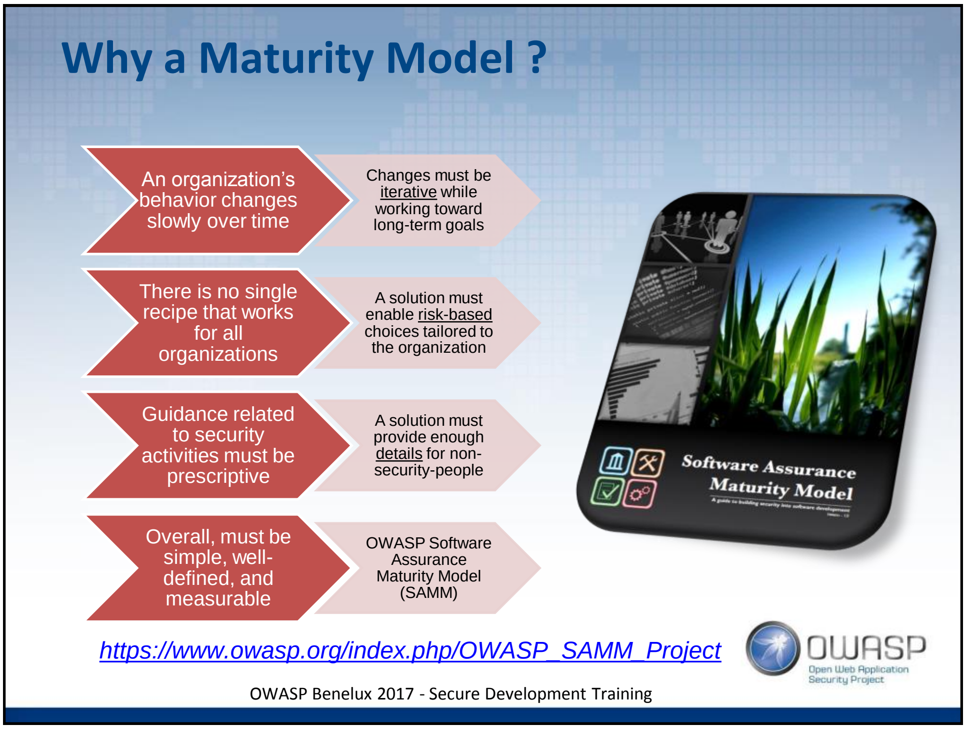
OWASP Benelux 2017 - Secure Development Training

Software Assurance

Is NOT …

But is …

OWASP Benelux 2017 - Secure Development Training



Why a Maturity Model ?

An organization's behavior changes slowly over time → Changes must be iterative while working toward long-term goals

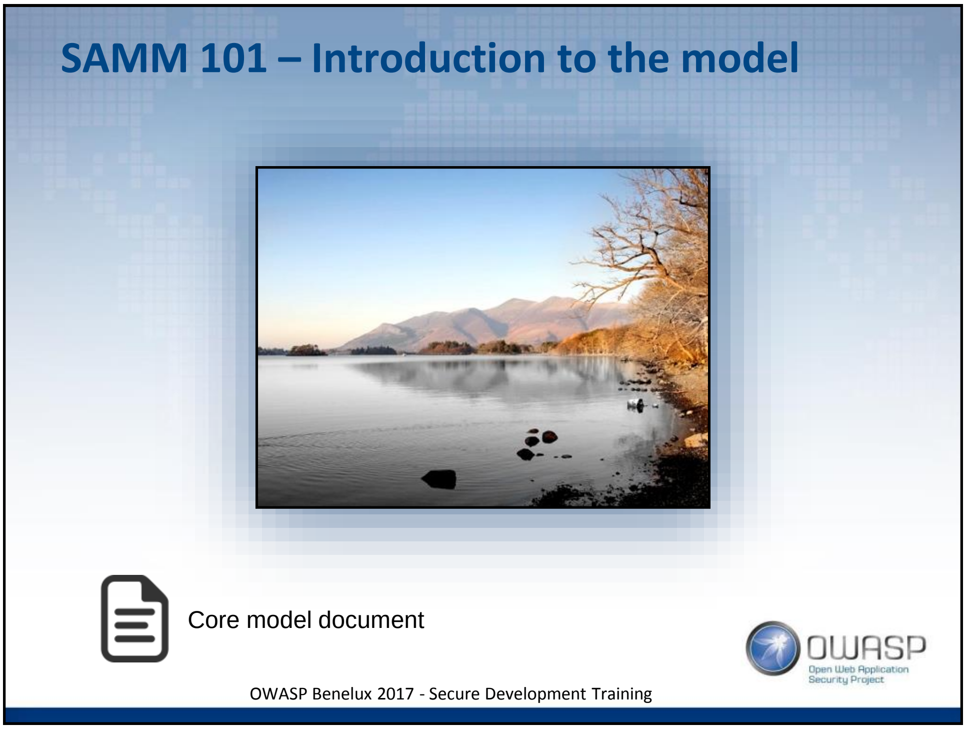
There is no single recipe that works for all organizations → A solution must enable risk-based choices tailored to the organization

Guidance related to security activities must be prescriptive → A solution must provide enough details for non-security-people

Overall, must be simple, well-defined, and measurable → OWASP Software Assurance Maturity Model (SAMM)

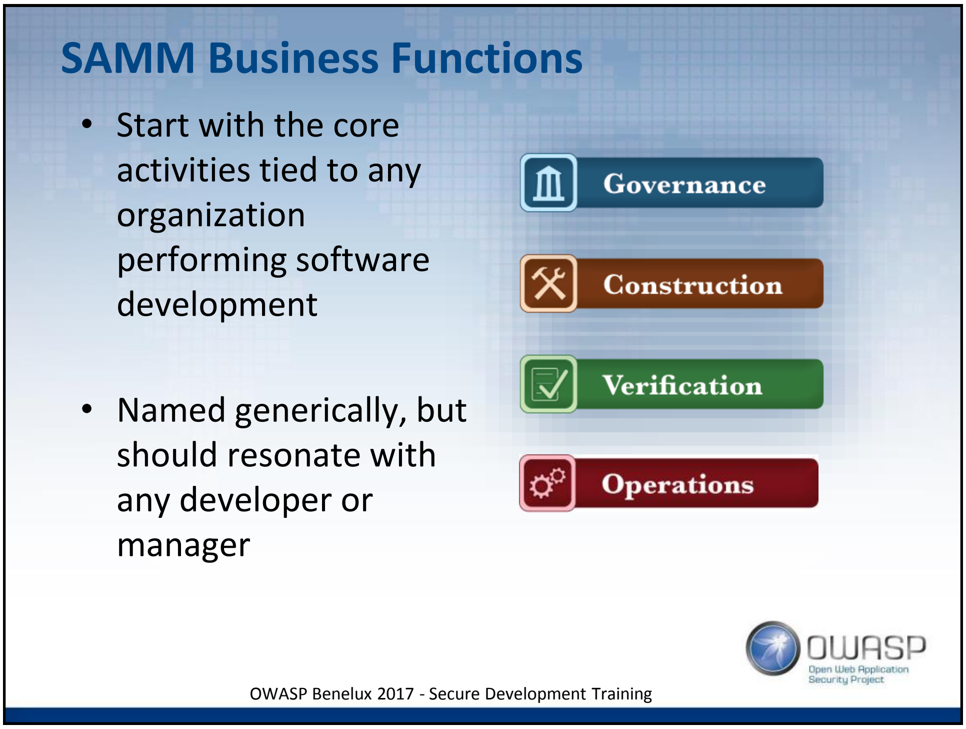
https://www.owasp.org/index.php/OWASP_SAMM_Project

OWASP Benelux 2017 - Secure Development Training



SAMM 101 – Introduction to the model

Core model document



SAMM Business Functions

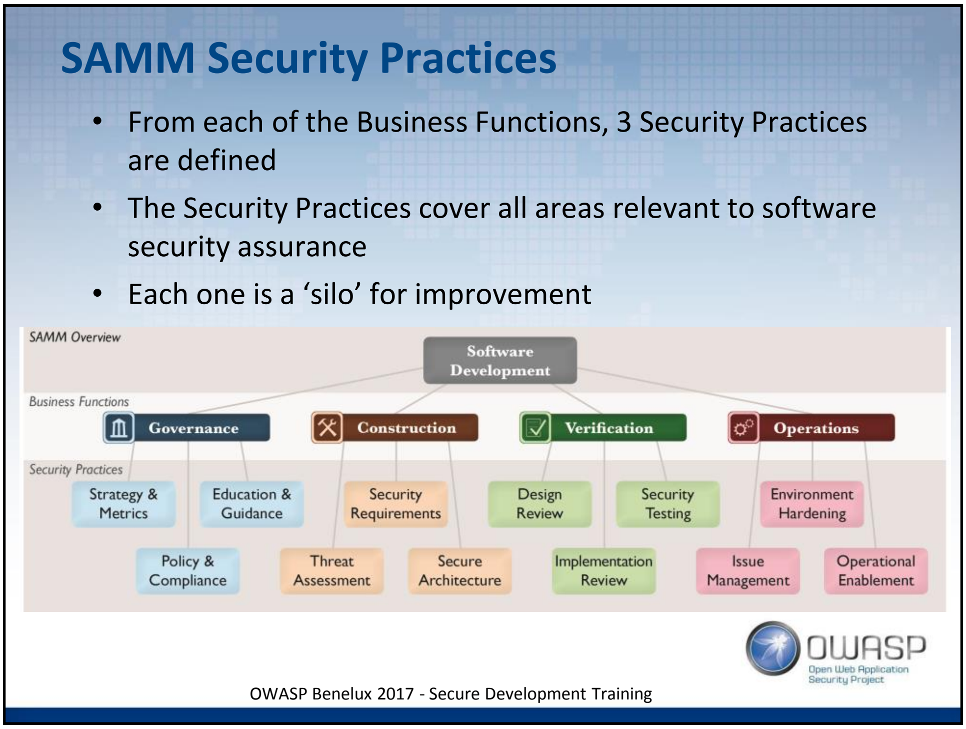
- Start with the core activities tied to any organization performing software development
- Named generically, but should resonate with any developer or manager

Governance

Construction

Verification

Operations



SAMM Security Practices

- From each of the Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance
- Each one is a ‘silo’ for improvement

SAMM Overview

Software Development

Business Functions

Governance

Construction

Verification

Operations

Security Practices

Strategy & Metrics

Education & Guidance

Policy & Compliance

Security Requirements

Threat Assessment

Secure Architecture

Design Review

Security Testing

Implementation Review

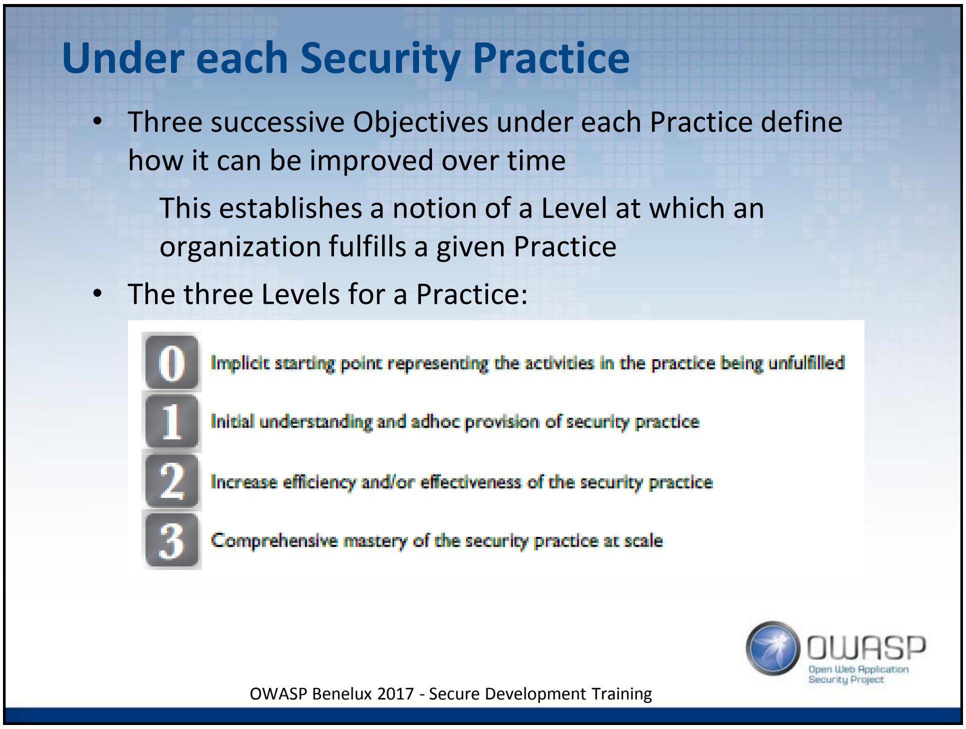
Environment Hardening

Issue Management

Operational Enablement

OWASP Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



Under each Security Practice

- Three successive Objectives under each Practice define how it can be improved over time

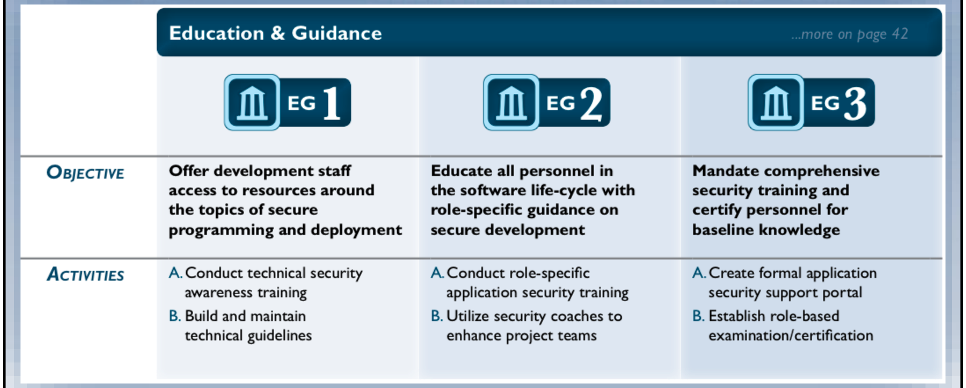
  This establishes a notion of a Level at which an organization fulfills a given Practice
- The three Levels for a Practice:

0	Implicit starting point representing the activities in the practice being unfulfilled
1	Initial understanding and adhoc provision of security practice
2	Increase efficiency and/or effectiveness of the security practice
3	Comprehensive mastery of the security practice at scale

OWASP Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training

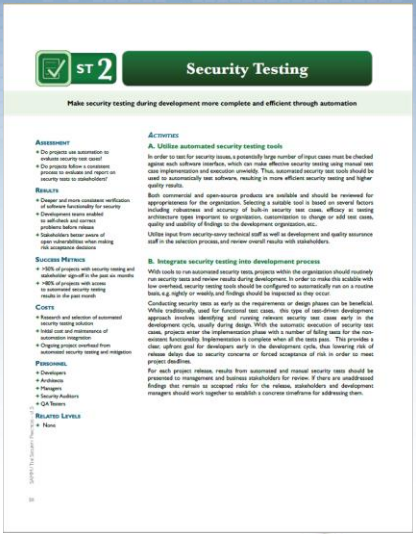
Check out this one...



	Education & Guidance EG 1	...more on page 42 EG 2	EG 3
Objective	**Offer development staff access to resources around the topics of secure programming and deployment**	**Educate all personnel in the software life-cycle with role-specific guidance on secure development**	**Mandate comprehensive security training and certify personnel for baseline knowledge**
Activities	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification

Per Level, SAMM defines...

- Objective
- Activities
- Results
- Success Metrics
- Costs
- Personnel
- Related Levels



ST 2 **Security Testing**

Make security testing during development more complete and efficient through automation

Assessment

- Do projects use automation to evaluate security test cases?
- Do projects follow a consistent process to evaluate and report on security tests to stakeholders?

Results

- Deeper and more consistent verification of software functionality for security
- Development teams enabled to self-check and correct problems before release
- Stakeholders better aware of open vulnerabilities when making risk acceptance decisions

Success Metrics

- >50% of projects with security testing and stakeholder sign-off in the past six months
- >80% of projects with access to automated security testing results in the past month

Costs

- Research and selection of automated security testing solution
- Initial cost and maintenance of automation integration
- Ongoing project overhead from automated security testing and mitigation

Personnel

- Developers
- Architects
- Managers
- Security Auditors
- QA Testers

Related Levels

- None

Activities

A. Utilize automated security testing tools

In order to test for security issues, a potentially large number of input cases must be checked against each software interface, which can make effective security testing using manual test case implementation and execution unwieldy. Thus, automated security test tools should be used to automatically test software, resulting in more efficient security testing and higher quality results.

Both commercial and open-source products are available and should be reviewed for appropriateness for the organization. Selecting a suitable tool is based on several factors including robustness and accuracy of built-in security test cases, efficacy at testing architecture types important to organization, customization to change or add test cases, quality and usability of findings to the development organization, etc.

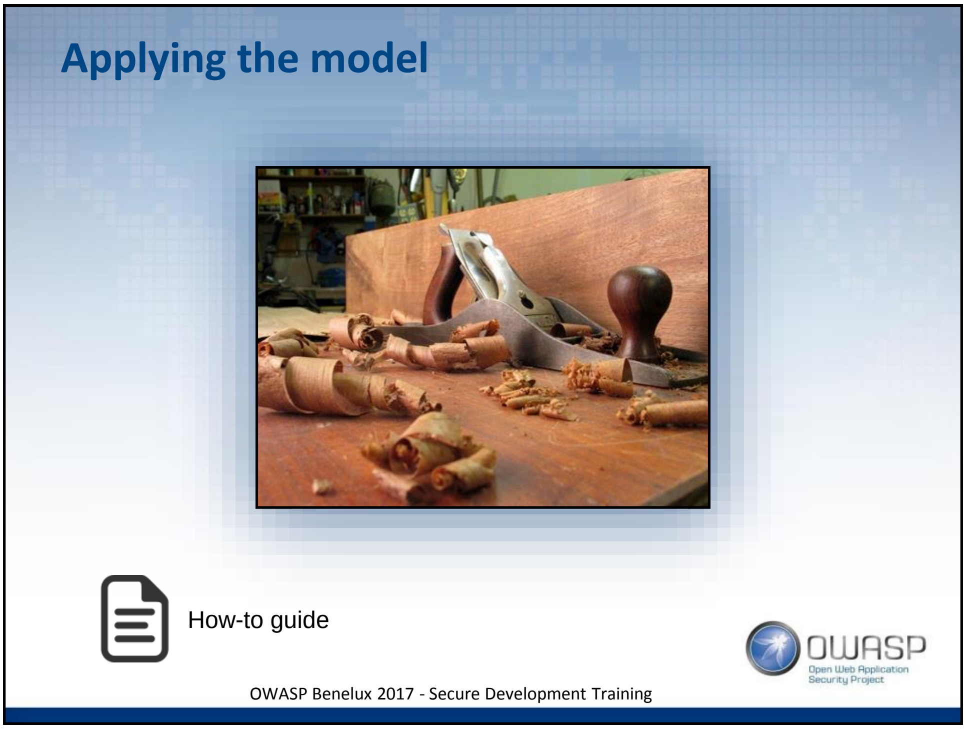
Utilize input from security-savvy technical staff as well as development and quality assurance staff in the selection process, and review overall results with stakeholders.

B. Integrate security testing into development process

With tools to run automated security tests, projects within the organization should routinely run security tests and review results during development. In order to make this scalable with low overhead, security testing tools should be configured to automatically run on a routine basis, e.g. nightly or weekly, and findings should be inspected as they occur.

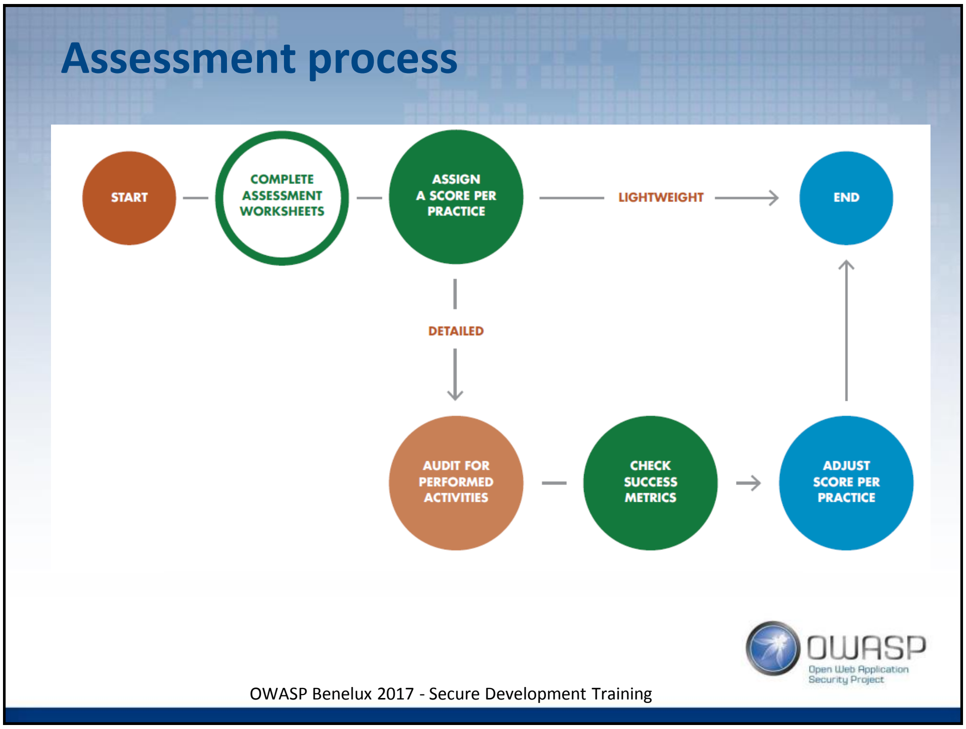
Conducting security tests as early as the requirements or design phases can be beneficial. While traditionally, used for functional test cases, this type of test-driven development approach involves identifying and running relevant security test cases early in the development cycle, usually during design. With the automatic execution of security test cases, projects enter the implementation phase with a number of failing tests for the non-existent functionality. Implementation is complete when all the tests pass. This provides a clear, upfront goal for developers early in the development cycle, thus lowering risk of release delays due to security concerns or forced acceptance of risk in order to meet project deadlines.

For each project release, results from automated and manual security tests should be presented to management and business stakeholders for review. If there are unaddressed findings that remain as accepted risks for the release, stakeholders and development managers should work together to establish a concrete timeframe for addressing them.



Applying the model

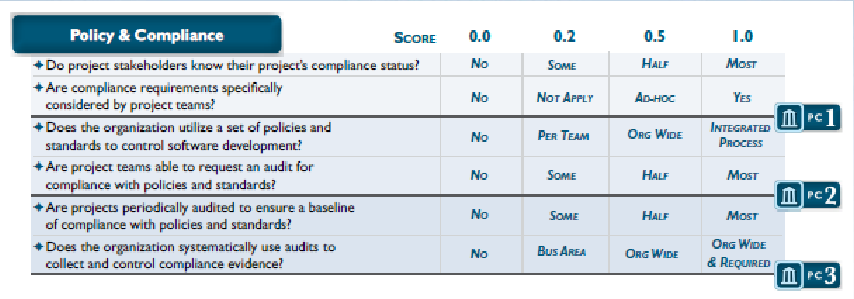
How-to guide



Assessment process

START
COMPLETE ASSESSMENT WORKSHEETS
ASSIGN A SCORE PER PRACTICE
LIGHTWEIGHT
END
DETAILED
AUDIT FOR PERFORMED ACTIVITIES
CHECK SUCCESS METRICS
ADJUST SCORE PER PRACTICE

Assessment worksheets



Policy & Compliance

	Score 0.0	0.2	0.5	1.0	
✦ Do project stakeholders know their project's compliance status?	No	Some	Half	Most	
✦ Are compliance requirements specifically considered by project teams?	No	Not Apply	Ad-hoc	Yes	PC 1
✦ Does the organization utilize a set of policies and standards to control software development?	No	Per Team	Org Wide	Integrated Process	
✦ Are project teams able to request an audit for compliance with policies and standards?	No	Some	Half	Most	PC 2
✦ Are projects periodically audited to ensure a baseline of compliance with policies and standards?	No	Some	Half	Most	
✦ Does the organization systematically use audits to collect and control compliance evidence?	No	Bus Area	Org Wide	Org Wide & Required	PC 3

OWASP Benelux 2017 - Secure Development Training

Intermezzo – how to measure

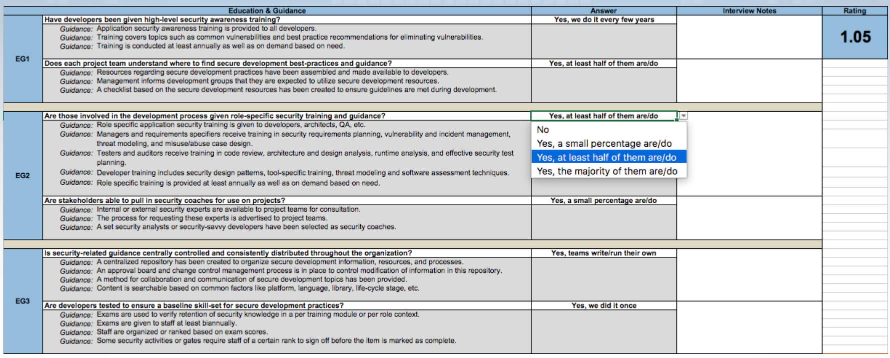
What?

How well?

How wide?

OWASP Benelux 2017 - Secure Development Training

Assessment Toolbox



	Education & Guidance	Answer	Interview Notes	Rating
EG1	**Have developers been given high-level security awareness training?**	Yes, we do it every few years		1.05
	Guidance: Application security awareness training is provided to all developers. *Guidance:* Training covers topics such as common vulnerabilities and best practice recommendations for eliminating vulnerabilities. *Guidance:* Training is conducted at least annually as well as on demand based on need.			
	Does each project team understand where to find secure development best-practices and guidance?	Yes, at least half of them are/do		
	Guidance: Resources regarding secure development practices have been assembled and made available to developers. *Guidance:* Management informs development groups that they are expected to utilize secure development resources. *Guidance:* A checklist based on the secure development resources has been created to ensure guidelines are met during development.			
EG2	**Are those involved in the development process given role-specific security training and guidance?**	Yes, at least half of them are/do		
	Guidance: Role specific application security training is given to developers, architects, QA, etc. *Guidance:* Managers and requirements specifiers receive training in security requirements planning, vulnerability and incident management, threat modeling, and misuse/abuse case design. *Guidance:* Testers and auditors receive training in code review, architecture and design analysis, runtime analysis, and effective security test planning. *Guidance:* Developer training includes security design patterns, tool-specific training, threat modeling and software assessment techniques. *Guidance:* Role specific training is provided at least annually as well as on demand based on need.	No Yes, a small percentage are/do Yes, at least half of them are/do Yes, the majority of them are/do		
	Are stakeholders able to pull in security coaches for use on projects?	Yes, a small percentage are/do		
	Guidance: Internal or external security experts are available to project teams for consultation. *Guidance:* The process for requesting these experts is advertised to project teams. *Guidance:* A set security analysts or security-savvy developers have been selected as security coaches.			
EG3	**Is security-related guidance centrally controlled and consistently distributed throughout the organization?**	Yes, teams write/run their own		
	Guidance: A centralized repository has been created to organize secure development information, resources, and processes. *Guidance:* An approval board and change control management process is in place to control modification of information in this repository. *Guidance:* A method for collaboration and communication of secure development topics has been provided. *Guidance:* Content is searchable based on common factors like platform, language, library, life-cycle stage, etc.			
	Are developers tested to ensure a baseline skill-set for secure development practices?	Yes, we did it once		
	Guidance: Exams are used to verify retention of security knowledge in a per training module or per role context. *Guidance:* Exams are given to staff at least biannually. *Guidance:* Staff are organized or ranked based on exam scores. *Guidance:* Some security activities or gates require staff of a certain rank to sign off before the item is marked as complete.			

OWASP Benelux 2017 - Secure Development Training

Creating Scorecards

- Gap analysis

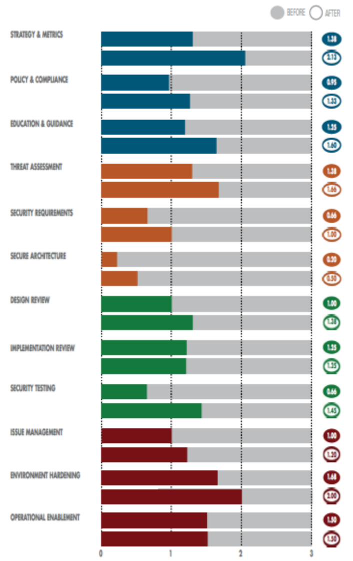
  Capturing scores from detailed assessments versus expected performance levels

- Demonstrating improvement

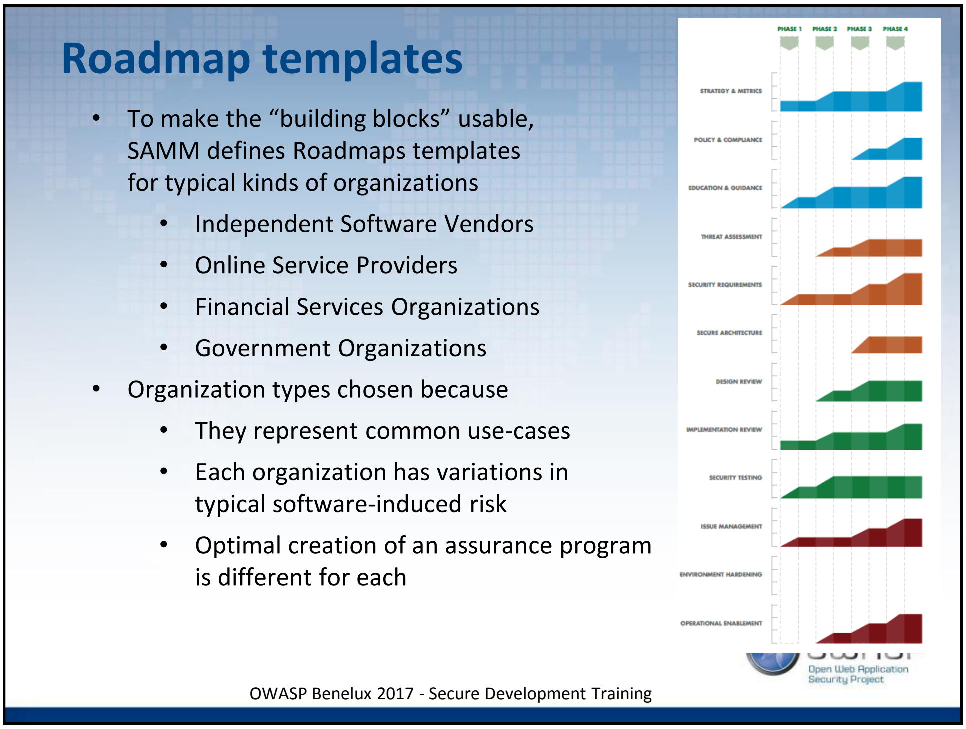
  Capturing scores from before and after an iteration of assurance program build-out

- Ongoing measurement

  Capturing scores over consistent time frames for an assurance program that is already in place



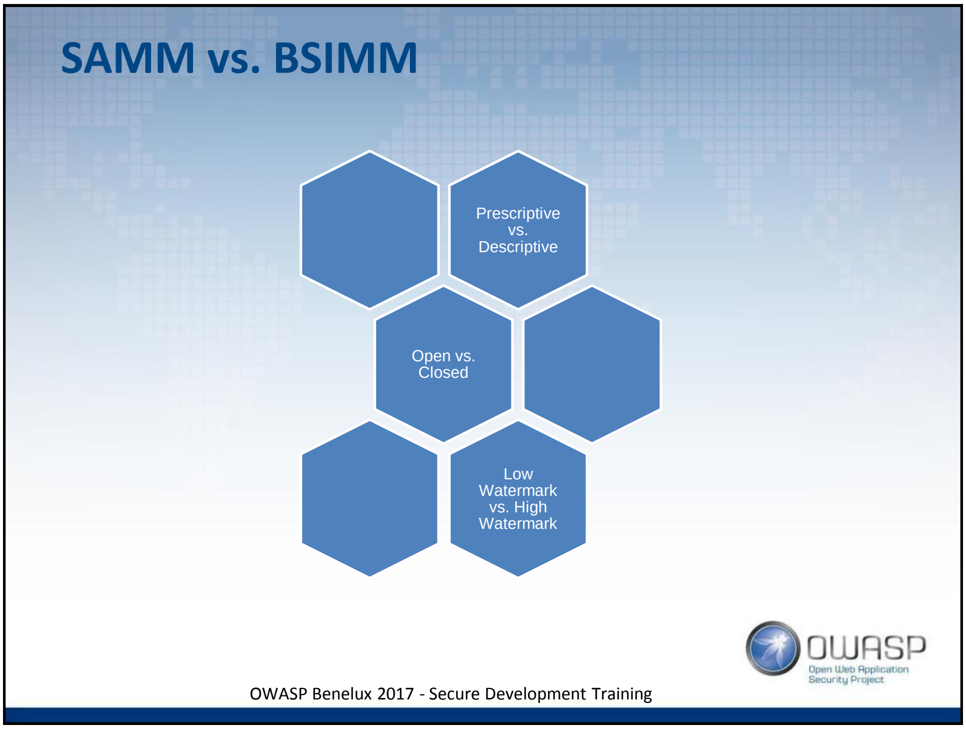
OWASP Benelux 2017 - Secure Development Training



Roadmap templates

- To make the “building blocks” usable, SAMM defines Roadmaps templates for typical kinds of organizations
  - Independent Software Vendors
  - Online Service Providers
  - Financial Services Organizations
  - Government Organizations
- Organization types chosen because
  - They represent common use-cases
  - Each organization has variations in typical software-induced risk
  - Optimal creation of an assurance program is different for each

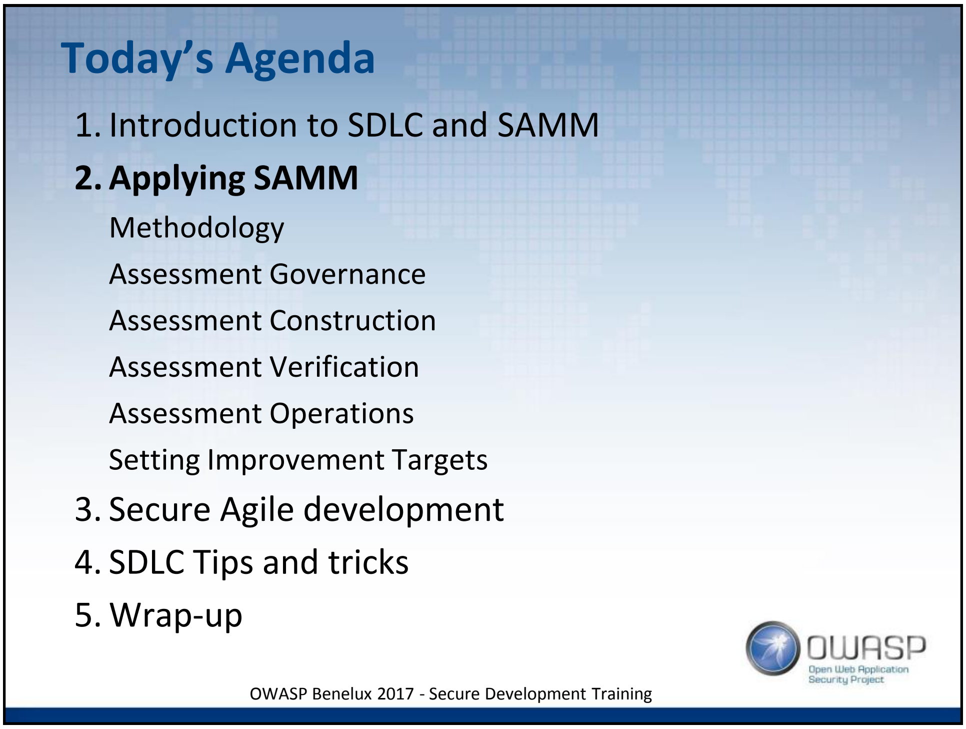
OWASP Benelux 2017 - Secure Development Training



SAMM vs. BSIMM

Prescriptive vs. Descriptive
Open vs. Closed
Low Watermark vs. High Watermark

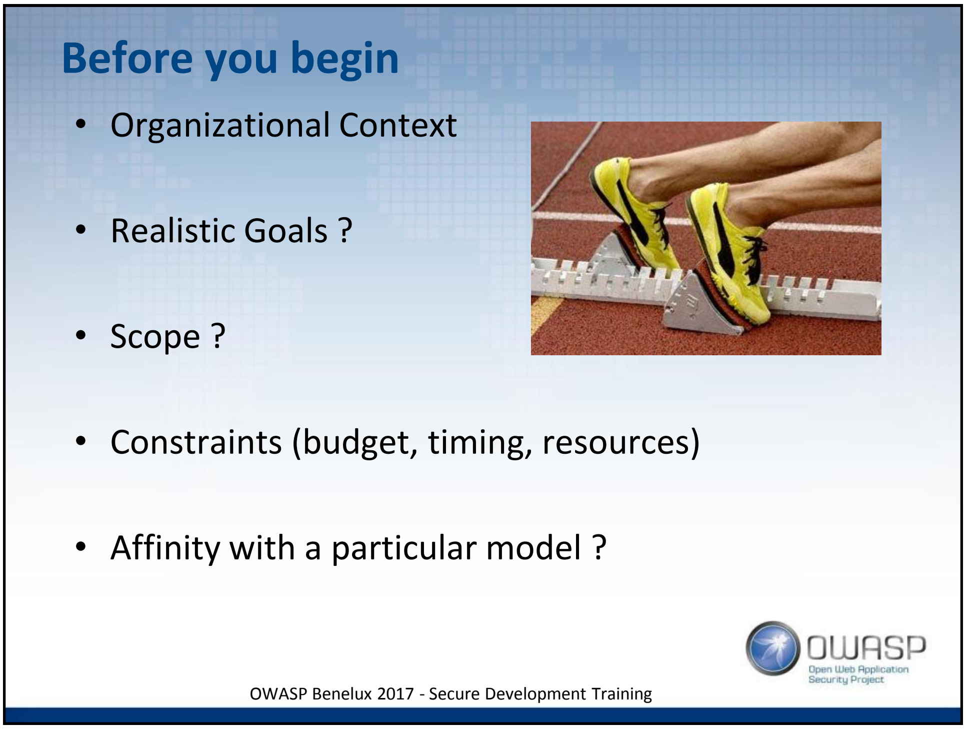
OWASP Benelux 2017 - Secure Development Training



Today's Agenda

1. Introduction to SDLC and SAMM
2. **Applying SAMM**
   - Methodology
   - Assessment Governance
   - Assessment Construction
   - Assessment Verification
   - Assessment Operations
   - Setting Improvement Targets
3. Secure Agile development
4. SDLC Tips and tricks
5. Wrap-up

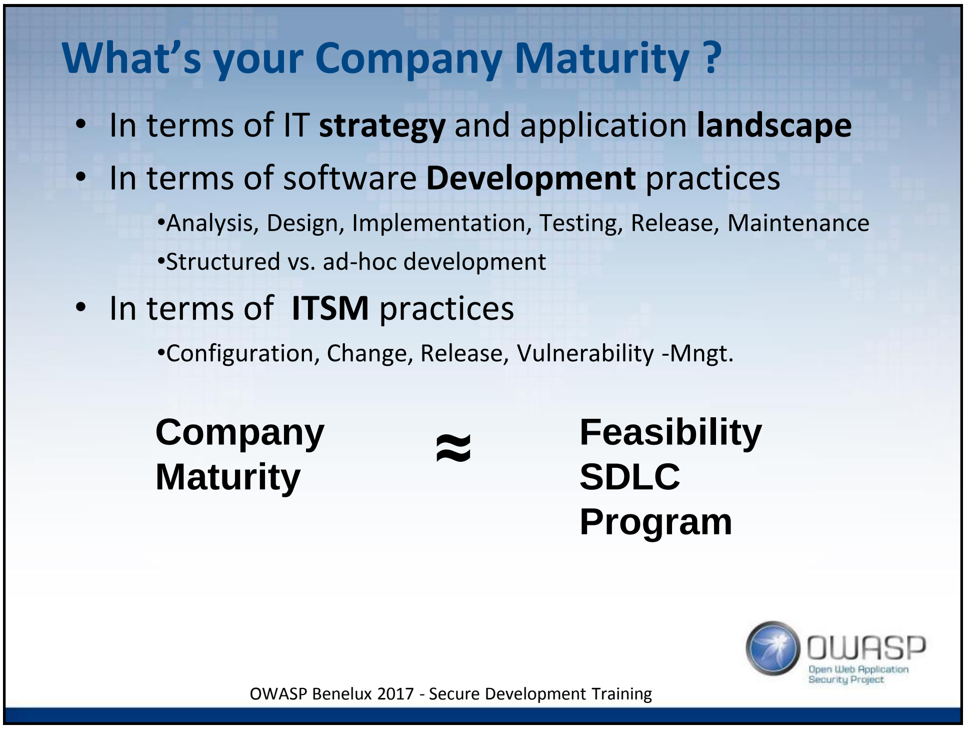
OWASP Benelux 2017 - Secure Development Training



Before you begin

- Organizational Context
- Realistic Goals ?
- Scope ?
- Constraints (budget, timing, resources)
- Affinity with a particular model ?

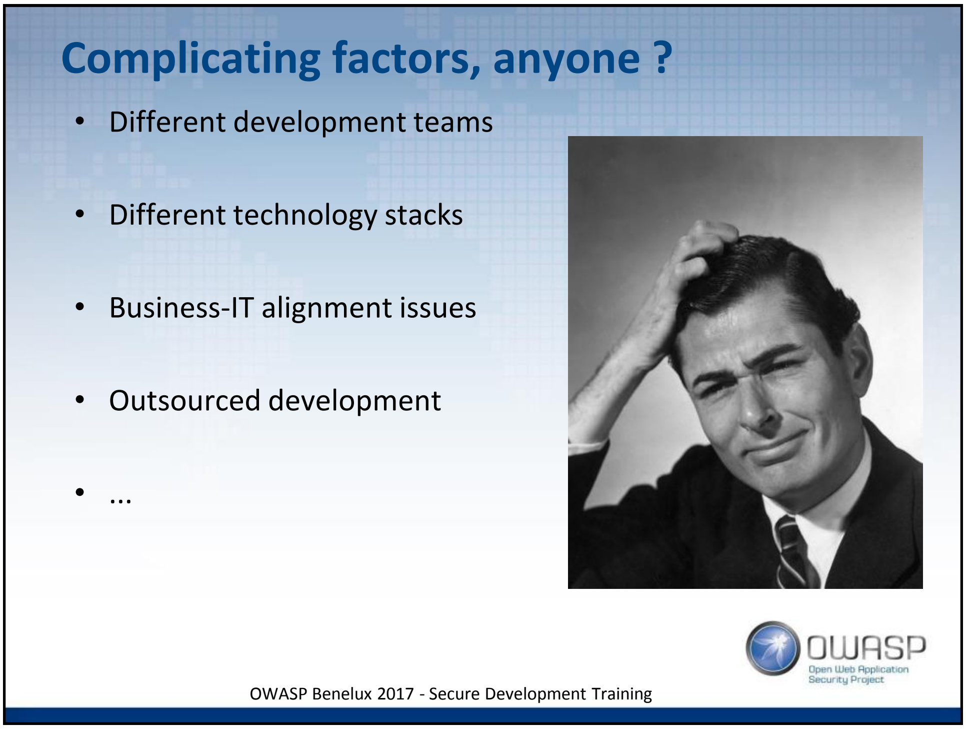
OWASP Benelux 2017 - Secure Development Training



What’s your Company Maturity ?

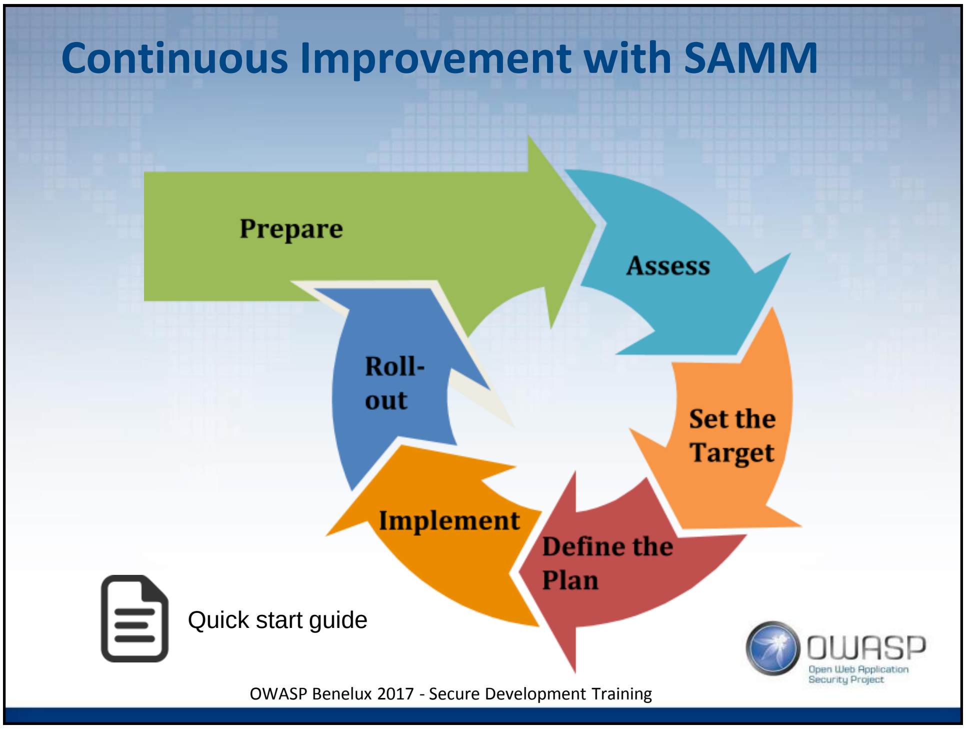
- In terms of IT **strategy** and application **landscape**
- In terms of software **Development** practices
  - •Analysis, Design, Implementation, Testing, Release, Maintenance
  - •Structured vs. ad-hoc development
- In terms of **ITSM** practices
  - •Configuration, Change, Release, Vulnerability -Mngt.

Company Maturity ≈ **Feasibility SDLC Program**



Complicating factors, anyone ?

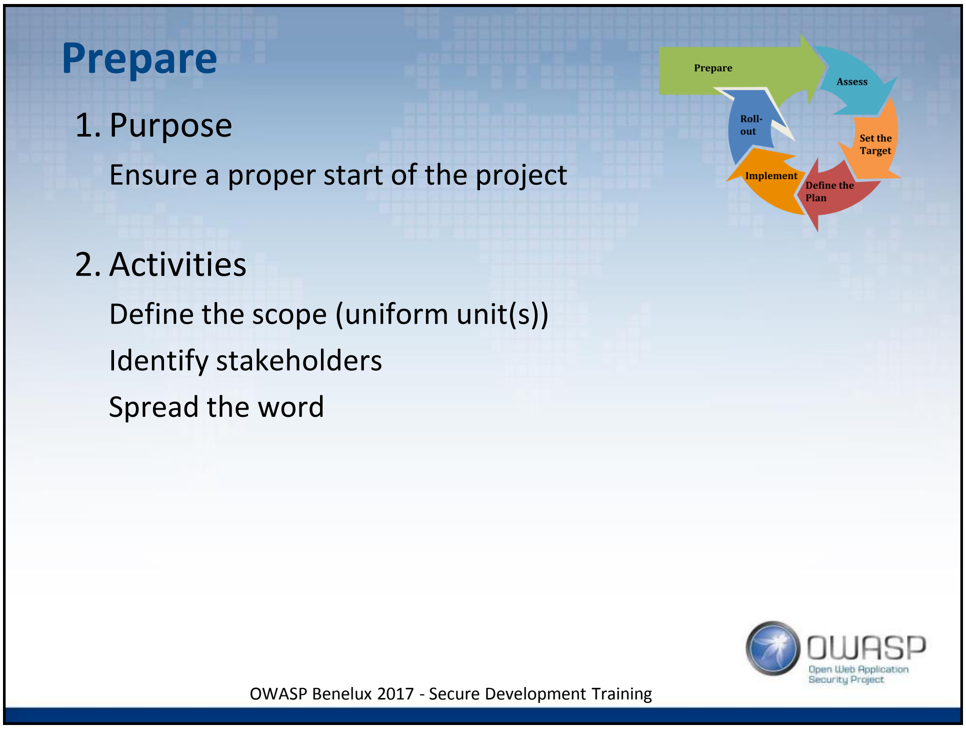
- Different development teams
- Different technology stacks
- Business-IT alignment issues
- Outsourced development
- …



Continuous Improvement with SAMM

Quick start guide

OWASP Benelux 2017 - Secure Development Training

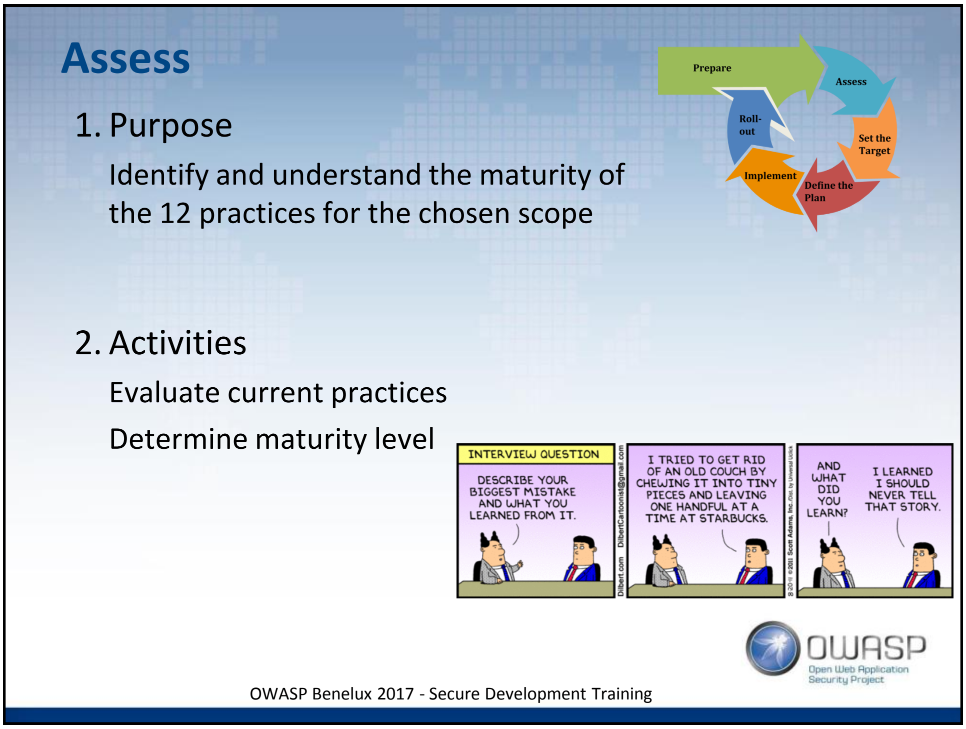


Prepare

1. Purpose
   - Ensure a proper start of the project
2. Activities
   - Define the scope (uniform unit(s))
   - Identify stakeholders
   - Spread the word

Prepare
Assess
Roll-out
Set the Target
Implement
Define the Plan

OWASP Benelux 2017 - Secure Development Training



Assess

1. Purpose

   Identify and understand the maturity of the 12 practices for the chosen scope

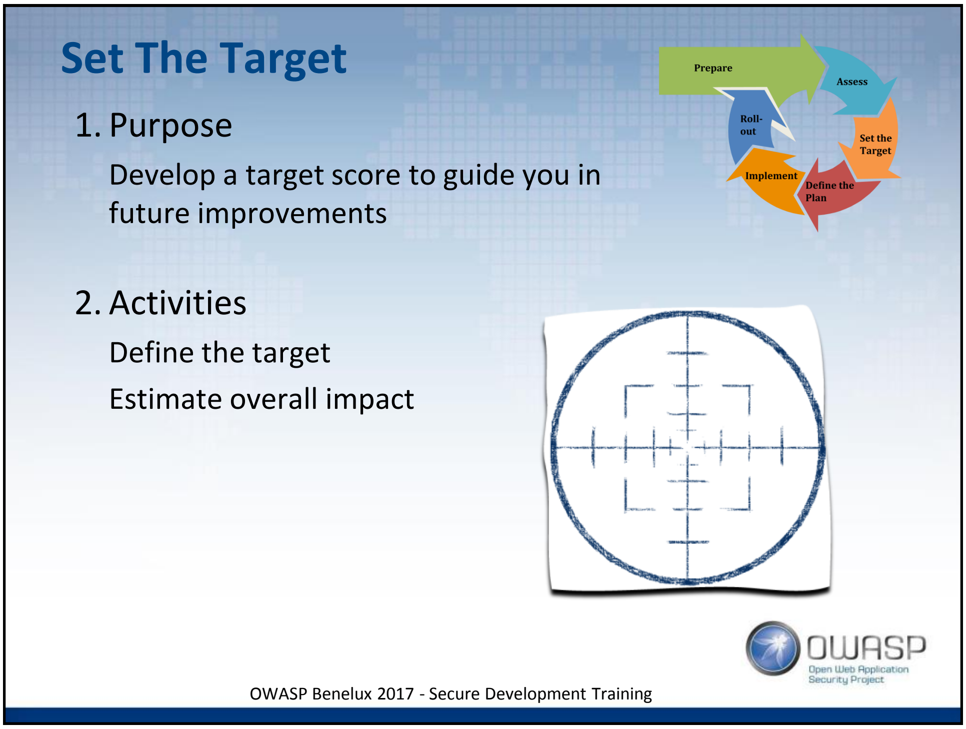
2. Activities

   Evaluate current practices

   Determine maturity level

Prepare
Assess
Set the Target
Define the Plan
Implement
Roll-out
INTERVIEW QUESTION
DESCRIBE YOUR BIGGEST MISTAKE AND WHAT YOU LEARNED FROM IT.
I TRIED TO GET RID OF AN OLD COUCH BY CHEWING IT INTO TINY PIECES AND LEAVING ONE HANDFUL AT A TIME AT STARBUCKS.
AND WHAT DID YOU LEARN?
I LEARNED I SHOULD NEVER TELL THAT STORY.
Dilbert.com DilbertCartoonist@gmail.com
8-20-11 ©2011 Scott Adams, Inc./Dist. by Universal Uclick
OWASP
Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



Set The Target

1. Purpose

   Develop a target score to guide you in future improvements

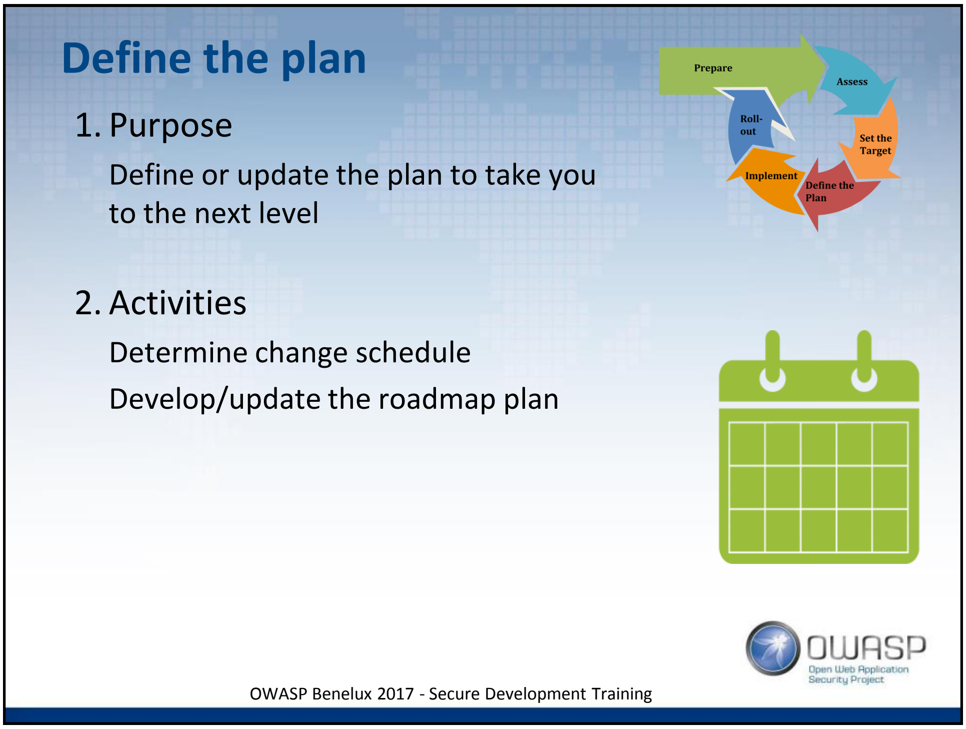
2. Activities

   Define the target

   Estimate overall impact

Prepare
Assess
Set the Target
Define the Plan
Implement
Roll-out
OWASP
Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training


Prepare
Assess
Set the Target
Define the Plan
Implement
Roll-out

Define the plan

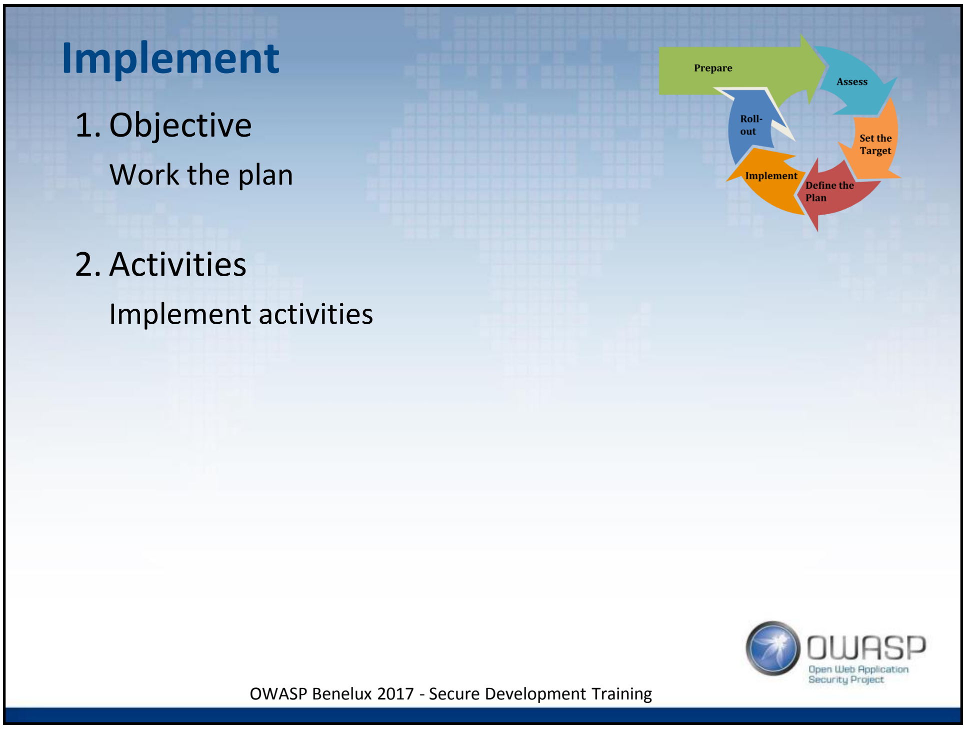
1. Purpose

   Define or update the plan to take you to the next level

2. Activities

   Determine change schedule

   Develop/update the roadmap plan


Prepare
Assess
Set the Target
Define the Plan
Implement
Roll-out

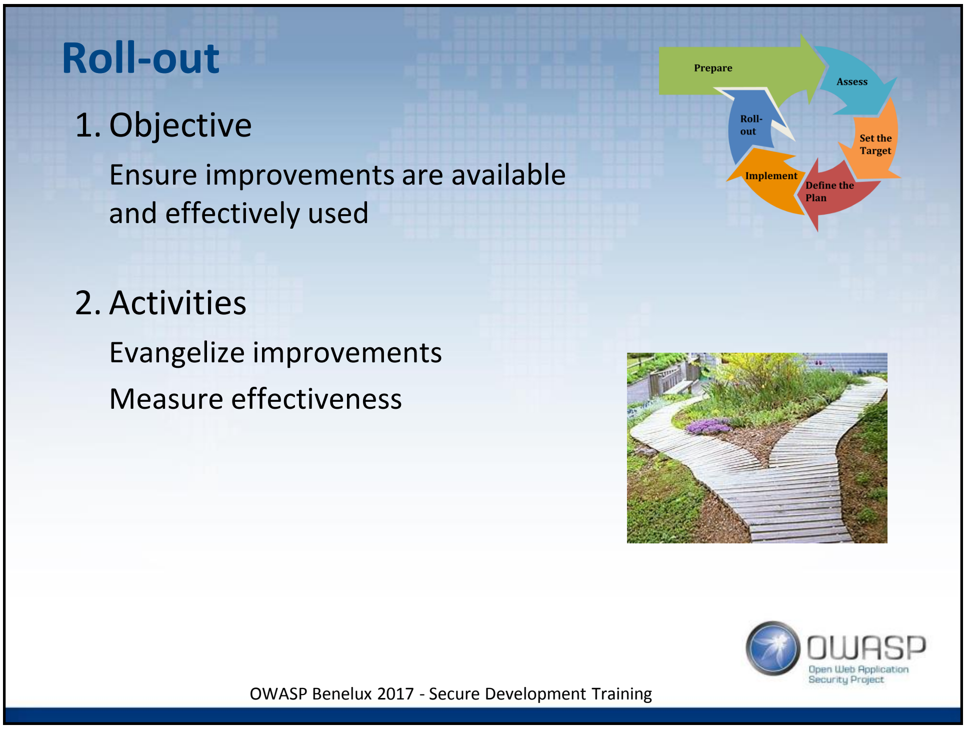
Implement

1. Objective

   Work the plan

2. Activities

   Implement activities



Roll-out

1. Objective

   Ensure improvements are available and effectively used

2. Activities

   Evangelize improvements

   Measure effectiveness

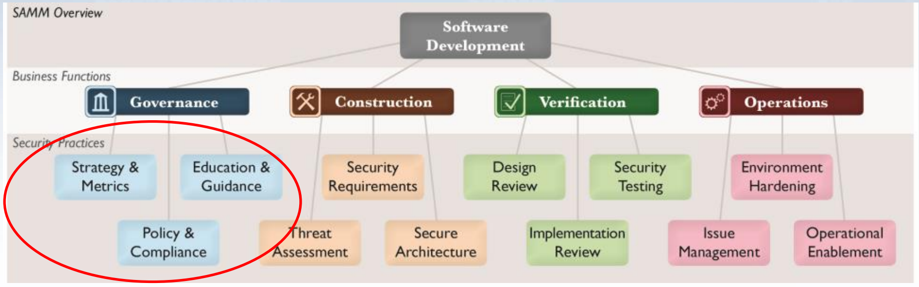
OWASP Benelux 2017 - Secure Development Training

OWASP Benelux 2017 - Secure Development Training

Governance
Business Function

OWASP Benelux 2017 - Secure Development Training

12 Security Practices



SAMM Overview

Software Development

Business Functions

- Governance
- Construction
- Verification
- Operations

Security Practices

- Strategy & Metrics
- Education & Guidance
- Policy & Compliance
- Security Requirements
- Threat Assessment
- Secure Architecture
- Design Review
- Security Testing
- Implementation Review
- Environment Hardening
- Issue Management
- Operational Enablement

OWASP Benelux 2017 - Secure Development Training

Strategy & Metrics

1. Goal is to establish a software assurance framework within an organisation

   Foundation for all other SAMM practices

2. Characteristics:

   Measurable

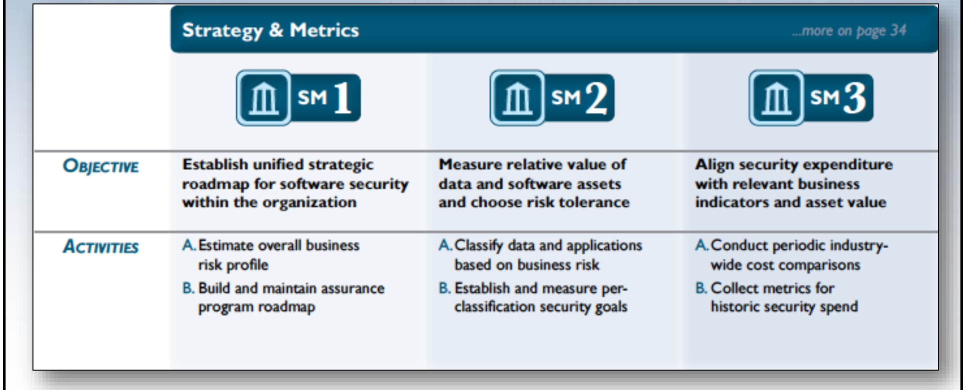
   Aligned with business risk

3. Driver for continuous improvement and financial guidance

VS.

OWASP Benelux 2017 - Secure Development Training

Strategy & Metrics



Strategy & Metrics ...more on page 34

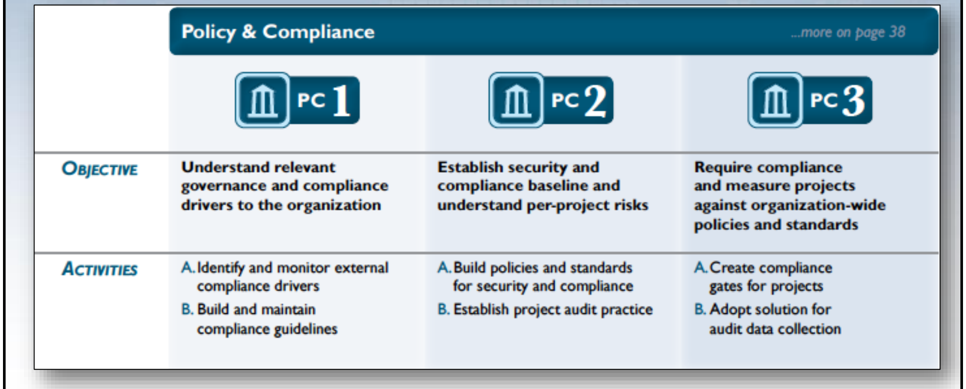
	SM 1	SM 2	SM 3
Objective	**Establish unified strategic roadmap for software security within the organization**	**Measure relative value of data and software assets and choose risk tolerance**	**Align security expenditure with relevant business indicators and asset value**
Activities	A. Estimate overall business risk profile B. Build and maintain assurance program roadmap	A. Classify data and applications based on business risk B. Establish and measure per-classification security goals	A. Conduct periodic industry-wide cost comparisons B. Collect metrics for historic security spend

OWASP Benelux 2017 - Secure Development Training

Policy & Compliance

1. Goal is to understand and adhere to legal and regulatory requirements
   - Typically external in nature
   - This is often a very informal practice in organisations !
2. Characteristics
   - Organisation-wide vs. project-specific
   - Scope
3. Important driver for software security requirements

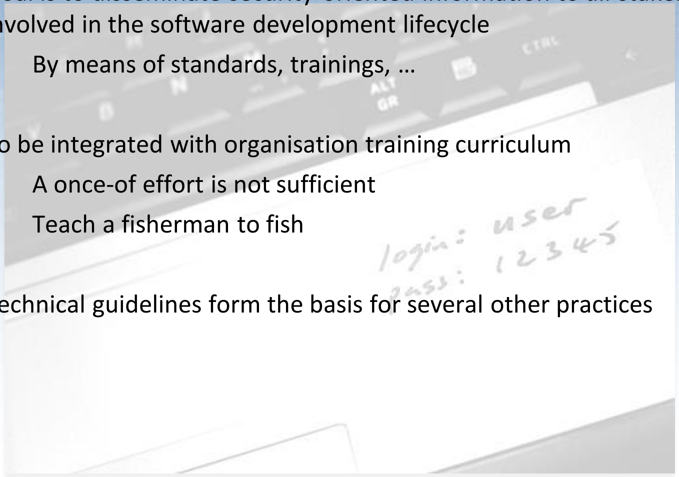
Policy & Compliance



Policy & Compliance ...more on page 38

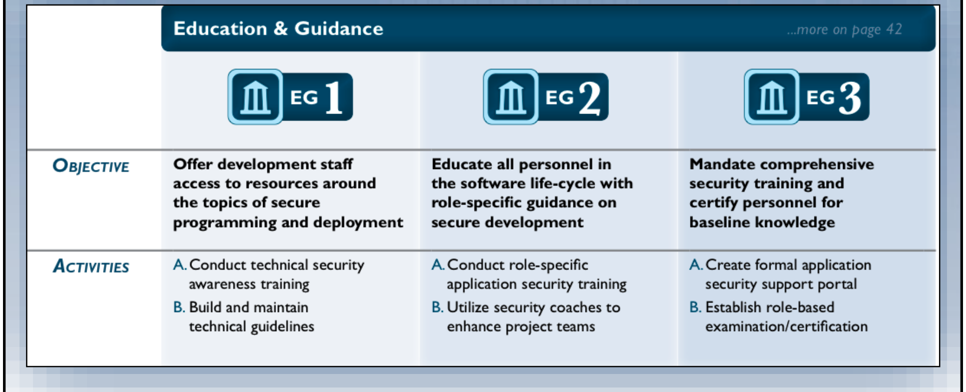
	PC 1	PC 2	PC 3
OBJECTIVE	**Understand relevant governance and compliance drivers to the organization**	**Establish security and compliance baseline and understand per-project risks**	**Require compliance and measure projects against organization-wide policies and standards**
ACTIVITIES	A. Identify and monitor external compliance drivers B. Build and maintain compliance guidelines	A. Build policies and standards for security and compliance B. Establish project audit practice	A. Create compliance gates for projects B. Adopt solution for audit data collection

Education & Guidance



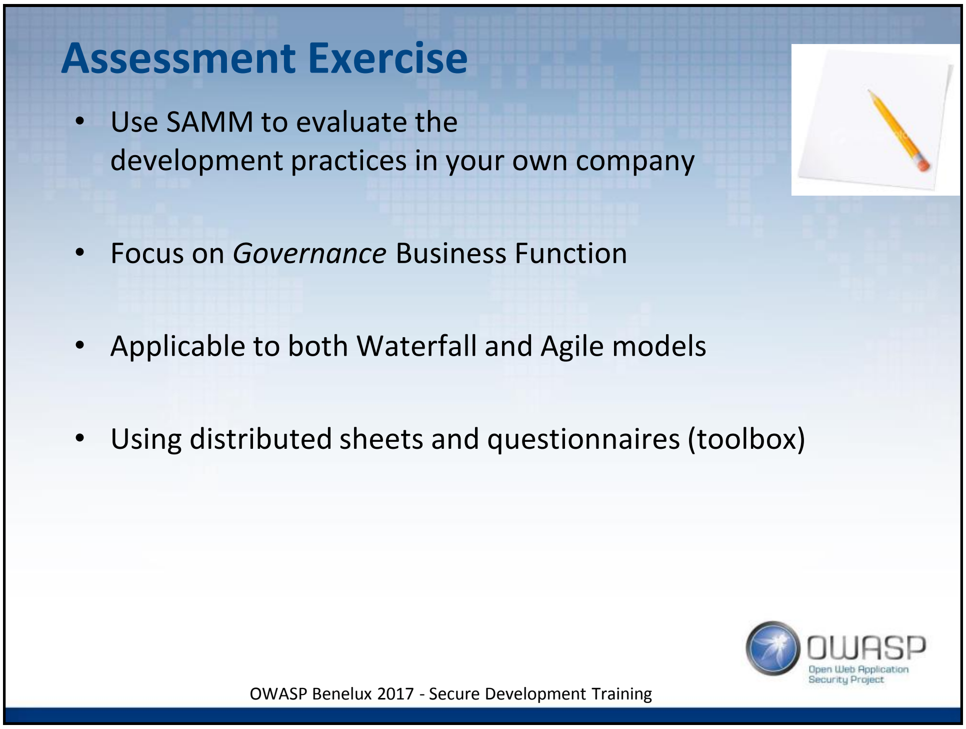
1. Goal is to disseminate security-oriented information to *all* stakeholders involved in the software development lifecycle
   - By means of standards, trainings, …
2. To be integrated with organisation training curriculum
   - A once-of effort is not sufficient
   - Teach a fisherman to fish
3. Technical guidelines form the basis for several other practices

Education & Guidance



Education & Guidance *…more on page 42*

	EG 1	EG 2	EG 3
OBJECTIVE	**Offer development staff access to resources around the topics of secure programming and deployment**	**Educate all personnel in the software life-cycle with role-specific guidance on secure development**	**Mandate comprehensive security training and certify personnel for baseline knowledge**
ACTIVITIES	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification



Assessment Exercise

- Use SAMM to evaluate the development practices in your own company
- Focus on *Governance* Business Function
- Applicable to both Waterfall and Agile models
- Using distributed sheets and questionnaires (toolbox)

OWASP Benelux 2017 - Secure Development Training

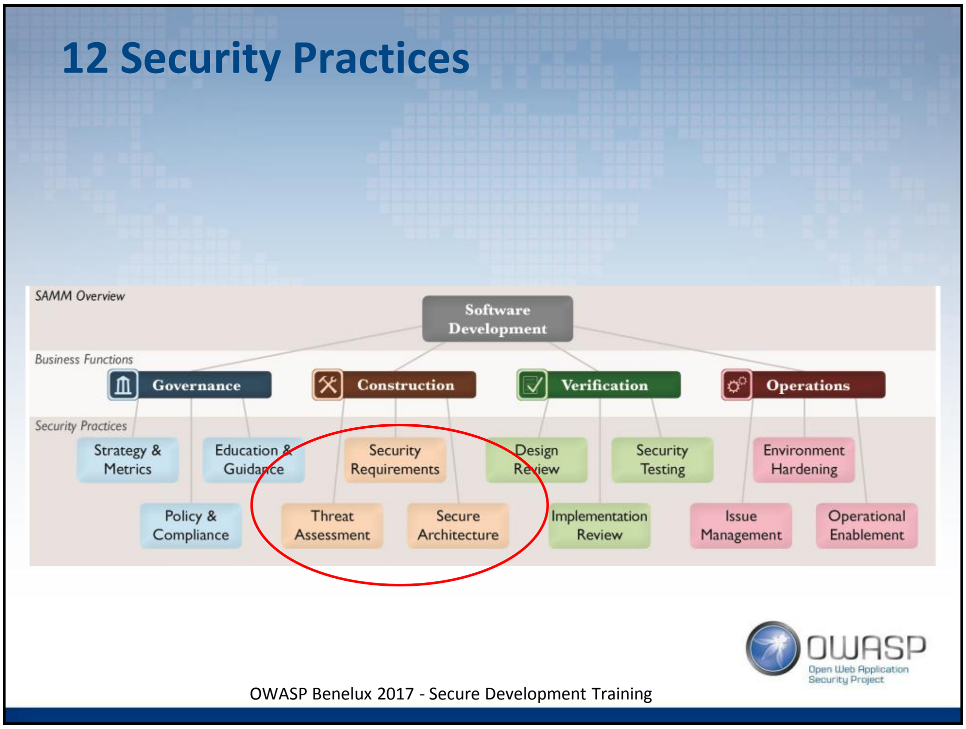
Assessment wrap-up

- What's your company's score ?
- What's the average scores for the group ?
- Any odd ratings ?

OWASP Benelux 2017 - Secure Development Training

Construction
Business Function

OWASP Benelux 2017 - Secure Development Training



12 Security Practices

SAMM Overview

Software Development

Business Functions

Governance

Construction

Verification

Operations

Security Practices

Strategy & Metrics

Education & Guidance

Policy & Compliance

Security Requirements

Threat Assessment

Secure Architecture

Design Review

Implementation Review

Security Testing

Environment Hardening

Issue Management

Operational Enablement

OWASP Benelux 2017 - Secure Development Training

Threat Assessment

1. The goal of this practice is to focus on the attacker perspective of things

   To make sure that security is not only functionality-driven

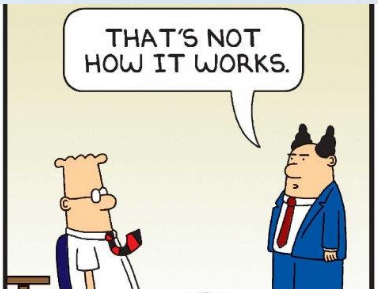
   Remember that software security = white + black

2. Very common practice in safety-critical systems

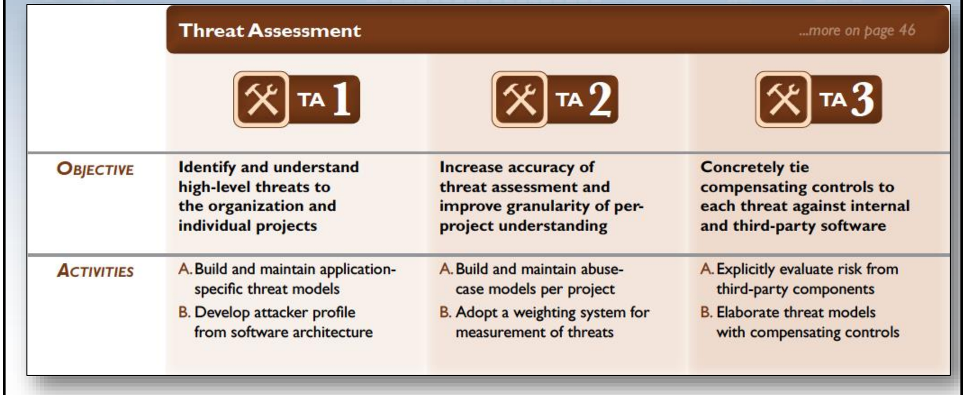
   Less so in others

3. This is where “the magic” kicks in

   Your imagination is the limit



Threat Assessment



Threat Assessment ...more on page 46

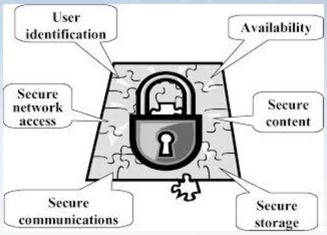
	TA 1	TA 2	TA 3
Objective	**Identify and understand high-level threats to the organization and individual projects**	**Increase accuracy of threat assessment and improve granularity of per-project understanding**	**Concretely tie compensating controls to each threat against internal and third-party software**
Activities	A. Build and maintain application-specific threat models B. Develop attacker profile from software architecture	A. Build and maintain abuse-case models per project B. Adopt a weighting system for measurement of threats	A. Explicitly evaluate risk from third-party components B. Elaborate threat models with compensating controls

Security Requirements

1. Goal is to make security specification more explicit

   Turn security into a positively-spaced problem

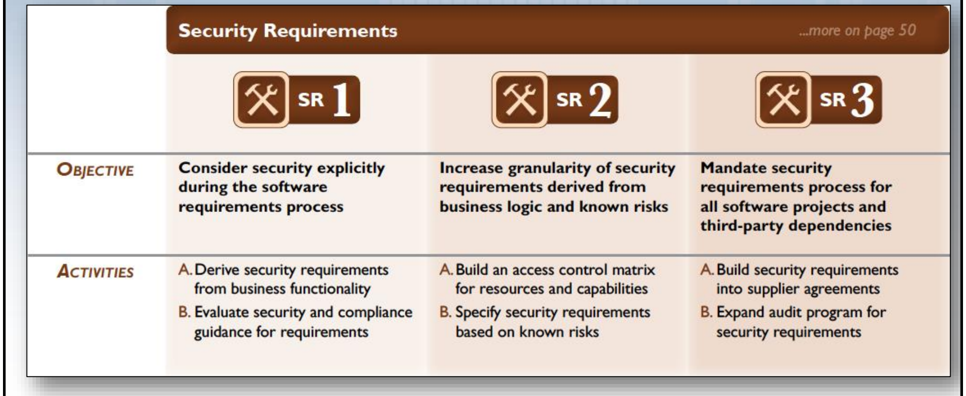
2. Source of security requirements
   - Compliance
   - Risk
   - Functionality
   - Quality



3. Requirements should be specified in a S.M.A.R.T. way

OWASP Benelux 2017 - Secure Development Training

Security Requirements



Security Requirements ...more on page 50

	SR 1	SR 2	SR 3
Objective	**Consider security explicitly during the software requirements process**	**Increase granularity of security requirements derived from business logic and known risks**	**Mandate security requirements process for all software projects and third-party dependencies**
Activities	A. Derive security requirements from business functionality B. Evaluate security and compliance guidance for requirements	A. Build an access control matrix for resources and capabilities B. Specify security requirements based on known risks	A. Build security requirements into supplier agreements B. Expand audit program for security requirements

OWASP Benelux 2017 - Secure Development Training

Secure Architecture

1. Key practice for security

   Poor decisions at this step can have major impact, and are often difficult (or costly) to fix.

2. Characteristics

   Take into account security principles

   Risk is a factor of all components (incl. 3rd party)

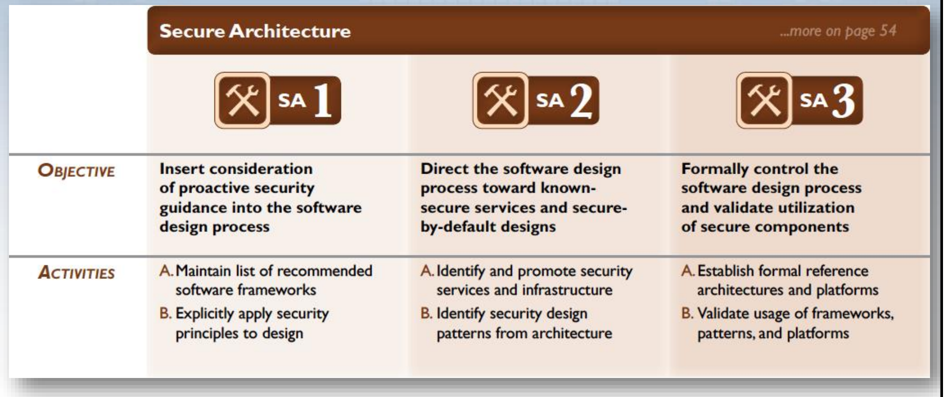
3. Use proven solutions

   Don't roll you own crypto

   Use company standards and best practices

OWASP Benelux 2017 - Secure Development Training

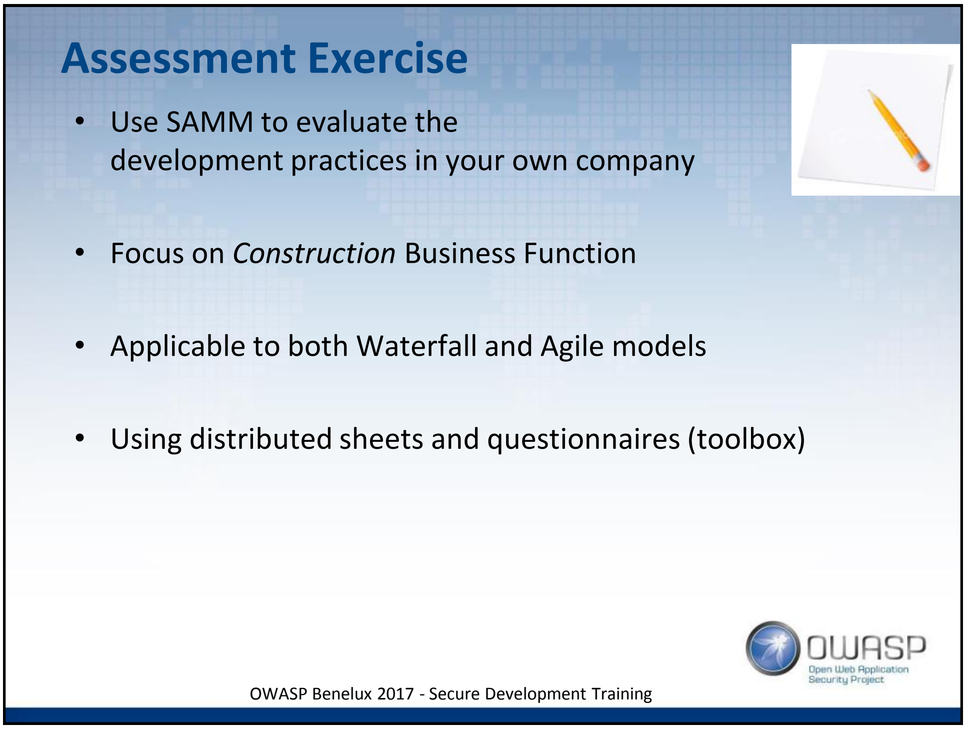
Secure Architecture



Secure Architecture ...more on page 54

	SA 1	SA 2	SA 3
Objective	Insert consideration of proactive security guidance into the software design process	Direct the software design process toward known-secure services and secure-by-default designs	Formally control the software design process and validate utilization of secure components
Activities	A. Maintain list of recommended software frameworks B. Explicitly apply security principles to design	A. Identify and promote security services and infrastructure B. Identify security design patterns from architecture	A. Establish formal reference architectures and platforms B. Validate usage of frameworks, patterns, and platforms

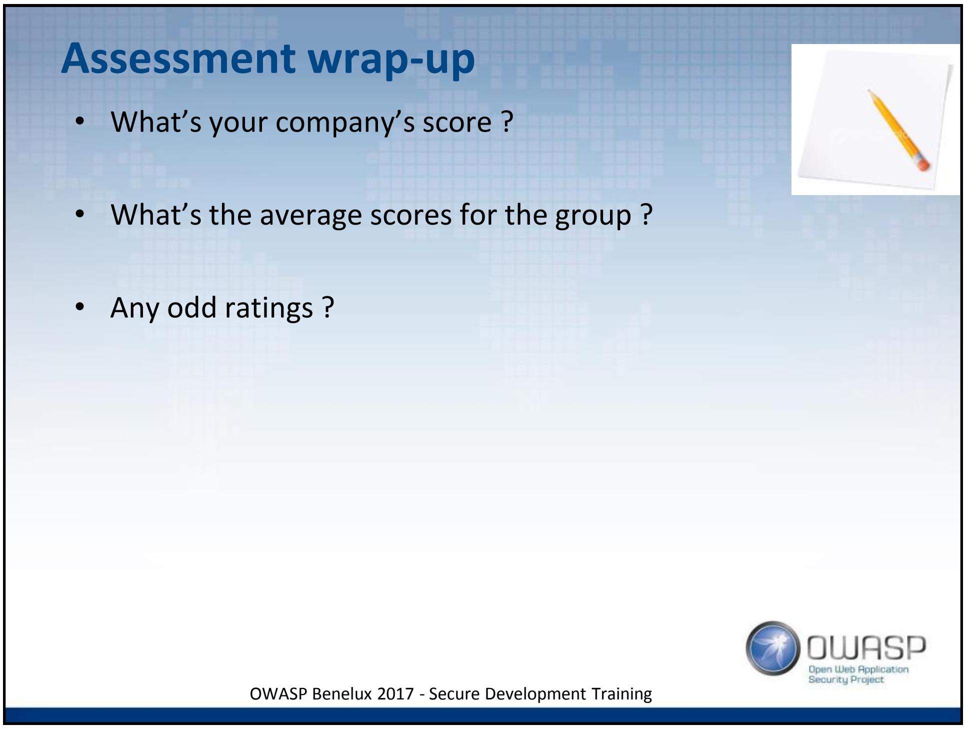
OWASP Benelux 2017 - Secure Development Training



Assessment Exercise

- Use SAMM to evaluate the development practices in your own company
- Focus on *Construction* Business Function
- Applicable to both Waterfall and Agile models
- Using distributed sheets and questionnaires (toolbox)

OWASP Benelux 2017 - Secure Development Training



Assessment wrap-up

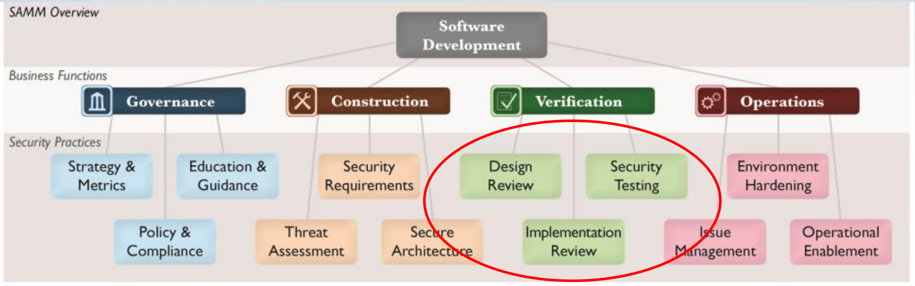
- What’s your company’s score ?
- What’s the average scores for the group ?
- Any odd ratings ?

OWASP Benelux 2017 - Secure Development Training

Verification
Business Function

OWASP Benelux 2017 - Secure Development Training

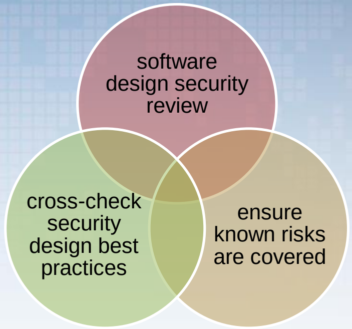
12 Security Practices



OWASP Benelux 2017 - Secure Development Training

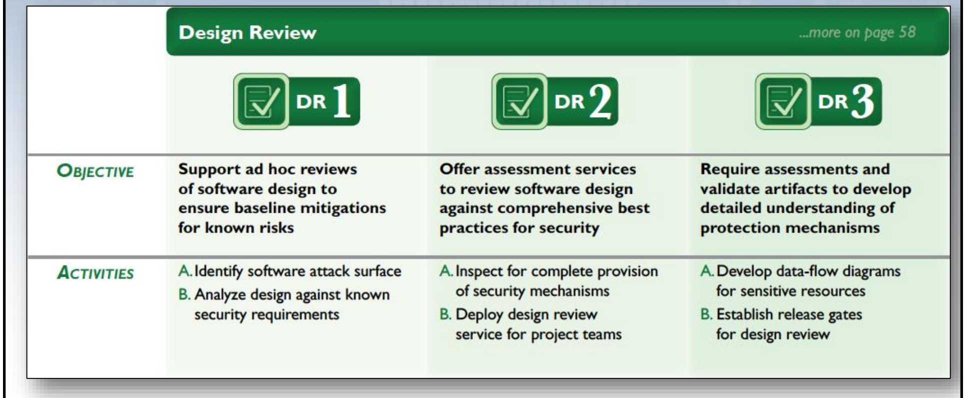
Design Review

- security assessment of attack surface, software design and architecture
- lightweight activities => formal inspection of data flows & security mechanisms
- enforcement of baseline expectations for conducting design assessments and reviewing findings before releases are accepted.


software design security review
cross-check security design best practices
ensure known risks are covered

⇒ Assess and validate artifacts to understand protection mechanisms

Design Review



Design Review *...more on page 58*

	DR 1	DR 2	DR 3
OBJECTIVE	**Support ad hoc reviews of software design to ensure baseline mitigations for known risks**	**Offer assessment services to review software design against comprehensive best practices for security**	**Require assessments and validate artifacts to develop detailed understanding of protection mechanisms**
ACTIVITIES	A. Identify software attack surface B. Analyze design against known security requirements	A. Inspect for complete provision of security mechanisms B. Deploy design review service for project teams	A. Develop data-flow diagrams for sensitive resources B. Establish release gates for design review

Implementation Review

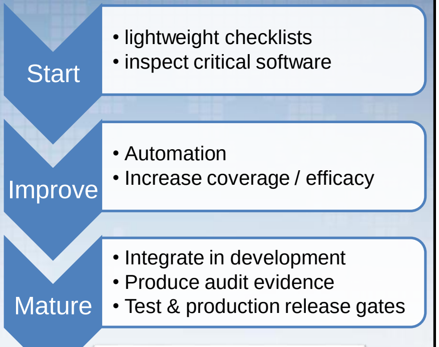
Assessment of source code:

- vulnerability discovery
- related mitigation activities
- establish secure coding baseline

Will require tool investment:

- Language specific
- Basic open source tooling
- Commercial tools maturing

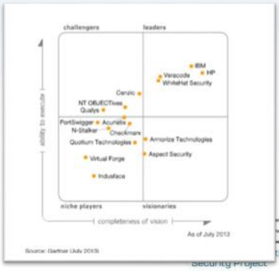
Process & education important!



Start
- lightweight checklists
- inspect critical software

Improve
- Automation
- Increase coverage / efficacy

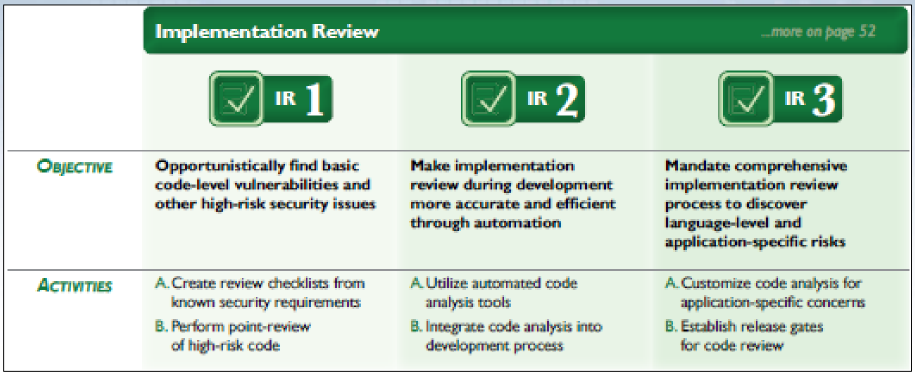
Mature
- Integrate in development
- Produce audit evidence
- Test & production release gates



Source: Gartner (July 2013)

OWASP Benelux 2017 - Secure Development Training

Implementation Review

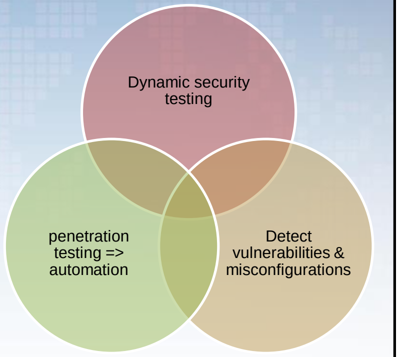


	Implementation Review		...more on page 52
	IR 1	IR 2	IR 3
Objective	Opportunistically find basic code-level vulnerabilities and other high-risk security issues	Make implementation review during development more accurate and efficient through automation	Mandate comprehensive implementation review process to discover language-level and application-specific risks
Activities	A. Create review checklists from known security requirements B. Perform point-review of high-risk code	A. Utilize automated code analysis tools B. Integrate code analysis into development process	A. Customize code analysis for application-specific concerns B. Establish release gates for code review

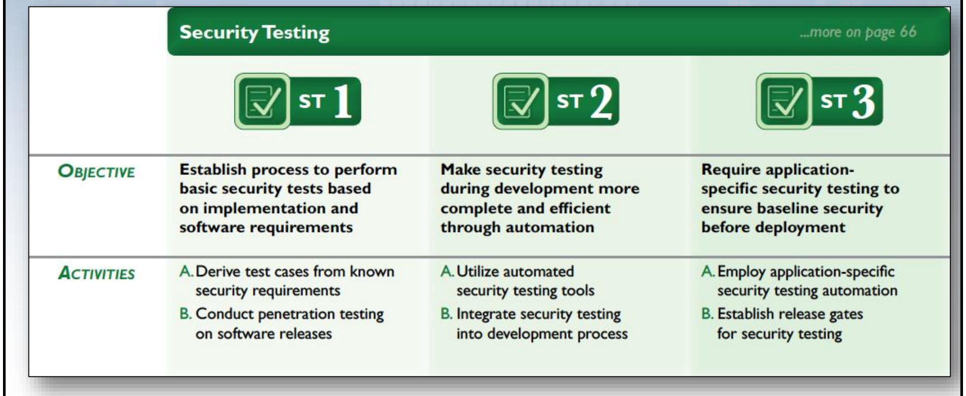
OWASP Benelux 2017 - Secure Development Training

Security Testing

- Based on security & compliance requirements / checklist of common vulnerabilities
- Manual testing can be done, scaled with tooling: intercepting proxy and/or scanner
- Detected defects will require validation, risk analysis & recommendations to fix
- Automate to repeat tests for each release
- Introduce security test-driven development
- Test results to be reported to & accepted by owner for each deployment

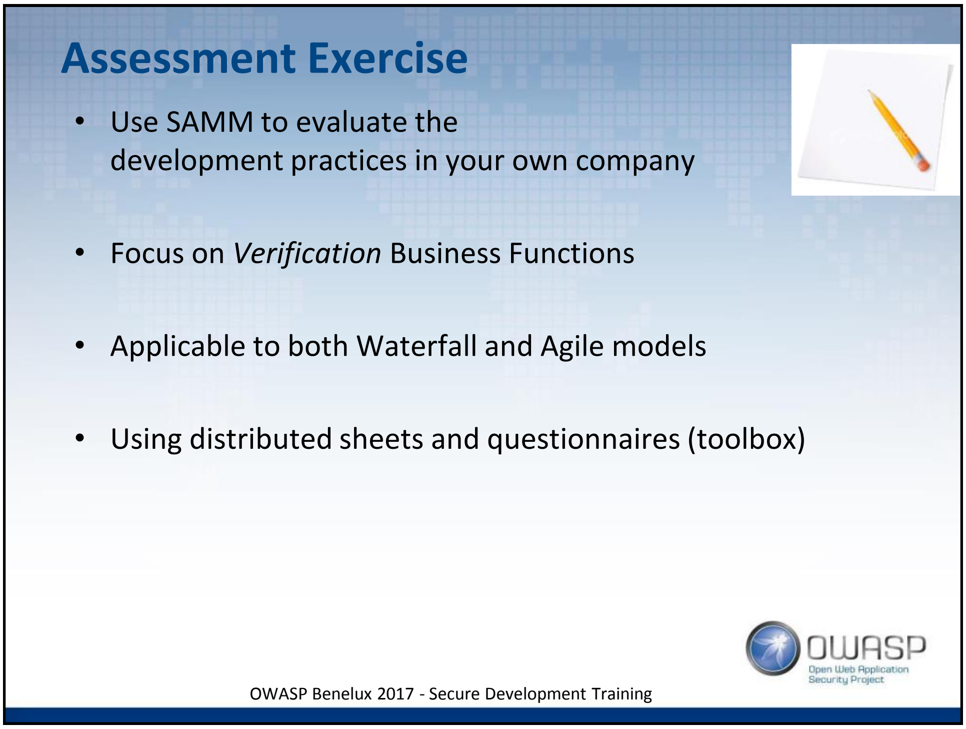

Dynamic security testing
penetration testing => automation
Detect vulnerabilities & misconfigurations

Security Testing



Security Testing *...more on page 66*

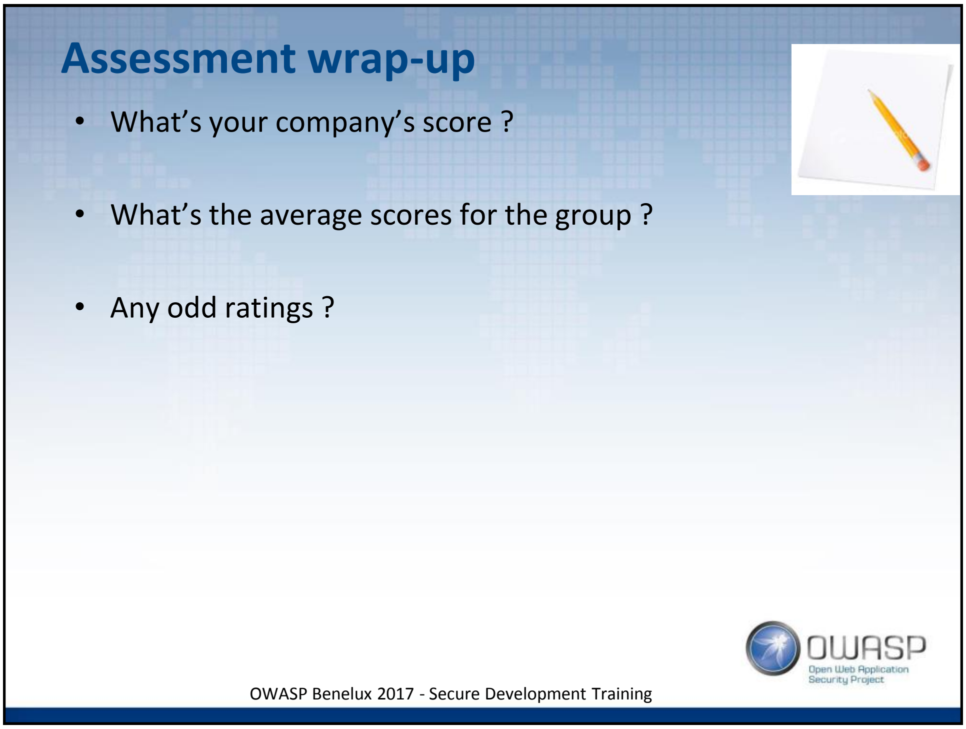
	ST 1	ST 2	ST 3
OBJECTIVE	Establish process to perform basic security tests based on implementation and software requirements	Make security testing during development more complete and efficient through automation	Require application-specific security testing to ensure baseline security before deployment
ACTIVITIES	A. Derive test cases from known security requirements B. Conduct penetration testing on software releases	A. Utilize automated security testing tools B. Integrate security testing into development process	A. Employ application-specific security testing automation B. Establish release gates for security testing



Assessment Exercise

- Use SAMM to evaluate the development practices in your own company
- Focus on *Verification* Business Functions
- Applicable to both Waterfall and Agile models
- Using distributed sheets and questionnaires (toolbox)

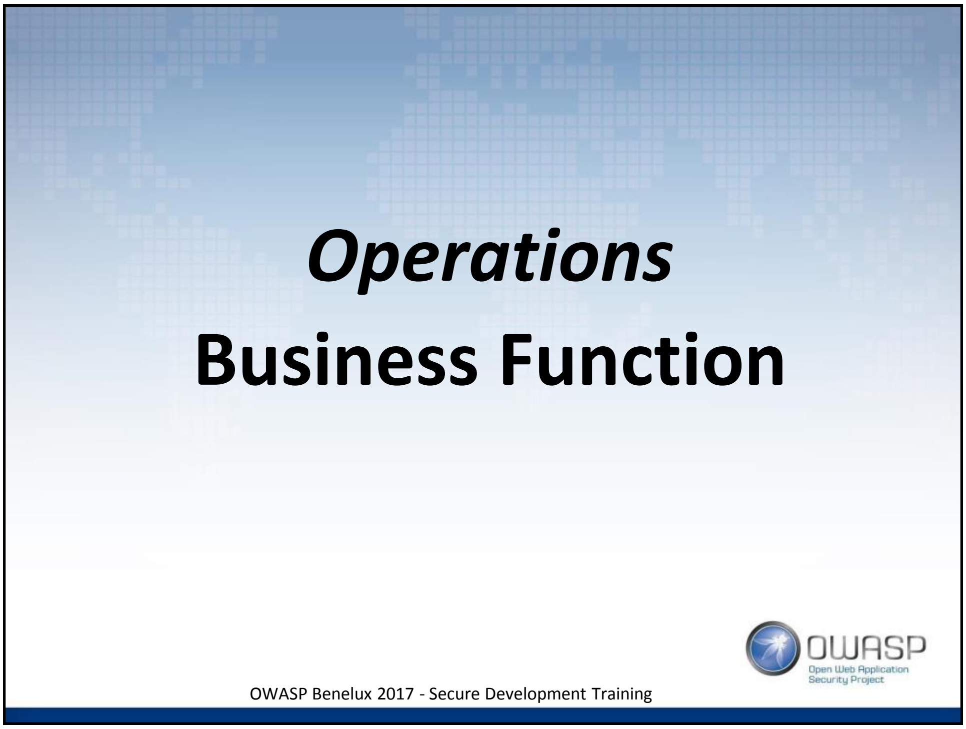
OWASP Benelux 2017 - Secure Development Training



Assessment wrap-up

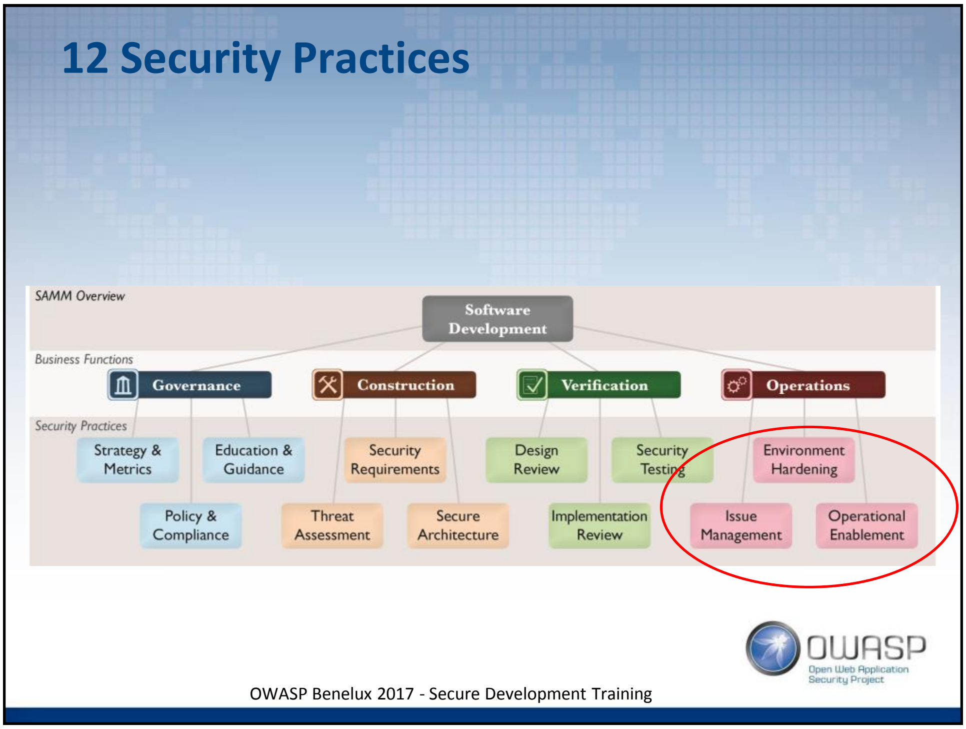
- What's your company's score ?
- What's the average scores for the group ?
- Any odd ratings ?

OWASP Benelux 2017 - Secure Development Training



Operations
Business Function

OWASP Benelux 2017 - Secure Development Training



12 Security Practices

OWASP Benelux 2017 - Secure Development Training

Issue Management

Prepare for WHEN, not IF!

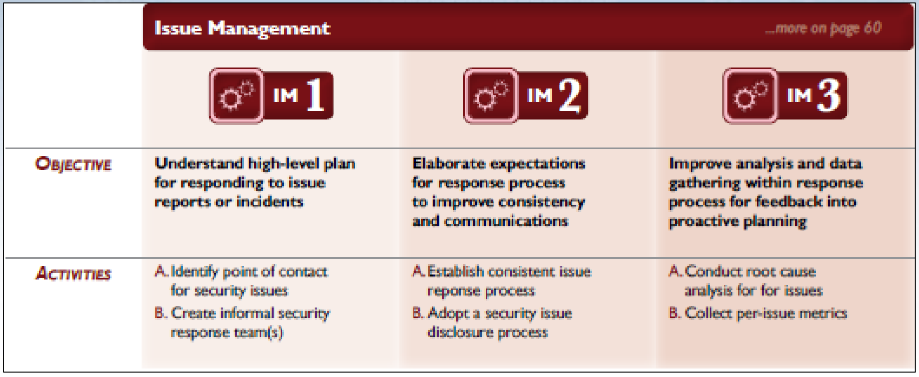
Symptoms of malfunctioning SDLC

- handling vulnerability reports and operational incidents
- lightweight assignment of roles=> formal incident response & communication process
- Use vulnerability metrics and root-cause analysis to improve SDLC

- spoc per team & security response team
- communication & information flow is key!
- patch release process & responsible/legal disclosure

OWASP Benelux 2017 - Secure Development Training

Issue Management



	Issue Management		...more on page 60
	IM 1	IM 2	IM 3
Objective	Understand high-level plan for responding to issue reports or incidents	Elaborate expectations for response process to improve consistency and communications	Improve analysis and data gathering within response process for feedback into proactive planning
Activities	A. Identify point of contact for security issues B. Create informal security response team(s)	A. Establish consistent issue reponse process B. Adopt a security issue disclosure process	A. Conduct root cause analysis for for issues B. Collect per-issue metrics

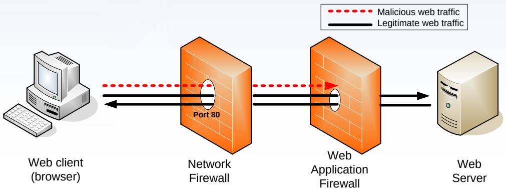
OWASP Benelux 2017 - Secure Development Training

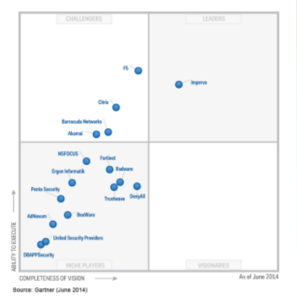
Environment Hardening

- Underlying infrastructure hardening & patching
- Track (3rd party) libraries & components

  TOP-10 - A9 – Using Known Vulnerable Components
- Add WAF layer (virtual patching)

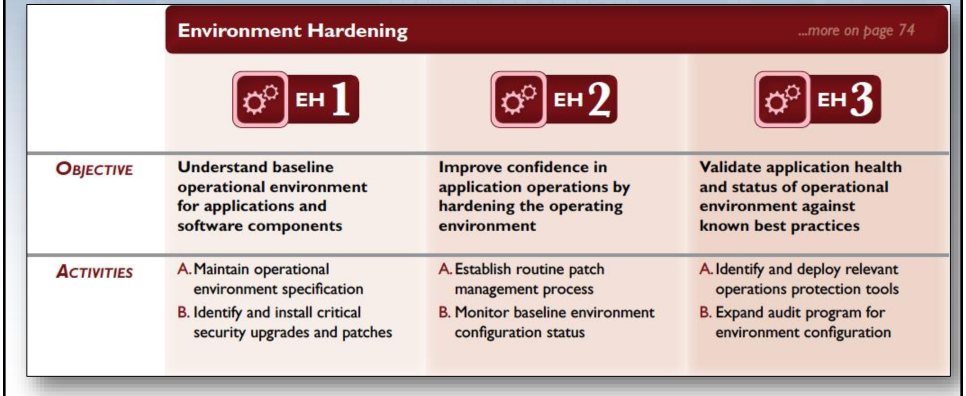
  ModSecurity





Source: Gartner (June 2014)

Environment Hardening

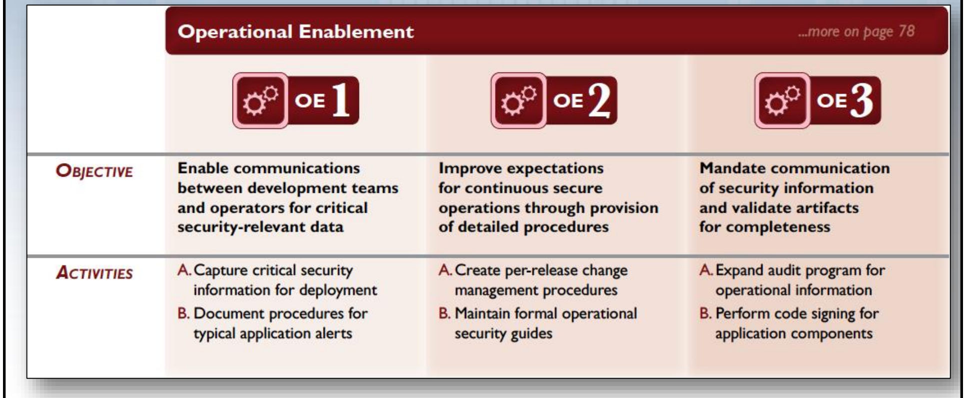


	Environment Hardening ...more on page 74		
	EH 1	EH 2	EH 3
OBJECTIVE	Understand baseline operational environment for applications and software components	Improve confidence in application operations by hardening the operating environment	Validate application health and status of operational environment against known best practices
ACTIVITIES	A. Maintain operational environment specification B. Identify and install critical security upgrades and patches	A. Establish routine patch management process B. Monitor baseline environment configuration status	A. Identify and deploy relevant operations protection tools B. Expand audit program for environment configuration

Operational Enablement

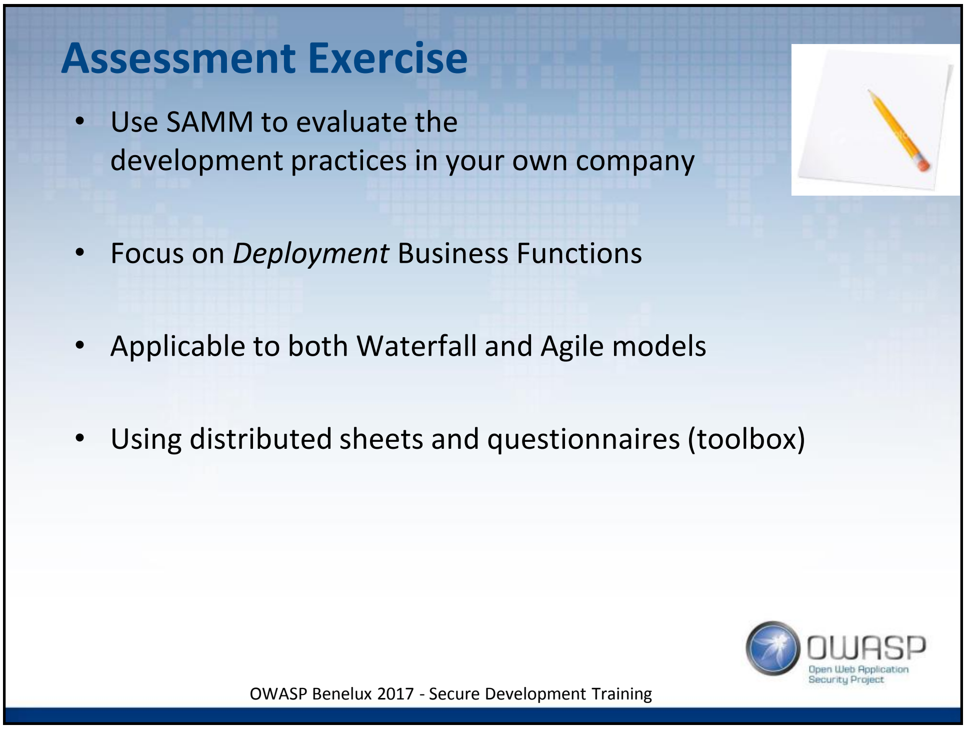
- Support users & operators
- Security documentation!
- Feed/document application security logs into SIEM
- Lightweight documentation => operational security guides
- Change management & end to end deployment integrity
- Even more important for outsourced development!

Operational Enablement



Operational Enablement ...more on page 78

	OE 1	OE 2	OE 3
Objective	**Enable communications between development teams and operators for critical security-relevant data**	**Improve expectations for continuous secure operations through provision of detailed procedures**	**Mandate communication of security information and validate artifacts for completeness**
Activities	A. Capture critical security information for deployment B. Document procedures for typical application alerts	A. Create per-release change management procedures B. Maintain formal operational security guides	A. Expand audit program for operational information B. Perform code signing for application components



Assessment Exercise

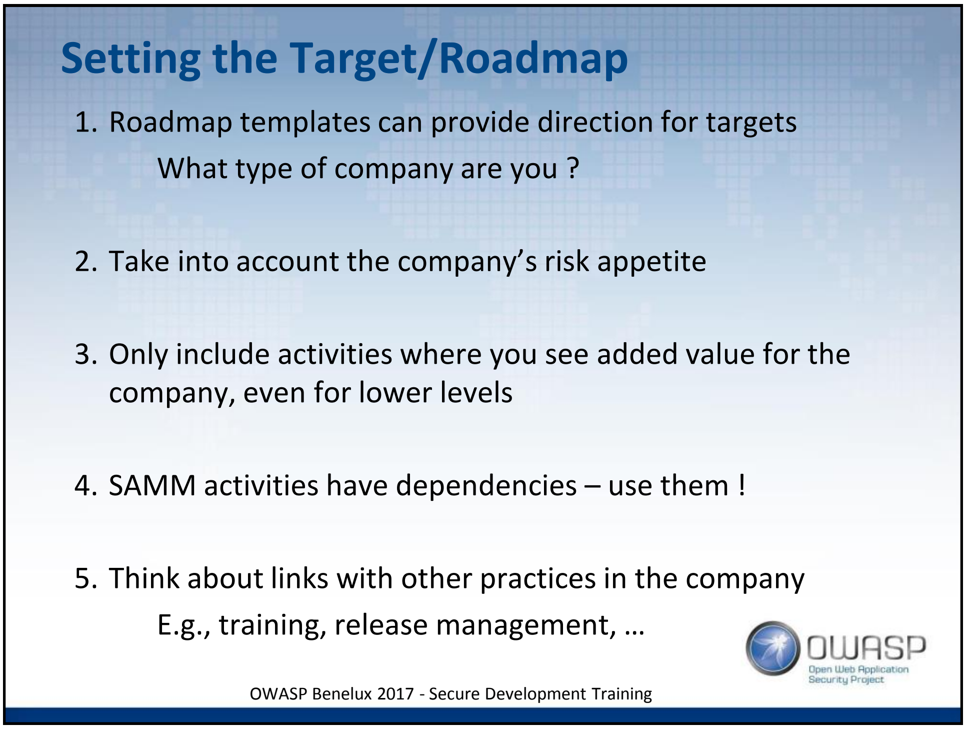
- Use SAMM to evaluate the development practices in your own company
- Focus on *Deployment* Business Functions
- Applicable to both Waterfall and Agile models
- Using distributed sheets and questionnaires (toolbox)

OWASP Benelux 2017 - Secure Development Training

Assessment wrap-up

- What's your company's score ?
- What's the average scores for the group ?
- Any odd ratings ?

OWASP Benelux 2017 - Secure Development Training



Setting the Target/Roadmap

1. Roadmap templates can provide direction for targets

   What type of company are you ?

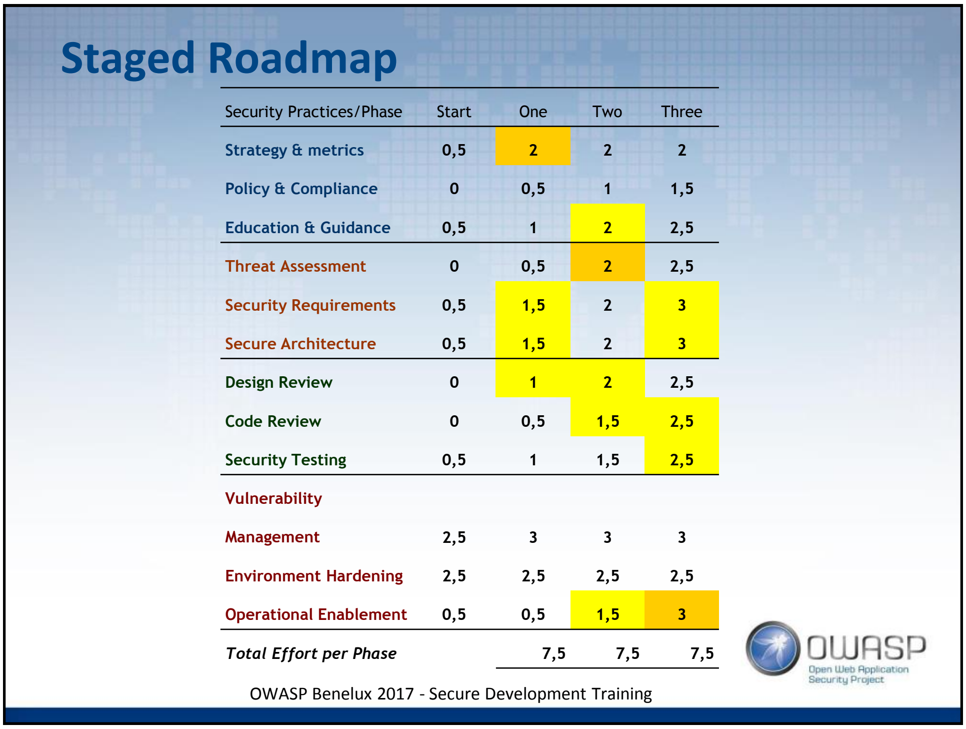
2. Take into account the company’s risk appetite

3. Only include activities where you see added value for the company, even for lower levels

4. SAMM activities have dependencies – use them !

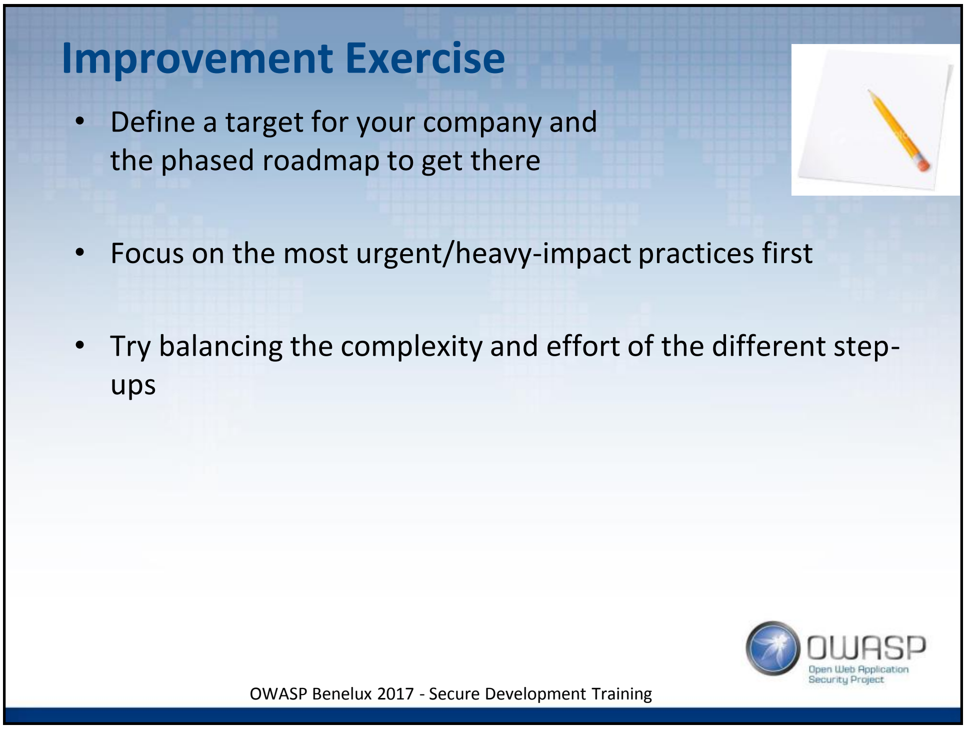
5. Think about links with other practices in the company

   E.g., training, release management, …



Staged Roadmap

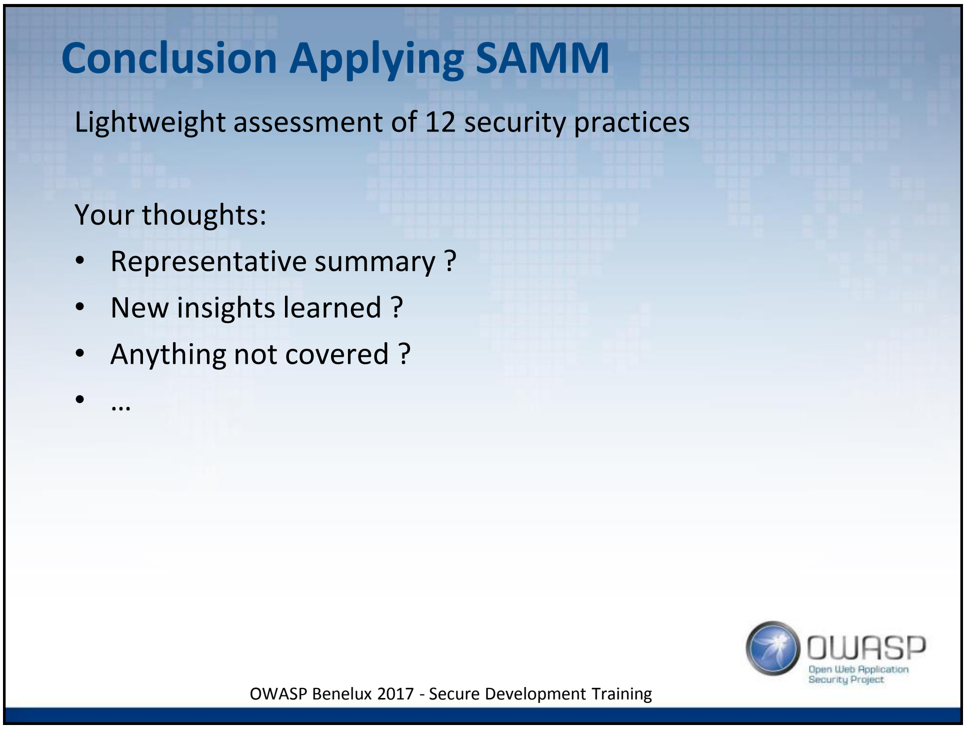
Security Practices/Phase	Start	One	Two	Three
Strategy & metrics	0,5	2	2	2
Policy & Compliance	0	0,5	1	1,5
Education & Guidance	0,5	1	2	2,5
Threat Assessment	0	0,5	2	2,5
Security Requirements	0,5	1,5	2	3
Secure Architecture	0,5	1,5	2	3
Design Review	0	1	2	2,5
Code Review	0	0,5	1,5	2,5
Security Testing	0,5	1	1,5	2,5
Vulnerability Management	2,5	3	3	3
Environment Hardening	2,5	2,5	2,5	2,5
Operational Enablement	0,5	0,5	1,5	3
Total Effort per Phase		7,5	7,5	7,5



Improvement Exercise

- Define a target for your company and the phased roadmap to get there
- Focus on the most urgent/heavy-impact practices first
- Try balancing the complexity and effort of the different step-ups

OWASP Benelux 2017 - Secure Development Training



Conclusion Applying SAMM

Lightweight assessment of 12 security practices

Your thoughts:

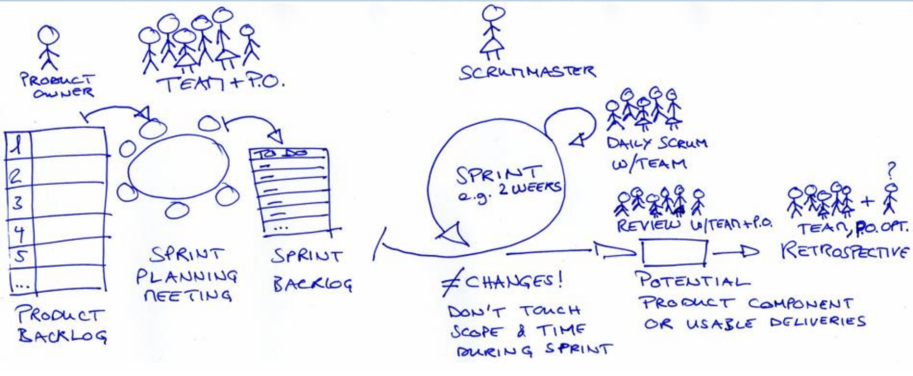
- Representative summary ?
- New insights learned ?
- Anything not covered ?
- …

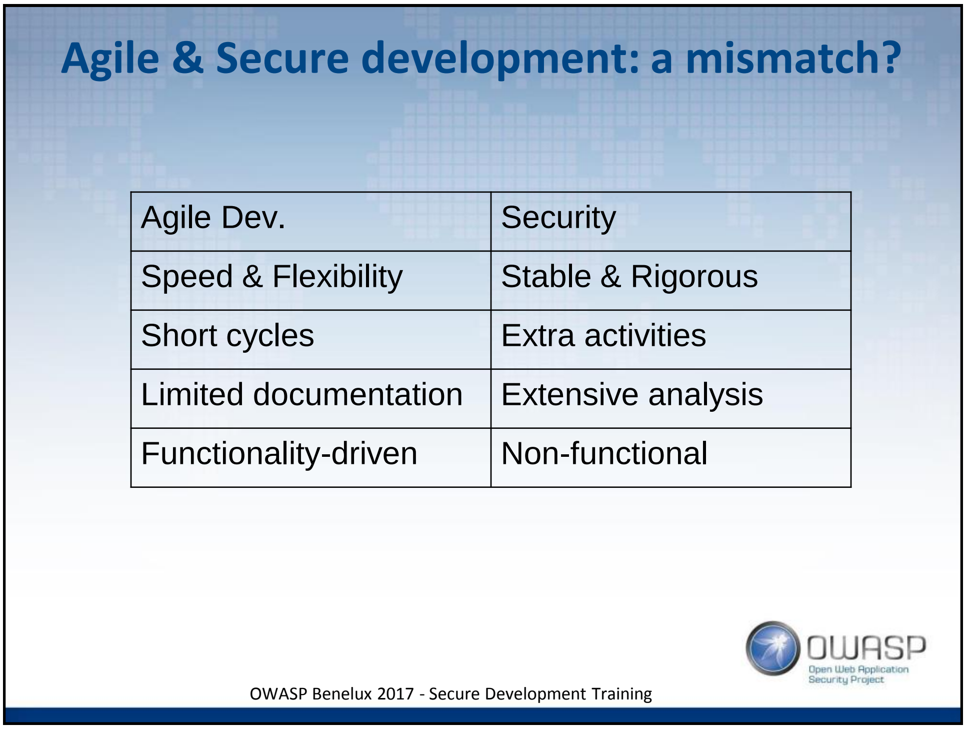
OWASP Benelux 2017 - Secure Development Training

Today's Agenda

1. Introduction to SDLC and SAMM
2. Applying SAMM
   - Methodology
   - Assessment Governance
   - Assessment Construction
   - Assessment Verification
   - Assessment Operations
   - Setting Improvement Targets
3. **Secure Agile development**
4. SDLC Tips and tricks
5. Wrap-up

Agile Models: Scrum

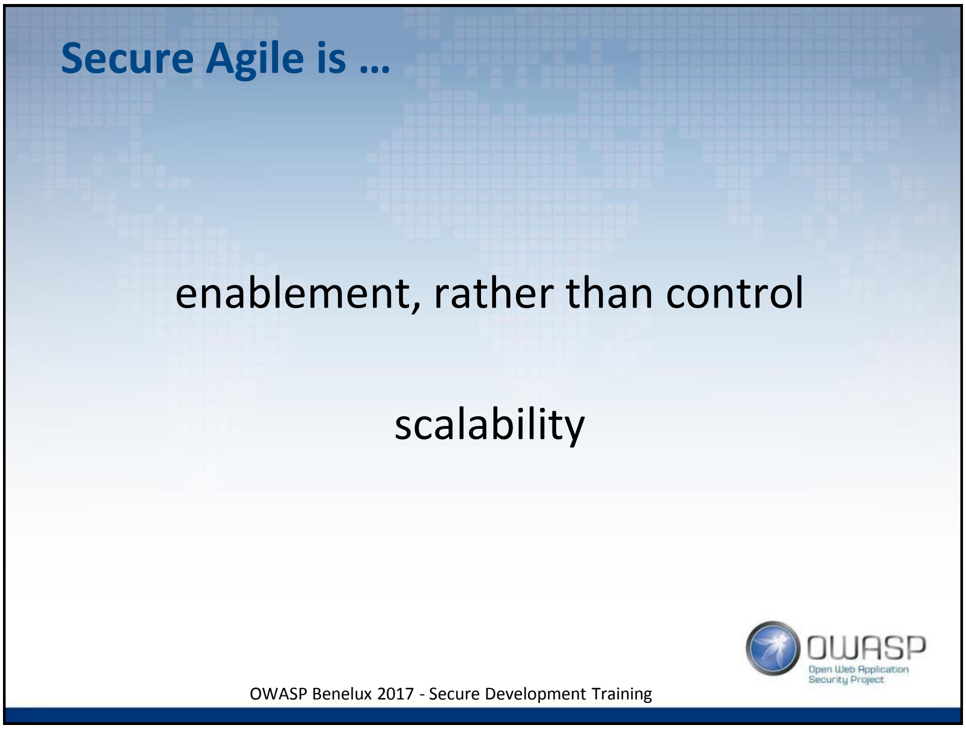




Agile & Secure development: a mismatch?

Agile Dev.	Security
Speed & Flexibility	Stable & Rigorous
Short cycles	Extra activities
Limited documentation	Extensive analysis
Functionality-driven	Non-functional

OWASP Benelux 2017 - Secure Development Training

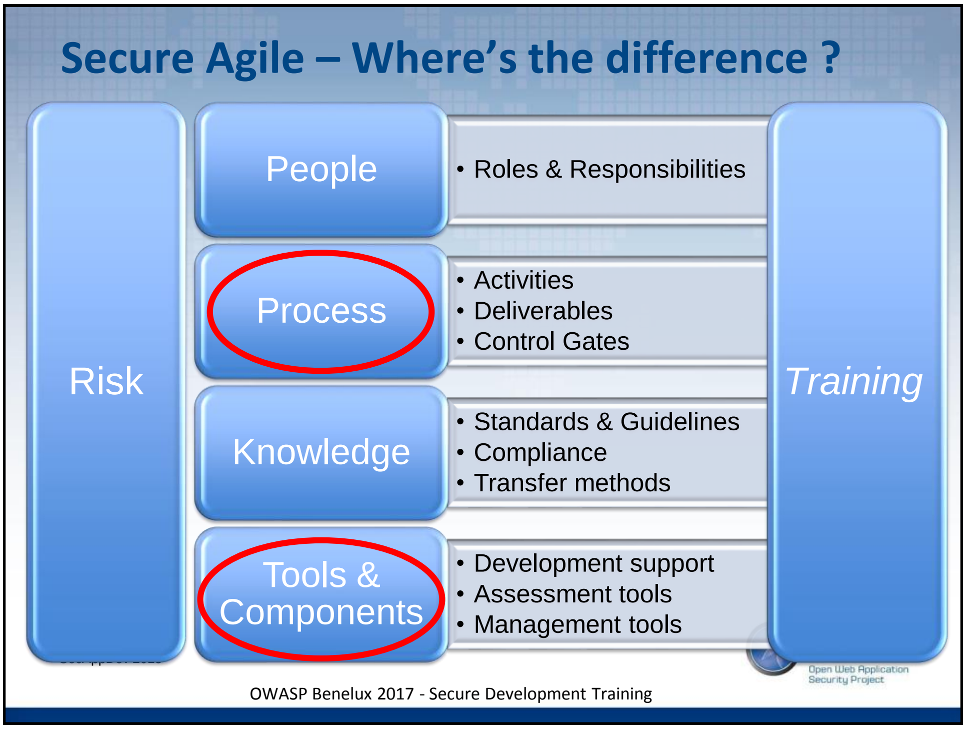


Secure Agile is …

enablement, rather than control

scalability

OWASP Benelux 2017 - Secure Development Training



Secure Agile – Where's the difference ?

Risk

People
- Roles & Responsibilities

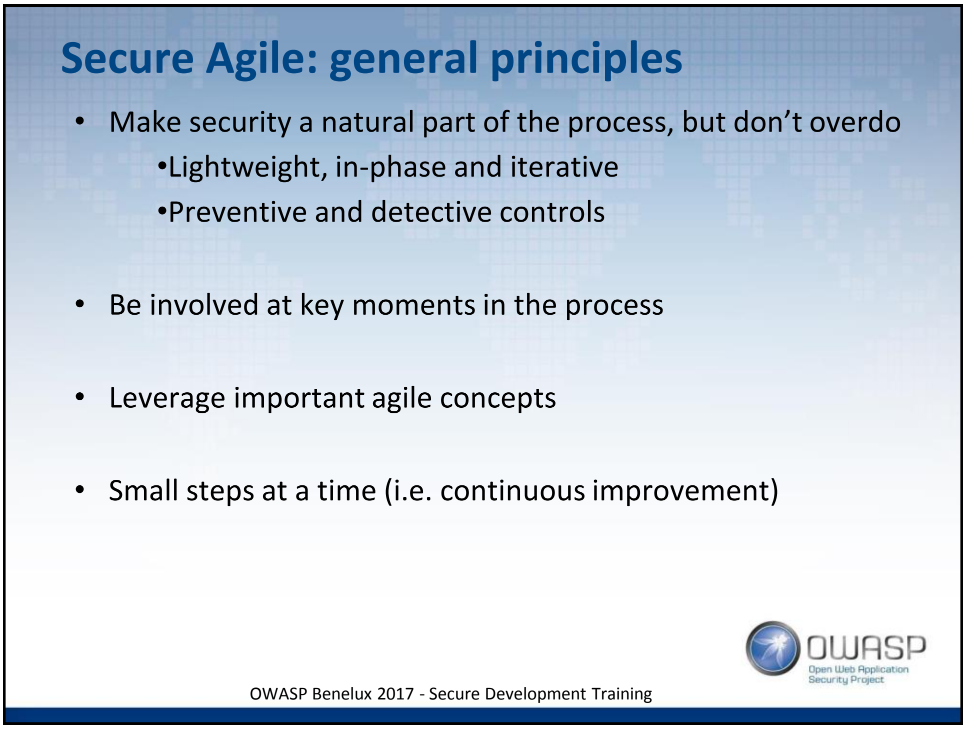
Process
- Activities
- Deliverables
- Control Gates

Knowledge
- Standards & Guidelines
- Compliance
- Transfer methods

Tools & Components
- Development support
- Assessment tools
- Management tools

Training

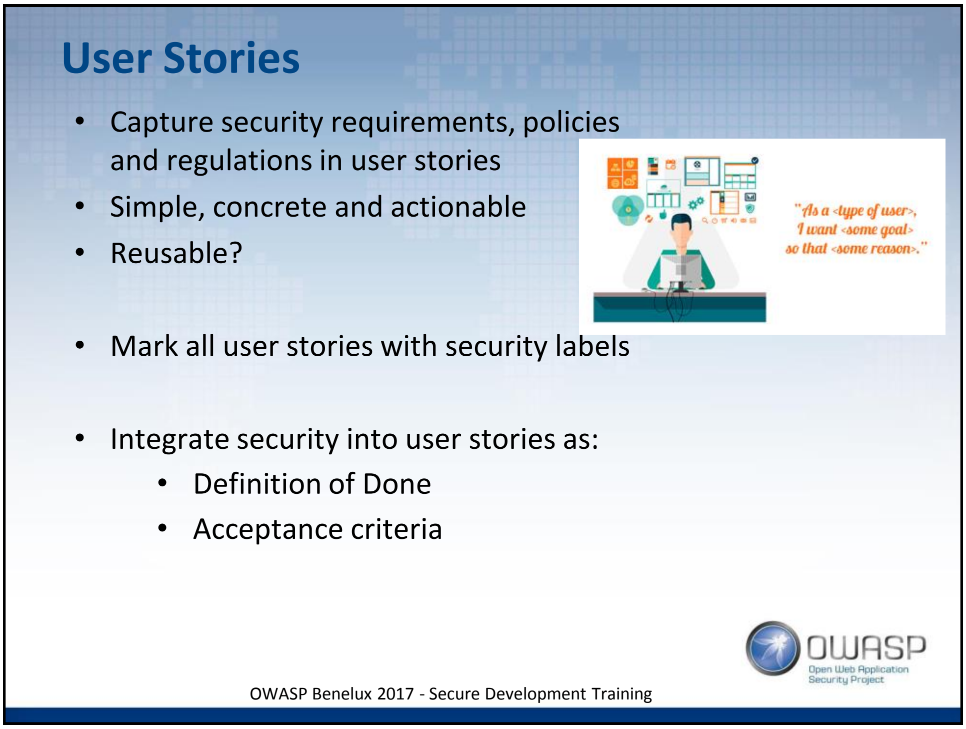
OWASP Benelux 2017 - Secure Development Training



Secure Agile: general principles

- Make security a natural part of the process, but don't overdo
  - Lightweight, in-phase and iterative
  - Preventive and detective controls
- Be involved at key moments in the process
- Leverage important agile concepts
- Small steps at a time (i.e. continuous improvement)

OWASP Benelux 2017 - Secure Development Training



User Stories

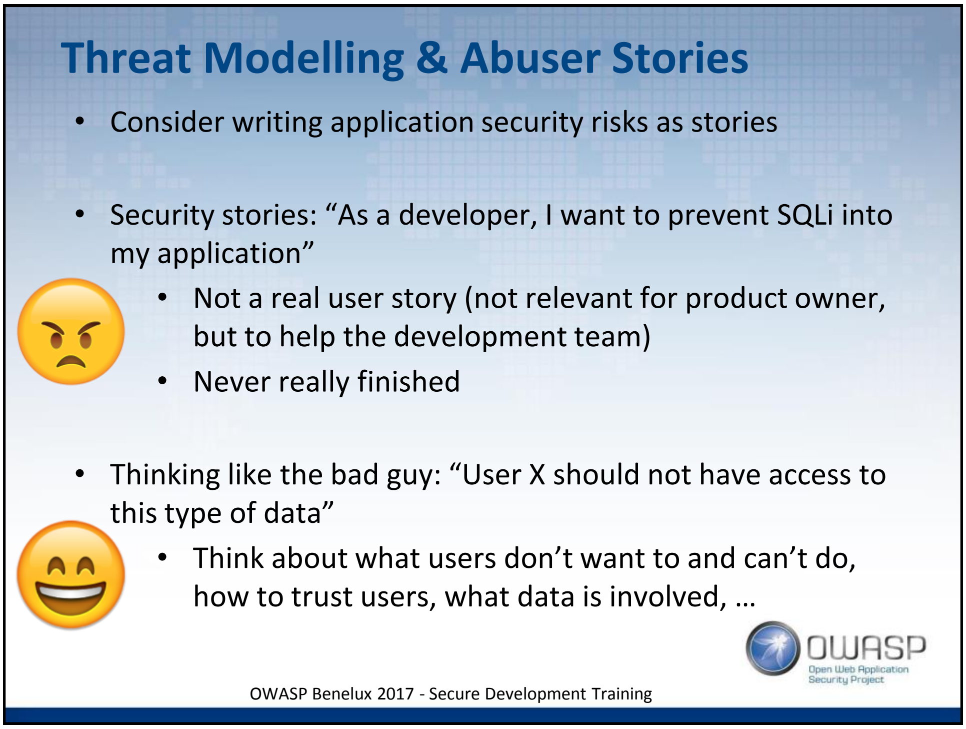
- Capture security requirements, policies and regulations in user stories
- Simple, concrete and actionable
- Reusable?

"As a <type of user>, I want <some goal> so that <some reason>."

- Mark all user stories with security labels

- Integrate security into user stories as:
  - Definition of Done
  - Acceptance criteria

OWASP Benelux 2017 - Secure Development Training



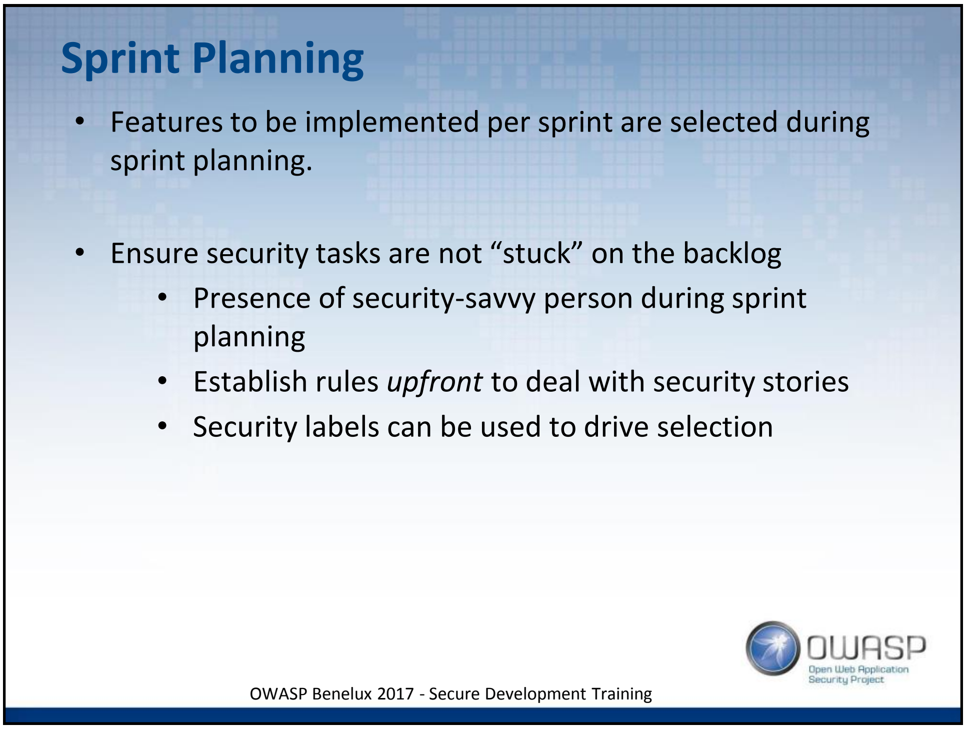
Threat Modelling & Abuser Stories

- Consider writing application security risks as stories

- Security stories: "As a developer, I want to prevent SQLi into my application"
  - Not a real user story (not relevant for product owner, but to help the development team)
  - Never really finished

- Thinking like the bad guy: "User X should not have access to this type of data"
  - Think about what users don't want to and can't do, how to trust users, what data is involved, …

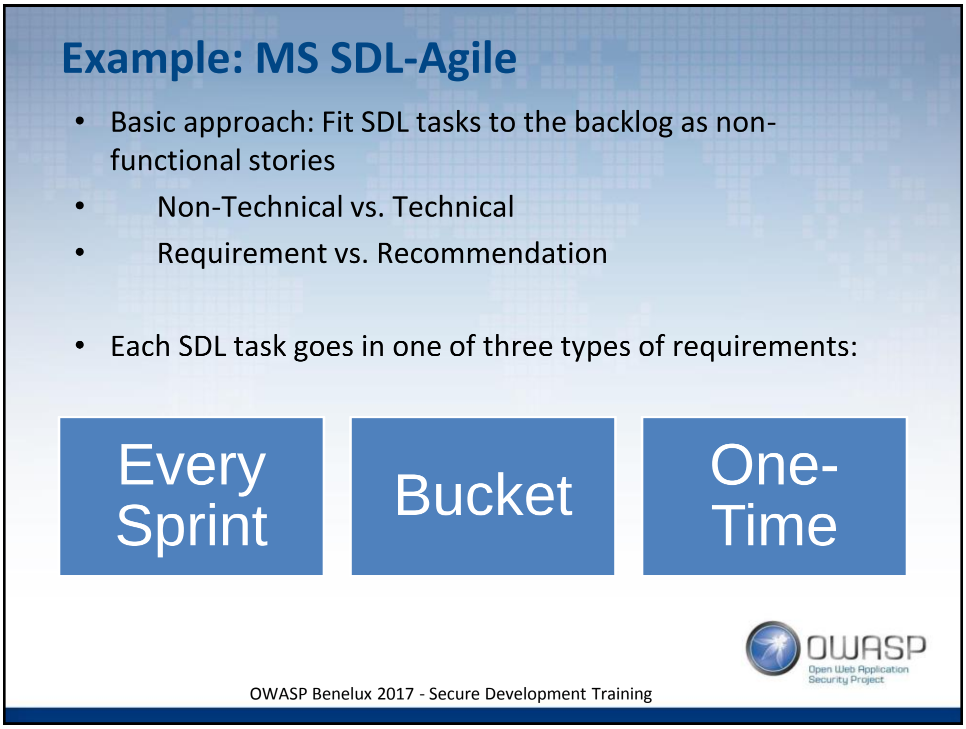
OWASP Benelux 2017 - Secure Development Training



Sprint Planning

- Features to be implemented per sprint are selected during sprint planning.

- Ensure security tasks are not “stuck” on the backlog
  - Presence of security-savvy person during sprint planning
  - Establish rules *upfront* to deal with security stories
  - Security labels can be used to drive selection



Example: MS SDL-Agile

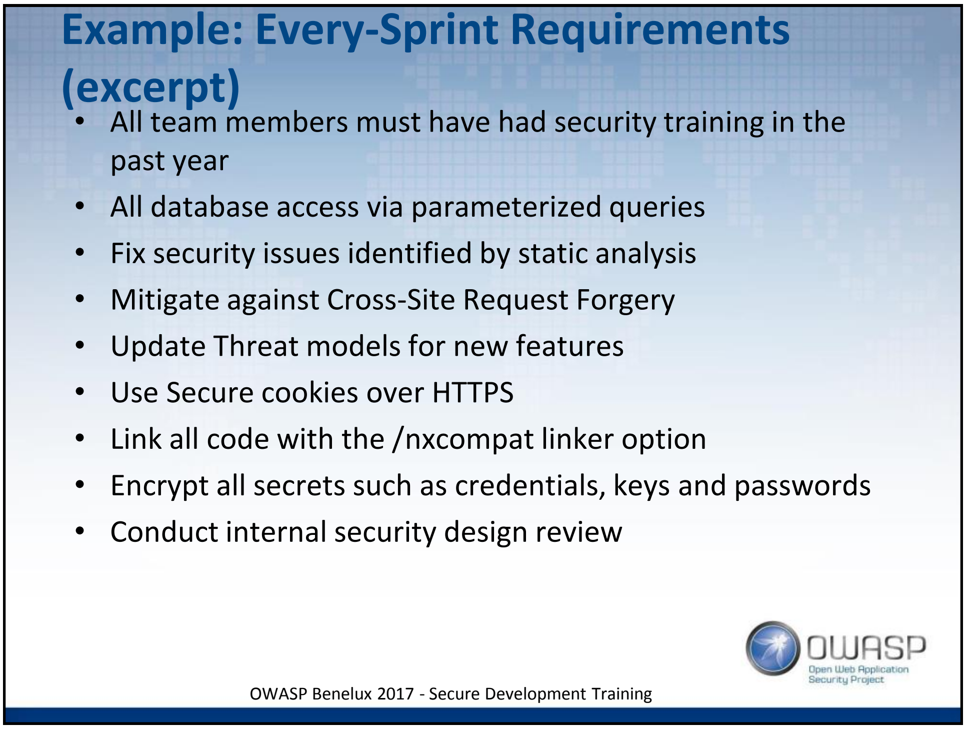
- Basic approach: Fit SDL tasks to the backlog as non-functional stories
- Non-Technical vs. Technical
- Requirement vs. Recommendation

- Each SDL task goes in one of three types of requirements:

Every Sprint

Bucket

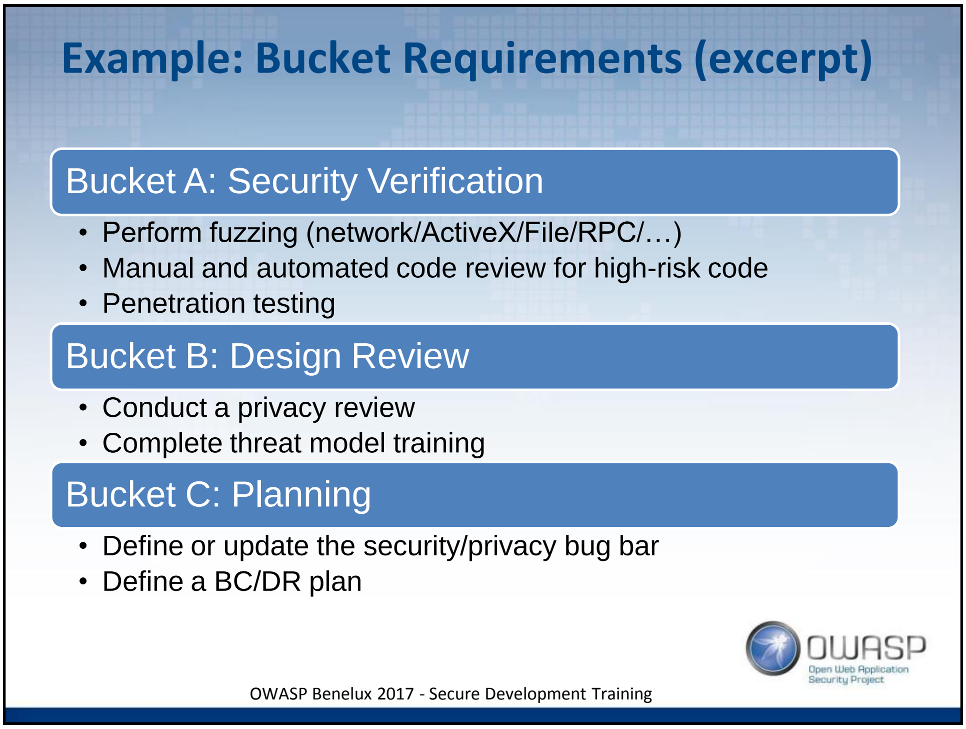
One-Time



Example: Every-Sprint Requirements (excerpt)

- All team members must have had security training in the past year
- All database access via parameterized queries
- Fix security issues identified by static analysis
- Mitigate against Cross-Site Request Forgery
- Update Threat models for new features
- Use Secure cookies over HTTPS
- Link all code with the /nxcompat linker option
- Encrypt all secrets such as credentials, keys and passwords
- Conduct internal security design review

OWASP Benelux 2017 - Secure Development Training



Example: Bucket Requirements (excerpt)

Bucket A: Security Verification

- Perform fuzzing (network/ActiveX/File/RPC/…)
- Manual and automated code review for high-risk code
- Penetration testing

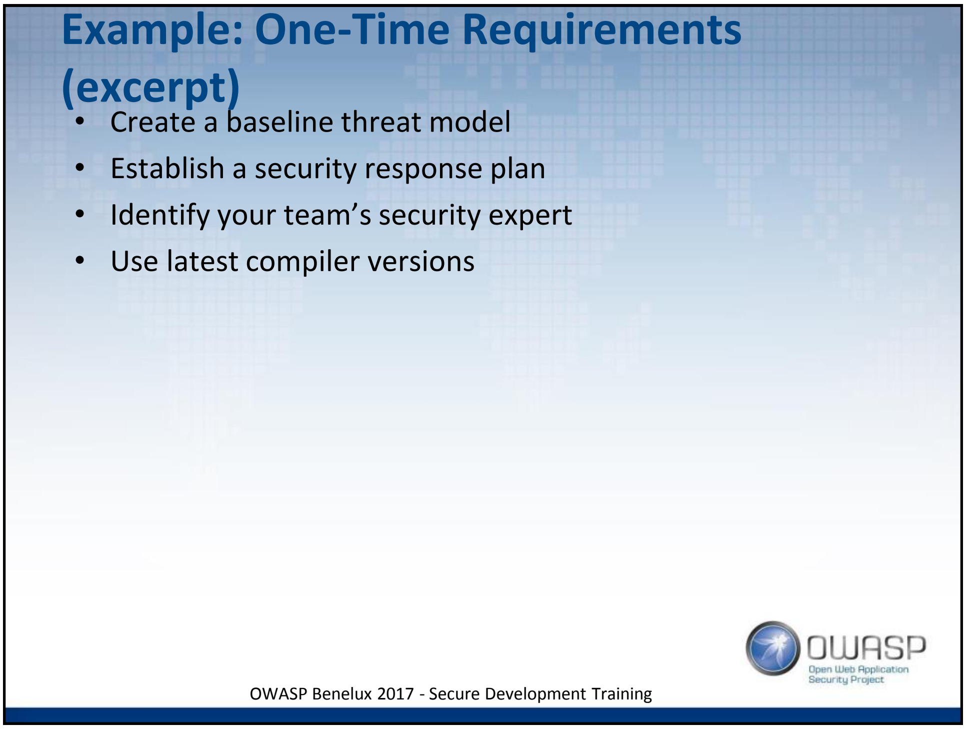
Bucket B: Design Review

- Conduct a privacy review
- Complete threat model training

Bucket C: Planning

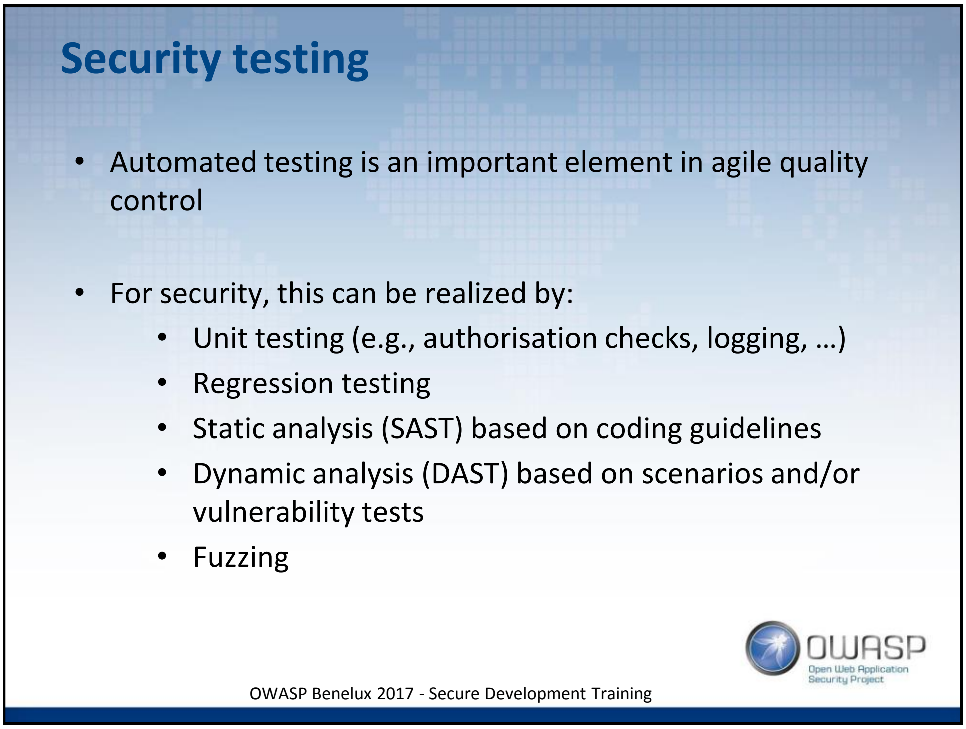
- Define or update the security/privacy bug bar
- Define a BC/DR plan

OWASP Benelux 2017 - Secure Development Training



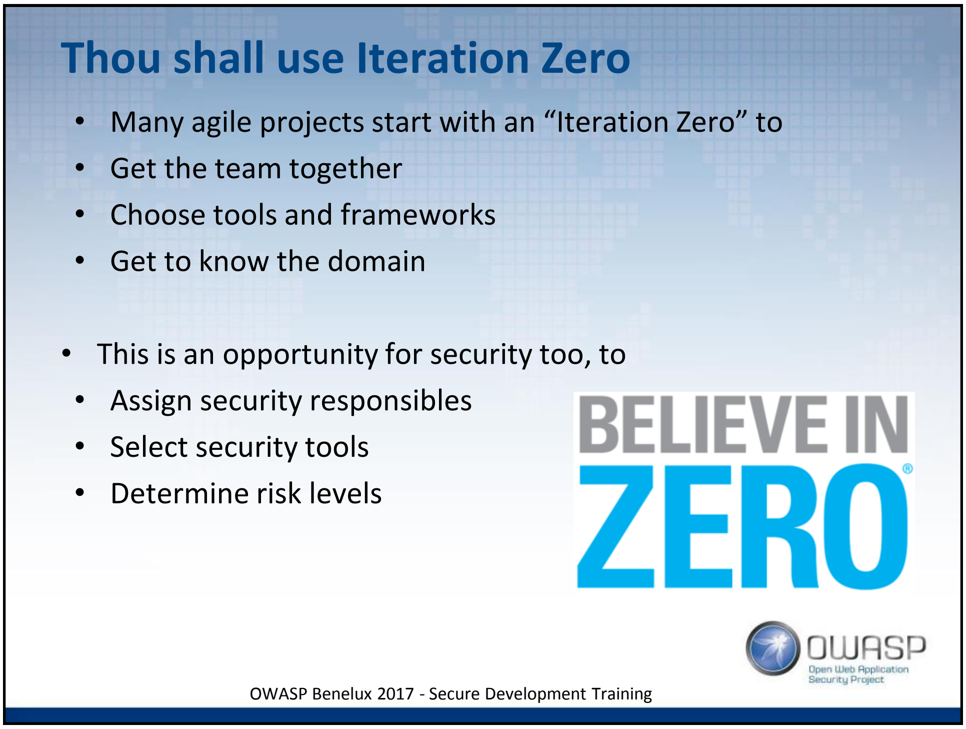
Example: One-Time Requirements (excerpt)

- Create a baseline threat model
- Establish a security response plan
- Identify your team's security expert
- Use latest compiler versions



Security testing

- Automated testing is an important element in agile quality control
- For security, this can be realized by:
  - Unit testing (e.g., authorisation checks, logging, …)
  - Regression testing
  - Static analysis (SAST) based on coding guidelines
  - Dynamic analysis (DAST) based on scenarios and/or vulnerability tests
  - Fuzzing

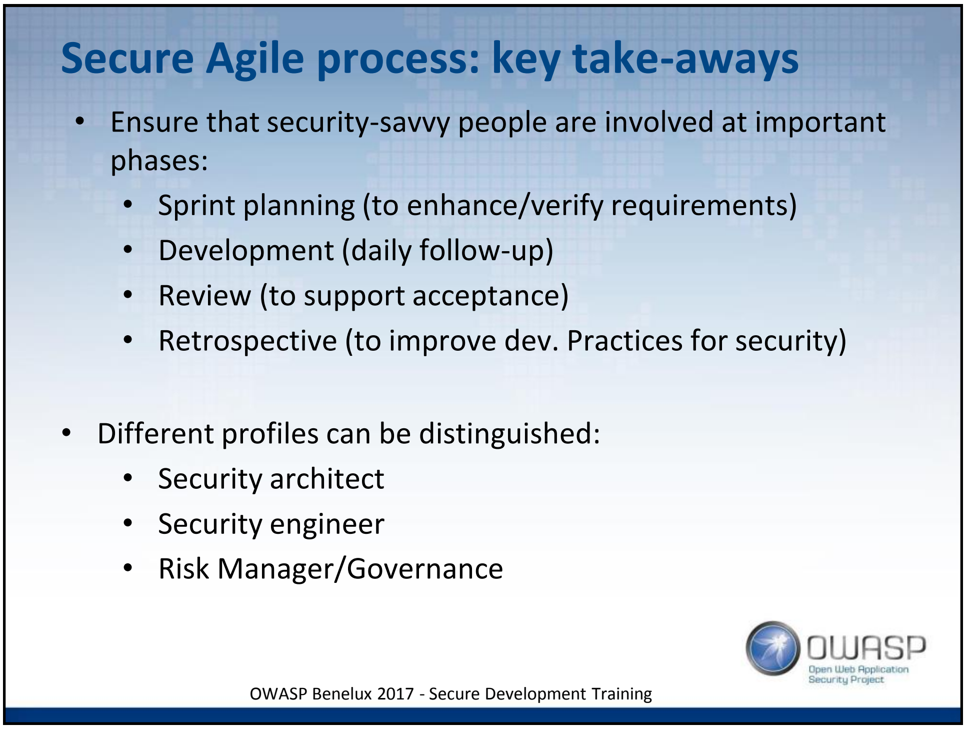


Thou shall use Iteration Zero

- Many agile projects start with an “Iteration Zero” to
- Get the team together
- Choose tools and frameworks
- Get to know the domain

- This is an opportunity for security too, to
  - Assign security responsibles
  - Select security tools
  - Determine risk levels

BELIEVE IN
ZERO®



Secure Agile process: key take-aways

- Ensure that security-savvy people are involved at important phases:
  - Sprint planning (to enhance/verify requirements)
  - Development (daily follow-up)
  - Review (to support acceptance)
  - Retrospective (to improve dev. Practices for security)

- Different profiles can be distinguished:
  - Security architect
  - Security engineer
  - Risk Manager/Governance

Secure Agile Tool Chain: general principles

- Secure agile is about enabling, rather than controlling
  - Embedding security tools to support development
- Given short sprint cycles, automation is important.

- Good tools:
  - Work continuously (to avoid developers being blocked)
  - Integrate well into developer’s world
  - Avoid causing too much overhead or confusion

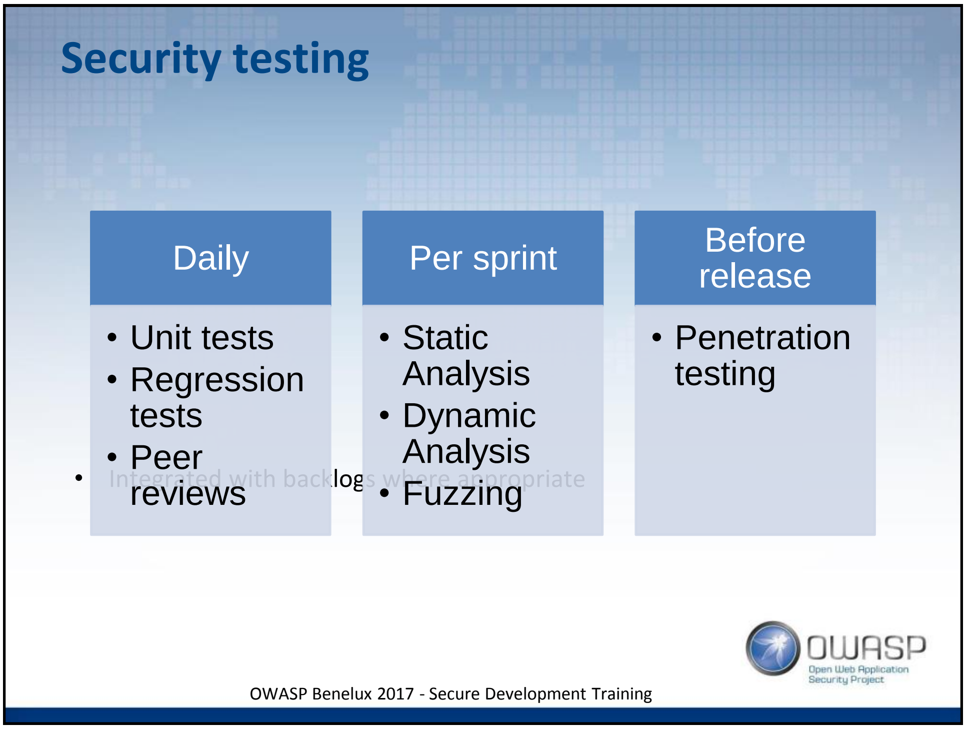
- Evaluate carefully which tools to implement (and which to avoid)

Secure Coding

- Integrate security tools in the development IDE’s:
  - Support for secure coding guidelines
  - Static analysis tools

- Ensure common development environment:
  - Programming run-time
  - Security components (e.g., SSO IdP’s, ...)

- Proper source control and versioning



Security testing

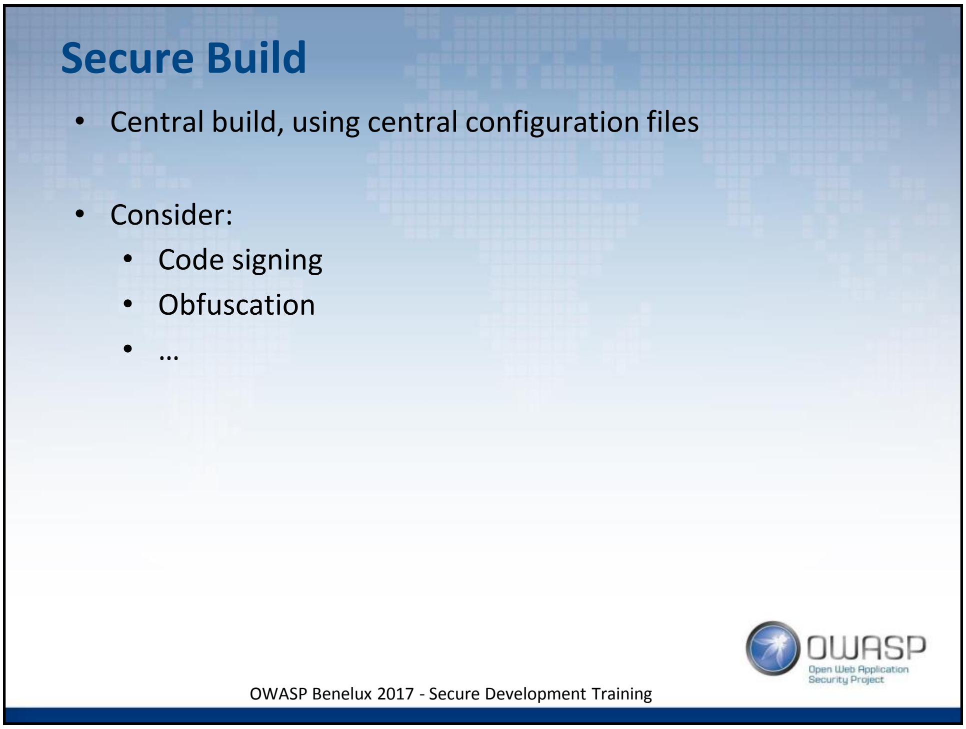
- Integrated with backlogs where appropriate

Daily
- Unit tests
- Regression tests
- Peer reviews

Per sprint
- Static Analysis
- Dynamic Analysis
- Fuzzing

Before release
- Penetration testing

OWASP Benelux 2017 - Secure Development Training



Secure Build

- Central build, using central configuration files
- Consider:
  - Code signing
  - Obfuscation
  - …

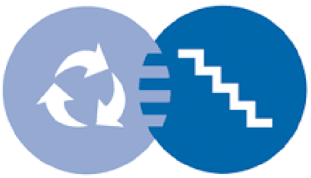
OWASP Benelux 2017 - Secure Development Training

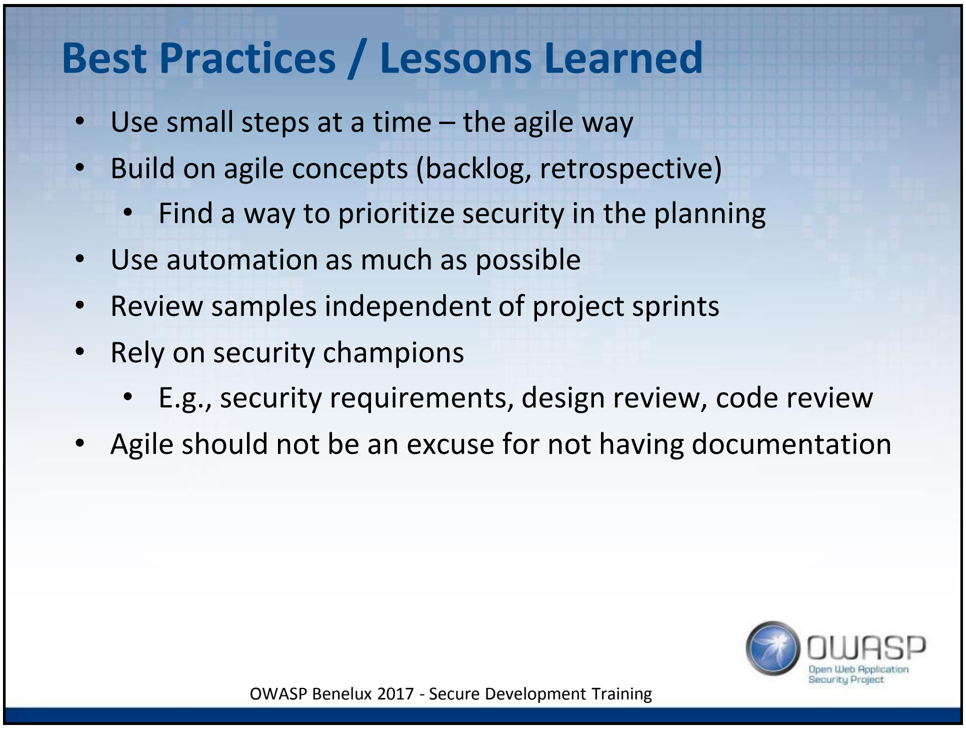
Secure Deploy / DevOps

- Automated deploy, using central configuration files
- Consider:
  - Random key generation
  - Appropriate key/certificate protection (config files, key stores, …)
  - Proper hardening of application servers
  - Security appliance configuration (e.g., WAF)
  - Security monitoring
  - …

Hybrid models

- Many companies are combining waterfall and agile
  - Studies indicate better resulting quality
- For security, easier to hook into
  - E.g., full architecture cycle

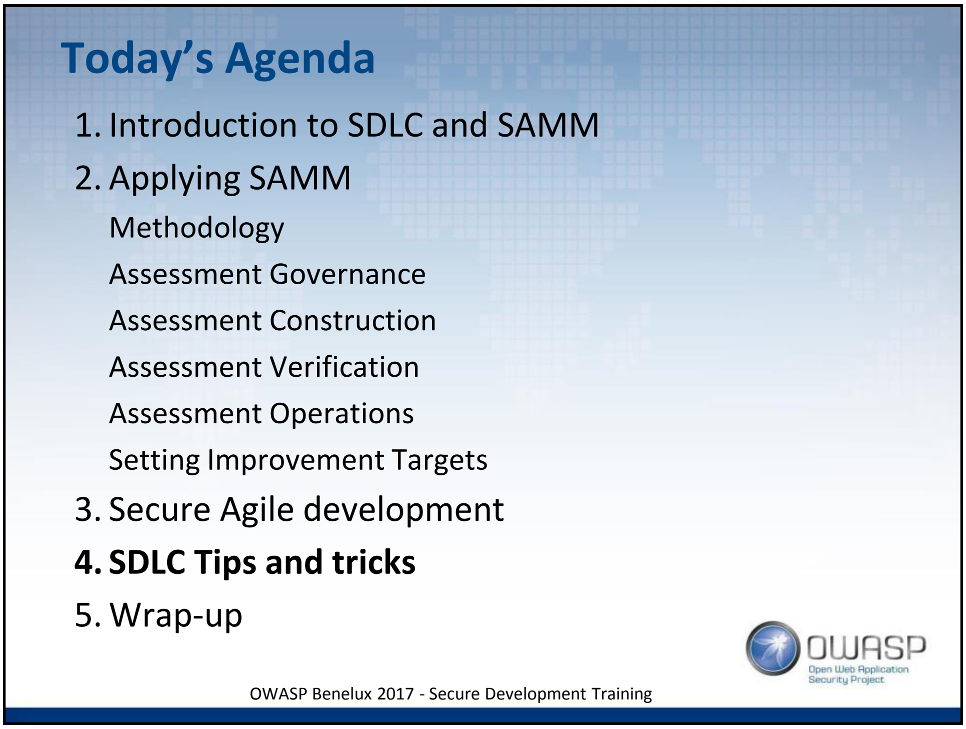




Best Practices / Lessons Learned

- Use small steps at a time – the agile way
- Build on agile concepts (backlog, retrospective)
  - Find a way to prioritize security in the planning
- Use automation as much as possible
- Review samples independent of project sprints
- Rely on security champions
  - E.g., security requirements, design review, code review
- Agile should not be an excuse for not having documentation

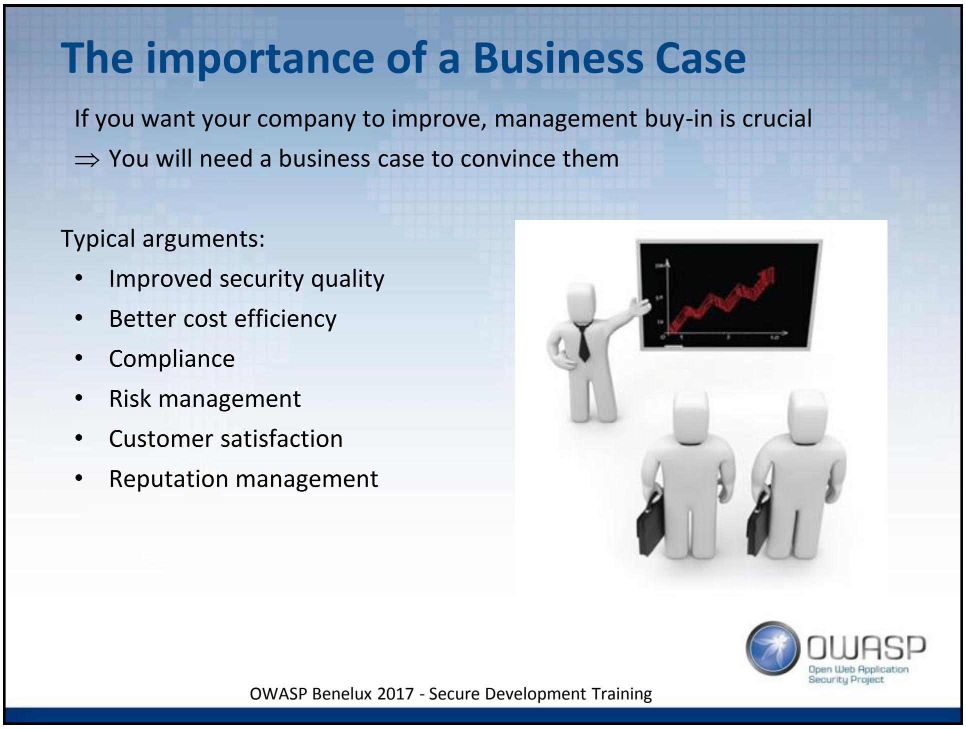
OWASP Benelux 2017 - Secure Development Training



Today's Agenda

1. Introduction to SDLC and SAMM
2. Applying SAMM
   Methodology
   Assessment Governance
   Assessment Construction
   Assessment Verification
   Assessment Operations
   Setting Improvement Targets
3. Secure Agile development
4. **SDLC Tips and tricks**
5. Wrap-up

OWASP Benelux 2017 - Secure Development Training



The importance of a Business Case

If you want your company to improve, management buy-in is crucial

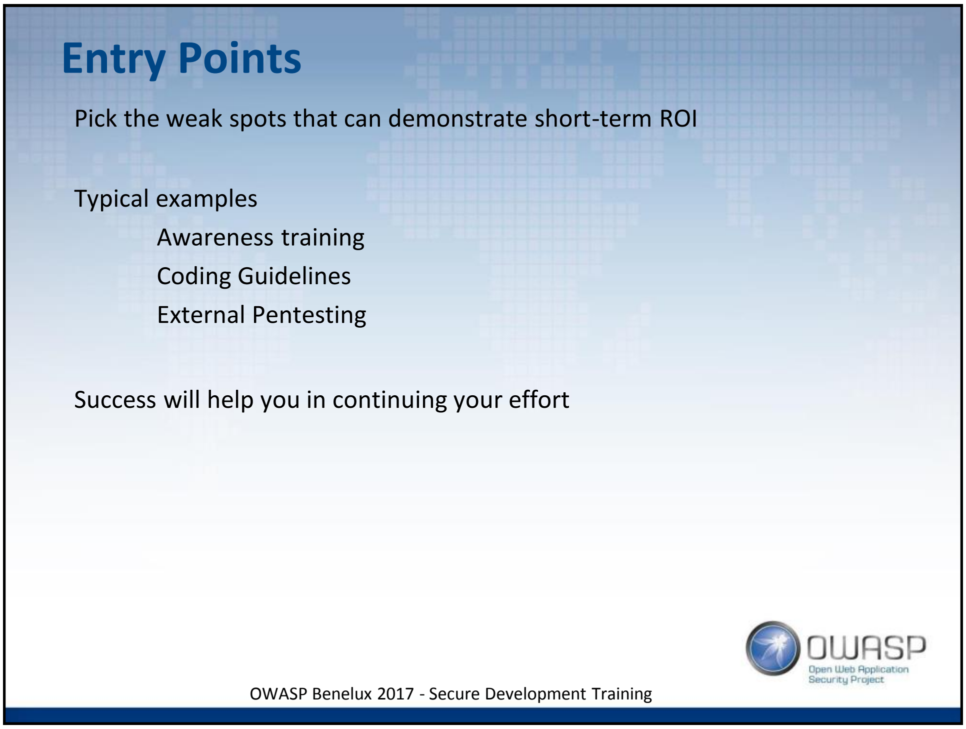
⇒ You will need a business case to convince them

Typical arguments:

- Improved security quality
- Better cost efficiency
- Compliance
- Risk management
- Customer satisfaction
- Reputation management

OWASP Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



Entry Points

Pick the weak spots that can demonstrate short-term ROI

Typical examples

Awareness training

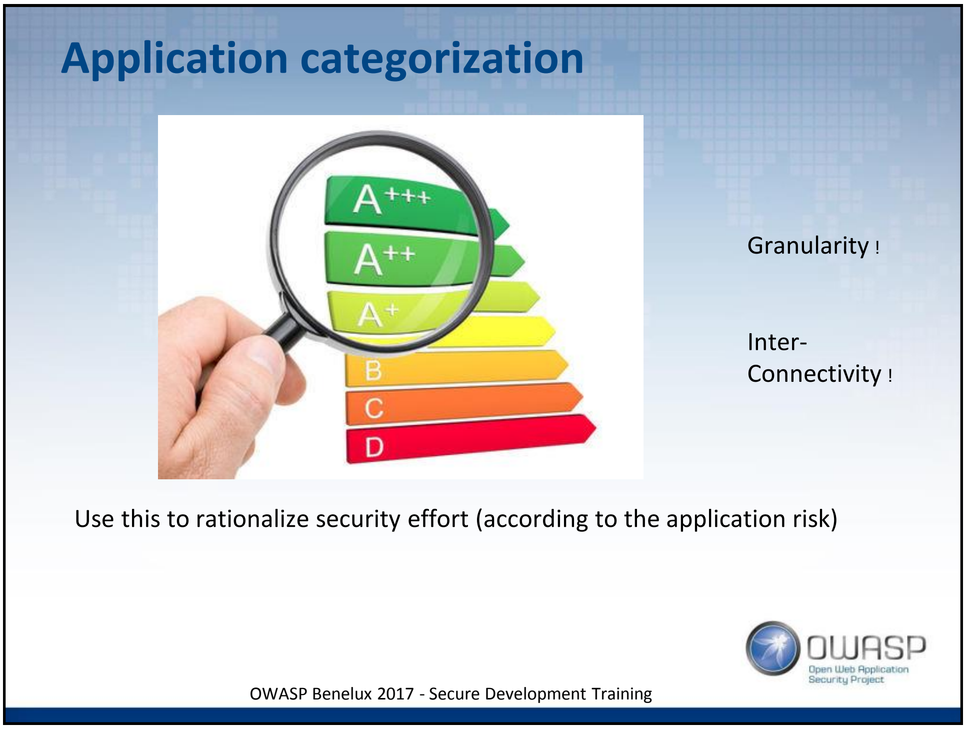
Coding Guidelines

External Pentesting

Success will help you in continuing your effort

OWASP Open Web Application Security Project

OWASP Benelux 2017 - Secure Development Training



Application categorization

Granularity !

Inter-Connectivity !

Use this to rationalize security effort (according to the application risk)

OWASP Benelux 2017 - Secure Development Training

Communication & Support

Critical success factor !

Spreading the message – broad audience

Setup a secure applications portal !

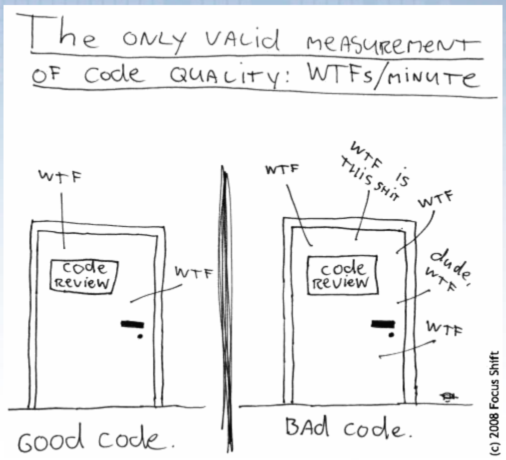
Regular status updates towards management

OWASP Benelux 2017 - Secure Development Training

Monitoring & Metrics

Project vs. Enterprise dashboard

Manual vs. Automated data collection



Responsibilties

Core Security team

- Support vs. Responsible role

Security Satellite

- Analysts
- Architects
- Developers
- Operations
- Management

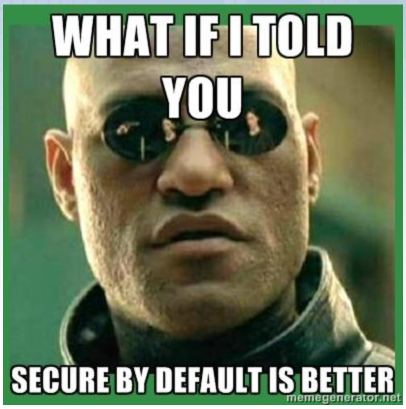
Formalized RACI will be a challenge

The Power of Default Security

Construct development frameworks that are secure by default

Minimizes work for developers

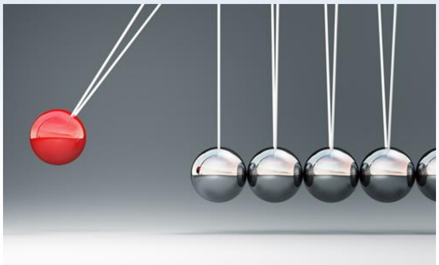
Will lower number of vulns.



SDLC impact

Difficult to predict, but:

- Projects are estimated to increase with 5 – 15% for security
- ROI is achievable taking maintenance and incident management into account
- SDLC capability costs approx. 1 FTE/100 developers



Today's Agenda

1. Introduction to SDLC and SAMM
2. Applying SAMM
   - Methodology
   - Assessment Governance
   - Assessment Construction
   - Assessment Verification
   - Assessment Operations
   - Setting Improvement Targets
3. Secure Agile development
4. SDLC Tips and tricks
5. **Wrap-up**

OWASP Benelux 2017 - Secure Development Training

Conclusions

Developing secure software gets more and more complex

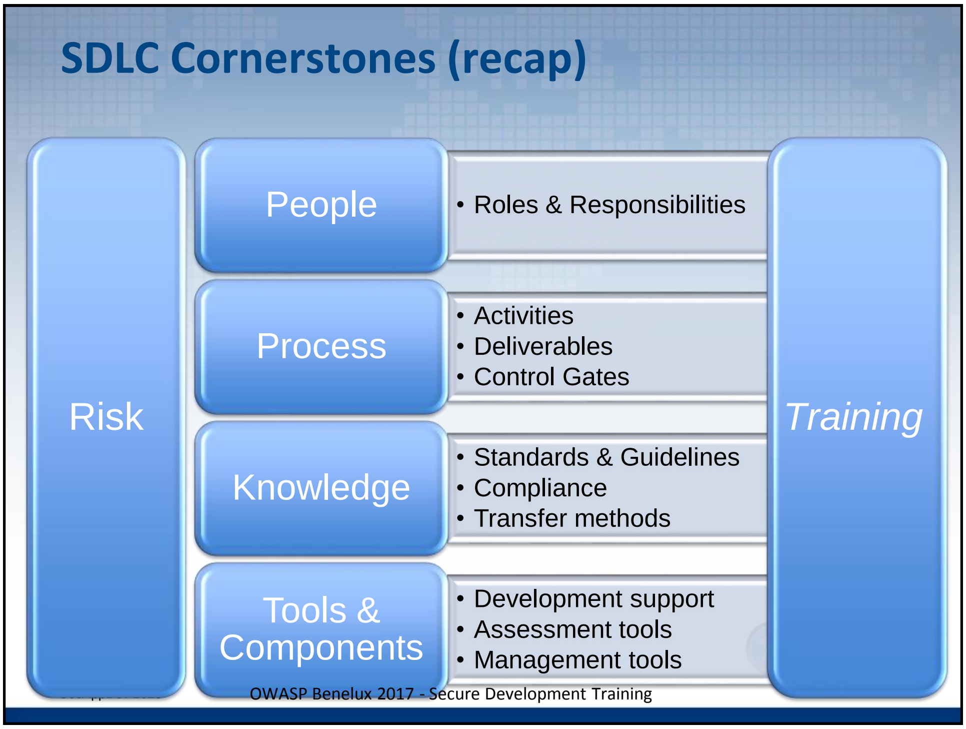
SAMM = global maturity foundation for software assurance

Applying SAMM =

- Assessment
- Roadmap
- (Continuous) Implementation

Be ready to face the organisational challenges that will pop up during the journey

OWASP Benelux 2017 - Secure Development Training



SDLC Cornerstones (recap)

Risk

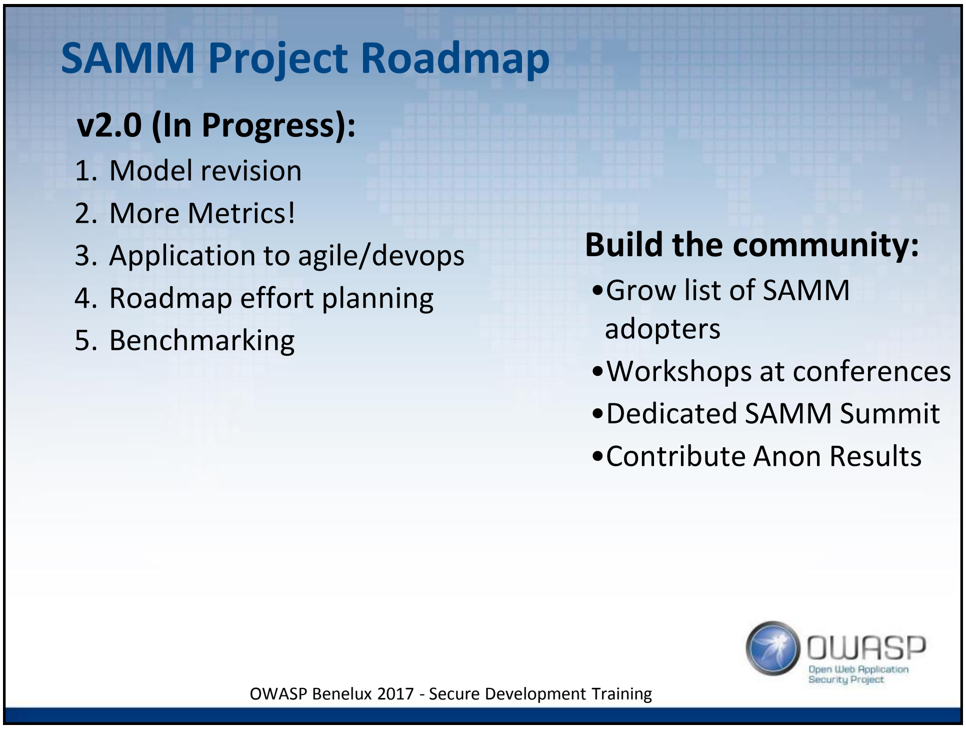
People
- Roles & Responsibilities

Process
- Activities
- Deliverables
- Control Gates

Knowledge
- Standards & Guidelines
- Compliance
- Transfer methods

Tools & Components
- Development support
- Assessment tools
- Management tools

Training



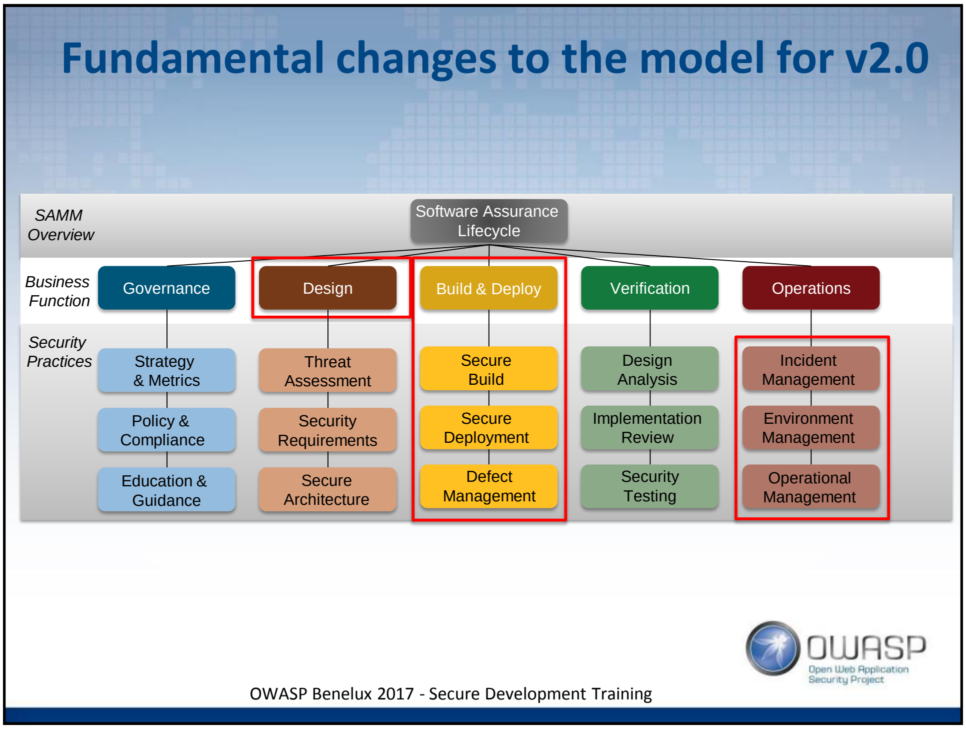
SAMM Project Roadmap

v2.0 (In Progress):

1. Model revision
2. More Metrics!
3. Application to agile/devops
4. Roadmap effort planning
5. Benchmarking

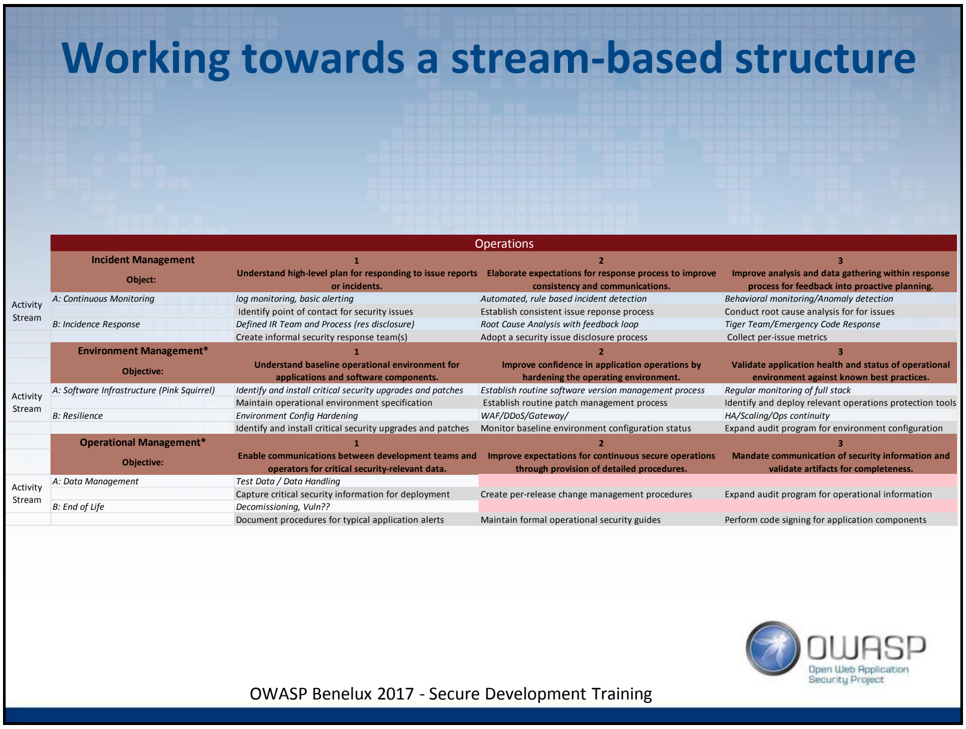
Build the community:

- Grow list of SAMM adopters
- Workshops at conferences
- Dedicated SAMM Summit
- Contribute Anon Results



Fundamental changes to the model for v2.0

OWASP Benelux 2017 - Secure Development Training



Working towards a stream-based structure

	Operations			
	Incident Management **Object:**	**1** **Understand high-level plan for responding to issue reports or incidents.**	**2** **Elaborate expectations for response process to improve consistency and communications.**	**3** **Improve analysis and data gathering within response process for feedback into proactive planning.**
Activity Stream	*A: Continuous Monitoring*	*log monitoring, basic alerting*	*Automated, rule based incident detection*	*Behavioral monitoring/Anomaly detection*
		Identify point of contact for security issues	Establish consistent issue reponse process	Conduct root cause analysis for for issues
	B: Incidence Response	*Defined IR Team and Process (res disclosure)*	*Root Cause Analysis with feedback loop*	*Tiger Team/Emergency Code Response*
		Create informal security response team(s)	Adopt a security issue disclosure process	Collect per-issue metrics
	Environment Management* **Objective:**	**1** **Understand baseline operational environment for applications and software components.**	**2** **Improve confidence in application operations by hardening the operating environment.**	**3** **Validate application health and status of operational environment against known best practices.**
Activity Stream	*A: Software Infrastructure (Pink Squirrel)*	*Identify and install critical security upgrades and patches*	*Establish routine software version management process*	*Regular monitoring of full stack*
		Maintain operational environment specification	Establish routine patch management process	Identify and deploy relevant operations protection tools
	B: Resilience	*Environment Config Hardening*	*WAF/DDoS/Gateway/*	*HA/Scaling/Ops continuity*
		Identify and install critical security upgrades and patches	Monitor baseline environment configuration status	Expand audit program for environment configuration
	Operational Management* **Objective:**	**1** **Enable communications between development teams and operators for critical security-relevant data.**	**2** **Improve expectations for continuous secure operations through provision of detailed procedures.**	**3** **Mandate communication of security information and validate artifacts for completeness.**
Activity Stream	*A: Data Management*	*Test Data / Data Handling*		
		Capture critical security information for deployment	Create per-release change management procedures	Expand audit program for operational information
	B: End of Life	*Decomissioning, Vuln??*		
		Document procedures for typical application alerts	Maintain formal operational security guides	Perform code signing for application components

OWASP Benelux 2017 - Secure Development Training

The end

OWASP Benelux 2017 - Secure Development Training