University of BRISTOL

CyBOK

Cyber Security Body of Knowledge: Secure Software Lifecycle

bristol.ac.uk

University of BRISTOL

CyBOK

The CyBOK project would like to understand how the CyBOK is being used and its uptake. The project would like organisations using, or intending to use, CyBOK for the purposes of education, training, course development, professional development etc. to contact it at contact@cybok.org to let the project know how they are using CyBOK.

bristol.ac.uk

Security Software Lifecycle

CyBOK

The components of a comprehensive software development process to prevent and detect all classes of security defects and to respond in the event of an exploit.

Agenda

CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommece/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

History (and Current ☹): Penetrate and Patch

- **Security is assessed when "complete"; discovered vulnerabilities then fixed/patched**
  - Costly
  - Attackers can find and exploit vulnerabilities without being noticed
  - (Urgent) Patches can introduce new vulnerabilities
  - Patches go unapplied by customers

- 1998: Gary McGraw DARPA funded project on the application of software engineering to assessment of security vulnerabilities
- McGraw (and with John Viega) advocated for proactive and rigorous software analysis to assess and prevent vulnerabilities
  - Book: Building Secure Software

- 2002: Bill Gates announces the Trustworthy Computing Initiative
- 2004: Turned into a structured process, the SDL (http://microsoft.com/sdl)
  - Evolved to Version 5.2 in 2012, Version 6.0 in 2013
  - Microsoft offers many (free) tools and templates to support SDL

"Trustworthy Computing is the highest priority for all the work we are doing. We must lead the industry to a whole new level of Trustworthiness in computing."

Agenda

CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommece/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

1: Provide Training

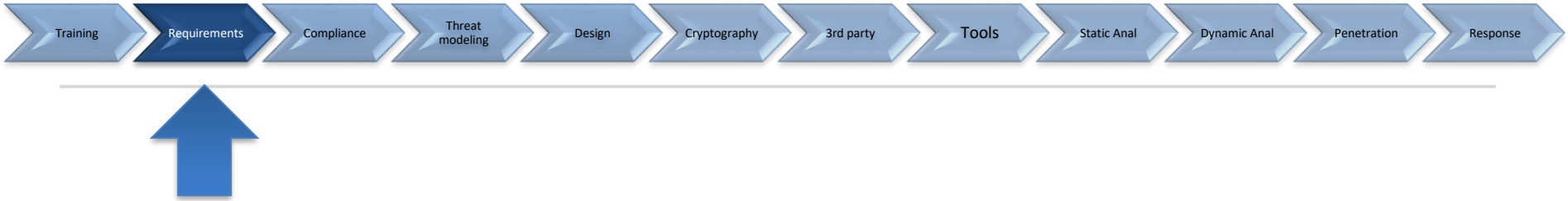
CyBOK

Assess organizational knowledge on security and privacy – establish training program as necessary

- Establish training criteria
  - Content covering secure design, development, test and privacy
  - (e.g. 80% of all technical personnel trained)
- Establish minimum training frequency – attackers are a moving target
  - Employees must attend n classes per year
- Need to cover all the topics in the SDL

2: Security Requirements

CyBOK

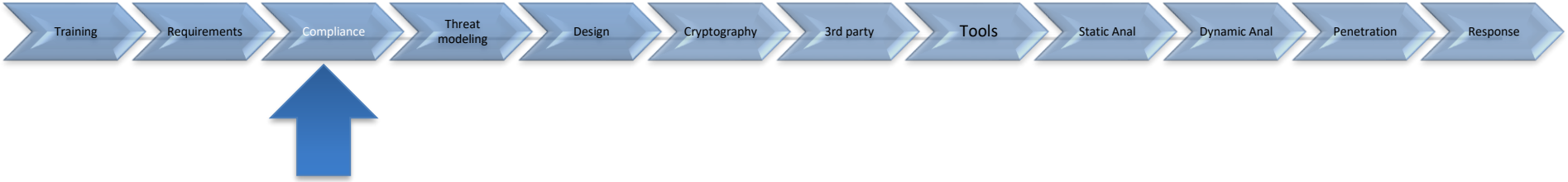


Consider security at the outset of a project

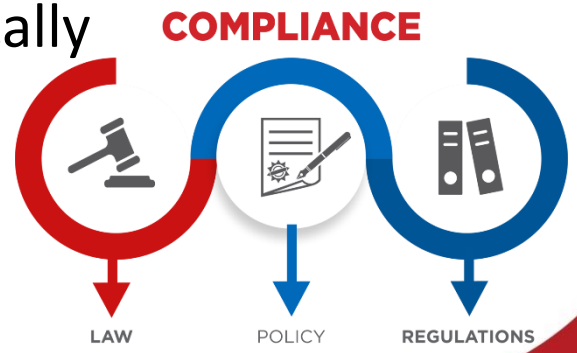
- Development team identifies security requirements
- Factor in security implications of functional requirements, legal and industry, compliance, standards, known threats, previous security incidents.
- Techniques for systematically developing: examples: SQUARE, abuse cases, i*, and KAOS

3: Define Metrics and Compliance Reporting

CyBOK

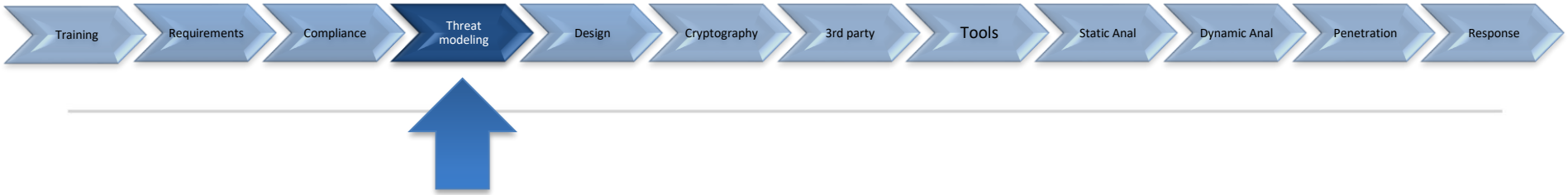


- Define and document a bug bar for security
  - Classification of what are moderate, important, or critical security and privacy bug types
  - User to set priority for fixing and to determine if the product can ship
- Ensure that bug reporting tools can track security and issues and that a database can be queried dynamically for all security bugs
- Understand requirement compliance and associate reporting requirements

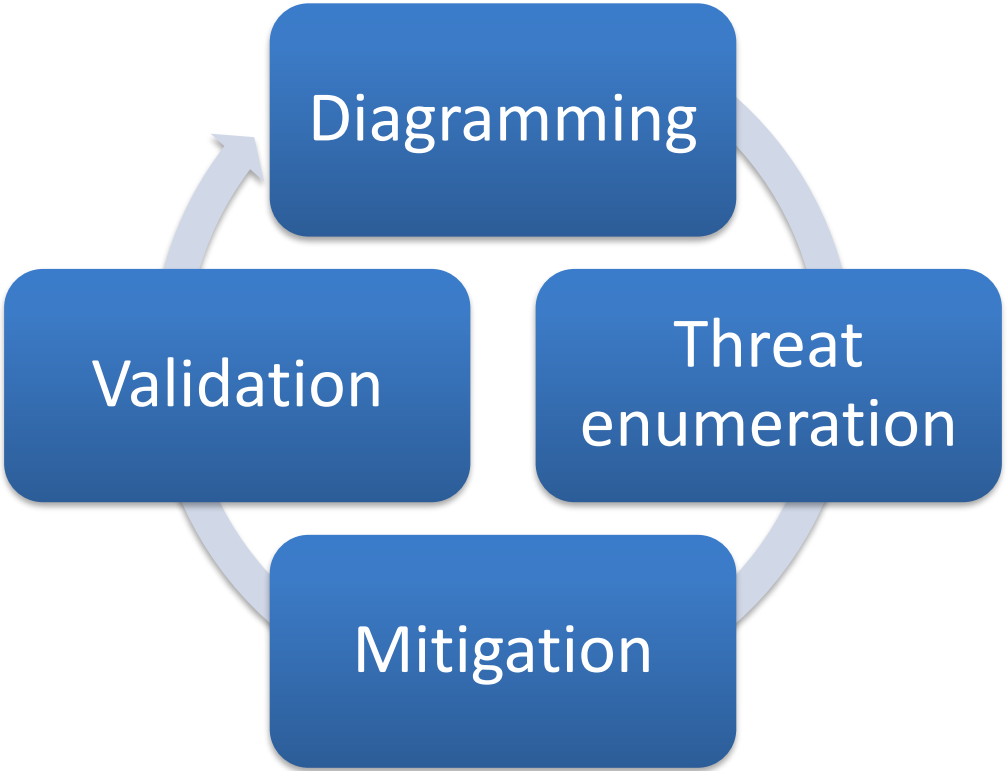


4: Perform Threat Modeling

CyBOK

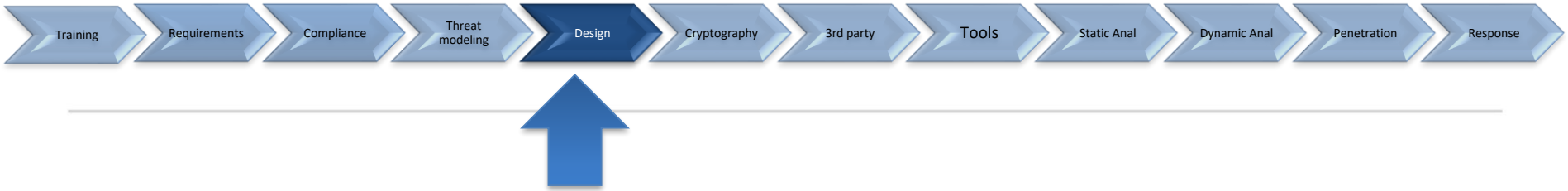


Threat modeling is a process to understand (potential) security threats to a system, determine risks from those threats, and establish appropriate mitigations.



5: Establish design requirements

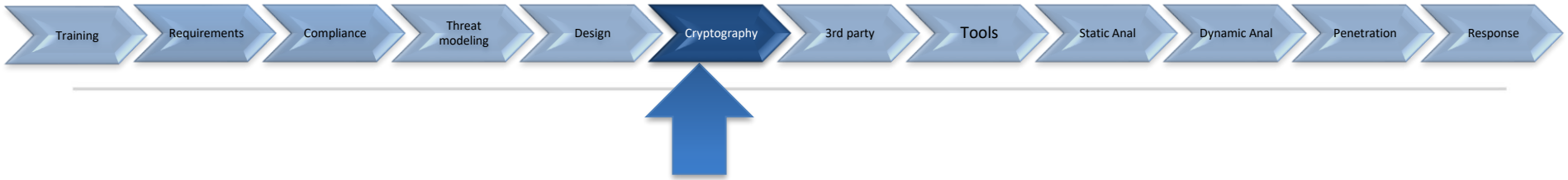
CyBOK



- **Consider design principles, such as:**
  - Economy of mechanism
  - Fail-safe defaults
  - Complete mediation
  - Separation of privilege
  - Least privilege
  - Least common mechanism
  - Secure the weakest link
  - Defense in depth
  - Give or earn but never assume trust
  - Always consider the users

6: Define and use cryptography standards

CyBOK



Through the proper use of cryptography, one can protect the confidentiality of data, protect data from unauthorized modification, and authenticate the source of data.

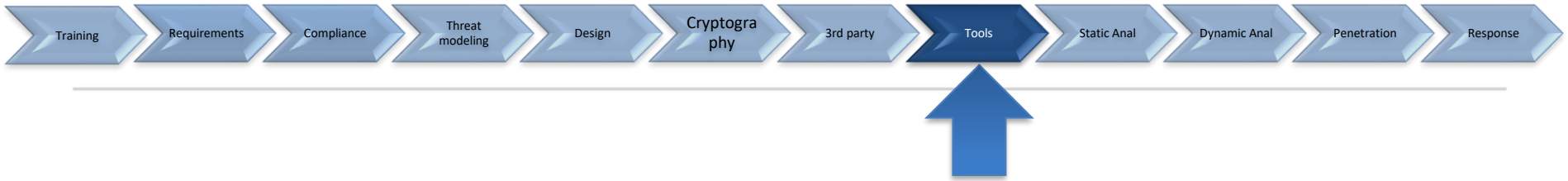
7: Manage the security risk of using 3rd party components

CyBOK

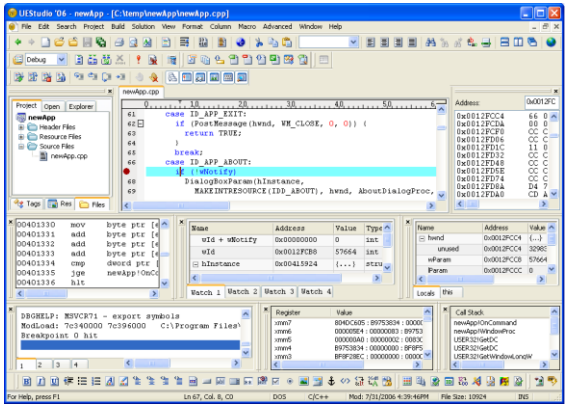
https://snyk.io/stateofossecurity/

8: Use approved tools

CyBOK



- Specification of approved build tools and options
  - Compilers and code generators
  - Static analysis
  - Debuggers
  - Dynamic analysis
  - Test verification
  - IDE



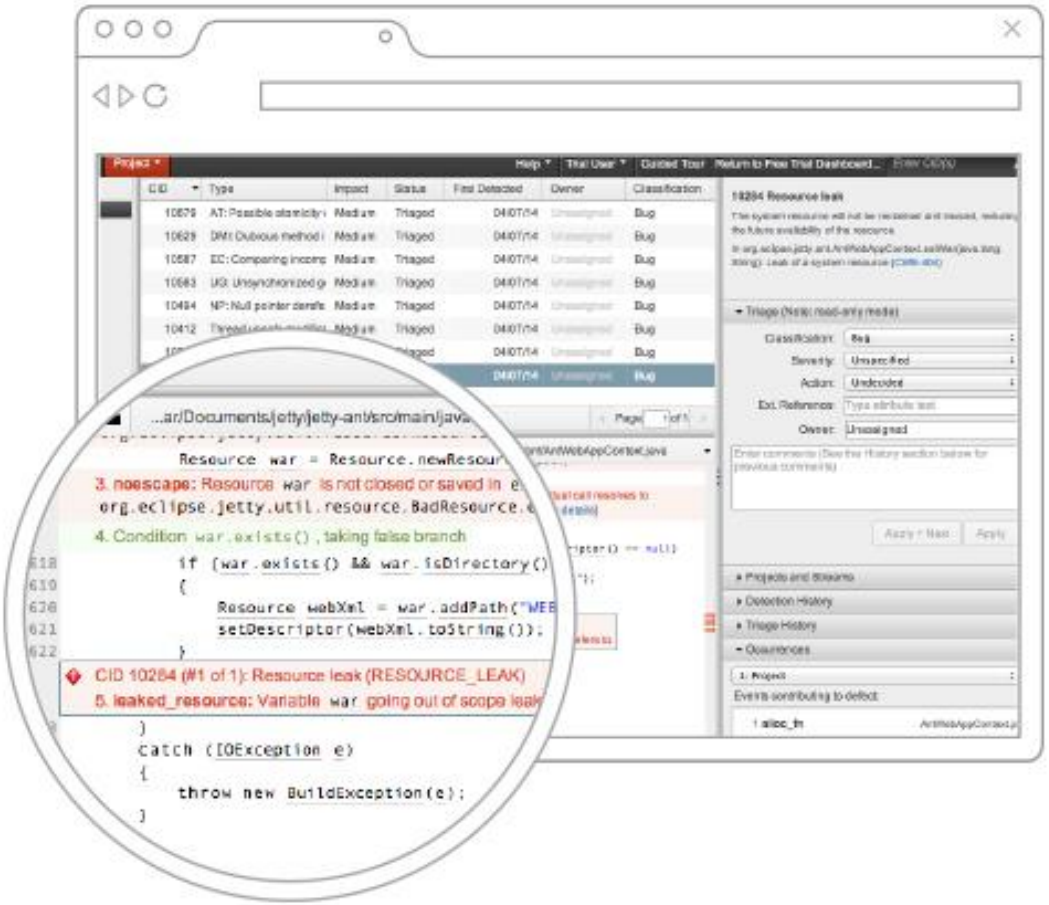
- Decide on settings and ensure settings are correct
- Security isn't static
  - Update your tools because they change with dynamic threat environment

http://www.ultraedit.com/support/tutorials_power_tips/uestudio/integrated_debugger.html

9: Perform static analysis security testing (SAST)

CyBOK

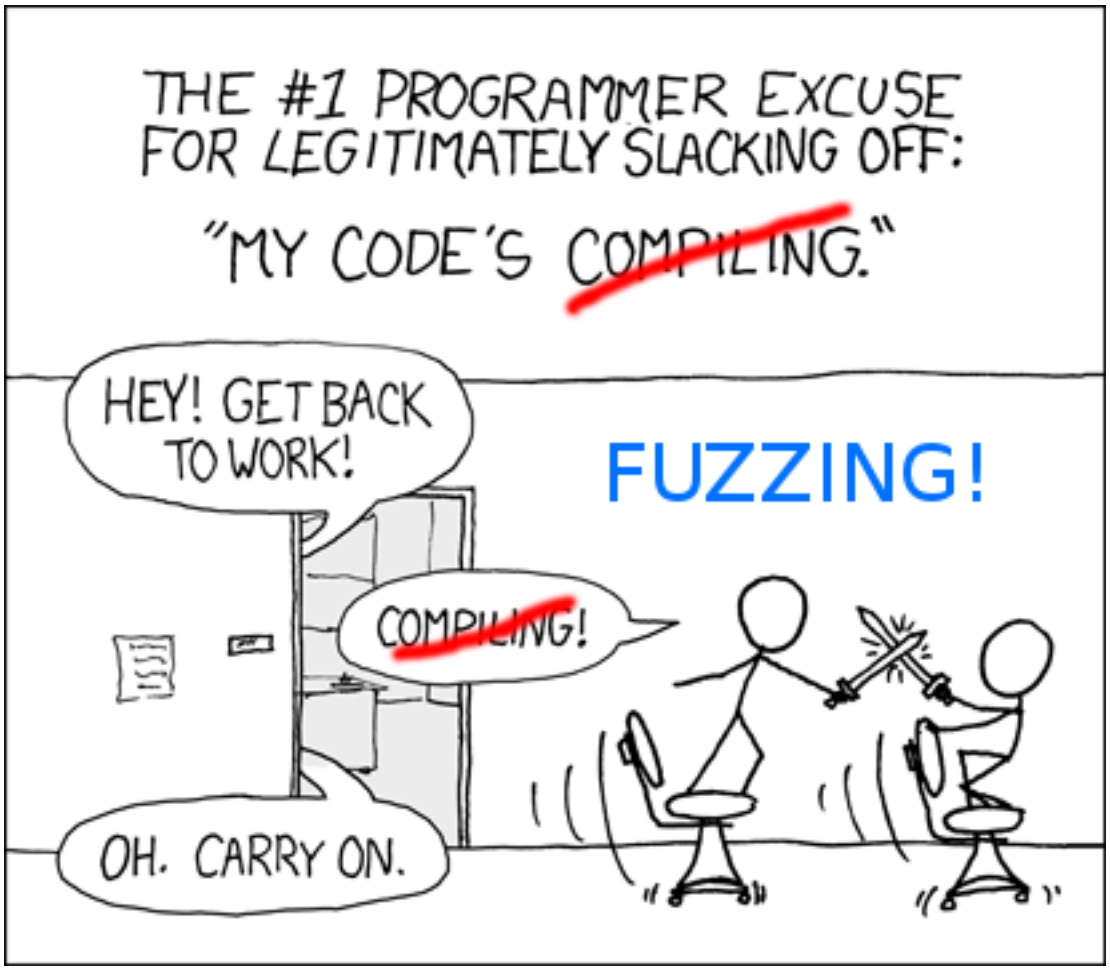
Training | Requirements | Compliance | Threat modeling | Design | Cryptography | 3rd party | Tools | **Static Anal** | Dynamic Anal | Penetration | Response



10: Perform dynamics analysis security testing (DAST)

CyBOK

Training | Requirements | Compliance | Threat modeling | Design | Cryptography | 3rd party | Tools | Static Anal | **Dynamic Anal** | Penetration | Response



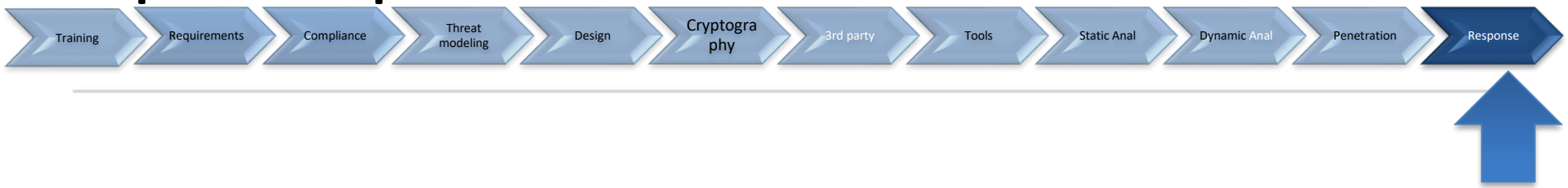
11: Perform penetration testing

CyBOK

Training | Requirements | Compliance | Threat modeling | Design | Cryptography | 3rd party | Tools | Static Anal | Dynamic Anal | **Penetration** | Response

12: Establish a standard incident response practice

CyBOK



Creation of a clearly defined support policy

- Prepare Cyber Security Incident Response Plan (CSIRP)
  - Identify contact for Cyber Security Council and resources to respond to incidents
  - 24x7x365 contact information for engineering, marketing and management individuals with decision-making authority
- Ensure ability to service all code “emergency” releases and all licensed 3rd party code.

Agenda

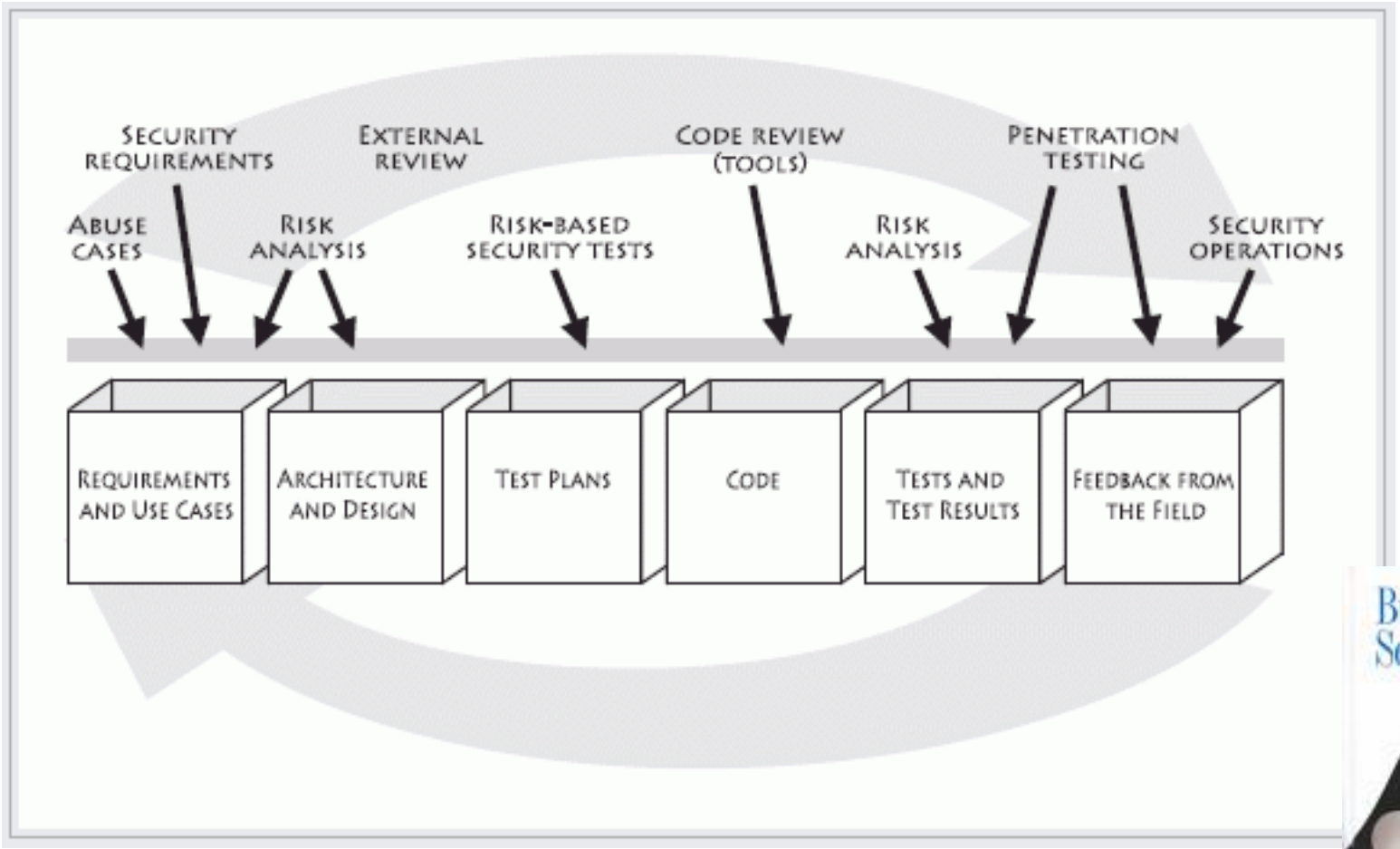
CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommece/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

Touchpoints

CyBOK

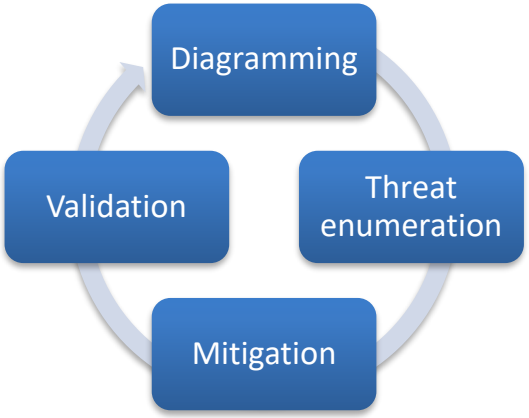
Code review (tools) → Architecture risk → Penetration Testing → Risk-based testing → Abuse cases → Security requirements → Security ops



4. Risk-based security testing

CyBOK

Code review (tools) → Architecture risk → Penetration Testing → **Risk-based testing** → Abuse cases → Security requirements → Security ops



5. Abuse cases

CyBOK

Code review (tools) | Architecture risk | Penetration Testing | Risk-based testing | **Abuse cases** | Security requirements | Security ops

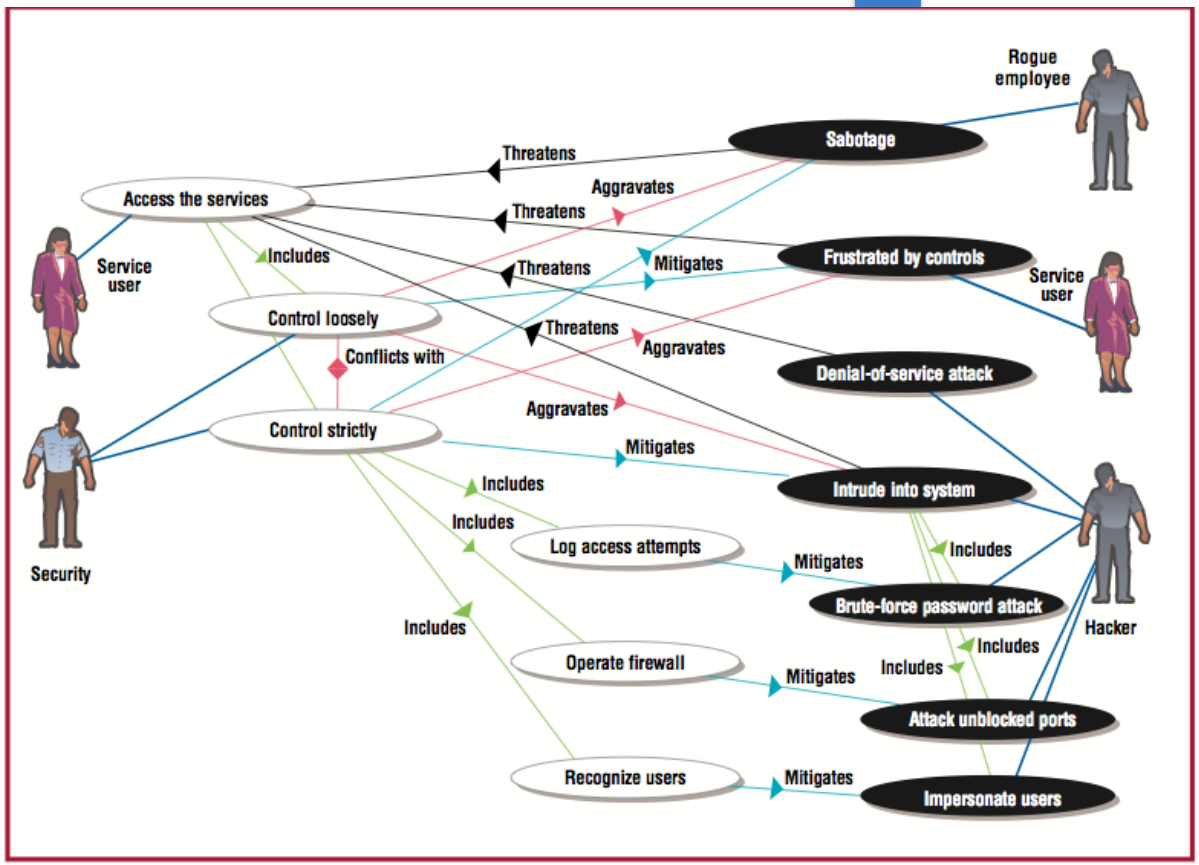


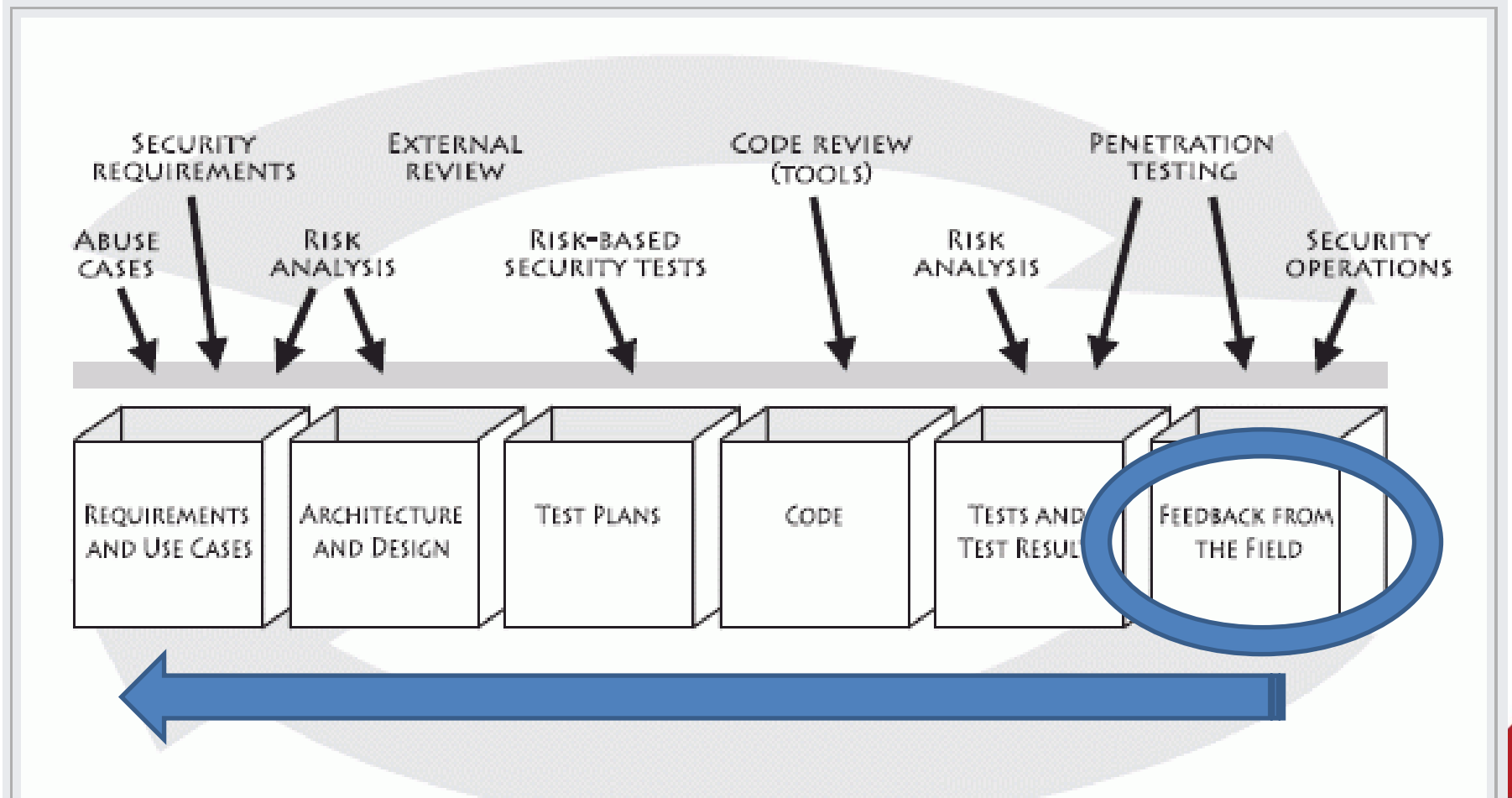
Figure 4. Use and misuse cases for Web portal security.

I. Alexander, *Misuse Cases: Use Cases with Hostile Intent*, IEEE Software, Jan/Feb 2003.

7. Security operations

CyBOK

Code review (tools) | Architecture risk | Penetration Testing | Risk-based testing | Abuse cases | Security requirements | Security ops



Agenda

CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommece/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

SAFECode

CyBOK

SAFECode Charter Members

Adobe
ca technologies
DELL
intel
Microsoft
SIEMENS
Symantec.

3. Secure coding practices

CyBOK

Security control | Design | Secure coding | 3rd party | Testing | Security findings | Vulnerability response | Deploy SDL

8. Planning the implementation and deployment of secure development

CyBOK

Security control → Design → Secure coding → 3rd party → Testing → Security findings → Vulnerability response → Deploy SDL

Agenda

CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommerce/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

Adaptations

CyBOK

- Agile:
  - SAFECode provides security user stories
  - Microsoft Secure DevOps
    - Keeping credentials safe
    - Continuous learning and monitoring
- Mobile
  - OWASP resources for secure testing and threat modeling
- Cloud
  - SAFECode practices for cloud
    - Cloud-based security threats

Adaptations - 2

- Internet of Things (IoT)
  - NIST practices: RFID, not allowing default passwords, use of Manufacturer Usage Description (MUD) specifications, secure update process
- Road Vehicles
  - US Highway/Traffic Safety Administration guidelines, particularly related to the systems engineering
- eCommerce/ Payment Card Industry (PCI)
  - Data Security Standard

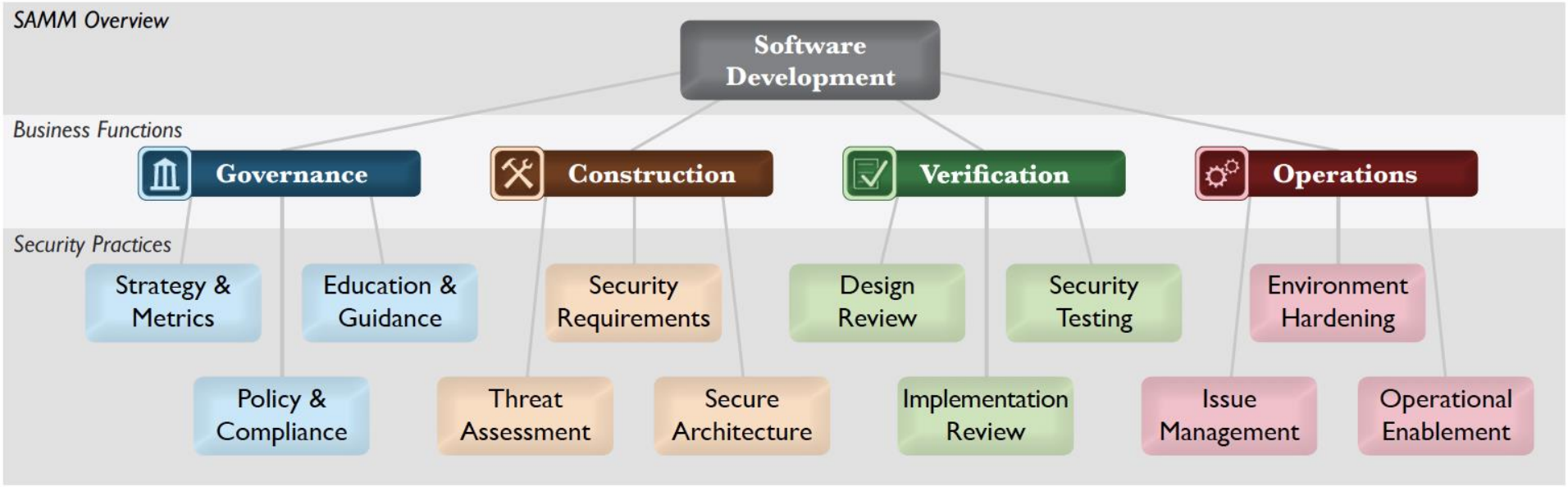
Agenda

CyBOK

- History
- Prescriptive Secure Software Lifecycle Processes
  - Microsoft Security Development Lifecycle (12 practices)
  - Touchpoints (7 practices)
  - SAFECode (8 practices)
- Adaptations of the Secure Software Lifecycle
  - Agile, Mobile, Cloud, Internet of Things (IoT), Road Vehicles, eCommece/Payment card Industry
- Assessing the Secure Development Lifecycle
  - SAMM
  - BSIMM
  - Common Criteria

Open SAMM Framework

CyBOK



BSIMM Framework

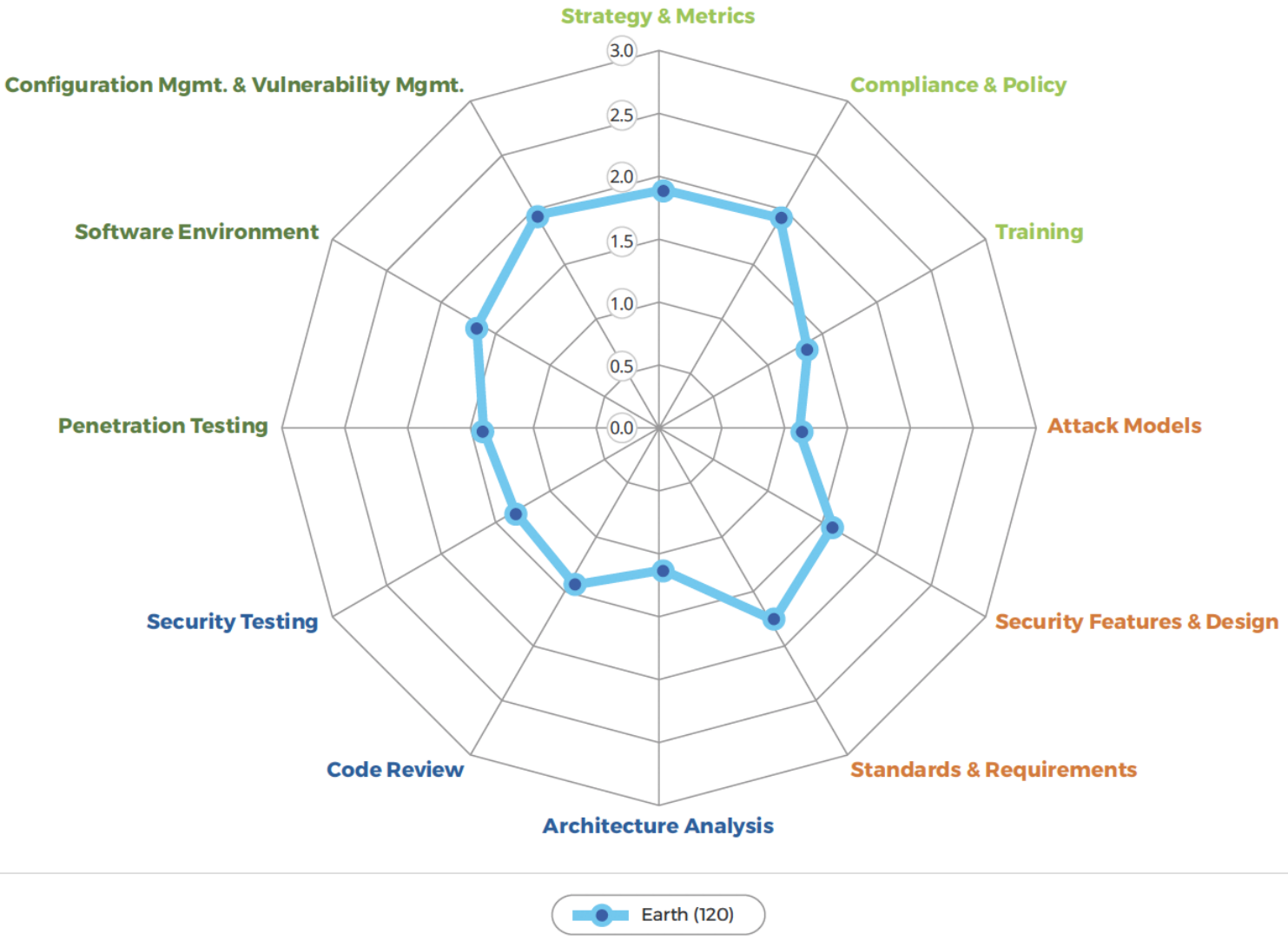
CyBOK




Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy & Metrics (SM)	Attack Models (AM)	Architecture Analysis (AA)	Penetration Testing (PT)
Compliance & Policy (CP)	Security Features & Design (SFD)	Code Review (CR)	Software Environment (SE)
Training (T)	Standards & Requirements (SR)	Security Testing (ST)	Configuration Management & Vulnerability Management (CMVM)

CyBOK

EARTH SPIDER CHART



Common Criteria

CyBOK

Objectives of the Common Criteria

- Permits comparability between the results of independent security evaluations by providing common set of requirements for secure functionality of IT products;
- Establishes level of confidence in the security functionality of IT products; and
- May help consumers determine whether the IT product fulfills their security needs

http://www.commoncriteriaportal.org/theccra.html

CC Assurance* Levels

CyBOK

- EAL1: Functionally Tested
- EAL2: Structurally Tested
- EAL3: Methodically Tested and Checked
- EAL4: Methodically Designed, Tested, and Reviewed
- EAL5: Semiformally Designed and Tested
- EAL6: Semiformally Verified Design and Tested
- EAL7: Formally Verified Design and Tested

Assurance: grounds for confidence that an entity meets its security objectives - http://www.commoncriteriaportal.org/files/ccfiles/ccpart1v2.3.pdf

Summary

- Important for an organization to have a comprehensive secure software lifecycle
- Organizations usually customize their own secure software lifecycle (rather than take a prescriptive approach)
- Successful adoption requires cultural change (in addition to adopting the technical practices)

Images

- https://online.stanford.edu/courses/xacs101-software-security-foundations
- http://www.greatpriceshere.com/2008/06/30/bill-gates-dethroned/
- https://www.govtech.com/dc/articles/Time-for-a-Cybersecurity-Overhaul.html
- https://news.virginia.edu/content/qa-technology-expert-and-uva-grad-gary-mcgraw-talks-cybersecurity
- http://www.webdesigncompany.net/website-optimization-moving-target/
- https://www.ioausa.com/employee-benefits/compliance/
- http://www.michiganemploymentlawadvisor.com/technology-employment-issues/employees-the-weakest-link-in-the-company-data-security-defense/
- https://towardsdatascience.com/the-basics-of-cryptography-80c7906ba2f7
- http://www.ultraedit.com/support/tutorials_power_tips/uestudio/integrated_debugger.html
- https://jrebel.com/rebellabs/developers-guide-static-code-analysis-findbugs-checkstyle-pmd-coverity-sonarqube/
- https://medium.com/@dieswaytoofast/fuzzing-and-deep-learning-5aae84c20303
- https://www.synopsys.com/blogs/software-security/automated-secure-code-review/
- https://www.opsfolio.com/risk-center/use-secure-coding-standards/
- https://content.wisestep.com/implementing-change-workplace-best-tips/