



IBM Software Group

面向 **SOA** 的业务驱动开发平台

Bao peng



@business on demand.

Rational背景



- 成立于1981
- 90年代初关注面向对象
- 90中期致力于UML统一
- 90末期工程化平台建设
- 嵌入式和企业级开发领域
- 2002. 12 并入IBM
- 新的发展

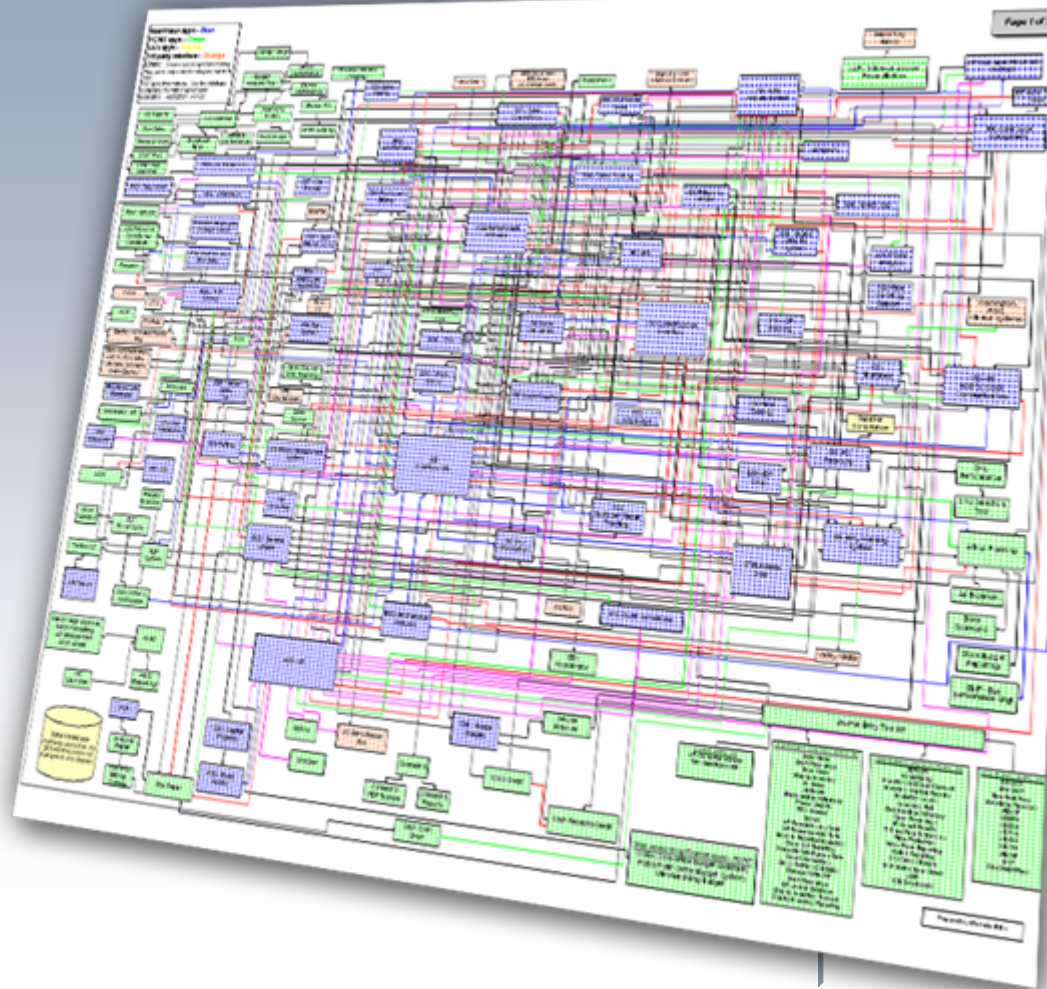


内容

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

IT架构现状

- 各种各样的平台
- 多种多样的应用
- 多种多样的架构
- 系统难以升级与维护
- 应用难以集成和重构
- 数据难以整合和共享
- 流程难以改变和重用
-



IT架构现状

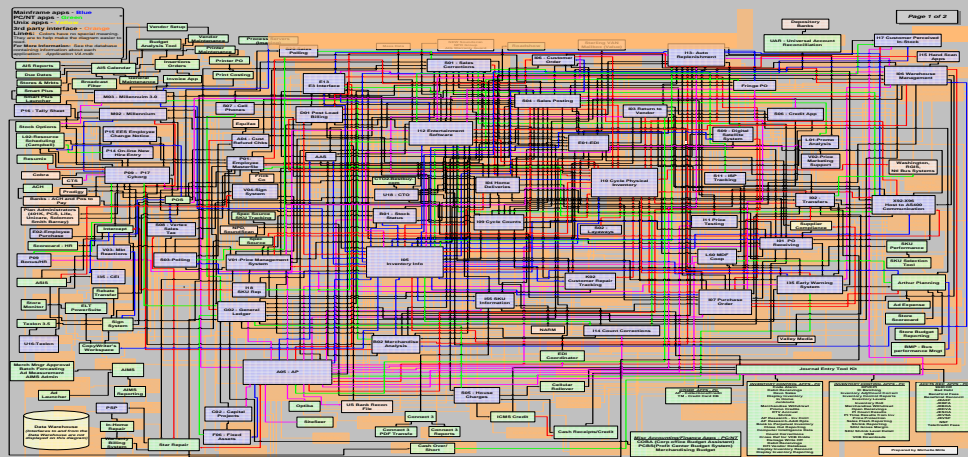
“今天的 IT 架构已经成为很多企业灵活改变业务策略的障碍。”

—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



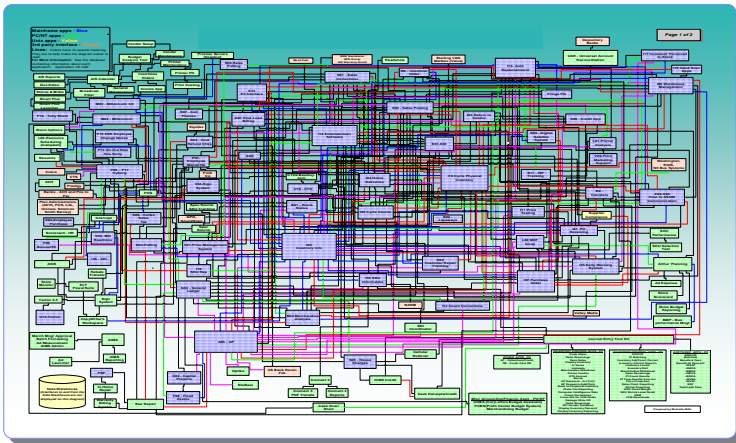
Actual application architecture for a consumer electronics company

SOA是必然选择

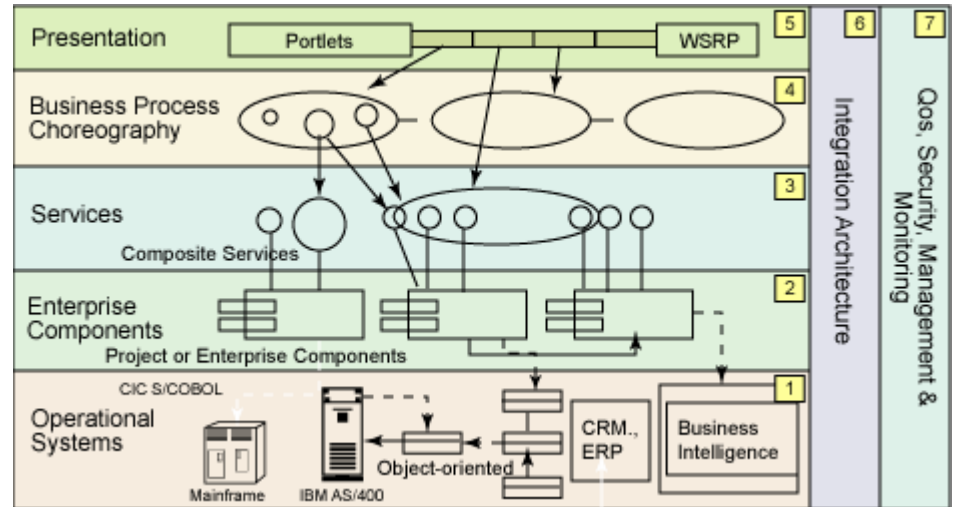


SOA的能力

使用SOA前



使用SOA后



■ 业务方面

- ▶ 提高效率：将业务流程从冗余重复的流程向维护成本较低、高度利用、共享服务的流程进行转变；
- ▶ 加快响应：迅速实现关键业务服务来满足市场需求，为客户、雇员和合作伙伴提供更高水准的服务；
- ▶ 降低成本：降低整个重构业务的复杂性，达到节约时间和资金的目的。

■ IT方面

- ▶ 增加重用：通过重用以前开发和部署的共享服务，实现了更有效的应用程序项目开发和交付；
- ▶ 降低复杂性：基于标准的兼容性，与点到点的集成相比降低了复杂性；
- ▶ 集成遗留系统：用作可重用服务的遗留应用程序降低了维护和集成的成本。

内容

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



面向服务的设计和开发

构建、组装、测试服务组件

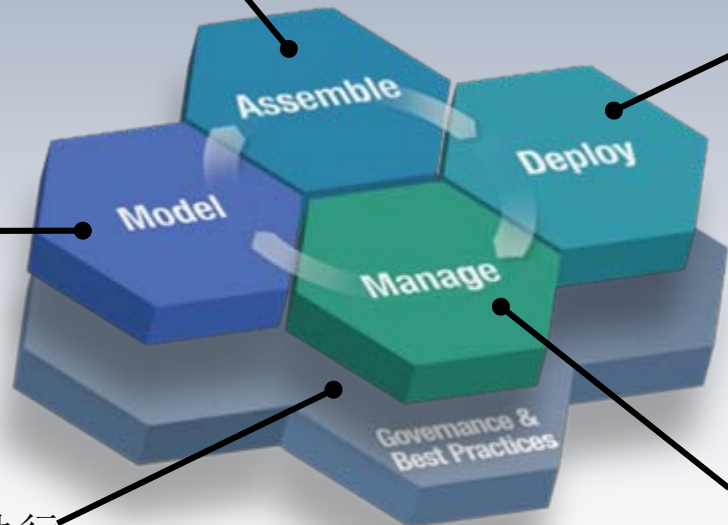
Rational Application Developer
WebSphere Integration Developer
Rational Functional Tester

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

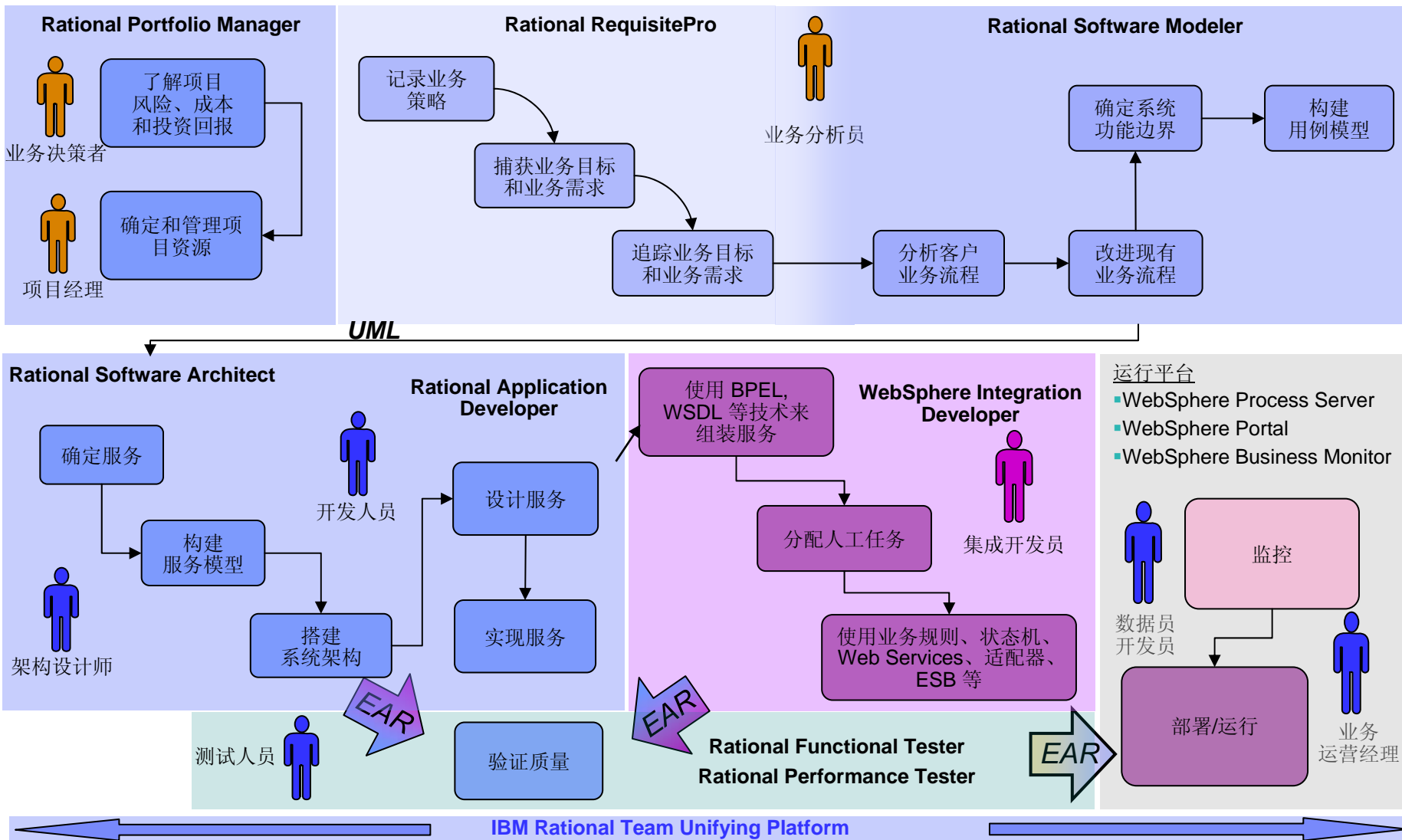
Rational Method Composer
Rational Portfolio Manager

监视SOA应用性能

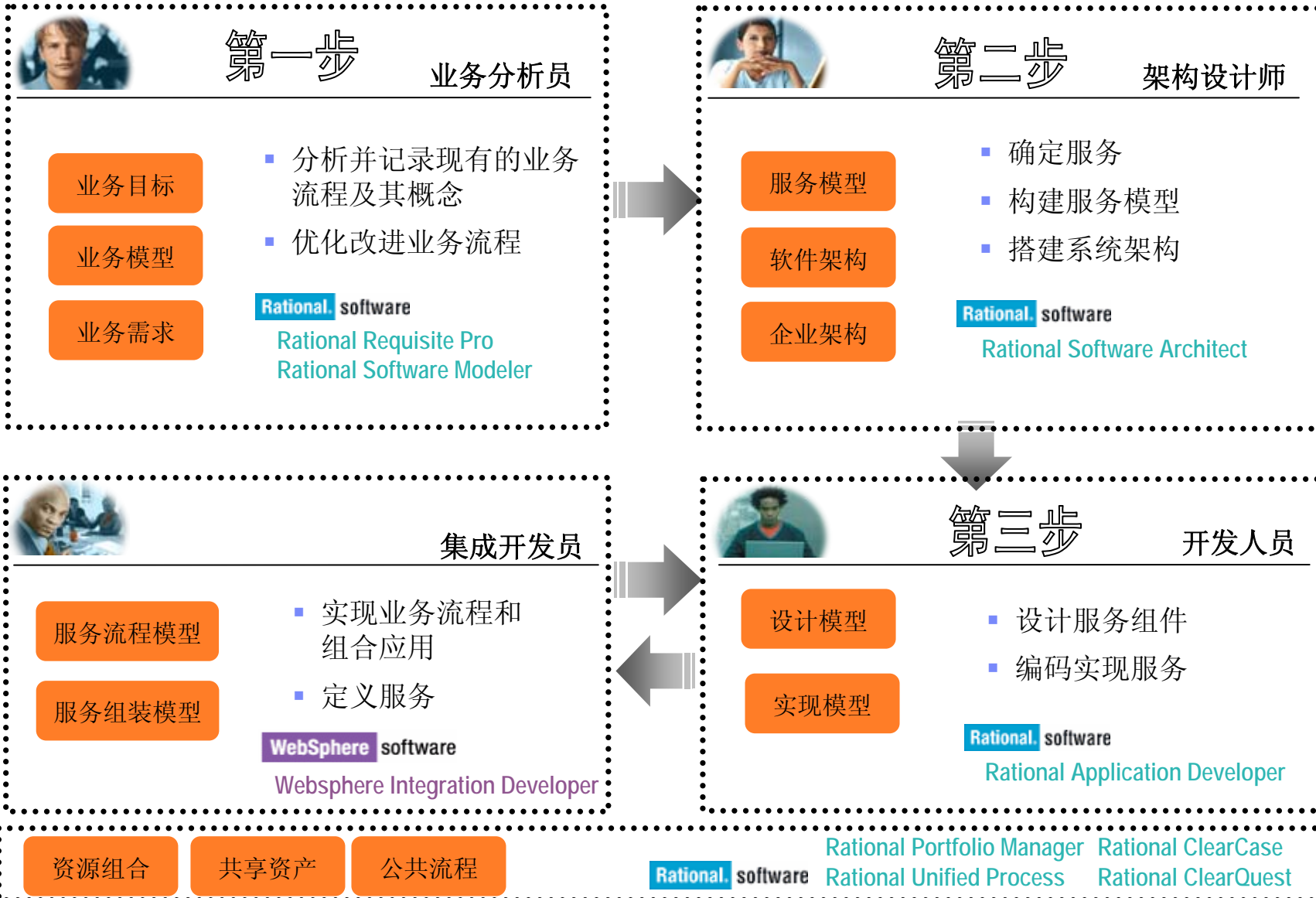
Tivoli Composite Application Manager
WebSphere Business Monitor



什么是 SOAD



SOA 开发三步曲

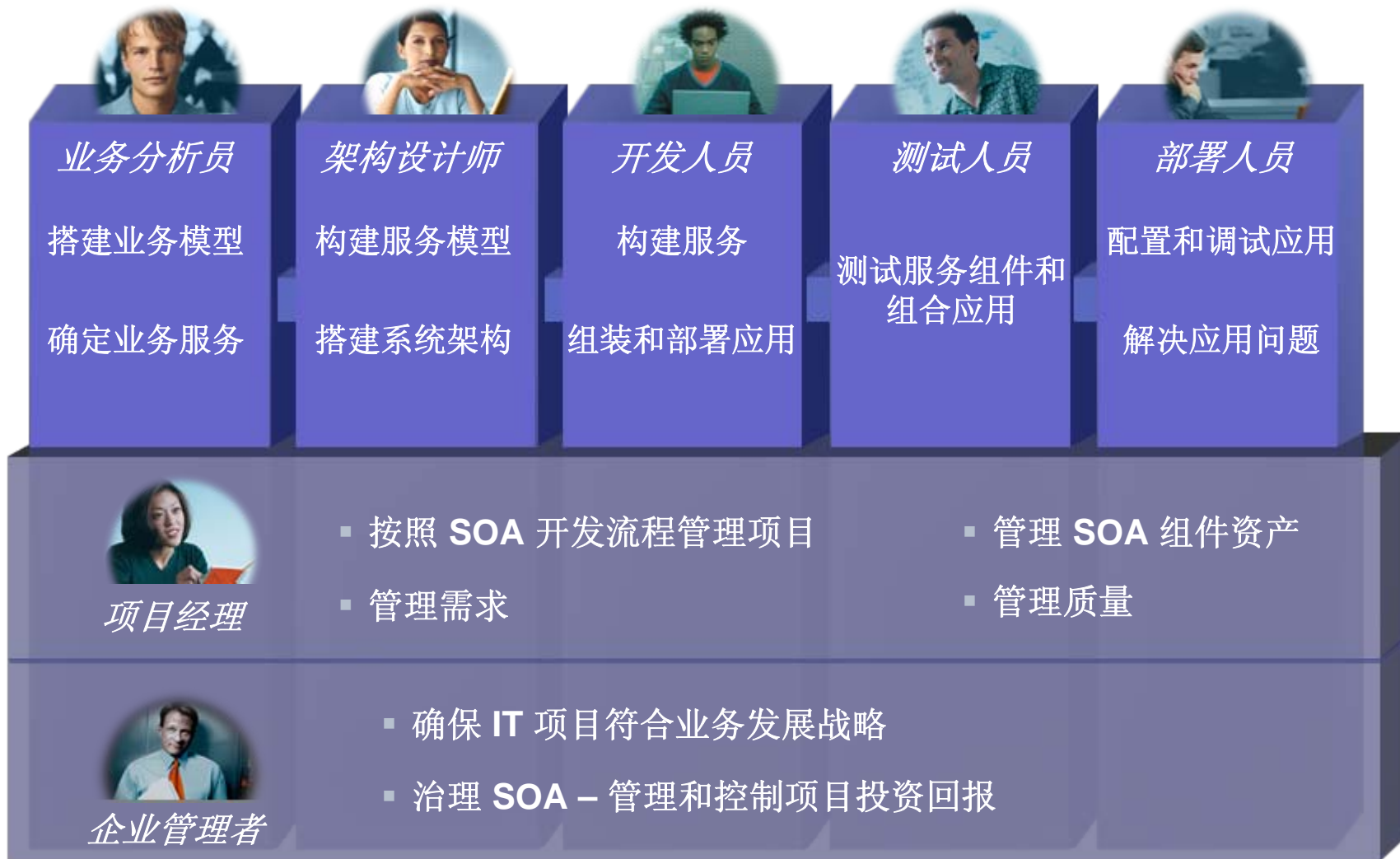


内容

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

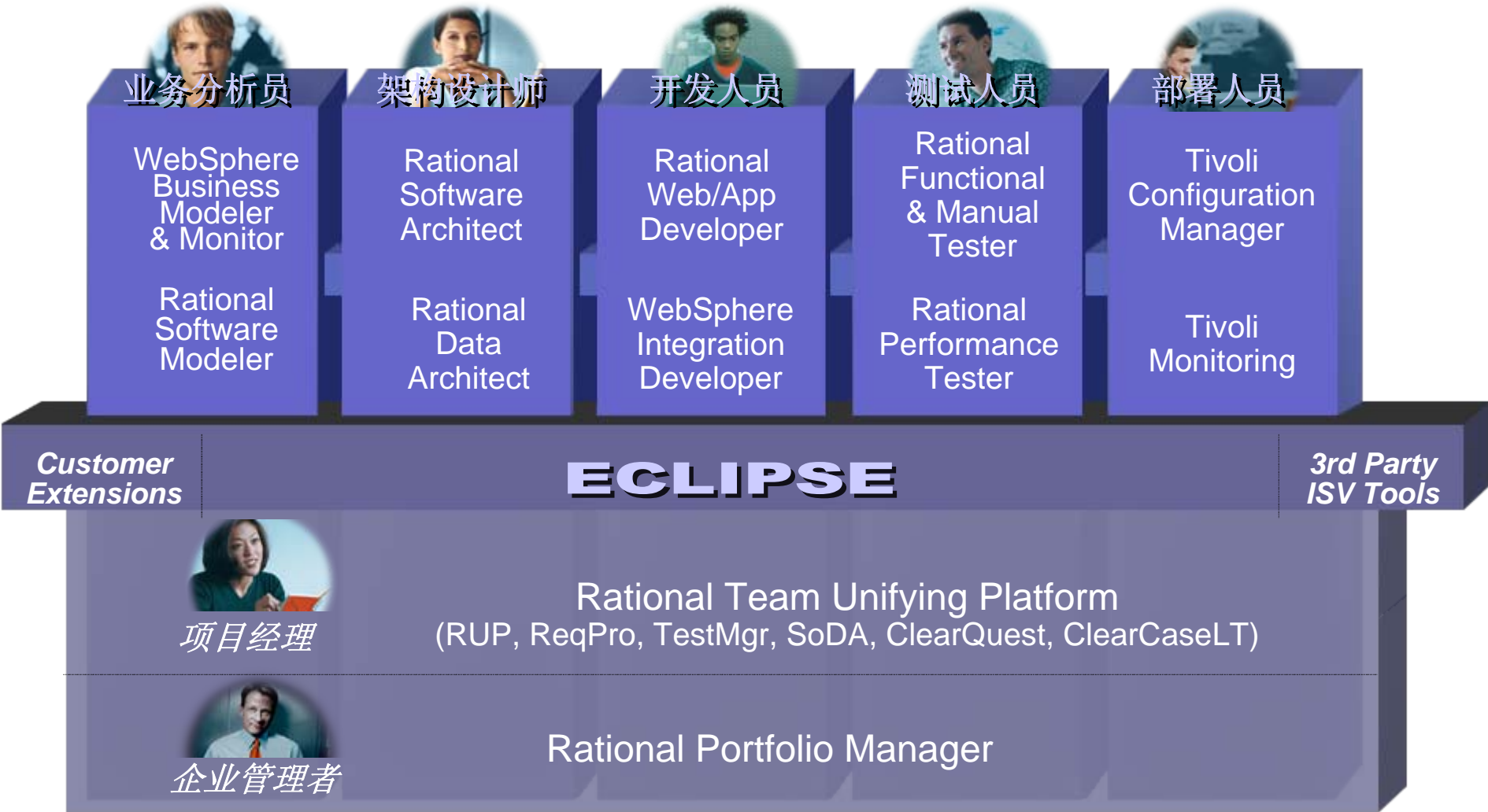


面向服务的设计和开发

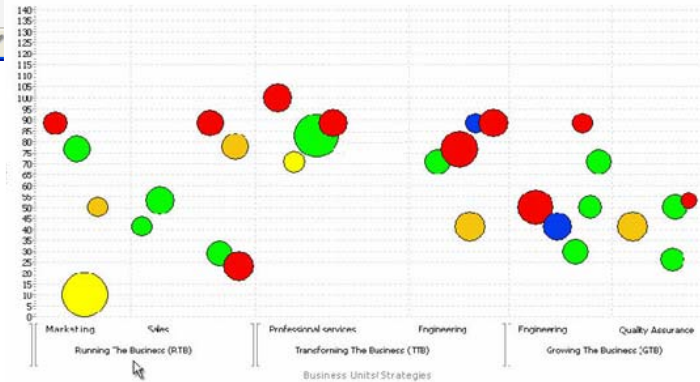
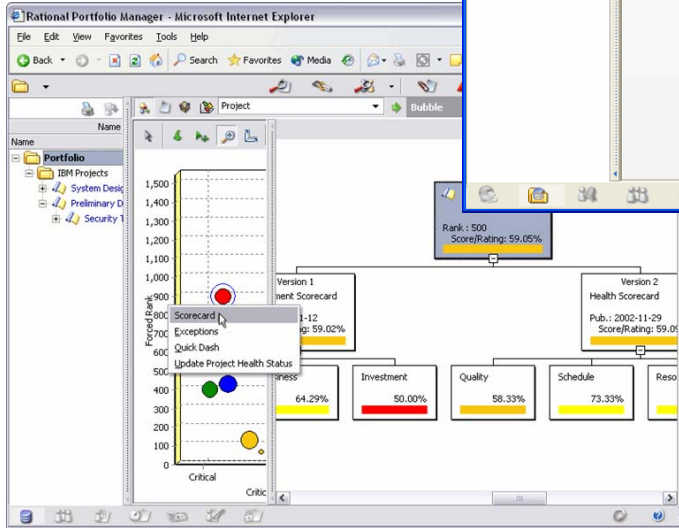
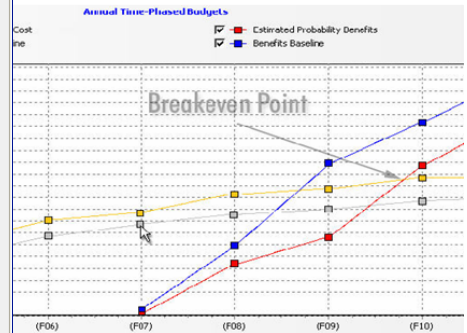
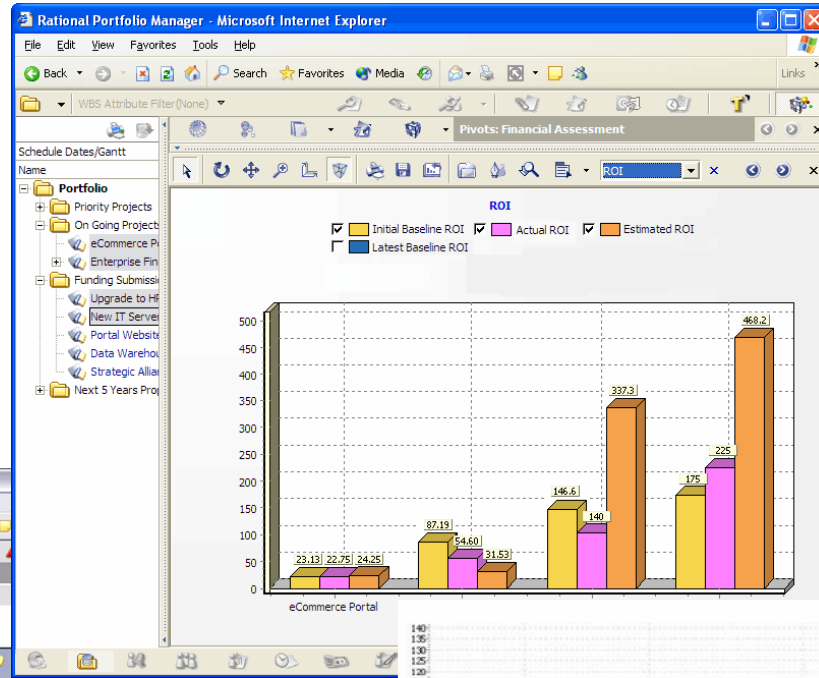
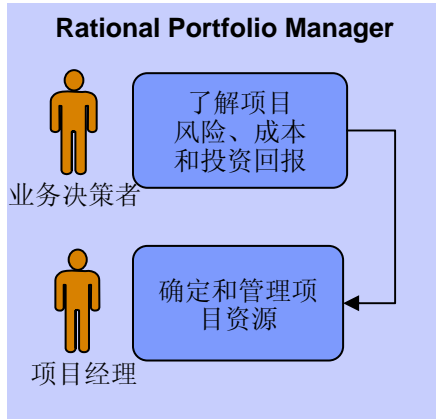


面向 SOA 的软件开发平台 SDP

IBM Rational Software Development Platform

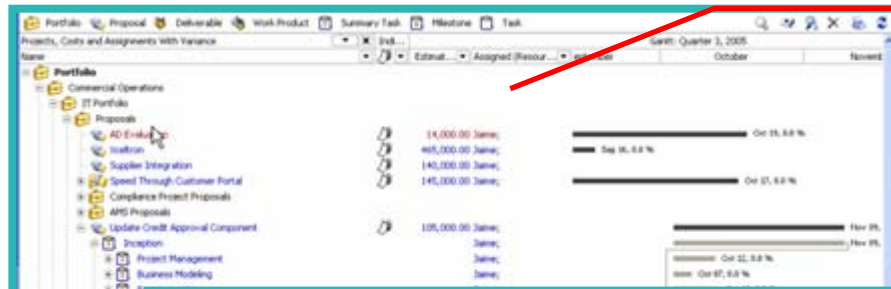
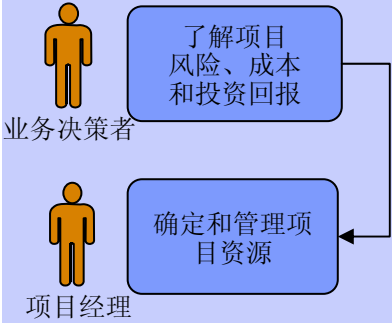


项目组合管理 – 控制和管理项目投资

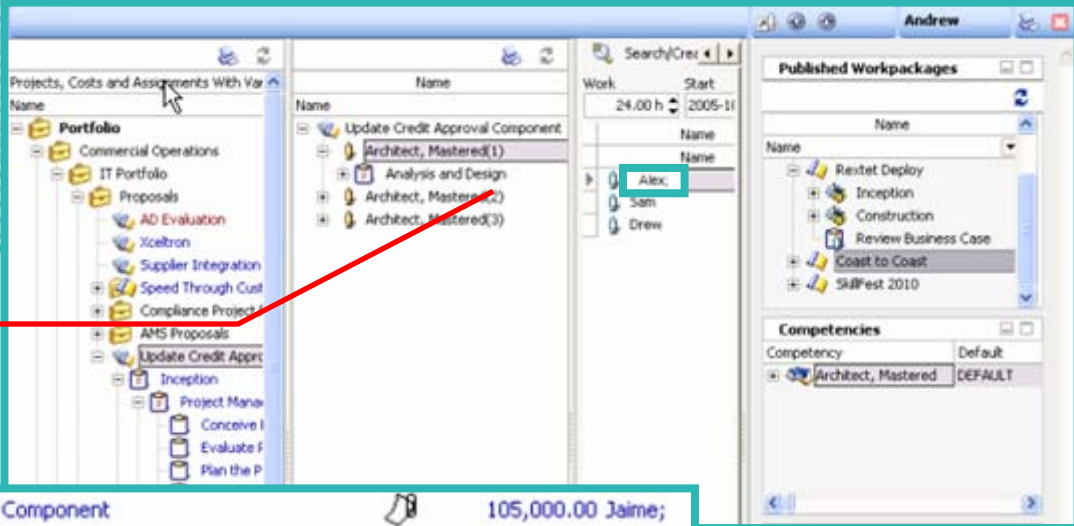


项目管理 - 计划、预算和分配资源

Rational Portfolio Manager



根据预先定义的标准模板来制定项目计划

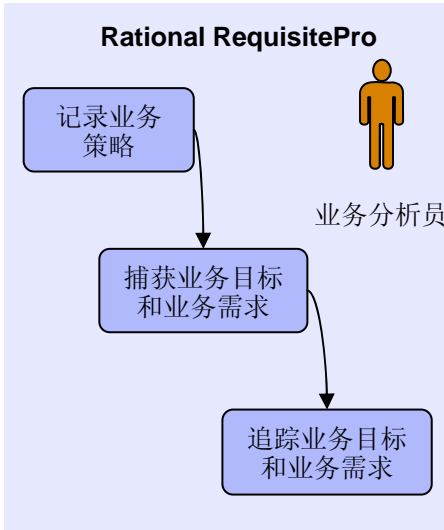


保证项目资源，提高资源利用率



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

Vision

Vision.VIS

1. Introduction

1.1 Purpose of the Vision Document

The purpose of this document is to define and to define high-level business requirements for the ClassicsCD.com Web Shop system. This system is intended to provide a new channel of sales for retail operation.

Rational RequisitePro - Learning Project - Use Cases - [FEAT: All Features]

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

Rational RequisitePro - Learning Project - Use Cases - [GOAL: Business Goals]

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

Rational RequisitePro - Learning Project - Use Cases - [RULE: Business Rules]

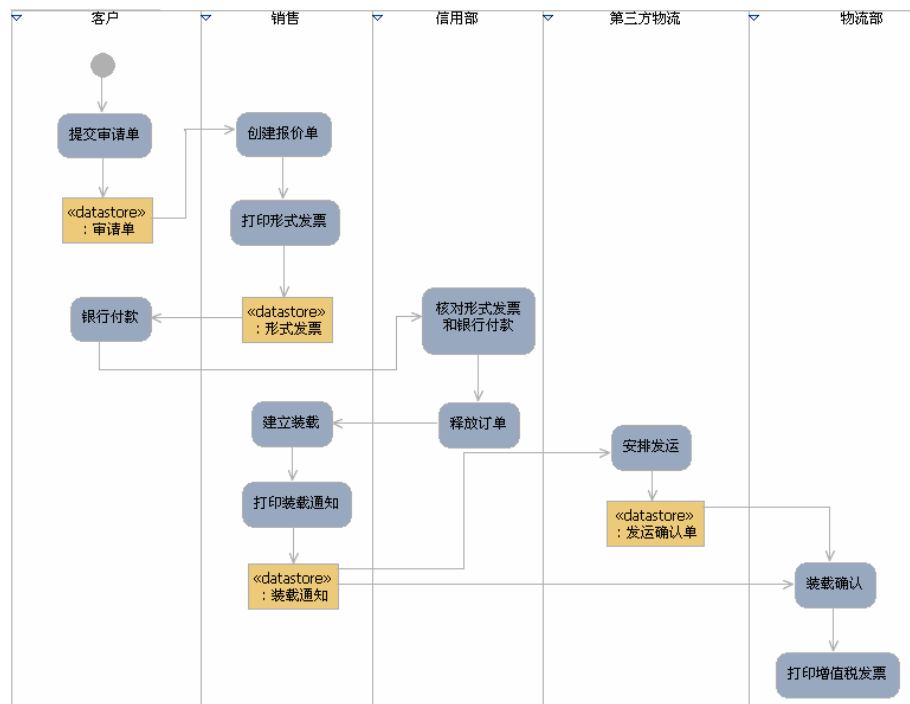
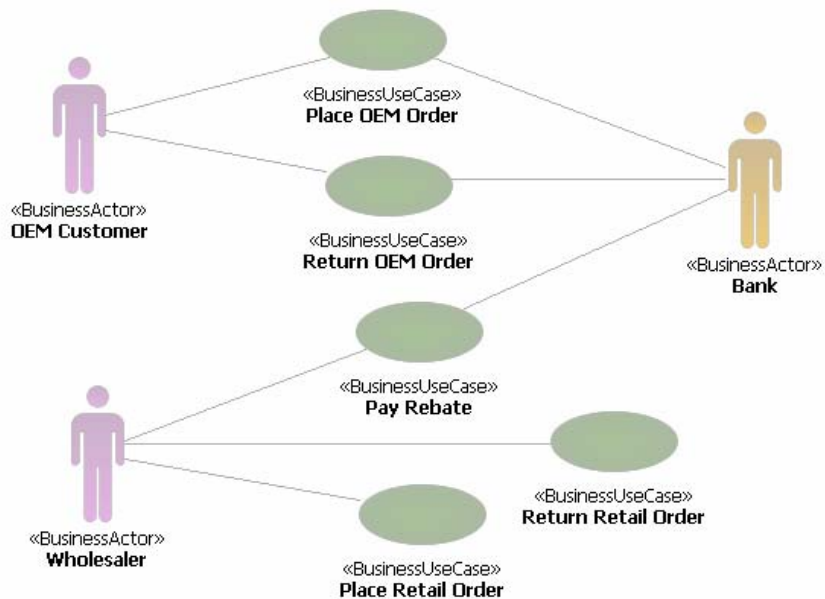
Requirements	Author	Revision	Reason
RULE1: Minimum of 3 CDs free shipping	Unknown	1.0001	Requirement Text C
RULE2: Maximum of 10 CDs, results in a 10% discount	Unknown	1.0001	Requirement Text C
RULE3: If the inventory # of a particular CDs falls below 3...	Unknown	1.0002	Requirement Text C
* <Click here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

RULE3: If the inventory # of a particular CDs falls below 3, then an automatic re-order shall happen

Requirement RULE3 modified

3 requirements

业务建模 - 了解业务流程，确定业务服务

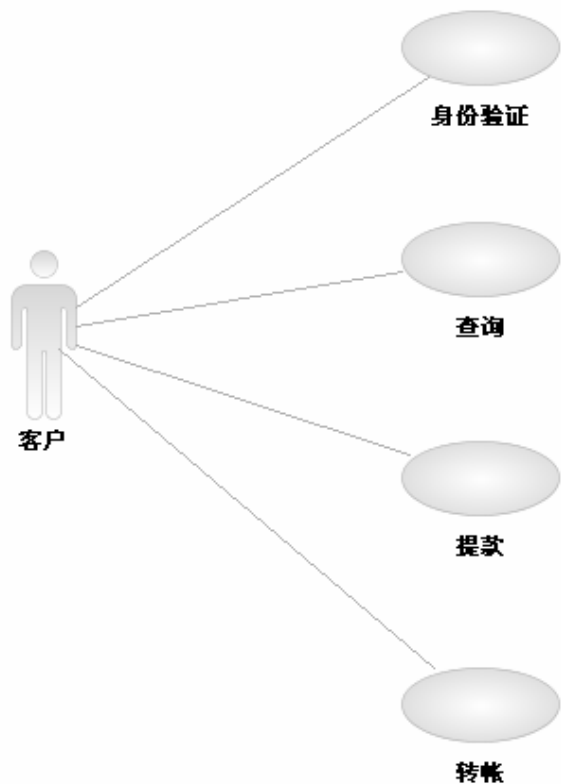
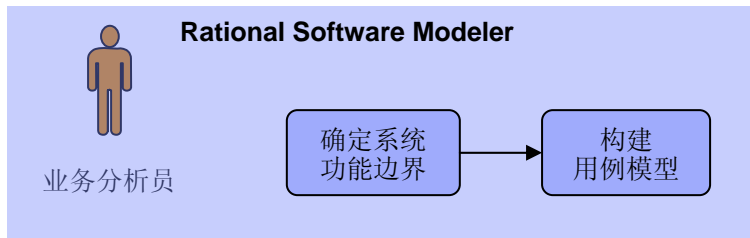


**Rational Software Modeler
UML Profile for Business Modeling**

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

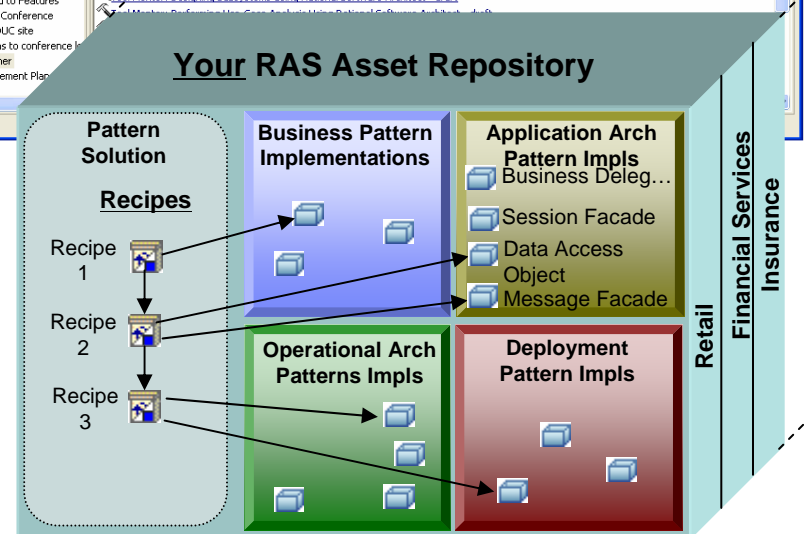
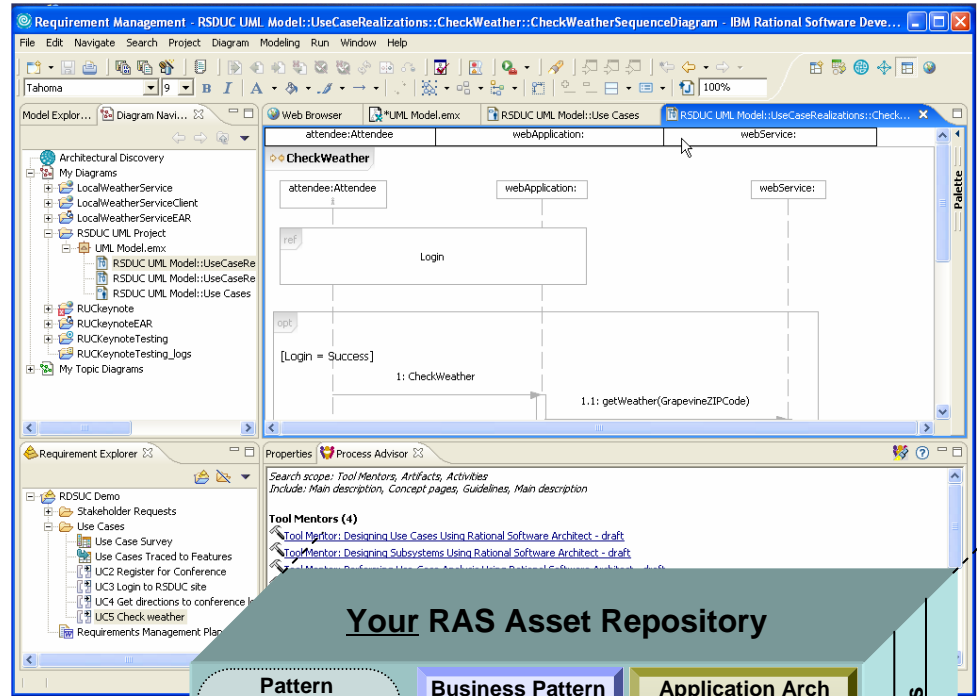
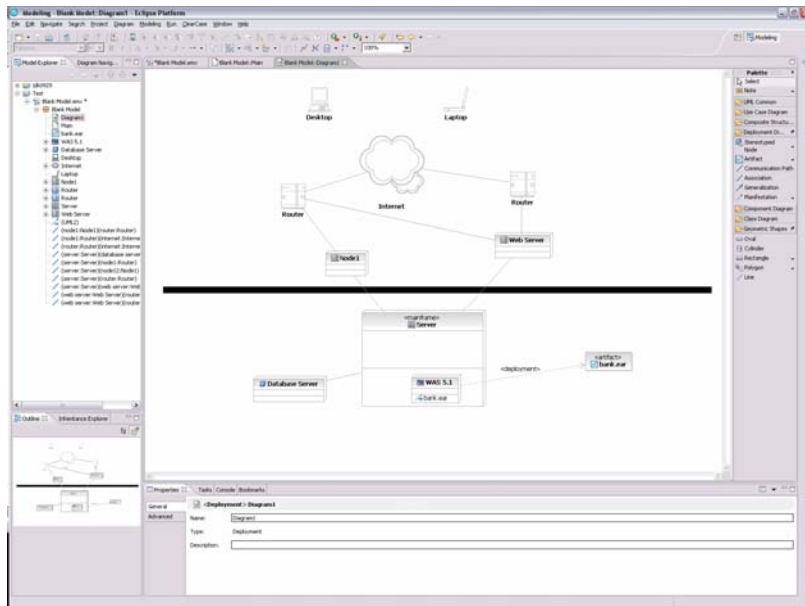
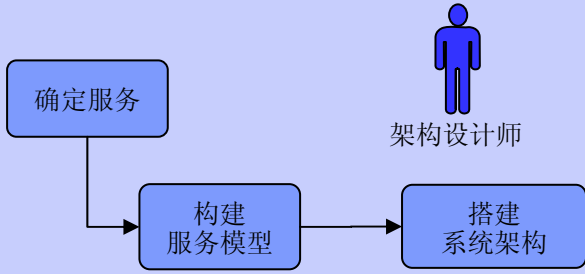
无效信用卡: 基本流, 无效信用卡

重新输入客户密码: 基本流, 客户密码不正确

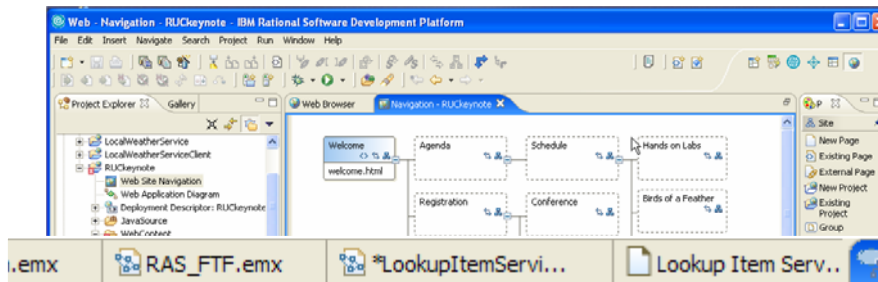
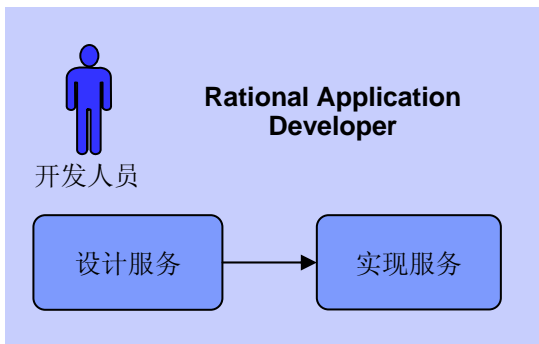
没收客户信用卡: 基本流, 客户密码不正确 (密码输错 3 次以上)

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



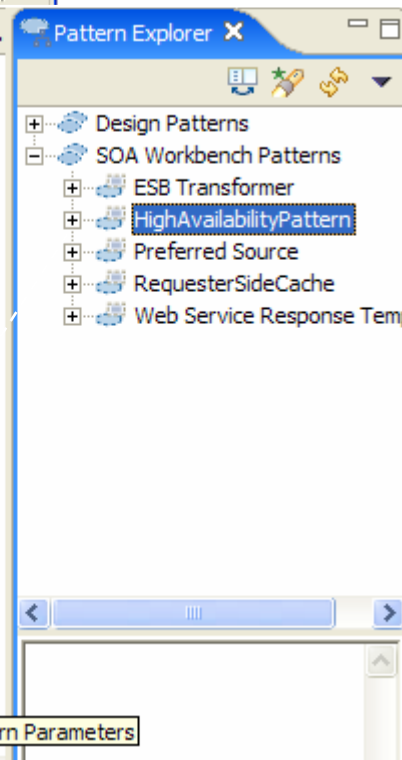
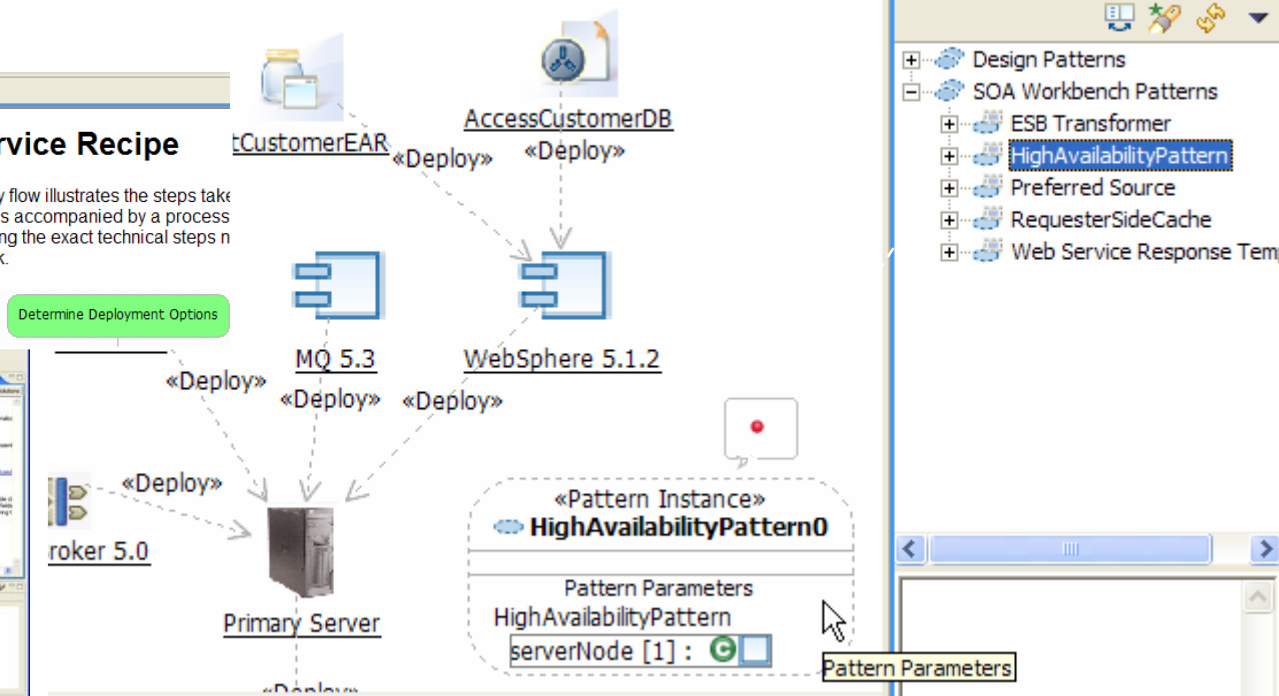
Properties Tasks Console Bookmarks **Solution Guide** Search

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.

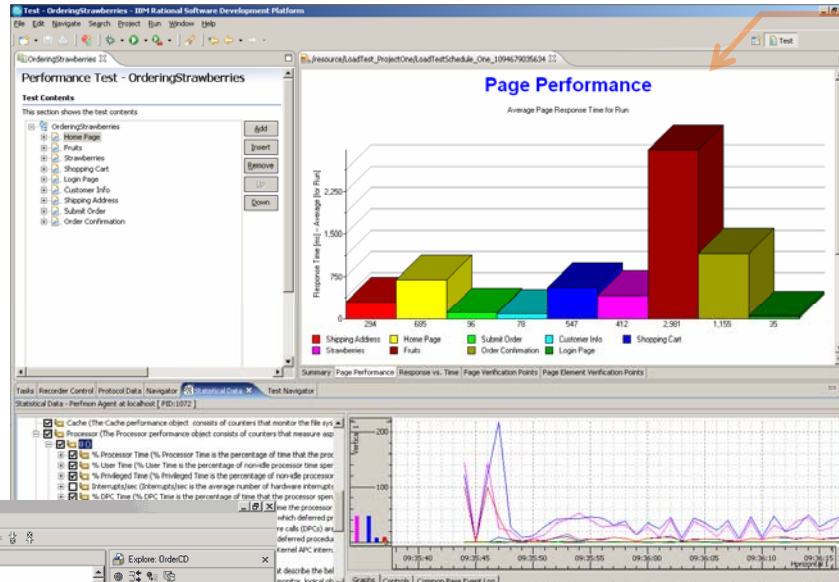


测试服务组件和组合应用

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
    
```

Launch Browser by clicking the icon shown on your Desktop:

- Enter Address http://
- Logon For Trading
- NewName
- Confirm that the
- Enter User Name
- Enter Password
- Select "Start in:"
- Does the "Submit"
- Does the "Remer
- Click "Submit"
- Logged in ok?
- Navigate to Trading
- Click "Investing"
- Click "Executions"
- Verify that trad
- Sell Mutual Fund
- Click "Mutual Fun

管理可重用的服务组件

Rational ClearCase
Rational ClearQuest

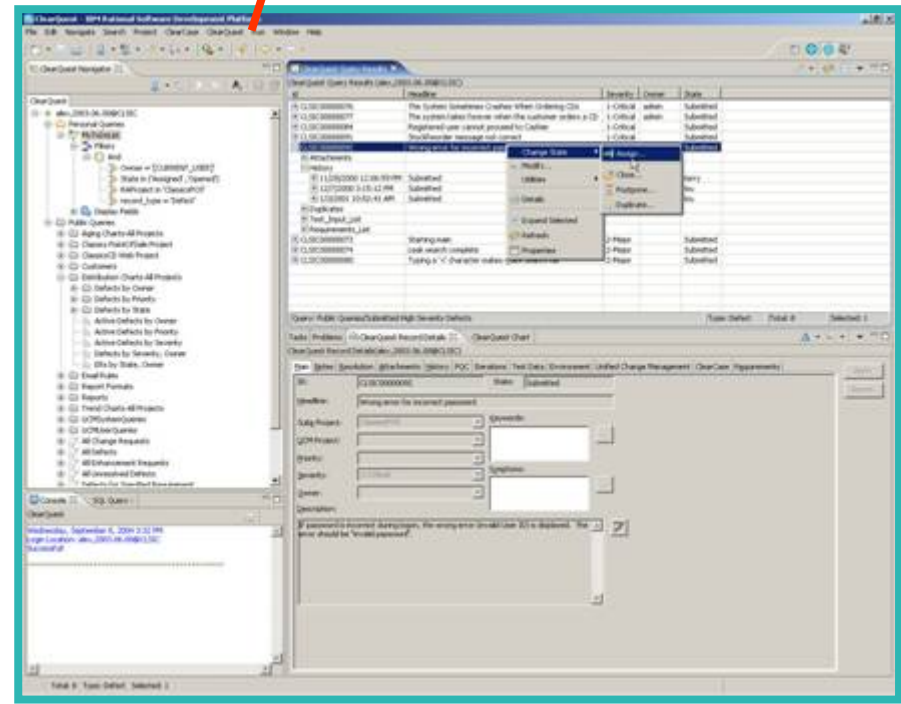
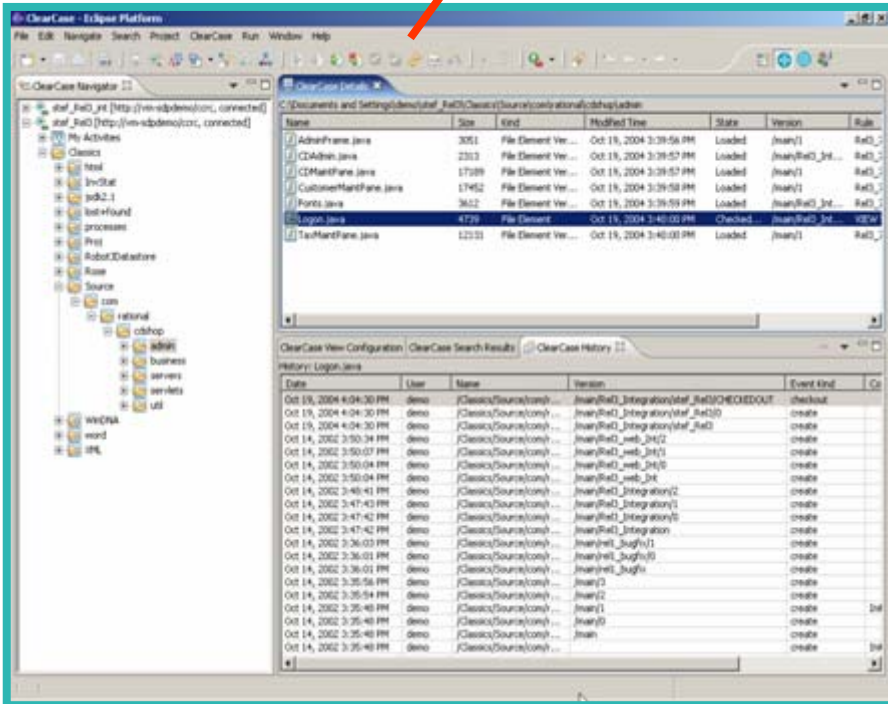
管理可重用
服务资产



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) software interface within a Mozilla Firefox browser window. The main content area is titled "Workflow Detail: Design Services" and includes a "Design services brief description." section. A central diagram illustrates the workflow: a "Designer" (represented by a person icon) leads to "Service Design" (represented by a document icon), which then branches into two parallel paths. The top path involves "Service Specification" (document icon) leading to "Service" (gear icon) and "Service Model" (document icon). The bottom path involves "Service Specification" (document icon) leading to "Service" (gear icon) and "Design Model" (document icon). A "Topics" sidebar on the left lists links for "Description", "Related Information", "Timing", and "How to Staff". 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".

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

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

IBM在SOA领域的领导地位

- IBM在SOA领域的领先优势
 - ▶ IBM是SOA相关技术标准和规范的主要制定者和坚定执行者
 - ▶ IBM拥有完整的SOA技术体系架构
 - ▶ IBM拥有先进的SOA实施方法论
 - ▶ IBM可以提供SOA分析咨询和技术服务
 - ▶ IBM软件产品全线支持SOA
- IBM SOA设计中心
 - ▶ IBM在全球设立了四个SOA设计中心，帮助客户进行SOA的流程设计以及实施
 - ▶ 在中国北京、上海的两个SOA设计中心将帮助重要合作伙伴和客户进行SOA架构设计、SOA评估
- IBM SOA社区建设
 - ▶ SOA人才培养计划：100/1000/10000
 - ▶ SOA社区



总结

- SOA将给企业IT架构带来一场意义深远的变更!
- SOA能够极大推动中国企业实现业务按需应变的进程!
- SOA对中国企业是一个非常好的机遇!
 - ▶ 做领导者还是追随者
 - ▶ 不必重复别人的老路
 - ▶ 企业希望复用的是业务
 - ▶ 避免一落地就是“遗留系统”



Thank
YOU

