Current IT Topics: IPE And Change Management

IIA St. Louis Chapter – December 16, 2020
Don Mapes, CIA, CISA, CSX, CRMA
What We’ll Cover Today

- Information Provided by Entity (IPE): What is it and what does it mean to the Business/Management?
- Change Management: For applications and databases
Each company, auditor, and partner are unique to their tolerance for risk. What works for one may not work for all.

This does not constitute legal, accounting, or audit advice. Weigh the advice given against your own experiences, and discuss with your paid advisors before implementing any of this.

My interpretations and opinions are my own, and not necessarily representative of my employer.

I am not a CPA. I do have a few other relevant designations.

I have been involved in auditing or designing (and for parts of my career outside of Audit, operating) Internal Controls for over 24 years, from a company perspective.
My Background

- Manager, RubinBrown Business Advisory Services
- President of Denver IIA Chapter 2001 and 2003, Various other committees
- President of ISACA Denver Chapter 2014-2016, currently VP Education
- BSBA – Accounting; MBA in Technology Management
- Certified Internal Auditor (IIA)
- Certified Information Systems Auditor (ISACA)
- Certified in Risk Management and Assurance (IIA)
- CyberSecurity Fundamentals Certificate (ISACA)
- Over 21 years of Internal Audit experience, 11 years as a Chief Audit Executive
- Spent 2+ years outside of Audit in IT, and a few more years in other operational/IT areas
Auditing Standard (AS) 1001 Responsibilities and Functions of the Independent Auditor, clearly states that

- “the auditor has a responsibility to plan and perform the audit to obtain reasonable assurance about whether the financial statements are free of material misstatement, whether caused by error or fraud. Because of the nature of audit evidence and the characteristics of fraud, the auditor is able to obtain reasonable, but not absolute, assurance that material misstatements are detected.”

- (emphasis mine)
IPE

Learning Objectives:

- Difference in IPE vs. information used in controls (IUC)
- Items to consider when using reports/information to make decisions
- Possible methods to address completeness and accuracy
Different Audience

Information Provided by Entity (IPE)
- Reports, interfaces, etc. generated by the company for the external auditor

Information Used in a Control (IUC)
- Reports, interfaces that management uses within the company to make decisions, evaluate controls, and produce results. And often, to book financials
What’s The Difference?

**IPE**
- For external to place reliance, has to be “sufficient evidence”
- Remember that “reasonable, but not absolute” language

**IUC**
- The Public Company Accounting Oversight Board (PCAOB) oversees external audit firms, not companies
  - Many companies still stick with “reasonable”
  - See the above graphic for the effect on the external auditor
1. Name, frequency & format (e.g., CSV, Excel, PDF, text) of report

2. System(s) used to produce the report & key elements below?
   - Including underlying app/database & report writer/tool used to extract the data

3. What management controls is it used in?

4. Key fields (RDEs = relevant data elements) in the report?
   - RDEs affect a control operator's decisions OR support a key input or assumption
IPE (More Questions)

5. How is the information input/captured in the IT application?
   - “CSR selects a service code in the billing system. The service code generates certain work order type(s) to the technician's hand held device. Technician indicates the work order is completed via the hand held, which interfaces back to the billing system. The closed work order transfers through ODS (operational data store) to XDW, where the key report is sourced from.”

6. Is the System(s) covered by effective ITGCs (processing integrity)?

7. How report is generated (screenshots of the steps), who runs it (scheduled job or on demand)?

8. Report manipulated (e.g., filtered, sorted, etc.) prior to completing the control?
But Wait, There’s More...

How do you know the report (or interface) is complete and accurate?
IPE Completeness

Manual – Ad hoc queries

1. What is done to ensure completeness (including data integrity, extraction, and manipulation)?
   - Tie totals (row counts, subtotals) or sum counts back to the source system
   - Positive/negative test of the system generated report

2. If the control is **manual**, the verification will need to take place for **every occurrence of the control**. Like, each month, quarter week, etc.
Automated – System generated reports

1. Be able to show the last change date of the report (screenshot)

2. Coordinate with IT resources to verify the query code and parameters used to run the report are established to completely & accurately extract the data
   - Tie totals (row counts, subtotals) or sum counts back to the source system
   - Positive/negative test of the system generated report

3. Will only have to show code behind the query one time per year
IPE Accuracy

Expected of Management by the External Auditor

Screenshot from system, tied to the report, for EACH Relevant Data Element

Make sure you include a date (see screenshot below)!
External auditor asked for a three HR listings: Active, New Hires, & Terminated Employees from 10/1/2019 onward

Separately, we requested a listing of all users in the same HR system

In other testing, using a complete listing of all users from the same HR system, the external auditor noted a terminated employee user ID was not on ANY of the initial three reports they received
Veracity Of The Reports?

- Are the reports unreliable???
- This casts doubt on the validity of all reports coming out of the HR system!!!
- How can this be?
Here’s How It Happened

- The parameter used to run the terminated employee report was “>= 10/1/2019”
- The employee in question had a termination date prior to 10/1/2019
  - There’s nothing wrong with any of the reports. The only issue was the parameters used in the first three reports precluded him showing
- Lesson learned: Watch those parameters – better yet, get the entire list of employees (active, inactive) with hire and termination dates
As covered previously, similar to IPE; information (typically reports & interfaces) that management uses in controls, or making decisions, and possibly making financial entries.

How do you know it's complete and accurate?

- Best answer: Tie it back to the source by count, amount, and sample item(s).
- That’s starting to sound like the external auditor’s standard...
- Don’t see/hear this happening outside of an auditor asking the business owner to do so.
How Do You Know The Report Is Complete And Accurate?

- What’s acceptable: Be able to articulate that the report matches expectations, and hasn’t changed
  - Approximately what you expect for size and amounts
  - Has item(s) you expect to see there
  - Does not have items you know should not be there
  - Report is included in IT Change Management
  - Able to have IT show last modified date, and it corroborates your statement that the report hasn’t changed since xx/xx/xxxx

- Not acceptable:
  - “I don’t know”
  - “It’s the same report I’ve used for the past xx years”
Sufficient Evidence

- None
- Reasonable
- Absolute
With respect to **Inventory**:

The firm selected for testing an automated control and certain information technology (“IT”) dependent manual controls over inventory that used data or reports that were derived from the issuer’s inventory system, for which the firm had identified a significant deficiency that was not remediated until the fourth quarter. The firm did not test any instances of the automated control and the controls over the accuracy and completeness of these data and reports subsequent to the remediation of the significant deficiency. (AS 2201.55 and .56)

The firm selected for testing a control over program changes that consisted of the approval, testing, and monitoring of changes made during the year to the issuer’s inventory system. The firm did not identify and test any controls over the completeness of the report that was generated by the inventory system affected by the significant deficiency discussed above and that the firm used to select program changes for testing. (AS 2201.39)
How Do You Know Which Way To Go?

- Are you publically traded and subject to an external SEC Attest-type audit?
- If you are, check with your external auditor:
  - One firm may take the IPE approach
  - Another may accept the IUC – so long as you can support what you assert
    - The external auditor will test to your assertions (see previous slides). It doesn’t change their burden
- If you are not subject to SEC Attest (& SOX), its up to your company’s risk appetite
Wrap-Up In IPE

- Management is free to do what they decide is best in terms of IUC:
  - Controls cost money
  - This is classic risk versus reward
  - Haven’t seen an external auditor note as a deficiency… yet.
- For IPE: If you are publicly traded, you are going to have to satisfy your external auditor
Change Management

Learning Objectives

- Put in place proper change monitoring over applications
- Risk assess direct database access (outside of the application)
- Monitor and filter to focus on changes made outside of the application
Change Management – Why?

- Unauthorized changes can create havoc with systems
  - Introduce weaknesses in security
  - Allow manipulation/misrepresentation of data
  - Cause the system to be unavailable
  - Delayed project implementations
  - A high number of emergency changes to fix
Bad Changes Cost Time And $$

- A large percentage of IT time is spent correcting unauthorized/improperly tested changes.

- "... in 2000... measured software quality for North American teams... ranged from 6 defects per function point down to less than 3 per 100 function points, a range of 200 to 1. The midpoint is approximately 1 defect per 0.6 to 1.0 function points. This implies that it is common for teams to spend more than 90 percent of their effort fixing defects." The author cites an example provided by one of his colleagues of a company that spends 90% of the time fixing their bugs.

  "Software Assessments, Benchmarks, and Best Practices," by Capers Jones
Change Management – Basics

Changes need to follow a disciplined process (usually in a ticketing system)

- Approval to start work
- Changes should be risk assessed, H M L, or similar
- Testing:
  - In a test environment, not Production, unless not possible
  - Approval of testing by business?
- Approval to implement the change
  - Backout strategy – in case the change goes wrong
  - Acceptance by the stakeholder (business)

How much of the above is required depends on the risk involved in the change.
Segregation of Duties

- Programmers should not have access to Production
- If they do, they can (potentially) make changes that bypass the change management process
- Sometimes, it’s unavoidable
- If so, our mitigation strategy is: Monitor changes and trace to tickets
Change Management – Monitoring

- What is the population of changes?
  - Ticketing system?
  - Code Library?
  - Production monitoring?
- Many turn to their ticketing system
  - Anyone see an issue with that method?
Hackers don’t turn in tickets

Nor do people who make accidental mistakes

- Change promoter (or heaven forbid, a programmer) thought they were in test, but made the change in Production

- They are not hiding it deliberately; they don’t even realize they changed Production

Production environments need to be monitored for changes
Applications – How To Monitor

Tools to use

- Version Number (File modified date and size)
- TripWire
- RPM Package Manager (Linux)
- Powershell
- SolarWinds
- Raygun
- And there are many more…
Off-The-Shelf Software (Third-Party Code)

- You do not have the source code
- Version number is usually sufficient to prove no modifications
  - File modified date and size
  - Periodically (quarterly perhaps), compare the version number/file size/modified date to the prior period
- They match, you’re done!
- They don’t match find out who approved and put in the change
Off-The-Shelf Software – Ticket?

- Should there be a ticket for an upgrade or patch?
  - Yes! Patches/version upgrades still should be approved, tested and verified
  - Make sure it operates properly in your environment
- Example: Company used ProCount software
  - New version of Java released
  - ProCount stopped working with the new Java version
  - Had to roll back and use older Java version
  - Vulnerable...
Got Code?

- Code Library: Ideally, you have a repository that contains the source code
- And requires check out/in for changes
  - Team Foundation Server
  - Package Manager on Linux
  - Visual Source Safe
  - Etc.
Business Involvement in Changes

- There should be a clear path showing the Business side requested the changes to be made applications
  - Could be IT if the application is only used by IT
- The change needs to be made in a Dev/Test area on a clone of Production data
- Access to this given to Business (or perhaps QA) for testing
  - Save supporting test results!
- Once the Business/QA approves, then the change can be made in Production
- Requestor (Business) should be giving final sign off
  - “I asked you to do this, and I can see that I got what I asked for”
What’s Your Exit Strategy?

- What if... it worked great in Dev/Test
- When you put it in Production, things didn’t go quite right
- Need a Roll Back or Backout Plan
- Know how you can quickly undo an errant change
Applications – Company Code

- If you have access to the source code, you’ll need something better than version number
- This is where other solutions come into play
- Most work as follows:
  - The Change Monitoring software “hashes” the files that comprise the application
  - Compares this hash to what is supposed to be there
  - If different, alerts; if same, no alert
  - Good software compiles a query-able list of changes made
What’s A Hash?

- Algorithm that takes file attributes (size, date, name, etc.) and creates a large numeric string
- MD5 is a very common one, SHA256 is another, there are more
- This is typically unique to a file, it’s a million (or more) to one that two dissimilar files could arrive to the same hash
- Many solutions use this; you can even write a PowerShell to compare to the prior day
What Tool Is Right For You?

- PowerShell to compare to the prior day
  - Advantage: It’s free, and it will monitor for changes
  - Disadvantage: It only alerts the day of the change
- RPM Package Manager (Linux) will compare the hash to the approved files in PacMan
  - Advantages: It’s also free, and keeps alerting until you rollback the unauthorized code or approve a new matching version in PacMan
  - Disadvantage: You have to be on Linux
- TripWire, Raygun, SolarWinds,
  - Advantage: All very good
  - Disadvantage: All cost money
Database Agenda

- Why do we need to examine databases
- Approaches
- Risk analysis
- Demonstrating access
- Logging
What’s In A Database?

- Databases house all the transactional detail supporting the reports/interfaces of the application
- Oftentimes, the security within the application (roles, rights, groups) is also stored in the database
- Each database has its own database software; popular ones are:
  - SQL Server
  - Oracle
  - DB2
  - Many others
Database Change Management - Program

- Important to know what database and version you are running, as well as the underlying operating system
  - This may impact security, e.g., SQL Server can utilize Windows Server Active Directory
  - DB2 is integrated with OS390/iSeries/Power8
- Upgrades of the database version should follow normal change management:
  - Make sure the new version runs properly in your environment
  - Make sure your application works correctly with the new version
  - Got customizations?
    - Make sure they are brought over
    - Do they still work?

**Bottom line: Do this in Test before you do it in Production!**
Why Worry About The Database?

It’s just data, right? Nope
- Schemas, views, objects & stored procedures (oh my!)
- Focus on the objects (data) and stored procedures

Direct Data Modifications
- Can be done using a variety of tools, if the UserID has access
  - Change the data; change the results!

Stored Procedures
- Many applications are coupled with Stored Procedures in the Database – these are instructions that operate very much like programs.
  - Change the program; change the results!
Bypass of Application Controls

- Application controls/permissions can be bypassed if one can modify data/stored procedures directly
  - If one can reach the direct modification of the items on the previous slide, your application controls never come into play.
  - Only controls over actions are those enforced by the database itself.

- Controlling user access to the database
  - Most important and common control
  - Very few people should have direct access to the database.
  - Normal users make changes via the application.

- Is enforcing limited access good enough?
  - Depends – risk versus reward
Approaches

- **Prevent:** I’m going to stop you from making any unauthorized changes!
- **Detect:** I’m going to find you if you make any unauthorized changes!
  - And change them back!
- Do one, the other or both?
  - In a perfect world, do both
  - But sometimes, both is not quite so practical
- Risk Analysis:
  - How important is it, and
  - What could go wrong?
Risk Analysis For A Database

- What is it used for?
  - Financial reporting?
  - Payroll?
  - User access reviews?
  - Sales quantities to identify purchasing habits?
  - Health insurance claims?
  - Customer information?
  - Results of the Sunday Football pool?

- What we are trying to answer is, “So what?” And if the database is compromised, what’s the impact on the firm?
  - How important: financially, public relations, lawsuits?
  - What could possibly go wrong?
    - Data changed to distort results
    - Data destroyed to harm operations
    - Data exposed to the public
## Risk Analysis For Direct Database Changes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many people have the ability to change data</td>
<td>Few</td>
<td>Many</td>
</tr>
<tr>
<td>How do they authenticate</td>
<td>SSO, Windows Authentication</td>
<td>Direct Login, unencrypted</td>
</tr>
<tr>
<td>Do they have Dev capabilities</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Volume of direct data mods</td>
<td>Few</td>
<td>Many</td>
</tr>
<tr>
<td>Frequency of data mods</td>
<td>Seldom</td>
<td>Often</td>
</tr>
<tr>
<td>Do the modifications have financial impact</td>
<td>Little</td>
<td>Large</td>
</tr>
<tr>
<td>Saved script versus interactive not saved</td>
<td>Saved script</td>
<td>Interactive</td>
</tr>
</tbody>
</table>
Risk Results

- **Low/Medium Risk**
  - Perhaps I can simply rely upon demonstrating limited access
  - Better be able to prove it

- **Medium/High Risk**
  - Time for logging
  - This can be accomplished thru commercial products (Oracle Database Vault, Idera, etc.)
  - Can also be done via triggers

- **OMG, THAT LOG IS GONNA BE HUGE!!!**
Many, many, changes are made. Hopefully, only a few are important to examine

Whether we use a commercial tool or not, the challenge is to filter what we examine

Every time a command is sent to a database, it contains information about what, who, and when it was sent, and what it’s doing

For simplicity’s sake: Select, Update, Delete

So – we start chopping the chaff!
Focus On What Is Important

Our monitoring system needs to examine each command:

- What tool is making the change: If coming from the authorized application – **IGNORE**
- Who/When: no help here, we usually can’t eliminate based on this. It does assist in reconciling to a ticket
- What type of command:
  - Select – **IGNORE**; unless you are concerned about data privacy!
  - Update or delete: **LOG IT**
- What is the target table?
  - May be able to ignore unimportant tables, if you know your data well enough
What’s Left

Now that you have saved the important stuff:

- Match to authorized change tickets
  - Sometimes it will be caused by an upgrade; no excuse for not having a ticket
- Investigate what isn’t authorized; roll it back
  - Find out whodunit
  - Take corrective action!
If you find an inordinate amount of data has to be looked through, it speaks to one of two things:

1. You missed filtering out legitimate transactions
   - If so, consider coding the authorized applications to include an authorized name

2. You have a broken process, in that too much maintenance is happening to data directly
   - Perhaps it’s time to look at features that need to be built into the application

Lastly
Questions?

Don Mapes
RubinBrown

don.mapes@rubinbrown.com
Ph: 303.819.4393