FoRMA: An Open Framework of Risk Management & Analysis Applied to SDLC

Kris Kahn, CISSP, CISA, OPSA

June 29, 2006

Agenda

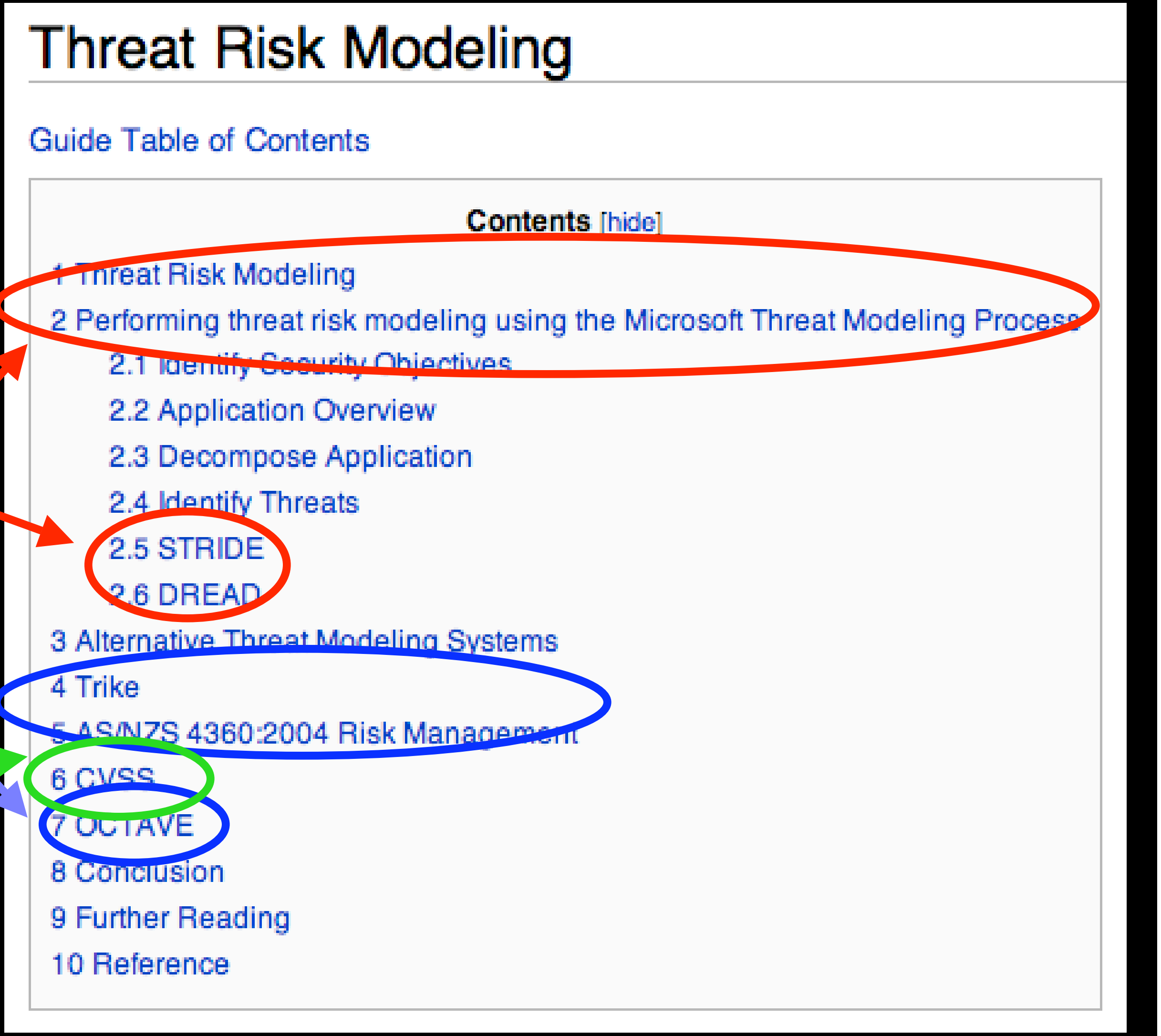
- Introduction
- Comparison
- Benefits
- Overview
- SDLC Controls & Risks
- Objective
- Building the Foundation
- Risk Mitigation Cycle
- Implementation Phases
- FoRMA Model
- Results
- References

Introduction

- About the author
  - □ **Kris Kahn, CISSP, CISA, OPSA**
  - □ **Sr Governance Analyst, Seagate Technology LLC**
- About the model
  - □ **Three years in development**
  - □ **Based on author's security and audit experience**
- About this presentation
  - □ **Risk Management applied to the Software Development Life-Cycle (SDLC)**

Model Comparison

- Threat Models
- Risk Models
- Scoring System



Threat Risk Modeling

Guide Table of Contents

Contents [hide]

1 Threat Risk Modeling
2 Performing threat risk modeling using the Microsoft Threat Modeling Process
    2.1 Identify Security Objectives
    2.2 Application Overview
    2.3 Decompose Application
    2.4 Identify Threats
    2.5 STRIDE
    2.6 DREAD
3 Alternative Threat Modeling Systems
4 Trike
5 AS/NZS 4360:2004 Risk Management
6 CVSS
7 OCTAVE
8 Conclusion
9 Further Reading
10 Reference

Source: http://www.owasp.org/index.php/Threat_Risk_Modeling

Benefits of FoRMA

- **Big Picture:** FoRMA will help to provide a holistic vision and strategic understanding of the relationships of many of our current and familiar security models.

- **Technology independent:** FoRMA is flexible enough that it can be applied to information security, physical security, even medical risk management issues.

- **Business Focused:** FoRMA help achieve business objectives by minimizing risk, not by maximizing security.

Overview

- New model focusing on Risk Management & Analysis
- An Open Framework for integrating industry standard models, such as CIA*, STRIDE* and others
- Addresses Risk and Control elements:
    - □ **Risk**
        - Threat
        - Vulnerability
    - □ **Control**
        - Technology
        - Process

*: See references at the end of the presentation material

Software Development Life-Cycle

- SDLC Framework
  - □ **Design**
    - Include **Standards and Coding Principles**
  - □ **Develop**
    - Include **Best Practices** for coding and configuration
  - □ **Test**
    - **Testing** is performed by developers to ensure program meeting design requirements
  - □ **Staging**
    - **Quality Assurance** is performed to certify the program for production deployment
  - □ **Post-Production**
    - **Reaccredidation** is performed as part of the **change control process** for significant changes to ensure certification

Risks (SDLC)

- Awareness Gaps
  - □ **Lack of security best practice knowledge**
- Protection Gaps
  - □ **Code is promoted by developers**
- Detection Gaps
  - □ **Coding errors cause information leaks**
- Assurance Gaps
  - □ **Application is unstable and prone to attacks**

Controls (SDLC)

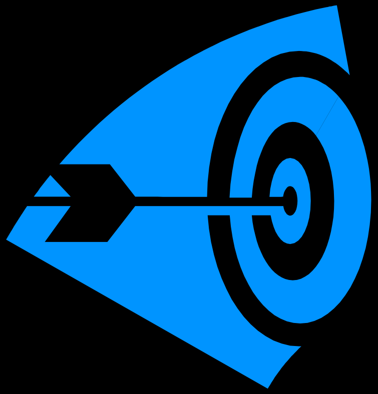
- Awareness
  - **Establish an SDLC standard and train developers**
- Protection
  - **Use tollgates in SDLC to advance between key steps**
- Detection
  - **Perform code reviews to ensure consistency**
- Assurance
  - **Conduct Quality Assurance Certification**

FoRMA Goal: Risk Mitigation

- **I.e. Control risks within acceptable limits to support business objectives**

Establish Your Boundaries

- Define relevant **policies**, standards and best-practices
- Protect assets and resources in accordance with **policy**
- Detect **policy** violations
- Assure **policy** compliance

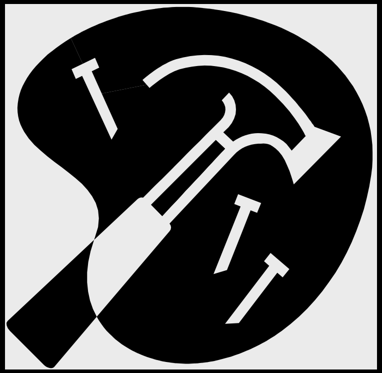


Building the foundation

Start from the ground level and work your way up!

- Construct a strong security foundation to build your security policies, standards and best-practices. Use industry established security methodologies and codes of best practice to guide your standards and practices.

- A security foundation supports all layers (including physical, network, application, etc), and addresses each security implementation phase (Awareness, Protection, Detection, and Assurance).



Building the foundation:
The Relationships

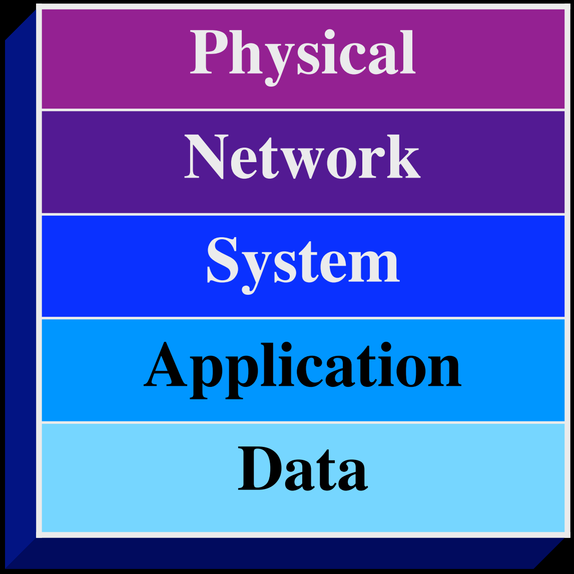
Methodology	Sub-Model	Subject
Threat Management	STRIDE*	Threat
Security Architecture	APAIN*	Technology
Security Management	RIVET*	Process
Asset/Resource Management	CIA*	Vulnerability

Use **Methodology** with **Sub-Model** to evaluate **Subject**

*: See references at the end of the presentation material

Building the foundation:
The Layers

- This is a layered model based on the ISO Protocol model* which identifies five (of the original seven) layers of critical assets and resources we want to protect.



Physical

Network

System

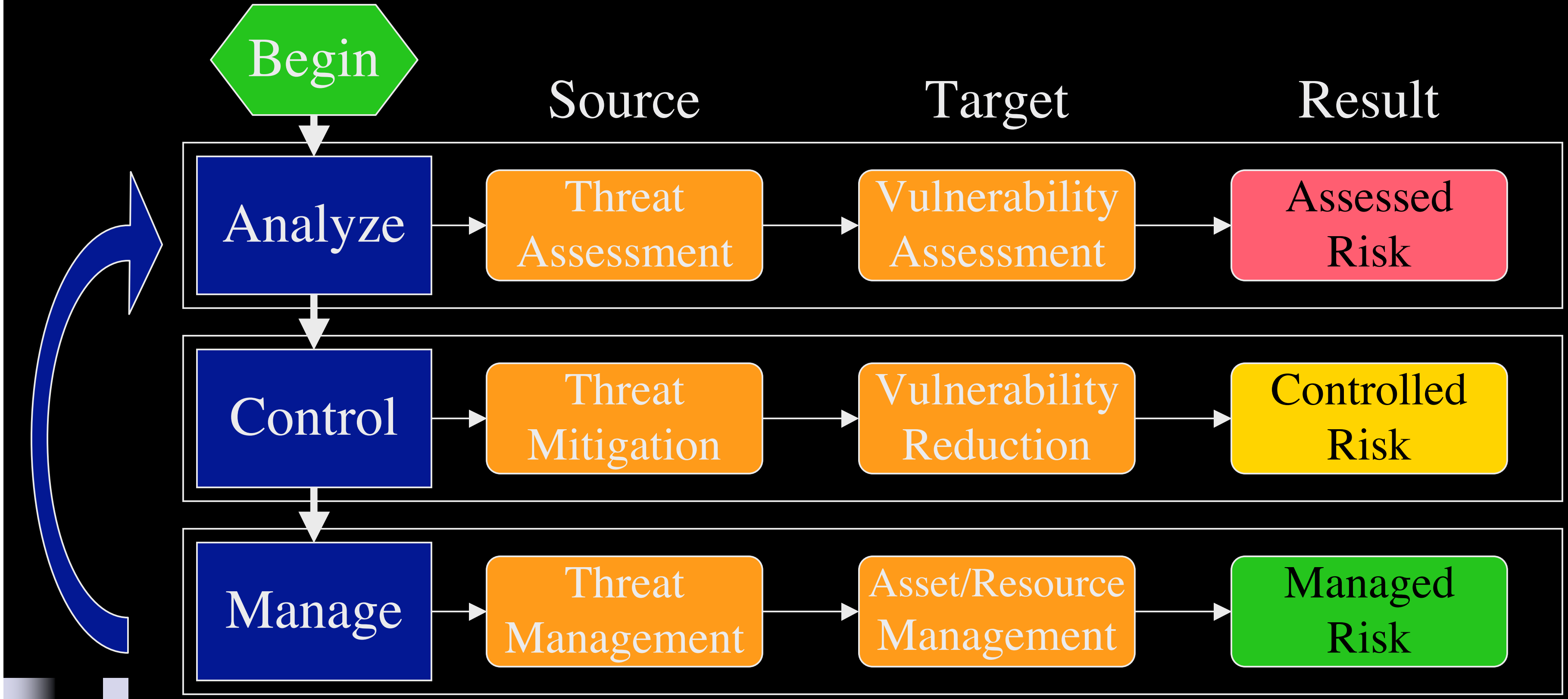
Application

Data

*: See references at the end of the presentation material

Risk Mitigation Cycle

- **Analyze, Control, Manage, repeat.**
- This process life cycle will guide you through the security model to the appropriate security resolution.

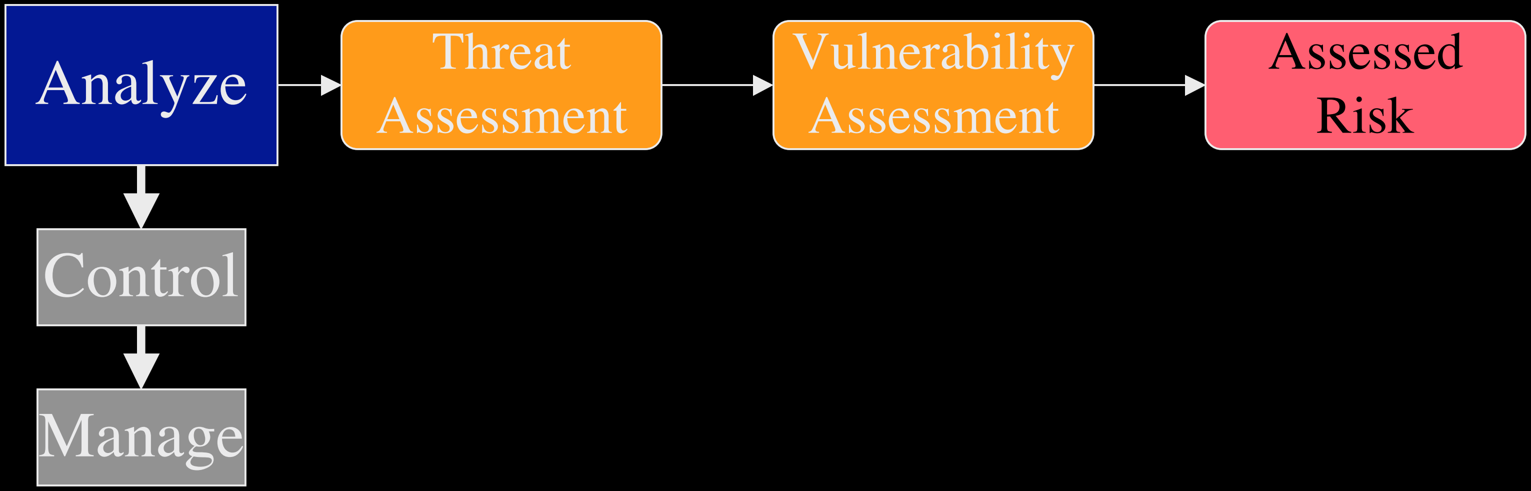


Begin

	Source	Target	Result
Analyze	Threat Assessment	Vulnerability Assessment	Assessed Risk
Control	Threat Mitigation	Vulnerability Reduction	Controlled Risk
Manage	Threat Management	Asset/Resource Management	Managed Risk

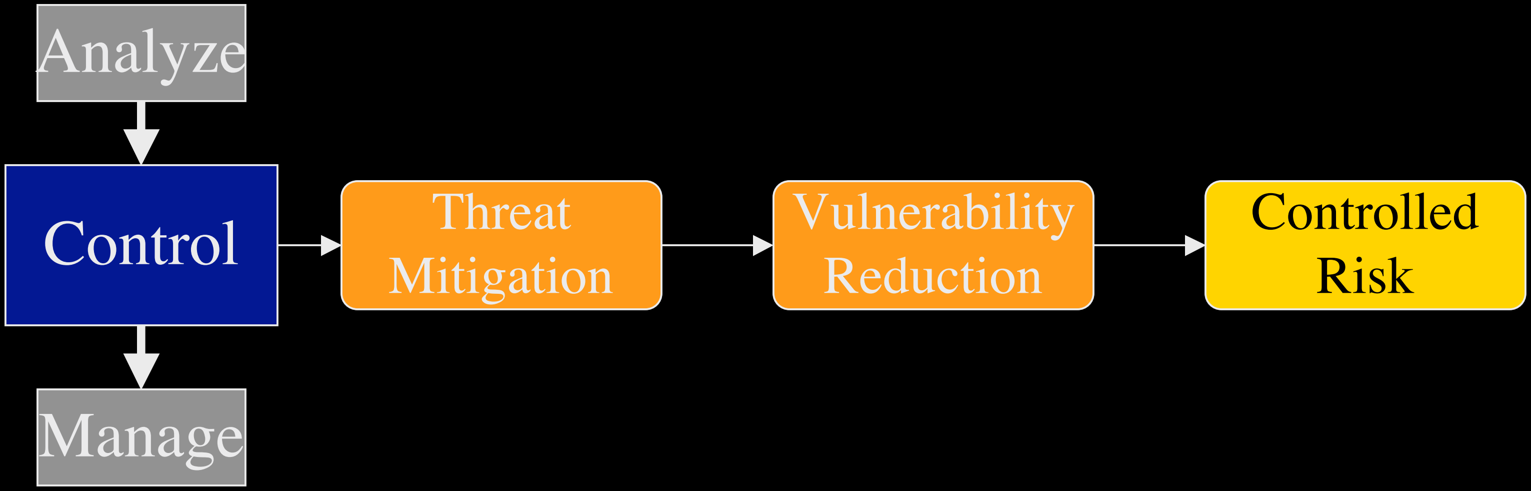
Risk Mitigation Cycle: Analyze

- First, to determine the risk, you must understand the threat of attack and the vulnerability of the asset or resource. We measure and analyze these items in detail to determine the corresponding risk.



Risk Mitigation Cycle: Control

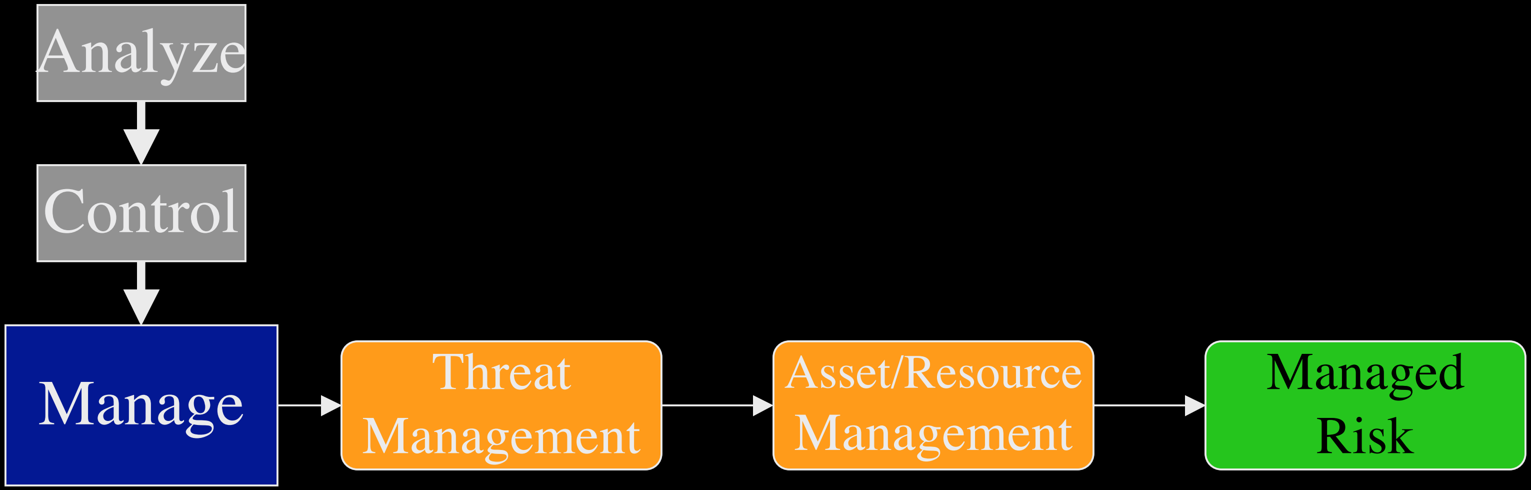
- Once you have assessed the risk, you can apply control-mechanisms in the form of technology to mitigate the threat or reduce the vulnerability.



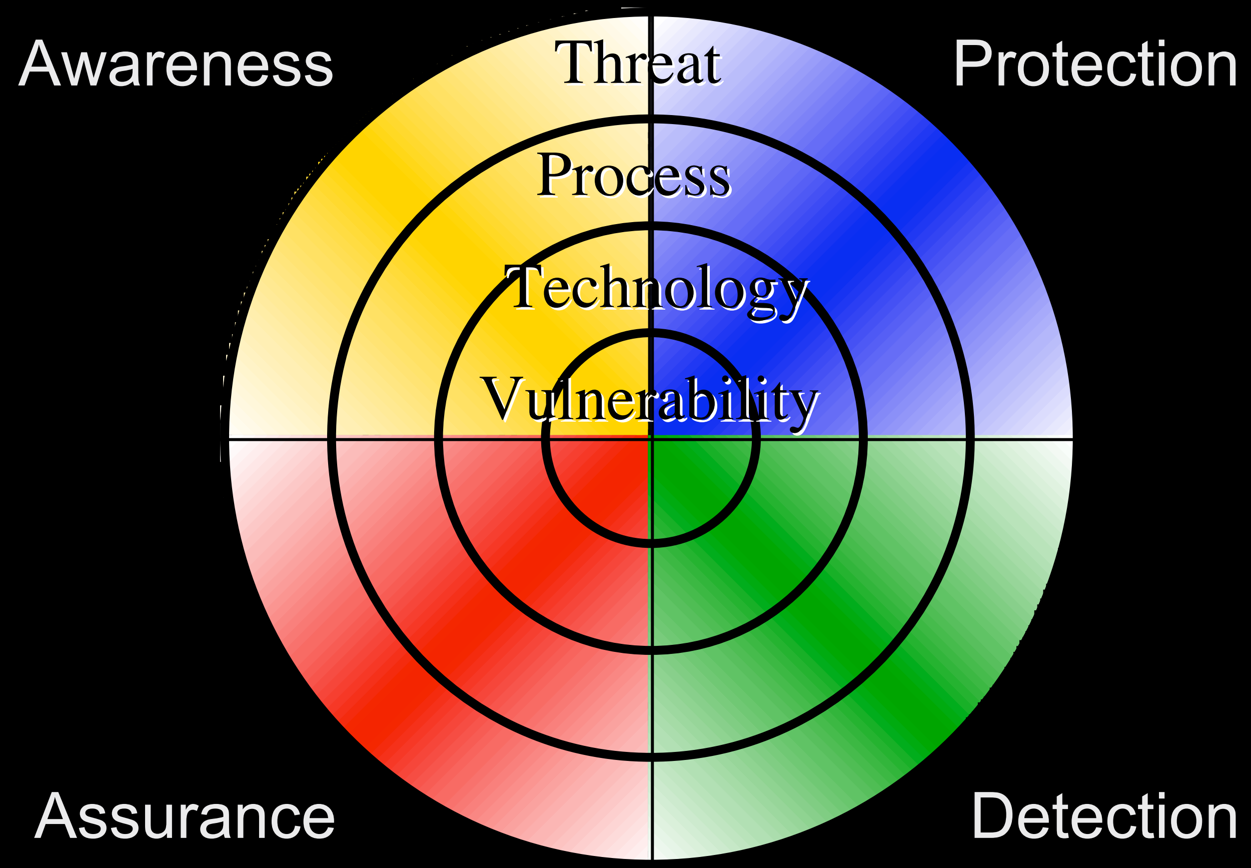
Risk Mitigation Cycle: Manage

- Once a system is live, you apply counter-measures in the form of processes in the event of an attack (Incident Response) or to assure the integrity of the technology (Security Assessments).

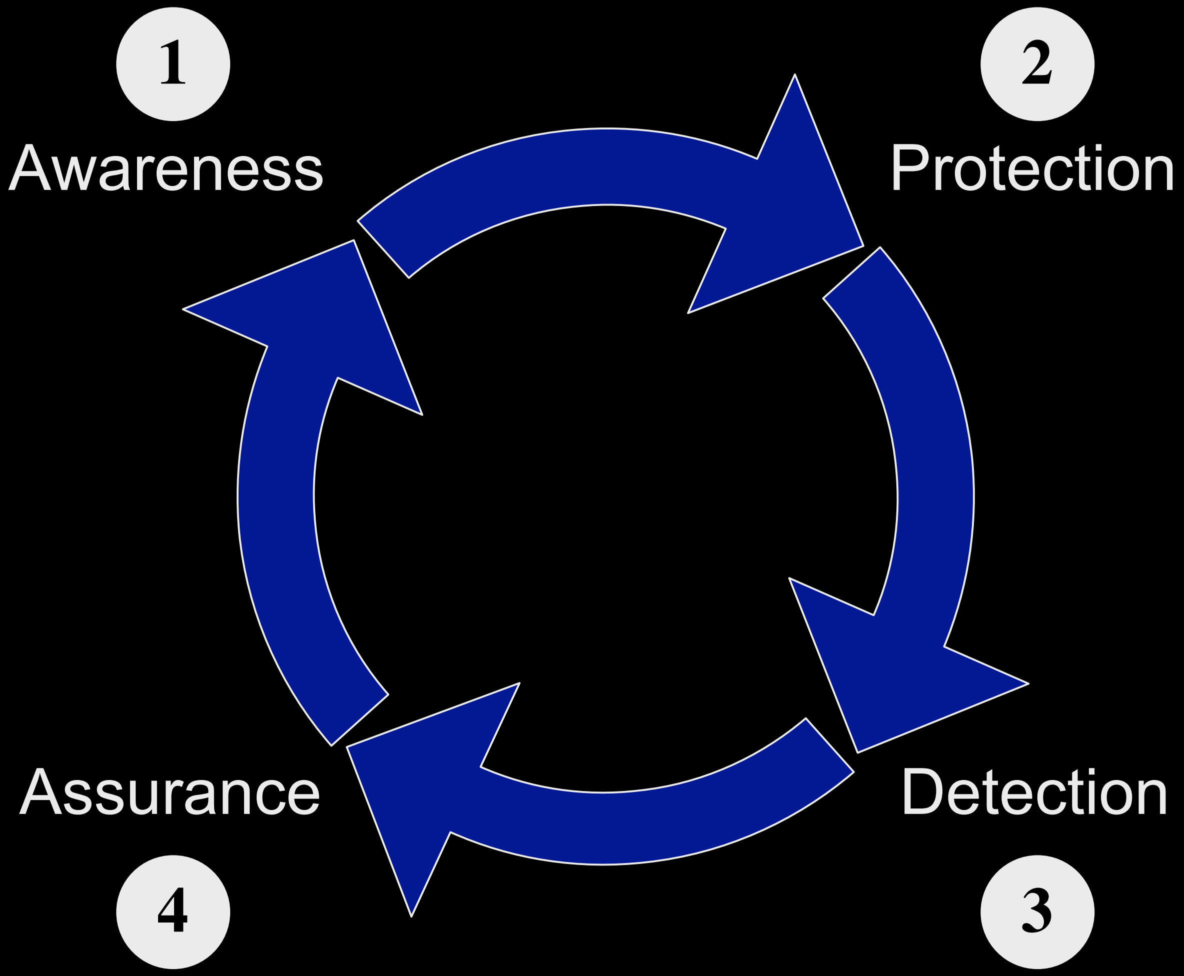
- Implement change control and regular audit processes to verify when an aspect of the formula has changed.



FoRMA Model Overview



Implementation: Phases



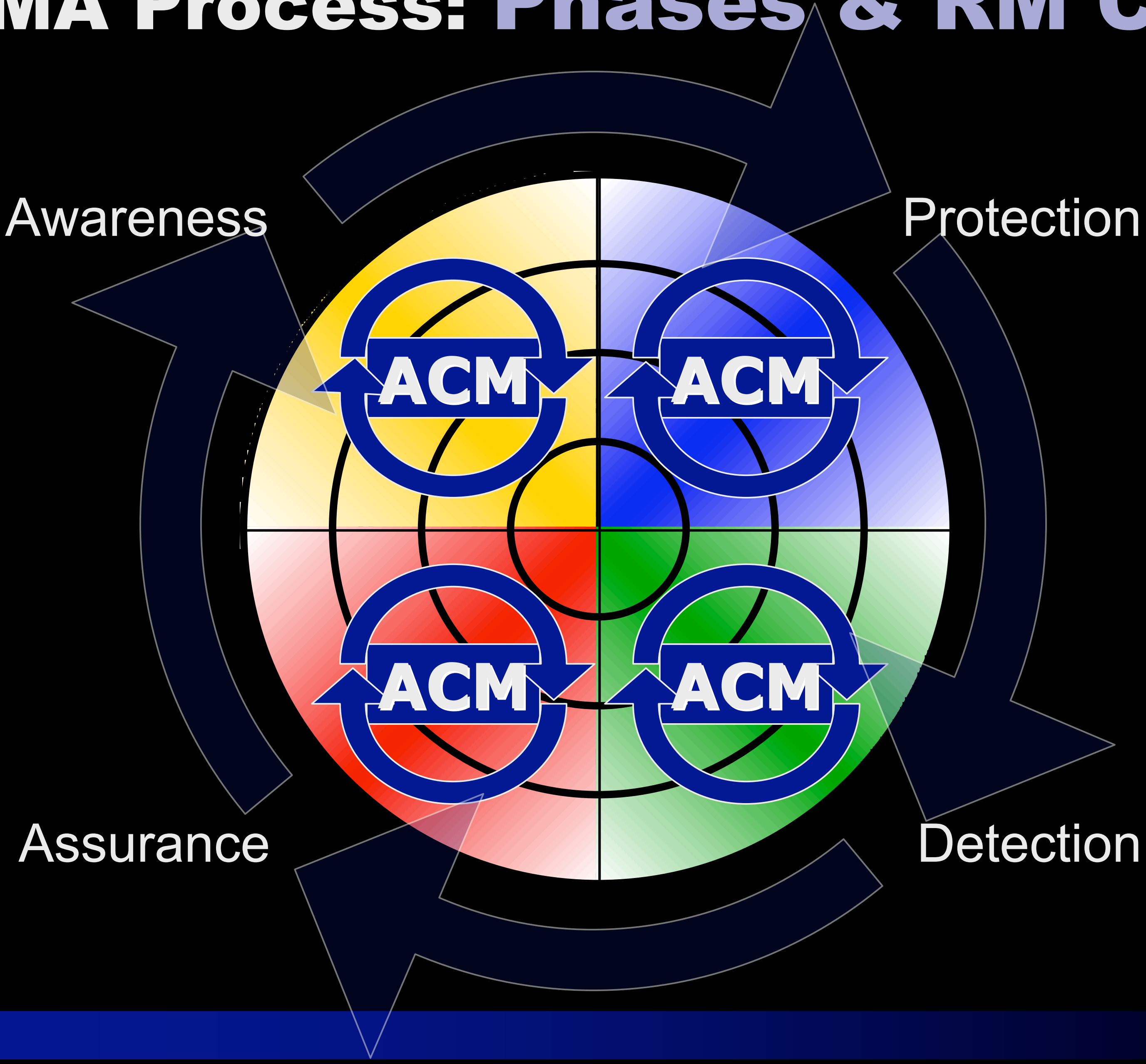
1 Awareness

2 Protection

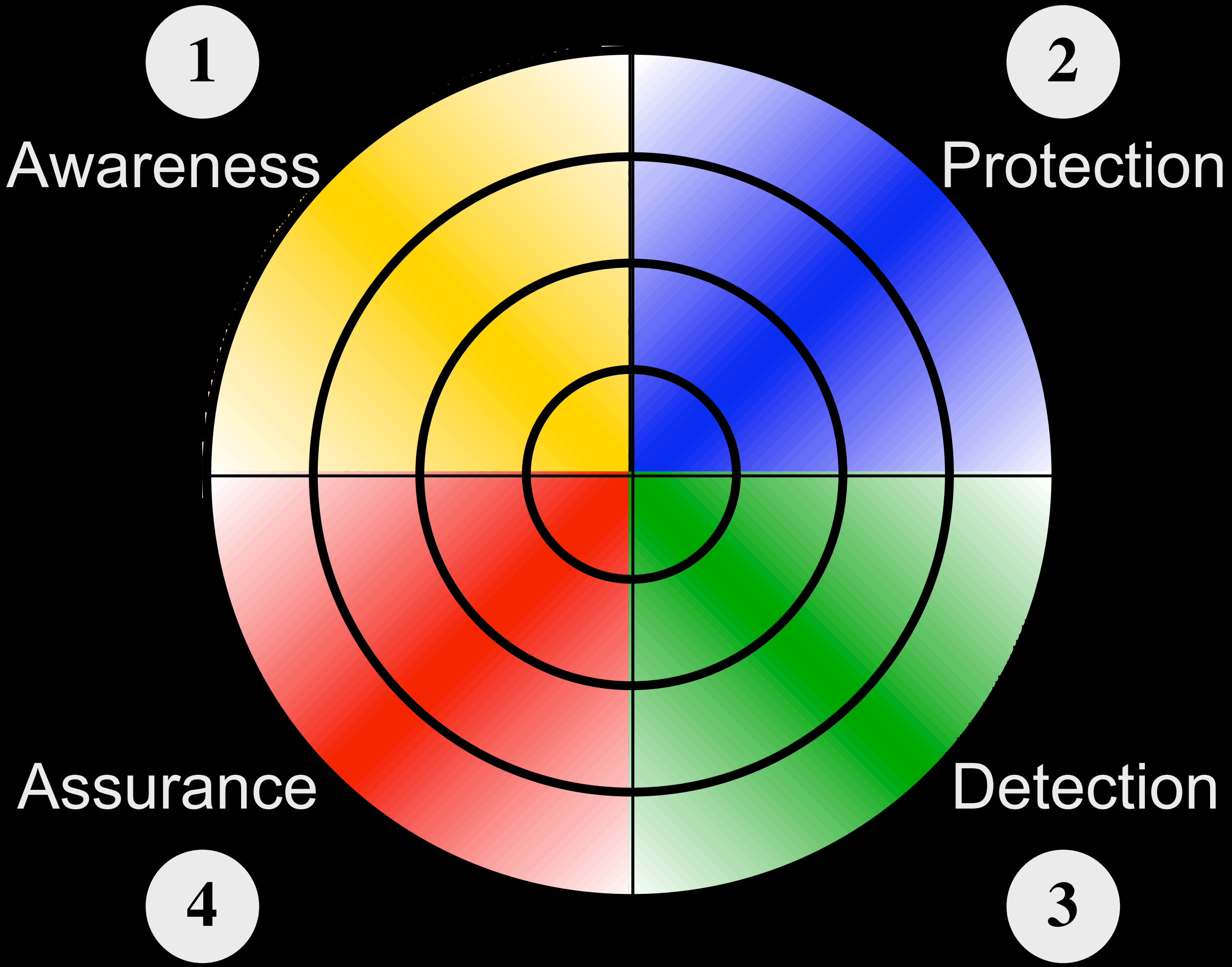
3 Detection

4 Assurance

FoRMA Process: Phases & RM Cycle



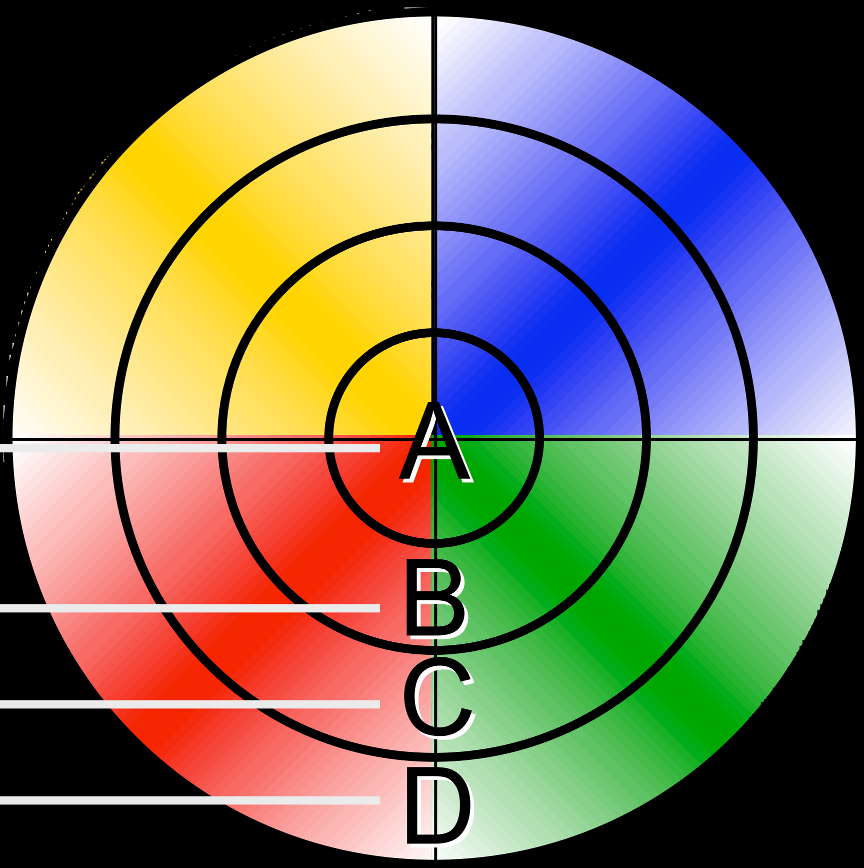
Overview of elements, Quadrants



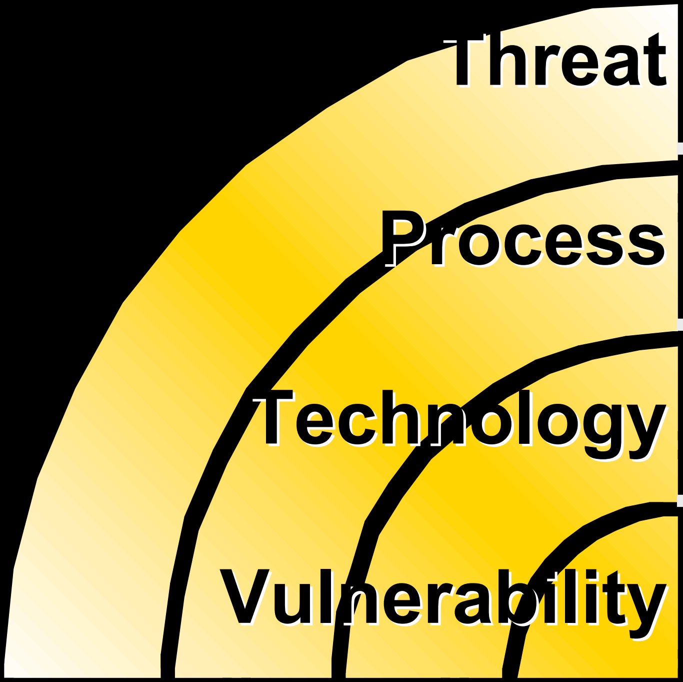
Overview of elements, Rings

- Vulnerabilities

  Of Assets & Resources
- Technology
- Process
- Threats

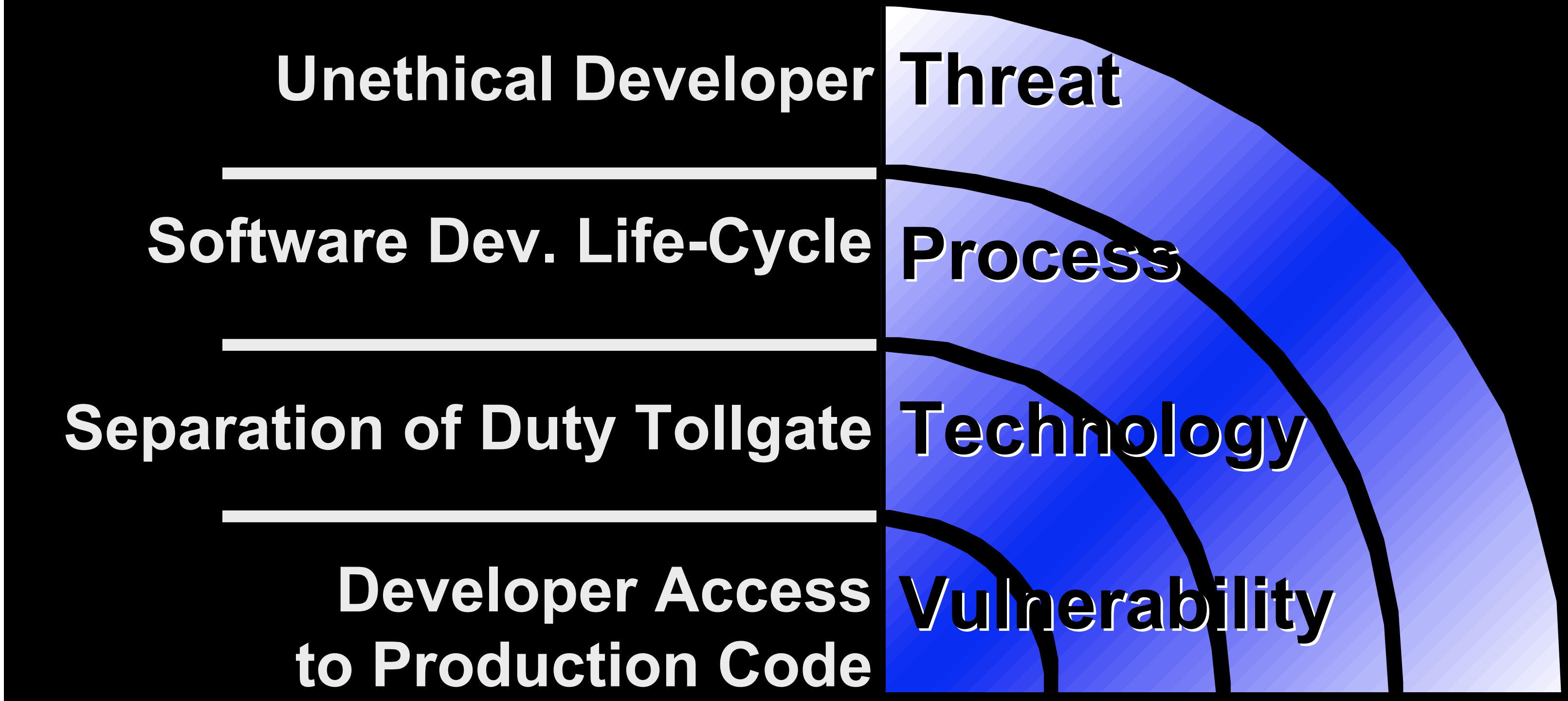


Quadrant 1, Awareness example



Threat	No Security Awareness
Process	Education Program
Technology	Development Standard
Vulnerability	Low Development Expectations

Quadrant 2, Protection example



Unethical Developer Threat

Software Dev. Life-Cycle Process

Separation of Duty Tollgate Technology

Developer Access
to Production Code Vulnerability

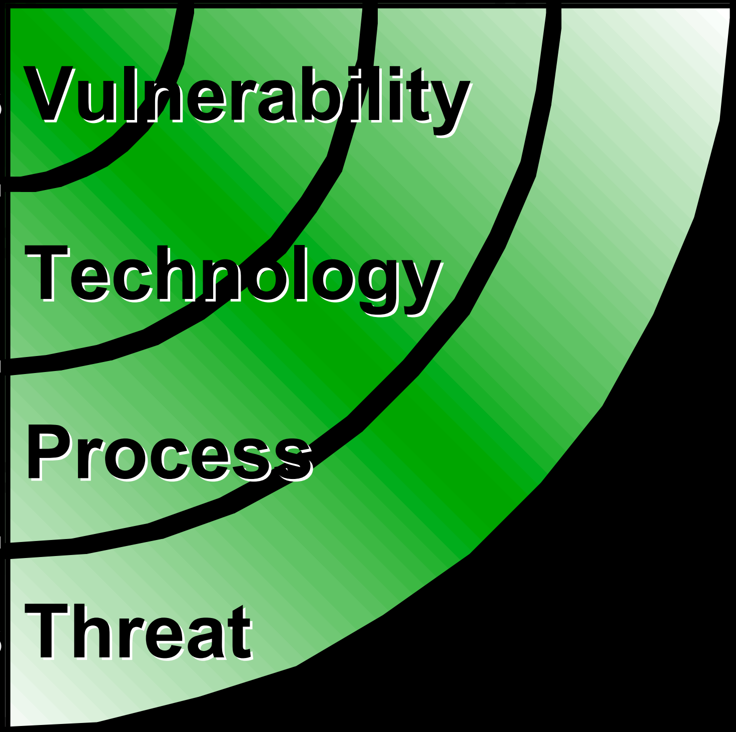
Quadrant 3, Detection example

Unreviewed Promotions

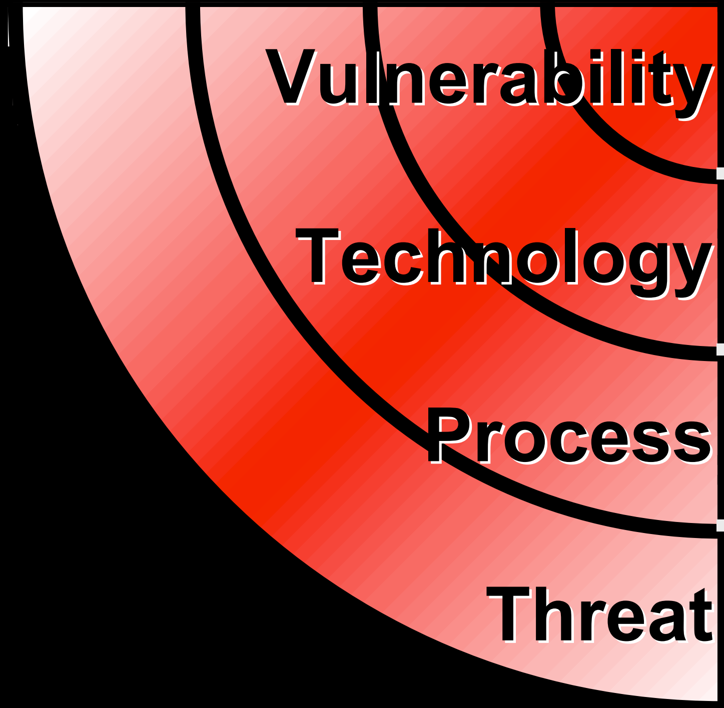
Sanity Checking Tool

Code Review & Approval

Undetected Coding Errors



Quadrant 4, Assurance example



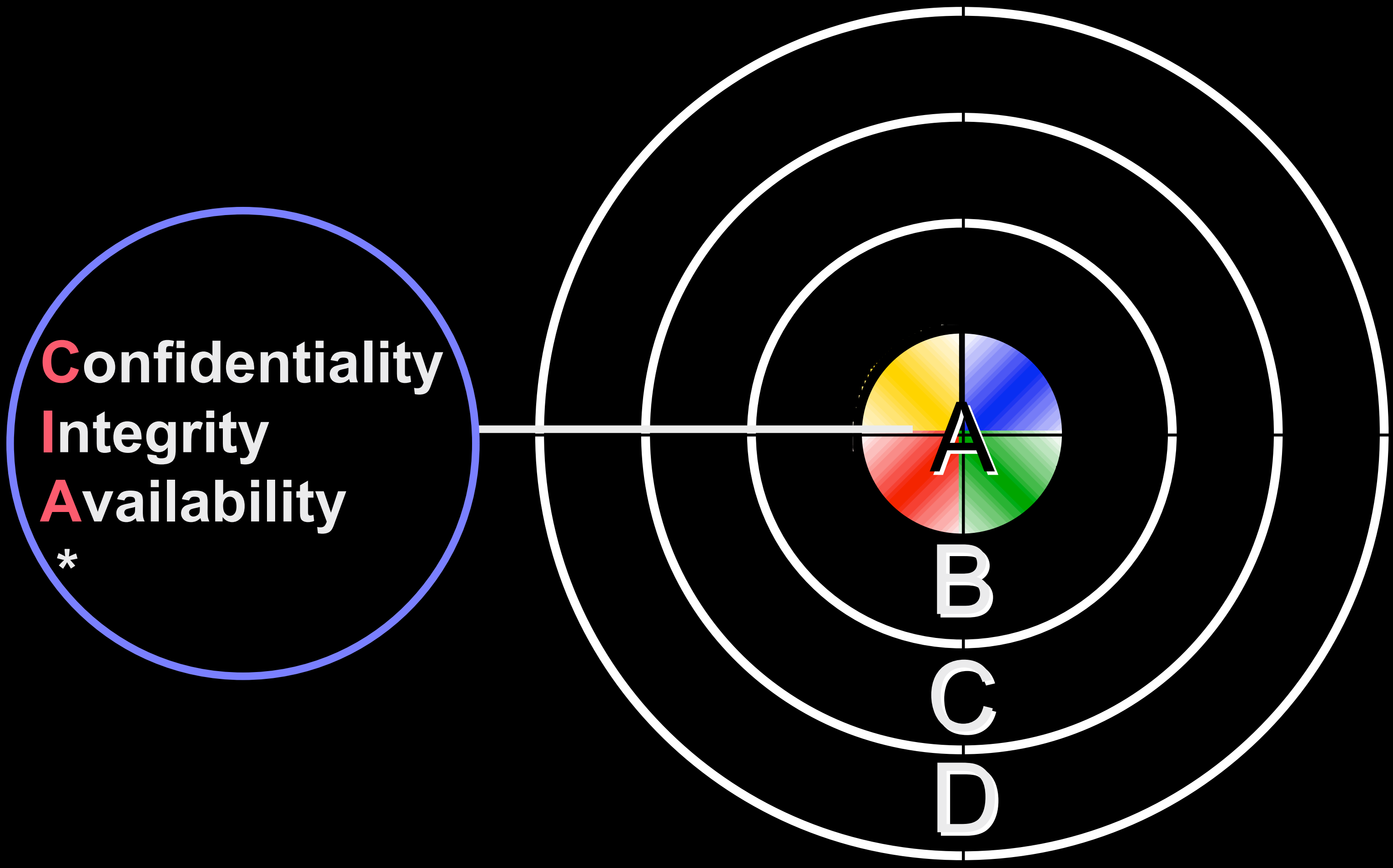
Unstable Application

Binary Analysis Tools, Performance Testing

Quality Assurance Certification

Excessive Load/Utilization

Ring A, Asset/Resource Mgmt

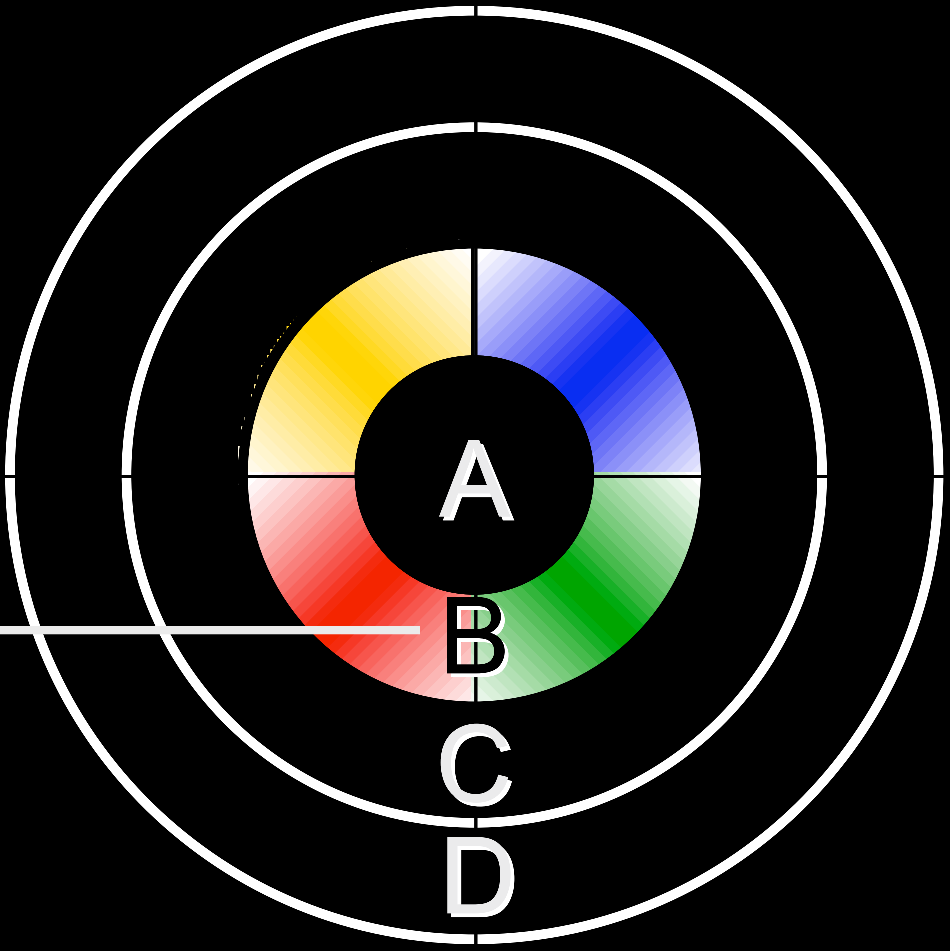


Confidentiality
Integrity
Availability
*

*: See references at the end of the presentation material

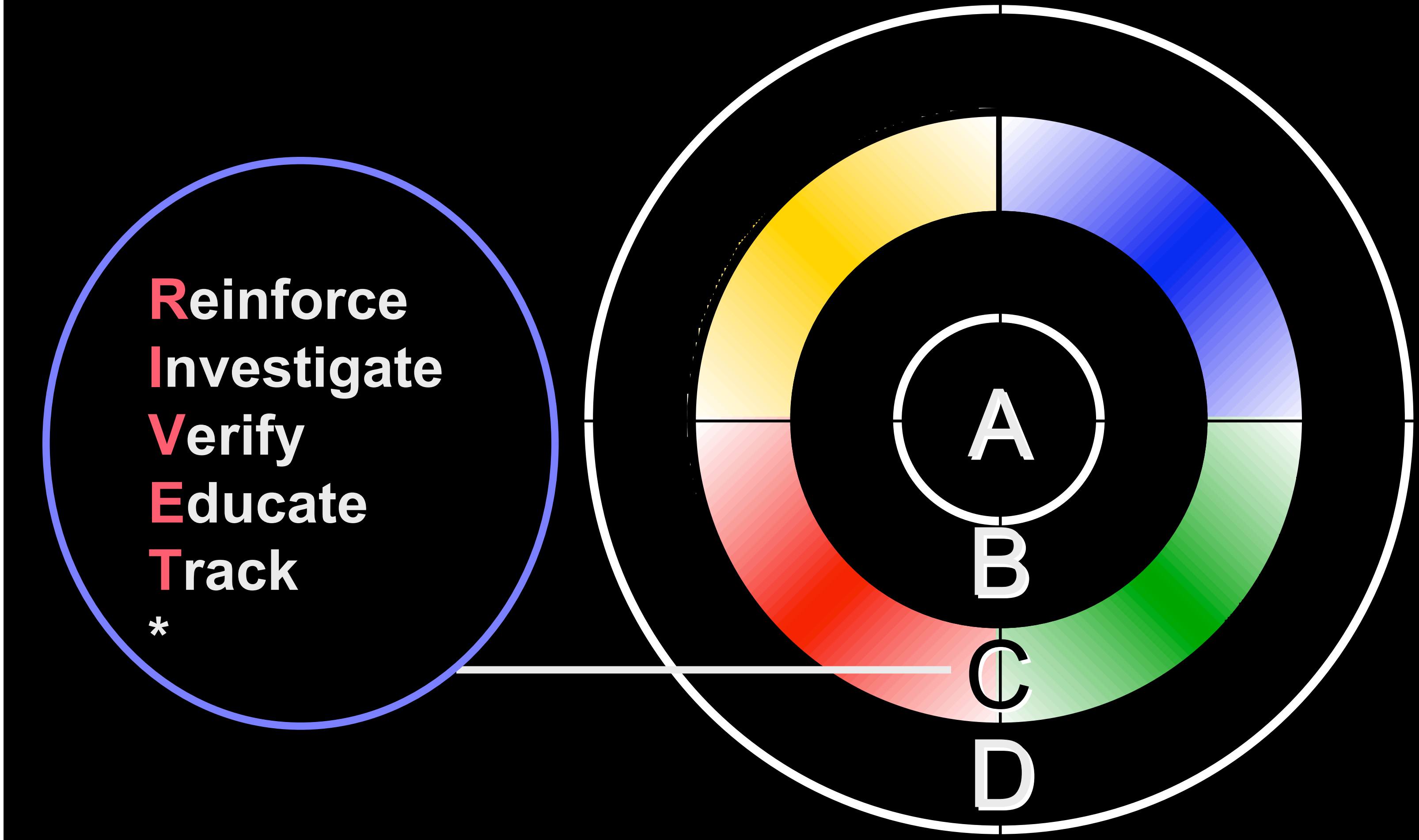
Ring B, Security Architecture

Authentication
Privacy
Authorization
Integrity
Non-repudiation
*



*: See references at the end of the presentation material

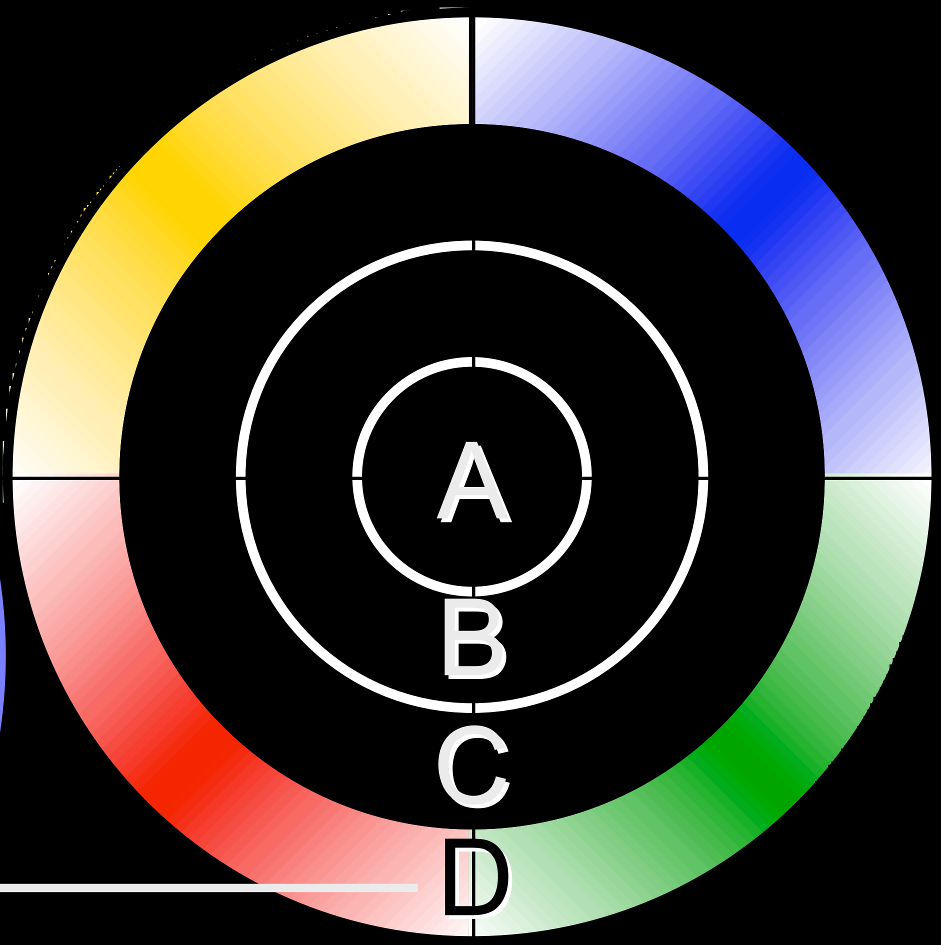
Ring C, Security Management



Reinforce
Investigate
Verify
Educate
Track
*

*: See references at the end of the presentation material

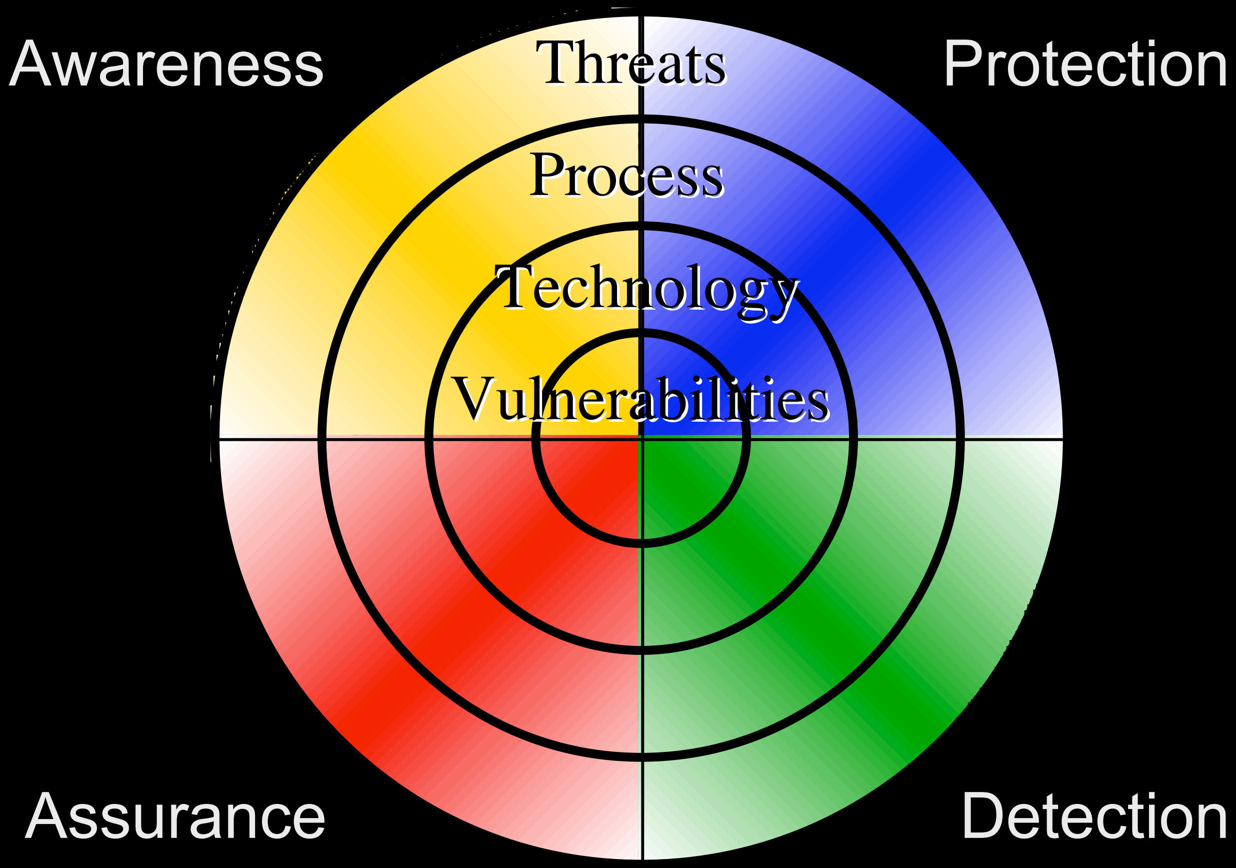
Ring D, Threat Management



Spoof
Tamper
Repudiate
Intercept
Denial-Of-Service
Elevation of
* Privileges

*: See references at the end of the presentation material

FoRMA Model Review

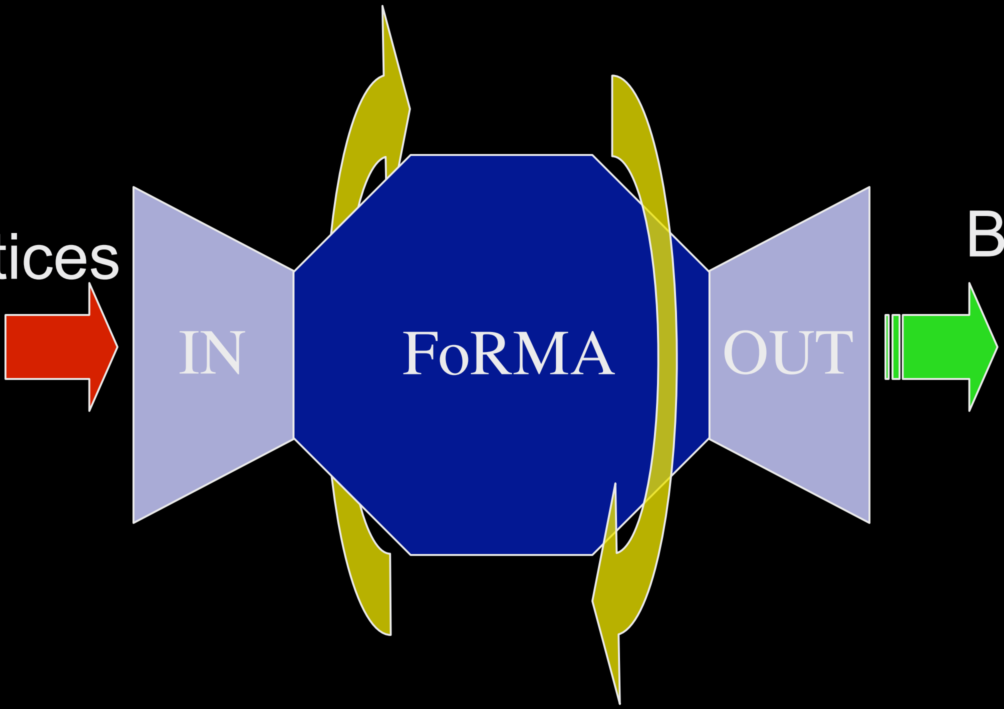


Applying the Model to SDLC

***Before*, Incomplete coverage**

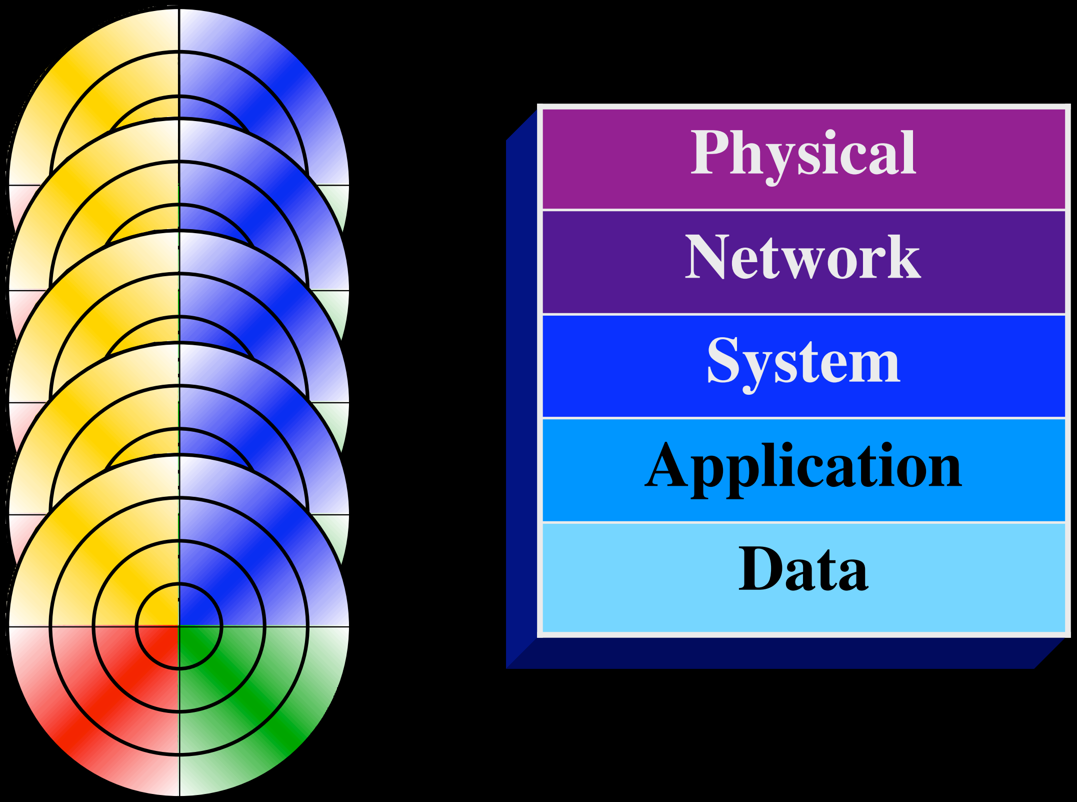
***After*, Improved security**

Coding Practices
Procedures
Concepts



Policies
Best-Practices
Standards
Architectures
Processes

Applied to the 5 Layer model*



Physical

Network

System

Application

Data

*: See references at the end of the presentation material

Results

- Applying the FoRMA model to your current environment will provide a security benefit in the following areas:
  - **Successful Vision & Strategy**
  - **Balanced Technology & Operations**
  - **Performing Security Gap Analysis & Audits**
  - **Demonstrating Reduced and Acceptable Risk**
  - **Trouble Shooting Security Process problems**

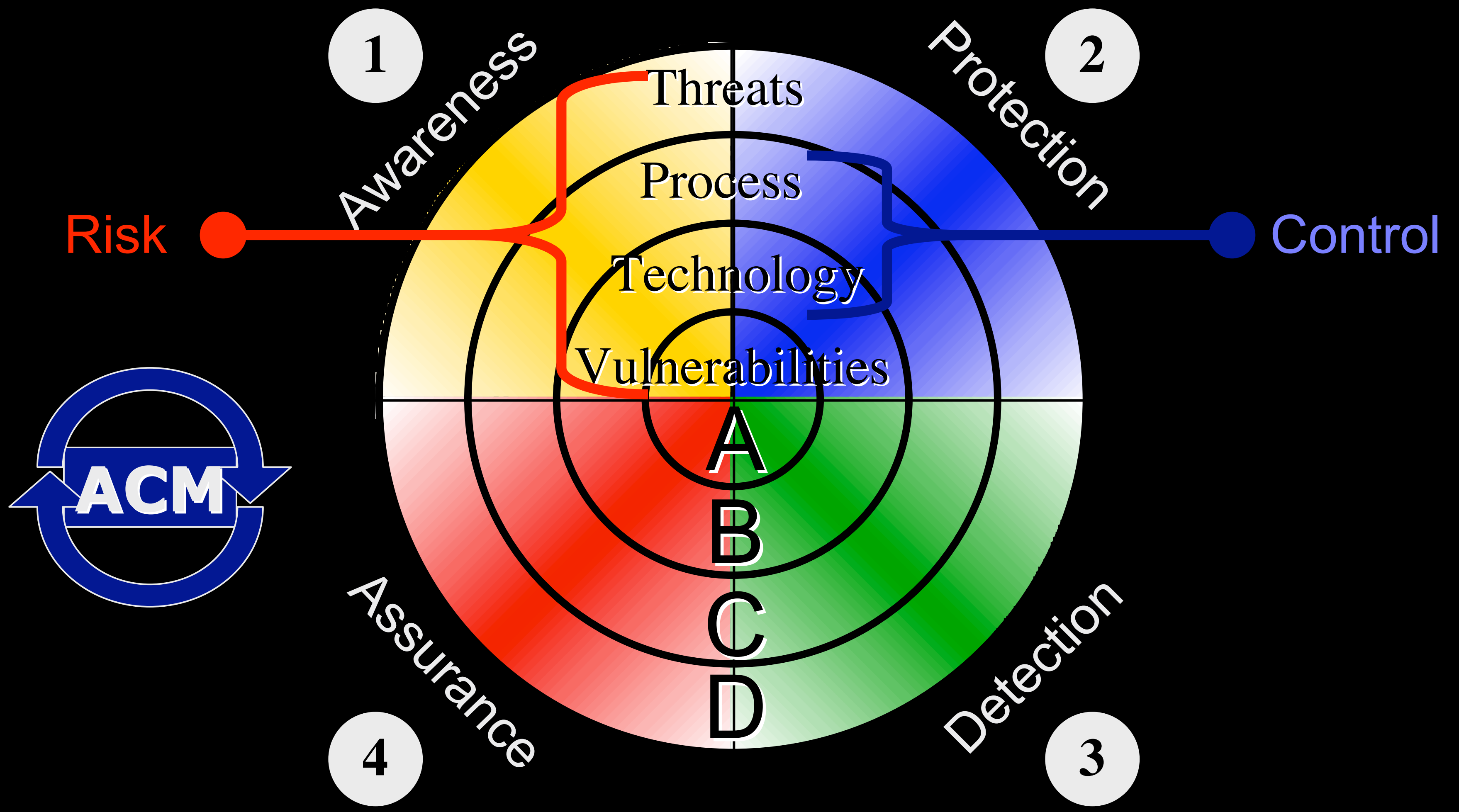
Questions?

- Contact information:
  - □ **Kris.Kahn@mac.com**
  - □ **831-336-5577**

References (*)

- **Open System Interconnection (OSI) reference model** was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered the primary architectural model for intercomputer communications.
- **STRIDE** Threat Model, conceived, built upon, and evangelized at Microsoft by Loren Hohnfelder, Praerit Garg, Jason Garms, and Michael Howard. Explained further in "Writing Secure Code, 2nd Ed" (ISBN 0-7356-1722-8), pages 83-86.
- **CIA** Security Model, author unknown, taught as part of the Common Body of Knowledge for CISSP curriculum.
- **APAIN** Acronym for Security Architecture, developed by Curtis Coleman in 2001.
- **RIVET** Acronym for Security Management, developed by Kris Kahn 2004.

FoRMA Model


1
Awareness
2
Protection
3
Detection
4
Assurance
Risk
Control
Threats
Process
Technology
Vulnerabilities
A
B
C
D
ACM