

**Architecture-Driven Modernization (ADM)  
Workshop:  
“A Model-driven Approach to Modernizing  
Existing Systems”**

*William M. Ulrich –*

*Tactical Strategy Group, Inc.*

**[www.systemtransformation.com](http://www.systemtransformation.com)**

# Seminar Overview

- I. **Objective Setting**
- II. **Existing Applications**
- III. **Integration & Modernization Overview**
- IV. **Modernization Infrastructure Requirements**
- V. **Enterprise Assessment**
- VI. **Project-level Assessment**
- VII. **Application & Data Migration, Structuring & Rationalization**
- VIII. **Architecture Modernization & Redeployment**
- IX. **Enabling Tools & Technologies**
- X. **Modernization Scenarios & Case Studies**
- XI. **Getting Started**
- XII. **Selling Management / ROI**
- XIII. **Close / Questions**

# I. Seminar Objectives

- Define existing architecture challenges
- Introduce modernization concepts
- Discuss architecture transition strategy
- Outline techniques for performing application assessments, integration and modernization
- Describe practical application of these concepts through project-based scenarios
- Introduce enabling tools and technologies
- Help attendees get started

# CIO Existing Systems Survey\*

How useful are your company's existing systems in helping you reach your strategic goals?

- Very useful / somewhat useful 63.7%
- Somewhat limiting / very limiting 36.3%



Existing systems are still important

\* CIO Insight – December 2002

# CIO Existing Systems Survey\*

What are the major reasons for keeping the existing systems your company is currently planning to retain?

- Still able to support business processes 54.4%
- Still reliable 49.7
- Staff available to support them 44.3
- Still more cost effective than alternatives 41.6
- We do not have the budget to make any changes 36.9
- Still able to support strategic goals 36.9



Still relevant and reliable

# CIO Existing Systems Survey

If you are currently migrating or planning to migrate off any of your current existing systems in the next year, what are the main reasons?

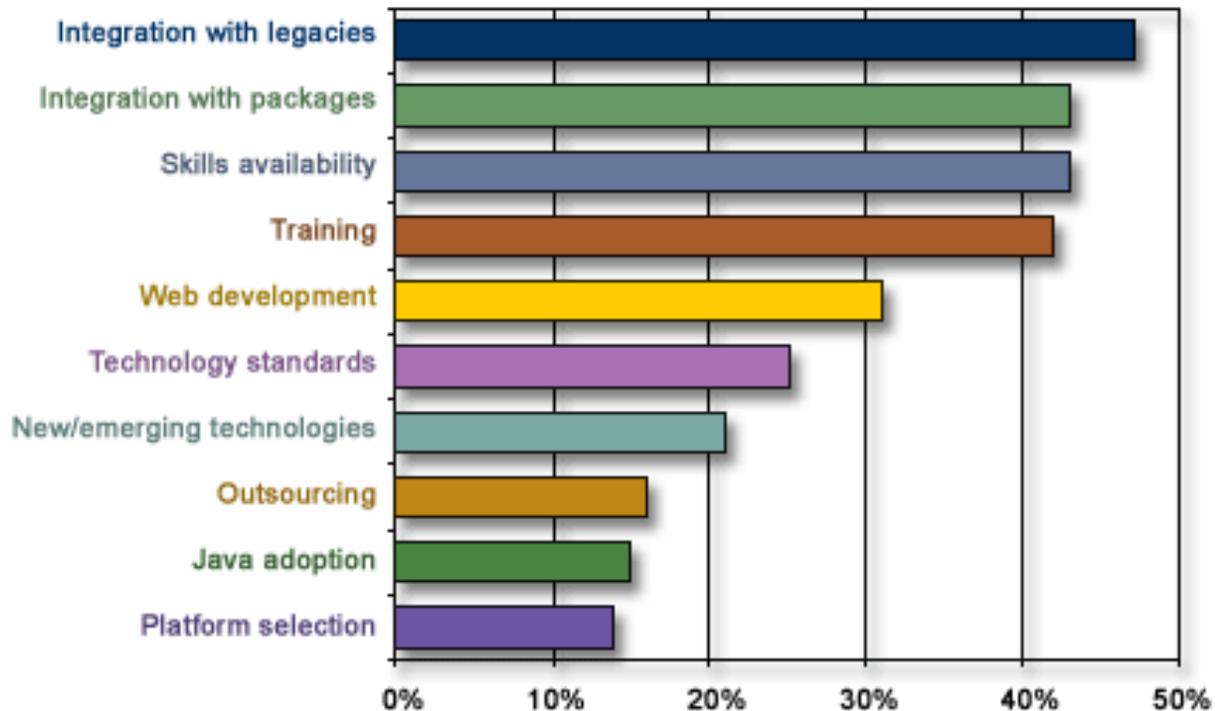
- Meet new strategic goals 65.2%
- Existing systems can't support business processes 59.9
- Existing systems can't support current strategic goals 56.1
- New systems would be more cost effective 48.5
- Existing systems provide poor interoperability 41.7



Migration / Modernization plans

# Gartner Survey

## Issues reported by IT managers as their main concerns



## II. Existing Applications vs. Emerging Technologies

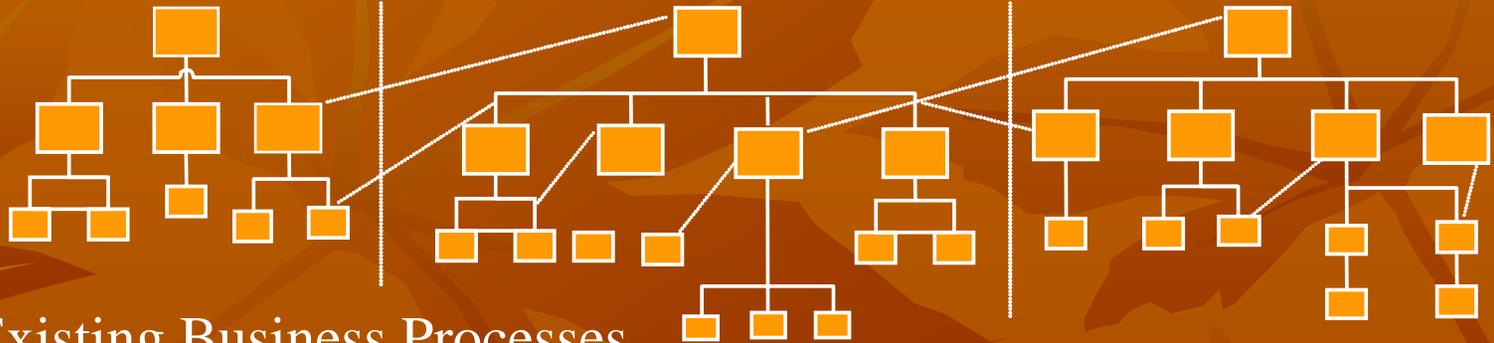
- Existing architectures run in sharp contrast to modern IT disciplines.
- The challenge is how to migrate to emerging architectures while meeting critical business requirements.

# Existing Applications & Architectures

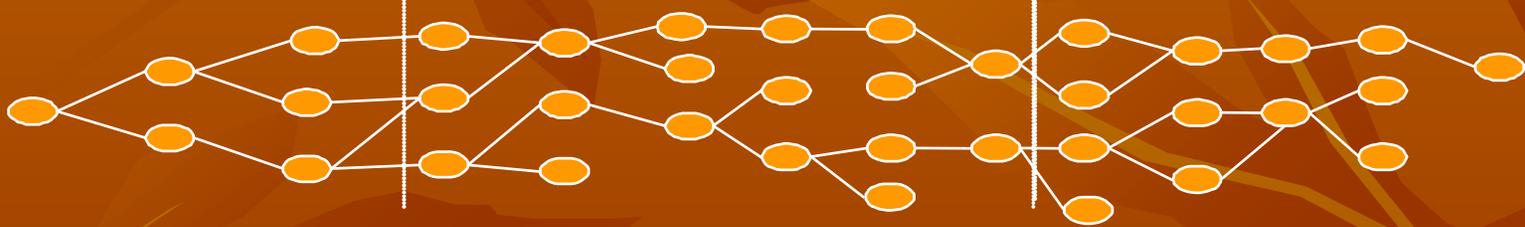
- Numerous languages and platforms
- 200 billion lines of Cobol (60% of total)
- Haphazard design and stovepipe architectures
- Segregated functionality and data structures
- Layers of middleware and data warehouses
- Web-based interfaces

# Poorly Integrated Information Architectures

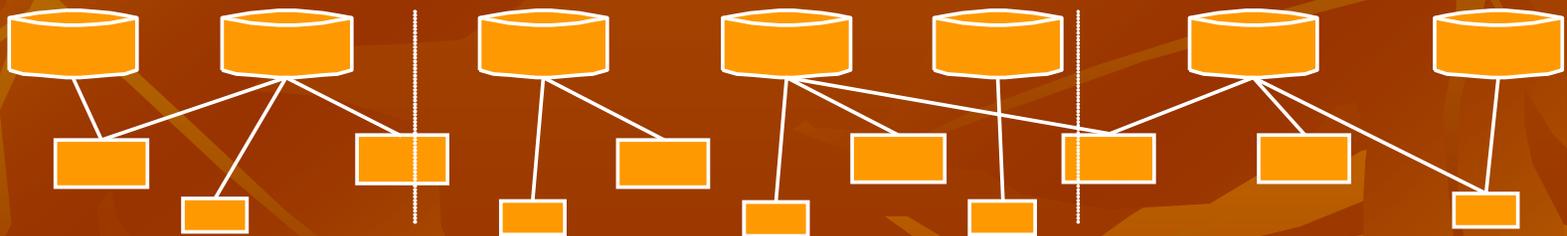
## Existing Organizational Governance Structures



## Existing Business Processes



## Existing Data & System Architectures



Business processes, applications and data mirror stovepipe information governance structures.

# Existing Architectures Impede Ability to Deliver Business Value

- 85% of IT projects: late or never delivered\*
- Only 9% of IS projects come in on time or within budget\*
- ERP projects: years to implement, canceled 35% of the time & rarely fully deployed\*
- \$16.5 billion is spent annually on systems that users never see (Information Week)
- Management has been duped into believing “quick & easy” solutions can solve complex IT challenges

\* Standish Group International

# III. Integration & Modernization Overview

- Integration: Non-invasion approach that connects processes, data and applications
- Modernization: Invasive solutions to address existing architecture challenges
- These disciplines are interdependent and interrelated
- Modernization is an augmentation strategy

# Concepts and Terminology

**Integration:** Connecting business processes, data and applications across Information architectures.

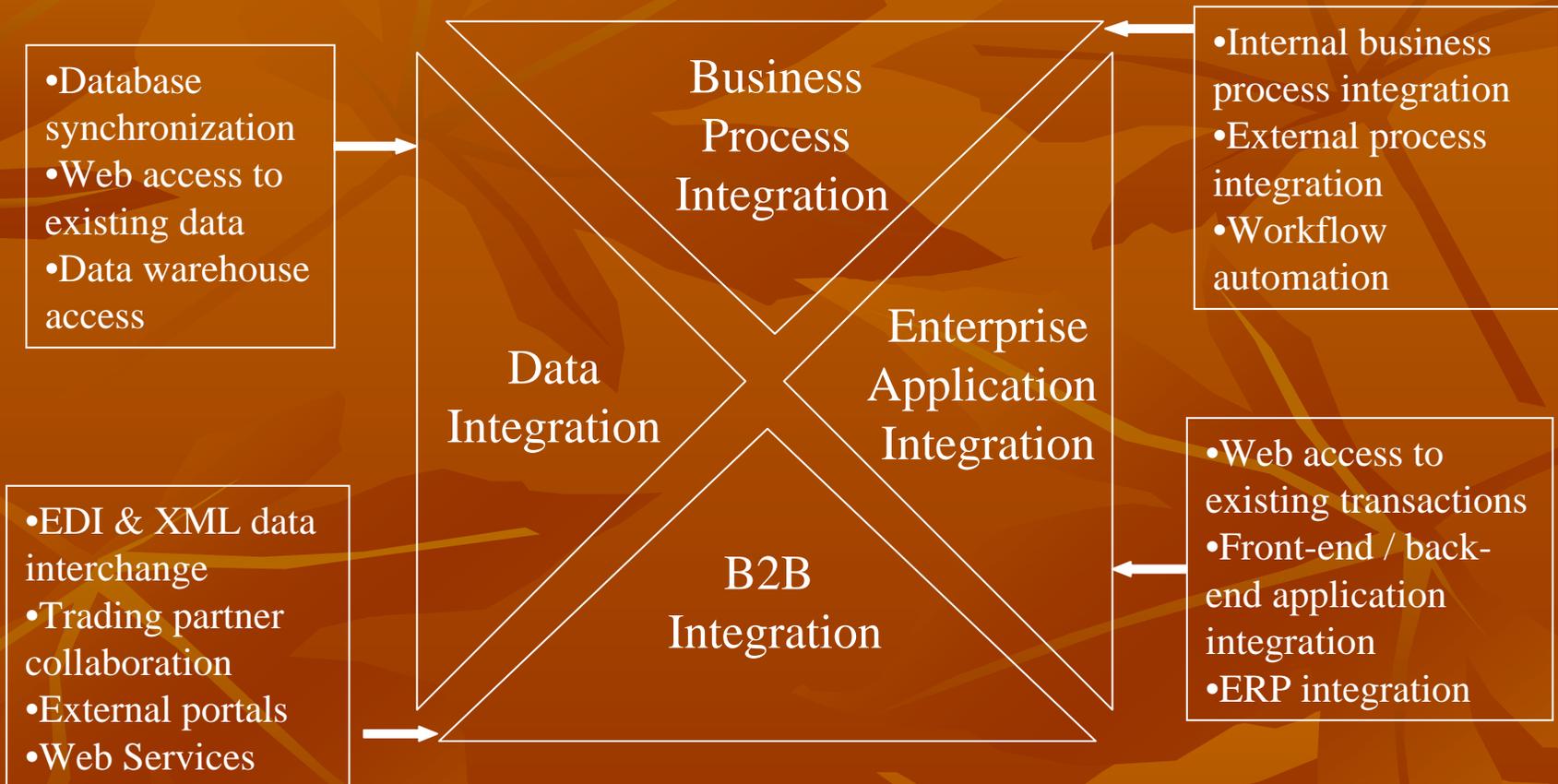
**Modernization:** The analysis, improvement, redesign and reuse of existing software systems to meet changing information requirements.

**Reverse Engineering:** Process of analyzing systems to identify components and interrelationships, and create representations in another form or a higher level of abstraction.

**Reuse:** Practice of employing the same logical or physical components within an information architecture to reduce redundancy and facilitate change.

**Architecture-Driven Modernization (ADM):** Process of understanding and evolving existing software assets. Modernization starts where existing practices fail to deliver against business objectives.

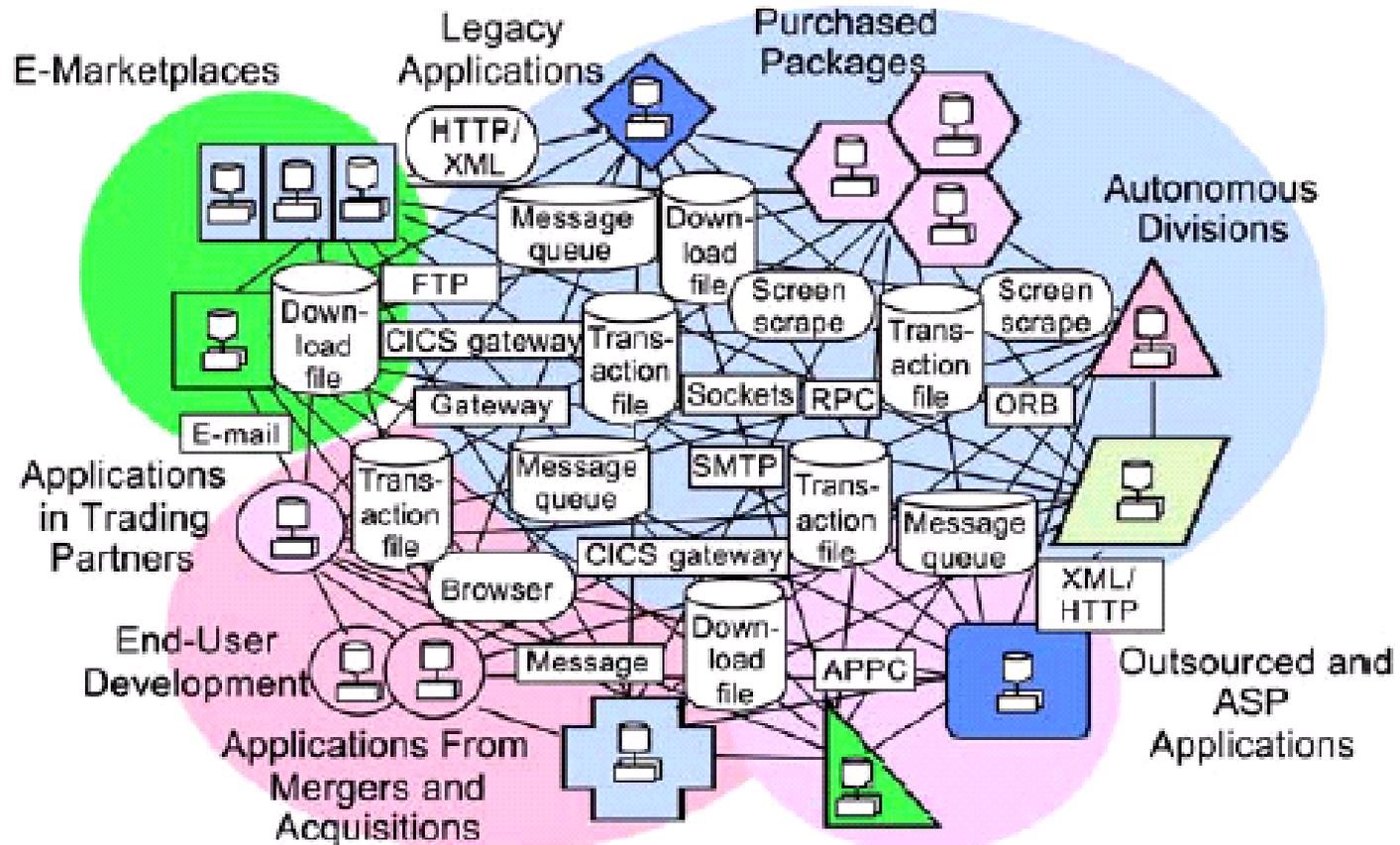
# Non-Invasive Integration Options



A conceptual framework for non-invasive integration allows planning teams to position integration technologies within an overall strategy.

# EAI Architectures can get Messy

## Application Integration Spaghetti

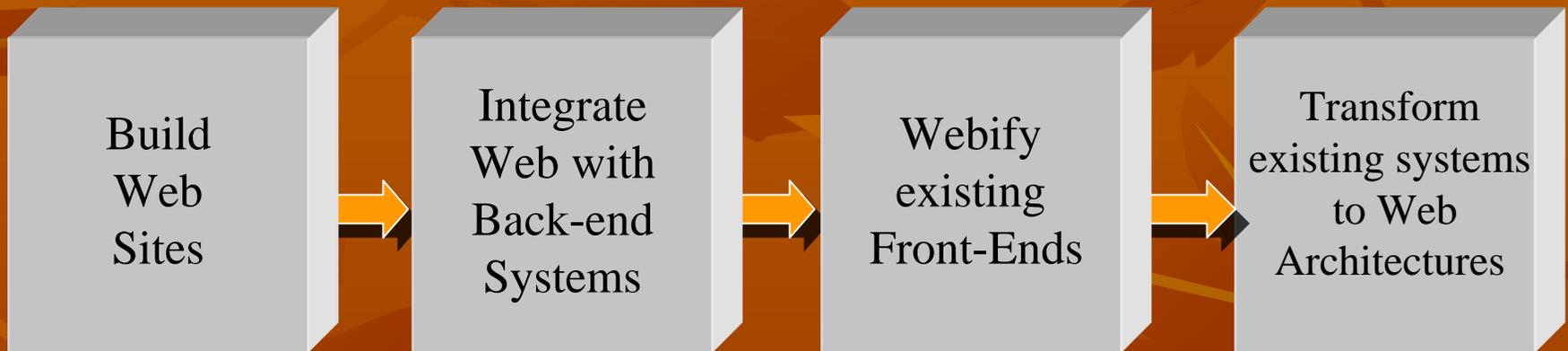


Source: Gartner

Source: Gartner Forecast Newsletter Issue 3, 2001

Vendor-driven integration solutions create layers of chaos across information architectures.

# IT Architecture Evolution



Four stages of Architecture Evolution

# Modernization: Bridges Gap Between Existing & Emerging Architectures

Business-Driven Requirements



Existing Architecture Modernization



Existing Architectures



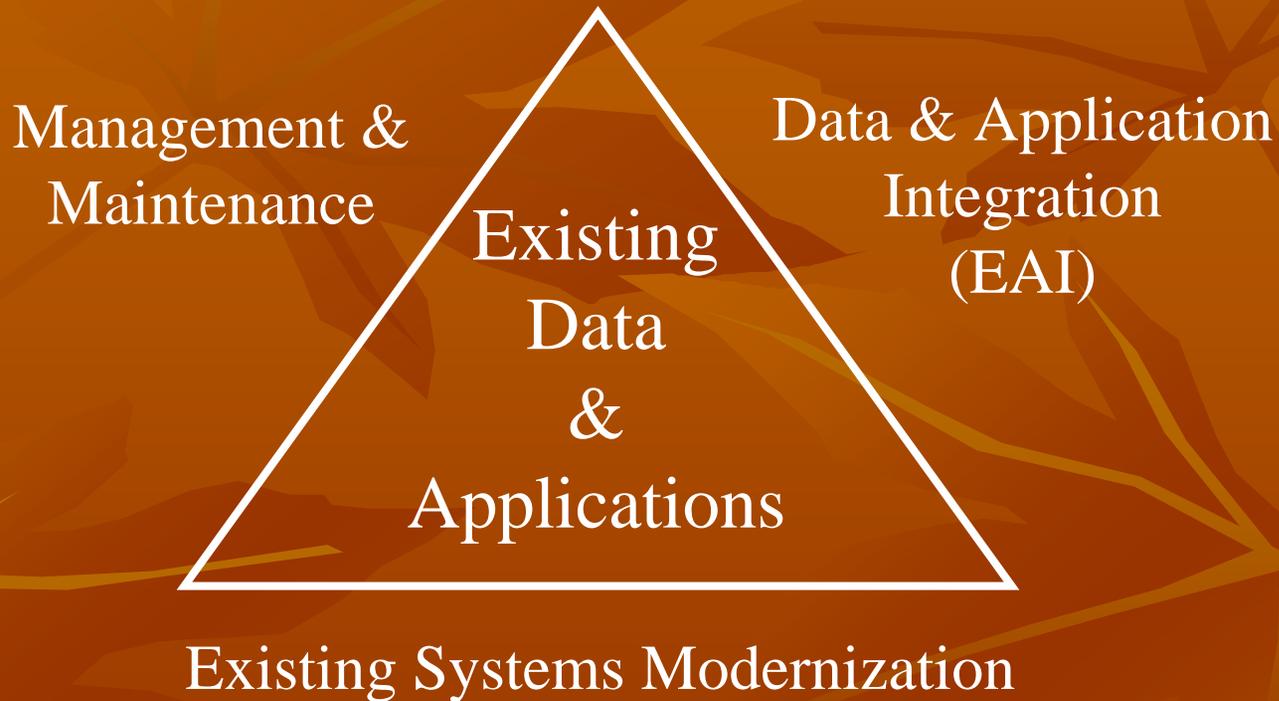
Strategic Architectures

# Existing Systems

## Modernization Includes:

- Understanding application and data architecture and functionality
- Making existing systems more reliable and adaptable
- Extracting and rationalizing data definitions, data and business rules
- Redesigning and reusing existing rules and data within the context of a strategic architecture

# Managing & Leveraging Existing Architecture's Requires Certain Disciplines



IT will need to use a combination of maintenance, EAI and modernization options to address existing system challenges.

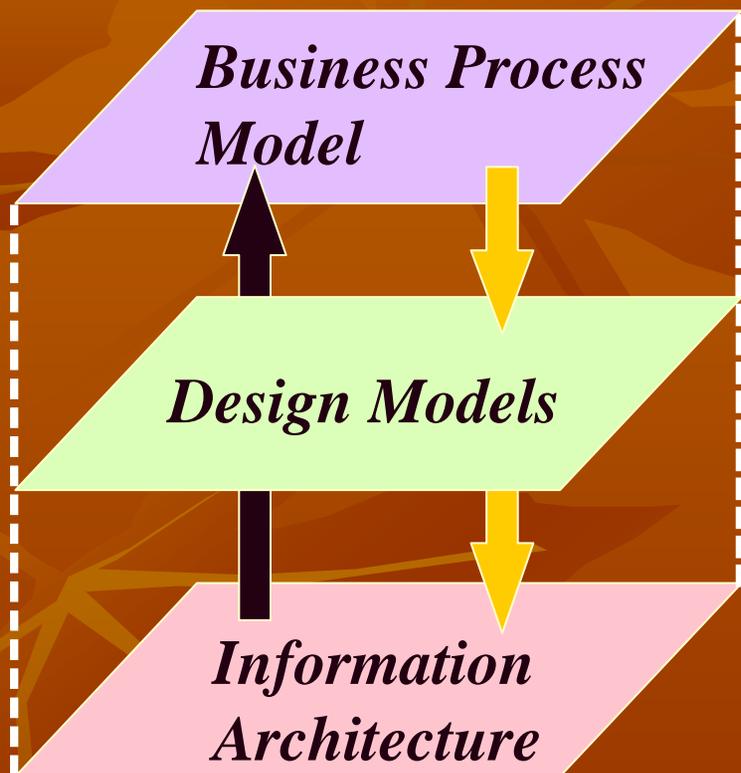
# IV. Modernization Infrastructure Requirements

- Recognition of existing value
- Process for assessing existing architectures and capturing, reusing and migrating existing components
- Process for analyzing, designing, building and deploying target architectures
- Software that facilitates and integrates modernization and development processes

# Shift from “From Scratch” Development Philosophy to Phased Reuse

- Replace “throwaway” philosophy with “reuse” philosophy
- Shift from an “all or nothing / go for broke” approach to a phased deployment approach
- Seek lower risks, higher returns and faster delivery through phased delivery strategy

# Existing Systems Reuse Should be Built into Project Specific Re-Design Process



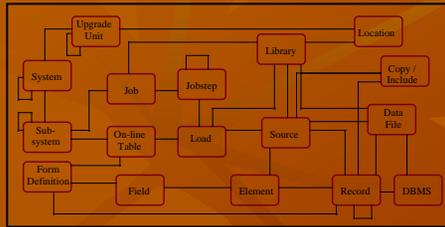
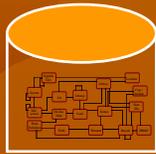
*Business models represent as-is & reengineered business processes*

*Design models depict business processes to be automated*

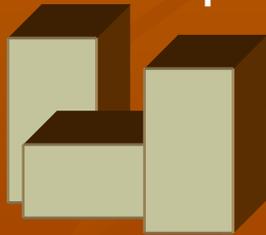
*Existing systems: input to redesign process & source of reusable data & business rules*

Mapping existing architecture to requirements and reusing existing components in new design and implementation.

# Modernization Process & Tool Infrastructure



**Repository Technology**



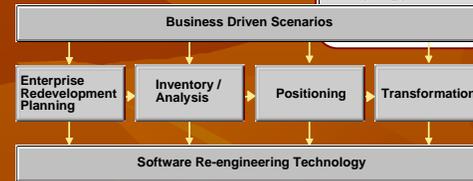
**Validation Tools**



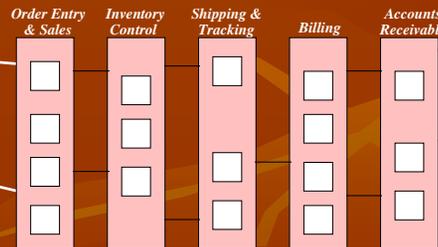
**Analysis & Upgrade Tools**



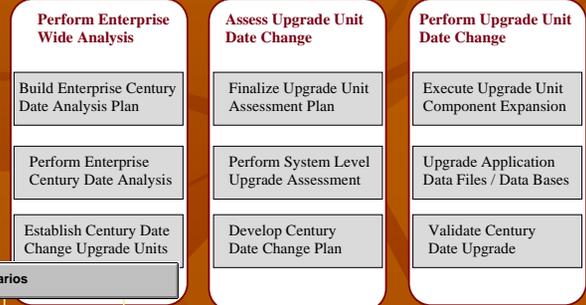
**Process Management**



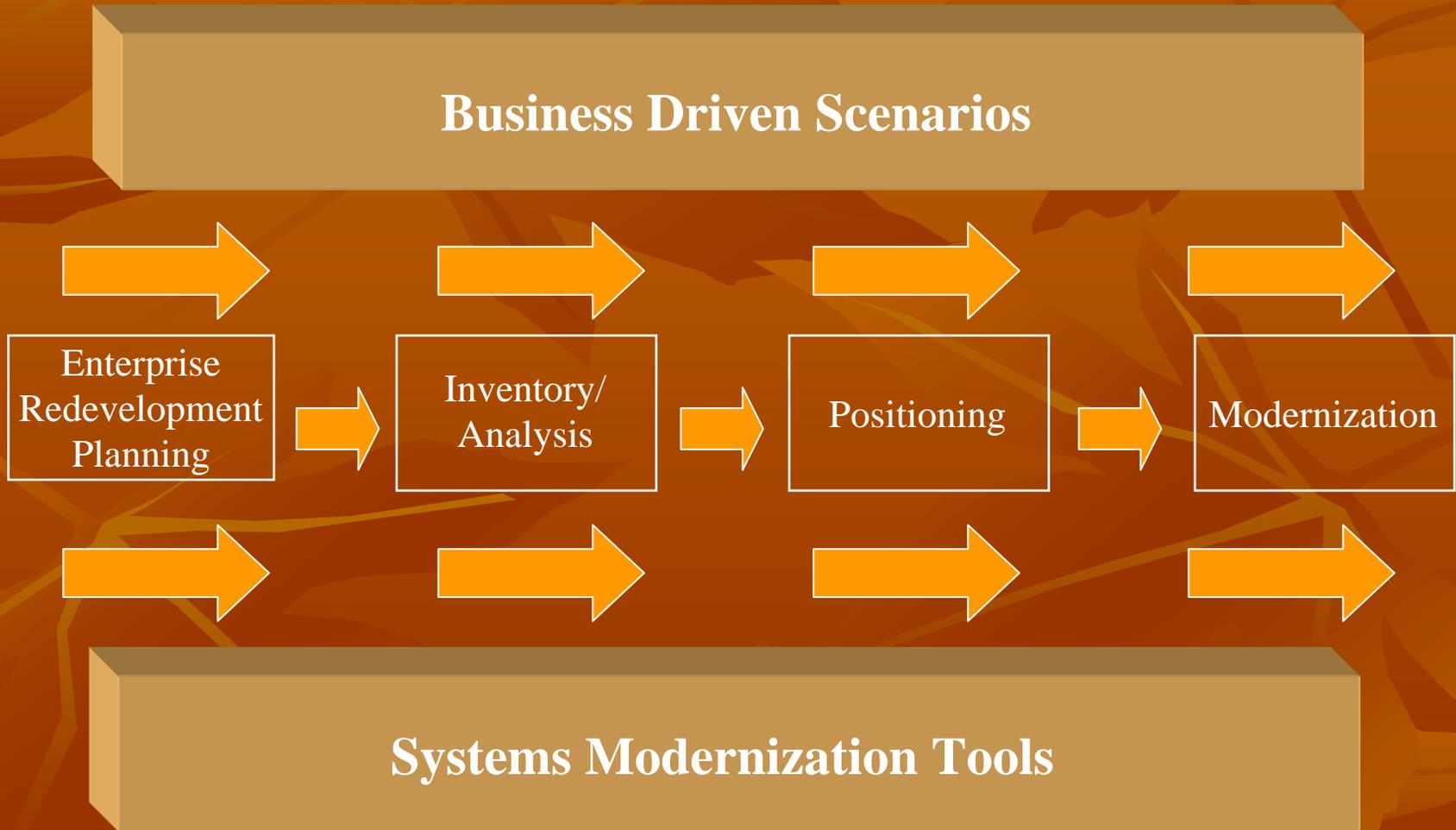
**Process Model / Blueprint**



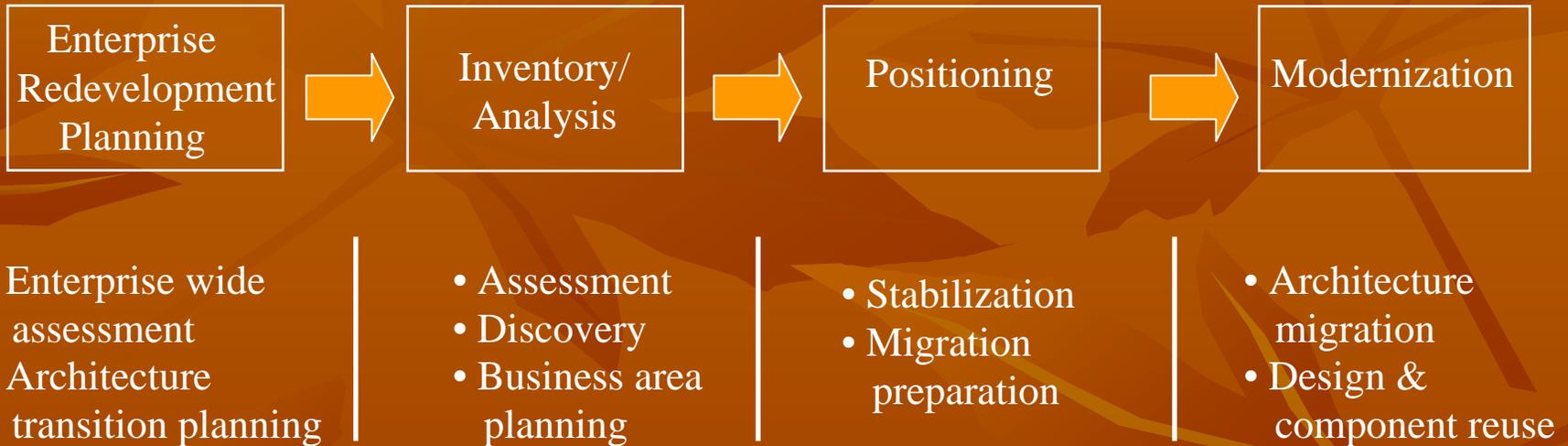
**Change Control & Configuration Mgmt**



# Sample Modernization Methodology



# The Value of a Modernization Framework

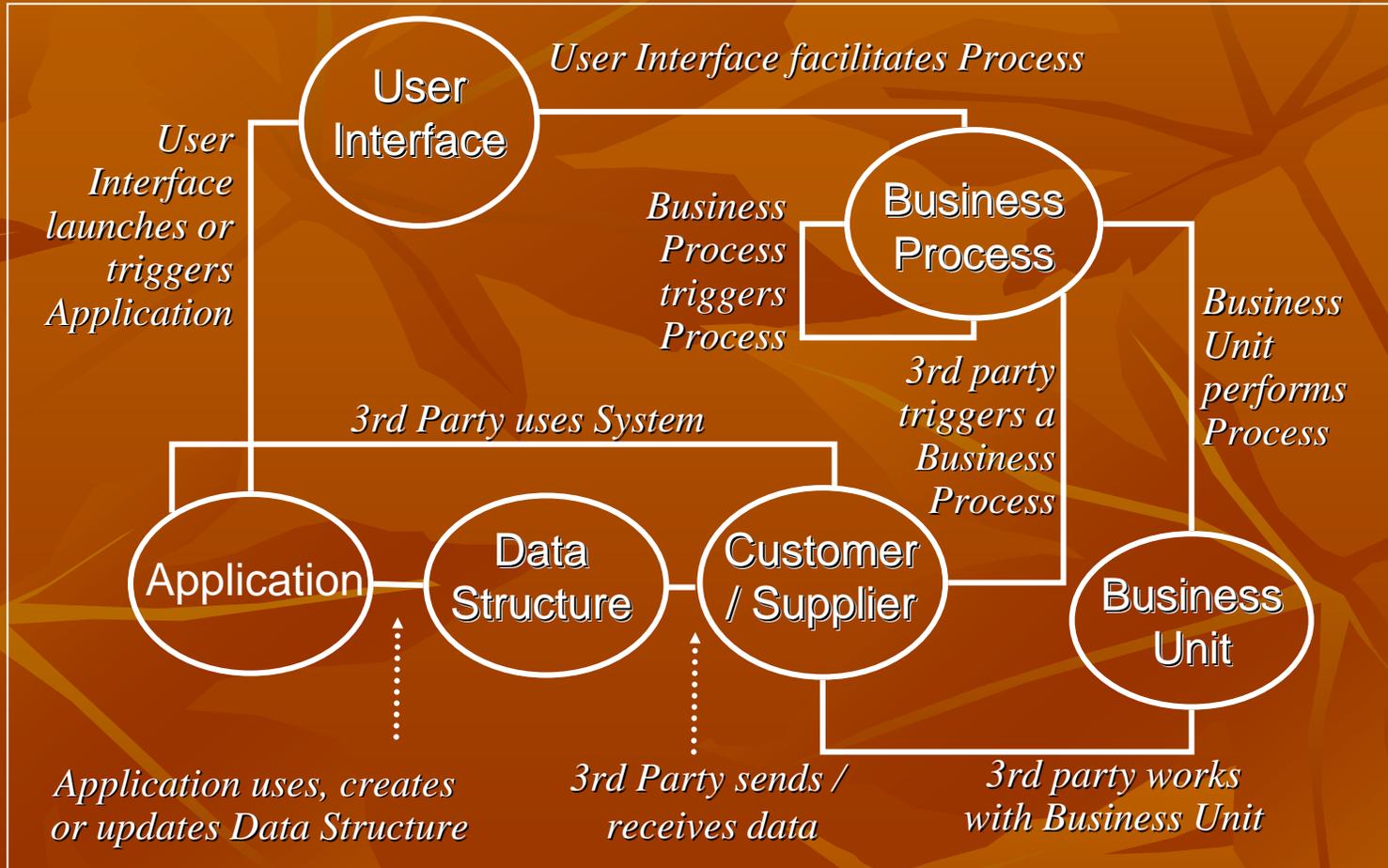


- Based on results oriented case studies and success stories
- Each stage contains multiple tasks and each task contains multiple steps
- Supports phased implementation strategy

# Measurement & Metrics

- Metrics support analysis and planning of modernization and integration projects
- Range from simple counts to more complex scores
- **EXAMPLES:**
  - Total # batch jobs in system
  - # Data Elements in current system in common with data elements in ERP system
  - Function points, McCabe, Halstead, etc.
  - Total integration points

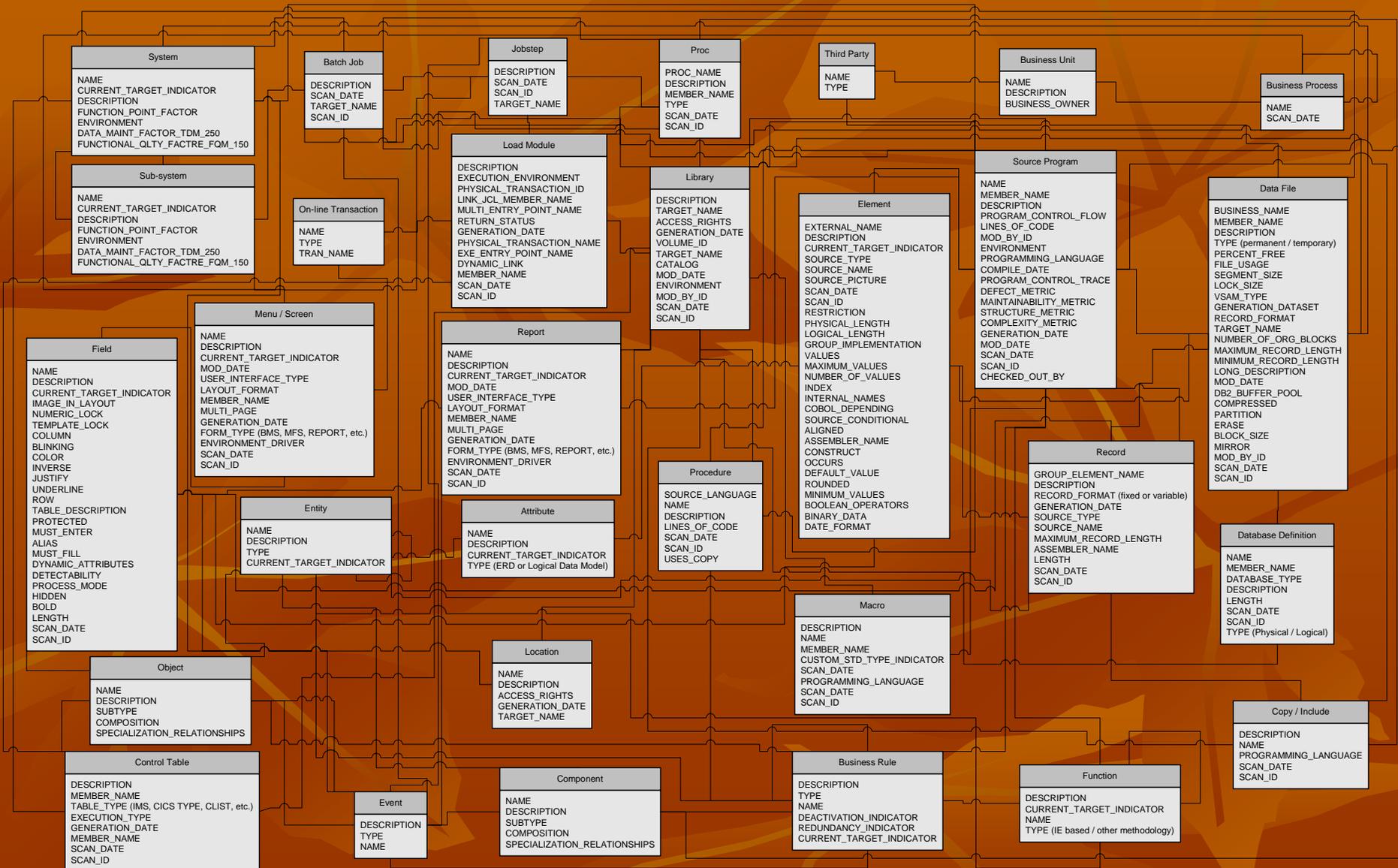
# Modernization Repository Facilitates Architecture Mapping



Enterprise information architecture can be captured in modernization repository to support various projects.

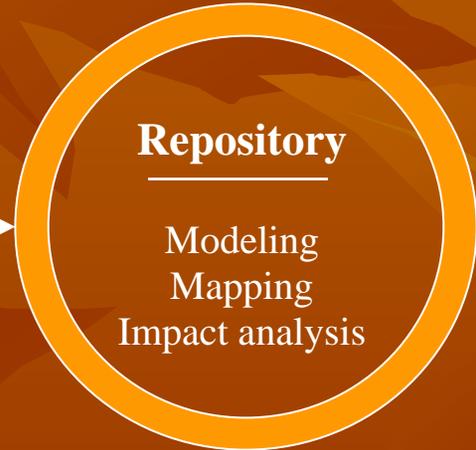
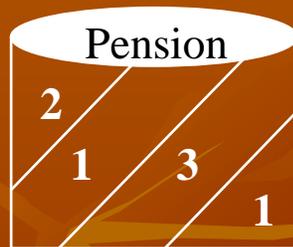
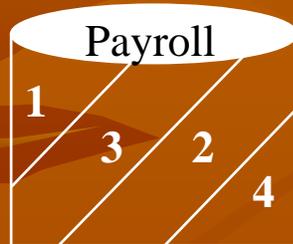


# Project Level Repository

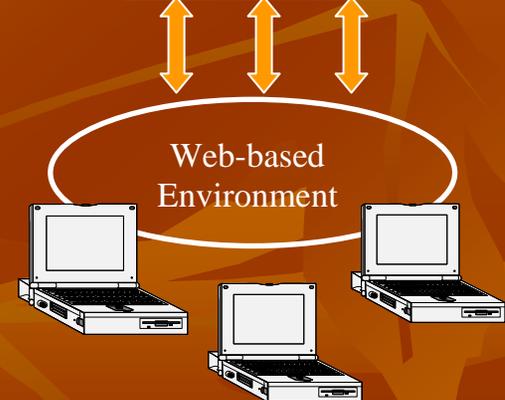
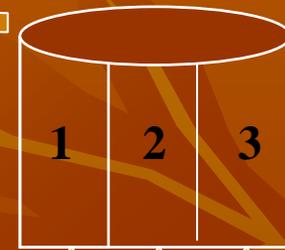


# Repository Support – Transition & Concurrency Mapping

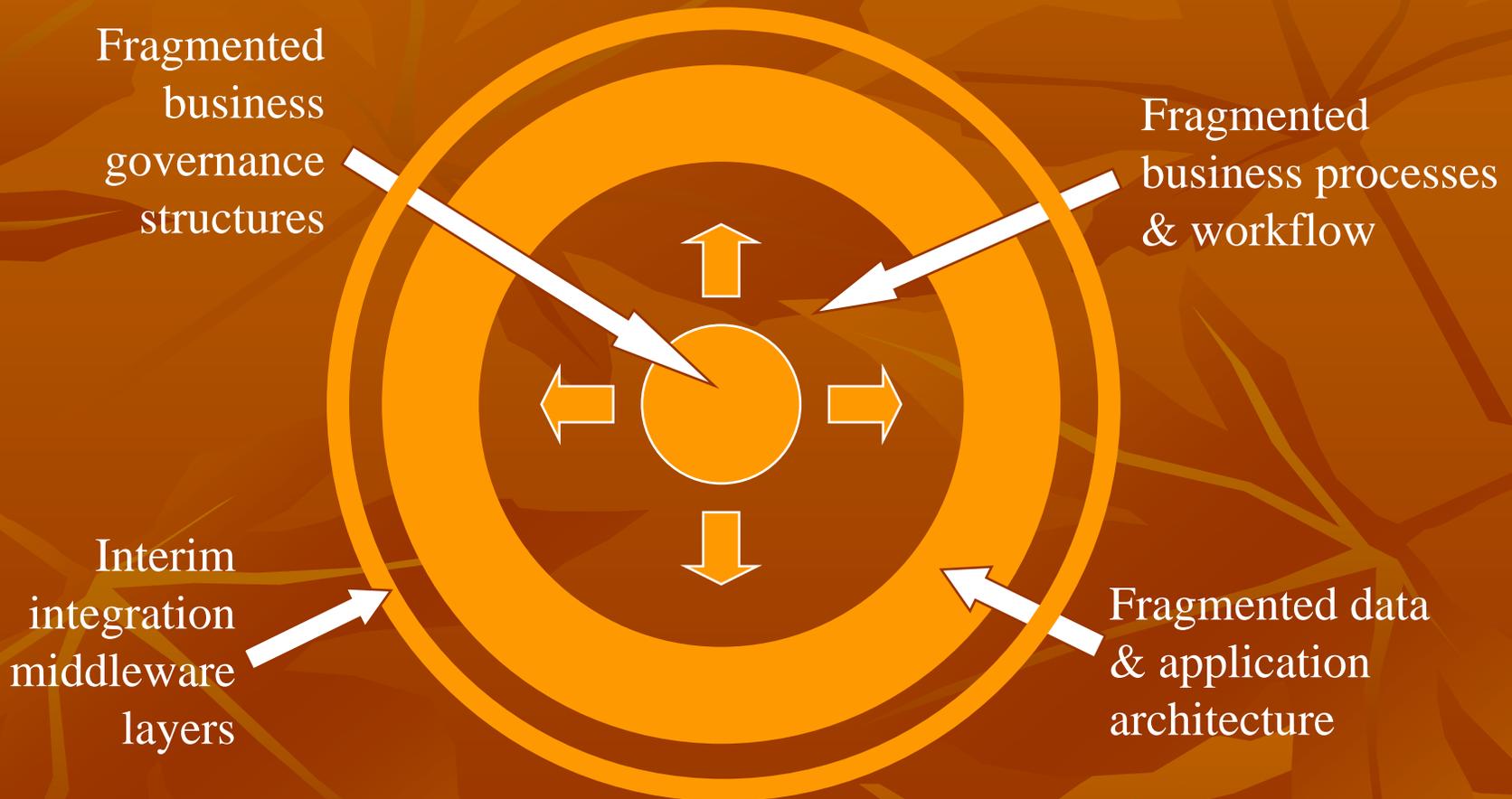
Existing Architecture



Target Application & Data Architecture



# Modernization must Address Complex Dimensions of Existing Systems Challenge



Integration solutions must assess fragmentation impacts between business units and IT while considering the impact of interim integration layers.

# Business Project / Modernization Task Mapping

<b>Transformation Task</b> <b>Business-Driven Project</b>	Enterprise Inventory	Application Documentation	Application Cleanup	Data Mining / Cleanup	Data Redesign / Migration	Language Upgrade / Conversion	EAI & User Interface Upgrade	Application Modularization	Architecture Migration
Strategic IT Planning	X								
Outsourcing Initiative	X	X	X						
Maintenance Upgrades		X	X						
BPI / BPA	X	X					X		X
Rehosting	X	X				X			
Package Assessment	X	X							
Package Implementation		X		X	X				X
HIPAA (data upgrade)	X	X	X	X					
Web-to-Host	X						X		
CRM Initiative	X	X			X		X		
Data Warehouse		X		X	X		X		
Application Consolidation	X	X	X		X		X	X	
Component Development		X	X		X			X	X
Web Services Migration		X	X		X			X	X

# V. Enterprise Assessment

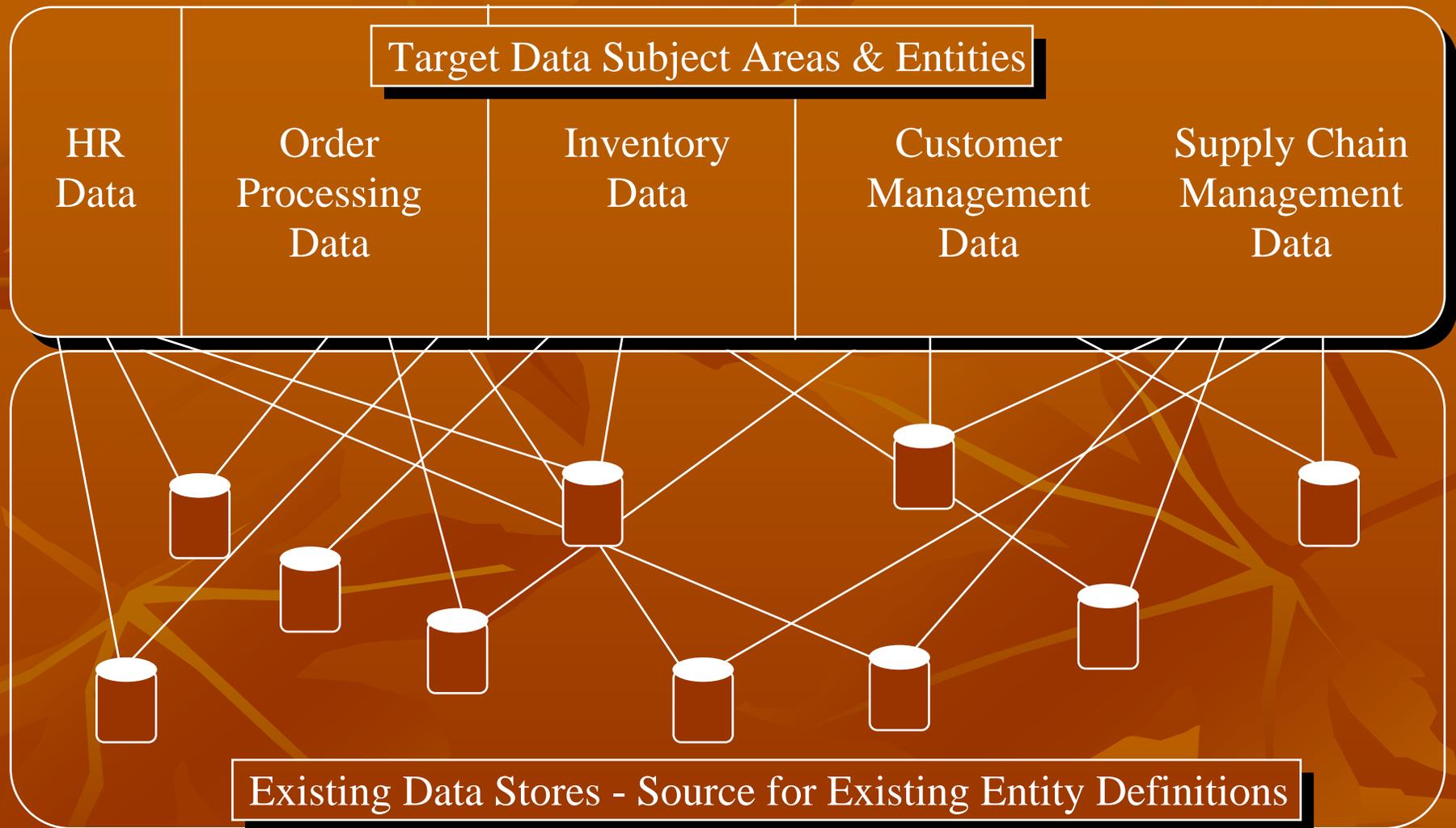
## Objectives

- Identify strategic information architecture requirements
- Inventory existing application portfolio
- Establish architecture transition strategy
- Link priority application projects with suggested redevelopment scenarios

# Enterprise Redevelopment Planning Deliverable Summary

Technical Architecture Assessment	Business Architecture Assessment	Data Architecture Assessment	Information Redevelopment Planning
<ul style="list-style-type: none"> <li>• Current physical systems inventory</li> <li>• Current physical systems attributes / metrics</li> <li>• Current to target technical architecture variances</li> </ul>	<ul style="list-style-type: none"> <li>• Current systems summary</li> <li>• Current physical systems to current functions matrix*</li> <li>• Current physical systems to target functions matrix</li> <li>• Current systems inter-relationship DFD*</li> <li>• Functional mapping summary</li> </ul>	<ul style="list-style-type: none"> <li>• Current data stores description*</li> <li>• Current data stores to current systems matrix*</li> <li>• Current data stores to target entity types matrix</li> <li>• Data mapping summary</li> </ul>	<ul style="list-style-type: none"> <li>• Current to target enterprise architecture summary</li> <li>• Architecture transitioning &amp; impact summary</li> <li>• Suggested integration &amp; modernization project options</li> </ul>

# Application-Data-to-Business-Data Mapping



Current-to-target data mapping highlights redundancies, inconsistencies and fragmentation within the existing architecture.

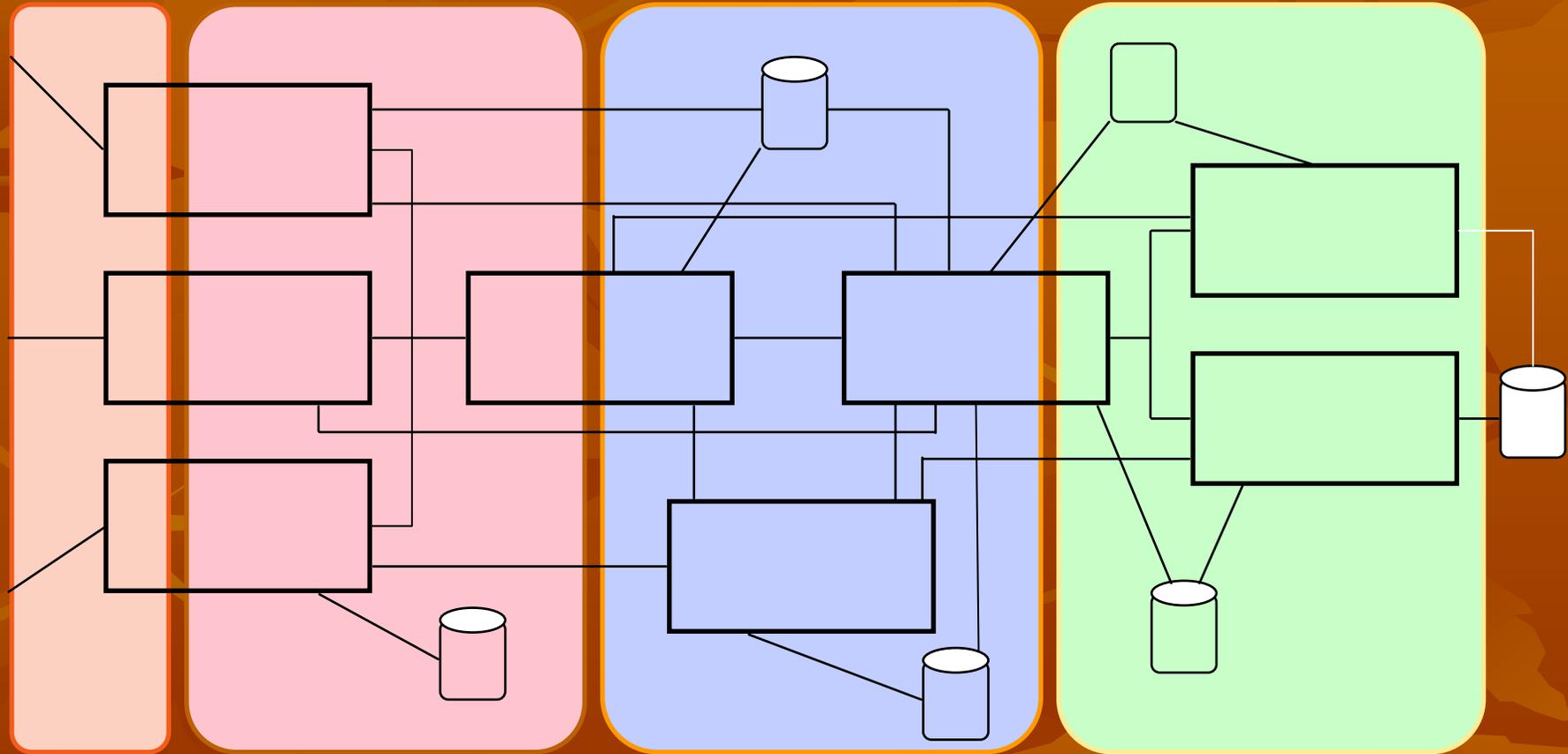
# Existing Systems Architecture Laid Over Target Architecture

Order Processing

Procurement Handling

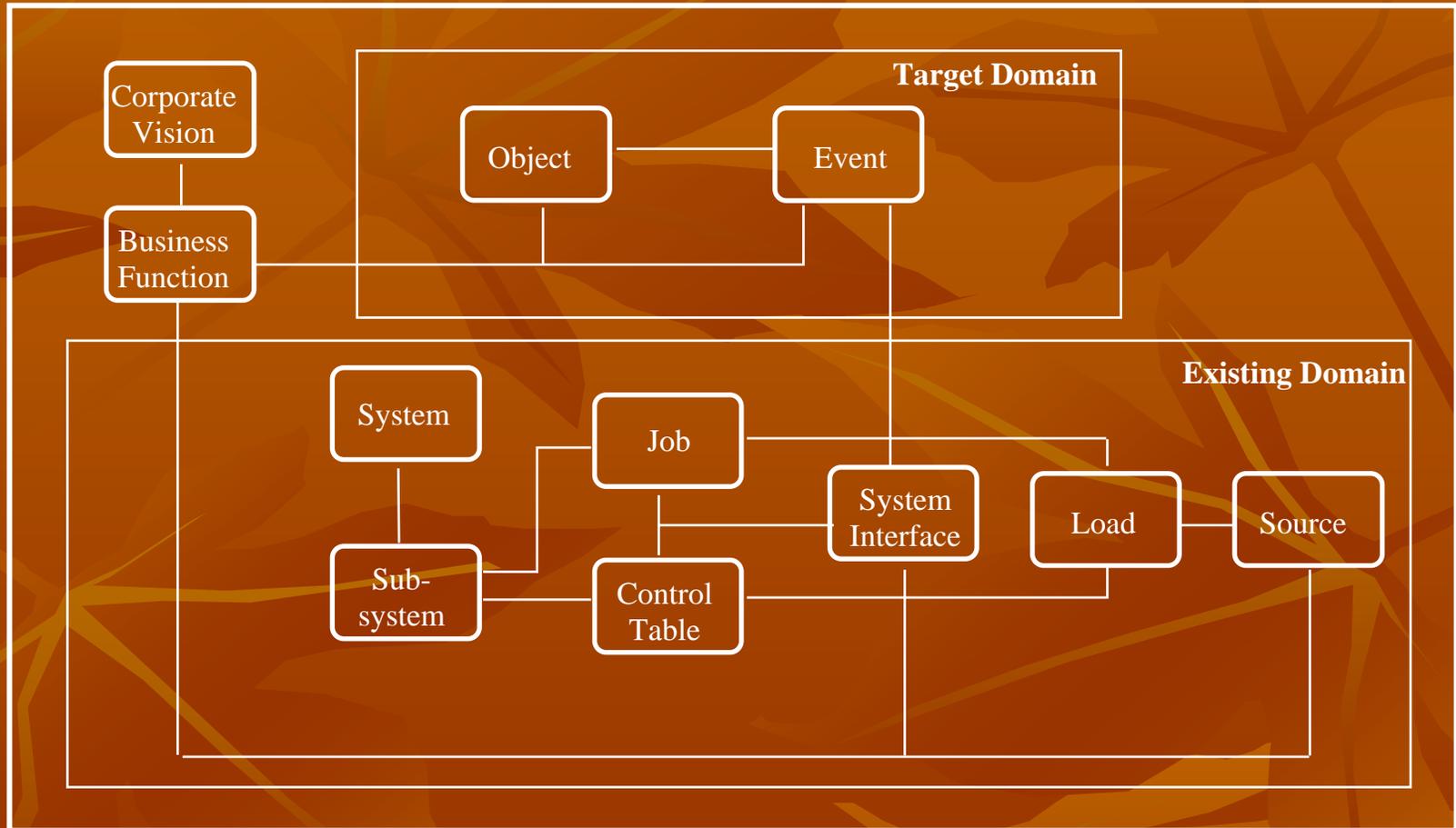
Inventory Management

Customer & Billing



Helps planning teams define project scope.

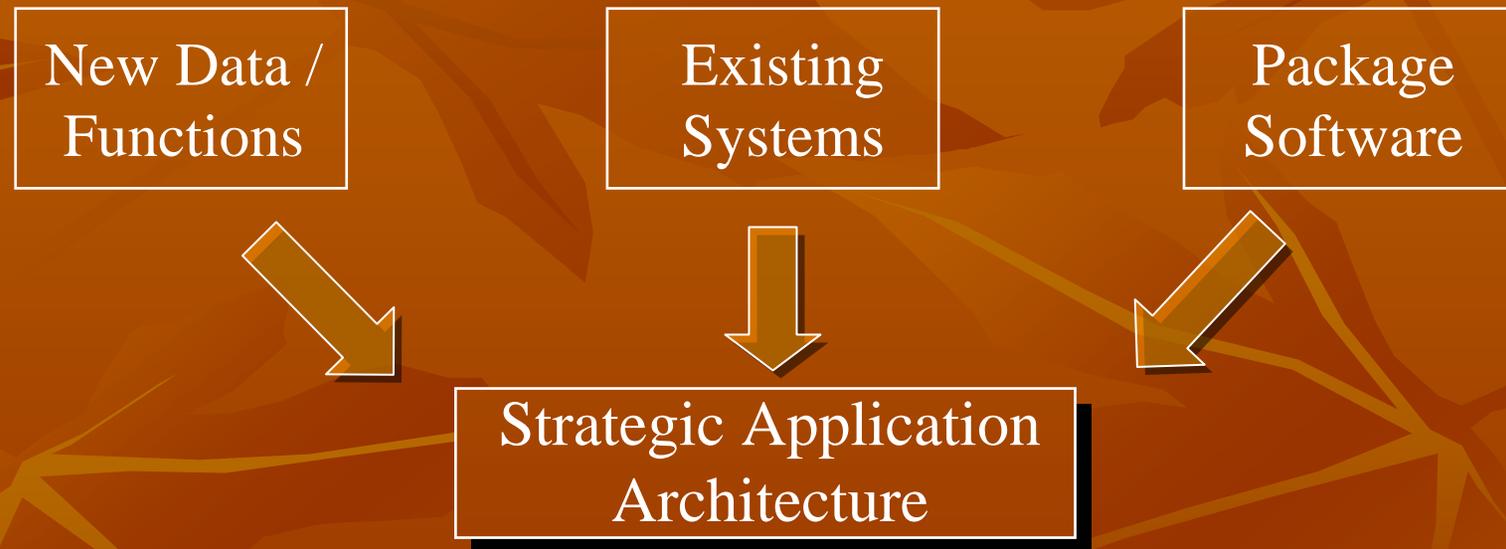
# Mapping Business Functions to Application Systems



Repository model facilitates mapping of business processes to current system functions and components. Model also supports object / event derivation and rule reuse in target design.

# Application Modernization Planning

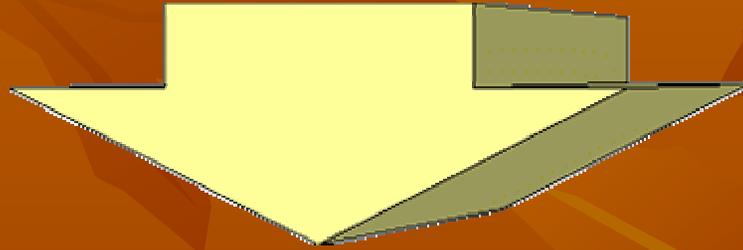
## Integration Technology



- What mix of existing, new and package components will define an organization's application systems under the target architecture?
- How will organizations partition and transition existing systems to achieve strategic targets?

# Key Tasks Regarding Modernization & Integration Planning

*Enterprise Modernization Plan identifies path forward*



## Information Systems Architecture

- Business Architecture
- Data Architecture
- Technical Architecture
- Define model driven, implementation independent design

## Architecture Transition Strategy

- Assess existing systems
- Map existing to target architecture
- Establish enterprise transition strategy
- Establish maintenance strategy
- Manage shifting target specifications

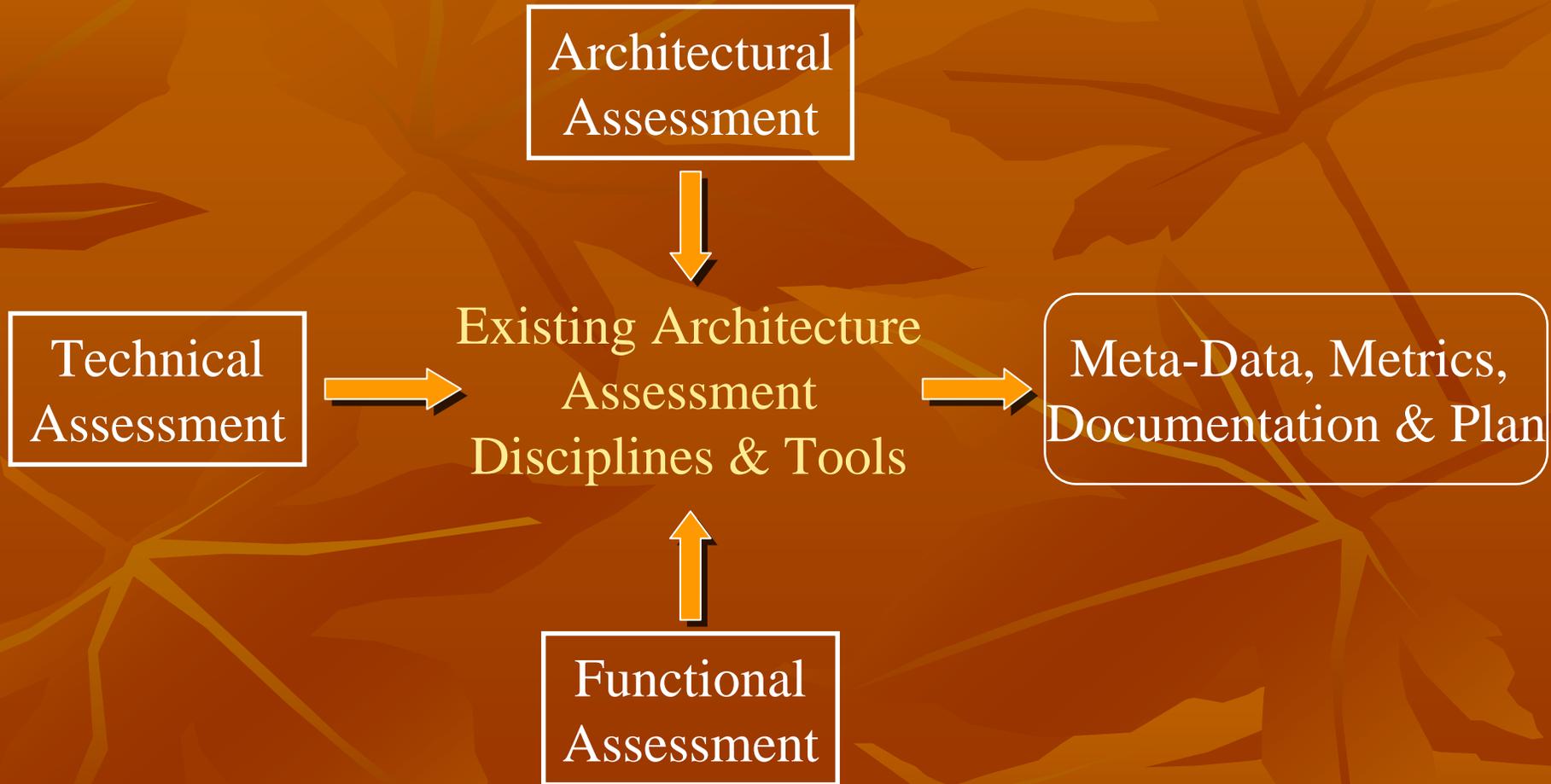
# VI. Project-Level Assessment

## Objectives

- Assess long-term business and information plans
- Identify the best migration strategy to meet short and long-term information requirements
- Eliminate "guess work" typically involved in IT planning efforts
- Establish phased, cost effective migration plan based on current position and future needs

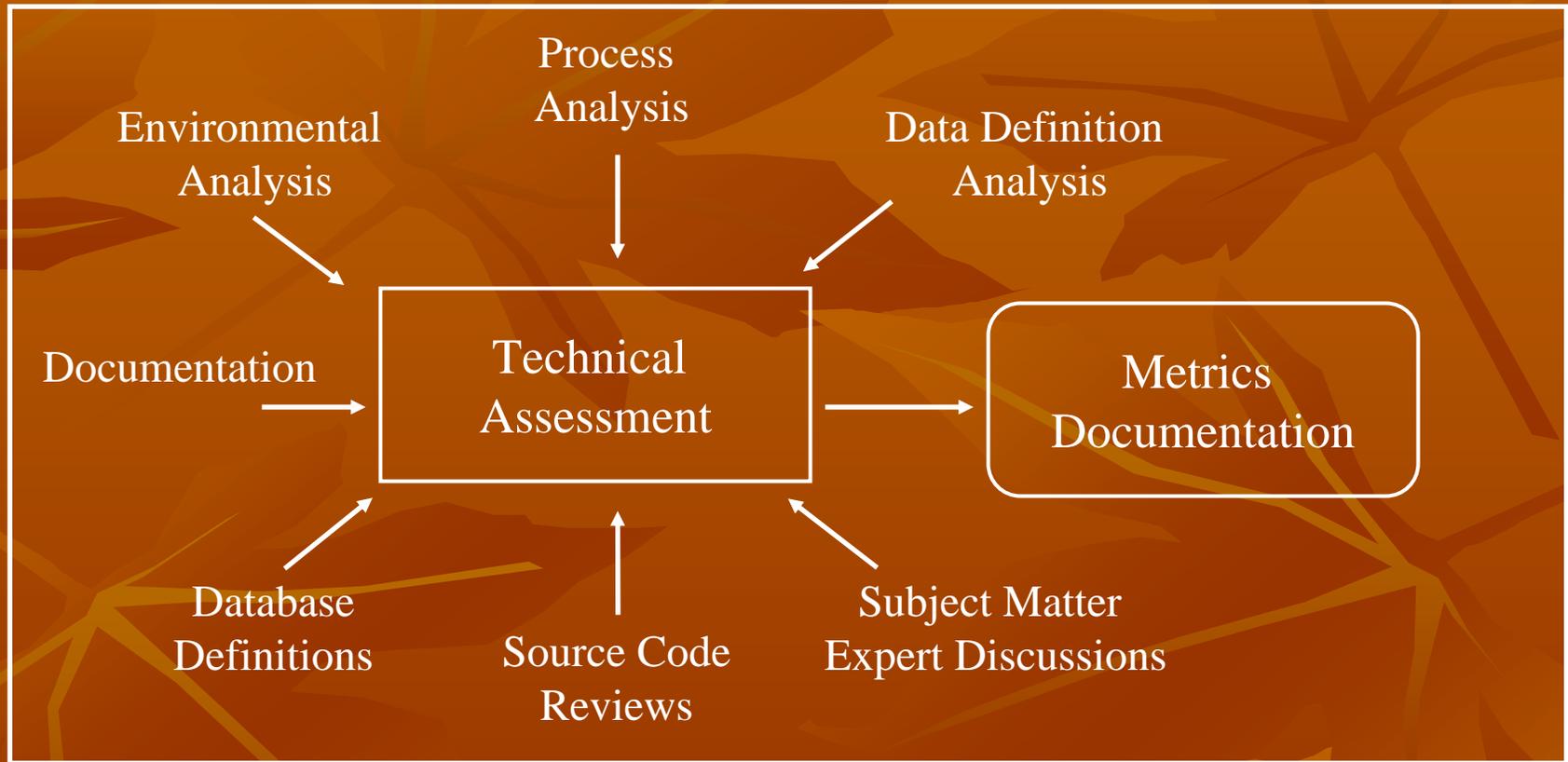
*Note: Assessments leverage Enterprise Assessment*

# Project-Level Assessment Options



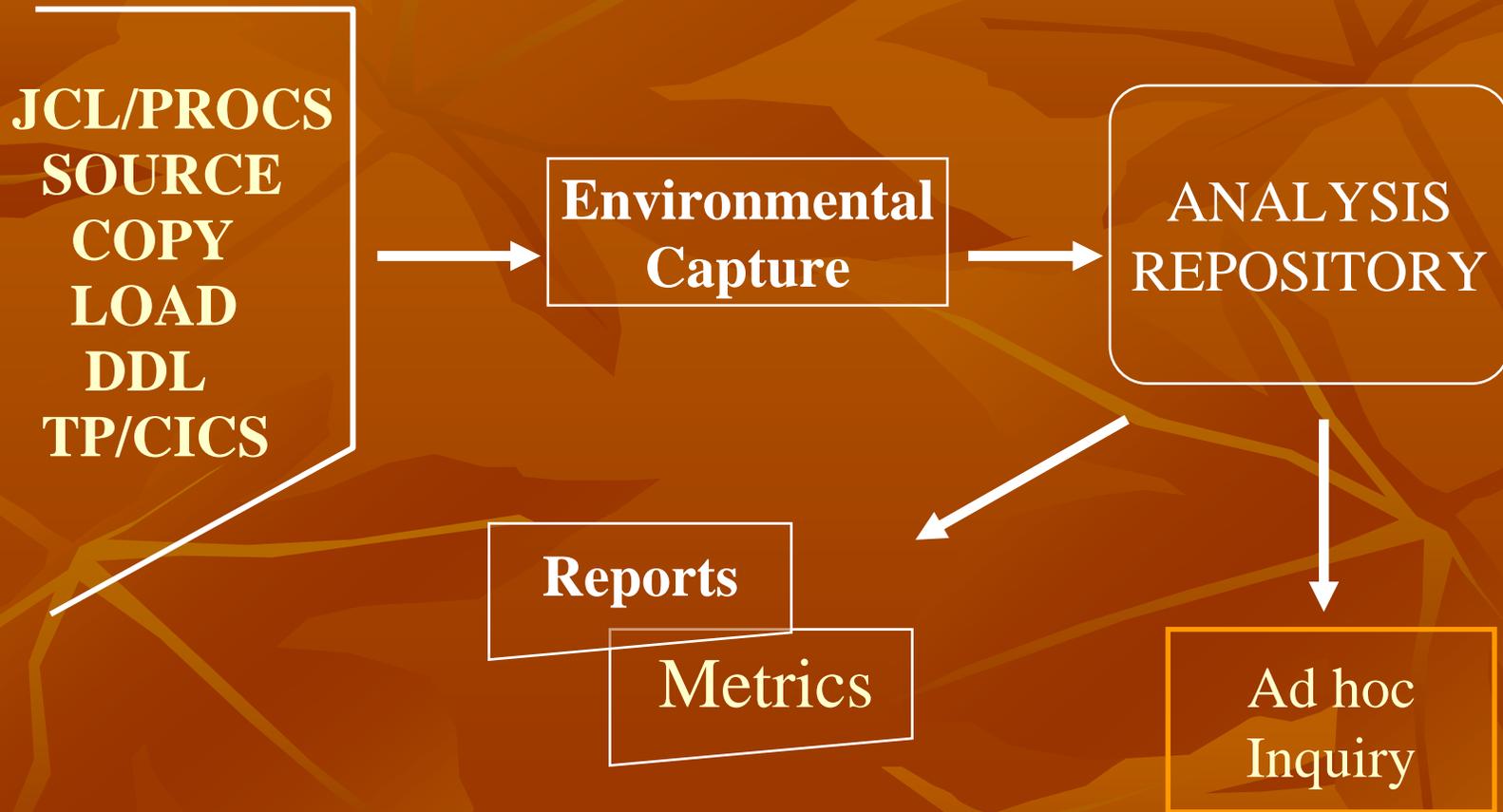
Three sets of tasks are basis for integration and modernization projects. Relies on enterprise assessment.

# Technical Assessment



Technical assessment inventories and documents existing environments, data usage and program flow.

# Environmental Analysis



# Dataset-to-Source Cross-Reference

<b>DATASET NAME</b>	<b>SOURCE PROGRAM</b>	<b>SOURCE LIBRARY</b>	<b>DDNAME</b>	<b>LAYOUT COPY MEMBER</b>	<b>OPEN MODE</b>
PROD.LEDGER.FILE	GLREAD	PROD.SOURCE.COBOL	INDD1	LEDGERIN	INPUT
	GLUPDATE	PROD.SOURCE.COBOL	UPDGL2	LEDGERIN	MIXED
	GLVERFIY	PROD.SOURCE.ASM	VERIFY1	LEDGER	INPUT
PROD.HISTORY.FILE	HIST001	PROD2.SOURCE	HISTOUT	HIST001	OUTPUT
PROD.TRANS.LOG	P001DLOG	UTIL.SOURCE	LOGOUT	LOGFILE	OUTPUT
	P002ULOG	UTIL.SOURCE	LOG	LOGFILE	INPUT
	P003XLOG	UTIL.SOURCE	SYSIN	LOGFILE	MIXED

Sample environmental analysis depicts relationships between physical data and source programs. This type of information expedites research and changes involving data and program definitions.

# Sample Tool Support – System Flow

The screenshot displays the Micro Focus Revolve interface for a project named 'nordic.PRJ'. The top menu bar includes 'Project', 'Tools', 'Browsers', 'Utilities', 'Window', and 'Help'. A toolbar with various icons is located below the menu. The main workspace is divided into three panes:

- Left Pane:** A hierarchical tree view showing the project structure, including folders and sub-items.
- Top Center Pane:** A system flow diagram showing a sequence of modules: 'nordic.PR' calls 'ONLINE', which calls 'ONLINE1', which in turn calls 'ONLINE3', 'ONLINE2', and 'ONLINE7'. The 'ONLINE3' node is highlighted in yellow.
- Bottom Center Pane:** A code editor window showing the source code for 'entry ONLINE1 (ONLINE1.CBL)'. The code includes a return statement and a comment: 'return {ONLINE1.CBL at 853,17}'. Below the code is a table of assembly-like instructions.

The code editor shows the following instructions:

```
211100 SYST-RETURN                SECTION.                21110008
211200 EXEC CICS RETURN                21120008
211300 TRANSID ('PPO3')                21130008
211400 COMMAREA (WS-COMMON-AREA)        21140008
211500 LENGTH (450)                    21150008
211600 RESP (EIBRESP)                  21160008
211700 END-EXEC.                        21170008
211800*                                21180008
211900 IF EIBRESP NOT = DFHRESP(NORMAL) 21190008
212000 PERFORM RETURN-OUT-TO-CICSR     21200008
21210008
21220008
21230008
212400*****21240008
212500* TRANSFER CONTROL TO NEXT PROGRAM(ONLINE2) *21250008
212600*****21260008
212700 XCTL-NEXT-PROGRAM-GAME          SECTION.          21270008
212800 MOVE 'ONLINE1' TO WS-CA-LAST-PROGRAM. 21280008
212900*                                21290008
213000 EXEC CICS XCTL                  21300008
213100 PROGRAM ('ONLINE2')              21310008
213200 COMMAREA (WS-COMMON-AREA)        21320008
213300 LENGTH (450)                    21330008
213400 RESP (EIBRESP)                  21330008
213500 END-EXEC.                        21330008
213600*                                21330008
213700 IF EIBRESP NOT = DFHRESP(NORMAL) 21370008
```

System flow shows program flow

Source: Micro Focus

# Sample Tool Support – System Level Analysis

Micro Focus Revolve - nordic.PRJ - [Project Manager - nordic.PRJ]

Project Tools Browsers Utilities Window Help

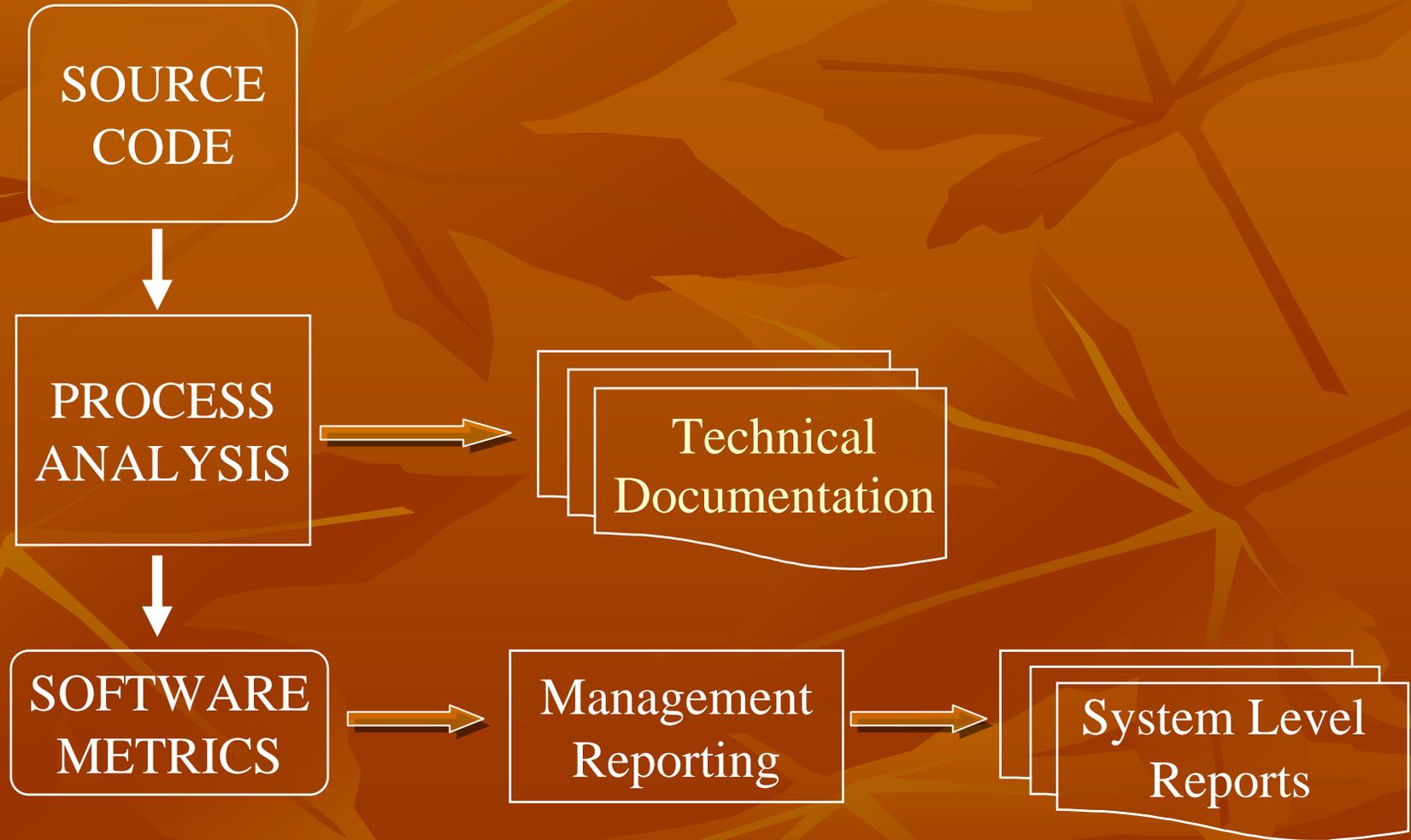
Type \* Match

GAME.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
GAMEMS.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
HEVRIFY.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
MAILLIST.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
PROSPECT.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
PROSTAG.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
SALES.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
TREND.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
WORKFLD.CPY	copybook	Loaded	.. \REVOLVE\SAMPLEAPP\COPYLIB\	07/17/00	12:27:00 AM
cobol files (9)					
BATCH1.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\		
BATCH2.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\		
ONLINE1.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\		
ONLINE2.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\		
ONLINE3.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\		
ONLINE4.CBL	cobol	Loaded	.. \REVOLVE\SAMPLEAPP\COBOL\	07/17/00	12:27:00 AM

System level view of program to program relationships

Source: Micro Focus

# Program Level Analysis



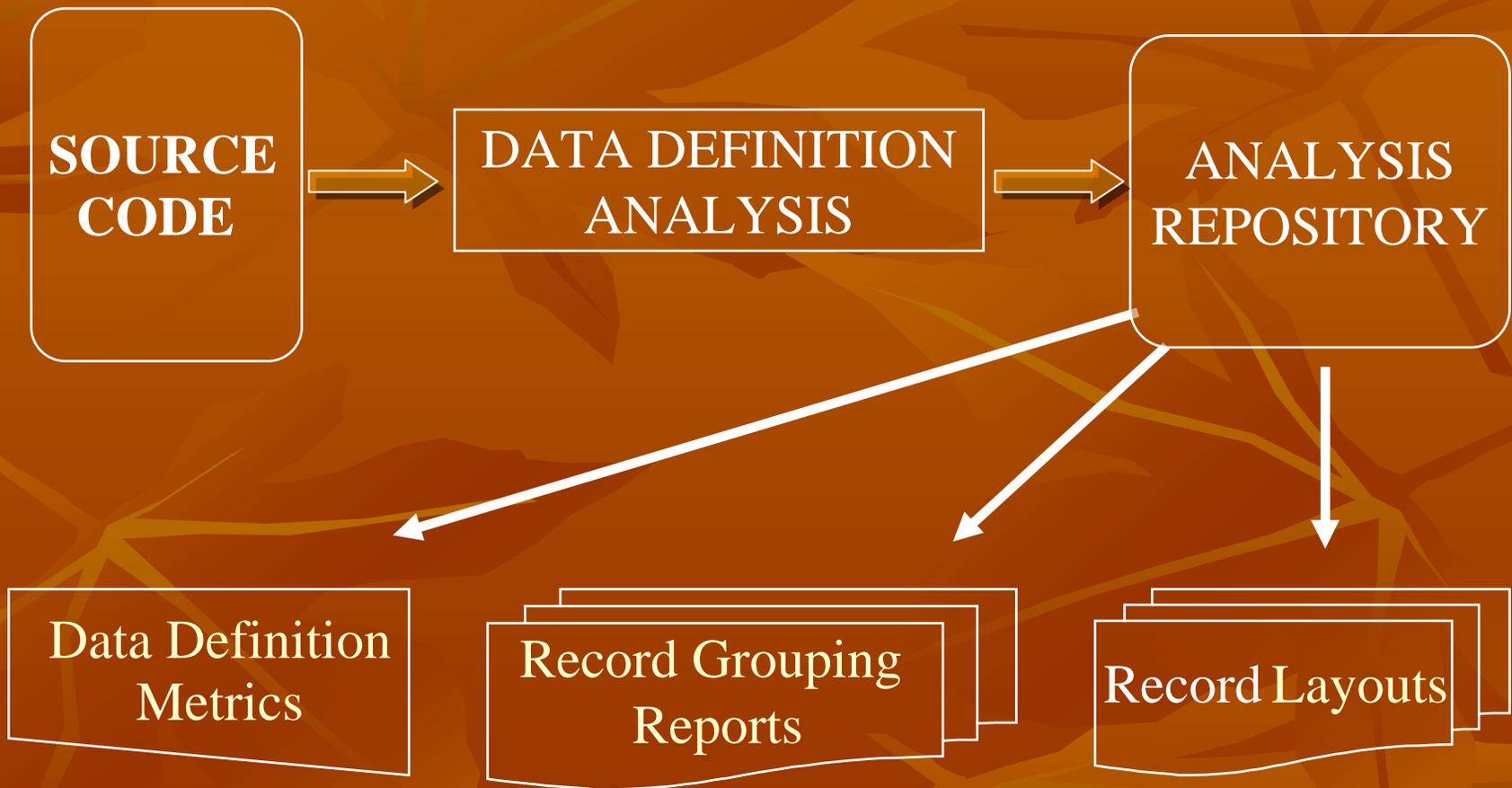
# Program Flow – Path Analysis

LINE	TRANSFER FROM	TRANSFER TO	TRANSFER TYPE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	00786	00786	ENTRY	SAMPLE																			
2	00787	00787	NULL FALL	BEGIN-PROGRAM																			
3	00788	00788	NULL FALL	BEGIN-PROGRAM-010																			
4	00789	01564	PERF	GET-ENTITY-TITLE																			
5	01569	01587	PERF	. 9100-LINK-TO-805																			
6	01588	00000	CALL	. . FAM806																			
7	01573	01589	COND PERF	. 9000-DISPLAY																			
8	01590	00000	CALL	. . FAM800																			
9	01575	01589	COND PERF	. 9000-DISPLAY >7																			
10	01576	01593	COND PERF	. 9900-ABORT-JOB																			
11	01594	00000	CALL	. . ILBOABN0																			
12	01594	00000	STOP	. . NOM RETURNING CALL																			
13	01588	01587	PERF	. 9100-LINK-TO-805 >5																			
14	01584	01589	COND PERF	. 9000-DISPLAY >7																			
15	01586	01593	COND PERF	. 9900-ABORT-JOB >10																			
16	00790	01519	PERF	S100-MAIN-RTN																			
17	01521	00971	PERF	. D000002-XX																			
18	00974	00976	PERF	. . 0220-PROCESS-OPT-B																			
19	00977	01446	PERF	. . . 0680-READ-DETAIL-INPUT																			
20	01448	01450	COND GO TO	. . . . 0685-CLOSE-DETAIL-INPUT																			
21	01454	01595	PERF	. . . . . TEST-FOR-USAM-E1																			
22	01596	01596	NULL FALL	. . . . . TEST-FOR-USAM-E1-A																			
23	01603	01589	COND PERF	. . . . . 9000-DISPLAY >7																			
24	01604	00000	COND CALL	. . . . . ENDIT																			
25	01605	01605	FALL THRU	. . . . . TEST-FOR-USAM-E1-EXIT																			
26	01455	01455	FALL THRU	. . . . . 0690-EXIT																			
27	01449	01455	GO TO	. . . . . 0690-EXIT																			
28	00978	01489	COND GO TO	. . . . . 0800-PRINT-E1-TOTALS 018																			
29	01491	01526	COND GO TO	. . . . . 0850-NO-E1S																			
30	01528	01373	COND PERF	. . . . . 0610-CLEAR-STORAGE																			
31	01380	01380	FALL THRU	. . . . . 0620-EXIT																			
32	01530	01318	COND PERF	. . . . . 0550-PRINT-E1S																			

Source: Legacy Systems: Transformation Strategies

Program flow analysis allows an analyst to quickly determine the overall logic flow of a program slated for an update or modernization.

# Data Definition Analysis



# Data Definition Analysis Process

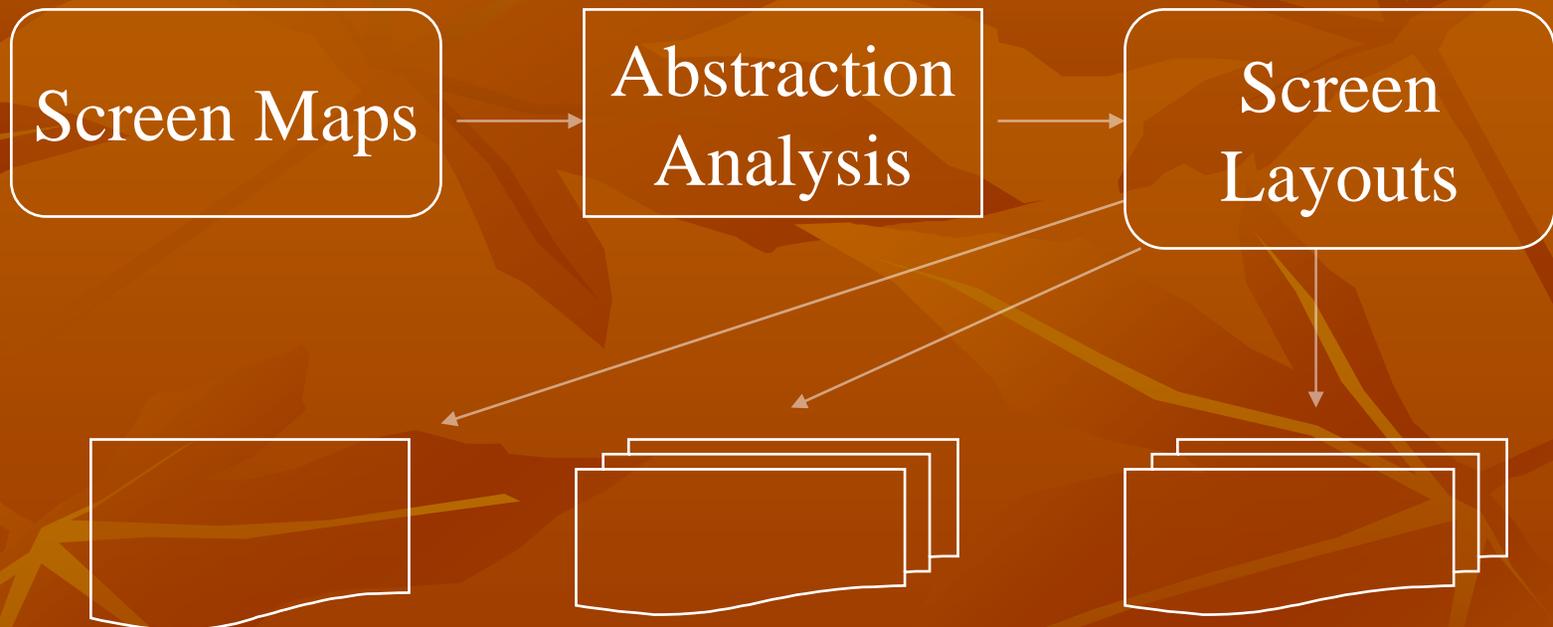
<p><b>Sample I/O record group - Group 005</b></p> <p>PM222100. (Copybook)</p> <p>01 PAY-MAST.          05 MAST-KEY.          10 EMP-NAME.            15 LAST-NAME PIC X(20).            15 FIRST-NAME PIC X(10).            15 MIDDLE-INIT PIC X.          10 EMP-NUMBER PIC 9(6).          10 EMP-BIRTH-DT.            15 EMP-BIRTH-MO PIC 99.            15 EMP-BIRTH-DA PIC 99.            15 EMP-BIRTH-YR PIC 99.          05 DATE-OF-HIRE PIC 9(6).          05 SOC-SEC-NO PIC 9(9).          05 DIVISION-NO PIC 9(3).          05 DEPT-NO PIC 9(3).          05 P-MODE PIC X.          05 YRLY-SAL PIC 9(8).          05 AV-DAYS PIC 99.          05 GROSS-PAY PIC 9(5).          05 NET-PAY PIC 9(5).          05 TOTAL-DED.            10 FED-TAX PIC 9(3).            10 FICA PIC 9(3).            10 SS-TAX PIC 9(3).            10 ST-TAX PIC 9(3).            10 DED-401K PIC 9(3).            10 DED-HEALTH PIC 9(3).          05 YTD-GROSS PIC 9(8).          05 YTD-NET PIC 9(8).          05 FILLER PIC X(13).</p>	<p>PAY-MASTER. (From Pgm PM22400.)</p> <p>01 PAY-MASTER.          05 MASTER-KEY.          10 NAME PIC X(31).          10 NUMBER PIC 9(6).          10 BIRTH-DATE PIC 9(6).          05 FILLER PIC 9(21).          05 MODE PIC X.          05 YR-PAY PIC 9(8).          05 VAC-DAYS PIC 99.          05 GRS-PAY PIC 9(5).          05 NET-PAY PIC 9(5).          05 FILLER PIC 9(52).          05 FILLER PIC X(13).</p>
<p>T-REC. (From Pgm LCT0700.)</p> <p>01 T-REC.          05 T-TYPE PIC X(3).          05 T-DT PIC 9(6).          05 EMP-ID PIC X(15).          05 T-AMT PIC 9(9).          05 FILLER PIC X(117).</p>	<p>MAST-REC. (From Pgm PM22500.)</p> <p>01 MAST-REC.          05 MAST-HEADER.            10 M-NAME PIC X(31).            10 M-NO PIC 9(6).            10 M-DOB PIC 9(6).          05 M-DOH PIC 9(6).          05 M-SS-NO PIC 9(9).          05 M-DIV-NO PIC 9(3).          05 M-DEPT-NO PIC 9(3).          05 M-MODE PIC X.          05 M-YRLY-PAY PIC 9(8).          05 M-VAC-DAYS PIC 99.          05 M-GROSS-PAY PIC 9(5).          05 M-NET-PAY PIC 9(5).          05 FILLER PIC 9(36).          05 M-YTD-GROSS PIC 9(8).          05 M-YTD-NET PIC 9(8).          05 FILLER PIC X(13).</p> <p>In this sample record group of length 150 bytes, T-REC may be excluded from the group, as it clearly defines a different physical file. The other three records should remain in this I/O record group.</p>



Record Grouping Analysis

Data definition analysis - essential to a variety of projects.

# On-line Presentation Layer Analysis



- Reverse screen online user views
- Assess potential on-line design reuse

# Sample Tool Support - I/O Analysis

The screenshot displays the Micro Focus Revolve interface. The main window, titled 'Screen - PP1.BMS', shows a terminal window with the following text:

```
ONL0001      HOME BUYING NETWORK
              PROD VERIFICATION MENU
              ENTER VERIFY CODE: 99 X

              VERIFICATION MENU

01 = VERIFY SALES MAJOR LINES
02 = VERIFY PACKAGE CONTENT
03 = VERIFY ORDER INFORMATION
04 = VERIFY RANK ADJUSTMENT
05 = SALES RANKING VERIFICATION
06 = MAILING ADDRESS VERIFICATION
```

The 'Screens' window on the right lists several screens, including:

- HEBVERIFY of PP3.BMS
- ONL0001 of PP1.BMS
  - ONLINE1.CBL
    - ONLINE1.CBL Line 679: 073600 EXEC CICS RECEIVE
    - ONLINE1.CBL Line 723: 198200 EXEC CICS SEND
    - ONLINE1.CBL Line 732: 199100 EXEC CICS SEND
- ONL0005 of PP4.BMS
- ONL0006 of PP7.BMS
- ONL0007 of PP5.BMS
- ONL0008 of PP2.BMS
- ONL1001 of PP8.BMS
- ONL2002 of PP9.BMS
- ONL3003 of PP10.BMS

The bottom portion of the terminal window shows COBOL code for verification screens:

```
ATTRB=(ASKIP,NORM)
VERIFY DFHMD F LENGTH=035,POS=(007,003),
        INITIAL='01 = VERIFY SALES MAJOR LINES
        ATTRB=(ASKIP,NORM)
VERIFY DFHMD F LENGTH=035,POS=(009,003),
        INITIAL='02 = VERIFY PACKAGE CONTENT
        ATTRB=(ASKIP,NORM)
VERIFY DFHMD F LENGTH=044,POS=(013,003),
        INITIAL='04 = VERIFY RANK ADJUSTMENT
        ATTRB=(ASKIP,NORM)
VERIFY DFHMD F LENGTH=037,POS=(015,003),
        INITIAL='05 = SALES RANKING VERIFICATION
        ATTRB=(ASKIP,NORM)
VERIFY DFHMD F LENGTH=035,POS=(017,003),
        INITIAL='06 = MAILING ADDRESS VERIFICATION
        ATTRB=(ASKIP,NORM)
```

The bottom right of the terminal window shows assembly code:

```
073500
073600 EXEC CICS RECEIVE
073700 MAP ('ONL0001')
073800 INTO (ONL0001I)
                                (BRESP)
074000 IF BRESP NOT = DFHRESP(NORMAL)
074300 PERFORM RETURN-OUT-TO-CICSE.
074400
074500 IF (RIBAID = DFHPF3)
074600 OR (RIBAID = DFHENTER)
074700 NEXT SENTENCE
074800 ELSE
074900 MOVE -1
075000 MOVE WS-PROS-MSG-5 TO
```

Tool support for user interface analysis

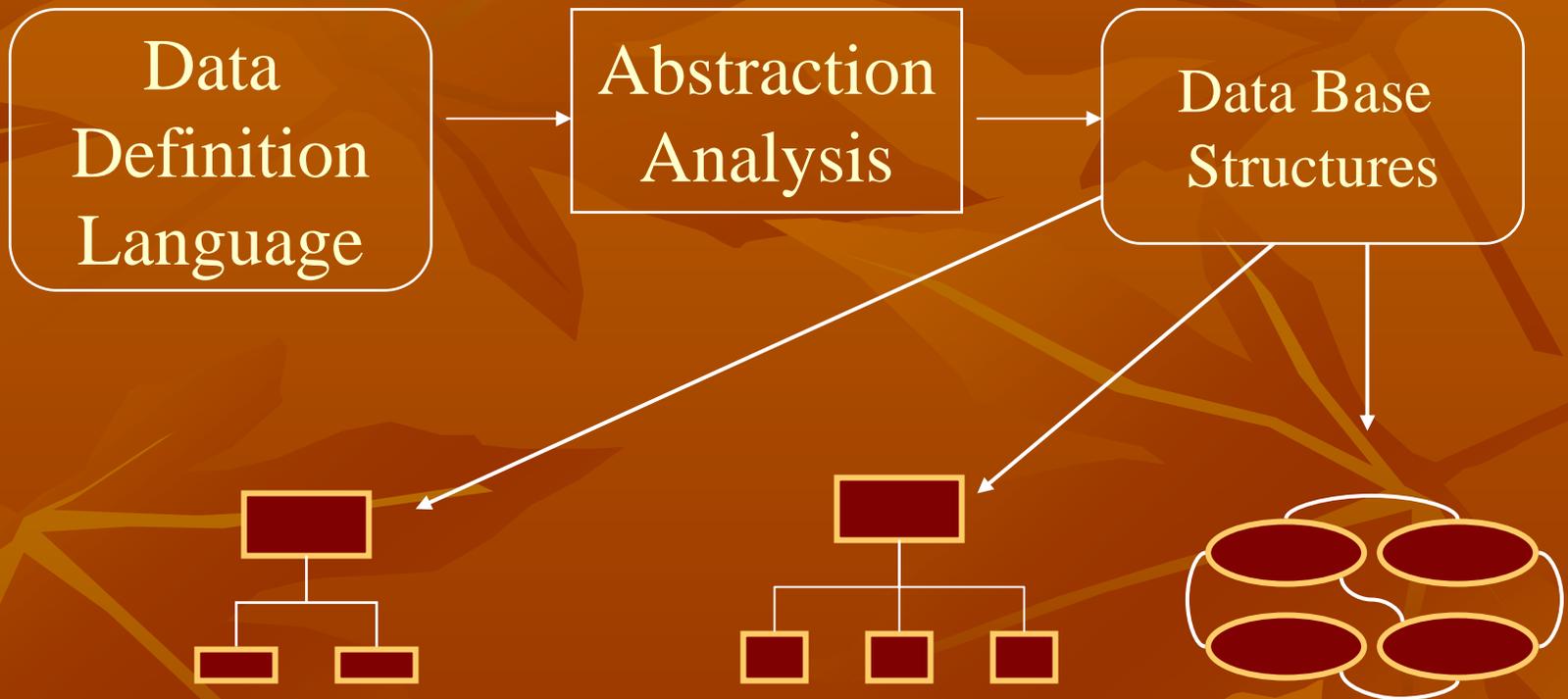
Source: Micro Focus

# Batch Presentation Layer Analysis



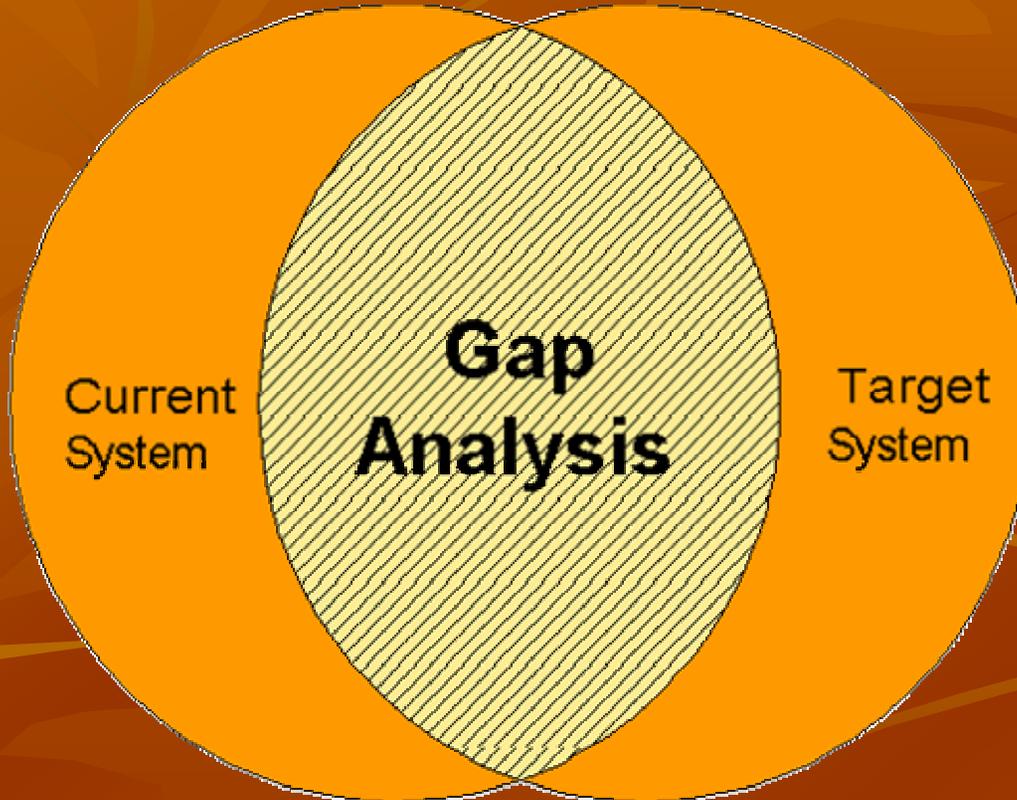
- Reverse engineer reports
- Assess report design reuse

# Data Access Layer Analysis



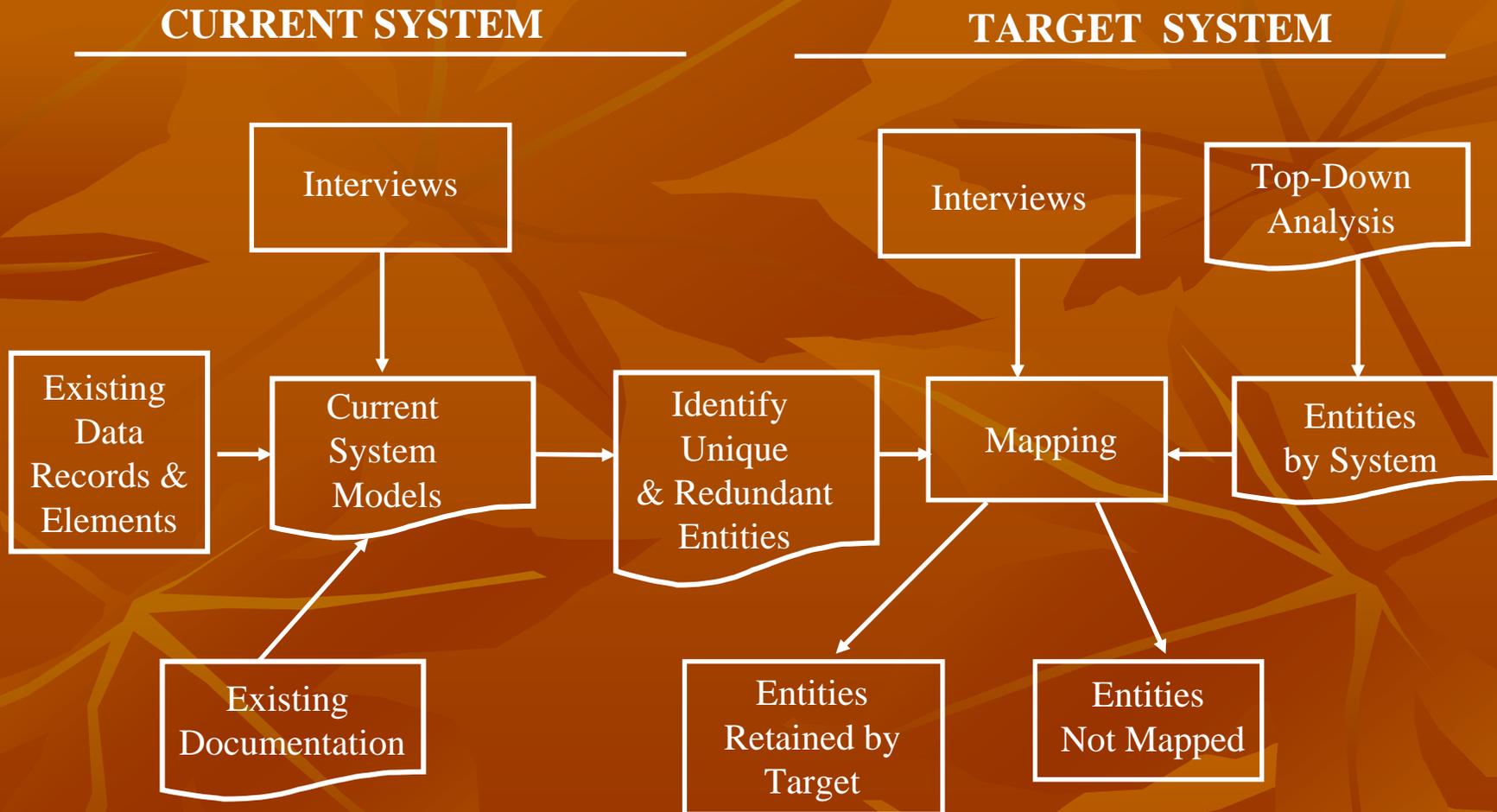
- Depicts existing data usage structures
- Input to data redesign and migration planning

# Functional Assessment: Gap Analysis



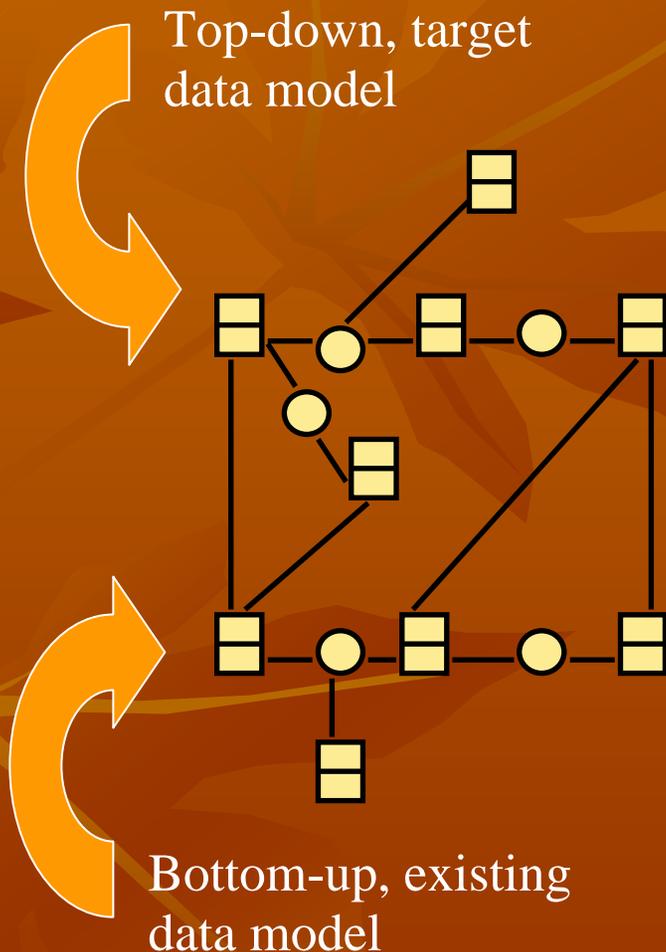
- Assess the potential for data / rule reusability
- Determine % of target existing in current systems
- Determine % of target to be developed from scratch

# Current-to-Target Data Entity Mapping



Current-to-target entity mapping extracts bottom-up entities and attributes to create or validate target data models.

# Building a High Integrity Data Architecture

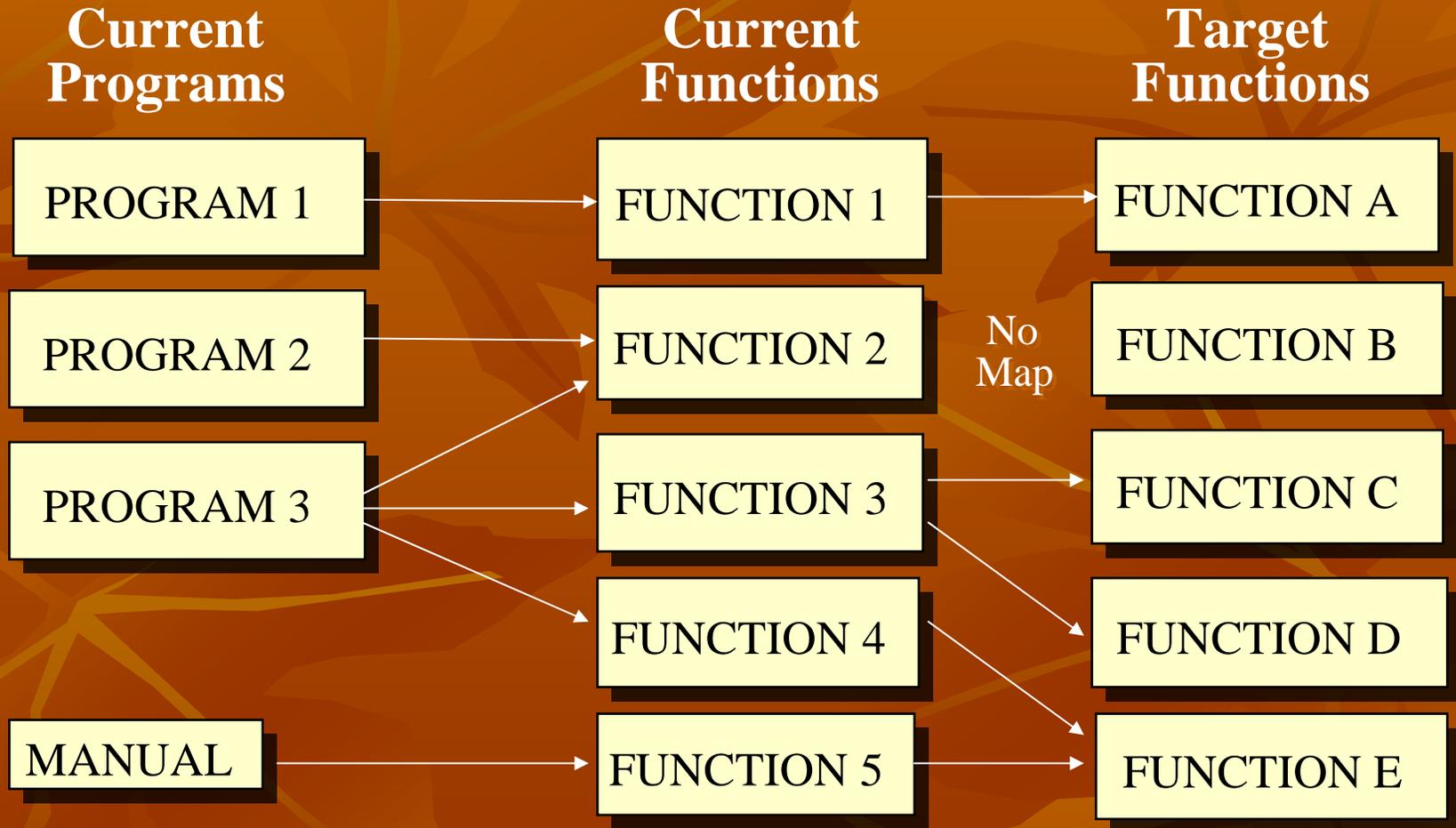


## Deliverables

- Integrated top-down, bottom-up data model
- Gap analysis determining missing data entities & attributes
- Reconciliation of inconsistent data usage
- An understanding of existing systems to target data migration requirements

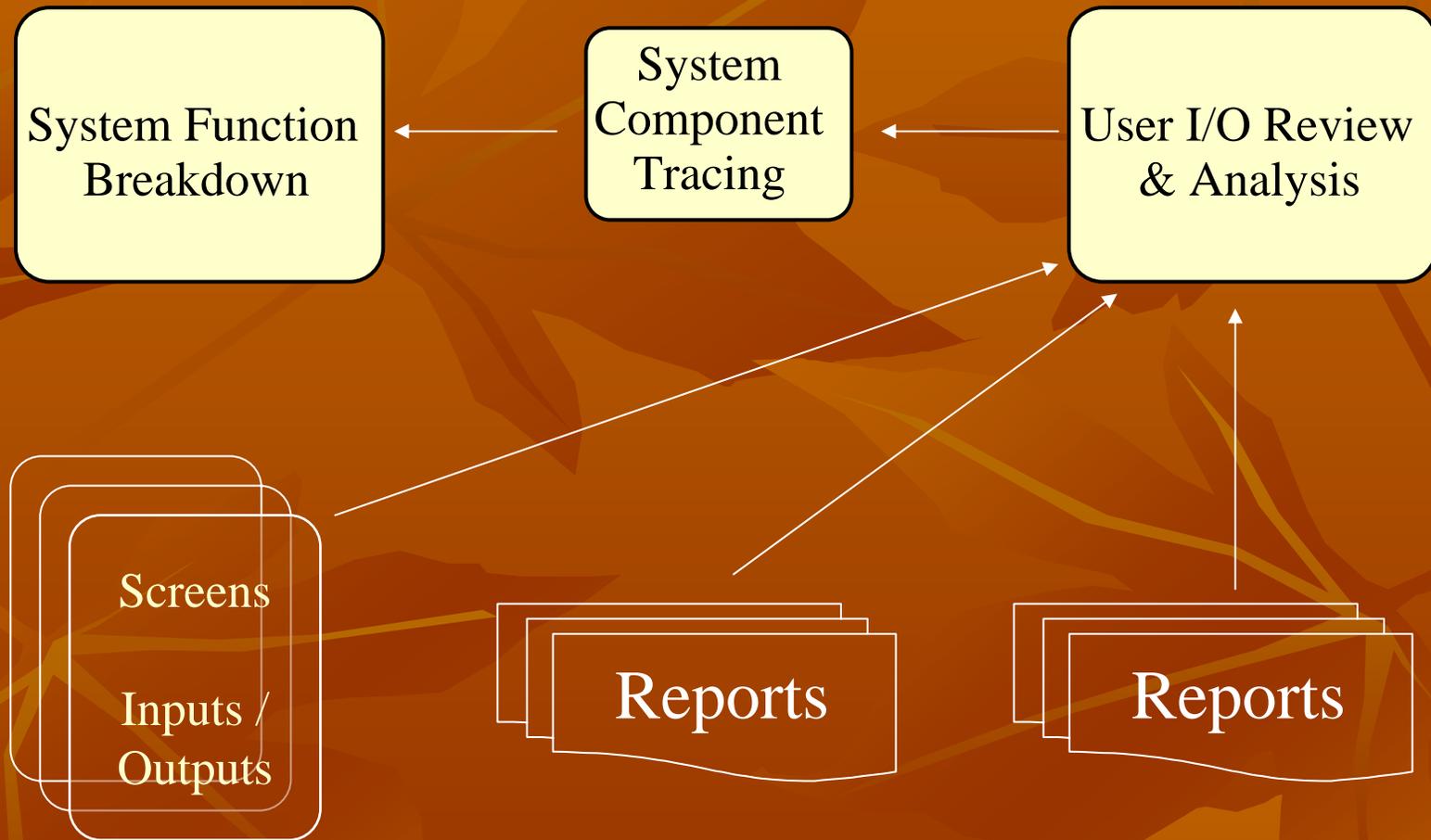
Consolidating top-down / bottom-up data models produces a higher quality data architecture results.

# Program / Function GAP Analysis



Program-to-function mapping helps determine level of existing application conformance to target architecture.

# Reverse Requirements Tracing



- Delivers data flow diagram of current application
- Serves as current systems documentation and as basis for functional model development

# Reverse Requirements Tracing of Functions

**FORM 004 - Create current function / process hierarchy - establish current functions**

PROGRAM SOURCE	CURRENT PROCESS	CURRENT FUNCTION	TARGET FUNCTION
		Purchasing	
		Sending/Receiving	
		Inventory	
		Manufacturing	
		Financial Services	
		Etc...	

Purchasing Department  
Vendor / Product Report      Date

Vendors/Products Added  
xxxxxxx 999999 xxxxxxxx 99/99/99

Vendors/Products Changed  
xxxxxxx 999999 xxxxxxxx 99/99/99

Vendors/Products Discontinued  
xxxxxxx 999999

*Source: USRM / Comsys*

Reverse requirements tracing determines application functionality by tracking user views back into the source code.

# Program / Process / Function Map

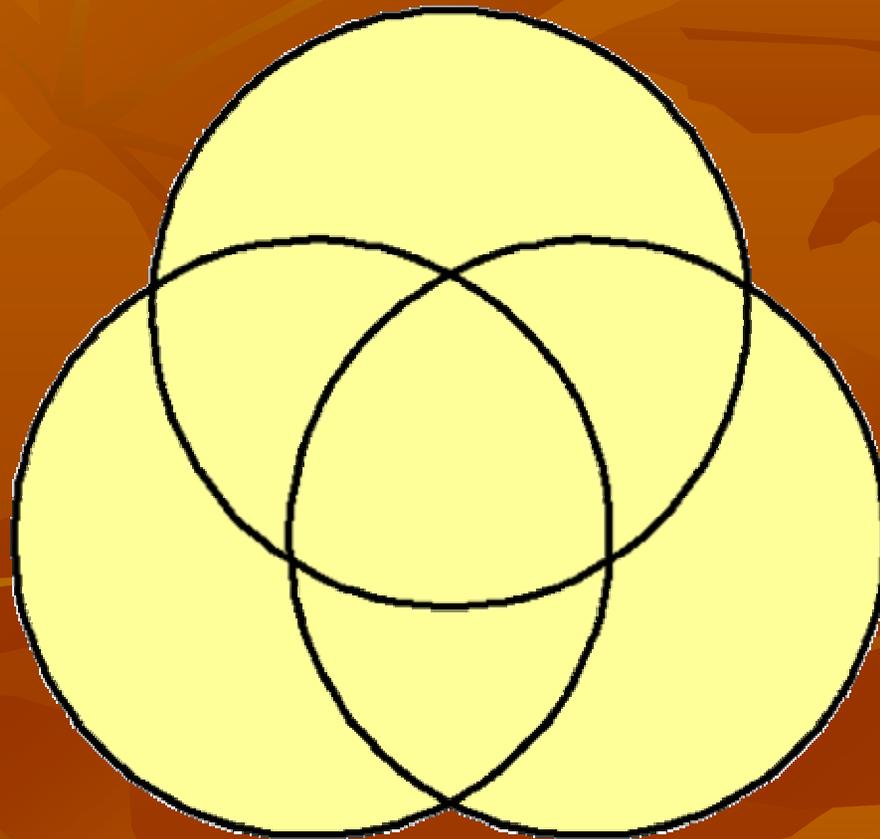
Mapping functions to program source modules			
PROGRAM SOURCE	CURRENT PROCESS	CURRENT FUNCTION	TARGET FUNCTION
PU300200	Select vendor / product	Purchasing	
PU300200	Change vendor / product	Purchasing	
PU300200	Roll off vendor / product	Purchasing	
PU300520	Order supplies	Purchasing	
PU500000	Pay vendors	Purchasing	
SR774200	Track vendor shipments	Sending/Receiving	
SR600350	Disperse supplies	Sending/Receiving	
INV08040	Maintain supply stock	Inventory	
Manual	Maintain supply stock	Inventory	
INV00100	Request supplies	Inventory	
SR400100	Ship corporate products	Sending/Receiving	
SR400200	Ship corporate products	Sending/Receiving	
MA906600	Manage materials	Manufacturing	
Manual	Manage materials	Manufacturing	
MA665000	Request supplies	Manufacturing	
MA240030	Create products	Manufacturing	
MA240050	Create products	Manufacturing	

Current-to-target functional analysis maps existing programs, processes and functions to target architecture functional definitions.

# Three-way vs. Two-way Mapping

Requirements

Existing  
Systems



Package

Current-to-target-to-proposed mapping supports package assessments.

# Implementation Project Planning

- Project-level analysis establishes detailed documentation and implementation plan
- Essentially becomes first steps in a modernization project

# VII. Application & Data Migration, Structuring & Rationalization

- Improve source code quality so that it is easier to analyze, modify, debug and test
- Prepare systems for design level and physical component level reuse within target architectures

# Application Management & Maintenance Facilitation Scenarios

## Data Integrity

- Mining
- Cleanup
- Consolidation

## Application Improvement

- Structuring
- Rationalization
- Modularization

Existing  
Applications  
&  
Data

## Application Documentation

Documentation, improvement and data integrity scenarios streamline management and upgrade tasks.

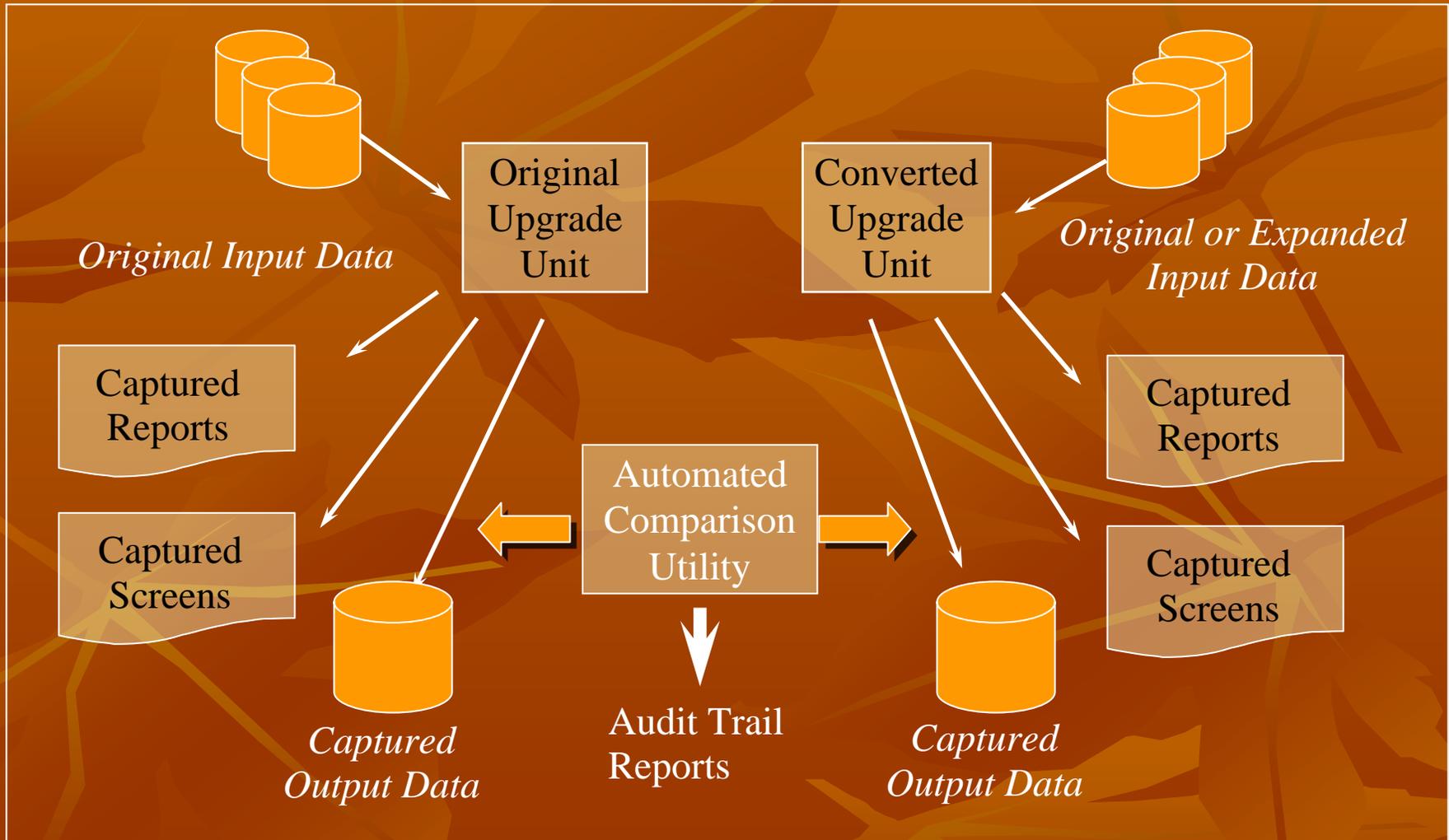
# Application Staging

- Establish working versions of production source code
- Establish Positioning configuration management procedures
- Facilitate change control procedures to manage and assimilate production upgrades

# Validation

- Verifies logical equivalence of retooled source code
- Involves executing comparisons to verify that no functional changes were inadvertently introduced
- It is NOT testing - testing looks for differing results, validation looks for equivalency
- Requires 60 - 70% execution coverage and good test suites

# Validation Process



Validation testing attempts to automate as much of the process as possible.

# Sample Tool Support – Execution Simulation

The screenshot displays the Micro Focus Revolve IDE. The main window shows a code editor with a list of statements and their addresses. A callout box highlights the simulation environment. A dialog box titled 'Execution Simulator Branch Choices' is open, showing a specific conditional statement.

**Execution simulation simulates testing in a static environment**

Address	Statement	Address	
037600	MAINLINE	SECTION.	09070009
037700	IF EIBCALEN > 0	09080009	
037800	MOVE DFHCOMMAREA TO WS-COMMON-AREA.	09090009	
037900*		09100009	
038000	IF WS-CA-LAST-PROGRAM = 'ONLINE1'	09110009	
		09120009	
		09130009	
		09140009	
		09150009	
		09160009	
		09170009	
		09180009	
		*****09190009	
		*09200009	
		*****09210009	
		09220009	
		09230009	
		09240009	
		09250009	
		09270009	
		09280009	
		09290009	
		09280009	
039700*		10620009	
039800	START-SCREEN-PROCESSA-EXIT. EXIT.	10630009	
039700*		10640009	
053100	READ-FILE-KEY-PREV	SECTION.	10650009
053200	MOVE 4 TO NDX-1.	10660009	
053300	MOVE WS-CA-PREV-ALPHA TO WS-CA-NEXT-ALPHA.	10670009	
053400	MOVE WS-CA-PREV-ALPHA TO STATE-KEY.	10680009	
053500*			
053600	EXEC CICS STARTER		
053700	FILE ('MAILXREF')		

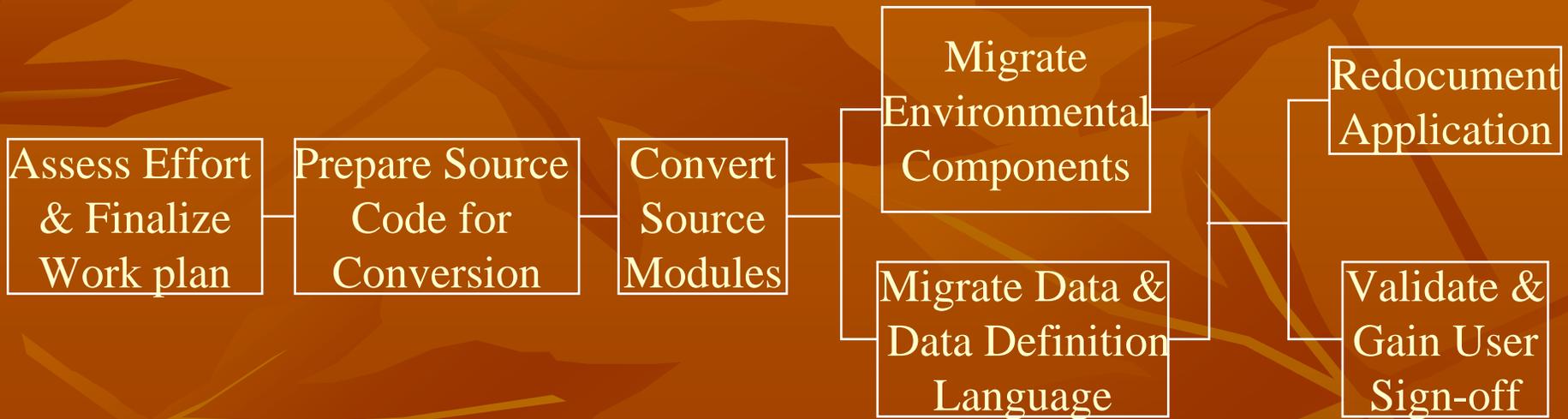
**Execution Simulator Branch Choices**

```
037700 IF EIBCALEN > 0
```

Buttons: Help, OK, Cancel

Source: Micro Focus

# Conversions & Language Level Upgrades



- Focus is on 2nd to 3rd, 3rd to 3rd or 4th to 3rd generation language conversion
- Does not address paradigm shift to objects

# In-Line Program Cross-Reference

```
00973          MOVE 1 TO LINE-ERROR
00974          PERFORM 0220-PROCESS-OPT-B.                                00976
00975  511100*

-----*
* . GO TO CONDITIONAL FROM 00906 0181-OLD-HIST-EOF.                    *
* . PERFORMED BY 00957 0200-F18TM-READ-NEXT, 00971 D000002-XX.        *
* . BEGIN GO TO LOOP RETURNING FROM 01000 0250-CONTINUE.              *
* . PERFORM RANGE VIOLATION TO 01489 0000-PRINT-E1-TOTALS.            *
-----*

00976  511800 0220-PROCESS-OPT-B.

00977  512500 PERFORM 0600-READ-DETAIL-INPUT THRU 0690-EXIT.            01446 01455
00978  513200 IF OPT-A GO TO 0800-PRINT-E1-TOTALS.                      01489
00979  513900*

-----*
* . BEGIN GO TO LOOP RETURNING FROM 00957 0200-F18TM-READ-NEXT.        *
* . FALL THRU FROM 00976 0220-PROCESS-OPT-B.                          *
* . PERFORM THRU 00999 0240-EXIT BY 01030 0200-OPT-C-E1, 01052 0300-IS-U1-ON-FILE, *
*   01478 0260-OPT-C-READ.                                             *
-----*

00980  514600 0230-VERIFY-DETAIL-INPUT.

00981  515300 IF DP-ID NOT = '001'
00982  516000 MOVE 1 TO E1-NO (1) ADD 1 TO E1-SWITCH.
00983  516700 IF DP-SORT-CODE NOT = 'B'
00984  517400 MOVE 1 TO E1-NO (2) ADD 1 TO E1-SWITCH.
00985  518100 IF OPT-B
00986  518800 IF DP-SEQ IS NOT NUMERIC
00987  519500 MOVE 1 TO E1-NO (3) ADD 1 TO E1-SWITCH.
00988  520200 IF OPT-B
00989  520900 MOVE DP-SEQ TO CK-CRD-SEQ.
00990  521600 IF DP-AMOUNTX = SPACES OR ZEROS NEXT SENTENCE
00991  522300 ELSE IF DP-AMOUNT IS NUMERIC
00992  523000 MOVE DP-AMOUNT TO CE-PAY-AMT
00993  523700 MOVE 1 TO AMOUNT-SWITCH
00994  524400 ELSE MOVE 1 TO E1-NO (8) ADD 1 TO E1-SWITCH
00995  525100 MOVE DP-AMOUNTX TO CE-PAY-AMTXX.
00996  525800 MOVE DP-DOCUMENT TO CK-DOC-NO.
00997  526500 IF DP-FILLER NOT = SPACES
```

Source: *Legacy Systems:  
Transformation Strategies*

In-line cross-reference reports ensure that source code analysis reflects all explicit or implicit references to program logic.

# Sample Tool Support – Program Flow

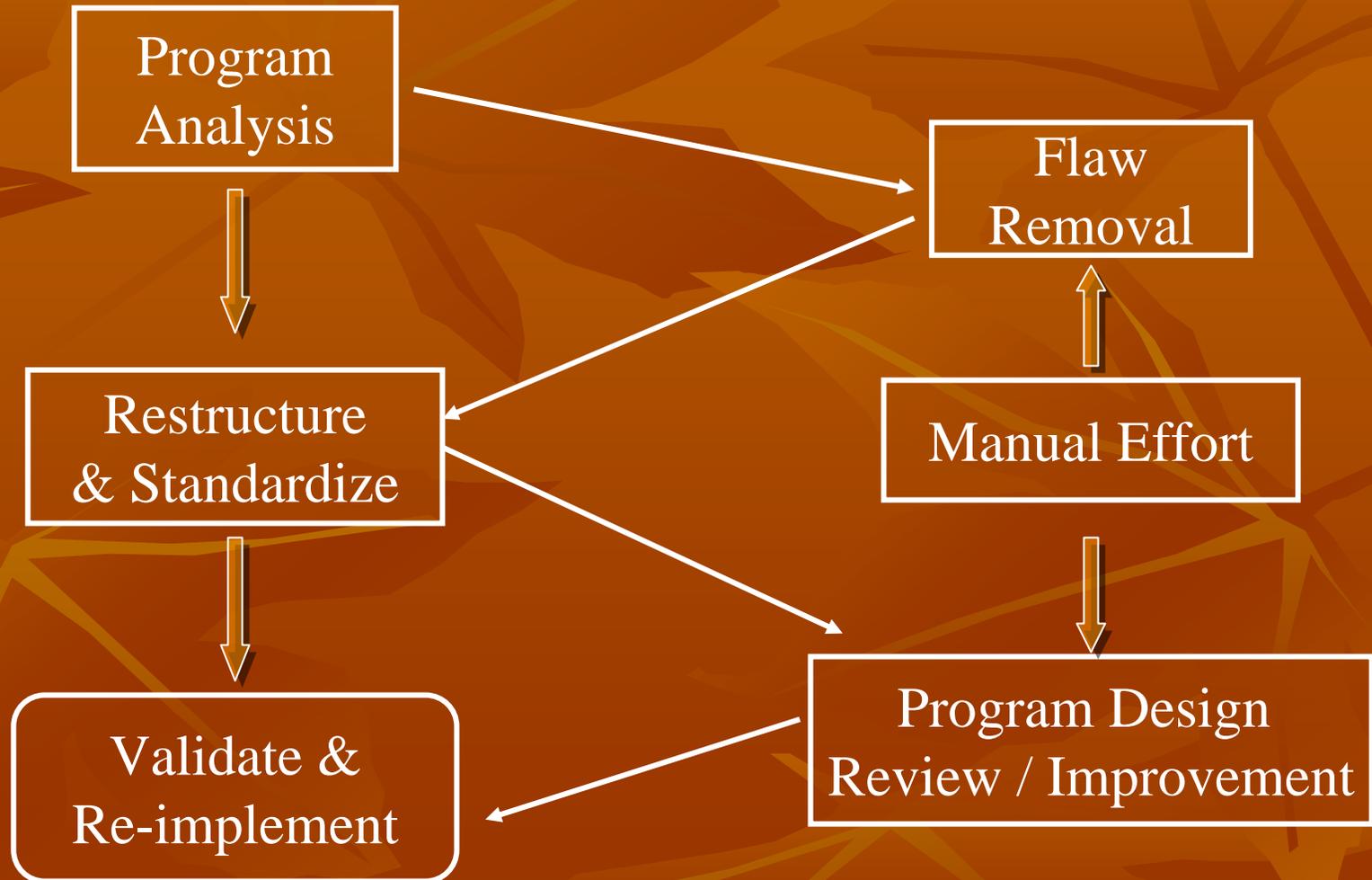
The screenshot displays the Micro Focus Revolve interface. The top window shows a code flow diagram for a program named 'mainline'. The diagram consists of several interconnected boxes representing program sections: 'mainline', 'main-receive', 'main-return', 'error-proces', 'send-main-ds', and 'return-to-pr'. Arrows indicate the flow of execution between these sections. A callout box on the right side of the diagram contains the text: 'System-wide and program-level view of logic'. Below the diagram, the COBOL source code is visible, with line numbers ranging from 215100 to 217000. The code includes sections for 'RETURN-TO-PREVIOUS1' and 'ONLINE4', and 'ONLINE5'. The bottom right corner of the screenshot features a white box with the text: 'Source: Micro Focus'.

System-wide and program-level view of logic

```
215100*****21480006
215200 RETURN-TO-PREVIOUS1          SECTION.          21490006
215300   IF WS-CA-XCTL-FROM = 'ONLINE4'              21500006
215400     EXEC CICS XCTL                             21510006
215500       PROGRAM ('ONLINE4')                      21520006
215600       COMMAREA (WS05-COMMON-AREA)              21530006
215700       LENGTH (1200)                            21540006
215800       RESP (EIBRESP)                           21550006
215900     END-EXEC.                                  21560006
216000*                                              21570006
216100   IF EIBRESP NOT EQUAL DFHRESP(NORMAL)        21580006
216200     PERFORM ERROR-PROCSING-MAIN.                21590006
216300*                                              21600006
216400   IF WS-CA-XCTL-FROM = 'ONLINE5'              21610006
216500     EXEC CICS XCTL                             21620006
216600       PROGRAM ('ONLINE5')                      21630006
216700       COMMAREA (WS05-COMMON-AREA)              21640006
216800       LENGTH (1200)                            21650006
216900       RESP (EIBRESP)                           21660006
217000     END-EXEC.                                  21670006
```

Source: Micro Focus

# Restructuring & Program Stabilization



# Runaway Logic - Program Anomaly

## Runaway logic path due to forced Abend

MOVE X TO FIELD-3.  
IF SEVERE-CONDITION-FOUND  
    COMPUTE X = 1/0.     (*Causes Abend*)  
MOVE Y TO FIELD-4.

## Runaway logic path eliminated

MOVE X TO FIELD-3.  
IF SEVERE-CONDITION-FOUND  
    COMPUTE X = 1/0  
    GOBACK. (*Terminates program*)  
MOVE Y TO FIELD-4.

Source: USRM / Comsys

Runaway logic can cause a program failure and result in the capture and reuse of errant business logic.

# Switches Can Mask Business Logic

## Unwanted switch variable

```
IF PROC-TYPE = 312 AND SEX = 'M'  
  MOVE 'G' TO ERR-SW  
  MOVE MAST-HEADER TO ERR-HEADER  
  MOVE PROC-TYPE TO ERR-REC  
  MOVE SEX TO ERR-REC  
ELSE  
  MOVE PROC-TYPE TO MAST-PROC.  
PERFORM PROCESS-REC.  
PROCESS-REC.  
MOVE ...  
MOVE ...  
COMPUTE ...  
IF ERR-SW = 'G'  
  PERFORM GENDER-ERR  
  ELSE ...  
GENDER-ERR.  
MOVE ...  
ADD 1 TO ERR-CT.  
WRITE ERR-REC.
```

## Unwanted switch variable removed

```
IF PROC-TYPE = 312 AND SEX = 'M'  
  (Switch usage and definition removed)  
  MOVE MAST-HEADER TO ERR-HEADER  
  MOVE PROC-TYPE TO ERR-REC  
  MOVE SEX TO ERR-REC  
ELSE  
  MOVE PROC-TYPE TO MAST-PROC.  
PERFORM PROCESS-REC.  
PROCESS-REC.  
MOVE ...  
MOVE ...  
COMPUTE ...  
IF PROC-TYPE = 312 AND SEX = 'M'  
  PERFORM GENDER-ERR  
  ELSE ...  
GENDER-ERR.  
MOVE ...  
ADD 1 TO ERR-CT.  
WRITE ERR-REC.
```

Source: USRM / Comsys

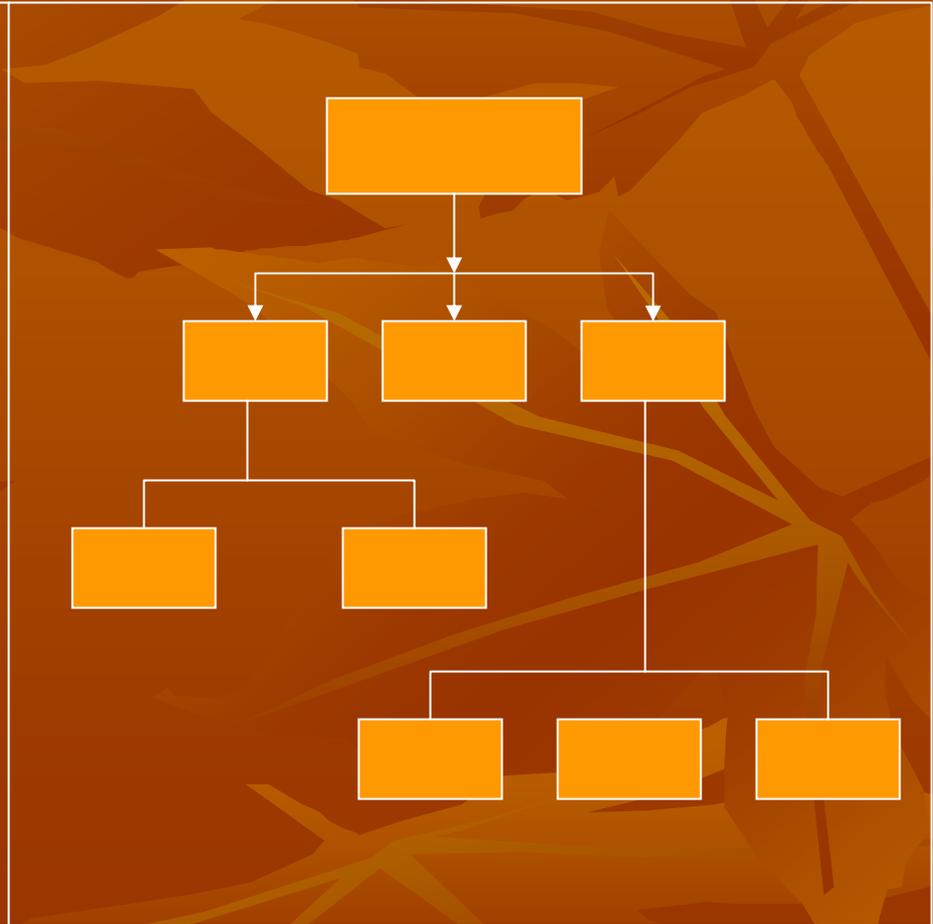
Switches can mask business logic. This should be addressed in business critical applications targeted for an upgrade or business rule reuse.

# Code Structuring

UNSTRUCTURED



STRUCTURED

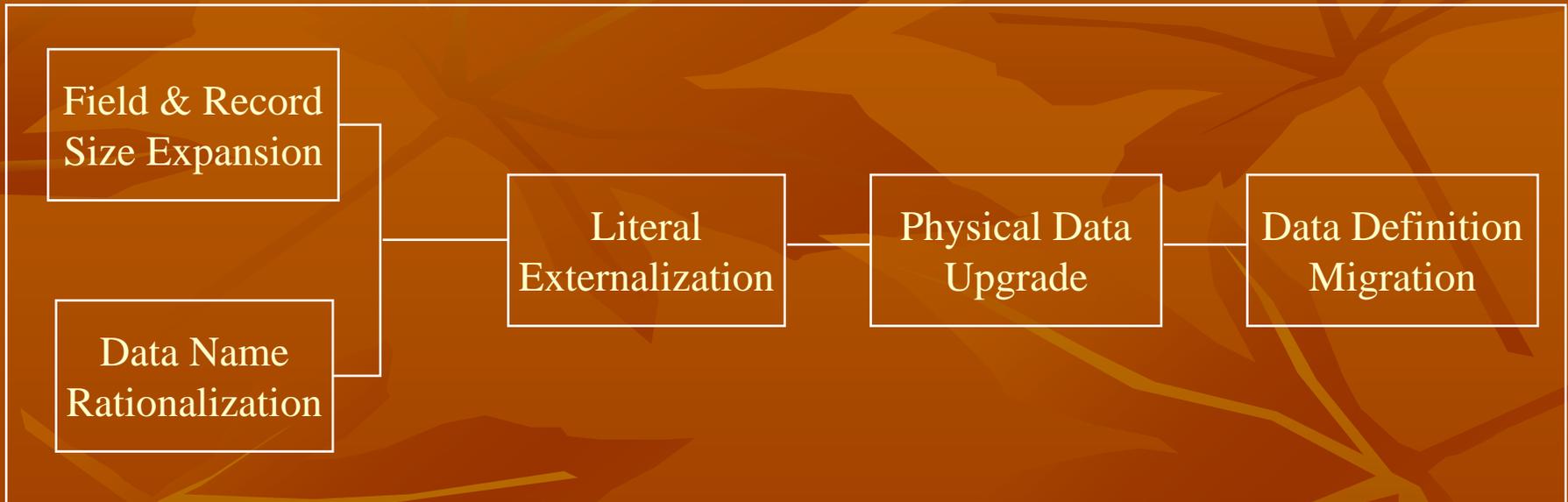


Restructuring of COBOL is highly automated.

# Program Level Design Review & Improvement

- Redundancy reduction
- Switch removal
- Name cleanup
- Spurious decision logic cleanup

# Data Definition Rationalization



- Simplifies system-wide data tracing and analysis
- Externalizes data usage and representations
- Eliminates definitional redundancy and inconsistency
- Facilitates most strategic migration efforts

# Input Code – Before Rationalization

## Example

Sample I/O record group - Group 005

PM222100. (Copybook)

```

01 PAY-MAST.
  05 MAST-KEY.
    10 EMP-NAME.
      15 LAST-NAME PIC X(20).
      15 FIRST-NAME PIC X(10).
      15 MIDDLE-INIT PIC X.
    10 EMP-NUMBER PIC 9(6).
    10 EMP-BIRTH-DT.
      15 EMP-BIRTH-MO PIC 99.
      15 EMP-BIRTH-DA PIC 99.
      15 EMP-BIRTH-YR PIC 99.
  05 DATE-OF-HIRE PIC 9(6).
  05 SOC-SEC-NO PIC 9(9).
  05 DIVISION-NO PIC 9(3).
  05 DEPT-NO PIC 9(3).
  05 P-MODE PIC X.
  05 YRLY-SAL PIC 9(8).
  05 AV-DAYS PIC 99.
  05 GROSS-PAY PIC 9(5).
  05 NET-PAY PIC 9(5).
  05 TOT-DED.
    10 FED-TAX PIC 9(3).
    10 FICA PIC 9(3).
    10 SS-TAX PIC 9(3).
    10 ST-TAX PIC 9(3).
    10 DED-401K PIC 9(3).
    10 DED-HEALTH PIC 9(3).
  05 YTD-GROSS PIC 9(8).
  05 YTD-NET PIC 9(8).
  05 FILLER PIC X(13).
```

T-REC. (From Pgm LCT0700.)

```

01 T-REC.
  05 T-TYPE PIC X(3).
  05 T-DT PIC 9(6).
  05 EMP-ID PIC X(15).
  05 T-AMT PIC 9(9).
  05 FILLER PIC X(117).
```

PAY-MASTER. (From Pgm PM22400.)

```

01 PAY-MASTER.
  05 MASTER-KEY.
    10 NAME PIC X(31).
    10 NUMBER PIC 9(6).
    10 BIRTH-DATE PIC 9(6).
  05 FILLER PIC 9(21).
  05 MODE PIC X.
  05 YR-PAY PIC 9(8).
  05 VAC-DAYS PIC 99.
  05 GRS-PAY PIC 9(5).
  05 NET-PAY PIC 9(5).
  05 FILLER PIC 9(52).
  05 FILLER PIC X(13).
```

MAST-REC. (From Pgm PM22500.)

```

01 MAST-REC.
  05 MAST-HEADER.
    10 M-NAME PIC X(31).
    10 M-NO PIC 9(6).
    10 M-DOB PIC 9(6).
  05 M-DOH PIC 9(6).
  05 M-SS-NO PIC 9(9).
  05 M-DIV-NO PIC 9(3).
  05 M-DEPT-NO PIC 9(3).
  05 M-MODE PIC X.
  05 M-YRLY-PAY PIC 9(8).
  05 M-VAC-DAYS PIC 99.
  05 M-GROSS-PAY PIC 9(5).
  05 M-NET-PAY PIC 9(5).
  05 FILLER PIC 9(36).
  05 M-YTD-GROSS PIC 9(8).
  05 M-YTD-NET PIC 9(8).
  05 FILLER PIC X(13).
```

In this sample record group of length 150 bytes, T-REC may be excluded from the group, as it clearly defines a different physical file. The other three records should remain in this I/O record group.

# Original Source to Composite Record Mapping

Sample composite record for Group 005	Sample composite record with descriptive, standardized data names
01 PAY-MAST.	01 <b>PM-PAY-MAST.</b>
05 MAST-KEY.	05 <b>PM-MAST-KEY.</b>
10 EMP-NAME.	10 <b>PM-EMP-NAME.</b>
15 LAST-NAME PIC X(20).	15 <b>PM-LAST-NAME</b> PIC X(20).
15 FIRST-NAME PIC X(10).	15 <b>PM-FIRST-NAME</b> PIC X(10).
15 MIDDLE-INIT PIC X.	15 <b>PM-MIDDLE-INIT</b> PIC X.
10 <b>EMP-NUMBER</b> PIC 9(6).	10 <b>PM-EMP-NO</b> PIC 9(6).
10 EMP-BIRTH-DT.	10 <b>PM-EMP-BIRTH-DT.</b>
15 EMP-BIRTH-MO PIC 99.	15 <b>PM-EMP-BIRTH-MO</b> PIC 99.
15 EMP-BIRTH-DA PIC 99.	15 <b>PM-EMP-BIRTH-DA</b> PIC 99.
15 EMP-BIRTH-YR PIC 99.	15 <b>PM-EMP-BIRTH-YR</b> PIC 99.
05 <b>DATE-OF-HIRE</b> PIC 9(6).	05 <b>PM-HIRE-DT</b> PIC 9(6).
05 SOC-SEC-NO PIC 9(9).	05 <b>PM-SOC-SEC-NO</b> PIC 9(9).
05 DIVISION-NO PIC 9(3).	05 <b>PM-DIVISION-NO</b> PIC 9(3).
05 DEPT-NO PIC 9(3).	05 <b>PM-DEPT-NO</b> PIC 9(3).
05 <b>P-MODE</b> PIC X.	05 <b>PM-PAY-MODE</b> PIC X.
05 <b>YRLY-SAL</b> PIC 9(8).	05 <b>PM-YEARLY-SALARY</b> PIC 9(8).
05 <b>AV-DAYS</b> PIC 99.	05 <b>PM-ACCUM-VAC-DAYS</b> PIC 99.
05 GROSS-PAY PIC 9(5).	05 <b>PM-GROSS-PAY</b> PIC 9(5).
05 NET-PAY PIC 9(5).	05 <b>PM-NET-PAY</b> PIC 9(5).
05 <b>TOT -DED.</b>	05 <b>PM-TOTAL-DED.</b>
10 FED-TAX PIC 9(3).	10 <b>PM-FED-TAX</b> PIC 9(3).
10 FICA PIC 9(3).	10 <b>PM-FICA</b> PIC 9(3).
10 SS-TAX PIC 9(3).	10 <b>PM-SS-TAX</b> PIC 9(3).
10 ST-TAX PIC 9(3).	10 <b>PM-ST-TAX</b> PIC 9(3).
10 DED-401K PIC 9(3).	10 <b>PM-DED-401K</b> PIC 9(3).
10 DED-HEALTH PIC 9(3).	10 <b>PM-DED-HEALTH</b> PIC 9(3).
05 YTD-GROSS PIC 9(8).	05 <b>PM-YTD-GROSS</b> PIC 9(8).
05 YTD-NET PIC 9(8).	05 <b>PM-YTD-NET</b> PIC 9(8).
05 FILLER PIC X(13).	05 FILLER PIC X(13).

Source: USRM / Comsys

Composite records should use descriptive data names.

# Post-Rationalization Code View

Composites propagated into programs	Program references changed to correspond												
<pre>PROGRAM PRP4030.  COPY ....   ↗ 01 <b>PM-PAY-MAST.</b> 05 <b>PM-MAST-KEY.</b> 10 <b>PM-EMP-NAME.</b>     15 <b>PM-LAST-NAME</b> PIC X(20).     15 <b>PM-FIRST-NAME</b> PIC X(10).     15 <b>PM-MIDDLE-INIT</b> PIC X. 10 <b>PM-EMP-NO</b> PIC 9(6). 10 <b>PM-EMP-BIRTH-DT.</b>     15 <b>PM-EMP-BIRTH-MO</b> PIC 99.     15 <b>PM-EMP-BIRTH-DA</b> PIC 99.     15 <b>PM-EMP-BIRTH-YR</b> PIC 99. 05 <b>PM-HIRE-DT</b> PIC 9(6). 05 <b>PM-SOC-SEC-NO</b> PIC 9(9). 05 <b>PM-DIVISION-NO</b> PIC 9(3). 05 <b>PM-DEPT-NO</b> PIC 9(3). 05 <b>PM-PAY-MODE</b> PIC X. 05 <b>PM-YEARLY-SALARY</b> PIC 9(8). 05 <b>PM-ACCUM-VAC-DAYS</b> PIC 99. .....</pre>	<pre>PROGRAM PRP4030.  MOVE P-MODE TO WS-P-MODE. IF BIWEEKLY   PERFORM CHECK-WEEK   ADD .385 TO WS-AV-DAYS. IF MONTHLY   PERFORM CHECK-MONTH   ADD 1.25 TO WS-AV-DAYS. MOVE WS-AV-DAYS TO AV-DAYS. ...   ↓ MOVE <b>PM-PAY-MODE</b> TO WS-P-MODE. IF BIWEEKLY   PERFORM CHECK-WEEK   ADD .385 TO WS-AV-DAYS. IF MONTHLY   PERFORM CHECK-MONTH   ADD 1.25 TO WS-AV-DAYS. MOVE WS-AV-DAYS TO <b>PM-ACCUM-VAC-DAYS.</b> ...</pre>												
<pre>01 WS-P-MODE PIC X. 88 BIWEEKLY PIC X VALUE 'B'. 88 MONTHLY PIC X VALUE 'M'. 01 WS-YRLY-SAL PIC 9(8). 01 WS-AV-DAYS PIC 99. .....</pre>	<p><b>Name replacement list for related subordinate references which are locally defined</b></p> <table border="0"> <thead> <tr> <th><u>ORIGINAL</u></th> <th><u>PROPOSED</u></th> </tr> </thead> <tbody> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>WS-P-MODE</td> <td>WS-PAY-MODE</td> </tr> <tr> <td>WS-YRLY-SAL</td> <td>WS-YEARLY-SALARY</td> </tr> <tr> <td>WS-AV-DAYS</td> <td>WS-ACCUM-VAC-DAYS</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	<u>ORIGINAL</u>	<u>PROPOSED</u>	...	...	WS-P-MODE	WS-PAY-MODE	WS-YRLY-SAL	WS-YEARLY-SALARY	WS-AV-DAYS	WS-ACCUM-VAC-DAYS	...	...
<u>ORIGINAL</u>	<u>PROPOSED</u>												
...	...												
WS-P-MODE	WS-PAY-MODE												
WS-YRLY-SAL	WS-YEARLY-SALARY												
WS-AV-DAYS	WS-ACCUM-VAC-DAYS												
...	...												

Source: USRM / Comsys

Composite propagation ensures that all programs use consistent definitions for a given physical record, segment or table.

# Rationalization of Local Program

## Element Names

### Example

Name replacement list for related subordinate references which are locally defined

<u>ORIGINAL</u>	<u>PROPOSED</u>
...	
WS-P-MODE	WS-PAY-MODE
WS-YRLY-SAL	WS-YEARLY-SALARY
WS-AV-DAYS	WS-ACCUM-VAC-DAYS
...	

Local names replaced

...		
01	<b>WS-PAY-MODE</b>	PIC X.
88	BIWEEKLY	PIC X VALUE 'B'.
88	MONTHLY	PIC X VALUE 'M'.
01	<b>WS-YEARLY-SALARY</b>	PIC 9(8).
01	<b>WS-ACCUM-VAC-DAYS</b>	PIC 99.
...		

Source: USRM / Comsys

Program references changed to correspond

PROGRAM PRP4030.

MOVE PM-PAY-MODE TO WS-P-MODE.  
 IF BIWEEKLY  
 PERFORM CHECK-WEEK  
 ADD .385 TO WS-AV-DAYS.  
 IF MONTHLY  
 PERFORM CHECK-MONTH  
 ADD 1.25 TO WS-AV-DAYS.  
 MOVE WS-AV-DAYS TO  
 PM-ACCUM-VAC-DAYS.

...



MOVE PM-PAY-MODE TO **WS-PAY-MODE**.  
 IF BIWEEKLY  
 PERFORM CHECK-WEEK  
 ADD .385 TO **WS-ACCUM-VAC-DAYS**.  
 IF MONTHLY  
 PERFORM CHECK-MONTH  
 ADD 1.25 TO **WS-ACCUM-VAC-DAYS**.  
 MOVE **WS-ACCUM-VAC-DAYS** TO  
 PM-ACCUM-VAC-DAYS.

# Homonym Identification & Correction

Homonym analysis list			
NAME	DEFINITION	TIMES USED	
...			
ACCT-NO	PIC X(8)	8	← Correct unique definition of ACCT-NO
ACCT-NO	PIC 9(8) COMP-3	1	
ACCT-NO	PIC X(4)	1	
ACCT-TYPE	PIC X(3)	3	← Correct unique definition of ACCT-TYPE
ACCT-TYPE	PIC X	1	
...			

Homonym replacement list			
ORIGINAL NAME	DEFINITION	PROPOSED NAME	DEFINITION
...			
ACCT-NO	PIC 9(8) COMP-3	CLIENT-ACCT-NO	PIC 9(8) COMP-3
ACCT-NO	PIC X(4)	SERVICE-ACCT-NO	PIC X(4)
ACCT-TYPE	PIC X	ACCT-MODE	PIC X
...			

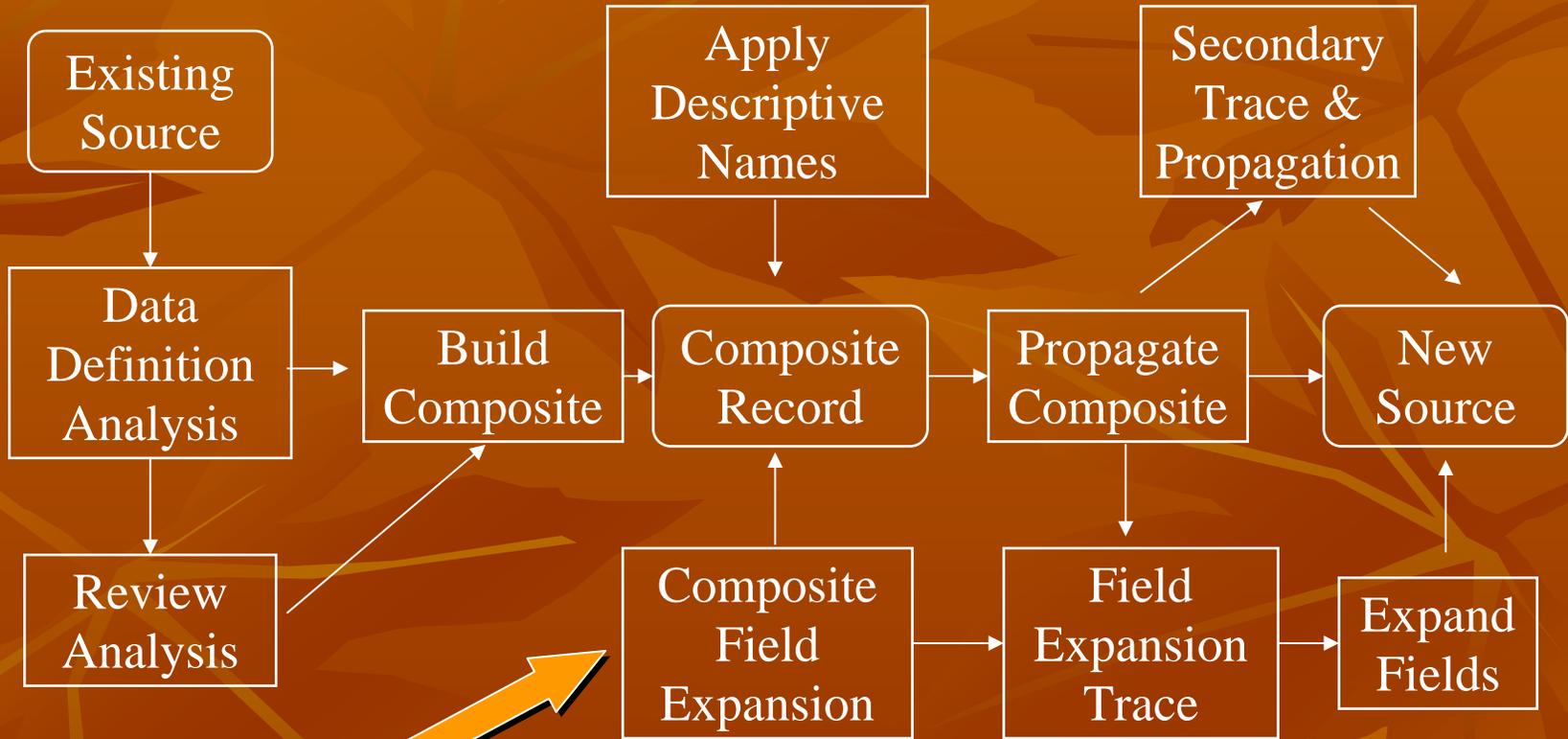
**Source: USRM / Comsys**

Data elements having the same name but referring to different physical data should be corrected as part of the rationalization process.

# Field & Record Size Expansion

- Expands business data and related definitions
- Includes physical data and all definitions of that data
- Examples: area codes, bar codes, dollar amounts, dates
- This is a business-driven initiative

# Combined Rationalization / Field Expansion View



*Field Expansion Option*

# Sample Tool Support – Data Tracing

**Data Files**

Match 11 Data File(s) matching ""

- PROD.CARDS(REPORT2)
- PROD.DEFAULT.MASTER
- PROD.GAME.MASTER
- PROD.HOMEBUY.MASTER
- PROD.MAIL.MASTER
- PROD.MAIL.XREF.FILE
- ONLINE.FCT
- BATCH2.CBL
  - BATCH2.CBL Line 37: 003500 SELECT MAIL-XREF
  - BATCH2.CBL Line 92: 009000 FD MAIL-XREF
  - BATCH2.CBL Line 465: 067100 OPEN I-O MAIL-XREF.
  - BATCH2.CBL Line 558: 148100 READ MAIL-XREF INTO**
  - BATCH2.CBL Line 600: 151600 WRITE MAIL-XREF-REC
  - BATCH2.CBL Line 1014: 154500 REWRITE MAIL-XREF-REC FROM
  - BATCH2.CBL Line 1041: 170900 CLOSE MAIL-XREF.
- ONLINE2.CBL
- REPL.JCL

**Programs**

Match 1 Program(s) matching "batch2"

**BATCH2**

- Copybooks Used
  - copybook BRHL01
  - copybook DEFAULTS
  - copybook MAILLIST
  - copybook SALES
- Data Accessed
  - file PROD.HOMEBUY.MASTER
  - file PROD.MAIL.MASTER
  - file PROD.MAIL.XREF.FILE

**Source Code (Left Pane):**

```

148000
148100 READ MAIL-XREF INTO HB01-MAIL-STATION-RECORD.
148200
148300 MOVE 'BATCH2' TO WS-VSAM-PGM-NAME.
148600 PERFORM F100-CHECK-VSAM-STATUS.
148700*
148800 IF WS-VSAM-STATUS = '00'
148900 PERFORM G100-ADD-TO-MAILSTAT THRU
149000 G100-ADD-TO-MAILSTAT-EXIT
149100 PERFORM U700-REWRITE-MAILSTAT-EXIT
149200 U700-REWRITE-MAILSTAT-EXIT
149300 ELSE
149400 PERFORM D200-CREATE-MAILSTAT THRU
149500 D200-CREATE-MAILSTAT-EXIT.
149600*
149700 G200-UPDATE-MAILSTAT-EXIT
    
```

**Source Code (Right Pane):**

```

154400*
154500 REWRITE MAIL-XREF-REC FROM HB01-MAIL-STATION-RECORD.
154600*
154700 PERFORM F100-CHECK-VSAM-STATUS.
154800 ADD +1 TO WS-MAIL-XREF-REWRITE.
154900*
155000 U700-REWRITE-MAILSTAT-EXIT.
155100*
155200*
155300*
155400*
155500*
155600*
155700*
155800*
155900*
156000*
156100*
156200*
156300*
156400*
156500*
156600*
156700*
156800*
156900*
157000*
157100*
157200*
157300*
157400*
157500*
157600*
157700*
157800*
157900*
158000*
158100*
158200* CLOSE FILES
158300*****
158400 U800-CLOSE-FILES.
158800
158900 CLOSE SALES-TRANS.
159000
159100 CLOSE GAME-MASTER.
    
```

**Data tracing across system to program level**

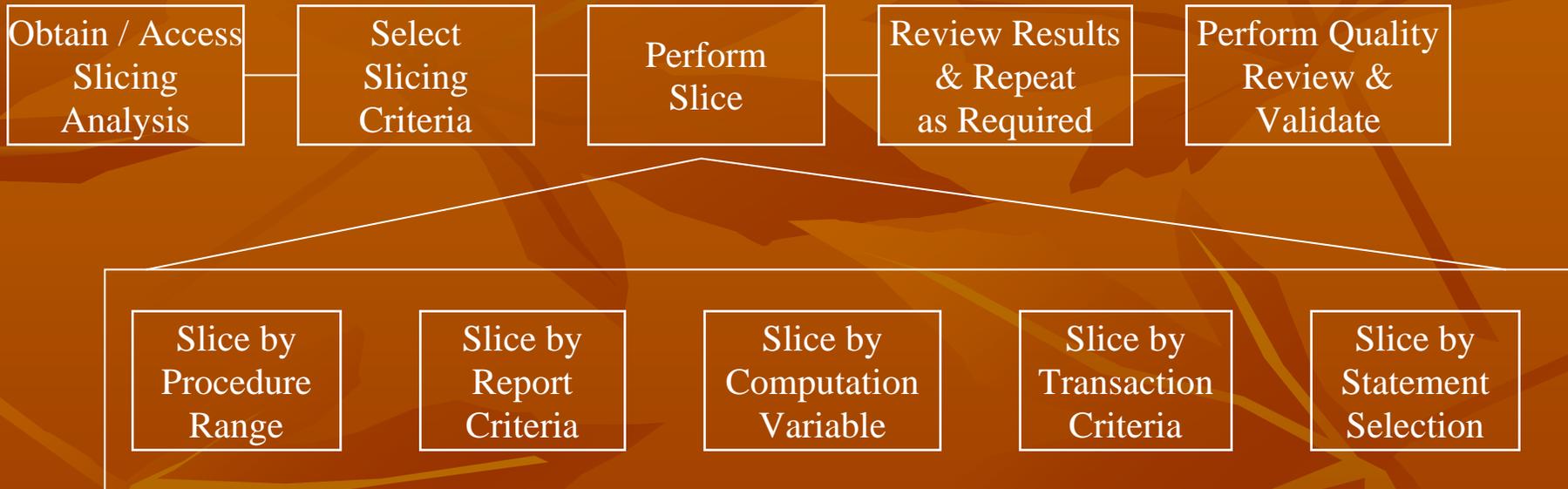
E:\TPS6\SAMPLES\REVOLVE\SAMPLEAPP\COBOL\BATCH2.CBL

Source: Micro Focus

# Literal Externalization

- Finds embedded business data
- Places it into user – modifiable tables or databases
- Examples: dates, plant-codes, area-codes, interest rates, etc.

# Code Slicing Techniques



- Isolates logic into new modules
- Supports functional modularity
- Simplifies maintainability for large programs
- Supports redundancy reconciliation and re-aggregation<sup>3</sup>

# Slicing Criteria Driven by Business Objective

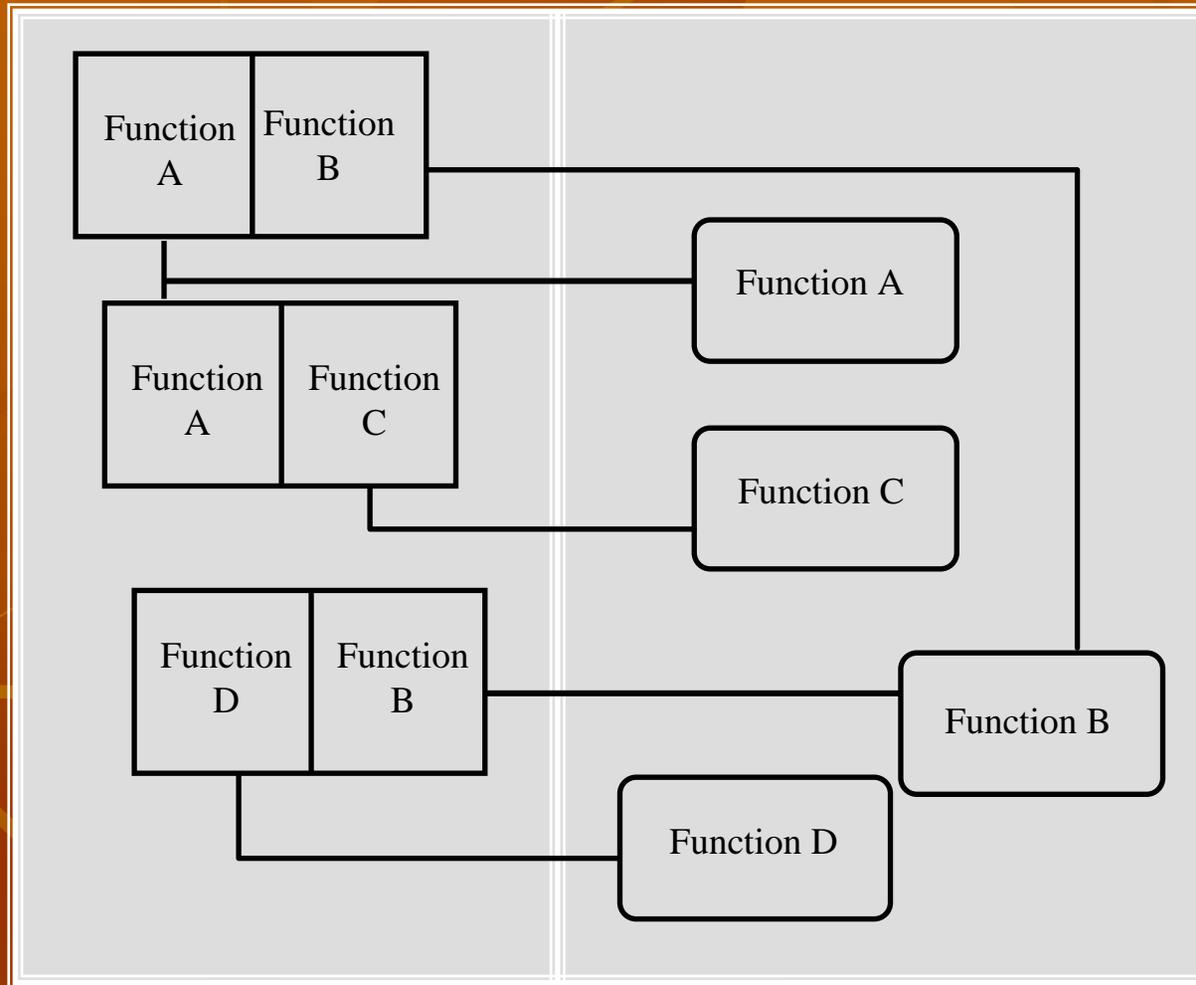
Slicing Criteria	Objectives
1. Execution Range Criteria (Structured code is required)	Large module size reduction Creation of shared business logic routines to develop usable code
2. Report Criteria	On-line migrations Reusability of report logic Creation of independent I/O layer Functional segregation Reusable routine isolation
3. Computation Variable Criteria	Functional isolation Reusable routine isolation Functional segregation / modernization preparation
4 Transaction Criteria	Functional re-aggregation support Reusable routines Creation of independent I/O layer Creation of independent data access layer
5. Statement Selection Criteria	Reusable routine isolation

Source: USRM / Comsys

# System Re-aggregation

Current System

Re-aggregated System

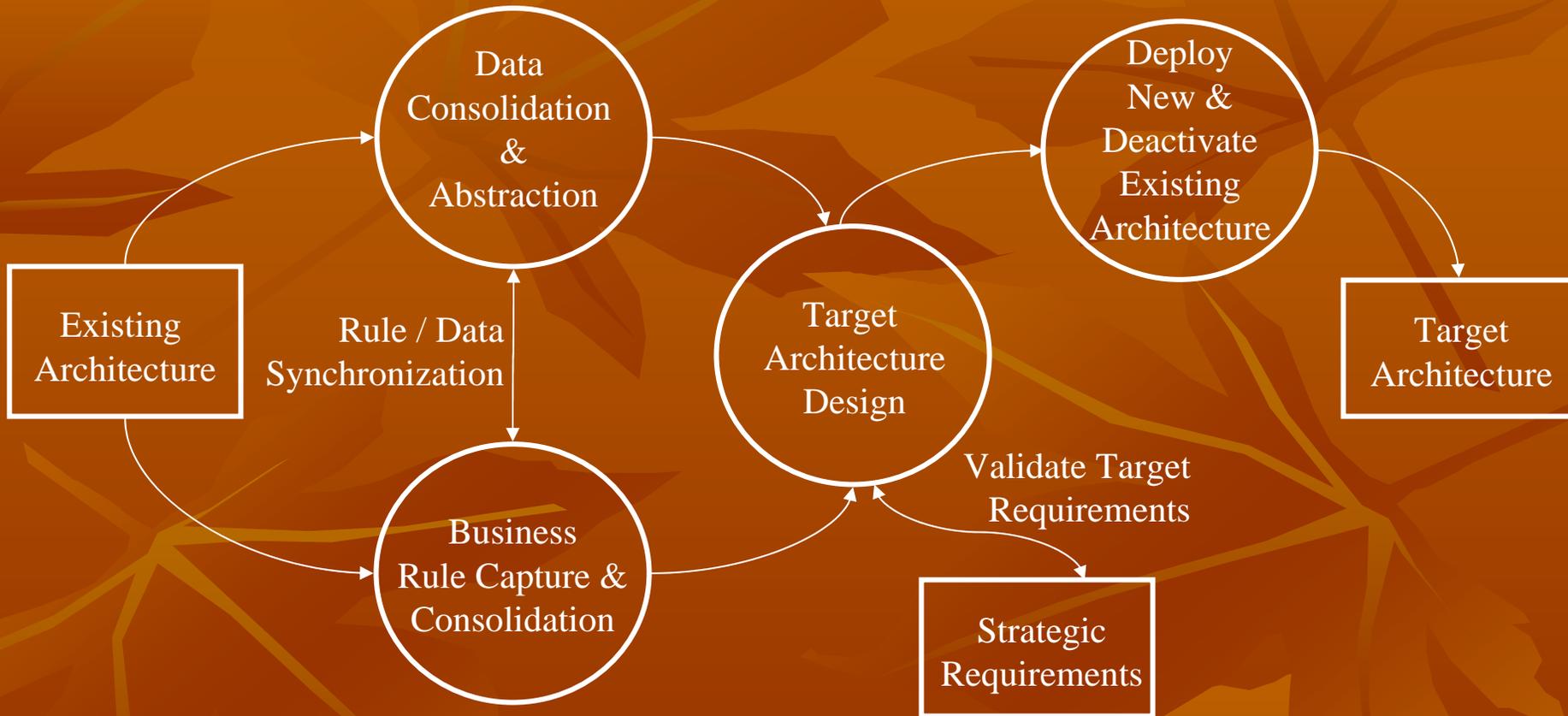


Functions can be split out and recombined to re-aggregate application functionality.

# VIII. Architecture Modernization & Redeployment

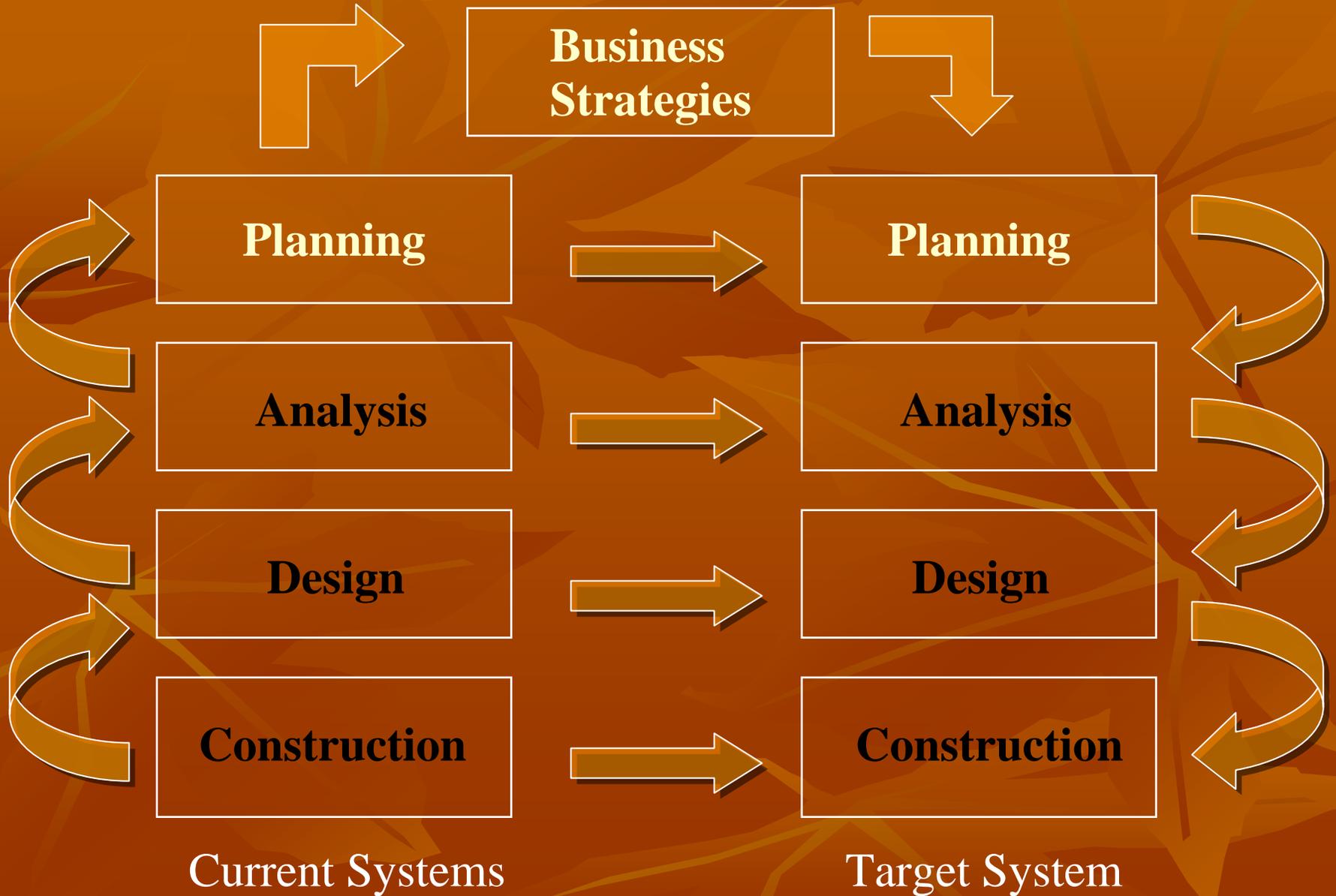
- Leverage existing applications to facilitate multiple redevelopment scenarios
- Reuse, where applicable, system and program level components
- Improve the integrity and completeness of new systems
- Shorten / streamline application development process
- Facilitate application integration efforts

# Architecture Modernization Options

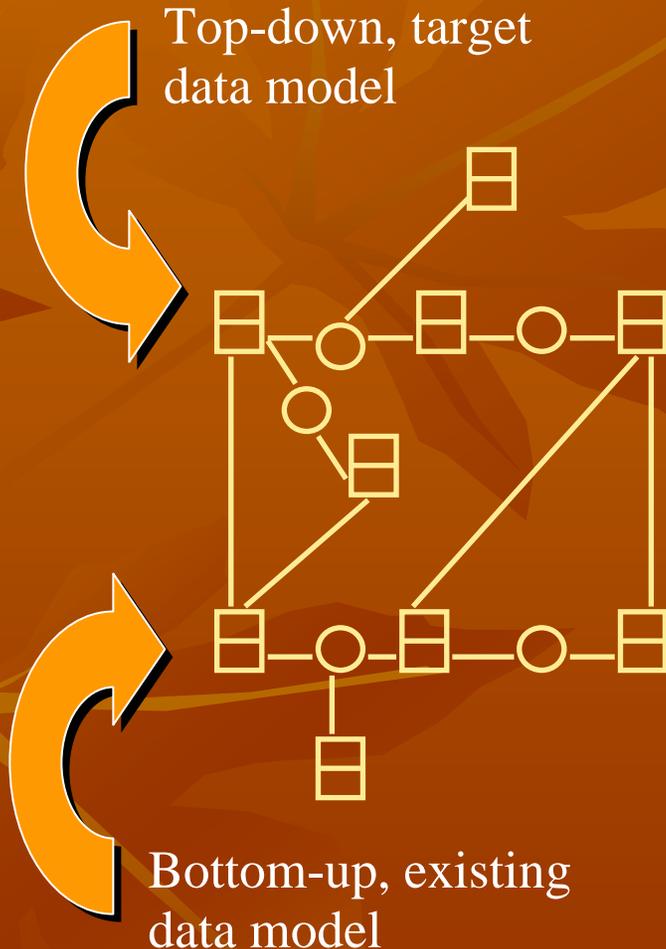


Modernization options include data and business rule capture, consolidation, redesign, validation and redeployment.

# Redesign & Re-Specification



# Building a High Integrity Data Architecture

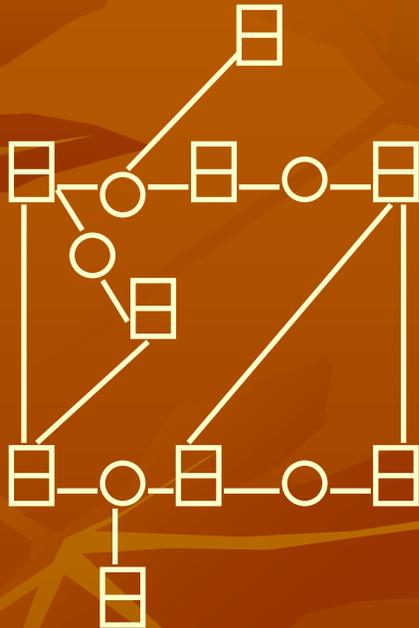


## Deliverables

- Integrated top-down, bottom-up data model
- Gap analysis determining missing data entities & attributes
- Reconciliation of inconsistent data usage
- An understanding of existing to target data migration requirements

**Consolidating top-down and bottom-up data models produces a higher quality data architecture and increases likelihood of project success.**

# Data Mapping Process



## Object Model Derivation

- Can utilize ER model or logical data model as input
- Extends role entities into object subtypes for inheritance
- Introduces composition and cascading hierarchies
- Extends model to support behavioral mapping

Note that many object oriented systems are implemented on RDBMS. This approach eases synchronization of objects and RDBMS design.

# Logical Functional Mapping Process

## Top Down Inputs

- I/A Functional Decomposition
- Logical Data Model
- I/A Functional Mapping

## Functional Mapping Process

- Expand I/A functional decomposition
- Expand I/A current to target mapping
- Complete current to target gap analysis
- Develop integrated functional decomposition & CRUD matrix

## Bottom Up Inputs

- Source code, screen layouts & JCL
- Screen & report layouts
- I/A assessment results

## Deliverables

- Detailed mapping of reusable source code components
- Identification of reusable I/O
- Integrated functional structures & data / functional matrix

# Reverse Requirements Tracing of Logical Processes

Purchasing Department Vendor / Product Report		Date
Vendors/Products Added xxxxxxx 999999 xxxxxxxx		99/99/99
Vendors/Products Changed xxxxxxx 999999 xxxxxxxx		99/99/99
Vendors/Products Discontinued xxxxxxx 999999		

FORM 004 - Creating current function / process hierarchy - establishing current processes			
PROGRAM SOURCE	CURRENT PROCESS	CURRENT FUNCTION	TARGET FUNCTION
	Select vendor / product	Purchasing	
	Change vendor / product	Purchasing	
	Roll off vendor / product	Purchasing	
	Order supplies	Purchasing	
	Pay vendors	Purchasing	
	Track vendor shipments	Sending/Receiving	
	Disperse supplies	Sending/Receiving	
	Maintain supply stock	Inventory	
	Request supplies	Inventory	
	Ship corporate products	Sending/Receiving	
	Manage materials	Manufacturing	
	Request supplies	Manufacturing	
	Create products	Manufacturing	

Source: USRM / Comsys

Reverse requirements tracing abstracts application functionality by tracking user views back into the source code.

# Cobol Business Rule Candidate

Scope of business rule candidate

B9000-POST-8000.

\* CALCULATE CURR AND OUTPUT ACCOUNT BALANCE TO POSTING SUSPENSE VSAM

PERFORM B9050-CALC-MONTHLY-CHARGES THRU B9050-EXIT.

IF DDACCT-ACCT-SRVC-CHG-FL NOT = "W" AND "G" AND "X"

IF HEA1010A-ACCT-BAL = ZERO

COMPUTE HEA1010A-ACCT-BAL = DDACCT-ACCT-BAL - (WS-TOT-MONTHLY-CHARGES \* 2)

IF HEA1010A-ACCT-BAL < ZERO

PERFORM U8010-CALC-PENALTY THRU U8010-EXIT

MOVE "Y" TO HEA1010A-PENALTY-FL

COMPUTE HEA1010A-ACCT-BAL = HEA1010A-ACCT-BAL - WS-TOT-PENLTY-CHG

ELSE NEXT SENTENCE

ELSE

COMPUTE HEA1010A-ACCT-BAL = DDACCT-ACCT-BAL - WS-TOT-MONTHLY-CHARGES

IF HEA1010A-ACCT-BAL < ZERO

PERFORM U8010-CALC-PENALTY THRU U8010-EXIT

MOVE "Y" TO HEA1010A-PENALTY-FL

COMPUTE HEA1010A-ACCT-BAL = HEA1010A-ACCT-BAL - WS-TOT-PENLTY-CHG

ELSE NEXT SENTENCE

ELSE IF DDACCT-ACCT-SRVC-CHG-FL = "W"

MOVE WS-TOT-MONTHLY-CHARGES TO HEA1010A-SRVC-CH-DFR-HLD.

\*\*\*\* DISABLED 05/03/89 BY BGG - HANDLED DIFFERNTLY NOW

\*\*\* ELSE IF DDACCT-ACCT-SRVC-CHG-FL = "G"

\*\*\* MOVE WS-TOT-MONTHLY-CHARGES TO HEA1010A-SRVC-CH-GIC-HLD.

PERFORM B9100-FILL-SGMENT THRU B9100-EXIT.

PERFORM B9200-OUTPT-POST THRU B9200-EXIT.

B9000-EXIT. EXIT.

Source: Netron, Inc. 2002

Rules can be reused as-is or as basis for existing redesign.

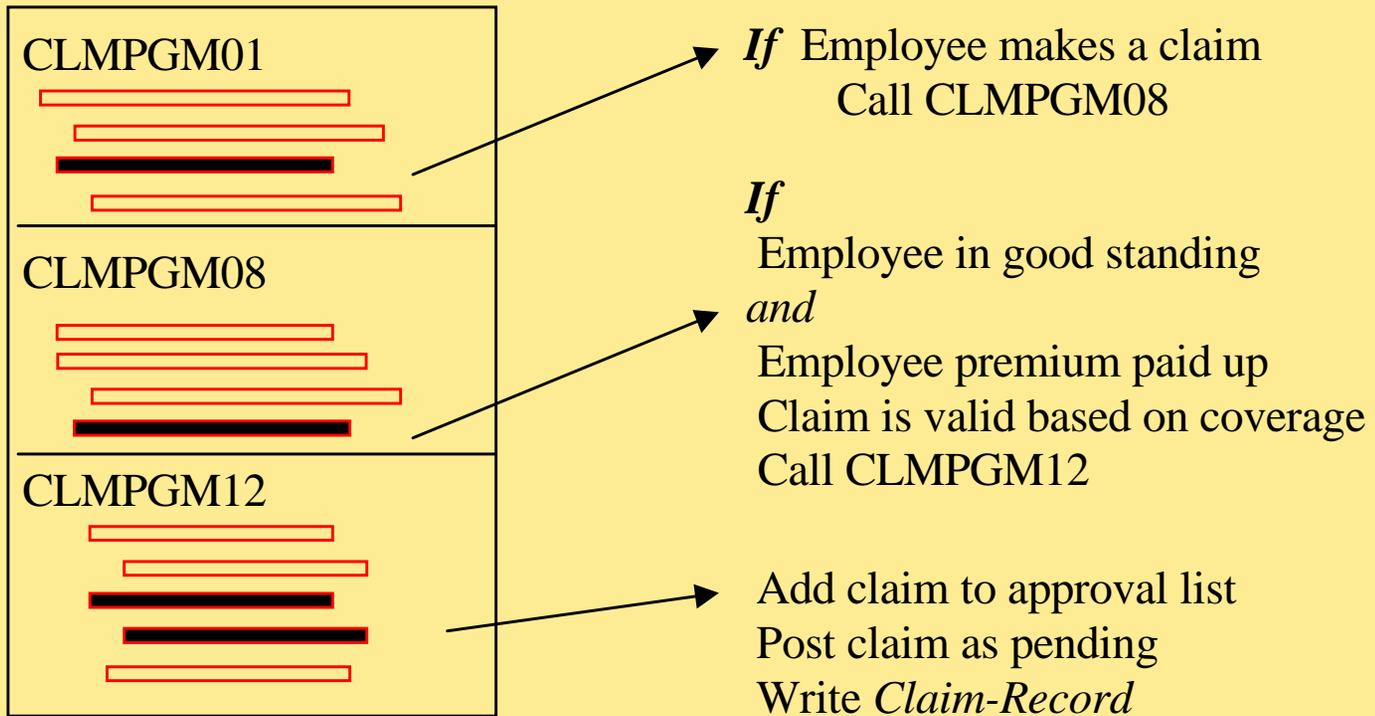
# Non-Business Logic Identification Table

<b>Non-Business Logic</b>	<b>Non-Business Logic Identification</b>
<i>Syntactically dead code</i>	Logic never executed regardless of data values.
<i>Semantically dead code</i>	Logic not executed based on setting of data values.
<i>Program initialization</i>	Logic that initializes element or record area values.
<i>Input / output logic</i>	Code accessing physical data. Includes call, read, write and other I/O statements.
<i>Output area build</i>	Logic that moves data to screen and report work areas.
<i>Status checking</i>	Tests directly after I/O commands checking communication or other status codes.
<i>Error handling</i>	Imperative logic that invokes exception reporting or module termination based on status code results.
<i>Data manipulation</i>	Database or file manipulation logic.
<i>Environmental logic</i>	Manages security, homegrown technology, date handling or similar routines.
<i>Extraneous logic</i>	Redundant conditionals, mutually exclusive tests or similar routines.

Identifying and discarding non-business logic should be based on a solid understanding of the logic types involved in this process.

# Rule Reuse Requires Functional Mapping at a Granular Level

## *Legacy Insurance System*



# Redundancy Tracking & Consolidation

- Should address unnecessary data and functional redundancies
- Streamlining / consolidating redundancy requires ROI analysis
- Process is tool supported but must be business-driven

# Sample Tool Support - Componentization

The screenshot displays the Net Express IDE interface. On the left, a code editor shows COBOL code for a form input section. The central pane shows a visual design of a form with input fields for OMF-ACCTNO, OMF-SAI FSREP, OMF-ITEM, OMF-REC-QTY, and OMF-REC-AMT. On the right, a project browser shows the HTML structure of the form, including a BODY1 container with a FORM1 element and its sub-elements: FORMHEADER, FORMBODY, and FORMFOOTER. A callout box points to the COBOL code, explaining that AssetMiner generates callable COBOL subroutines and copybooks, while NetExpress generates components in various target formats.

```
process-form-input-data section
  *> Accept the input from t
  perform browser-initialize
  accept htmlform
  exit.

*-----
convert-input section.

  *> Convert numeric input v
  perform input-conversion

  *> Add additional validation
  if v-all-ok = 0
    perform output-form-error
  end-if
  exit.

*-----
output-form-error-and-stop section
```

**AssetMiner generates callable COBOL subroutines and copybooks. NetExpress, generates "components" in the required target format (COBOL, Java Beans, COM objects, EJB's,...).**

Source: Micro Focus

# IX. Enabling Tools & Technologies

Tool Feature \ Method	Process Redesign	Asset Mgmt	Development	ReTool/ Redeploy	Maintenance
<b>System Wide Analysis</b>				X	X
Maintenance environment					X
Redevelopment environment				X	
System wide summary analysis		X		X	X
General purpose repository	X	X	X	X	X
<b>Program Level Analysis</b>					
Process flow metrics				X	X
Program flow diagrams				X	X
In-line cross reference				X	X
Interactive analysis tools				X	X
<b>Code Improvement</b>					
Code restructuring				X	
Data definition rationalization				X	
Remodularization tools				X	X
Physical data migration					
<b>Model Based Tools</b>					
Business planning tools	X		X		
Analysis tools	X			X	
Development tools			X	X	X
Design tools			X	X	
Reverse engineering tools			X	X	
Integration Tools			X	X	X

# Sample Modeling & Design Tools

- **Business Analysis and Knowledge Management**
  - Gadrian – Enterprise Analytics, Inc.
  - StarBEAM – Entreon Corporation
  - Process Analyzer – Fuego, Inc.
  - Fuego's Business Language – Fuego, Inc.
  - Process Orchestration Designer – Fuego, Inc.
  
- **Business Process Integration and Automation**
  - Component Manager – Fuego, Inc.
  - Orchestration Administrator – Fuego, Inc.
  - Orchestration Engine – Fuego, Inc.
  - Work Portal – Fuego, Inc.
  - Intalio|n<sup>3</sup> Server – Italo
  - Intalio|n<sup>3</sup> Designer – Italo
  - Intalio|n<sup>3</sup> Director – Italo
  - TeamWorks – Lombardi Software
  - webMethods Workflow – webMethods, Inc.
  - webMethods Manager – webMethods, Inc.

# Sample Modeling & Design Tools

## ■ UML Modeling Tools

- AllFusion Component Modeler – Computer Associates
- NeuArchitect – IBM Rational
- Rational Rose – IBM Rational
- System Architect – Popkin Software

## ■ Data Modeling Technology

- AllFusion ERwin Data Modeler – Computer Associates
- NeuArchitect – IBM Rational
- SILVERRUN ERX – Magna Solutions
- SILVERRUN RDM – Magna Solutions
- System Architect – Popkin Software

# Sample Analysis & Documentation Tools

- **Batch Environmental Analyzer**
  - PM/SS – Adpac Corporation
  - BluePhoenix IT Discovery – BluePhoenix
- **Interactive Environmental Analyzer**
  - ASG-Vista – ASG Software Solutions
  - BluePhoenix IT Warehouse – BluePhoenix
  - Application Mining Suite – CAST
  - Headway reView – Headway Software, Inc.
  - WebSphere Studio Asset Analyzer – IBM
  - inSight – klocwork
  - Micro Focus Revolve – Micro Focus
  - Micro Focus Revolve Enterprise Edition – Micro Focus
  - Hotrod – Netron
  - PathPoint – PathPoint Software, Inc.
  - Application Profiler – Relativity Technologies
  - Application Analyzer – Relativity Technologies
  - SEEC Mosaic – SEEC
  - eVolution 2000 – Software Revolution, Inc.
  - DASE/Ware Understand IT – SWS Software Technologies
  - Transoft evolveIT Application Mining – Transoft

# Sample Analysis & Documentation Tools

- **Batch Static Analyzer**
  - ASG-Recap – ASG Software Solutions
  - VISION:Inspect – Computer Associates
  - McCabe Reengineer – McCabe & Associates
  - Fermat Migration System – Software Migrations Ltd
- **Interactive Static Analyzer**
  - ASG-Insight – ASG Software Solutions
  - VISION:Assess – Computer Associates
  - XPEDITER – Compuware Corporation
  - inForce – klocwork
  - inSpect – klocwork
  - inTellect – klocwork
  - Application Analyzer – Relativity Technologies
  - Application Profiler – Relativity Technologies
  - Understand for Ada – Scientific Toolworks, Inc.
  - Understand for C++ – Scientific Toolworks, Inc.
  - Understand for Java – Scientific Toolworks, Inc.
  - Understand for FORTRAN – Scientific Toolworks, Inc.
  - Understand for JOVIAL – Scientific Toolworks, Inc.
  - eVolution 2000 – Software Revolution, Inc.

# Sample Analysis & Documentation Tools

- **Dynamic Analysis**
  - XPEDITER – Compuware Corporation
  - McCabe Reengineer – McCabe & Associates
- **Data Definition Analyzer**
  - XPEDITER – Compuware Corporation
  - Micro Focus Revolve Enterprise Edition – Micro Focus
  - Hotrod – Netron
  - SEEC Mosaic – SEEC
- **Repository Tools**
  - BluePhoenix IT Warehouse – BluePhoenix
  - PLATINUM Repository – Computer Associates
  - Flashline 4 – Flashline, Inc.

# Sample Application Improvement Tools

- **Language Change/Upgrade Tool**
  - *Anything\_to\_.NET* – ArtinSoft
  - ASG-SmartEdit – ASG Software Solutions
  - BluePhoenix LanguageMigrator – BluePhoenix
  - Language Conversions – Datatek, Inc.
  - NXTware – eCube Systems, L.L.C.
  - CCCA – IBM
  - Transaction Platform – LegacyJ
  - PerCobol – LegacyJ
  - DialogSys2Java – LegacyJ
  - Net Express – Micro Focus
  - Micro Focus Enterprise Server – Micro Focus
  - Micro Focus Application Server – Micro Focus
  - Transformation Assistant – Relativity Technologies
  - Fermat Migration System – Software Migrations Ltd
  - eVolution 2000 – Software Revolution, Inc.
  - iNET – Stryon, Inc.
  
- **Program Structuring Tool**
  - VISION:Recode – Computer Associates
  
- **Code Modularization Facility**
  - ASG-Encore – ASG Software Solutions
  - VISION:Recode – Computer Associates
  - VISION:Assess – Computer Associates
  - McCabe Reengineer – McCabe & Associates

# Sample Application Improvement Tools

- **Data Definition Rationalization**
  - WebSphere Studio Asset Analyzer – IBM
  - Micro Focus Revolve Enterprise Edition – Micro Focus
  - Hotrod – Netron
  - SEEC Mosaic – SEEC
- **Data Definition Expansion**
  - BluePhoenix EuroEnabler – BluePhoenix
  - BluePhoenix GTIN/UPCEnabler – BluePhoenix
  - Micro Focus Revolve Enterprise Edition – Micro Focus
  - Reasoning5 – Reasoning Systems

# Sample Extraction & Redesign Tools

- **Presentation Layer Extraction**
  - eXtremeVista – Open Connect
  - Transoft Screen Component Adapters – Transoft
  - WinFast & GUI – Visual Legacy Group

# Sample Extraction & Redesign Tools

- **Transaction Flow Analysis**
  - Hotrod – Netron
  - Application Profiler – Relativity Technologies
  - Application Analyzer – Relativity Technologies
  - SEEC Mosaic – SEEC
- **Data Reverse Engineering and Reengineering**
  - NeuArchitect – IBM Rational
  - SILVERRUN ERX – Magna Solutions
  - SILVERRUN RDM – Magna Solutions
  - Hotrod – Netron
  - Application Profiler – Relativity Technologies
  - Application Analyzer – Relativity Technologies
  - SEEC Mosaic – SEEC

# Sample Business Rule Capture & Consolidation Tools

- **Business Rule Capture, Consolidation and Componentization**
  - Application Mining Suite – CAST
  - Hotrod – Netron
  - Business Rule Manager – Relativity Technologies
  - SEEC Mosaic – SEEC
  - eVolution 2000 – Software Revolution, Inc.
  - Transoft evolveIT BRM Business Rules Mining – Transoft

# Sample Validation Tools

- **Change Integration Tool**
  - ChangeMan – Serena Software
- **Transaction Simulation Facility**
  - Matador – Matador Technologies Corp.
  - Mainframe Express – Micro Focus
  - Illuma – Reasoning Systems

# Sample Validation Tools

- **Source & Object Code Comparison Tool**
  - XPEDITER – Compuware
  - Comparex – Serena Software
- **Test Coverage Analyzer**
  - ASG-SmartTest – ASG Software Solutions
  - XPEDITER – Compuware
  - QAHiperstation – Compuware
  - Matador – Matador Technologies Corp.
- **Source & Object Code Comparison Tool**
  - XPEDITER – Compuware
  - Comparex – Serena Software

# Sample Maintenance & Modernization Workbenches

- Application Mining Suite – CAST
- Hotrod – Netron
- Transformation Assistant – Relativity Technologies
- SEEC Mosaic – SEEC
- eVolution 2000 – Software Revolution, Inc.
- Transoft Legacy Liberator – Transoft

# Sample Data Analysis (Mining) & Migration Tools

- **Data Mining Technology**
  - CleverPath Predictive Analysis Server – Computer Associates
  - ETI EXTRACT – Evolutionary Technologies International
- **Physical Data Analyzer & Cleansing Tool**
  - Advantage Data Transformer – Computer Associates
  - Advantage InfoRefiner – Computer Associates
  - Evoke Axio Product Suite – Evoke Software
  - ETI EXTRACT – Evolutionary Technologies International
  - Mediator – Ontology Works

# Sample Data Analysis & Migration Tools

- **Physical Data Design Tool**
  - All Fusion ERwin Data Modeler – Computer Associates
  - Evoke Axio Product Suite – Evoke Software
  - SILVERRUN ERX – Magna Solutions
  - SILVERRUN RDM – Magna Solutions
  
- **Data Migration Technology**
  - BluePhoenix DBMS Migrator – BluePhoenix
  - BluePhoenix DataMigrator – BluePhoenix
  - Advantage InfoRefiner – Computer Associates
  - File-Aid products – Compuware Corporation
  - File Manager – IBM
  - Startool – Serena Software
  - HIREL – SWS Software Technologies
  - IXREL – SWS Software Technologies
  - VREL – SWS Software Technologies

# Sample Project Management & Administration Tools

- **Configuration Management**
  - Integrated Software Engineering Platform – Digité
  - Flashline 4 – Flashline
- **Project Management Tool**
  - Integrated Software Engineering Platform – Digité
- **Collaborative Work Environments**
  - Integrated Software Engineering Platform – Digité
  - Rational Suite AnalystStudio – IBM Rational
- **Change Integration Tool**
  - ChangeMan – Serena

# X. Selected Modernization Scenarios and Case Studies

- Scenarios depict practical application of modernization discipline, tools and strategies
- Scenarios have a business-driven project focus and are typically delivered in phases
- Case studies reflect actual project results

# Scenario Architecture Impact Summary

Scenario Category	Impact	Documentation	Legacy Application and / or Data Impact	Target Architecture Impact
Portfolio Management		X		
Application Stabilization		X	X	
Data Integration / Data Warehouse		X	X	
Enterprise Application Integration (EAI)		X		X
Distributed Front-end Migration		X	X	
Relational Database Migration		X	X	X
Application Consolidation		X	X	X
ERP / Package Replacement		X	X	
Rehosting		X	X	X
Component-Based Migration		X	X	X

**Scenario impact on existing and target architectures.**

# Portfolio Management Scenario

## Issues:

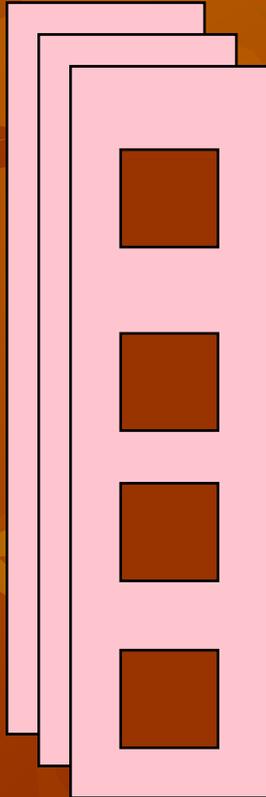
- Large IT portfolio, difficult to navigate
- Existing systems poorly documented
- Cross-functional / cross-application project being proposed
- Outsourcing plan proposed

## Strategy:

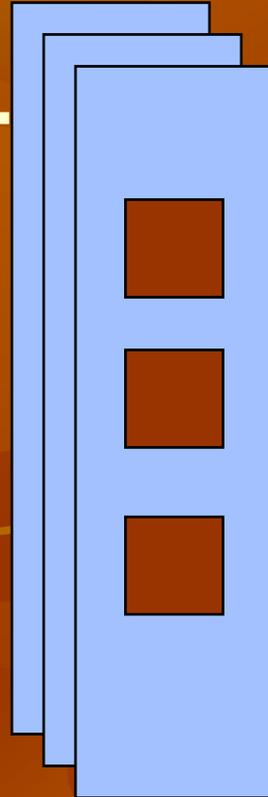
- Document cross-functional applications and relation to business functions in relevant business units using repository

# Portfolio Management

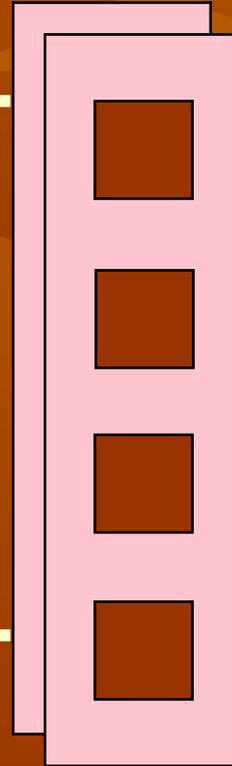
*Human  
Resources*



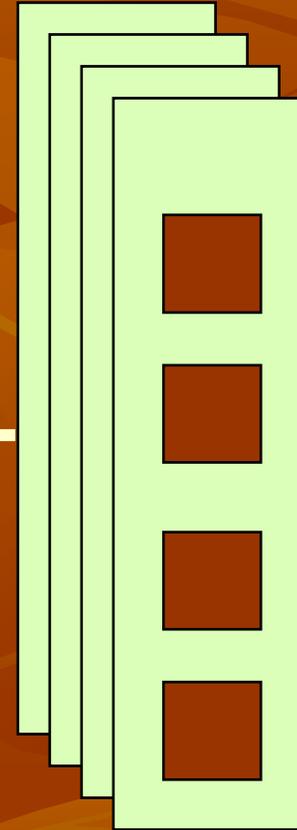
*Accounting  
& Revenues*



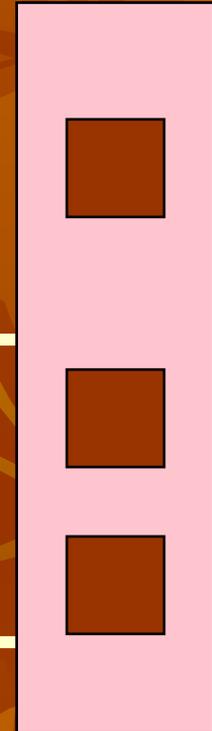
*Sales &  
Marketing*



*Back Office  
Processing*



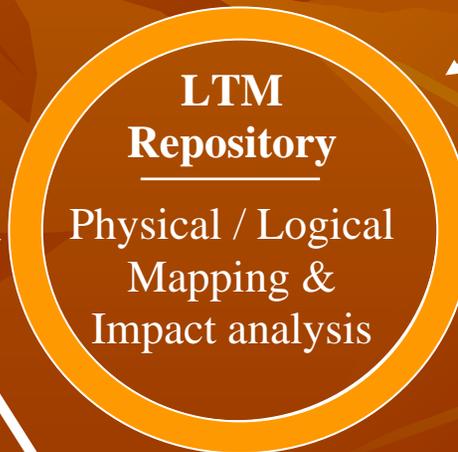
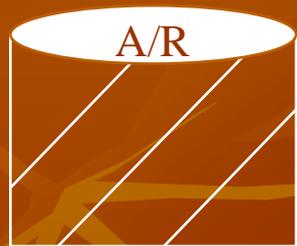
*Inventory  
Controls*



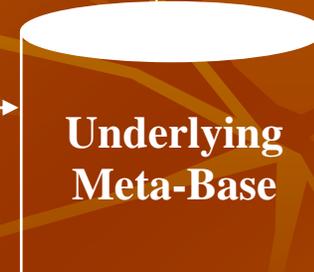
Enterprise wide analysis scenario inventories systems portfolio, defines functional / physical boundaries and supports enterprise level transition planning.

# Portfolio Management: Information Capture & Management

Existing  
Applications



Access to on-line  
Documentation



# Application Stabilization Scenario

## Issues:

- Systems and programs hard to change and understand
- Poorly structured, designed source programs
- Data definitions redundant, and hard to understand

## Strategy:

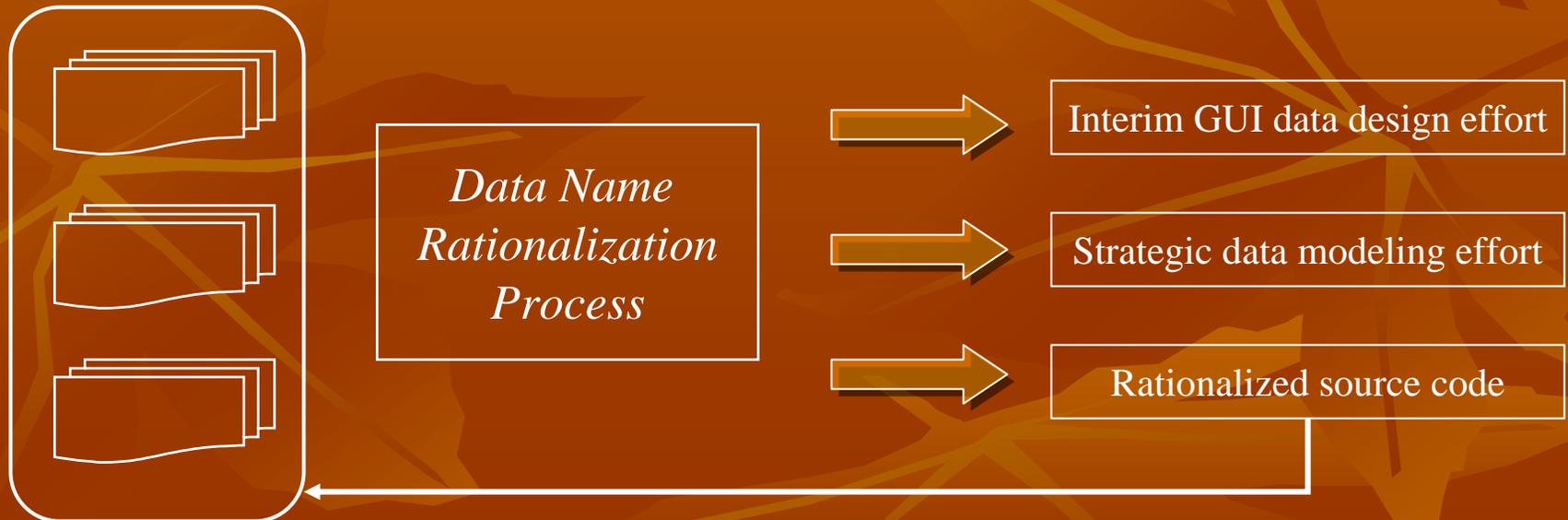
- Analyze and cleanup volatile programs and systems

# Application Stabilization Scenario

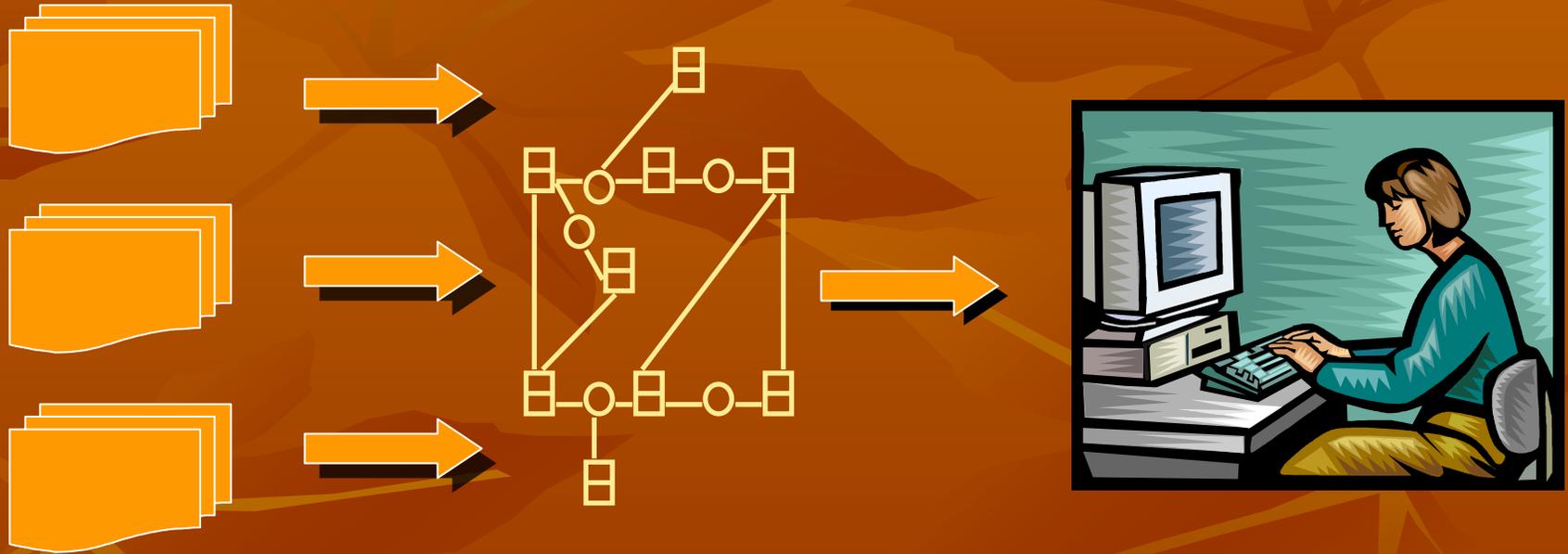
Convert, stabilize, structure, and/or slice selected modules in highly modified systems



Rationalize data definitions across systems of interest – determined during assessment



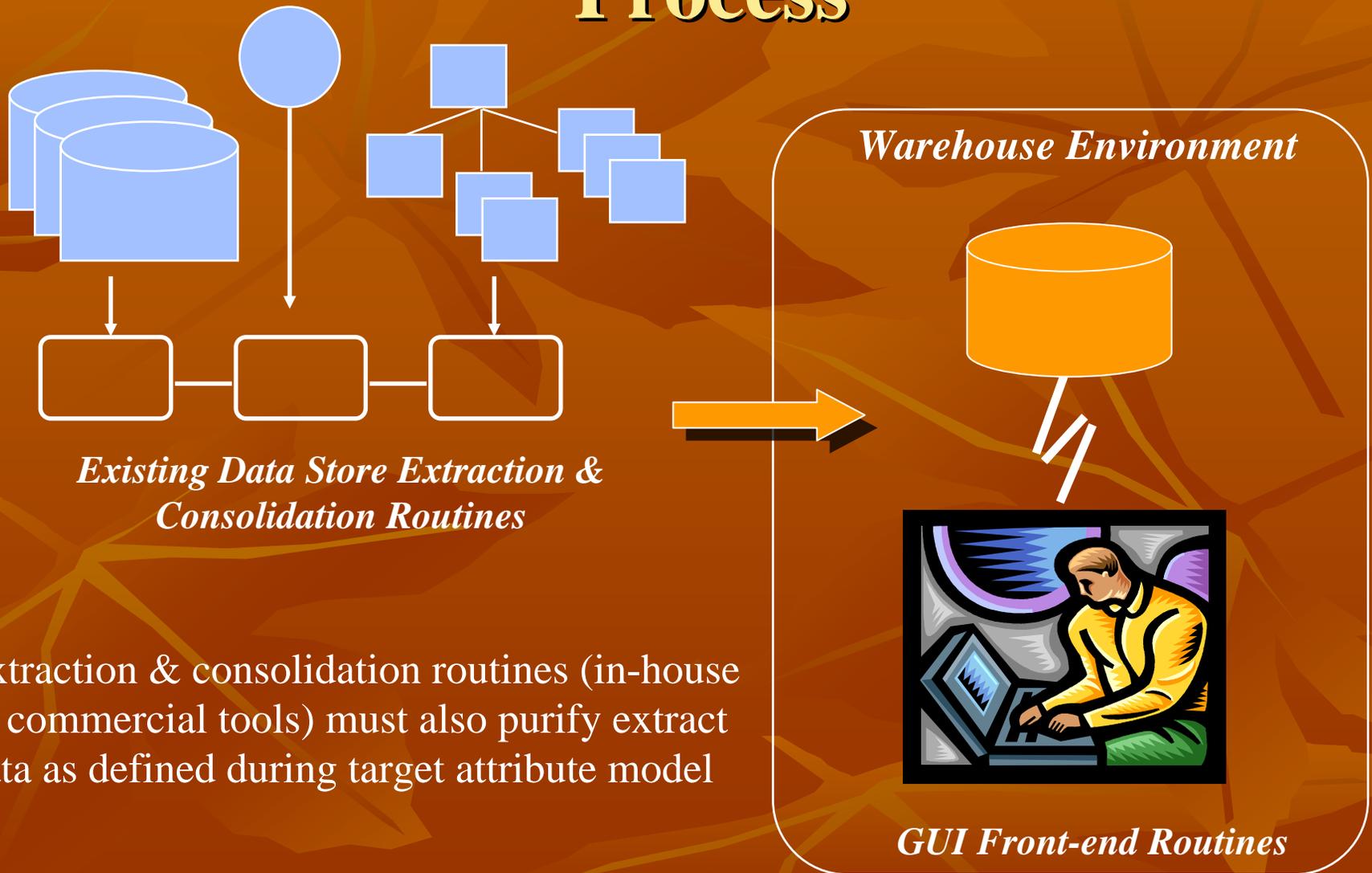
# Data Warehouse Concept



Selective migration / integration  
of cross-functional data

End user access to cross-  
functional data

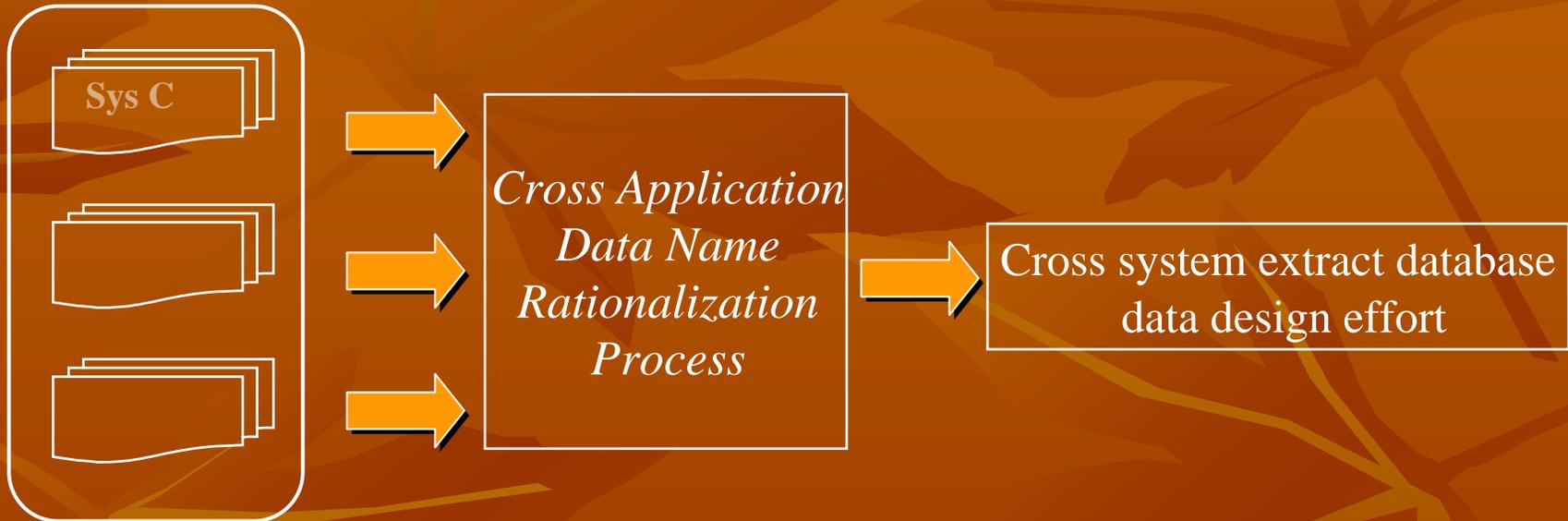
# Data Warehouse: Data Extraction Process



*Existing Data Store Extraction & Consolidation Routines*

Extraction & consolidation routines (in-house or commercial tools) must also purify extract data as defined during target attribute model

# Warehouse Scenario Composite Record Capture Rationalization



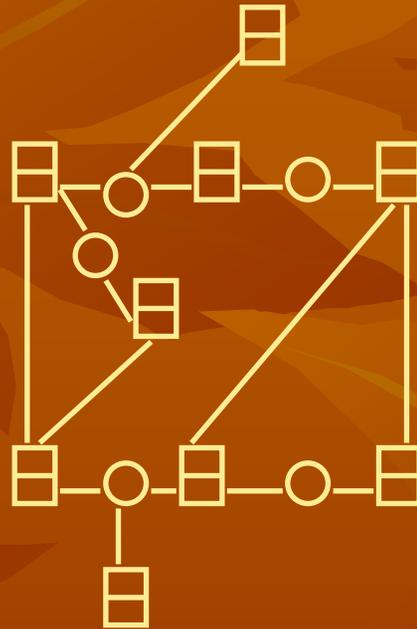
- *Data definitions are rationalized across systems of interest*
- *Composite views are used as basis for integrated extract database design*

# Developing Warehouse Database

Relational Modeling  
Refinements



Rationalized Existing  
Composite Records



Integrated Data Model  
to Support Host Resident  
Queries

- *Rationalized data views used to create integrated, relational extract database*
- *Extract database may reside on host or be downloaded to client environment*
- *Extract programs are developed to load & refresh extract data*

# Application Consolidation Scenario

## Issues:

- Redundant or stovepipe applications cannot share data
- Segregated functions hindering business agility
- Redundant applications, data, functions inflating operating costs

## Strategy:

- Design integrated target architecture
- Develop and perform phased migration of relevant applications

# Application Consolidation Approach

"Personnel System Example"

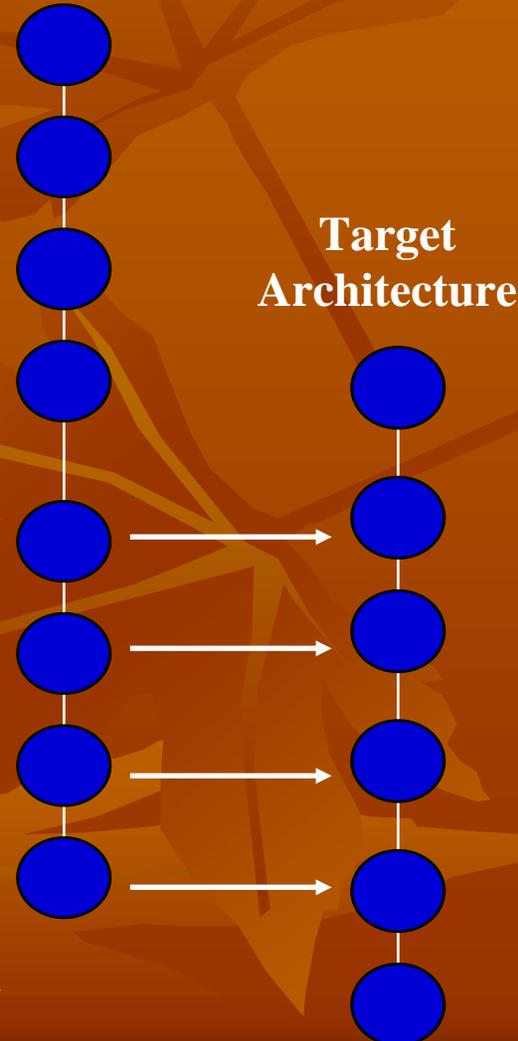
## POSITIONING STEPS

- Step 1 Code stabilization / restructuring
- Step 2 Data name rationalization
- Step 3 Redundancy reconciliation & re-aggregation
- Step 4 Client/server architecture finalization

## Modernization STEPS

- Step 5 Redevelop person management sub-system
- Step 6 Transform base payroll functions
- Step 7 Transform base pension functions
- Step 8 Transform base insurance functions
- Step 9 Finalize target functionality / deploy integrated system

Existing Architecture



# Phased Modernization & Consolidation: One Application From Many

## Existing Applications

### Baseline Applications



### Business Unit 1



### Business Unit 2



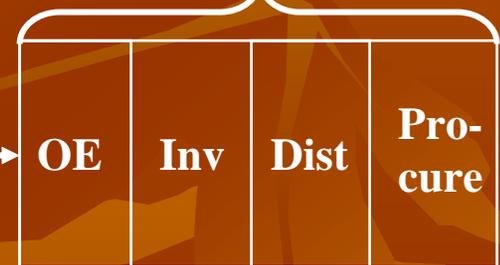
### Business Unit 3



## Integration & Modernization Tasks

- Integrate & automate common processes across business units
- Consolidate & redesign cross-functional data
- Migrate baseline systems to new architecture
- Migrate & consolidate subsequent business unit applications under new architecture
- Web-enable selected user interfaces as required

## Target Architecture



# ERP/Package Replacement Scenario

## Issues:

- Current systems not meeting business needs
- Packages under review

## Strategy:

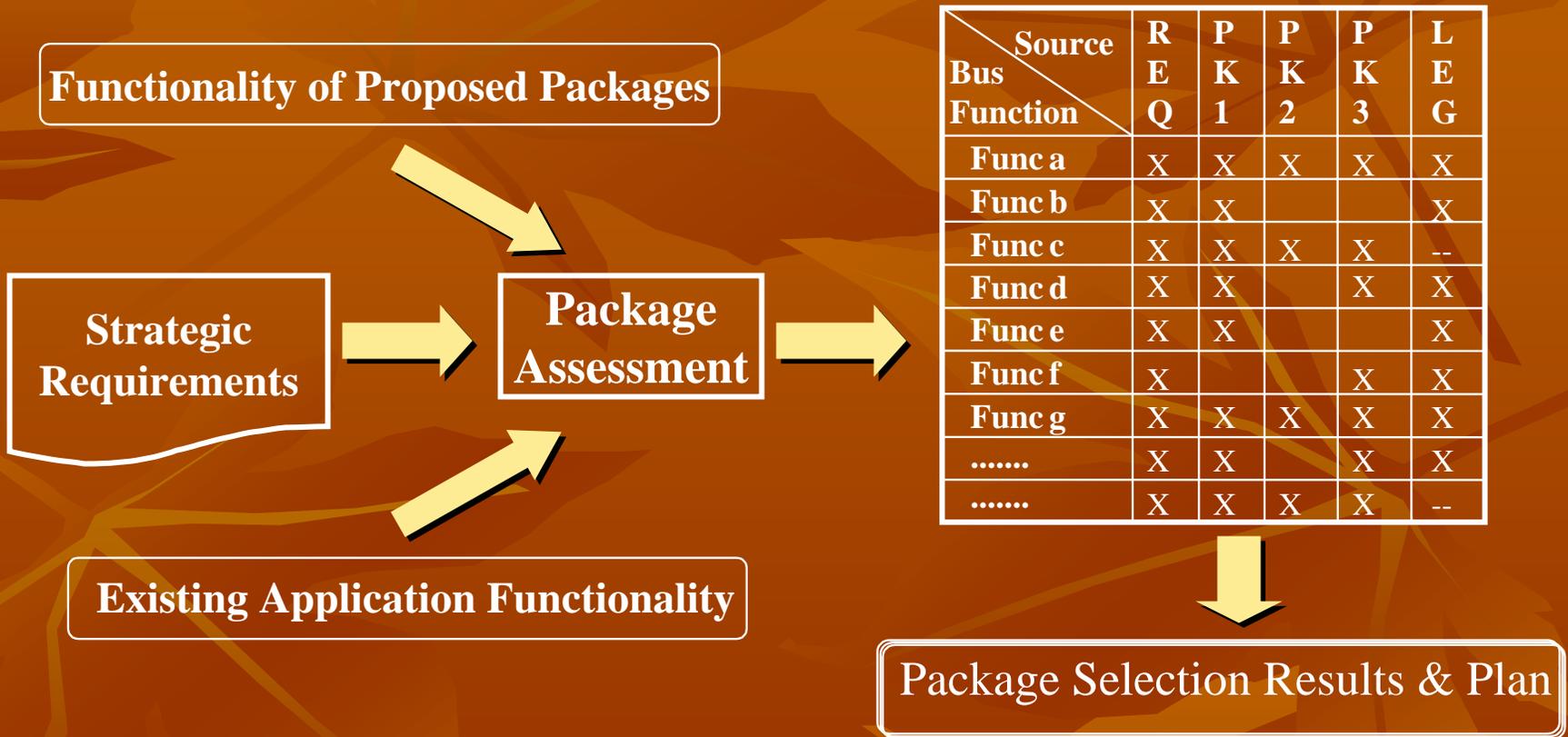
- Assess package candidate inability
- Map selected package to existing environment
- Perform phased package deployment and existing systems deactivation

# Basic Package Replacement Options

<b>Replacement Option</b>	<b>Utilizes Legacy Analysis Techniques</b>	<b>Reuses Legacy Application in Target Architecture</b>
1. Standard Package Selection	No	No
2. Package Selection with Gap Analysis	Yes	No
3. Standard Package Implementation	No	No
4. Package Implementation with Gap Analysis & Legacy Reuse	Yes	Yes
5. Legacy Consolidation & Migration to Target Architecture	Yes	Yes

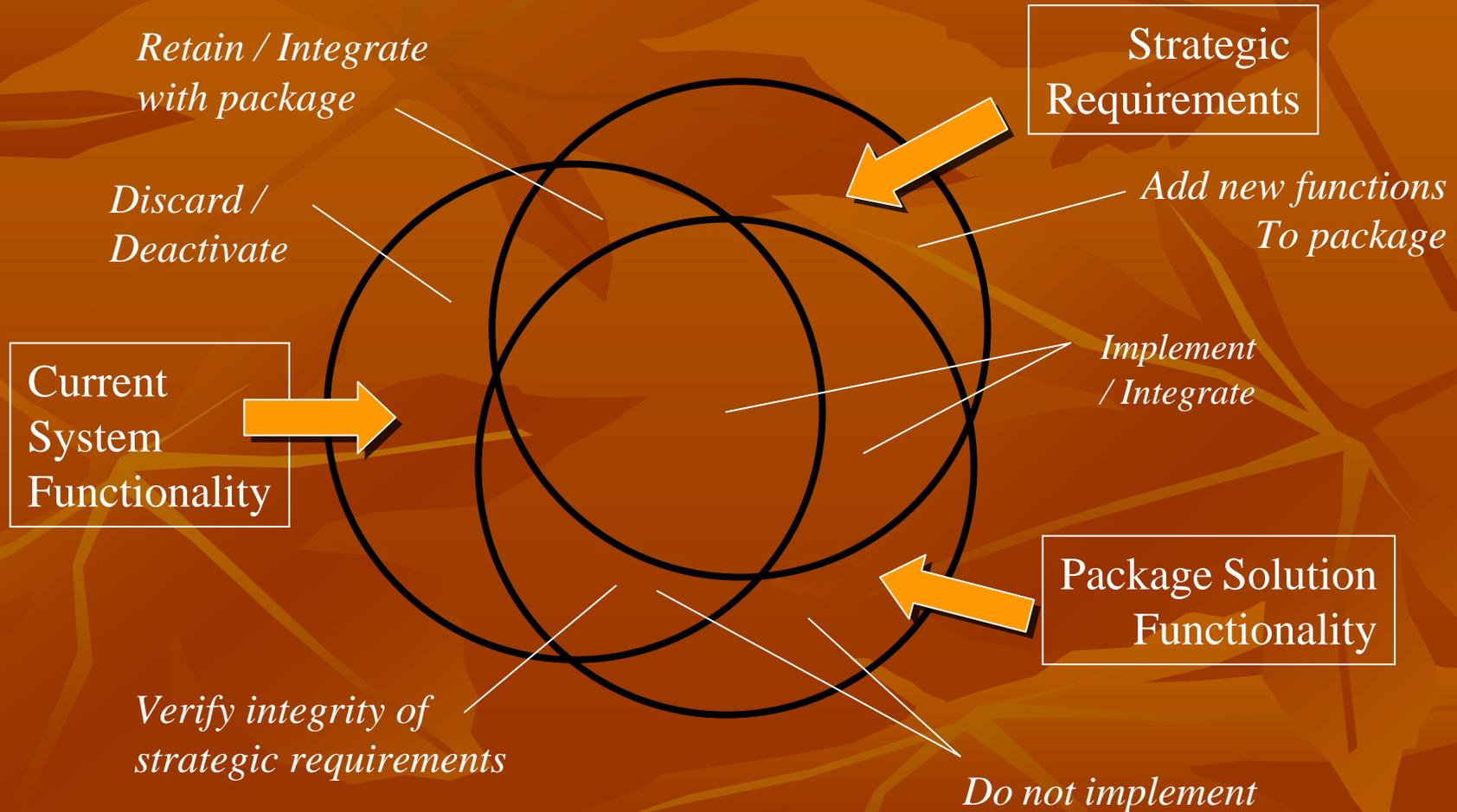
Package selection and replacement options can leverage or ignore existing architectures. Option 5 uses the existing architecture as the baseline for the new application as an alternative to licensing and implementing a package.

# ERP Selection: Mapping Existing Functions to Requirements & Candidate Packages



Multi-package assessment defines how packages map to requirements (REQ) and to legacy (LEG) functions.

# ERP Deployment: Mapping Existing Systems, Requirements & Package Functions



# Component-Based Migration / Scenario

## Issues:

- Desire to move existing functions to Web services
- Need to consolidate and reuse existing functionality

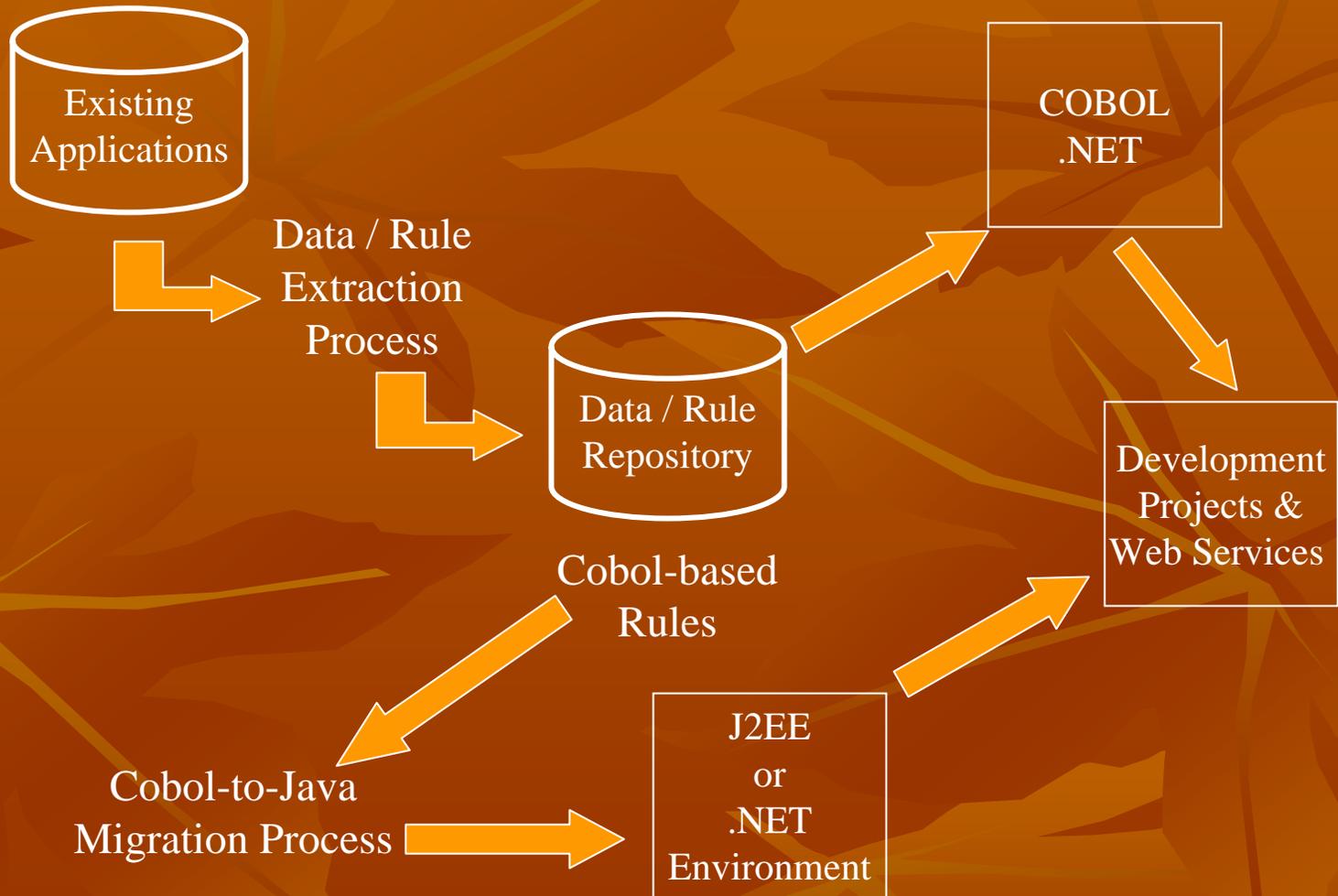
## Strategy:

- Define target architecture
- Perform phased existing systems consolidation
- Migrates to component-based environment
- Deploy Web services

# Component-Based Migration Tasks

- Address redundancy and reuse issues
- Perform consolidation as needed
- Select, extract and componentize rules
- Deploy under services-based architecture

# Componentization Strategy



# Componentization Scenario Case Study

- Situation: 30-year old inventory management system supporting 91 bases worldwide
  - 6000 people accessing Unisys 2200 mainframe daily
  - 1.6 million lines of code, maintenance became unfeasible
  - After 3 years' efforts, off-the-shelf software delivered only 27% functionality
- RescueWare® Solution:
  - Analysis and Knowledge Mining used to remove unused code and extract COBOL business rules
  - Java components generated from extracted COBOL business rules

Source: Relativity

# XI. Getting Started: Assessments

- Snapshot the enterprise / develop an architecture transition strategy
- Objectively gather facts and requirements
- Identify hypotheses for various systems
- Build an interim and strategic plan
- Develop a phased migration strategy

# Getting Started: Pilots

## Develop and implement a pilot project to:

- Test drive and refine a selected strategy  
Integrate redevelopment task set
- Get the application team on board  
Verify tool requirements
- Fine tune validation approach
- Adjust application proposal, contract, workplan and cost justification analysis

# **XII. Selling Management / ROI**

# Identifying Business-Driven Opportunities

- Maintenance is drawing too many resources
- System reliability is an issue
- Integration requirements exist
- Migration to model driven approach is planned
- Any funded project where existing systems could play a minor or major role

A portfolio list identifying systems, platforms, size, & strategic requirements should be kept as a baseline from which to identify redevelopment candidates

# Business Requirements Drive Value-Based ROI

- Reducing time-to-market for new products and services
- Shifting to a customer-driven philosophy
- Streamlining transaction flow across supply and distribution chains
- Creating flexible information systems to achieve business agility

# Current State of ROI on IT Projects

- 83% of companies do not track or measure (ROI) on technology projects.\*
- Companies that measure ROI track time lines and costs but not value.\*
- Value must be determined to assess ROI

*\* Building and Measuring ROI for Application Integration Projects, Julia King, Computerworld ROI, Brainstorm Business Integration Conference, October 29, 2001*

# Value-Driven ROI Principles

- Existing data structures and applications are highly valued information assets
- Application management and maintenance must consider business value - not just IT cost reduction
- Rewrite, package replacement and integration projects must consider existing information assets as key ROI factors
- Projects impacting existing environments should be cost justified in phases

# Business-Driven Value Assessment Shifts with Economic Conditions

- In 2000, top spending priorities at financial service firms: customer relations (37%), competition (18%), cost savings (14%) and efficiency (10%)\*
- By 2001, priorities shifted to cost savings (32%), efficiency (29%), market share (23%) and customer demands / requirements (15%)\*
- Customers fell in priority while cost saving priorities increased

*\*Budget Season is Here: Financial Services IT Spending is in for Big Changes, by John Hagerty, AMR Research, May 2001*

# Phased ROI: Reduces Project Risk & Helps Manage Shifting Priorities

- “Assessing the business value, related costs and annual projected TCO for a project should be incorporated into a cost model that quantifies ROI by project phase.”\*
- “If a project is put on hold or cancelled due to a shift in business priorities, that project will have delivered some value prior to being cancelled.”\*

\**Legacy Systems: Transformation Strategies*, William M. Ulrich, Prentice Hall, 2002

# Value-Based ROI for IT Projects\*

- Identify various options to meet a given set of business requirements
- Estimate revenue growth and / or cost savings to be gained from each approach
- Calculate cost of delivering each approach as the implementation cost plus annual operating costs
- Determine “net value” for each option: Subtract project costs from projected annual revenue growth and / or cost savings

*\*Legacy Systems: Transformation Strategies, William M. Ulrich, Prentice Hall, 2002*

# Information Asset Management & Maintenance ROI

- Application assets must be documented whether maintained in-house or offsite
- Proactive upgrade activities can streamline delivery of business value or prepare applications for strategic modernization
- Outsourcing / insourcing agreements should incorporate value-added activities - not just cost reduction

# Modernization Support for Outsourcing ROI

- IT cost cutting is currently the driving force behind outsourcing initiatives
- Value-added outsourcing ROI goes beyond cost cutting and includes:
  - Documenting applications in an open repository
  - Integrating applications using EAI technology
  - Upgrading, migrating and / or consolidating enterprise data and applications

# Phased ERP Deployment Leverages Existing Systems Modernization Disciplines

**Phased alternative to standard ERP deployment approach\***

- Migrate existing applications to an industry-standard database
- Migrate application front-ends to a client/server environment
- Add core ERP functionality to this environment
- Integrate additional ERP features with best-of-breed solutions

*\*ERP: Time for a Rethink*, Richard Todd, a white paper from [www.max-international.com](http://www.max-international.com), June 1998

# **XIII. Summary / Close / Questions**



# Existing Architecture Modernization Summary

- Retains and reuses valuable business knowledge from existing applications
- Augments an enterprise's ability to deliver strategic, time-critical projects
- Reduces risks and increases odds of success for new application initiatives
- Achieves strategic goals more quickly, more reliability and at a reduced cost