

Software-Modernisierung - Der Abendländische Weg

- ◆ **Interactive Objects**
- ◆ **SOA and BPM**
- ◆ **Modernization Options**
- ◆ **Selecting a Strategy**
- ◆ **Legacy Transformation Factory**

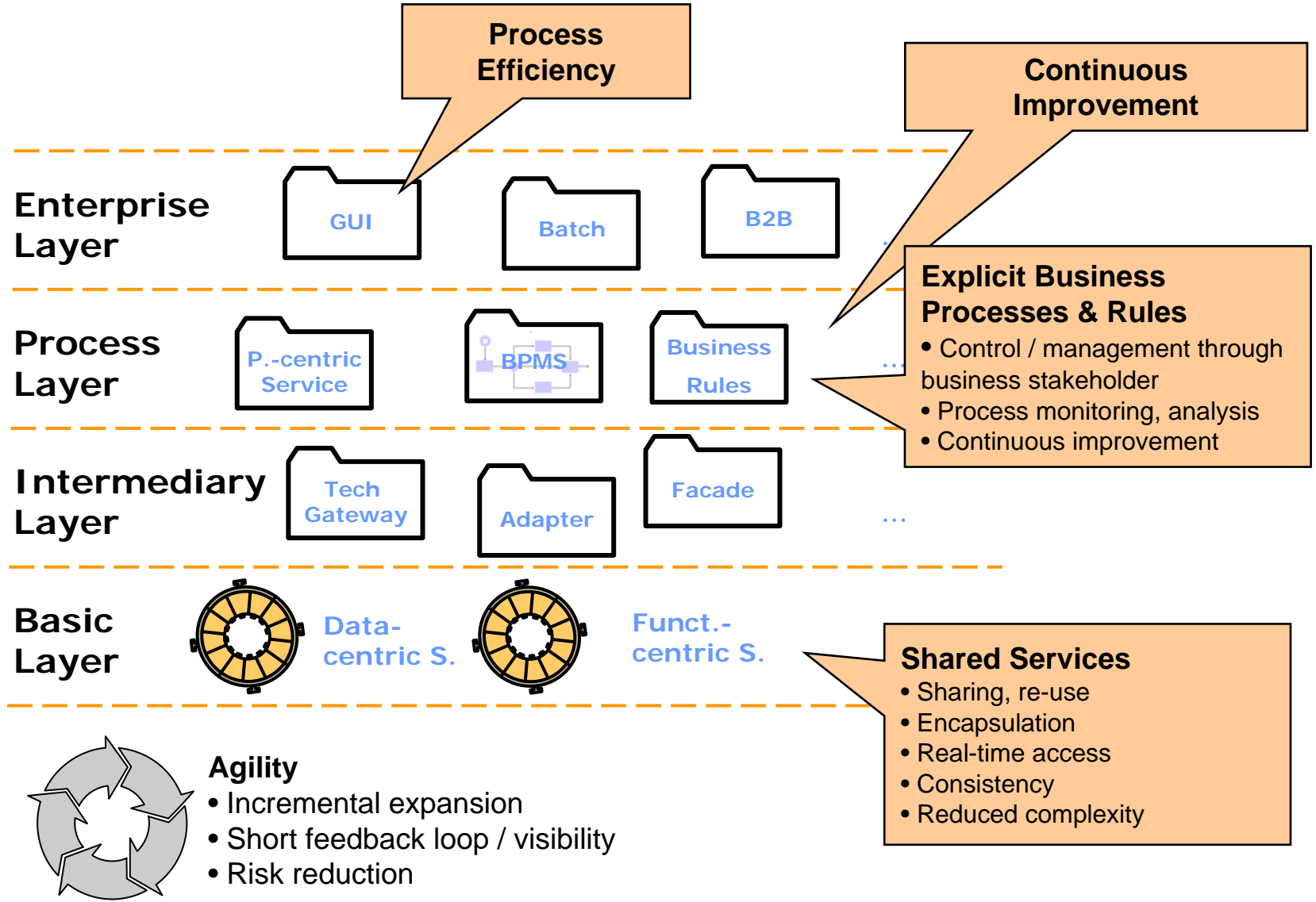


Over 16 Years of Success and Experience



- 2006 Credit Suisse officially nominated Interactive Objects as Preferred Supplier
- 2005 ArcStyler 5 and PS generate 10 success stories with blue chip companies
- 2005 Repositioning of Interactive Objects
- 2004 New Management Team on board
- 2002 Two OMG Awards for best MDA applications
- 2001 Received venture capital
- 1999 Development of the first MDA product on the Market
- 1994 OMG Design Award for first CORBA application
- 1990 Interactive Objects was founded

Benefits of SOA



SOA & BPM complement one another

SOA

How IT is structured ...

**Enable and fertilize
each other**



- ◆ Building blocks
- ◆ Business-oriented components
- ◆ Loosely coupled
- ◆ Foundation for the management of application landscapes
- ◆ Made for reuse

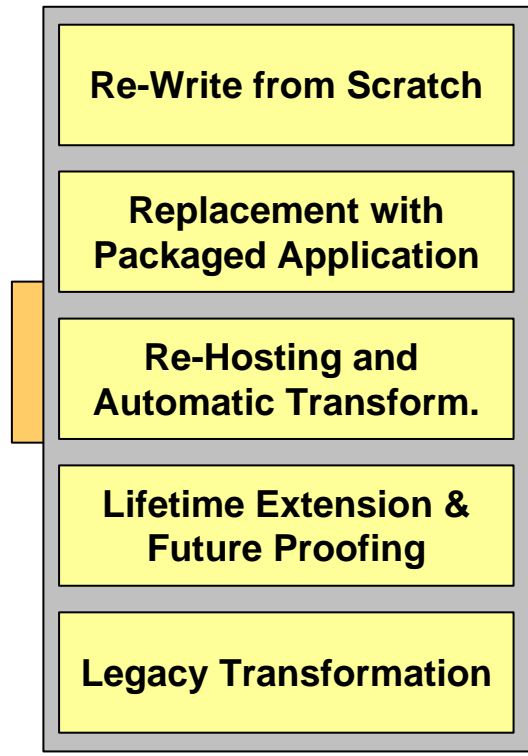
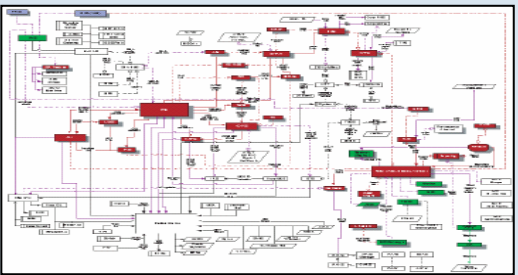
BPM

What IT does ...

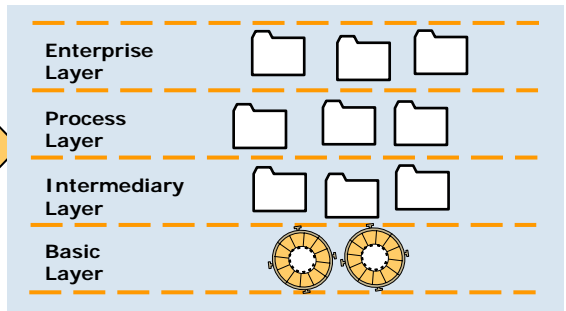
- ◆ Model of business processes and their IT support
- ◆ High-level view of business demands
- ◆ Implicit construction plan for applications
- ◆ Made for specific business purpose

What are the options?

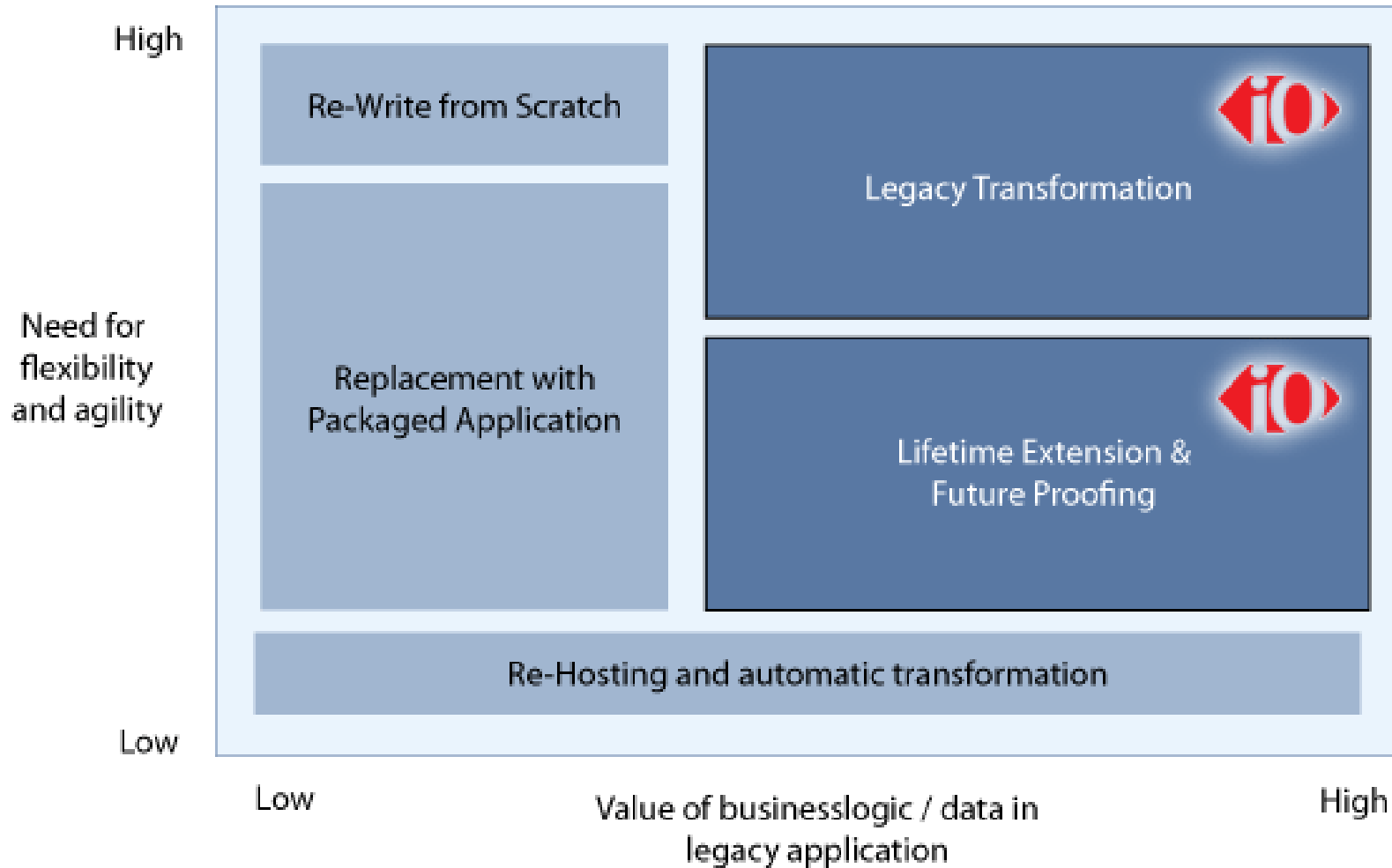
Legacy System



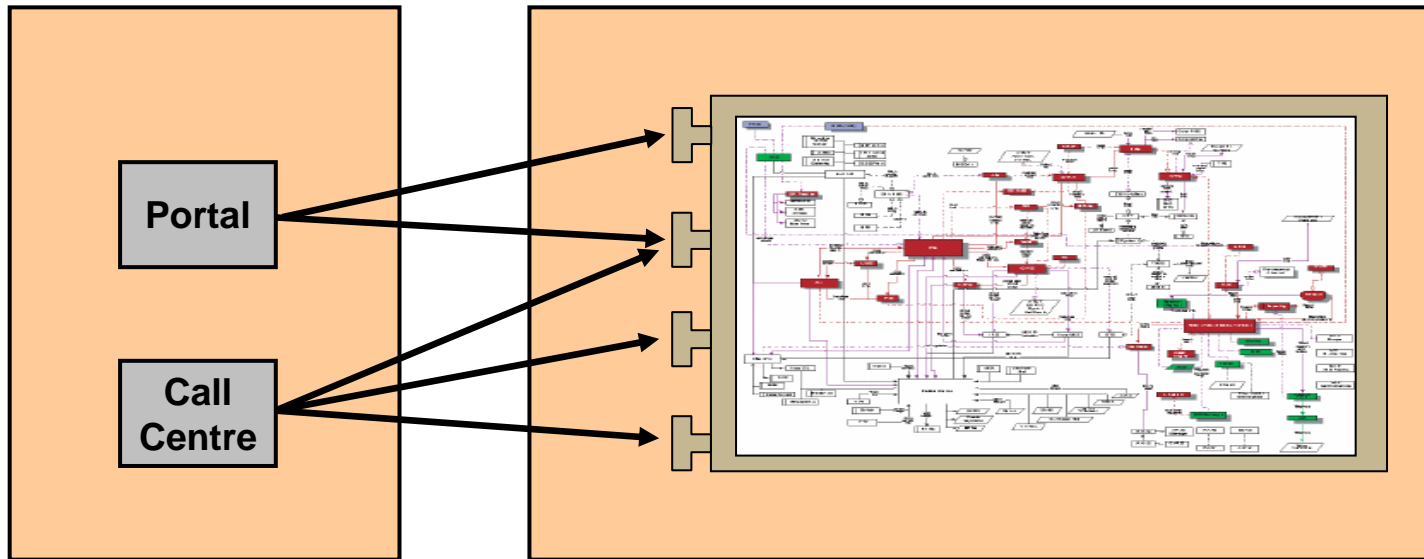
Service Oriented Architecture



Selecting a Strategy



SOA interfacing is important, but it does not address the real problems of monolithic silos



Benefits

- Enables innovative new front-ends
- Initial quick wins

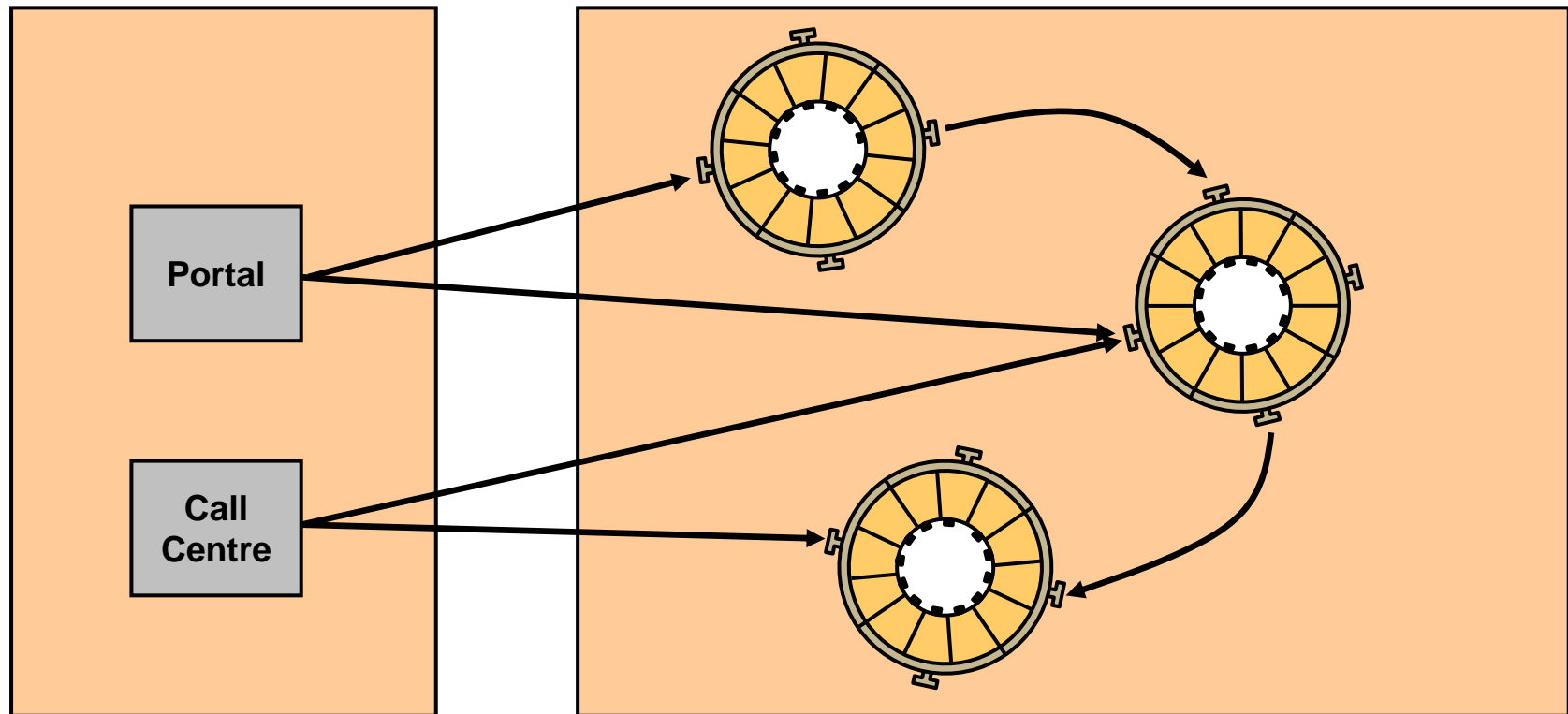
Disadvantages

- Uncontrollable „spaghetti“ behind the interfaces
- Trapped into „big bang“ rollouts of new releases...
- ...usually leading to high number of emergency requests
- Making changes in small steps would create fewer problems, but not possible because of tight coupling
- No way to develop and test components independently

Consequences

- Time to market very long
- Backend functionality pushed to front-end, causing inconsistencies
- High maintenance costs

Loosely coupled SOA Components



SOA Component

- Full encapsulation: access only via public interfaces
- Autonomous lifecycle: deployment independent of other components

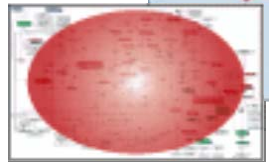
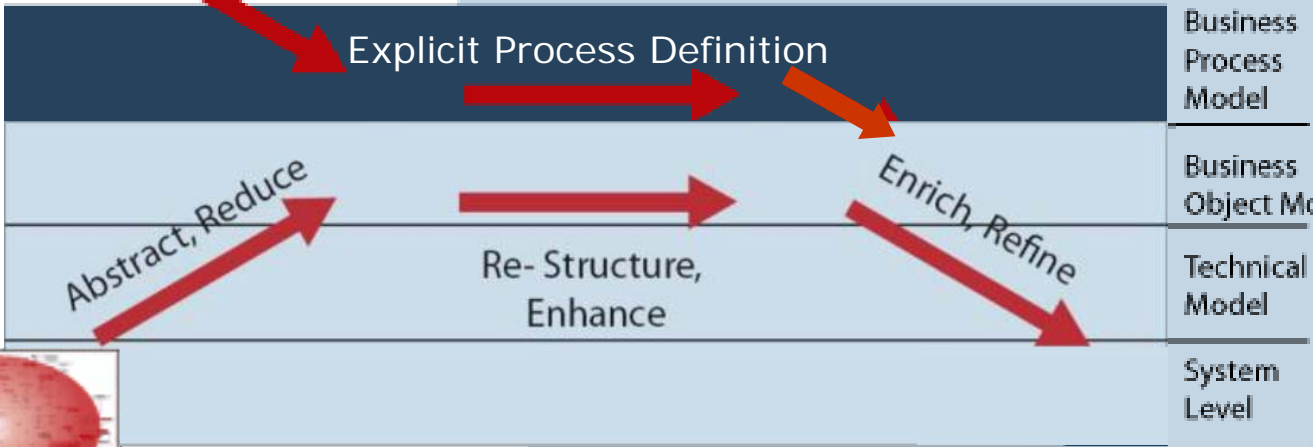
Example for relatively large bank

- 30-60 Components
- 300-1200 public Service Interfaces

iO's Legacy Transformation Process



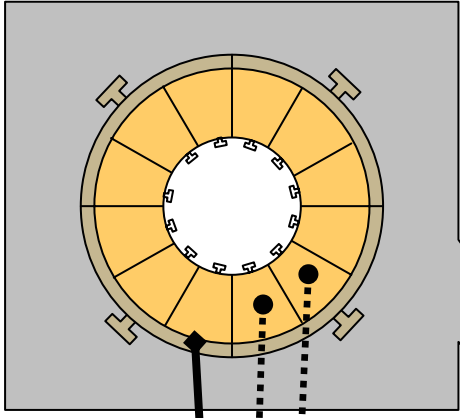
Sourcing :
Model Driven Offshoring



Legacy System

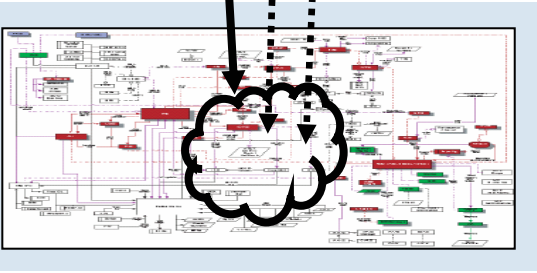
New System

SOA Componentization Process

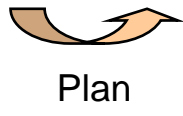
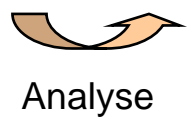
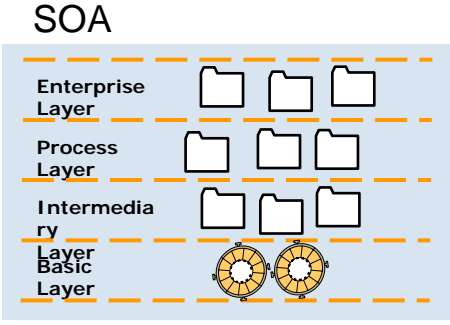
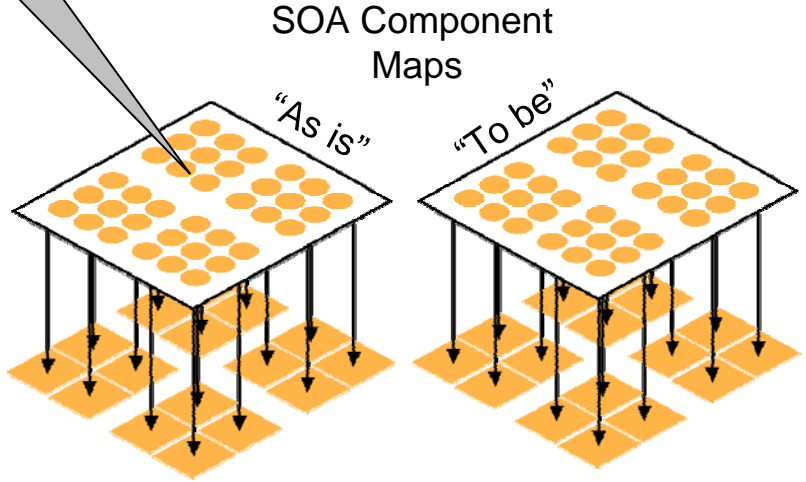


Process Overview

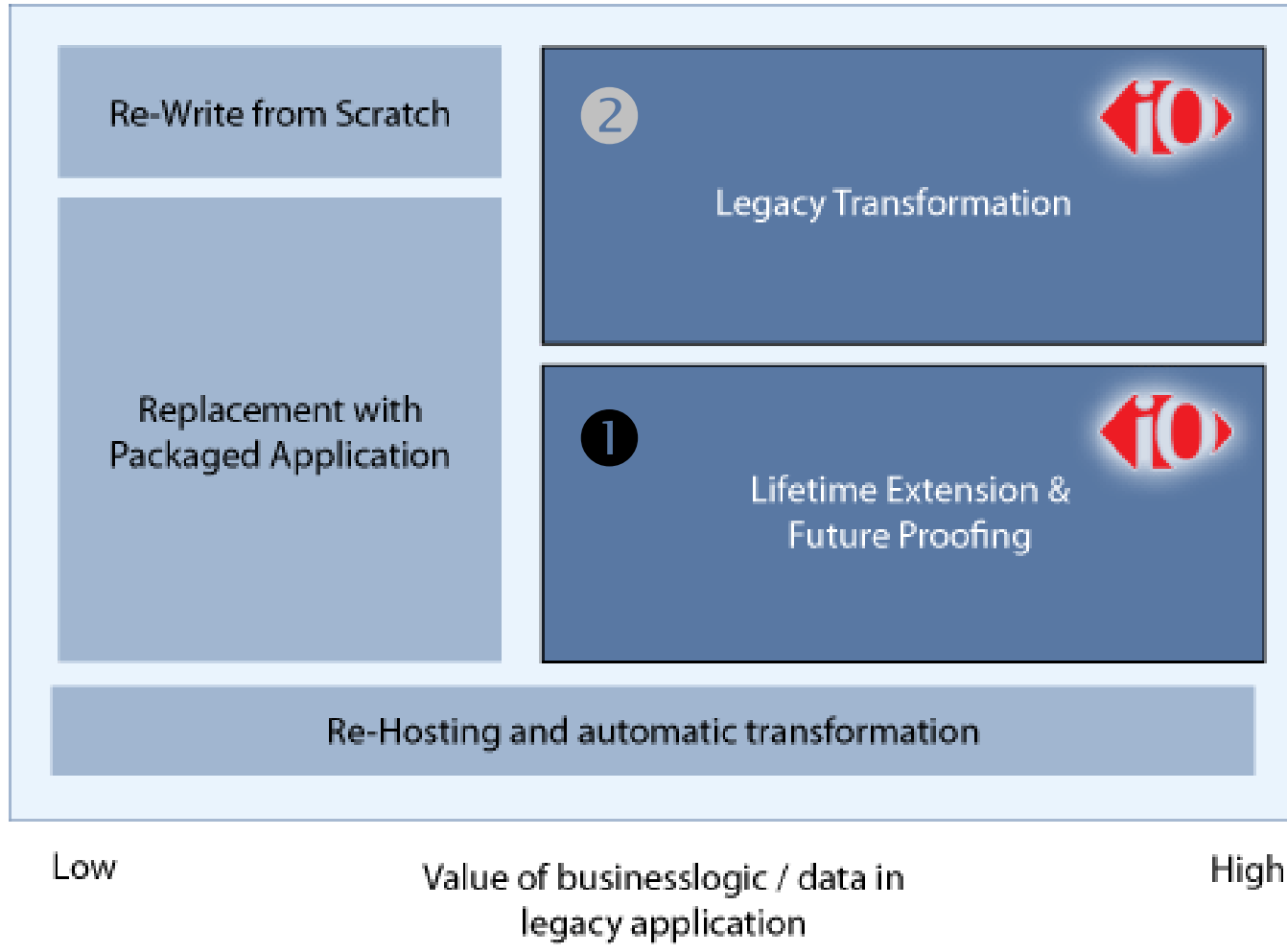
1. „As is“ SOA Component Map
 - Analyse Legacy System
 - Create high-level map
 - Enrich (detailed Asset Mapping)
2. „To be“ planning
3. Componentization, Disentanglement



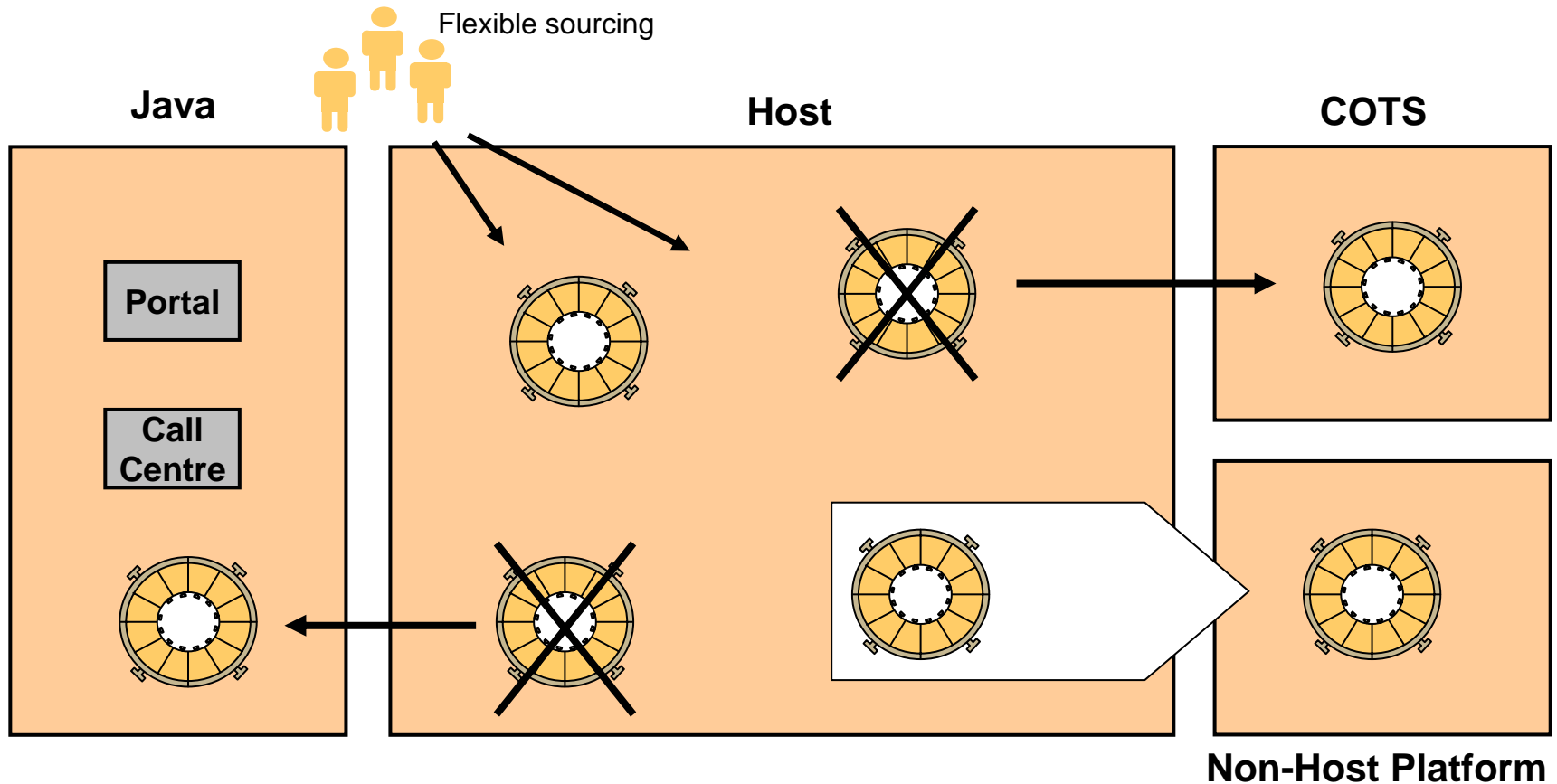
„Spaghetti“ System



Legacy Modernization



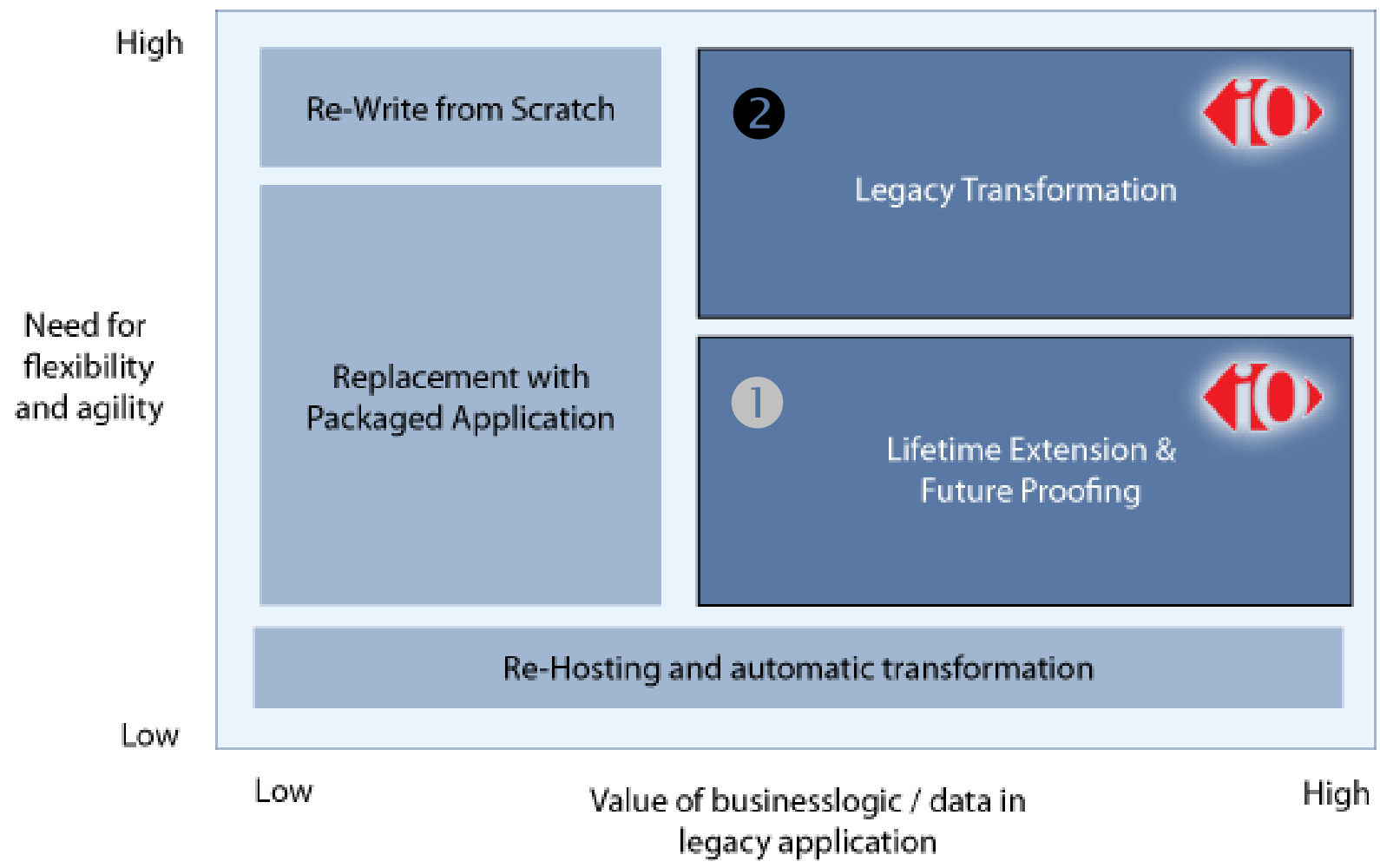
Disentanglement results in flexibility



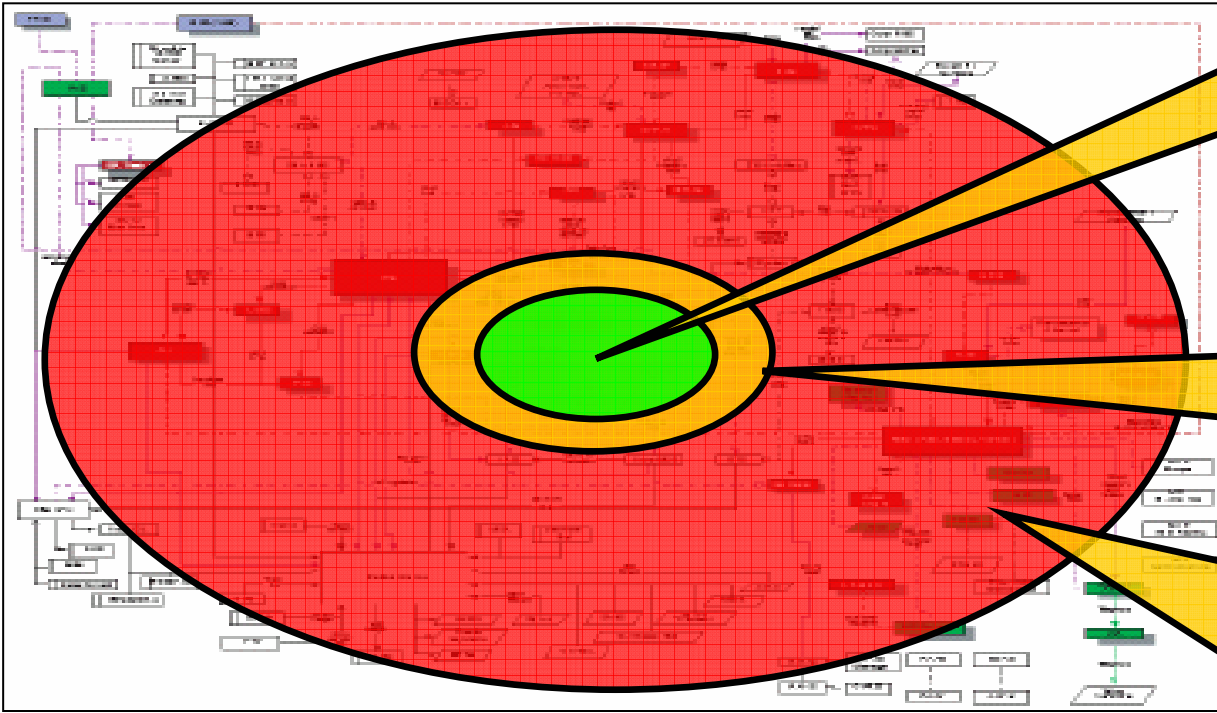
Disentanglement creates options

- Migrate frequently changing components to more agile platform, e.g. Java
- Replace selected components with Common-Off-the-Shelf Software (COTS)
- Re-host components on non-Mainframe environment, e.g. Microfocus on Linux

Legacy Modernization



Removing dead freight is key!



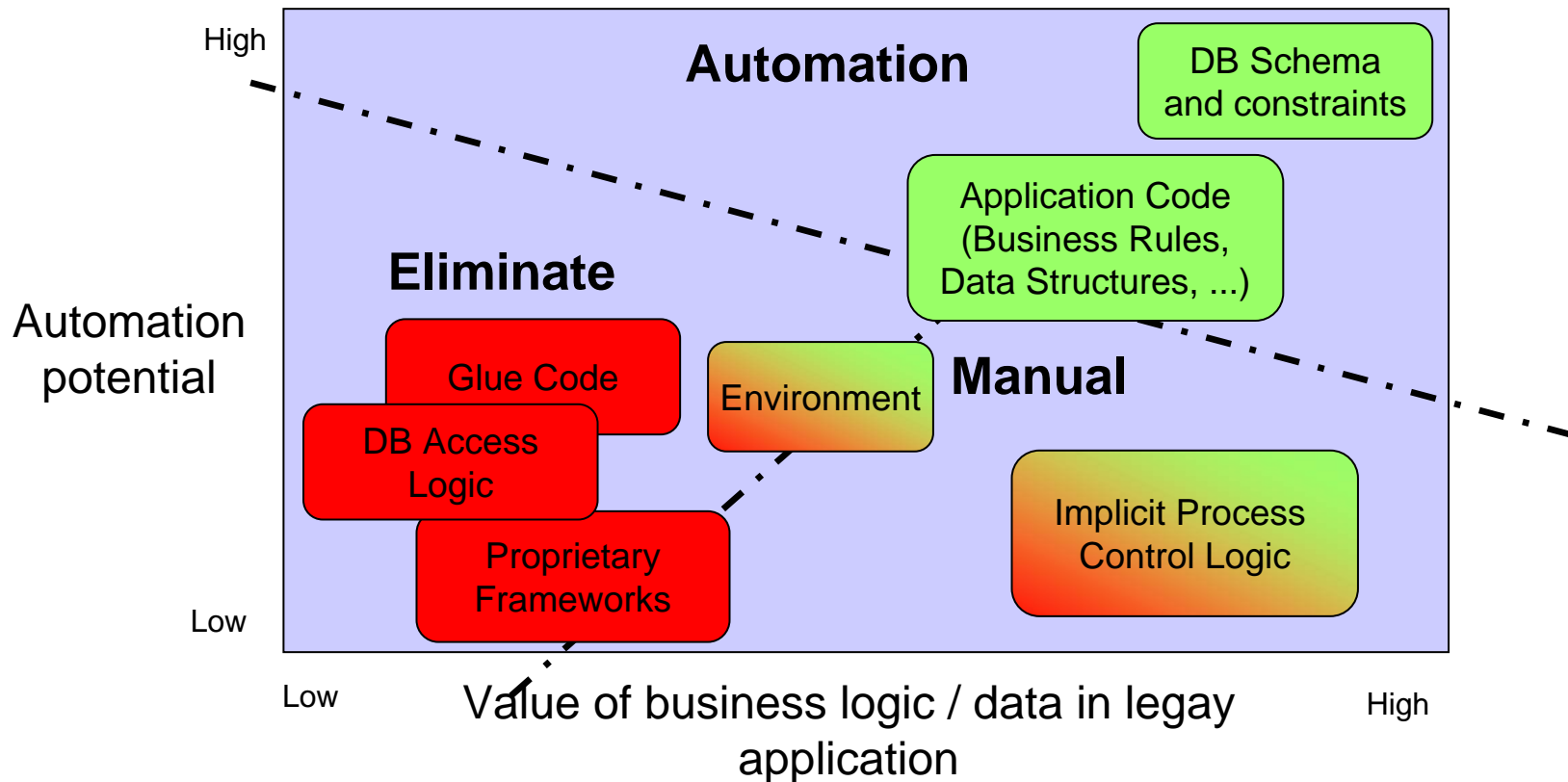
10-15%: Core Assets
E.g. structure of business data, relationships, business rules, constraints, screen formats, etc

5-10%: Hidden Assets
Scope (!),
Implicit Process Control,
Etc.

~80%: Dead Freight
Glue Code, data transformations, data redundancy management and exception handling caused through architectural “rank growth”

Focus on transforming key assets only!

- ◆ Only transform the 20% of the legacy system which are really valuable
- ◆ Use a combination of automation and manual transformation
- ◆ Leverage high automation potential of MDA

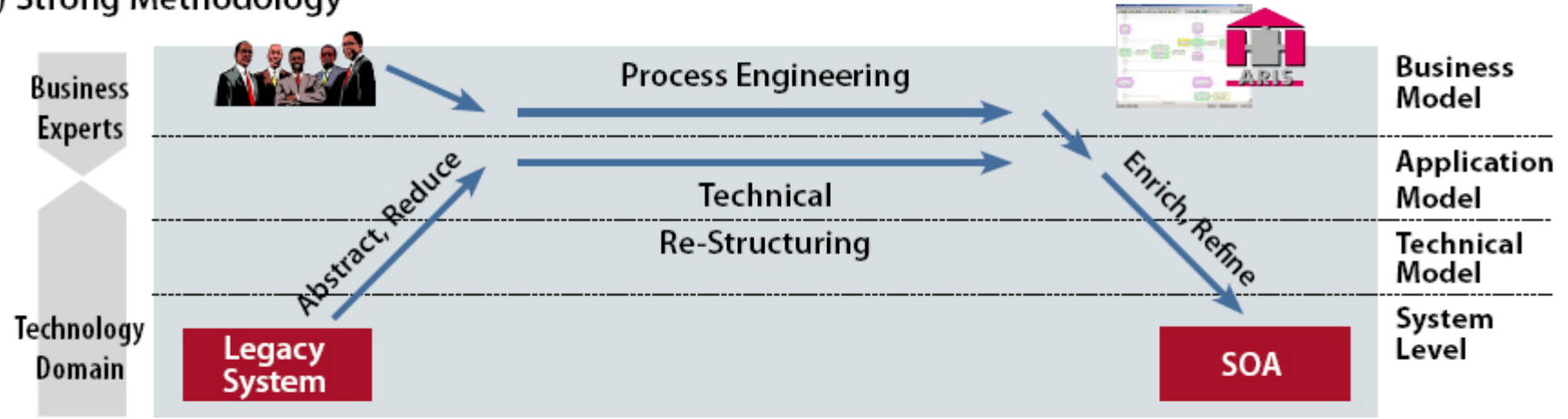


Processes – Methodology - Tools

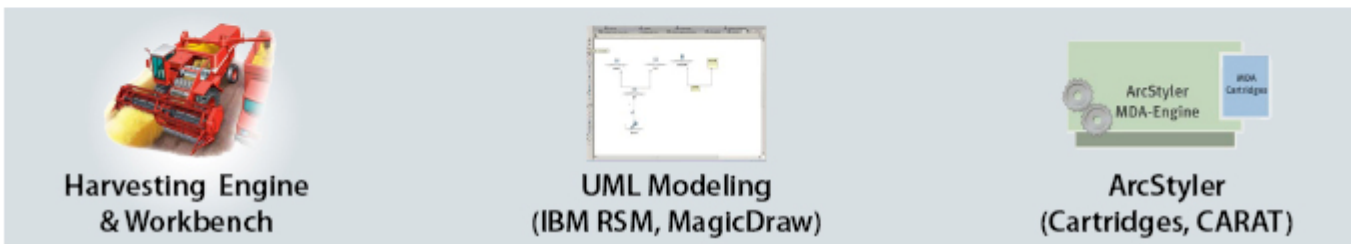
1) Industrial Transformation Process



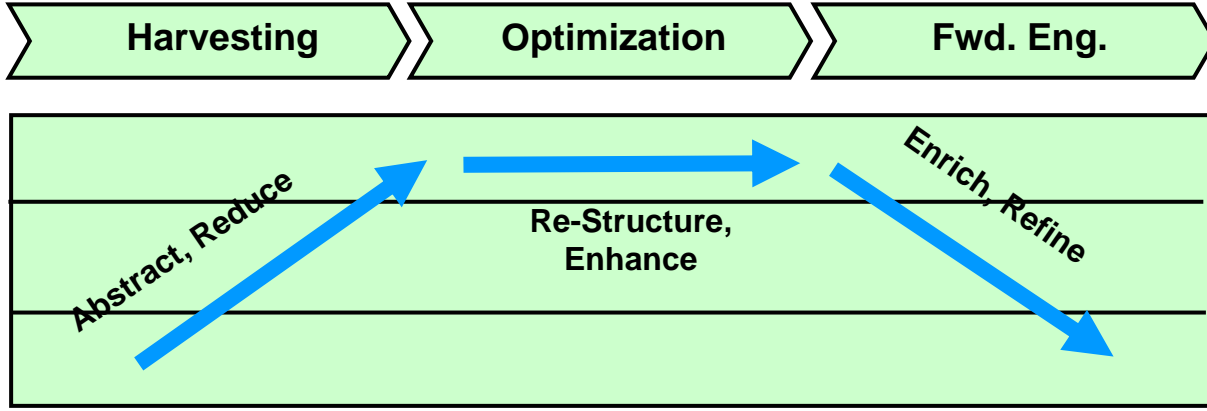
2) Strong Methodology



3) Flexible Tools



Automation vs. Manual Transformation



Automation Potential

60 - 80%

0 - 20%

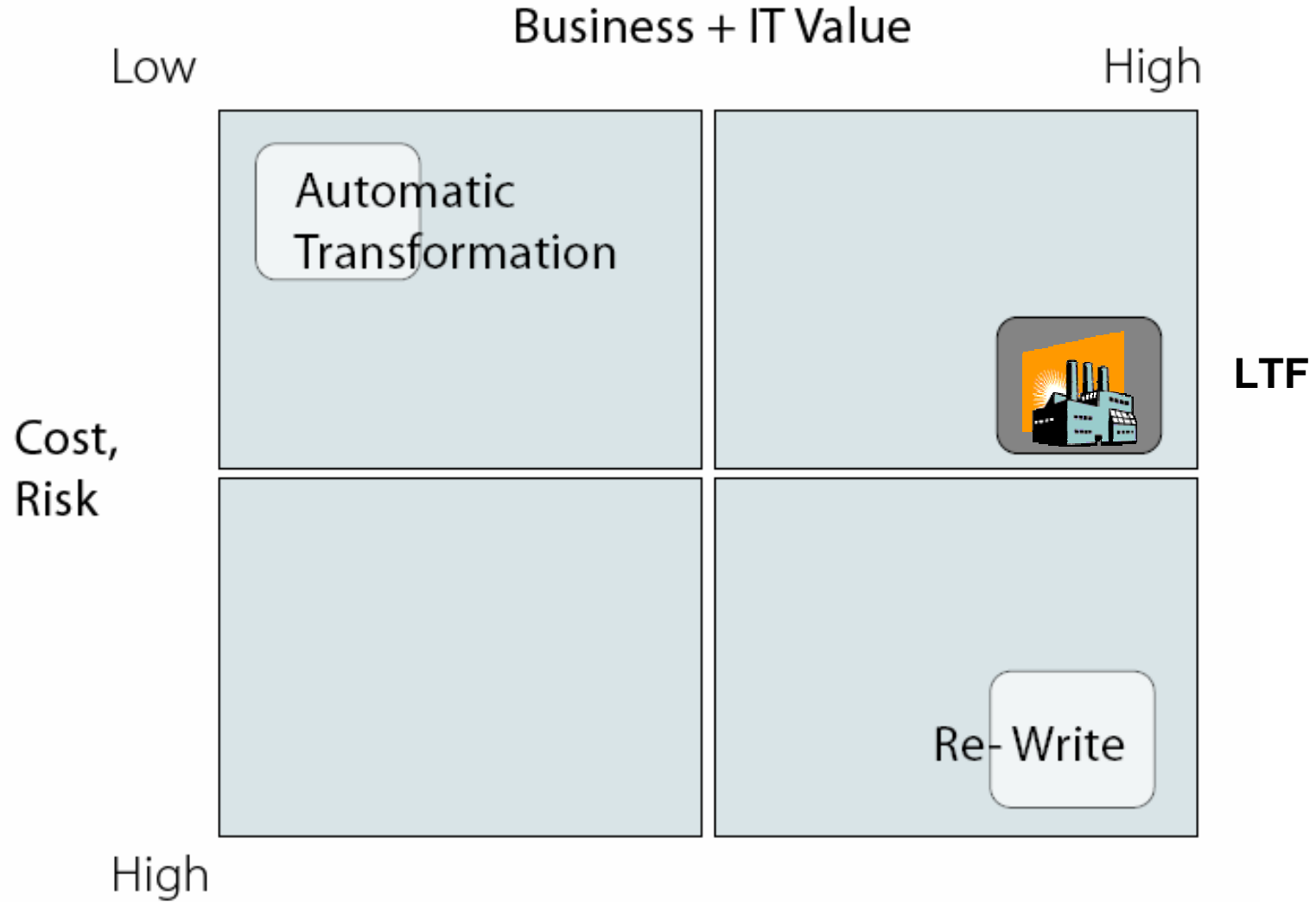
50 - 70%

Recall: only ~20% of the legacy code based will be harvested, of which then 60 – 80% can be harvested automatically.

This is the creative part, which usually doesn't lend itself to automation. This is why it's important, that this is done on a **high level of abstraction**, and after the reduction process!

Some parts of the new system can be automatically generated, e.g. Data Centric Services

Comparison



Benefits for the Concerned Roles

◆ Business Analysts

- Methodological Support for high quality Business Process Architecture

◆ Architects

- Creation of integral Process Oriented Architectures (POA) consisting of Business Process Architectures and technical Service Oriented Architectures (SOA)

◆ Designers

- Direct Usage of Business Analyst output for implementation

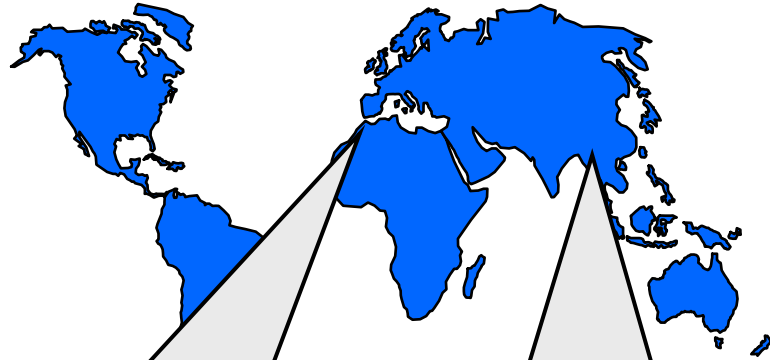
◆ All

- Navigability from Business Model to Code and vice versa during modernization process

◆ Managers

- Process Control and reports

Case Study: Bank Coop Credit Management System



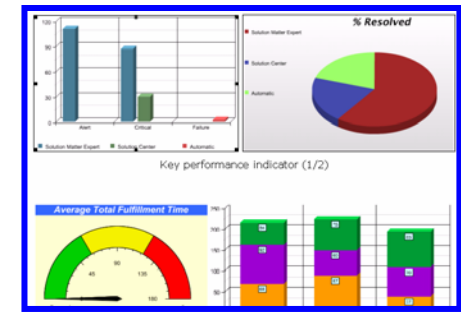
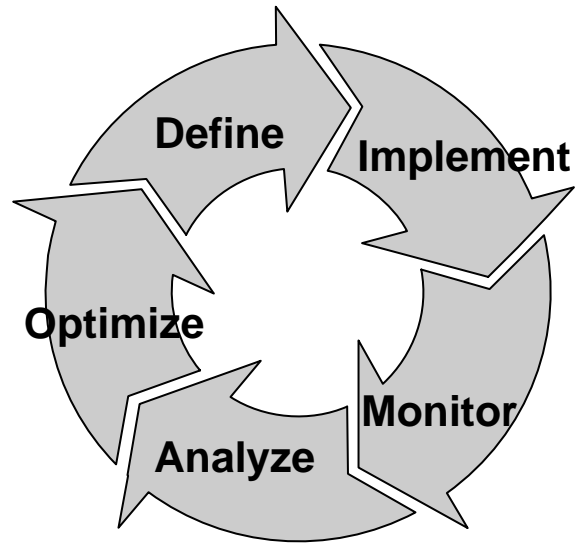
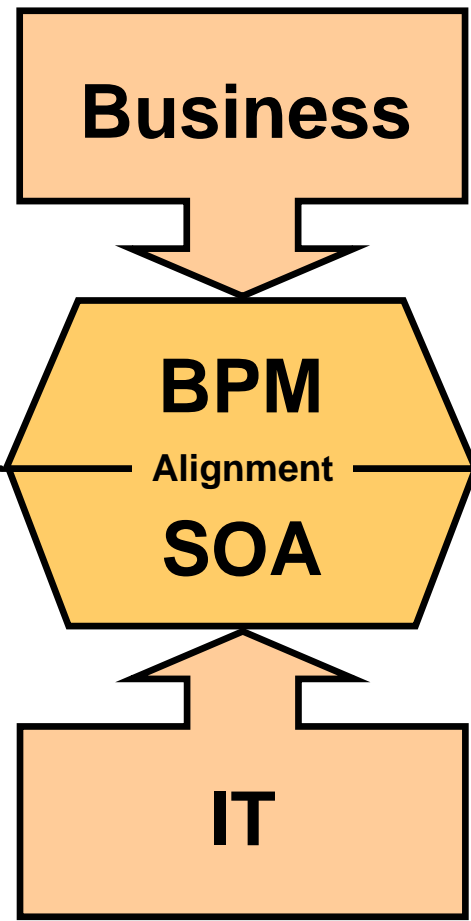
CSC Switzerland:
1 Project Leader
1 Technical & Test Manager
2 Business Analysts (UML)
1 Technical Architect

Interactive Objects:
1 MDA Expert (20%)

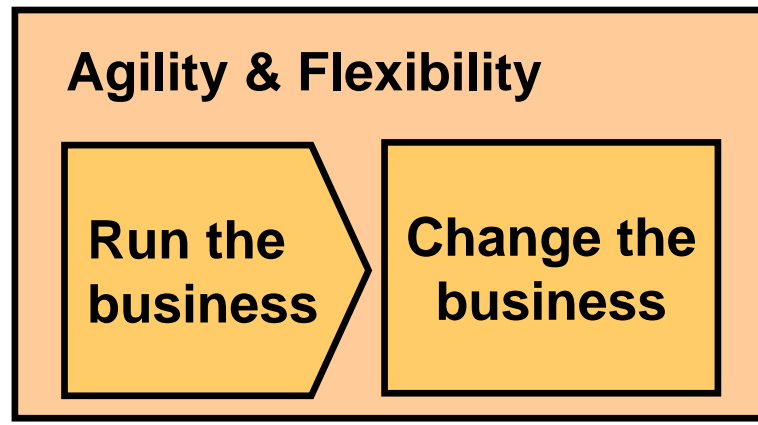
CSC India:
1 Local Project Leader
13 Java Developers

M Buser, Director Application Engineering, CSC Switzerland: The LTF approach in combination with Mix-Shoring resulted in 65% cost savings compared to a manual migration approach.

Summary Legacy-to-SOA



- ⇒ Explicit processes
- ⇒ Transparency
- ⇒ Continuous improvement





**Interactive Objects Software GmbH
Basler Strasse 61
79100 Freiburg, Germany**

**Tel. [+49] 761 / 4 00 73 - 0
Fax [+49] 761 / 4 00 73 - 73**

www.interactive-objects.com

info@interactive-objects.com