

## 质量超越客户期望的三个关键

**Serge Lucio**

质量管理、安全性、产品及战略主管



# 议程

- 定义质量挑战
- 利用生命周期质量管理加强协作和决策
- 利用模型驱动测试提高生产效率
- 将测试整合到质量管理的生命周期
- 适用于系统质量管理的 **Rational** 解决方案
- 问与答



## 每次我们都发现产品变得更智能

- 2007 年，照相手机的销量为十亿台，是 2006 年的两倍
- 独特的可定制电话、电子邮件、音乐、上网、照相机、全球定位系统（GPS）、应用程序、录像机等等都集中在一台设备上
- 在 1999 年时，这听起来像是科幻小说，然而结果是生产力的确飞速增长
- 在 2011 年，它已经过时了！

**2019 年可能会是怎样呢？**

35,000 种应用程序。并且还在继续增加。



# 业务挑战

66%

新产品包含了内置软件组件

Aberdeen Group

41%

软件项目无法提供预期的业务价值和 ROI

Dynamic Marketing Ltd

49%

软件项目超出原来估算的时间和费用

Dynamic Marketing Ltd

200 倍

在维护阶段修正错误的成本与在分析阶段就修正错误的成本相比较

Leefingwell and Widrig



# 运营挑战

质量

太多缺陷产品被发布到市场上

可见性

管理人员不知道是否符合质量目标

复杂度

越来越难测试和诊断设备

成本

测试非常昂贵 - 占项目的 **30%- 50%**

时间

周期后段出现质量意外，延误产品上市时间



# 质量管理产业的现状： 较差的业务可预见性

规划的业务结果



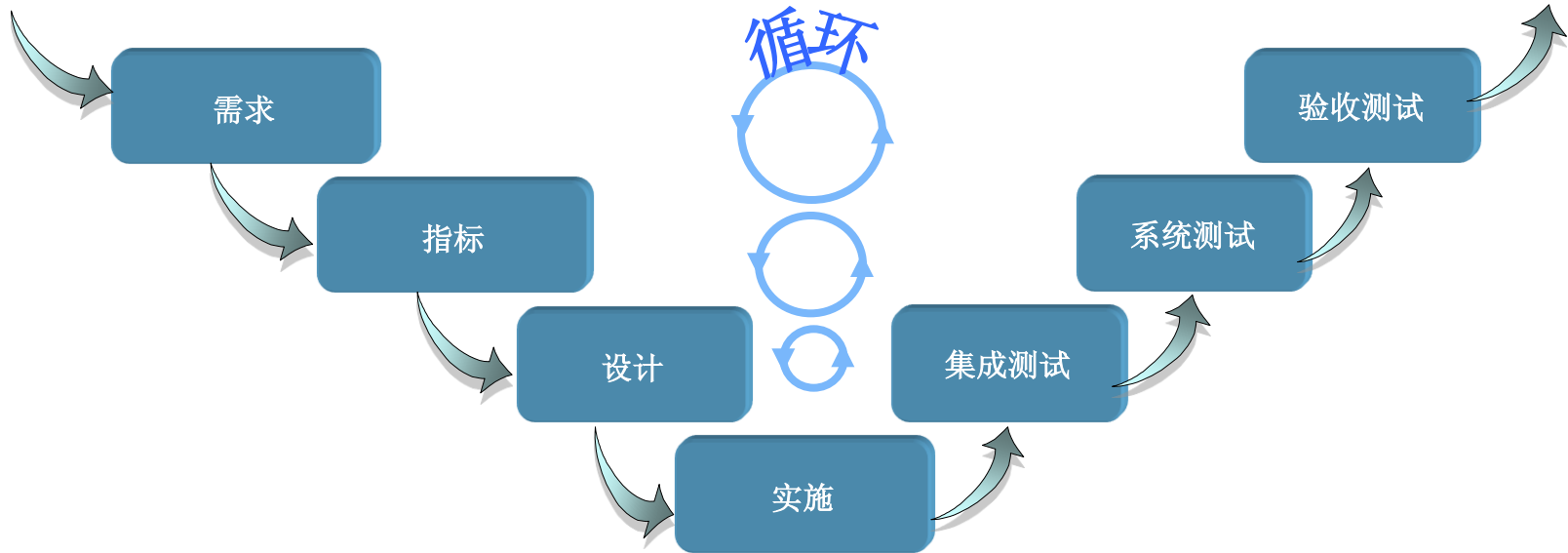
成本（电子表格）  
质量标准  
时间（甘特图）

≠

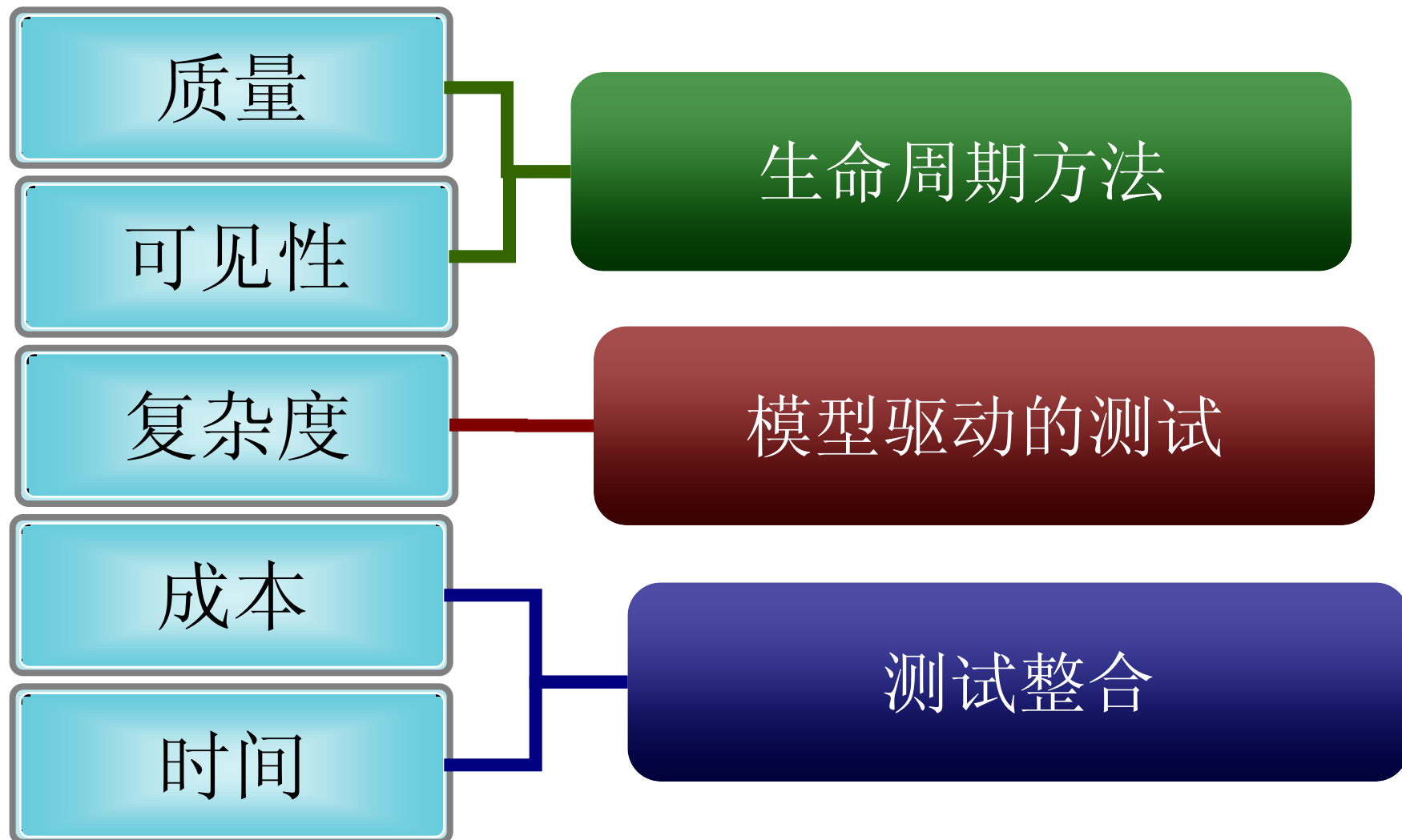
实际的业务结果



成本（电子表格）  
质量标准  
时间（甘特图）



# 应对挑战的三个关键





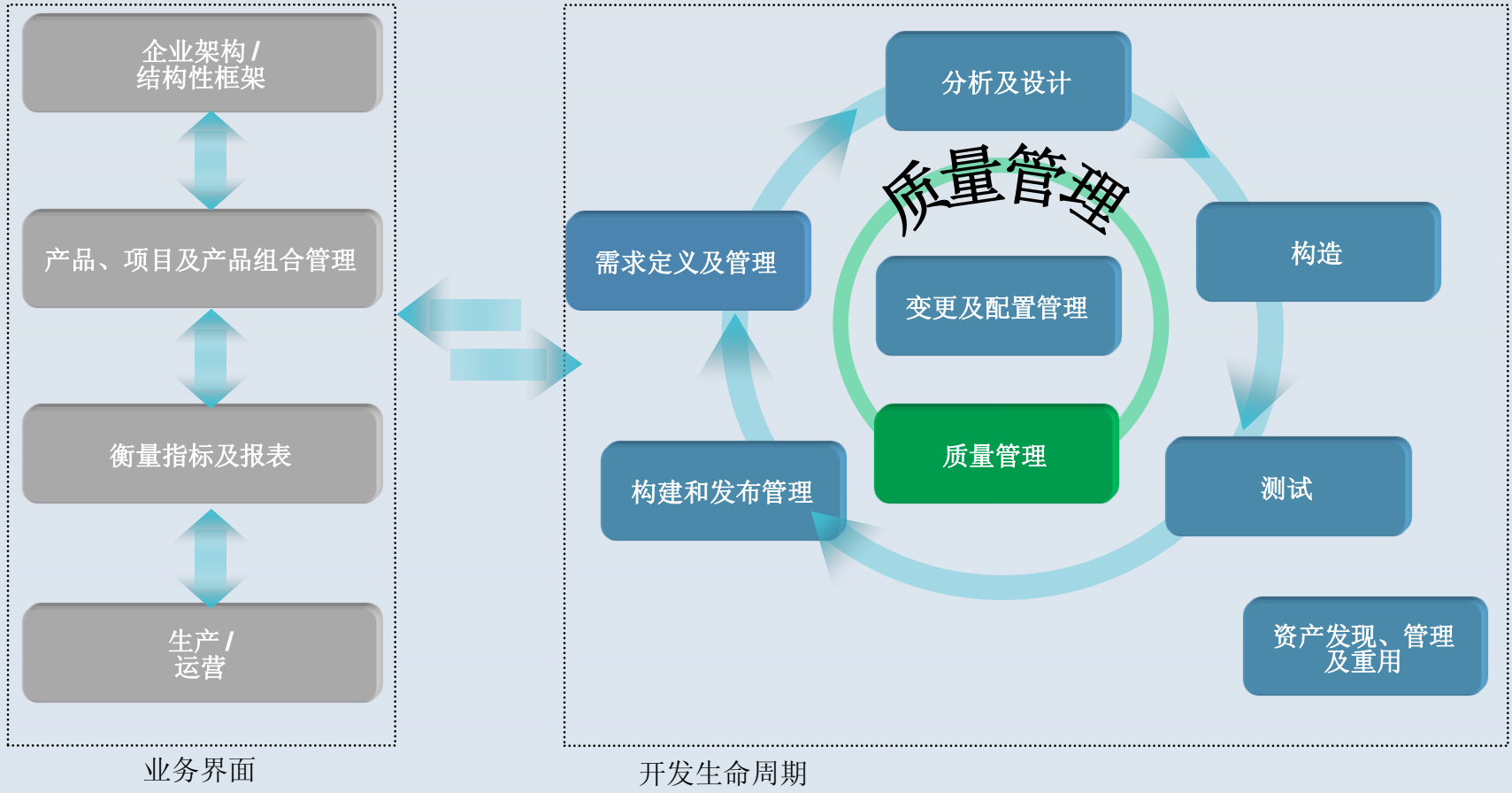
# 议程

- 定义质量挑战
- 利用生命周期质量管理加强协作和决策
- 利用模型驱动测试提高生产效率
- 将测试整合到质量管理的生命周期
- 适用于系统质量管理的 **Rational** 解决方案
- 问与答

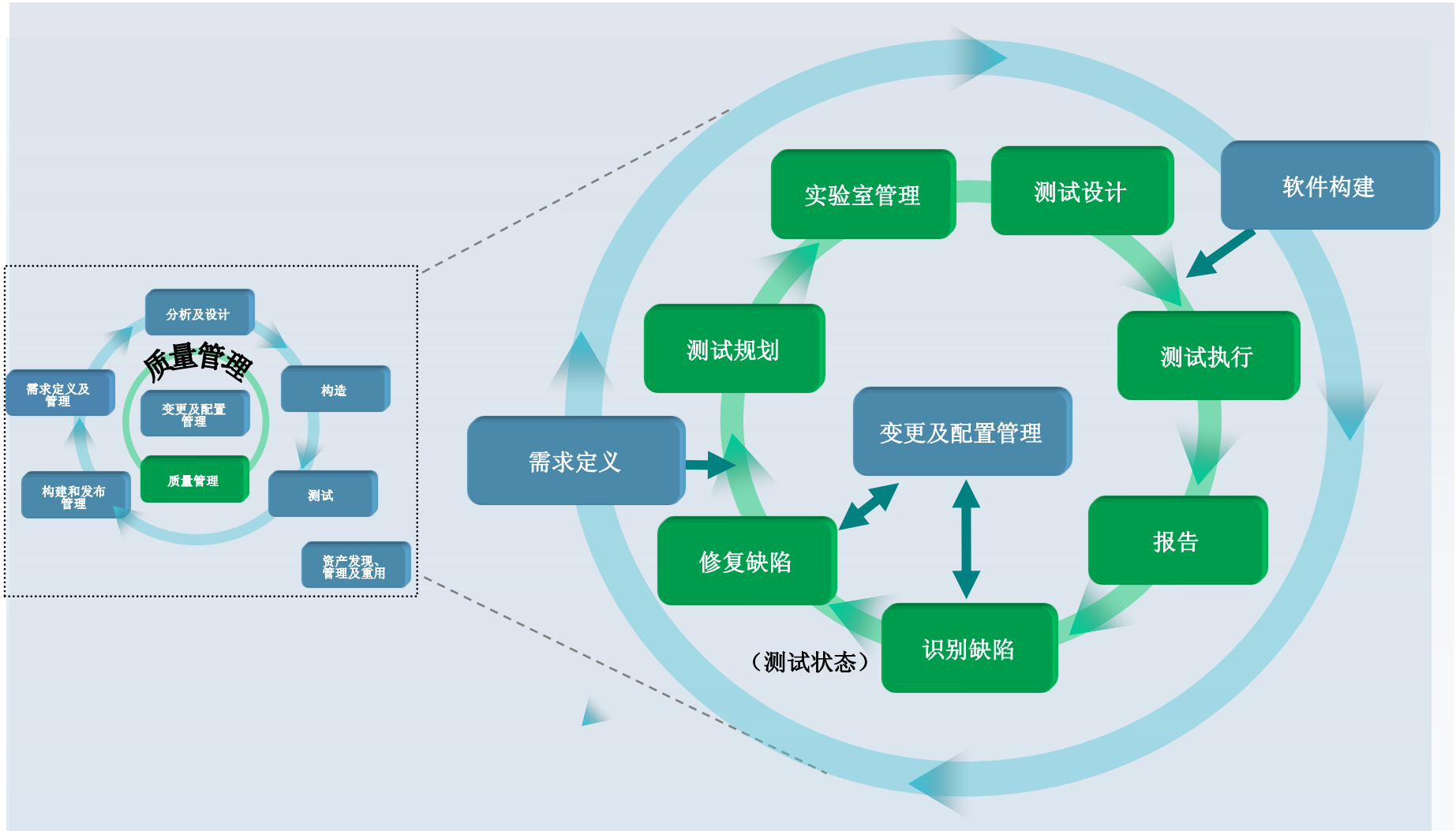




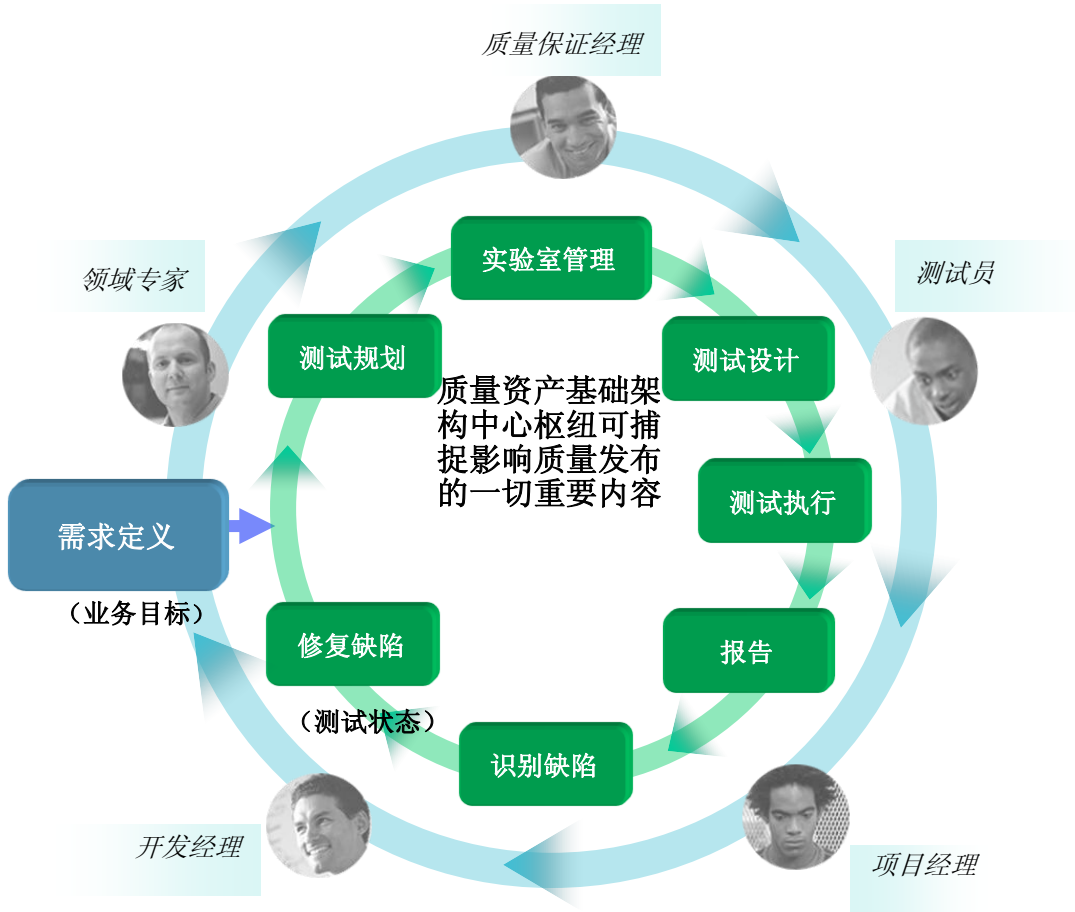
# 开发生命周期



# 质量生命周期



# 在实时合作中统一团队



- 利用质量资产的共享视图统一整个团队
  - ▶ 资产的中心位置  
(例如, 需求、可用资源、目标、现有标准)
- 综合的动态规划及更新
  - ▶ 由整合的工作流程驱动而不是任务驱动团队活动、传递、审查/审批和签发
- 了解其他人的工作进度以及其他人对您的期望
  - ▶ 个人和团队的任务管理



# 可执行的报告

- 利用实时分析制定合理的决策并主动修改
- 测量并管理质量、项目和团队的状态、绩效及结果
- 基于量化信息制定高质量的决策
- 基于相关的质量及项目数据采取实时操作

The image shows two screenshots of the Rational Quality Manager interface. The top screenshot displays a dashboard with a main chart titled 'Execution Status by Test Case using TER Count' and a secondary chart 'Plan Requirements Coverage by Test Case'. A yellow callout box labeled '质量指标仪表盘' (Quality Metrics Dashboard) points to the dashboard area. The bottom screenshot shows a detailed view of the 'Execution Status by Test Case using TER Count' report, featuring a horizontal bar chart with a legend for 'Verdict' (Blocked, Failed, Not Run, Passed). A yellow callout box labeled '定制的报告' (Customized Report) points to this detailed report view.

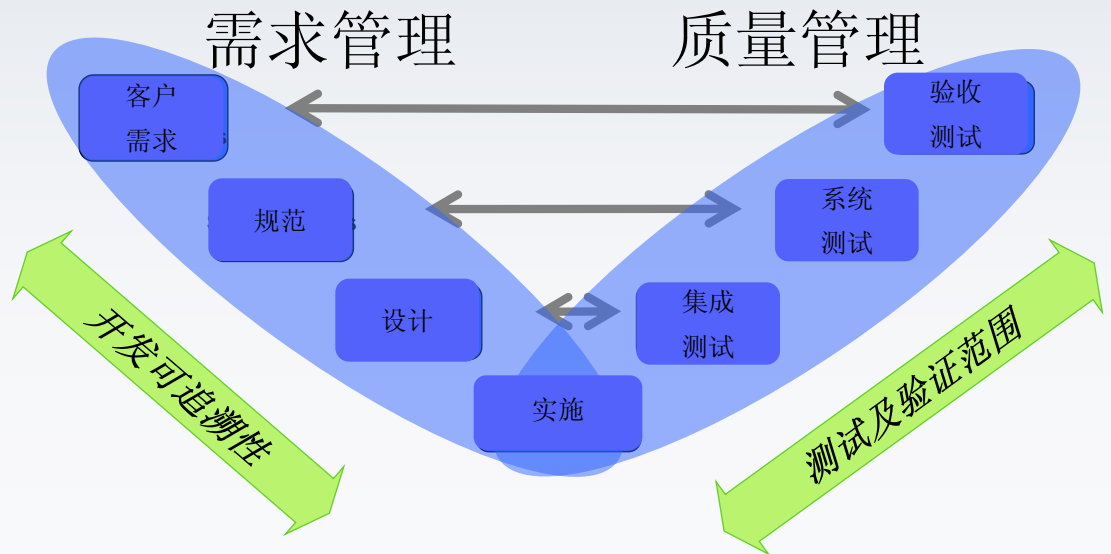


# 在生命周期中推动质量

降低项目风险并改善结果

在整个生命周期中提供收益

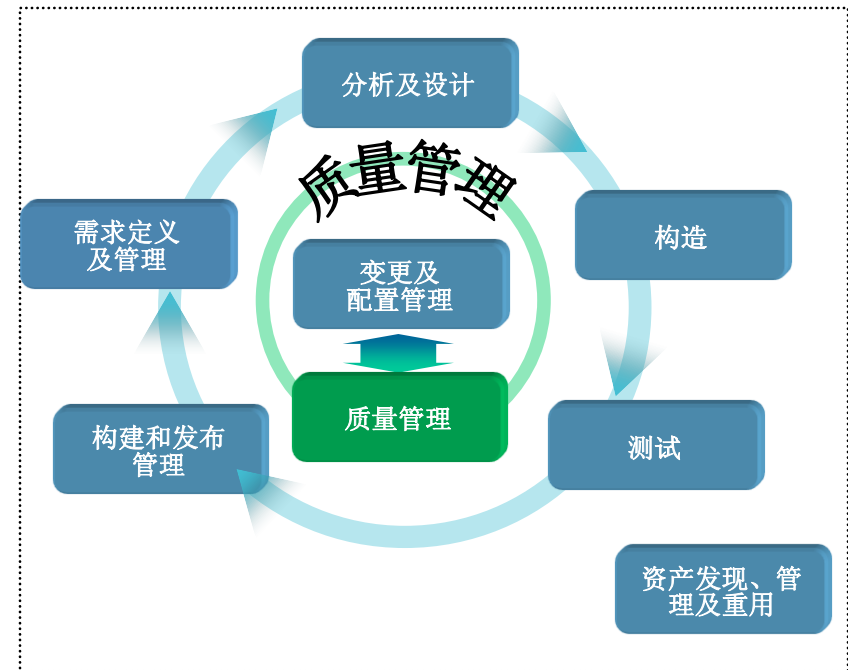
- ✓ 团队共享一套公共的明确的需求
- ✓ 不要漏掉关键需求
- ✓ 评估需求变更的影响
- ✓ 识别最关键的需求并进行测试
- ✓ 证明合规性（可审计）



# 提高项目的弹性

变化是不可避免的 – 使用工具及流程支持它

- 整合缺陷跟踪及变更管理
  - ▶ 开发人员和测试人员需要针对相同的内容工作
- 理顺测试人员与开发人员之间的信息流
  - ▶ 在两个环境之间重复使用信息，而不是采用易于出错的手工移交方式
- 用配置来管理质量管理任务
  - ▶ 通过管理跨多种产品配置的复杂度来提高质量



## 生命周期质量管理获益

- ✓ 质量是不是某个特殊过程、工具或可交付物的输出——它是一个‘健康’的、协作的生命周期的输出
- ✓ 通过优化信息流使相关各方获得相同的信息，协作使质量成为可能
- ✓ 需求使项目可以“做正确的事”
- ✓ 通过有效地执行必要的变更， workflow、规划和变更管理支持的质量提高了
- ✓ 通过促进更好的决策，报告和衡量质量提高





# 议程

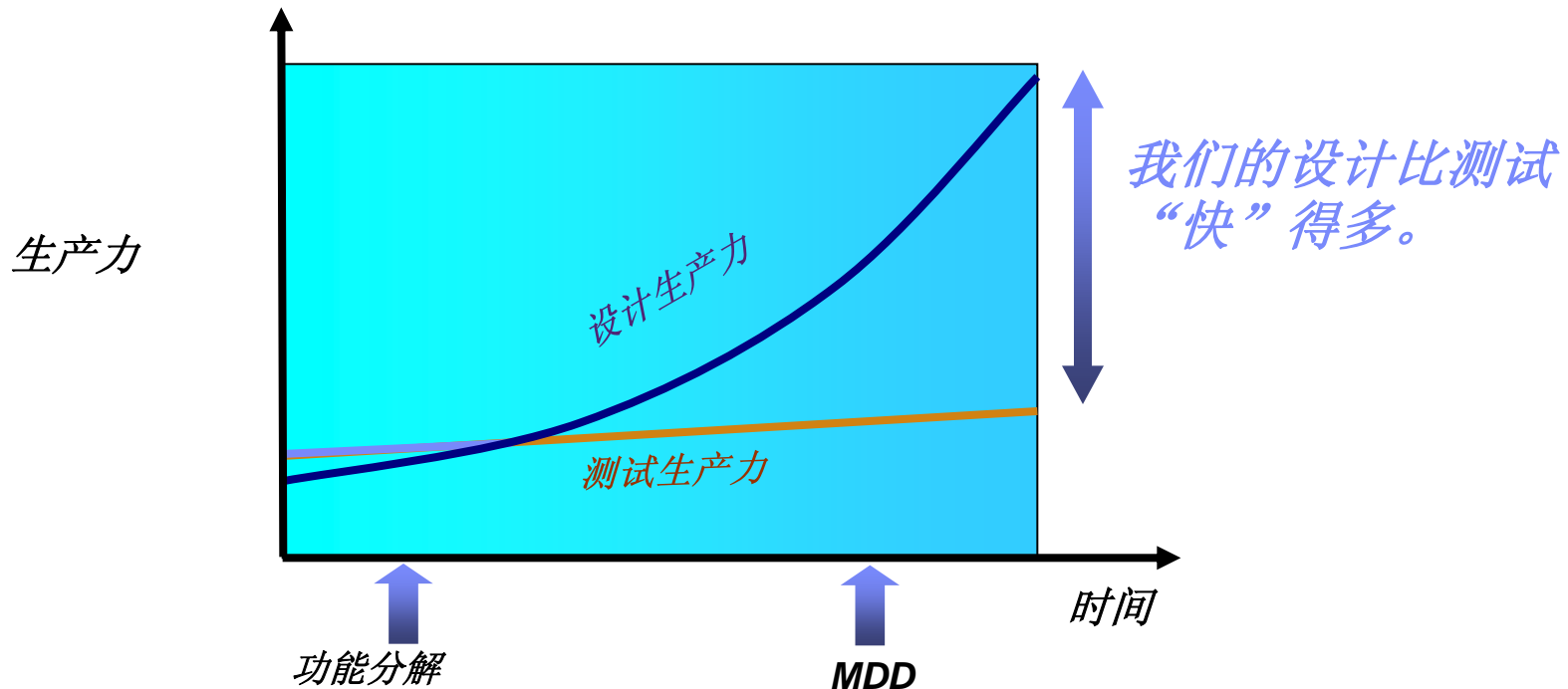
- 定义质量挑战
- 利用生命周期质量管理加强协作和决策
- 利用模型驱动测试提高生产效率
- 将测试整合到质量管理的生命周期
- 适用于系统质量管理的 **Rational** 解决方案
- 问与答





## 但 MDD 还不足够...

- 通过采用 MDD，开发人员提高了他们的生产力，以在更短时间内提供更复杂和更具竞争力的产品...
- 测试仍然停留在 80 年代的代码水平！



- 开发已发展到建模... 测试也同样需要进步



# 使用模型任务测试

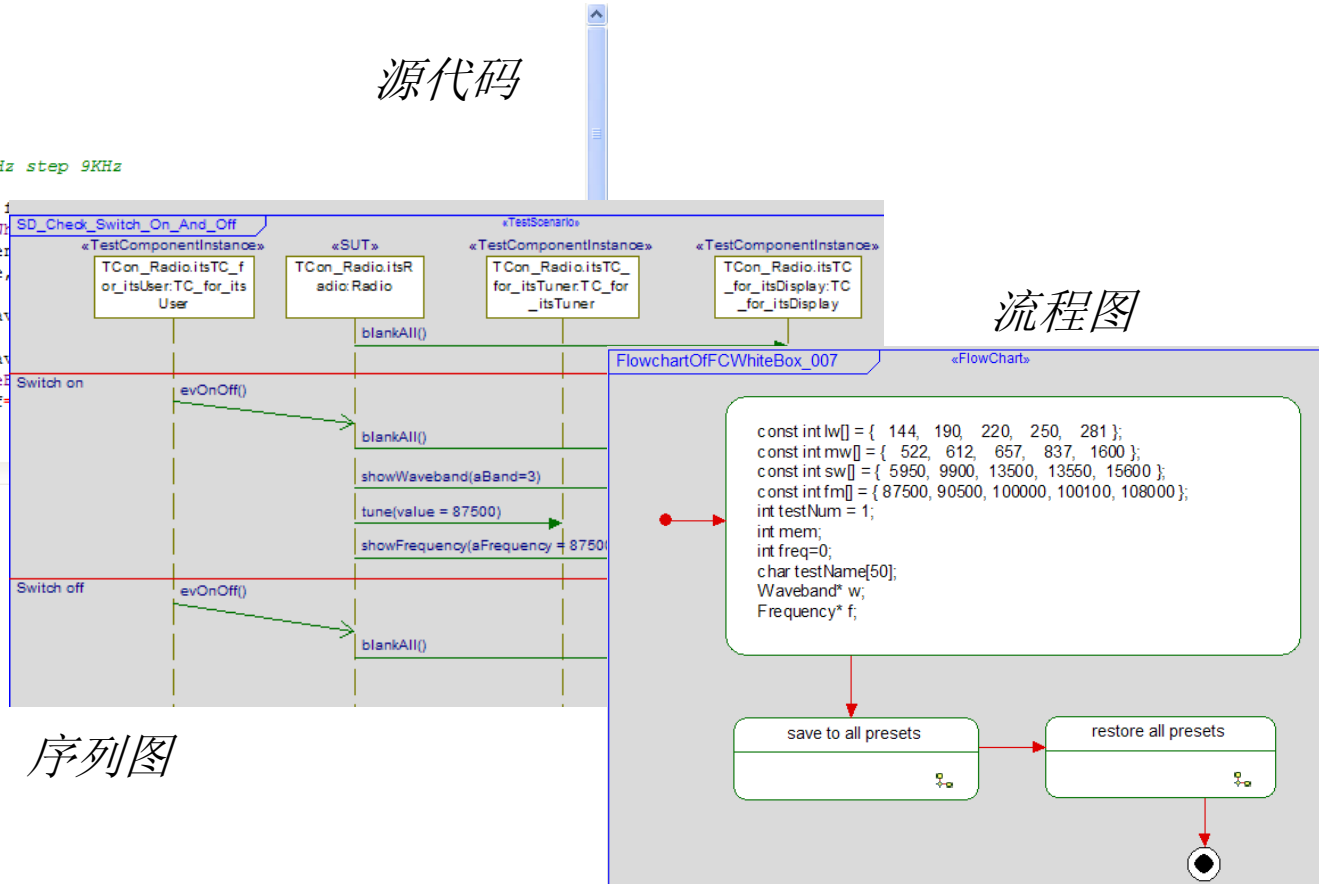
- 用设计模型的方式去测试模型

```
int f;
int freq;
int testNum = 1;
char testName[50];

itsRadio.nextWaveband();

// Test MW 522KHz to 1620KHz step 9KHz
itsRadio.nextWaveband();
for ( freq=522; freq<=1620; f
    sprintf ( testName, "CDW
    f = itsRadio.getItsCurrent
    RTC_ASSERT_NAME (testName,
    testNum++;
    itsRadio.getItsCurrentWav
}
f = itsRadio.getItsCurrentWav
sprintf ( testName, "CDWhite
RTC_ASSERT_NAME (testName, (f
```

源代码



序列图

流程图

- 在您的桌面和您的目标上执行测试

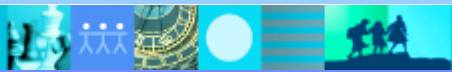


# 议程

- 定义质量挑战
- 利用生命周期质量管理加强协作和决策
- 利用模型驱动测试提高生产效率
- 将测试整合到质量管理的生命周期
- 适用于系统质量管理的 Rational 解决方案
- 问与答



# 系统测试是复杂的



# 将测试整合到生命周期中 需要什么？

The screenshot displays the IBM Rational Quality Manager interface. The main content area shows a test script titled "Redirect user to correct lot" with a "Test Script Overview" and "View Snapshots" link. The script is in a "Draft" state, created by "Ted Tester". The test data is "Unassigned". The test description is "Redirect user to correct lot when current lot is full".

The "Manual Steps" section lists seven steps:

Step	Description	Status
1	User arrives at incorrect lot and lot is full	✗
2	Record parking lot location	✗
3	Identify vehicle	✗
4	Check vehicle is on list of authorised users for current day in alternate lot, and not for current lot	✓
5	Check space is not available in current lot for reallocation	✓
6	Supply directions to alternate lot	✓
7	Vehicle departs from current lot without the barrier opening	✓

The interface also includes a "Keyword View" panel on the right, which is currently empty, showing "No items found." and a search filter "Type Filter Text".

- ▶ 整合以实现按步骤的记录及执行测试
- ▶ 在执行过程中的丰富错误记录
- ▶ 简单直观的界面，执行快速测试
- ▶ 专用测试解决方案的开放式集成
- ▶ 支持测试环境的可追溯性
- ▶ 报告...



# 议程

- 定义质量挑战
- 利用生命周期质量管理加强协作和决策
- 利用模型驱动测试提高生产效率
- 将测试整合到质量管理的生命周期
- 适用于系统质量管理的 **Rational** 解决方案
- 问与答



# Rational Quality Manager



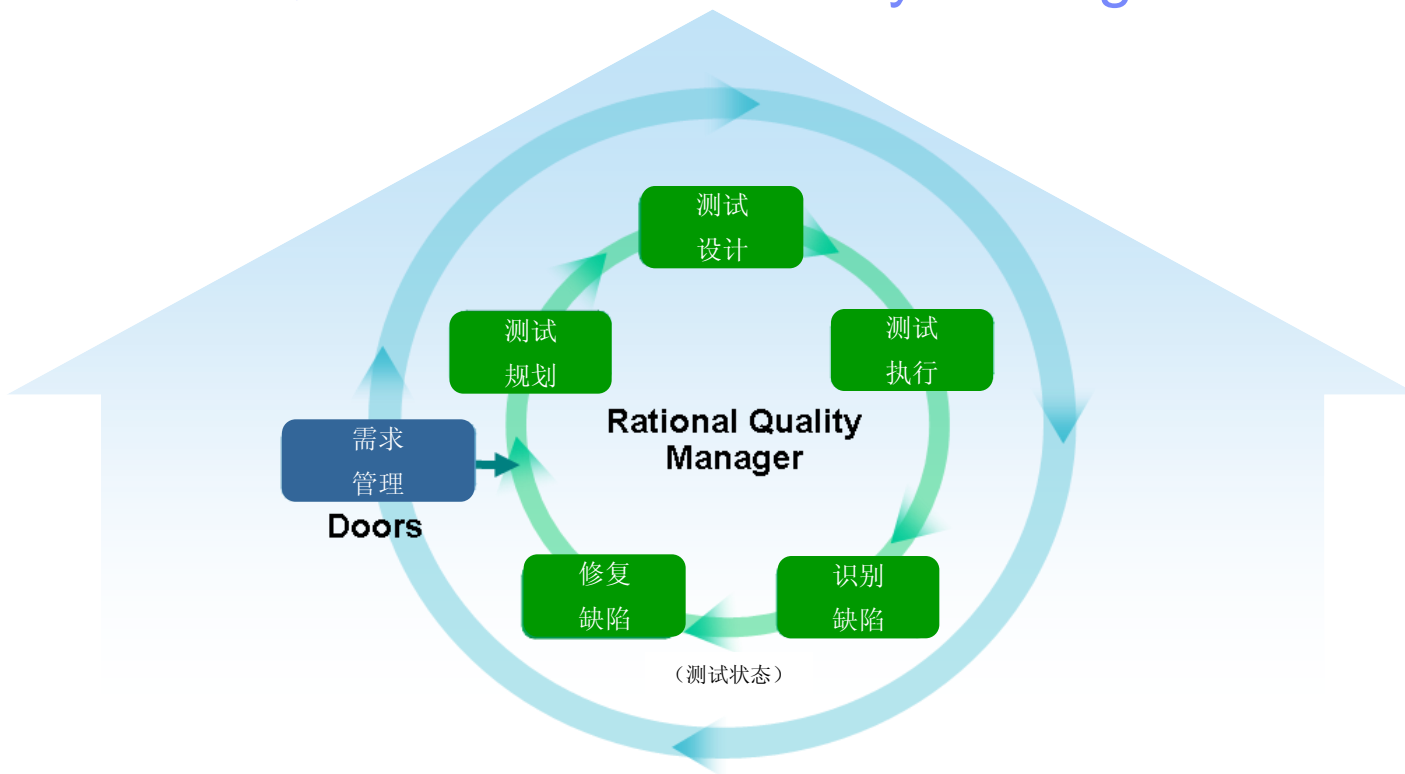
跨所有类型的测试及平台提供集中测试管理中枢以及整个生命周期的支持



自主开发的



# 整合 DOORS 与 IBM Rational Quality Manager



1. 尽早规划测试
2. 尽早执行测试
3. 将测试与需求关联起来

4. 将缺陷与需求关联起来
5. 根据需求测量进度

“基于需求的测试确保了可交付物符合客户的期望”



# 使用 Test Conductor 进行模型驱动的测试

## 测试执行与测试报告

Execute SD\_Check\_Successful\_Connection

SD	Iteration	Status	Progress
SD_tc_0	1	FAILED	57% (8/14)

Develop Test Case

Develop Test Case

执行/报告测试执行

- 输入到 SUT 及 余下行为将自动执行
- 意外的行为被突出显示
- 可以定制测试执行报告以符合企业/项目标准

**Test Case Result**

Test Case: SD\_BB\_TST001  
12:11:43, Friday, July 20, 2007

Environment Info

Test executed on machine:	TEMPRANILLO
Test executed by user:	lgmari
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	7.1, build 893427
Used TestConductor version:	2.0, build 616

Tested Project

Operation Call - In Parameter values do not match.

SD Instance 'check stopwatch initialization'

Iterations:	1
Status:	passed
Progress:	100% (1/1)

SD Instance 'check stopwatch start'

Iterations:	1
Status:	passed
Progress:	100% (8/8)

SD Instance 'check stopwatch stop'

Iterations:	1
Status:	passed
Progress:	100% (3/3)

SD Instance 'check stopwatch restart'

Iterations:	1
Status:	passed
Progress:	100% (7/7)

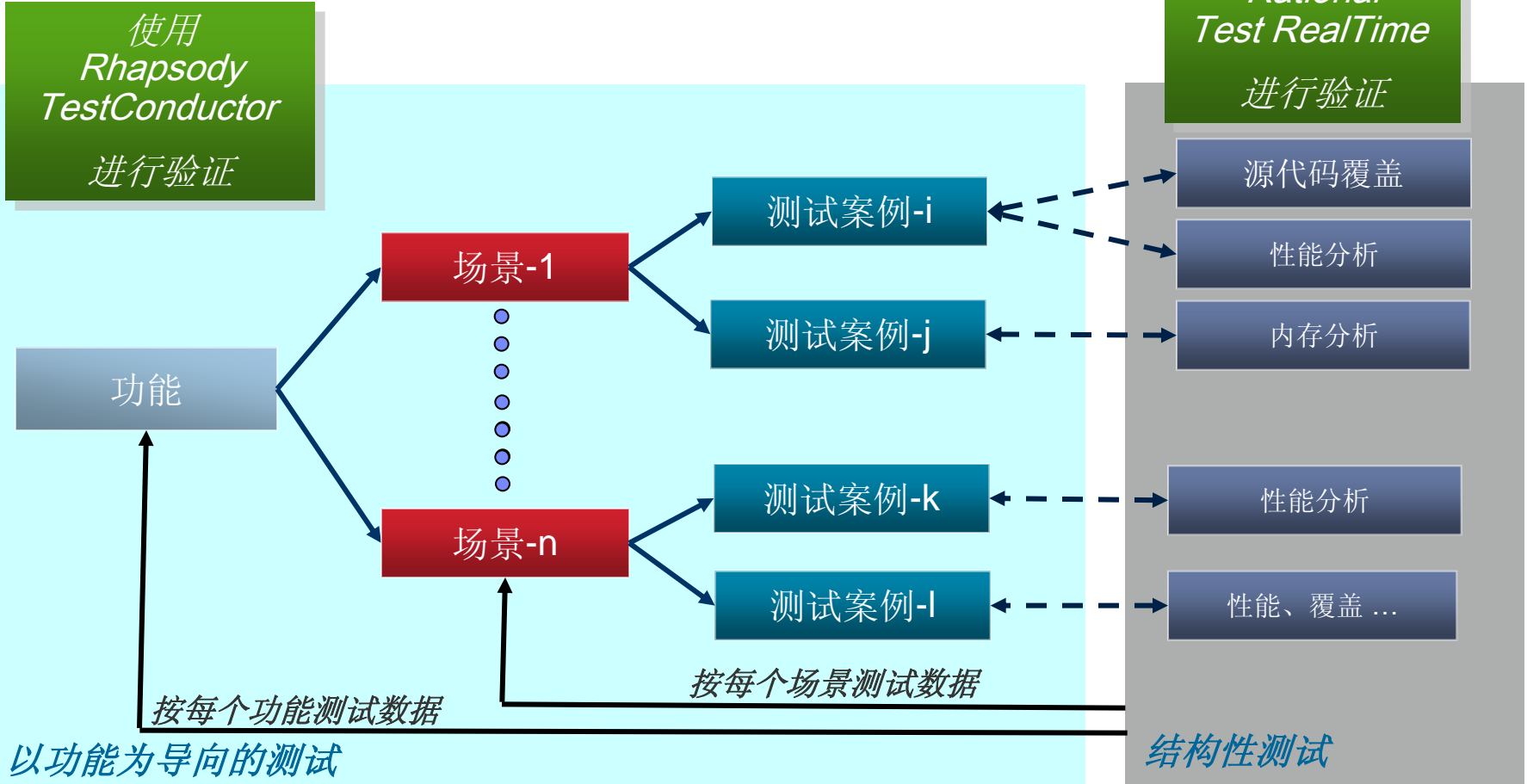
SD Instance 'check stopwatch reset'

Iterations:	1
Status:	passed
Progress:	100% (3/3)

## Rational Rhapsody TestConductor 与 Rational Quality Manager 集成

- 从 RQM 启动针对基于模型的 Rhapsody TestConductor 的测试案例的全面执行控制及管理
- 可通过 RQM 获得执行状态（通过/失败）和结果报告（执行结果、覆盖结果）
- RQM 可以利用 TestConductor 的执行结果来持续提供透明和最新质量保证统计及质量保证报告

# 两全其美... 模型测试与源代码测试同步



- 测试数据包括与执行测试案例相关的任何运行时间信息，如性能分析、源代码覆盖、运行时间复杂度等。



# 整合全部功能： IBM Rational 系统质量管理解决方案

在开发中管理变更

在整个产品开发过程中  
记录并管理需求及变更

测试生态系统

**Rational Change**

**Rational ClearQuest**

**Rational Team Concert**

**Rational DOORS**

**Rational Quality Manager**

**Rational Rhapsody**

**Rational TestRT**

其它 Rational / 第三方测试工具

手工测试

**Rational Rhapsody TestConductor**

在设计与测试过程中使用建模管理复杂度

# IBM Rational 质量管理解决方案 获益综述

- 建立在一个全生命周期协作的平台之上
- 通过及时获得正确的质量信息改善决策制定
- 确保项目达到业务目标
  - ▶ 将需求整合到质量管理生命周期中
- 释放模型驱动的开发的力量
  - ▶ 将模型驱动测试整合到质量管理生命周期中
- 将测试与生命周期联系起来





# QUESTIONS





请访问以下网址，了解更多相关信息：

<http://www-01.ibm.com/software/rational/offerings/quality/>

© 版权所有 IBM Corporation 2009。保留所有权利。本材料中包含的信息仅用于提供信息的目的，并且只根据现状提供，没有任何隐含或者明确的担保。IBM 不承担因为使用本文内容和相关内容而造成损害的责任。本文中不包含的内容不打算、也不应该作为 IBM 或其供应商或其许可证销售商的担保或表示，或者修改适用于 IBM 软件的许可证协议的条款和条件。本文中对 IBM 产品、计划和服务的引用不代表它们可用于所有 IBM 运营的国家。本文中引用的产品发布日期和/或功能可能在任何时候变更，这根据 IBM 单方对市场机遇或其他因素的判断力，不代表 IBM 对未来产品或功能可用性的承诺。IBM、IBM 徽标、Rational、Rational 徽标、Telelogic、Telelogic 徽标以及其它 IBM 产品和服务是国际商业机器公司在美国和/或其他国家或地区的商标。其他公司、产品或服务名称可能是其他公司的商标或服务标志。

