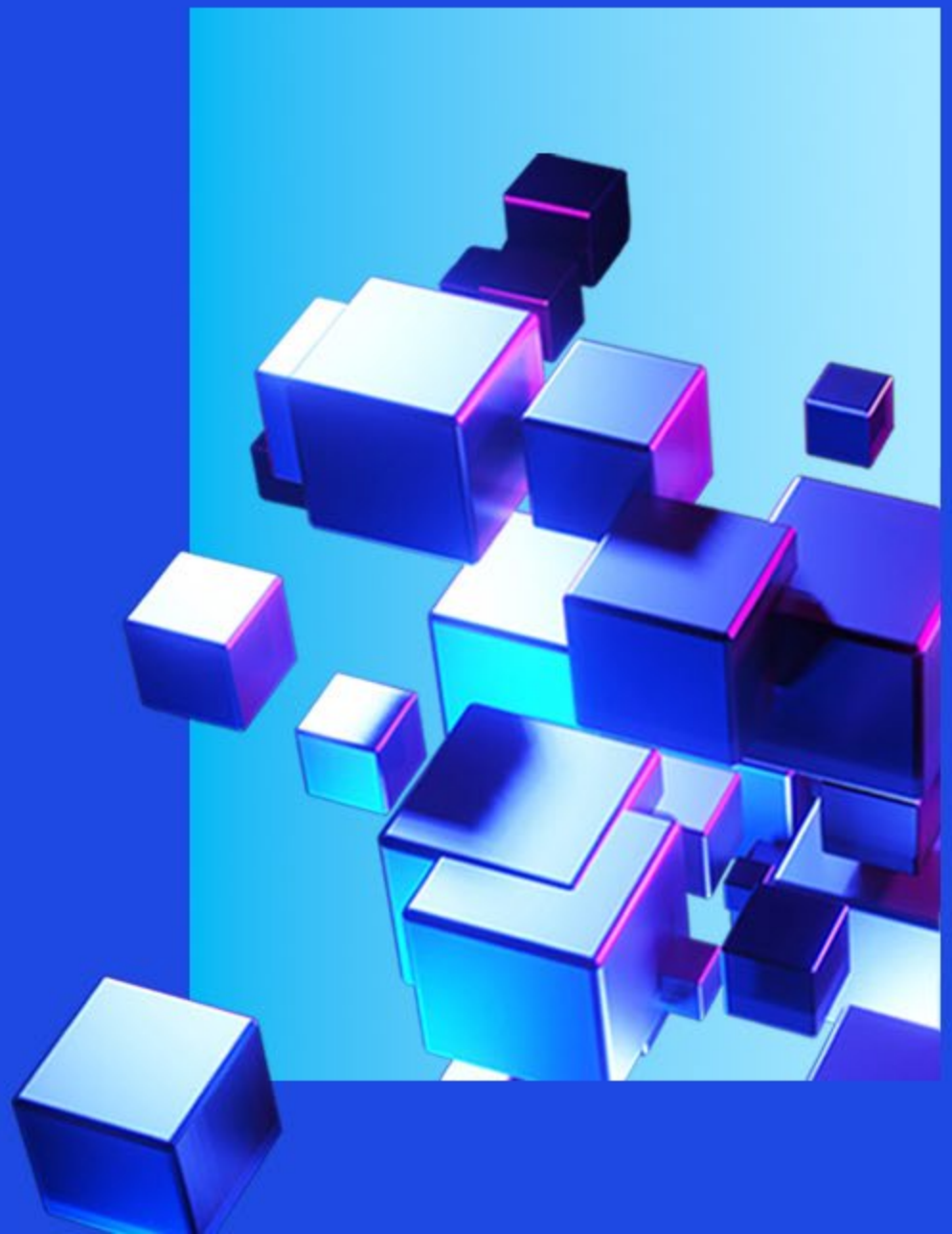
KPMG

From Theory to Practice: Mastering IT Scoping for SOX Compliance



A KPMG *Future of SOX* webcast

—

February 26, 2026 | 11:00 a.m.–12:30 p.m. ET

With you today

Copresenters

Sue King

US SOX Leader
KPMG LLP

Steven Estes

US SOX Leader
KPMG LLP

Subash Samuels

Principal,
Internal Audit & Controls
KPMG LLP

Agenda

1. ICFR risk assessments
2. Scoping of Systems and Architecture
3. Scoping of IT layers
4. IT Hot Topics
   - I. Code review
   - II. IT audit inspection findings
   - III. Agile SDLC risks
   - IV. User access reviews

ICFR risk assessments

SOX methodology overview – Phases

Strategy

- Articulate the strategy behind the SOX program at the company.
- Focus on communication between external and internal auditors.
- Consider SOX like any other strategy in the organization—what should it do for the company?

Risk assessment

- Perform a strong risk assessment that connects risks and audit assertions, leverages direct ELCs, and ultimately impacts control testing.
- Consider this the key element to "rightsizing" a SOX program.

Entity-level controls

- Help identify and define ELCs to act as the company's "insurance policy" against significant control failures.

Control selection

- Scope and select the right controls to be tested.
- Focus on alignment between the types of controls—preventative versus detective and automated versus manual.
- Consider potential for improvement in the design and automation of controls.

Testing strategy

- Test the execution for the design and effectiveness of key controls.
- Document testing results and exceptions.
- Consider the use of data analytics and continuous monitoring.

Evaluating results

- Follow steps to evaluate the identified deficiencies.
- Help management assess whether deficiencies aggregate to significant deficiencies or material weaknesses.

Governance

- Determine SOX governance structure within the company.
- Consider where the SOX program reports and whether there is appropriate management involvement/oversight.

Importance of a robust risk assessment

The risk assessment is the most vital part of the scoping process as it establishes priorities and supports the rationale for subsequent actions.

It's the foundation of the top-down, risk-based approach

A top-down, risk-based audit is fundamental to SOX. The risk assessment is what makes it "risk based." Without it, you are not following the required methodology.

It creates efficiency and focus

It allows the team to concentrate effort on the systems and controls that truly matter. Instead of testing every single application, you focus only on those that mitigate a specific, high-risk area. This saves an enormous amount of time and resources.

It provides a defensible position

It provides the clear rationale for why a system is in scope. When you decide an application needs to be audited, it's not a subjective choice; it's because your risk assessment identified it as being critical to preventing a material misstatement.

It directly identifies key controls

You can't know which ITACs are important until you know which risks you are trying to mitigate. The risk assessment process is what logically filters numerous potential controls down to key controls that are essential for the audit.

Key components of an effective risk assessment

In the context of SOX compliance, a good risk assessment is not a generic business exercise. It is highly focused on financial reporting and includes these key steps:

Identifying “what could go wrong” (WCGW)

- For each key business process (i.e., financial reporting), you identify the specific points where an error or fraud could occur that would lead to a material misstatement in the financial statements (“What Could Go Wrong”).

Linking risks to financial assertions

- Each identified risk is then linked to a specific financial statement assertion. For example:
  - Risk: “A duplicate payment could be made to a vendor.” → This impacts the accuracy/valuation assertion.
  - Risk: “A shipment could be made but never invoiced.” → This impacts the completeness assertion for revenue.

Evaluating risk impact

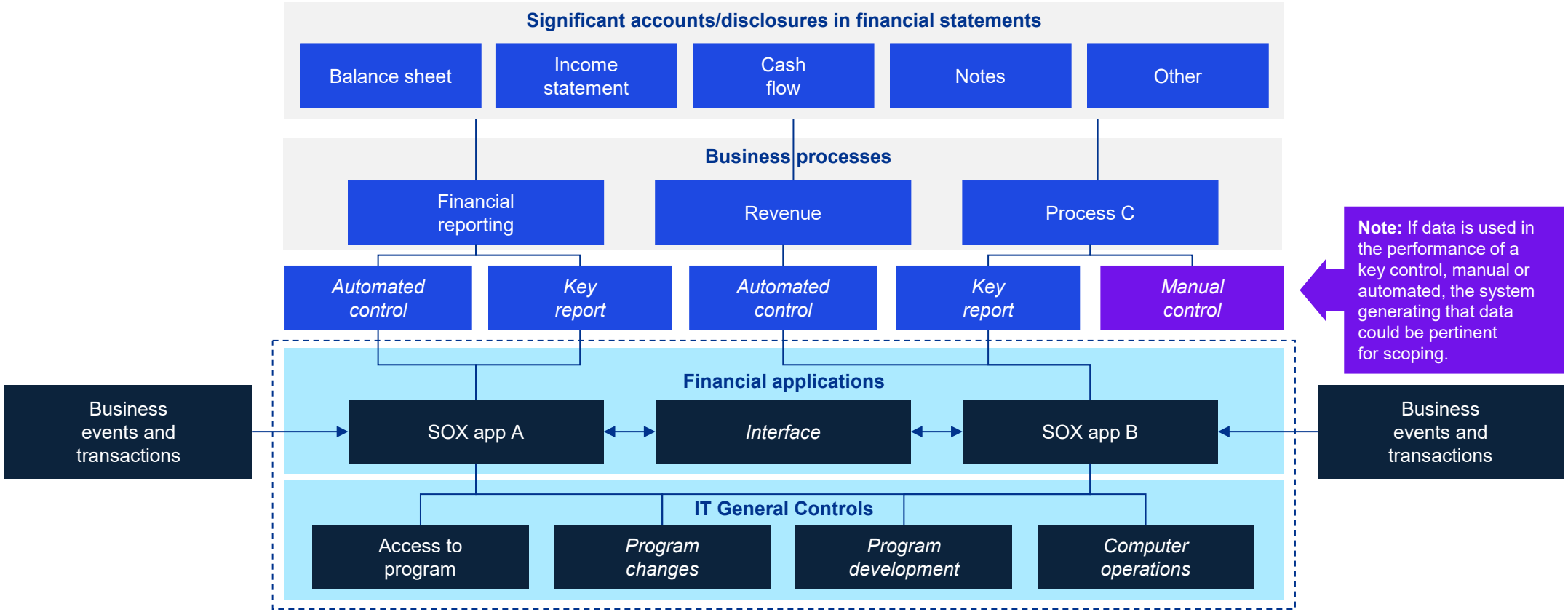
- You assess the potential magnitude of the risk. The focus is always on risks that are significant enough to be considered “material” to the financial statements, not minor typos.

Identifying existing controls

- For each significant risk, you identify the key controls that are already in place to mitigate it. This is the step that directly leads you to the critical manual controls and automated application controls (ITACs).

ICFR scoping and risk assessment: You need a defensible position

Integrated financial and IT controls



Scoping of systems and architecture

The scoping workflow

From business process to IT controls

Start with the business process

- An **end-to-end** approach reflects the modern audit focus on dataflow and the potential risks that exist in the connections between systems.
- Begin by performing detailed walkthroughs of each initial in-scope business process.
- The goal is to deeply understand the process, identify key process risk points (PRPs), and create a detailed list of necessary controls and potential gaps.
- **It is the key to understand the evolving view on “dataflow.”**

Perform application scoping

- This scoping is driven by the business processes identified earlier and involves identifying the applications that support those processes.
- We create an initial list of relevant applications, which serves as a starting point that is then validated and refined through further discussions.

Identify controls & assess risk

- The ultimate goal is to assess IT risks, including risks of material misstatements (RMMs).
- We do this by connecting the business “risk points” to the relevant automated control activities (ITACs) within the in-scope applications.
- This is documented with evidence such as dataflow diagrams and control descriptions.

Importance of process understanding

Process

The actual steps necessary to record an amount in the financial records or prepare a disclosure in accordance with the applicable financial reporting framework

Reminder: A process owner (i.e., preparer) reviewing their own work cannot be considered a control.

Process step =

Step taken to record an amount or prepare a disclosure. Process steps introduce RMMs (i.e., where error could occur)

PRP =

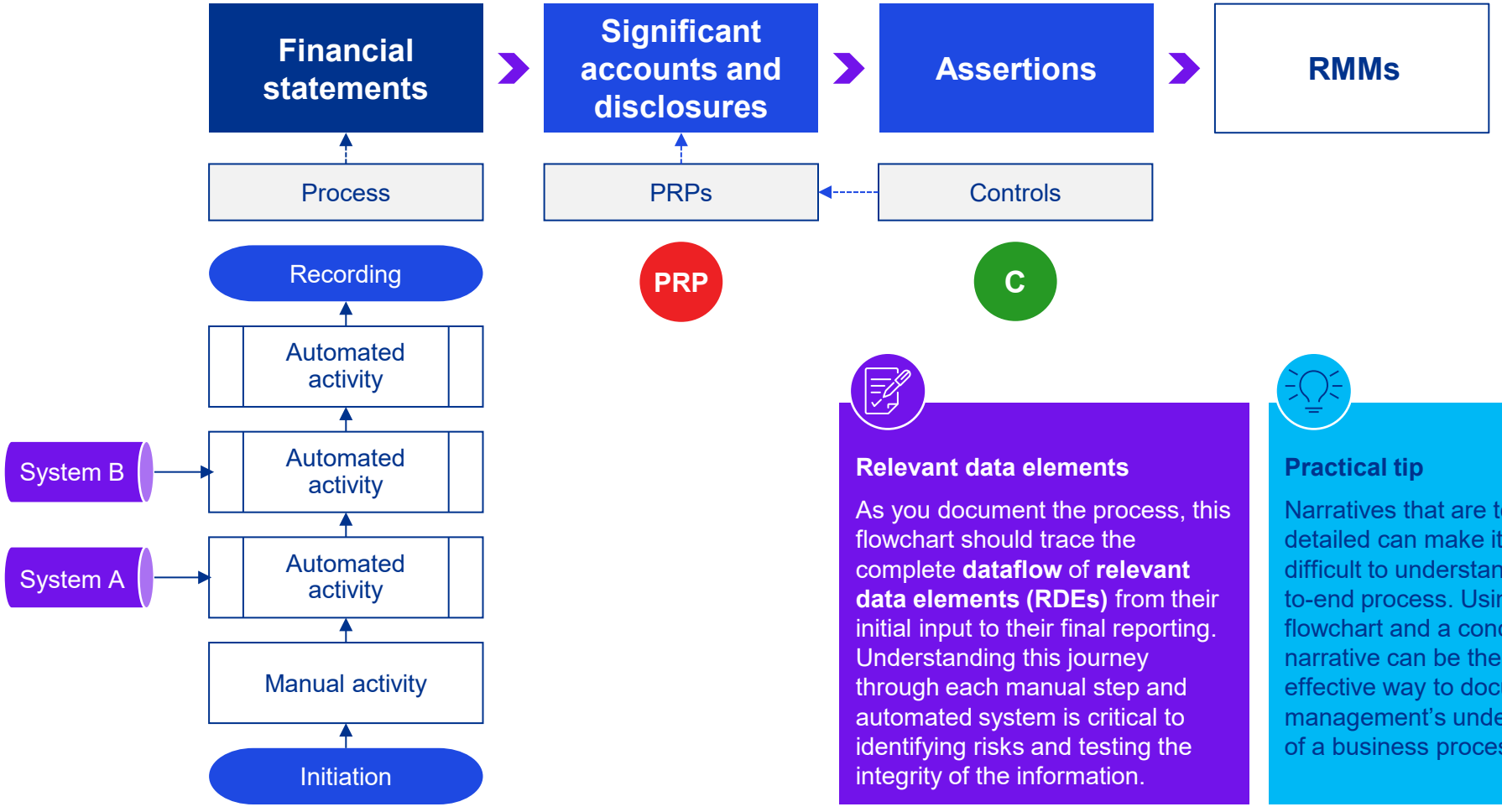
Point in the entity's process where a misstatement could occur and, individually or in aggregate, yield a material misstatement to the financial statements

Process control activity =

Specific actions taken to mitigate PRPs identified

End-to-end process documentation

Flowchart



Relevant data elements

As you document the process, this flowchart should trace the complete **dataflow** of **relevant data elements (RDEs)** from their initial input to their final reporting. Understanding this journey through each manual step and automated system is critical to identifying risks and testing the integrity of the information.

Practical tip

Narratives that are too long and detailed can make it more difficult to understand the end-to-end process. Using both a flowchart and a concise narrative can be the most effective way to document management's understanding of a business process.

When to scope a system in or out

The fundamental principle

We scope a system in when our risk assessment identifies a clear “chain of impact” between that system and our internal controls over financial reporting (ICFR). Conversely, we scope a system out when that chain of impact is broken and it is determined to have no effect on our financial controls.

The Litmus Test (Question/Rationale)	
Does the system create, transform, aggregate, or route data that is (directly or indirectly) recorded to the GL, subledgers, and disclosures or used in estimates/judgments (e.g., revenue recognition)?	A direct or indirect linkage to a financial assertion is the primary scoping driver in a top-down risk assessment.
Is any key control executed in this system (e.g., an automated calculation and a configured control), or does an end-user control rely on this system’s output (IPE)?	The location of key controls (whether fully automated or IT-dependent manual controls) dictates where IT general control (ITGC) and IPE testing is required.
Do downstream key controls (such as reconciliations, management reviews, or exception monitoring) depend on the completeness and accuracy of data that comes from this system or its interfaces?	If the evidence for the C&A of a key control originates from this system, then ITGCs or specific interface testing for this system is necessary to prove the data is trustworthy.
Could improper privileged access (e.g., by a system administrator) or a defective change to this system allow someone to bypass detective controls and enable a material misstatement? (Consider sensitive configurations such as rate tables, posting rules, or revenue logic).	The presence of sensitive, financially impactful configurations or custom code magnifies the system’s risk profile, making robust ITGCs (such as access and change management) critical.
Is the system a software as a service (SaaS) or otherwise outsourced application for which you will rely on a SOC report to provide assurance over key controls or the integrity of its data (IPE)?	Even if controls are outsourced, the system remains SOX relevant. The audit scope shifts to evaluating the SOC report and testing the company’s complementary user entity controls.

You can generally scope out an IT system if the answer to all of the following questions is ‘No’

The scoping decision: A practical walk-through of the litmus test

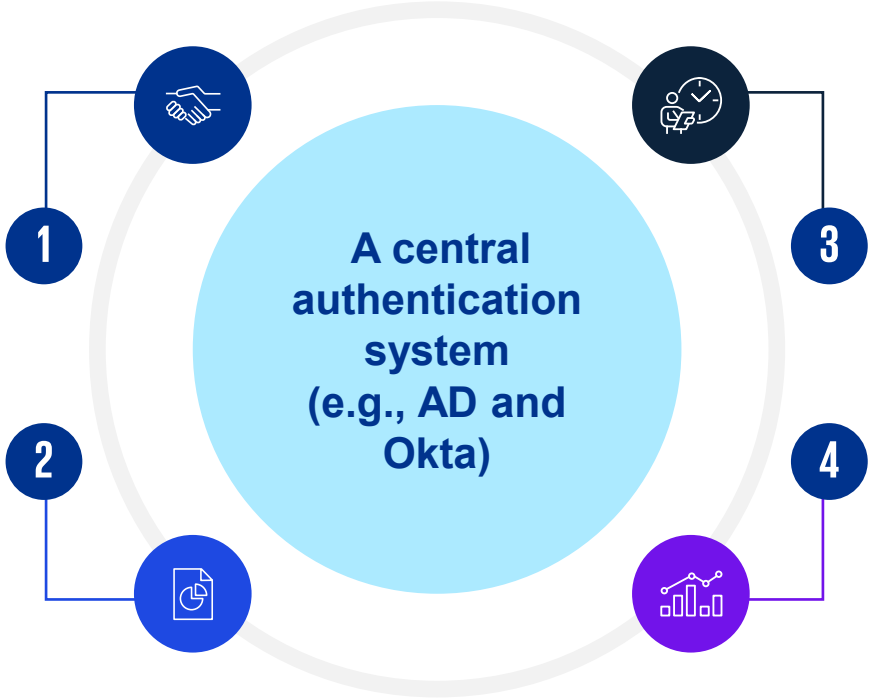
The scenario: A company uses a central identity and access management system, such as Active Directory (AD), to control user authentication (log-ins) for nearly all corporate applications, including the in-scope ERP and financial reporting systems. AD itself does not store any dollar amounts or financial transaction data.

Does the system create, transform, aggregate, or route data that is (directly or indirectly) recorded to the GL?

No. AD's primary function is to manage and route identity data (e.g., user credentials and group memberships), not financial data. It answers the question "Is this user who they say they are?" but it does not create, process, or store invoices, journal entries, or revenue figures.

Is any key control executed in this system, or does an end-user control rely on this system's output (IPE)?

No. Key automated controls, such as a three-way match or a segregation of duties (SOD) check, are executed within the ERP, not in AD. Furthermore, the IPE for a user access review is typically generated from the target application (e.g., "a list of all users with access to the ERP"), not directly from the central AD system.



Do downstream key controls depend on the completeness and accuracy of data that comes from this system?

Yes. The effectiveness of every single access-based control in the downstream ERP system (e.g., only managers can approve invoices over $10,000) is entirely dependent on the accuracy of the user's identity authenticated by AD. If AD incorrectly identifies a user, the ERP's control becomes useless because it will apply rules to the wrong person.

Could improper privileged access or a defective change to this system allow someone to bypass detective controls and enable a material misstatement?

Yes. This is the most significant risk. A malicious administrator with privileged access to AD could assign a finance director's permissions to their own account. They could then log in to the ERP as the finance director and approve fraudulent payments, completely bypassing the ERP's controls. The ERP would be operating perfectly, but it would be acting on compromised identity information, leading directly to a potential misstatement.

When to scope out a system

The main idea

The core purpose of “scoping out” is to efficiently focus audit resources by concentrating efforts on systems that materially impact the accuracy of financial statements rather than testing every system in the company.

There is no one-size-fits-all answer. The decision to scope out a system is not based on a simple checklist; it is a professional judgment that depends on the unique circumstances of each system and process.

Key reasons to scope out a system:

- It’s not financially relevant: The system has no connection to a key financial process and does not handle data that could lead to a material misstatement in the financial reports.
- It’s a “pass-through” system: The system acts like a pipe, simply passing data from one place to another without modifying it. In this case, the audit focus should be on the source and destination systems, not the pipe itself.
- The risk is covered elsewhere: The risks associated with the system are already effectively managed by another layer of IT architecture.

The ‘manual control’ trap

The old way of thinking: It was once common to argue that a system could be scoped out because a person manually reviewed a report from that system. The logic was that the human check was the real control.

The current audit guidance: If the manual control relies on data generated by the system, then the integrity of that system is still critical. Auditors will often insist that the system remains in scope for IT general controls unless the manual control is validated against independent, original source documents (e.g., paper invoices), which is becoming less common.

Scoping out: A practical walk-through of the litmus test

The scenario (pass-through system): A company uses a middleware application (IBM Sterling Connect: Direct) to automatically transfer large, encrypted daily sales files from its regional point-of-sale (POS) server to the central ERP system for processing.

Does the system create, transform, aggregate, or route data that is (directly or indirectly) recorded to the GL?

No. While the system routes the data package, it acts as a simple conduit and does not create, transform, or aggregate the financial data itself. Its function is purely logistical transport; the substantive financial touchpoint risk lies with the source and destination (ERP) systems.

Is any key control executed in this system, or does an end-user control rely on this system's output (IPE)?

No. Key SOX controls (such as data validation or approvals) are performed in the source or destination systems. The middleware's operational logs (e.g., success/failure reports) are not used as IPE for any key financial reconciliation or review control.

Do downstream key controls (such as reconciliations, management reviews, or exception monitoring) depend on the completeness and accuracy of data that comes from this system?

No. In fact, the opposite is true. The key downstream control—a reconciliation in the ERP to confirm all sales files were received completely and accurately—is designed to independently verify the outcome of the transfer process. This control is meant to detect a failure in the middleware, not rely on its data.

Could improper privileged access or a defective change to this system allow someone to bypass detective controls and enable a material misstatement?

No. The control architecture is designed to mitigate this risk elsewhere. Any data manipulation or transmission failure within the middleware would be detected by robust detective controls in the destination ERP system (e.g., file hash validations or record count reconciliations against the source). The risk is effectively covered by another in-scope layer.

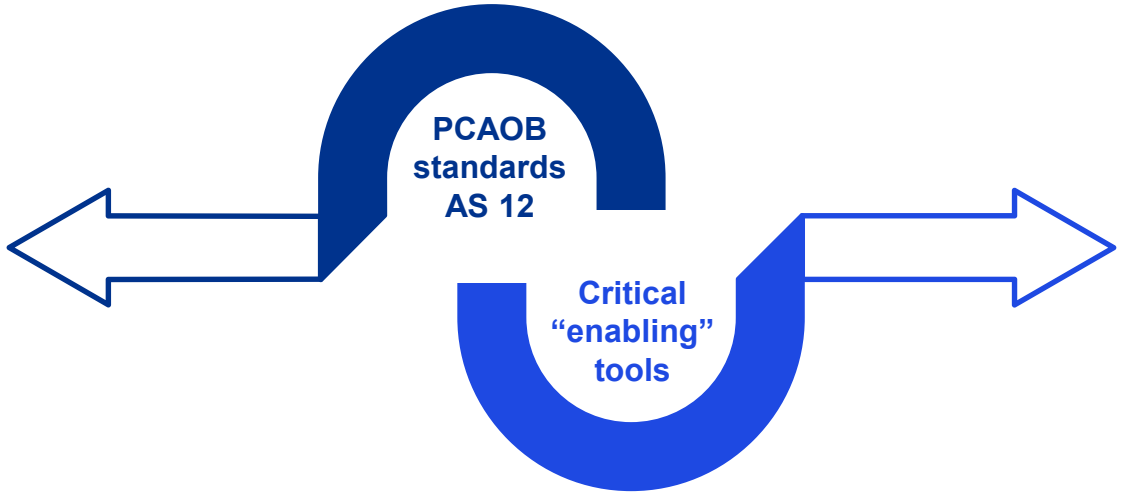
SOX scoping: The standard and the new reality

The foundation: the PCAOB standard (AS 12)

Our approach is anchored in the official standard, which dictates that a system is in scope if it:

- Processes financial transactions (initiates, records, processes, or reports them), or
- Is relied upon by key manual controls (for approvals, reviews, reconciliations, etc.).

Our scoping philosophy is built on two core pillars: the foundational guidance from the PCAOB and the practical reality of modern IT environments.



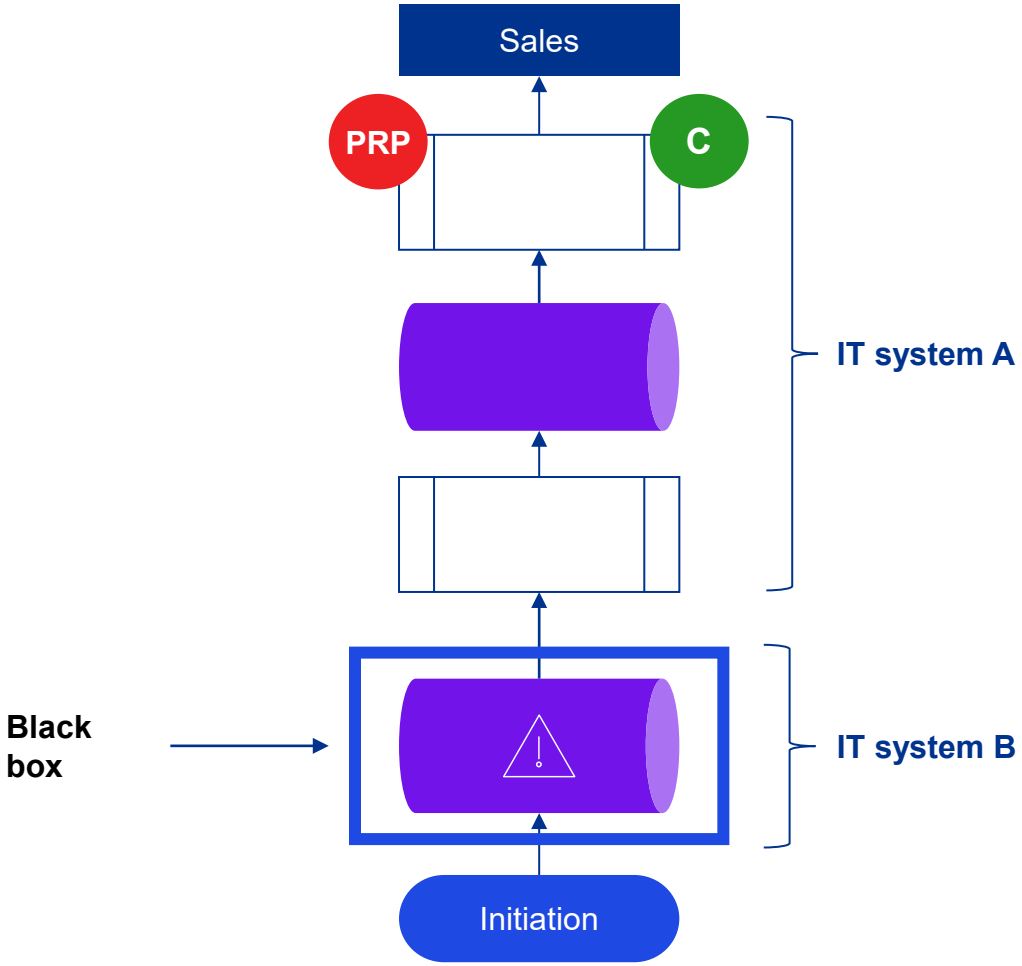
The main point

A complete SOX IT scope includes not only core financial applications but also the critical enabling tools that are relied upon to secure them.

The new reality: critical "enabling" tools

A system is in scope if it is used to initiate, record, process, or report transactions that affect significant financial accounts.

Common challenge example

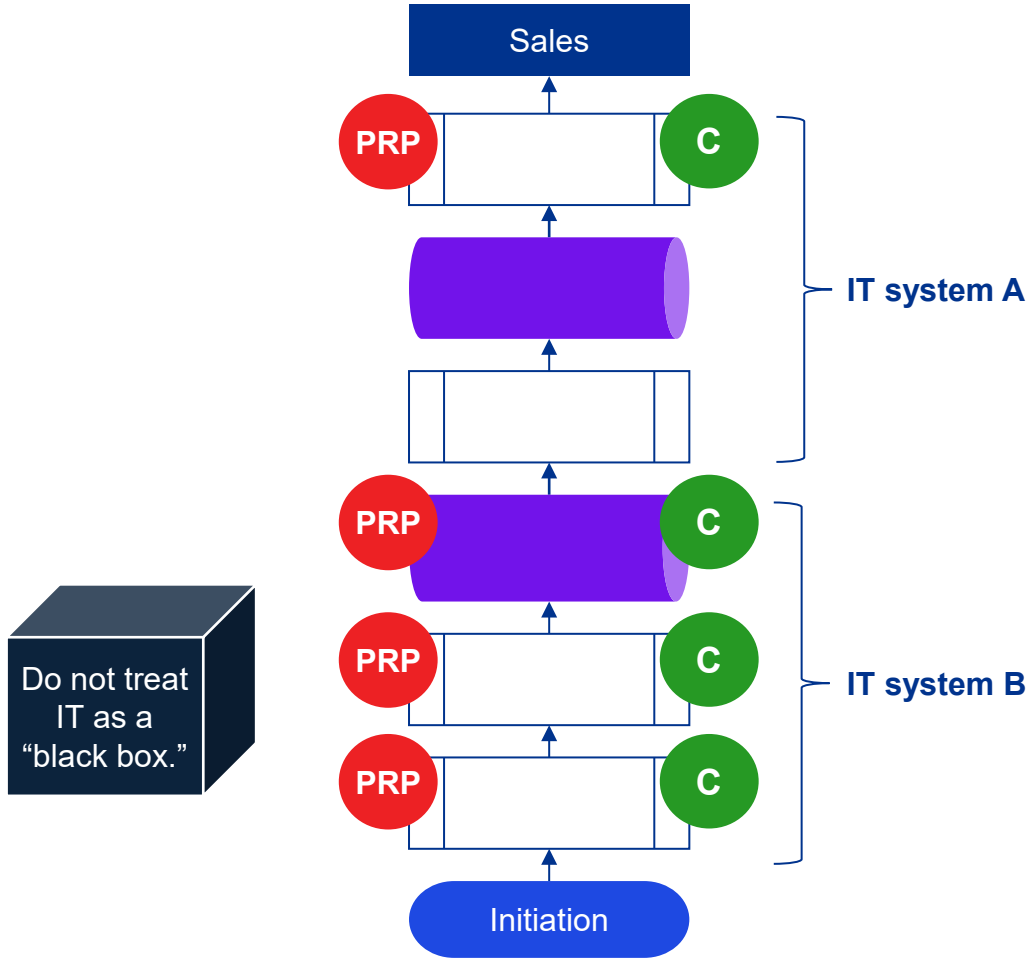


Challenge

Failed to fully understand the process and treated the IT system B as a “black box,” including:

- Not understanding the source of information, including the initiation point of RDEs (e.g., price)
- Not assessing risks related to integrity of data within system B
- Not identifying and testing automated controls and the related GITCs relevant to system B

Sample solution



Solution

- Encourage better communication between accounting, finance, SOX, and IT professionals.
- Involve IT professionals in walk-throughs and documentation of the process understanding, including:
  - Preparing a system overview diagram
  - Identifying all reports and data elements
  - Designing and implementing automated controls to address PRPs
  - Designing and implementing relevant GITCs to address the risks arising from IT.

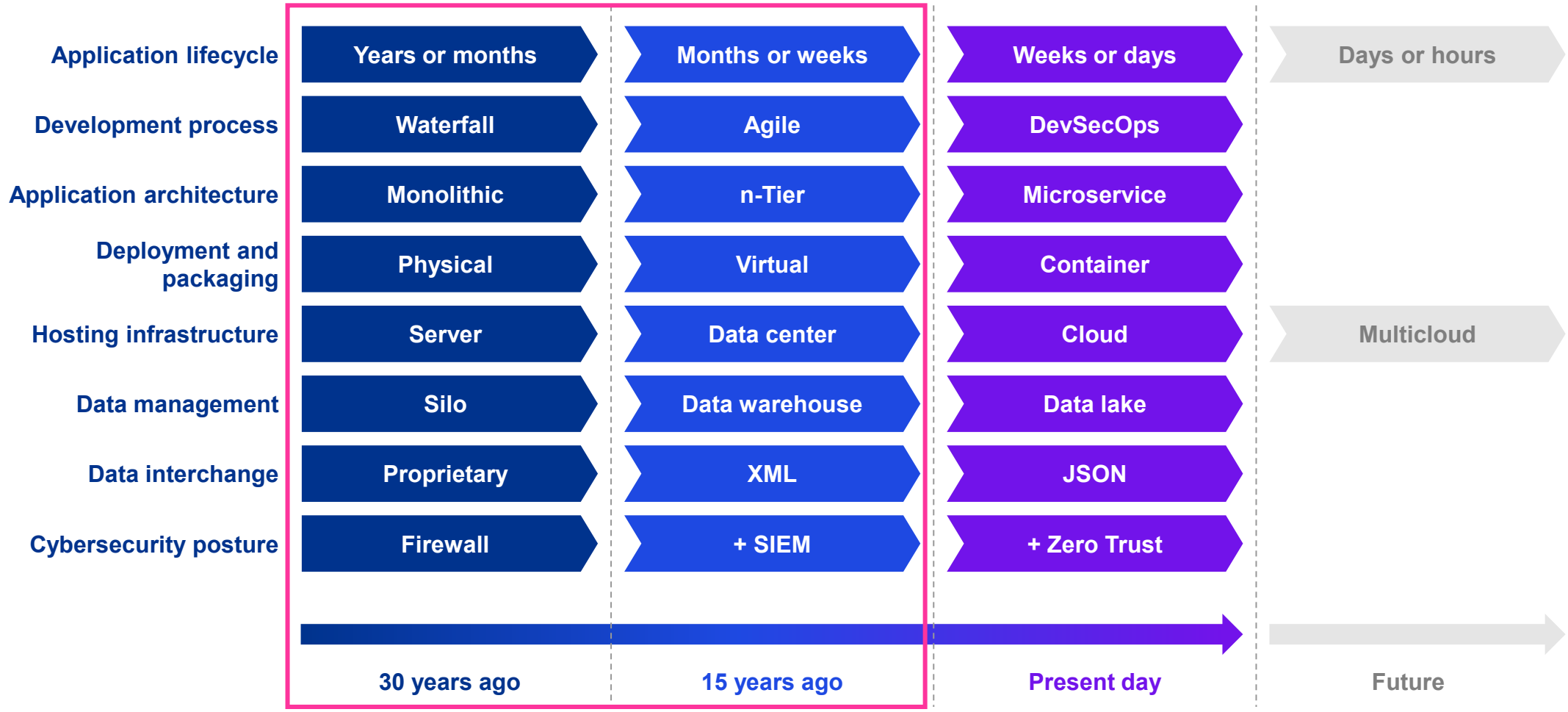
Scoping of IT layers

From owning stacks... to managing services

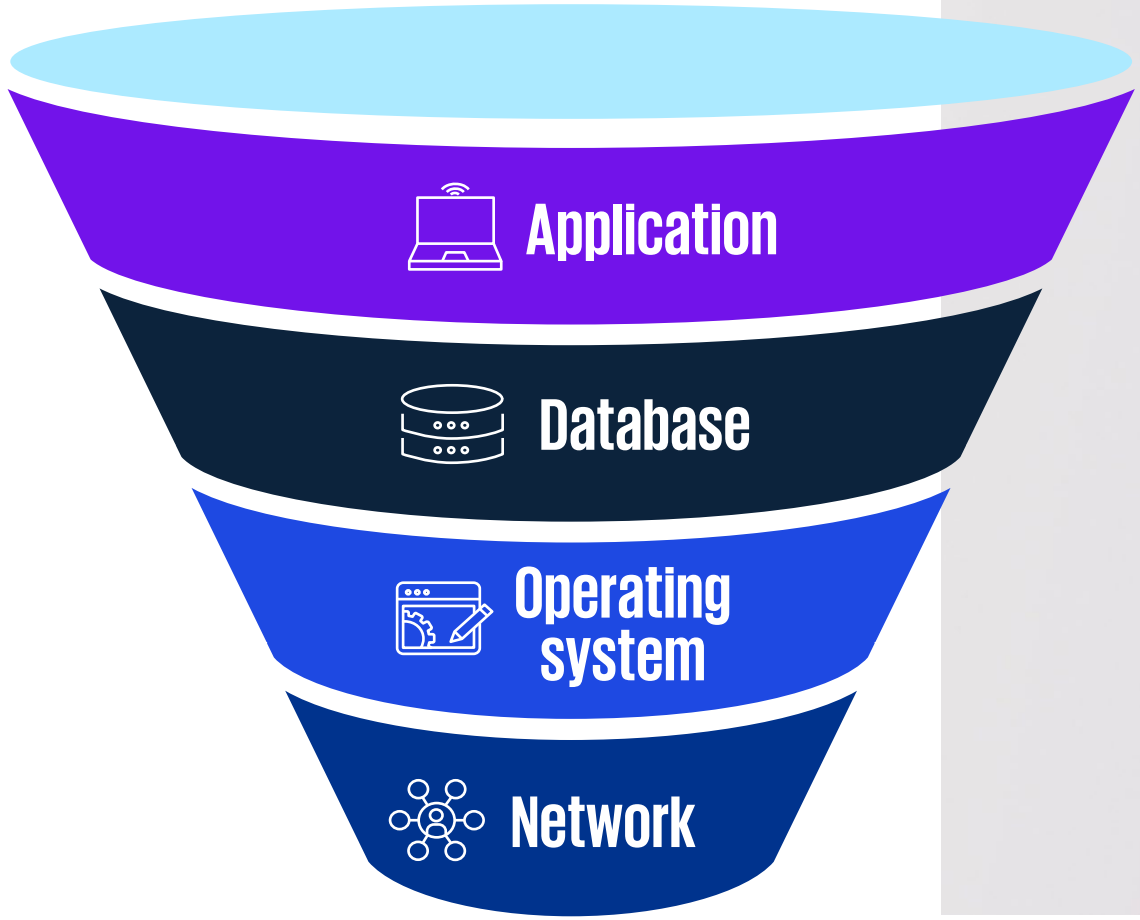
The audit approach is evolving. We are shifting from scoping the technology we own and manage to a model based on the cloud service we use. This is governed by the shared responsibility model, which divides security responsibility between our organization and the cloud service provider.

Technology layer	Traditional on premise	Infrastructure as a service	Platform as a service	SaaS
Application	Management	Management	Management	Vendor
Data	Management	Management	Management	Management
Middleware	Management	Management	Vendor	Vendor
Operating system	Management	Management	Vendor	Vendor
Virtualization	Management	Vendor	Vendor	Vendor
Servers	Management	Vendor	Vendor	Vendor
Storage	Management	Vendor	Vendor	Vendor
Networking	Management	Vendor	Vendor	Vendor

Technology has drastically evolved across all dataflow layers



Traditional/Historical dataflow architecture



Complexity of modern dataflow architecture

Data origins	Transformations		Consumption
Sources	**Ingestion**	**Storage**	**Access**
Systems of record containing business data	Applications used to connect and extract both batch and streaming data from source systems using automated pipelines	Data layer storage for natural and raw formats, consisting of structured and unstructured files, including cloud-native platforms	Controlled access to establish communication and information exchange between applications, as well as internal and external consumption, including microservices and APIs
Additional unstructured and structured data sources	**Data transformation and integration**		**Presentation layer**
	Industry-standard frameworks to convert, cleanse, and structure data into a common data model format for improved analytics and analysis		Data consumption via operational reporting, visual analytics, and application tools to support multiple personas
AI/ML, analysis, and analytics	Scalable, predictive modeling tools to enable enhanced and accelerated AI/ML models		
Data quality and metadata	Quality, observability, accuracy, completeness, lineage, and reliability		
Governance	Standard industry software to manage the accessibility, availability, usability, consistency, and trustworthiness of data for consumer usage		
Platform security	Modern technologies for protecting usability and integrity of an organization's infrastructure		

Scoping change management tools in a DevOps world

This table illustrates a modern DevOps toolset, which is an ecosystem used to move application code from development to a production environment. From a SOX perspective, we need to evaluate which tools should be scoped in for ITGCs. We can no longer look at approval tickets in isolation; we must also understand the tools involved in developing and deploying changes.

Plan	**Platforms**
Develop	**Collaboration**
Build	**Security**
Test	
Release	
Deploy	
Run/Operate	
Monitor	

Conclusion: The new reality of IT scoping

As technology evolves from simple stacks to complex ecosystems, understanding how each layer and dataflow impacts your SOX program is more critical than ever. The fundamental principles of risk and control remain, but how and where we apply them has fundamentally changed.

From the on-premise stack

Auditing a simple, linear technology stack (application, DB, OS, and network) that lived within a well-defined company perimeter

To the modern distributed ecosystem

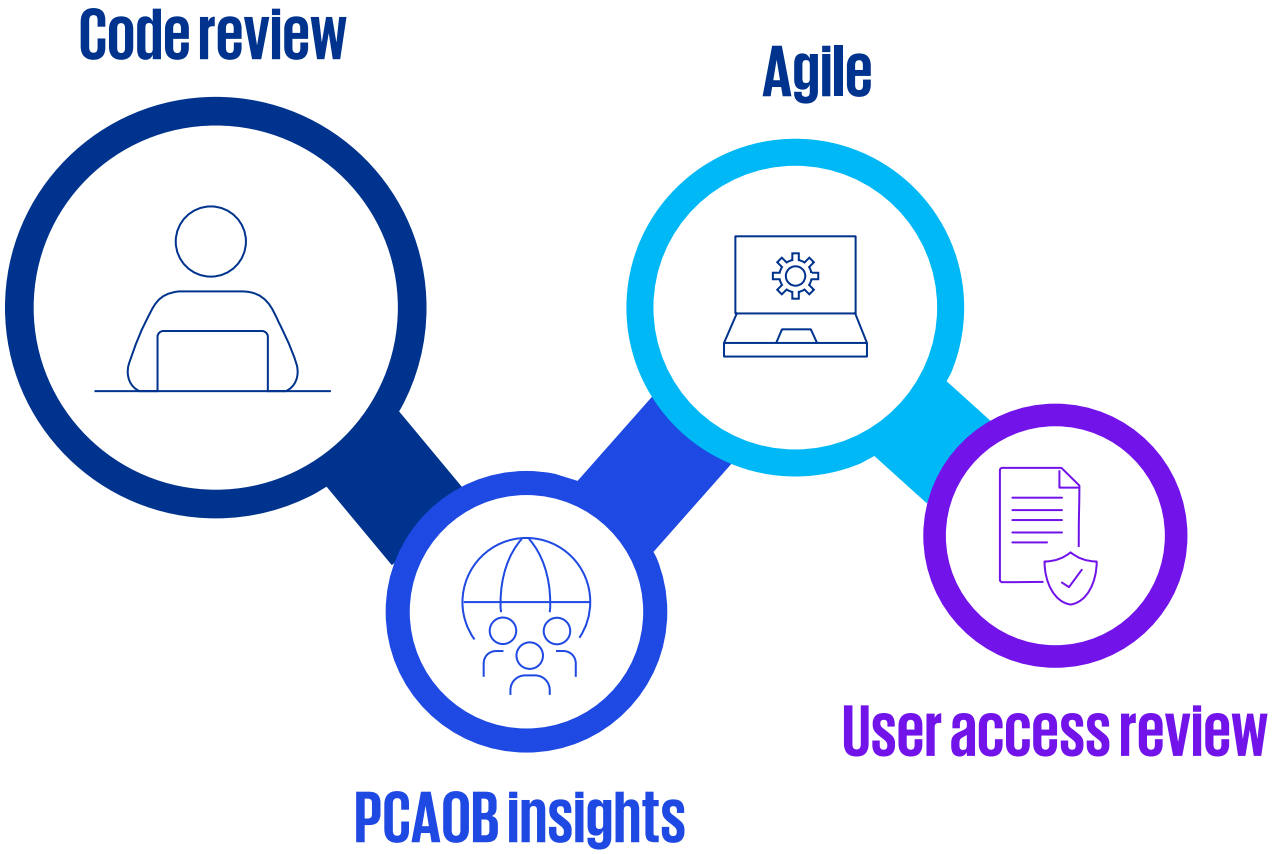
Analyzing a complex ecosystem that requires us to ask critical new questions:

- Which technology layers are our responsibility?
- What is covered by the vendor's SOC report?
- What is the data's end-to-end journey?
- Was the data transformed along the way?

Effective SOX scoping is no longer a simple IT checklist; it demands a deep, integrated understanding of how data moves through your modern technology landscape. Navigating this complexity to build a complete and defensible scope is precisely how internal controls teams partner with you to protect and add value to your business.

ICFR Hot Topics

Overview



Placeholder for text

Code review

From a SOX external audit perspective, there isn't a blanket requirement to "document and explain all code."

What PCAOB/AICPA standards require is sufficient, appropriate evidence that **(1) the IT-dependent controls** (e.g., configurable ITACs), **(2) interfaces**, and **(3) key reports** are designed and operating effectively and that any information produced by the company (IPE) used in the audit is complete, accurate, and sufficiently precise.

What the standards require (at a glance)

Integrated audit objective (SOX/ICFR): Auditors must obtain enough evidence to provide an opinion on the effectiveness of ICFR, using a top-down, risk-based approach (design and operating effectiveness). This often includes IT-dependent controls because they affect key assertions.

Audit evidence quality: Auditors must ensure evidence is sufficient (quantity) and appropriate (relevance and reliability). If evidence is system generated (reports, extracts, and logs), auditors must test its accuracy, completeness, and precision or test controls over those attributes.

Documentation: Auditors must prepare workpapers that enable an experienced auditor to understand the procedures performed, evidence obtained, and conclusions reached. This may include preserving report parameters, logic references, configurations inspected, or excerpts of code/queries if those are what support the conclusion.

(1) Configurable IT application controls

Goal

Conclude that the configuration enforces the intended control objective consistently.

Typical external audit procedures

Design evaluation: Auditors confirm that a control is designed correctly by inspecting its specific configuration values (e.g., tolerance limits) and verifying who is authorized to make changes.

Operating effectiveness: They test if the control worked over time by either reperforming a sample transaction or reviewing system evidence, such as logs, that shows it operated as intended.

Code review (when necessary): If a control is based on custom logic rather than a simple setting, auditors may need to inspect the underlying code or reperform the logic with independent data to get reliable evidence of how it functions.

Documentation to expect

Point-in-time **configured snapshots** (screenshots/export), **evidence of change approvals**, and **period-of-use evidence** (e.g., logs and exception reports); if custom logic is material to the control, include the specific section of the rule/SQL or a tested reperformance with results

(2) Interfaces (system to system)

Goal

Conclude that data transferred is complete, accurate, and timely and that failures are detected and addressed.

Typical external audit procedures

Design evaluation: Auditors understand the interface's design by reviewing its dataflow, mapping, and error-monitoring controls (e.g., control totals), along with the supporting ITGCs.

Operating effectiveness: They test if the interface worked correctly by inspecting execution logs and reconciliations and by reperforming a source-to-target data tie-out for a sample run.

Code review (when necessary): If the interface uses custom transformation logic (such as an ETL script), auditors may either inspect the code or independently reperform the transformation to validate the logic that was applied.

Documentation to expect

Dataflow diagram or narrative, **control descriptions, sample run evidence** (pre-/postcounts and control totals), and **exception resolution**

Include mapping/logic excerpts or reperformance workpapers when they are the basis for your conclusion.

(3) Key reports (used in controls or as audit evidence)

Under PCAOB and AICPA guidance, key reports are IPE. Auditors must either:

a) Test the report's logic/parameters directly (e.g., inspect SQL, report definition, filters, and joins), or

b) Test effective controls over the report's accuracy/completeness, plus obtain evidence of precision and capture the exact parameters/version used.

What "good" IPE testing evidence looks like

Source tieback: Proof that the report pulls from the intended system/source

Logic evidence: the report definition/SQL or spec sufficient to understand how the fields and population are derived (or a successful, independently executed reperformance)

Parameters/Filters: date range, entity, status codes, user selections, and exact values used

Failures in foundational ITGCs

Ineffective change management controls

Industry: Industrial manufacturing

The finding: The audit team did not properly test the company's change management controls for their main ERP system.

The breakdown (what the auditors missed)

- They did not verify whether business approvals were obtained *before* changes were implemented.
- They failed to check whether developers were improperly testing their own changes.
- They relied only on a manager's explanation of the review process instead of inspecting the actual evidence of the review.

Key takeaway: Relying on inquiry alone is a recurring audit failure; key ITGC processes such as change approvals must be validated with direct evidence.

Ignoring automated controls in DevOps

Industry: Insurance

The finding: The team tested a *manual* SOD process instead of testing the *automated* SOD workflow controls built directly into the Azure DevOps tool.

The breakdown (what the auditors missed)

- They did not evaluate the actual workflow configurations within the Azure DevOps tool to see who could approve what.
- Because they didn't test the automated control, their testing of user access lists was incomplete.
- They formed a conclusion based on a manual process that was not the primary control.

Key takeaway: As companies automate, auditors must test the configured, automated controls, not just the legacy manual processes that may surround them.

Flaws in scoping and control precision

Inadequate testing of cryptoasset controls

Industry: Financial services

The finding: The team did not sufficiently test the controls for safeguarding cryptoassets held in cold storage.

The breakdown (what the auditors missed)

- They failed to identify and test controls over the mobile device that held a private key share.
- They did not test the completeness and accuracy of reports used to review who had access to the secure space.
- They failed to assess if the review control was precise enough to actually detect unauthorized access.

Key takeaway: For novel or complex assets such as crypto, auditors must go beyond standard checks and deeply analyze the specific, unique risks (such as device security and control precision).

Improper "homogeneity" assumptions

Industry: Industrial manufacturing

The finding: The team incorrectly concluded that controls were the same ("homogeneous") across 47 different subsidiaries that were using approximately 30 different IT systems.

The breakdown (what the auditors missed)

- Their testing only covered 11 of the approximately 30 IT systems.
- They used this limited sample to improperly extrapolate their conclusion to the entire group.
- This left 20 IT systems completely untested, despite being part of the "homogeneous" group.

Key takeaway: Assuming controls are the same across different locations or systems is a significant risk; homogeneity must be rigorously proven before a sampling approach can be relied upon.

Failures in testing system data (IPE)

Overreliance on third-party data

Industry: Media and telecommunications

The finding: The team failed to test the reliability of a critical viewership report that was produced by an outsourced, third-party system.

The breakdown (what the auditors missed)

- The SOC report for the third-party vendor did not cover the specific reporting tool being used.
- They did not test the company's own controls for "cleansing" the data after receiving it from the vendor.
- No ITGCs over the reporting tool itself were identified or tested.

Key takeaway: A vendor's SOC report is not a blanket solution; you must ensure it covers the specific systems and services you rely on.

Using an incomplete data population

Industry: Life sciences

The finding: The team used an incomplete set of logs to conclude that no unauthorized changes were made to journal entry workflows in NetSuite.

The breakdown (what the auditors missed)

- The logs they inspected only captured the activity of IT personnel.
- The logs did not capture the activity of the company's controller who also had administrative rights to make configuration changes.
- They concluded no changes were made without ever testing a complete population of all possible changes.

Key takeaway: When testing controls over system configurations, you must ensure the population of changes or logs being reviewed is complete and includes all users with privileged access.

Failures in testing system data (IPE) (continued)

Failure to test data from source to report

Industry: Insurance

The finding: The team tested manual reconciliation controls but failed to gain assurance that the underlying data was complete and accurate from the source system.

The breakdown (what the auditors missed)

- They did not assess whether reconciliation controls were precise enough to ensure all RDEs were transferred accurately.
- For a separate control, they tested a report reconciliation but never tested how management ensures the data was completely and accurately recorded in the *source system* in the first place.

Key takeaway: A reconciliation control is only effective if the source data feeding it is reliable and the reconciliation itself is precise enough to catch meaningful errors.

Incomplete testing of high-risk valuation data

Industry: Banking and finance

The finding: The team did not sufficiently test the completeness and accuracy of key data elements used in a high-risk fair value model for residential loans.

The breakdown (what the auditors missed)

- The valuation specialists used several data elements that were not supported by either controls testing or substantive audit procedures.
- The reconciliation control test they did was incomplete, as it failed to include several critical data fields (e.g., delinquent interest and remaining loan term).

Key takeaway: For high-risk estimates, every critical data element used in the calculation must be subject to either controls testing or direct substantive testing.

Failures in testing system data (IPE) (continued)

Failure to validate third-party vendor data

Industry: Banking and finance

The finding: The team tested the tie-out of a footnote disclosure to a report from a third-party vendor but failed to test the accuracy of the third-party report itself.

The breakdown (what the auditors missed)

- Their testing focused only on the mechanical process of tying the footnote numbers to the vendor report's totals.
- They did not identify or test any controls over the accuracy and completeness of the underlying report provided by the third party.

Key takeaway: Tying out to a report is not sufficient audit evidence; you must also have evidence that the underlying report itself is complete and accurate, especially when it originates from a third party.

Failure to test "input risk" of critical data

Industry: Technology

The finding: The team relied on multiple RDEs from a critical homegrown revenue application but failed to test the "input risk" of the key data elements within them.

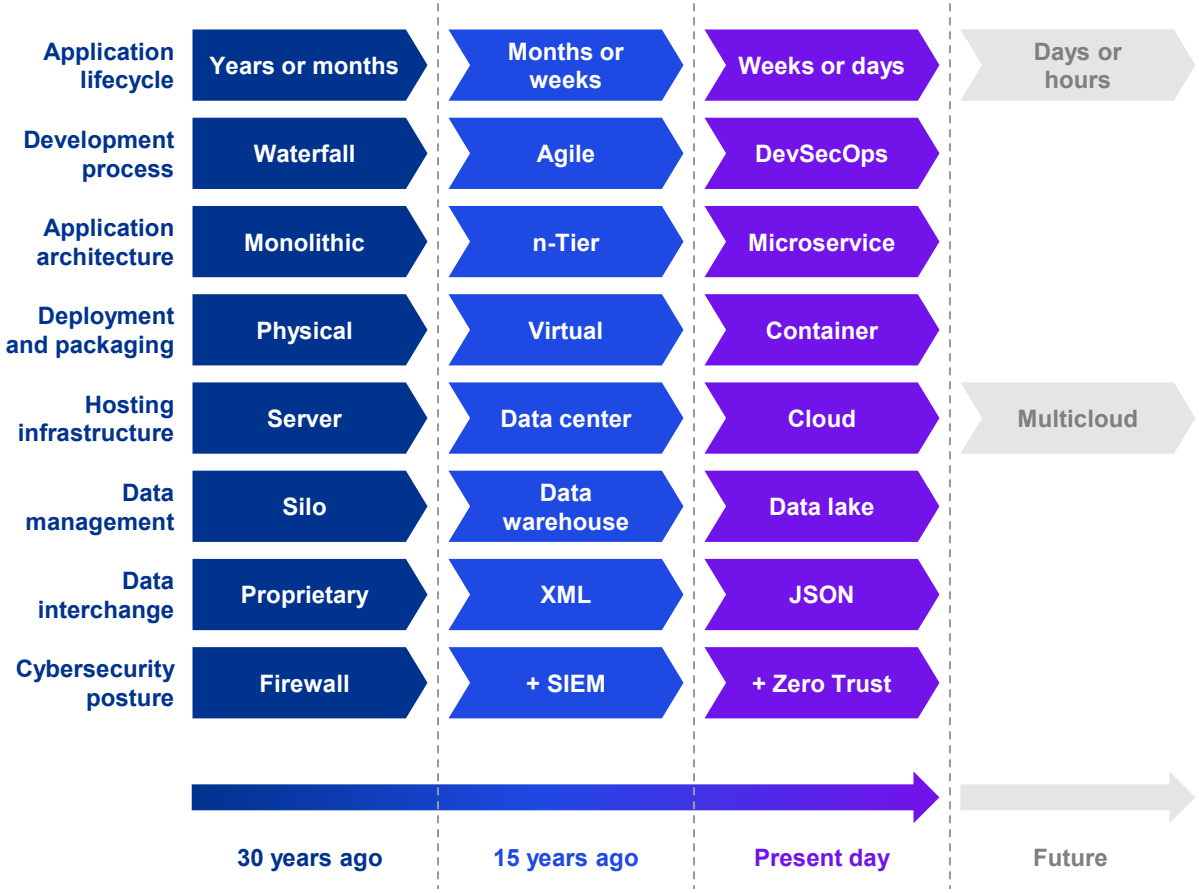
The breakdown (what the auditors missed)

- For key revenue controls, they used RDEs without testing the reliability of the "state" and "created-at" fields.

Key takeaway: When using an RDE, auditors must test not only the report's validity but also the "input risk" of the critical data elements contained within it.

Agile SDLC risks

Evolution of software development



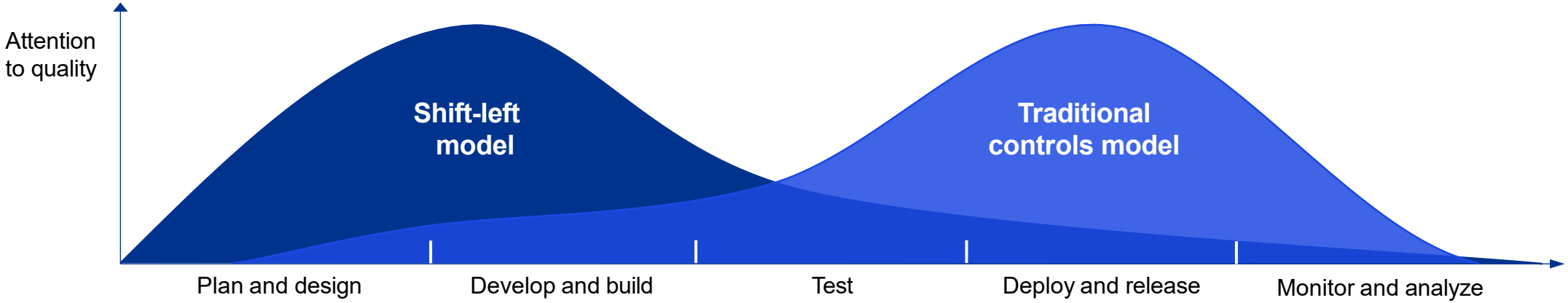
Source: Department of Defense, Office of Prepublication and Security Review, May 12, 2021.

Software development best practices are constantly evolving as new ideas, frameworks, capabilities, and radical innovations become available.

Over time, we witness technological shifts that relegate what was once state of the art to be described as legacy or deprecated.

Software is no different, and the graphic depicts the broad trends over the last 30 years. Different programs and application teams may be more advanced in one aspect and lagging in another.

Shift left



DevSecOps-enabled model

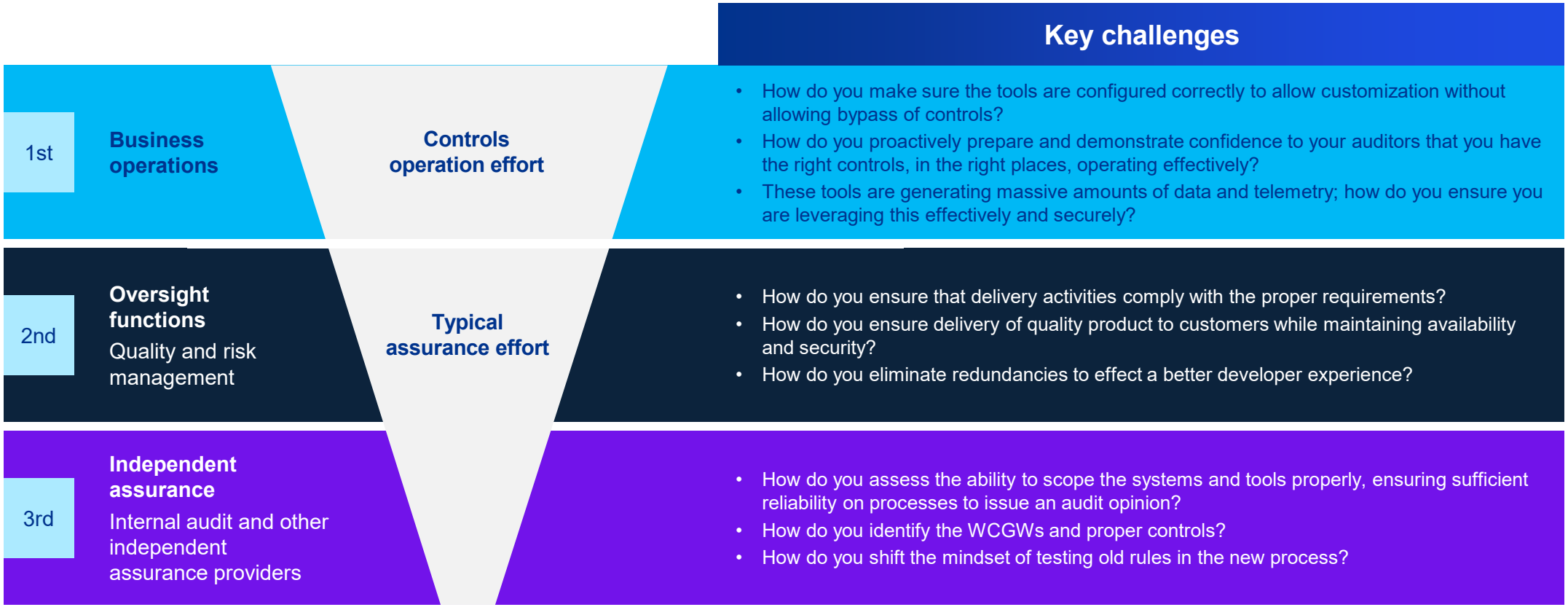
- Leverages automated CI/CD processes, resulting in a low number of manual processes and controls
- Changes are reviewed at the unit level for appropriateness and tested in a production-mirrored environment throughout the lifecycle, resulting in greater detail review.
- Changes that require rework are able to be actioned quickly, increasing speed to production.
- Less manual controls resulting in less time and effort spent on both control performance and control testing
- When control departures/overrides are identified, teams are notified real time, and escalation/resolution takes at the unit-level place before production impact.

Traditional change management model

- High number of manual touchpoints, resulting in increased control portfolio
- Change advisory boards review hundreds of changes in a release as a stage gate before approval at a higher level, not at the unit level, and meet infrequently, slowing down speed to production.
- Change denials or rework requests involve large time investments.
- Manual controls are recurring in nature, which requires dozens of samples for the testing of each manual control.
- Limited visibility into the state/compliance of a change until it reaches the change advisory board, and increased effort to tie to work items/code changes

Challenges presented across organizations

While these new techniques and tools provide many benefits to organizations, they also present large-scale risks across the three lines of defense.



Risks of using agile and DevOps methodologies

Lack of documentation

Inconsistent application of principles driven by individual experience and/or knowledge

High levels of autonomy across teams and business units

Inconsistent approaches to meeting control objectives increase the risk of objectives not being met.

Dependencies on “soft” controls (i.e., team skills, knowledge, and communication)

Soft controls may lead to compliance challenges.

Continuous changes in design

Design requirements may change over the course of product development without revisiting security or control requirements.

Scaling requires careful management

Large, cross-functional teams and complex solutions can cause additional work, not less.

The balancing act of risk and controls

Agile and DevOps methodologies introduce new risks into the control environment due to the high-speed, high-volume nature of change.

User access reviews

User access reviews – Why do they keep being an issue?

Completeness and accuracy

- How will we evaluate the accuracy and completeness of the list being reviewed with respect to users, departments, locations, etc.?
- Are there adequate C&A procedures over all listings, including shared account listings?
- Is there precision in the review of shared accounts, and who has access to the accounts?
- Was the precision of the review performed at user role and permission level to determine appropriateness?

Timeliness

- Is there a defined timeline for the completion of review, and are there escalation procedures when the review is not completed within the defined timeline?

Reviewer competency

- Does management and the control operators understand the rights and privileges attached to the role?
- Are control operators being trained on the overall control risk?
- Who is authorized to perform the access reviews, and how are all reviews coordinated?

Lack of SOD as part of the review

- Who reviews the access of managers performing the reviews?

Root cause/impact analysis

- If inappropriate access is flagged, then how does management assess the impact of the inappropriate access (i.e., what activities were performed by that user account)?

A sample two-pronged solution for effective user access reviews

Implement an automated identity governance platform

A strategic solution that leverages technology to automate manual tasks for greater reliability and efficiency

- Automates data integrity: directly connects to source systems to generate complete and accurate user lists with a full audit trail, eliminating the need for manual IPE validation
- Enforces process by design: Automated workflows enforce deadlines with reminders and escalations while systemically preventing self-approval to help ensure SOD.
- Integrates impact analysis: automatically creates tickets for both immediate access revocation and formal impact analysis when issues are flagged, helping ensure crucial follow-up

Formalize the process with an evidence-based checklist

An immediate, tactical approach that uses a formal checklist to enforce diligence and create a strong audit trail, ideal for nonautomated environments

- Mandate as official evidence: The checklist becomes the required, official evidence for every review; its submission is necessary for completion.
- Prove diligent review: forces reviewers to formally attest to performing critical steps, such as checking for terminated users, SOD conflicts, and generic accounts and verifying specific permission levels
- Establish clear accountability: Formal sign-offs create clear accountability for both preparers and reviewers, which discourages “rubber-stamping” and reinforces ownership of the control.

Persistent audit findings in user access reviews stem from weaknesses in manual processes and inconsistent oversight. A leading-practice approach directly targets these issues through a dual approach: leveraging technology for automation and data integrity while formalizing the human element with a structured, evidence-based review process.

Upcoming *Future of SOX* webcast

Revolutionizing Internal Controls | April 30, 2026 | 11:00AM-12:30PM ET

In the latest installment of the Future of SOX webcast series, Sue King and Steve Estes, KPMG US SOX Solution Leaders, will explore how evolving external audit tools are influencing internal control practices and driving the need for modernization across all three lines of defense. Participants will examine how emerging technologies, including AI and Agentic AI, can strengthen control environments through continuous monitoring, improved efficiency, and smarter risk mitigation. Attendees will walk away with practical strategies to leverage technology responsibly and position internal audit for the future of assurance.

Scan the QR code below to register.

Thank you

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

Learn about us: **kpmg.com**