Architecting Systems-of-Systems for Response

Andrew McDermott and Hart Rossman

June 29, 2009

SAIC
From Science to Solutions

Agenda

- The evolving development environment
- Computer Emergency Response Team role in the system development life cycle and supply chain
- Useful strategies and technical solutions

Baseline Questions

- System development background?
  - Acquisition
  - Design
  - Development
  - Operations and maintenance
  - Disposal

Market Drivers

- What's currently altering the development environment?
  - Regulation and standards
  - Evolution of tools
  - Training requirements and certifications

Regulation

- International Organization for Standardization 27001 and National Institute of Standards and Technology Special Publication 800-64v2
  - Build security into the system throughout development, deployment, operations and maintenance
  - Integrate security into requirements base and functional testing
  - Provides opportunity for incorporating Computer Emergency Response teams into the system development life cycle

Tools

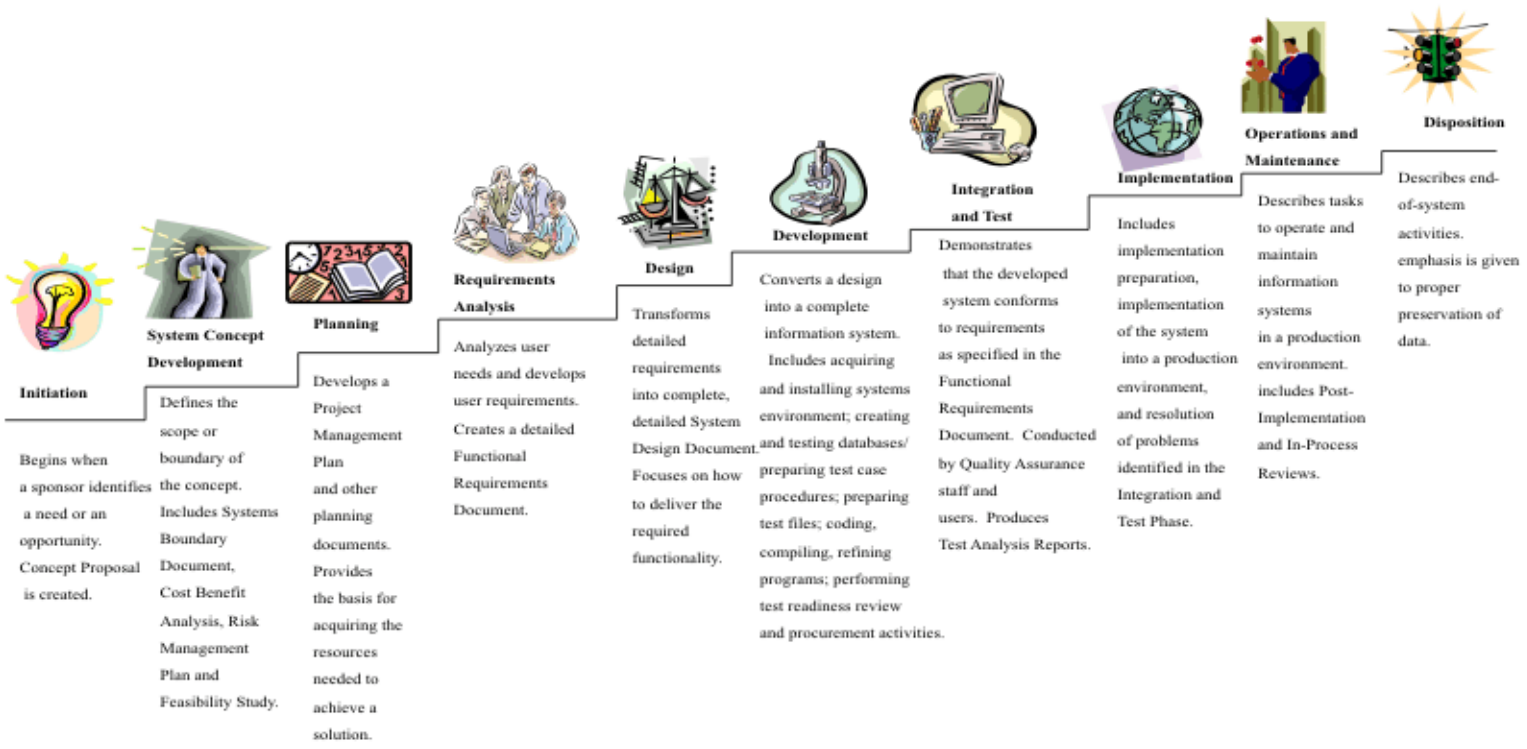
- Current Computer Emergency Response Team tools:
  - Verdasys®
  - Encase®
  - Splunk®
  - Manager of managers (ArcSight®, Tivoli® Netcool®)
- Primarily used during Operations and Maintenance phase
- How can they be employed in other phases of the system development life cycle?

Training/Certification

- Certified Secure Software Lifecycle Professional (CSSLP®)
  - First certification to explicitly address security in the system development life cycle (SDLC) process
- SANS Institute Secure Coding Assessment
  - Focus on specific coding languages rather than SDLC process
- Department of Defense 8570
  - Revised in May 2008, expect future updates with more detailed requirements
  - Currently doesn't require CSSLP for system engineers or developers

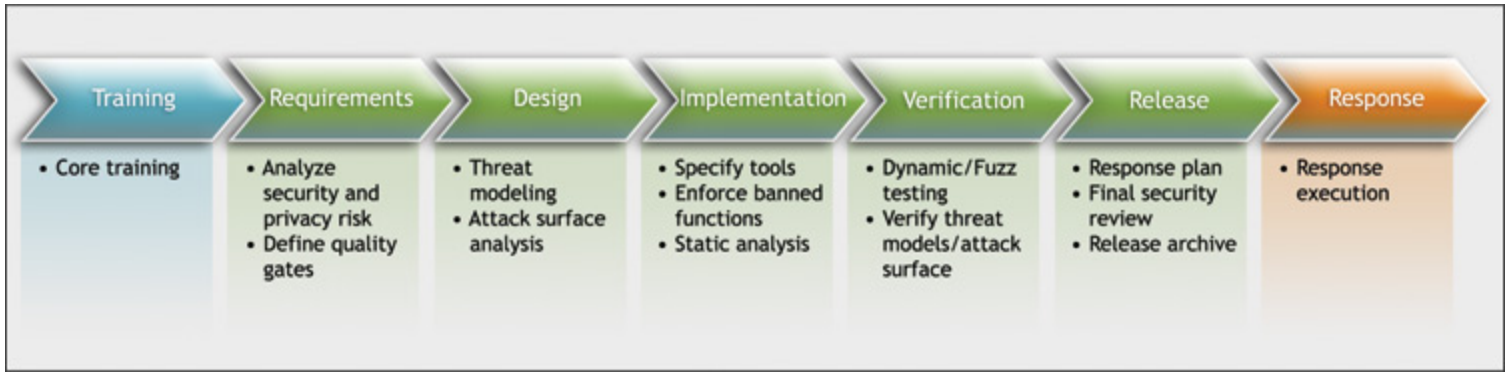
The System Development Life Cycle

Systems Development Life Cycle (SDLC) Life-Cycle Phases



Source: Department of Justice SDLC Guidance

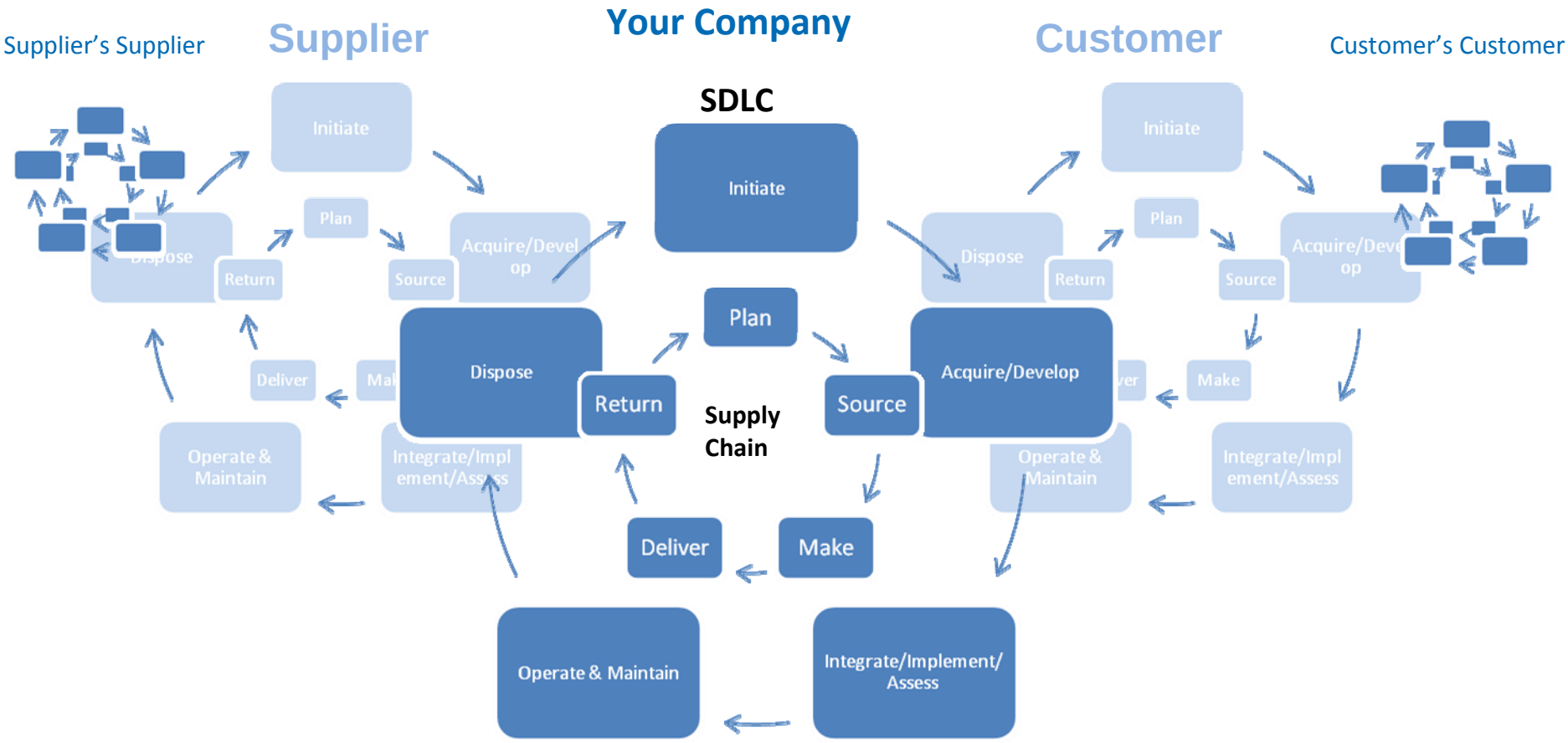
Microsoft Corporation's Security Development Life Cycle



- Integration of security in all phases has lead to markedly more secure products
- Focus on eliminating vulnerabilities during design and coding, not through patches after release

Source: http://msdn.microsoft.com/en-us/security/cc448177.aspx

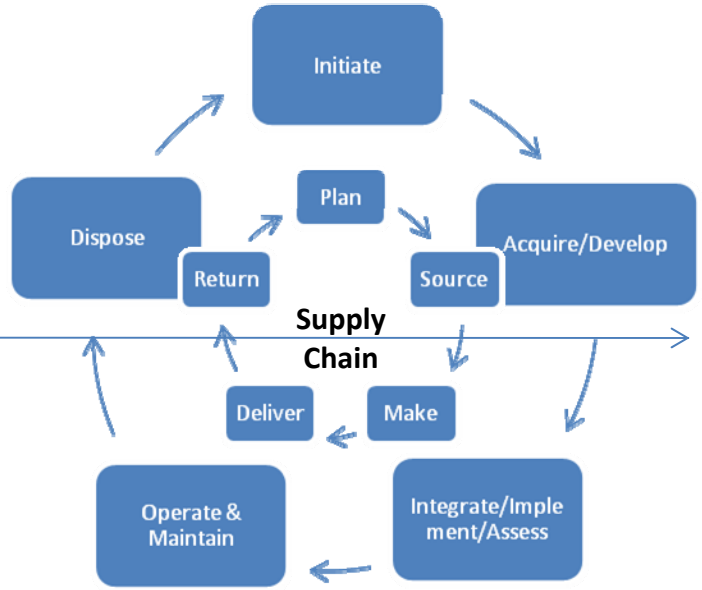
System Development Life Cycle (SDLC)/ Supply Chain Ecosystem



Energy | Environment | National Security | Health | Critical Infrastructure

System Development Life Cycle (SDLC)/ Supply Chain Interdependencies

Supplier		Your Company		Customer	
		SDLC			
SC: Plan	SDLC: Initiate	⟷		SC: Plan	SDLC: Initiate
SC: Source	SDLC: Acquire			SC: Source	SDLC: Acquire
SC: Make	SDLC: Integrate	⟷		SC: Make	SDLC: Integrate
SC: Deliver	SDLC: O&M			SC: Deliver	SDLC: O&M
SC: Return	SDLC: Dispose	⟷		SC: Return	SDLC: Dispose



O&M = Operations and Maintenance

Role of the Computer Emergency Response Team

- Current role
  - Not considered a valid stakeholder
  - Some basic, relevant baseline requirements introduced in initial phases
- Future role
  - Requirements, Design, Operations and Maintenance, Disposal phases
  - Active role in ongoing requirements development (joint application development/rapid application development)

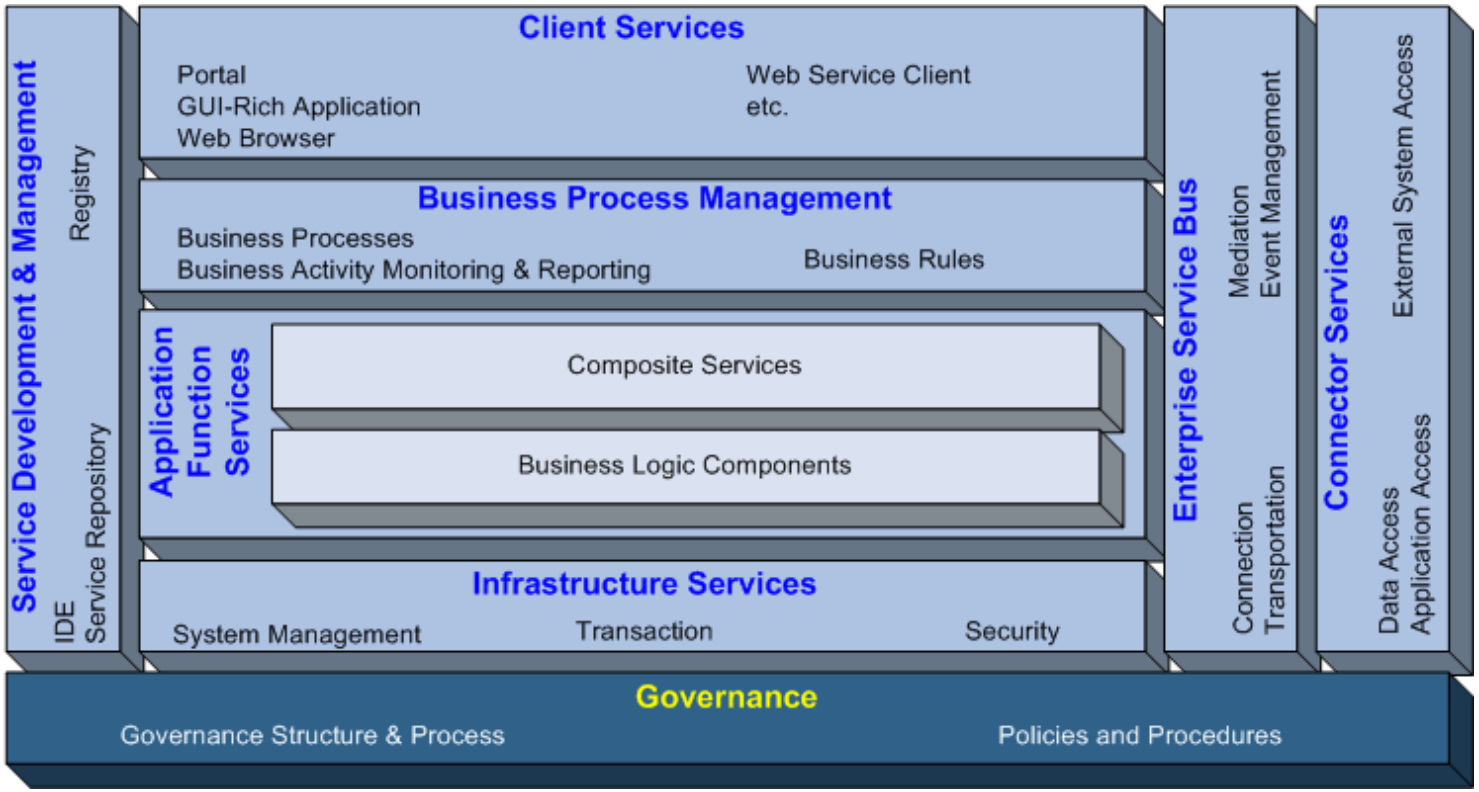
Discussion Questions

- Anyone currently have a role in development of systems they monitor?
- Opportunities to influence the development of monitoring and forensic capabilities?
- Product Security Incident Response Team versus Computer Emergency Response Team functions? How are they related? Where do they overlap?

SAIC Common Service-Oriented Architecture (SOA) Framework

Models are needed to depict a managed architecture

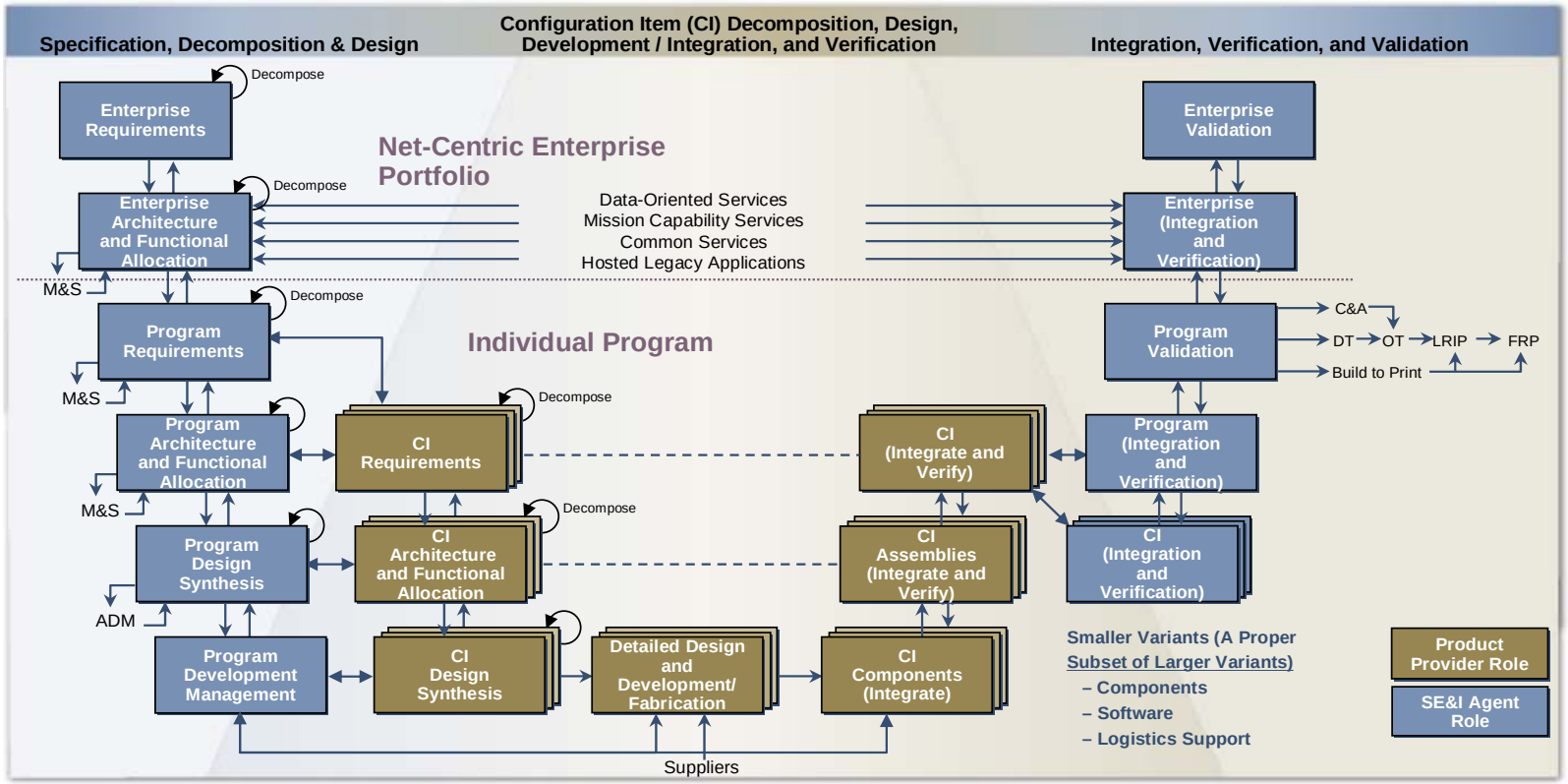
….our experience across many IT programs has helped us evolve and develop a common SOA framework that is "technology-agnostic" while bearing forth distinct layers for consideration on any project.



IDE = Integrated Development Environment GUI = graphical user interface

System-of-Systems Engineering and Integration Agent Role

- **Inner "V's" depict development/integration of each configuration item (CI)**
- **Outer "V" depicts overall systems engineering and integration at the program and portfolio level**
- **Requires innovation in incident response programs too!**



M&S = modeling and simulation C&A = Certification and Accreditation DT = demonstration testing
OT = operational testing LRIP = low-rate initial production FRP = full-rate production
ADM =Architecture Development Method SE&I = system engineering and integration

Reconciling Viewpoints

Requirements development

- Security requirements are functional requirements
- Security requirements of various government standards are bare minimums, yet are generally considered to be all that are necessary to produce a system with adequate/good security
- Secure system-of-system design/architecture requires moving beyond compliance with bare minimum of requirements

Reconciling viewpoints on security requirements

- Difference in opinion between developers, certifiers and the designated approving authority
- Early agreement necessary to avoid costly changes later in the process

Assignment of requirements to system components

- What's satisfied locally by the platform/applications?
- Using external security services?
- Reuse of previously approved solutions

A Solution-Oriented Framework

- Usage model:
  - [Design], configure, generate, validate, [alert, index, search, retrieve], archive
  - An example: Splunk® and grep
  - Another example: Verdasys® and Encase® network

The Building Security In Maturity Model

Ten Core Activities Everybody Does	
Objective	**Activity**
build support throughout organization	create evangelism role/internal marketing
meet regulatory needs or customer demand with a unified approach	create policy
promote culture of security throughout the organization	provide awareness training
see yourself in the problem	create/use material specific to company history
create proactive security guidance around security features	build/publish security features (authentication, role management, key management, audit/log, crypto, protocols)
build internal capability on security architecture	have SSG lead review efforts
drive efficiency/consistency with automation	use automated tools along with manual review
use encapsulated attacker perspective	integrate black box security tools into the QA process (including protocol fuzzing)
demonstrate that your organization's code needs help too	use external pen testers to find problems
provide a solid host/network foundation for software	ensure host/network security basics in place

Three Core Activities that Most Organizations Do	
Objective	**Activity**
understand the organization's history	collect and publish attack stories
meet demand for security features	create security standards
use ops data to change dev behavior	identify software bugs found in ops monitoring and feed back to dev

- Source: www.bsi-mm.com

SSG =

Concluding Discussion

- Possible solutions:
  - Virtualization
  - Service-oriented architecture
  - Cloud