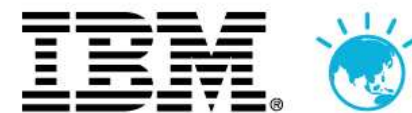


IBM Rational 软件创新论坛



Let's **build** a
smarter planet.

开发有道
创新“智”造

Innovate**2010**



理性的企业软件过程改进

- 中文演讲主题：理性的企业软件过程改进

- 中文姓名：李卫锋

- 中文Title（公司、部门、职称）：Rational架构师

- 演讲主题中文介绍：CMMI是一个非常受欢迎的软件评估模型。然而组织常常难以达到预期的水平，Rational提供了端到端的平台来帮助企业基于CMMI开展软件过程改进。

IBM Rational不仅可以助力企业构建整个过程改进的体系架构，

而且能够使企业基于Rational的软件交付平台

自动化的遍历整个过程改进所需的最佳实践，

并通过真实客观的过程数据、报表和

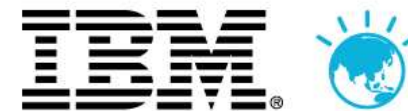
仪表盘进行科学严格的过程度量。

此交流主题将和大家讨论如何基于

IBM Rational定义和实施满足企业需求的过程

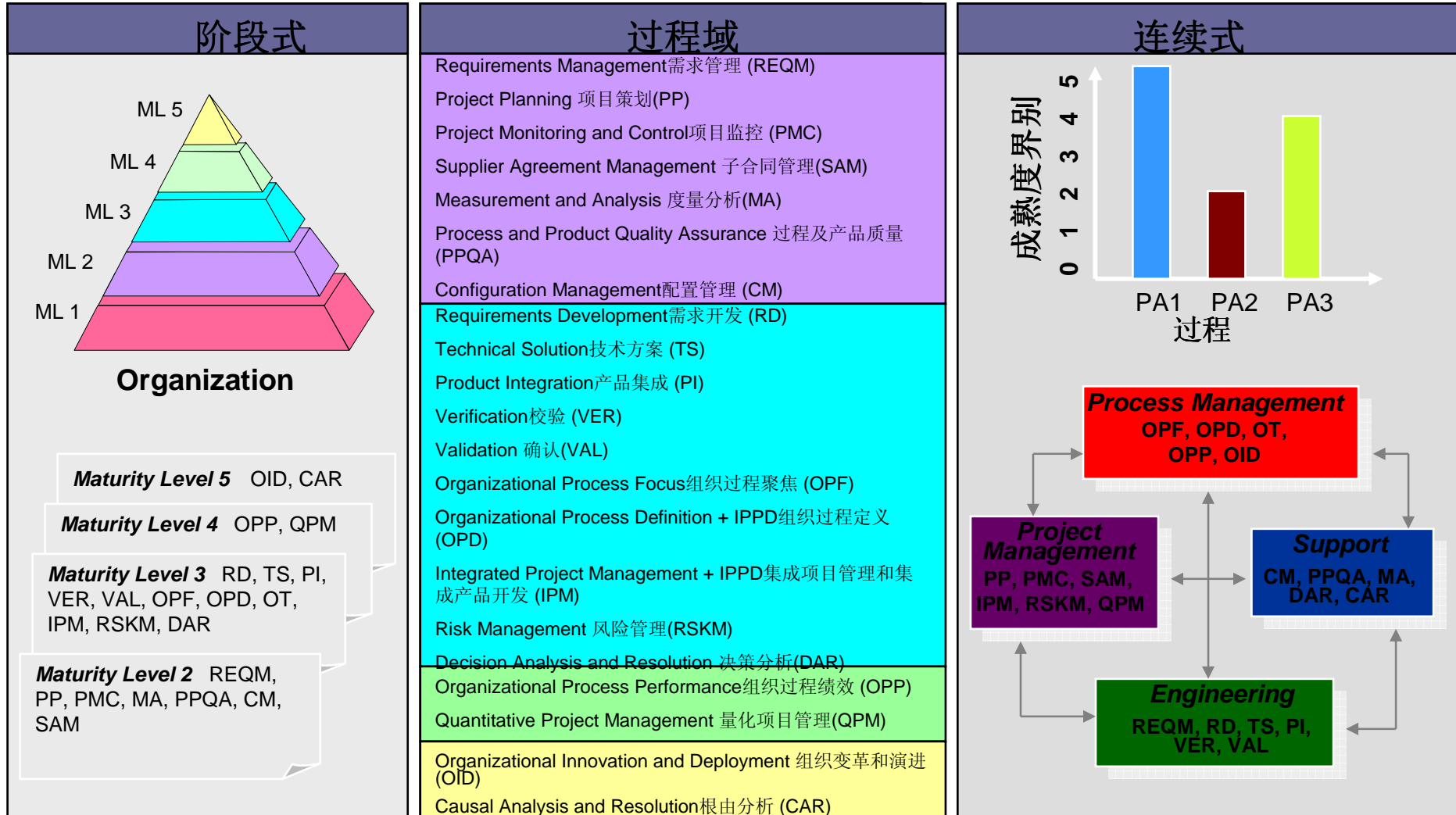
改进平台，并如何自动化这个过程，以及如何

使用Rational洞察收集的数据从而使企业真正受益。



Innovate2010

CMMI 模型



基于CMMI进行过程改进遇到的挑战和陷阱

- 挑战

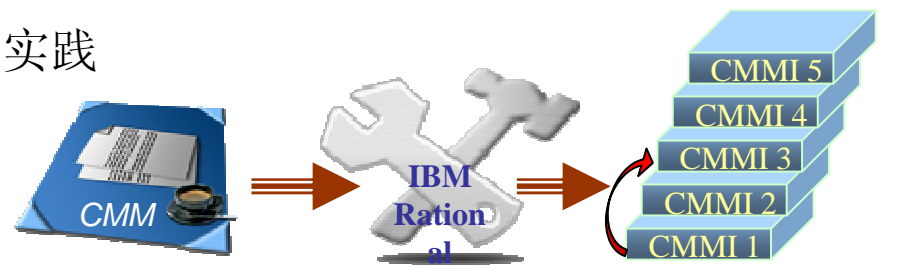
- CMMI模型提供了组织要经历的最佳实践但没有如何落地最佳实践
可绝大多数最佳实践没有自动化的工具支持
组织不清楚应该度量哪些和怎样度量可用来审计的过程数据
为了文档化花费太多的时间
不清楚如何设计裁剪过程以适用不同类型的项目

- 陷阱

- 在一次改进中涉及太多领域的流程，或没有一个明确的改进路线图
- 部署中矫枉过正，或与实际现状脱节
- 改进中没有相关的度量过程，难以有效监督和控制改进有效性

IBM SDP 全面支持CMMI实施

- IBM 软件交付平台 (SDP) 恰能补充CMMI以下方面的空白：
 - RMC 为 CMMI 实施提供了流程框架
 - CMMI 只是定义了一些标准，为了保证产品质量，一个开发团队应该做到什么，而RMC 帮助企业针对产品开发定义了一个完善的软件开发流程，并通过MCIF告诉软件团队如何去达到CMMI所定义的目标。
 - IBM SDP 为 CMMI中的很多关键过程域提供软件工程技术；
 - CMMI中的很多目标都需要工具的支持：
 - 工具平台自动化能力能够让规程成为实践
 - 工具平台自动化能够简化规程的实践
 - 工具平台有效保证过程质量，从而提高产品质量

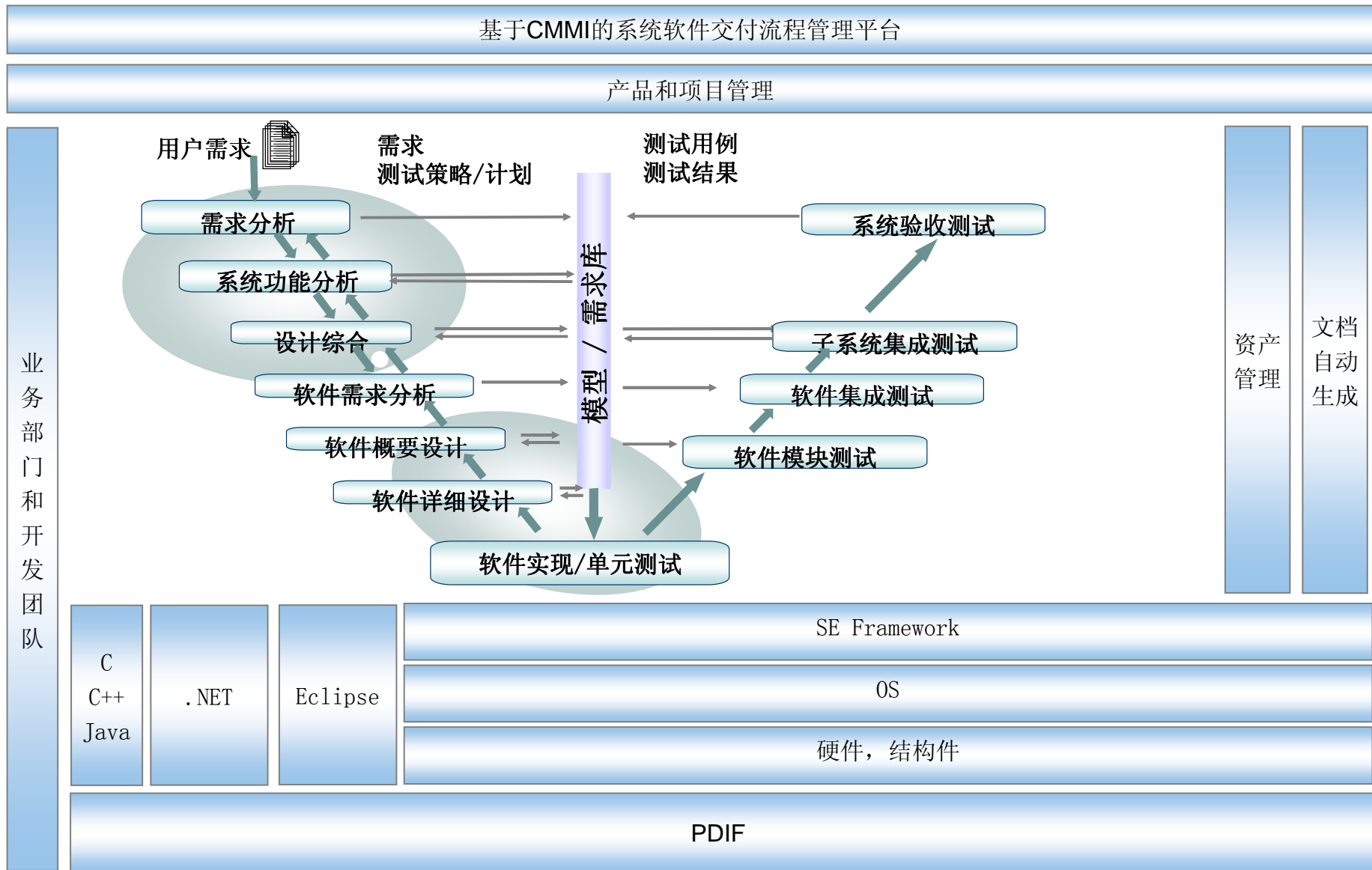


标准：指明目标

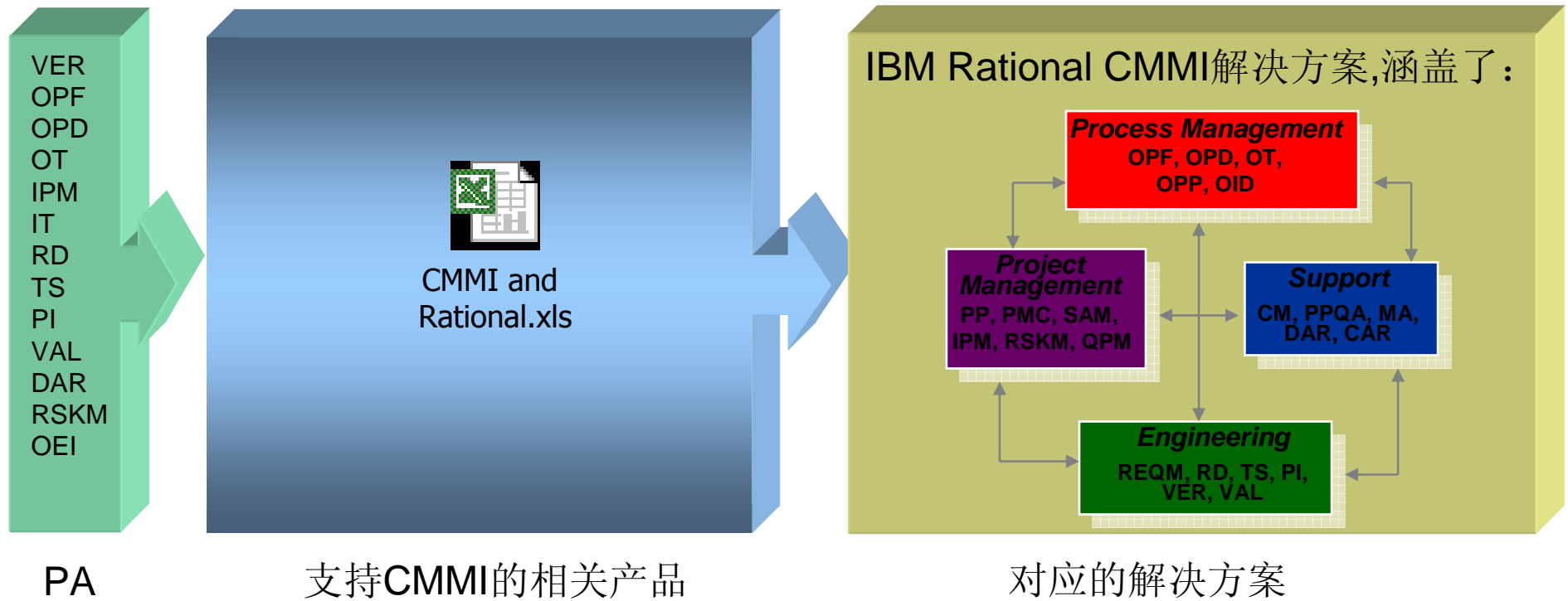
工具：解决如何实现

达到更高的CMMI成熟度级别

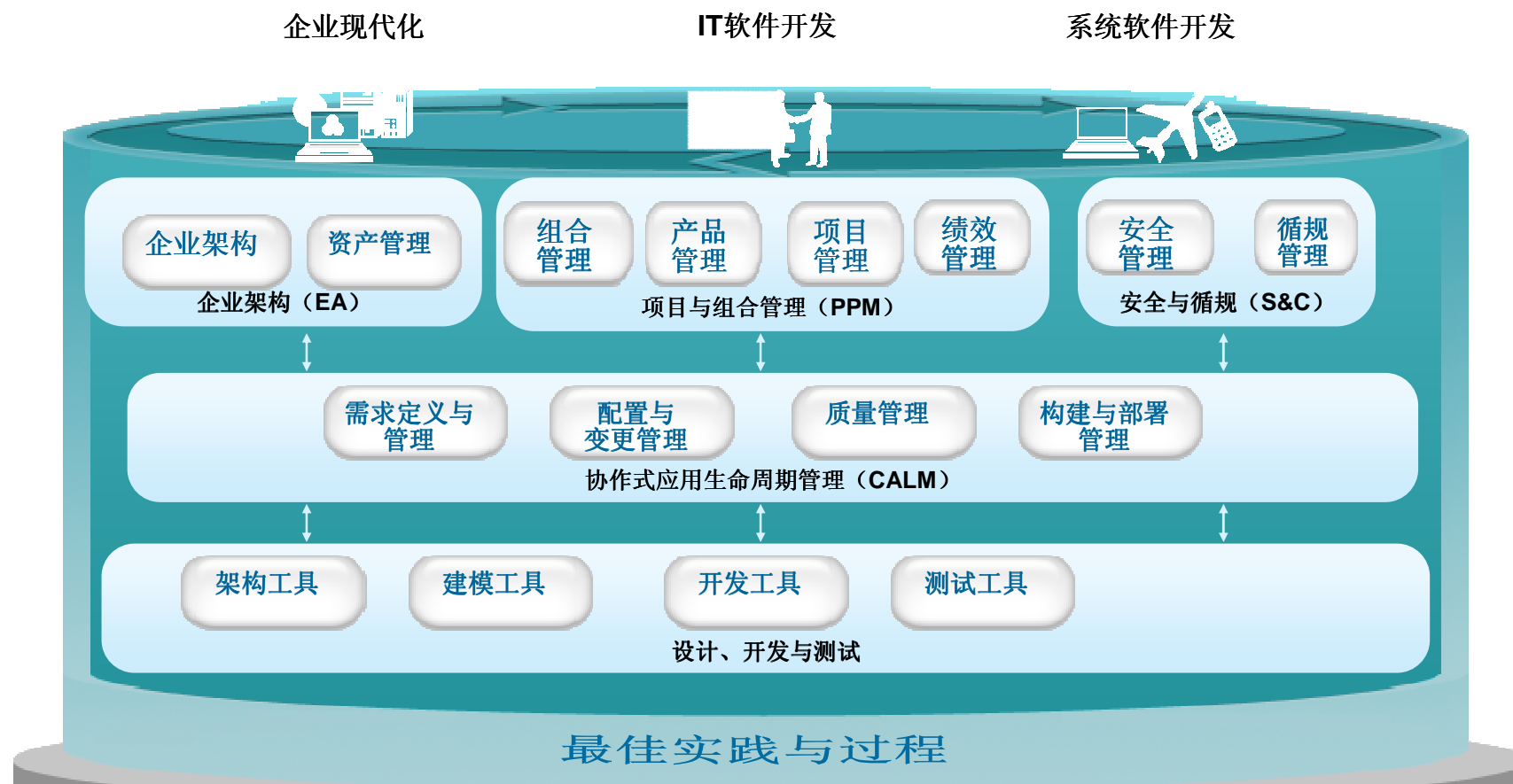
基于IBM Rational 软件交付平台 (SDP) 的CMMI PI支撑平台和部署模型图



基于IBM Rational 软件交付平台 (SDP) 的CMMI PI支撑平台构成



基于IBM Rational 软件交付平台 (SDP) 的CMMI PI支撑平台构成模块



Rational Software Delivery Platform *powered by jazz*

IBM and Business Partner Ecosystem

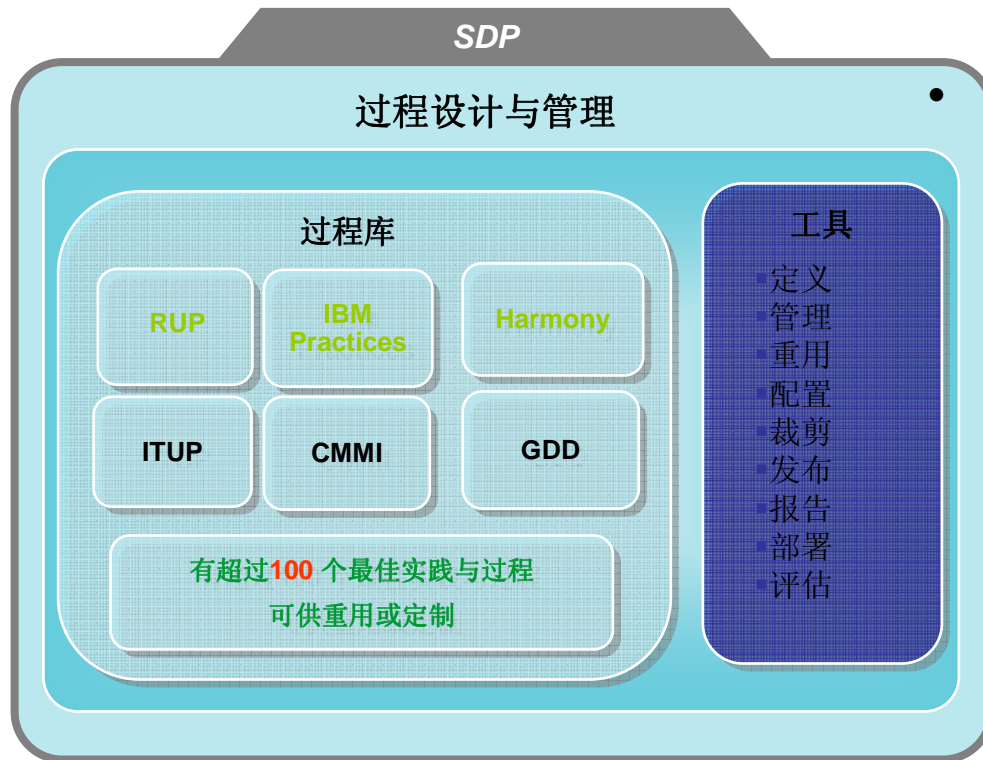
Innovate2010

Let's build a smarter planet.



IBM Rational Method Composer (RMC)

软件与系统开发的最佳实践与过程



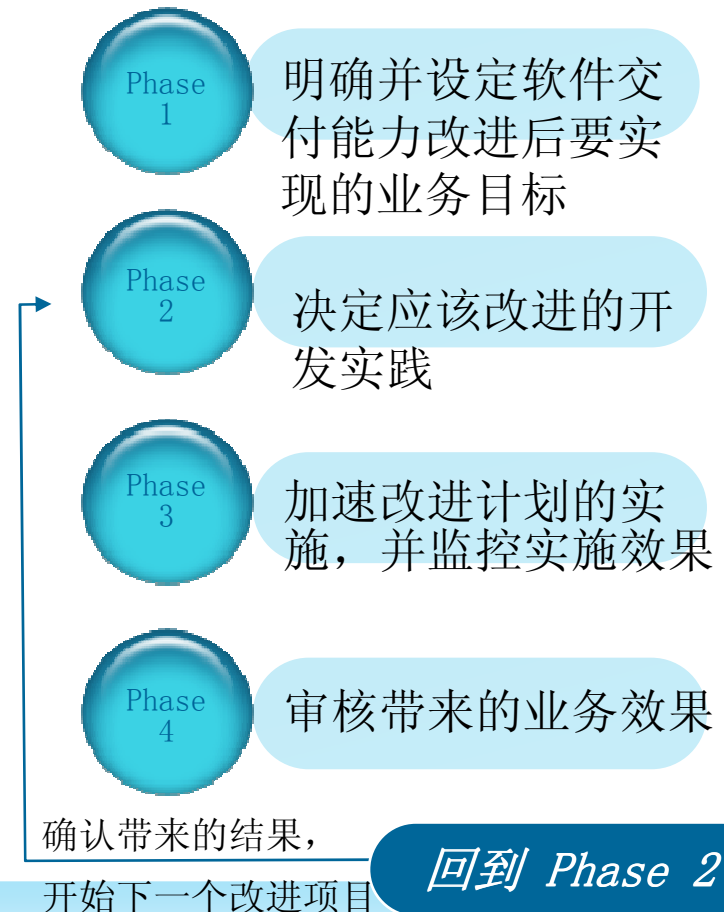
- **工具**: 让你以非常简便的方式进行创建、编辑、维护和发布自己的开发过程
- **过程库**: 可重用、可定制的最佳实践经验与过程库, 包括:
 - **RUP for CMMI®**: 指导如何使用RUP实现CMMI的方法论。等等。
 - **Rational Unified Process (RUP)**: 经典的迭代化软件开发最佳实践经验。
 - **IBM Practices**: 融合了RUP和一些主流敏捷方法论, 形成更加灵活的、模块化的、可增量式实施的大规模敏捷开发过程。
 - **Harmony**: 系统工程与嵌入式软件开发方法论。

Measured Capability Improvement Framework (MCIF)

这是种先进的过程改进方法，用于提升企业的软件交付能力

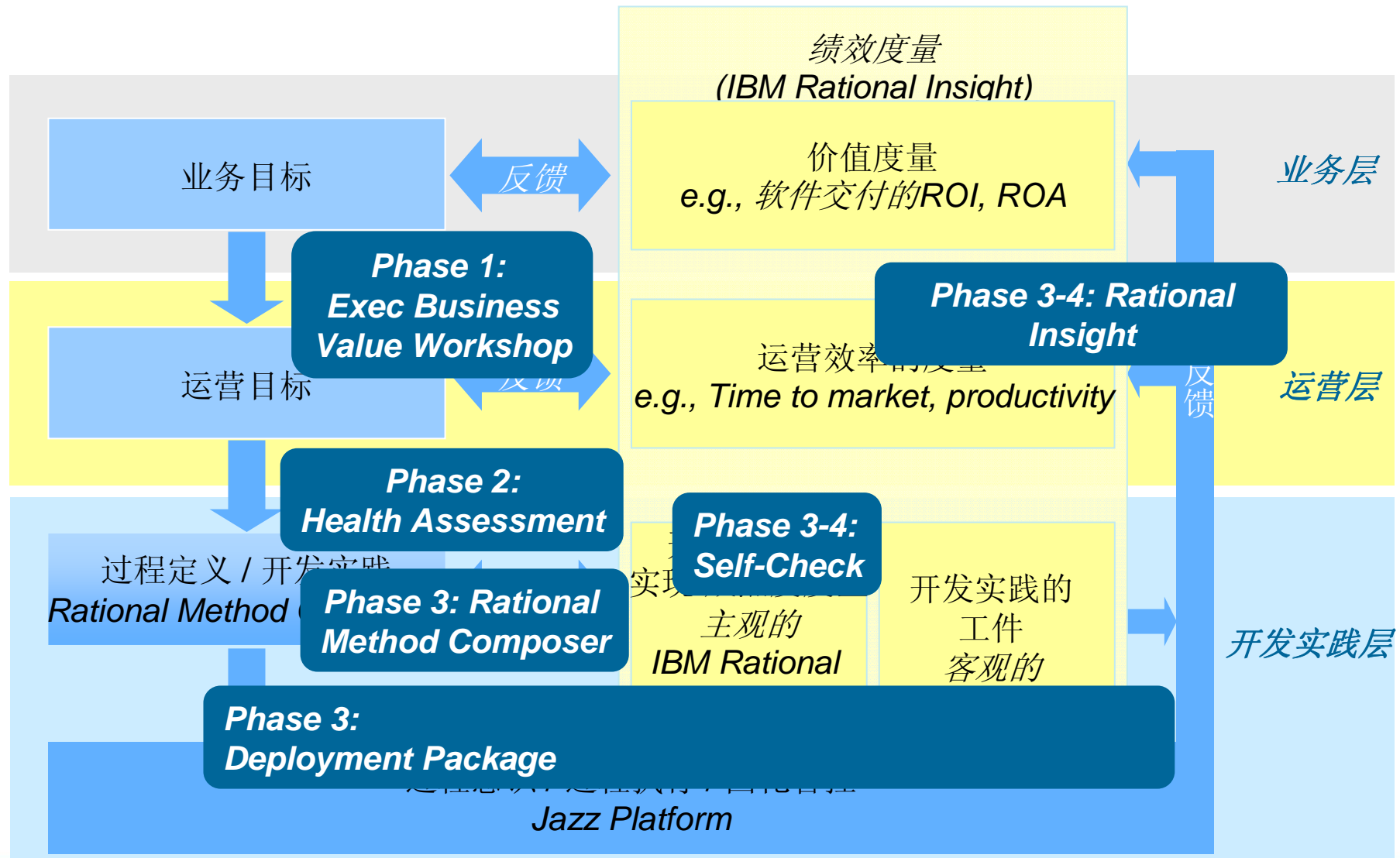


- 目标：通过可测量的、持续的过程改进，可靠地提升企业软件开发与交付的能力，驱动企业的业务创新、降低成本
- 这是一种分阶段的方法，帮助软件开发团队：
 - 采用一种增量式的、可测量的方法，来提升软件开发与交付能力
 - 关注哪些对企业软件交付能力影响最大的软件开发领域和实践
 - 通过IBM在10多年里积累的“开箱即用”软件资产来加速团队实施MCIF方法
 - 通过带来的业务价值，来体现软件交付能力的提升结果
 - 支持任何软件开发过程与方法
 - 在支持Agile开发方面做了更多的优化



MCIF使开发团队能够测量、管理，并逐步地改进软件交付能力。

MCIF方法与资产可帮助企业逐步构建软件交付控制框架



MCIF包含的服务组件

从这里开始
决定正确的解决方案

Phase 1

明确并设定软件交付能力改进后要实现的业务目标

- IBM Executive Business Value Workshop for Software Delivery

Phase 2

决定应该改进的开发实践

- IBM Health Assessment for Software Delivery

从这里开始
实施解决方案

Phase 3

使用工具与服务来加速实施

- IBM Rational Method Composer
- IBM Rational Rapid Deployment Package for Agility @ Scale
- IBM Rational Self Check for Software Teams
- IBM Rational SDP工具

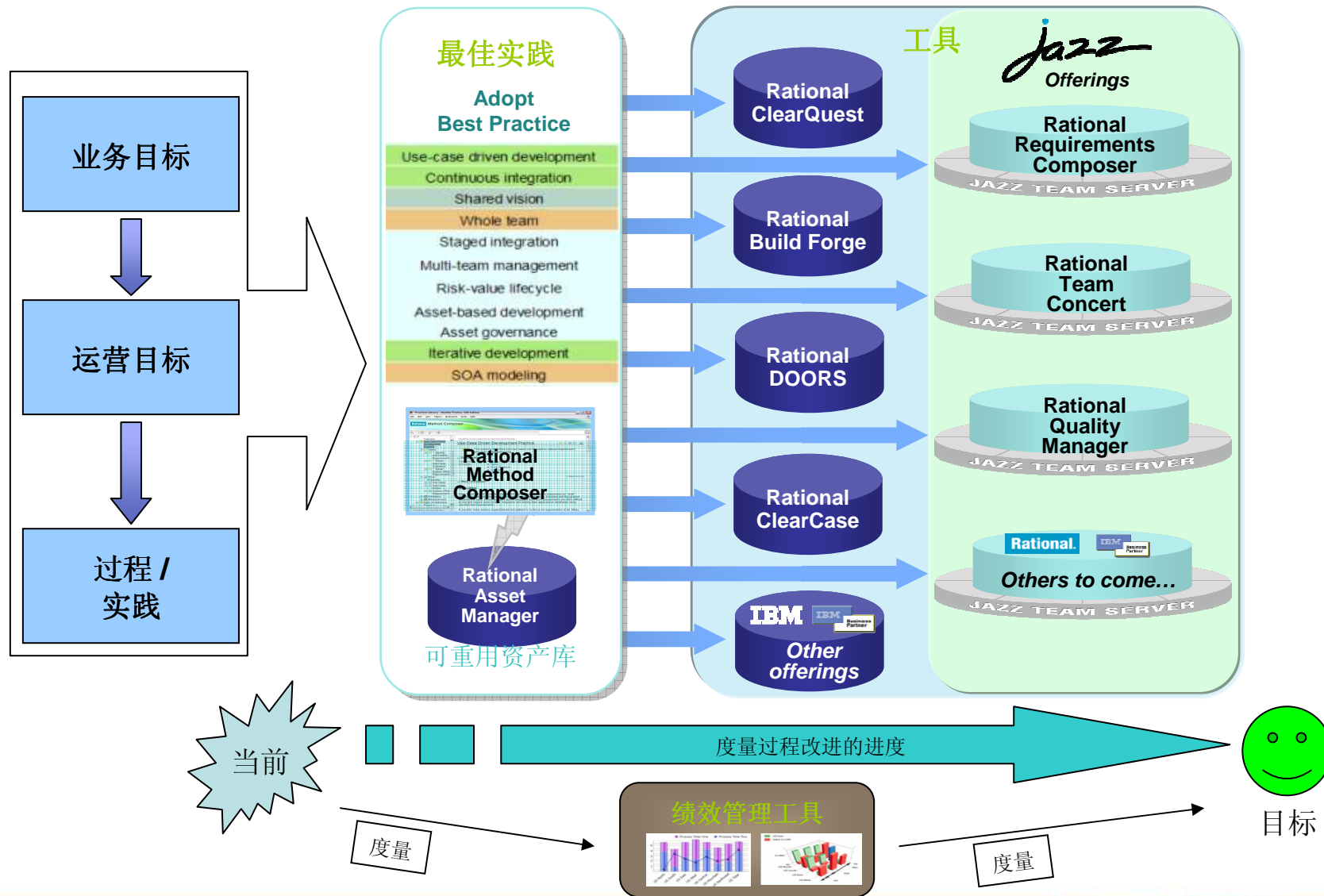
Phase 4

基于结果进行报告、分析并采取行动

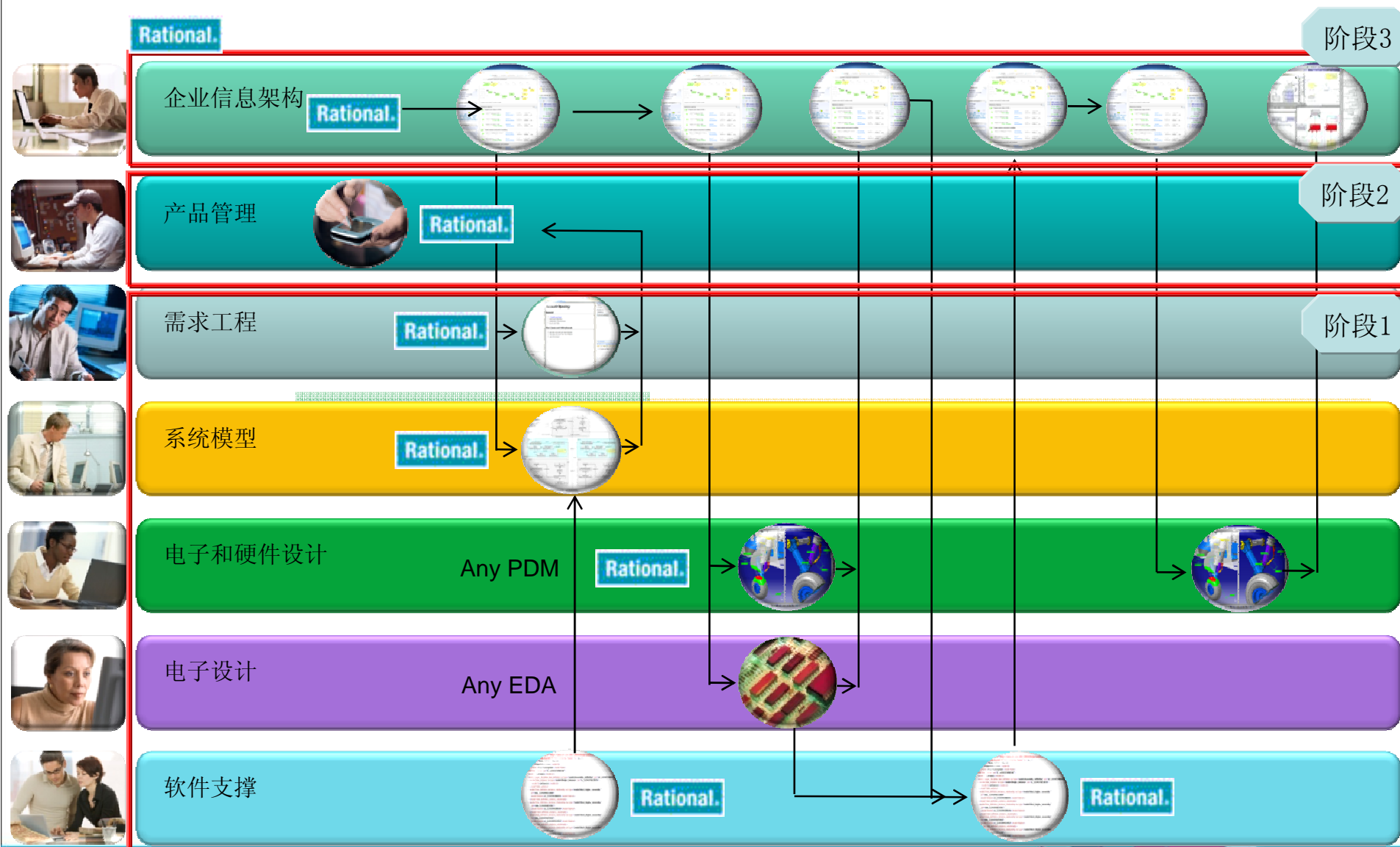
- IBM Rational Insight工具
- IBM Rational Self Check for Software Teams

开始下一次增量的改进

MCIF: 一种软件交付能力与绩效改进的方法

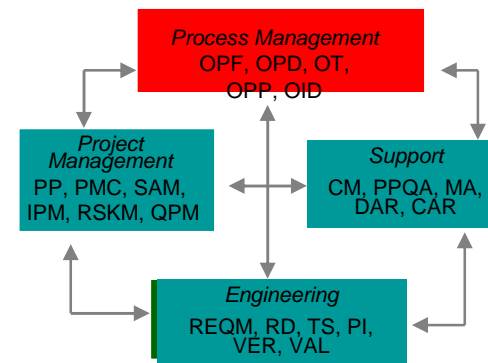


IBM Rational针对CMMI的PI的解决方案目标分解

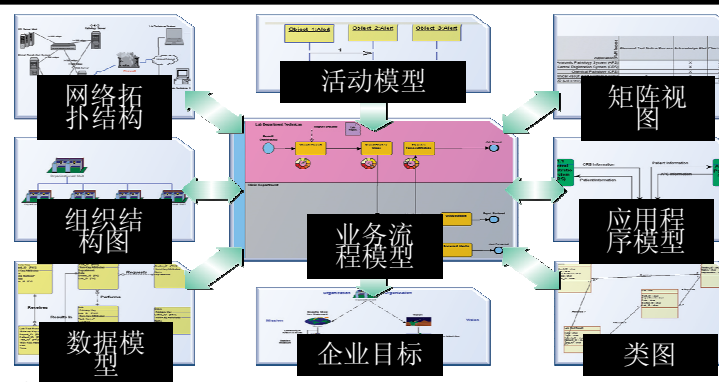


企业信息架构管理

Solution Enablement Software	Required	Recommended	Optional
Rational SA	X		
RMC/MCIF			

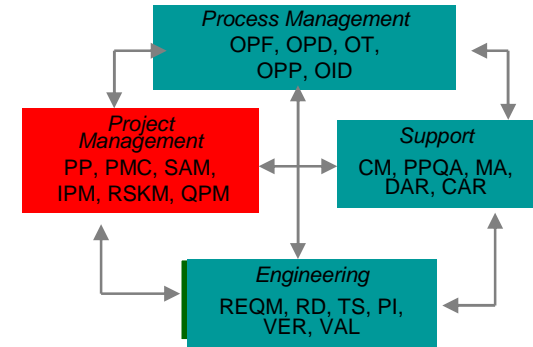


<p>解决问题</p>	<p>如何让CMMI支撑业务流程 如何规划CMMI PI 如何实施CMMI 如何规划和实施PI数据以适应多个系统和应用中 如何基于CMMI让Cx0 level和企业各个层面互联互通</p> <p>简单说： 不能掌握ITPI（如业务、流程、应用、数据）的现状以及相互关系， 导致ITPI投资浪费； 不能建立ITPI和战略、过程的关联，导致ITPI投资不能满足战略发展需要</p>
<p>解决方案</p>	<p>应用Rational企业架构管理解决方案，Rational System Architect。</p> <p>SA通过建立整体PI架构模型，管理海量、复杂的PI架构信息，并通过建立不同架构（业务、应用、数据、技术）之间的关联关系来全面的展现PI架构资产。</p> <p>SA通过结构化的数据管理架构信息，便于进行架构级的PI统计分析（应用覆盖分析、应用变更影响分析）。</p> <p>SA通过信息发布建立整体战略层次、整体架构层次和具体解决方案的沟通渠道。可以更好的驱动和约束项目，并及时的得到项目的反馈。</p>



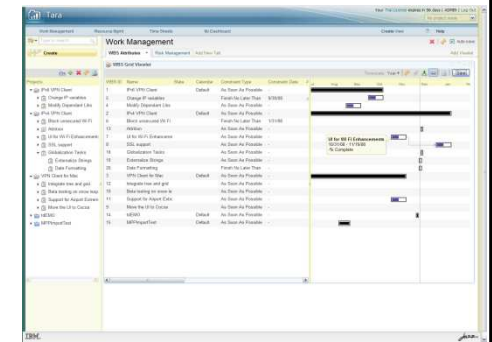
产品和项目管理

Solution Enablement Software	Required for PDIF	Recommended	Optional
Rational RPC/RMC	X		
Rational Focal Point		X	



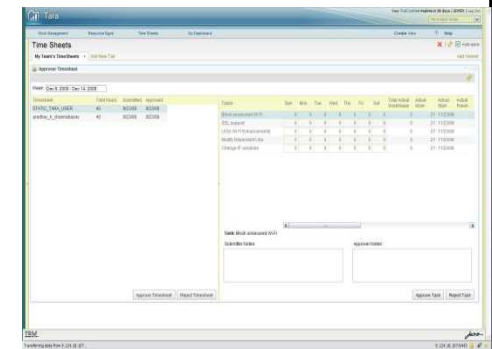
解决问题

- 缺乏基于CMMI L2-L5的项目级和企业级项目管理方法
- 如何项目里程碑和关键节点进行有效控制，实现有效执行和监控
- 如何对企业重要资源进行有效管理
- 如何进行项目组合管理，确保投资与业务保持一致
- 如何提高组织过程资产重用，减少生产中的大量重复性工作

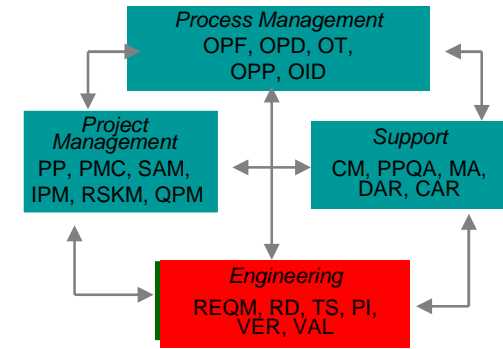


解决方案

- 推行基于CMMI的企业标准化项目管理流程和最佳实践经验；
- 把想法、提案发展为可测量的项目和项目组合
- 进行项目的计划、执行和监控；
- 基于准确的项目数据，作出及时的、准确的决策
- 主动地管理项目风险和问题
- 高度集合软件开发过程

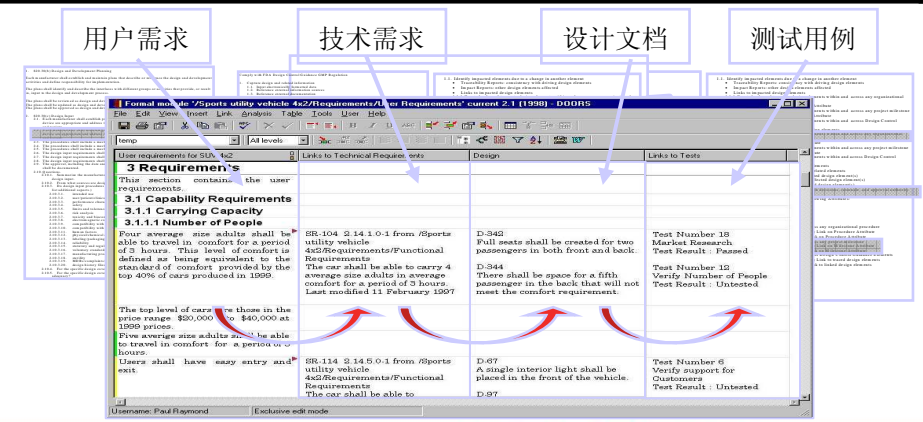


需求工程



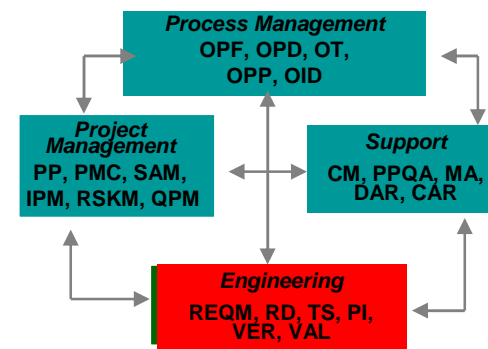
Solution Enablement Software	Required for P/DF	Recommended	Optional
Rational Requirement Composer	X		
Rational Doors	X		

<p>挑战</p>	<ul style="list-style-type: none"> •确保产品与需求之间的一致性 •构建复杂的大型系统时，如何将最初的高级客户需求细化为具体的技术规范，同时保持需求与规范之间的溯源性，以便在客户需求发生变化时，验证其一致性或促进影响分析？ •在每次发行产品后，您们如何追踪该产品对需求的满足情况，以确定应该在何时向市场或军方发布增补产品？
<p>解决方案</p>	<ul style="list-style-type: none"> •集中需求捕获、改进和跟踪以提高信息的可见性和协作性，支撑和规范RM和RD PA •按优先顺序排列需求，并使结果直观可见，以确定需要实现的需求 •在整个开发生命周期中跟踪需求，以确保客户需求的切实满足和相关行业或政府规定的遵守

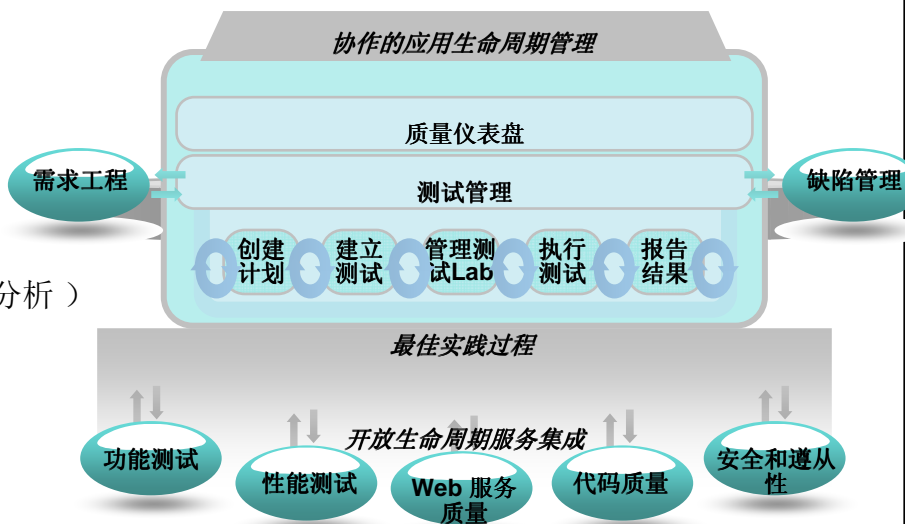


持续不断的质量保证

Solution Enablement Software	Required	Recommended	Optional
Rational TestRT	X		
Rational Logicscope	X		
Rational AppScan		X	
RQM	X		

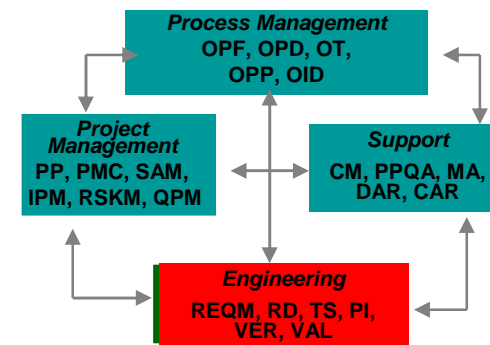


<p>挑战</p>	<ul style="list-style-type: none"> •如何有效实施产品质量确认 •如何有效实施产品质量校验 •如何助力企业中应用的复杂性环境的复杂性和流程的复杂性，制定有效的改进平台
<p>解决方案</p>	<ul style="list-style-type: none"> • 代码级测试工具（主要是开发人员使用） <ul style="list-style-type: none"> - 动态代码分析（可靠性测试、覆盖分析、性能分析） - 静态代码分析（Java为主） - 静态代码分析（C/C++为主） - 代码质量图形化评估 - 嵌入式实时软件测试 • 系统级测试工具（主要是测试人员使用） <ul style="list-style-type: none"> - 性能测试 - 自动功能测试 - 手工功能测试 - SOA功能与性能测试



模型驱动开发

<i>Solution Enablement Software</i>	Required	Recommended	Optional
Rational Rhapsody	X		
Rational Harmony	X		
RTC	X		



挑战

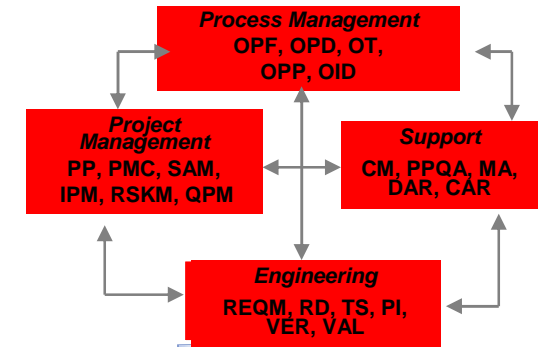
- 系统的开发向着大型化、复杂化的方向发展
- 今天的嵌入式开发的复杂度上升迅速,功能越来越多,系统组件越来越少
- 由于产品中软件含量越来越高,所以增加了完整的产品需求定义的难度.
- 现在更为需要能够松散耦合的系统组件,这些组件组装起来能够能够紧密地集成成为一个系统,而且能够紧密地协作

解决方案

- 1.理解系统在其工作环境中的工作模式。将系统当作黑盒子,找出系统的交互对象、实际交互与结果。——上下文层次
- 2.决定如何初步将系统分解成为相互交互的元素。递归地从黑盒透视图到白盒透视图地检验每个元素直到全部的关键功能和非功能需求得到满足。——分析层次
- 3.考虑系统元素如何被实现 ——设计层次
- 4.系统被构建、测试,准备发布 ——实现层次

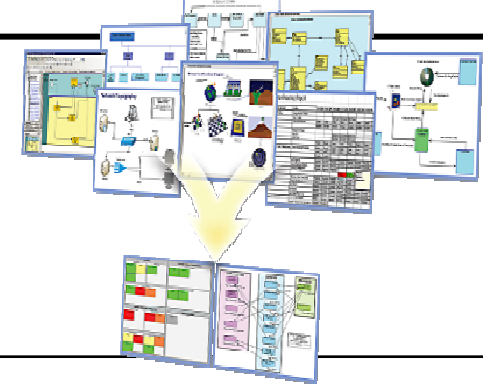
项目度量和统计

Solution Enablement Software	Required	Recommended	Optional
Rational Insight	X		
Rational RTC/CC/CQ/RQM/...		X	



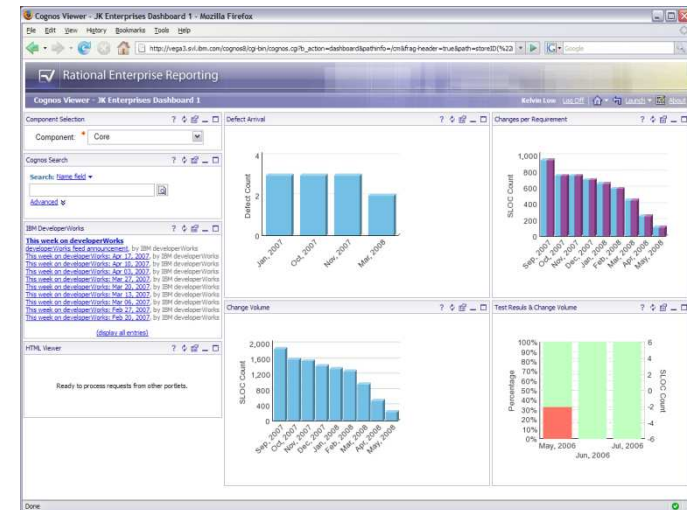
挑战

- PI中如何架构度量体系
- 对开发部门以及项目缺乏各个视角的度量手段和评估标准
- 缺乏风险控制，决策支持往往缺少依据
- 需要定位过程改进机会，针对项目衡量改进成果
- 项目执行过程中各个软件工程平台的信息无法整合



解决方案

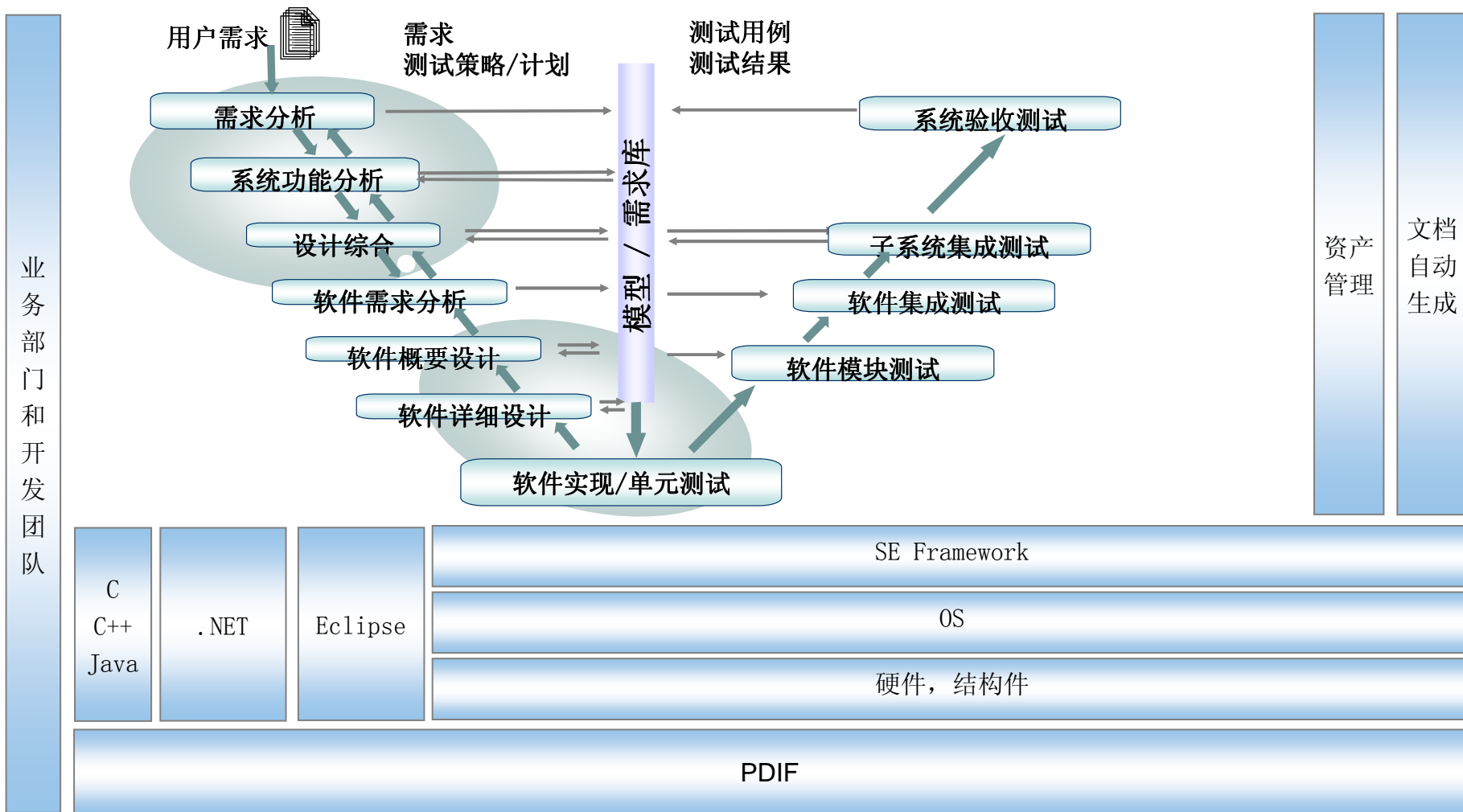
- 提升整个部门和项目开发过程的透明度，降低成本，防范风险，为正确决策提供客观依据
- 衡量流程和各种实践的效率，从而改进组织的投入产出
- 提供跨地域，跨组织的对开发项目的统一洞察和监控能力
- 真正的软件工程生产线的信息收集、度量决策工具，基于Jazz架构。
- 与MCIF结合，将CMMI中给出的度量实践真正落地



总结

基于CMMI的系统软件交付流程管理平台

产品和项目管理



Thank You

Innovate2010

Let's build a smarter planet.

