
IT Dependencies Testing

Session Objectives

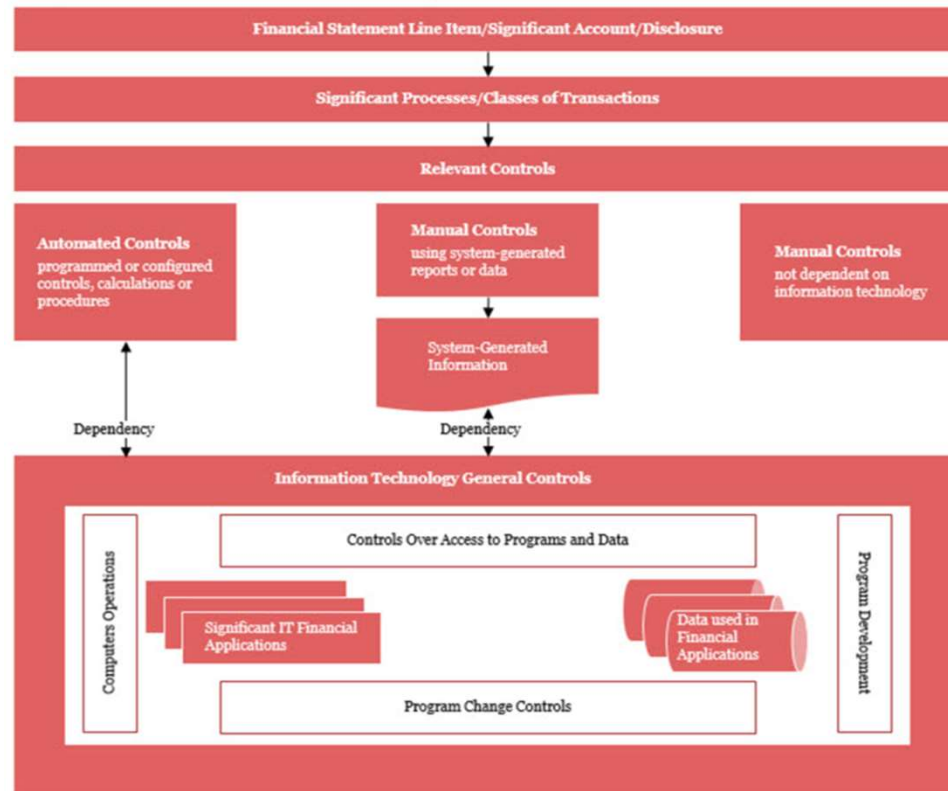
- Discuss common risks and pitfalls for testing IT Dependencies
- Discuss key requirements and considerations to improve quality and consistency of IT Dependencies testing

Agenda

- Types of IT Dependencies
- Testing Strategy Considerations – Key Reports
- Testing Strategy Considerations – System Interfaces
- Testing Strategy Considerations – Automated Controls
- Testing Strategy Considerations – Automated Reconciliation
- Testing of Continued Operating Effectiveness
- Robotics Process Automation (“RPA”)
- Key Take Aways
- Questions?

Types of IT Dependencies

IT Dependencies Scoping



Key Reports and Queries

Defined as:

- A system generated document in any format serving one of the following 3 purposes during an audit:
 - Directly supporting an accounting entry or management decision
 - Supporting the execution of a control by management
 - Supporting audit procedures (i.e., populations)

Key Risks & Pitfalls:

- Does the system generated information completely and accurately reflect the source data in the system? Is the source data in the system complete, accurate and valid?
- Lack of understanding of intended use of the report and its appropriateness for that purpose

Types:

- Standard (“Canned”)
 - Customized (subject to ITGCs)
 - Ad-hoc/Query (not subject to ITGCs)
-

System Interfaces

Defined as:

- A boundary across two independent systems, which meet and act on or communicate with each other

Key Risks & Pitfalls:

- Completeness and accuracy of data transfer, as well as validity of the data received
- Lack of understanding of the end to end data flow & data modification in the process

Types:

- Manual
 - Automated
-

Automated Controls

Defined as:

- A system functionality providing consistent application of processing, validations or computations based on established rules

Key Risks & Pitfalls:

- Various iterations of the core functionality
- Lack of understanding of the intended purpose of the functionality

Types:

- System configurations
- Edit & validation checks
- Calculations

Reconciliations

Defined as:

- A process for ensuring that the two sets of records are in agreement

Key Risks & Pitfalls:

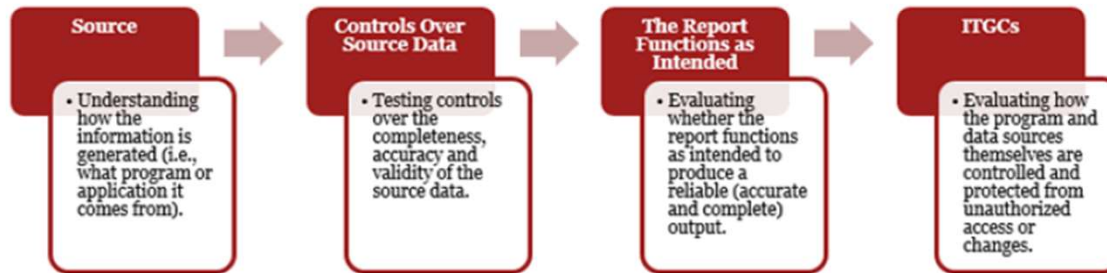
- Completeness, accuracy and validity of data, reports and iterations of reconciliation logic
- Lack of understanding of the end to end data flow & objective of reconciliation

Types:

- Manual
- Automated

Testing Strategy Considerations – Key Reports

Key Reports Considerations



- What is the source of the data?
 - How does management determine that the system represents complete, accurate and valid source of data?
 - What is the intended use of the report? What attributes (totals, sub-totals, specific columns, mathematical calculations, etc.) is management relying upon?
 - What does the system do with the data? How does the system (system logic) produce an output? How many different ways does the system perform a control (unique iterations)? Is the logic appropriate for intended use? Is the report output complete and accurate based on the logic?
 - Can manual overrides be made to the output of the system?
 - Have changes been made to the key report? What triggered the change?
 - Is there ITGCs coverage?
-

Complexity of the Report



Less Persuasive Evidence	Factors	More Persuasive Evidence
<ul style="list-style-type: none">• Non-complex system	Complexity of System	<ul style="list-style-type: none">• Complex system
<ul style="list-style-type: none">• Standard report	Report Type	<ul style="list-style-type: none">• Ad-hoc Query
<ul style="list-style-type: none">• Seeking lower level of assurance from testing	How the Report is Used in the Audit	<ul style="list-style-type: none">• Seeking higher level of assurance

Accuracy & Completeness Procedures

Report Type	Procedures
Standard (“canned”) report	<ul style="list-style-type: none">• Validate that the report is standard and no changes were made to the report since system implementation/standard vendor upgrade or patch• Test ITGCs supporting continued reliability of the report• If input parameters are used, verify the input parameters for each report supporting control testing.
Customized Report/Query (subject to ITGCs)	<ul style="list-style-type: none">• Test completeness and accuracy of the report• Test ITGCs supporting continued reliability of the report• If input parameters are used, verify the input parameters for each report supporting control testing.
Customized Report/Query (not subject to ITGCs)	<ul style="list-style-type: none">• Test completeness and accuracy of each report supporting control testing, including parameters used to generate the report.

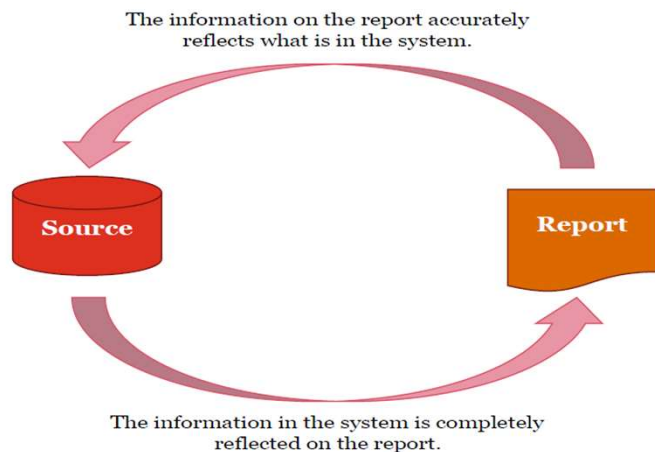
Standard (“Canned”) Report – Example Documentation

1. Inspected System User guides obtained directly from the vendor website noting that the “Report” is a standard report.
2. On MM/DD/YY, inspected the program used to execute the “Report” and noted it was last modified on MM/DD/YY by System User ID.

Procedures to Test Reliability of System-Generated Information

1. Perform Full/False Accept Reject Testing
2. Use Source data extract to reperform report/query
3. Perform an independent code review

Full/False Accept Reject Testing



- To assess whether the data included in the report is accurate, select a sample of items from the report and agree key attributes within the report to the underlying system.
- To assess whether the data in the report is complete, select a sample of items (different to the one used to test accuracy) from the system and confirm that those items are included in the report using unique key attributes.

Using a data extract, reperform the report/query

This approach is based on the ability to obtain an extract of the data used in the report from the database supporting the source application generating the report.

- Obtain an understanding of the logic used to generate the report.
- Observe the extraction of the data and confirm that the extraction is complete and apply the logic obtained
- Compare the result to the contents of the report used in management's control to replicate the output by running independent queries on the extracted data on the back end database and match to the output on the report.

Perform an independent code review

Example A – Simple Language

```
select distinct(a.roleuser), b.*  
from psroleuser a, psoprdefn b  
where a.roleuser = b.oprid  
AND B.acctlock = 0  
order by 1
```

Example B - Complex

```
/*dgpc0371 Modified fields on report */  
/*dgpc0384 Added output for filename */  
/*dgpc0588*/ /* Add logic to update the xcustpo table when a successfull  
/* extranet order is placed */  
/*dgpc0589*/ /* Bring in the customer po price for comparison */  
/*dgpc14492*/ /* Display message when customer is "inactive" */  
  
{mfdeclre.i}  
{gplabel.i} /* EXTERNAL LABEL INCLUDE */  
  
define temp-table tt_rpt_list  
  field tt_rpt_seq like edmfs_mfd_seq  
  field tt_rpt_status like edmfs_status  
  field tt_proccess like edxfsd_proc_sess  
  index tt_rpt_seq_idx tt_rpt_seq.  
  
define input parameter table for tt_rpt_list.  
define input parameter prntdetail like mfc_logical no-undo.  
  
define variable l-msgtext as character format "x(38)" no-undo.  
/*dgpc14492*/ define variable l-msg-inactive as character format "x(38)"  
/*dgpc14492*/ l-msg-inactive = "".  
  
/* VARIABLE INDICATING WHETHER TO PRINT PASSED/FAILED/  
define shared variable l_PrntFailPassBoth as character  
  format "x(1)" no-undo.
```

Perform an independent code review

- Assess the technical report logic to determine whether the report is generated as intended (i.e. combination or exclusions)
- Utilize sufficient technical ability and knowledge to formulate an independent point of view on the sufficiency of the code to satisfy the intended purpose, including both simple query languages (i.e. SQL) and complex mainframe programming languages (i.e. COBOL)
- Even code written in a simple query language may be complex, affecting the ability to efficiently perform a code review

Third-Party Reports – Considerations

- Review the SOC1 report to determine whether the system-generated information is included in the scope of the testing performed
- Evaluate user entity controls in place to assess reliability of system-generated information coming from the service provider
- When the information/report is not covered in the SOC1:
 - Obtain additional evidence from the service organization or the service auditor
 - Request additional procedures by the service auditor
 - Perform testing over the system generated information
 - Consider continued operation of the report

Testing Strategy Considerations – System Interfaces

Interfaces Considerations

- What are the ways in which data flows from source system to target system, including pass through systems?
- What is the type of data being sent over the interface, and any specific key attributes? Is the data being modified (e.g. filtered, excluded, aggregated) during the transmission?
- How often is the job executed/its frequency for the purposes of the control?
- Where is the job configuration held, either in the source or destination application, or in a job scheduler like AutoSys? Is it subject to relevant ITGCs?

Interface – Interface Job Testing

- Identify the interface job and obtain job details and configurations
- Identify and test controls for handling job abends for the job
- Perform procedures over continued operation of the interface during the period

Interfaces – Testing Completeness

- Extract from source application of the file(s) being sent to destination application, including completeness considerations (e.g. script review and analysis)
- Extract from the destination application of the file(s) received from source application, including completeness considerations (e.g. script review and analysis)
- Reperform the interface by comparing the source file (e.g. total entries, total balance) to the file received in the destination application

Interfaces – Testing Accuracy

For a sample line item(s) selected from the destination extract:

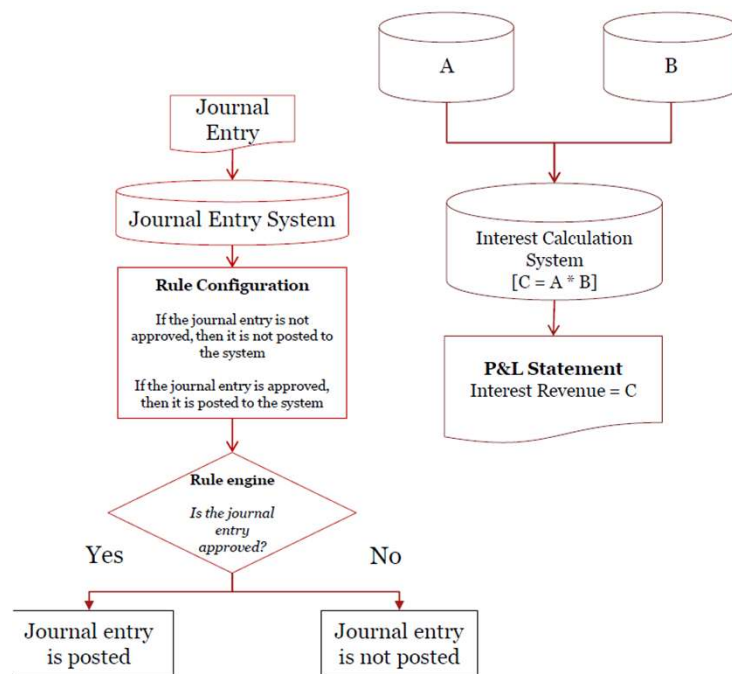
- Obtain transaction details/key fields in the destination system
- Compare those key fields to the transaction details in the source system for accuracy

Testing Strategy Considerations – Automated Controls

Automated Controls Considerations

- What are the various iterations in which the control can be configured and which ones are in use (e.g. calculation methods, validation checks, etc.)?
- What is the process for periodic validation of the system functionality?
- How and who can overwrite system functionality? And what is the process for implementing the changes?

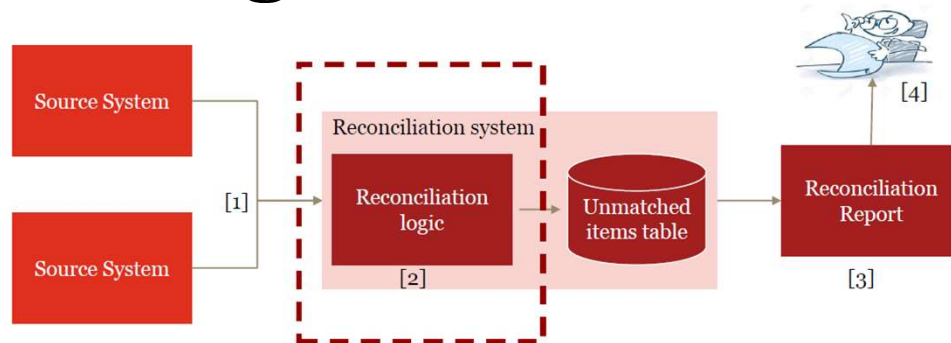
Testing Automated Controls



- Gain an understanding and evaluate the underlying logic/functionality
- Perform a walkthrough of transactions to demonstrate the operation of the functionality is consistent with business purpose.
- Perform positive and negative scenario testing for each functionality
- Perform testing over each iteration of the functionality

Testing Strategy Considerations – Automated Reconciliation

Testing Automated Reconciliations



An automated reconciliation typically has the following components:

1. Feed from the source systems to reconciliation system (interface)
2. Reconciliation logic that performs the matching (automated functionality or manual process)
3. Reporting of unmatched items (key report)
4. Research and resolutions of unmatched items (manual control)

Reconciliations Considerations

- How the reconciliation is performed (automated vs manual)?
- What is being reconciled (i.e. cash, positions, balances, etc.)?
- What are the key fields being matched?
- Is the source data, including filtering criteria and exclusions, appropriate?
- How many iterations on the matching logic exist?
- What is defined as a break (tolerances/thresholds)?
- How are the results of reconciliation displayed (i.e. exception only, etc.)?
- What is the process for aging & resolution of the reconciling items?
- What is the monitoring over aged items?

Testing Automated Reconciliations

Method 1: Using source data/inputs

- Define Expectation - Obtain source data/inputs & apply necessary filters/exclusions based on systematic logic
- Define Outcome - Obtain results of reconciliation performed by the system (report, dashboard, etc.)
- Reperform Reconciliation - Compare the expectation from #1 vs outcome from #2

Testing Automated Reconciliations

Method 2: System Validation

1. Perform walkthrough to gain an understanding of the systems, feeds, and logic associated with the reconciliation
2. Define Expectation - Obtain Positive iterations and Negative iteration (data that matches and data that does not match) and submit it into the application/reconciliation tool for processing
3. Define Outcome - Obtain results of reconciliation performed by system for both Positive iterations (which do not create reconciling items) and Negative iterations (which do create reconciling items)
4. Reperform Reconciliation - Compare results of expectation in #2 vs outcome in #3
5. Generation of Key Report - Test that reconciliation results appears completely and accurately

Note: consider the effectiveness of ITGCS & test the continued operation of the control throughout the period

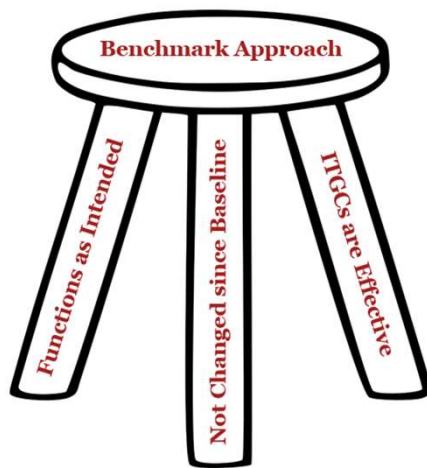
Testing of Continued Operating Effectiveness

Baseline Approach



- Obtain evidence directly from the system to validate that the system performs the control as designed.
- Evaluate changes to the core functionality of the automated control or key report during the period.
- ITGCs are in place and are effective

Benchmark Approach



- An alternate test approach where reliance can be placed on prior period testing, when all of the following are true:
- Automated control or key report functions as intended (it was tested it in the past)
- Generally, “past” = within the past 3 years
- ITGCs are in place and are effective
- No changes have been made to the core functionality of the automated control or key report since it was last tested.

Robotics Process Automation ("RPA")

RPA Considerations

- Entity level controls - RPA framework, policy and procedures, etc.
- Access to programs and data - How is access provisioned to the bot? will the bot have direct access to data? How will access be monitored?
- Program change - How is the RPA configured and implemented? Who owns the controls associated with development, test and implementation?
- Computer Operations - Is the business monitoring the bot processing? How are errors/failures identified and corrected?
- Other considerations - Software vendors, business resilience & impact to the business.

Documenting use of Technology

- Scope and purpose of the work to be performed, including audit risks to be addressed
- Procedures over the completeness and accuracy of the inputs
- How/where/what/and from whom were the input(s) obtained
- Procedures performed and/or logic applied to generate the results, including actual script/programming (e.g., SQL) or an export of the workflow or system logic
- Testing procedures and summary of analysis and findings

Key Take Aways

Key Take Aways

- Understanding of the end to end process and intended use/purpose of the IT Dependency is crucial for development of a strong and efficient testing approach
- There is no one right way to test IT Dependencies
- Most IT Dependencies are tested at a point in time and considerations should be given to the continued operation during the period in scope (i.e. changes to ITD, effective ITGCs, etc.)

Questions?

This publication has been prepared for general guidance on matters of interest only, and does not constitute professional advice. You should not act upon the information contained in this publication without obtaining specific professional advice. No representation or warranty (express or implied) is given as to the accuracy or completeness of the information contained in this publication, and, to the extent permitted by law, PricewaterhouseCoopers LLP, its members, employees and agents do not accept or assume any liability, responsibility or duty of care for any consequences of you or anyone else acting, or refraining to act, in reliance on the information contained in this publication or for any decision based on it.

© 2019 PricewaterhouseCoopers LLP. All rights reserved. In this document, "PwC" refers to PricewaterhouseCoopers LLP which is a member firm of PricewaterhouseCoopers International Limited, each member firm of which is a separate legal entity.
