“How Security and Quality ‘Mesh’ within the SDLC”

Rhonda Farrell
ASQ SW SIG 509
March, 2011

Agenda

- Problem Space
- SDLC Phases
- Project Methodologies
- Quality Programs
- Risk Management
- Security within the Lifecycle
- Importance of “Decision Gates”
- Methods of Gauging Readiness
- Utilization of Resources to Enhance Quality, Cut Costs, and Improve Productivity
- References

Poll Question 1

How many think that quality can be effectively “tested in”?

Poll Question 2

How many think that security can effectively be “bolted on” ?

Problem Space

- Disenfranchised management and employee base
- Product and service quality levels much lower than those which are actually attainable
- Enterprise liability due to customer or internally found security issues
- Constant “fire-fights” versus “well-planned”, attainable project progression
- Increased costs due to schedule delays, excessive re-work, contractual liability

Premise

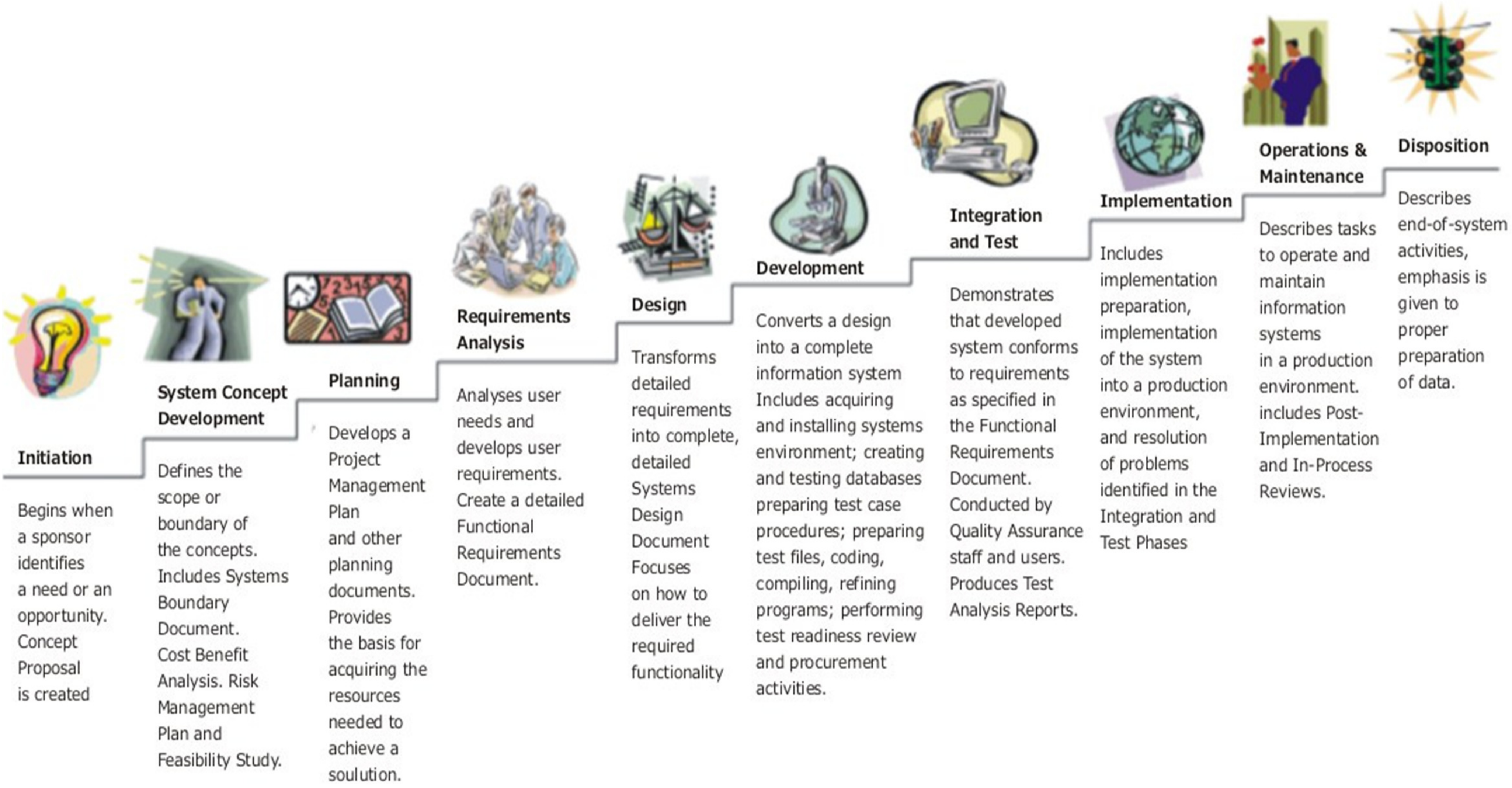
In order to maximize resources and reach successful project outcomes, personnel must consider the organization holistically, thus both ***quality*** *and* ***security*** *must be taken into consideration during the systems development life-cycle. Each and every person must be empowered to be* <u>*responsible*</u> *and* <u>*accountable*</u> *for these initiatives within the enterprise.*

SDLC Phases

Focus Point

In order to allocate and utilize resources effectively, the teams must understand what comprises a project cycle and how their particular skills can be utilized across the separate phases to best optimize actual outcomes.

Systems Development Life Cycle (SDLC) Phases



Initiation

Begins when a sponsor identifies a need or an opportunity. Concept Proposal is created

System Concept Development

Defines the scope or boundary of the concepts. Includes Systems Boundary Document. Cost Benefit Analysis. Risk Management Plan and Feasibility Study.

Planning

Develops a Project Management Plan and other planning documents. Provides the basis for acquiring the resources needed to achieve a soulution.

Requirements Analysis

Analyses user needs and develops user requirements. Create a detailed Functional Requirements Document.

Design

Transforms detailed requirements into complete, detailed Systems Design Document Focuses on how to deliver the required functionality

Development

Converts a design into a complete information system Includes acquiring and installing systems environment; creating and testing databases preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

Integration and Test

Demonstrates that developed system conforms to requirements as specified in the Functional Requirements Document. Conducted by Quality Assurance staff and users. Produces Test Analysis Reports.

Implementation

Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phases

Operations & Maintenance

Describes tasks to operate and maintain information systems in a production environment. includes Post-Implementation and In-Process Reviews.

Disposition

Describes end-of-system activities, emphasis is given to proper preparation of data.

Project Methodologies

Focal Point

Once the specific system development phases are agreed upon and published, a program/project methodology should then be selected, which is applicable to the task at hand and enables the teams to successfully achieve their overall objectives.

Oftentimes organizations select and extract those components of a methodology which are most readily applicable to their specific environment. These are then combined with best practices, additional frameworks, and associated methodologies, in order to design one that best allows for successful outcomes to be achieved, given specific project specific constraints.

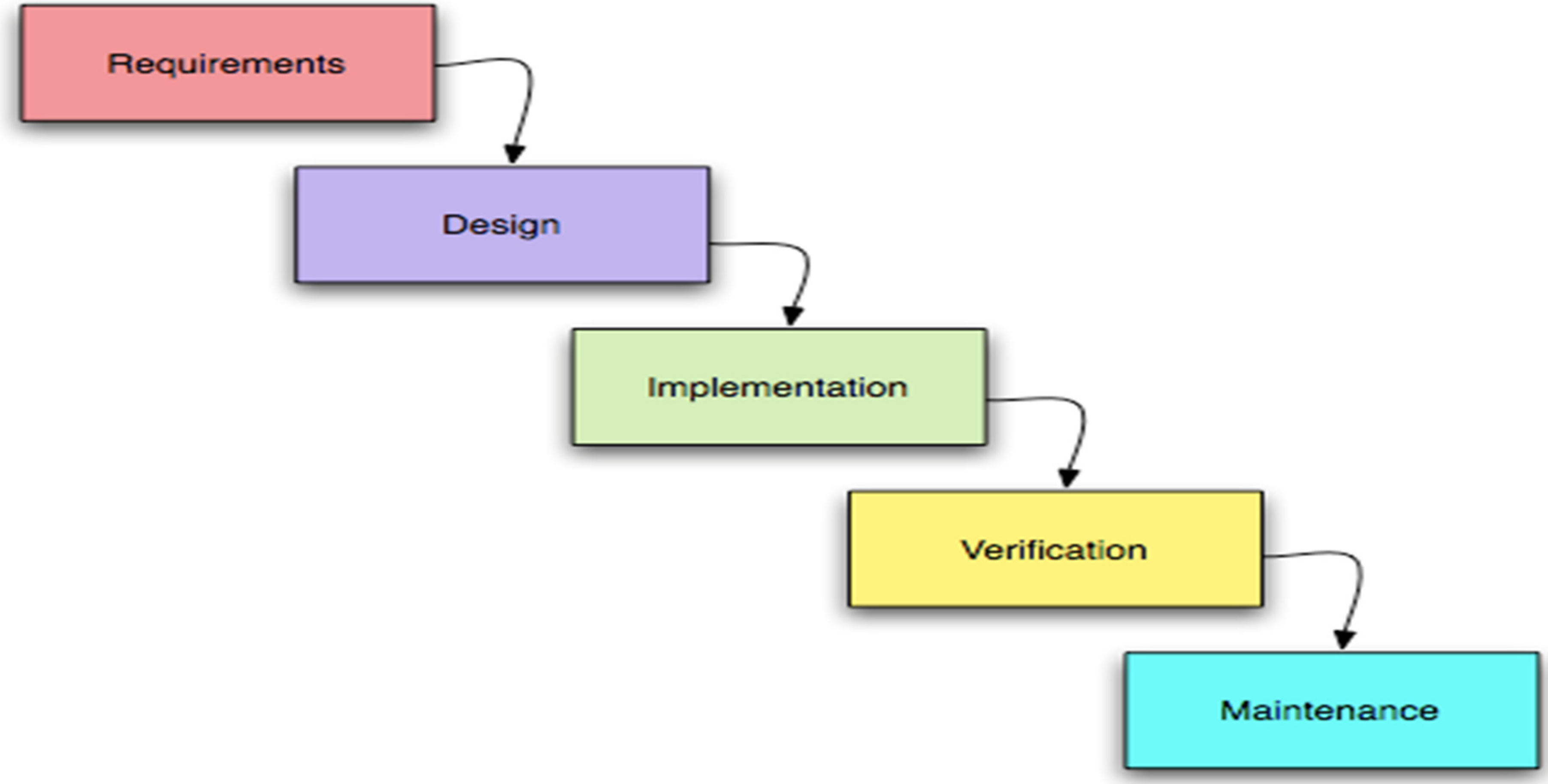
Project Methodologies

- **Plan Driven**
  - Systematic approach according to a documented methodology
    - Typical life cycle orientation: requirements, design, build, test, and deploy framework
    - Focus on standardization, traceability, and verification activities
    - Strengths: Predictability, stability, repeatability, high-assurance, process improvement
      - ❖ Waterfall
      - ❖ Spiral
      - ❖ Vee
- **Iterative or Mixed**
  - Used when requirements are not finalized up front, focuses on rapid value and responsiveness
    - Agile
    - Rapid Development/Prototyping
- **Lean, Lean SE**
  - Based on Just In Time (JIT) system
  - Maximum value (mission assurance) to stakeholders and removal of waste from the process

Plan Driven

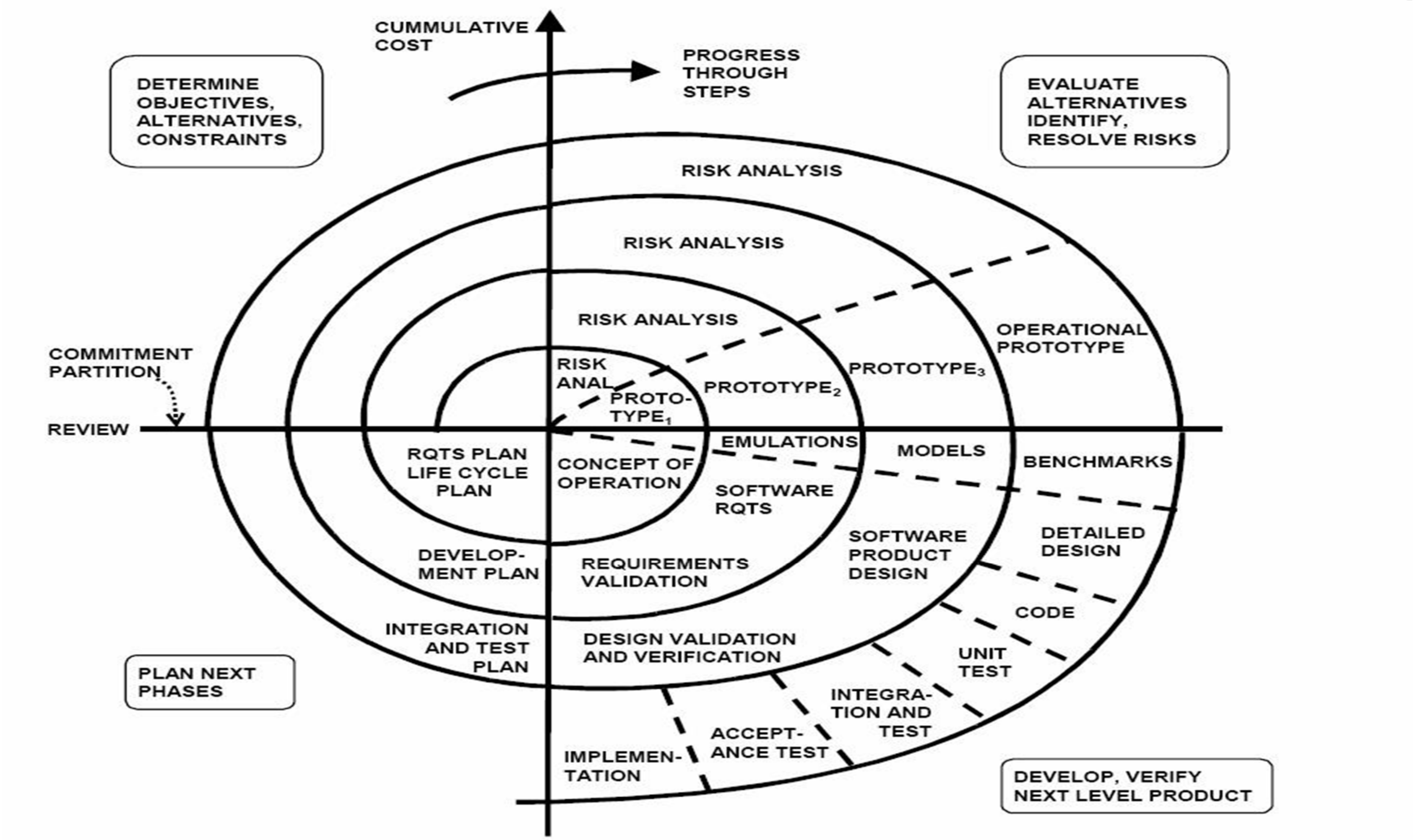
Waterfall

- SEQUENTIAL SDLC process which has formalized phase entry and exit criteria - typically termed a “top down” approach



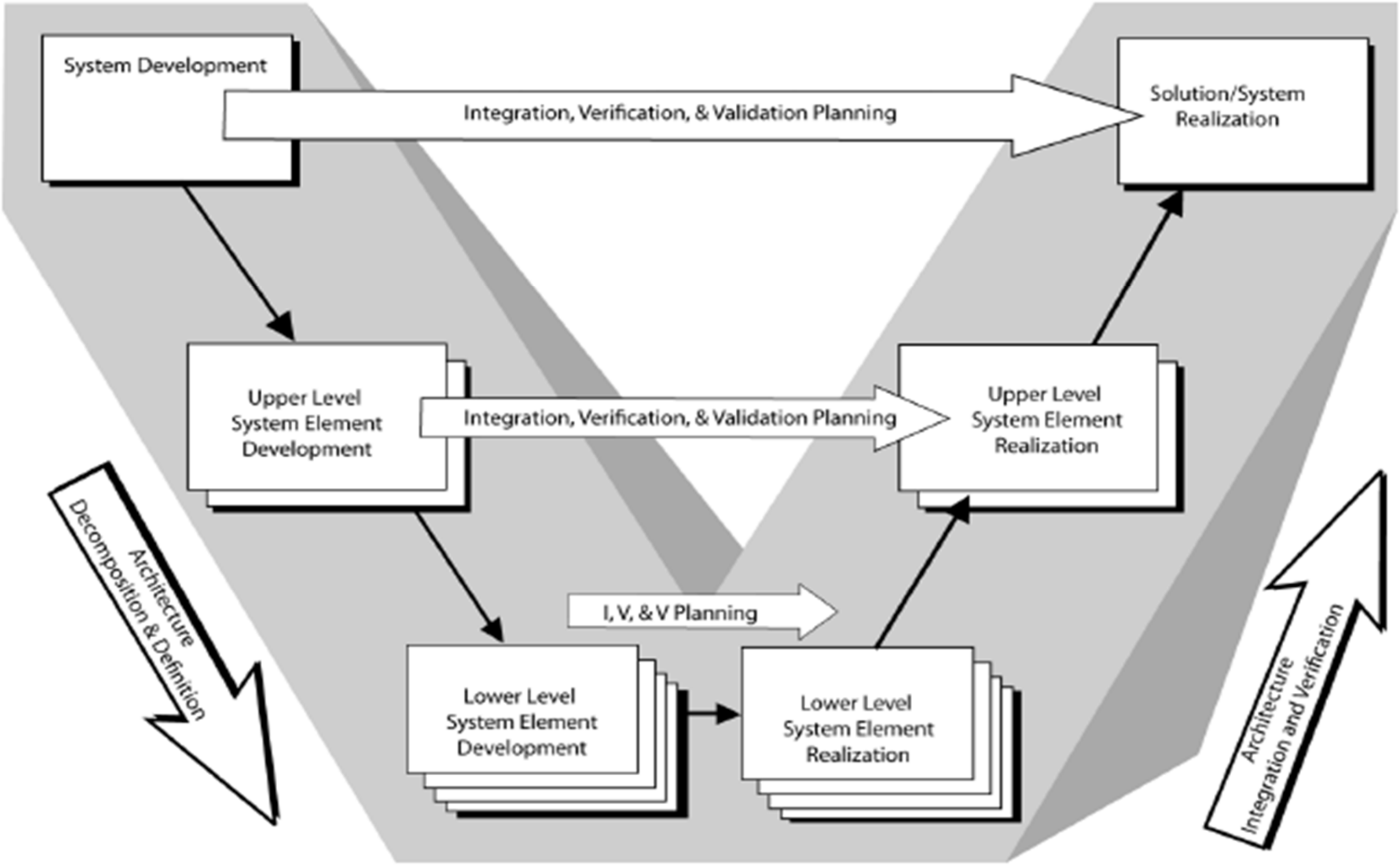
Spiral

- Combines formalized design with iterative processing to combine advantages of the top-down and bottom-up approaches

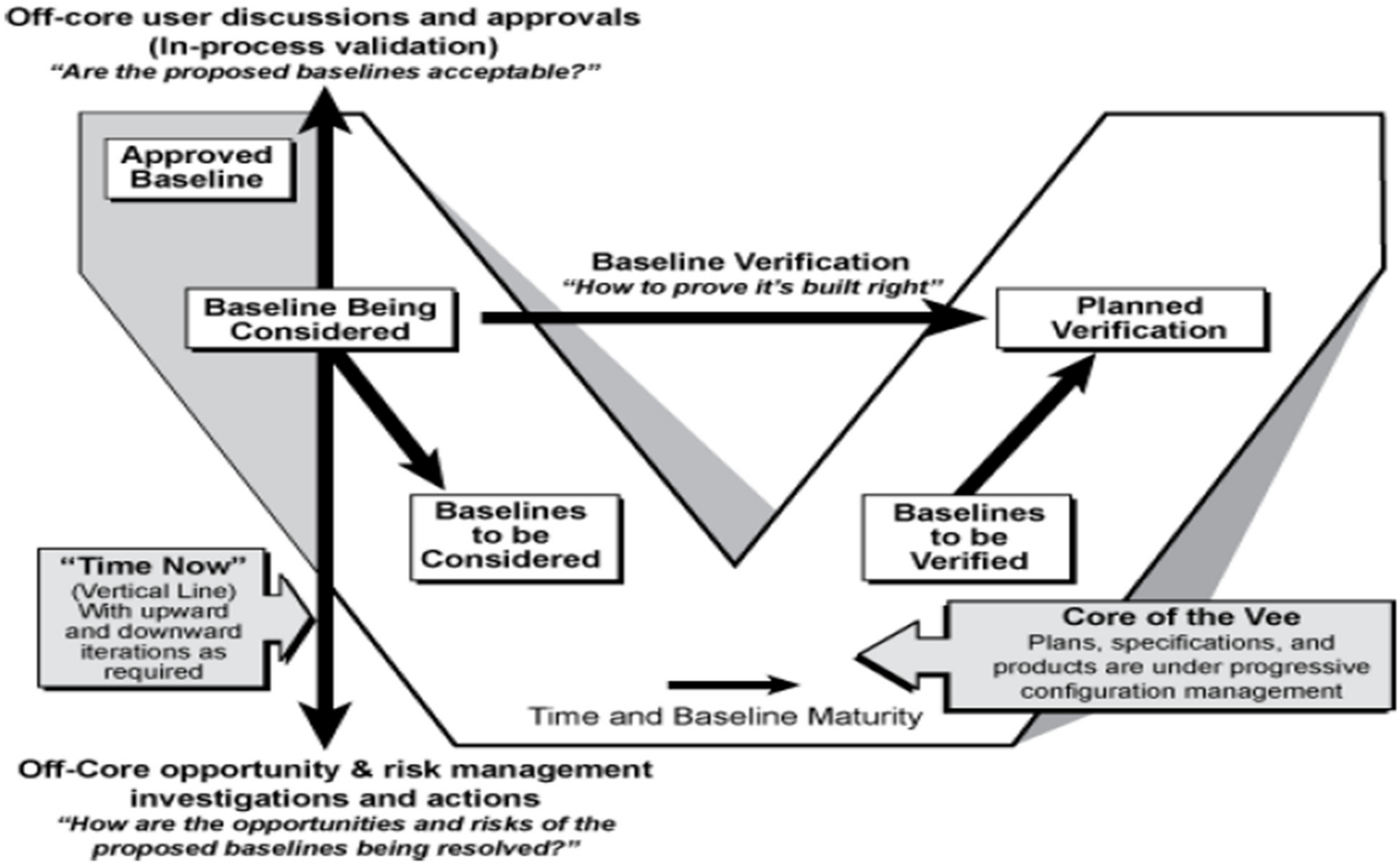


Vee

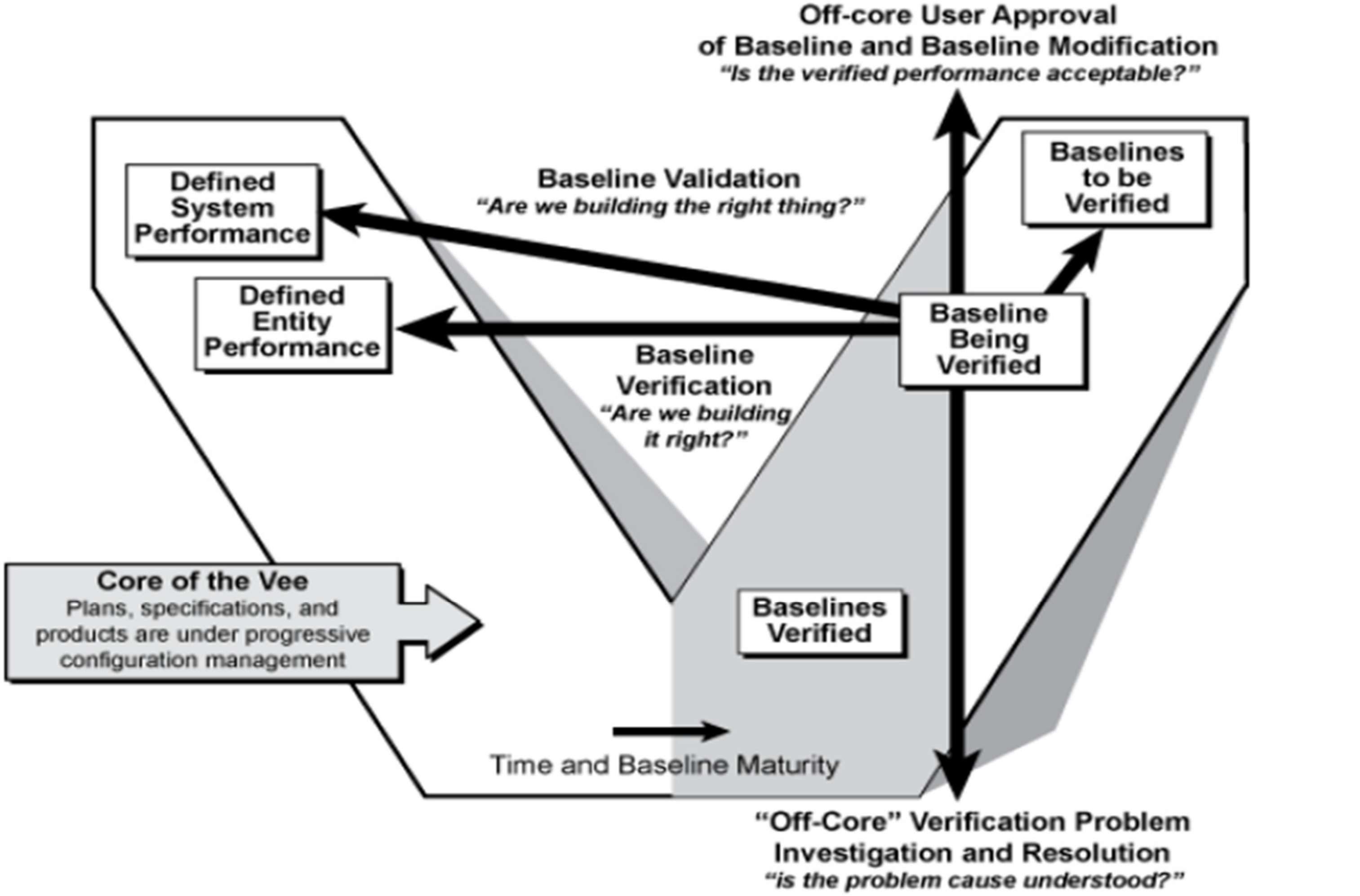
- Visualization methodology used when implementing the Systems Engineering lifecycle, highlights verification activities, and continuous risk and opportunity assessment


System Development
Integration, Verification, & Validation Planning
Solution/System Realization
Upper Level System Element Development
Integration, Verification, & Validation Planning
Upper Level System Element Realization
I, V, & V Planning
Lower Level System Element Development
Lower Level System Element Realization
Architecture Decomposition & Definition
Architecture Integration and Verification

Vee


Off-core user discussions and approvals
(In-process validation)
"Are the proposed baselines acceptable?"
Approved Baseline
Baseline Being Considered
Baseline Verification
"How to prove it's built right"
Planned Verification
Baselines to be Considered
Baselines to be Verified
"Time Now" (Vertical Line) With upward and downward iterations as required
Core of the Vee
Plans, specifications, and products are under progressive configuration management
Time and Baseline Maturity
Off-Core opportunity & risk management investigations and actions
"How are the opportunities and risks of the proposed baselines being resolved?"

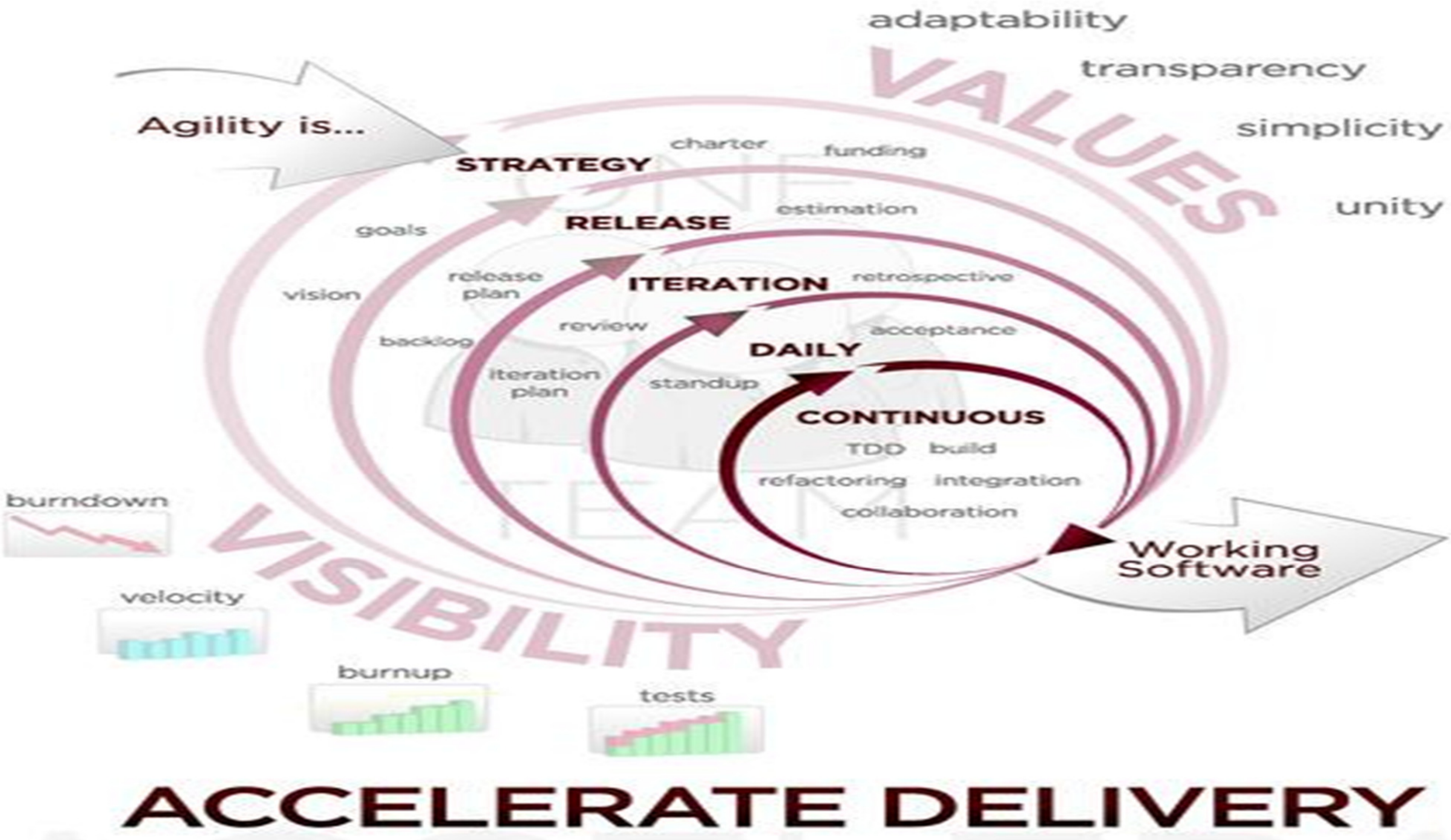
Vee


Off-core User Approval
of Baseline and Baseline Modification
"Is the verified performance acceptable?"
Defined System Performance
Baseline Validation
"Are we building the right thing?"
Baselines to be Verified
Defined Entity Performance
Baseline Being Verified
Baseline Verification
"Are we building it right?"
Core of the Vee
Plans, specifications, and products are under progressive configuration management
Baselines Verified
Time and Baseline Maturity
"Off-Core" Verification Problem
Investigation and Resolution
"is the problem cause understood?"

Iterative or Mixed

Agile

- Focus on flexibility versus formalized process, rapid response, customer satisfaction, and collaboration



Other Rapid Development/Prototyping

- Examples
  - Scrum
  - Rapid Application Development (RAD)
  - Joint Application Development (JAD)
  - Extreme Programming (XP)
- Merger of various iterative and prototyping methodologies (waterfall mixed with spiral approach)
- Utilizes
  - Preliminary data models
  - Business process models
  - Requirement verification using prototypes
  - Software re-use
  - Small teams (some self-managing)
- Goals
  - Faster product development
  - Higher code quality

Lean

Lean

- Based on Toyota system of Just in Time “JIT” Development
- Eliminate all Waste (*muda*)
  - Ambiguity in the requirements
  - Unnecessary functionality or process steps
  - Delay in following the documented processes
  - Bureaucracy
  - Non-optimized communications
- Complete focus on ensuring value to the customer (achieving mission assurance)
- Enhance learning across the organization
- Base decisions on fact, eliminate uncertainty as much as possible
- Delivery as fast as possible
- High priority on empowerment, integrity, and holistic approach

Lean SE

- Application of LEAN to Systems Engineering (SE) Framework (INCOSE)
- Effects typical SE, organization, and project management activities
- Enable optimum life-cycle “value” with minimal waste
- Increased mission assurance focus
- Greater levels of authority, responsibility, and accountability
- Foundation of respect for all participants

Quality Programs

Focal Points

In conjunction with the project framework selected for the various organizational teams, it is often extremely beneficial from a customer satisfaction, empowerment, and rigor standpoint to implement quality programs which enable enterprise-wide goals to ultimately be achieved.

*Implementation of a **formalized quality program** enables the enterprise's personnel to work from a common baseline, speak with a common vocabulary, utilize organizational synergies, and focus on achieving customer satisfaction within a structured, yet flexible environment.*

Much like the program/project methodologies, enterprises often select the most beneficial portions of various quality programs and intertwine them to ultimately produce a sustainable program which addresses their particular needs, unless they are specifically required to certify to a particular quality program for industry compliance purposes.

Quality programs affect each part of an organization differently, but should be undertaken with the long-term view in mind and with the realization that the enterprise must be mentally prepared to handle the "discomfort" that may occur due to heightened operational scrutiny .

Quality Programs

- TQM
- ISO 900x
- Baldrige
- CMM & CMMi
- Six-Sigma

TQM

- **Total Quality Management** (TQM) is the baseline quality management program implemented in the 1980s timeframe
- A **holistic, integrative, philosophic approach** which empowers everyone in an organization to strive for continual product or service quality improvements
- **Quality is everyone's responsibility**
- **9 common practices**
  - Cross-functional product design
  - Process management
  - Supplier quality management
  - Customer involvement
  - Information and feedback
  - Committed leadership
  - Strategic planning
  - Cross-functional training
  - Employee involvement

ISO

- ISO 9000: 2005 describes the fundamentals of a Quality Management System
  - For use for familiarity purposes prior to implementing this system within the organization
- ISO 9001: 2008 framework for a Quality Management System
  - **Systematic approach** towards achieving quality goals with the intent of achieving certification
- **8 basic principles**
  - Customer focus
  - Leadership
  - Involvement of people
  - Process approach
  - System approach to management
  - Continuous improvement
  - Factual approach to decision making
  - Mutually beneficial supplier relationship

Baldrige

- Baldrige Criteria are divided in the following buckets
  - Business & Non-profit entities
    - Government
    - Manufacturing
    - Other for-profit
    - Non-profit
  - Healthcare
  - Education
- Typically focuses on **7 areas** of **performance excellence**
  - Leadership
  - Strategic Planning
  - Customer Focus
  - Measurement, analysis, and knowledge management
  - Workforce focus
  - Operations focus
  - Results

CMM & CMMI

- Capability Maturity Model (CMM)
  - CMM-SEI, based on DoD research across multiple organizations
  - CMMi superseded CMM
  - Focus on **process improvement**
- Structured Approach
  - 5 Maturity Levels
  - Key Process Areas (KPA)
  - Goals
  - Common Features
  - Key Practices
- Utilized in the following areas
  - Software Engineering
  - System Engineering
  - Project Management
  - Software Maintenance
  - Risk Management
  - System Acquisition
  - IT Services
  - Business Processes
  - Human Capital Management

Maturity Levels

- **Level 1** - *Initial (Chaotic)* - processes are (typically) undocumented and in a state of dynamic change, *ad hoc* in nature. Usually a chaotic or unstable environment.
- **Level 2** - *Repeatable* - processes are in partiality repeatable, possibly with consistent results. No rigorousness.
- **Level 3** - *Defined* - processes at this level are defined and documented and subject to some degree of improvement over time.
- **Level 4** – *Managed* - processes use process metrics, management can effectively control and identify ways to adjust and adapt the process to particular projects without measurable losses of quality or deviations from specifications. Process Capability is established from this level.
- **Level 5** - *Optimizing* - processes at this level focus on continually improving process performance through both incremental and innovative technological changes/improvements.

Key Practice Areas

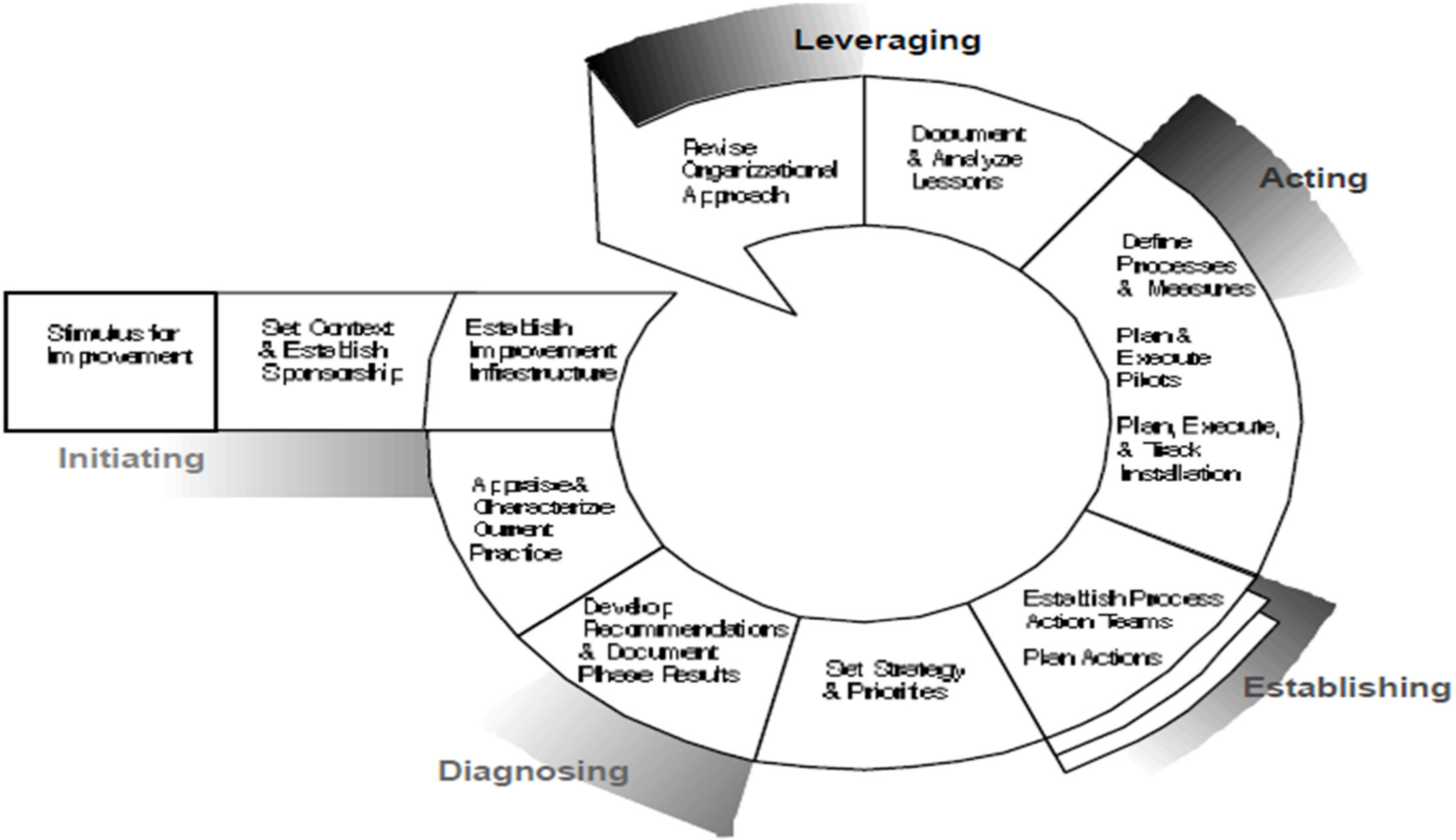
- More stringent **Key Practice Areas** (KPAs) are associated with each higher Maturity Level
- Each **KPA** consists of a
  - Goal
  - Commitment
  - Ability
  - Measurement
  - Verification
- Organizations typically attain a maturity level status over a **continuum of time** in a step-by-step manner
- Organizations are typically measured against **five checklist areas** for each maturity level
  - Policy
  - Standard
  - Process
  - Procedure
  - Level overview

“IDEAL” Model

- 5 Stages of the Process Improvement cycle
  - **I**nitiating
  - **D**iagnosing
  - **E**stablishing
  - **A**cting
  - **L**everaging

- Framework which covers necessary components to ensure a **successful process improvement effort**
  - Phases
  - Activities
  - Resources

5 Stages


Leveraging
Revise Organizational Approach
Document & Analyze Lessons
Acting
Define Processes & Measures
Plan & Execute Pilots
Plan, Execute, & Track Installation
Stimulus for Improvement
Set Context & Establish Sponsorship
Establish Improvement Infrastructure
Initiating
Appraise & Characterize Current Practice
Develop Recommendations & Document Phase Results
Set Strategy & Priorites
Establish Process Action Teams
Plan Actions
Establishing
Diagnosing

Six-Sigma

- **Six Sigma** seeks to **identify and eliminate the causes of defects**, while minimizing variability in manufacturing and business practices
- It uses a set of quality management methods, including statistical analysis to gauge quality
- Specialized infrastructure of certified (expert) personnel:
  - Master Black Belts, Black Belts, Green Belts
- Follows defined sequence of steps and quantifiable financial targets (cost reduction or profit increase)
- Six Sigma defines 99.99966% of the products manufactured are statistically expected to be free of defects (3.4 defects per million)
- Focuses is on:
  - Reduction of process variation
  - Processes characteristics which can be measured, analyzed, improved, and controlled
  - Commitment from the entire organization in order to sustain a quality improvement process

Six-Sigma

- Program Features
  - Achieving measurable and quantifiable financial returns
  - Increased emphasis on strong and passionate management leadership and support
  - Expert “champions” which lead and implement the approach
  - Decision making based on verifiable data

- Two main project methodologies used
  - **DMAIC** – improving existing business processes and products
  - **DMADV** – creating new products or process designs

DMAIC

- **D**efine goals which are consistent with customer requirements and enterprise strategy
- **M**easure characteristics which are Critical to Quality (CTQs), product capabilities, production process capability, and risks
- **A**nalyze data in order to investigate cause and effect relationships and determine root cause
- **I**mprove and/or optimize processes and set up pilot runs in order to establish process capability
- **C**ontrol processes by implementing process control and continuous monitoring to remove defects before they adversely effect the organization

DMADV

- **D**efine goals which are consistent with customer requirements and enterprise strategy
- **M**easure characteristics which are Critical to Quality (CTQs), product capabilities, production process capability, and risks
- **A**nalyze data in order to develop and design viable alternatives, and constructively evaluate options
- **D**esign activities which optimize the design and includes design verification activities
- **V**erify the design, run pilots, implement production processes, hand-off to process owner(s)

Risk Management

Focus Points

No matter the extent of prior planning activities, organizational and project ***risks*** *are simply a fact of life. Market forces, international drivers, supply chain constraints, as well as organizational constraints and interdependencies must be studied, understood, and fully taken into consideration by overarching programs and associated goal achievement plans.*

In order to effectively mitigate these identified risks it is beneficial for the organization to proactively create and maintain a risk management program in order to minimize adverse effects to relevant projects, the systems under development, and ultimately the customer.

Risk Management activities are also recommended to be incorporated across the entire SDLC in order to maximize the potential success of the organization and the associated critical programs.

Risk Management Programs

- **Risk Management (Traditional Approach)**
- **NIST-Risk Management Framework (RMF)**
  - Six phase program
  - Maps to the SDLC to provide proper risk related programmatic support
  - Takes into account varying levels of security postures and system characterization in order to successfully combat project risks via implementation of relevant controls
  - Will eventually map SP 800-53 security controls to ISO 27001 controls (Information Security Management Systems) so that organizations are able to be simultaneously compliant with both standards
- **Risk Management - Continuous Monitoring Programs**
- **Computer Emergency Readiness Team (CERT) Resilience Management Model (CERT-RMM)**
  - 26 process areas enable achievement of mission assurance

Typical Framework

- **Focus Areas**
  - Enterprise Risk Management
  - Project Management (Risk Management related activities)
  - Business Continuity
- **Useful standards**
  - ISO 31000: Principles and Guidelines on Implementation (Risk Mgmt in general)
  - IEC 31010: Risk Management - Risk Assessment Techniques
  - ISO/IEC 73: Risk Management – Vocabulary
  - ISO/IEC 27005:2008 : Information technology -- Security techniques -- Information security risk management
- **Risk Management is typically made up of the following activities:**
  - Risk Identification
  - Risk Assessment
  - Risk Prioritization
  - Risk Handling

Risk Related Activities

- **Risk Identification**
  - Identity, characterize, and assess threats
  - Can utilize multiple methods: objective-based, scenario-based, taxonomy-based, common-risk checking (per industry basis), risk charting (weighting mechanisms)
- **Risk Assessment**
  - Assess the vulnerability of critical assets to specific threats
  - Assess potential severity of the loss and the probability of occurrence
  - Qualitative or quantitative methodology depending on need
- **Risk Prioritization**
  - Prioritize risks based on cost/asset value
  - Select risk handling method on a per risk basis
- **Risk Handling**
  - *Reduce* (Mitigate properly based on cost/benefit analysis)
  - *Transfer* (Insure or Outsource)
  - *Avoid* (Do not pursue business opportunity)
  - *Accept* (Budget accordingly to pay for the liability the risk may impose)

Importance of the Risk Function

- **Objectives**
  - Creation of value for the enterprise
  - Integral to the organization’s decision making practices
  - Systematically addressed in a structured manner and incorporated into organizational processes and procedures
  - Focus on inclusiveness, transparency, flexibility, and continuous improvement efforts

- Vital to the success of any selected risk framework is the creation of a formalized plan and proper communication

NIST - RMF

- **Main Standard:** SP 800-30 Risk Management Guide for Information Technology Systems
- **Associated Standards:**
  - SP 800-37 rev. 1 Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach
  - SP 800-39 DRAFT Integrated Enterprise-Wide Risk Management: Organization, Mission, and Information System View
  - SP 800-53 rev3 Recommended Security Controls for Federal Information Systems and Organizations
  - SP 800-53A rev1 Guide for Assessing the Security Controls in Federal Information Systems and Organizations, Building Effective Security Assessment Plans
  - SP 800-64 Security Considerations in the System Development Life Cycle
  - SP 800-137 DRAFT Information Security Continuous Monitoring for Federal Information Systems and Organizations
- **Processes**
  - Risk Assessment
  - Risk Mitigation
  - Evaluation and Assessment

Risk Assessment

- **Phases**
  - Categorize the Information Systems
  - Select Proper Security Controls
  - Implement Security Controls
  - Assess Security Controls
  - Authorize the Information Systems
  - Monitor Security Controls (Continuously)
- **Activities**
  - System Characterization
    - Boundaries, Functions, Criticality, Sensitivity
  - Threat Identification
    - History and Industry knowledge
  - Vulnerability Identification
    - History, audit findings, security requirements and test/evaluation results – output is an itemized list of vulnerabilities

Risk Assessment

- **Activities (cont'd)**
  - Control Analysis
    - Current and planned controls
  - Likelihood Determination
    - Threat source motivation, capacity, nature of vulnerability, current control gap
  - Impact Analysis
    - Mission impact, loss of confidentiality, integrity, or availability, output is an impact rating
  - Risk Determination
    - Likelihood of exploitation, magnitude of impact, adequacy of controls, output is itemized list of risks with correlated risk level
  - Control Recommendations
    - Formalized control recommendations list
  - Results Documentation
    - Formalized Report which addresses threats, vulnerabilities, measures the risk, and provides recommendations for control implementation

Risk Mitigation

- **Actions/Steps**
  - Prioritize Actions
  - Evaluate recommended controls
  - Conduct Cost Benefit analysis
  - Select Controls
  - Assign Responsibility
  - Develop Safeguard Implementation Plan
  - Implement Selected Controls
  - Ascertain Residual Risk and repeat process if necessary
- **Risk Handling**
  - *Mitigate (Reduce)* Risk
  - *Transfer* Risk
  - *Avoid* Risk
  - *Assume* Risk

Evaluation and Assessment

- **Activities**
  - Ongoing and evolving risk management activities due to changes
- **Changes which require additional cycles**
  - Network changes or expansions
  - Personnel changes
  - Hardware and software upgrades, etc.
- **Cycle Time**
  - FISMA requires every 3 years or whenever major changes are introduced
  - Now moving to a continuous monitoring model

Control Framework

- **Control Families**
  - The NIST 800-53 rev 3 framework is broken out into **18 control families**, across **three control types**: management (M), operational (O), technical (T)
- **Control Baselines**
  - Controls are implemented per the **FIPS-199/FIPS-200** system categorization and security posture “buckets” of High, Moderate, and Low and relevant control baselines created
- **Individual Control Data**
  - Control Id & Name
  - Control Text
  - Supplemental Guidance
  - Control Enhancements
  - References
  - Priority
  - Applicability to a specific Baseline
- It is recommended that personnel get familiarized with the control families and associated detail data and utilize the information for test planning and infrastructure implementation purposes

NIST – RMF – Control Families

Control ID	Control Family Name	Type of Control	Control ID	Control Family Name	Type of Control
AC	Access Control	T	**MP**	Media Protection	O
AT	Awareness & Training	O	**PE**	Physical and Environmental Protection	O
AU	Audit & Accountability	T	**PL**	Planning	M
CA	Sec. Assessment & Authorization	M	**PS**	Personnel Security	O
CM	Configuration Mgmt	O	**RA**	Risk Assessment	M
CP	Contingency Plan	O	**SA**	System & Services Acquisition	M
IA	Identification & Authentication	T	**SC**	System & Communications Protection	T
IR	Incident Response	O	**SI**	System & Information Integrity	O
MA	Maintenance	O	**PM**	Program Mgmt	M

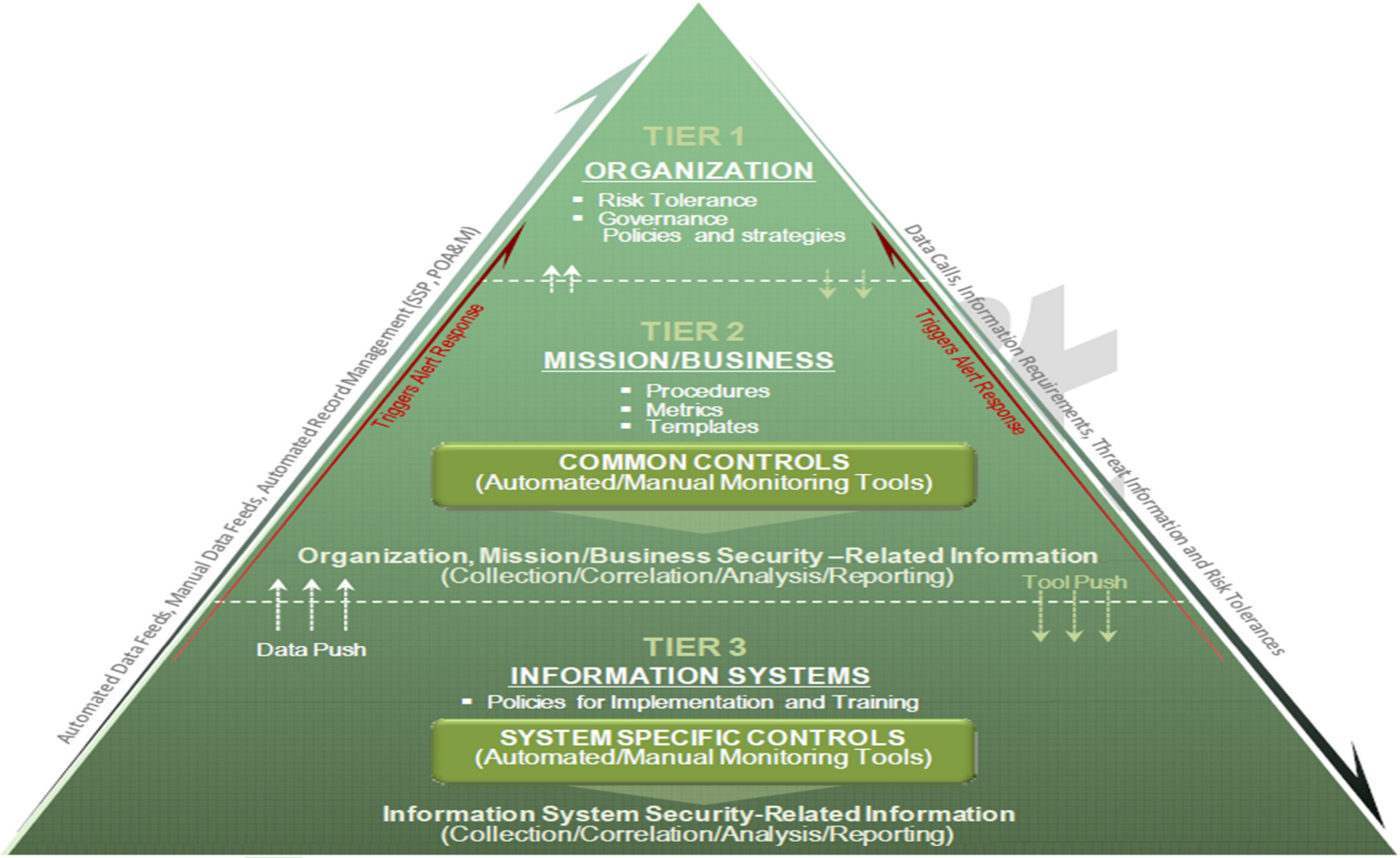
NIST - Continuous Monitoring

- **Main Specification**: SP 800-137 DRAFT Information Security Continuous Monitoring for Federal Information Systems and Organizations
- FISMA Requirements are evolving from a three-year reporting cycle to a continuous monitoring methodology which constantly examines vulnerabilities and associated controls, primarily focusing on utilization of automated methods whenever possible (Security Content Automation Protocol [SCAP])
- **Focus Areas:**
  - Technology
  - Processes
  - Procedures
  - Operating Environments
  - Personnel
- **Essential Elements**
  - Configuration Management/Change Control
  - Security Impact Analysis
  - Ongoing assessment of system security controls
  - Security status monitoring and reporting
  - Active executive and management participation

Continuous Monitoring

- **Tiered Risk Management Focus allows for "holistic" approach**
  - Tier 1: Organization
  - Tier 2: Mission/Business Area
  - Tier 3: Information System
- **Essential Steps**
  - Determine the security impact of proposed or actual changes
  - Assess required technical, management, and operational security controls
  - Conduct remediation activities based on findings, prior assessments, and plan of actions and milestone (POA&M) itemizations
  - Update the various documentation (security plan, assessment report, POA&Ms)
  - Report on security status
  - Determine whether acceptable risk levels exist given analysis, if not continue remediation activities
  - Implement decommissioning strategy when necessary

Continuous Monitoring



TIER 1

ORGANIZATION

- Risk Tolerance
- Governance
  Policies and strategies

TIER 2

MISSION/BUSINESS

- Procedures
- Metrics
- Templates

COMMON CONTROLS
(Automated/Manual Monitoring Tools)

Organization, Mission/Business Security –Related Information
(Collection/Correlation/Analysis/Reporting)

TIER 3

INFORMATION SYSTEMS

- Policies for Implementation and Training

SYSTEM SPECIFIC CONTROLS
(Automated/Manual Monitoring Tools)

Information System Security-Related Information
(Collection/Correlation/Analysis/Reporting)

Automated Data Feeds, Manual Data Feeds, Automated Record Management (SSP, POA&M)
Triggers Alert Response
Data Calls, Information Requirements, Threat Information and Risk Tolerances
Triggers Alert Response
Tool Push
Data Push

CERT- Resilience Management Model (RMM)

- **4 categories**

  1) Engineering 2) Operations

  3) Enterprise Management 4) Process Management

- **26 process areas** which aide in achievement of mission assurance with corresponding goals and practices
- **Goals**
  - Holistically improve risk and resilience management through convergence of disciplines:
    - Security Management
    - Business Continuity
    - IT Operations Management
  - Incorporate into everyday processes and practices, determine metrics and measurement, and initiate continual process improvement
  - Provide a foundation for process institutionalization and organizational process maturity

CERT- RMM: 26 process areas across 4 categories

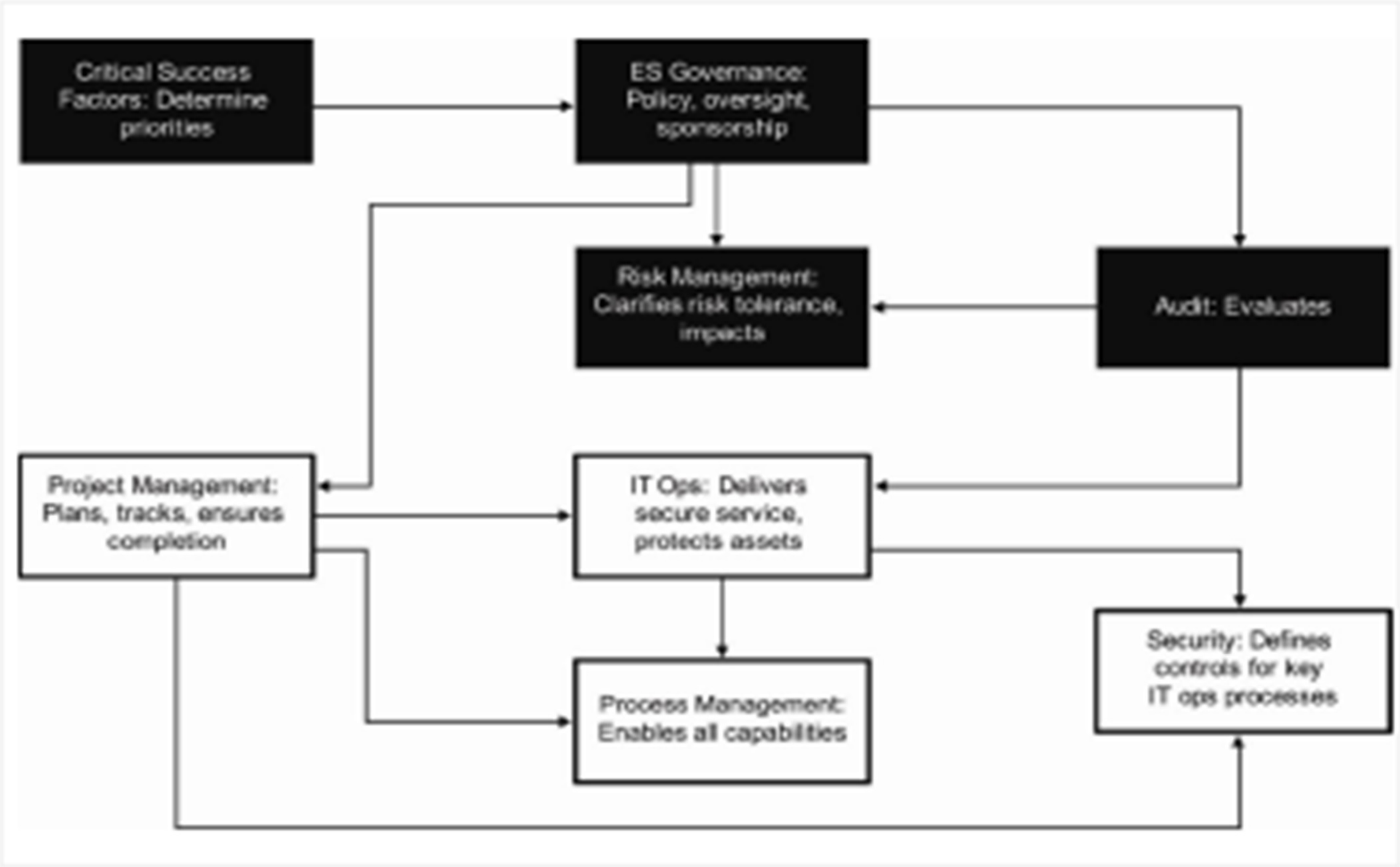
Engineering	
ADM	Asset Definition and Management
CTRL	Controls Management
RRD	Resilience Requirements Development
RRM	Resilience Requirements Management
RTSE	Resilient Technical Solution Engineering
SC	Service Continuity

Enterprise Management	
COMM	Communications
COMP	Compliance
EF	Enterprise Focus
FRM	Financial Resource Management
HRM	Human Resource Management
OTA	Organizational Training & Awareness
RISK	Risk Management

Operations Management	
AM	Access Management
EC	Environmental Control
EXD	External Dependencies
ID	Identity Management
IMC	Incident Management & Control
KIM	Knowledge & Information Management
PM	People Management
TM	Technology Management
VAR	Vulnerability Analysis & Resolution

Process Management	
MA	Measurement and Analysis
MON	Monitoring
OPD	Organizational Process Definition
OPF	Organizational Process Focus

CERT- RMM



Critical Success Factors: Determine priorities

ES Governance: Policy, oversight, sponsorship

Risk Management: Clarifies risk tolerance, impacts

Audit: Evaluates

Project Management: Plans, tracks, ensures completion

IT Ops: Delivers secure service, protects assets

Security: Defines controls for key IT ops processes

Process Management: Enables all capabilities

Security within the Lifecycle

Focal Points

*Like project, quality and risk issues, **security** related issues oftentimes cause projects to fail.*

In order to maximize the success proposition, security must be implemented across the entire lifecycle, preferably in parallel with implementation of the aforementioned formalized programs.

Risk Integration

- **Integrating Risk Management into the SDLC**
  - Phase 1 – Initiation
    - **Phase need:** Need for the system is expressed and the purpose and scope is documented
    - **RMF:** Identified risks are used to flesh out system and corresponding security requirements, as well as the security concept of operations
  - Phase 2 – Development or Acquisition
    - **Phase need:** System is designed, purchased, developed, or constructed
    - **RMF:** Supports the assessment of the system implementation against requirements and ultimate design choices made
  - Phase 3 – Implementation
    - **Phase need:** System security features should be configured, enabled, tested, and verified
    - **RMF:** Supports assessment of the system against security requirements within the modeled environment -- risk handling issues must be resolved prior to operation within the actual environment

Risk Integration

- Phase 4 – Operation or Maintenance
  - **Phase Need:** System is able to perform its functions (including security related functionality)
  - **RMF:** Periodic re-evaluation occurs if and when operational conditions warrant

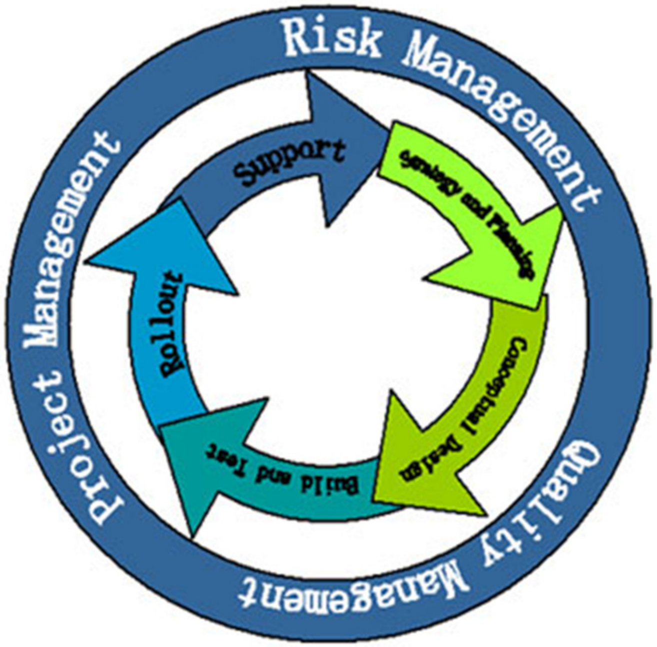
- Phase 5 – Disposal
  - **Phase Need:** Disposal of the system and associated data
  - **RMF:** Proper data and system disposal is handled per security requirements and system migration activities take place in an authorized manner

Holistic Approach Necessary

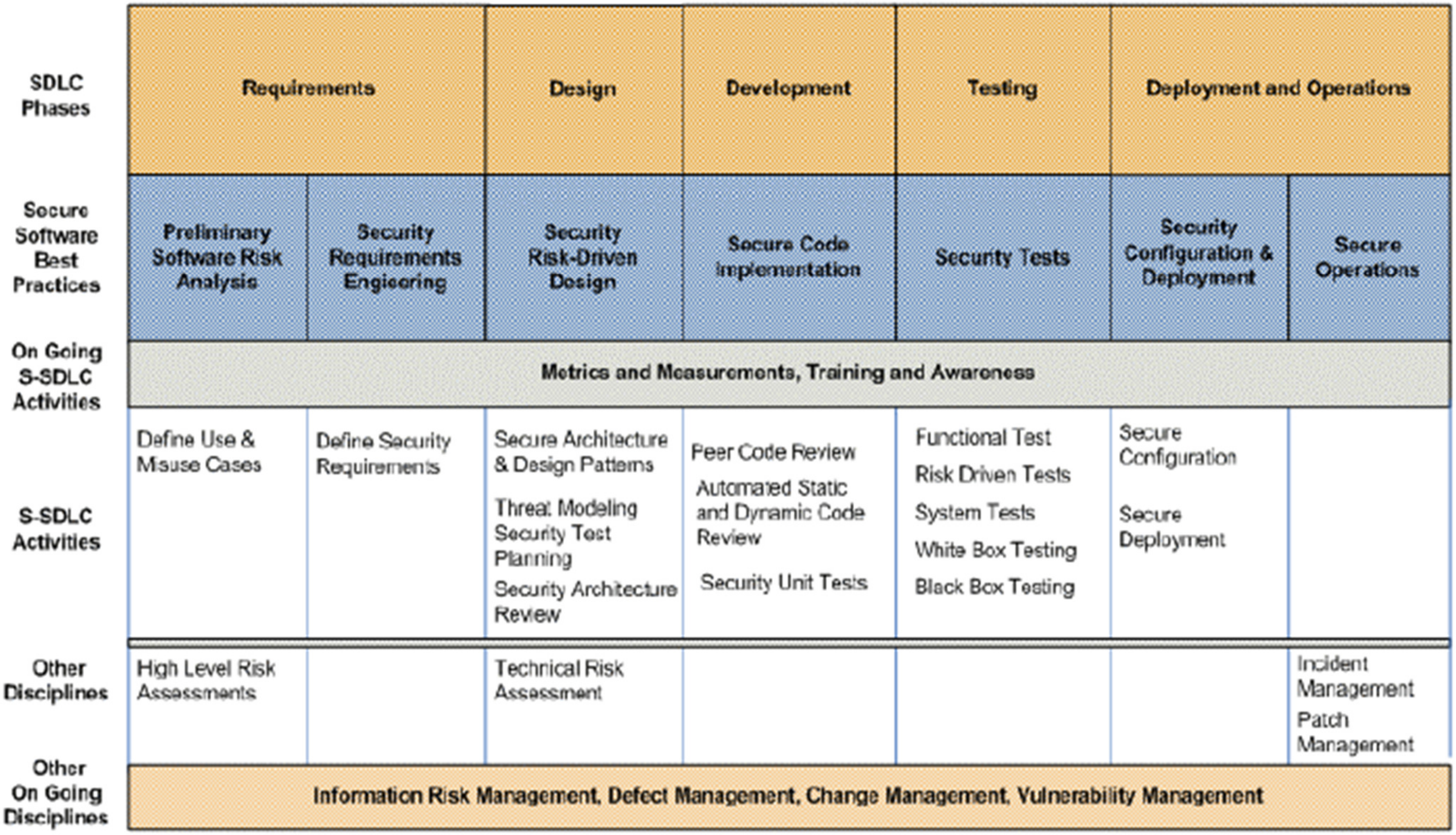
- Goal is to holistically combine all three areas cohesively to create positive synergies
  - Project Methodologies
  - Quality Programs
  - Risk Management Framework

- Ultimately for the project and organization to be successful, risk related activities need to be mapped to each of the major life cycle phases

- “Easier said then done”… however, there are plenty of resources available for use



Software Security Frameworks



SDLC Phases	Requirements		Design	Development	Testing	Deployment and Operations	
Secure Software Best Practices	Preliminary Software Risk Analysis	Security Requirements Engieering	Security Risk-Driven Design	Secure Code Implementation	Security Tests	Security Configuration & Deployment	Secure Operations
On Going S-SDLC Activities	Metrics and Measurements, Training and Awareness						
S-SDLC Activities	Define Use & Misuse Cases	Define Security Requirements	Secure Architecture & Design Patterns Threat Modeling Security Test Planning Security Architecture Review	Peer Code Review Automated Static and Dynamic Code Review Security Unit Tests	Functional Test Risk Driven Tests System Tests White Box Testing Black Box Testing	Secure Configuration Secure Deployment	
Other Disciplines	High Level Risk Assessments		Technical Risk Assessment				Incident Management Patch Management
Other On Going Disciplines	Information Risk Management, Defect Management, Change Management, Vulnerability Management						

Security should be “Built in” not “Bolted on”

- **Build Security In & SW Assurance sites**
  - National Cyber Security Division (NCSD) of the U.S. Department of Homeland Security
  - Standards, frameworks, methodologies, and life-cycle models which support security related activities
- **Build Security In Maturity Model (BSIMM2)**
  - Resources designed for aid in understanding, planning, and implementing a software security initiative
- **Microsoft**
  - Industry Best Practices & specific recommended actions on a per phase basis
- **NIST SP 800-64 Rev. 2 – Security Considerations in the System Development Life Cycle**
  - Phased approach, legacy and virtualized systems covered, supply chain and partner security considerations addressed

Additional Tools for your Toolkit

- **Open Software Assurance Maturity Model (OpenSAMM)**
  - Open framework for software security based on particular organizational risks
  - Integrates with BSIMM
- **Open Source Security Testing Methodology Manual (OSSTMM 3)**
  - Institute for Security and Open Methodologies (ISECOM)
  - Peer-reviewed methodology for performing security tests and metrics
- **The Open Web Application Security Project (OWASP)**
  - Control Libraries, security verification tools, Top 10 Web Application Security Risks, Guides (Code Review, Development, and Testing), CLASP, among other resources
- **Secure-SDLC**
  - Security Activities throughout the Security Lifecycle
- **Security Touchpoints**
  - Cigital-based security framework

Additional Tools for your Toolkit

- **Software Assurance Metrics and Tool Evaluation (SAMATE)**
  - U.S. Department of Homeland Security (DHS) National Cyber Security Division and NIST
  - Identification, enhancement and development of software assurance tools
- **Software Engineering Institute Carnegie Mellon - CERT**
  - Secure coding standards (Java, C, C++, and language independent practices)
- **SysAdmin, Audit, Network, Security Institute (SANS)**
  - Consensus Audit Guidelines (CAGs) - 20 Critical Security Controls
  - Top Cyber Security Risks, Top 25 Software Errors
- **Systems Security Engineering Capability Maturity Model (SSE-CMM)**
  - Process model that can be used to improve and assess the security engineering capability of an organization
  - The SSE-CMM provides a comprehensive framework for evaluating security engineering practices against the generally accepted security engineering principles

“Build Security In”

- **Cross Organizational Initiatives**
  - DHS
  - Dept of Commerce
  - Department of Defense
  - Industry
  - Academia
- **Build Security In Resources across the Lifecycle**
  - Tools
  - Guidelines
  - Rules
  - Practices
  - Principles
- **Applicable Audiences**
  - Developers/Test Engineers
  - Security Architects, Security Practitioners, Security Executives
  - Program, Project, Risk Managers

Build Security In – Software Assurance Efforts

- Same cross organizational structure
- **“Mission Assurance”** achievement is primary objective
- **Goals**
  - Reduce vulnerabilities
  - Minimize exploitations
  - Focus on improving development and deployment of secure and trustworthy software products
- **Program Focus**
  - Move from “reactive” patch management practices to “proactive” software assurance practices
- **Collaborative Program Components**
  - People
  - Processes
  - Technology
  - Acquisition

BSIMM2

- Resources to aid understanding, measuring, and planning a security initiative
- Case Studies from 30 actual security initiative implementations
- Software Security Framework (SSF)
  - **3 Maturity Levels**
    - Level 1 -- Common understanding of direction and strategy
    - Level 2 – Align behavior with strategy and verify
    - Level 3 – Practice risked-based management
  - **4 Domains**
    - Governance
    - Intelligence
    - Software Security Development Lifecycle (SSDL) Touchpoints
    - Deployment
  - **12 Practice Areas**
  - **109 security activities**

BSIMM2 - SSF

Software Security Framework			
Governance	**Intelligence**	**SDDL Touchpoints**	**Deployment**
Strategy & Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance & Policy	Security Features & Design	Code Review	Software Environment
Training	Standards & Requirements	Security Testing	Configuration & Vulnerability Management

BSIMM2 – Governance Domain

Software Security Framework		
Domain	**Area**	**Activity Focus**
Governance	Strategy & Metrics	Planning, assigning roles and responsibilities, identifying software security goals, determining budgets, identifying metrics and gates.
	Compliance & Policy	Identifying controls for compliance regimens, developing contractual controls (COTS SLA), setting organizational policy, auditing against policy.
	Training	Creation of a knowledgeable workforce and correcting errors in processes.

BSIMM2 – Intelligence Domain

Software Security Framework		
Domain	**Area**	**Activity Focus**
Intelligence	Attack Models	Threat modeling, abuse cases, data classification, technology-specific attack patterns.
	Security Features & Design	Threat modeling, abuse cases, data classification, technology-specific attack patterns.
	Standards & Requirements	Explicit security requirements, recommended COTS, standards for major security controls, standards for technologies in use, standards review board.

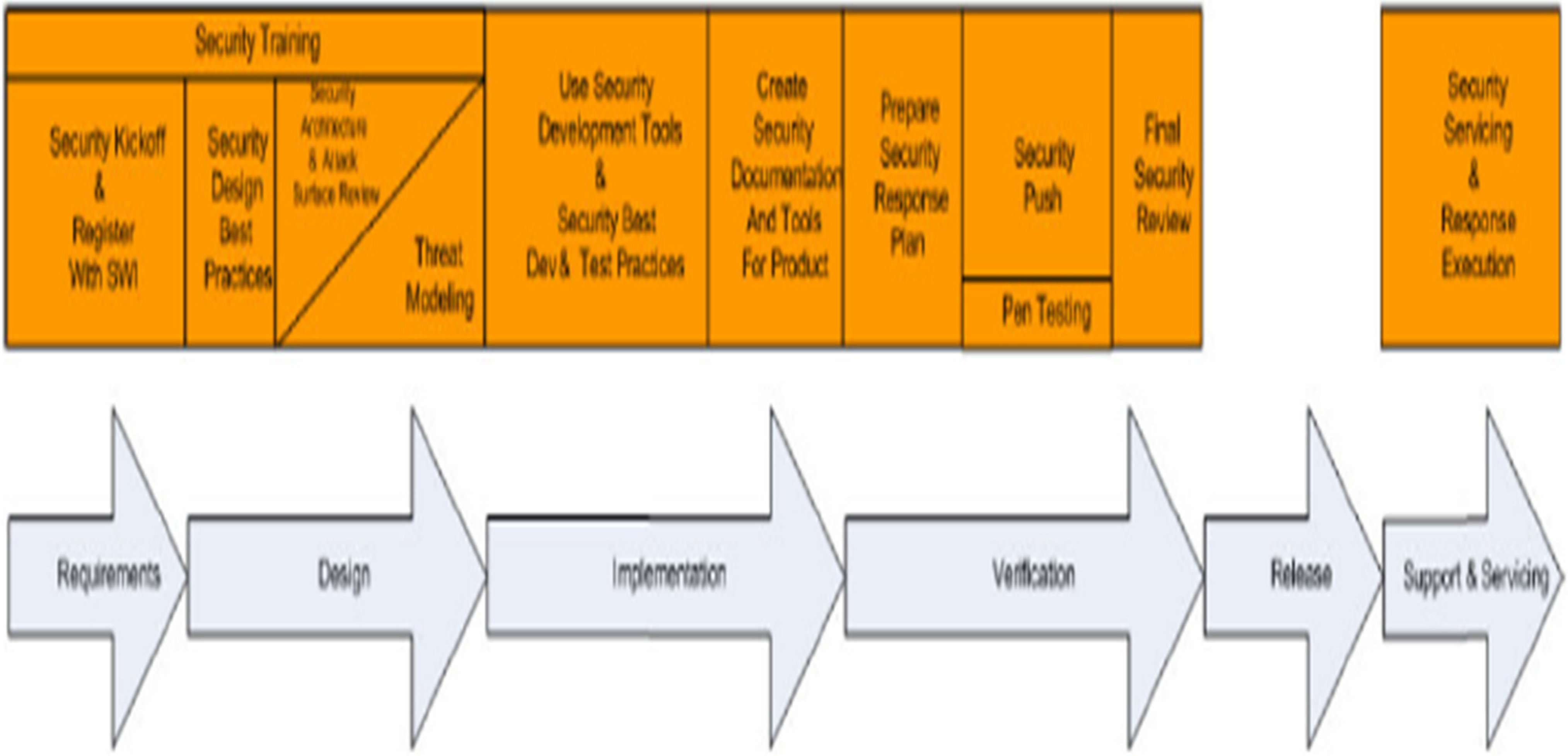
BSIMM2 – SDDL Touchpoints Domain

Software Security Framework		
Domain	**Area**	**Activity Focus**
SDDL Touchpoints	Architecture Analysis	Capturing software architecture diagrams, applying lists of risks and threats, adopting a process for review, building an assessment and remediation plan.
	Code Review	Use of code review tools, development of customized rules, profiles for tool use by different roles, manual analysis, ranking/measuring results.
	Security Testing	Use of black box security tools in QA, risk driven white box testing, application of the attack model, code coverage analysis.

BSIMM2 – Deployment Domain

Software Security Framework		
Domain	**Area**	**Activity Focus**
Deployment	Penetration Testing	Vulnerabilities in final configuration, feeds to defect management and mitigation.
	Software Environment	OS and platform patching, Web application firewalls, installation and configuration documentation, application monitoring, change management, code signing.
	Configuration & Vulnerability Management	Patching and updating applications, version control, defect tracking and remediation, incident handling.

Microsoft SDL


Security Training
Security Kickoff & Register With SWI
Security Design Best Practices
Security Architecture & Attack Surface Review
Threat Modeling
Use Security Development Tools & Security Best Dev & Test Practices
Create Security Documentation And Tools For Product
Prepare Security Response Plan
Security Push
Pen Testing
Final Security Review
Security Servicing & Response Execution
Requirements
Design
Implementation
Verification
Release
Support & Servicing

Microsoft SDLC

SDLC Life Cycle Phase	Security Focus Name	Brief Explanation
Requirements	Phase Training	SecureSDLC Training, Security "Buddies“
	Security Kickoff Meetings	Get everyone on the same page - build in, not bolt on
Design	Phase Training	SecureSDLC Training, Security "Buddies“
	Security Design Best Practices	Define security architecture and design guidelines
	Security Architecture & Attack Surface Review	Define supplemental ship criteria, Document the elements of the software attack surface
	Threat Modeling	The product team conducts threat modeling at a component-by-component level

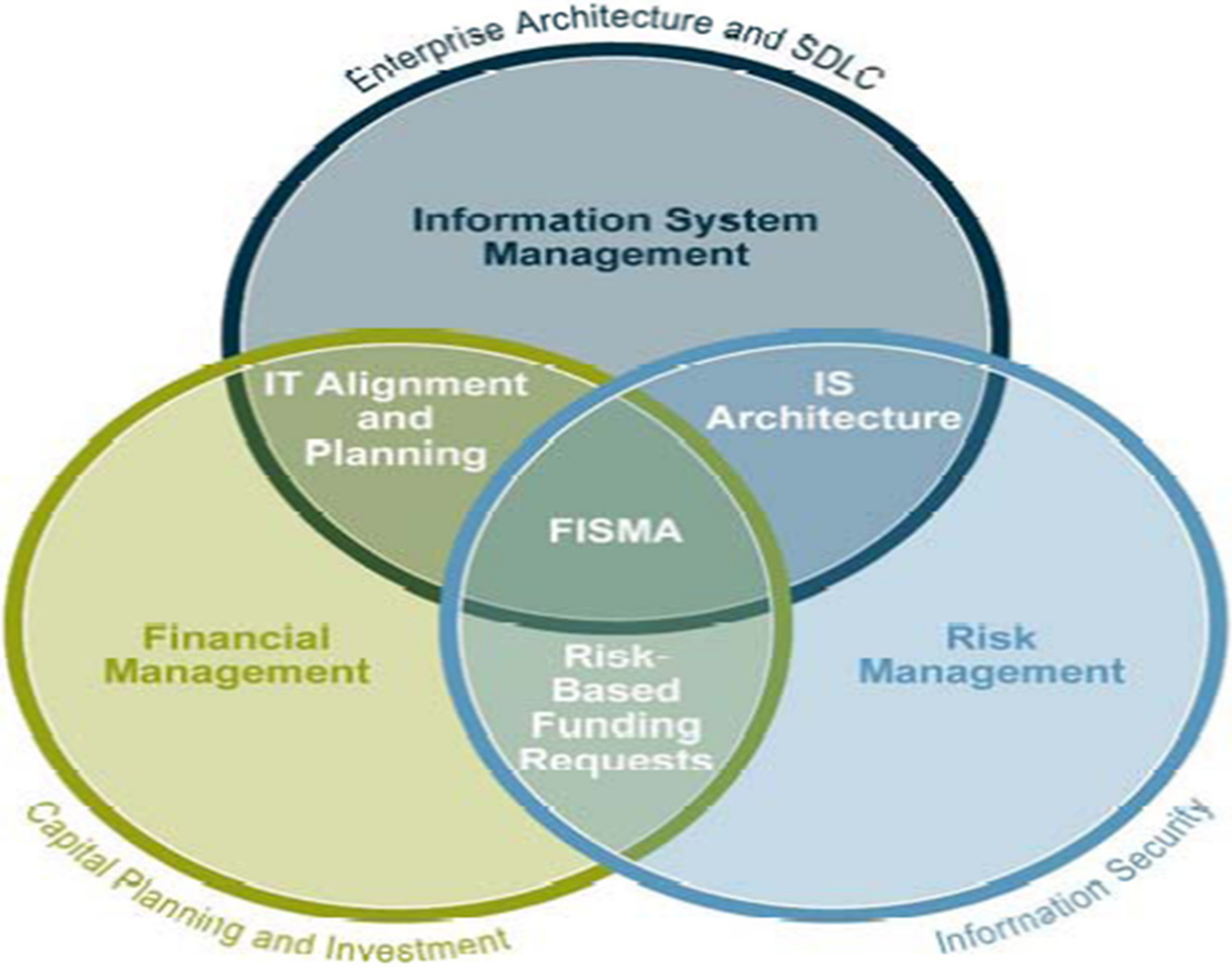
Microsoft SDLC

SDLC Life Cycle Phase	Security Focus Name	Brief Explanation
Implementation	Usage of Security Development Tools	Conduct code reviews, apply static-analysis code scanning tools
	Dev & Test Security Best Practices	Apply coding and testing standards, apply security-testing tools including fuzzing tools
	Documentation & Tools	Supporting documentation
Verification	Security Response Plan	Escalation and communication plan
	Security Validation/Verification	Security code reviews, focused security testing (especially on attack surface code areas)
	Penetration Testing	Test the current controls and configuration

Microsoft SDLC

SDLC Life Cycle Phase	Security Focus Name	Brief Explanation
Release	Final Security Review (FSR)	The FSR gives the product team and the organization's top management an overall picture of the security posture of the software and the likelihood that it will be able to withstand attack after it has been released to customers
	Dev & Test Security Best Practices	Apply coding and testing standards, apply security-testing tools including fuzzing tools
	Documentation & Tools	Creation of supporting documentation
Support & Servicing	Security Response Plan	Creation of an escalation plan
	Security Servicing	Conduct a post-mortem of reported vulnerabilities and taking action as necessary
	Response Execution	Prepare to evaluate reports of vulnerabilities and release security advisories and updates when appropriate

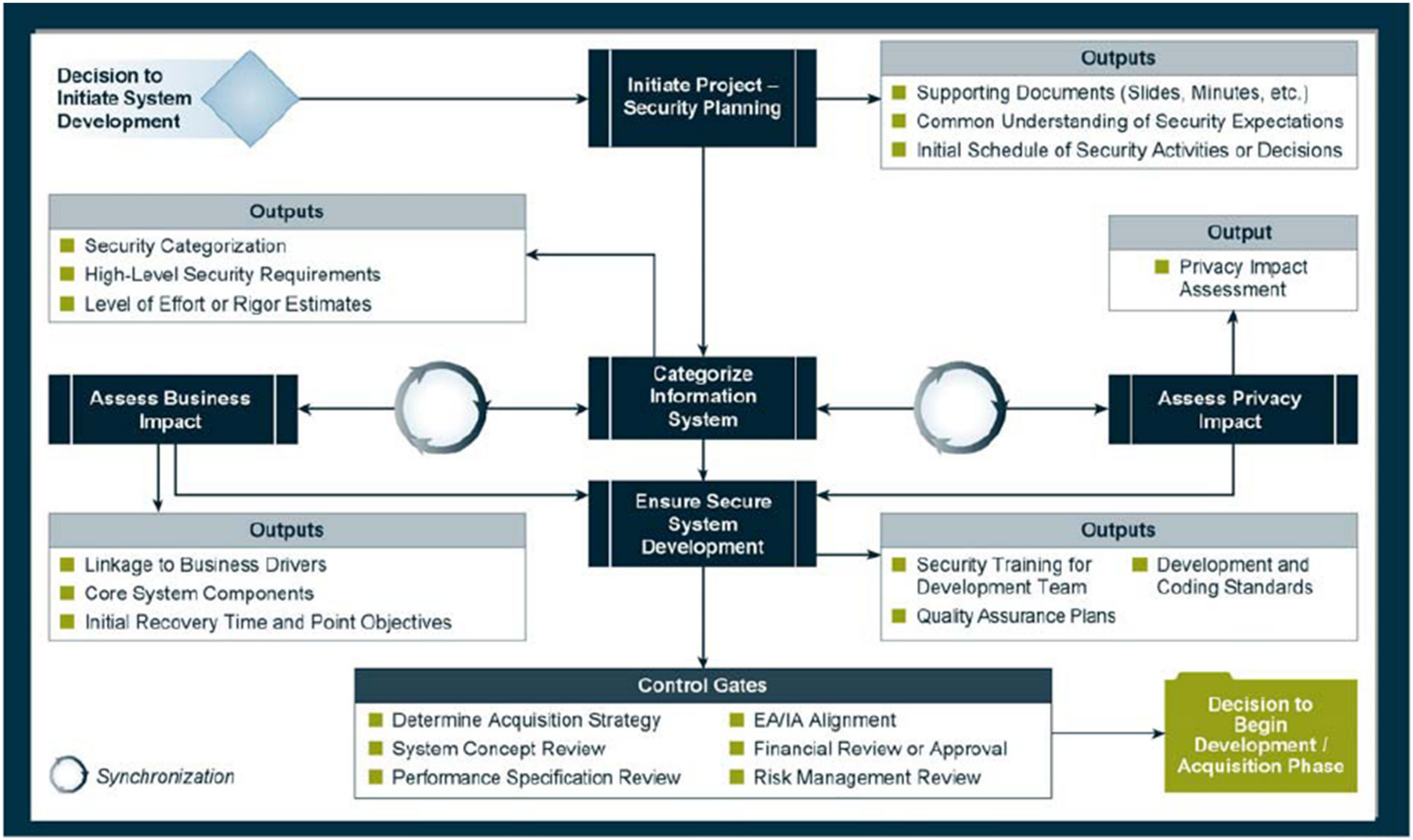
NIST 800-64 - Security Considerations in the System Development Life Cycle


Enterprise Architecture and SDLC
Information System Management
IT Alignment and Planning
IS Architecture
FISMA
Financial Management
Risk-Based Funding Requests
Risk Management
Capital Planning and Investment
Information Security

NIST 800-64 – Focus Areas

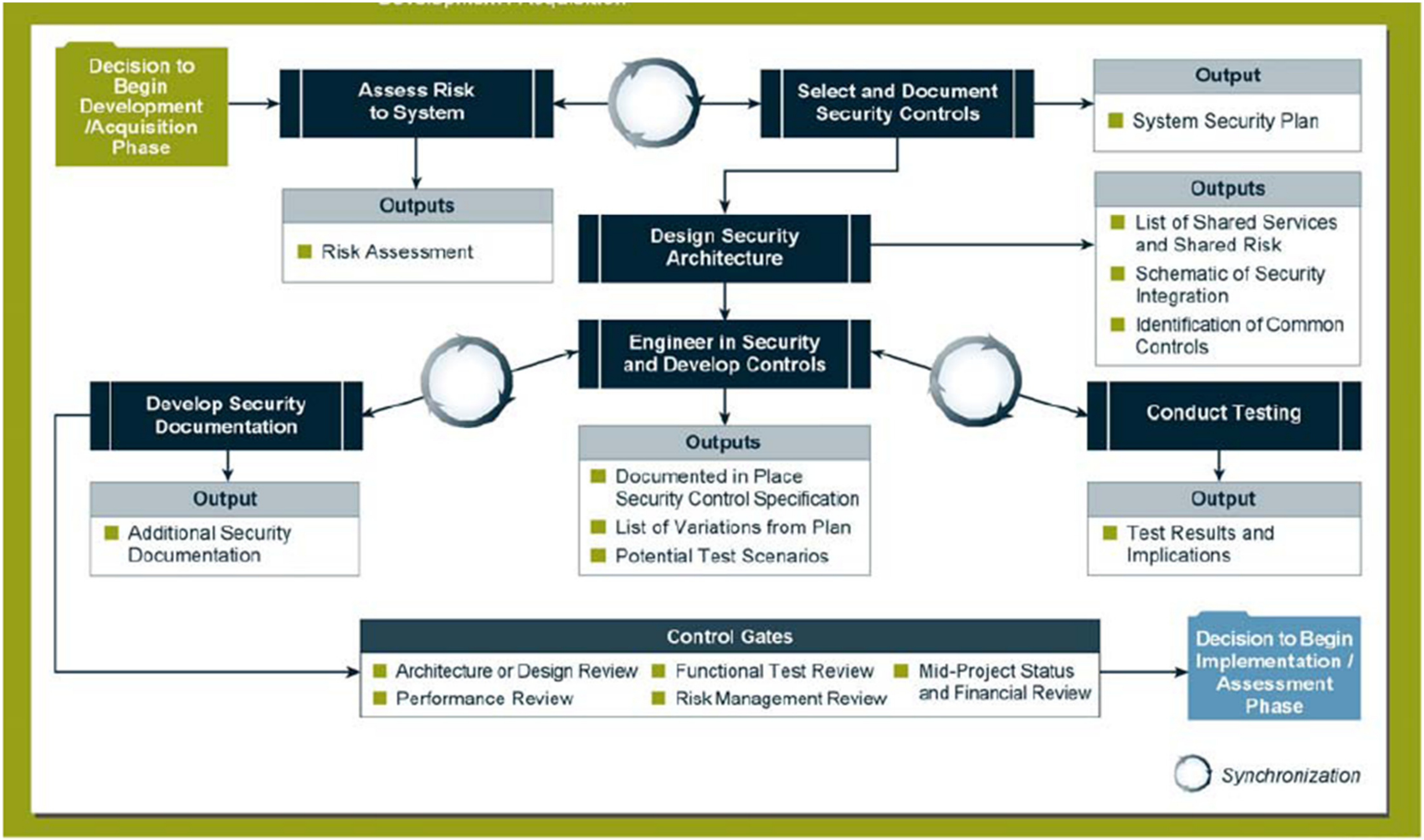
- **Integrated Security Model**
  - Plan
  - Acquire
  - Build-it-in
  - Deploy as an integral portion of the solution
- **Integral Program Points Across the Lifecycle**
  - Capital Planning and Investment Control Process
  - Security Architecture Activities
  - Legacy System Support
  - Roles & Responsibilities
- **SDLC/Security Points**
  - Major activity and milestone activities on a per phase basis
  - Security decision points/control gates
  - Specified outputs which provide relevant data into system design
  - Project accomplishments
  - System maintenance, security, and operational considerations
  - Synchronization and interdependencies

NIST 800-64 – Phase1: Initiation

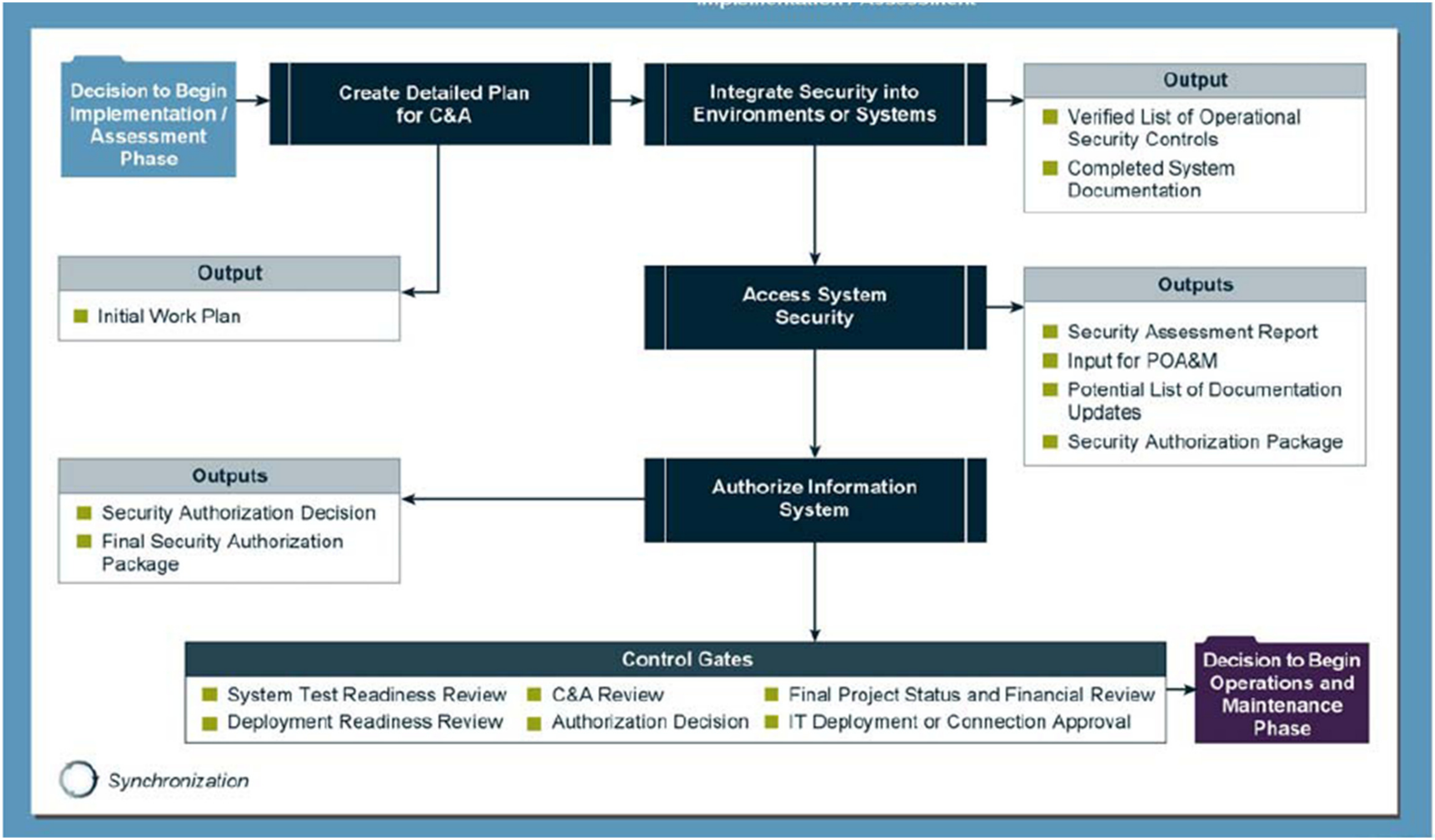

Decision to Initiate System Development
Initiate Project – Security Planning
Outputs
Supporting Documents (Slides, Minutes, etc.)
Common Understanding of Security Expectations
Initial Schedule of Security Activities or Decisions
Outputs
Security Categorization
High-Level Security Requirements
Level of Effort or Rigor Estimates
Output
Privacy Impact Assessment
Assess Business Impact
Categorize Information System
Assess Privacy Impact
Ensure Secure System Development
Outputs
Linkage to Business Drivers
Core System Components
Initial Recovery Time and Point Objectives
Outputs
Security Training for Development Team
Development and Coding Standards
Quality Assurance Plans
Control Gates
Determine Acquisition Strategy
System Concept Review
Performance Specification Review
EA/IA Alignment
Financial Review or Approval
Risk Management Review
Decision to Begin Development / Acquisition Phase
Synchronization

09-2030-NIST-Phase1

NIST 800-64 – Phase2: Development/Acquisition


Decision to Begin Development /Acquisition Phase
Assess Risk to System
Select and Document Security Controls
Output
System Security Plan
Outputs
Risk Assessment
Design Security Architecture
Outputs
List of Shared Services and Shared Risk
Schematic of Security Integration
Identification of Common Controls
Engineer in Security and Develop Controls
Develop Security Documentation
Conduct Testing
Outputs
Documented in Place Security Control Specification
List of Variations from Plan
Potential Test Scenarios
Output
Additional Security Documentation
Output
Test Results and Implications
Control Gates
Architecture or Design Review
Performance Review
Functional Test Review
Risk Management Review
Mid-Project Status and Financial Review
Decision to Begin Implementation / Assessment Phase
Synchronization

NIST 800-64 – Phase 3: Implementation/Assessment



Decision to Begin Implementation / Assessment Phase

Create Detailed Plan for C&A

Output
- Initial Work Plan

Integrate Security into Environments or Systems

Output
- Verified List of Operational Security Controls
- Completed System Documentation

Access System Security

Outputs
- Security Assessment Report
- Input for POA&M
- Potential List of Documentation Updates
- Security Authorization Package

Authorize Information System

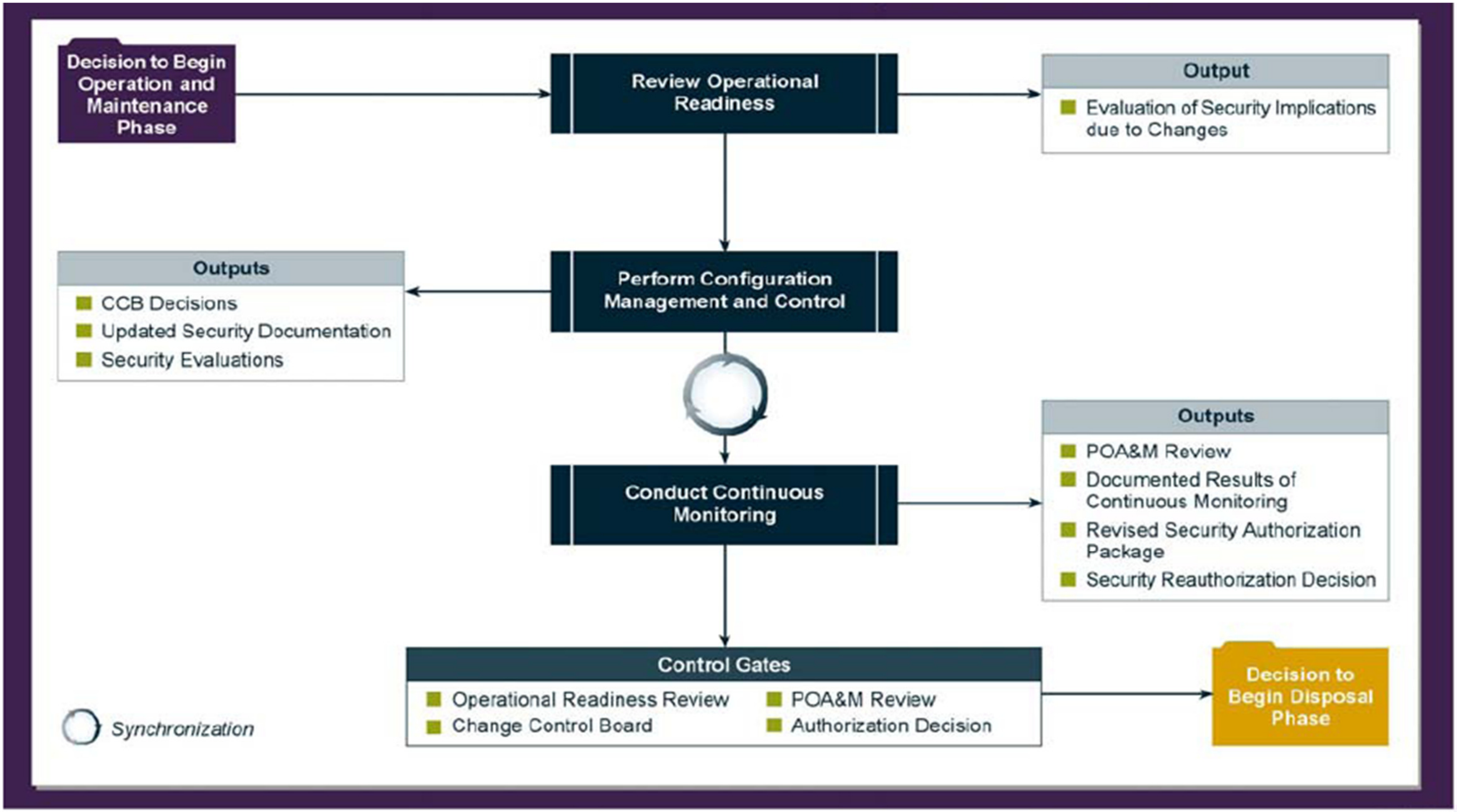
Outputs
- Security Authorization Decision
- Final Security Authorization Package

Control Gates
- System Test Readiness Review
- Deployment Readiness Review
- C&A Review
- Authorization Decision
- Final Project Status and Financial Review
- IT Deployment or Connection Approval

Decision to Begin Operations and Maintenance Phase

Synchronization

NIST 800-64 – Phase 4: Operations/Maintenance



Decision to Begin Operation and Maintenance Phase

Review Operational Readiness

Output

- 🟩 Evaluation of Security Implications due to Changes

Perform Configuration Management and Control

Outputs

- 🟩 CCB Decisions
- 🟩 Updated Security Documentation
- 🟩 Security Evaluations

Conduct Continuous Monitoring

Outputs

- 🟩 POA&M Review
- 🟩 Documented Results of Continuous Monitoring
- 🟩 Revised Security Authorization Package
- 🟩 Security Reauthorization Decision

Control Gates

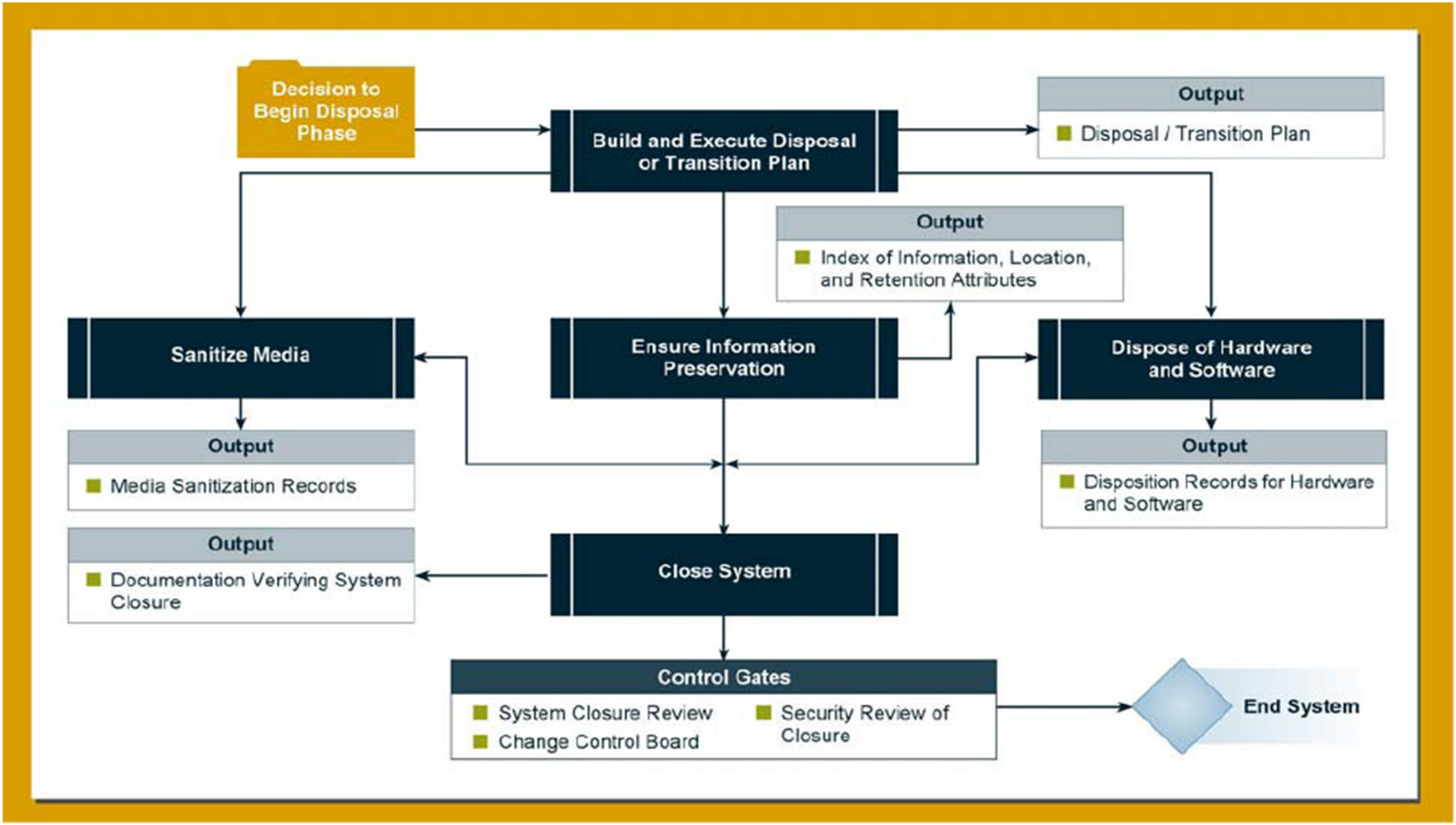
- 🟩 Operational Readiness Review
- 🟩 Change Control Board
- 🟩 POA&M Review
- 🟩 Authorization Decision

Decision to Begin Disposal Phase

Synchronization

09-3000-NIST-Phase4

NIST 800-64 – Phase 5: Disposal



Decision to Begin Disposal Phase

Build and Execute Disposal or Transition Plan

Output
- Disposal / Transition Plan

Output
- Index of Information, Location, and Retention Attributes

Sanitize Media

Ensure Information Preservation

Dispose of Hardware and Software

Output
- Media Sanitization Records

Output
- Disposition Records for Hardware and Software

Output
- Documentation Verifying System Closure

Close System

Control Gates
- System Closure Review
- Change Control Board
- Security Review of Closure

End System

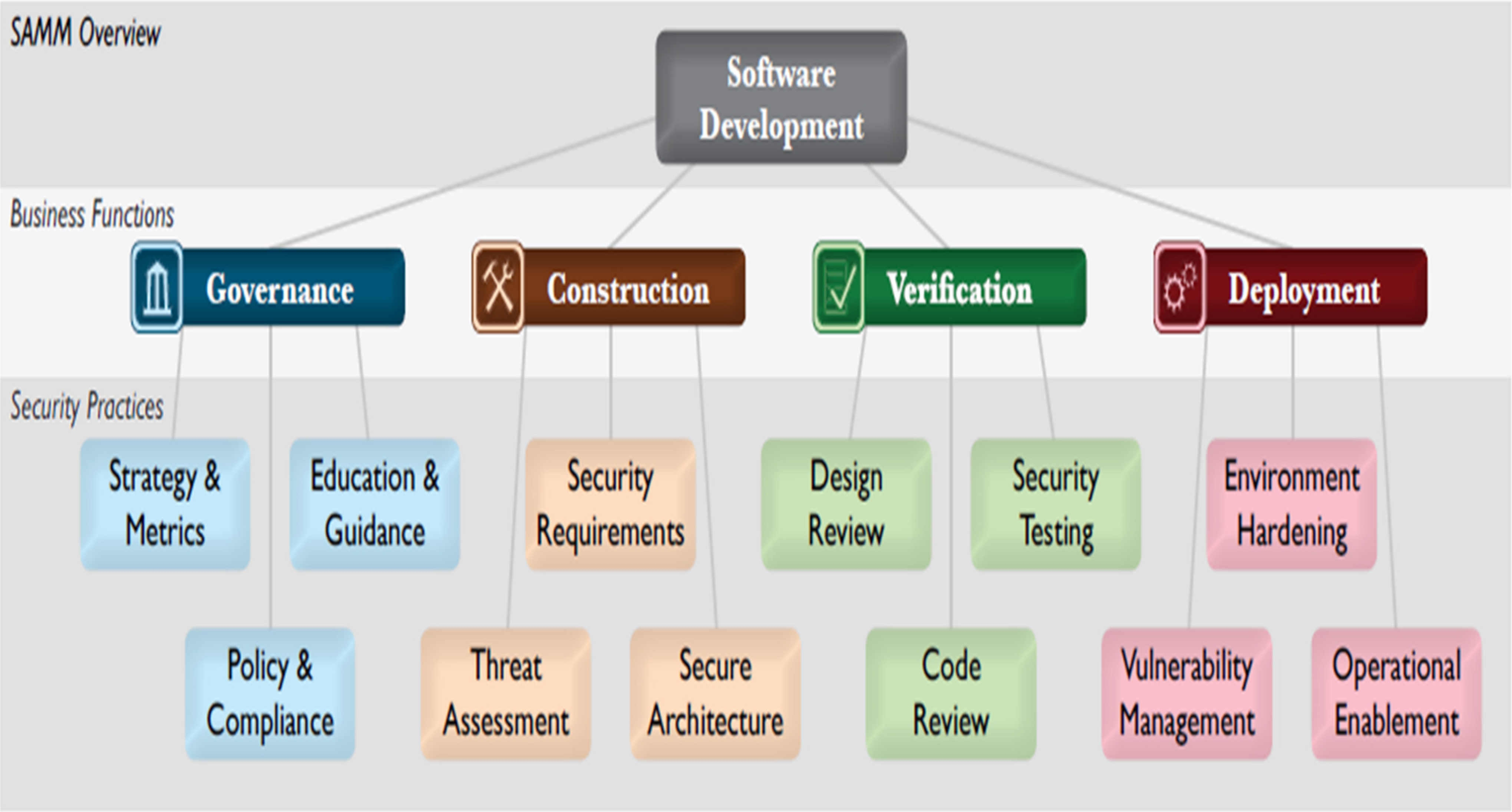
NIST 800-64 – Additional Information

- **Additional Security Focus Areas & Activities**
  - Security within the Supply Chain
  - Software Assurance
  - Service Oriented Architectures
  - Security Module Re-use
  - Cross Organizational Solutions (SLAs, MOA/MOU)
  - Technology Migrations
  - Facilities/Data Center Implementations
  - Virtualized Technologies

OpenSAMM

- Open, flexible framework
- Risk-based security strategy
- **Focus areas:**
  - Evaluation of existing software security practices
  - Building a cohesive and integrated software security assurance program
  - Implement concrete improvements
  - Define and measure security-related activities across the enterprise
- **"Words of Wisdom"**
  - Take it slow and works towards changing behavior over the long-term, balanced with short-term tangible security "wins"
  - No single solution fits all enterprises, program is adaptable and flexible based on risks
  - Ingrain changes in the long-term via creation of simple, well-defined, and measurable actions

OpenSAMM


SAMM Overview
Software Development
Business Functions
Governance
Construction
Verification
Deployment
Security Practices
Strategy & Metrics
Education & Guidance
Policy & Compliance
Security Requirements
Threat Assessment
Secure Architecture
Design Review
Security Testing
Code Review
Environment Hardening
Vulnerability Management
Operational Enablement

OSSTMM 3

- Comprehensive Security Test Cases for “in-sourced” teams
- Can also be used to perform “audits”
- Focused on the operational security layer and providing functional security verification
- **Correlates actions to**
  - Laws, Rules & Guidance
  - Processes & Technology
- **Main focus areas:**
  - Security Analysis
  - Operational Security
  - Trust Analysis
  - Work Flow Processing
  - Human Security
  - Physical Security
  - Wireless Security
  - Telecommunications Security
  - Data Network Security
  - Compliance

OWASP

- **Plethora of security resources**
  - Testing Guide
    - Testing activities across the SDLC
    - Testing Techniques
    - Security Requirements Test Derivation
  - Code Review Guide
    - Security Code Reviews across the SLDC
    - Security Code Review Coverage
    - Cross references to Technical Controls, Vulnerabilities, and Best Practices
    - Automated Tools
  - Developer Guide
    - Focus on development and deployment of web applications and related services
    - Identify Risks, define components, define functions, and perform threat modeling
    - Be aware of and properly handle the Top 10 threats

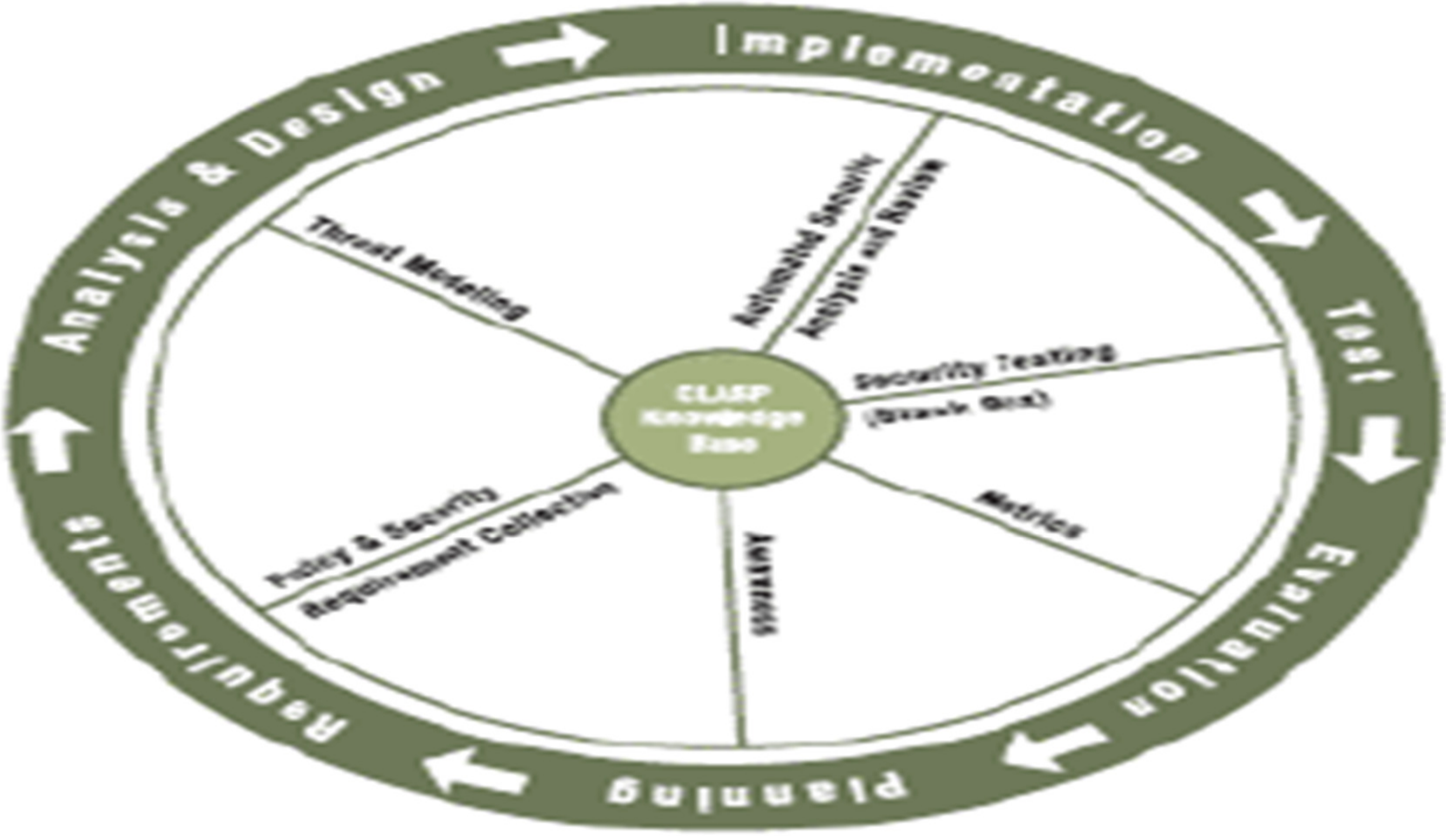
OWASP

- Application Security Verification Standard Project (ASVS)
  - **4 Levels**: Automated, Manual, Design, and Internal Verification
  - **14 requirement areas:**
    - Security Architecture Documentation
    - Authentication
    - Session Management
    - Access Control
    - Input Validation
    - Output Encoding/Escaping
    - Cryptography
    - Error Handling & Logging
    - Data Protection
    - Communication
    - HTTP
    - Security Configuration
    - Malicious Code
    - Internal Security

OWASP

- Enterprise Security API (ESAPI)
  - Multi-language control libraries
    - Security control interfaces
    - Reference implementation for each security control
    - Language/organization specific control implementation flexibility
  - Language Support
    - Java EE
    - JavaScript
    - .NET
    - Python
    - C
    - Ruby
    - Swingset
    - ASP
    - PHP
    - Cold Fusion/CFML

THE CLASP LIFECYCLE



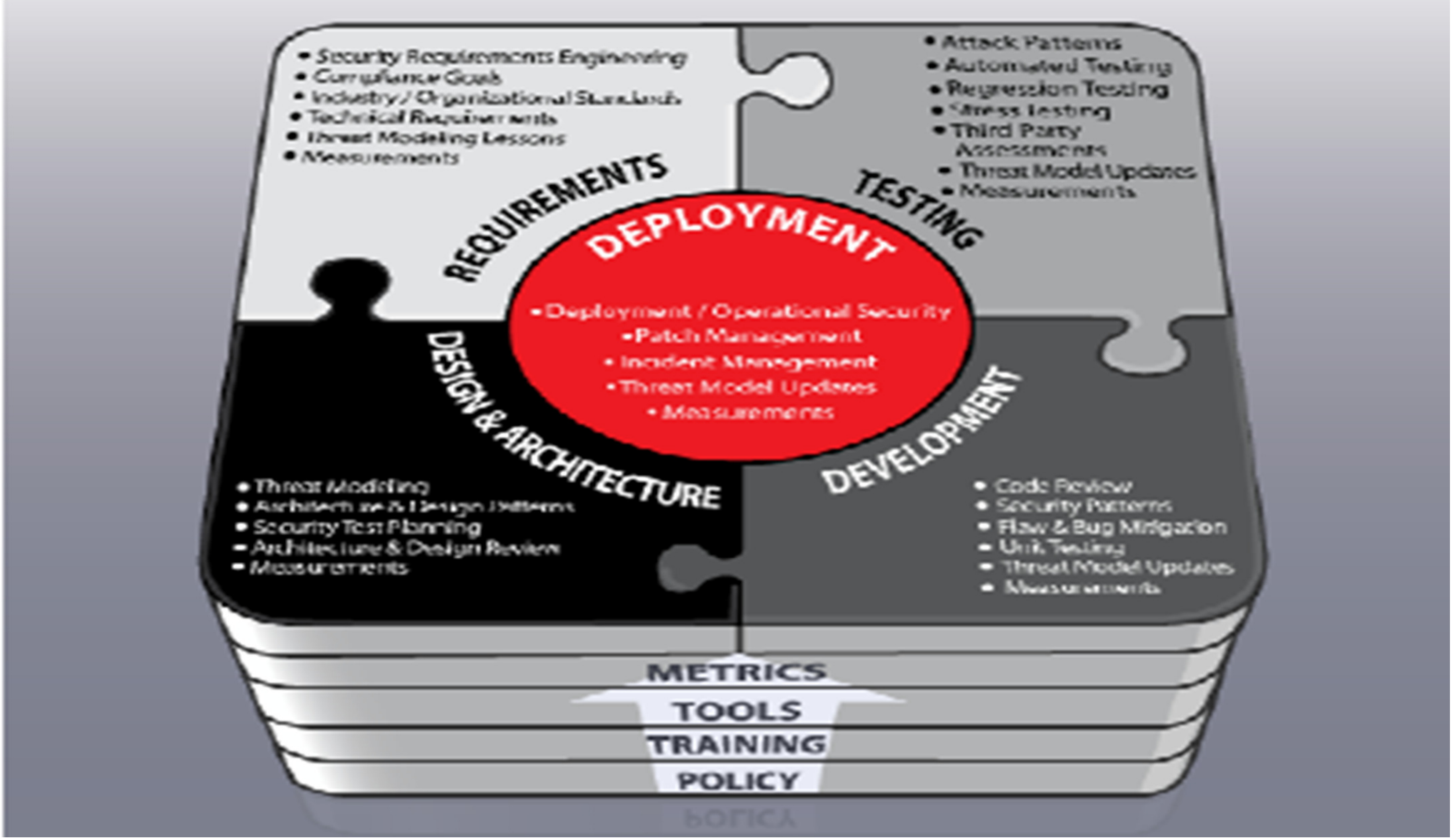
OWASP - CLASP

- OWASP Sponsored Security Related Initiative
  - **C**omprehensive
  - **L**ightweight
  - **A**pplication
  - **S**ecurity
  - **P**rocess
- Focus on moving security related activities earlier in the lifecycle (as well as corresponding quality related functions)
- Offers
  - Supporting data on key concepts, best practices, and core principles
  - Summarizes security services and defines the typical associated roles
  - Offers insight into vulnerabilities and provides access to a cohesive vulnerability spreadsheet
  - Offers guidance in the form of activity collections, advice on engineering activities and related roadmaps as it applies to the SDLC, as well as coding guidelines in checklist form

OWASP - CLASP

- The high-level process flow maps to **24** security related activities across **5** major high-level problem categories containing **104** typical problem types
- It also offers corresponding data on the consequences of non-mitigation, potential exposure time periods, and corresponding avoidance and mitigation techniques
- **Best Practices** include:
  - Institute awareness programs
  - Perform application assessments
  - Capture security requirements
  - Implement secure development practices
  - Build vulnerability remediation procedures
  - Define and monitor metrics
  - Publish operational security guidelines
- Mapping of **security resources** across the lifecycle to better understand and achieve related security goals in the areas of : access control, authentication, confidentiality, data integrity, availability, accountability, and non-repudiation is integral to the process

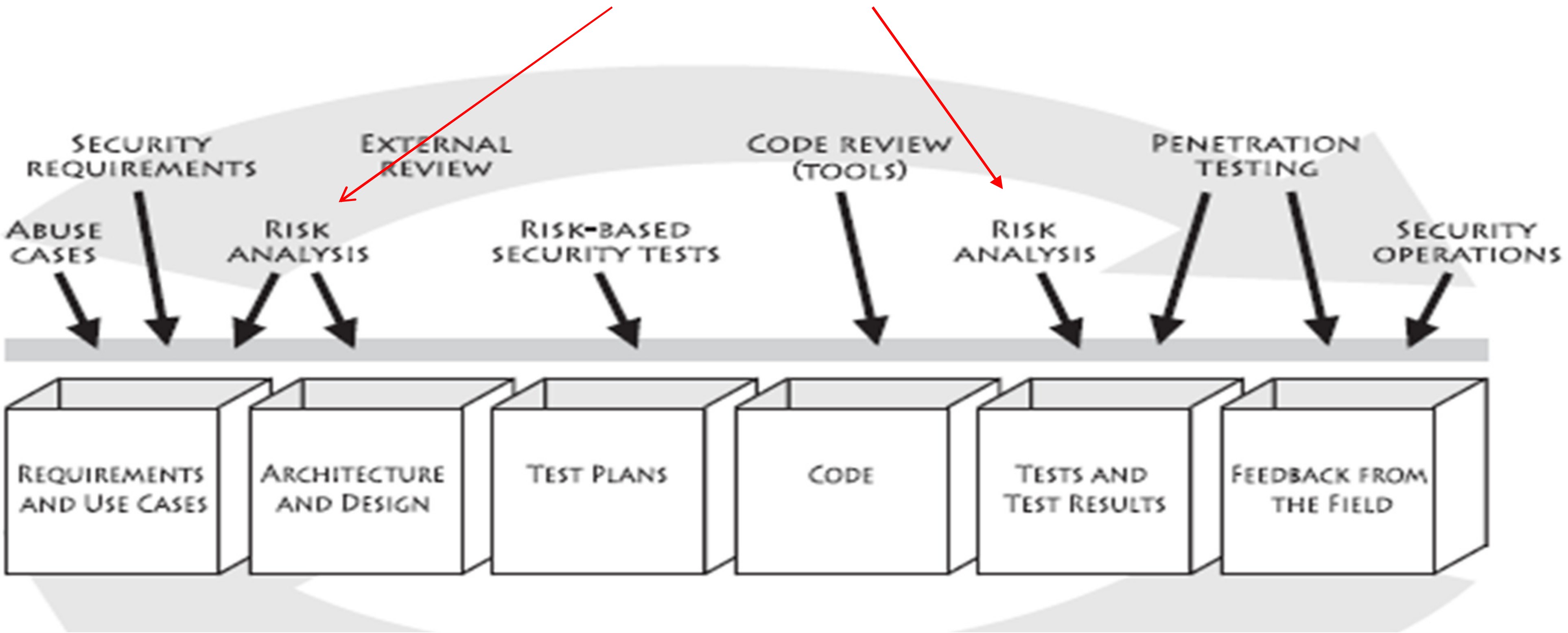
Secure SDLC (S-SDLC) Approach



Cigital's Security Touchpoints Model

- Cross SDLC phase approach to integrating security within the lifecycle with corresponding recommended activities

Architectural Risk Analysis



Touchpoint Model

- Architectural Risk Analysis
  - Activity: Build a one page system overview showing component interconnects
  - Process Steps:
    - Examine attack resistance
      - Utilize existing known attack approaches and patterns (STRIDE, CAPEC)
      - Examine existing taxonomy data (CWE, CVE, McGraw/Hogland [48 Attack Patterns], McGraw [7 Pernicious Kingdoms], Howard, LeBlanc, Viega [19 Deadly Sins])
      - Identify and flag components which are vulnerable to attack
      - Calculate risk-based impact
      - Checklists are useful during this step
    - Conduct ambiguity analysis
      - Examine system functionality, frameworks, and interfaces in-depth to identify additional potential attacks
      - Expose invalid assumptions (constructively)
      - Model the system functionality (Trust, Data Sensitivity, Threat)
      - Design relevant solutions (mitigations, controls, design, and code changes)

Touchpoint Model

- Process Steps (continued)
  - Conduct weakness analysis
    - Examine system dependencies
    - Focus on all aspects of system functionality and internal and external interconnedts
    - Thnk: In, Over, Under, Outside

- STRIDE (MS Model)
  - **S**poofing
  - **T**ampering
  - **R**epudiation
  - **I**nformation disclosure
  - **D**enial of Service
  - **E**levation of Privilege

Touchpoint Model

- Use the following methods to mitigate issues associated with each phase of the life-cycle
  - Identification of Security Requirements & Use of Abuse Cases in addition to other requirements and Use Cases
  - Use of Risk Analysis in addition to Architecture and Design best practices
  - Use of Code Reviews and Associated Tools
  - Use of Risk-based Security Tests in addition to other test plans, processes, and procedures
  - Inclusion of Security Operations findings in addition to feedback from the field, etc.

Software Engineering Institute Carnegie Mellon - CERT – Secure Coding Standards

- **Intent**
  - Essential resources for use within secure software development
  - Implementation and use of a uniform set of rules and guidelines
  - Sets baseline for future metrics
- **Priorities & Levels**
  - Each rule and recommendation is given a priority
  - Priorities are based on Failure Mode, Effects, and Criticality Analysis (FMECA)
  - Priority Components (Severity, Likelihood, and Remediation Cost)
- **Language Support**
  - Java
  - C
  - C++

SANS

- **Mapping of Top 20 Critical Controls to NIST 800-53 rev3 priority 1 controls**
- **Sub-control categorization**
  - Quick Wins
  - Improved Visibility and Attribution
  - Hardened Configuration & Improved IS Hygiene
  - Advanced
- **Additional control "metadata"**
  - Metrics
  - Testing and evaluation methods
- **Supports automated methods/tools as much as possible**

SANS – Top 20 Critical Controls

- **Automatable**
  - Inventory of authorized and Unauthorized devices
  - Inventory of Authorized and Unauthorized Software
  - Secure Configurations for Hardware and Software on Laptops, Workstations, and Servers
  - Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
  - Boundary Defense
  - Maintenance, Monitoring, and Analysis of Security Audit Logs
  - Application Software Security
  - Controlled Use of Administrative Privileges
  - Controlled Access Based on Need to Know
  - Continuous Vulnerability Assessment and Remediation
  - Account Monitoring and Control
  - Malware Defenses
  - Limitation and Control of Network Ports, Protocols, and Services
  - Wireless Device Control
  - Data Loss Prevention
- **Non-automatable**
  - Secure Network Engineering
  - Penetration Tests and Red Team Exercises
  - Incident Response Capability
  - Data Recovery Capability
  - Security Skills Assessment and Appropriate Training to Fill Gaps

SSE-CMM

- **Evaluation mechanism**
  - Describes the essential characteristics of an organization’s security engineering processes
  - Provides a way to measure and improve performance in the application of security engineering principles
  - 2 Primary categories
    - Security Engineering
      - o Engineering Processes
      - o Assurance Processes
      - o Risk Processes
    - Project and Organizational processes
  - 22 Process areas
  - Codified as ISO/IEC 21827 which is maintained by International Systems Security Engineering Association (ISSEA)

Importance of “Decision Gates”

Focus Point

Formalized *programs, policies, procedures, plans, and mitigation efforts all work ultimately towards achieving organization, team, customer, and personnel* ***success****. However, this is not without oversight cost to the organization. Since it is common knowledge that one cannot change what one cannot measure -- discovery, root cause analysis, and corrective action activities are necessary elements of a sustainable continuous improvement implementation.*

"Decision Gates" act as "public review forums" where entrance and exit criteria are examined for compliance, metrics and corresponding measurements are reviewed against expected values, and further continuous improvement activities are prioritized for execution.

“Decision Gates”

- Decision Gates are typically aligned with the major “**Phases**” of the SDLC
- Additionally, wherever a “**major**” milestone or deliverable exists within the schedule a decision gate should appear which addresses “go”, “no go”, “met”, “not-met” **decision analysis**
- **Enforcement/Compliance of Decision Gate criteria is key to the process**
- **Typical Decision Gate components**
  - Entrance Criteria
  - Exit Criteria
  - Relevant decision making bodies

“Definitions

- **Metric**
  - A rule used for quantifying some characteristic or attribute
- **Measurement**
  - A set of observations that reduce uncertainty

“Metrics”

- **Metric Attributes**
  - Simple and computable
  - Empirically and intuitively persuasive
  - Consistent and objective
  - Consistent in use of units and dimensions
  - Programming language independent
  - Provide an effective mechanism for quality feedback
- **Examples**
  - Functional Metrics
  - High Level Design Metrics
  - Component Level Metrics
  - Interface Design Metrics
  - Source Code Metrics
  - Testing Metrics
  - Maintenance Metrics

“Measurements”

- **Measurement Principles**
  - Formulation -- Appropriate for the solution under consideration
  - Collection -- Data accumulation mechanisms
  - Analysis -- Computation of metrics and application of mathematical tools
  - Interpretation -- Evaluation of the data in order to gauge quality level
  - Feedback -- Prioritized recommendations based on analysis findings
- **Take-aways**
  - Objectives of the measurements should be ascertained before data collection begins
  - All metrics should be unambiguous in nature
  - Metrics should be drawn from the proper domain of application
  - Metrics should be tailored specifically for a specific product or process
- **Critical points**
  - Automate whenever possible
  - Utilization of valid statistical techniques
  - Creation and utilization of guidelines and recommendations should be established

Examples

FURPS

- **Quality Factor Areas**
  - **F**unctionality
  - **U**sability
  - **R**eliability
  - **P**erformance
  - **S**upportability

Extended FURPS

Quality Factor Areas

- Auditability
- Accuracy
- Communication Commonality
- Completeness
- Complexity
- Concision
- Consistency
- Data commonality
- Error tolerance
- Execution efficiency
- Expandability
- Generality
- Hardware Independence
- Instrumentation
- Modularity
- Operability
- Security
- Self documentation
- Simplicity
- System Independence
- Traceability
- Training

Extended FURPS

- **Software Quality Metrics**
  - Correctness
  - Reliability
  - Efficiency
  - Integrity
  - Maintainability
  - Flexibility
  - Testability
  - Portability
  - Reusability
  - Interoperability
  - Usability

Methods of Gauging “Readiness”

Focus Area

Methods for gauging readiness vary depending on the formalness and comprehensiveness of the various programs being utilized within the enterprise.

Most important within this area is the selection of criteria which is most applicable to the enterprise's mission assurance goals.

Defect Removal Efficiencies (DREs)

- DRE = $E / (E + D)$

  Where E = number of errors found before customer delivery
  Where D = number of defects found after customer delivery

- DRE$_i$ = $E_i / (E_i + E_{i+1})$

  Where E_i = number of errors found during software engineering activity
  Where E_{i+1} = number of defects found during software engineering activity $_{i+1}$ that are traceable to errors which were not discovered in software engineering activity *i*

  Focus is on ensuring that DRE = 1; and DRE$_i$ = 1 (no defects released to the customer or to later engineering cycles)

Predicting Software Quality

- Uses the Process Quality Index which is a method to predict whether unit-tested components will cause down-stream defects
- Allows for enterprises to take proper corrective actions to reduce test and rework time
- Strive for defect free components and use that as a metric
- Measurements:
  - Design Time to Code time comparison
  - Design Review Time as a percentage of Design time (Goal: 50% or greater)
  - Code Review Time as a percentage of Coding Time (Goal: 50% or greater)
  - Compile Defects per KLOC (< 10 defects per KLOC)
  - Unit Test Defects per KLOC (< 5 defects per KLOC)

Predicting Software Quality

- Typical Project Benchmark Metrics
  - Schedule Deviation
  - Number of Defects in 100,000 lines of code
  - Percentage of defects removed prior to system test
  - Percentage of development time fixing system test defects
  - Cost of quality
  - Product Warranty duration

Use of Magic Numbers

- Key Performance Indicators (KPI)
  - Measures of performance, used for definition and evaluation of success against enterprise goals

Weighted Risk Trend (**WRT)**
Business-based representation of risk over a time period or through repeated iterations of development.

Defect Remediation Window (**DRW**)
Duration of time from defect identification through closure

Rate of Defect Reoccurrence (**RDR**)
Rate of defect re-introduction back into a solution, over time.

Specific Coverage Metric (**SCM**)
Coverage of total functionality that testing has achieved; Total functionality = known functionality + discovered functionality (use of UI, API, code coverage, + advanced discovery tools)

Security to Quality Defect Ratio (**SQR**)
Ratio of security defects to total number of defects

Utilization of Resources to Enhance Quality, Cut Costs, and Improve Security and Productivity

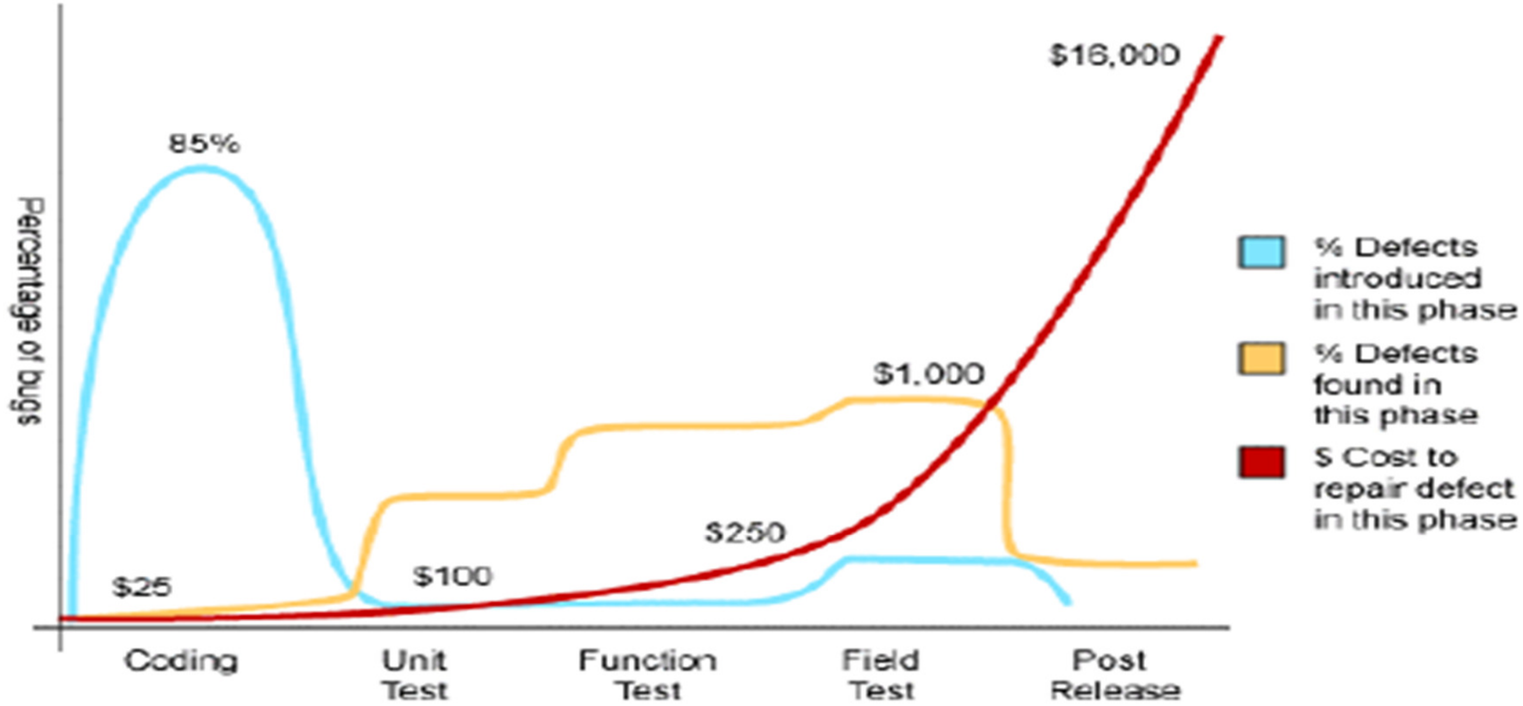
Focus Area

An essential art of an “enterprise-wide” quality, security, project or risk management program is support for re-allocatable resources with which to combat major programmatic and enterprise related issues.

*Mature organizations use personnel **holistically**, across the system development life cycle in order to isolate and eradicate problems as early as possible, so as to minimize costs, maximize schedule, and more effectively utilize resources.*

“Where Should Test Engineering Resources be Deployed? ”

- Answer – **EVERYWHERE** within the SDLC
- Defects **cost less** when found earlier



Source: Applied Software Measurement, Capers Jones, 1996

Quantification Methods

- Security **costs less** when “built-in”
- Common *quantification* mechanisms
  - **Return on Security Investment (ROSI)**

$$\textbf{ROSI} = \frac{(\text{Risk Exposure} * \%\ \text{Risk Mitigated}) - \text{Solution Cost}}{\text{Solution Cost}}$$

  - **Total Cost of Ownership (TCO)**

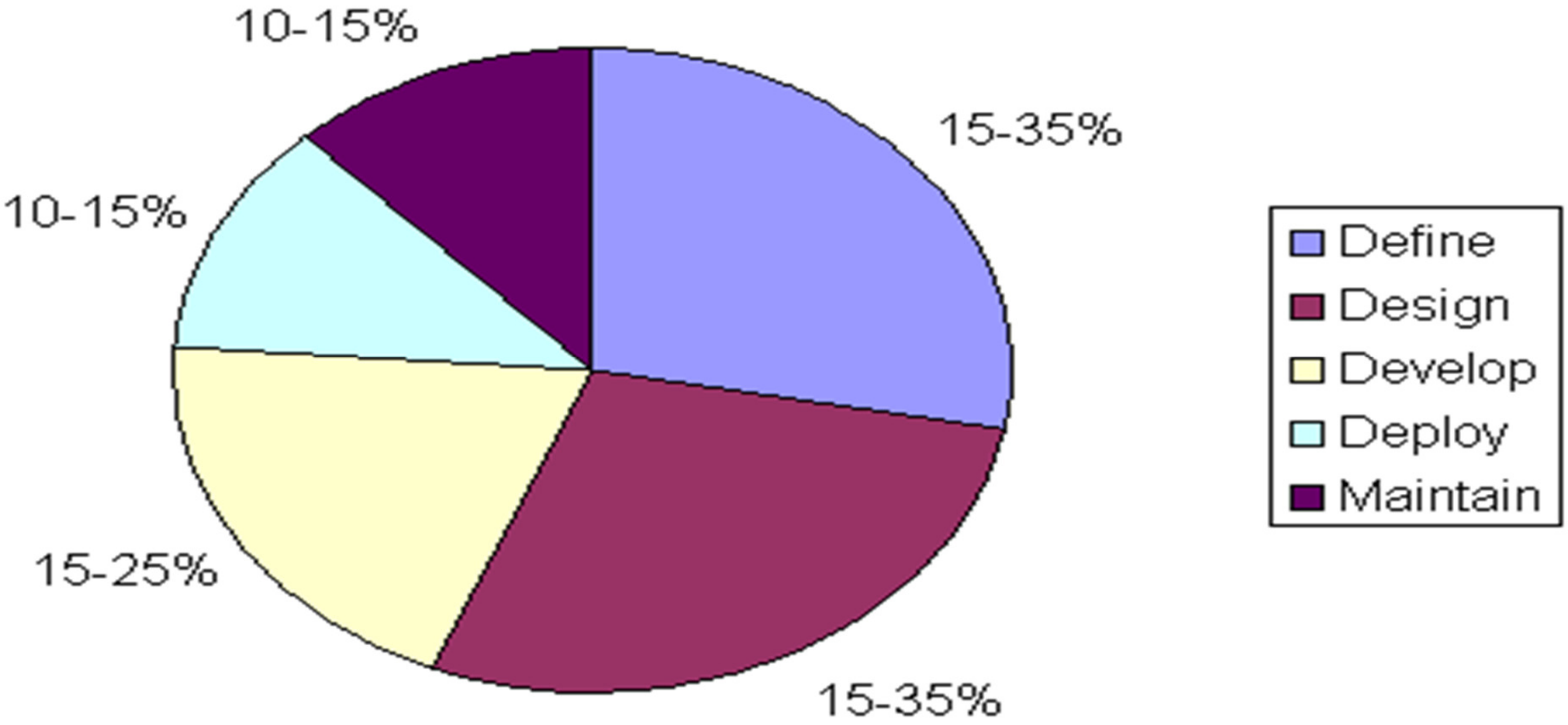
TCO = (cost to purchase + cost to install + cost to operate + cost to maintain)

  - **Cost Benefit (CB) Analysis**

$$\textbf{CB} = \frac{\text{Total Benefits}}{\text{Total Implementation Costs}}$$

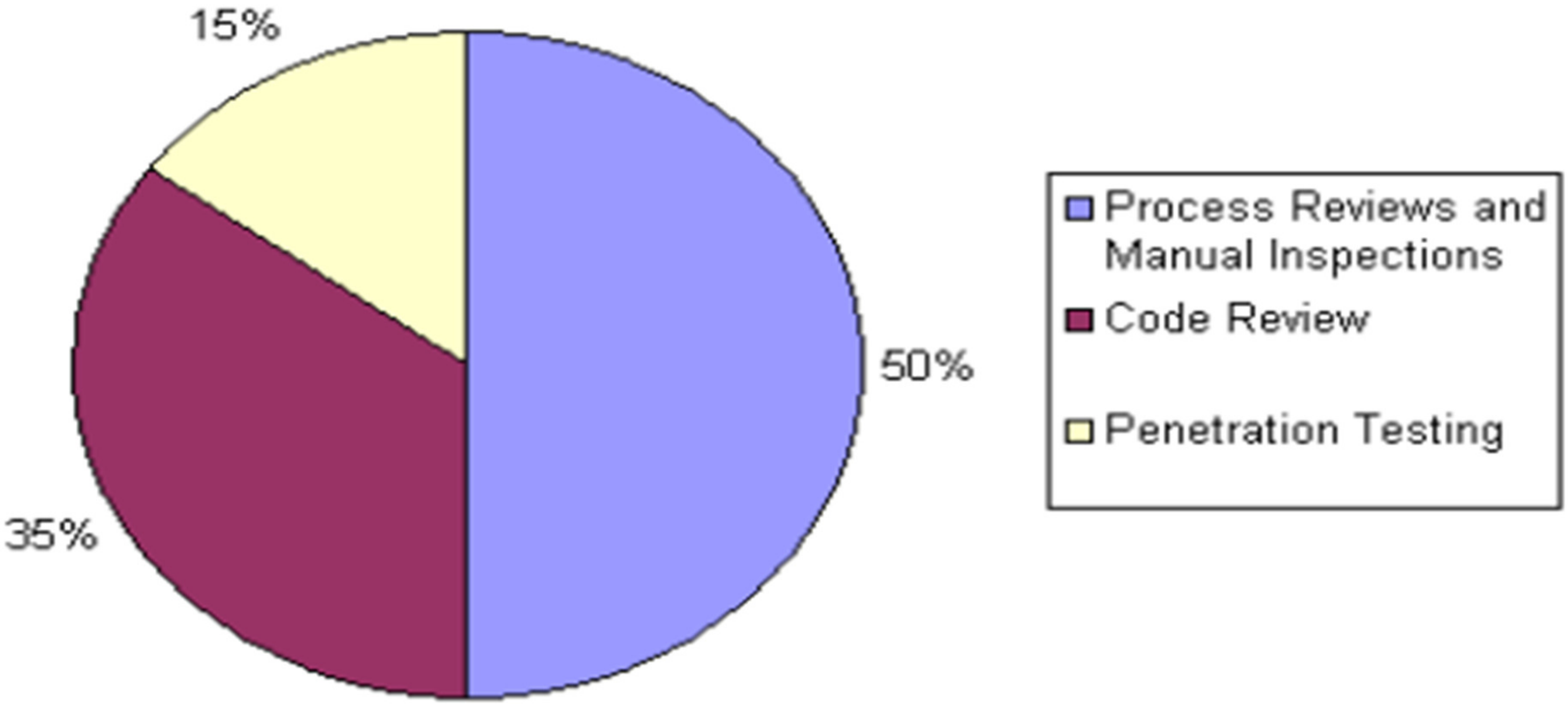
Balanced Approach

- OWASP documents that a balanced approach to deployment of test personnel across the SDLC, to ensure that defects are caught earlier in the cycle



Early in the Cycle

- Additionally, they document the need for personnel across various types of testing activities related to security in order to find defects earlier in the cycle



“Importance of Continuous Improvement”

- Continuous Monitoring Applies to Organizations as well as Security
- Allows for “Lessons Learned” to be implemented across the Lifecycle
- Keeps the personnel and organizations motivated and focused on corrective actions
- Enables organizational maturity and operational excellence

Final Message

It Takes Everyone’s Effort to Make the Necessary Changes

*Each and every one of you is **vital** and instrumental in making these changes happen throughout various organizations within industry.*

It will take your efforts, willingness, and vision to ensure that the “possible” becomes the new “reality”.

*The Challenge -- **GET and STAY INVOLVED**!*

Questions ??

References

References

[1] http://upload.wikimedia.org/wikipedia/commons/b/bb/Systems_Development_Life_Cycle.jpg

[2] INCOSE, SE Handbook, January 2010

[3] http://en.wikipedia.org/wiki/Waterfall_model

[4] http://en.wikipedia.org/wiki/Spiral_model

[5] http://en.wikipedia.org/wiki/Rapid_application_development

[6] http://en.wikipedia.org/wiki/Agile_software_development

[7] http://en.wikipedia.org/wiki/Lean_software_development

[8] http://en.wikipedia.org/wiki/Total_quality_management

[9] http://www.iso.org/iso/iso_9000_essentials

[10]http://www.iso.org/iso/iso_catalogue/management_and_leadership_standards/quality_management/qmp.htm

[11] http://www.nist.gov/baldrige/publications/criteria.cfm

References

[12] http://en.wikipedia.org/wiki/Six_Sigma

[13] http://en.wikipedia.org/wiki/Capability_Maturity_Model

[14] The Capability Maturity Model, CMU-SEI, 1995

[15] McFeeley, B. (1996). IDEAL: A User's guide of Software Process Improvement. CMU/SEI-96-HB-001

[16] http://en.wikipedia.org/wiki/Risk_management

[17] http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf

[18] http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf

[19] http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-39

[20] http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf

[21] http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf

References

[22] http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-137

[23] http://www.cert.org/resilience/download/CERT-RMM_v1.0.pdf

[24] http://www.sei.cmu.edu/reports/10tr012.pdf

[25]] http://www.microsoft.com/security/sdl/

[26] http://msdn.microsoft.com/en-us/library/ms995349.aspx

[27] https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/326-BSI.html

[28] http://bsimm.com/

[29] https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/326-BSI.html

[30] http://www.owasp.org/index.php/ASVS

[31] http://www.owasp.org/index.php/ESAPI

[32] http://www.owasp.org/index.php/Category:OWASP_Testing_Project

References

[33] http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project

[34] http://www.owasp.org/index.php/Category:OWASP_Guide_Project

[35] http://www.opensamm.org/

[36] http://www.isecom.org/mirror/OSSTMM.3.pdf

[37] http://www.sans.org/critical-security-controls/guidelines.php

[38] https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards

[39] Pressman, R. (1997). Software Engineering A Practitioner’s Approach. 4th Edition.

[40] Seshagiri, G. (2010). Predicting Software Quality Early in the SDLC and Producing Secure Software

[41] Los, R. (2010). Magic Numbers An In-Depth Guide to 5 Key Security Metrics for Web Application Security

References

[42] http://www.blackhat.com/presentations/bh-usa-06/bh-us-06-Morana-R3.0.pdf

[43] http://www.infosecwriters.com/text_resources/pdf/ROSI-Practical_Model.pdf

[44] http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6918/ps9145/ps9152/white_paper_c11_512820.pdf

[45] http://www.owasp.org/index.php/Testing_Guide_Introduction

[46] http://www.google.com/url?sa=t&source=web&cd=3&ved=0CCgQFjAC&url=http%3A%2F%2Fwww.cert.org%2Farchive%2Fpdf%2FSQUARE_Cost.pdf&rct=j&q=cost%20benefit%20analysis%20example%20and%20security&ei=vxESTe3qF4GdlgeW2sn5Cw&usg=AFQjCNHzSJ3eNcL00avMAhM-wIuxvxd_9Q

[47] http://www.owasp.org/index.php/CLASP_Security_Principles

[48] http://www.owasp.org/index.php/Category:OWASP_CLASP_Project

[49] http://secappdev.org/handouts/2010/Gary%20McGraw/ARA%20lo%20res.pdf

References

[50] http://www.cigital.com/services/security/

[51] http://www.blackhat.com/presentations/bh-usa-06/bh-us-06-Morana-R3.0.pdf