IBM Rational 软件创新论坛

IBM

Let's build a
smarter planet.

开发有道
创新"智"造

Innovate2010

敏捷的模型驱动系统开发
——构筑高质量嵌入式软件产品

姚冬

dongyao@cn.ibm.com

Innovate2010

Agenda

- 嵌入式系统开发现状
- MDD模型驱动开发与Harmony
- Harmony与敏捷Agile
- Rational解决方案

嵌入式系统软件开发的机遇和挑战

2010年，平均每人拥有10亿个晶体管，每个晶体管的平均价值为百万分之一分钱.
先进的科技已经融入数亿计的设备中 --- 汽车，交通，仪器等

2011年，全球约有20亿人通过各种各样的设备接入到英特网

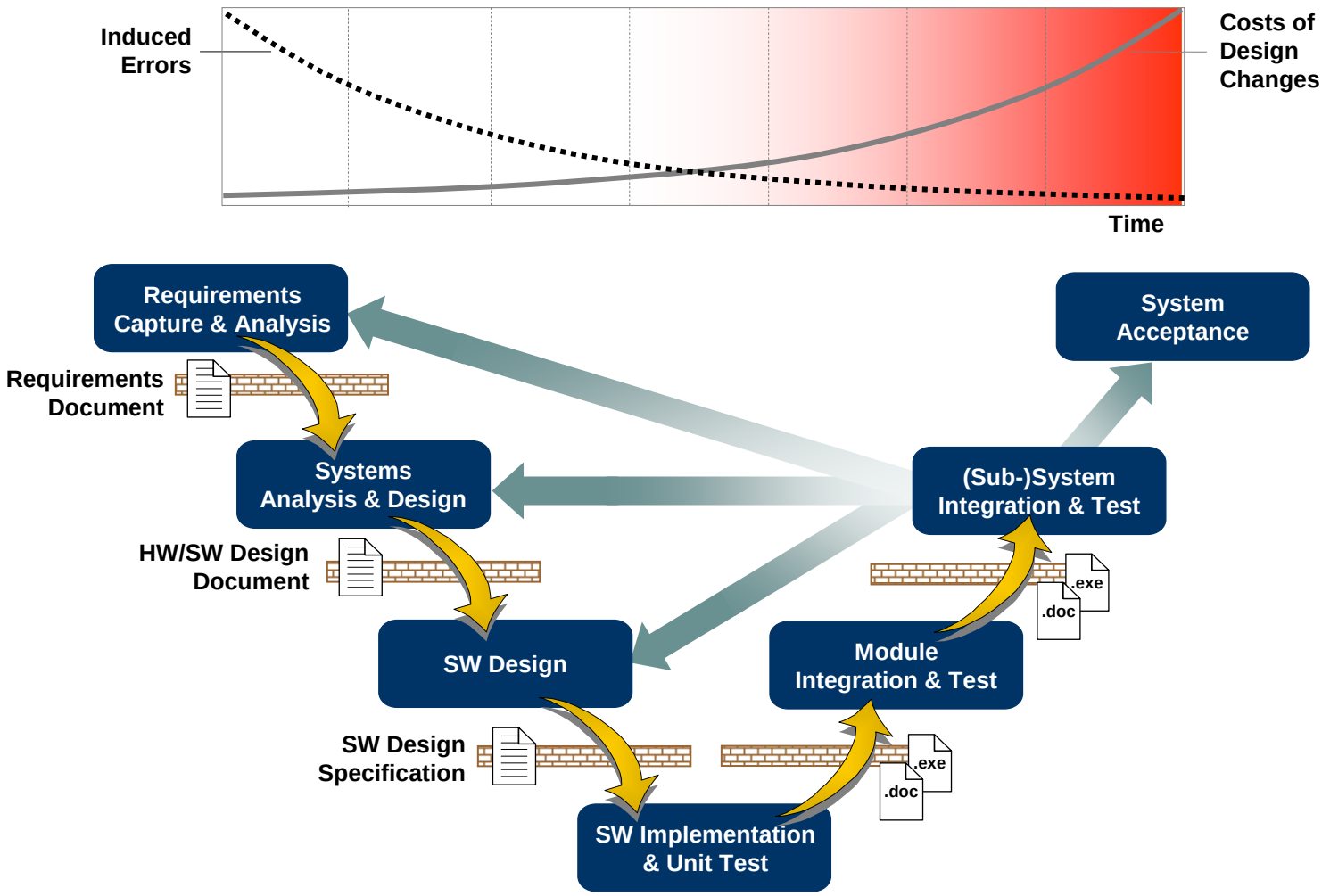
90%的汽车中的创新来自于电子设备

80%的创新来自于嵌入式软件

嵌入式系统软件开发的现状

- 文档驱动的开发
- 需求管理难度大
  - 需求描述不清晰，理解存在二义性
  - 需求变更影响难以追踪
  - 需求覆盖难以统计
- 目标硬件与运行环境在项目早期通常无法就位
- 系统复杂，团队协作存在问题
- 难以适应变更
- 设计、代码、文档一致性不好，与客户沟通存在问题

文档驱动的开发 – 实际的

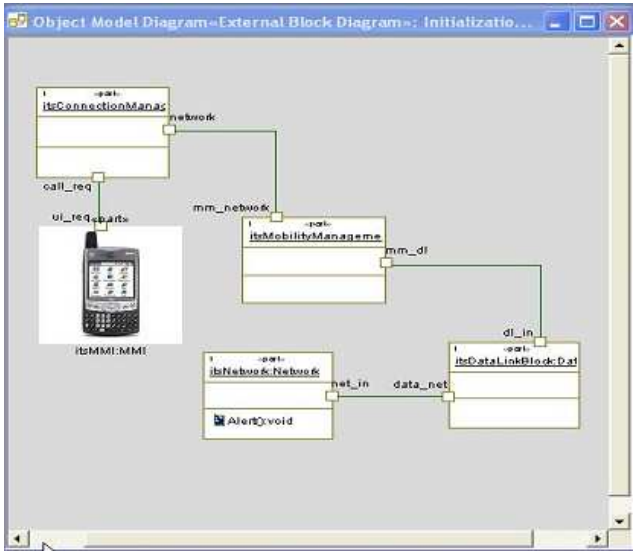

Induced Errors
Costs of Design Changes
Time
Requirements Capture & Analysis
Requirements Document
Systems Analysis & Design
HW/SW Design Document
SW Design
SW Design Specification
SW Implementation & Unit Test
Module Integration & Test
(Sub-)System Integration & Test
System Acceptance
.exe
.doc
.exe
.doc

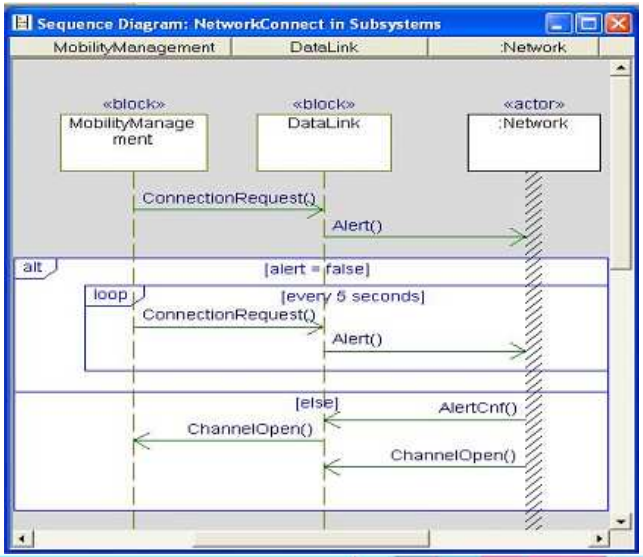
Agenda

- 嵌入式系统开发现状
- **MDD模型驱动开发与Harmony**
- Harmony与敏捷Agile
- Rational解决方案

模型——UML 标准

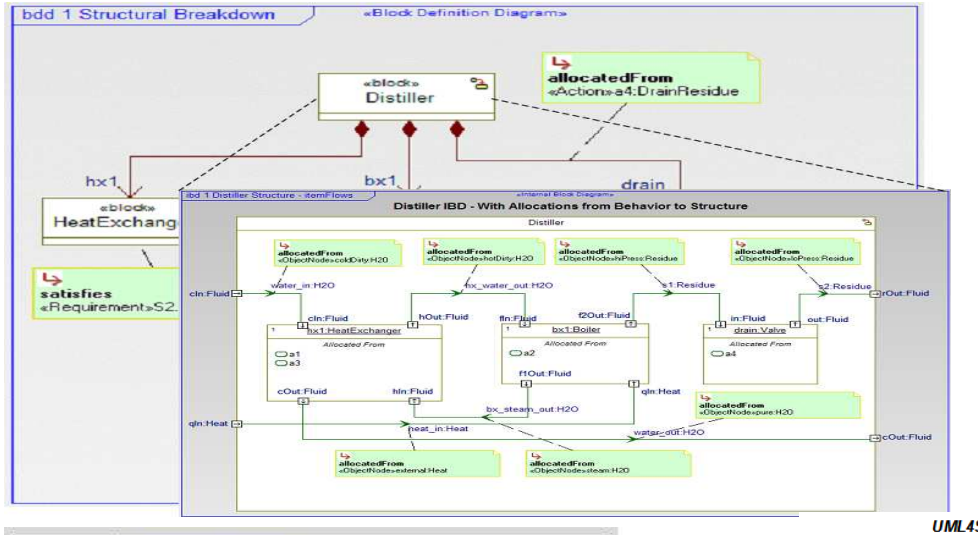
- 结构
  - Structure Diagram
  - Package Diagram
  - Component Diagram
  - Object Model Diagram
    - Class and Object
  - Deployment Diagram
- 行为
  - StateCharts
  - Activity Diagram
  - Use case Diagram
- 交互
  - Sequence Diagram

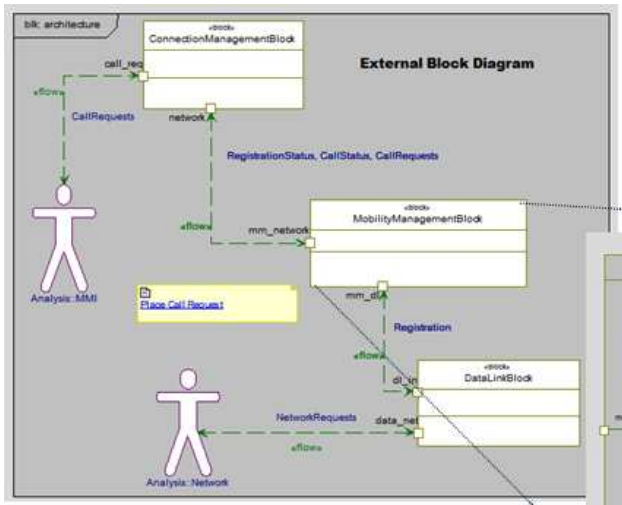


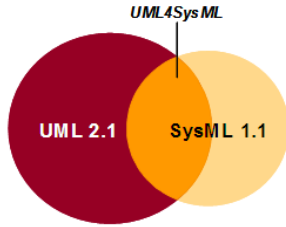


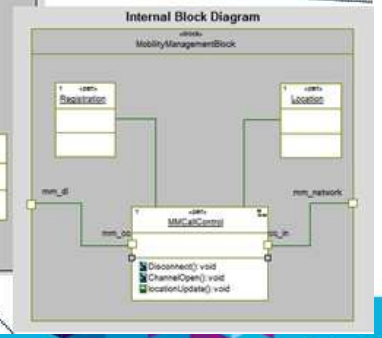
模型——SysML - System Modeling Language

- SysML 模型
  - 需求
    - Requirement Diagram
    - Use case Diagram
  - 结构
    - External block Diagram
    - Internal block Diagram
  - 行为
    - StateCharts
    - Activity Diagram
    - Sequence Diagram
  - 约束
    - Parametric Diagram



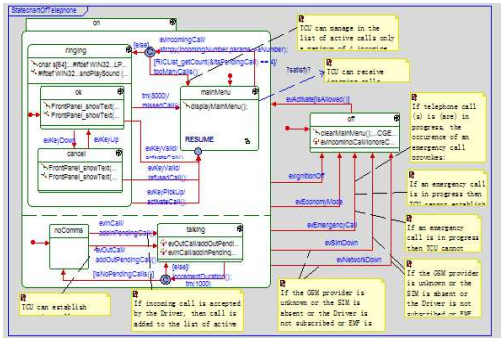


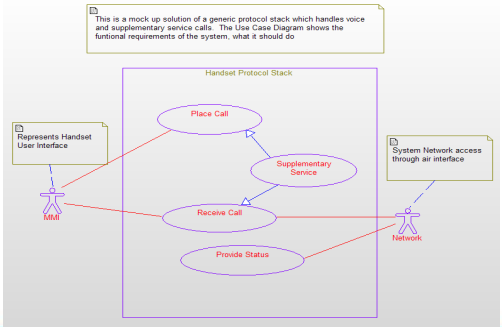


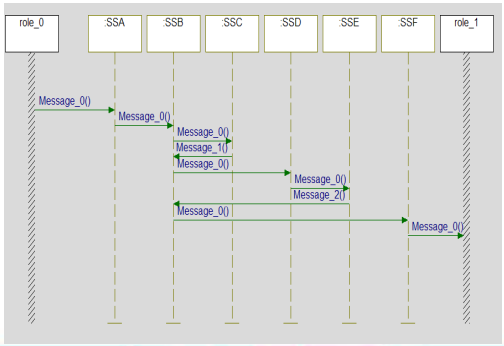


模型驱动的系统软件开发

- 模型驱动的系统软件开发是一种可视化的开发方法，形式化地理解并实现需求的内容
- 专注于具体领域的建模和抽象，而不是程序语言和算法实现。
- 从不同层面，不同角度，通过模型对复杂系统进行抽象和描述
- 迭代与增量的开发与测试，通过快速原型验证设计
- 改善文档驱动的设计方法带来的弊病, 优点在于:
  - 更好的团队沟通和合作
  - 精准设计和设计重用
  - 需求描述和追踪
  - 实现系统架构和功能的对接
  - 接口描述

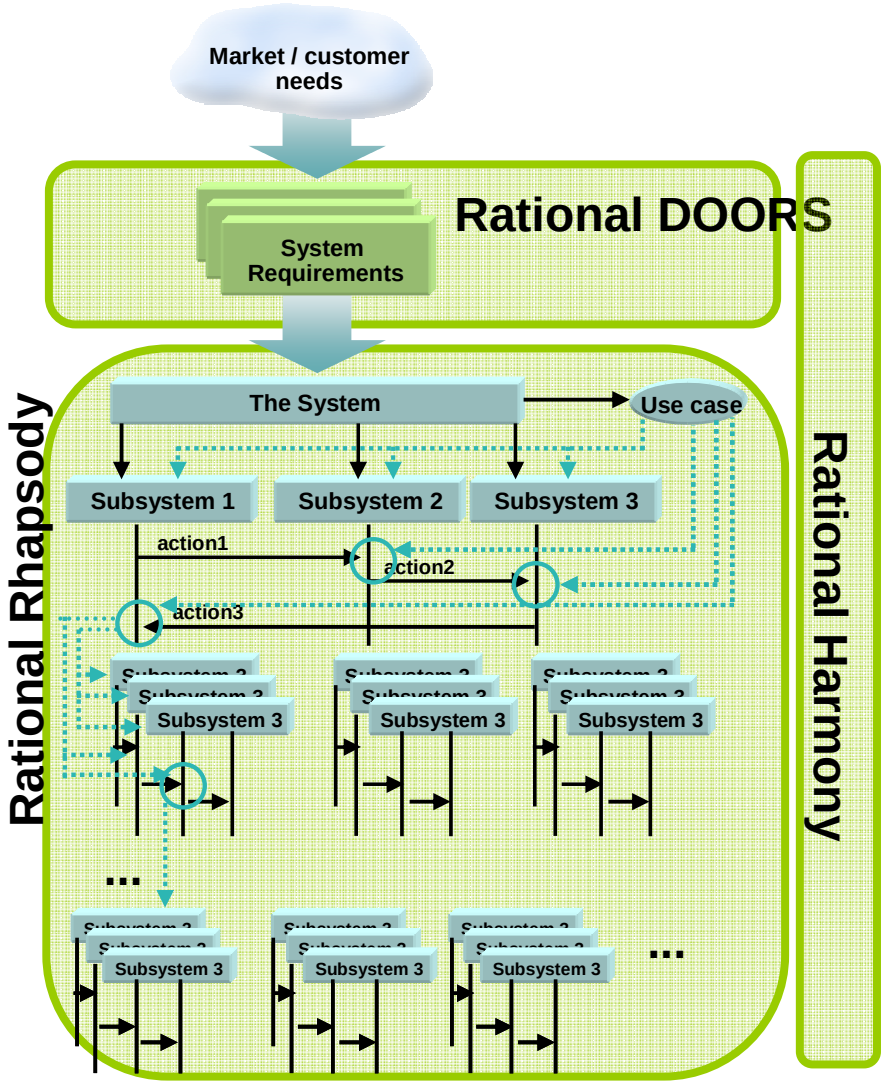






模型驱动的系统开发——Rational方案

- UML、SysML，保证设计遵循行业标准和安全规范
- 可视化的需求、系统和软件建模
- 通过早期系统原型找出最佳系统设计方案
- 建立模型和需求的追踪关系
- 模型级别的仿真，保证系统设计缺陷在早期被发现
- 自动代码生成保证了开发效率和质量
- 快速响应用户需求或系统架构的变化，保证端到端的追踪
- 增加团队的协作性
- 系统开发最佳实践－Harmony



模型驱动的系统开发的最佳实践——Harmony

Hans-Peter Hoffmann

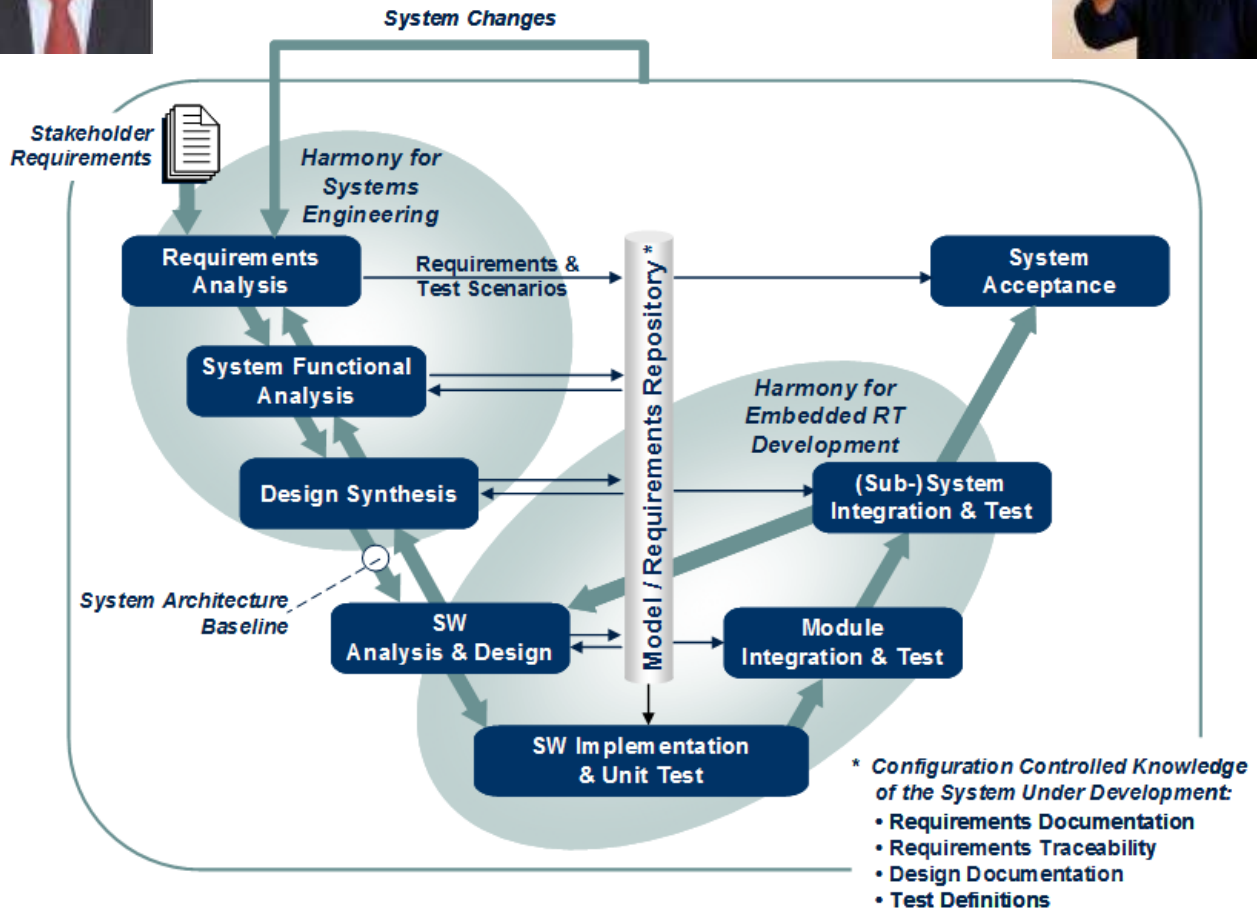
Bruce Powel Douglass

1.系统工程

Harmony /SE

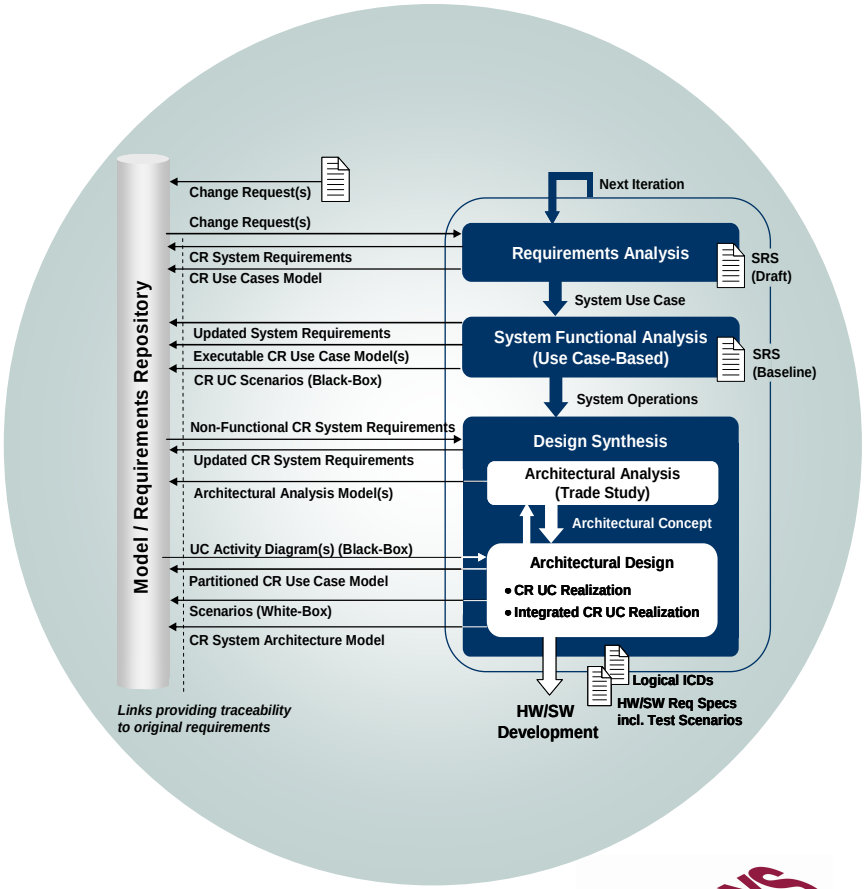
2.嵌入式软件开发

Harmony/ESW



Harmony/SE -变更驱动的系统设计

- 基于模型的系统工程流程
- 基于SysML
- 迭代的过程
- 用例驱动需求捕获与分析
- 通过可执行模型验证需求
- 系统设计过程中全程追踪需求
- 执行场景 转化为测试用例
- 核心步骤:
  - 需求分析(Requirements Analysis)
  - 系统功能分析(System Functional Analysis)
  - 架构分析和设计 (architectural analysis and design)



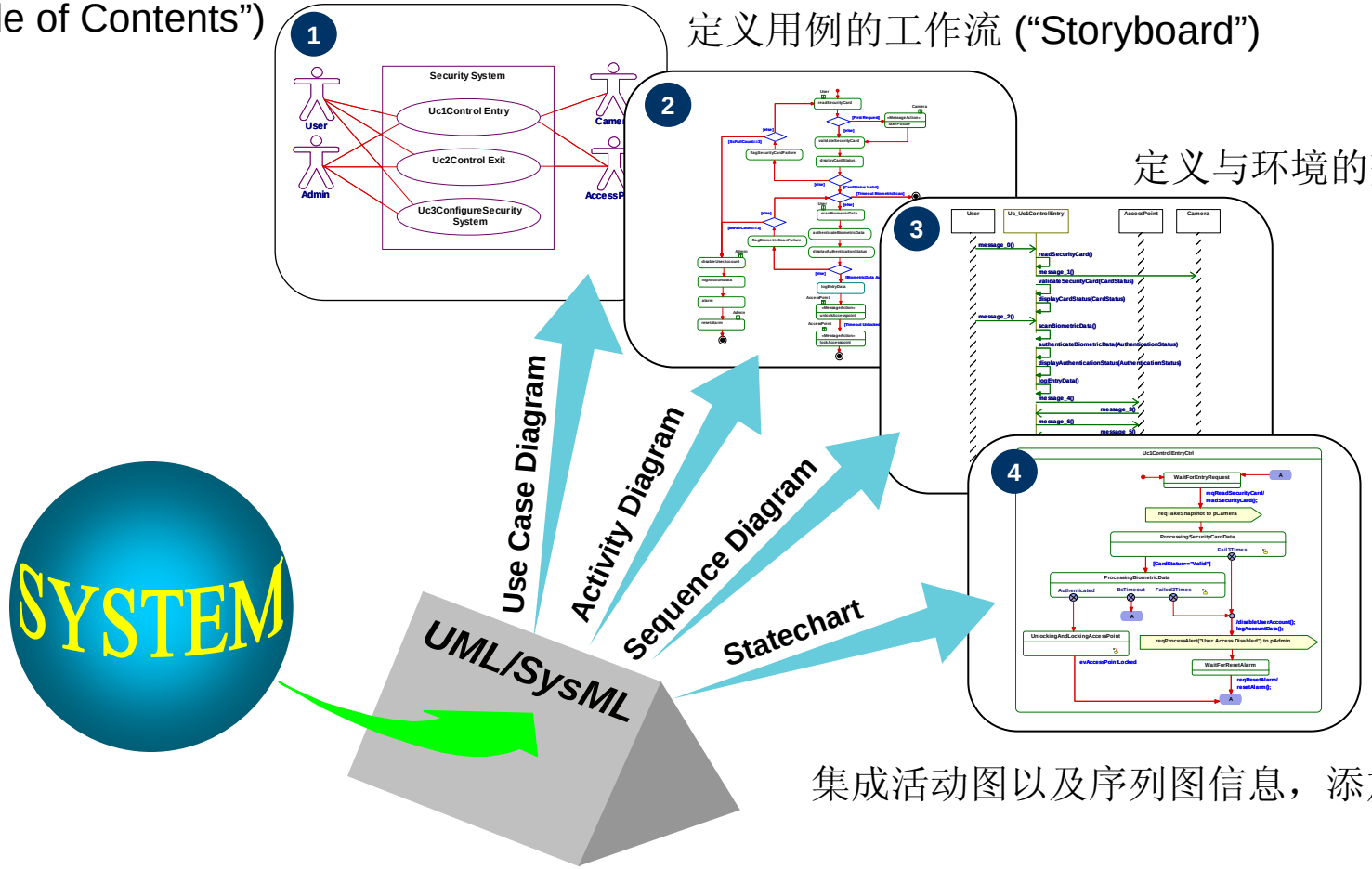
SysML 工件在Harmony/SE中的应用

捕获系统行为

定义系统边界，范围，关联需求至用例
("Table of Contents")

定义用例的工作流 ("Storyboard")

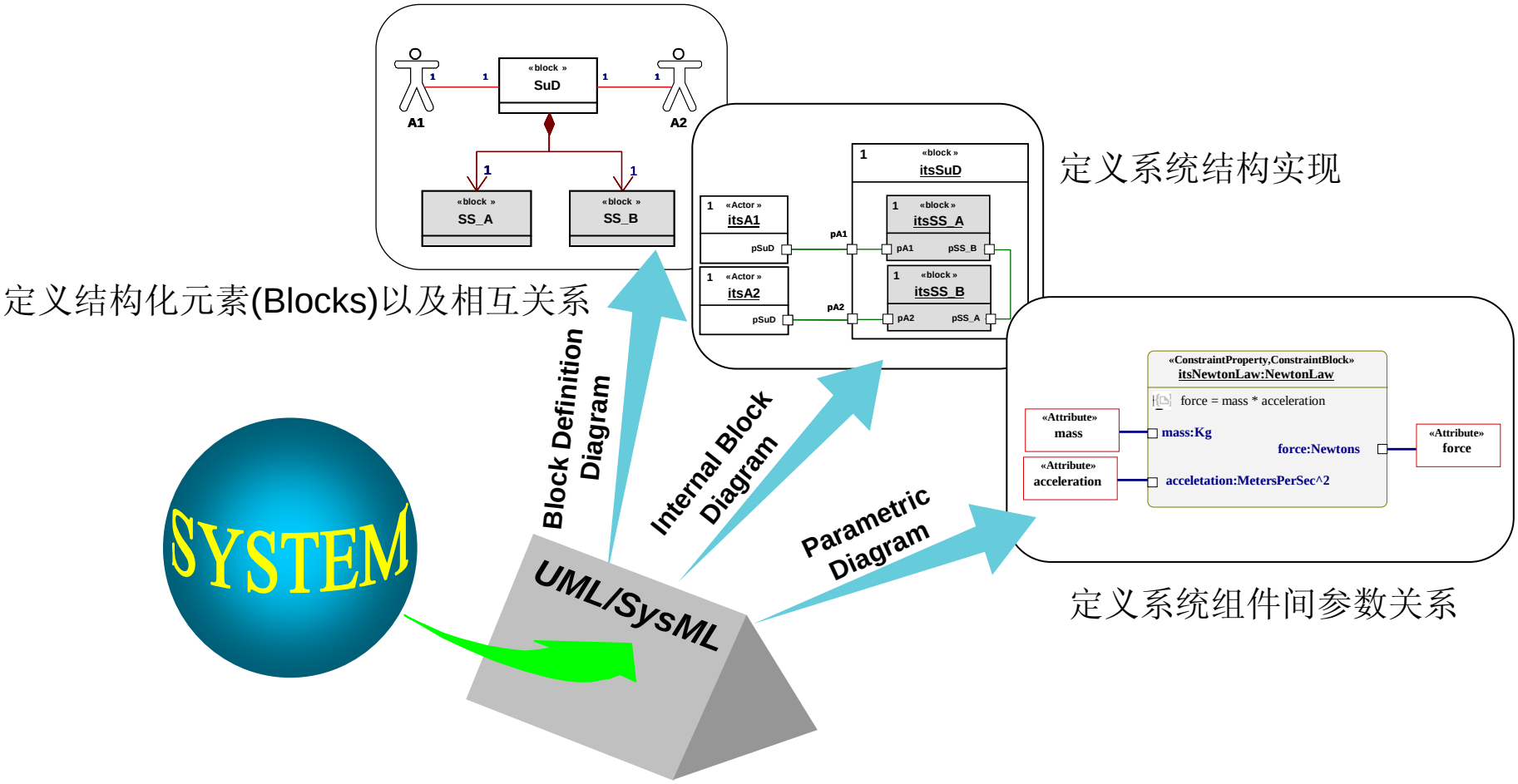
定义与环境的交互



集成活动图以及序列图信息，添加状态行为

SysML 工件在Harmony/SE中的应用

捕获静态视图



Harmony/ESW

- 螺旋、迭代的开发流程
- 需求驱动
- 以构架为中心
- 专为实时和嵌入式软件
- 软硬件协同开发
- 提供了嵌入式软件开发最佳实践
  - 嵌入式架构
  - 设计模式
  - MDA架构
  - 安全关键系统分析和开发

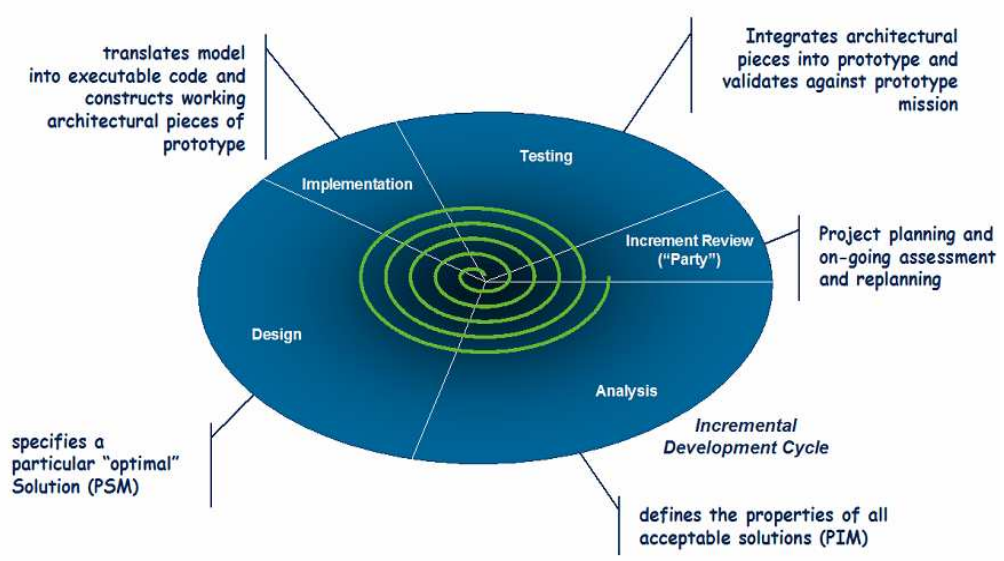
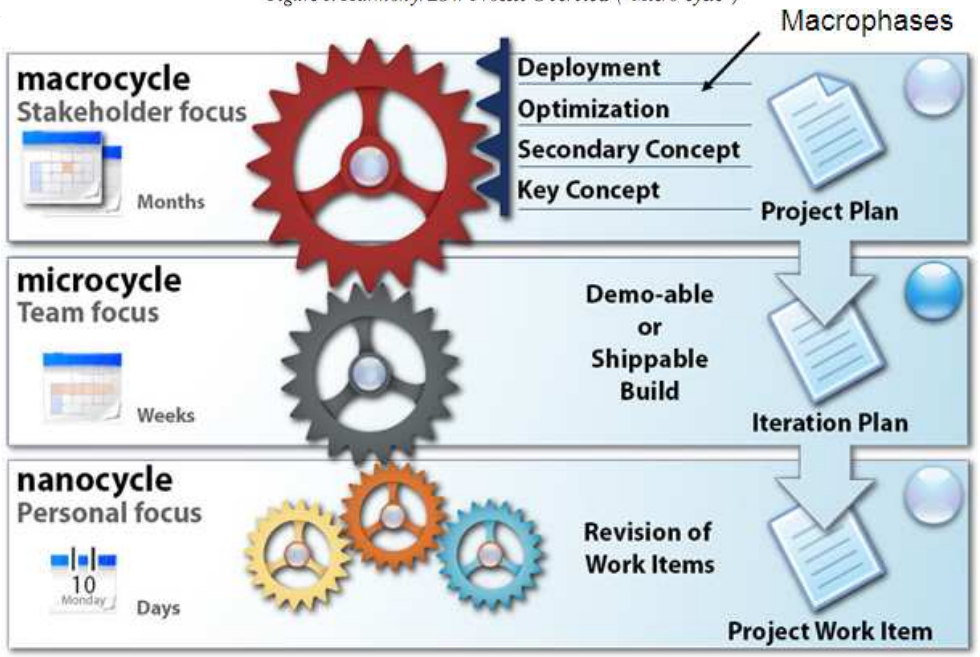


Figure 1. Harmony/ESW Process Overview ("Micro-cycle")



Harmony/ESW

- Harmony ESW是一个敏捷过程，遵循一些基本的核心原则
  - 可工作的软件为核心和评估标准
  - 模型驱动
  - 使用UML捕获应用行为与结构
  - 计划，追踪与调整
  - 持续反馈，重视质量及优化
  - 持续的代码生成
  - 持续的调试与测试
  - 持续集成

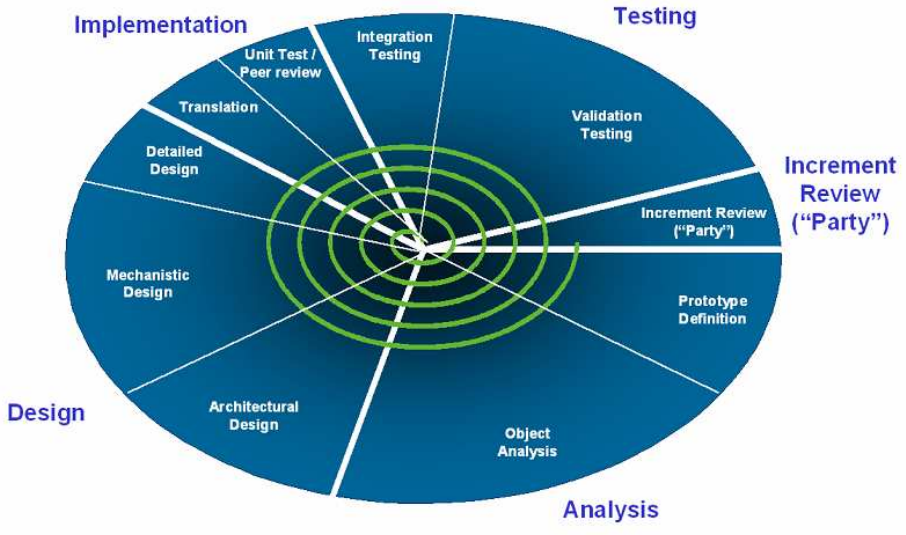


Figure 3. Software spiral ("Micro-cycle") in Harmony/ESW

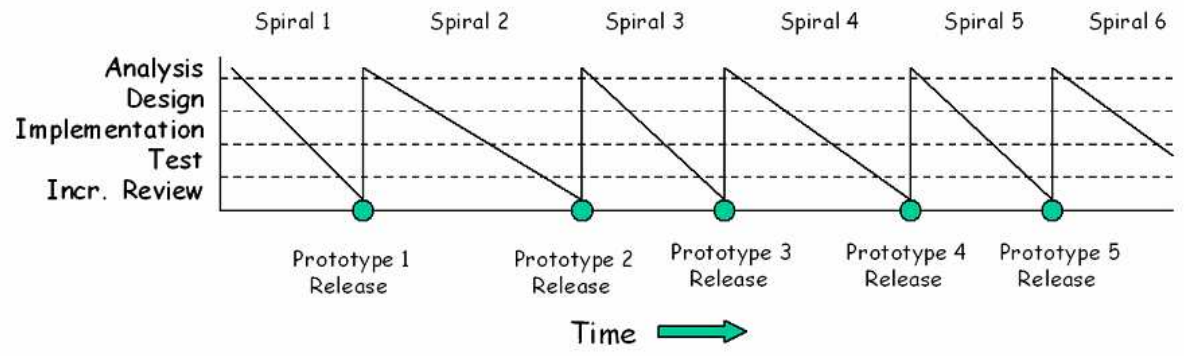


Figure 2. "Unrolled" micro-cycles

Agenda

- 嵌入式系统开发现状
- MDD模型驱动开发与Harmony
- **Harmony与敏捷Agile**
- Rational解决方案

敏捷开发解决产品开发挑战

主要挑战	传统敏捷开发技术	模型驱动/Harmony与敏捷的结合
需求不断变化，难以确定	▪客户的参与 ▪需求、设计以及原型的结合 ▪团队协作 ▪迭代演进	▪**基于模型的系统工程** ▪构架可执行的需求与构架模型 ▪基于模型交付成果 ▪**基于模型的嵌入式软件工程** ▪可维护的设计 ▪自动生成代码，与设计保持一致 ▪支持基于模型的测试以及测试自动化 ▪**敏捷工程最佳实践** ▪持续质量反馈 ▪测试驱动的开发 ▪持续验证过程与结果 ▪基于现状动态调整计划
如何协调产品开发时间与成本要求	▪定义需求优先级 ▪增量开发高优先级任务 ▪持续发布 ▪更小更专注更集中的团队 ▪精简的过程与文档	
与新技术新科技的配合	▪系统构架师与开发人员互动 ▪在项目早期验证高风险任务 ▪持续测试Daily Testing	

辨正的看待敏捷宣言

个体和交互 胜过 过程和工具

- *不代表：你可以忽略使用模型以及协作设计工具*

工具有助于提速，模型相比起没有说明文档的代码而言，更直观易于理解

可以工作的软件 胜过 面面俱到的文档

- *不代表：你可以忽略需求捕获及对应的设计*

模型与文档对理解系统非常重要，提供敏捷所依赖的深层了解

客户合作 胜过 合同谈判

不代表：你可以忽视合同、限制及规范

响应变化 胜过 遵循计划

不代表：你可以不作计划

有计划不代表我们不能改变，模型帮助预测系统行为，提高变更对系统影响的分析精度，增量的模型开发有助于提供快速响应。

Harmony 敏捷指导原则

- 首要目标：开发可工作的软件
- 持续反馈至关重要
- 计划、追踪与调整
- 模型至关重要
- 忽略风险是项目失败最大敌人
- 持续关注质量必不可少
- 以目标持续验证过程
- 足够胜于多余
- 测试驱动开发

- Harmony的理念与过程与Agile/Scrum存在一致性

原则：首要目标: 开发可工作的软件

- 所有不能直接帮助开发可工作的软件的活动，都一定是次要的
- 实践: 每日活动应该关注在
  - 理解现在应该做什么
  - 尽快运行工件
  - 对每一个小的增量进行验证，确保在做正确的事
  - 避免浪费时间在对运行没有贡献的任务上
- 但是，文档以及其他任务同样不可或缺，只是并非首要目标

原则：持续反馈至关重要

- “It ain’t right if it don’t run” – Law of Douglass
- “Optimism is the enemy of realism” – Law of Douglass
- “Optimism is a disease – feedback is the cure” – Kent Beck

- 瀑布模型一个关键前提是可以从一个阶段到另一个阶段而不引入显著错误
  - 我们现在都知道这是**不对的!**

- 实践: 我们以天、周和月为单位开发软件，我们需要确保在做正确的事情：
  - 持续的执行不断提供反馈（至少软件是可以执行的）
  - 执行结果与预期进行比较，保证设计与实际保持一致
  - 非正式的调试，鉴定并修改设计或实现
  - 正式的测试保证预期与实际的一致性
  - 一天或许需要多次运行验证

原则：计划、追踪与调整

- "The more you know, ***the more you know!"*** - Law of Douglass
- "Plan to Replan" – Law of Douglass

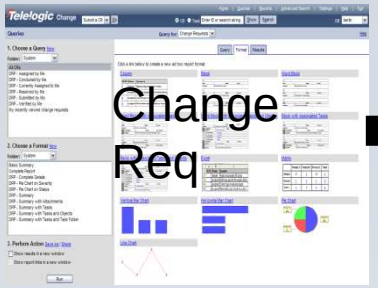
- 太多的管理者使用"弹道型计划"
  - 制定计划，祈祷几年以后项目将按预期的开支，达到预期的目标。
  - 不去跟踪实际进展，发现问题并采取适当调整
- 敏捷的一个关键原则是"动态计划"
  - 计划制定时往往无法获取完整信息
  - 随项目推进，理解更加深入
  - 变化总是存在
    - 需求
    - 环境
    - 能力
  - 计划提供蓝图，随着理解的深入及时更新，并随时调整以应对变化

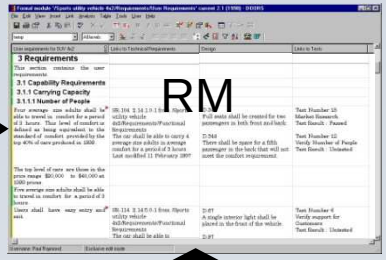
Agenda

- 嵌入式系统开发现状
- MDD模型驱动开发与Harmony
- Harmony与敏捷Agile
- **Rational解决方案**

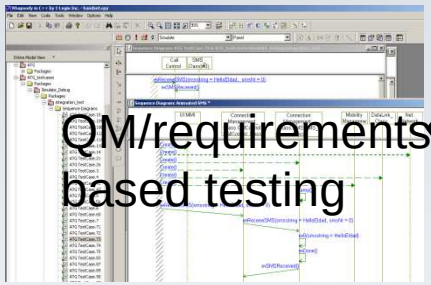
MDD在ALM中的集成

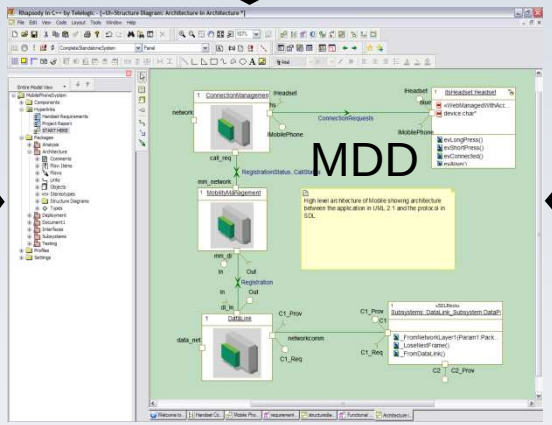
有效的团队协作

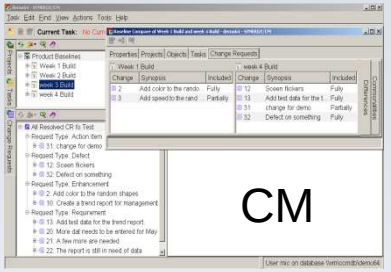


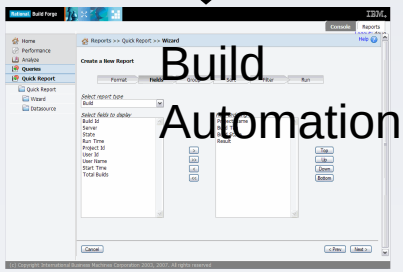


- 快速响应变更需求，通过可视化变更影响分析
- 通过模型驱动测试，自动验证修改
- 围绕模型驱动保持团队同步

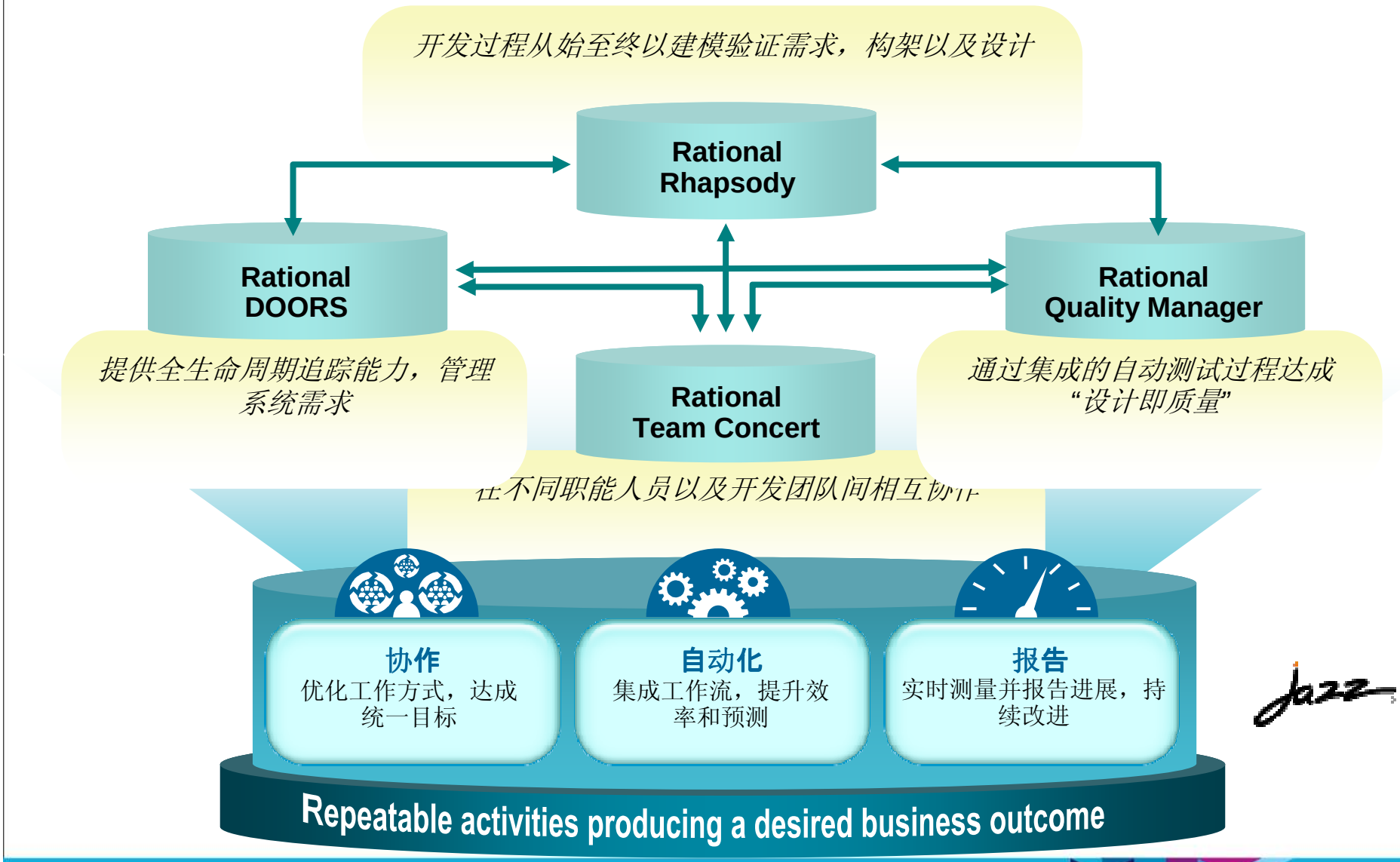






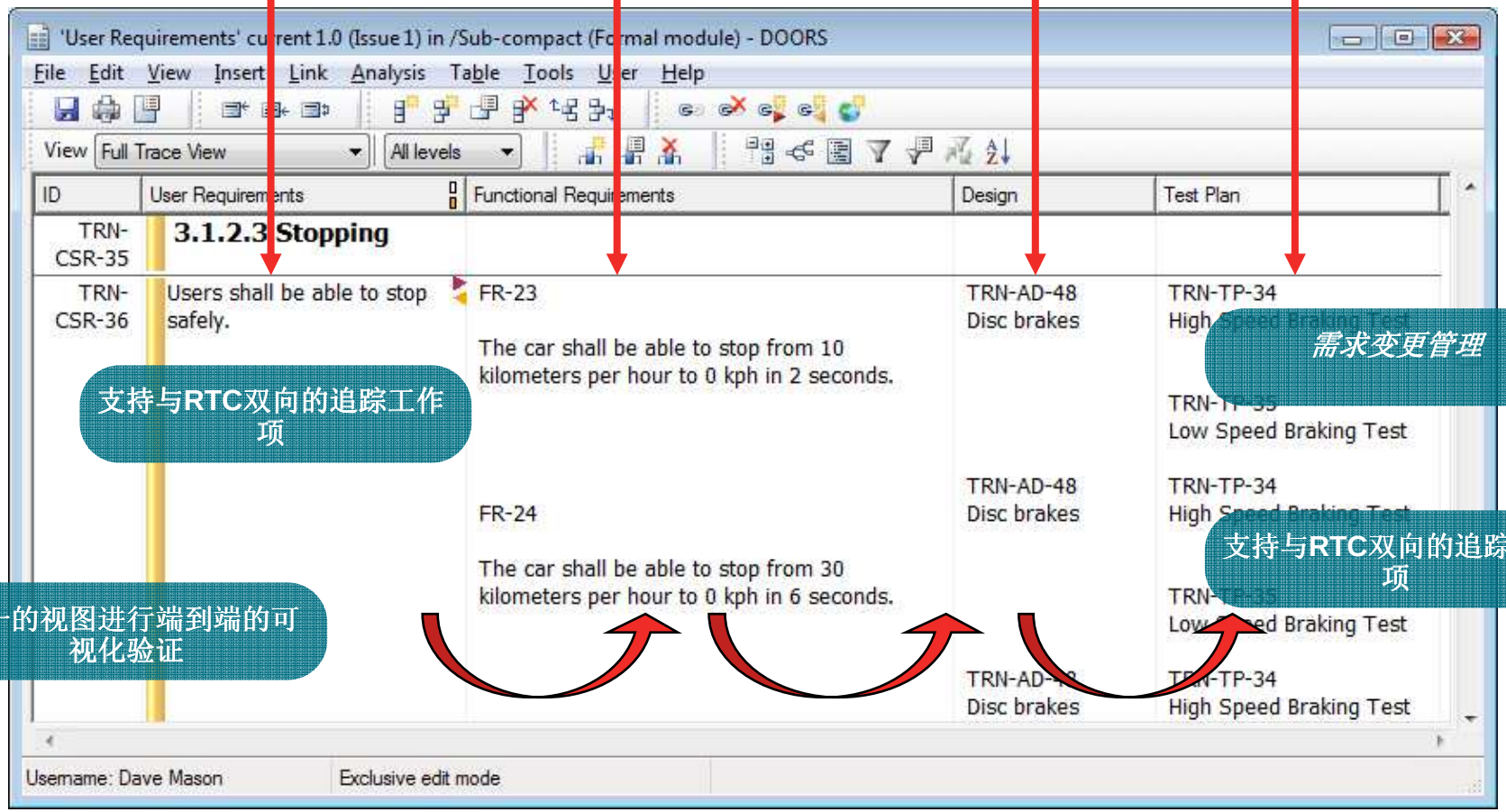


Rational 系统和软件工程工作台



使用Rational Doors在全生命周期管理需求

用户需求　　技术需求　　设计　　测试用例



支持与RTC双向的追踪工作项

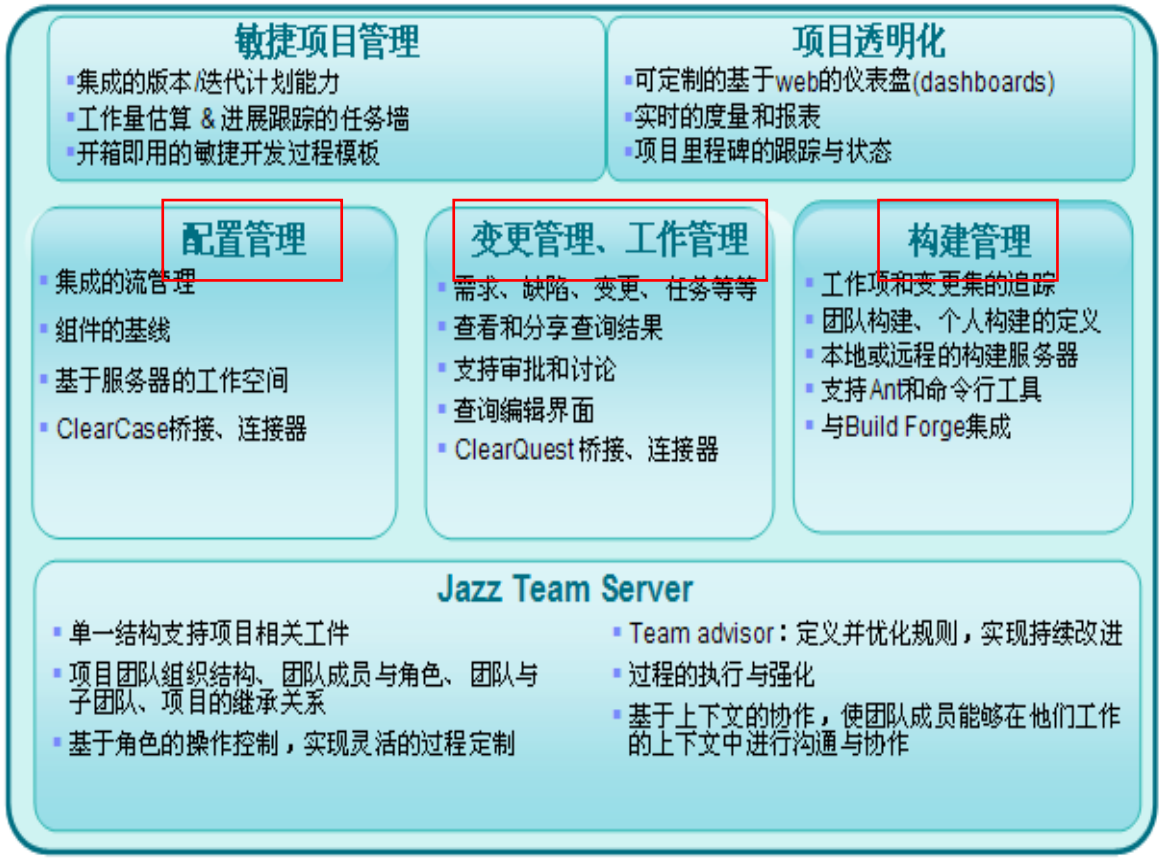
需求变更管理

支持与RTC双向的追踪测试项

统一的视图进行端到端的可视化验证

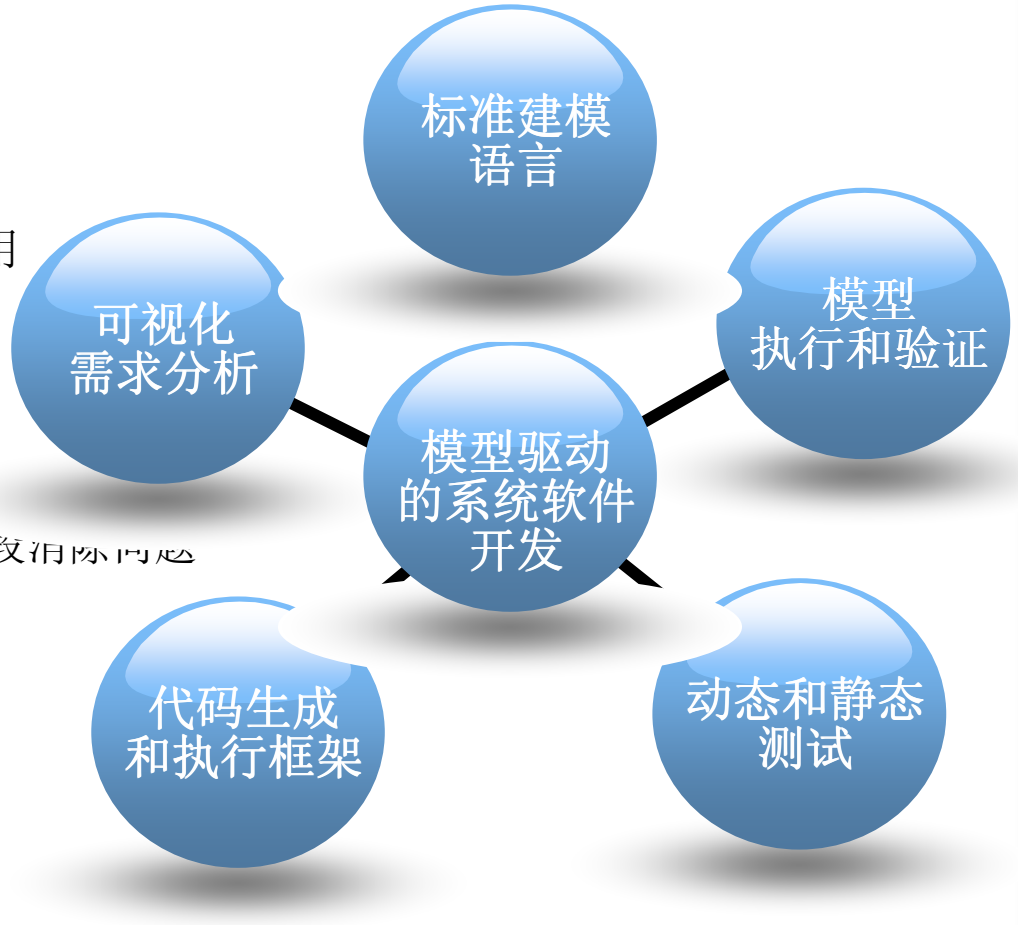
Rational Team Concert——软件协作管理中心

- 一个实时的协作式软件开发平台，目标是提升软件开发团队的协作能力、创新能力和生产率
- 一个可伸缩的、轻量级的ALM解决方案，面向企业中的敏捷开发团队（Agile Development）
- 包含功能
  - 工作管理（需求管理）
  - 配置管理、变更管理
  - 构建管理



Rational Rhapsody——模型驱动实时及嵌入式软件开发平台

- 特点
  - 针对嵌入式及实时操作系统，最新支持Android系统，支持多核框架
  - 捕获需求，设计系统及软件实现
  - 在整个生命周期，基于模型的仿真与测试
  - 开发完整的C, C++, C#, Java以及Ada应用
  - 模型与代码动态关联
- 收益
  - 优化沟通与协作
  - 持续测试，持续质量提升，在项目最早阶段消除问题
  - 自动生成应用及文档

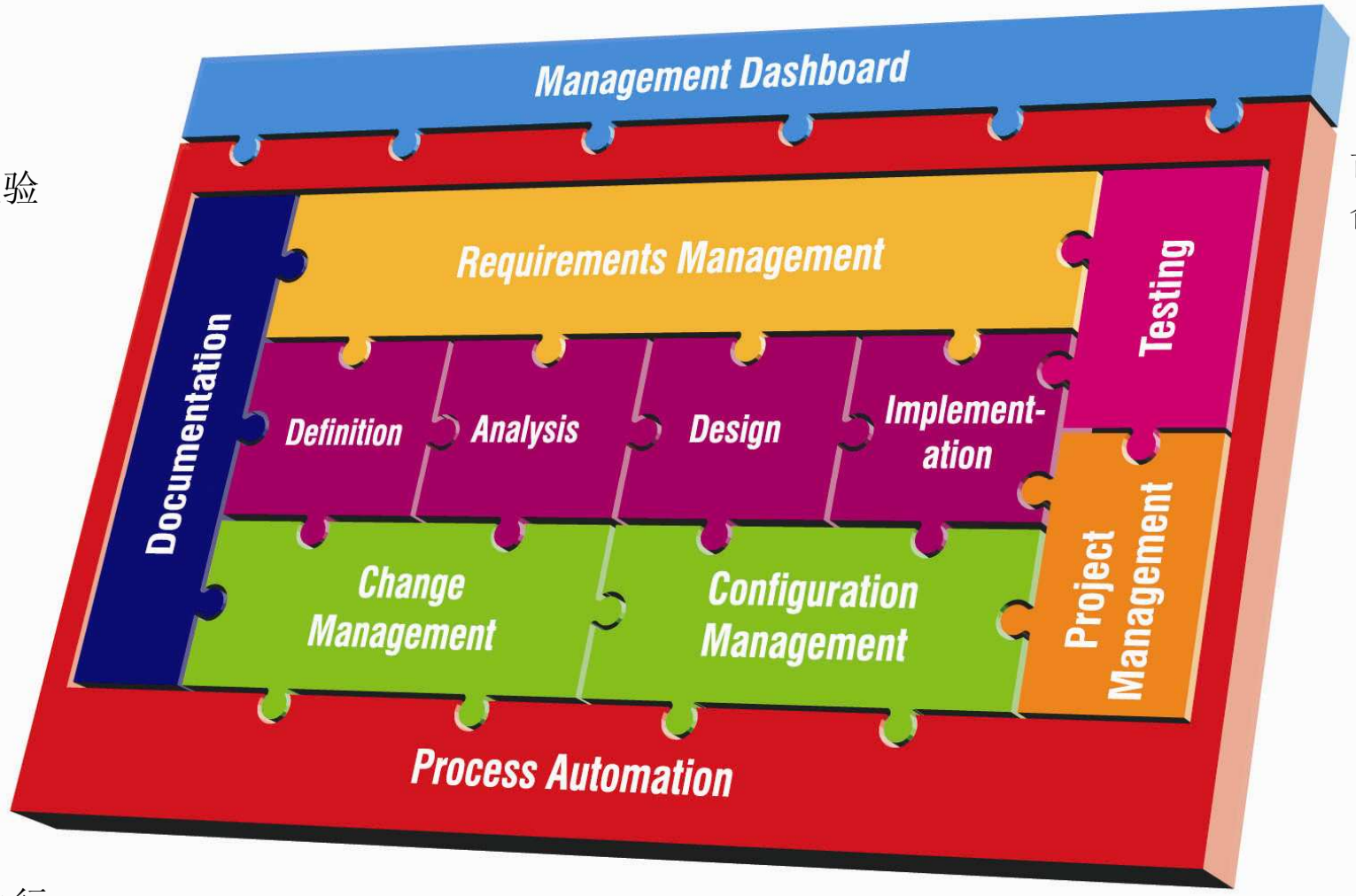


Rational Quality Manager——基于团队，质量驱动软件交付

- **协作**
  - 质量管理中心，为软件交付团队服务
  - Web 2.0 风格协作
  - 重用已有的或是第三方的测试资源
  - 可扩展的plug-in 构架
- **自动化**
  - 自动化文档生成
  - 自动化测试环境管理
- **管理**
  - 贯穿业务对象，能力，测试用例，资源以及缺陷的追踪能力
  - 动态设置风险优先级
  - 质量管理、测试管理、测试开发、试验室管理、自动化测试

有机的集成整个开发生命周期

成功经验



可视化生命周期

卓越的执行

沟通和协作

www.ibm.com/software/rational