

Architecture-Driven Modernization (ADM) Healthcare DTF

Legacy Modernization

ADM/Healthcare
Concepts, Methodologies, Case Studies



Michael Oara

CTO

© RELATIVITY TECHNOLOGIES 2005



Legacy Modernization

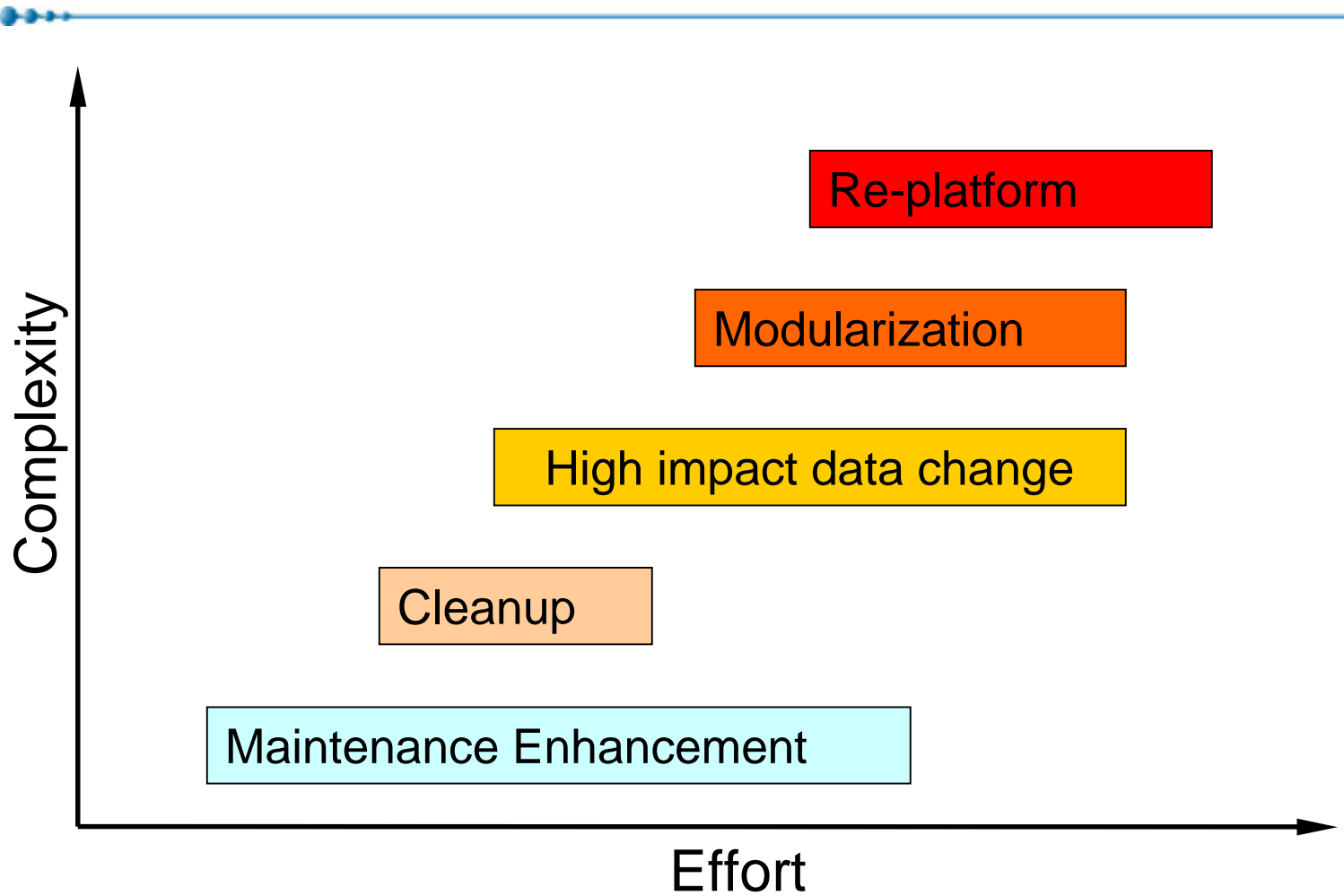
Any modification performed on a Legacy application that would bring it in line with the requirements of the business

It may take place in a number of dimensions and involve various degrees of change

Dimensions

- **Internal functionality** - Capture patient family history
- **External functionality** - Interoperability with other applications, based on a set of standards
- **User interface** - Green screen -> Browser based
- **Platform** - Hierarchical to Relational database
- **Architecture** - Flexibility, performance, robustness

Degrees of change





Knowledge Discovery and Modernization Tools

- The degree of change and effort required by demands for modernization is overwhelming
- Unlike green field development, modernization implies analysis and use of already existing assets
- Knowledge Discovery & Modernization tools help capture knowledge from existing assets and perform various degrees of change in multiple dimensions.



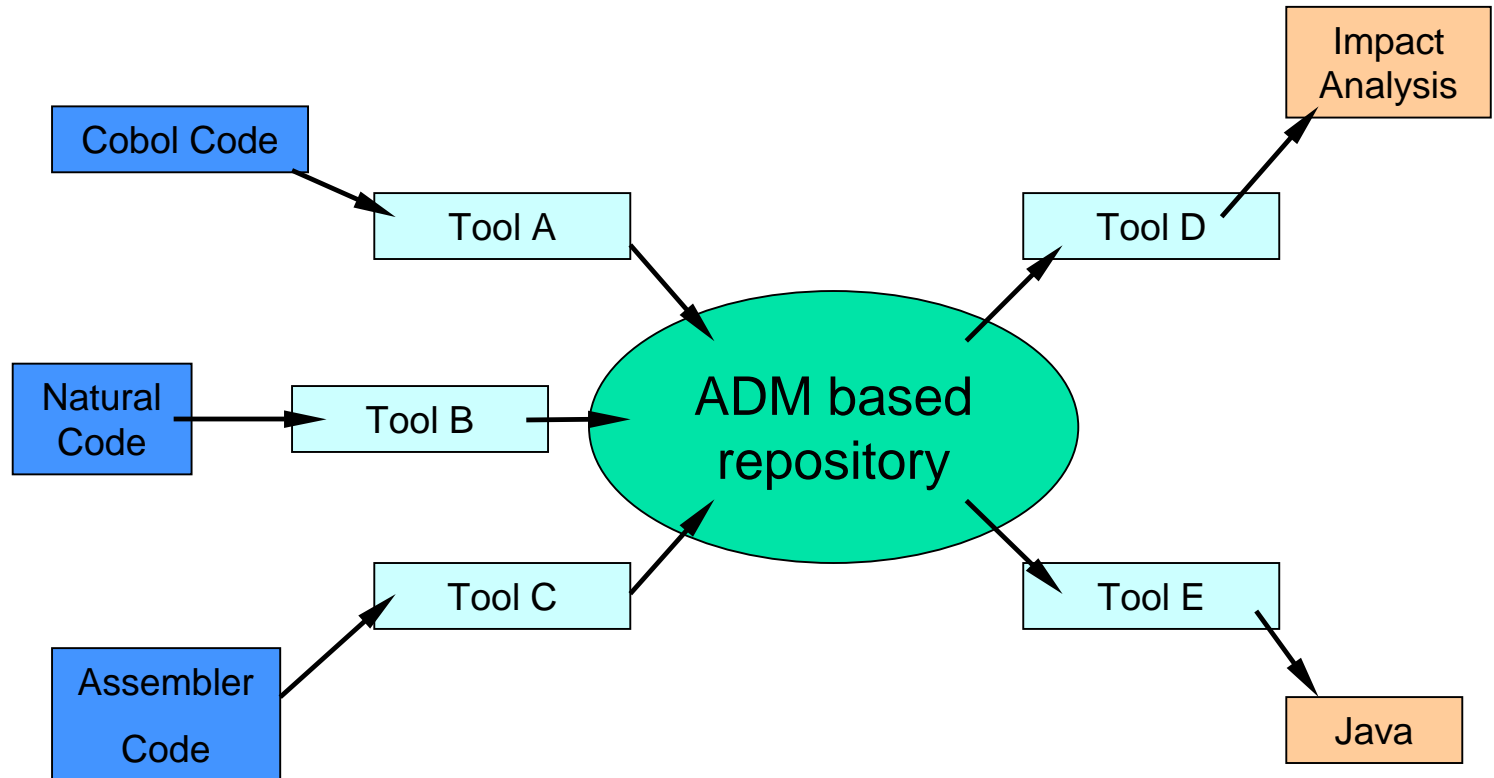
Knowledge Discovery and Modernization Tools

- Analyze legacy artifacts
- Extract knowledge and populate a repository, describing:
 - Legacy artifacts (programs, screens, files, etc.)
 - Relationships between them
 - Application logic
- Act upon the Legacy application, to either transform it or generate new code

The Need for Standards

- Just as in healthcare, one supplier or one technology cannot satisfy all requirements of the market.
- Consumers are faced with multiple suppliers which use proprietary technologies, which do not communicate with one another
- Standardization will:
 - Allow the consumer to pick best-of-breed technologies specialized in particular tasks
 - Help suppliers focus on their area of expertise

Standards based Legacy Analysis and Modernization



Cleanup Projects (1)

- Major preparation step for any other modernization projects
 - Eliminates artifacts which are not really needed but continue to be maintained and executed
 - Source of “dead wood”
 - Change in requirements
 - Programming mistakes
 - Customization
 - Obsolesce
 - Will result in major savings in any other subsequent projects involving the application.
 - Are susceptible to a high degree of automation
-

Cleanup projects (2)

- **Program Cleanup** - Eliminate the unused legacy artifacts
 - Programs
 - JCL, procs, steps, sort cards
 - Data
 - **Dead Code** - Code which will never execute for logical reasons
 - **Obsolete Code** - Code which will not execute because the business case disappeared
 - **Dead-End Code** - Code which will execute but will have no effect on the application outputs
-

Data Impact projects

- Some data changes:
 - Is eliminated
 - Has a new format
- Requires detailed impact analysis
- Example: Y2K
- Case study:
 - A table used as an input was eliminated from the application
 - Impact tree: 240 levels deep, 12,000 nodes

Data Standardization

- Definition, facts and rules about data must be standardized
- Example: HIPA
- If format changes, a deep Impact Analysis is required
- Validation rules must be catalogued and made to conform to standards
 - **Derivations** - “Problem is called recurrent if it happened n times in a year”
 - **Triggers** - “If parents had this health problem, ask for specific test”
 - **Validations** - “Patient date of birth is required”

External Interface: SOA (1)

- SOA is emerging as a standard architecture which allows the linking of heterogeneous applications
- Many of the web services are exposing functionality which can be implemented in a variety of environments
- The implementation of such functionality already exists in legacy application, it's just a matter of exposing it in the form of Web Services

External Interface: SOA (2)

- Activities on the legacy side
- Discover best components suitable for wrapping
- Discover architectural traps
 - Hidden screen conversations
 - Very large communication areas
 - Communication through queues rather than commarea
 - Validations absent from data facing programs
 - Mixed screen/data programs
- Perform remedy
- Create full service specification



Batch to Online: Case Study

The problem

Current Operation

- Claims arrived in paper format
- Operators entered the claims in the system and they were stored in a large batch transaction file
- The transaction file was processed overnight. Claims were accepted or rejected and appropriate papers were generated to be sent to the Claimant
- 3-5 days between the submission of the claim and the notification of approval or rejection

Intended Operation

- Claims are entered directly into the system, not through a transaction file
- Validation of transactions is done in real time
- The time between submission and notification is collapsed to minutes or seconds

The modernization is driven by clear needs for competitive advantage

Batch to Online: Case Study

The solution (1)

- Code cleanup
- Inventory and estimation
 - What are the boundaries of the application?
 - Which programs participate in the affected features
 - What modernization activities will be performed against what programs, in which phases
 - Estimation based on inventory and project activities

Activity	Effort per unit	Units	Total effort
Standardize names	1 person-day per program	20 programs	20 person-days

Batch to Online: Case Study

The solution (2)

Large batch transaction program is split in order to separate affected transactions

Original program

```
IF TRAN-TYPE = "T1" THEN
    PERFORM PARAGRAPH1
ELSE
    PERFORM PARAGRAPH2
```


Program1

```
PERFORM PARAGRAPH1
```

Program2

```
PERFORM PARAGRAPH2
```

*A special technique, called “**domain slicing**” is being used*



Batch to Online: Case Study

The solution (3)

Large batch transaction
program processes
transactions T1, T2... T51

Domain slicing

Intermediate
program processes
transaction T1

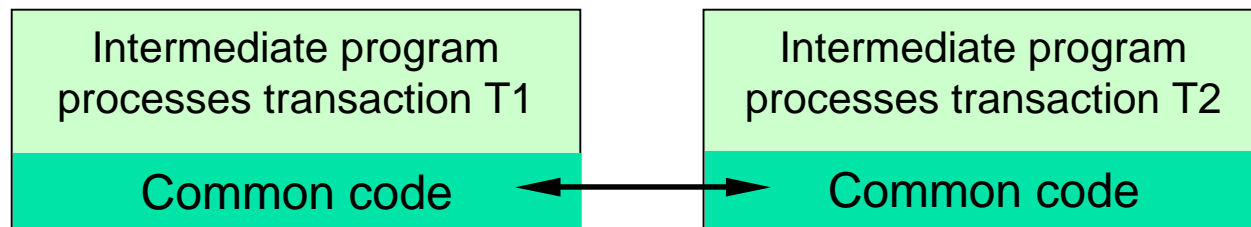
New batch transaction
program processes
transactions T3, T4... T51

Intermediate
program processes
transaction T2

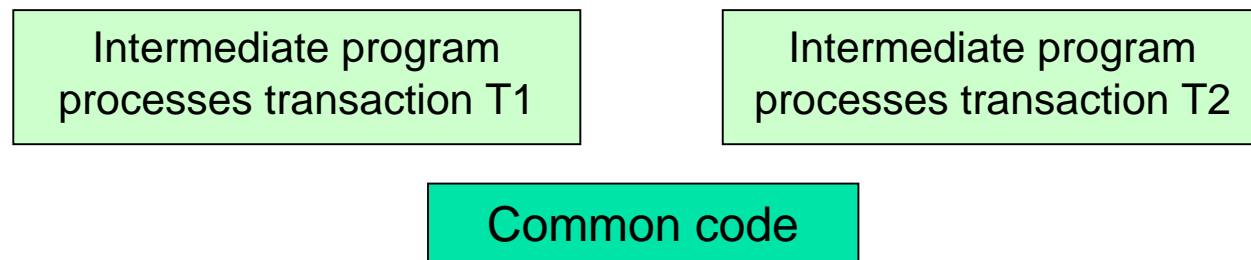
Batch to Online: Case Study

The solution (4)

Find and separate common routines in programs specialized in Transaction T1 and T2



Use “duplicate code identification” and “structure slicing”



Batch to Online: Case Study

The solution (5)

