Welcome Back to the CISSP Bootcamp

Your instructors:

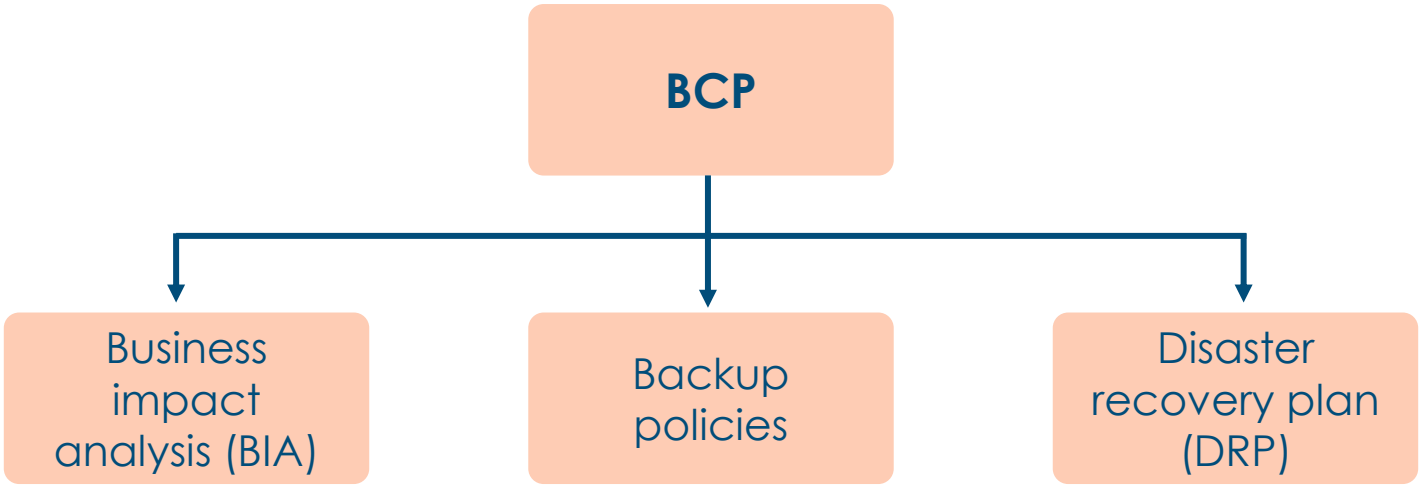
Michael J Shannon

and

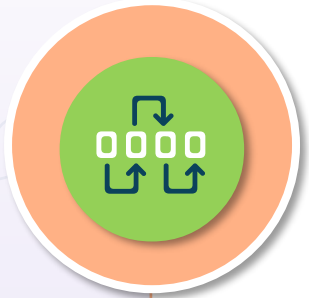
Carl Mullin

- **Class will begin at 10:00 A.M. Central Standard Time (CST)**

Business Continuity Planning (BCP)



Business Impact Analysis (BIA)



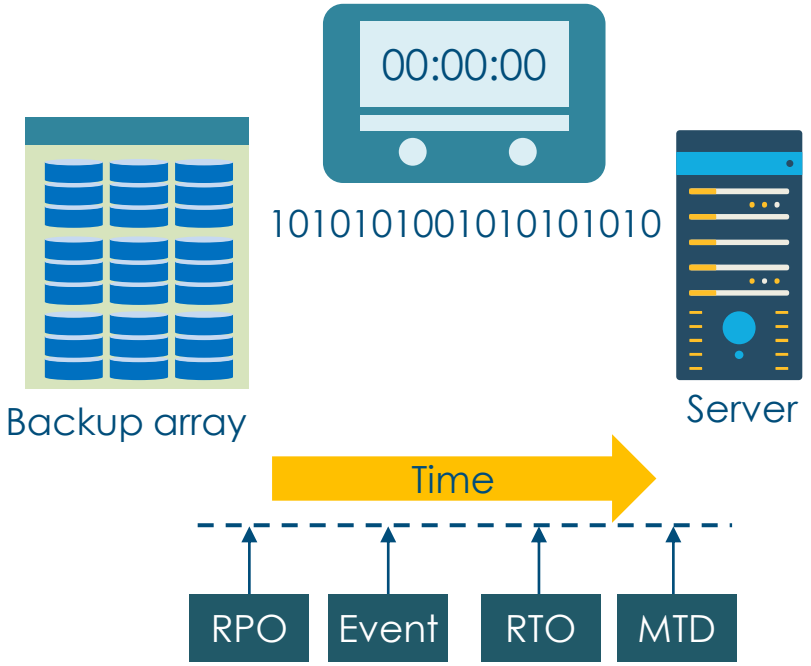
- The risk assessment aspect of the business continuity plan (BCP)
- Identify critical functions to the business and prioritize them based on need for survival
- Gather meaningful metrics and indicators
- Identify the risks associated with the critical functions
- The probability of the risk occurring (likelihood)
- The impact the risk will have (magnitude)
- Identify how to eliminate or reduce the risk

Key BIA Terminology

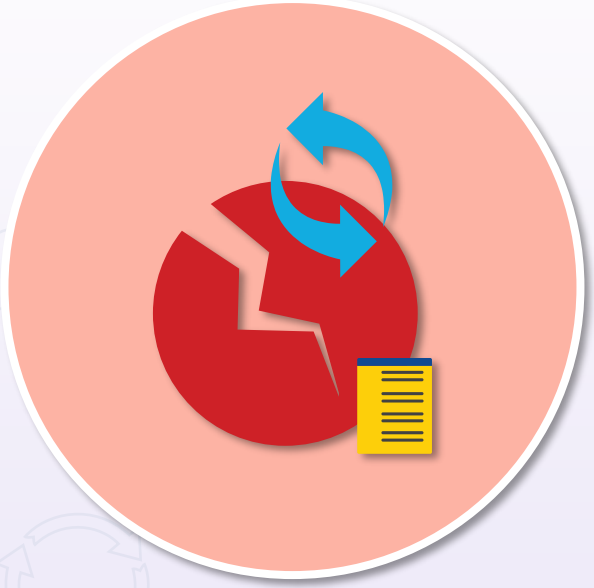
- **Recovery time objective (RTO)**
  - The target amount of time within which a process must be restored after disruption
- **Recovery point objective (RPO)**
  - The maximum targeted period in which an asset or data may be lost from an IT service due to a major event
- **Mean time to repair (MTTR)**
  - The average time needed to repair a failed system or module
- **Mean time between failures (MTBF)**
  - The number of failures per million hours for a product
- **Maximum Tolerable Downtime (MTD)**
  - Absolute maximum amount of time that a resource, service, or function can be unavailable before we start to experience a loss

Recovery Time Objective (RTO)

- The amount of time available to recover the resource, service, and function
- Must be equal to or less than MTD
- Any solutions must be accomplished within this time frame, or it is considered loss
  - Add physical security
  - Add redundancy
  - Purchase insurance
  - Invest in backup generators
  - Invest in faster supply chains
  - Safeguard media off-site



Recovery Point Objective (RPO)



- The point in time, relative to a disaster, where the recovery process begins
- In IT systems, it is often the point in time when the last successful backup is performed before the disruptive event occurs
  - How much work can be lost if a disruption occurs?
  - What impact will it have?
  - How do we make sure we don't lose more than "X" information?

Mean Time between Failures (MTBF)

- A measure of how reliable a hardware system or component is
- For most devices, the measure is in thousands or tens of thousands of hours between failures
- For example, an SSD drive may have a mean time between failures of 10 years

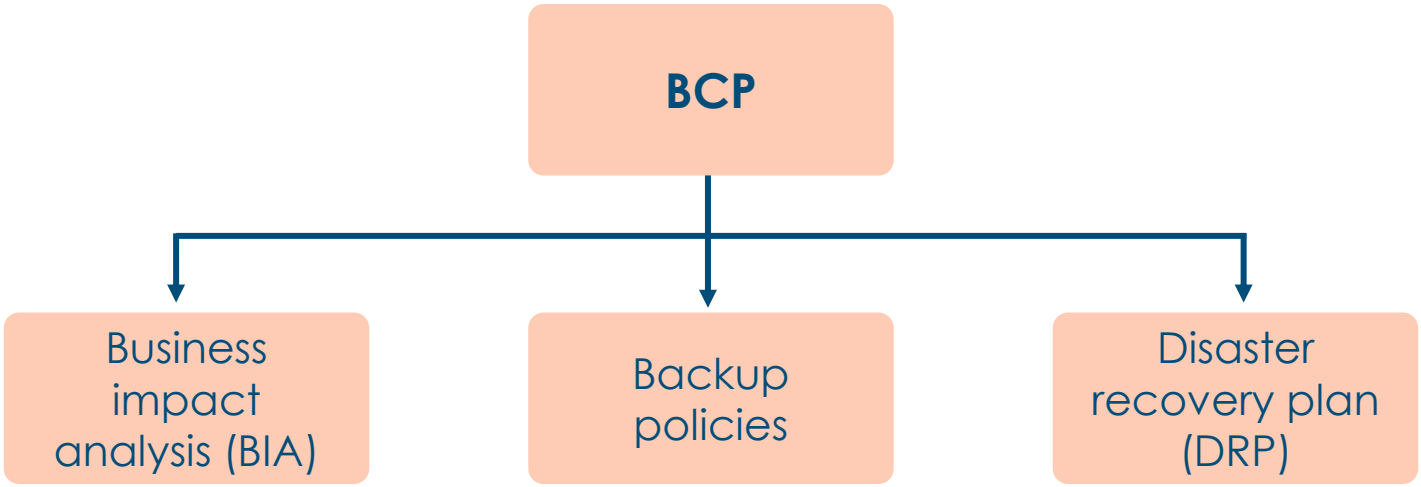
Mean Time to Repair (MTTR)

- How long does it take to repair?
- Measures time to fix
- Average value predicted based on experience and documentation
- (Total down time)/(number of breakdowns)

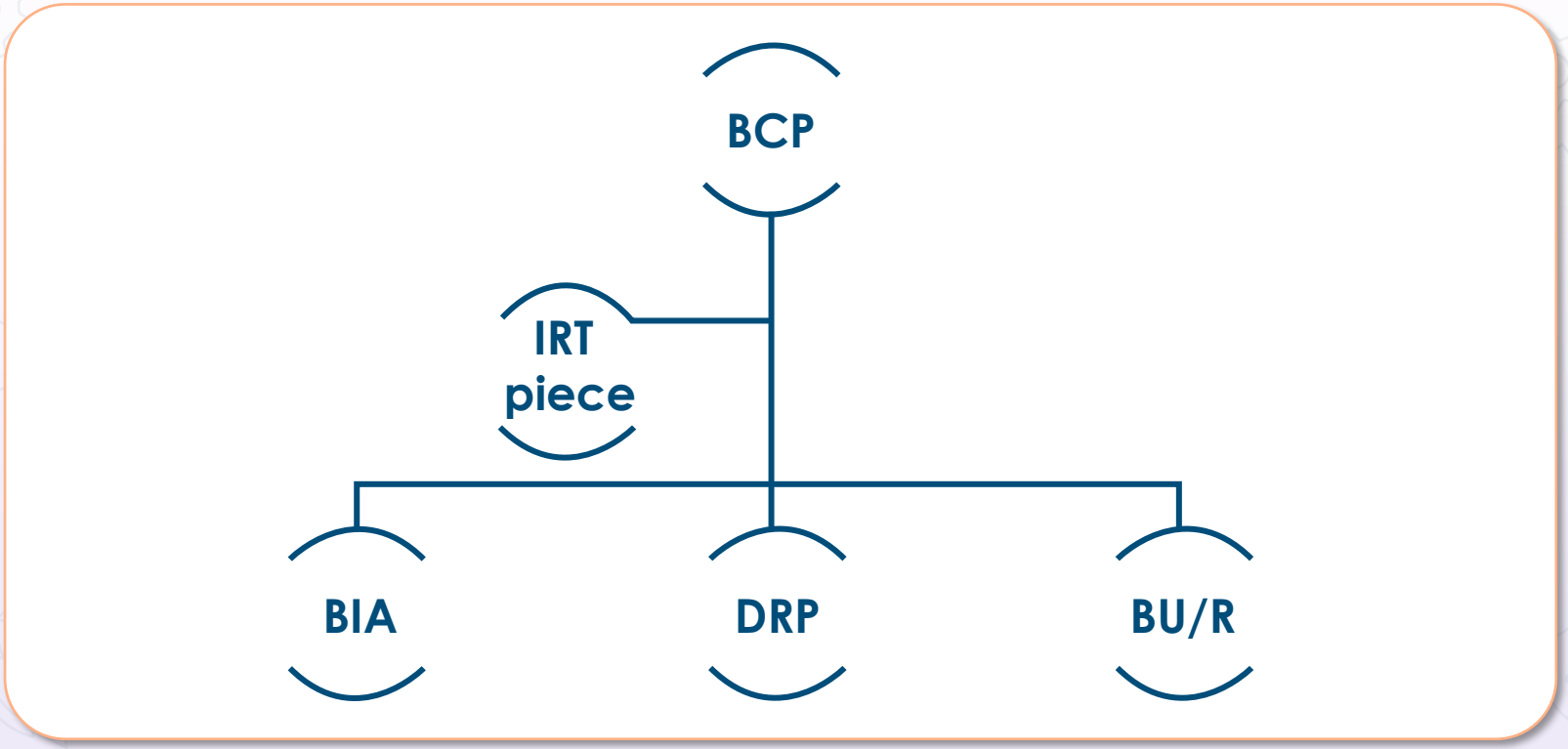
Business Continuity Planning (BCP)

- BCP involves the preparation of all activities and procedures deployed to avert the loss of critical business functions and services for a pre-determined acceptable amount of time
- For government agencies and other non-commercial entities, the term "Continuity of Operations" (COOP) is often used instead

Business Continuity Planning (BCP)



Continuity of Operations (COOP)



Disasters

Environmental	Man-made intentional	Man-made unintentional
• Earthquakes • Wildfires • Flooding • Snow • Tsunamis • Hurricanes • Tornadoes • Landslides • Asteroids	• Arson • Terrorist • Political • Break-ins • Theft • Damage • File destruction • Information disclosure	• Mistakes • Power outage • Illness • Epidemics • Information disclosure • Damage • File destruction • Coding errors

BCP from Ready.gov

Business impact analysis

- Develop questionnaire
- Conduct workshop to instruct business function and process managers how to complete BIA
- Receive complete BIA questionnaire forms
- Review BIA questionnaires
- Conduct follow-up interviews to validate information and fill any gaps

Recovery strategies

- Identify and document resource requirements based on BIAs
- Conduct gap analysis to determine gaps between recovery requirements and current capabilities
- Explore recovery strategy options
- Select recovery strategies with management approval
- Implement strategies

Plan development

- Develop plan framework
- Organize recovery teams
- Develop relocation plans
- Write business continuity and IT disaster recovery procedure
- Document manual workarounds
- Assemble plan
- Validate and gain management approval

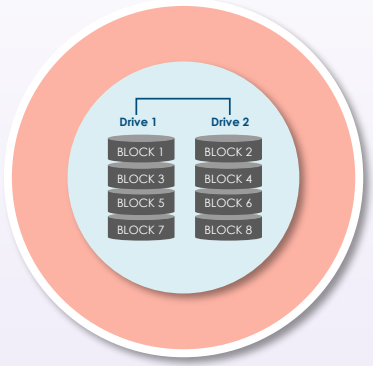
Testing & exercises

- Develop testing, exercise, and maintenance requirements
- Conduct training for business continuity team
- Conduct orientation exercises
- Conduct testing and document test results
- Update BCP to incorporate lessons learned from testing and exercises

Redundancy

- Passive redundancy uses additional capacity to reduce the impact of component failures
  - Active/passive failover
  - Hot spares
  - Snapshots
- Active redundancy eliminates performance problems by having simultaneous capacity in use
  - Active/active failover
  - Hot, mirrored, or parallel sites

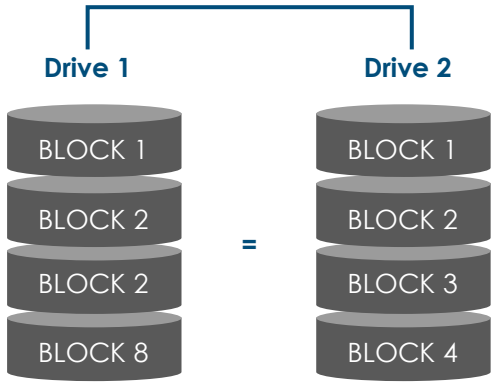
Redundant Array of Independent Disks (RAID 0)



- Data is split up into blocks
- Blocks are written across all array drives
- Uses at least two disks at a time
- Offers fast read and write speeds
- Not redundant = no fault tolerance

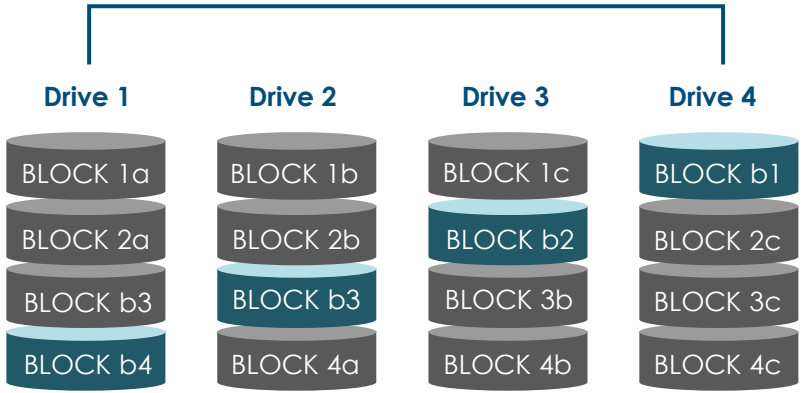
RAID Level 1 (Mirroring)

- Configuration of at least two drives that contain the exact same data
- If one drive fails, the others will still function
- RAID 1 offers high read performance, as data can be read off any of the drives in the array
- Since data needs to be written to all the drives in the array, the write speed is slower than a RAID 0 array



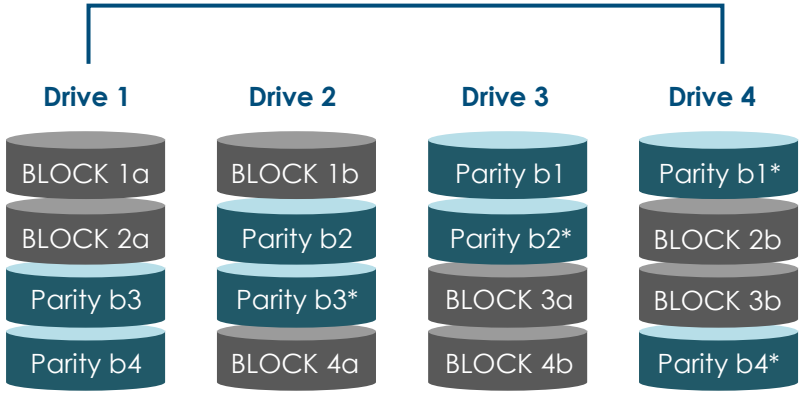
RAID Level 5

- RAID 5 requires at least three drives
- Data is striped across multiple drives like RAID 0, but also has "parity" data distributed across the drives
- In the event of a drive failure, data is restored by using the parity information stored across the other drives
- Read times are very fast but the write speed is slower due to the parity that must be calculated
- The most popular RAID 5 configurations use four drives, which lowers the lost storage space to 25 percent (it can work with up to 16 drives)



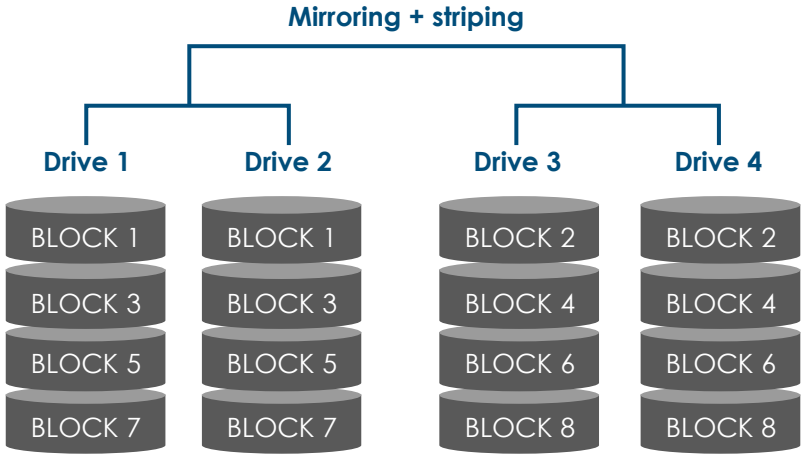
RAID Level 6

- RAID 6 is like RAID 5, but the parity data is written to two drives, so it requires at least four drives
- This solution can survive two drives failing simultaneously
- Read speeds are as fast as RAID 5, but the write speeds are slower than RAID 5 due to the additional parity data that must be calculated



RAID Level 10

- Consists of a minimum of 4 drives and combines the advantages of RAID 0 and RAID 1 into one single system
- Offers security by mirroring all data on secondary drives while using striping across each set of drives to speed up data transfers – the speed of RAID 0 with the redundancy of RAID 1
- Can lose any single drive, and feasibly even a 2nd drive without losing any data
- Compared to large RAID 5 or RAID 6 arrays, RAID 10 is an expensive way to have redundancy for fast databases, file servers, application servers

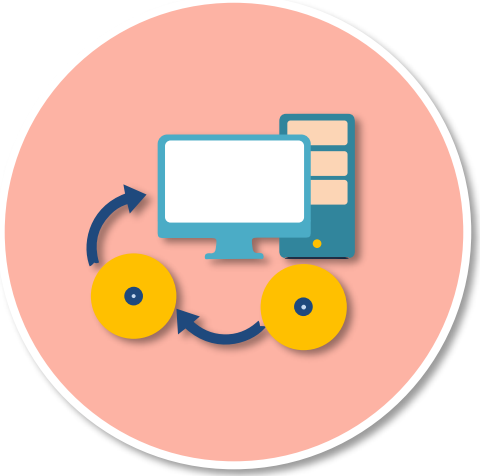


RAID Comparisons

Features	RAID 0	RAID 1	RAID 5	RAID 6	RAID 10
Minimum number of drives	2	2	3	4	4
Fault tolerance	None	Single-drive failure	Single-drive failure	Two-drive failure	Up to one disk failure in each sub-array
Read performance	High	Medium	Low	Low	High
Write performance	High	Medium	Low	Low	Medium
Capacity utilization	100%	50%	67-94%	50-88%	50%

Full Backups

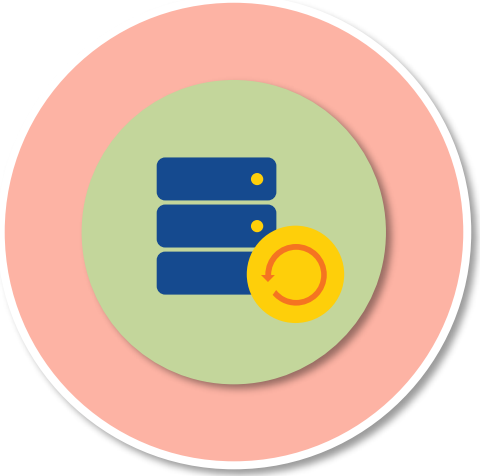
The main ransomware countermeasure



- Backs up everything regardless of archive bit being set or not
- Clears the archive bit once the backup completes
- It takes the longest to back up
  - Depends on how much must be backed up
- It is the quickest to restore
  - Only the most recent full backup is required

Incremental Backup

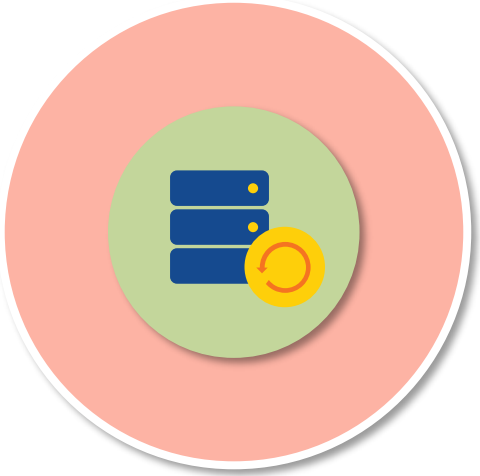
Clears the archive bit once the backup completes



- Backs up any new file or any file that has changed since
  - the last full backup, or
  - the last incremental backup
- Subsequent backups only store changes that were made since the previous backup
- The process of restoring lost data from an incremental backup is longer, but the backup process is much quicker
- **Should not be performed manually if possible**

Differential Backup

DOES NOT clear the archive bit when the backup completes



- Backs up any file that has the archive bit set
- Any new file or any file that has changed since the last full backup
- Slow to back up
- Quick to restore
- The last full backup and the most recent differential backup are needed for restoration
- **Not recommended to perform manually**

Snapshots

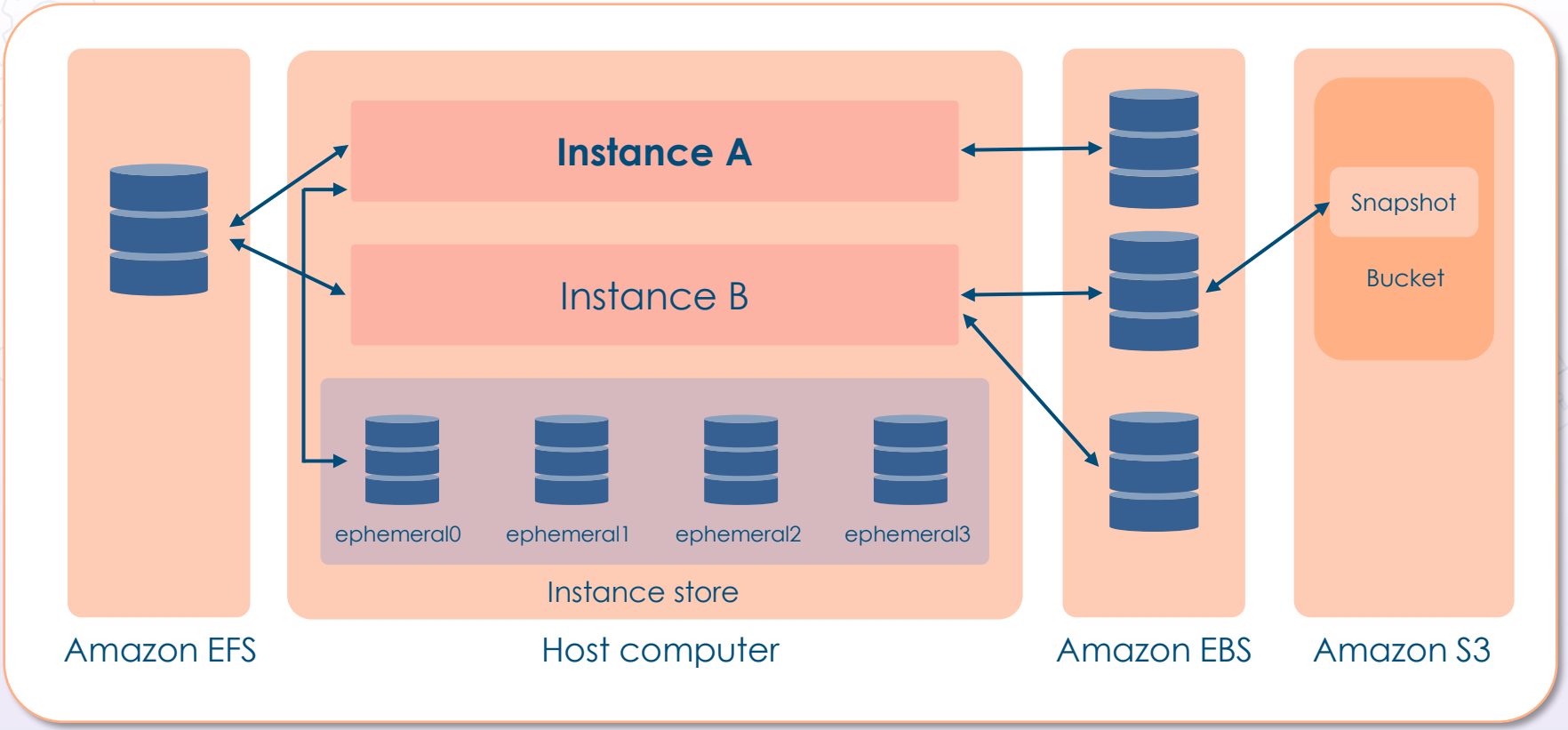
- Easier and faster backups and restores
- Immediate point-in-time virtual copy of source
- Should be replicated to another media or cloud storage to be considered a backup
- Time to back up does not increase with amount of data
- Improved RTO and RPO
- Restores are faster
- Less data is lost with an outage

Storage Media Comparisons

Medium	Date of invention	Lifespan	Capacity
HD	1956	5 - 10 years	From GB to TB
Floppy disk	1971	3 - 5 years	Hundreds of KB to a few MB
CD/CD-ROM	1979	25 - 50 years	80 minutes or 700 MB
MD (Mini Disc)	1991	25 - 50 years	60 minutes or 340 MB
DVD	1994/1995	25 - 50 years	4.7 GB
SD card	1994	10 years or more	A few MB to tens of GB
USB flash drive	2000	10 years or more	A few MB to tens of GB
SSD	1970 - 1990	10 years or more	From GB to TB

Barbosa, Daniel Cunha. “Types of Backup and Five Backup Mistakes to Avoid.” WeLiveSecurity. WeLiveSecurity, May 10, 2019. https://www.welivesecurity.com/2019/05/10/types-backup-mistakes-avoid/.

Amazon Block Storage (HDD and SDD)



AWS S3 Storage Plans (Tiers)

Standard	I-T	S-I A or 1 Z-I A	Glacier or Deep Archive
Eleven 9's of durability	Three 9's of availability	Infrequent access but rapid access when needed	Eleven 9's of durability
Four 9's of availability	Eleven 9's of durability	Lower per GB storage price and retrieval fee	Data archiving with flexible access options
Low-cost throughput	Cheaper than Standard S3	Lower throughput	Can store data for as little as $0.004 per gigabyte per month

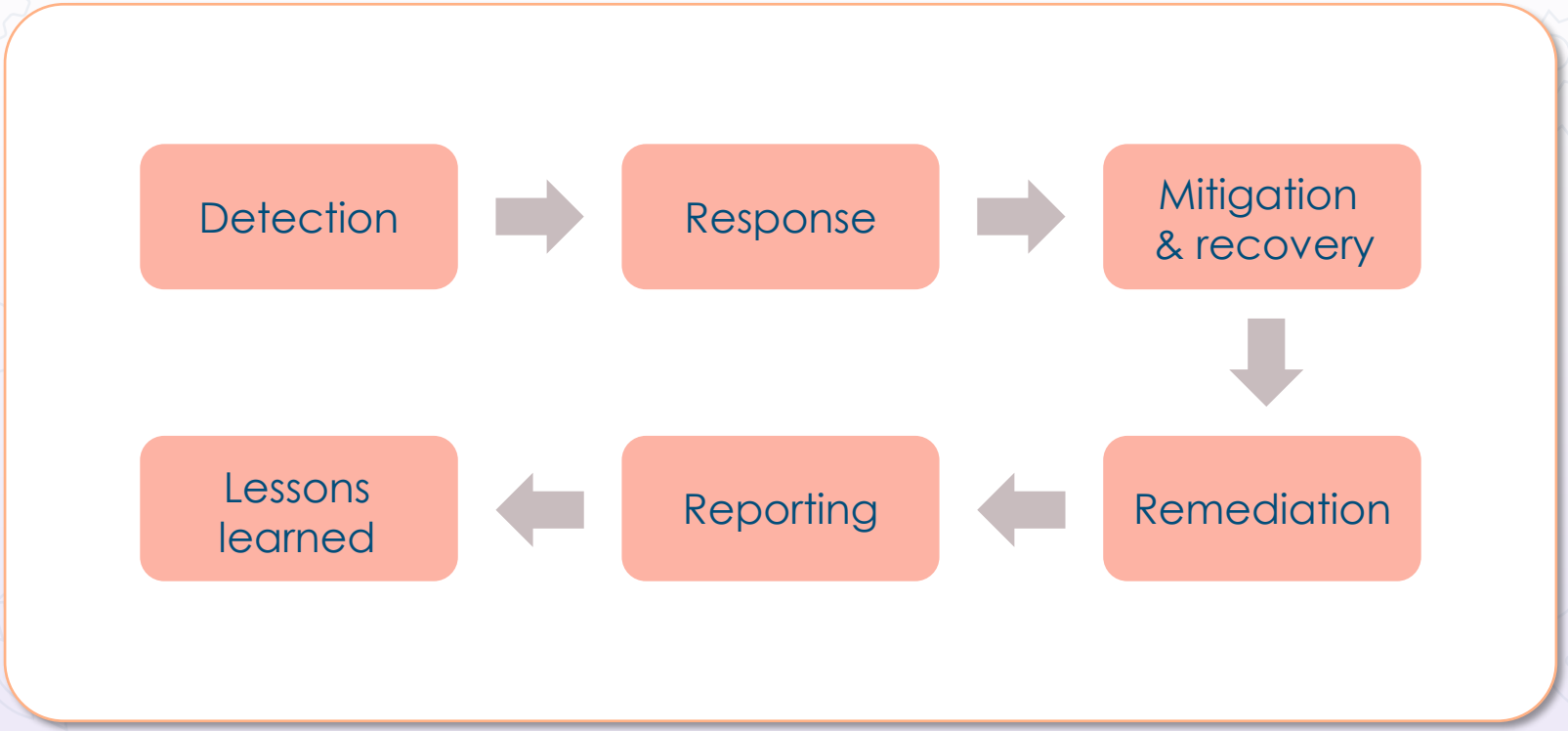
Disaster Recovery Planning (DRP)

- Ensuring that you can help the organization recover **to an acceptable level** from any type of catastrophic event
- A cataclysmic event can be a single drive ransomware attack to an entire facility or campus being put out of action
- The disaster recovery plan (DRP) should contain detailed steps for recovering from any kind of data loss or physical disaster

Disaster Recovery Planning

- Step-by-step instructions on how to recover each aspect of critical systems, applications, and data
- Backup and restore plans with order of restoration
- Contact information for key stakeholders, partners, and vendors
- Contact info for law enforcement, legal, insurance, media outlets
- Order of succession and command
- Location of hot spares, software and CD keys, security access keys and failsafe passwords, and other valuables
- Site locations and descriptions (cold, warm, hot, cloud)

DRP Lifecycle



Disaster Recovery Site Strategies

Recovery Strategy	Recovery Time	Advantages	Disadvantages	Comments
Commercial Hot site	24 to 48 Hours	• Best recovery time • Easiest to implement as equipment, application software, data, and OS are in place • Easy to test at any point in time • The best solution that is available to support on-going operations	• Most expensive options duplicate equipment and software plus on-going version control issues • Ongoing communication costs to duplicate data very high • Term of the agreement can limit the duration of use • If you are not the "most important customer" you could be bumped	Often the most cost-effective strategy for data center recovery strategies. Clear contract terms need to be defined which meets the enterprise service objectives. Consideration should be made for disasters that impact entire regions such as hurricanes and earthquakes.
Internal Hot	1 to 12 Hours	• Best recovery time • Easiest to implement as equipment, application software, data, and OS are in place • Easy to test at any point in time • The best solution that is available to support on-going operations	• Most expensive options duplicate equipment and software plus on-going version control issues • Ongoing communication costs to duplicate data very high	If costs can be shared among multiple facilities within the enterprise, internal provisioning can cost competitive with commercial alternatives. If no appropriate secondary space is available "co-location" facilities providers offer managed raised-floor space at very attractive rates as an alternative to building out secondary sites.
Warm Site	24 to 48 Hours	• Moderately priced • Basic infrastructure is in place to support recovery operations • Ability to pre-stage delivery and implementing of the necessary hardware, application software, OS software, data, and communications	• Not easy to test • Recovery time is longer than with hot site and is controlled by the time to locate and restore application • Facility equipment may not be exactly what is required – Once the recovery begins delays may occur because of equipment, software, or staffing shortfalls	If costs can be shared among multiple facilities within the enterprise, internal provisioning can cost competitive with commercial alternatives. If no appropriate secondary space is available "co-location" facilities providers offer managed raised-floor space at very attractive rates as an alternative to building out secondary sites.

Disaster Recovery Site Strategies

Recovery Strategy	Recovery Time	Advantages	Disadvantages	Comments
Mobile Site	24 to 48 Hours	• Moderately priced • Typically, can be in place for 36 to 72 hours • Can be placed in the "parking lot" adjacent to you impacted facility	• Recovery time typically is at least 2 to 5 days longer than a hot site. • Access to your impacted facility may be hindered because of the event • A trailer may not be configured exactly as you need it	This approach avoids employee travel issues but has limitations on equipment availability and outbound bandwidth if small aperture satellite terminal (VSAT) links must be used for communication. If the disaster profile includes events such as hurricanes, floods or toxic spills, these solutions may not be appropriate.
Cold Site	72 plus Hours	• Lowest cost solution • Basic infrastructure power, air, and communication are in place • Can rent the facility for a longer term at lower cost	• Longest recovery time • All equipment must be ordered, delivered, installed and made operational • Worst solution for supporting on-going operations	"Environmentally appropriate" space can be either provisioned internally or contracted form a commercial facilities service provider. Cold-site strategies are usually based on "quick-ship" delivery agreements to allow server, storage, and communications hardware and network service providers to quickly build out the data center and/or client workspace infrastructure.
Reciprocal Agreement	12 to 48 Hours	• Least costly solution • Better than no strategy	• Seldom works • Typically, in the same geographic area and a wide range disaster like an earthquake renders it of no use • No easy way to test	This is typically a formal agreement between two trusted, non-competing partners in different industries in which each provides secure sites for the other. This option is the least favorable and has the greatest risk associated with it.
Cloud	0 to 24 Hours	• Data and applications available immediately • Location independent • Easy to test	• Security • May no allow enough time for a daily cycle processing window	Data should be in place so activation would only be limited by connectivity and network addressing (DNS propagation)

Personnel Safety and Security Concerns

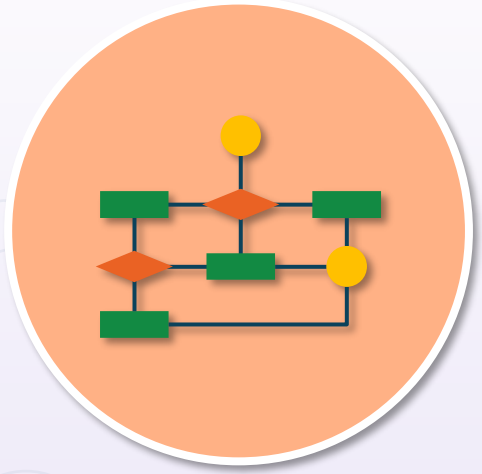
- Most large organizations have a separate department or third-party vendor that handles all employee and contractor travel arrangements
- Company may have fleet of autos or trucks for corporate use and may be assigned to employees on permanent or as-needed basis
- Disaster recovery must be addressed in security training and awareness programs
- May need HR to offer emergency management, counseling, and personal duress assistance

Test Disaster Recovery Plans

Read-through testing

- Read-through (plan review) is where the business continuity plan owner and business continuity team discuss the business continuity plan
- Look for missing elements and inconsistencies within the plan or with the organization
- A type of checklist test useful to train new members of a team, including the business function owner

Test Disaster Recovery Plans



Tabletop testing

- Participants gather in a room to execute documented plan activities in a stress-free environment
- Can use blueprints, topological diagrams, or computer models to effectively demonstrate whether team members know their duties in an emergency and if they need training
- Documentation errors, missing information, and inconsistencies across business continuity plans can be identified

Test Disaster Recovery Plans

Walkthrough testing

- Planned rehearsal of a possible incident designed to evaluate an organization's capability to manage that incident
- To provide an opportunity to improve the organization's future responses and enhance the relevant competences of those involved
- Often done on a limited basis or by scheduling each department or building separately for fire and active shooter drills

Test Disaster Recovery Plans

Simulation testing

- To determine if business continuity management procedures and resources work in a realistic situation, a simulation exercise is desirable
- May be the most elaborate test most entities ever conduct
- Uses established business continuity resources, such as the recovery site, backup equipment, services from recovery vendors, and transportation
- It can require sending teams to alternate sites to restart technology as well as business functions

Test Disaster Recovery Plans

Parallel testing

- A parallel test involves bringing the recovery site to a state of operational readiness, but maintaining operations at the primary site
- Staff are relocated, backup tapes are transferred, and operational readiness established in accordance with the disaster recovery plan while operations at the primary site continue normally
- May be the most comprehensive test most entities ever conduct

Test Disaster Recovery Plans

Full interruption testing

- Operations are completely shut down at the primary site to fully emulate the disaster
- Enterprise transfers to the recovery site in accordance with the disaster recovery plan
- A very thorough test, which is also expensive (may be cost-prohibitive)
- Has the capacity to cause a major disruption of operations if the test fails

Lessons Learned

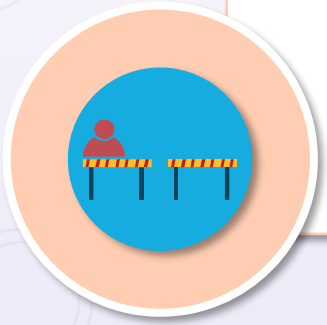
A section of After-Action Report

- Knowledge gained from the process of conducting the program, project, or task included in After-Action Report (AAR)
- Formal sessions usually held at the project close-out, near the completion of the initiative
- Recognized and documented at any point during the life cycle to
  - share and use knowledge derived from an experience
  - endorse the recurrence of positive outcomes
  - prevent the recurrence of negative outcomes

Gap Analysis

- A gap is the difference between the implemented existing controls and the predetermined control objectives
- Gap analysis is the outcome of corporate security strategy and governance
- Foundation for the fundamental information security initiative action plans and programs
- Current countermeasures should be established according to the organization's risk appetite for each asset class

Gap Analysis



- Conducted in a reusable and repeatable manner to assess and report on efficacy of executed controls
- Focus will be on established metrics such as key performance indicators (KPIs) and key goal indicators (KGIs)
- Also useful for security assessment and auditing

Threat Modeling

Prototyping

- Involves creating an abstraction of a system to identify risk and probable threats (private cloud/sandboxing) starting with all entry points to a system, service, or application
- With the widespread adoption of threat intelligence technologies, most enterprises are trying to adopt a threat-focused approach to risk management
- Provides visibility, increased security awareness and prioritization, and understanding of posture
- In addition to being a requirement for DoD acquisition, cyber threat modeling is very important to federal programs, including DHS and NASA

OCTAVE (Practice-based)

- Operationally Critical Threat, Asset, and Vulnerability Evaluation methodology is one of the first created specifically for cybersecurity threat modeling
- Focused on assessing organizational (non-technical) risks that may result from breached data assets, which are identified as datasets that have attributes based on the type of data stored
- Intent is to remove mistakes regarding the scope of a threat model and reduce excessive documentation for assets that are either poorly defined or are outside the purview of the project
- Provides a vigorous, asset-centric view, and organizational risk awareness, but the documentation can get overwhelming as it lacks scalability
- This highly-customizable method is most useful when creating a risk-aware corporate culture

Trike (Acceptable Risk-focused)

- Unique, open-source modeling process focused on sustaining the security auditing process from a cyber risk management perspective
- Risk-based approach with a distinctive implementation and risk modeling process
- Based on the "requirements model" that helps ensure the assigned level of risk for each asset is "satisfactory" to the various stakeholders
- Security engineers will create a data flow diagram (DFD) containing data stores, processes, data flows, and interactors
- DFDs and trust boundaries illustrate data flow in an implementation model and the actions users can perform within a system state

PASTA (Attacker-focused)

- Process for Attack Simulation and Threat Analysis (PASTA) is a relatively new application threat modeling approach
- Offers a seven-step platform-independent process for risk analysis
- Goal is to align business objectives with technical requirements, while considering business impact analysis and compliance requirements
- Combines an attacker-centric perspective on potential threats with asset-centric risk and impact analysis
- Works best for organizations that need to align threat modeling with strategic objectives as it integrates business impact analysis as an integral part of the process and magnifies cybersecurity responsibilities beyond the IT department

STRIDE (Developer-focused)

- Spoofing Tampering Repudiation Information Message Disclosure Denial of Service and Elevation of Privilege
- Microsoft threat modeling methodology that aligns with their Trustworthy Computing directive of January 2002
- Focus is to help ensure that Microsoft's Windows software developers think about security during the design phase
- Goal is to get an application to meet the security properties of CIA along with authentication, authorization, and non-repudiation
- Once the security SME builds the DFD-based threat model, system engineers or other experts check the application against the STRIDE threat model classification scheme

STRIDE

Threat	Definition	Property	Example
Spoofing	Pretending to be someone else	Authentication	Hack victim's email and to send messages as the victim
Tampering	Changing data or code	Integrity	Software executive file is tampered with by hackers
Repudiation	Claiming not to do a particular action	Non-repudiation	"I have not sent an email to users"
Information Disclosure	Leaking sensitive information	Confidentially	Making credit card information available on the internet
Denial of Service	Non-availability of service	Availability	Web application not responding to user requests
Elevation of privilege	Ability to perform unauthorized action	Authorization	Normal user can delete admin account

Admin. "STRIDE: Acronym of Threat Modeling System." All About Testing, February 21, 2019. https://allabouttesting.org/stride-acronym-of-threat-modeling-system/.

VAST (Enterprise-focused)

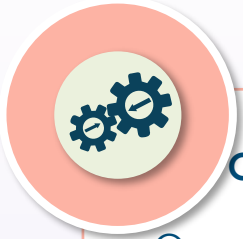
- Visual, Agile, and Simple Threat modeling was developed after observing perceived limitations and implementation challenges intrinsic to the other threat modeling methodologies
- Founding principle is that, in order to be effective, threat modeling must:
  - Scale with the infrastructure and entire DevOps portfolio
  - Integrate effortlessly into an Agile environment
  - Provide actionable, accurate, and reliable results for developers, programmers, security teams, and senior executives
- Fundamental difference is its practical approach in recognizing that security issues among development teams are often divergent from those of an infrastructure team

VAST Models

Application Threat Models

Application threat models for development teams are generated with process flow diagrams (PFD)

PDFs map the structures and communications of an application, much like the way developers and architects consider applications during the SDLC lifecycle



Operational Threat Models

Operational threat models are meant for the infrastructure teams

Though more like traditional DFDs than application threat models, the data flow information is presented from an attacker (not a data packet) perspective

By relying on PFDs instead of DFDs, VAST models do not require extensive systems expertise

Comparing Threat Modeling Methods

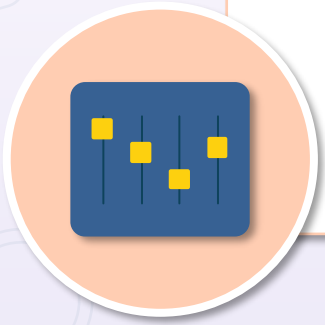
	OCTAVE	Trike	P.A.S.T.A	Microsoft	VAST
Implement application security at design time	✓	✓	✓	✓	✓
Identify relevant mitigating controls	✓	✓	✓	✓	✓
Directly contributes to risk management	✓	✓	✓		✓
Prioritize threat mitigation efforts	✓	✓	✓		✓
Encourage collaboration among all stakeholders	✓	✓			✓
Outputs for stakeholders across the organization	✓				✓
Consistent repeatability		✓			✓
Automation of threat modeling process		✓			✓
Integrates into an Agile DevOps Environment					✓
Ability to scale across thousands of threat models					✓

"Threat Modeling Methodologies." ThreatModeler Software, Inc. Accessed June 7, 2021. https://threatmodeler.com/threat-modeling-methodologies-/.

Security Control Assessment (SCA)

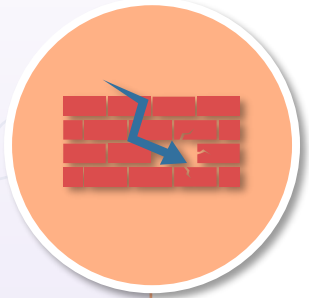
- Security Control Assessment (SCA) is a formal evaluation of a system against a pre-defined set of controls
- Performed with, or independently of, a full Security Test and Evaluation (ST&E), which is carried out as part of an official security authorization
- Often conducted as part of an official accreditation or certification process

Security Control Assessment (SCA)



- SCA and ST&E will appraise the operational plan (or planned implementation) of controls
- Results are a risk assessment report that represents a gap analysis, documenting the system, application, or data risk
- Tests conducted should include audits, security reviews, vulnerability scanning, and penetration testing

Penetration Testing Frameworks



SSAF - framework provided by Open Information Systems Security Group (OISSG), a not-for-profit organization based in London

OSSTMM - open-source security testing created by ISECOM (Institute for Security and Open Methodologies

OWASP - popular methodology used widely by security professionals, create by a non-profit organization focused on advancing software security

PTES - Penetration Testing Execution Standard (PTES) methodology was developed to cover the key parts of a penetration test

NIST - National Institute of Standards and Technology (NIST) provides a manual that is best suited to improve the overall cybersecurity of an organization

NIST Cybersecurity Framework

IDENTIFY	PROTECT	DETECT	RESPOND	RECOVER
• Asset management • Business environment • Governance • Risk assessment • Risk management strategy	• Awareness control • Awareness and training • Date security • Info protection and procedures • Maintenance • Protective technology	• Anomalies and events • Security continuous monitoring • Detection process • Communications	• Response planning • Communications • Analysis • Mitigation • Improvements	• Recovery planning • Improvements • Communications

SOC 2

- Developed by the American Institute of CPAs (AICPA)
- Defines criteria for managing customer data based on five trust service principles: security, availability, processing integrity, confidentiality, and privacy
- Unlike PCI DSS, which has very rigid requirements, SOC 2 reports are
  - Unique to each organization
  - Aligned with precise business practices, each with its own controls to comply with one or more of the principles
- There are two types of SOC reports:
  - Type I describes a vendor's systems and whether their design is appropriate to meet the applicable trust principles
  - Type II details the operational effectiveness of those systems

Center for Internet Security (CIS)

- CIS v7 lists 20 actionable cybersecurity requirements designed for improving the security standards of all organizations
- Most companies use the security requirements as best practices, since the CIS has a credible reputation for developing baseline security programs
- CIS v7 stands out from the rest as it empowers organizations to create budget-friendly cybersecurity programs and better prioritize their cybersecurity efforts

CISv7 Implementation Groups

1

Group 1 is for businesses and organizations with little or limited cybersecurity expertise and resources

2

Group 2 is for all organizations with moderate technical experience and resources in implementing the CIS controls

3

Group 3 is for organizations with immense cybersecurity expertise and resources

Cloud Security Alliance (CSA)

- World's leading organization dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment
- CSA Cloud Controls Matrix (CCM) is a cybersecurity control framework for cloud computing
  - Composed of 197 control objectives
  - Structured in 17 domains covering all key aspects of cloud technology
- Certified auditors for CSA STAR Certification

NIST 800-53

- NIST 800-53 publication enables federal agencies to realize effective cybersecurity practices
- Focuses on information security requirements designed to assist federal agencies in securing information and information systems
- Offers government agencies the requirements to comply with FISMA (Federal Information Security Management Act)
- NIST 800-53 contains more than 900 security requirements, making it among the most complicated frameworks for organizations to implement
- Some recommendations include controls for enhancing physical security, penetration testing, guidelines for implementing security assessments, and authorization policies or procedures

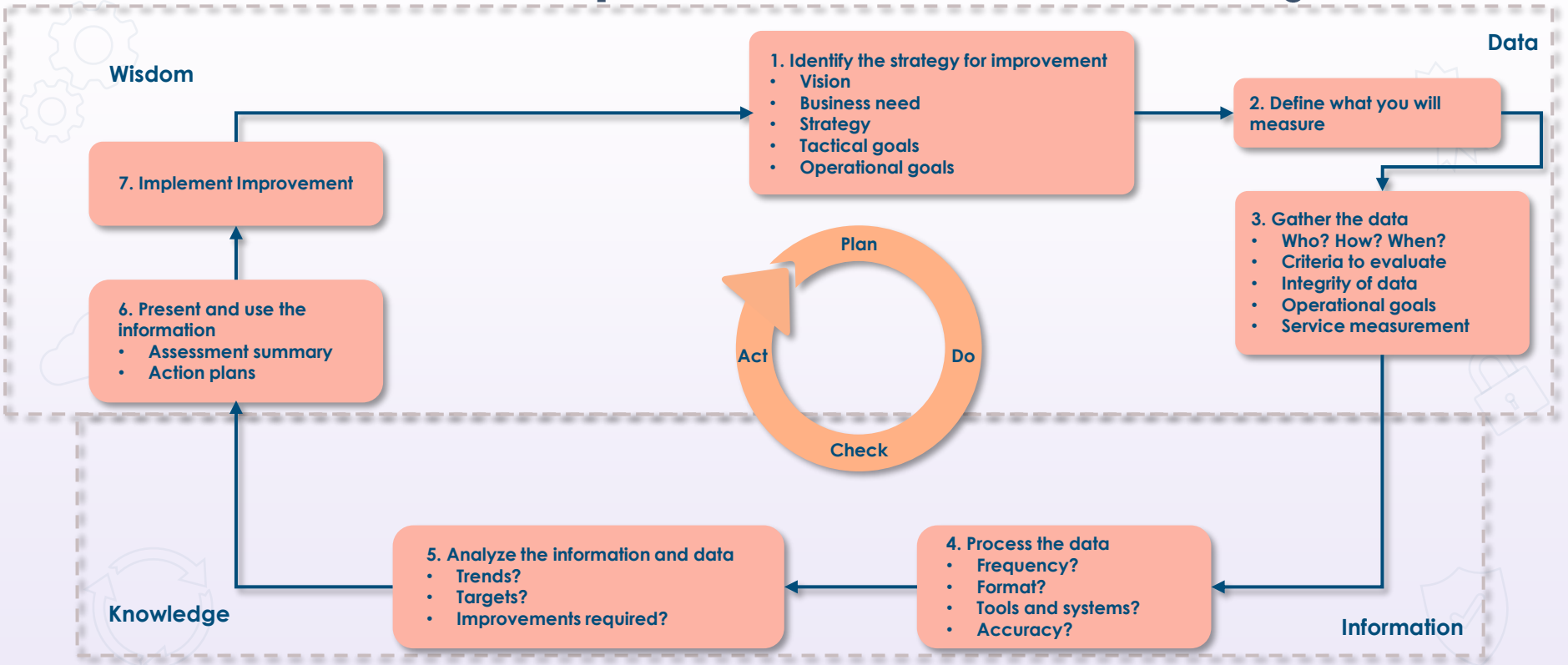
COBIT

- Control Objectives for Information and Related Technologies aligns a business's best features to its IT security, governance, and management
- ISACA developed and maintains this framework ,which is useful for companies trying to simultaneously improve production quality and observe heightened security practices
- Helps to meet all stakeholder cybersecurity expectations, end-to-end procedure controls for organizations, and the need to develop a single but integrated security framework

IASME

- Information Assurance for Small and Medium Enterprises Consortium (IASME) standards certification includes free cybersecurity insurance for businesses operating within the UK
- The IASME governance accreditation is similar to an ISO 27001 certification
  - Implementing and maintaining the standard comes with reduced costs, administrative overhead, and complexities over ISO 27001
- IASME governance refers to cybersecurity standards designed to enable small and medium-sized enterprises to realize adequate information assurance
  - Outlines a criterion in which a business can be certified as having implemented the relevant cybersecurity measures.
- The standard enables companies to demonstrate to their customers their readiness to protect business or personal data

Continual Improvement Models Overlay



Collecting Security Process Data

- Security analysis should involve powerful and automated security log analysis tools
- Since every single device or application on the network creates log files, administrators should start with log analyzers that collect data from a device's log files and translates it into a data format that is easy to read
- Output is often in a centralized graphical format
  - SolarWinds Security Event Manager tool automatically generates HIPAA, PCI-DSS, SOX, ISO, NCUA, FISMA, FERPA, GLBA, NERC CIP, GPG13, reports
  - Datadog Log Analysis and Troubleshooting
  - ManageEngine EventLog Analyzer
  - Splunk and Paessler PRTG Network Monitor

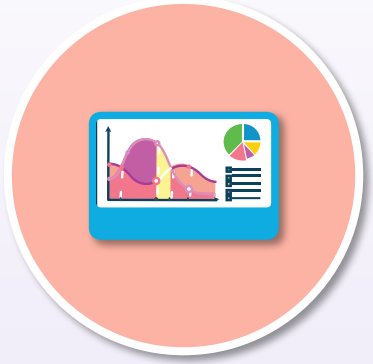
Automated Solutions

- **Amazon Inspector** - an automated security assessment service that helps improve the security and compliance of applications deployed on AWS
  - Automatically assesses applications for exposure, vulnerabilities, and deviations from best practices
- **Microsoft Azure Sentinel** - a scalable, cloud-native, security information event management (SIEM) and security orchestration automated response (SOAR) solution
  - Investigates threats with artificial intelligence, and hunts for suspicious activities at scale, tapping into years of cybersecurity work at Microsoft
- **Fortinet's Cyber Threat Assessment Program (CTAP)** – helps validate your network's current security accuracy, analyze application traffic, assess user productivity, and monitor network performance

Security Reporting Best Practices

- Align reporting to customer security guidance, policies, standards, and guidelines
- Reporting routine should correspond to the vulnerability scanning routine (daily, weekly)
- Maintain a consistent reporting structure over the lifecycle for improved results
- Purge outdated and obsolete security scan results data
- Engage report consumers frequently and assess possible improvements

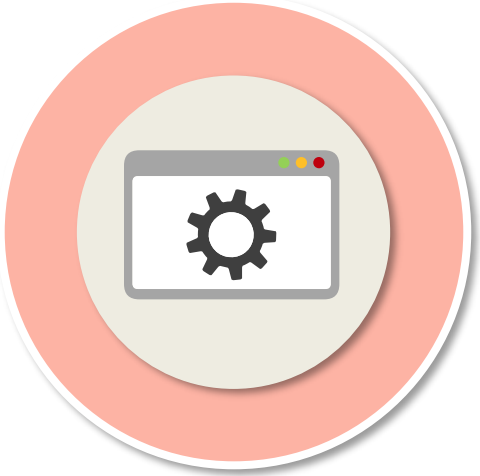
Security Reporting Best Practices



- Utilize engaging dashboards
- Generate reports with different audiences and technical knowledge in mind
- Provide glossaries and definitions – don't assume understanding
- Make reporting as interactive and dynamic as possible
- Avoid pie charts and histograms

Application Security Design

Secure by default



- This design consideration assumes the application is natively secure without any modifications or additional controls
- Example: a server application has certain possible unsecure functions, but they are disabled by default at deployment based on infrastructure-as-code

Application Security Design

Secure by design

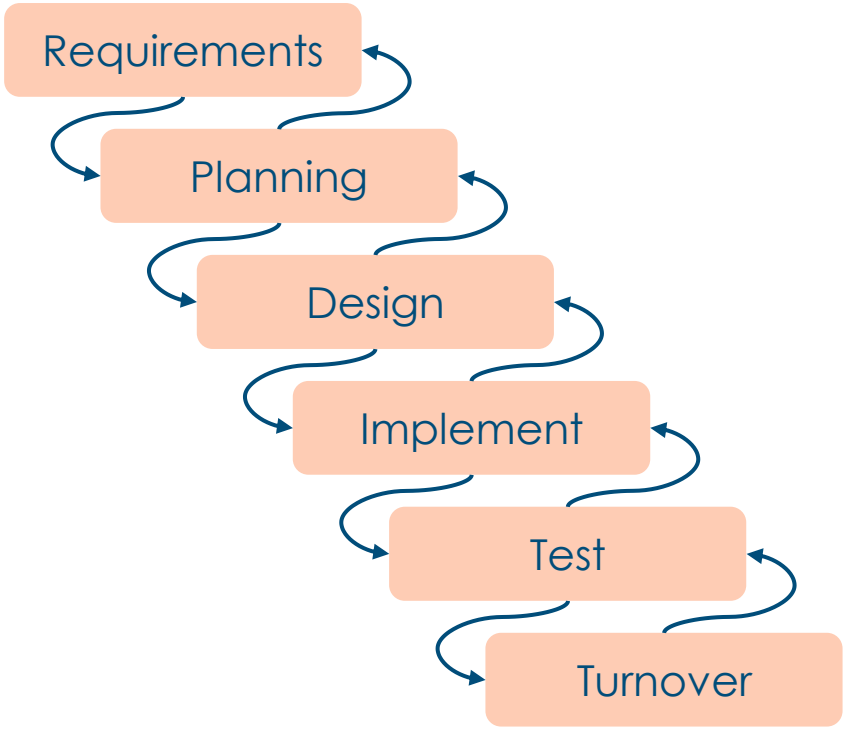
- The custom or outsourced program or application is developed with security integrated into the entire SDLC
- Attackers cannot simply overcome the security controls, even if they have white or gray box familiarity with the application design
- Security by design should not rely on security by obscurity
- Example: using cloud-native computing at AWS, GCP, or Azure build and run scalable applications in modern, dynamic environments, such as private and hybrid clouds for containers, microservices, serverless functions, and immutable infrastructure

Application Security Design

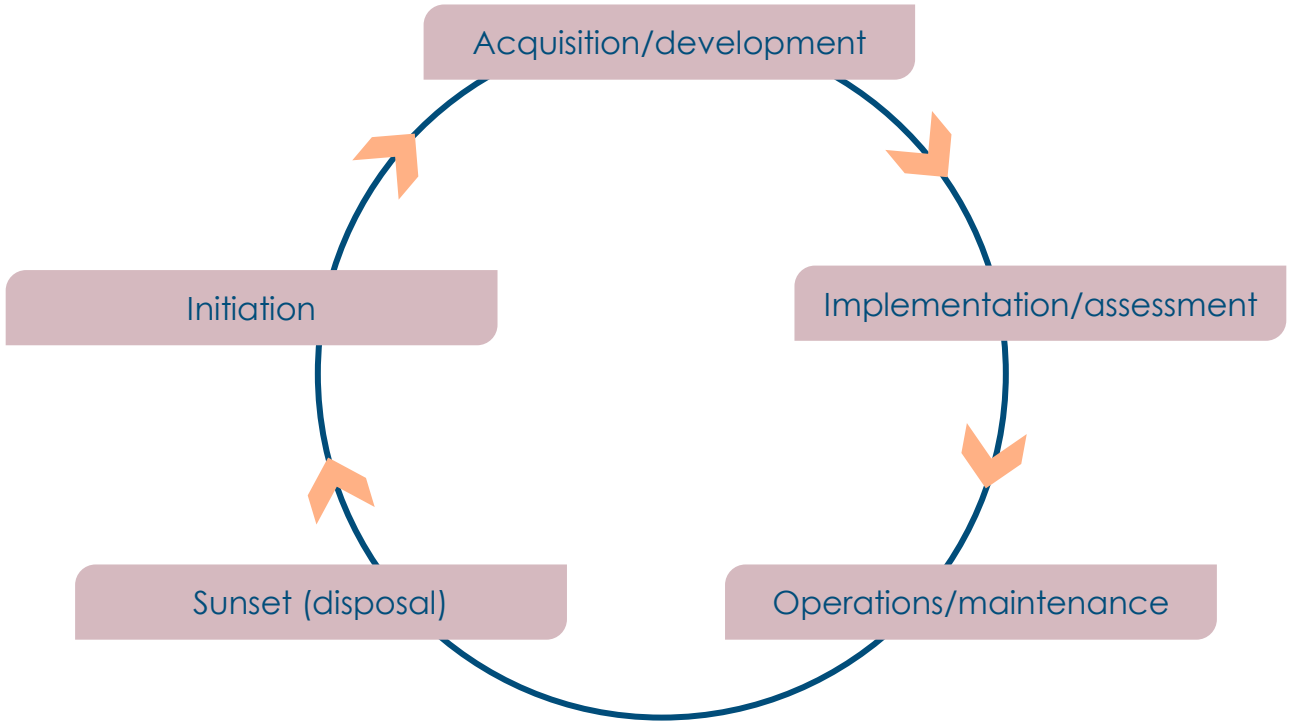
Secure by deployment

- The application was not developed with security integrated, but is deployed into an environment where security was considered in the network and system design
- Developers can take advantage of infrastructure-as-code to roll out well-tested application stacks
- Example: an application is deployed with air-gapped Docker container separated from any untrusted networks (a private cloud at a CSP or on-premises)
- Does not include security by obscurity

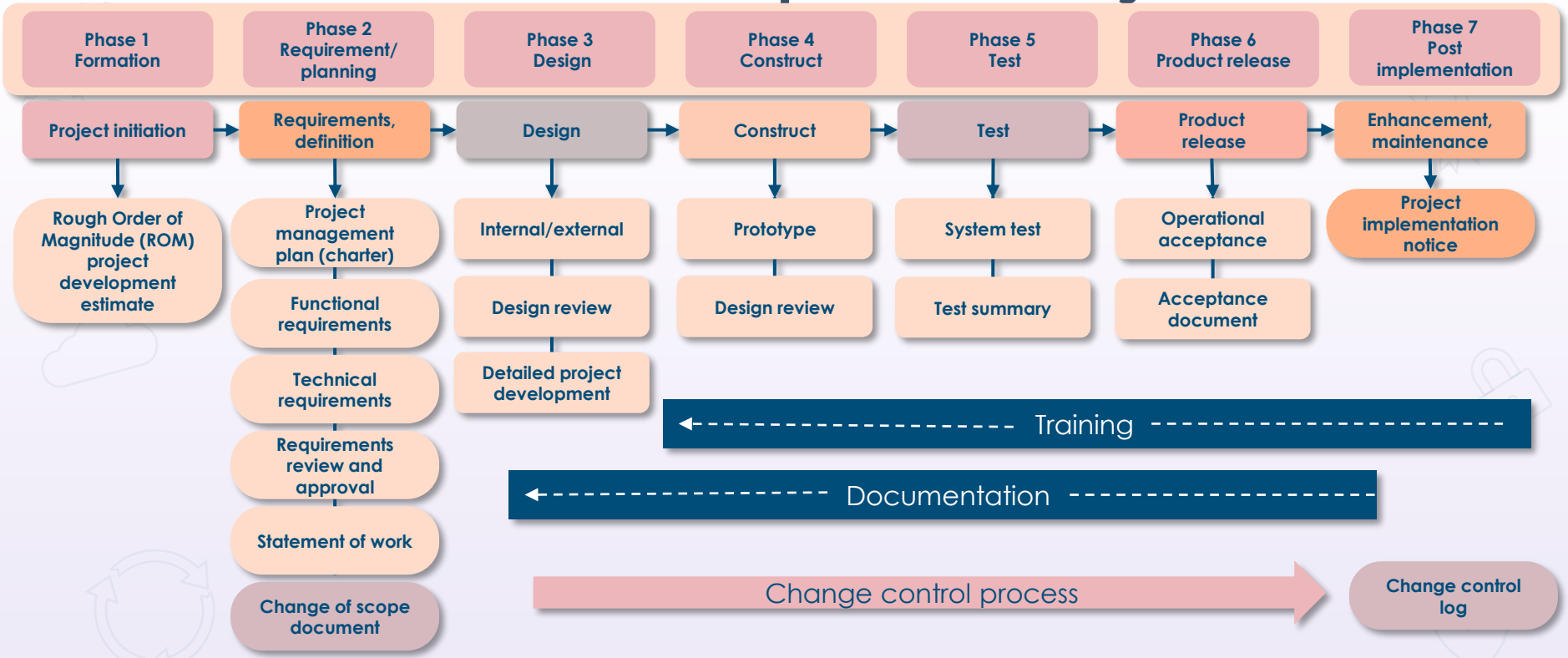
Waterfall Software Development



NIST Traditional SDLC

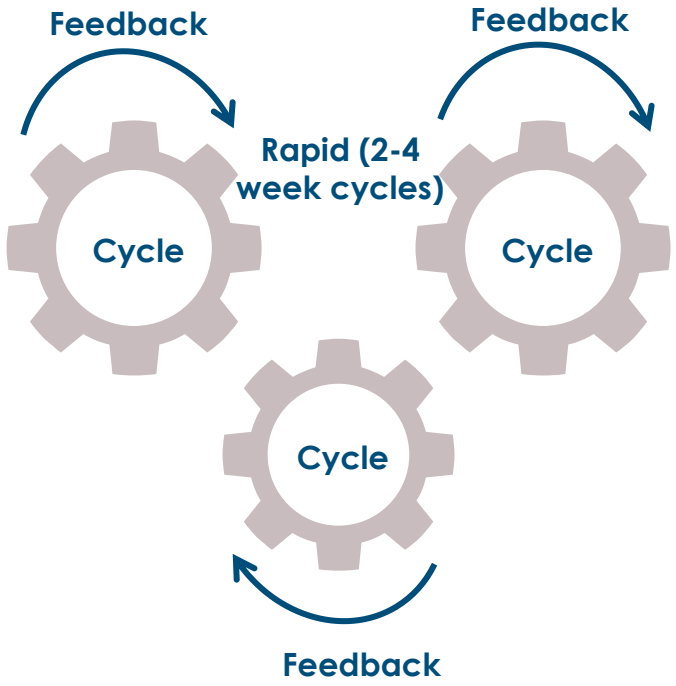


Software Development Life Cycle

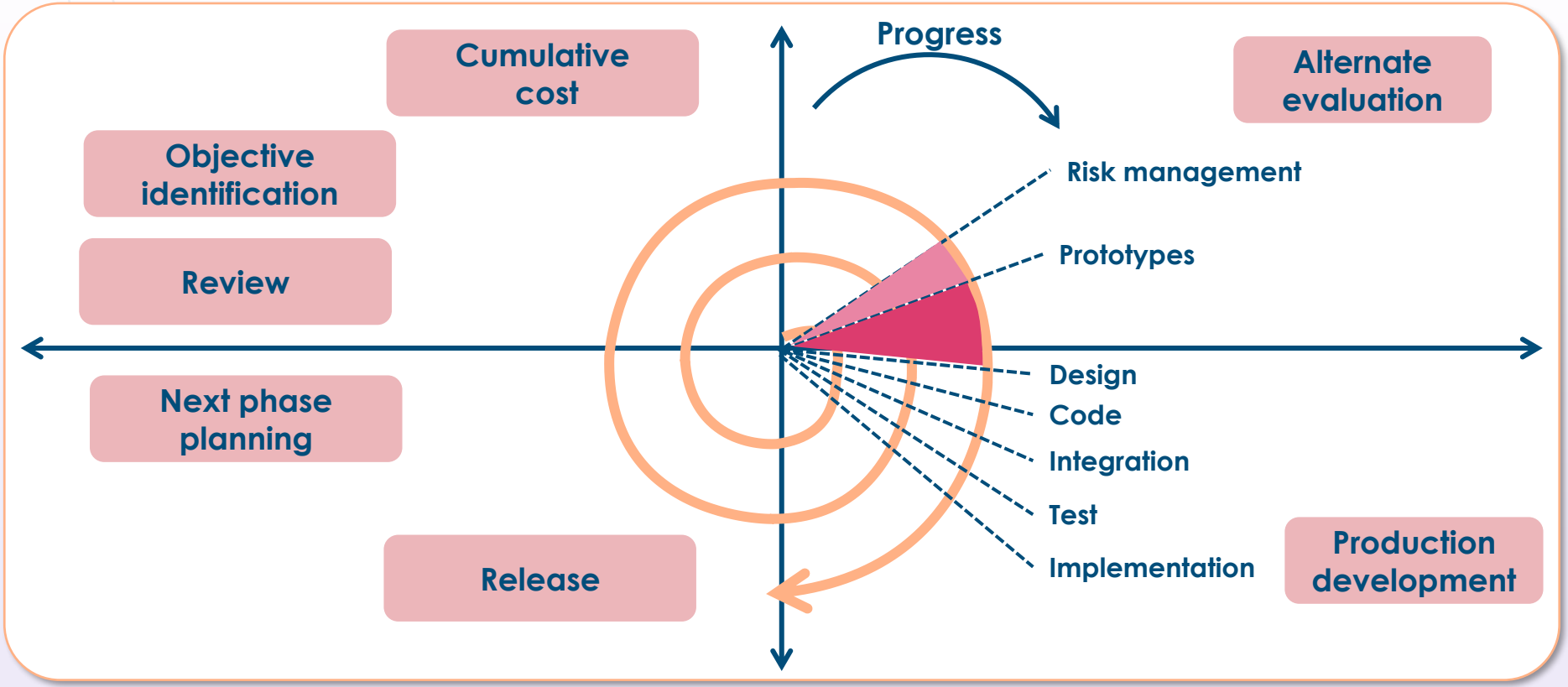


Agile Software Development

- For smaller projects, the concept of agile software development is becoming a viable alternative
- Evolutionary approach – measured in weeks – involving collaboration between cross-functional teams
- Very flexible, adaptable, not predictable, with testing done during development
- Estimates (i.e., budget, schedule, etc.) get more realistic as work progresses, because important issues are discovered earlier

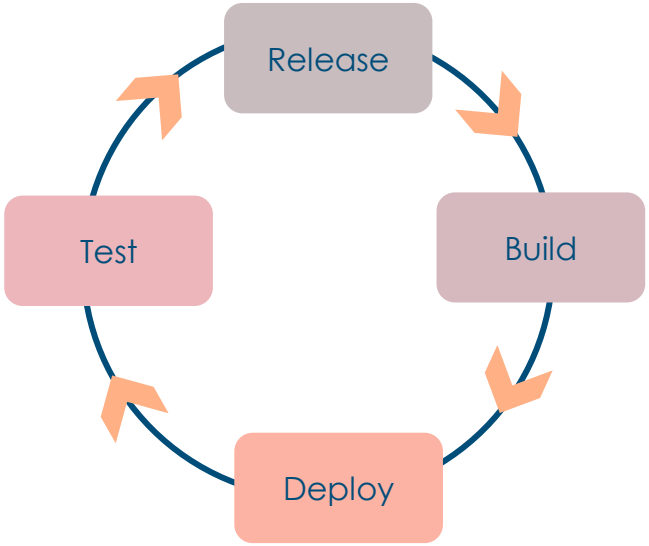


Spiral Software Development



Continuous Integration Development

- Continuous integration (CI) is a development technique that forces developers to integrate code into a shared repository several times a day
- Each check-in is then verified by an automated build, allowing teams to detect problems early
- The goal is to detect and locate bugs and security flaws quickly
- Developers on AWS commonly use continuous integration and continuous deployment (CI/CD)



DevSecOps

- DevOps is a methodology for building software quickly by linking development and operations
- It is a clipped compound of "software DEVelopment" and "information technology OPerationS," referring to a set of practices that accentuate the collaboration and communication of both software developers and IT professionals that automate the software delivery process
- DevSecOps involves considering application and infrastructure security from the start and automating some security gates to keep the DevOps workflow from slowing down
- Choosing the right tools to integrate security continuously is critical

Capability Maturity Models (CMM)

- The Capability Maturity Model (CMM) is a methodology used to advance and enhance an organization's software development process
- The model consists of a five-level evolutionary path of progressively prepared and systematically more mature processes
- It is like ISO 9001 standards that specify an effective quality system for manufacturing and service industries

Capability Maturity Models (CMM)

Initial (chaotic): chaotic, ad hoc, individual heroics

Level 1

Repeatable (implicit): process is not codified or defined and is still vulnerable to inconsistency

Level 2

Defined (early explicit): Process is defined and documented as a standard business process

Level 3

Managed (mature explicit): process is controlled and can be adjusted and adapted to particular projects without measurable losses of quality

Level 4

Optimized (purely explicit): process management includes deliberate process optimization and improvement

Level 5

CMM Level 1

- At the **initial** (chaotic) level 1, processes are disorganized, even chaotic
- Success most often depends on individual heroics and is not considered to be repeatable
- Processes are not sufficiently defined or documented for them to be replicated
- Decision-making is a free-for all and based on intuition and existing experience
- Decision-making authorities are poorly defined, with no KRI's, KPIs, CSFs, or real meaningful metrics
- Results are inconsistent and often unaligned with any executive leadership

Initial (chaotic):
chaotic, ad hoc, Individual heroics

Level 1

CMM Level 2

- At the **repeatable** (implicit) level 2, fundamental project management is established, and successes could be repeated, as the obligatory processes have been established, somewhat defined, and documented
- Decision-making is based on rote adherence to poorly-aligned standards and practices, and data provided to decision-makers is often superficial, not codified, and cannot hold up to scrutiny
- Roles and responsibilities are unclear, and it is common for people to make decisions outside their level of authority
- Risk is defined purely quantitatively and without much support of expert judgment or historical precedent
- Established KRIs and metrics, if any, are questionable as risk terminology, taxonomy, and policy is superficial or non-existent

Repeatable (implicit): process is not codified or defined and is still vulnerable to inconsistency

Level 2

CMM Level 3

- At the **defined** (early explicit) level 3, the enterprise has developed its own standard software process through greater attention to documentation, standardization, and integration
- Standardized terminology exists, and assessments are more up-to-date and better supported
- Visibility is improved as more robust and defensible analysis exists
- Well established standards, security teams, steering committees will exist
- Components like service desk and ITIL4 practices are put in use
- Risk registers, KPIs, and KRIs are defined
- Meaningful metrics, and calibrated and more precise semi-quant and quants are evident

Defined (early explicit): process is defined and documented as a standard business process

Level 3

CMM Level 4

- At the **managed** (mature explicit) level 4, the processes are controlled and can quickly be adjusted and adapted for development projects, security initiatives, or other endeavors without measurable loss of quality
- Quality is visible, and risk registers and assessments are up-to-date
- Data is actively used, and risk treatment/handling is adapted accurately based on more quantitative analysis
- Indicators and metrics are well-defined and tested
- Methods for assurance and certification are established if pertinent
- This is the highest level that most DevSecOps initiatives and organizations can hope to meet

Managed (mature explicit): process is controlled and can be adjusted and adapted to particular projects without measurable losses of quality

Level 4

CMM Level 5

- At the **optimized** (purely explicit) level 5, the process management has attained everything at level 4, including deliberate process optimization and continual improvement
- Level 5 is rarely achieved, but is still possible with proper leadership and resources
- For example, ITIL 4 mastery and maximum software development proficiency would be demonstrated in this organization

Optimized (purely explicit): process management includes deliberate process optimization and improvement

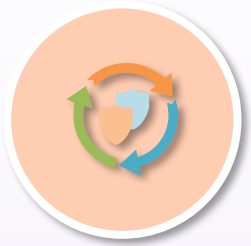
Level 5

Software Assurance Maturity Model (SAMM)

The Software Assurance Maturity Model (SAMM) is an open framework from OWASP to assist organizations in developing and deploying a secure software delivery strategy that is focused on the detailed risks facing the enterprise. The resources offered by SAMM will assist in:

- Appraising the organization's current software security initiatives
- Constructing a well-adjusted software security assurance program using established iterative processes
- Establishing tangible continual improvement methodologies to a software security assurance program
- Defining and gauging security-related tasks throughout the enterprise

SAMM Principles



An organization's activities gradually change over time

An effective software security initiative should be implemented in small repeatable iterations that result in tangible assurance wins, while incrementally pushing toward longer-term goals

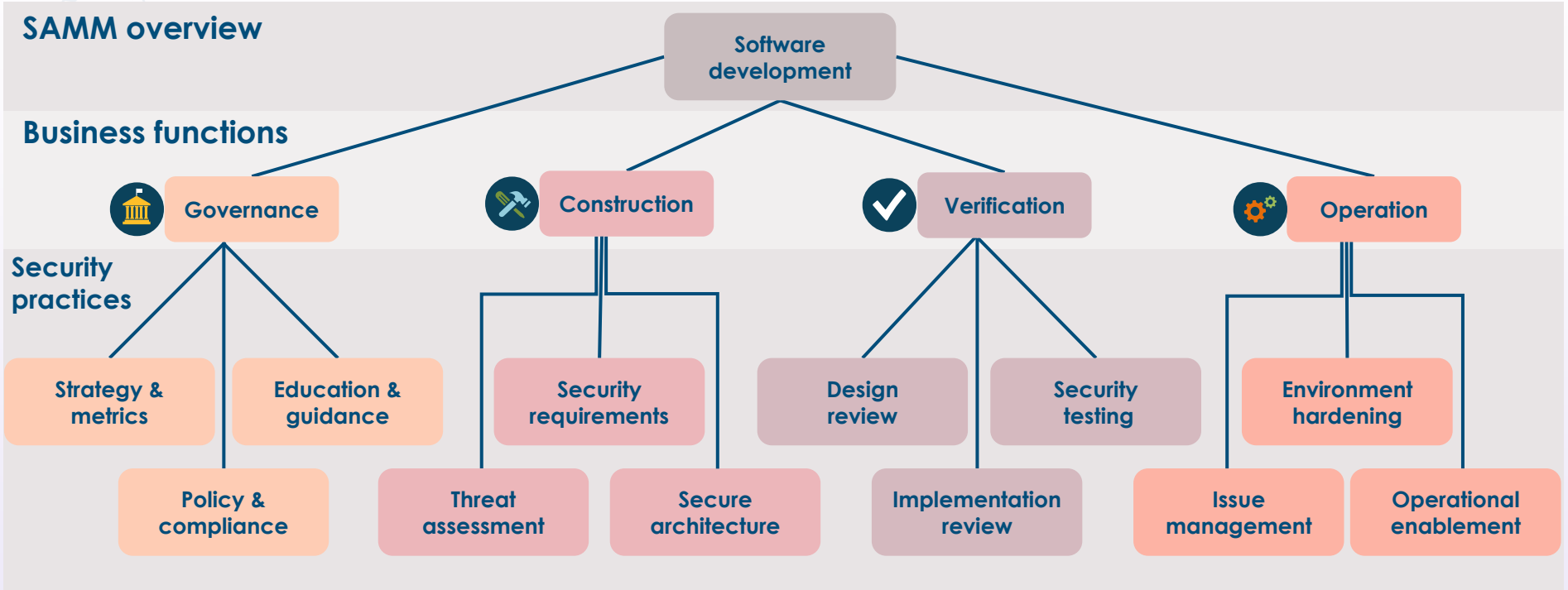
There is no "one-size-fits-all" solution that works across all organizations

A software security framework must be malleable and allow organizations to customize their solutions based on their risk treatment and the way they build or buy, and use software applications

Guidance related to security activities must be strict and comprehensive

All the steps used to build and assess an assurance program should be concise, well-defined, and with meaningful measurable metrics, providing roadmap templates for other types of like organizations

SAMM Overview



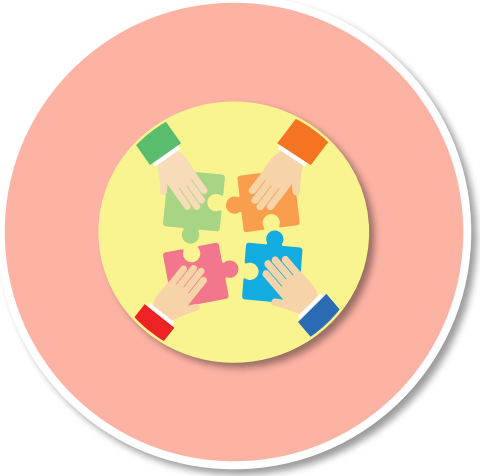
"Software Assurance Maturity Model." OWASP.org. Accessed June 8, 2021. https://owasp.org/www-pdf-archive/SAMM_Core_V1-5_FINAL.pdf.

Integrated Product Teams (IPT)

- An IPT is defined as a multidisciplinary team of people who are collectively responsible for delivering a well-defined product or service solution
- The Department of Defense (DoD) has adopted IPTs as their preferred approach for systems and software acquisition
- IPTs are maintained by different subject matter experts and can be leveraged throughout the entire development lifecycle
- These groups of key individuals represent varying ranges of expertise with the common goal of delivering the best product or service
- This cross-functional expert judgment also supports product acquisition activities and the development of system and software solutions

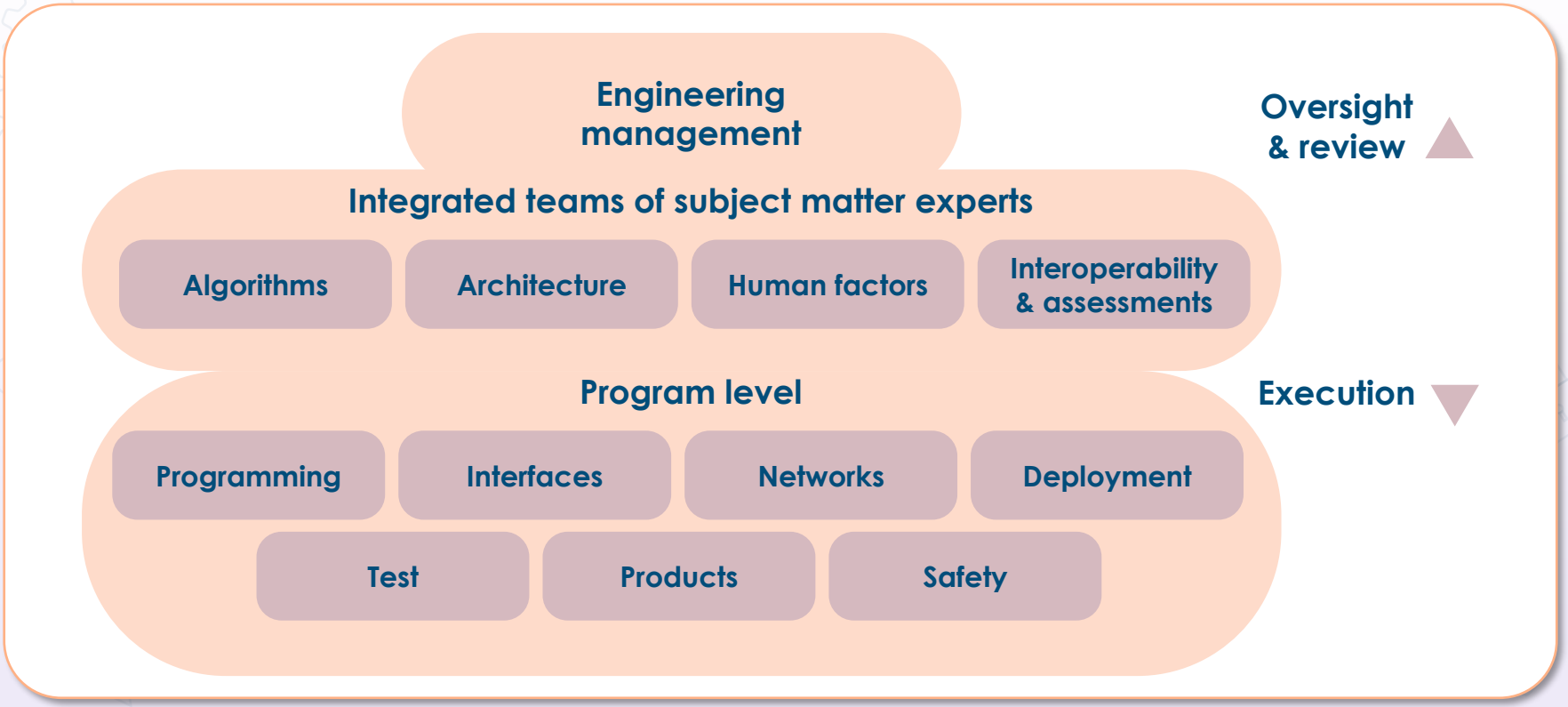
Goals of IPT

IEEE standard 1220-2005

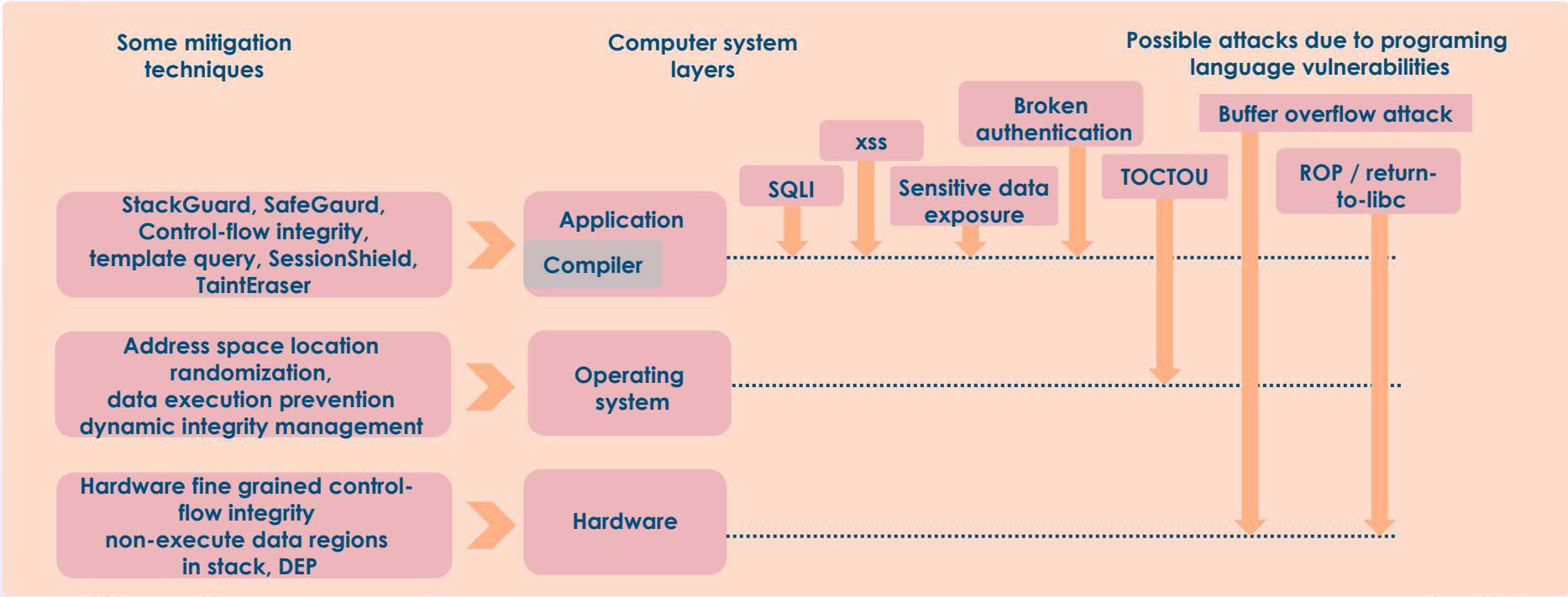


- The interdisciplinary tasks required throughout a system's life cycle to transform customer needs, requirements, and constraints into a system solution, are defined.
- In addition, the requirements for the systems engineering process and its application throughout the product life cycle are specified.
- The focus of this standard is on engineering activities necessary to guide product development while ensuring that the product is properly designed to make it affordable to produce, own, operate, maintain, and eventually to dispose of without undue risk to health or the environment.

Integrated Product Teams (IPT)



Identify Programming Security Controls



Khwaja, Amir A., Muniba Murtaza, and Hafiz F. Ahmed. "A Security Feature Framework for Programming Languages to Minimize Application Layer Vulnerabilities." Wiley Online Library. John Wiley & Sons, Ltd, November 7, 2019. https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.95.

Code Repository Security

- Your code is only as secure as the methods and systems used to generate it
- Some of the advantages that come with using a secure code repository are version control, peer review, and built-in auditing
- It is critical that the repository (such as GitHub or AWS) is an adequately secure central point of code storage and management
- Attackers can change a code base without your knowledge or permission due to loss/compromise of access credentials or breach of the core service
- If appropriate due diligence is applied to security measures, the benefits of using a code repository far outweigh the risks

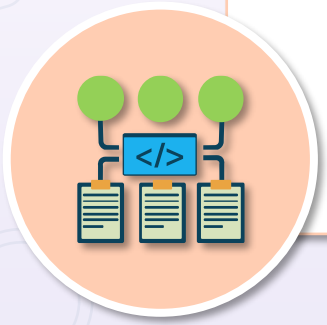
Code Repository Security

- Select a repository you trust completely
- Consider the exposure and customer base of your repository
- Protect access credentials
- Separate secret credentials from source code
- Repository access should be revoked quickly when not needed or if compromised

Code Repository Security

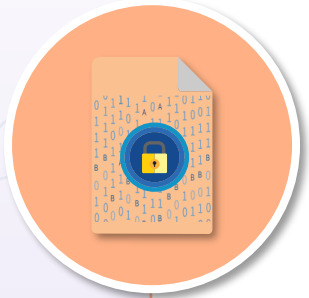
- Include open code in your risk model
- Review all code changes
- External code changes may be malicious
- Protect your identity if using a publicly accessible repository
- Ensure that your code is backed up

Software Configuration Management (SCM)



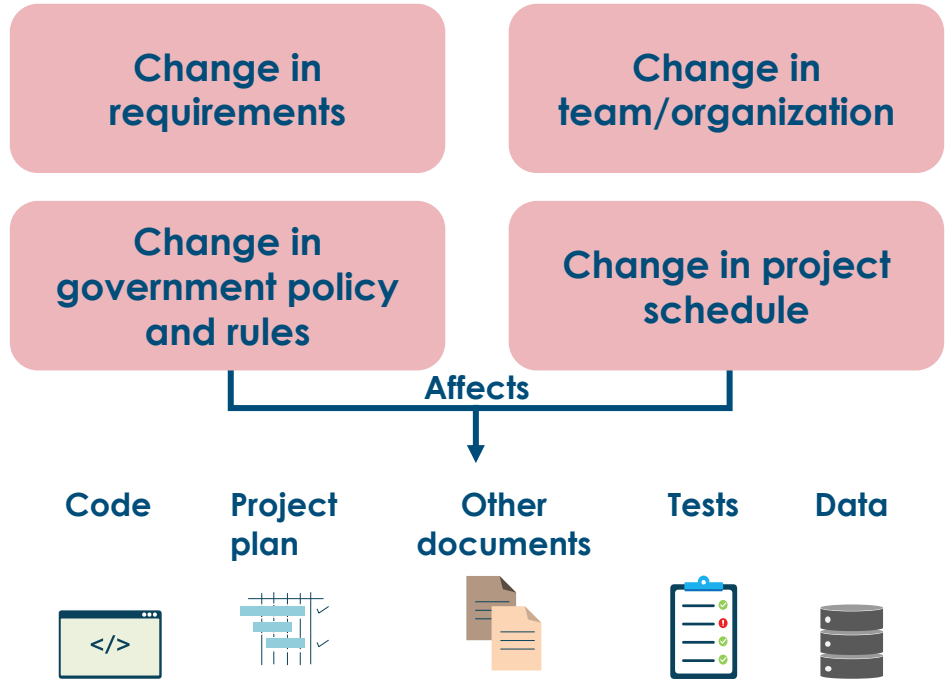
- Software configuration management (SCM) is a software engineering process to systematically manage, organize, and control the changes in the documents, codes, and other artifacts during the software development life cycle
- The primary goal is to enhance productivity and minimize errors
- SCM is part of the cross-disciplinary field of configuration management (integrated product teams – IPT) and can correctly determine the revision history

Reasons to Use SCM

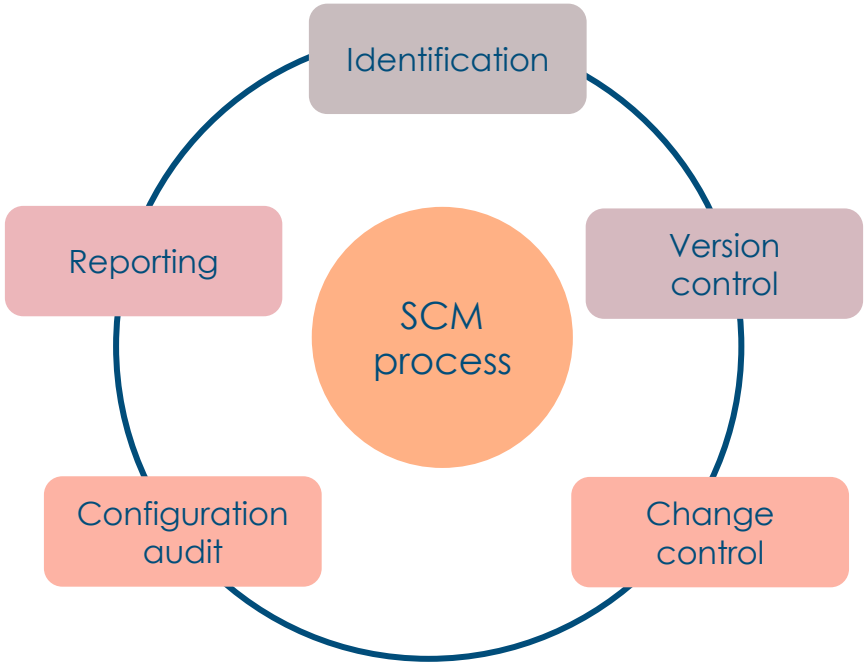


- There are several people working on applications that are continually updating (CI/CD or Spiral development)
- Multiple versions, branches, micro-services, and programmers are involved in a software project, and the team is geographically dispersed yet is working concurrently
- Changes in customer requirements, policy, budget, and schedules need to be accommodated
- Software must be able to run on different platforms and operating systems
- There is a critical need to develop coordination among cross-functional stakeholders
- Need to control the costs involved in making changes to an app

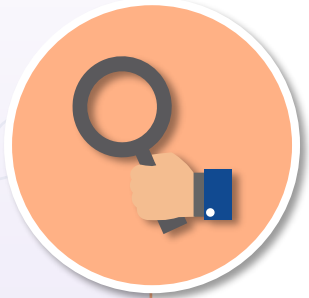
Software Configuration Management (SCM)



Software Configuration Management (SCM)

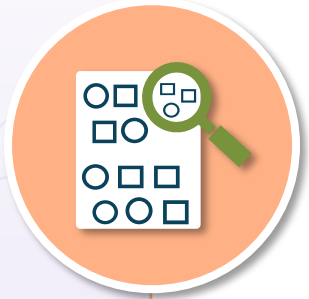


Static Application Security Testing (SAST)



- SAST tools are also known as code analyzers that conduct a direct white-box analysis of the application source code
- The analysis runs on a static view of code, in that the code is not running at the time of the assessment
- SAST security tools are mainstream and are widely adopted throughout the software industry
- They have broad programming language support and use concepts that are relatively easy to comprehend
- SAST code analyzers have no visibility of the execution flow, can be slow, inaccurate, and outdated, and often need additional customization and/or tuning

Dynamic Application Security Testing (DAST)



- DAST tools are most often web scanners like OWASP ZAP and Burp Suite (vulnerability scanners)
- When compared to SAST, they perform black-box analysis in that they do not have access to the code or the implementation specifics
- DASTs only inspects the system's responses to a series of tests designed to highlight vulnerabilities
- They function independently of the underlying application platform and offer solid support for manual pentesting
- A top-level DAST detects only ~20% of issues with no information provided on the location of the issue in the code base
- An experienced security background is necessary to interpret the results

SAST vs. DAST

SAST

- Offers a static and internal analysis of the application
- A SAST code scanner will find more security control holes than DAST, at the cost of having a high percentage of false positives
- Serves well in the development and QA stages

DAST

- Offers dynamic (runtime) and external analysis of the application
- A DAST web scanner is more useful during QA and production stages, since the application is running
- Does not require direct access to the source code, while SAST does

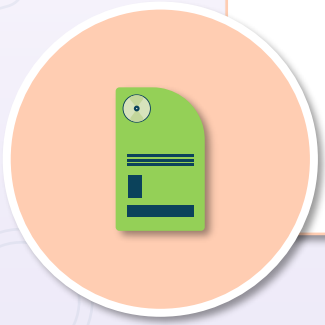
Interactive Application Security Testing (IAST)

- IAST combines the advantages of a SAST and a DAST solution
  - The benefits of a static view, because they can see the source code
  - The benefits of a web scanner approach, since they see the execution flow of the application during runtime
- Can detect ~100% of OWASP benchmark in real-time with no false positives
- Can flexibly be used in QA and production environments, analyzing dependencies as well as legacy components
- No need to scan or attack the application
- Continuous detection – DevOps-friendly
- Integrates and communicates with task management systems to create unified workflows

Commercial Off-the-shelf (COTS)

- A common commercial off-the-shelf as-is solution
- COTS products are intended to be easily installed and to interoperate tightly with existing system components
- Almost all software bought by the public computer user fits into the COTS category (operating systems, office product suites, word processing, and e-mail programs)
- One of the major advantages of mass-produced COTS software is that it is relatively low cost

Modified Off-the-shelf (MOTS)

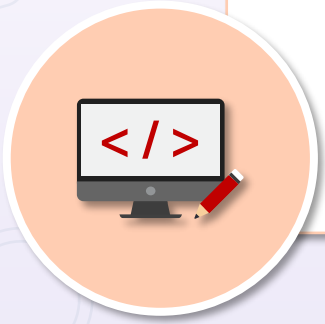


- MOTS stands for either modified or modifiable off-the-shelf, or military off-the-shelf product (dependent on the context)
- It is typically a product whose source code can be modified or customized by the purchaser, the vendor, or another party to meet the needs of the customer
  - In the military, it refers to an off-the-shelf product that is developed or customized by a vendor to respond to precise military requirements
- Since MOTS software specifications are written by external entities, some government agencies are often untrusting of these products, as they fear that future changes to the product will not be in their control

Government Off-the-shelf (GOTS)

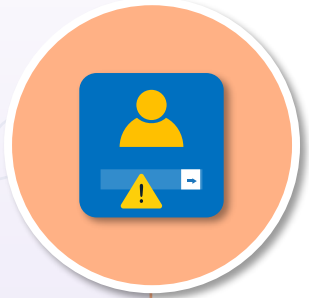
- A government off-the-shelf product is normally produced by the technical staff of the government agency for which it is created
- It can also be developed by an external contracted body, but with financing and specifications from the agency
- Since government agencies can directly manage all characteristics of GOTS solutions, these are commonly preferred for government use cases

Niche (or NATO) Off-the-shelf (NOTS)



- NOTS can stand for NATO off-the-shelf or niche off-the-shelf product (depending on the context)
- Generally, niche off-the-shelf refers to vendor-developed software that is for a particular well-defined market segment, as opposed to a broad market for COTS products
- If it is developed for NATO Consultation, Command, and Control (NC3A), then it is to meet explicit NATO requirements

Open-source Vulnerabilities



- Open-source software is a package whose code is accessible for public examination, alteration, and improvement
- Many enterprises and products (90% by some estimates) use at least one open-source component, often without being aware of it
- Normally, this software is built using public community collaboration and is preserved and updated on a voluntary basis
- Open-source software can be used according to a diversity of licenses, depending on what the developers have implemented

Open-source Vulnerabilities

- Vulnerabilities are publicly known
- No claims or obligations to be secure
  - Open-source software often includes or demands the use of vulnerable third-party libraries
- Intellectual property challenges
  - There are over 200 types of licenses that can be used with open-source software
- Lack of warranty for its security, support, or content

Open-source Vulnerabilities

- Lax integrations oversight and control
  - Dev teams often have non-existent review processes for open-source components
- Operational inadequacies requiring additional work for proper DevSecOps
- Poor development practices and procedures
  - Risks increase as developers commonly copy and paste chunks of code from open-source software
  - Developers often transfer components through email or use poor repository security practices

Source-code Weaknesses

- Code vulnerabilities exist because solid secure development is quite difficult
- Whether it's proprietary or open-source code, software will unavoidably have vulnerabilities
- Experts often agree that open-source code libraries are far more secure than commercial software
  - Open-source code isn't inherently more secure; **it is more securable**
- Most organizations do not have a clearly defined policy to confirm that developers wanting to use a section of software code go through an authorization process

Source-code Weaknesses

- Since open-source components exist in almost all codebases, keeping up with open-source components in your software is an overwhelming task – including tracking the forks, versions, and state of updates to the code
- Developers need to move beyond SAST and DAST solutions by implementing continuous code analysis
- New solutions should have all the advantages of the available IAST solutions with machine learning and advanced AI tools

Common Programming Weaknesses

- Poor error handling
- Poor exception handling
- Improper input validation
- Not relying on stored procedures
  - Precompiled groups of code, statements, and commands that can be called later
- Unsecure usage of code repositories
- Leaving inoperative dead code
- Redundancy in the code (no normalization)

Application Programming Interface (API) Security

- Do not embed credentials or access keys into API calls
- Use forward secrecy on API keys
- Protect remote communication channels with SSH2 or TLS
- Leverage IAM and SSO federated solutions for best results and add MFA solutions if feasible
- Developers should use OWASP API Top Ten: **https://owasp.org/www-project-api-security/**

OWASP API Top Ten

OWASP API Top 10

API1:2019 Broken Object Level Authorization

API2:2019 Broken User Authentication

API3:2019 Excessive Data Exposure

API4:2019 Lack of Resources & Rate Limiting

API5:2019 Broken Function Level Authorization

API6:2019 Mass Assignment

API7:2019 Security Misconfiguration

API8:2019 Injection

API9:2019 Improper Assets Management

API10:2019 Insufficient Logging & Monitoring

Secure Coding Techniques

- Normalization
  - Involves ensuring there is no redundancy in data and that similar items are stored together (also deduplication)
- Proper input validation
  - Verifying the input before entering it into the system
  - Also includes proper error/exception handling
  - Errors should be captured with secure logging (SIEM)

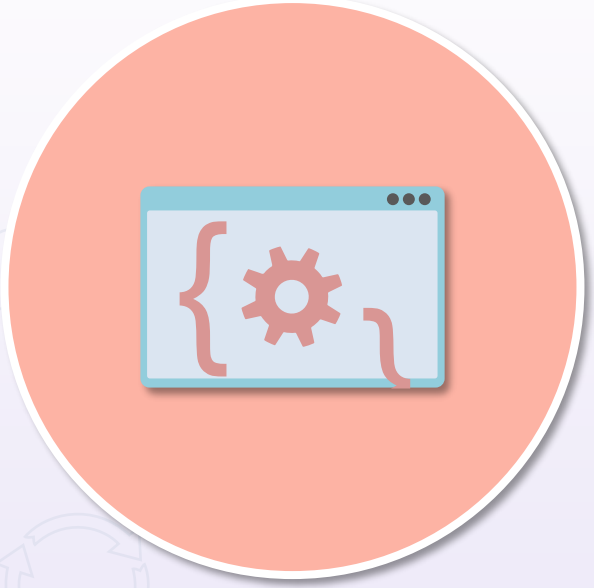
Secure Coding Techniques

- Stored procedures
  - Precompiled groups of code, statements, and commands that can be called later
  - Also called "code re-use", where one deliberately leverages existing, tested, and validated code that can be used again
  - Use Infrastructure as code when possible (Terraform, AWS CloudFormation)
- Obfuscation/camouflage
  - Involves writing code that humans have a hard time understanding
- Code signing
  - Digitally signing code to prove authorship, provide integrity, and enforce non-repudiation
  - Also ensure the confidentiality of the code in transit, at rest, and in use

Secure Coding Techniques

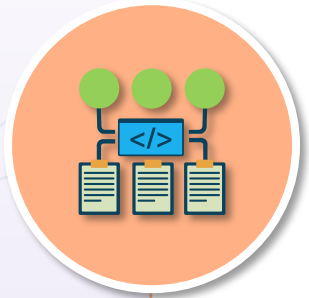
- Version control
  - Track changes to code
  - Document at GitHub or CSP, for example
- Change management
  - Direct, control, and support changes in code efficiently
  - Can be planned or unplanned
- Provisioning and deprovisioning
  - Providing or removing software access

Secure Coding Techniques



- Server-side vs. client-side execution and validation
  - Client-side is more efficient, but server-side is more secure
- Memory management
  - Allocate memory/buffer when needed, release it for re-use when no longer needed
- Use of third-party libraries and SDKs
  - May violate user's privacy, may damage the quality of the application

Software-defined Security (SDS)



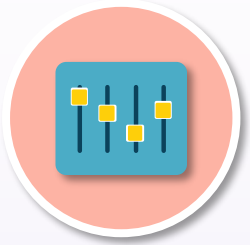
- Software-defined Security (SDS) is a model in which the information security is highly controlled often using virtualization
- The functionality of network security devices, such as next-gen firewalls intrusion detection and prevention, identity and access controls, and network segmentation are removed from hardware devices to a software layer
- SDS exploits the software-defined networking (SDN) initiative to enhance network security
- The concept of software-defined security is envisioned to define IT infrastructure security services as a transition from hardware based to a software-defined solution

SDN and SDS

HOST

The host role is to transmit or receive data through the network

For the SDS, all security techniques are transferred to the controller

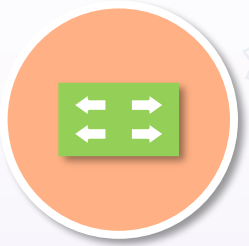


CONTROLLER

The controller is fully software-based

All security checks are done inside the controller, and it has total visibility of the traffic flows

It collects and processes information about the network



SWITCH

The switch checks with the controller to determine whether to accept or reject a request

A reactive caching mechanism is adopted in SDN, which does make SDN switches vulnerable to a DoS attack

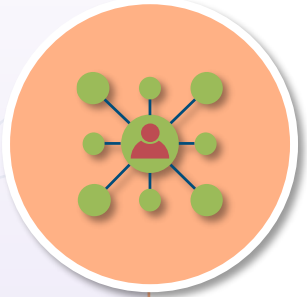
Advantages to SDS

- Offers resourceful and dynamic countermeasures to security attacks
- Separates security away from traditional hardware vulnerabilities
- Ability to dynamically configure existing network nodes allows for rapid attack mitigation from zero-day attacks
- Synchronized view of logical security policies exist within the SDN controller model (not tied to any server or specialized security device)
- Visibility of information provided from one source
- Integration with emerging technology to correlate events in a simpler way and respond more efficiently and intelligently to threats
- Enables centralized management of security, which is implemented, controlled, and managed by security software through the SDN controller
- Facilitates IoT & BYOD connectivity and security

Software Diversity

- Software diversity is an application development methodology where two or more functionally duplicate versions of the app are developed from the same specification
- The process uses different developers or programming teams
- The goal is better error detection, improved consistency, and fewer programming errors

Software Diversity



- End-user applications are often written in modern programming languages, like Java and others
- The operating systems, firmware/middleware, support libraries, and virtual machines are still written in low-level languages that place flexibility and performance over security
- Programming errors in low-level code are often exploitable and can sometimes give attackers unrestricted access to compromised host systems
- Automated software diversity techniques use randomization to significantly increase the difficulty of exploiting the huge amounts of low-level code in existence
- Diversity-based defenses are motivated by the assumption that a single attack will fail against multiple targets with unique attack surfaces

Software Assurance

- The key objective of the Software Assurance Program is to shift the security paradigm from patch management to software assurance
- Encourage developers to raise overall software quality and security from the start
- Emphasize the usage of tested standard libraries and modules
- Employ industry-accepted approaches that recognize that software security is fundamentally a software engineering issue that must be addressed systematically throughout the software development life cycle