Building Security In Maturity Model: A Review of Successful Software Security Programs

Gabriele Giuseppini

Technical Manager
Cigital, Inc.

OWASP-Italy Day IV
Milan
6th, November 2009

The OWASP Foundation
http://www.owasp.org

Agenda

- Why BSIMM?
- How We Built BSIMM
- BSIMM as a Tool
- Some Findings: USA
- Some Findings: EU
- Conclusions

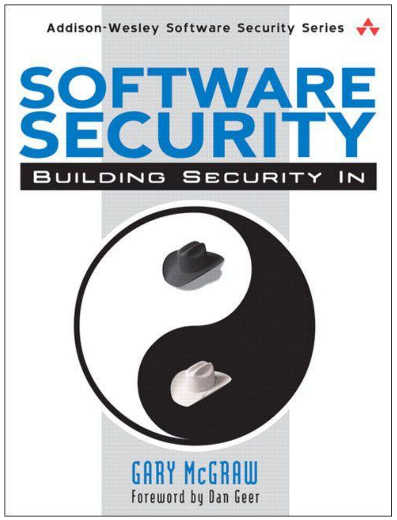
Cigital

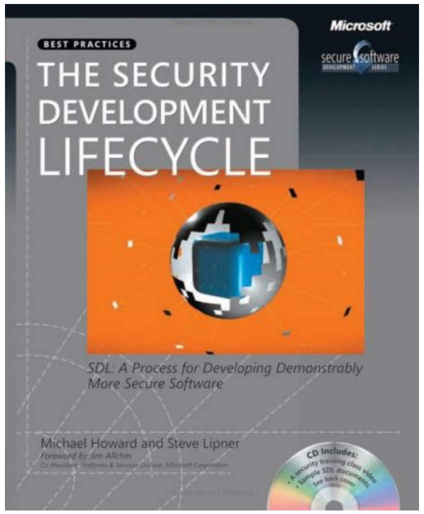
- Founded in 1992 to provide software security and software quality professional services
- Recognized experts in software security and software quality
  - Widely published in books, white papers, articles
  - Industry thought leaders

Why BSIMM?

A Shift from Philosophy to How-To

- Integrating best practices into large organizations
  - Microsoft's SDL
  - Cigital's touchpoints
  - OWASP adopts CLASP




Microsoft
secure software
BEST PRACTICES
THE SECURITY DEVELOPMENT LIFECYCLE
SDL: A Process for Developing Demonstrably More Secure Software
Michael Howard and Steve Lipner
Foreword by Jim Allchin
CD Includes:

More Questions than Answers

- Same activities for all software project?
- What about outsourcing?
- How to handle open source?
- Which vulnerabilities do I have to fix?
- How to get budget / internal support?

- Change focus
  - Old: Perform isolated activities in dev lifecycle only
  - New: Create a software security initiative

Breaking New Ground

- Building Security In Maturity Model
- Real data from real initiatives
- McGraw, Chess, & Migues

cigital

FORTIFY

How We Built BSIMM

Real Data

- Big idea: Build a maturity model from actual data gathered from 9 of 35 known large-scale software security initiatives

EMC² where information lives®

DTCC. The Depository Trust & Clearing Corporation

Adobe

Google™

QUALCOMM®

Microsoft®

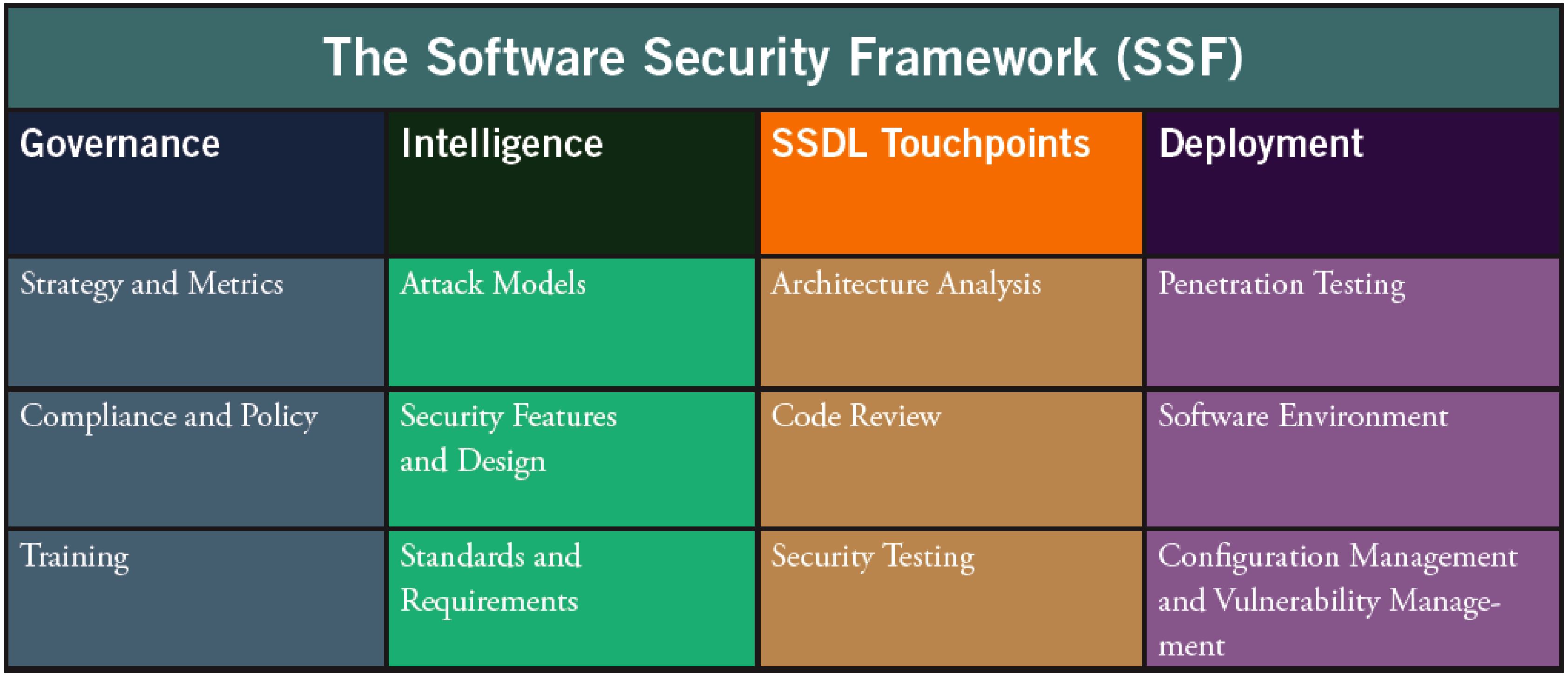
WELLS FARGO

Two more unnamed financial services firms

Building BSIMM

- Create a software security framework
- Nine in-person executive interviews
- Build bullet lists (one per practice)
- Bucketize the lists to identify activities
  - 110 activities supported by real data
- Create levels
  - Three levels of “maturity”

A Software Security Framework

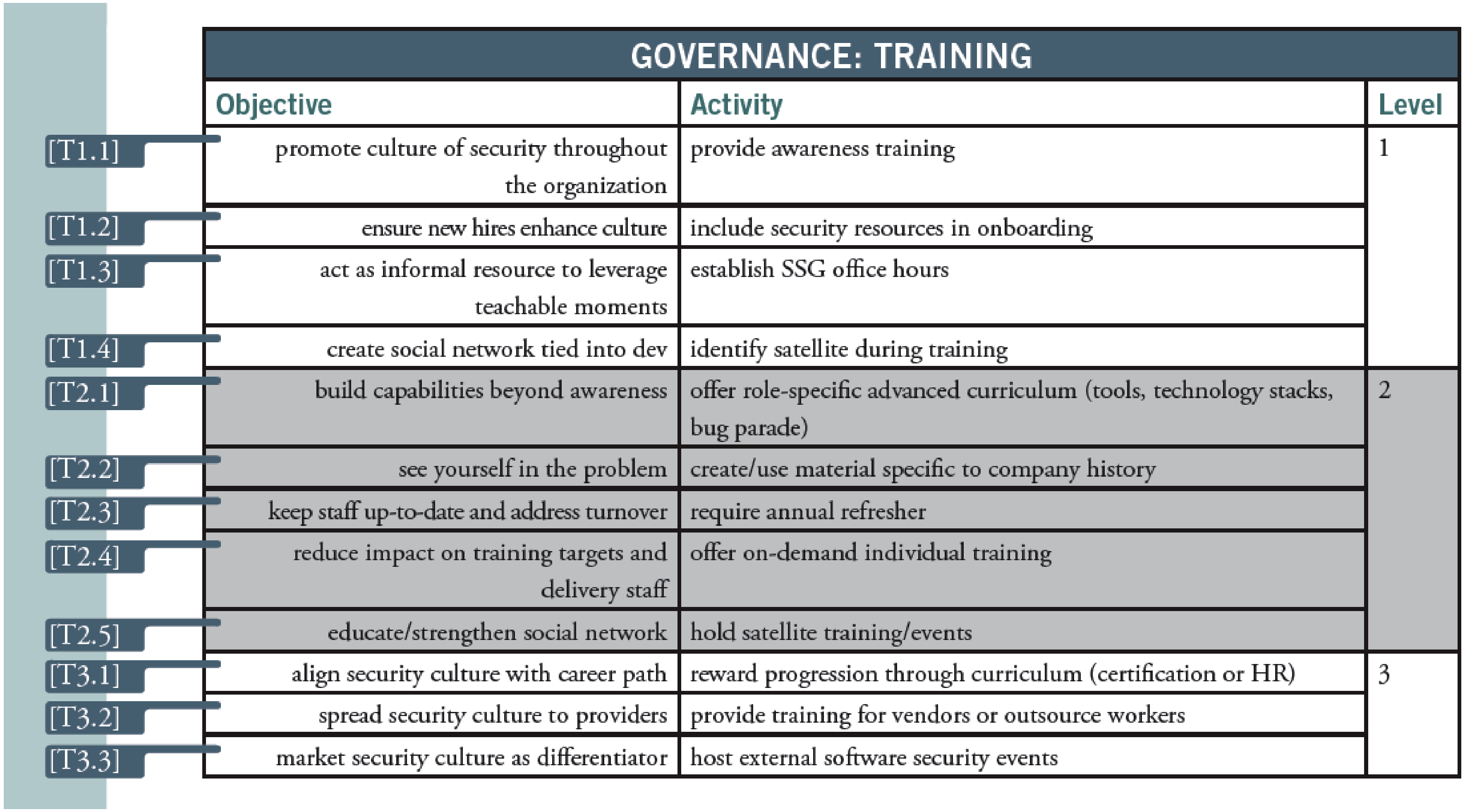


The Software Security Framework (SSF)

Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Manage-ment

- Four domains
- Twelve practices
- An "archeology grid"

Training Practice Skeleton



GOVERNANCE: TRAINING

	Objective	Activity	Level
[T1.1]	promote culture of security throughout the organization	provide awareness training	1
[T1.2]	ensure new hires enhance culture	include security resources in onboarding	
[T1.3]	act as informal resource to leverage teachable moments	establish SSG office hours	
[T1.4]	create social network tied into dev	identify satellite during training	
[T2.1]	build capabilities beyond awareness	offer role-specific advanced curriculum (tools, technology stacks, bug parade)	2
[T2.2]	see yourself in the problem	create/use material specific to company history	
[T2.3]	keep staff up-to-date and address turnover	require annual refresher	
[T2.4]	reduce impact on training targets and delivery staff	offer on-demand individual training	
[T2.5]	educate/strengthen social network	hold satellite training/events	
[T3.1]	align security culture with career path	reward progression through curriculum (certification or HR)	3
[T3.2]	spread security culture to providers	provide training for vendors or outsource workers	
[T3.3]	market security culture as differentiator	host external software security events	

Example Activity

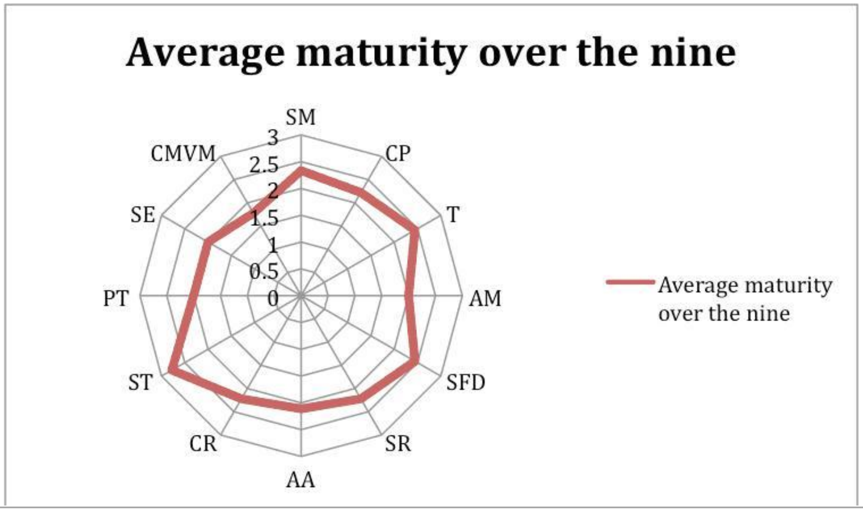
[T1.3] **Establish SSG office hours.** The SSG offers help to any and all comers during an advertised lab period or regularly scheduled office hours. By acting as an informal resource for people who want to solve security problems, the SSG leverages teachable moments and emphasizes the carrot over the stick. Office hours might be held one afternoon per week in the office of a senior SSG member.

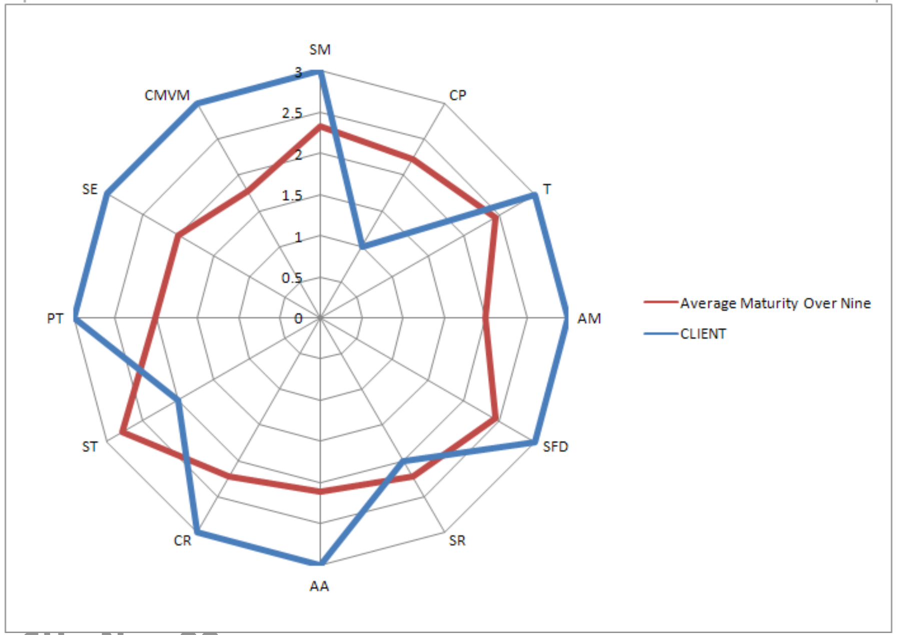
Monkeys Eat Bananas

- BSIMM is not about good or bad ways to eat bananas or banana best practices
- BSIMM is about observations

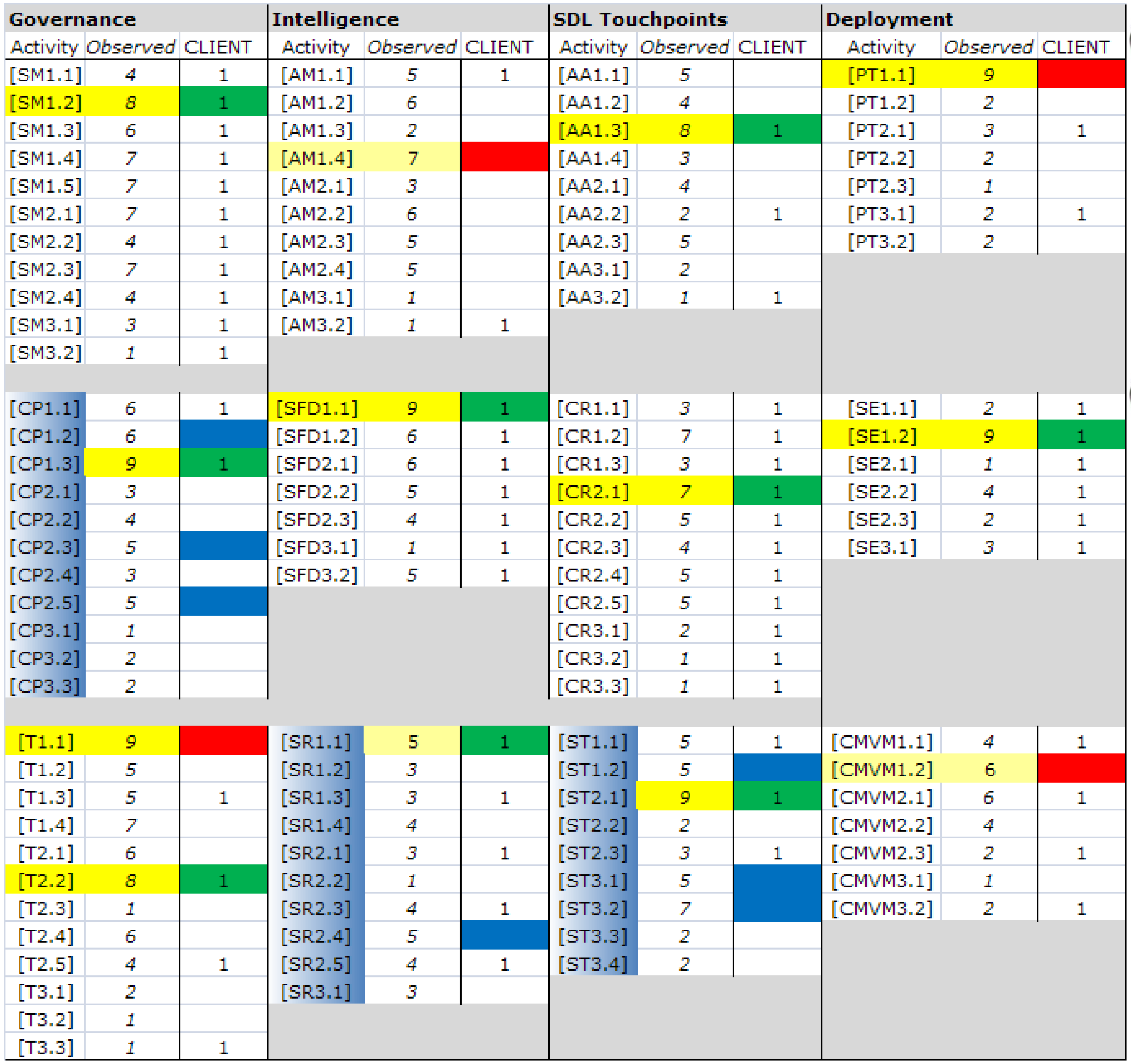
BSIMM as a Tool

Maturity Yardstick





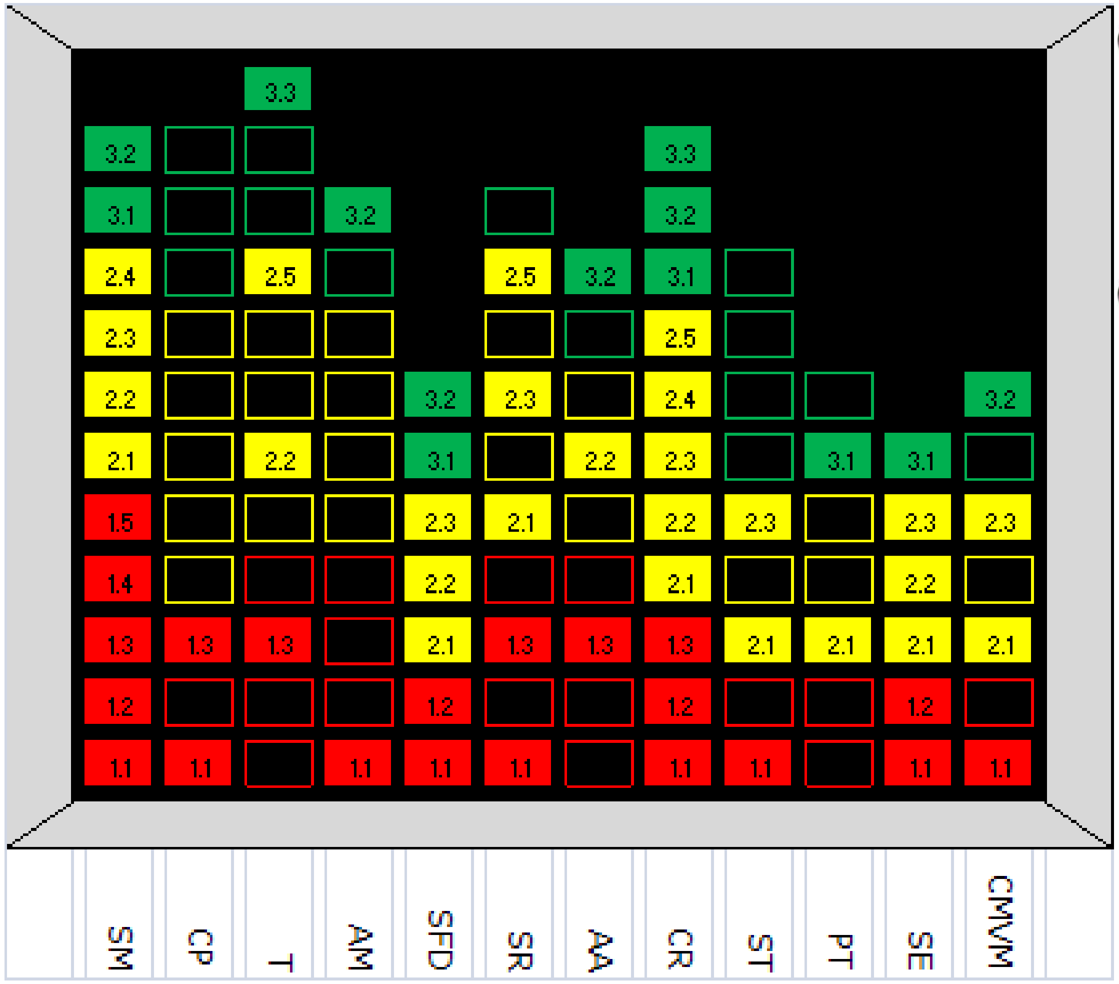
BSIMM Scorecard



Governance			Intelligence			SDL Touchpoints			Deployment		
Activity	*Observed*	CLIENT	Activity	*Observed*	CLIENT	Activity	*Observed*	CLIENT	Activity	*Observed*	CLIENT
[SM1.1]	*4*	1	[AM1.1]	*5*	1	[AA1.1]	*5*		[PT1.1]	*9*	🟥
[SM1.2]	*8*	🟩 1	[AM1.2]	*6*		[AA1.2]	*4*		[PT1.2]	*2*	
[SM1.3]	*6*	1	[AM1.3]	*2*		[AA1.3]	*8*	🟩 1	[PT2.1]	*3*	1
[SM1.4]	*7*	1	[AM1.4]	*7*	🟥	[AA1.4]	*3*		[PT2.2]	*2*	
[SM1.5]	*7*	1	[AM2.1]	*3*		[AA2.1]	*4*		[PT2.3]	*1*	
[SM2.1]	*7*	1	[AM2.2]	*6*		[AA2.2]	*2*	1	[PT3.1]	*2*	1
[SM2.2]	*4*	1	[AM2.3]	*5*		[AA2.3]	*5*		[PT3.2]	*2*	
[SM2.3]	*7*	1	[AM2.4]	*5*		[AA3.1]	*2*				
[SM2.4]	*4*	1	[AM3.1]	*1*		[AA3.2]	*1*	1			
[SM3.1]	*3*	1	[AM3.2]	*1*	1						
[SM3.2]	*1*	1									
[CP1.1]	*6*	1	[SFD1.1]	*9*	🟩 1	[CR1.1]	*3*	1	[SE1.1]	*2*	1
[CP1.2]	*6*	🟦	[SFD1.2]	*6*	1	[CR1.2]	*7*	1	[SE1.2]	*9*	🟩 1
[CP1.3]	*9*	🟩 1	[SFD2.1]	*6*	1	[CR1.3]	*3*	1	[SE2.1]	*1*	1
[CP2.1]	*3*		[SFD2.2]	*5*	1	[CR2.1]	*7*	🟩 1	[SE2.2]	*4*	1
[CP2.2]	*4*		[SFD2.3]	*4*	1	[CR2.2]	*5*	1	[SE2.3]	*2*	1
[CP2.3]	*5*	🟦	[SFD3.1]	*1*	1	[CR2.3]	*4*	1	[SE3.1]	*3*	1
[CP2.4]	*3*		[SFD3.2]	*5*	1	[CR2.4]	*5*	1			
[CP2.5]	*5*	🟦				[CR2.5]	*5*	1			
[CP3.1]	*1*					[CR3.1]	*2*	1			
[CP3.2]	*2*					[CR3.2]	*1*	1			
[CP3.3]	*2*					[CR3.3]	*1*	1			
[T1.1]	*9*	🟥	[SR1.1]	5	🟩 1	[ST1.1]	*5*	1	[CMVM1.1]	*4*	1
[T1.2]	*5*		[SR1.2]	*3*		[ST1.2]	*5*	🟦	[CMVM1.2]	6	🟥
[T1.3]	*5*	1	[SR1.3]	*3*	1	[ST2.1]	*9*	🟩 1	[CMVM2.1]	*6*	1
[T1.4]	*7*		[SR1.4]	*4*		[ST2.2]	*2*		[CMVM2.2]	*4*	
[T2.1]	*6*		[SR2.1]	*3*	1	[ST2.3]	*3*	1	[CMVM2.3]	*2*	1
[T2.2]	*8*	🟩 1	[SR2.2]	*1*		[ST3.1]	*5*	🟦	[CMVM3.1]	*1*	
[T2.3]	*1*		[SR2.3]	*4*	1	[ST3.2]	*7*	🟦	[CMVM3.2]	*2*	1
[T2.4]	*6*		[SR2.4]	*5*	🟦	[ST3.3]	*2*				
[T2.5]	*4*	1	[SR2.5]	*4*	1	[ST3.4]	*2*				
[T3.1]	*2*		[SR3.1]	*3*							
[T3.2]	*1*										
[T3.3]	*1*	1									

- Top 10 things
  - Green = good?
  - Red = bad?
- Blue shift = practices to emphasize
  - Activities you should maybe think about in blue

BSIMM Equalizer



- What you do (by level)
- Gaps are apparent and lead to good conversation

Some Findings: USA

Real-world Data: the Nine

- Initiative age: 5yrs 4months avg.
  - Newest: 2.5
  - Oldest: 10
- Satellite size: 79
  - Smallest: 0
  - Largest: 300
  - Median: 20
- SSG size: 41
  - Smallest: 12
  - Largest: 100
  - Median: 35
- Dev size: 7750
  - Smallest: 450
  - Largest: 30,000
  - Median: 5000

Average SSG size: 1% of dev

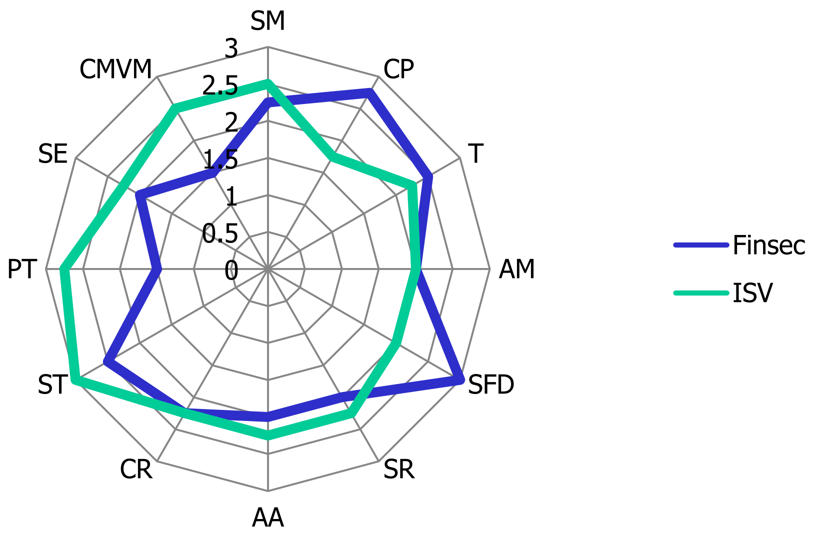
Ten Surprising Things

1. Bad metrics hurt
2. Secure-by default frameworks
3. Nobody uses WAFs
4. QA can't do software security
5. Evangelize over audit
6. ARA is hard
7. Practitioners don't talk attacks
8. Training is advanced
9. Pen testing is diminishing
10. Fuzz testing

Ten Things Everybody Does

- Evangelist role
- Create policy
- Awareness training
- History in training
- Security features
- SSG does ARA
- Code review tools
- Black box tools
- External pen testing
- Good network security

Everyone Is a Special Snowflake (Not)



- ISV results are similar to financial services
- Industry vertical has less impact that originally thought

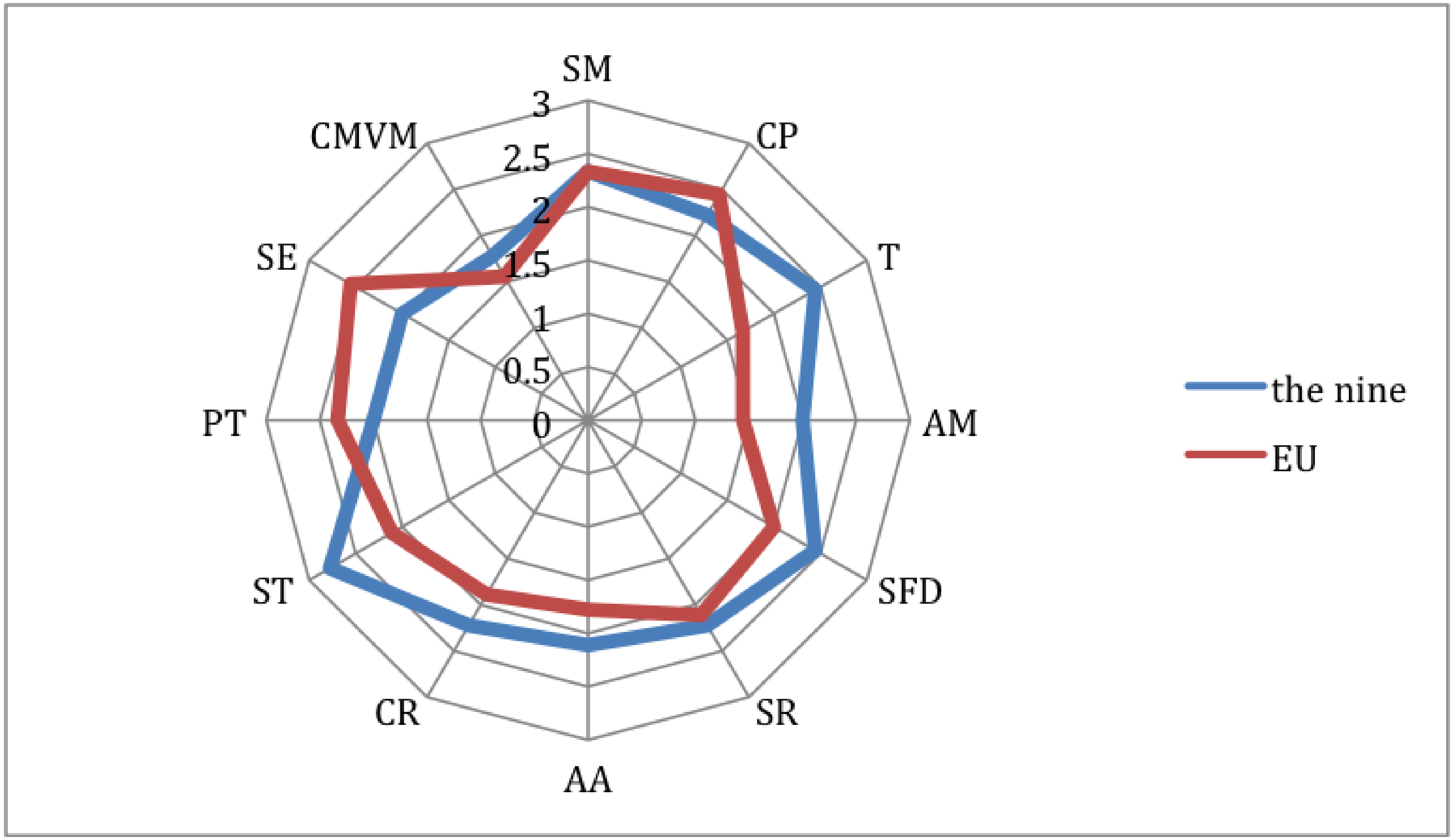
Some Findings: EU

Real-world Data: the Nine EU

- Initiative age: 6yrs 8months avg. (5y4m)
  - Newest: 1.5 (2.5)
  - Oldest: 14 (10)
- SSG size: 16 (41)
  - Smallest: 1 (12)
  - Largest: 50 (100)
  - Median: 10 (35)
- Satellite size: 45 (79)
  - Smallest: 0 (0)
  - Largest: 140 (300)
  - Median: 0 (20)
- Dev size: 4807 (7750)
  - Smallest: 400 (450)
  - Largest: 12,000 (30,000)
  - Median: 5,000 (5,000)

Average SSG size: 0.72% of dev (1%)

Maturity in the EU



Eleven Things Everybody Does in the EU

- **Publish process**
- **Identify gates**
- **Require sign-off**
- **Promote privacy**
- Create policy
- Security features
- **Security standards**
- **Review SF’s**
- External pen testing
- **Fix PT findings**
- Good network security

Fourteen Things Nobody Does in the EU

- Feedback to policy (CP)
- Require refresher (T)
- Forum for attacks (AM)
- Research new attacks (AM)
- Arm testers (AM)
- Code review labs (CR)
- Build a factory (CR)
- Eradicate bugs (CR)
- Share with QA (ST)
- Security in QA suites (ST)
- AA drives tests (ST)
- CC drives tests (ST)
- Learn from ops (CMVM)
- Change from ops (CMVM)

Conclusions

Using BSIMM

- BSIMM released March 2009 under creative commons
  - http://bsi-mm.com
  - Steal the data if you want
- BSIMM is a yardstick
  - Use it to see where you stand
  - Use it to figure out what your peers do
- BSIMM is growing
  - More BSIMM victims (+17 and counting) → BSIMM II
  - BSIMM Europe
  - BSIMM Begin

Where to Learn More

- bsi-mm.com
- www.informIT.com
- gem@cigital.com
- smigues@cigital.com
- chess@fortify.com