

Modernizing RPG & 2E Apps Using Design Recovery

© 2007 Databorough Ltd, Weybridge Business Centre, 66 York Road, Weybridge, KT13 9DY, United Kingdom, Tel: +44 (0)1932 848564, Fax: +44 (0)1932 859211, E-mail: info@databorough.com, Website: www.databorough.com

Stuart Milligan – Databorough
stuartm@databorough.com

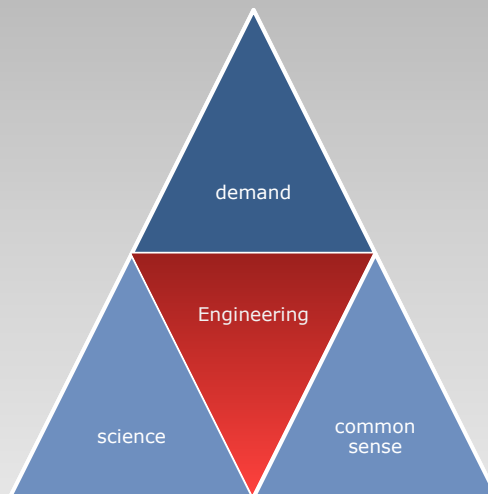


- Background
- Motivation
- Mechanics
- Methodology

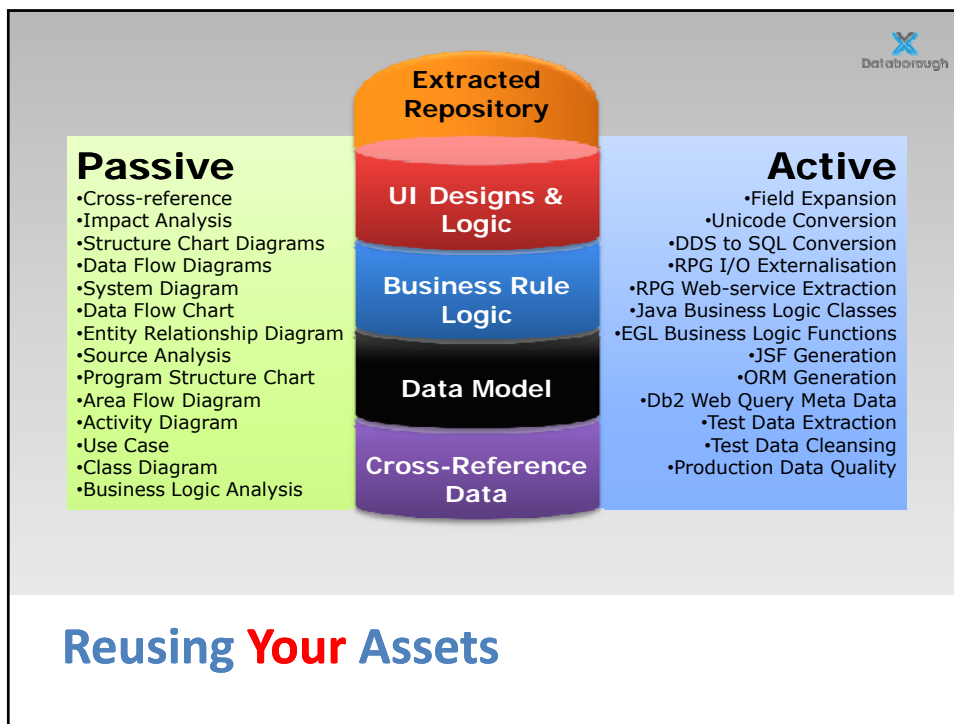


- 80's cross referencing tools
- Mid 90's Reengineering Automation
Graphical analysis & Documentation
- Early 00's Model Based Extractions
Data Model
- 2005 Model Based Extractions
Business Rule Logic
- 2005 Model Based Extractions
UML & DDL
- 2009 Database Modernization & Reengineering

Technological Milestones



Using an Engineered Approach



500+ Large Scale Reengineering Projects

Concepts Guide

Modernizing System i Applications Using Design Recovery

<http://www.databorough.com/designrecovery>

IBM
Modernizing and Improving the Maintainability of RPG Applications Using X-Analysis Version 5.6
 Graphically analyze application architecture
 Extract business rule logic and application data models
 Migrate existing code into the MVC architecture
 List of Authors: ...
Redpaper
 ibm.com/redpaper

Cumulative Knowledge Base

R & D Partners

Typical System i Application

- ▶ Hundreds to thousands of RPG or COBOL Programs
- ▶ Aging Systems – 30 to 40 years old, heavily modified
- ▶ Monolithic Programming model
 - Screen
 - Database
 - Business Logic
- ▶ Little or no Documentation
- ▶ Original Designers no longer available

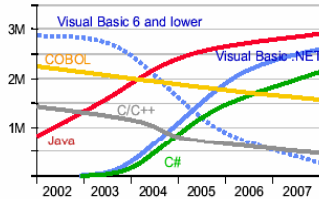


- ∞ Improve the responsiveness of development
- ∞ Improve long term maintainability
- ∞ Increased access to developers skilled in popular languages
- ∞ Increase availability of tools and application packages
- ∞ Improve application accessibility through open interfaces
- ∞ Improve portability of applications across platforms

Why Modernize?

Developers

From an estimated worldwide market size of 7 million "professional" developers



Gartner



Java is Critical for the Industry

The world's most widely used programming platform

What percent of your programming time will you spend using the following languages TODAY? (filtered for responses of 1% or greater)	Population	Percent of Cases
Java	9,007,346	61.6
C/C++	7,194,179	49.2
XML	7,194,179	49.2
C#	5,044,699	34.5
Visual Basic/Visual Basic.NET	4,913,098	33.6
Scripting Languages	10,426,711	71.3
Other	7,428,130	50.8

Global Developer Population and Demographics Survey, Volume I, © 2009 Evans Data Corp.



Developer Population?

Modern vs. Traditional Applications

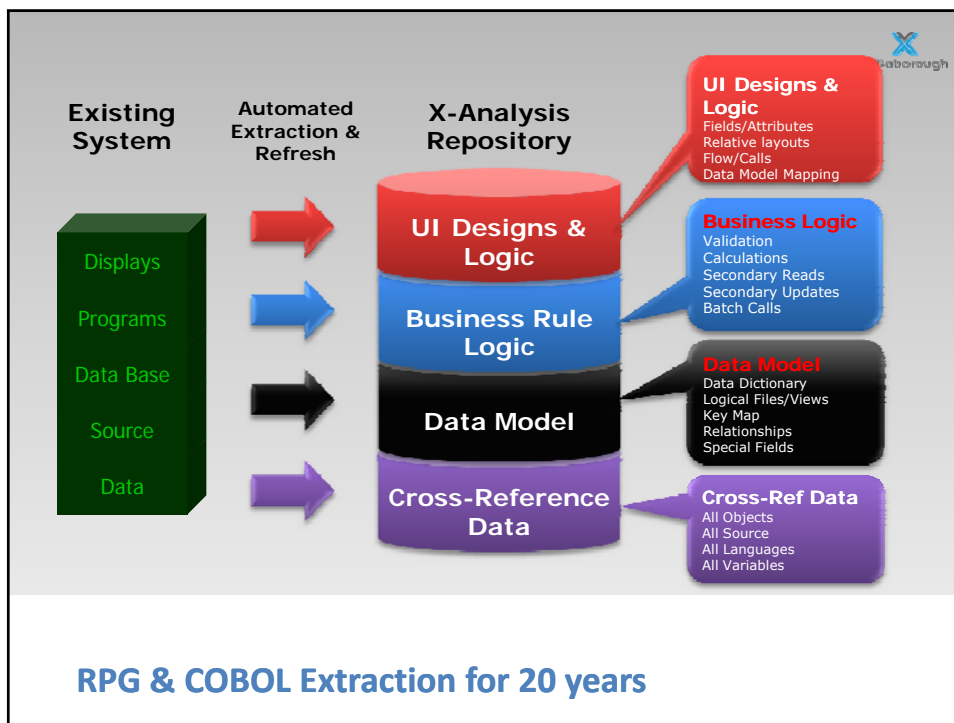
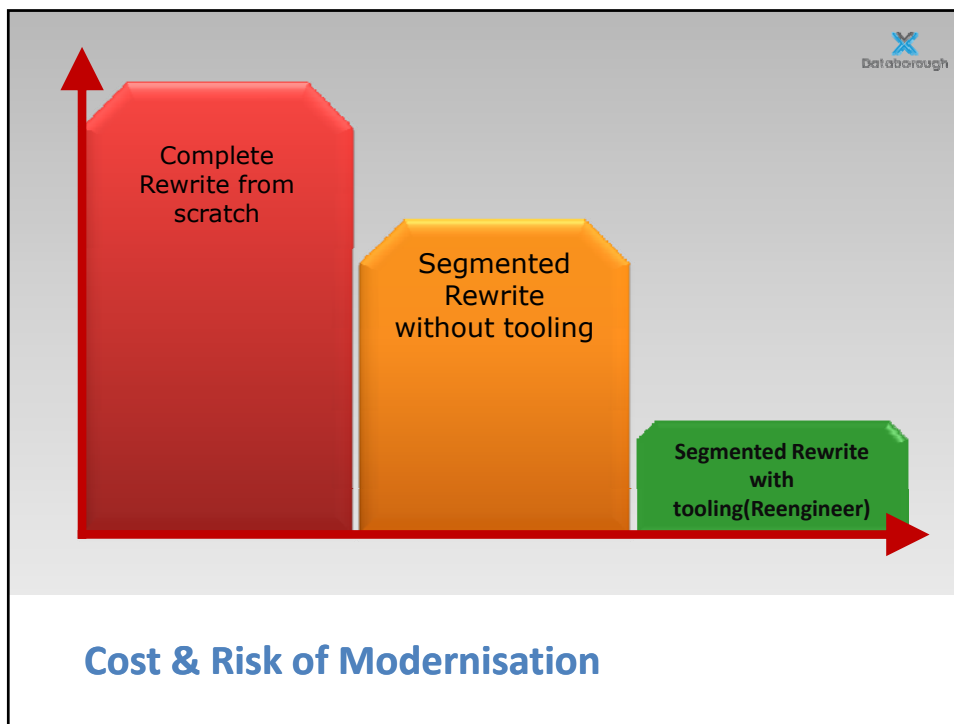
Characteristic	Modern Application	Traditional Application
Structure	Modular, reusable, component based	Monolithic, intertwined, repetitive code across modules
Componentization	Functionally cohesive	Arbitrary
Performance	Highly scalable	Limited scalability
User Interface	Multiple	5250
Portability	Diverse platforms	System i and its predecessors

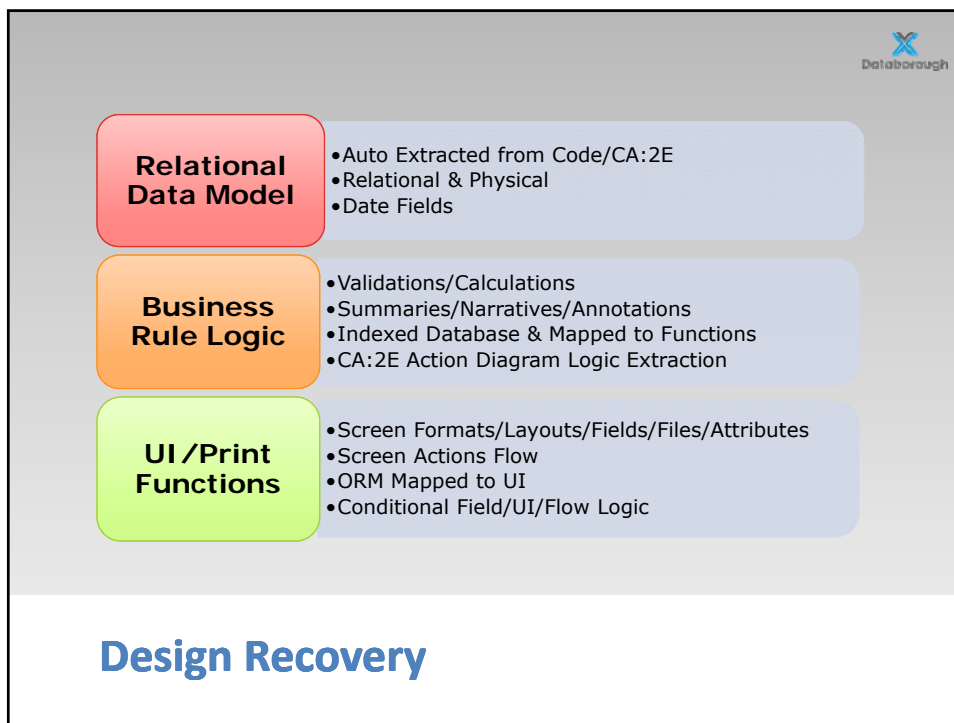
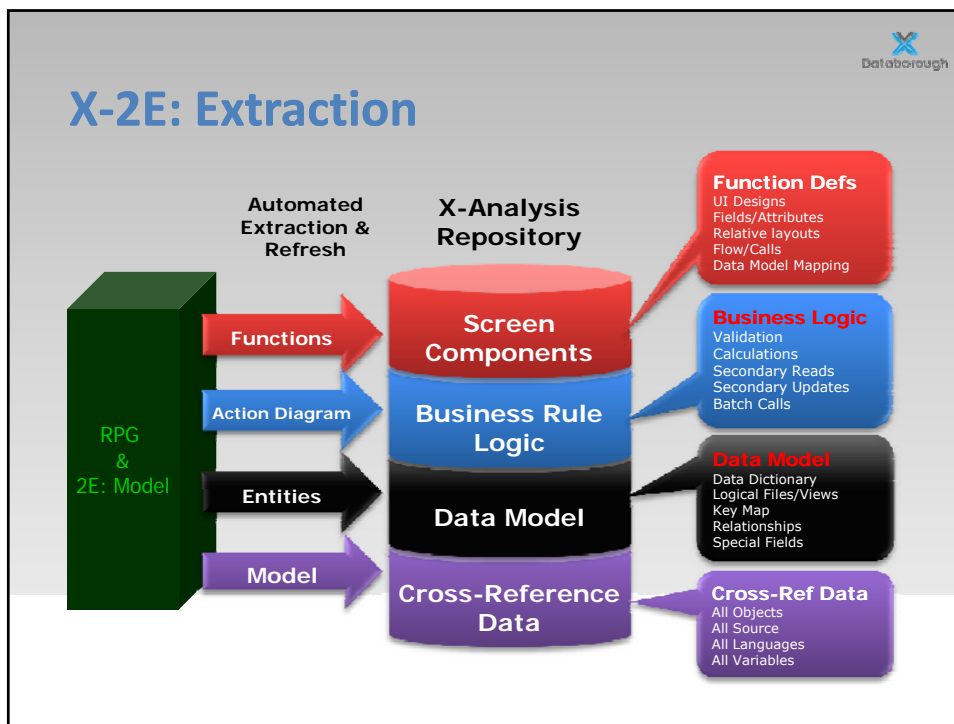


- ☛ Inherit Legacy Problems
- ☛ Not A Modern Application (Actually Legacy Emulation)
- ☛ **No-one recognizes the output**
- ☛ Virtually Un-maintainable
- ☛ Difficult To Retain Staff
- ☛ Poor Performance
- ☛ Difficult to Scale
- ☛ Not a Long Term Investment



Code Conversions





Design Recovery Documentation

Business Logic Analysis

Databorough

CA 2E data model as interactive diagram with relationship details

Variable Where Used | Data Content Diagram | Outline | DMD Details

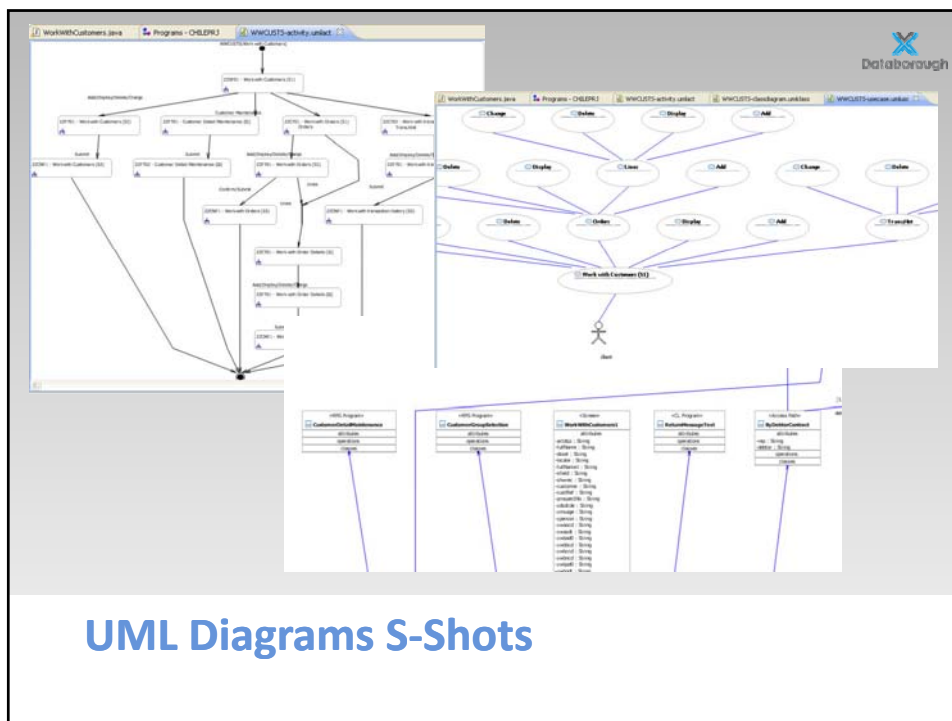
DMD Relations for DEM01/AEQCP, Total Relations: 11

Ref No.	Dependent File	Physical File	Relation Type	Parent File	Physical File	Dependent Fields	Parent Fields
1	PRODUCTS FOR CLIENTS	Physical file	OWNED BY	CLIENT	Physical file	001 Client Number	001 Client Number
2	PRODUCTS FOR CLIENTS	Physical file	OWNED BY	ROLES FOR PRODUCT	Physical file	032 Product Type, Role code	032 Product Type, Role code
3	ORIGINATOR	OGN Physical file	REFERS TO	PRODUCTS FOR CLIENTS	Physical file	001 Client Number, 032 Pr...	001 Client Number, 032 Product Type, ...
4	PPSR TRANS DEBTOR	PPD Physical file	REFERS TO	PRODUCTS FOR CLIENTS	Physical file	001 Client Number, 032 Pr...	001 Client Number, 032 Product Type, ...
5	CLIENTACCOUNT ARCHEVECAA	Physical file	OWNED BY	PRODUCTS FOR CLIENTS	Physical file	001 Client Number, 032 Pr...	001 Client Number, 032 Product Type, ...
6	CLIENT BILPAIN PUBLIC CSP	Physical file	REFERS TO	PRODUCTS FOR CLIENTS	Physical file	CSP Client, 032 Product Ty...	001 Client Number, 032 Product Type, ...
7	DISTRIBUTOR	DST Physical file	REFERS TO	PRODUCTS FOR CLIENTS	Physical file	001 Client Number, 032 Pr...	001 Client Number, 032 Product Type, ...
8	ACCOUNT ROLE ADDRESS	ARA Physical file	OWNED BY	PRODUCTS FOR CLIENTS	Physical file	001 Client Number, 032 Pr...	001 Client Number, 032 Product Type, ...

Data Model Analysis

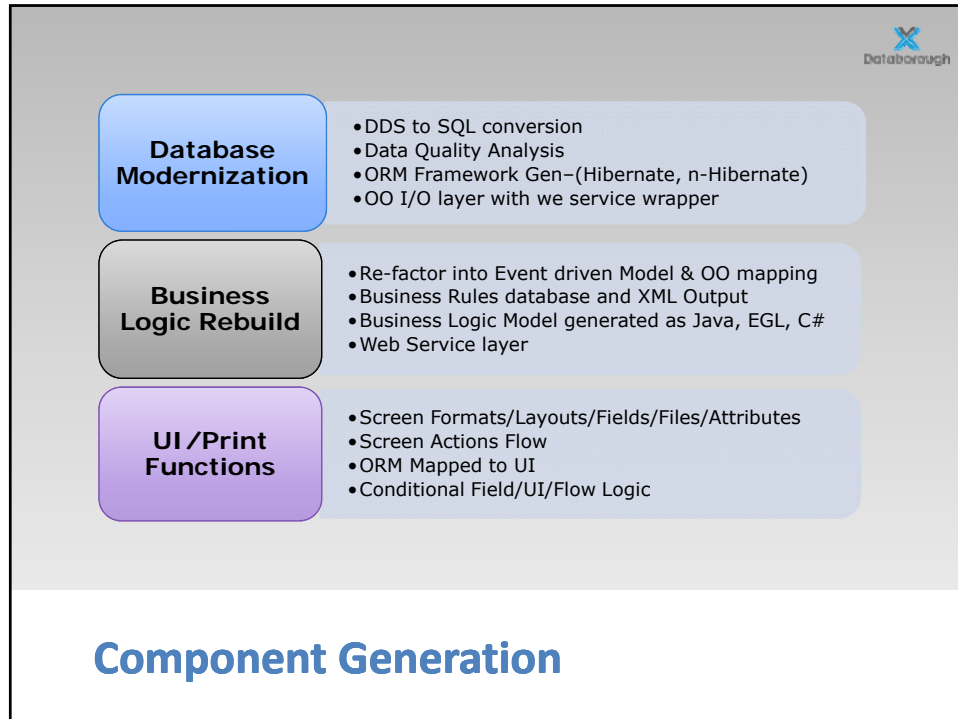
Databorough

UML Diagrams & UML Extensions



- ∞ Scientific Forensics
- ∞ System wide perspective
- ∞ Visualization
- ∞ Non Expert Inclusion
- ∞ Explicit Information
- ∞ Quality of Communication

Summary Benefits of Analysis



What is MVC?

Model-View-Controller (MVC)

- **Model:** Underlying data objects and Business Logic
- **View:** The user interface
- **Controller:** Takes requests from View; passes commands to Model; chooses resulting View

MVC means keeping these functions separate.

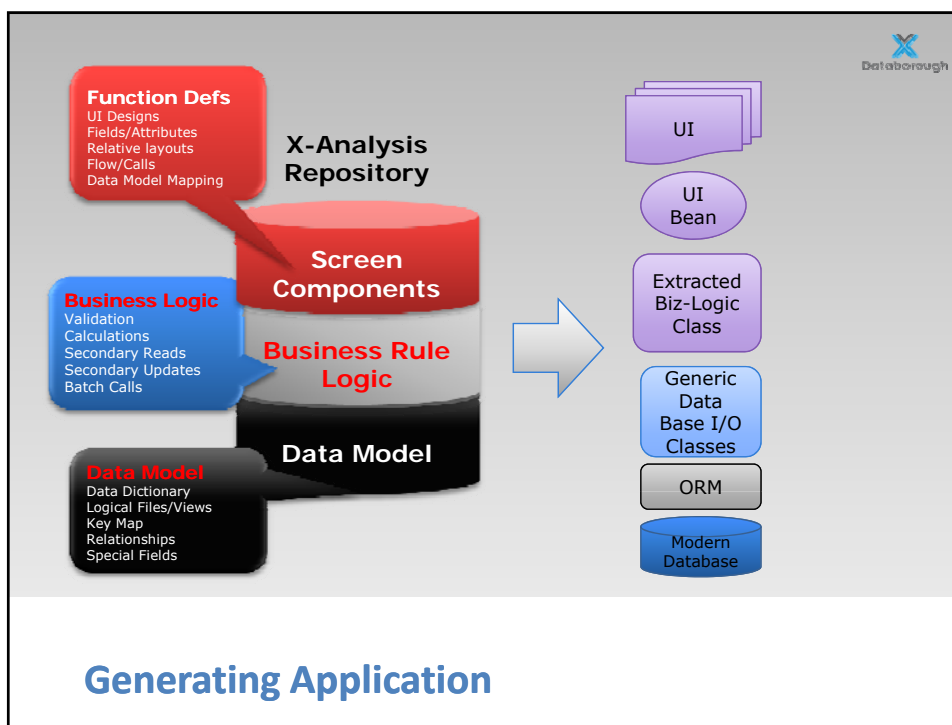
Why MVC?

The shift to MVC

Brian Goldfarb, director of developer and user experience platforms at Microsoft.

"In the same vein, Microsoft has recognized the importance of the MVC pattern to developers, and it views it as a proven, robust way of making applications, The needs of customers have changed now."

- Reduce Maintainable Code Base
- Agile Maintenance
- Consistent Architecture
- Lower Training Costs
- Optimise Performance
- Portability & Scalability
- Modular separation of function
- Easier to maintain
- Data Integrity





- Each Screen Format has business logic indexed
- UI Events Removed from code
- Indicators and UI messaging converted to standard messaging
- I/O Standardised into generic components in Java/C#
- Procedural Code is converted to event driven code
- Standardised Parameters and API's
- Variables Rationalised back to DB Fields if Possible
- Each Screen Format converted to JSF
- Java/EGL Controller Bean/Handler per screen
- Uses a open source/.Net application framework as necessary or useful

How is design Re-factored?

Web - JSF JEE MVC

Project Methodology

- **Project Discovery**
- **Proof of Concept**
 - Migration of a defined subset of functionality to primary desired variant. Customer can then inspect and benchmark actual migrated code.
- **Model Preparation**
 - Complete documentation
 - Application Subdivision
 - Model/Application clean-up/refactor
- **Test Case Development**
- **Migration**
- **Training**
- **Global & Specific Customization**

On Going Changes during Project

- **High Level Automation minimises impact of change**
- **Precise impact analysis upfront in tooling**
- **Last minute input**

Testing

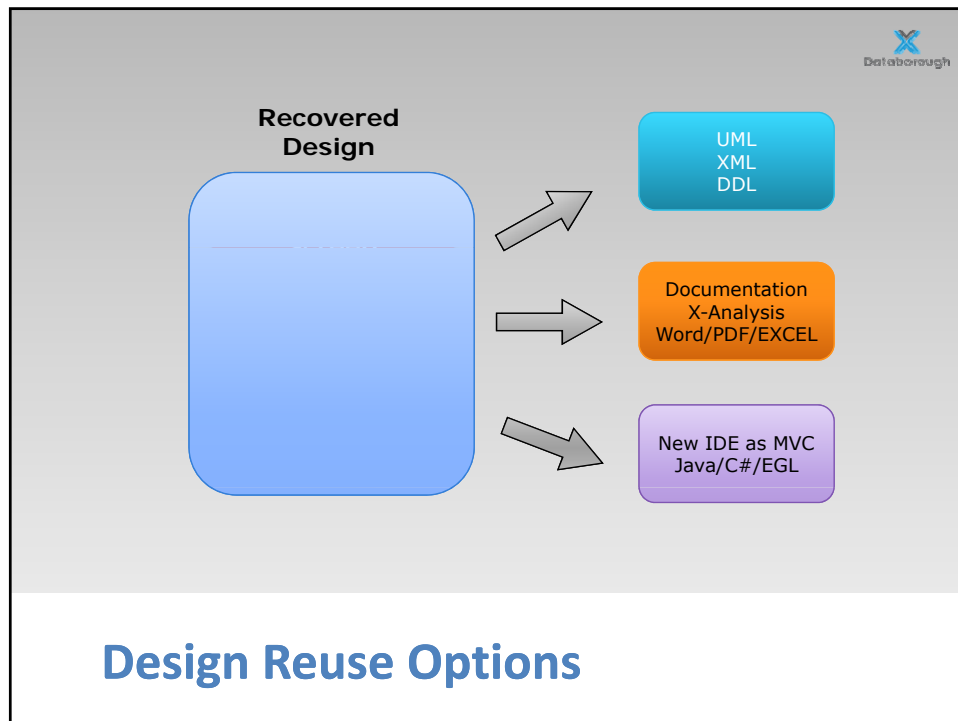
- **Distributed and so focussed and more manageable**
- **Consistency – code is consistent so results are more predictable**
- **Semi automated – OO and Event code model allows for automated unit testing**

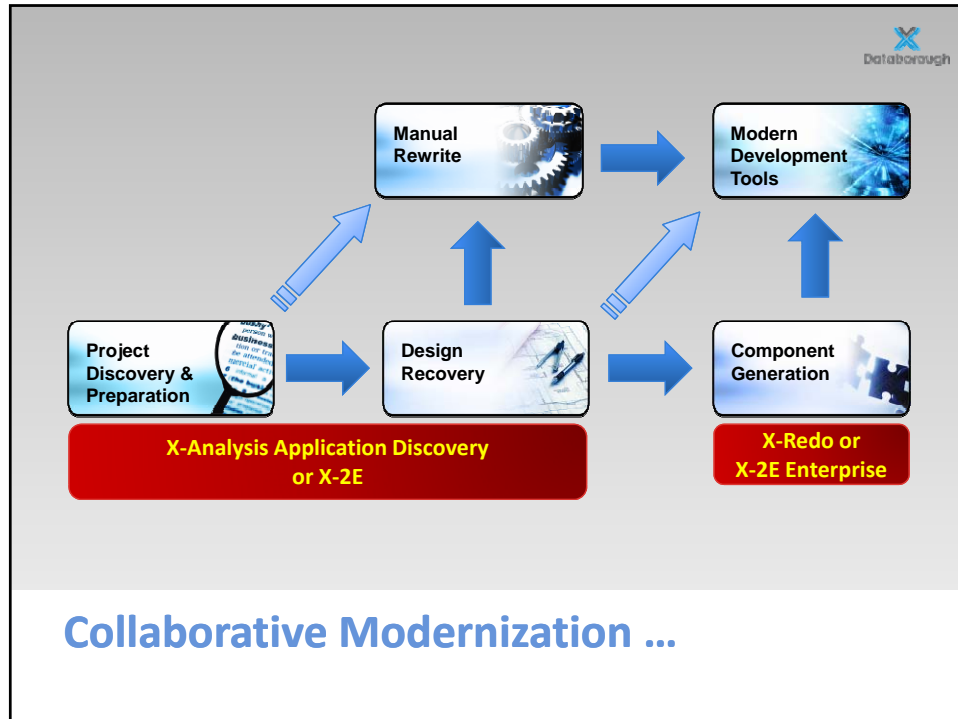
Future Maintenance

- **Potential for Automation – Consistent structure**
- **Distributed Architecture/Modular/Known(you know where to make the change or enhancement)**
- **Object Oriented - reuse**
- **Semi automated – generation from models**

Stability & performance

- **Distributed Architecture**
- **Modular Scalability**
- **Modern frameworks – Hibernate/Spring/nHibernate**
- **Automated benchmarking**





- ∞ Inherit Valuable Assets
- ∞ Exact and Accurate Knowledge Transfer
- ∞ Lower Risk
- ∞ Re-Use Existing Skills
- ∞ Configurable
- ∞ Gradual, Granular & Scalable
- ∞ Long term result - *maintainable*

Why Use Design Recovery?



Questions & Answers

Thank You

धन्यवाद
Hindi

多謝
Traditional Chinese

ขอบพระคุณ
Thai

Спасибо
Russian

Gracias
Spanish

Merci
French

Thank you for viewing our presentation
For more information or to book a demo or trial, please contact us
below:

info@databorough.com
www.databorough.com
North America: 705 458-8672
Europe: +44 1932 848564

شكراً
Arabic

Obrigado
Brazilian Portuguese

Grazie
Italian

多謝
Simplified Chinese

Danke
German

நன்றி
Tamil

ありがとうございました
Japanese

감사합니다
Korea



Refactoring Legacy Applications


© 2007 Databorough Ltd, Weybridge Business Centre, 66 York Road, Weybridge, KT13 9DY, United Kingdom, Tel: +44 (0)1932 848564, Fax: +44 (0)1932 859211, E-mail: info@databorough.com, Website: www.databorough.com

Stuart Milligan – Databorough
stuartm@databorough.com



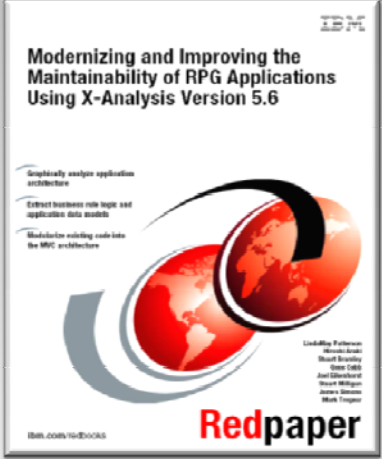
- Introduction
- Defining the Problem
- Resolving the problem
- Case Studies
- Summary

Agenda
















500+ Large Scale Reengineering Projects

Concepts Guide
 Modernizing System i Applications Using Design Recovery
<http://www.databorough.com/designrecovery>



Cumulative Knowledge Base

		
		
	 Global Services  Lab Services	
		

Mainstream



- 80's cross referencing tools
- Mid 90's Reengineering Automation
Graphical analysis & Documentation
- Early 00's Model Based Extractions
Data Model
- 2005 Model Based Extractions
Business Rule Logic
- 2005 Model Based Extractions
UML & DDL
- 2009 Database Modernization & Reengineering

Technological Milestones

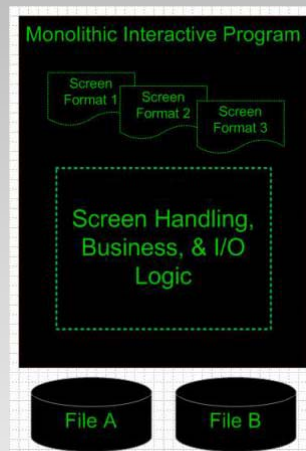


- Application Architecture
- Code Complexity & Growth
- Technology Evolution & Standards
- Knowledge Dispersion & Loss

Defining The problem



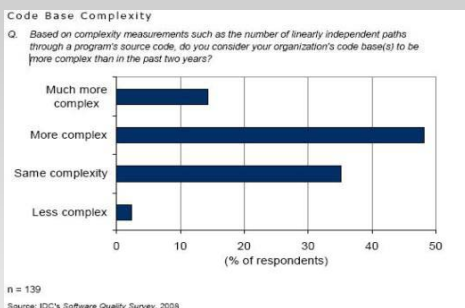
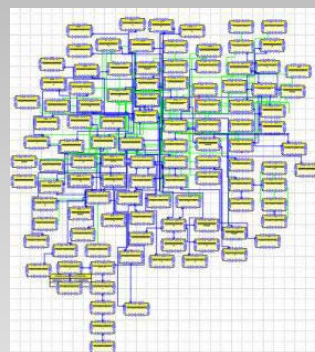
- Monolithic
 - Screen handling
 - Database I/O
 - Business Logic/Conditional Flow
- Inheritance & Erosion
 - Legacy - S36/S38 int/multi-frm/mbr files
 - ERP
 - Acquisition & growth
 - Regulatory




Application Architecture



- In-source, Outsource, Offshore
- Interfacing & Integration
- Regulatory Demands
- Competitive Business Pressures

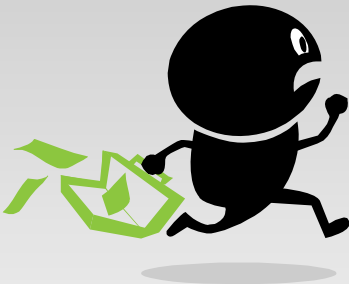


Code Complexity & Growth



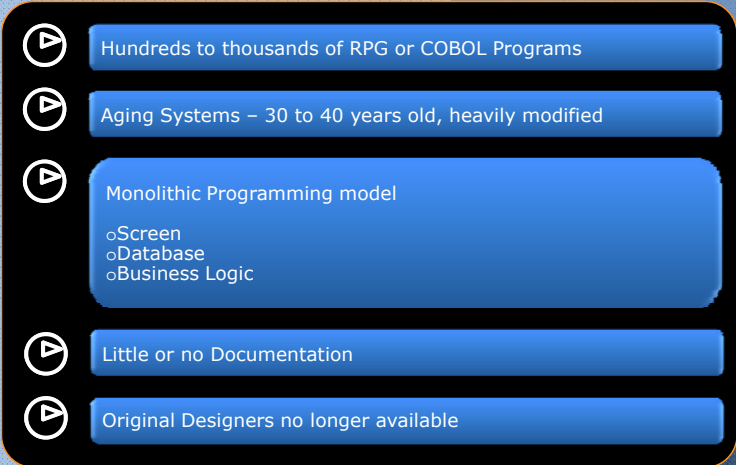
- + RPG II/III/IV/Free
- + GOTO/Indicators/Subroutines/procedures
- + Internally/Externally described files
- + OpenQueryFile/SQL/RLA
- + ILE
- + Naming conventions
- + Documentation/Comments
- + SCM
- + 4GL/Templates/Copying

= Aaaaaahhh!!!



Technology Evolution & Standards

Typical System i Application



- ▶ Hundreds to thousands of RPG or COBOL Programs
- ▶ Aging Systems – 30 to 40 years old, heavily modified
- ▶ Monolithic Programming model
 - Screen
 - Database
 - Business Logic
- ▶ Little or no Documentation
- ▶ Original Designers no longer available



- Application Discovery
- Refactor
- Automation
- Quick Wins & Rewrites

Resolving The problem



- Documentation
- Visual Analysis
- Functional Subdivision
- Communication
- Validation

Application Discovery

Object Where Used - Usage References for CUSF, Total Objects: 45

Object	Text
CUSCPY	Customer Copy
CUSFL1	Sites by Name
CUSFL2	Sites by Status
CUSFL3	Sites by Number
CUSFL6	Sites by Dist & Status
CUSFL6	Sites by Dist & Name
CUSFL7	Sites by Last Crt Date
CUSFL8	Sites by Next Crt Date

Usage Reference	Seq No	Condition	Field	File/Program	Rule
AFNWEFF	0267.00	IF LCL.FSM_Code <> *BLANK:			
ATBZPVP	0168.00	DTL.Fund_Source_Mech_Code_To = LCL.FSM_Code:			
AR000FP	1163.00	LCL.FSM_Code = PAR.AFP_Surrogate:			
AR000FP	1211.00	PAR.Audit_Heading_Line_1 = 'Fund Source :' + LCL.FSM_Code (2):			
ASDQZ10	0019.00	A	DQCZCI		TEXT ('FSM Code')
ASDQZ10	0062.00	A	R DQCZCI		
ASDQZ11	0019.00	A	DQCZCI		TEXT ('FSM Code')
ASDQZ11	0062.00	A	K DQCZCI		
ASDQZ12	0018.00	A	DQCZCI		TEXT ('FSM Code')
ASDQZ12	0017.00	A	DQCZCI		TEXT ('FSM Code')
ATBZPVP	0237.00	LCL.Attachment_Description = 'From' + BCD.FSM_Code (1):			
ATBZPVP	0092.00	DTL.FSM_Code = PAR.FSM_Code:			
ATBZPVP	0201.00	IF DTL.FSM_Code <> *BLANK:			

variable/field where used using CA 2E AND mnemonic names

Documentation

Business Rules for CNTCMaint, Number of Lines: 6

Source Member	Narration	Type	Rule No	Field	File/Program	Rule
CNTCMaint	If the field ZUSERNM is blank then it is invalid.	V	00001	USERNM	CNTACS	You must enter a contact name
CNTCMaint	If the field ZTELNO is not blank, verify the field ZTELNO against '0123456789'. If other values found then the field "Phone" is invalid.	V	00002	TELNO	CNTACS	The telephone no. is invalid.
CNTCMaint	If the field ZFAXNO is not blank, verify the field ZFAXNO against '0123456789'. If other values found then the field "Fax No." is invalid.	V	00003	FAXNO	CNTACS	The fax no. is invalid.


```

C* Telephone number
0169.96 =!BRC* If the field ZTELNO is not blank , verify the field ZTELNO against '
0169.97 =!BRC* 0123456789'. If other values found then the field "Phone" is invalid.
0169.98 %!BRC* Business Rule No. CDEMO/QRPGLESRC/CNTCMaint/170 Validation.
0169.99 @!BRC* V00002 CNTACS TELNO The telephone no. is invalid.
0170.00 B>!BRC if ztelno <> *blanks
0171.00 ->!BRC ' 0123456789' check ztelno z1
0172.00 ->!BRC if %found
0173.00 ->!BRC eval *in34 = *on
0174.00 ->!BRC eval msgid = 'OEM0014'
0175.00 ->!BRC callp(e) rtmsgtext(msgid:errormsg)
0176.00 ->!BRC eval valid = *off
0177.00 ->!BRC leavesr
0178.00 ->!BRC endif
0179.00 E>!BRC endif
    
```

Documentation – Business Rules


The screenshot displays a software interface with several panes. The top pane shows a 'Data Model Diagram for ATBRL1/AAQZPP, Total Objects: 12' with various data objects and their relationships. The middle pane shows a 'Program Centred Data Flow Diagram for ATBQFP, Total Objects: 22' with a flow diagram. The right pane shows a 'Data Content Diagram for AZATBDFP51' with a hierarchical tree structure. The bottom pane shows a 'Screen Flow Diagram for ATBQFP, Total Objects: 54' with a flow diagram. The interface also includes a 'Print display file' window showing a report for 'Maintain Funding Source Mechanisms'.

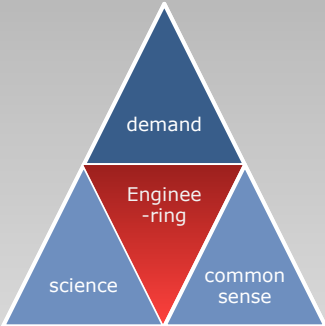
Visual Analysis – Data Model, OWU and field analysis

The screenshot displays a software interface with several panes. The top pane shows a 'Screen Design for ATBQFP - #KDD011' with a report for 'Maintain Funding Source Mechanisms'. The middle pane shows a 'Screen Action Diagram for AZATBDFP51' with a flow diagram. The right pane shows a 'Data Content Diagram for AZATBDFP51' with a hierarchical tree structure. The bottom pane shows a 'Screen Fields' table with field details for AZATBDFP51.

Label	Field	File	Line	Column	Attribute	Type	Length	Decimal Post
Code	FSM_Code	ASDQREP	4	24	Both	A	6	0
Description	FSM_Description	ASDQREP	5	24	Both	A	50	0
	DTL_Text_21	WorkField	7	2	Output	A	21	0
	FSM_Org_Link_Type_ID	ASDQREP	7	24	Both	A	1	0

Visual Analysis – Deconstructing UI Layer





Basic Code Refactor

- Create Subroutines and in the process, elimination of GOTO / CAB / TAG statements.
- Replacement of left hand indicators used to control program logic flow with IF, THEN, ELSE type logic statements.
- Removal of primary / secondary file structures – eliminate use of RPG cycle
- Inclusion of D-Spec statements as created by ILE conversion process
- Extraction of messages to a message file format
- Externalization of inline tables (compile time tables/arrays)
- Replacement of left hand indicators with named indicators (to include Level indicators L1-L9, MR, U indicators)
- Consistent use of up-to-date RPG ILE data handling capabilities
- Use of RPG ILE standard functions wherever possible, such as %EOF, %FOUND, %LOOKUP, etc.
- Replacement of ADD, SUB, MULT, DIV with EVAL whenever practical
- Removal (and archival) of commented out code


Database Refactor

- DDS to SQL
- Field Expansions
- UNICODE


Advanced

- I/O Externalization
- Subroutine to Procedures
- Business Logic Externalization

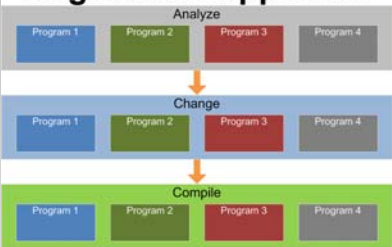
Refactor



Manual Approach



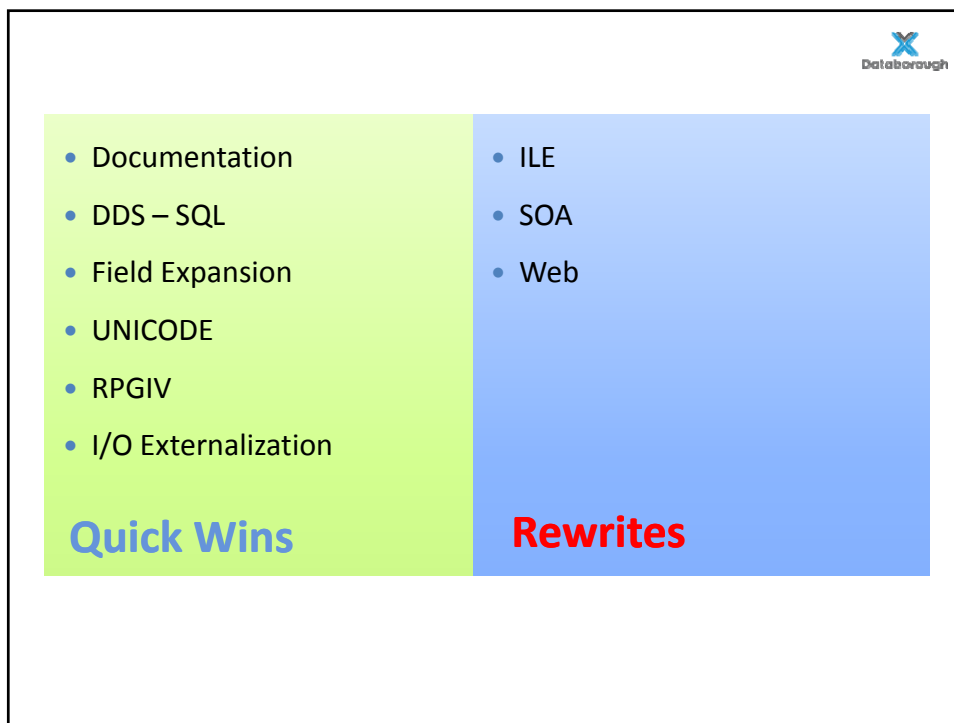
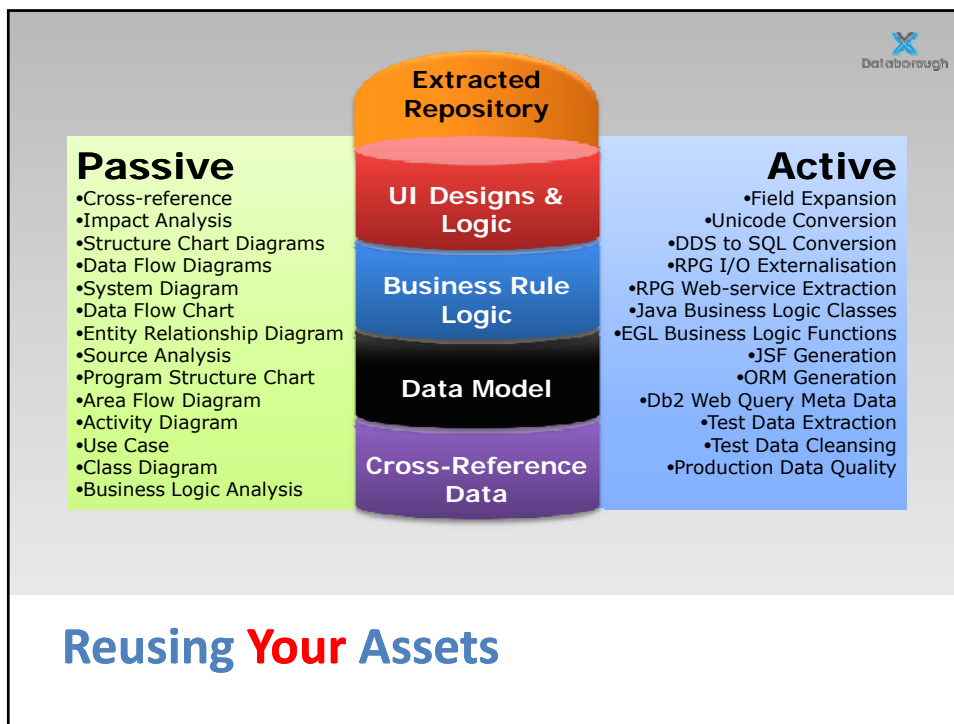
Engineered Approach



- Human Error
- Freeze Development
- Single Shot
- More Testing
- Labor Intensive

- One small change per pass
- Design Flexibility
- Repeatable
- Last Minute Changes
- Re-Usable

Automation





- 600 Plus projects
- Multi-Level Impact Analysis
- Case/Type Search Algorithms
- Case/Type Replacement Algorithms
- Exception Reporting

Case Studies – Y2K/Field Expansion/Euro/UNICODE



- Worldwide Insurance Company
- Million plus lines of code
- Integration with new package solution
- Need to use/update two sets of masters/transactions
- Automatically externalized RPG i/o into procedures/modules
- Change RLA into Callp's

Case Studies – Externalize i/o



- Large ERP system
- 55 Million Lines of code
- CA 2E & RPG
- Transforming into new application framework
- Auto Extract Business Logic and Data Model
- Generate documentation and XMI & DDL
- Import into new frameworks/tooling(Borland)

Case Studies - Design Extraction



- Large Financial Institution (Texas)
- Large database with heavy batch I/O processes
- DDS database
- Rewriting core processes
- Auto-Converted DD S to DDL and added security/constraint logic to DB

Case Studies - Database Modernization



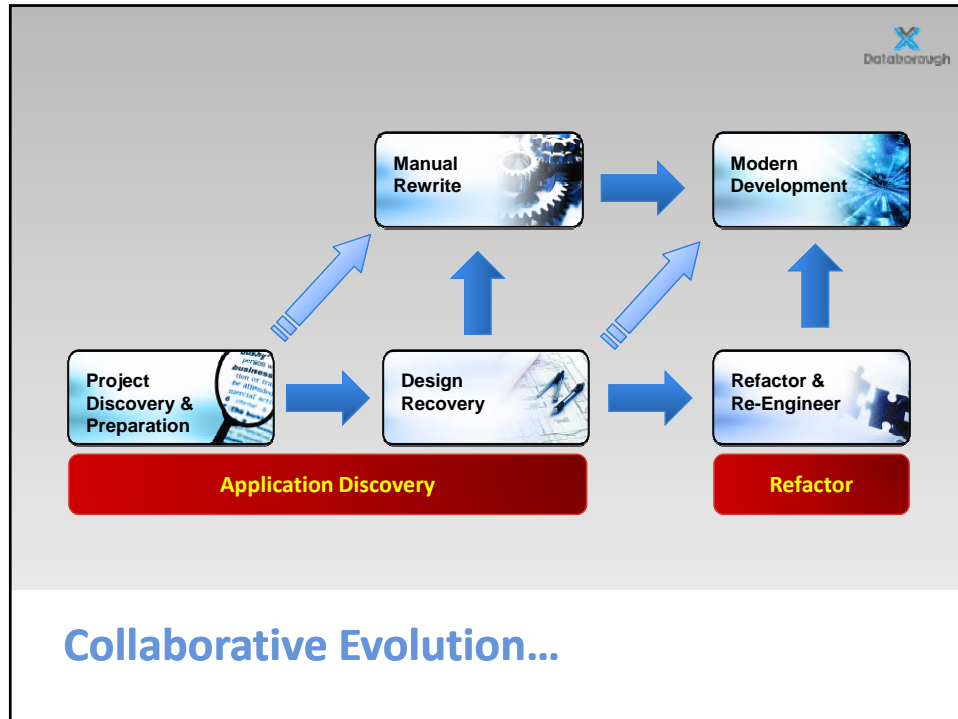
- Document
- Analyze
- Design & Plan
- Automation
- Rewrites

Summary



- **Accurately Measuring Impact of change**
accurate, current detailed, inferred and abstracted
- **Consistent & Current Information Sharing**
“even the most experienced developers need to communicate effectively”
- **Controlling Knowledge Loss**
systems become increasingly riskier and difficult to maintain
- **Education & Inclusion**
visualizing application design broadens user base(QA/Testing/non-400)
- **Design Recovery**
business rules, data models, process models, meta-data models
- **Automated System Re-engineering**
Field expansion, UNICODE, refactoring, I/O Externalization, Java, .Net

Improving Development Quality **AND** Reducing Cost



- 24 Years i series experience
- Skills in RPG/Java/COBOL/.Net/EGL/PHP
- Experts in legacy application designs
- Large Scale Reengineering
- Large Projects
- Global presence and workload distribution
 - Asia Pacific
 - Europe
 - Americas
- Certified Consultants
- System i Specialization

Databorough



- “We reduced our impact analysis from 4 weeks to 11 seconds.” Mark Rinker – Mapics
- “We saved close to a million dollars on the first projects” Chris Nickchen – New Penn
- “the key to unlocking the promise of both [web 2.0 and SOA] lies in leveraging of legacy business logic.” Ted Tritchew – Infor
- “[this] has provided a fundamental capability to increase quality and timeliness in development and support.” Mike Qusted - Fiserv
- “This helped us take complete control of our legacy applications, and has improved the quality of the development process through understanding and knowledge.” Julie Hollis – Davies Turner

Customer comments:



Questions & Answers