Effectiveness Measures for Continuous Monitoring

Completeness and Timeliness?

How do we make it affordable for things that require manual testing?

U.S. Department of State

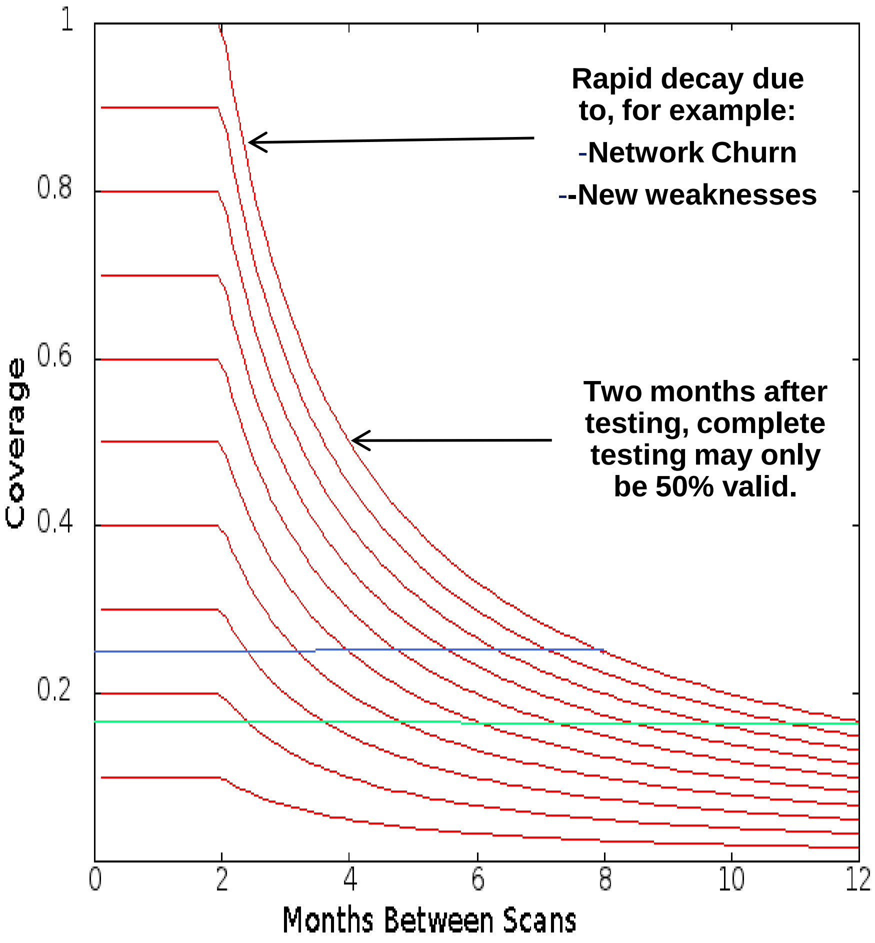
November 2011

Completeness/Timeliness Tradeoff: Part 1

- FISMA 1.0 is based on the assumption that security is founded on the **completeness** of the security program
  - It treats security controls as links in a chain, and if any one link breaks, the chain fails.
- FISMA 1.0 places little emphasis on **timeliness.**
  - OMB A-130 states checking controls every three years is adequate.
  - Adversaries are scanning our networks for weaknesses continually, and at automated speeds.
  - A strong defense requires a rapid response.

Completeness/Timeliness Tradeoff: Part 2

- **MIT Lincoln Labs** is conducting a mathematical modeling study of the tradeoff between complete testing and timeliness. Preliminary results show that:
  - Complete tests, when conducted frequently, are much better than incomplete tests
  - The benefits of “Complete” tests **decay rapidly** over time under any reasonable assumptions.
  - In this example, after one year, the complete test (once a year) is only as effective as a 17% complete test every 2 months. (See green line) Results depend on detailed assumptions.



Rapid decay due to, for example:

-Network Churn

--New weaknesses

Two months after testing, complete testing may only be 50% valid.

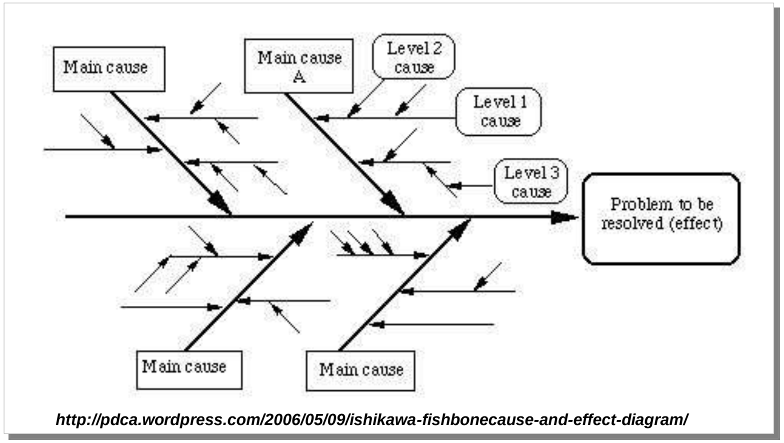
Economic Analysis: What to Test

- State conducted a **notional economic analysis of the cost/benefit of testing and remediation** considering the following parameters:
  - Cost of testing and remediation
  - Cost of not remediating (High value)
  - Probability of failure of the control over time.
- The study concluded that there are two kinds of things to monitor continuously:
  - Things that are ***very cheap to monitor*** (on the margin). For example, vulnerability and configuration checks.
  - Things that are ***very high risk/value*****.** For example, whether an integrated set of controls is working to produce an essential result/outcome.

Transition imperatives:

- **more “timeliness”** in testing priority “things”.
  - **less “completeness”** to pay for timeliness.
- ***Because*** guidance doesn’t define high priority things to test, we **need a model of high priority security OUTCOMES.**

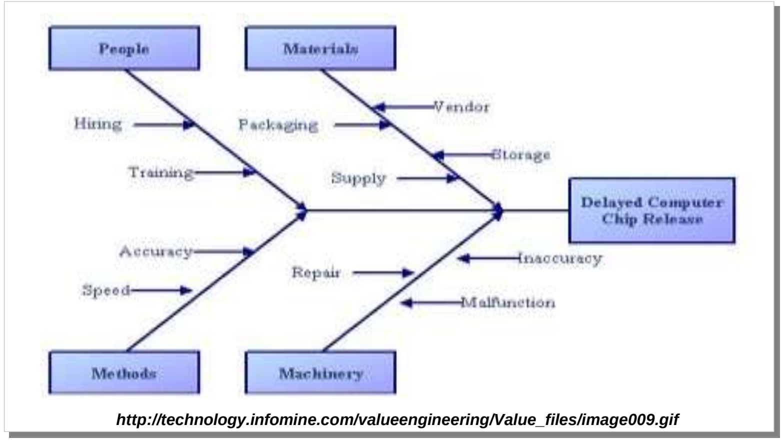
How? Fishbone diagrams



http://pdca.wordpress.com/2006/05/09/ishikawa-fishbonecause-and-effect-diagram/

The Ishikawa diagram (also known as a Fishbone diagram) is a graphical method for finding the most likely causes for an undesired effect.

Kaoru Ishikawa, a famous Japanese consultant developed this method in the 1960s



http://technology.infomine.com/valueengineering/Value_files/image009.gif

Next Step: focus on how to define "high value outcomes"

Methodology background – Part 1

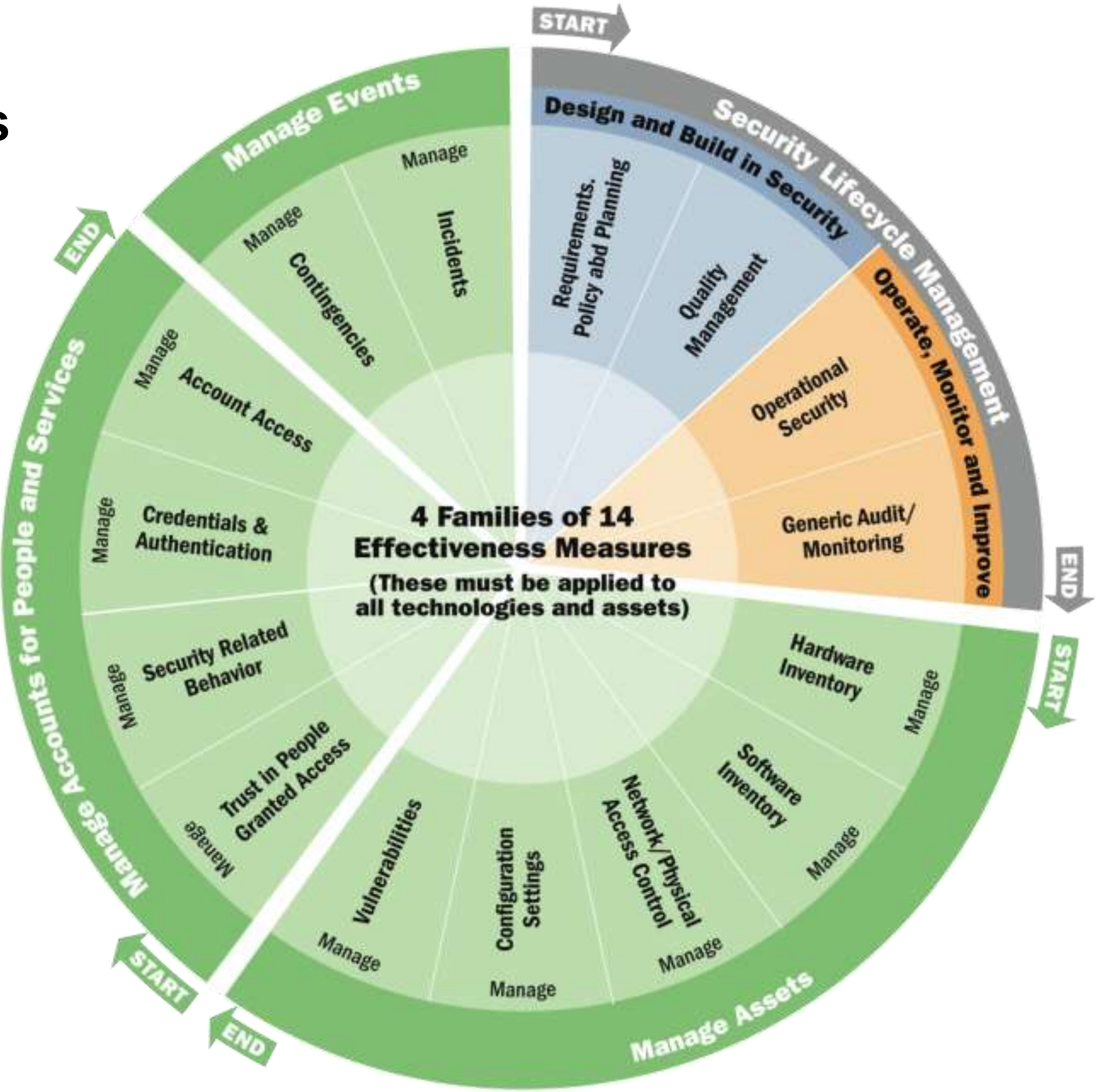
- A CMWG sub-group has developed a set of 15 Effectiveness Measures that cover all NIST 800-53 and CAG/CSC Controls
  - These were developed by starting with the detailed controls and inductively deriving the controls' purpose.
  - For each measure, a fishbone diagram was developed to indicate the main "requirements" to produce the desired effect.

Methodology background – Part 2

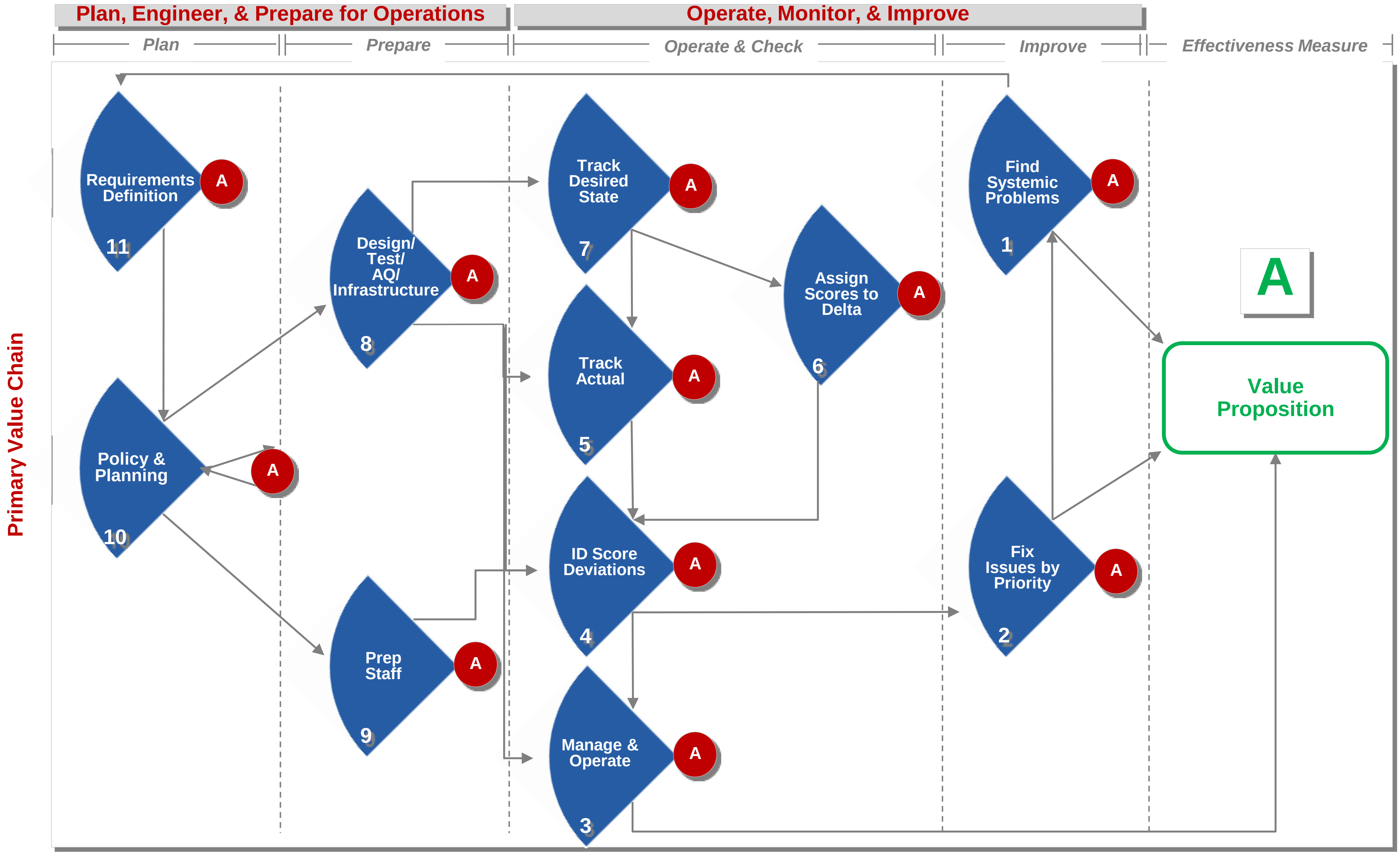
- A CMWG sub-group has developed a set of 15 Effectiveness Measures that cover all NIST 800-53 and CAG/CSC Controls
  - After the steps in the last slide were done the 800-53 controls elements were remapped to the resulting effectiveness measures and fishbone "requirement". This mapping was then reviewed independently by two different SME firms.

Conclusion: Continuous monitoring of the 15 effectiveness measures would fully cover both 800-53 and CAG/CSC.

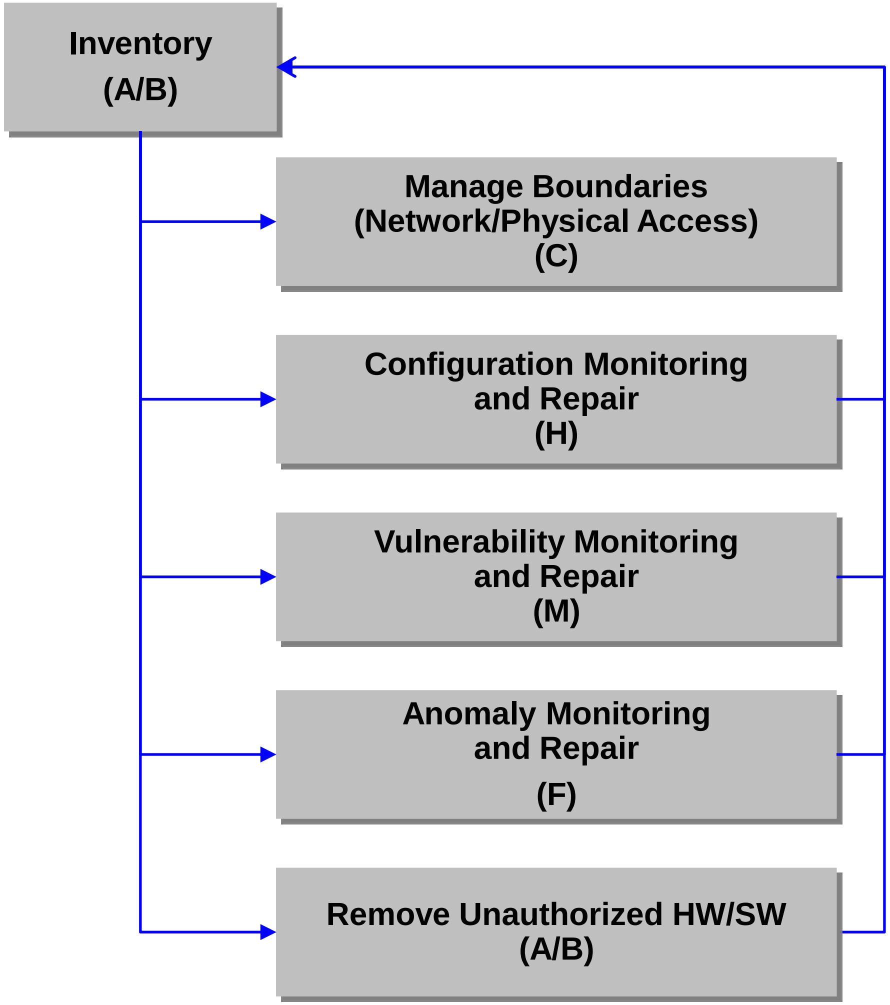
15 Proposed Effectiveness Measures


START
Security Lifecycle Management
Design and Build in Security
Requirements. Policy abd Planning
Quality Management
Operate, Monitor and Improve
Operational Security
Generic Audit/ Monitoring
END
START
Manage Assets
Hardware Inventory
Manage
Software Inventory
Manage
Network/Physical Access Control
Manage
Configuration Settings
Manage
Vulnerabilities
Manage
END
START
Manage Accounts for People and Services
Trust in People Granted Access
Manage
Security Related Behavior
Manage
Credentials & Authentication
Manage
Account Access
Manage
END
Manage Events
Contingencies
Manage
Incidents
Manage
4 Families of 14 Effectiveness Measures
(These must be applied to all technologies and assets)

This Fishbone Diagram works for EACH of the 15 Areas



The Hardware/Software Family



Inventory
(A/B)

Manage Boundaries
(Network/Physical Access)
(C)

Configuration Monitoring
and Repair
(H)

Vulnerability Monitoring
and Repair
(M)

Anomaly Monitoring
and Repair
(F)

Remove Unauthorized HW/SW
(A/B)

The Hardware/Software Family
Monitor Inventory – Step 1

Establish Responsibility

Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Capability Statement
A – Manage Hardware	Attackers continually scan for new, unprotected systems, including test or experimental systems, and exploit such systems to gain control of them.	**Manage Hardware Inventory** Remove unauthorized hardware with X hours to prevent attackers from • Gaining control of those systems.

The Hardware/Software Family
Monitor Inventory – Step 2

Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Value Statement
B – Manage Software	- Attackers continually scan for vulnerable software and exploit it to gain control of target machines. - Attackers distribute hostile content on Internet-accessible (and sometimes internal) websites that exploits unpatched and improperly secured client software running on victim machines. - Attackers use currently infected or compromised machines to identify and exploit other vulnerable machines across an internal network.	**Manage Software Inventory** Remove unauthorized software with X hours to prevent attackers from: • Exploiting vulnerable software (for example, placed there innocently by insiders to perform work without adequately addressing security). • exploiting unpatched and improperly secured software • Using the software to exploit other vulnerable machines across the internal network.

The Hardware/Software Family
Monitor Vulnerabilities and Configurations – Step 3a

Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Value Statement
H – Manage Configurations (CCEs)	- Attackers exploit weak default configurations of systems that are more geared to ease of use than security. - Attackers exploit and infiltrate through network devices whose security configuration has been weakened over time by granting, for specific short-term business needs, supposedly temporary exceptions that are never removed. - Attackers scan for remotely accessible services on target systems that are often unneeded for business activities, but provide an avenue of attack and compromise of the organization.	**Manage Configuration Settings** Prevent weaknesses from weak configuration settings (including port, protocols, and services) by defining an appropriate desired operational state for these settings and maintaining it in operation, thereby preventing attackers from: • Exploiting preventable configurational weaknesses.

The Hardware/Software Family
Monitor Vulnerabilities and Configurations – Step 3b

Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Value Statement
M – Manage Vulnerabilities (CVEs)	Attackers exploit new vulnerabilities on systems that lack critical patches in organizations that do not know that they are vulnerable because they lack continuous vulnerability assessments and effective remediation.	**Manage Known Vulnerabilities** Prevent vulnerabilities (for example, CVEs from the National Vulnerability Database) by finding and removing such vulnerabilities, thereby preventing attackers from: • Exploiting preventable vulnerabilities

The Hardware/Software Family
Monitor Boundaries – Step 4a

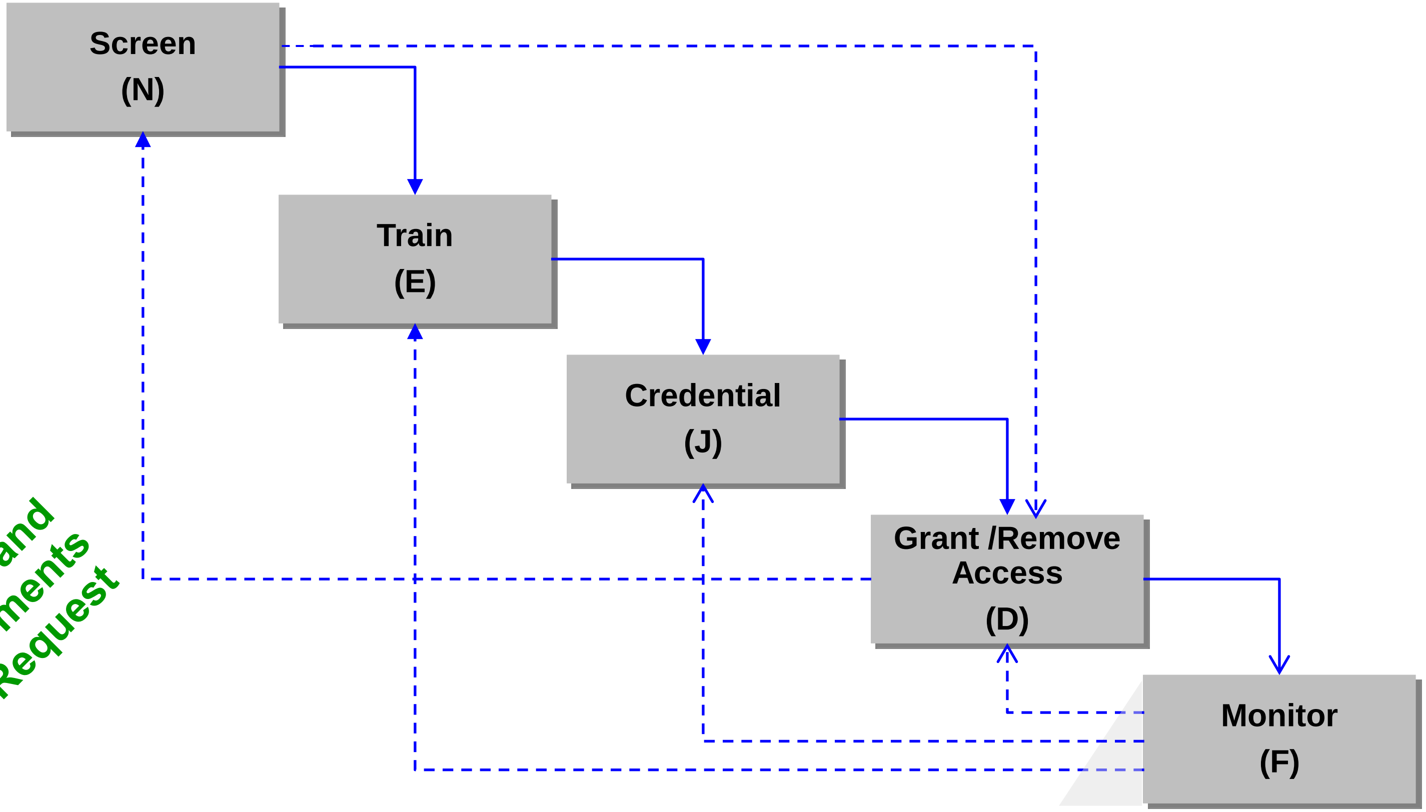
Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Value Statement
C – Manage Network Access	Attackers exploit boundary systems on Internet-accessible DMZ networks (and on internal network boundaries), and then pivot to gain deeper access on internal networks.	**Manage Network Access** Prevent, remove and limit unauthorized network connections/access to prevent attackers from: • exploiting internal and external network boundaries and then pivoting to gain deeper network access and/or capture network resident data in motion or at rest. Note: Boundaries include things like firewalls, but also encryption (as in VPNs).

The Hardware/Software Family
Monitor Boundaries – Step 4b

Area	Attack Scenarios (adapted from 800-53 and the CSC Version 3.0)	Value Statement
C – Manage Physical Access	Attackers exploit physical boundaries to gain access to facilities, networks, etc. and then pivot to gain deeper access to, or cause harm to those resources and/or data.	**Manage of Physical Access** Prevent, remove and limit unauthorized physical access, and to prevent attackers from: • exploiting that access and then pivoting to gain deeper access to, or cause harm to those resources and/or data.

Manage Accounts (for people and services)

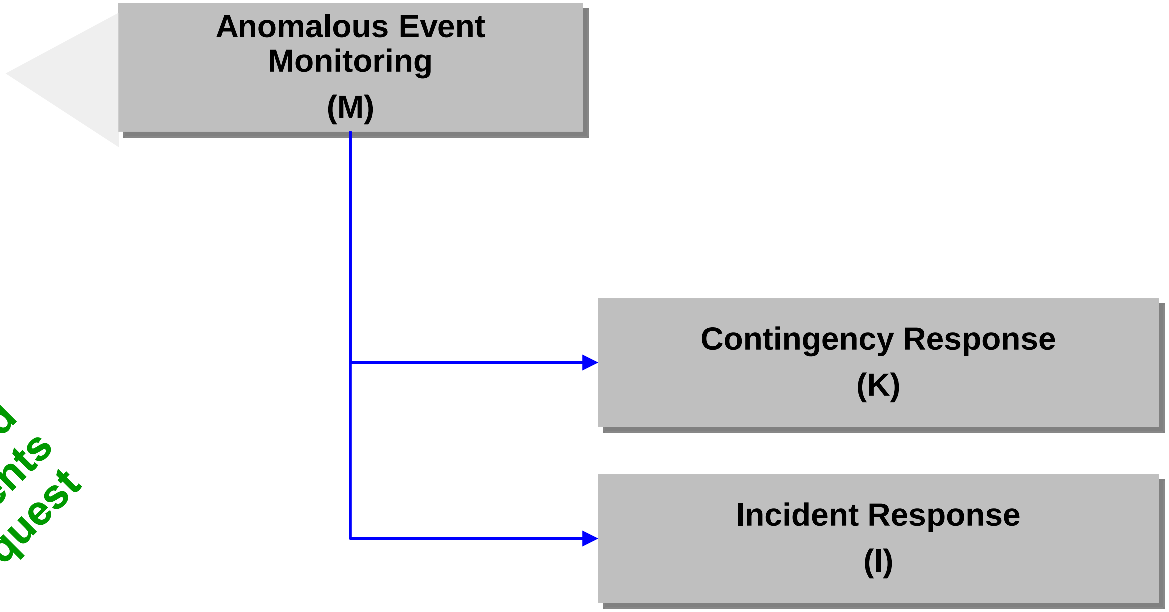
Interactions of Effectiveness Measures



Attack Scenarios and Capability Statements Available on Request

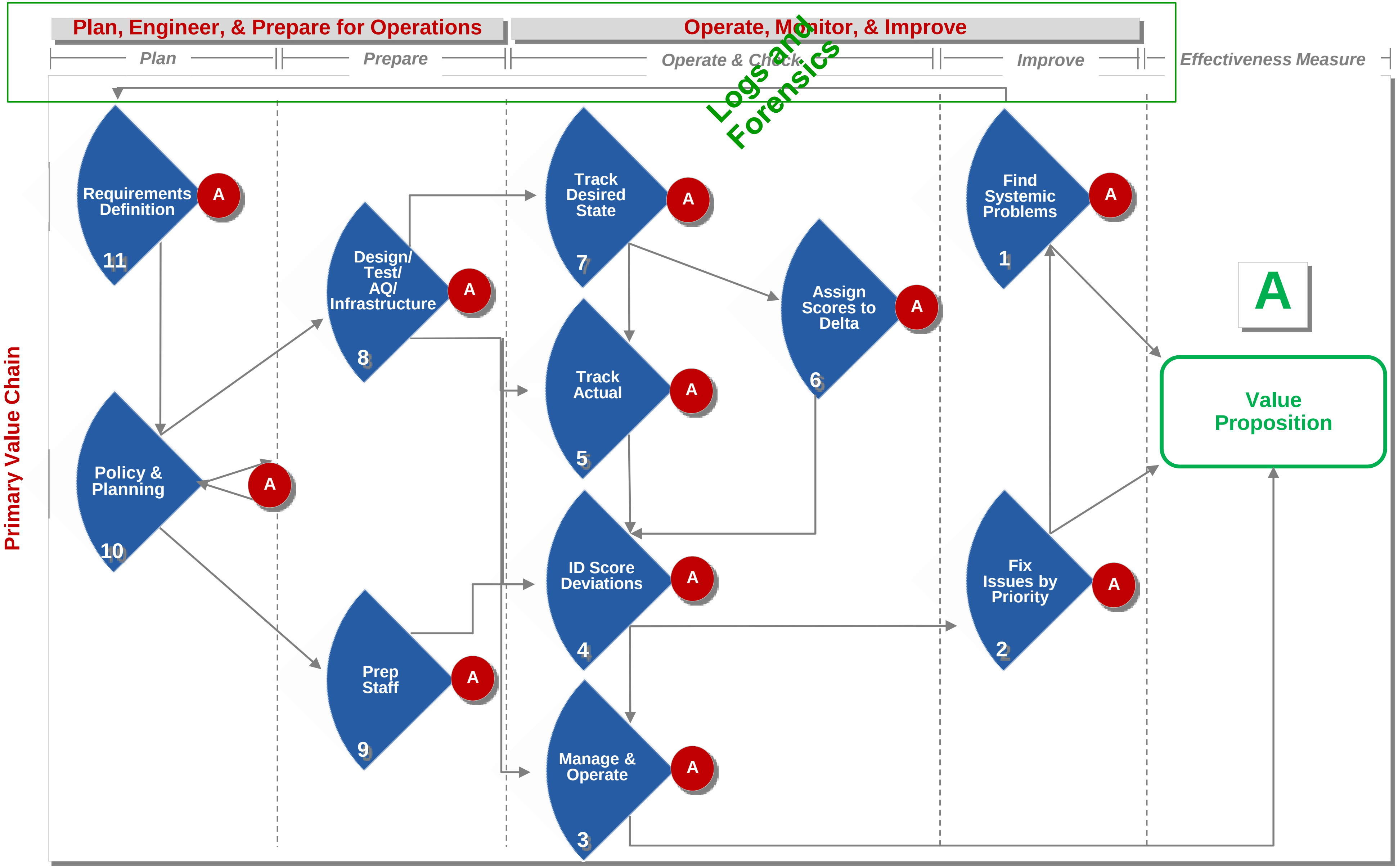
Manage Events

Interactions of Effectiveness Measures



Attack Scenarios and Capability Statements Available on Request

The Lifecycle Outcomes fit Over the Operational Fishbones



Audit Log Management
is Key to All Other Areas

- Audit Logs and Forensics apply to all operational families
  - Hardware/Software Behavior
  - Account Behavior
  - Events (Contingencies and Incidents)
- Integration of this log data across the enterprise is essential **at later stages of maturity.**

Coverage Model

	Technologies and Assets											
4 Families of 15 Effectiveness Measures **(These must be applied to all technologies and assets)**	Networks	Applications	Data	People	Wireless	Cloud	Maintenance	Media	Physical	Environmenta.	Malware	Etc…………
Security Lifecycle Management: Design and Build in Security *Requirement, Policy and Planning (L)* *Quality Management (G1)* Operate, Monitor and Improve *Operational Security (G2)* *Generic Audit/Monitoring (F)*												
Manage Hardware and Software Assets *Manage Hardware Inventory (A)* *Manage Software Inventory (B)* *Manage Network /Physical Access Control (C)* *Manage Configuration Settings (H)* *Manage Vulnerabilities (M)*												
Manage Accounts for People and Services *Manage Trust in People Granted Access (N)* *Manage Security Related Behavior (E)* *Manage Credentials & Authentication (J)* *Manage Account Access (D)*												
Manage Events *Manage Contingencies (I)* *Manage Incidents (K)*												

Measures apply across technology.

Measures should be technology agnostic

Conclusions:

- Improved security strategy should support a risk-based **tradeoff between completeness and timeliness** of testing/remediation.
- Recent modeling by MIT shows that even incomplete/timely testing and remediation can be as effective as complete/untimely testing.
- We should test things that are
  - relatively cheap to test and
  - things that are of high security risk/value.
- Testing the 15 proposed effectiveness measures offers:
  - the best available set of high value security outcomes for “complete” monitoring
  - of 800-53 “controls” using event driven testing
  - at a reasonable cost .