



IBM Software Group

# 面向 **SOA** 的业务驱动开发平台

IBM 软件部 崔心佳



@business on demand.

# 内容

- 为什么要 SOA?
- 什么是 SOA?
- 面向 SOA 的业务驱动开发平台
- 总结

# 内容

- 为什么要 SOA?
- 什么是 SOA?
- 面向 SOA 的业务驱动开发平台
- 总结

## 全球前 450 家企业的 CEO 在想些什么呢？



### CEO 要求

- 以更低的成本获得更高的营业收入
- 关键的能力：快速响应
- 成功的关键因素：最大地发挥人的效率

Source: CEO Study of 456 WW CEOs, IBM Corp. 2004



### CIO 的挑战

- 使 IT 投资与业务发展目标相适应，从而在降低成本的同时获得业务的增长
- 通过 IT 技术来增强企业的快速应变能力和灵活性
- 如何利用 IT 技术来帮助员工和团队提高生产率

Source: Operating Environment Market Drivers Study, IBM Corp. 2004

# 市场竞争对于业务提出了更高的要求

## 灵活性



### ... 更快地增长

- Bekins, a moving company, increased revenue by \$75M through integration with business partners to serve a new market
- PineBank increased customer traffic by 300% and revenues by \$8M

## 执行效率



### ... 降低成本

- Kookmin Bank should save \$250 million from reduction of duplicate processes
- Volkswagen realized a 20% productivity gain

## 快速响应



### ... 提高客户满意度

- Dassault Aviation reduced concept-to-runway development time by 30%
- British Petroleum decreased user-provisioning time from 5 days to 10 minutes

# IT 灵活性造就业务灵活性

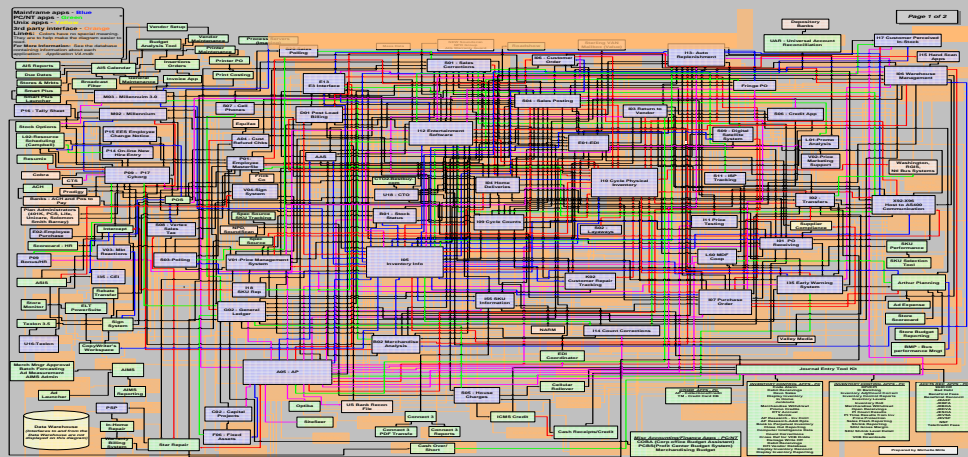
“今天的 IT 架构已经成为很多企业灵活改变业务策略的障碍。”

—McKinsey  
“Flexible IT, Better Strategy”



为什么今天的 IT 架构成为障碍？

- 应用的复杂性
- 单一的竖井应用
- 未公开的接口
- 硬编码的接口
- 设计时就没有考虑将来的变化



Actual application architecture for a consumer electronics company

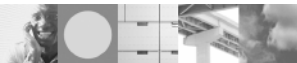
# SOA 给业务带来灵活性

- 构件化
  - ▶ 为各种应用提供标准化的服务接口
- 模块化
  - ▶ 可以灵活地拼装出各种业务流程和应用架构
- 互操作性
  - ▶ 简化应用之间信息的互换
- 可扩展性
  - ▶ 开始时刚好够用，但很容易增加新的功能



# 内容

- 为什么要 SOA?
- 什么是 SOA?
- 面向 SOA 的业务驱动开发平台
- 总结





# 一些基础概念

## ... 服务(Service)?

一项可重复的业务任务，  
如：客户信用审查、创建  
新帐号

## ... Service Orientation?

一种将业务组织模式，在  
这种模式中业务是由一组  
服务相互协作来实现的

## ... Service Oriented Architecture (SOA)?

一种支持面向服务的 IT 架  
构风格

## ... 组合应用(Composite Application)?

一组按照 SOA 架构相互集  
成起来的服务构成了组合应  
用，该组合应用实现了业务  
流程



# 什么是 SOA (Service-Oriented Architecture) ?

对不同的角色 **SOA** 有不同的含义:

- ▶ 业务分析员
  - 一个企业向她的客户或商业伙伴公开的一组服务
  
- ▶ 架构设计师
  - 一种架构风格，其中定义了服务供应者、请求者和服务描述
  - 一组架构设计的原则、模式和标准，强调模块化、封装、松耦合、重用等
  
- ▶ 软件开发人员
  - 一种编程模式，包括了各种标准、工具和标准如 **Web Service**
  - 一种专为服务的组装、监控和管理而定制的中间件解决方案



# 热点 SOA 开发技术

系统元素	描述	实现技术
用户界面	用户如与服务、业务流程和组合应用相交互	JSF, Portlets Rich Clients
系统架构	服务之间如何整合并相互操作	SCA, ESB
系统组合	将各种服务组装成在一起来支持一个可运行的业务流程，形成一个组合应用	SCA, BPEL
业务组件	业务逻辑的基本单元，具有定义明确的接口，并且这些接口独立于任何具体的物理实现	SCA
信息数据	用一种一致的方式来访问数据	SDO

SCA – Service Component Architecture

JSF - Java Server Face

SDO - Service Data Objects

BEPL - Business Process Execution Language

ESB - Enterprise Service Bus

# 内容

- 为什么要 SOA?
- 什么是 SOA?
- 面向 SOA 的业务驱动开发平台
- 总结



# 业务驱动的SOA开发流程

构建、组装、测试服务组件

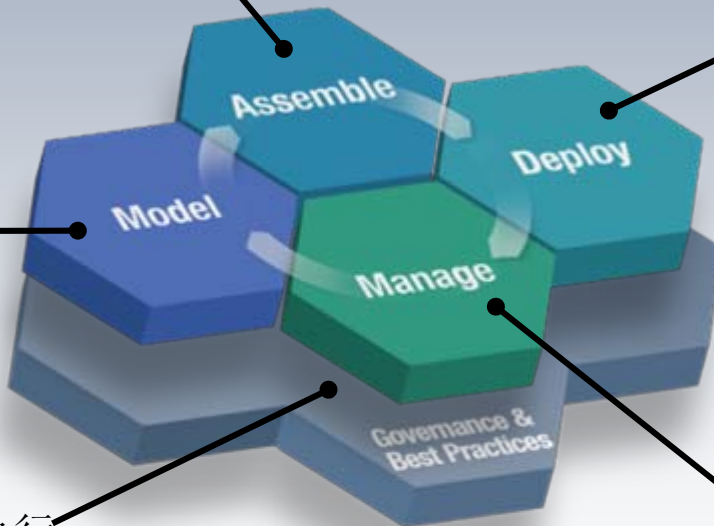
*Rational Application Developer*  
*WebSphere Integration Developer*  
*Rational Functional Tester*

部署SOA应用  
管理服务组件资产

*Rational ClearCase*  
*Rational ClearQuest*  
*Rational Performance Tester*

了解业务流程、定义需求  
设计SOA系统架构

*Rational Software Modeler*  
*Rational RequisitePro*  
*Rational Software Architect*



管理SOA项目执行  
提供SOA开发流程指南

*Rational Method Composer*  
*Rational Portfolio Manager*

监视SOA应用性能

*Tivoli Composite Application Manager*  
*WebSphere Business Monitor*



# SOA 影响到应用的整个生命周期

## 建模

## 组装

## 部署

## 管理



业务分析员



开发人员



集成人员



测试人员



IT 服务经理



IT 运营

“哪些业务服务需要优化？这对我们的 IT 系统有什么影响？”

“应该如何调试这个应用来重现问题呢？”

“我怎么样在现有系统的基础上设计和实现服务？”

“我们应该怎样才能达到我们的业务目标呢？”

“在部署到生产系统之前，我如何来保证我们定义的服务流程会按照我们的设计来工作？”

“我们的组合应用中用到了很多外部了服务，应该如何测试这样的应用呢？”

“我们新建的SOA应用可以满足性能指标吗？”

“我如何才能确认这个服务组件能够可靠安全地与其他服务组件一起正常工作？”

“我们有一些服务组件是供其他的合作伙伴使用的，必须保证这些服务组件的质量能够达到我们所承诺的服务水平”

“是SOA架构的哪一部分引起了这个服务问题，是应用服务器还是消息总线呢？”

“是什么原因引起了这个服务问题 – 是需求、服务流程还是应用本身？”

Rational

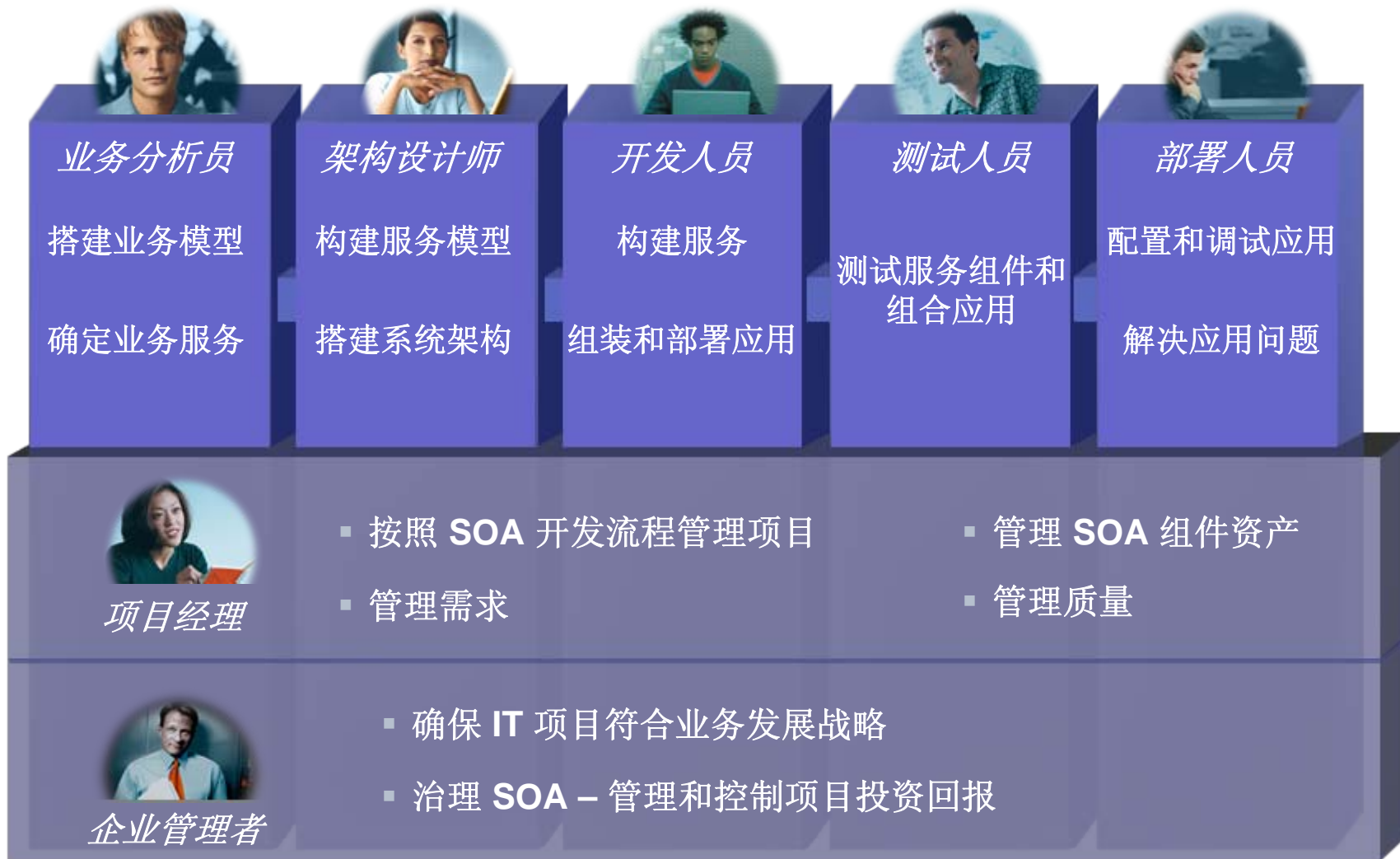
WebSphere

Lotus

DB2

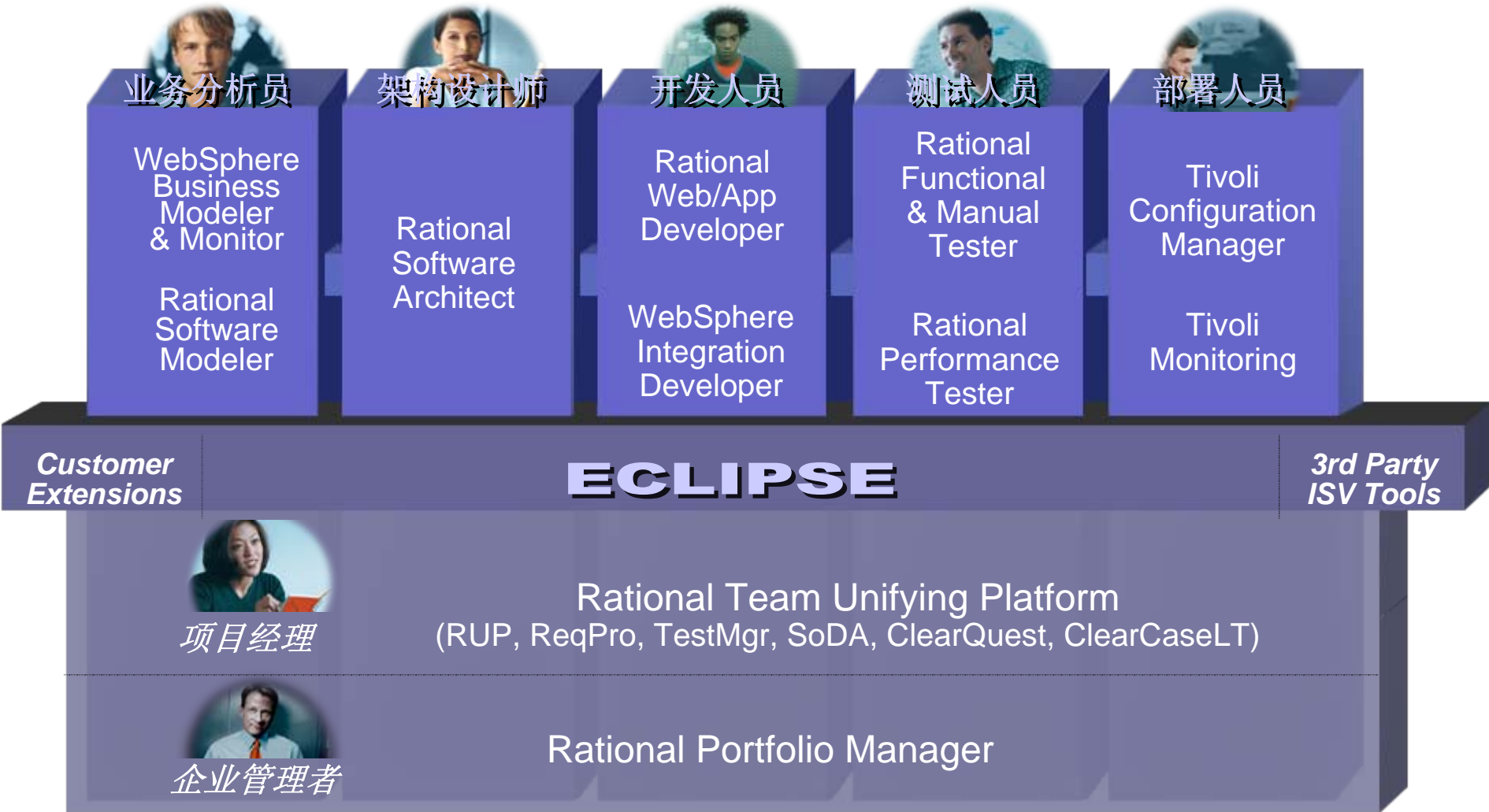
Tivoli

# 面向 SOA 的业务驱动开发团队



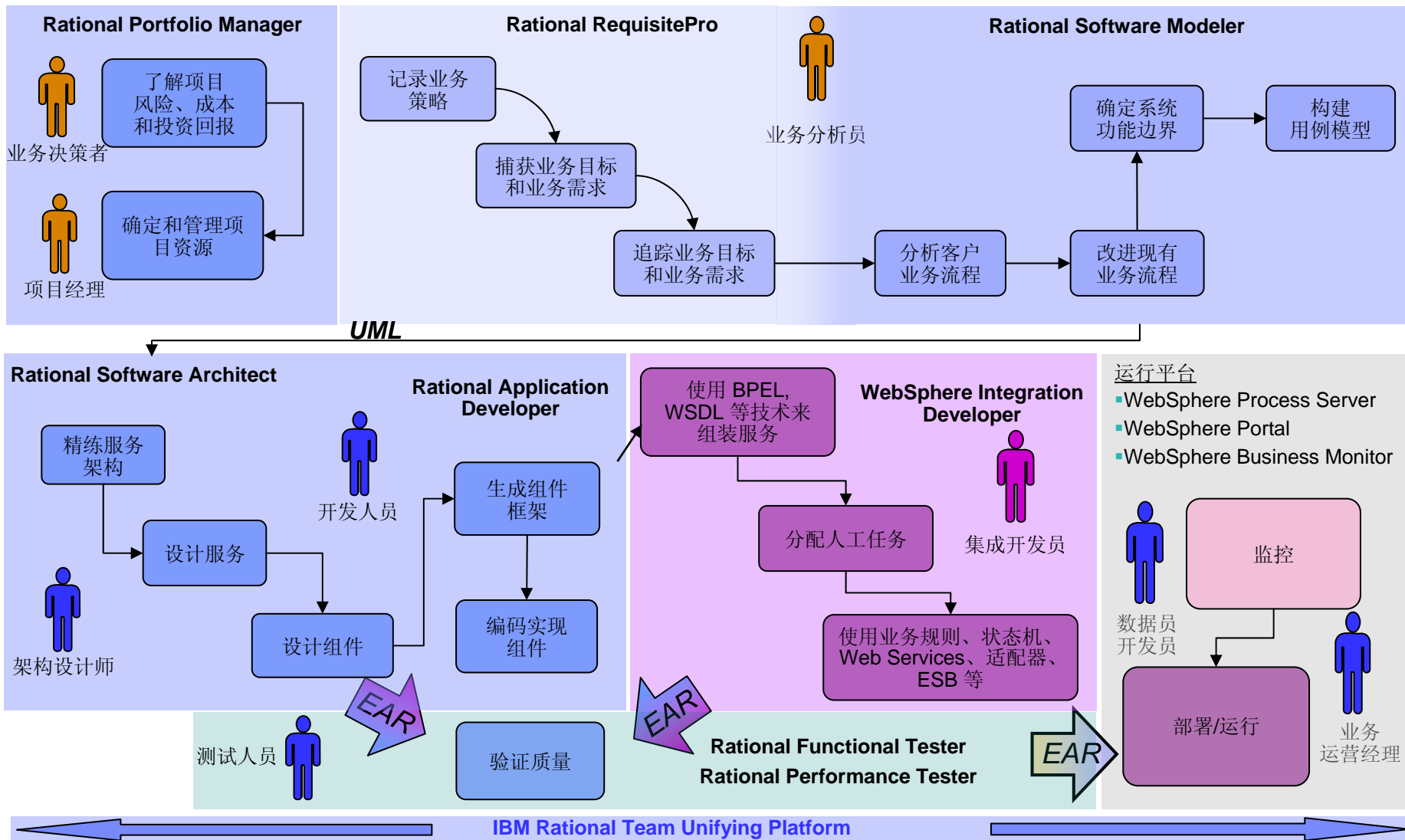


# 面向 SOA 的软件开发平台 SDP IBM Rational Software Development Platform

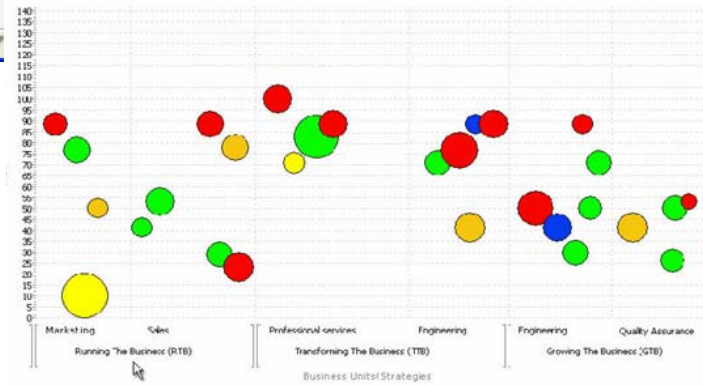
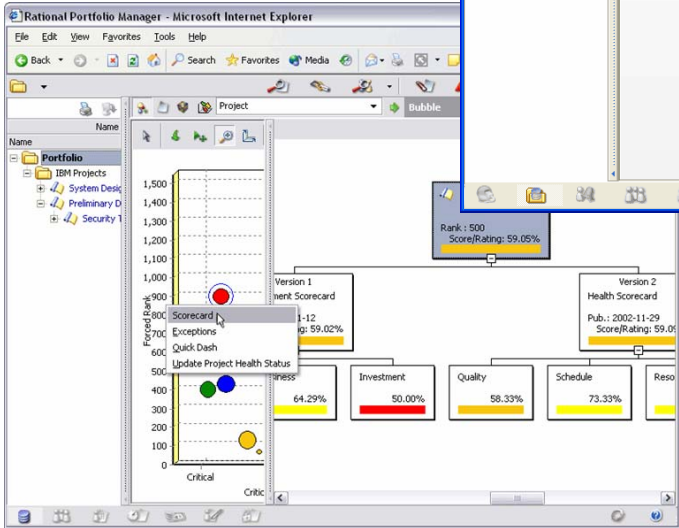
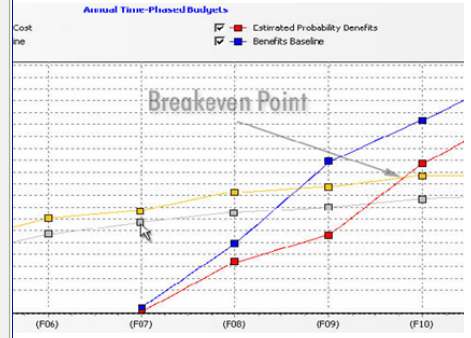
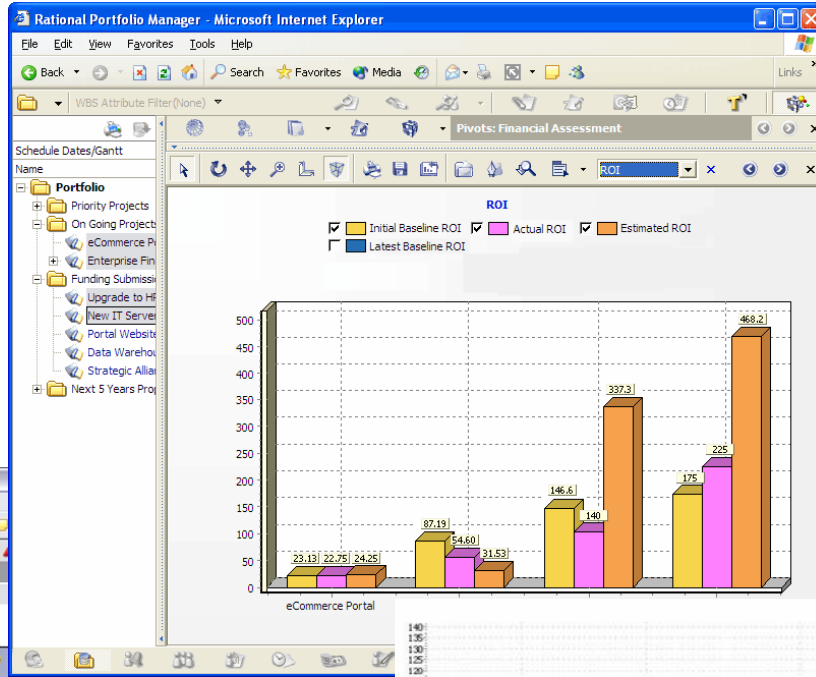
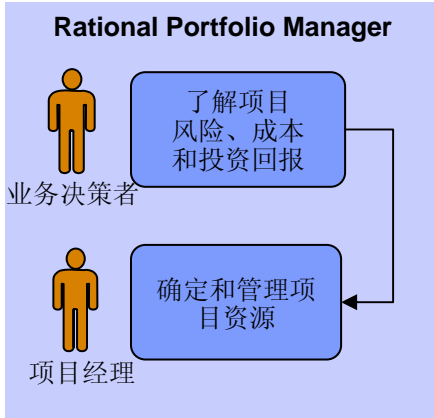




# 面向 SOA 的业务驱动开发流程

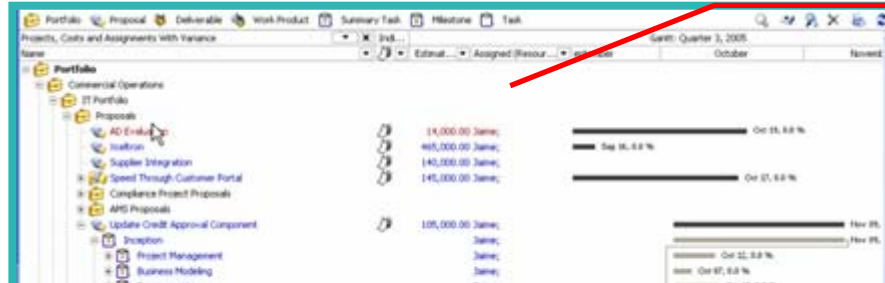
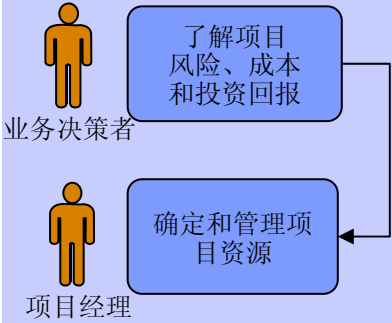


# 项目组合管理 – 控制和管理项目投资

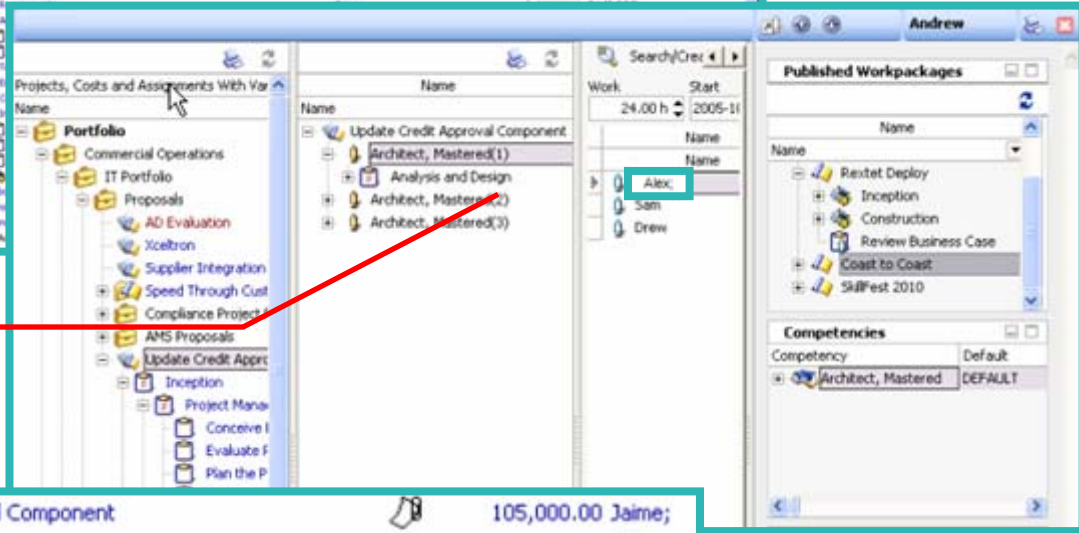


# 项目管理 - 计划、预算和分配资源

## Rational Portfolio Manager



根据预先定义的标准模板来制定项目计划

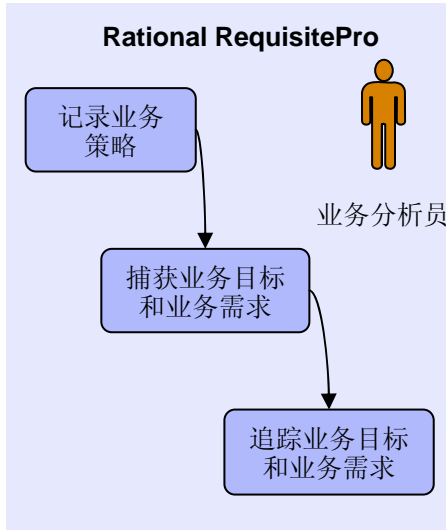


保证项目资源，提高资源利用率



根据任务优先级和资源技能来分配资源

# 需求管理 – 定义 SOA 方案的业务需求



vision.VIS - Microsoft Word

ClassicsCD Web Shop

1. Introduction

1.1 Purpose of the Vision Document

The purpose of this document is to define and to define high-level business re... ClassicsCD.com Web Shop system is an... is intended to provide a new channel of s... retail operation.

Requirements	Type	Package	Priority
<b>FEAT1: Secure payment method</b> Secure payment method	Functional	Web Shop System	Must
<b>FEAT2: Easy browsing</b> Easy browsing for available titles	Functional	Web Shop System	Should
<b>FEAT3: Search by multiple criteria</b> Ability to search for CDs by multiple criteria		Web Shop System	Must

Business Goals	Measure	Change Value	Change Kind	Verification	Timeframe
Classic CDs over the internet	Qualitative	25	Percent		Q3
business	Quantitative	100	Percent	Purchases CDs	Q4
use the time to order CDs	Quantitative	-200	Percent	Compare current time agai	Q1
ase the number staff	Quantitative	3	Direct	Reduce staff	
order business to the entire...	Qualitative	5	Direct	Reduced components	

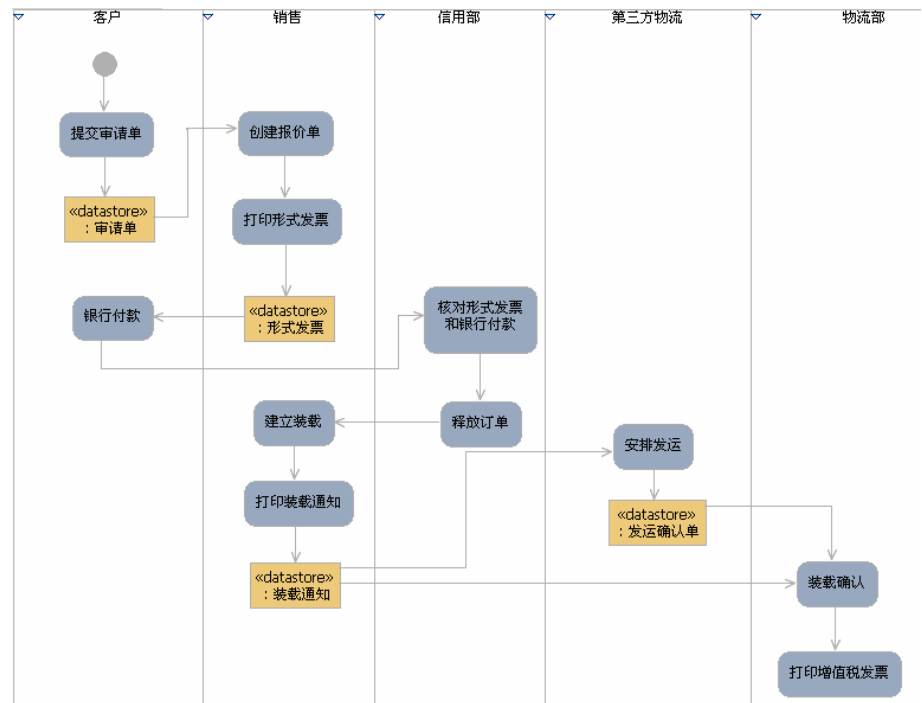
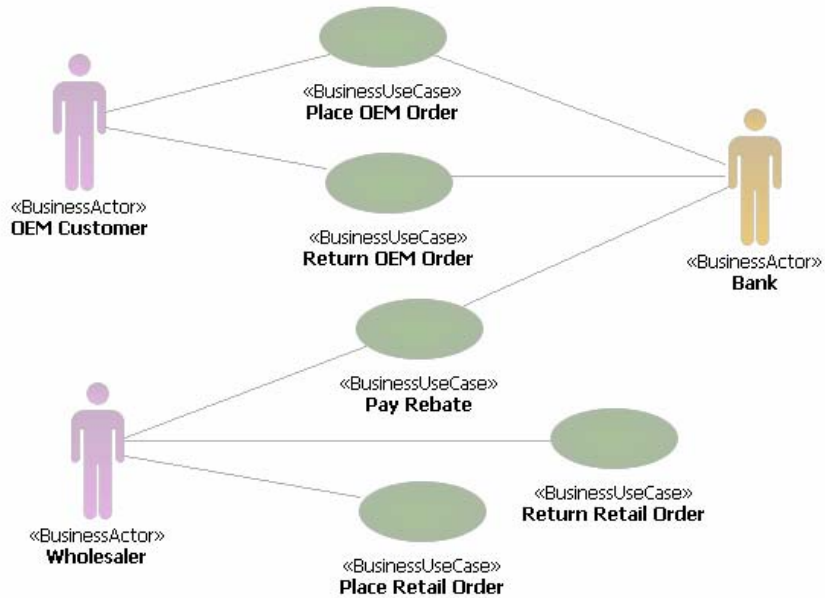
5 requirements

Requirements	Author	Revision	Reason
RULE1: Minimum of 3 CDs free shipping	Unknown	1.0001	Requirement Text C
RULE2: Maximum of 10 CDs, results in a 10% discount	Unknown	1.0001	Requirement Text C
RULE3: If the inventory # of a particular CDs falls below 3...	Unknown	1.0002	Requirement Text C
* <Click here to create a requirement>	Unknown	1.0000	

Requirement RULE3 modified

3 requirements

# 业务建模 - 了解业务流程，确定业务服务

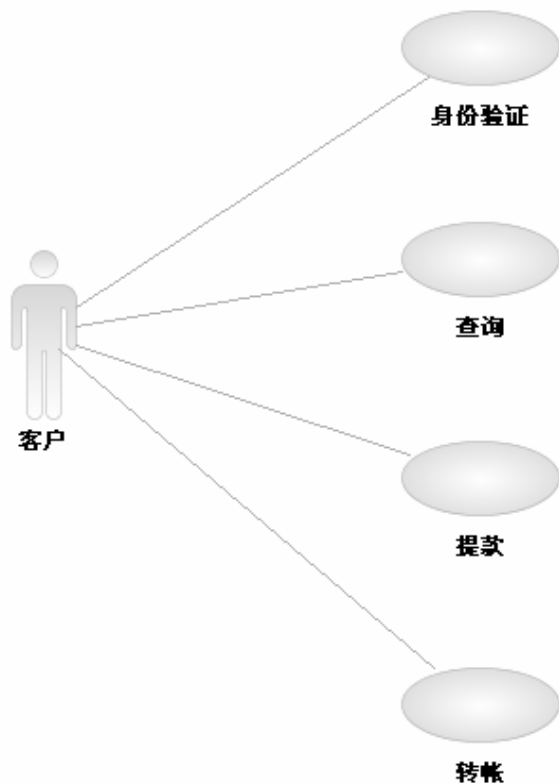
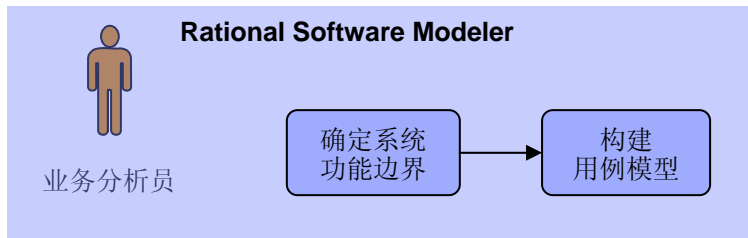


**Rational Software Modeler**  
UML Profile for Business Modeling

业务分析员

分析客户业务流程 → 改进现有业务流程

# 用例建模 - 定义系统需求



ATM系统	版本: 1.1
用例规约: 客户身份验证	日期: 2002-08-27
用例规约-客户身份验证.doc	

## 用例规约: 客户身份验证

### 1. 简要说明

该用例描述 ATM 机是如何验证客户身份的。

### 2. 事件流

客户使用 ATM 机提供的各项服务之前, 必须先通过该用例来进行用户身份验证。

#### 2.1 基本流

##### 1. 插入信用卡

客户将信用卡插入系统, 系统读取信用卡上的客户帐号信息, 并向后台服务器系统确认该信用卡的有效性。

##### 2. 输入密码

系统提示客户输入信用卡密码, 客户输入 6 位密码, 系统向后台服务器检查用户密码是否正确。

##### 3. 选择服务

客户通过身份验证后, 系统显示操作主菜单供客户选择查询, 提款, 转帐服务, 客户选择他所需要的服务。

#### 2.2 备选流

##### A1. 无效信用卡

在基本流步骤 1 中, 客户插入的信用卡在后台服务器中没有对应的帐号, 系统显示错误信息并退出信用卡, 用例结束。

##### A2. 客户密码不正确

在基本流步骤 2 中, 客户输入错误的信用卡密码, 系统提示客户重新输入密码, 客户重新输入正确信用卡密码后继续基本流中的下一个步骤; 如客户输入密码错误超过 3 次, 系统没收客户插入的信用卡, 用例结束。

##### A3. 退出

在基本流的任何一个步骤中, 客户都可以选择“取消(Cancel)”退出, 系统退出信用卡, 用例结束。

### 3. 用例场景

#### 3.1 成功场景

通过客户身份验证: 基本流

取消客户身份验证: 基本流, 退出

#### 3.2 失败场景

无效信用卡: 基本流, 无效信用卡

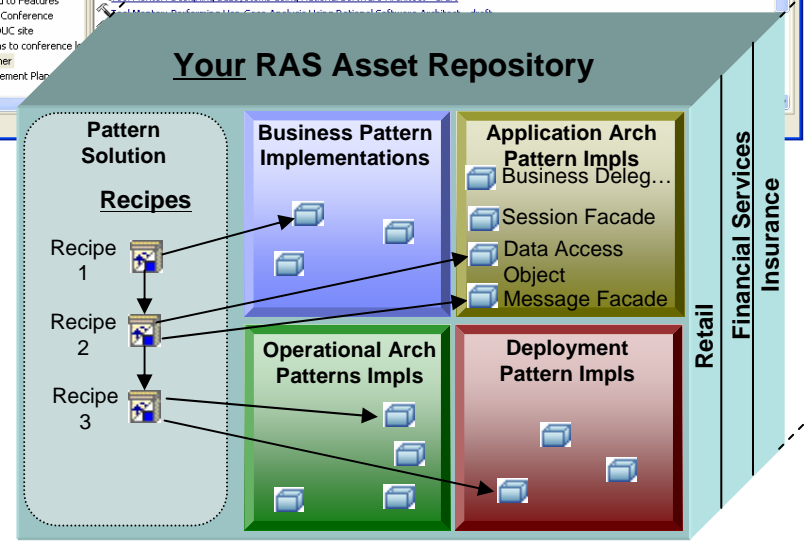
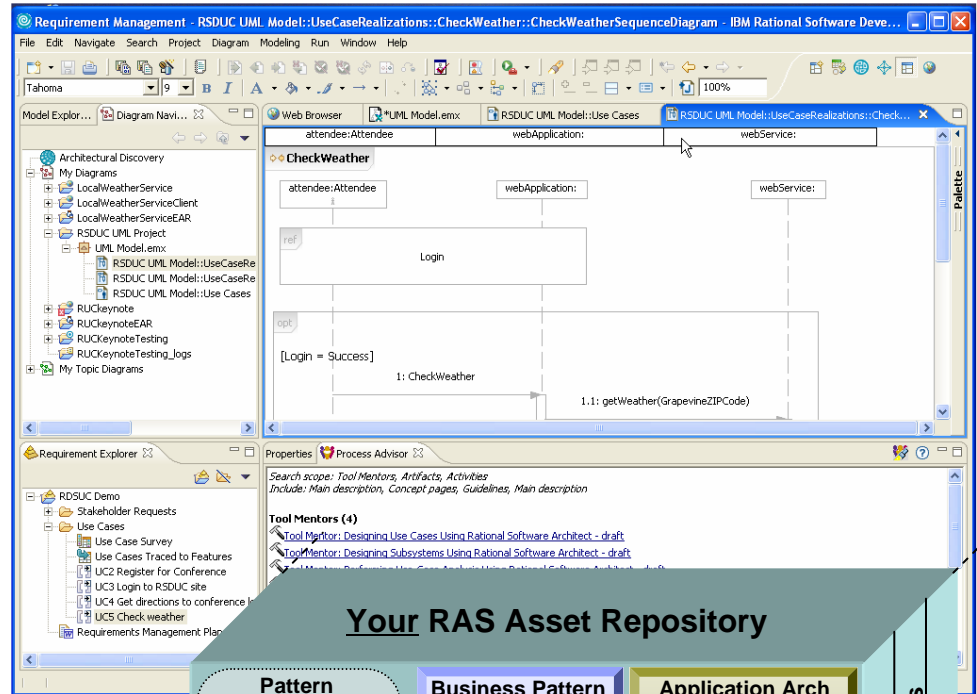
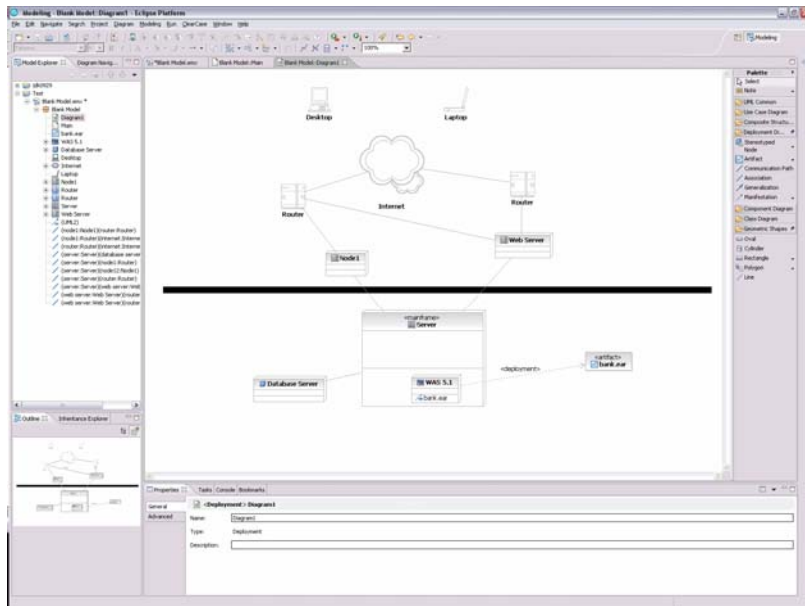
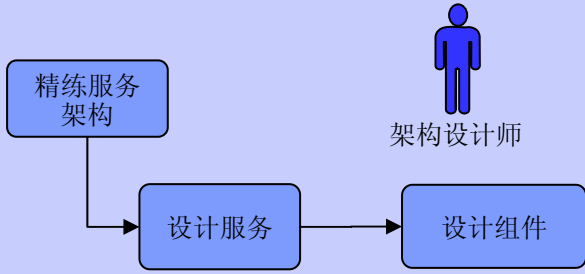
重新输入客户密码: 基本流, 客户密码不正确

没收客户信用卡: 基本流, 客户密码不正确 (密码输错 3 次以上)

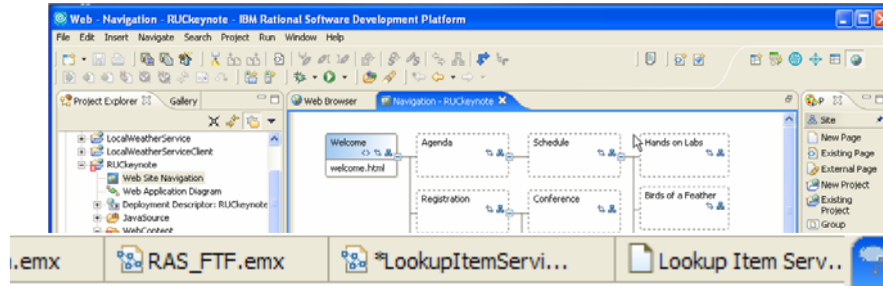
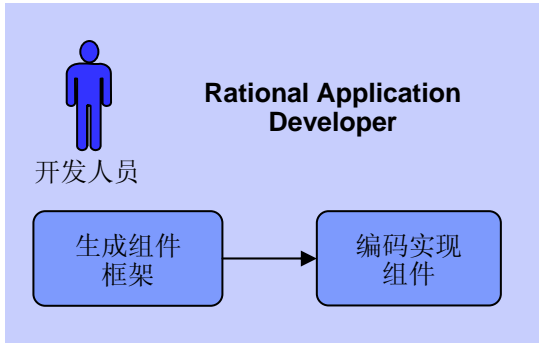


# 可视化建模 - 搭建系统架构

## Rational Software Architect UML Profile for Software Service



# 构建服务组件 - 快速实现服务



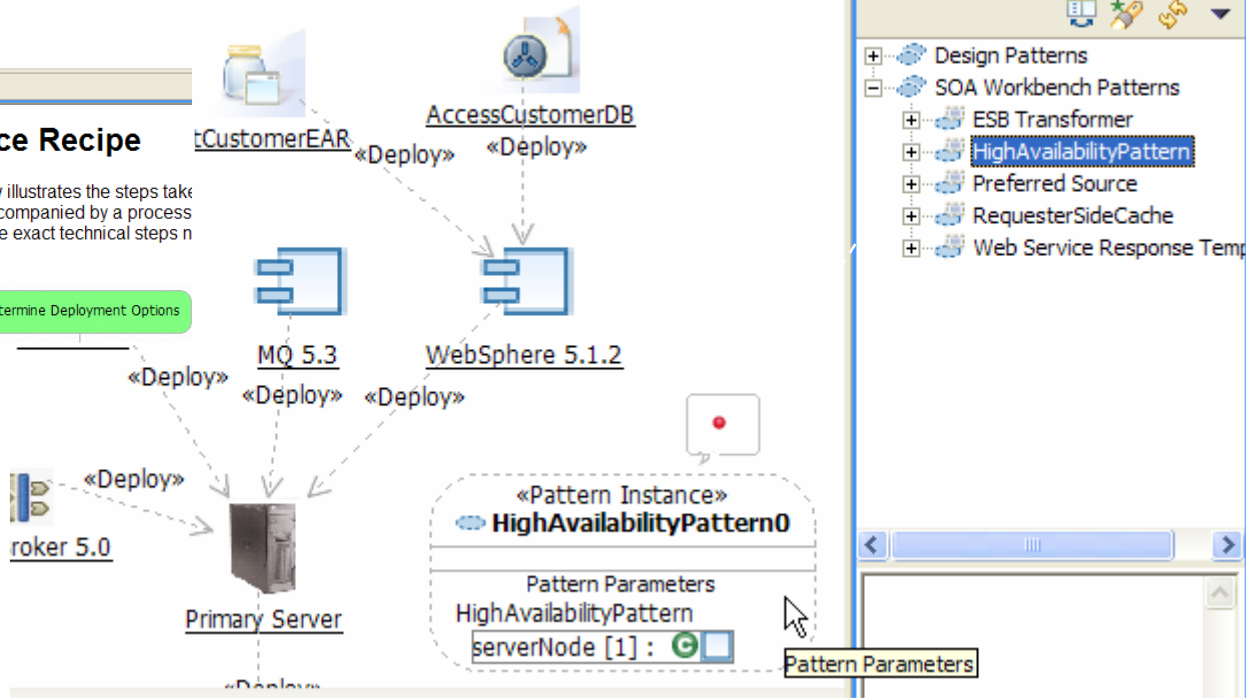
Properties Tasks Console Bookmarks **Solution Guide** Search

**Solution Guide for RSDC\_Deploy Service Recipe**

1. Determine Deployment Options
2. Create and Configure the Deployment Model
3. Apply Deployment Patterns
  - High Availability Pattern
  - High Availability Pattern
4. Deploy

**Deploy Service Recipe**

The following Activity flow illustrates the steps take solution. Each step is accompanied by a process cheat sheet explaining the exact technical steps n to complete the work.



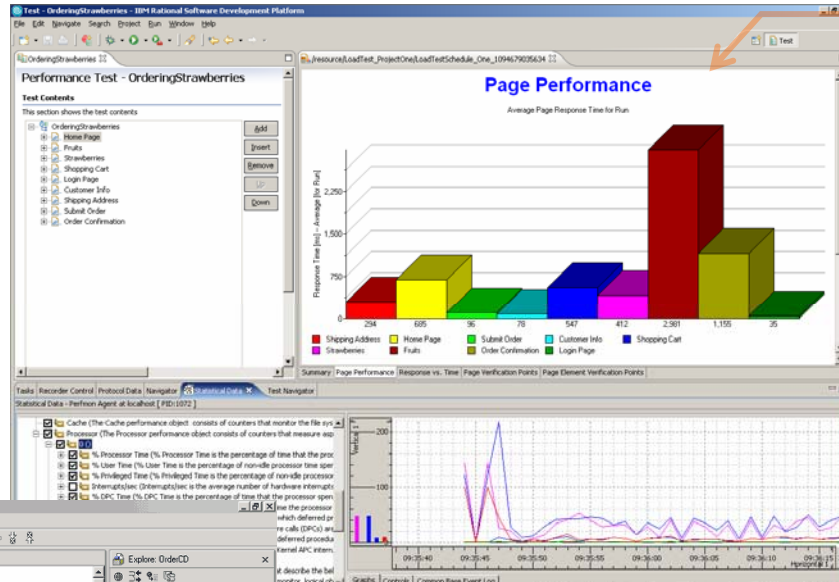


# 测试服务组件和组合应用

**Rational Functional Tester**  
**Rational Performance Tester**

验证质量

测试人员



根据从业务需求导出的用例来执行自动化的测试，并且测试系统的性能

```

public void testMain(Object[] args)
{
    startApp("ClassicsWeb");
    // Browser: MS Internet Explorer
    Link_Explorestorefront().click();
    // Browser: MS Internet Explorer: Classics Home Page
    Image_button_catalog().click();
    // Browser: MS Internet Explorer: Classics Catalog Page
    Image_putinshoppingcart().click();
    // Browser: MS Internet Explorer: Classics Catalog Page
    Image_putinshoppingcart2().click();
    // Browser: MS Internet Explorer: Classics Login Page
    Text_customerID().click();
    CDCustomerIDEntry().inputKeys("jmoore");
    Html_password().click();
    CDCustomerIDEntry().inputKeys("*****");
    CheckBox_chkRememberON().click();
    Button_loginsubmit().click();
    // Browser: MS Internet Explo
    HtmlTableContent_DataVP().pe
    // Browser: MS Internet Explo
    Document_CashierInMoore().cl
    Button_PlaceOrdersubmit().cli
    click() void GuiTestObjec
    click(MouseModifiers) void GuiTestObjec
    click(MouseModifiers.Pane) void GuiTestObjec
    click(Point) void GuiTestObjec
    close() Object Object
    doubleClick() void GuiTestObjec
    doubleClick(MouseModifiers) void GuiTestObjec
    
```

**Launch Browser by clicking the icon shown on your Desktop:**

- Launch Browser
- Enter Address http://www.ibm.com
- Logon For Trading
- Navigate to Trading
- Sell Mutual Fund
- Verify Transaction in

# 管理可重用的服务组件

Rational ClearCase  
Rational ClearQuest

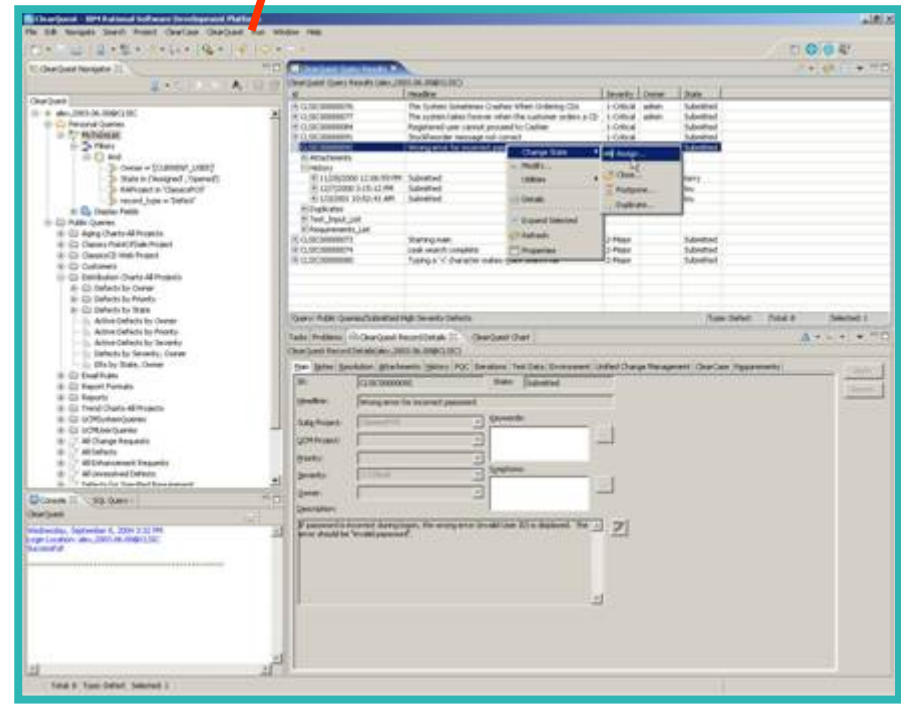
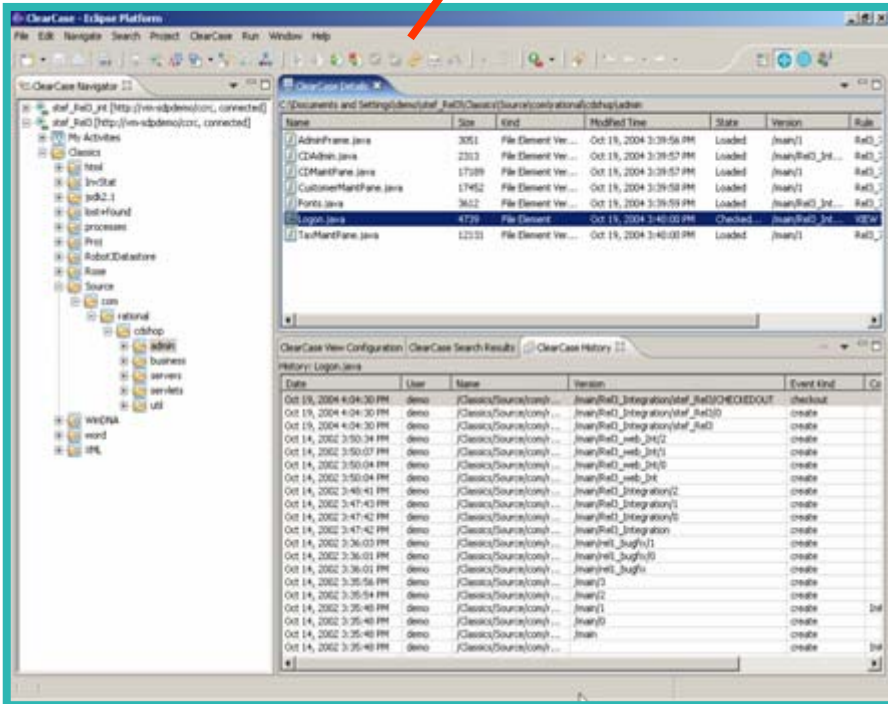
管理可重用  
服务资产



配置管理员

使用 Rational  
Clearcase 管理可  
重用服务资产

使用 Rational  
ClearQuest 管理变更



# 专为 SOA 开发定制的软件开发流程

The screenshot displays the Rational Unified Process (RUP) interface within a Mozilla Firefox browser. The main window title is 'Rational Unified Process - Mozilla Firefox'. The address bar shows the file path: 'file:///D:/temp/rupsoa-v3/index.htm'. The interface includes a menu bar (File, Edit, View, Go, Bookmarks, Tools, Help) and a toolbar with navigation icons. Below the browser window, the Rational Unified Process logo is visible, along with navigation links for Glossary, Index, Feedback, and About. The main content area is titled 'Workflow Detail: Design Services' and shows a 'Design services brief description.' with a 'Topics' sidebar containing links for Description, Related Information, Timing, and How to Staff. The central diagram illustrates a workflow starting with a 'Designer' icon, leading to a 'Service Design' box, which then branches into two parallel paths: one involving 'Service Specification' and 'Service Model', and another involving 'Software Architecture Document' and 'Design Model'. Below the diagram, a 'Description' section states: 'This Workflow Detail has the following goals: Refine the definitions of service design elements by providing detailed specifications of service'. The bottom status bar shows 'Done'.

Rational Unified Process  
RUP Plug-in for SOA

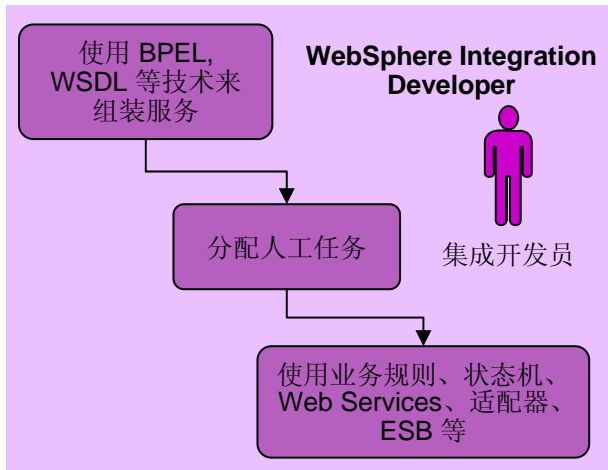
定义SOA  
开发流程



流程工程师

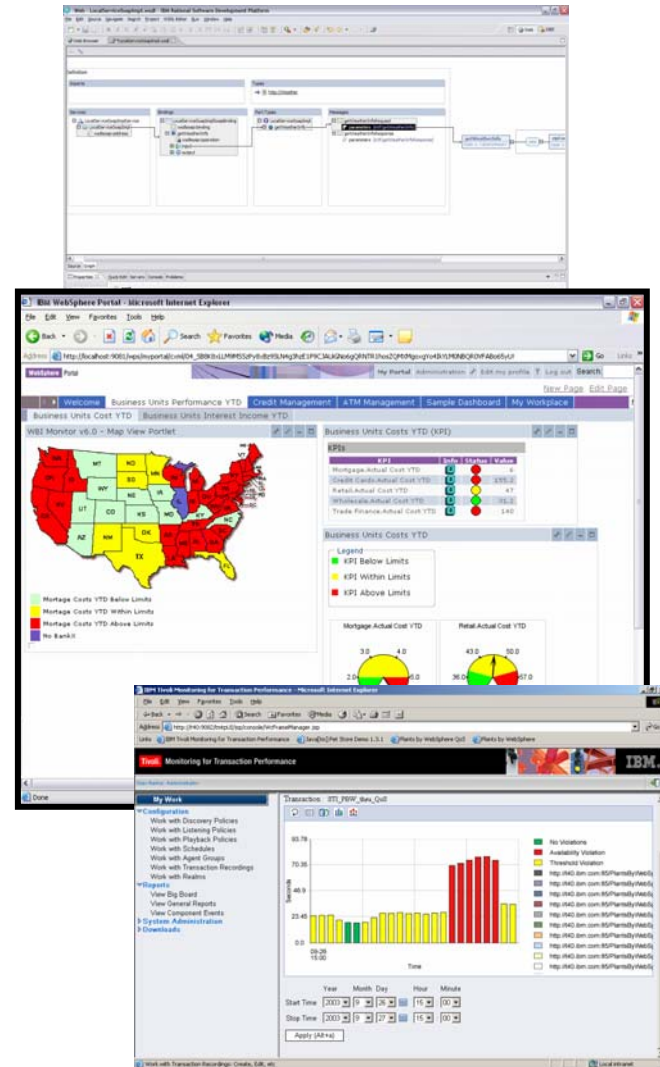
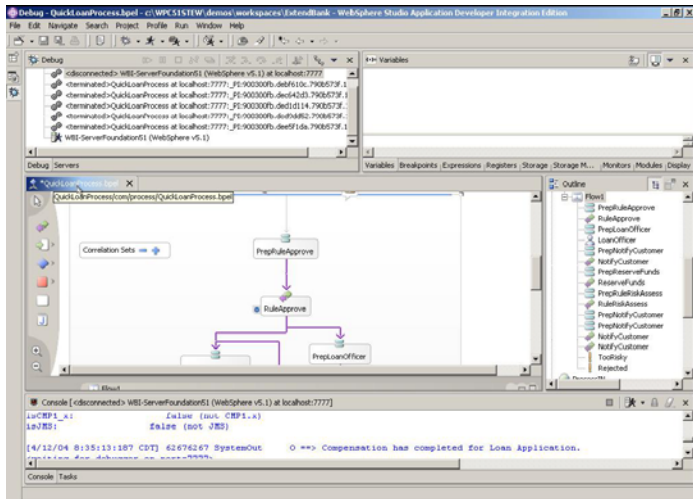
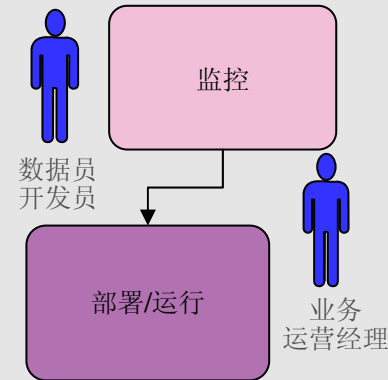
专为 SOA 开发定义的  
工作流程

# 组装、部署和监控 SOA 应用



## 运行平台

- WebSphere Process Server
- WebSphere Portal
- WebSphere Business Monitor



# IBM Rational SDP for SOA – 入门级配置

角色	SOA 开发过程中的职责	推荐工具
IT 主管和项目经理	确保项目投资和业务目标相一致	Rational Portfolio Manager
业务分析员	分析和定义业务流程	Rational Software Modeler Rational RequisitePro
架构设计师	设计业务服务	Rational Software Architect UML Profile for Software Service UML Profile for Business Modeling
开发人员	构建服务组件	Rational Application Developer
测试人员	测试服务组件	Rational Functional Tester Rational Performance Tester
集成开发员	组装和部署服务	Websphere Integration Developer
项目开发团队	管理软件资产生命周期	Rational ClearCase Rational ClearQuest
	软件开发最佳实践经验和指南	Rational Unified Process ▶ RUP Plug-in for SOA



# IBM Rational SDP for SOA – 中型团队配置

角色	SOA 开发过程中的职责	推荐工具
IT 主管和项目经理	确保项目投资和业务目标相一致	Rational Portfolio Manager
业务分析员	分析和定义业务流程	Rational Software Modeler Rational RequisitePro
架构设计师	设计业务服务	Rational Software Architect UML Profile for Software Service UML Profile for Business Modeling
开发人员	构建服务组件	Rational Application Developer
测试人员	测试服务组件	Rational Functional Tester Rational Performance Tester
集成开发员	组装和部署服务	Websphere Integration Developer
项目开发团队	管理软件资产生命周期	Rational ClearCase Rational ClearQuest
	软件开发最佳实践经验和指南	Rational Unified Process ▶ RUP Plug-in for SOA

# IBM Rational SDP for SOA – 企业级解决方案

角色	SOA 开发过程中的职责	推荐工具
IT 主管和项目经理	确保项目投资和业务目标相一致	Rational Portfolio Manager
业务分析员	分析和定义业务流程	Rational Software Modeler Rational RequisitePro
架构设计师	设计业务服务	Rational Software Architect UML Profile for Software Service UML Profile for Business Modeling
开发人员	构建服务组件	Rational Application Developer
测试人员	测试服务组件	Rational Functional Tester Rational Performance Tester
集成开发员	组装和部署服务	Websphere Integration Developer
项目开发团队	管理软件资产生命周期	Rational ClearCase Rational ClearQuest
	软件开发最佳实践经验和指南	Rational Unified Process RUP Plug-in for SOA

# 内容

- 为什么要 SOA?
- 什么是 SOA?
- 面向 SOA 的业务驱动开发平台
- 总结





# 总结

## CEO 面临的挑战

### 业务增长

- 提高营业额
- 降低成本

### 提高效率

- 灵活性
- 人员效率
- 流程效率

### 快速响应

- 对客户需求的响应
- 市场需求的变化
- 业务需求的变化

## SOA 的能力

灵活性

可重用

高效率

适应变化的设计

## 业务价值

↑ 提高 IT 灵活性

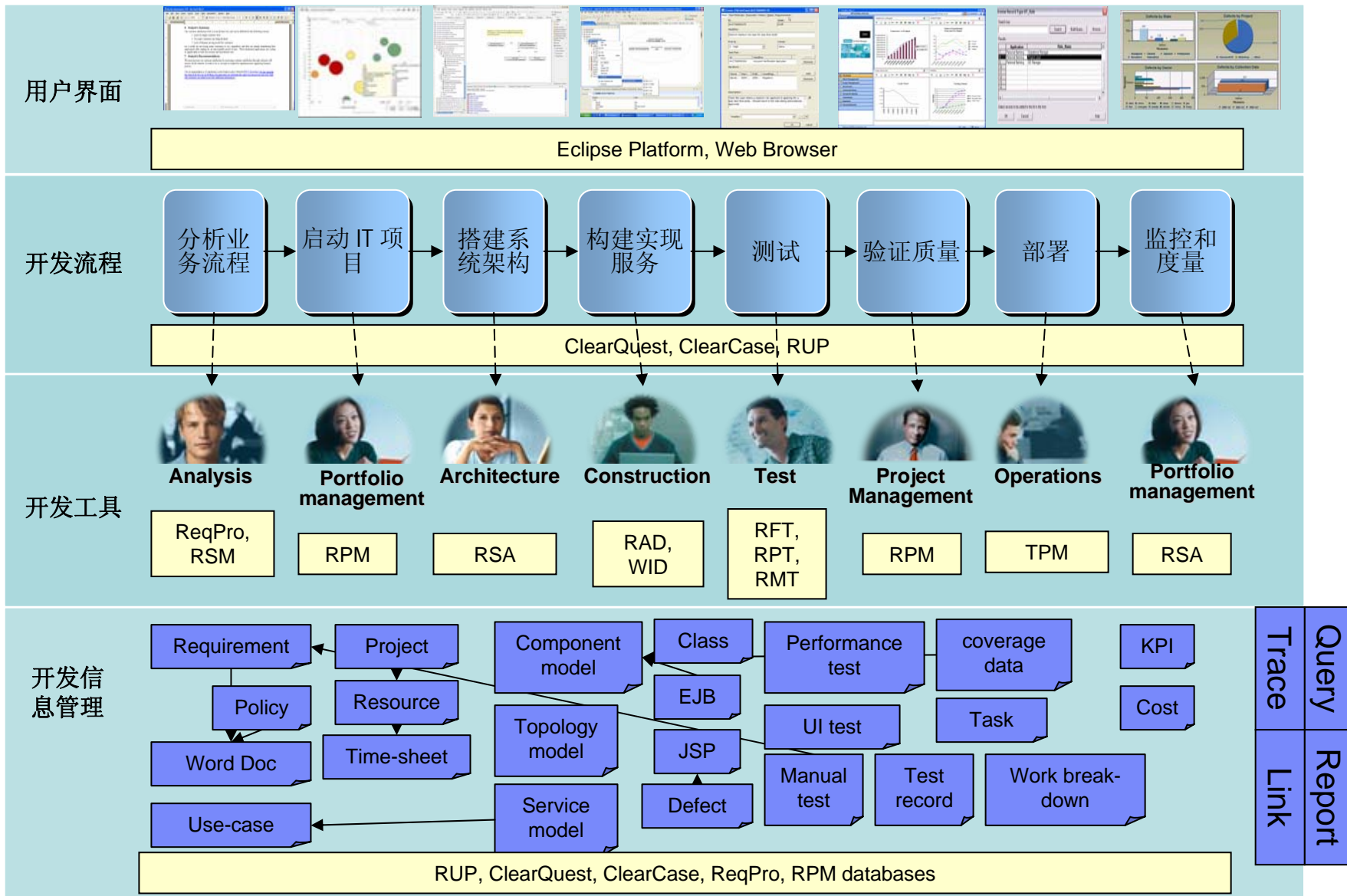
↓ 降低开发和维护成本

↑ 增加成功概率

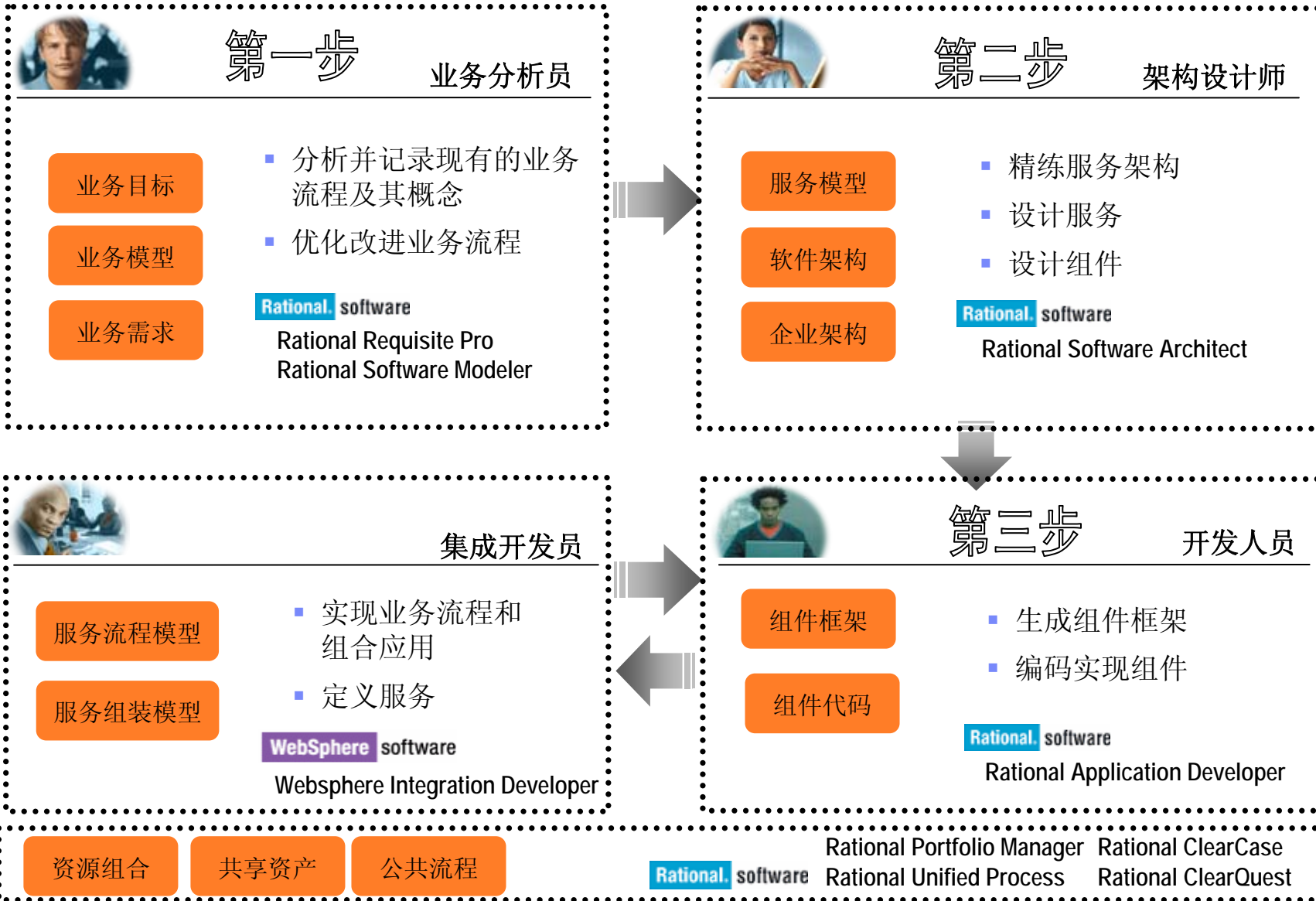
↑ 提高质量

↓ 缩短产品上市时间

# 面向 SOA 的软件开发流程



# SOA 开发三步曲



Thank  
You

