RISK AND RESILIENCE BOOTCAMP

TEKsystems Global Services

WORKFORCE DEVELOPMENT

CONTROLS

This section is an introduction to controls

- Preventive Controls
- Detective Controls
- Corrective Controls

PREVENTIVE CONTROLS

- Preventive controls
  - Proactive risk and resiliency management
  - Measures put in place before an incident occurs
    - Designed to stop or reduce the likelihood that a risk event occurs or to inhibit its progress
    - The goal is to prevent undesirable events from happening
- In ISACA terms
  - Preventive controls are part of “control activities”
  - Safeguards to maintain integrity, availability and confidentiality
  - In risk vocabulary, they reduce inherent risk by influencing likelihood of events occurring
- In DRI / resilience thinking
  - Preventive controls are the “absorb” or “anticipate” buffer for the impact of events
  - They help avoid or reduce the shock before it becomes a disruption

PREVENTIVE CONTROLS

- Perform risk assessment and threat modeling
  - Identify key risk scenarios, threat types, and vulnerabilities
  - Estimate likelihood, impact, and residual risk
- Prioritize by risk appetite and cost-benefit
  - Use cost, complexity, expected reduction in risk, and residual risk to rank candidate controls
  - Choose controls whose cost is justified by risk reduction
- Consider control layering and defence in depth
  - Don’t rely on a single control
  - Use overlapping layers
    - For example: combine network controls, authentication, and process controls

PREVENTIVE CONTROLS

- Ensure feasibility and compatibility
  - The control must integrate with existing architecture without undue fragility
  - Consider performance, usability, maintainability
- Map to frameworks and standards
  - ISACA governance framework
  - COBIT/COBIT-derived frameworks
  - Align control objectives to domain goals

PREVENTIVE CONTROLS

- Define control requirements and specifications
  - What must the control accomplish?
    - For example: block unauthorized access, enforce encryption
  - Determine metrics, thresholds, SLAs, exception handling, and review cycles
- Document in a control catalog or risk-control matrix
  - Map risks to controls, mark preventive / detective / corrective
  - Tracks ownership & testing
  - ISACA often expects controls to be documented, tested, and periodically reviewed

PREVENTIVE CONTROLS

IT Domain / Function	Preventive Control Example	Description / Why Preventive
Access / Identity	Role-Based Access Control (RBAC), least privilege	Prevents unauthorized access by limiting permissions
Network / Perimeter	Firewalls, Network Access Control (NAC)	Blocks unwanted traffic before it enters network segments
Endpoint / System Hardening	Patch management, OS hardening, disabling unused services	Prevent vulnerabilities from being exploited
Authentication / Cryptography	Multi-Factor Authentication, password policies, encryption of data at rest	Prevent unauthorized authentication or data exposure
Change Management	Approval workflows, segregation of duties (SoD)	Prevent unauthorized or risky changes from being applied
Development / SDLC	Secure coding standards, static code analysis, code reviews	Prevent introduction of insecure coding or vulnerabilities
Physical / Environmental	Door locks, server room environmental controls, CCTV	Prevent physical intrusion, damage, or tampering
Training / Awareness	Security awareness training, phishing simulations	Prevent human error or social engineering from succeeding

DETECTIVE CONTROLS

- Detective control
  - Mechanisms designed to identify, detect, or alert when an adverse event or anomaly occurs
    - Backup if preventive controls fail or something unexpected happens
    - Their goal is to provide visibility, raise alarms, and feed information to response functions
    - Validate via data collection that preventative controls are working
- In ISACA and assurance practice
  - Detective controls allow validation that preventive controls are functioning
  - Detective controls are also designed to respond to exceptions or anomalies
- In resilience terms
  - Detective controls contribute to the “sensing” or “detect and respond” capabilities
  - They are often triggers for compensatory responses

DETECTIVE CONTROLS

- Identify key indicators, signals and events
  - Determine which system events, logs, patterns, or deviations are meaningful
  - Define thresholds, anomaly detection, correlation logic
- Ensure coverage across layers
  - Not only network or host, but also application, database, user behavior, business processes
- Balance sensitivity, noise and false positives
  - Tune thresholds to avoid “alert fatigue”
  - Use correlation, context, and filters to reduce irrelevant alerts

DETECTIVE CONTROLS

- Ensure timeliness
  - Detect quickly enough that corrective action can beat escalation
  - Define SLAs or time thresholds for detection
- Link to response processes
  - Alerts should feed to incident response plans, escalation paths, dashboards
  - The outputs of detective controls must be actionable
- Log and auditability
  - Maintain secure, tamper-evident logs or event capture
  - Ensure logs are retained for forensic analysis

DETECTIVE CONTROLS

- Periodic review, tuning and testing
  - Regularly test detection rules and simulate attacks (red teaming, logging exercises)
  - Evaluate gaps or blind spots
- In ISACA or audit frameworks
  - Detective controls often are tested during assurance engagements to validate control effectiveness

DETECTIVE CONTROLS

IT Domain / Function	Detective Control Example	Description / Role
Network / Traffic	Intrusion Detection Systems (IDS), Network Packet Capture / Flow Logging	Detect unusual traffic patterns or known signatures
Security Monitoring / SIEM	SIEM systems aggregating logs, correlating events, alerting	Central hub to correlate events across systems for suspicious behavior ISACA +3
Host / Endpoint	Host intrusion detection (HIDS), malware detection, file integrity monitoring	Detect anomalous file or process behavior
Application / API	Application logging, anomaly detection, input validation alerts	Detect unusual API usage, failed authentication spikes, abnormal transactions
Database / Data	Audit logs, data integrity checks, anomaly detection on data access	Detect unauthorized queries, changes, or exfiltration
User / Behavior	User and Entity Behavior Analytics (UEBA), anomalous login patterns	Detect deviation from normal user behavior (e.g., access at strange times)
Audit / Review / Reconciliation	Periodic audit of system logs, reconciliation reports, exception reports	Detect financial mismatches or unexpected discrepancies
Physical / Environmental	CCTV, badge access logs, environmental alarms	Detect unauthorized physical access or environmental anomalies (e.g. temperature, humidity)

CORRECTIVE CONTROLS

- Corrective controls
  - Restoration, damage control, prevent recurrence
- Are put in place after a detected event
  - Intended to restore systems, mitigate damage, contain ongoing issues
  - Designed to prevent recurrence of the same or similar event
  - Respond to the incident, remediate, and improve controls to avoid repeat incidents
- In resilience DRI resilience context
  - Corrective controls often combine containment, recovery, restoration, and learning
- In ISACA assurance
  - Corrective controls are part of incident response, business continuity and disaster recovery processes, patching, and root cause remediation

CORRECTIVE CONTROLS

- Incident response and root cause analysis
  - Once a control failure or security event is detected, perform root cause analysis
  - Identify what went wrong
    - For example: control failure, process gap, misconfiguration, human error
- Define remediation steps and plan
  - What must be done to contain, isolate, remove, or patch the threat
  - Define roles, timelines, rollback plans, dependencies
- Ensure rapid recovery and restoration
  - Use backups, redundancy, failover, or alternate systems
  - Prioritize restoring critical services first

CORRECTIVE CONTROLS

- Implement preventive enhancements
  - After recovery, modify preventive and detection controls so that the same event doesn't recur
  - Use it as a "lesson learned" to improve preventative controls
- Test and validate the remediation
  - After corrective changes, test to ensure they work and don't break other systems
- Update policies, procedures and training
  - Update documentation, checklists, standard operating procedures, and train staff
  - Ensure that human processes incorporate the new lessons
- Monitoring and audit of the corrected state
  - Monitor over time to detect if the fix holds, or if regressions occur
  - Periodic reviews and audits

CORRECTIVE CONTROLS

IT Domain / Function	Corrective Control Example	Description / Role
Incident / Security	Incident Response Plan, Forensic Investigation, Patch Deployment	After detection, respond, isolate affected systems, apply patches or fixes
Data / Storage	Backup restoration, data integrity correction, data recovery procedures	Recover lost or corrupted data to functional, consistent state
System / Application	System reimaging, repair scripts, hotfix application	Restore system binaries, remove malware, rebuild system if compromised
Change / Configuration	Rollback changes, corrective change deployment, configuration revision	Correct misconfigurations or faulty changes
Process / Control Gap	Process redesign, control strengthening, root cause training	Adjust internal procedures to prevent recurrence
Continuity / Resilience	Failover to backup systems, activate alternate site, switchover	Use continuity strategies to maintain operations while correcting the failure
Monitoring / Logging	Update detection rules, tighter thresholds, improved logging	Enhance detective capabilities to catch similar future events earlier

EXAMPLE

- A web application vulnerability is exploited
- The corrective control steps might be
  - Quarantine or take the web server offline (containment)
  - Restore from clean backup or re-deploy the patch (restoration)
  - Apply patch or upgrade the vulnerable module
  - Conduct root cause analysis to see why the patch was not applied earlier
  - Update patch management procedures and strengthen monitoring so similar exploit attempts are caught earlier
  - Validate the patch works and monitor logs to confirm no re-infection

EXAMPLE

- A mis-configuration causes downtime
- The corrective control steps might be
  - Rollback to the last known good configuration
  - Fix the misconfiguration script
  - Add automated pre-deployment checks
  - Retest

INTEGRATION & CONTROL DESIGN

- ISACA assurance controls
  - Controls are evaluated for design effectiveness and operating effectiveness
  - Auditors will map
    - Requirements (laws, policies, standards) to control objectives then to specific controls
  - Control framework or control catalog
    - Includes control types (preventive, detective, corrective)
    - Ownership, testing frequency, and metrics
  - Assurance engagements
    - Test that preventive controls are in place
    - Test that detective controls detect failure
    - That corrective controls properly restore and prevent recurrence

INTEGRATION & CONTROL DESIGN

- DRI, resilience and continuity perspective
  - Controls are integral to resilience strategies
    - Preventive controls reduce disruption frequency
    - Detective controls enable quick detection and response
    - Corrective controls enable recovery and adaptation
  - In business continuity and disaster recovery planning
    - Often see corrective controls and recovery strategies merging (DR plans, backup, alternate site)
  - The DRI community emphasizes
    - Lessons learned, post-incident review, and improvement
    - Feedback loops from corrective to preventive

STANDARD FORMS - TEMPLATES

- Often used in ISACA or enterprise risk management settings:
  - Risk-control matrix or control mapping matrix
    - Maps risks to controls, control types (prevent/detect/correct), control owner, control frequency, test procedures
  - Control catalog
    - Repository of defined controls with descriptions, type classifications, performance criteria
  - Test plans and control test scripts
    - Implemented for each control
    - How a control will be tested for design and operating effectiveness
  - Root cause template
    - Captures event, detection, corrective actions, lessons learned, preventive enhancements
  - Change request templates
    - Include required control impact analysis
  - Control self-assessment (CSA) forms
    - For control owners to self-certify control status, exceptions, and remediation

PROCESS MATURITY AND CONTROLS

- Controls are only as strong as the processes that define, apply, and monitor them
  - Maturity directly affects control design, consistency, effectiveness, and adaptability
- . Control design quality
  - In immature organizations, controls are reactive, informal, or not properly aligned to risks
  - Mature organizations design controls based on risk analysis, data, and policy frameworks, leading to more precise preventive, detective, and corrective mechanisms
- Control execution and consistency
  - Immature processes depend on people's discretion ("tribal knowledge")
  - Mature processes embed controls in standardized workflows, automated pipelines, and governance tools: ensuring consistency and repeatability

PROCESS MATURITY

Maturity Level	Typical Characteristics	Control Environment Implications
Level 1 – Initial (Ad hoc)	Processes are informal, inconsistent, person-dependent	Controls are inconsistent or nonexistent; rely on individual vigilance
Level 2 – Repeatable (Managed)	Basic discipline and documentation; processes repeatable by trained staff	Some preventive and detective controls exist, but may lack full coverage or testing
Level 3 – Defined	Standardized and documented processes across the organization	Controls are embedded, standardized, and consistently applied
Level 4 – Managed (Measured)	Processes are quantitatively measured and monitored	Controls are measured for performance; deviations trigger action automatically
Level 5 – Optimized (Continuous Improvement)	Processes continuously improved based on metrics and feedback	Controls are adaptive, automated, and continually enhanced to handle emerging risks

PROCESS MATURITY AND CONTROLS

- Monitoring and measurement
  - Low-maturity organizations rarely test control effectiveness; failures go unnoticed
  - At higher maturity, control performance is monitored with defined KPIs/KRIs (e.g., number of incidents detected, MTTR, patch cycle compliance)
  - This feedback allows continual refinement: necessary for resilience
- Adaptability and resilience
  - Immature controls struggle to adapt to new risks (e.g., emerging cyber threats, regulatory changes)
  - Mature controls are part of a living system: continuously assessed and improved via post-incident reviews, audits, and lessons learned.

PROCESS MATURITY AND CONTROLS

- Integration across functions
  - Early maturity: siloed departments, fragmented control ownership
  - High maturity: integrated governance (e.g., GRC systems, shared risk registers) where controls align with strategic, operational, and compliance goals

PROCESS MATURITY AND CONTROLS

Maturity Level	Preventive Controls	Detective Controls	Corrective Controls
1 – Initial	Ad hoc access restrictions; manual approvals	Manual log reviews; reactive troubleshooting	Informal recovery efforts; untested backups
2 – Repeatable	Documented procedures; password policies	Scheduled monitoring; basic event logs	Manual incident response playbooks
3 – Defined	Automated access control; change management workflows	SIEM, automated alerts, audit trails	Documented DR plans; tested restoration
4 – Managed	Policy-driven automation; configuration management tools	Real-time monitoring, behavioral analytics	Automated failover; continuous improvement tracking
5 – Optimized	Predictive security analytics; AI-driven prevention	Adaptive anomaly detection; threat intelligence integration	Self-healing infrastructure; continuous learning and control tuning

PROCESS MATURITY AND CONTROLS

- From a risk management perspective
  - Higher process maturity reduces control risk, which is the risk that controls will fail to prevent or detect issues
  - Mature organizations can more accurately predict, prevent, and recover from disruptions
- From a resilience management (DRI/CERT-RMM) perspective
  - Mature processes foster operational resilience expressed as the ability to maintain critical functions under stress
  - Resilience maturity grows in tandem with control maturity

ASSESS AND IMPROVE MATURITY

- Perform a control maturity assessment
  - Use frameworks like COBIT 2019's Process Capability Model or CERT-RMM
  - Evaluate for each process: definition, documentation, measurement, and continuous improvement
- Identify control weaknesses or gaps
  - Compare current controls to framework expectations or regulatory standards
  - Determine if gaps arise from low maturity (e.g., no ownership, no automation, poor testing)
- Prioritize improvements
  - Focus first on processes with the highest risk exposure and lowest maturity
- Institutionalize continuous improvement
  - Build post-incident reviews, audits, and self-assessments into governance cycles
  - Create feedback loops that automatically refine controls over time

Q&A AND OPEN DISCUSSION