Software Security Assurance

Nadim Barsoum

HPE Fortify EMEA Professional Service

Managing Principal, CISSP, CSSLP, CCSK

OWASP

Agenda

- Software & Industry Breaches
- Software Security Assurance (SSA)
- SSA Program Diagram
- SSA Activities
- Wrap up

Why Software Security?

- Software is Ubiquitous
  - It's in the phones we carry
  - It's in the cars we drive
  - It's in the planes we board
  - It's in….pretty much in everything...possibly even the air we breath :)

Energy Industry Breach

The US Department of Energy's July 2013 Cyber Security Breach

Story

- Early warning signs that certain personnel- related information systems were at risk ignored
- Department did not take action to protect PII of a large number of past and present employees
- July 2013 Exfiltration of PII for 104,000+ individuals.

Impact

-$1.6 Million Credit Monitoring and Call Center

- $2.1 Million in Corrective Efforts

Investigation

- PII info not encrypted in storage
- System was publically accessible
- Lack of Assurance (Security Testing)
- Presence of Critical and High Risk Vulnerabilities
- Failure to update or replace outdated system

What can we do?

Use a systematic approach to treating software risk

- Goal: Reduce Risks posed by Software
- Objectives:
  - Identify Risks
  - Prioritize Risks
  - Treat Risks
- Assurance Activities:
  - Awareness (Secure Coding Training/Guidelines)
  - Security Testing
  - Remediation of findings with respect to impact

Defining Software Security Assurance

- What is assurance?
  - Building confidence / trust in a certain claim
- What is security?
  - Defending the Confidentiality, Integrity and Availability (CIA) of System/Assets
- What is software security assurance?
  - Building Confidence that C I A requirements are met

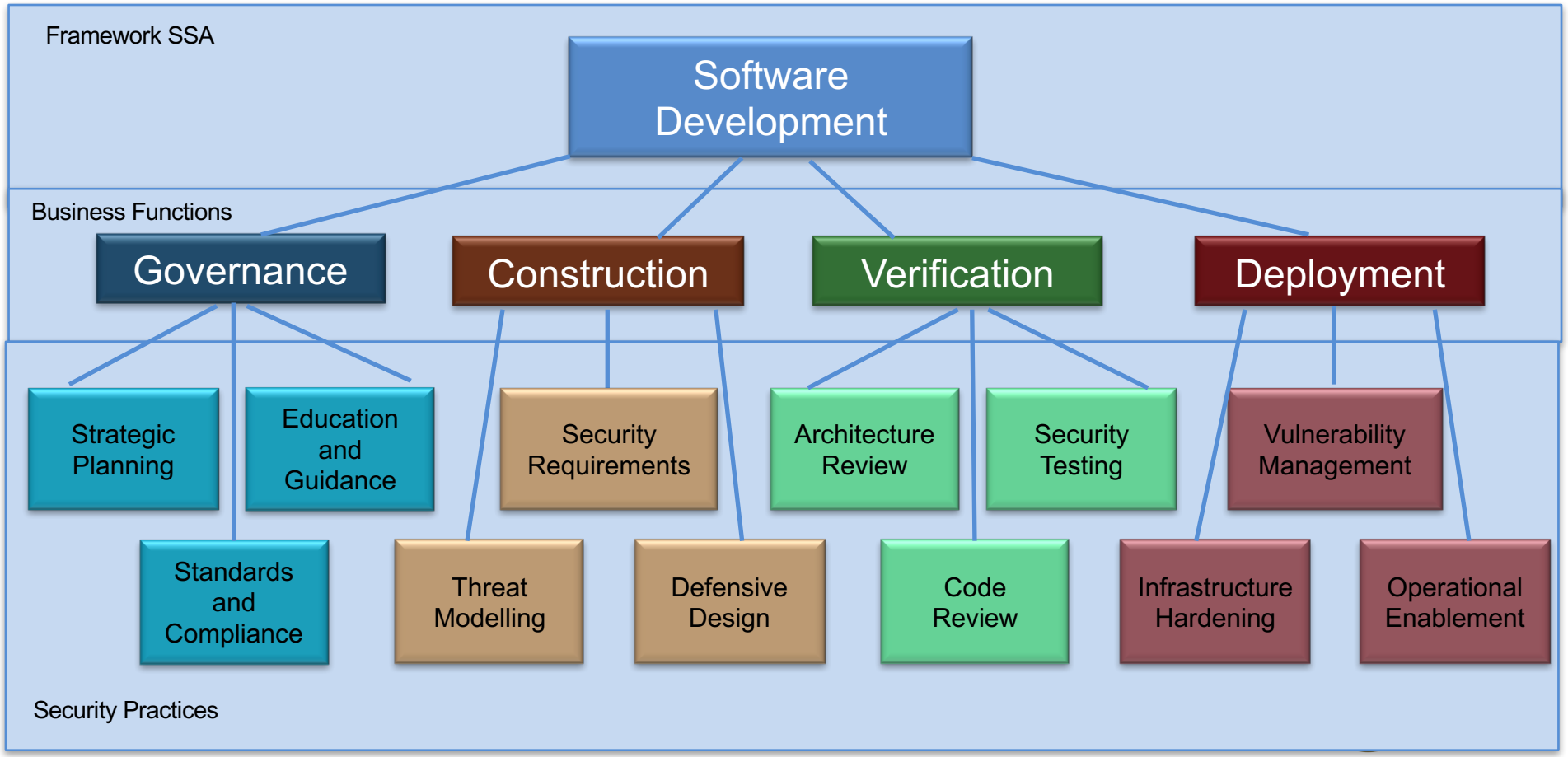
Frameworks and Standards

- Constituents of a structured framework

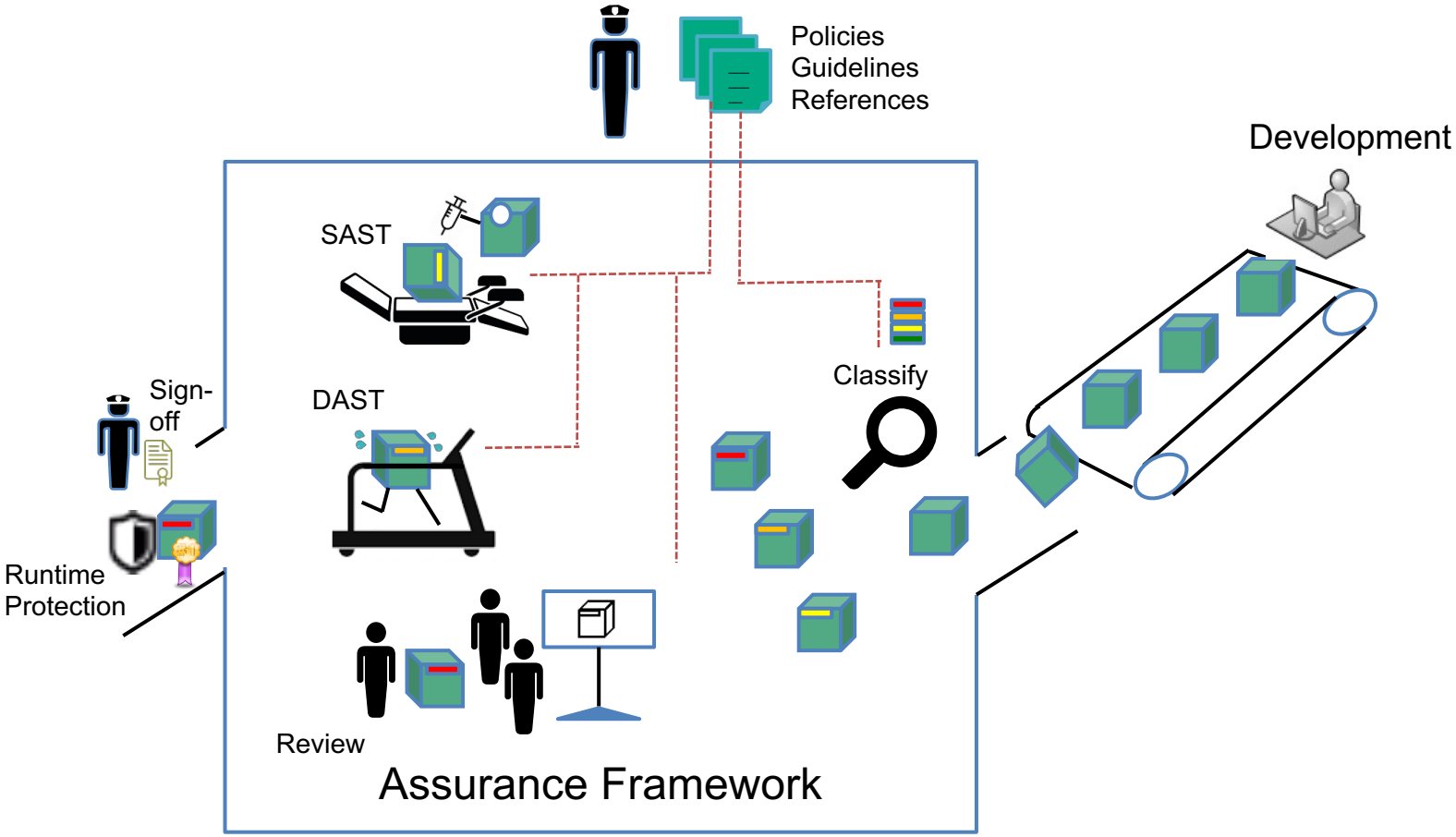
  Provides a structured definition of:

  – Goals
  – Functions that serve those Goals
  – Roles and Duties for each Function
  – Measurements and Performance Indicators
  – Supporting Tools and Resources

Framework*SSA / OpenSAMM



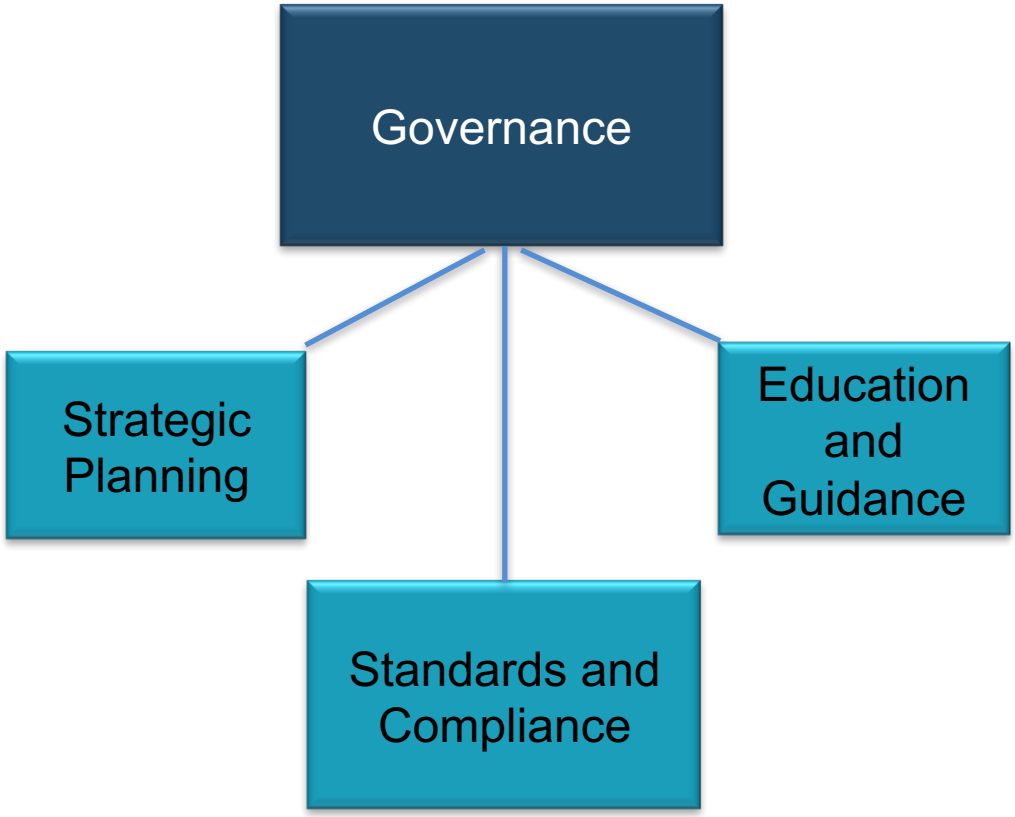
SSA in Practice



Governance | Construction | Verification | Deployment

Strategic Planning
Standards and Compliance
Education and Guidance

GOVERNANCE



Strategic Planning

Initiate a Software Security Assurance Program

- Define Goals of the program with stakeholders
  - Goals, e.g "Lower risks posed by Internet Facing Systems"
  - Motivation, e.g. Government regulation / Competitive advantage
  - Risks, e.g. Loss of Customer Confidence/Business, Penalties
- Define Strategy
  - "Identify and analyse risks specific to Internet Facing Applications used by the organisation"
- Raise Awareness
  - Publish program goals and conduct awareness sessions
- Develop evaluation criteria

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

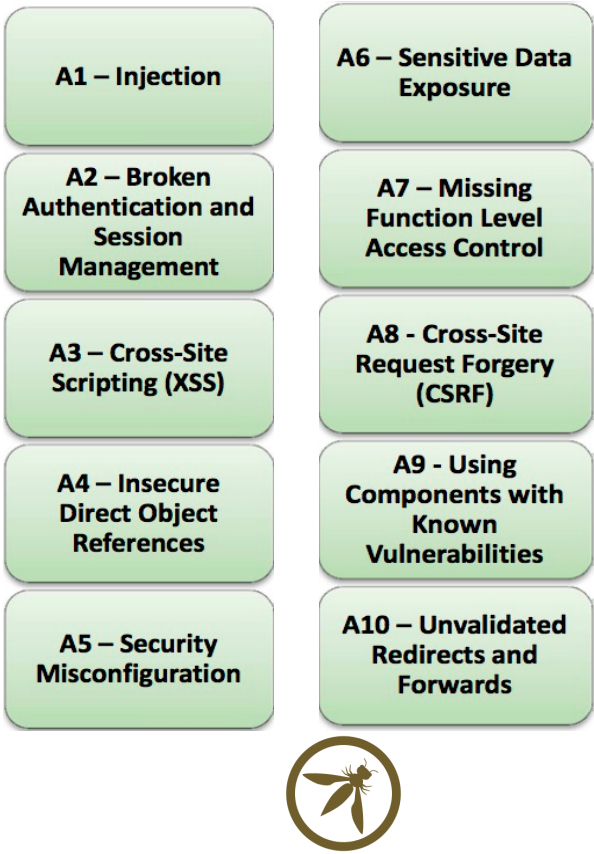
Standards and Compliance

Identify Compliance Drivers

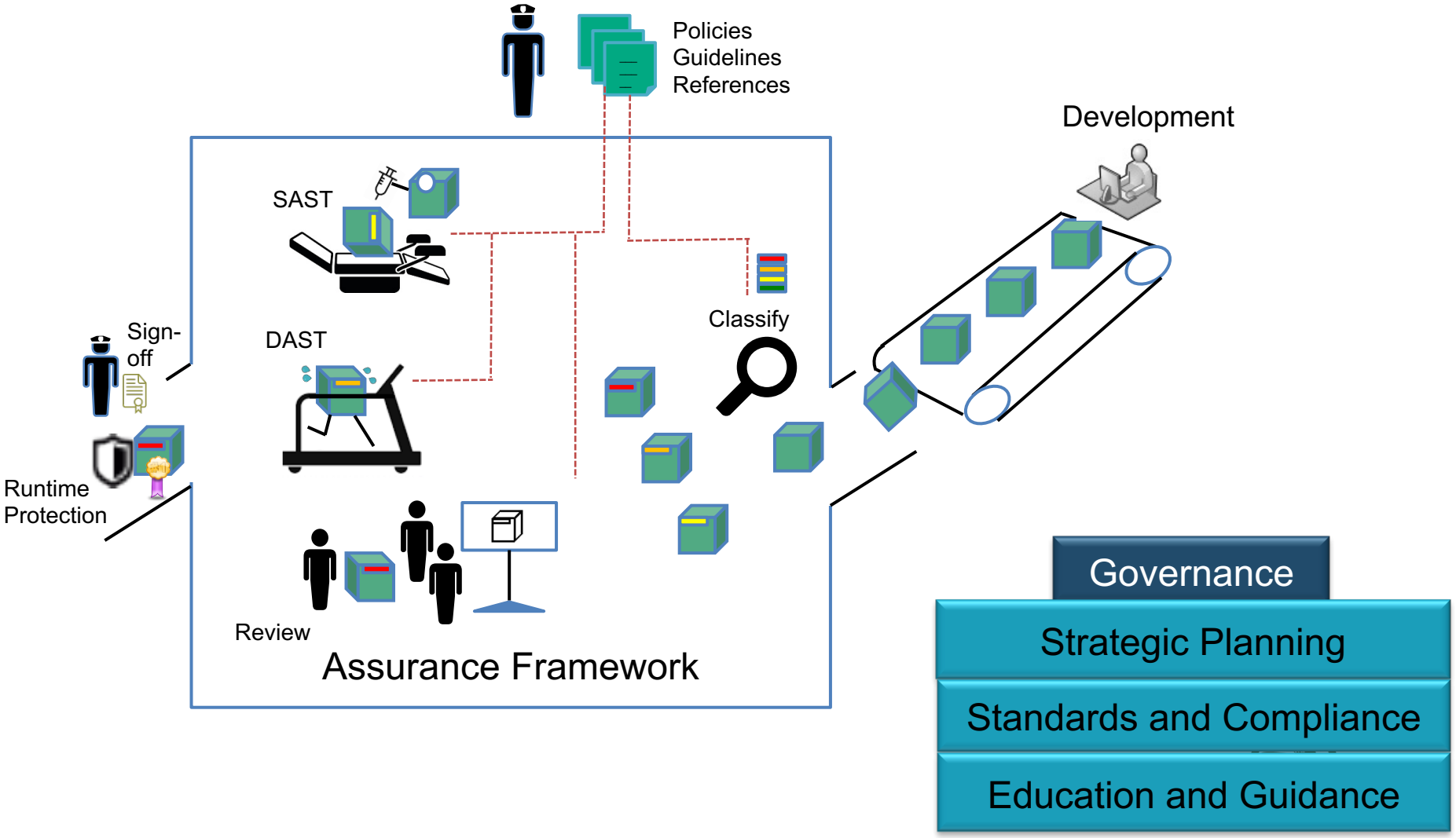
- Identify your software/data obligations to
  - Customers, Partners, Government
- Translate to Security Requirements for projects
  - e.g. Project Requirements Documents state: “All user input must be validated”
  - All projects will state security requirements using template xyz
- Keep Up-to-date

Education and Guidance

- Educate Teams
  - Developers
    - Application Security Awareness
    - Common Vulnerabilities
    - Secure Coding Best Practices
- Assemble Guidance Resources
  - Secure Coding Guidelines
  - Secure Coding Checklists
- Annual Training (refreshers)

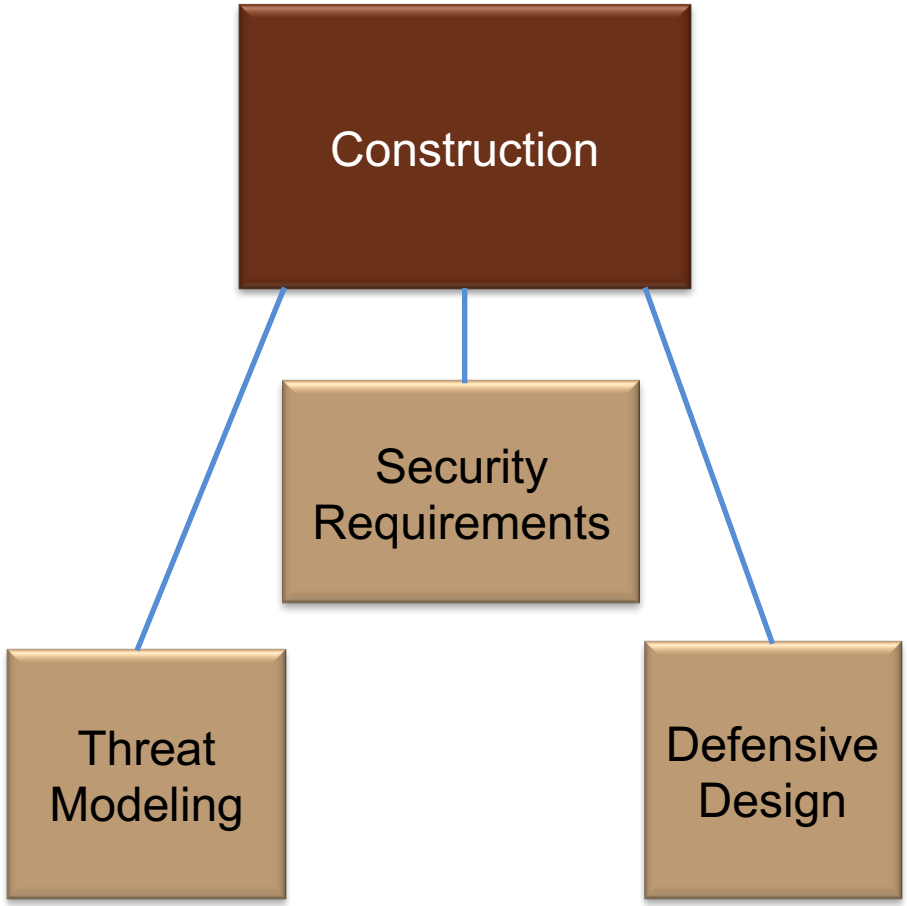


SSA in Practice

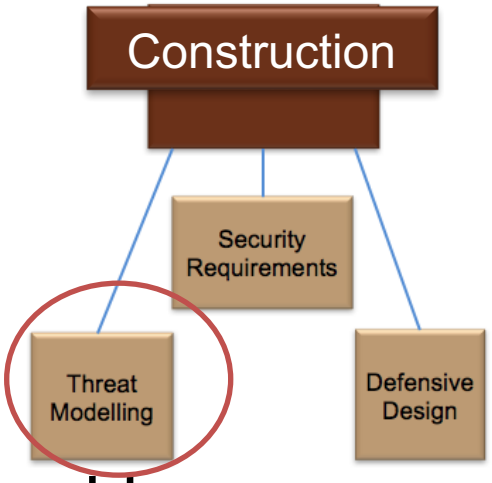


Threat Modeling
Security Requirements
Defensive Design

CONSTRUCTION



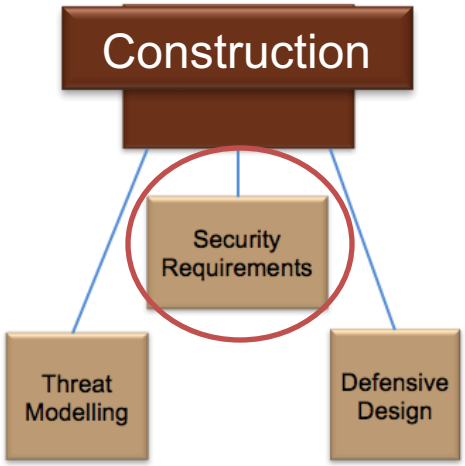
Threat Modeling



Identify high-level threats for projects

- Projects should list relevant Threats and how they handle them
  - Find threats by listing worst-case scenarios (abuse Cases)
  - Recognise threat agents (Internal and External)
  - Identify controls and gaps
  - Collect these for further adoption by organisation (maturity)
  - Repeat as new features added to projects

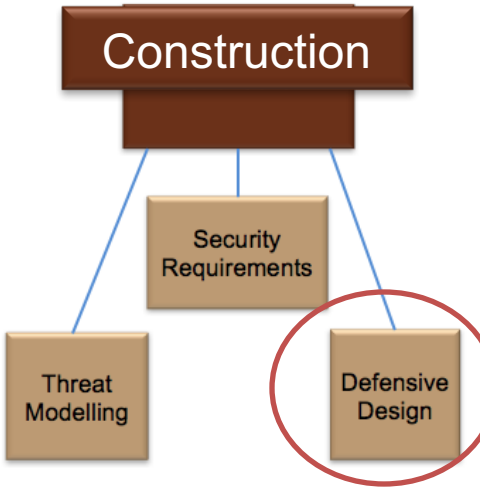
Security Requirements



Identify Security Requirements for each project

- Derive from Functional Requirements
- SMART (specific, measurable, attainable, realisable, traceable)
- Based on Industry Best Practices and Standards
- Documented
- Analyse existing systems for missing requirements
- Refactor existing systems to implement missing requirement by priority

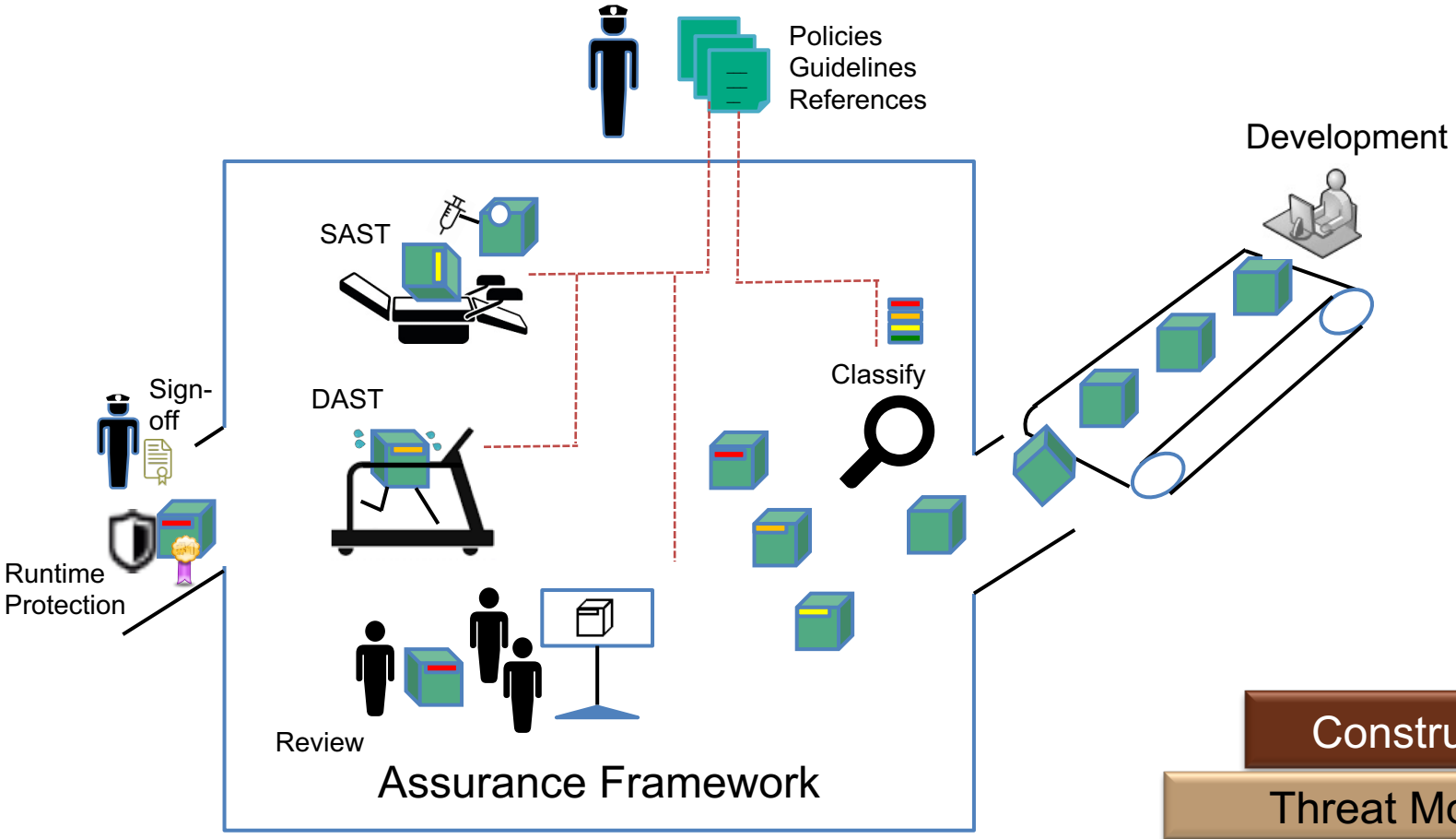
Defensive Design



Identify Dependencies, Acceptable Third Party Components & Interfaces

- Teams document their usage of third party components
  - Open Source Libraries / Free tools
  - Commercial Libraries
- Secure Design Principles Guidance
  - Secure Design Guidelines
  - Secure Design Checklists

SSA in Practice



Construction

Threat Modeling

Security Requirements

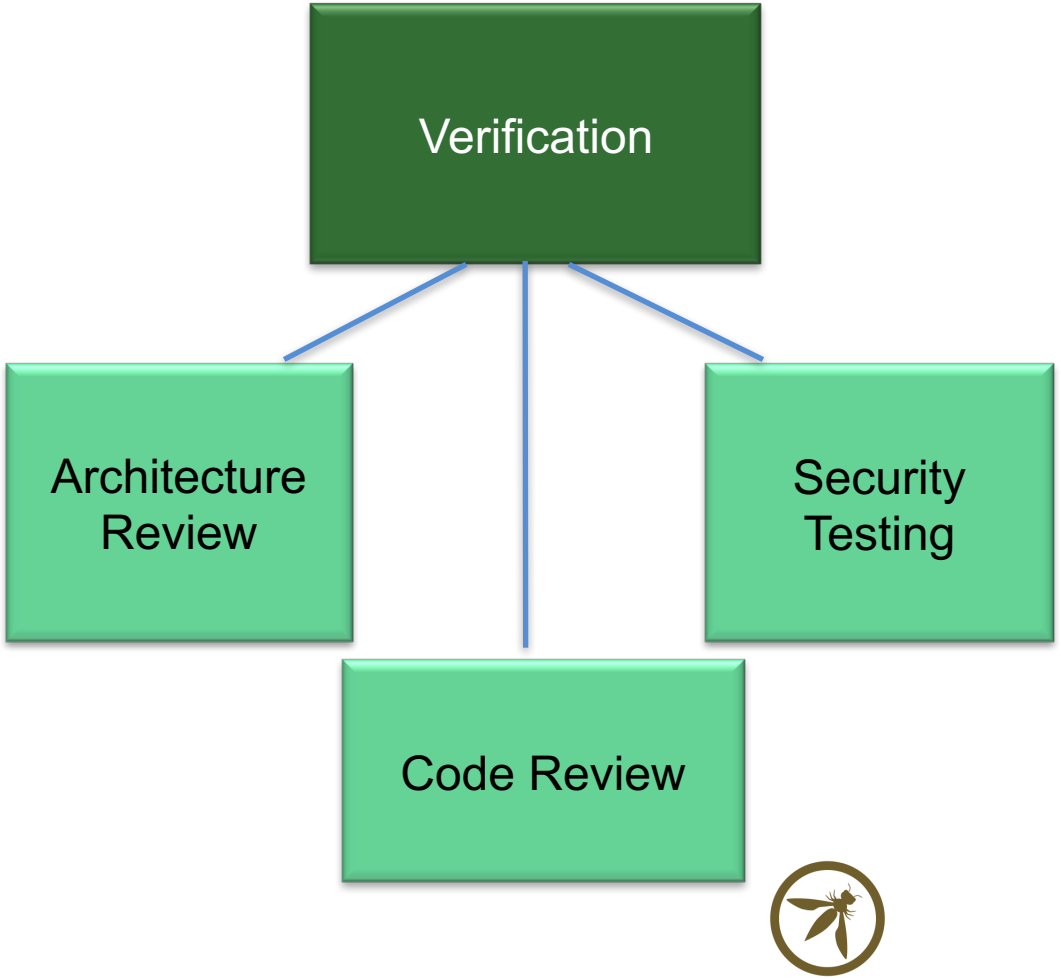
Defensive Design

Architecture review

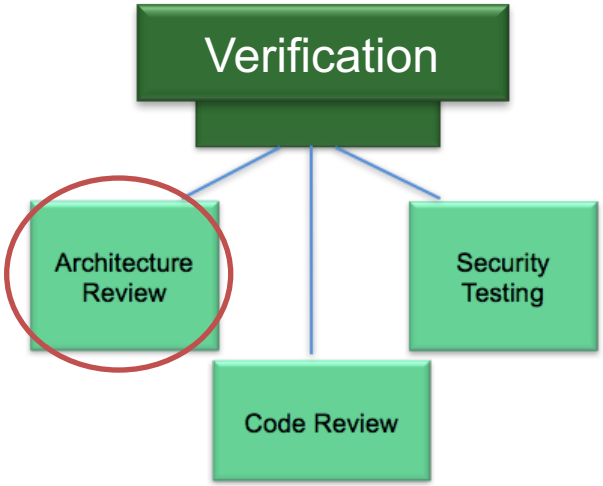
Code Review

Security Testing

VERIFICATION

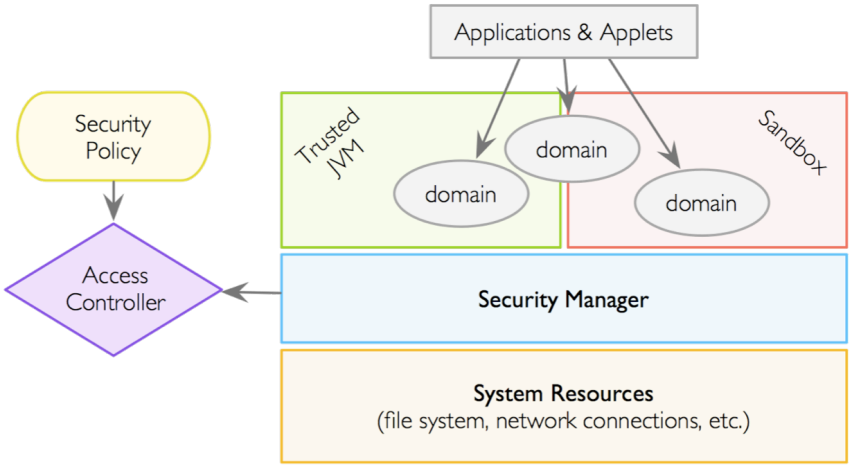


Architecture Review



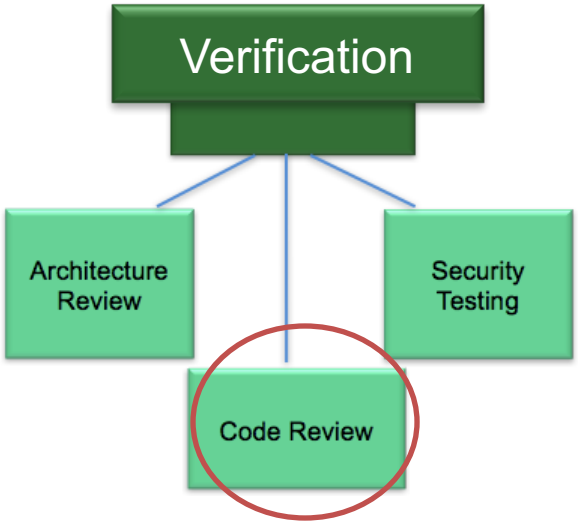
Annotated Architecture Diagram is provided for each project

- Describes User-roles and exposed functionalities
- Describes Assumptions and Trust-boundaries
- Identifies related security functionalities
- Checked for missing security features
- Updated as system changes



Code Review

```
98    ThreadWatcher watcher;
99
100   try
101   {
102       // start the command
103       child = Runtime.getRuntime().exec(command);
104
105       // get the streams in and out of the command
106       InputStream processIn = child.getInputStream();
107       InputStream processError = child.getErrorStream();
108       OutputStream processOut = child.getOutputStream();
109
110       // start the clock running
111       if (timeout > 0)
112       {
113       watcher = new ThreadWatcher(child, interrupted, timeout);
114       new Thread(watcher).start();
115       }
116
117       // Write to the child process' input stream
118       if ((input != null) && !input.equals(""))
```

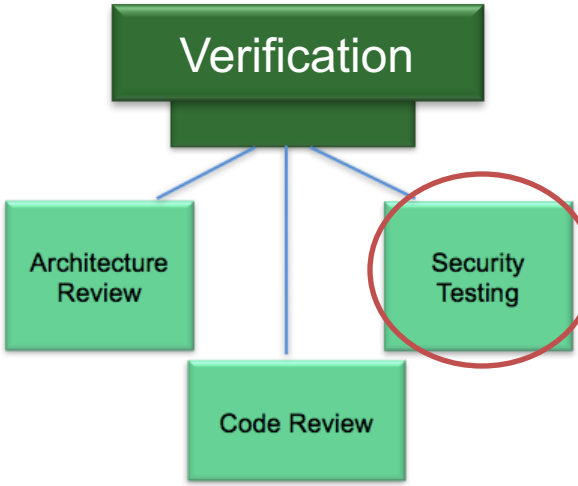


Project utilise Secure Coding Gu

Documentation

- Secure Coding Checklists
- Training on Secure Coding and Checklist usage
- High-risk projects are reviewed first, findings handled by priority
- Tools for review (SAST)
  - Internal, External, Consultant-driven...etc

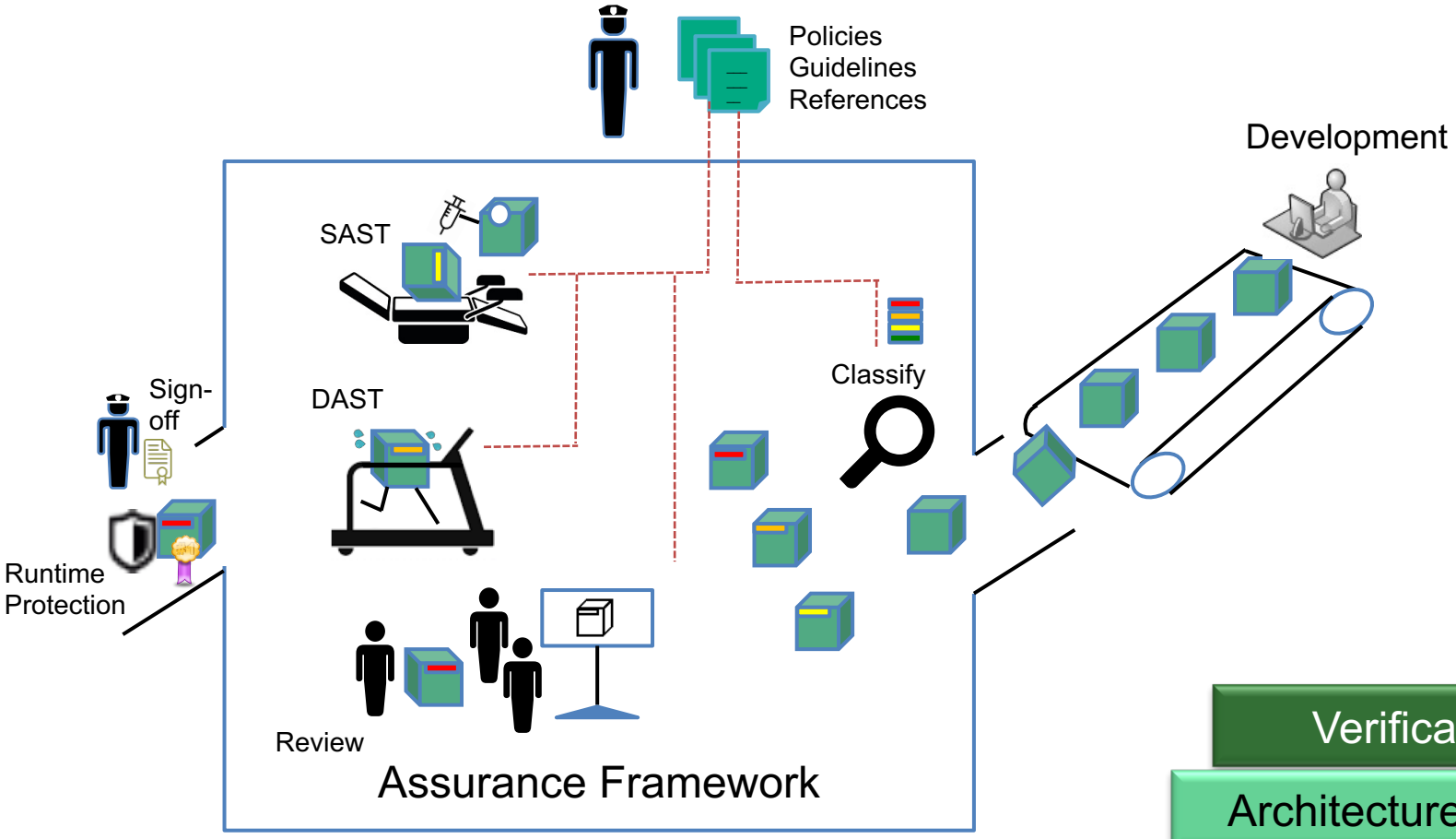
Security Testing



Define Testing Requirements based on Security Requirements

- Projects define test cases
- Projects perform penetration test before release
- Findings are reviewed and treated as per prioritisation with stakeholders
  - Tools for review (DAST)
    - Internal, External, Consultant-driven...etc

SSA Clock-work



Verification

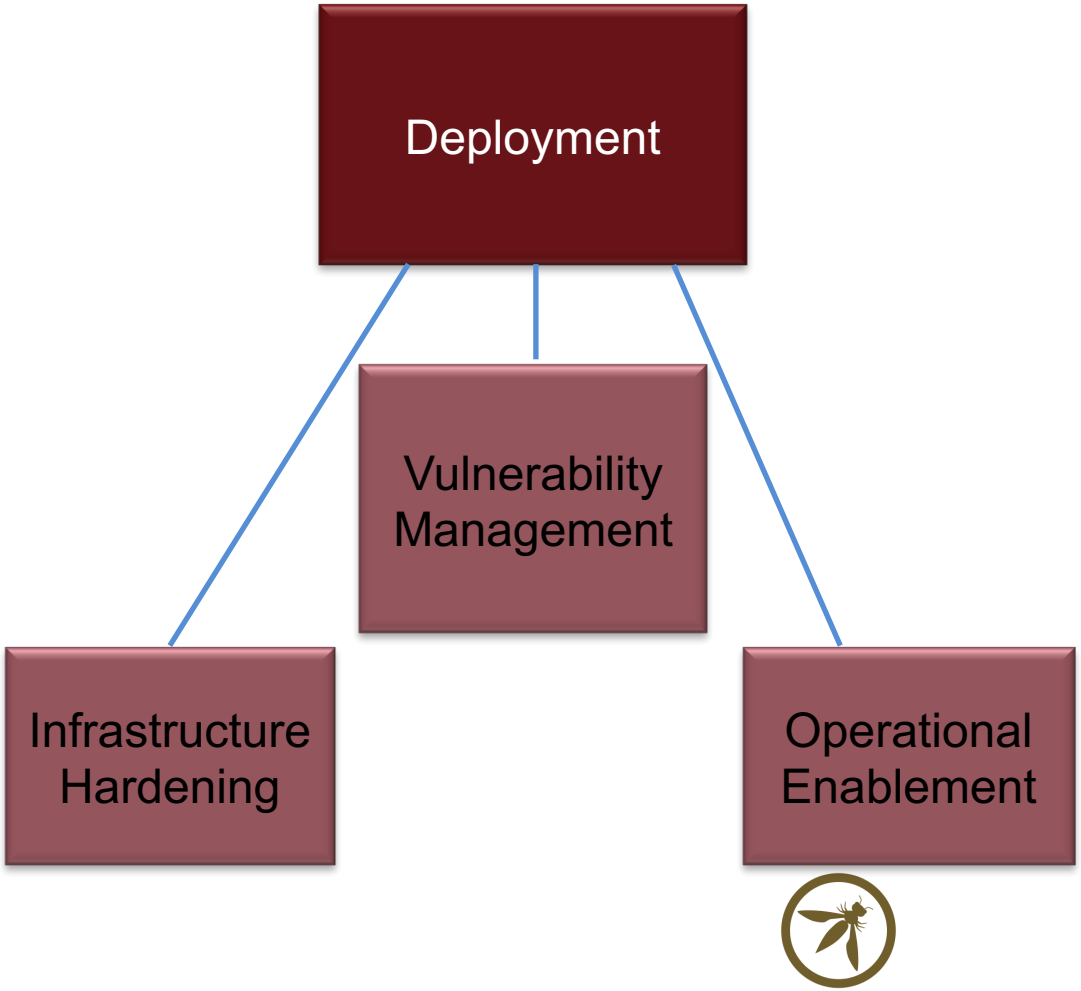
Architecture Review

Code Review

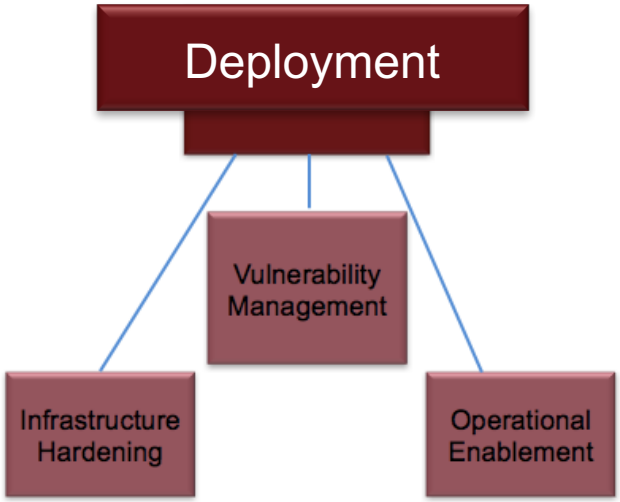
Security Testing

Vulnerability Management
Infrastructure Hardening
Operational Enablement

DEPLOYMENT



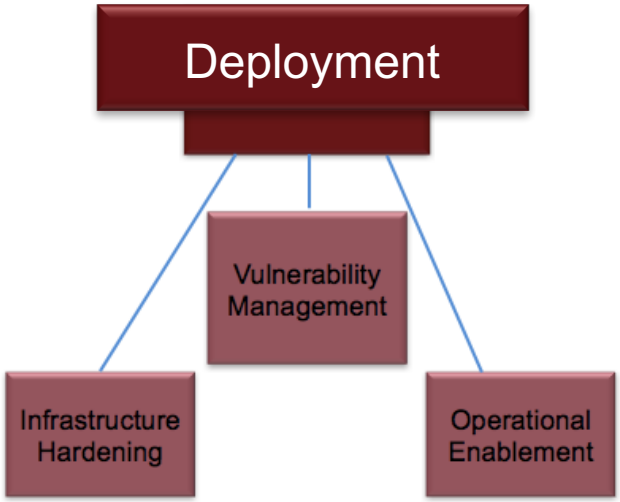
Vulnerability Management



Identify Key Security Resource

- Security Response Team
  - Security-Savvy developer per project
- Incident Response Process
  - Manage Incidents
  - Report Vulnerabilities to project
  - Follow-up on Remediation
  - Report to Stakeholders

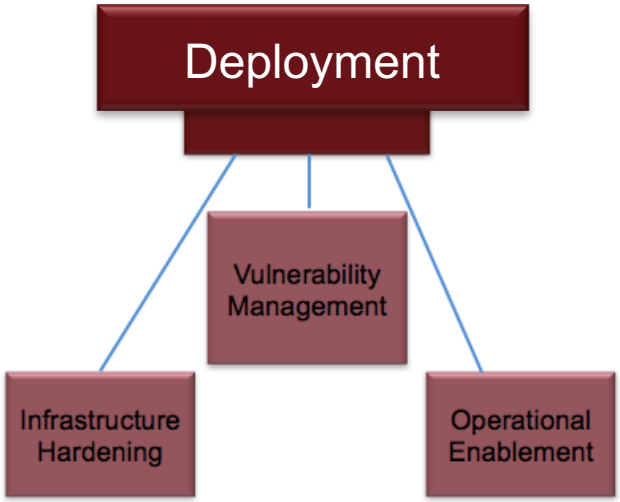
Infrastructure Hardening



Set and Agree on Expectations for Operating Environment

- Describe Baseline Configurations available for projects
- Document project run-time requirements and deviations from baseline
- Monitor critical updates for infrastructure, platform and dependencies (e.g. Libraries)
- Separation of Duties (Operations / Development)

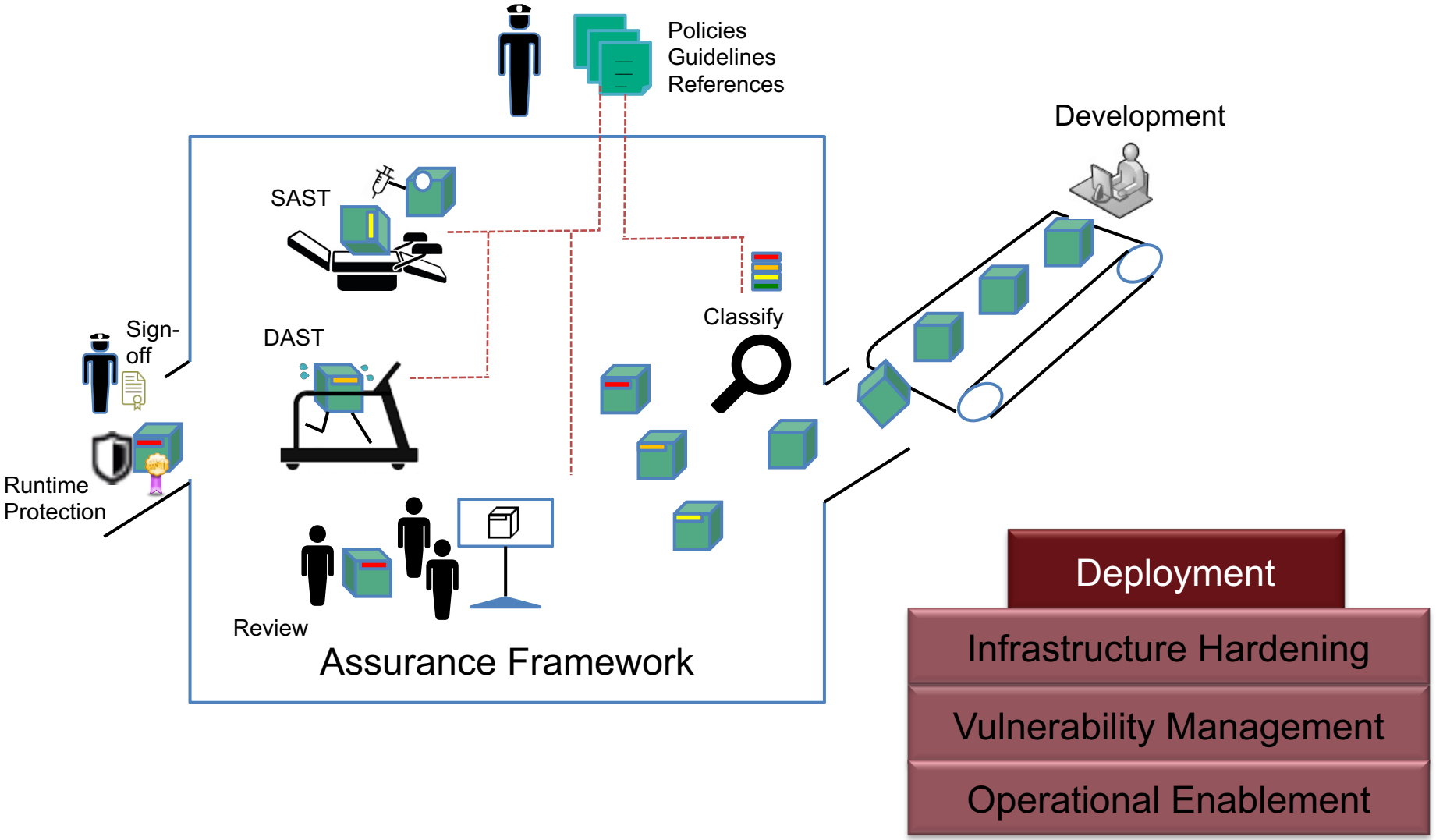
Operational Enablement



Inform Operations on secure usage of system

- Project documents
  - Security Features and how to configure them
  - Secure Defaults and the security impacts of changing configurations
  - Security-relevant alerts and error conditions
- Operations to Monitor system and Take Proper Action

SSA Clock-work



Incident Response

- Application design
  - Security Events – what is recorded?
- Application Implementation
  - Frameworks for logging
  - Interfaces and Integrations with common detection systems (e.g. SIEM)
- Documentation – how to respond from a contextual point of view
- Protection and response

SSA IN PRACTICE

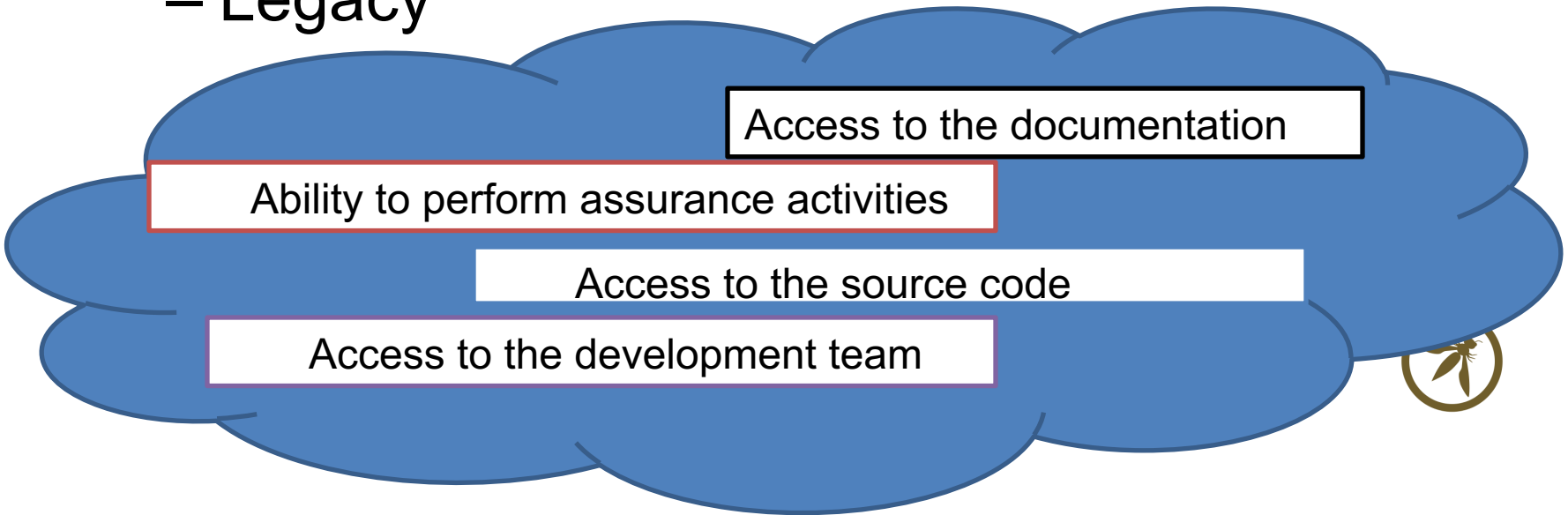
Classification

- How do you classify your projects risk?
  - Data Processed (C I A)
  - Business Functions served
  - Surface of attack / Exposure
  - Threats and their Likelihood

Project Types

Consider how you may handle these differently:

- Internal Development Projects
- Commercial Off-the-shelf (COTS)
- Vendor Developed / Outsourced Projects
- Legacy



SSA

- Resourcing SSA Programs
  - Size of development effort and number of projects
  - Size of development community/department
  - Current Resources
    - Developers, Managers, Business Owners
    - QA Testers
    - Auditors / Consultants
  - Development Lifecycle Stage of Projects
    - New, Stable, BAU, Retiring

Solutions and Resources

- Tools and Solutions
  - Internal vs. External Resourcing
    - SAST: Are we allowed to share our code?
    - DAST: Is our system accessible to the outside world?
    - Threat Modeling: Do we have access to the necessary documents or people?

In Sum

- Start by assessing where you stand
- Inspect Business Functions
- Identify Current Security Practices
- Identify Gaps
- Set a roadmap
- Update your SSA strategy/plan
- Implement Changes
- Repeat

Thank You

- Questions?

Contact: Nadim.barsoum@hpe.com