Data Integration & Information Quality

Case Studies Addressing Interface Risk

Andy Zone
Deloitte & Touche
azone@deloitte.com

Mark Houston
Deloitte & Touche
mahouston@deloitte.com

Executive Summary/Abstract: Effective exchange of data between different systems is critical to the operational and reporting needs of most businesses. Organizations face various financial and operational risks when this exchange of data does not work appropriately. Addressing these risks requires an approach that considers multiple layers of an interface risk model: the process layer, the application layer, and the data layer.

Overview

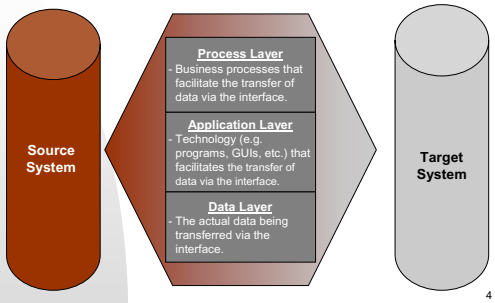
- Interfaces and their impact
- Layers of interface risk
- Considerations for assessing interface risk
- Case studies
- Lessons learned from case studies
- Applying the lessons learned
- Questions

Interfaces and Their Impact

- **What is an Interface?**
  - Simply stated, an interface is any point where data is transferred between systems

- **Why is it important to address interface risk?**
  - *Interface issues can result in significant revenue loss and operational inefficiencies*
    - A major telecommunications provider was losing nearly $1 Million annually when billable call records were not transferring to a billing system due to outdated business rules.
    - For a single city studied, a telecommunications provider was losing approximately $50,000 annually due to undetected, incomplete interface transactions.
    - A state disbursement agency lost corrected address information despite multiple attempts to process updates, resulting in an inability to contact parties responsible for payments.

Layers of Interface Risk: Our Point of View

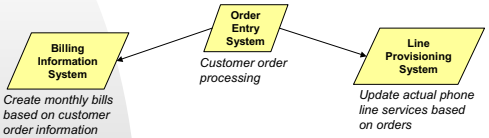


Layers of Interface Risk: Considerations

- **Process Layer**
  - Is data ownership clearly defined?
  - Are thorough change management procedures in place?
  - Are reconciliations performed between systems?
  - Are procedures for initiating interfaces sufficiently documented and communicated?
- **Application Layer**
  - Are the process layer objectives supported with an appropriate configuration?
  - Are translation tables accurate and reviewed periodically?
  - Are overwrite and update rules correct and have they been validated?
  - Have systems dependencies and timing issues been addressed?
- **Data Layer**
  - Is the data in the source and receiving systems processed in the same context?
  - Are data standards and data formats in the source and receiving systems documented and communicated?
  - Is the data in the receiving system, post any translations, consistent with the source system?
  - Is the data accurate, timely, consistent, and complete?

Case Study #1: Revenue Loss

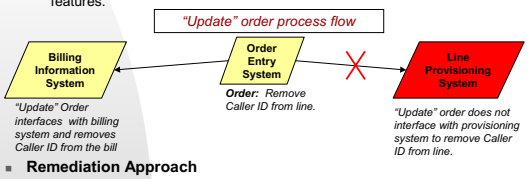
- **Background**
  - Orders for a telecommunications company were being passed to billing and provisioning (the system that changes your phone line service) systems via two different interfaces.



- **Issue**
  - In some cases, the company was under-billing customers. Certain line features (e.g. Caller ID) were remaining active on the line, but not in the billing system, after order processes were complete.

Case Study #1: Revenue Loss

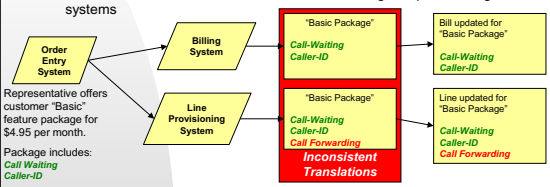
- **Root Cause #1**
  - Customer service agents were using "Update" order types to remove features from a line. Update order routines were intended only for updating customers' personal information (e.g. Name, SSN), and consequently did not interface the provisioning system that controls line features.



- **Remediation Approach**
  - Improve training process for service agents. Instruct agents to use "Change" orders to remove features from lines.
  - Utilize monitoring procedures to track features removed with a "non-Change" order.

Case Study #1: Revenue Loss

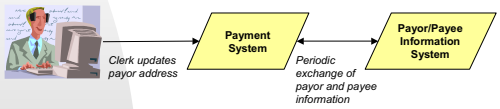
- **Root Cause #2**
  - Translation tables existed in 2 interfaces to process feature packages (e.g. "Base package" includes Caller ID and Call-Waiting for $4.95 per month.)
  - Translation tables were inconsistent between billing and provisioning systems



- **Remediation Approach**
  - Centralize edit processes and assign ownership to translations
  - Utilize monitoring procedures to track mismatches between line and bill

Case Study #2: Address updates lost

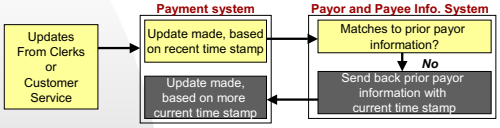
- **Background**
  - A disbursement unit of a state government had 2 systems communicating via interfaces: a payment system and a central payor/payee information system.



- **Issue**
  - Updates to payor information (e.g., addresses), being made in the payment system, were not "sticking" and were being made multiple times, unsuccessfully.

Case Study #2: Root Cause Analysis

- **Root cause #1**
  - Payor information system was returning old addresses with current time stamps to the payment system



- **Root cause #2**
  - The 2 systems spanned organizational boundaries. The owners of the payor information system viewed this system as the "system of record" for payor information.
- **Remediation Approach**
  - Make business decisions based on "system of record". Update documentation around existing applications, and update processes so that address changes are input into the right system.

Lessons Learned: Data Layer

- **Interface problems had a direct impact on the data layer:**
  - Telecommunications Case: Data was inconsistent between billing and provisioning systems
  - Disbursements Case: Data being passed across interfaces & updated in systems was out-of-date, inaccurate, and inconsistent across systems

- **Data Layer**
  - Is the data in the source and receiving systems processed in the same context?
  - Are data standards and data formats in the source and receiving systems documented and communicated?
  - Is the data in the receiving system, post any translations, consistent with the source system?
  - **Is the data accurate, timely, consistent, and complete?**

- **Conclusion: Interface issues impacted data quality.**

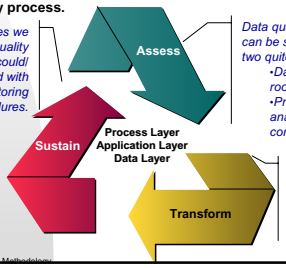
Lessons Learned: Process & Application Layers

- **Root causes of interface problems were found in the application layer:**
  - Telecommunications Case: Translation tables & logic were inconsistent across systems
  - Disbursements Case: Update logic on interfaces (esp. the use of new time stamps on old data) caused unexpected overwrites
- **Root causes of interface problems were also found in the process layer:**
  - Telecommunications Case: Application functionality was used in a manner inconsistent with approved operating procedures
  - Disbursements Case: Updates were not being made in the system of record, because not everyone shared the same view of the system of record
- **Conclusion: In our case studies, the root causes of interface issues (and data quality issues) were found in both the application and process layers.**

Lessons Learned: Where do I find risks in my interfaces?

- **Based on our experience, successfully mitigating data quality risks around interfaces requires a sustained approach to assess and continually monitor risks inherent to all layers of the risk framework. As presented in prior conferences, Deloitte uses a 3 step closed loop data quality process.**

Most interface issues we identify in Data Quality Assessments could/ should be identified with ongoing monitoring procedures.



Data quality and interface risks can be successfully identified with two quite different approaches:
- *Data Analysis, followed by root cause analysis*
- *Process and controls analysis, followed by controls testing*

Before fixing an interface-related data quality problem, understand how the issue relates to all 3 layers of interface risk.

Lessons Learned: Techniques for addressing interface risks

- **Assess**
  - A structured self-assessment with input from end-users through management assists in effectively identify high-risk areas supported by interfaces
  - The documentation and review of an organization's internal control structure (consistent with Sarbanes Oxley requirements) assists in identify deficiencies
- **Transform**
  - Define data ownership and accountability for interfaces
  - Modify interface controls by implementing new automated validations as necessary (e.g. # records updated in system A = # updated in system B). Use of third-party software can enable automation
  - Enhance system controls on user processes (e.g. restrict user access to translation data and interface batches. Ensure access security authorization process requires appropriate approvals.)
  - Define change management processes
- **Sustain**
  - Perform ongoing source to target reconciliations
  - Trend interface data volumes to detect unexpected fluctuations
  - Continually monitor noted control deficiencies and action plans in place to address them (consistent with Sarbanes Oxley requirements)

Questions?