IBM Software Group

有效进行软件缺陷的管理和分析

---正交缺陷分类(ODC)在CQ中的应用

Rational. software

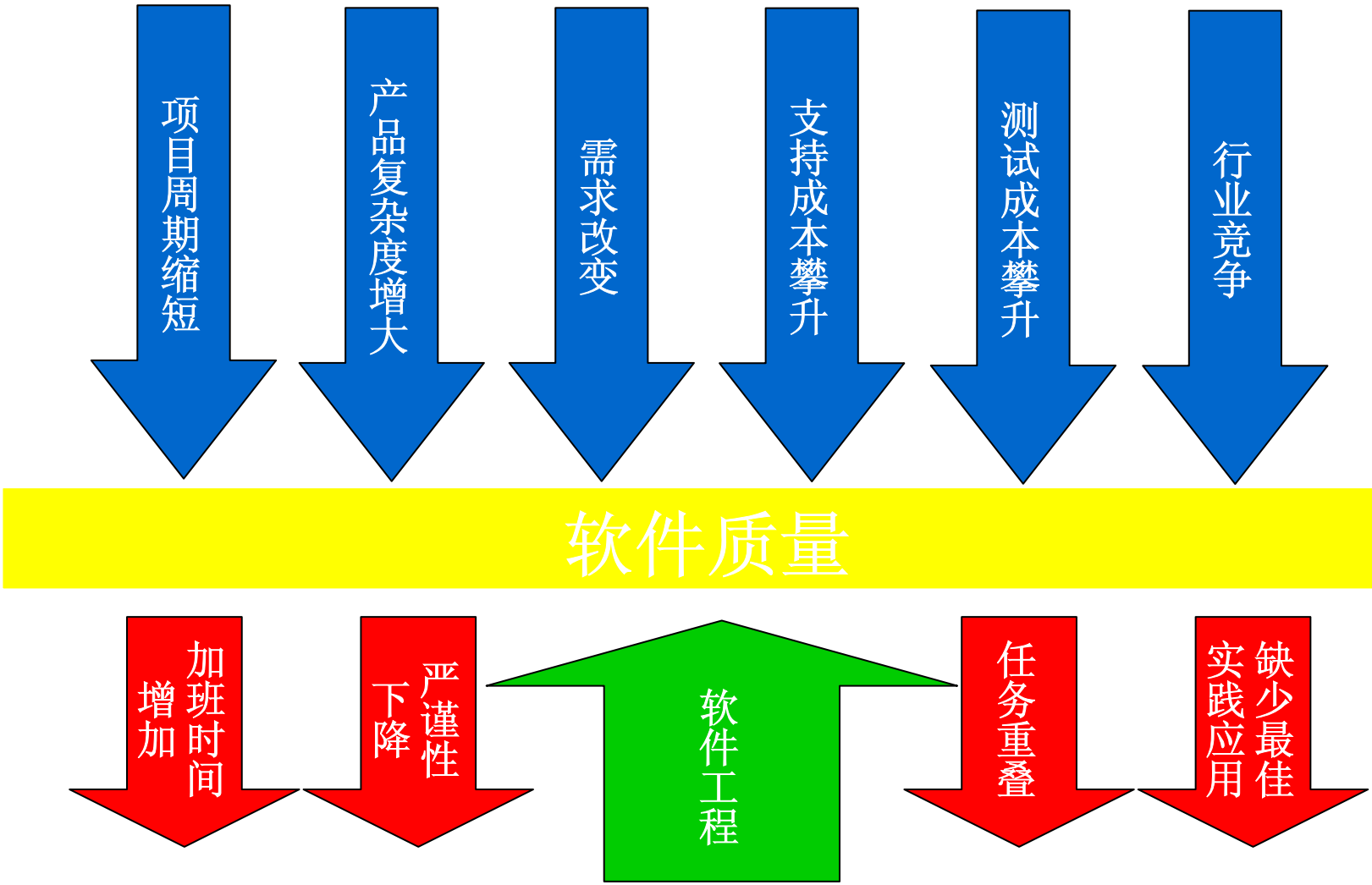
IBM软件部 李明
limingsw@cn.ibm.com

ON DEMAND BUSINESS

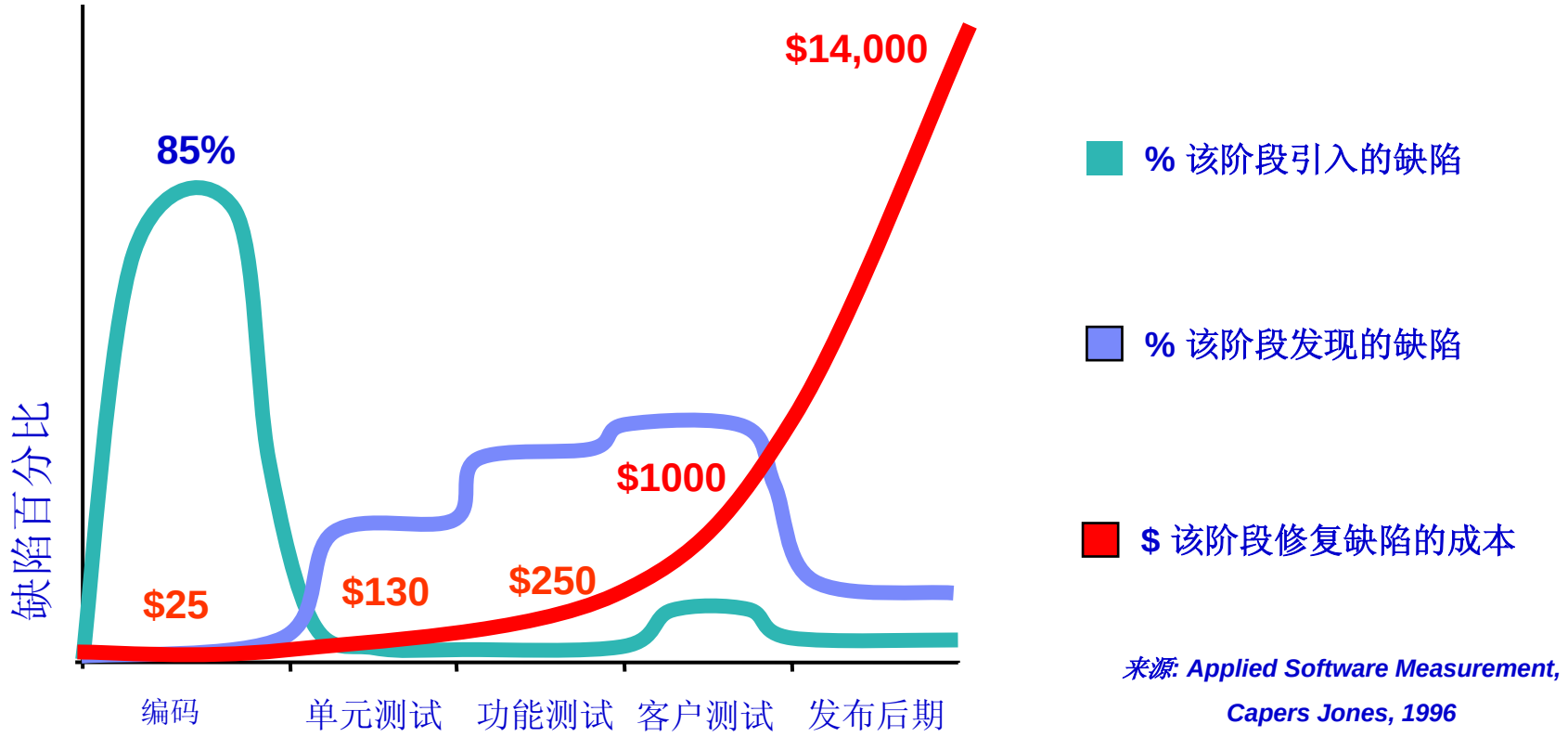
议程

- 软件软件质量现状和缺陷分类方法
- 缺陷正交分类方法、应用和流程
- 如何在**Rational ClearQuest**中使用正交缺陷分类技术

软件质量所面临的压力


项目周期缩短
产品复杂度增大
需求改变
支持成本攀升
测试成本攀升
行业竞争
软件质量
加班时间增加
严谨性下降
软件工程
任务重叠
缺少最佳实践应用

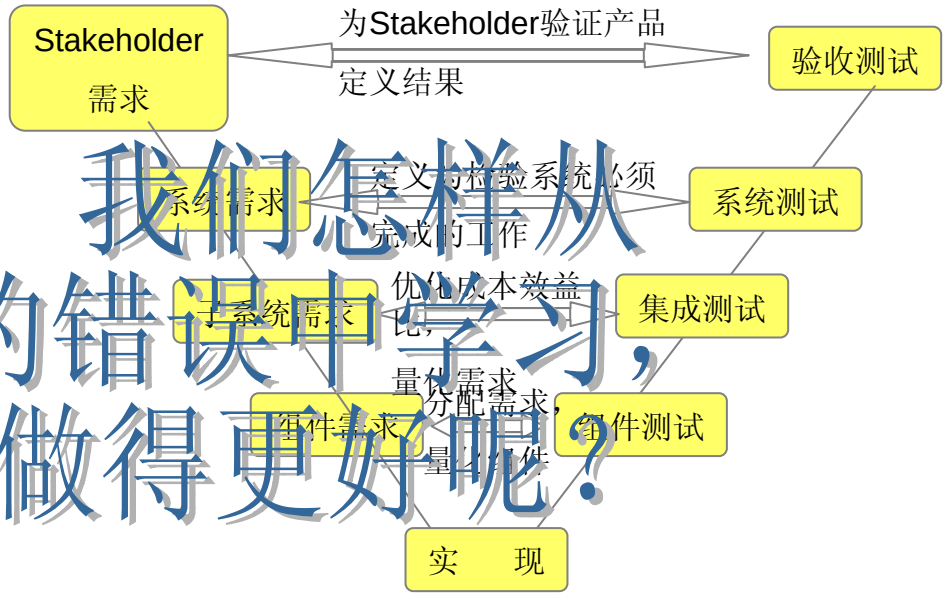
软件质量低下的代价



来源: Applied Software Measurement, Capers Jones, 1996

如何保证质量-流程

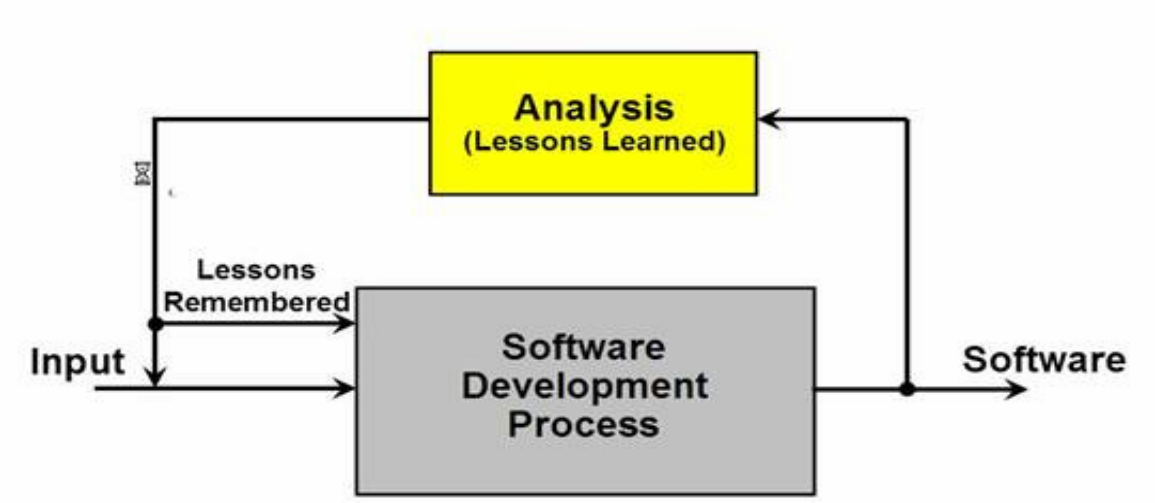
- 软件开发中都包含有控制软件开发的流程。
  - 需求分析
  - 系统设计
  - 开发代码
  - 产品测试
  - 发布产品



但是，我们怎样从以前的错误中学习，怎样做得更好呢？

如何保证质量-工件

- 通常我们会将输出的数据来进行分析，以便知道应该怎么样和进行什么样的改进

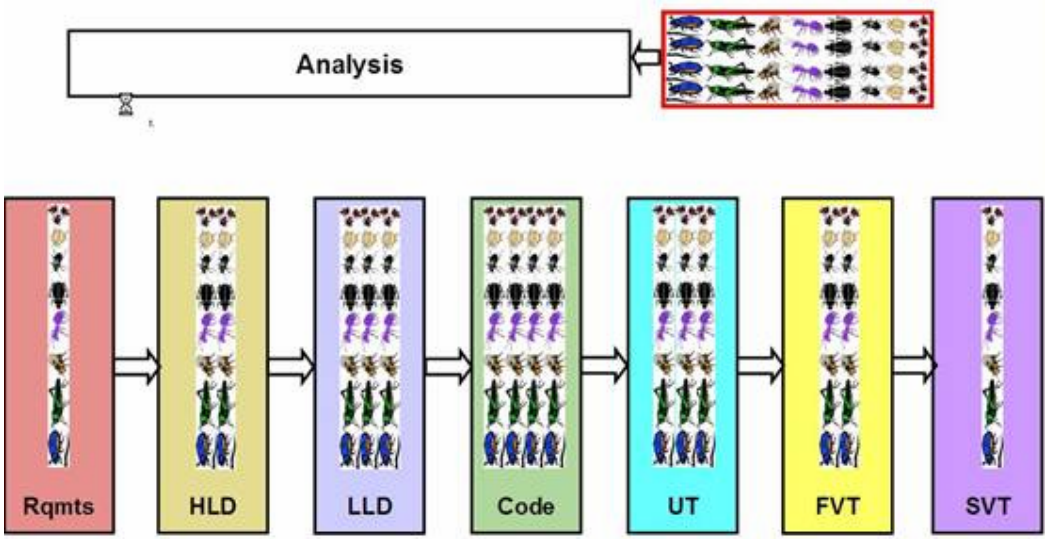


- 如何确定我们做的努力是真正有用的呢？

缺陷分类方法 (Defect Classification)

缺陷分类方法（Defect Classification）帮助改进质量

- **根源分析法(root causal analysis)**
  - 软件开发过程中我们会在不同的阶段发现数量不等的defect，对于所发现的defect我们可以逐一的对它们进行分析



这种分析方法占用了大量的时间和资源

缺陷分类方法（Defect Classification）帮助改进质量

- 缺陷 严重程度分类法
  - 在测试过程中会根据defect的严重程度对defect 进行分类，在这里将严重度称为severity
    - Severity 1：最严重的缺陷。它使系统根本不能运转，需要立即进行改正；
    - severity 2：一般功能性错误的缺陷。这些缺陷是需求中所要求的，必须改正才能实现系统完整的功能。
    - Severity 3：细小的缺陷。它不影响功能的实现，但可能引起用户的误解或者使用不当。
    - Severity 4：测试人员建议改进的地方，如果时间允许开发人员可以选择性的改正，或者等到下个版本中再改进。

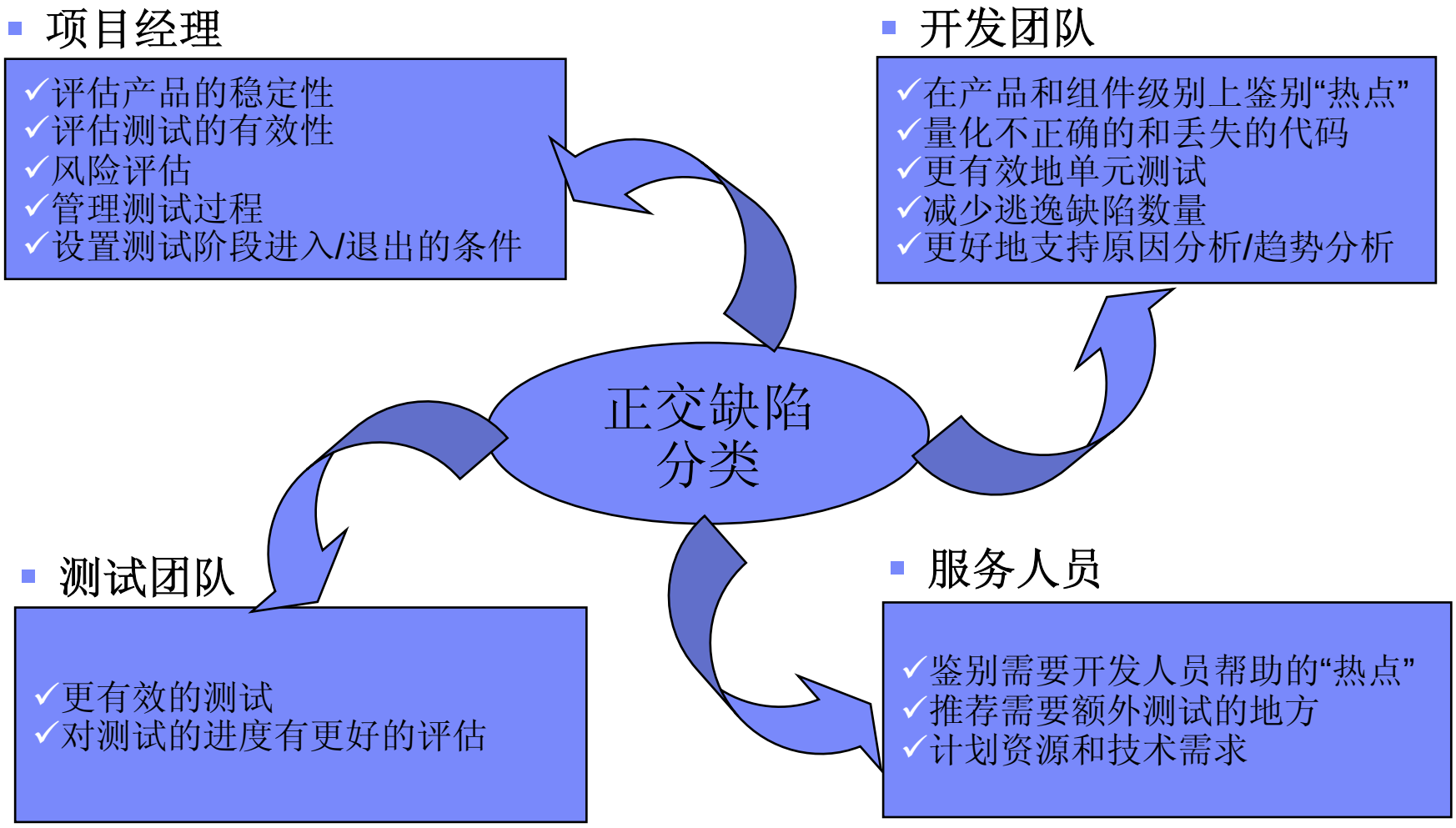
不知道每个缺陷引起的原因和无法评估这些员工的工作效率

缺陷分类方法（Defect Classification）帮助改进质量

- 正交缺陷分类法
  - ODC在高层次上，是帮助获取缺陷信息的一个缺陷分类方案。
  - 它不仅仅是一个分类方法，ODC是一个软件过程的度量系统，它是建立在包含于缺陷流中的语义信息基础上的。
  - 它可以帮助我们评估测试的效力和效率，可以进行错误跟踪，通过ODC背后的分析机制评估顾客的满意度。

- 正交缺陷分类法适用对象
  - 开发生命周期相对来说是一个很漫长的过程，包括后续的改进工作。例如，这个项目包括多个软件版本或者一个版本有多次迭代。
  - 潜在的缺陷数目是相当大的。缺陷数目越多，客观的分析结果也越多，对了解软件质量越有好处。
  - 这个项目已经将“高可靠”设定为它的主要目标之一。

正交缺陷分类的好处


项目经理
✓评估产品的稳定性
✓评估测试的有效性
✓风险评估
✓管理测试过程
✓设置测试阶段进入/退出的条件
开发团队
✓在产品和组件级别上鉴别“热点”
✓量化不正确的和丢失的代码
✓更有效地单元测试
✓减少逃逸缺陷数量
✓更好地支持原因分析/趋势分析
正交缺陷分类
测试团队
✓更有效的测试
✓对测试的进度有更好的评估
服务人员
✓鉴别需要开发人员帮助的“热点”
✓推荐需要额外测试的地方
✓计划资源和技术需求

议程

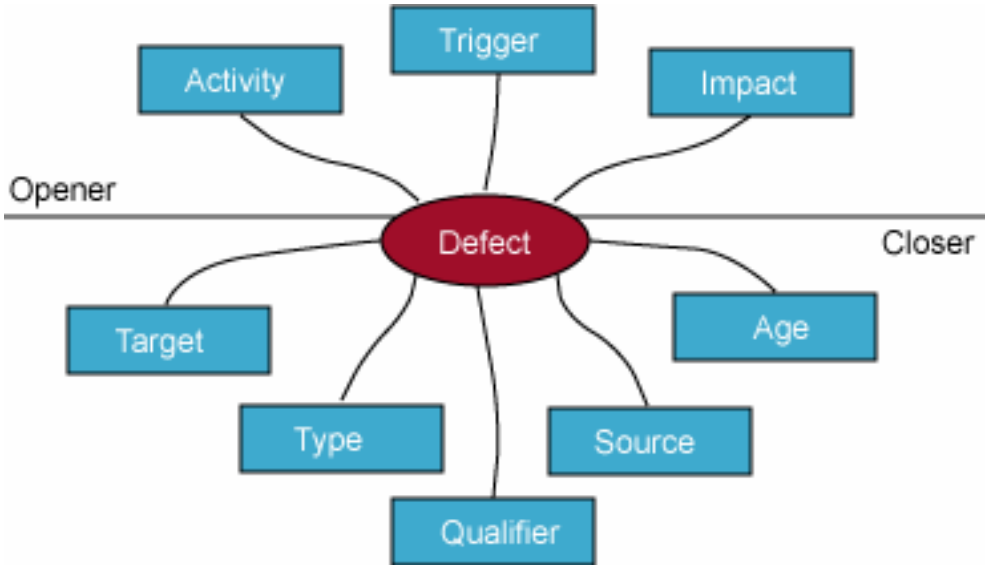
- 软件软件质量现状和缺陷分类方法
- 缺陷正交分类方法、应用和流程
- 如何在**Rational ClearQuest**中使用正交缺陷分类技术

正交缺陷分类方法、应用和流程

- **ODC通过搜集有用的信息丰富了缺陷的属性。ODC一共有8个属性：**
  - 测试人员
    - 活动（Activity）：是当缺陷被发现时实际的处理步骤（代码审查，功能测试等等）
    - 触发（Trigger）：描述了暴露缺陷时存在的环境或者条件
    - 影响（Impact）：是对用户或者是认识到的，或者是实际的影响
  - 开发人员
    - 阶段（Age）：确定可拥有这个缺陷目标（比如设计，代码，ID等等）历史
    - 来源（Source）：指引起缺陷的起因
    - 限定符（Qualifier）：指明了所进行的修复应归于缺失，错误或者还是外来的代码或者信息
    - 类型（Type）：表示所进行的实际修正的种类
    - 目标（Target）：表示被修复实体的高层特性（例如，设计，代码，ID，等等）

正交缺陷分类方法、应用和流程

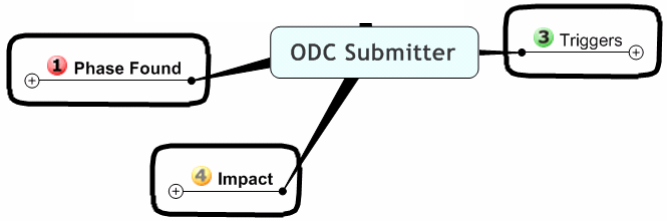
- **ODC8个属性示意图：**



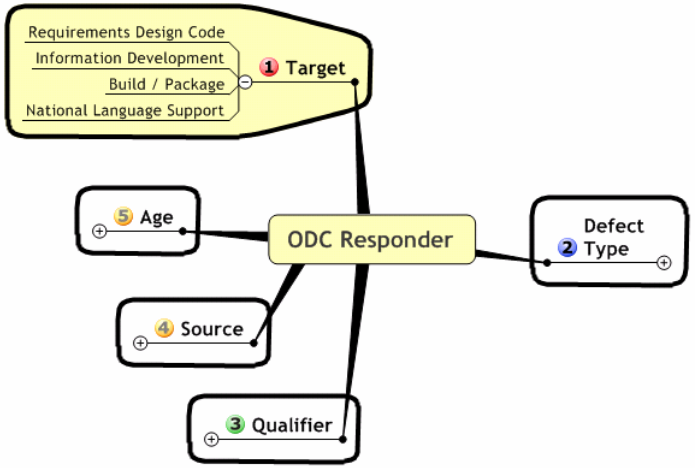
- 当测试人员发现并提交缺陷时，可以给这个缺陷分配“活动（**Activity**）”、“触发（**Trigger**）”、“影响（**Impact**）”这三个属性。
- 当开发人员修复或者回应了缺陷时，可以分配“阶段（**Age**）”、“来源（**Source**）”、“限定符（**Qualifier**）”、“类型（**Type**）”以及“目标（**Target**）”这些属性。

正交缺陷分类方法、应用和流程

- 当测试人员发现并提交缺陷时，可以给这个缺陷分配“活动（**Activity**）”、“触发（**Trigger**）”、“影响（**Impact**）”这三个属性。

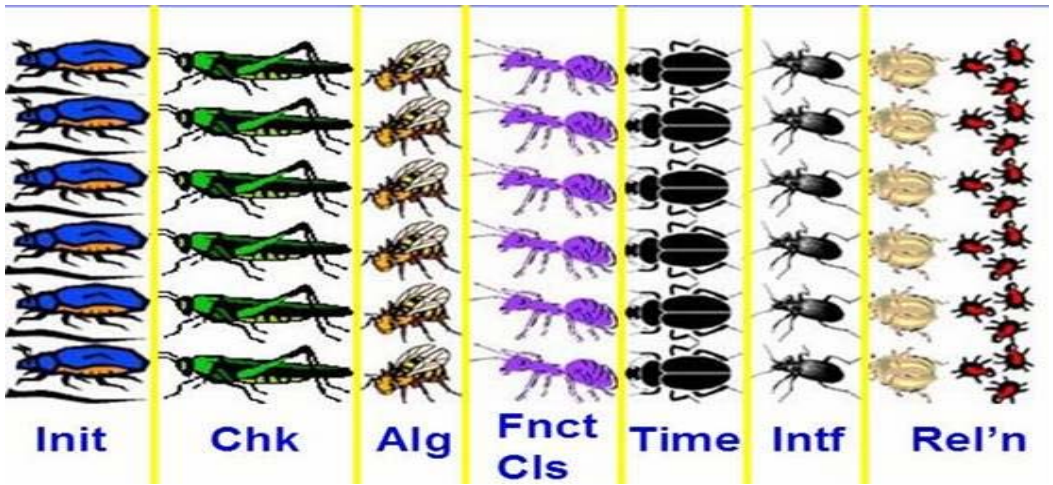


- 当开发人员修复或者回应了缺陷时，可以分配“阶段（**Age**）”、“来源（**Source**）”、“限定符（**Qualifier**）”、“类型（**Type**）”以及“目标（**Target**）”这些属性。



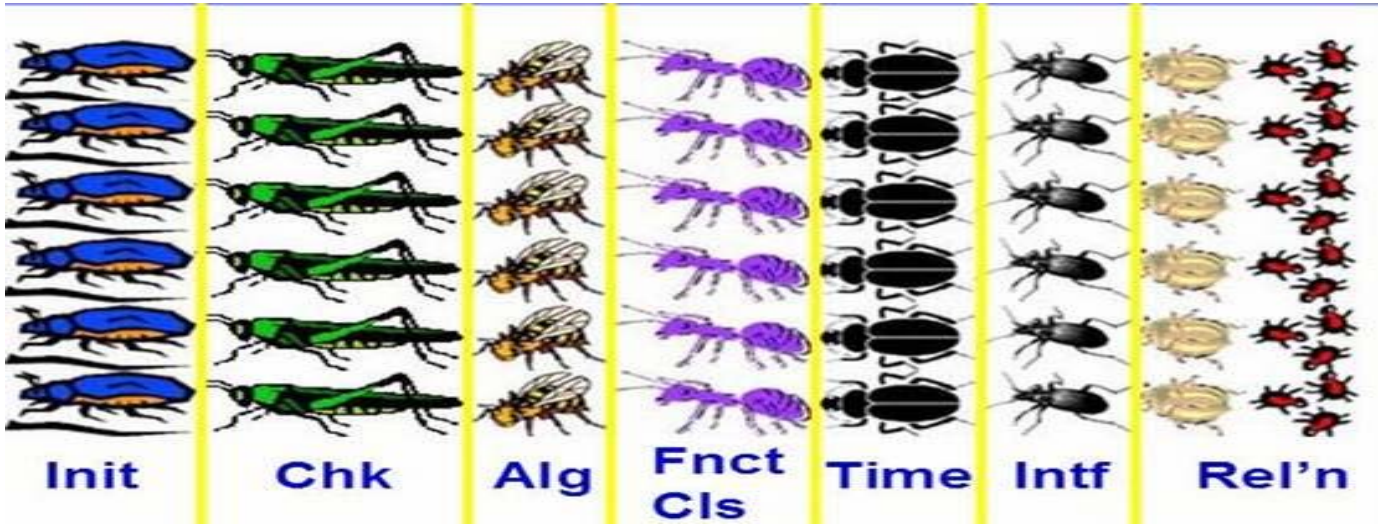
正交缺陷分类方法、应用和流程

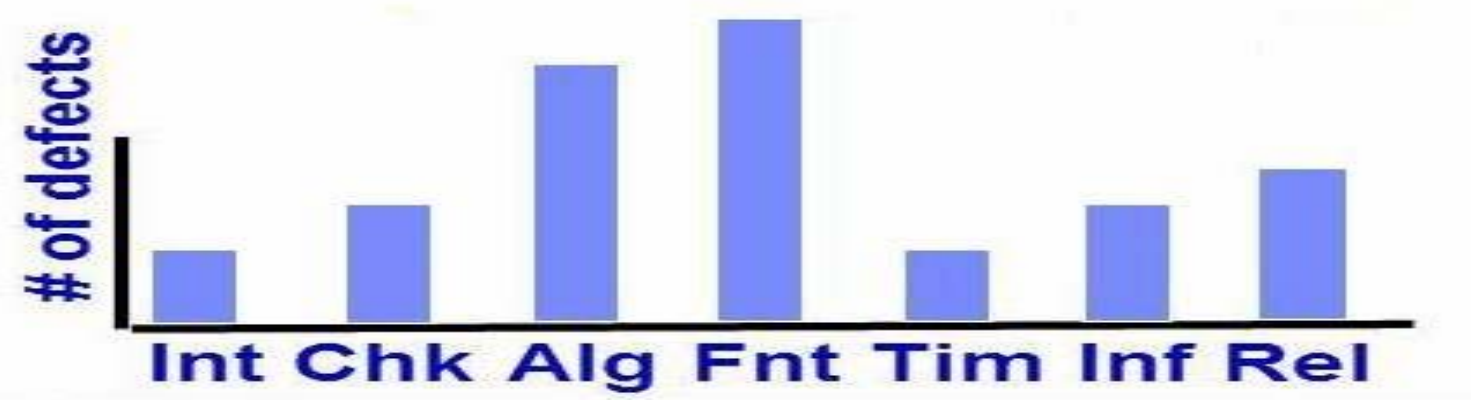
- 开发中应用**ODC**
  - 开发人员发现问题如果按类型（type）分类可以有如下选择：
    - 没有正确的初始化 （Init）
    - 代码没有正确的check-in （Chk）
    - 算法问题 （Alg）
    - 功能性的错误，可能是模块内的功能没有被正确实现，也可能是模块与模块之间相联系的部分没有被正确实现。（Fnct Cls）
    - 有可能是有关时间的错误 （Time）
    - 界面相关的错误 （Intf）
    - 代码之间相关联的错误，例如错误的继承关系 （Rel'n）



正交缺陷分类方法、应用和流程

- 开发中应用**ODC**



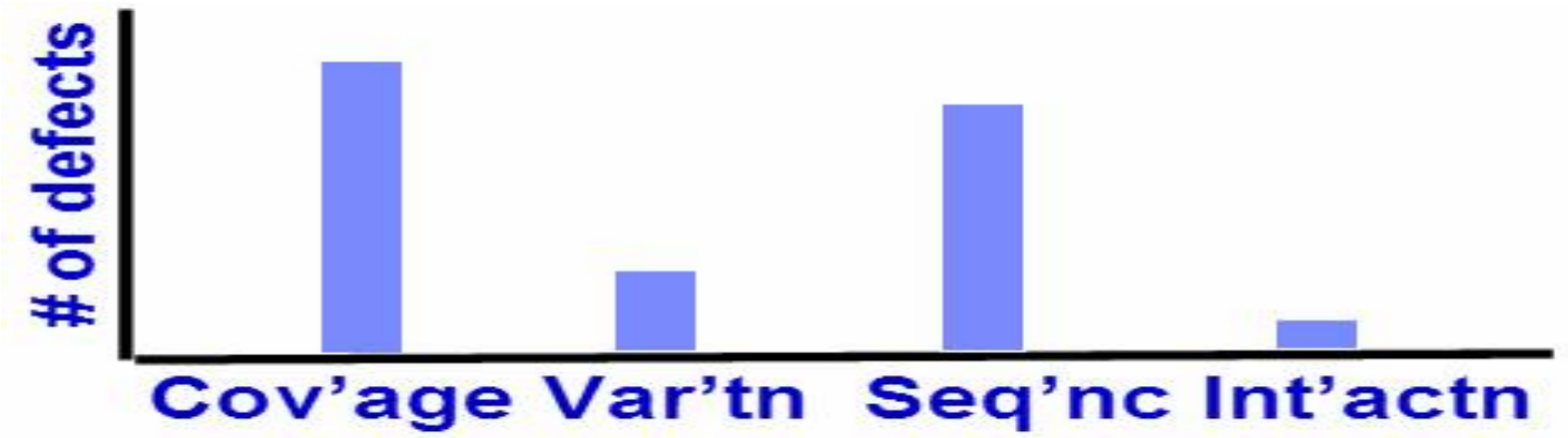


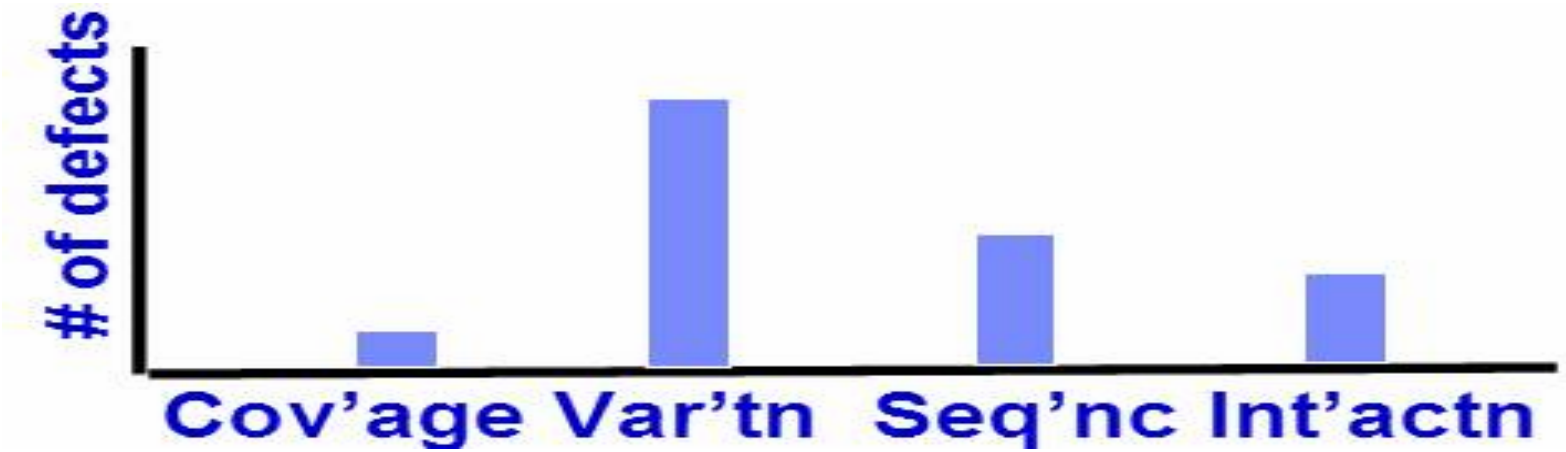
正交缺陷分类方法、应用和流程

- **功能测试中应用ODC**
  - 在FVT（功能测试）中，Trigger是帮助FVT做得更好的重要指标。
  - 在ODC中trigger可以简单的理解为是什么样的测试发现了这个defect。
  - 在FVT中可以定义4个trigger：
    - Coverage （是指normal function, 是任何用户都会用到的功能，基本的、简单的功能）；
    - Variation （有些对产品比较熟悉的用户，有可能会愿意用不常用的有创造性方法或者输入来完成同一种动作或者功能，或者单单就是为了挑错，在这些尝试中往往会发现很多漏掉的defect）；
    - Sequencing （用和以前不同的操作流程来完成一种任务功能)；
    - Interaction （当两个或者多个功能模块互相交互时可能会发生一些错误）；

正交缺陷分类方法、应用和流程

- 功能测试中应用**ODC**

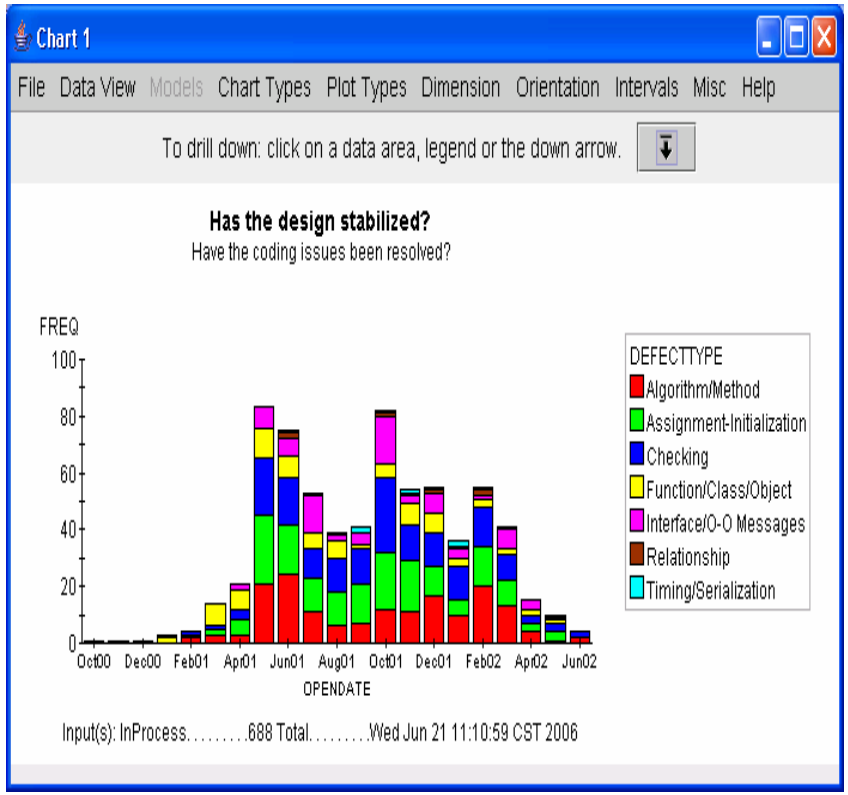




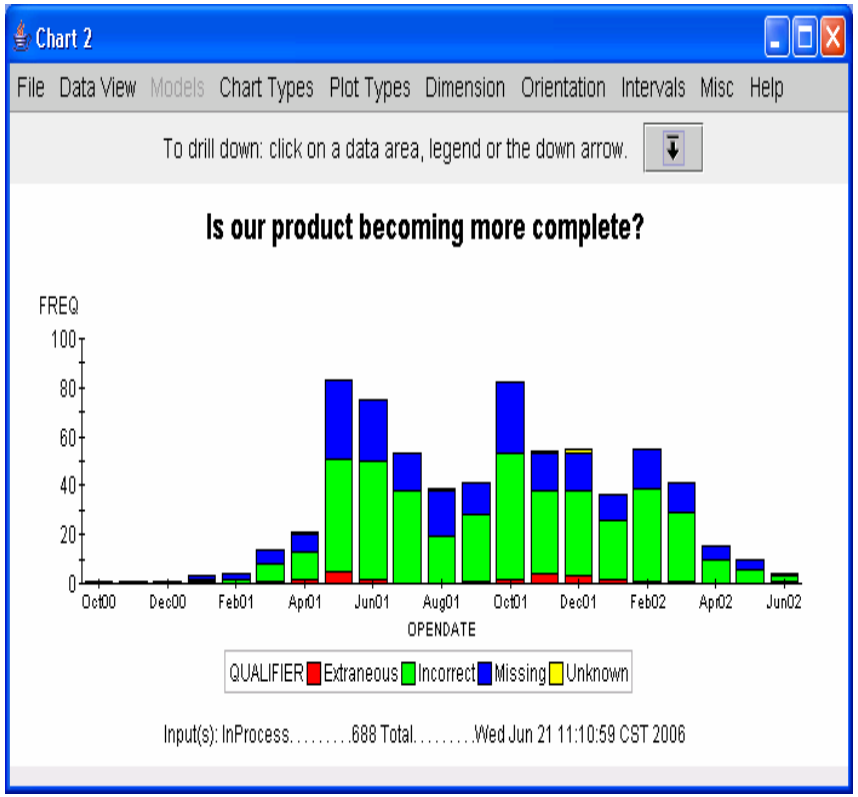
正交缺陷分类方法、应用和流程

- 评估测试有效性

1. 软件产品的设计是否已经稳定？



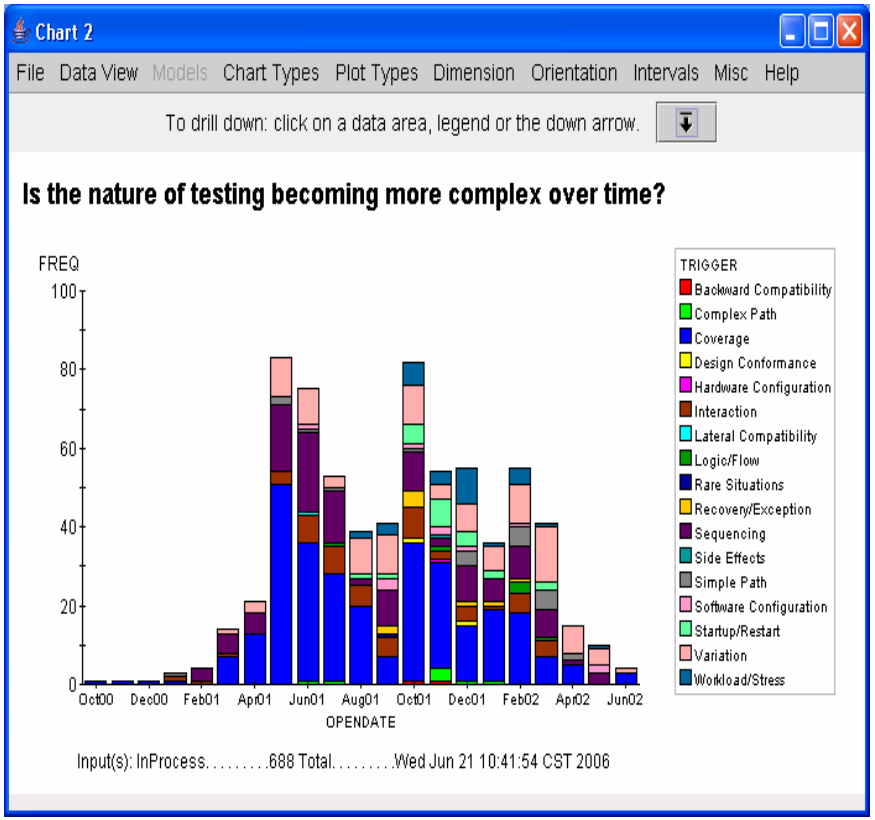
2. 软件产品的代码是否完整？



正交缺陷分类方法、应用和流程

- 评估产品稳定性

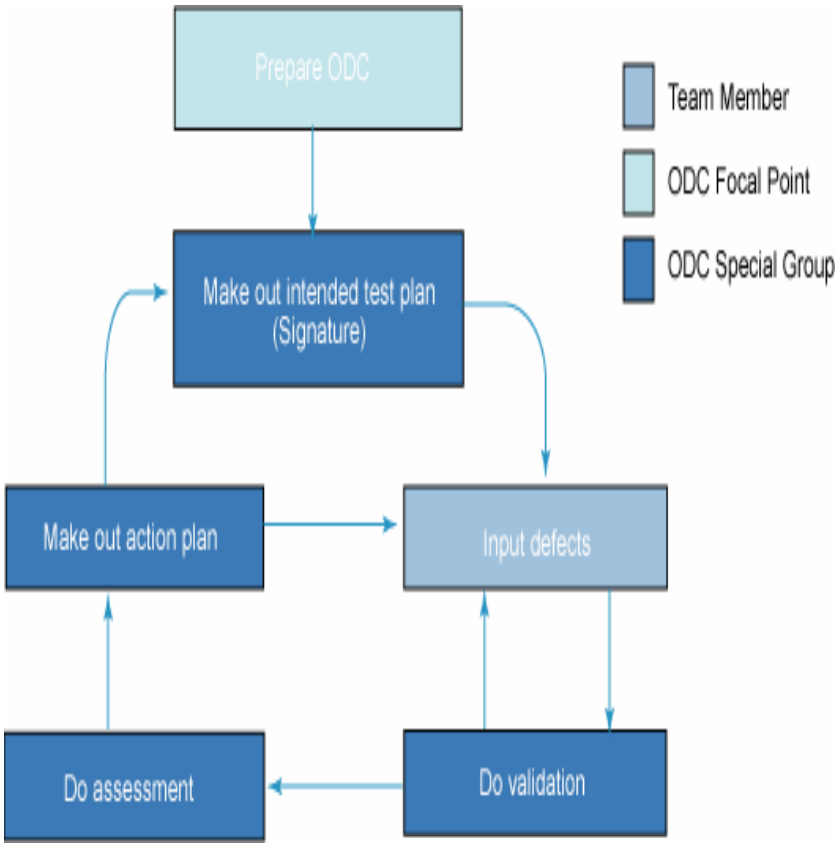
1. 测试是否使用了足够复杂的测试手段？



2. 测试中是否有逃逸的缺陷？

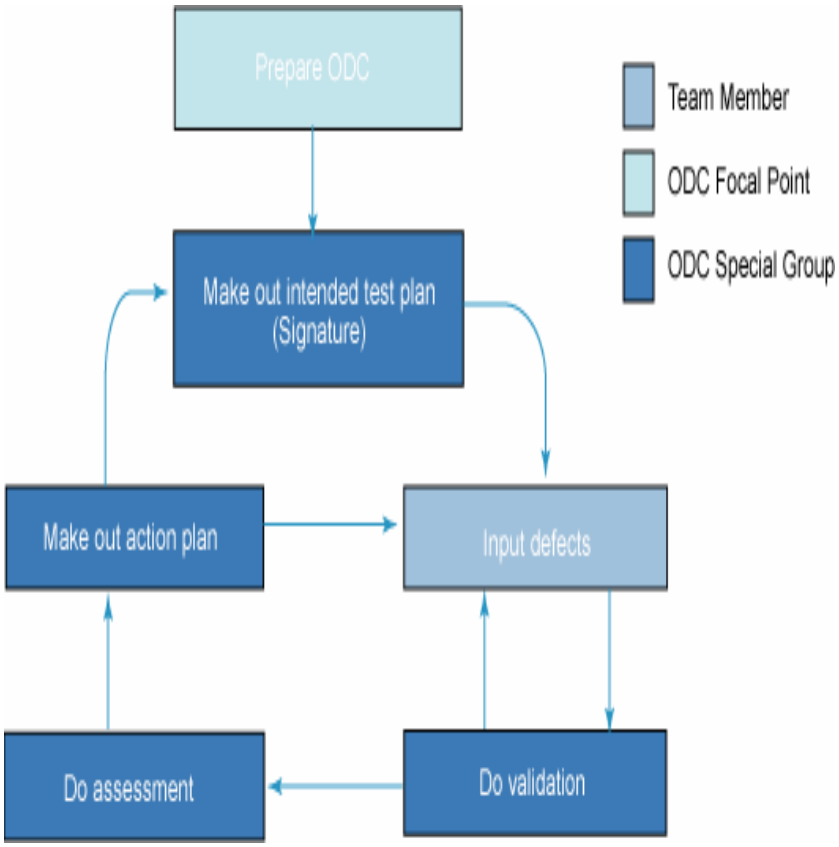
正交缺陷分类方法、应用和流程

- **三个角色：ODC生命周期包括三个不同的角色**
  - 团队成员：这是ODC中最普通的角色。团队成员就是开发者，测试人员或者用户，他们负责输入数据。
  - ODC核心：这个角色是一个ODC专家，其熟悉ODC的执行，并且可以帮助项目团队进行正确的计划和分析工作。ODC核心人物可以来自项目团队的外部。
  - ODC的特殊团体：ODC的特殊团体负责活动计划的创建、确认以及评估。这个团体是由来自不同团队的人员组成的团队。



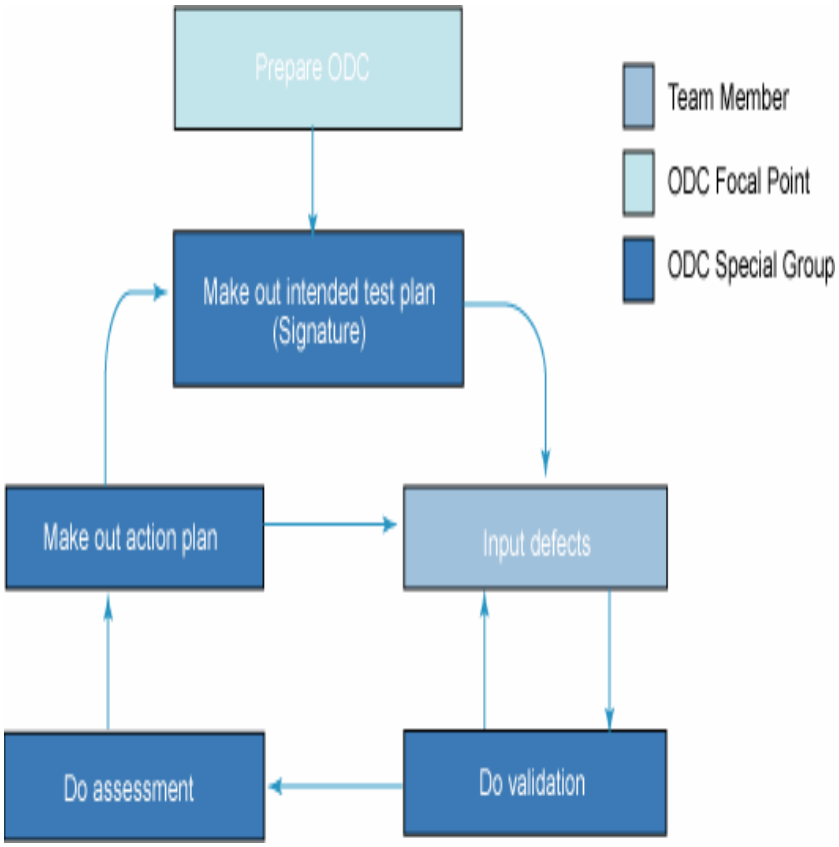
正交缺陷分类方法、应用和流程

- **三个循环：根据ODC所需的步骤的数量，它有三个可能的循环：**
  - 大循环：除了预备步骤，这个循环本身含有五个步骤。IDC计划步骤与ODC的评估是相关联的，并且它可以使评估更加有效。
  - 中等循环：它包含四个步骤，这几个步骤是ODC生命周期中的核心组成部分。尽管完整的ODC评估在这个循环中是不能得到的，一些有用的评估是可以被执行的。
  - 小循环：这个循环包含两个步骤。也就是说只要找到一定数量的缺陷，随时可能发生确认的活动。
  - ODC的实现：六个步骤



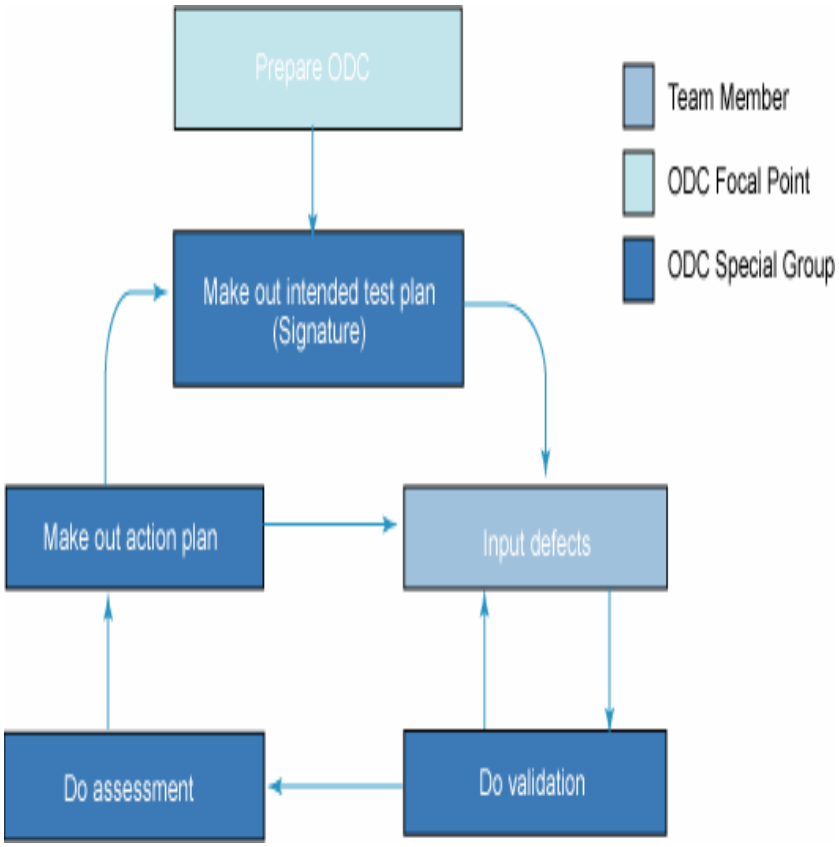
正交缺陷分类方法、应用和流程

- **步骤1：预备阶段**
  - 获得采取ODC方法操作的批准和支持
  - 采取ODC，要获得开发团队和测试团队的允许
  - 找到ODC的中心人物
  - 调查项目当前的状况
  - 给开发人员和测试人员指派ODC角色
  - 在一个缺陷跟踪工具上部署ODC计划。比如Rational ClearQuest
  - 培训



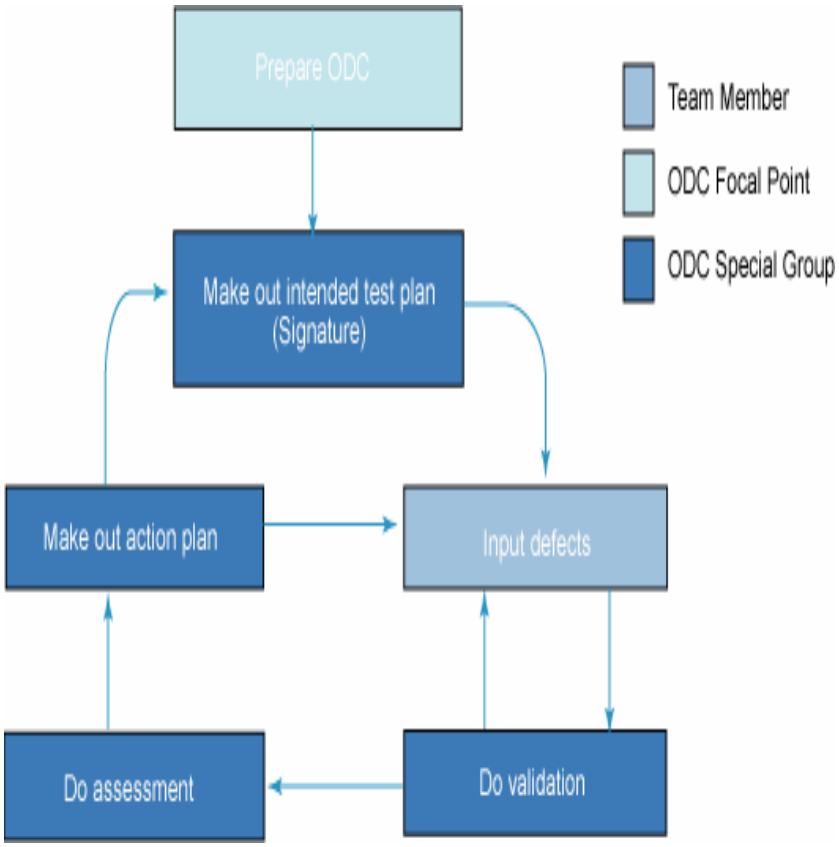
正交缺陷分类方法、应用和流程

- **步骤2：计划**
  - 确定属性
  - 将项目分成组件
  - 为ODC的评估决定项目检查点
  - 创建计划文档
  - 评审ODC计划



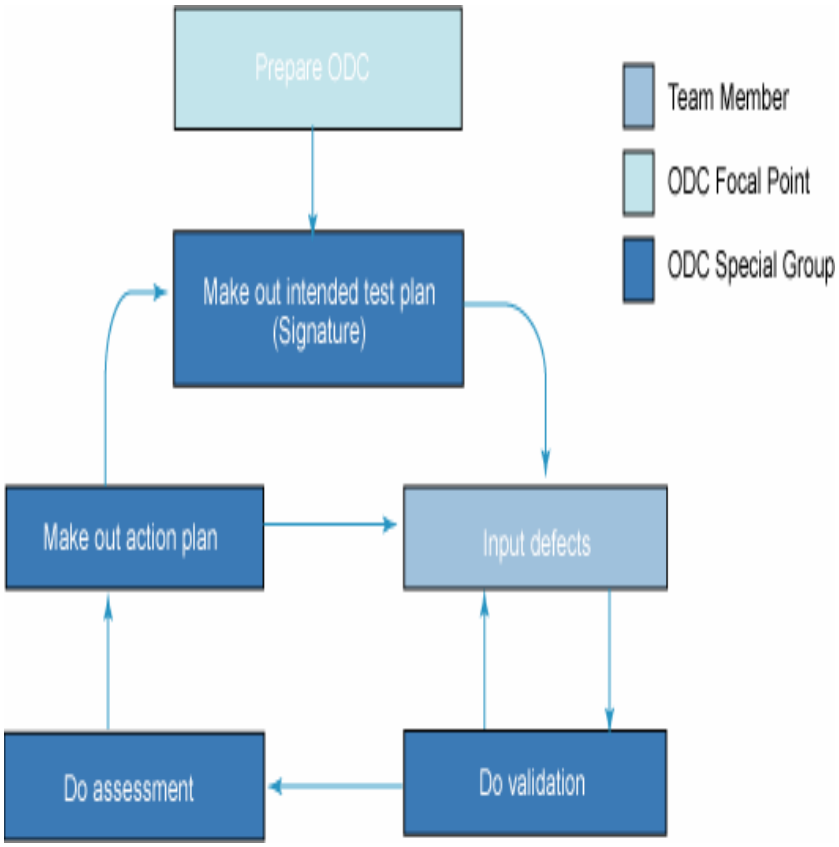
正交缺陷分类方法、应用和流程

- **步骤3和4：数据输入和确认**
  - 数据输入之前，确保所有的开发人员和测试人员都清楚地了解每个属性的含义
  - 在数据输入过程中，数据的格式应该由工具来控制。这些程序应该与缺陷状态的转换保持一致
  - 数据输入以后，需要完全的确认。



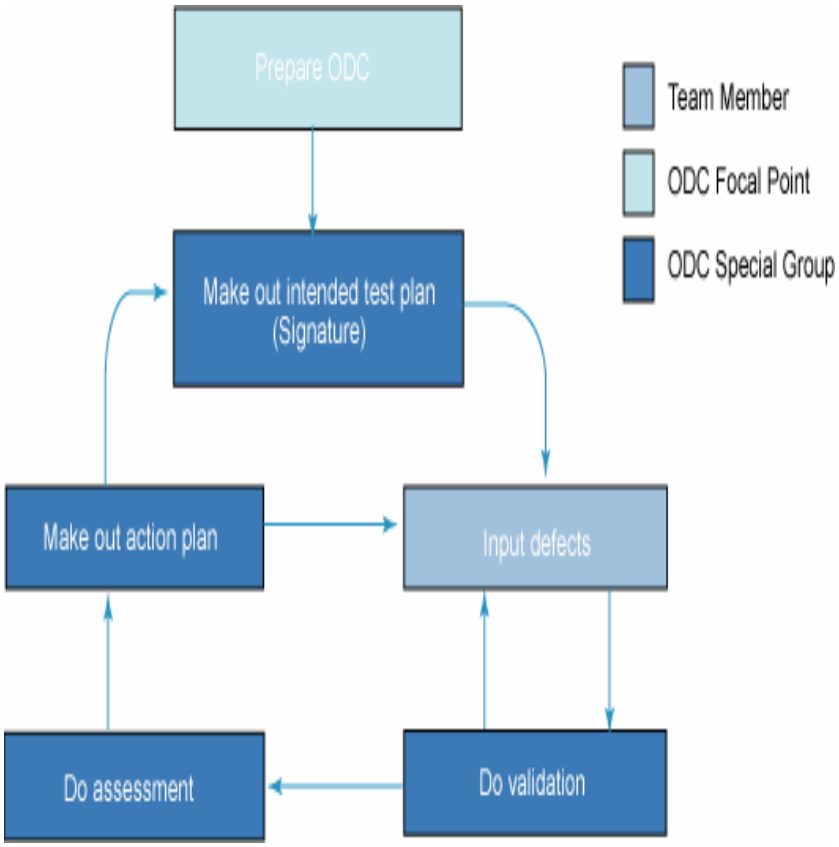
正交缺陷分类方法、应用和流程

- **步骤5：评估**
  - 这是一个收获的步骤，收集到如此多优良的数据以后，可以用下几种方法来分析
    - 选项1：生成一个ODC“退出评估报告”，并从它开始分析
    - 选项2：为这个项目生成并选择一些有意义的图表



正交缺陷分类方法、应用和流程

- **步骤6：活动**
  - 活动是最后一个步骤，这个时候应该设计一个正式的活动计划，这个计划能够帮助我们不断地提高软件质量。这个活动计划包括来自设计文档的材料，特殊代码组件，开发步骤或者测试方法。但是最重要的部分是下一次迭代或者下一次发布所要采取的活动。这些活动的目标必须清楚而且是可度量的。紧记下一个阶段中出现的问题往往是先前步骤中的错误导致的。

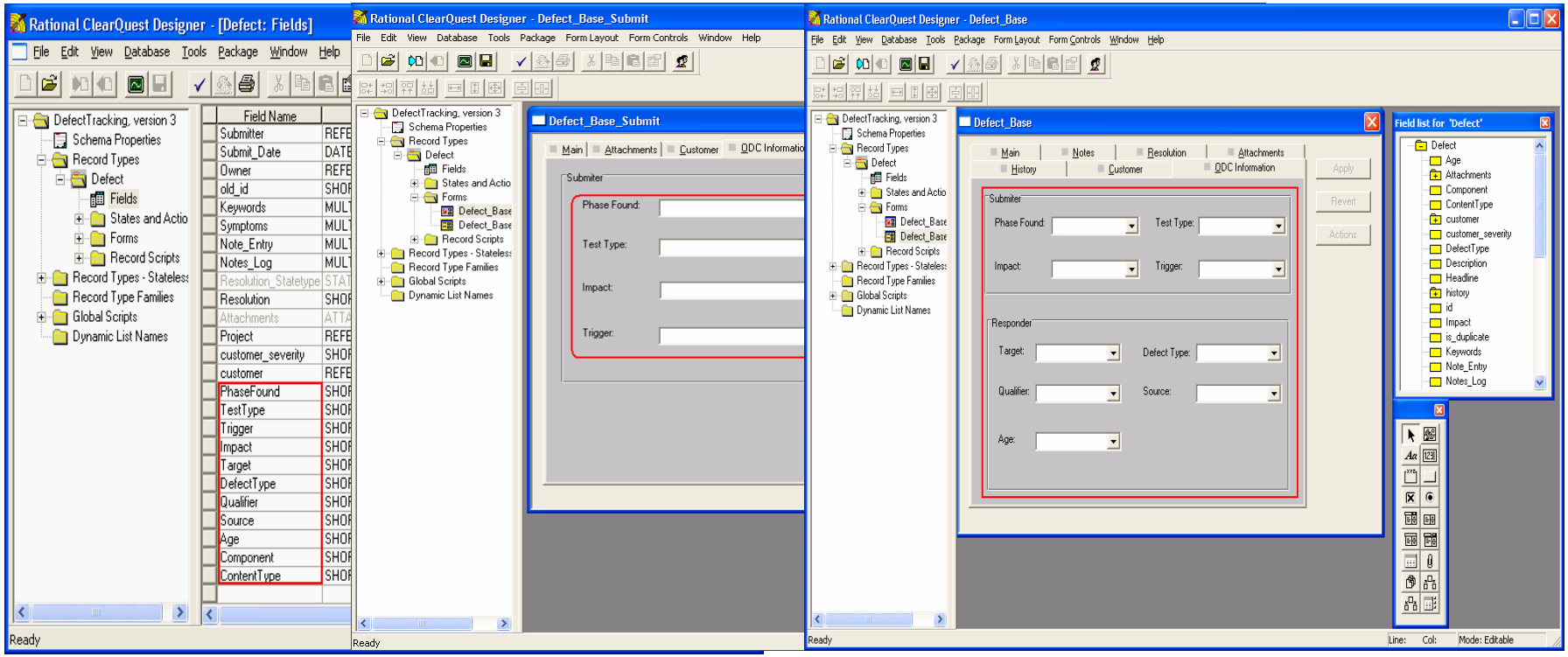


议程

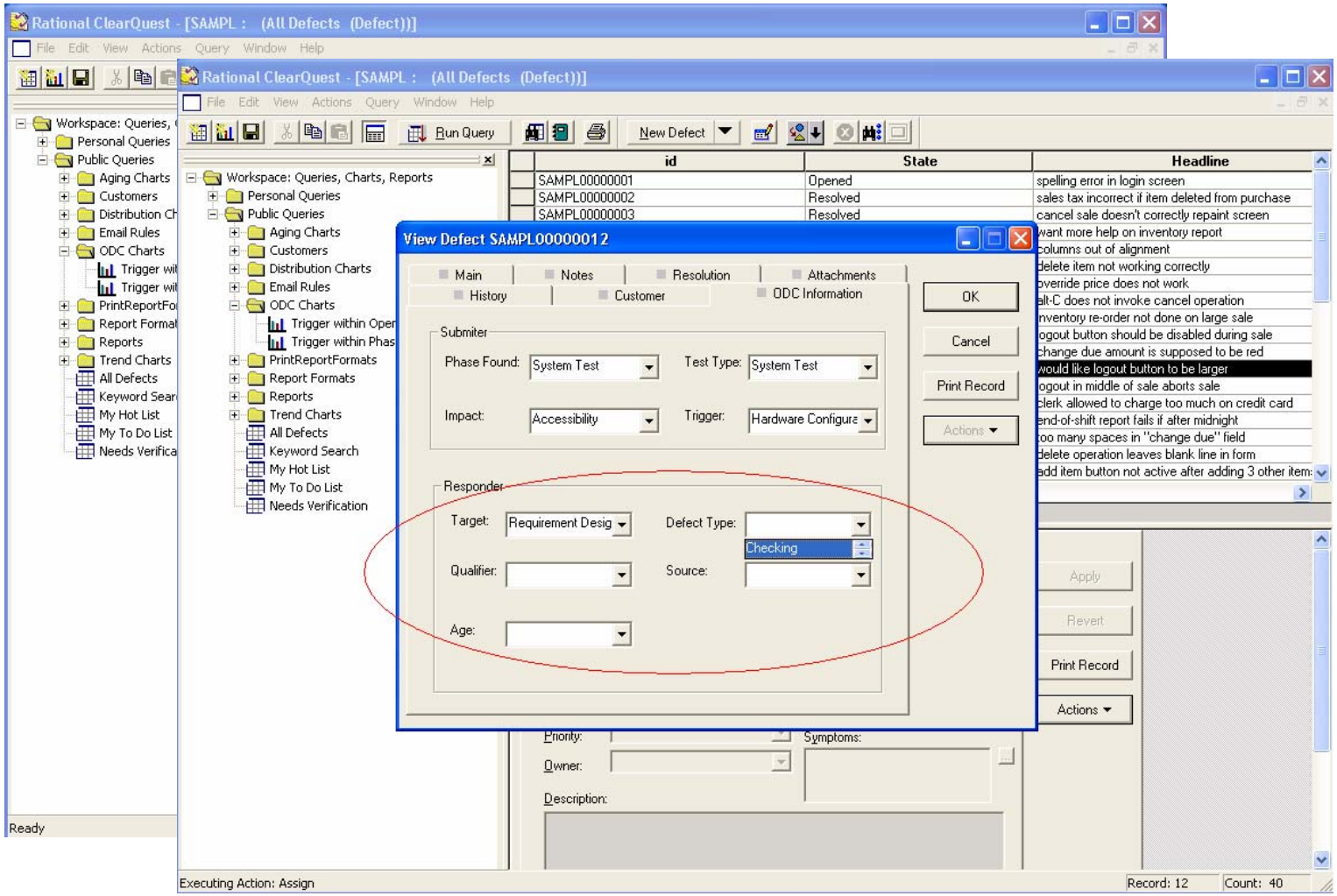
- 软件软件质量现状和缺陷分类方法
- 缺陷正交分类方法、应用和流程
- 在**Rational ClearQuest**中使用正交缺陷分类技术

定制ClearQuest的Schema来引入ODC属性

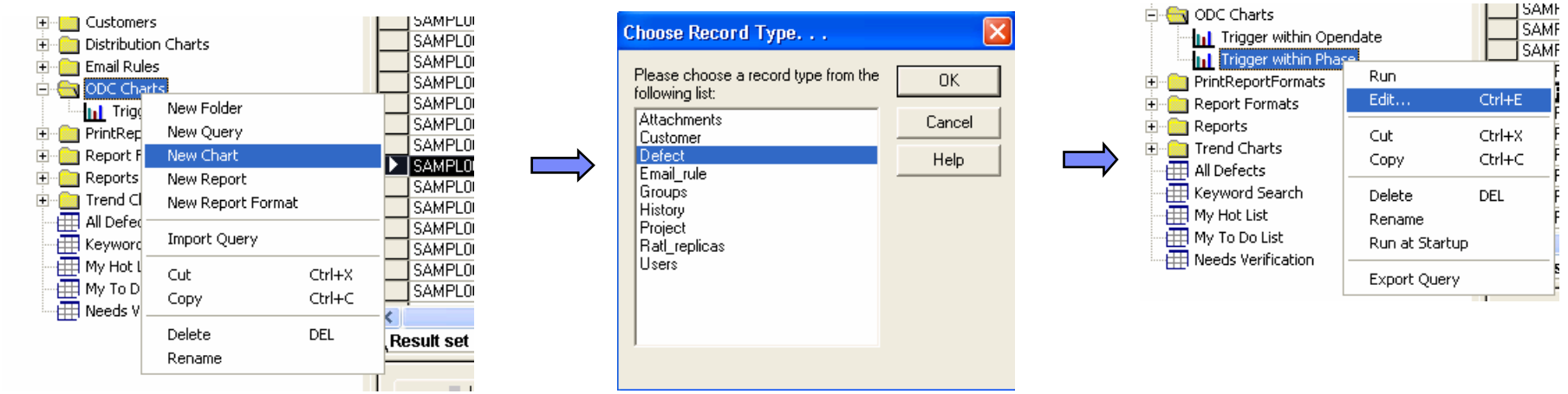
- 基于ClearQuest内置的Schema：DefectTracking
  - 字段: Phase, Test Type, Trigger, Impact, Target, Defect Type…
  - 表单: Defect_Base_Submit, Defect_Base

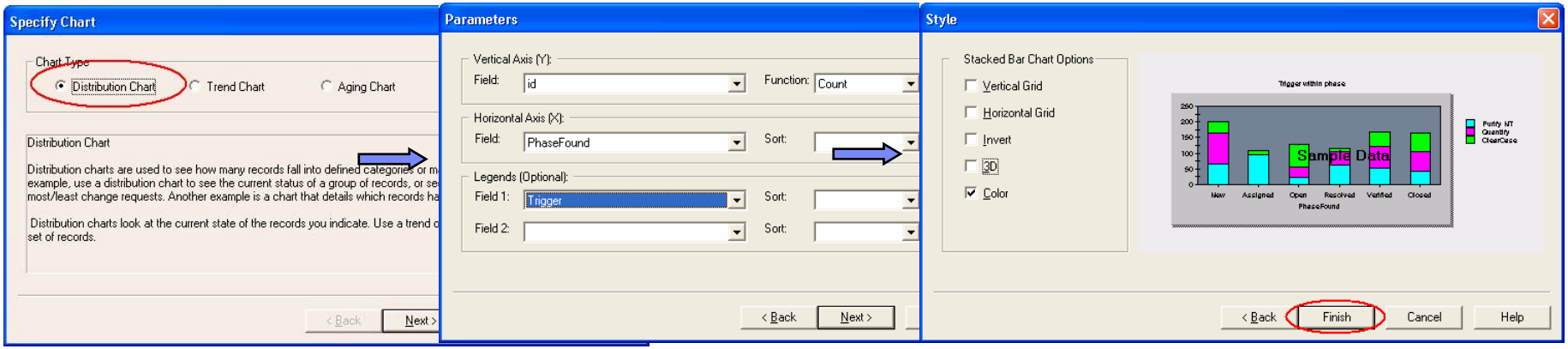


对缺陷数据进行分类

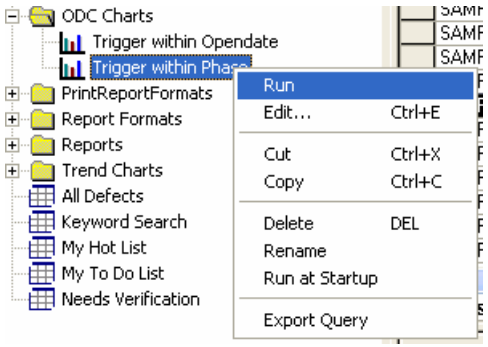

Rational ClearQuest - [SAMPL : (All Defects (Defect))]
File Edit View Actions Query Window Help
Run Query
New Defect
Workspace: Queries, Charts, Reports
Personal Queries
Public Queries
Aging Charts
Customers
Distribution Charts
Email Rules
ODC Charts
Trigger within Oper
Trigger within Phas
PrintReportFormats
Report Formats
Reports
Trend Charts
All Defects
Keyword Search
My Hot List
My To Do List
Needs Verification
id
State
Headline
SAMPL00000001 Opened spelling error in login screen
SAMPL00000002 Resolved sales tax incorrect if item deleted from purchase
SAMPL00000003 Resolved cancel sale doesn't correctly repaint screen
want more help on inventory report
columns out of alignment
delete item not working correctly
override price does not work
alt-C does not invoke cancel operation
inventory re-order not done on large sale
ogout button should be disabled during sale
change due amount is supposed to be red
would like logout button to be larger
ogout in middle of sale aborts sale
clerk allowed to charge too much on credit card
end-of-shift report fails if after midnight
too many spaces in "change due" field
delete operation leaves blank line in form
add item button not active after adding 3 other item
View Defect SAMPL00000012
Main
Notes
Resolution
Attachments
History
Customer
ODC Information
OK
Cancel
Print Record
Actions
Submiter
Phase Found: System Test
Test Type: System Test
Impact: Accessibility
Trigger: Hardware Configura
Responder
Target: Requirement Desig
Defect Type:
Checking
Qualifier:
Source:
Age:
Apply
Revert
Print Record
Actions
Priority:
Symptoms:
Owner:
Description:
Ready
Executing Action: Assign
Record: 12
Count: 40

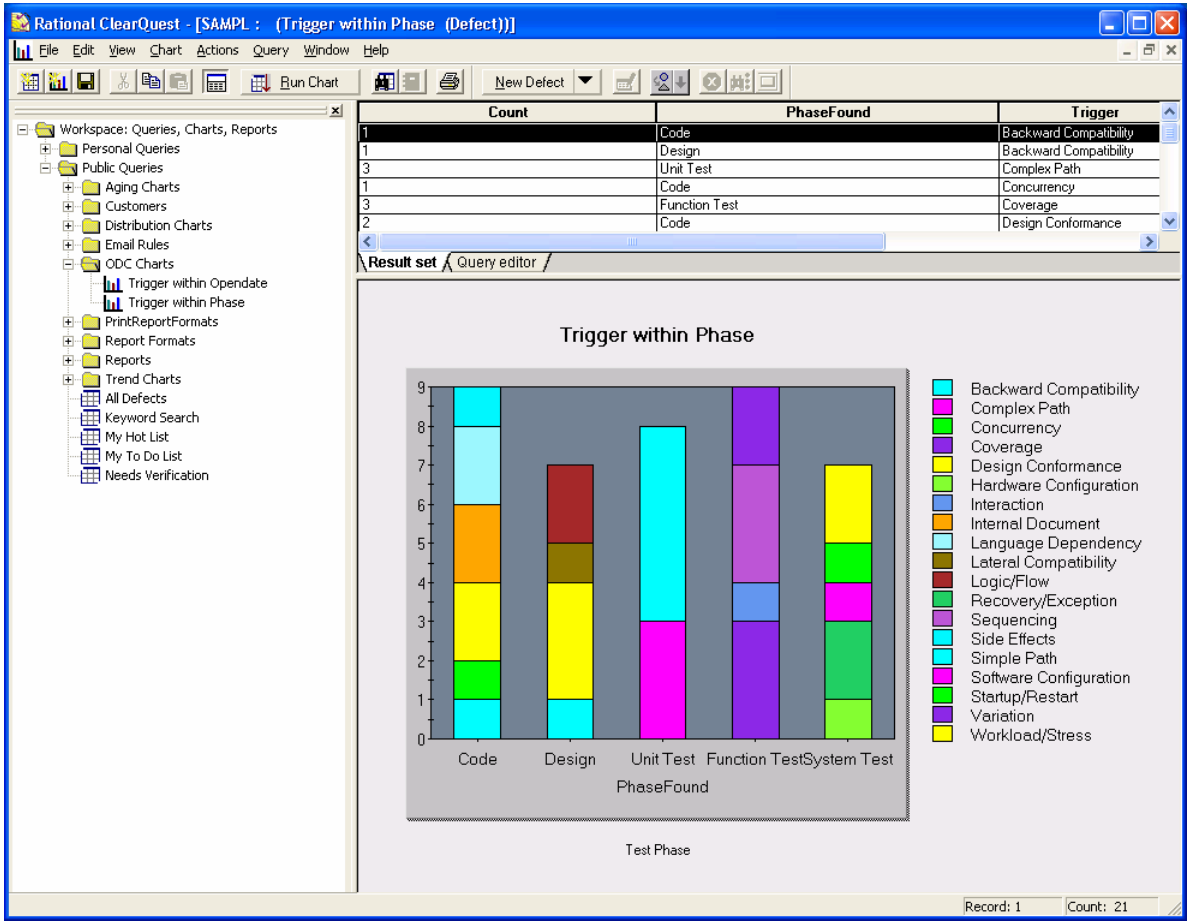
创建分布图来进行ODC分析





创建分布图来进行ODC分析（续）





议程

- 软件软件质量现状和缺陷分类方法
- 缺陷正交分类方法、应用和流程
- 在**Rational ClearQuest**中使用正交缺陷分类技术


Questions


Thank
You