

Enhance mainframe developer productivity for SOA applications with Rational Business Developer

May 2008

Stefano Sergi
Product Line Manager
Rational Business Developer
sergi@us.ibm.com

Rational. software

Agenda

- **Business drivers to IT transformation**
- **Enterprise Modernization**
- **Application development challenges**
- **Rational Business developer overview**
- **Case studies**
- **Summary**

What 765 CEOs from 20 industries and 11 geographies told us in IBM's Global CEO Study 2006^a

- **Renewed focus on top-line growth**
 - Cost-cutting and efficiency are still important but are no longer main drivers
 - Innovation drives profitable growth
- **Numerous external forces are bearing down on the organization**
 - Radical business change is inevitable
 - Seize opportunities for differentiation
 - Innovate in new ways to succeed
 - Need to expand the innovation mix



CEOs are embracing change

“CEOs do not seem intimidated or content simply to cope... [Instead they see change] as both reason and license to expand their innovation horizon...”

^aIBM Global CEO Study 2006

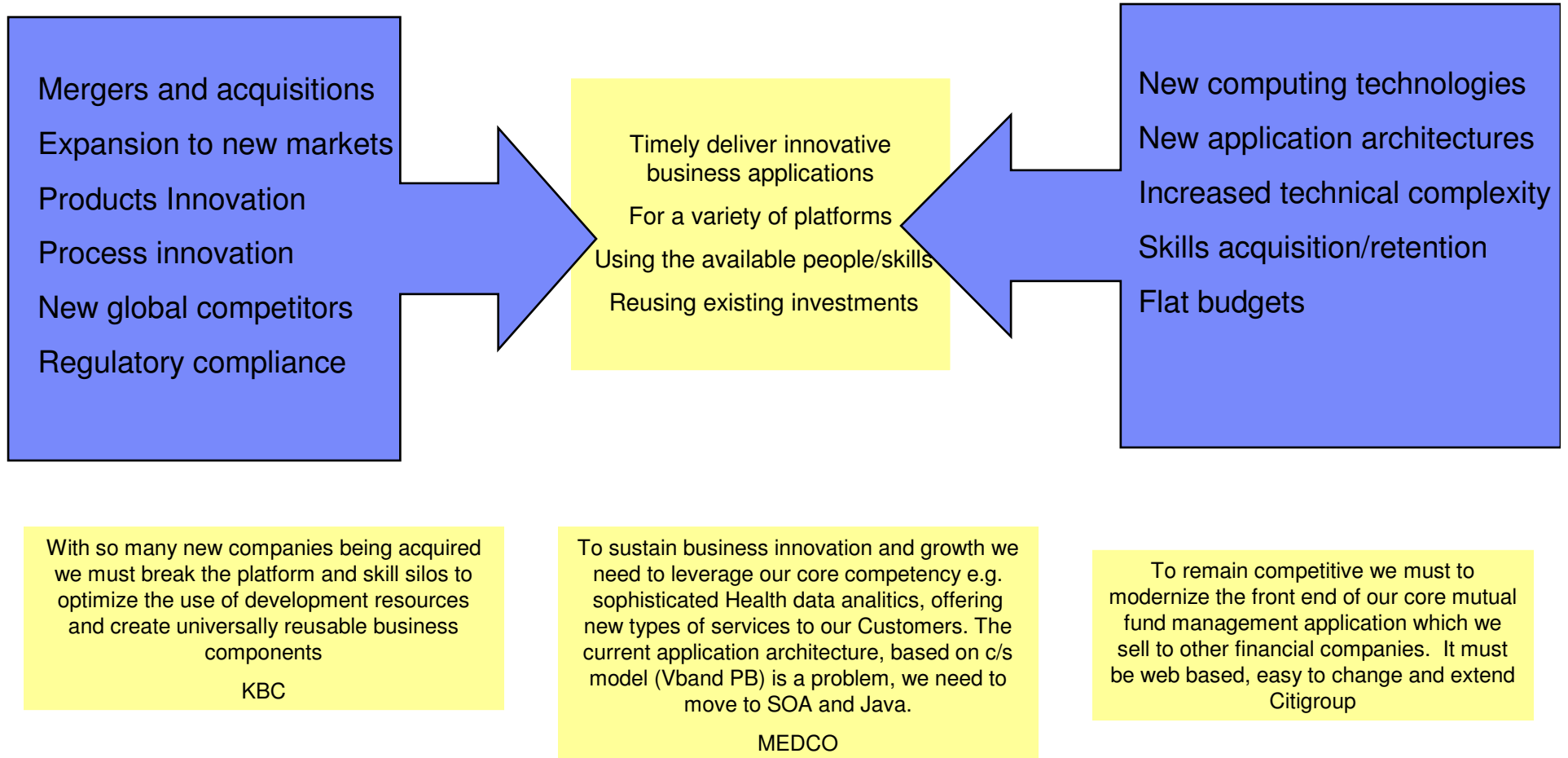
IT as a Core Business Process

“IT investments made 10 to 20 years ago are ... constraining the speed with which these businesses can change and respond to industry threats.” – Gartner’s Top Predictions for Industry Leaders, 2007 and Beyond

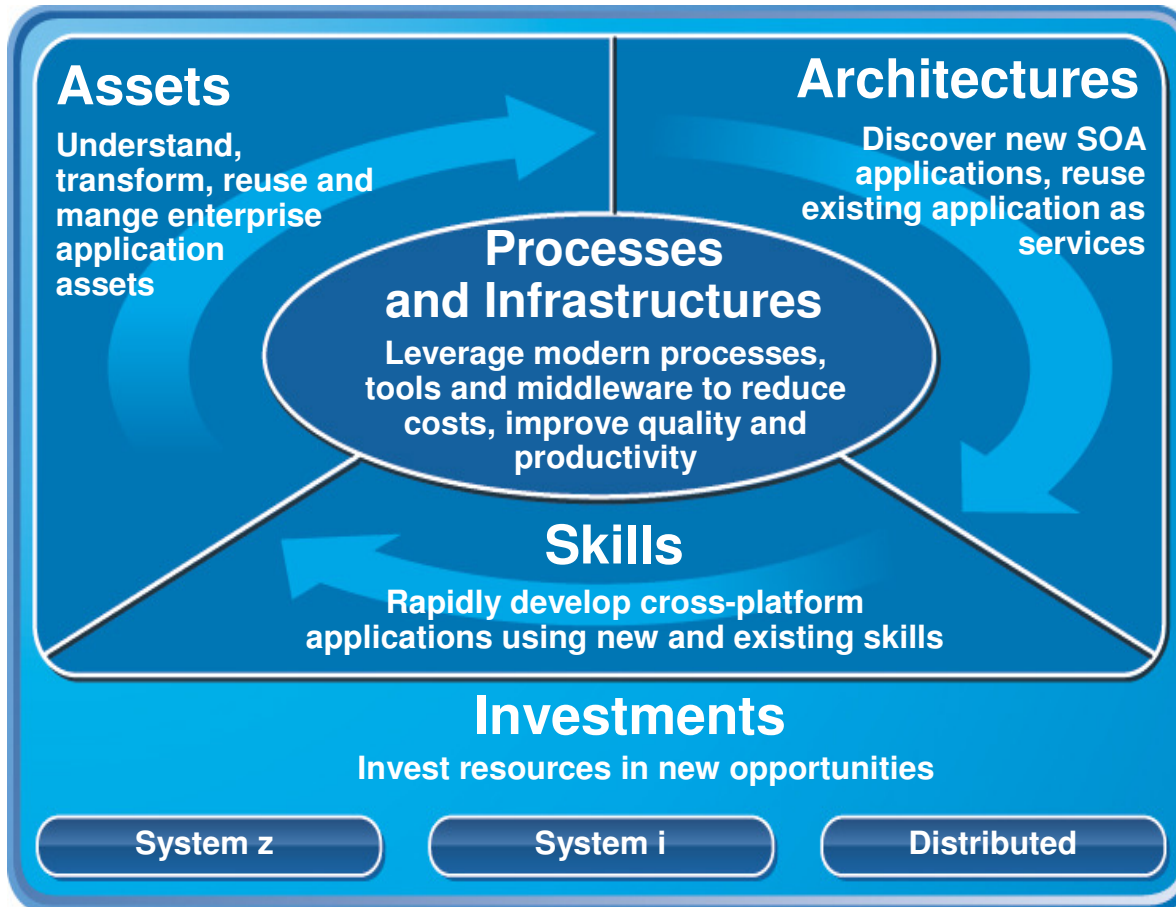
- ▶ IT flexibility is a key enabler for today’s businesses
- ▶ To be successful, you must mature and modernize your IT tools and processes
- ▶ However, doing so can be rather challenging...

“Aligning IT with the business relates directly with an IT group attaining real flexibility.” – Butler Group

IT Conundrum and the need for Modernization



Enterprise Modernization – Solution Overview



Leverage value in existing assets

Drive innovation with SOA and web technology advancements

Leverage existing and new staff on multi-platform projects

Improve quality and flexibility with consolidated team infrastructure

Reduce maintenance costs

<http://www-306.ibm.com/software/info/developer/solutions/em/>

Rational. software

Key development challenges

SAP HR system
Custom .Net applications
Back-office legacy systems
Home-grown line of business apps.
Oracle Siebel CRM



Windows
AIX
Java
5iOs
COBOL/zOS
Linux
.Net

Existing Applications

- Costly to maintain
- Monolithic
- Hard to reuse in new ways

Skills

- Skills silos
- Skills mismatch
- Erosion of legacy platforms skills

Platforms / Middleware

- Proliferation
- Coexistence
- Complexity



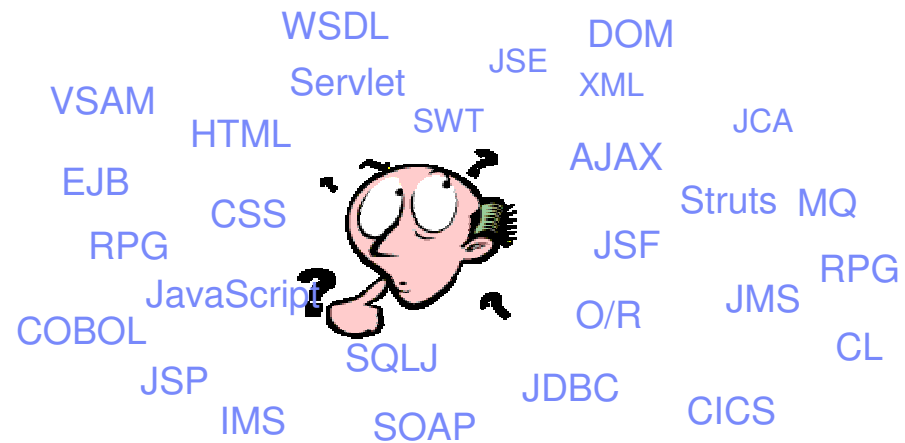
High costs

Compromise

Slow response



Building applications today is not easy



- Learn and master a myriad of middleware programming interfaces.
- Understand new programming paradigms, frameworks, libraries.
- Cross platform solutions require totally different programming skills.
- Constantly emerging new technologies, frameworks, standards, etc.

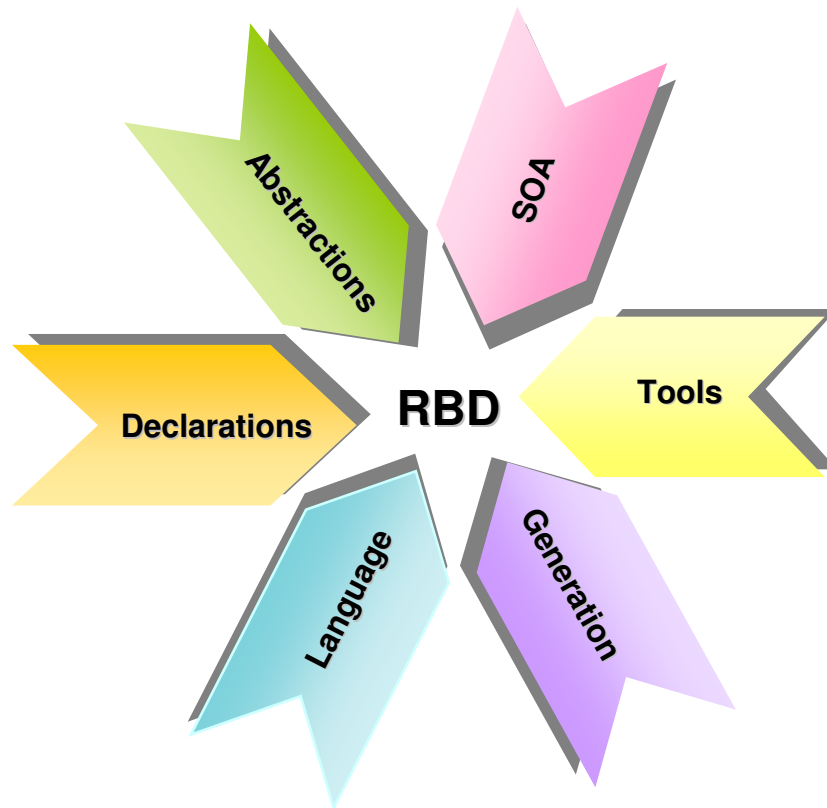
Building applications today can be easy



- provides programming “abstractions” that increase productivity and quality
- very easy to learn, and supports broad runtime choices
- modern comprehensive language (not your father 4gl!)
- shields programmers from intricacies of middleware and runtimes
- inter-operates seamlessly with legacy, so that new modern solutions can easily reuse and extend existing investments

Rational Business Developer

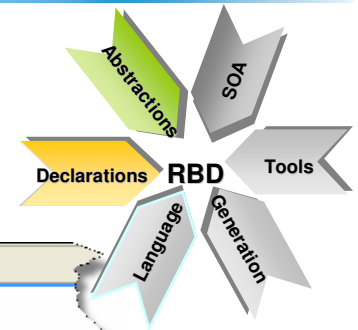
A simple, robust, unified approach to end-to-end construction of application and services that shields developers from intricacies of runtimes and middleware



Rapidly Deliver...

Modern Innovative Solutions...

With the available Skills



The power of abstractions and Declarative

■ Data access:

- "Records" provide access to:
 - SQL, Indexed, Relative, Serial, DL/I, Service data, Message Queues
- Common Verbs for data access (**Get, Add, Replace, Delete**).
- Allows complete access to SQL statement if needed.
- Common Error Handling.

```
*sampleProgram.egl x
function allLoans()
  loans LoanRec[];
  get loans;
end
```

■ Remote Invocation

- Call COBOL, RPG, C, Java.
- Linkage information separated from code.
- Data mapping, protocol invocation all resolved at runtime, NO code necessary!

```
*sampleProgram.egl x
function callHelloWorldOniSeries()
  salutation char(30);
  call helloworld salutation;
end
```

■ User Interface:

- EGL "handlers" hide complexities of
 - Web (JSF) UI
 - Reports (BIRT)
 - Portlets
 - * Rich Web (Ajax)

* Tech Preview

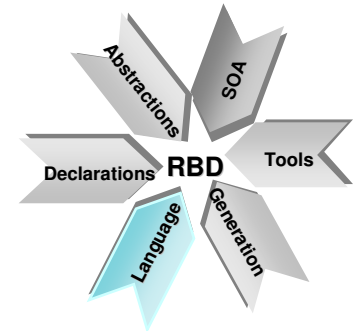
```
*sampleProgram.egl x
DataItem SSN Password char(9) {
  validatorFunction = "ValidateSSN()",
  displayUse = secret,
  pattern = "XXX-XX-XXXX",
  displayName = "Social Security No",
  inputRequired = yes;
end
```

■ Validation/Editing Rules:

- Via properties in "Data Items.
- Define formatting & validation rules in a common place.
- Reuse data items for Records, screens, web pages, reports.

The power of the Language

Simple and familiar.... yet robust and complete



- **Rich data types**

- Simple (int, string, boolean, etc.) or Complex (any, static arrays, dynamic arrays, dictionaries, array dictionaries, etc.)

- **Keywords**

- Case, if-then-else, while, for loop, for loop cycling through a database result set, etc.

- **High power language capabilities**

- Automated Casting (e.g. using AS operator)
- Mixing data types in assignments and expressions
- Exception handling

- **Rich libraries of built-in functions**

- Math, string, date/time, system, i5/OS resources (Data Areas, Data Queues) access, ...

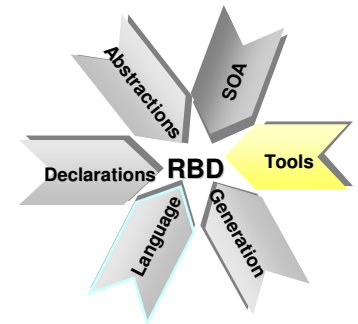
- **Robust integration with existing investment or access to low level APIs**

- Call RPG, COBOL, C, etc.
- Full Java interoperability
 - Invoke Java from EGL (map Java classes with EGL External Types)
 - Invoke EGL from Java

```

hello.egl x
1 // Hello World basic program
2
3 program hello type BasicProgram
4
5     // Data Declarations
6     name string = "World";
7
8     function main()
9         writeStdOut("Hello " +name);
10    end
11
12 end
13
    
```

The power of tools

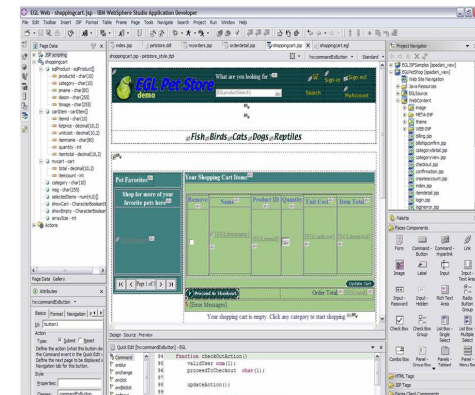


■ Web development

- Integrate seamlessly with Page Designer and JSF tools
- EGL native coding within JSF, mask all JSF complexity
- Drag drop, code behind

■ Powerful IDE

- Smart EGL editor, Code templates and snippets, Code completion
- SQL visualization, editing, validation
- References and declarations, Refactoring
- **Integrated Debugger (no compile necessary, just animate source as you define it!)**

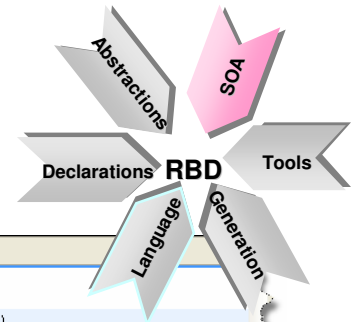


■ Model Driven Development tools

- Generate fully functional CRUD applications from UML models or Database tables
 - including default JSF, and EGL or Web Services!

The power of Services

Built into the language



Service part:

- a generatable part containing code that will be accessed:
 - from EGL code by way of a local or TCP/IP connection (*EGL Service*).
 - from any code by way of an HTTP connection (*EGL Web service*).

```
customerService.egl x
// service
Service CustomerService
Function getCustomer(custid String) returns (string)
//
end
//
end
```

Interface part:

- Used to access external services as EGL services or simply to provide separation of concern.

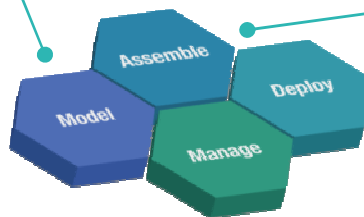
```
customerService.egl creditCheck.egl x
// interface
Interface creditCheck
function checkCredit(SSAN string in) returns (string);
// ...
end
```

At development time...

- Focus on the business logic
- Implement SOA design elements: services and interfaces
- Leverage existing business developers for new SOA development
- Ignore deployment targets/technology while coding/testing

Leverage external web services...

- EGL Interfaces
 - represent external web services
 - are created via import from WSDL
 - allow the EGL developer to stay within the context of EGL programming

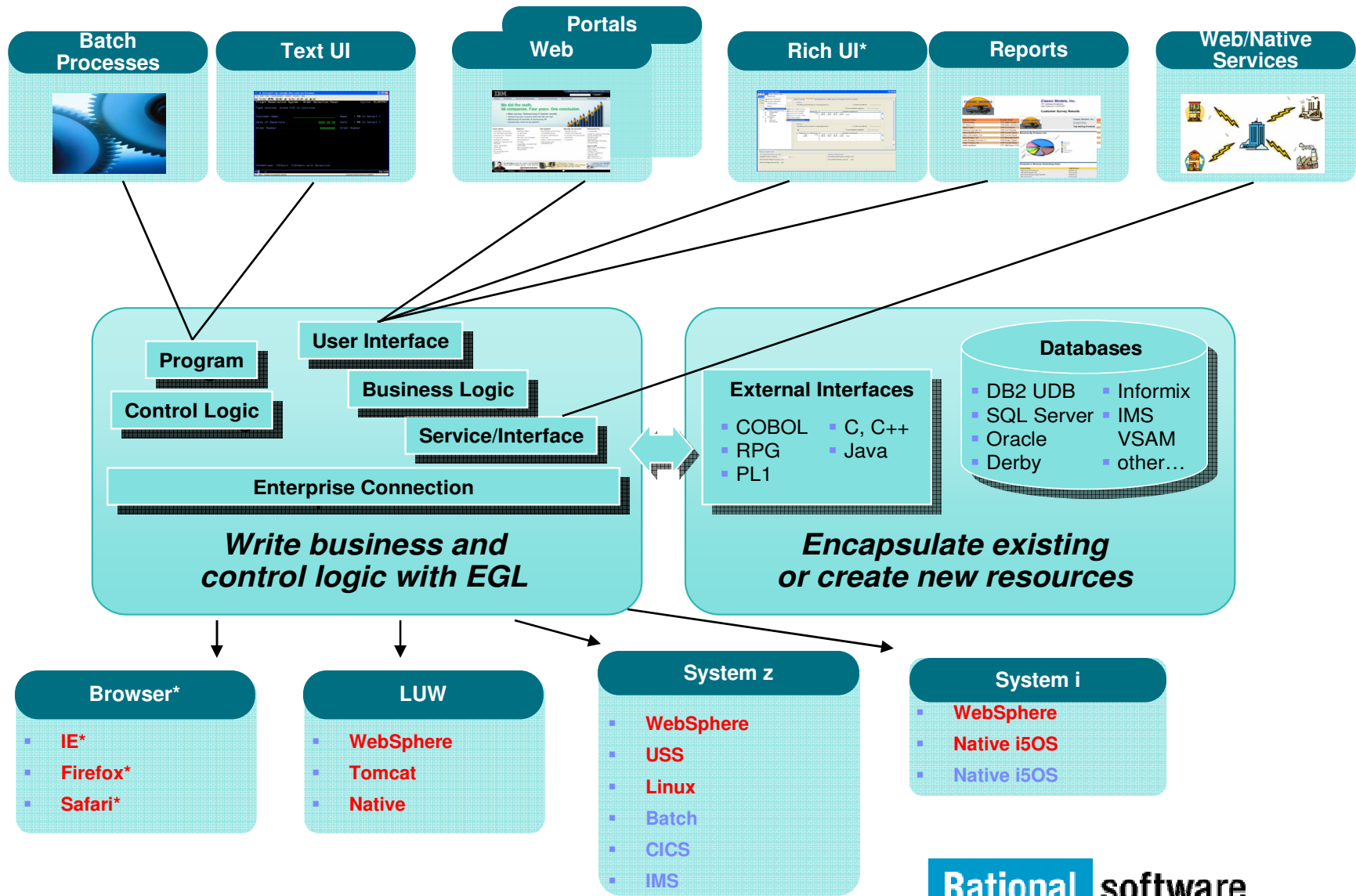


Deploy EGL services...

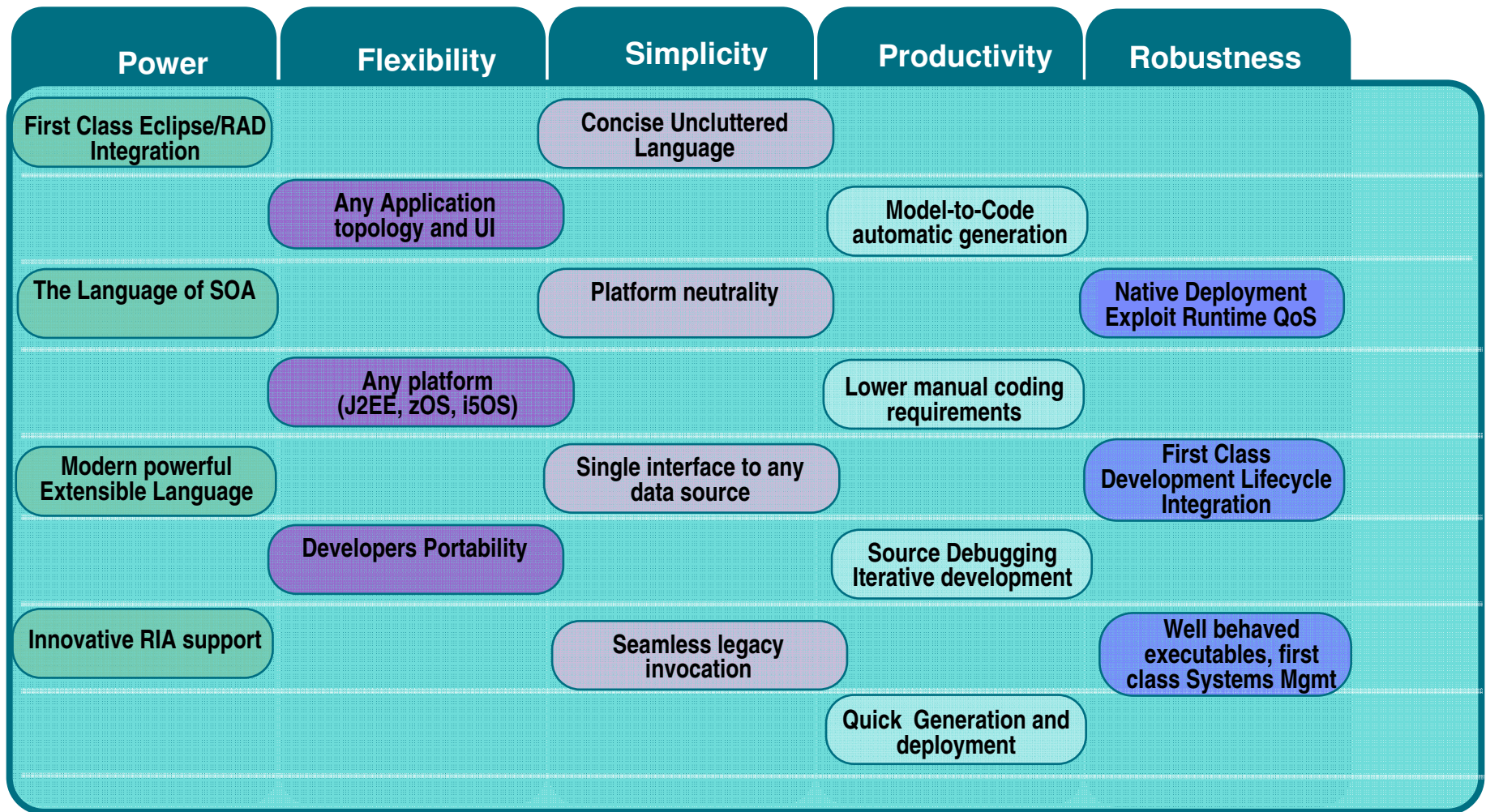
To any platform (Java to WAS/Tomcat, COBOL to CICS, i5/OS)

- As a Web service (uses SOAP)
- As a private service (uses CICS ECI or TCP)

Respond to Broadest Application Needs



The RBD Difference



Accelerate enterprise modernization

With Rational Business Developer

Rapidly integrate existing mainframe assets into service oriented solutions



Leverage trusted mainframes quality of service

Reduce mainframe and server maintenance and new development cost

Break Skills Silos



Maximize use of programming resources and business domain knowledge

Reduce retraining costs

Simplify cross platform development



Respond to business needs independent of underlying technology implementation

Improved quality and architectural integrity

Rational Business Developer is the right solution tailored to specific mainframe and cross-platform needs



KBC

Unify application dev across platforms and transaction managers

Challenge:

- Cost reduction through synergy (reuse solutions for multiple subsidiaries) and integration (support disbursed development team)

Solution:

- Leverage EGL to achieve the Cross-System components development and to eliminate skills silos among their dispersed development teams.

Results:

- Created interchangeable developers. Shift from monolithic (3270) applications to browser based and open systems. Transform 600 mainframe developers to multi platform developers (Unix, mainframe, WAS, IMS).
- Enable component based architecture. Shift to component based architecture, product factories and multi channel.

“... We want to avoid the ‘skill silos,’ what we really need is a large group of general developers who should not worry about target platforms and focus on developing business components, and only a small number of technology specialists, so that we can swiftly allocate general developers to upcoming business needs... EGL is helping us achieve this goal...”

- Lieven Gouwy, IT Architect, KBC, Redmonk Podcast

APIS

Deliver new EU compliance systems based on SOA

NPIS IT d.o.o.

Challenge:

- Tight deadlines for delivery of new systems to comply with EU requirements
- Adhere to IT wide SOA architectural framework and deploy new services to the trusted CICS environment
- Insufficient time to skill up new development staff on CICS

Results:

- “EGL programming model is simple enough and easy to learn for developers with different backgrounds and different initial skills”
- “ ...training costs very low (compared to other technologies)”
- “ We can count on a very flexible pool of “business developers” and move people around projects..”

“We expect 2-3 times faster development than using traditional programming technologies....”

- Robert Stanko Application Development Manager , APIS IT

Summary

- **RBD is a powerful solution that helps**
 - Break the skill silos and leverage business know-how of all developers.
 - Build innovative state-of-the-art solutions without the bite of skill ramp
 - Lower risk and increase project success rate.
 - Lower costs of training.
 - Lower Cost of development.
 - Deliver Projects faster.
- **EGL: IBM newest business language and a key component of IBM Rational Enterprise Modernization solution**
- **For More information**
 - External RBD Web Site (www.ibm.com).
<http://www-306.ibm.com/software/awdtools/developer/business/>
 - RBD/EGL zone on developerWorks
<http://www-128.ibm.com/developerworks/rational/products/rbde/>

