

基于HT-RCA缺陷分析的测试 改进

张玲玲

目录

- 1 现状分析&问题提出
- 2 HT-RCA理论基础
- 3 HT-RCA工程方法

网上问题分析案例（1）

- 这是需求还是BUG？

- 某产品上传超大资源文件后系统重启失败，用户无法使用业务。在测试根因分析后发现测试过程中产品测试人员曾经提出此问题，但产品开发人员认为现网客户不会上传这么大的文件，问题就此放过。测试根因分析结果是需求问题。

- 客户质量标准在哪里？

- 某产品上网后由于网络大量丢包，出现服务质量急剧劣化，严重影响用户体验，被通报为网上事故。产品需求描述支持x%丢包率，所以当丢包率超过需求要求则必然会影响业务。测试根因分析认定非测试漏测问题。

网上问题分析案例（2）

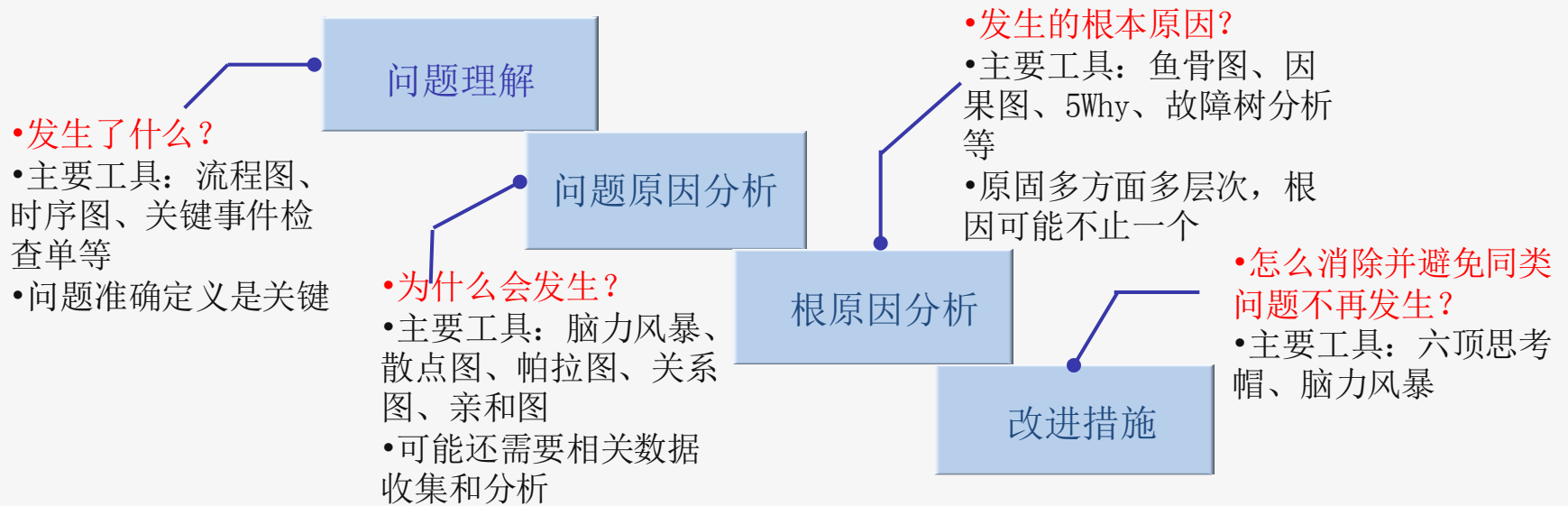
■ 第三方造成的问题也算系统BUG？

- 某产品因客户第三方备份平台异常，没有按照约定时间取走文档，以至于磁盘空间满、待业务中断后才被用户发现、上报事故。现网紧急启动放通机制、应用恢复。测试根因分析结果是非系统问题。

• 这到底是不是人的问题？

- 某产品上网后出现与第三方设备对接失败。测试根因分析结果是测试人员执行不正确，改进措施：增强责任心、提高人员技能。进一步交流发现：该产品对接接口中有N个相似含义字段、且取值各不同，例：A接口0代表OK，而B接口则1代表OK，测试人员执行过程中以为自己是对的，但实际上记错了、搞混了。

根因分析（RCA）概述



- 质量管理工具，通用的问题根因分析模型，应用广泛
- 属于逆向工程，通过RCA寻求问题解决方案，多应用于管理或过程改进
- 对于软件系统来说，软件程序是由人写的，人注定会犯错，缺陷注定会发生。通过**缺陷根因分析**减少缺陷发生

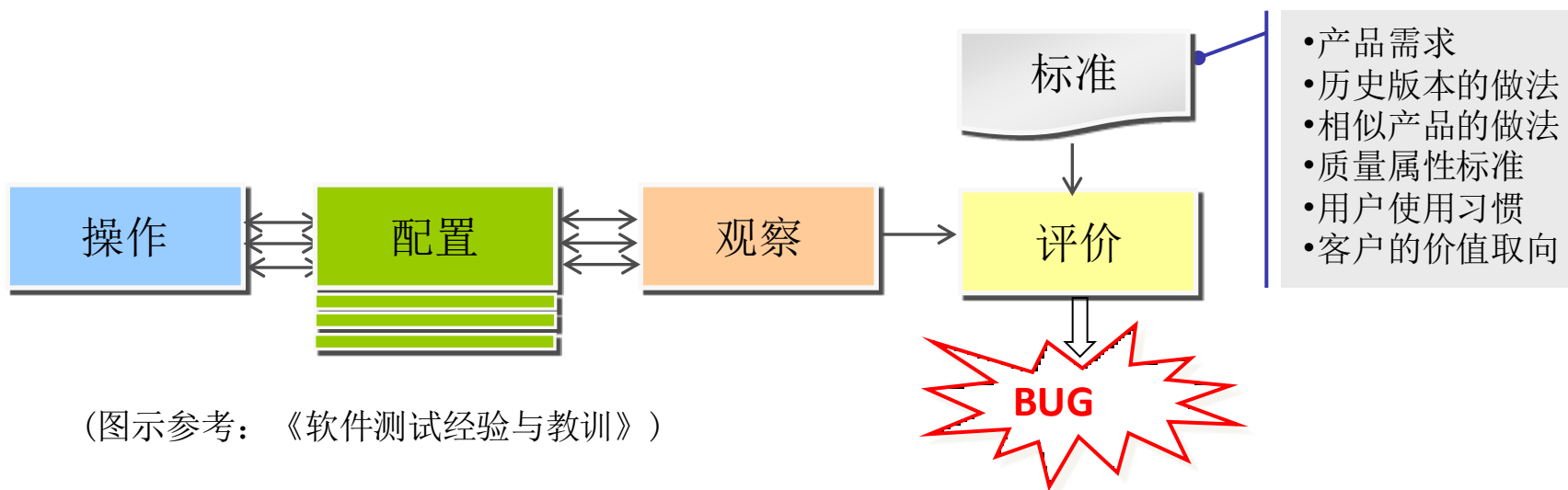
缺陷根因分析现状

- 如何进行缺陷内容的测试经验提取和长效改进
 - 来源于客户问题是一个给我们的回馈，如何最大化的有效利用、固化经验？
 - 对缺陷分析多数停留在度量、测试策略上应用，对于每个缺陷内容有零度分析，但不系统，且也缺少分层。
- 缺陷根因分析模型如何凸现测试特点
 - 多数落脚在测试流程或测试活动的某个方面
 - 对于测试技术专门性的分析不足
- 人因错误如何分析与改进
 - 对人因不分析或笼统的加以归类(例:测试技能不足),并没有作进一步的细致分析
 - 对人因错误改进缺少工程化的考虑

目录

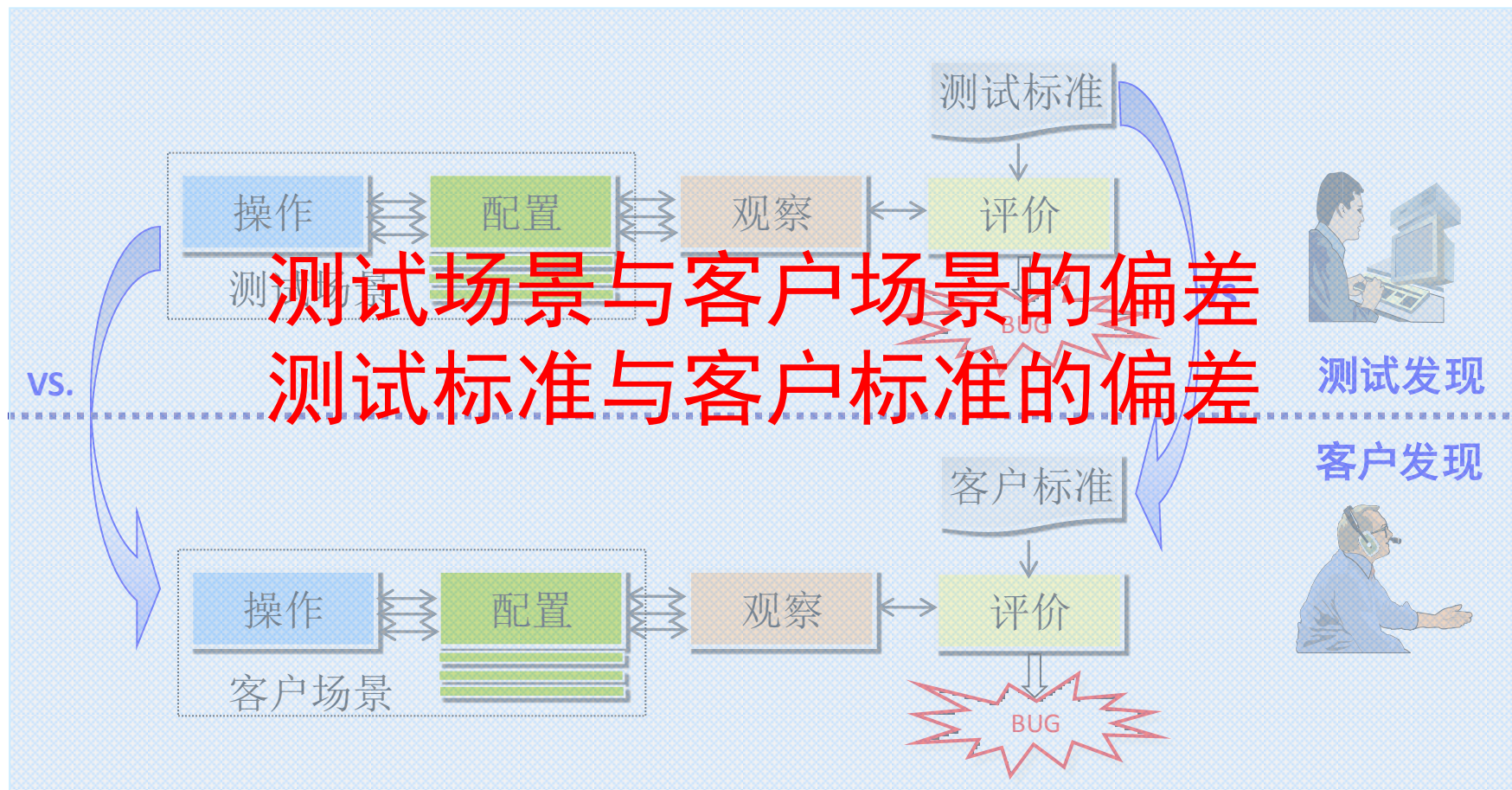
- 1 现状分析&问题提出
- 2 HT-RCA理论基础
- 3 HT-RCA工程方法

BUG是怎么发现的？

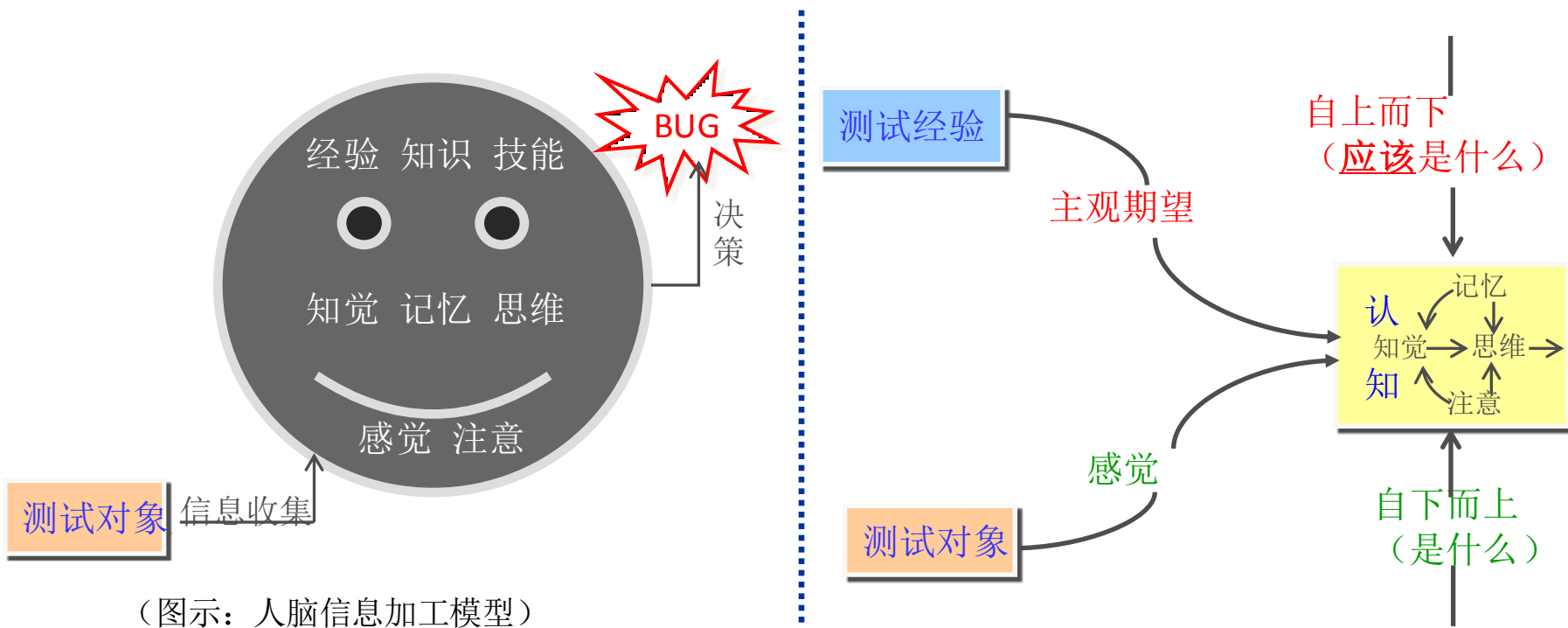


- 图示BUG可以定义为：在特定配置下对特定系统进行特定操作,将观察到的信息与标准加以比对、并作出评价，如果发现偏差即作为BUG提出
- 百度定义：在电脑系统或程序中,隐藏着的一些未被发现的缺陷或问题统称为bug。
- **BUG是一种感知。有不同期望的人对BUG的感知是不一样的。**

如何客户知道BUG而我们不知道?



人为什么会犯错？

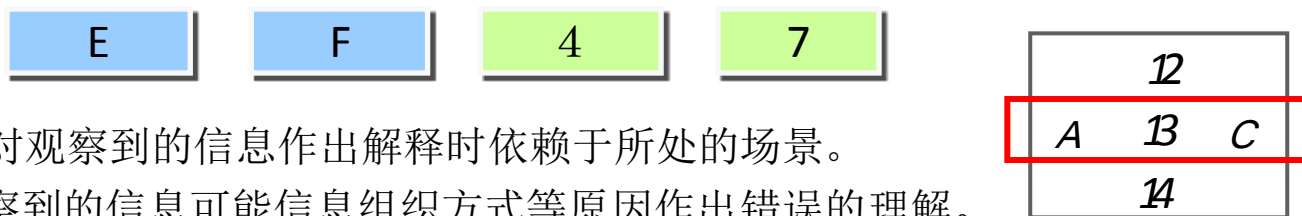


- 人脑是一个功能不确定的黑盒系统，受系统容量、负荷、处理能力/效率、知识经验等因素影响
- 人脑信息加工的各个环节均可能出现错误：错觉、选择性注意、心理模型（期望与倾向）、认知偏差（记忆、知觉、思维）等
- 自上而下加工的主观期望是影响测试结果客观性、造成错误的主要原因

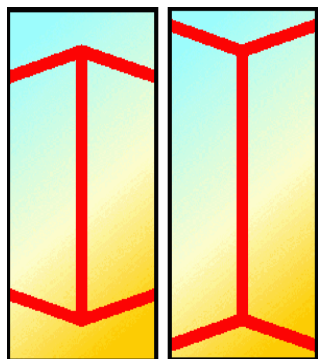
人的错误与偏见 (1)

- **证实倾向**。对于规则或假设,人更倾向于去证实它们,而很少去证伪它们。

【四卡片选择作用】卡片一面写用字母、另一面写有数字。有规则“若卡片的一面是元音字母,则另一面是偶数”。如果要证明该规则,需要翻看哪些卡片



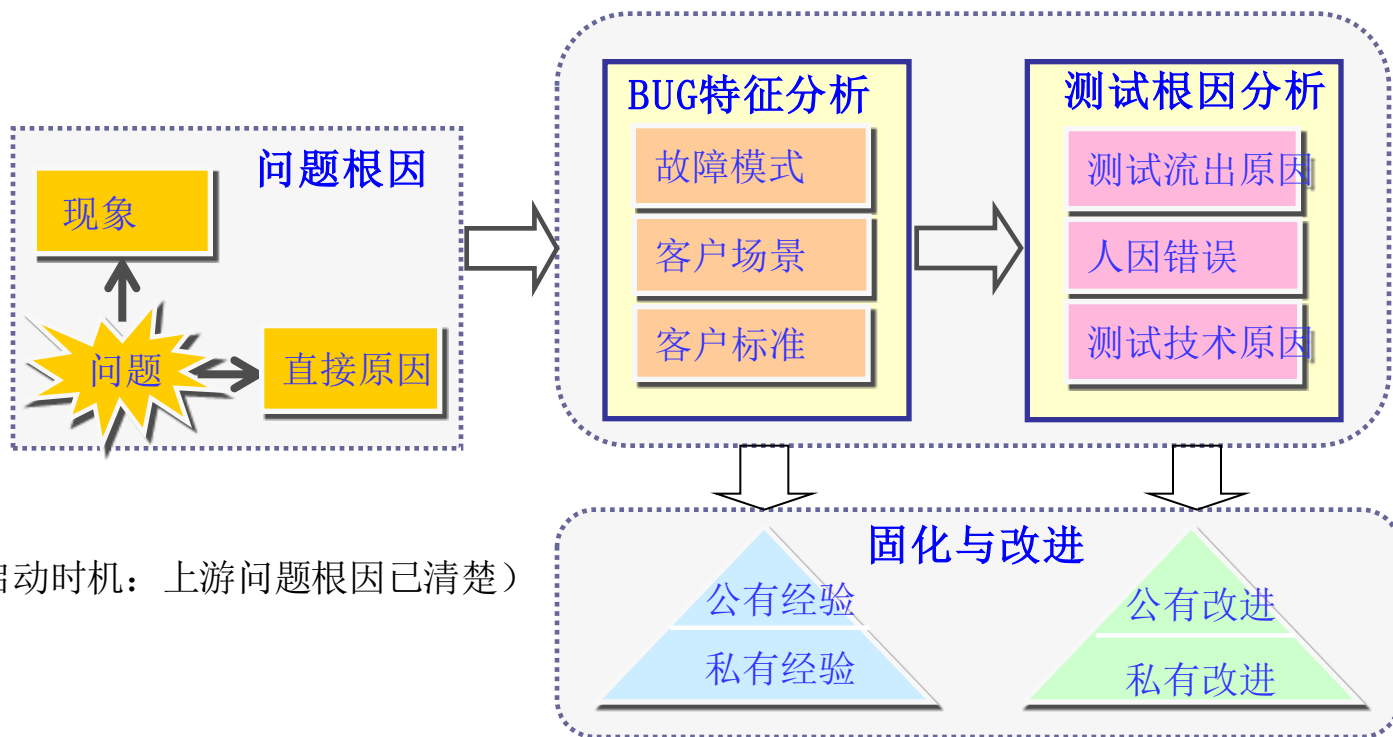
- **情景倾向**。人对观察到的信息作出解释时依赖于所处的场景。
- **错觉**。人对观察到的信息可能信息组织方式等原因作出错误的理解。
- **选择性注意**。人只会注意到新异和自己关注的信息。人更易于关注和选择那些与形成的产品经验相关的测试结果,而忽略那些自己经验之外的测试结果。
- **信息不确定性**。如果信息有歧义、模糊,则不同人可能产生不同结果解释 (双歧图形)



目录

- 1 现状分析&问题提出
- 2 HT-RCA理论基础
- 3 HT-RCA工程方法

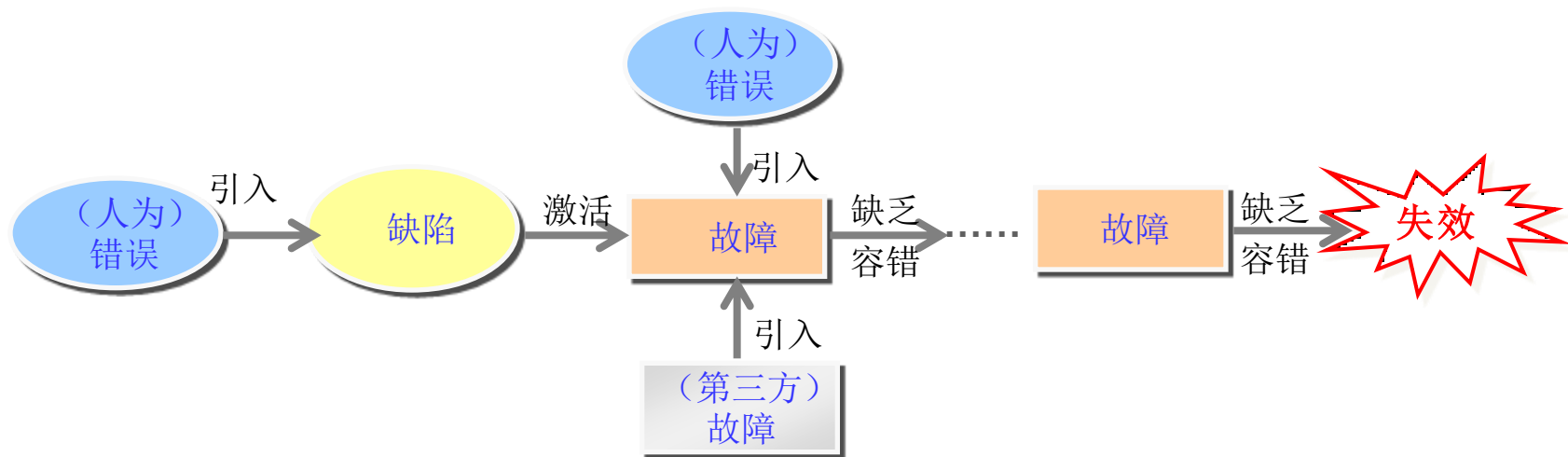
HT-RCA模型介绍



(启动时机：上游问题根因已清楚)

- 强化了面向客户的BUG特征多维度分析、对问题内容信息的深度提取和利用。
- 增强了测试技术原因和人因错误分析，对问题根因的多方面分析和改进识别。
- 从公有和私有2级进行经验提取和测试改进。

BUG特征分析：故障模式分析



XX系统 → (抽象化) → 故障模式 → (实例化) → YY系统

术语	特征	定义
错误	(人-) 动作-(XX) 对象	人针对特定的对象所做的特定动作
缺陷	(XX) 对象-(XX) 状态	从特定程序对象角度阐述, 体现为静态存在
故障	(XX) 实体-(XX) 状态	从系统角度阐述, 体现为运行状态. 故障-旦激活后则可以反复传递
失效	(XX) 业务-(XX) 状态	从用户角度阐述, 体现为所用户体验的业务状态

BUG特征分析：故障模式分析

【示例】 由于用户申请的license文件中缓存消息数一项参数设置过大，导致XX系统内存分配出现异常，造成该节点频繁coredown，影响正常业务处理

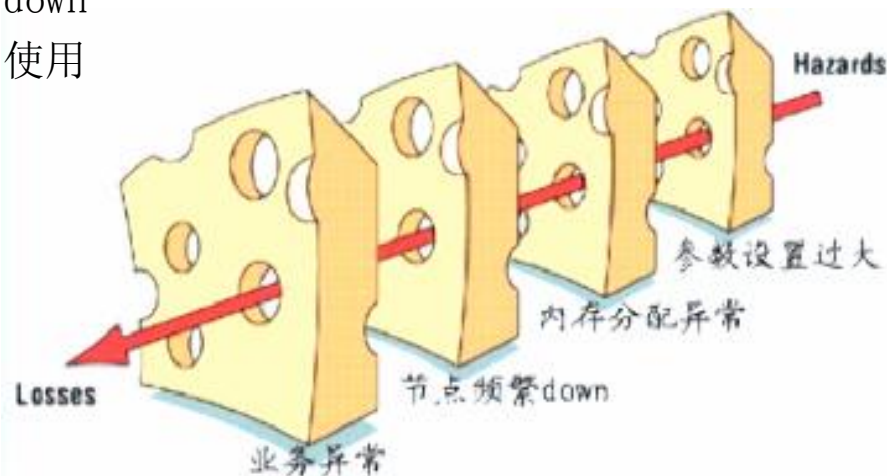
【分析】

人为错误:用户 XX参数设置过大

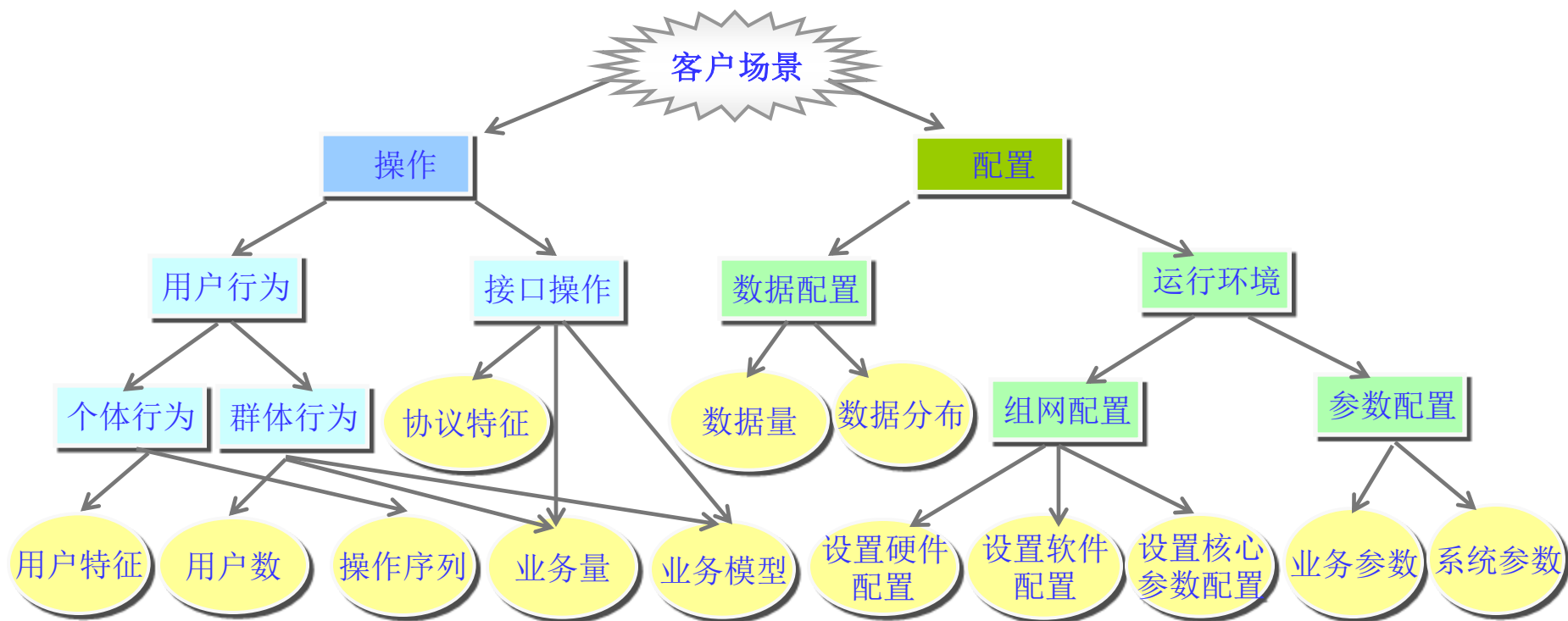
故障1: XX系统 内存分配异常

故障2: XX节点 频繁coredown

失效: XX业务 不能正常使用



BUG特征分析：客户场景分析

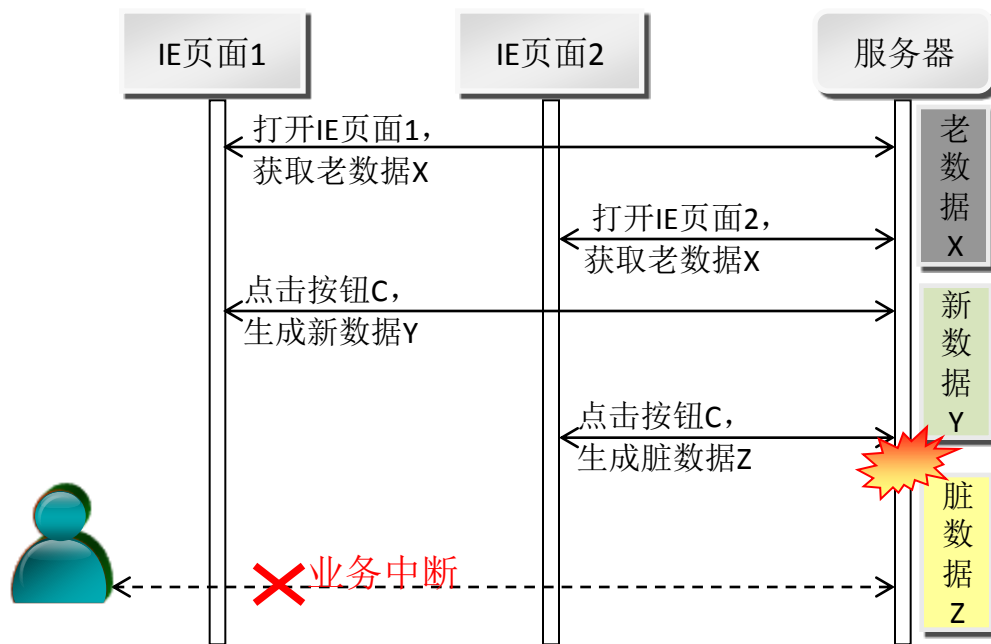


- 分析客户场景与测试场景的偏差，例：用户的使用习惯、用户与系统的交互模式、哪些操作和配置场景会引起业务失效，逐步建立起客户场景信息库。
- 通过场景分类树的方法，将场景层层挖掘深入下去、不断填充细节。

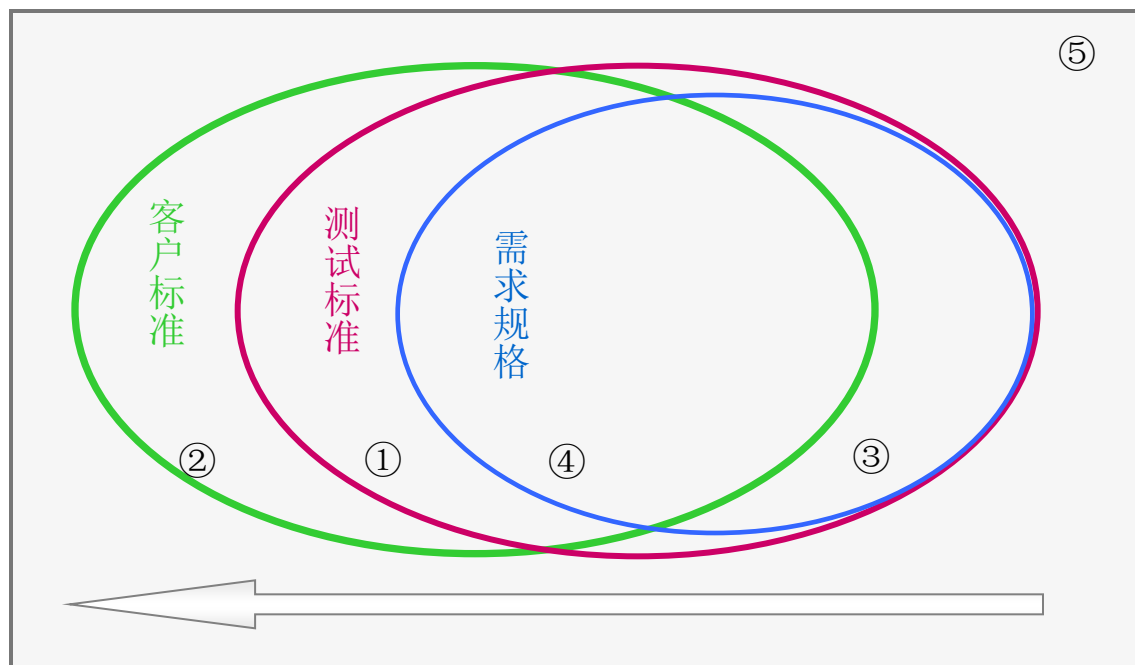
BUG特征分析：客户场景分析

【示例】 某页面C按钮只可能操作一次，因为页面在操作后按钮自动变成不可点击。现场在操作之前，同时打开了两个IE窗口。在其中一个IE窗口上点击操作保存数据后，又在另外一个IE窗口上再次执行相同操作，导致第一次正确数据被覆盖，引发业务失效。

【分析】 这个案例中属于典型的用户操作序列问题：打开IE页面A → 打开IE页面B → 在页面A点击按钮C → 在页面B点击按钮C



BUG特征分析：客户标准分析



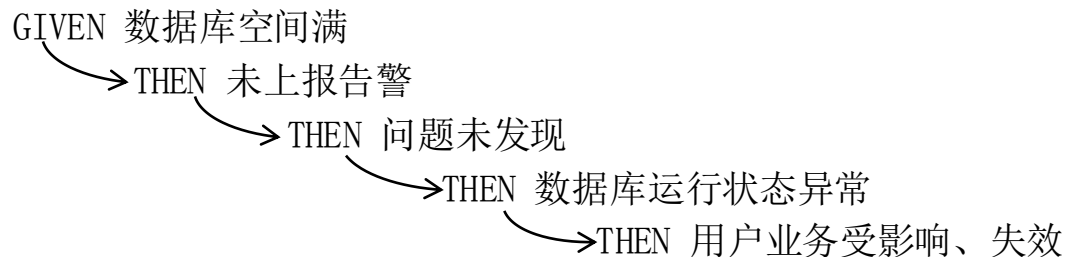
- ① 测试标准—需求规格
- ② 客户标准—测试标准
- ③ 测试标准—客户标准
- ④ 三者一致
- ⑤ 未知

- 理想状况下，测试标准应是严于客户标准且又涵盖客户标准。
- 从问题发生过程的因果链出发，针对一个具体的错误场景，N-Then方法进行层层推进，不断明晰问题的结果影响、客户所使用的合格判据、背后所承载的真实期望，从中提取出测试标准。
- 对于共性问题按质量属性领域可以进一步抽取出通用的测试标准

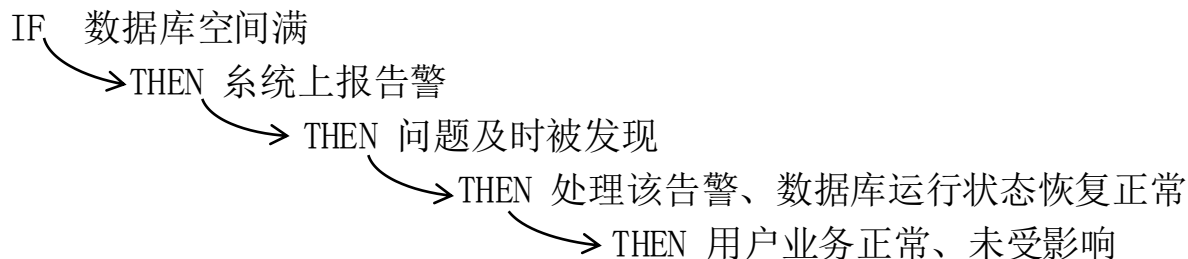
BUG特征分析：客户标准分析

【示例】某版本修改引入问题导致数据库空间满，但由于告警失败，问题到当天下午才被发现、规避，导致3000多个用户受到影响，多个用户投诉，客户认为我们的系统可靠性比较低，无法及时暴露系统的异常运行状态。改进措施为：增加数据库空间满的告警。

【分析】针对问题发生过程，可以描述成：



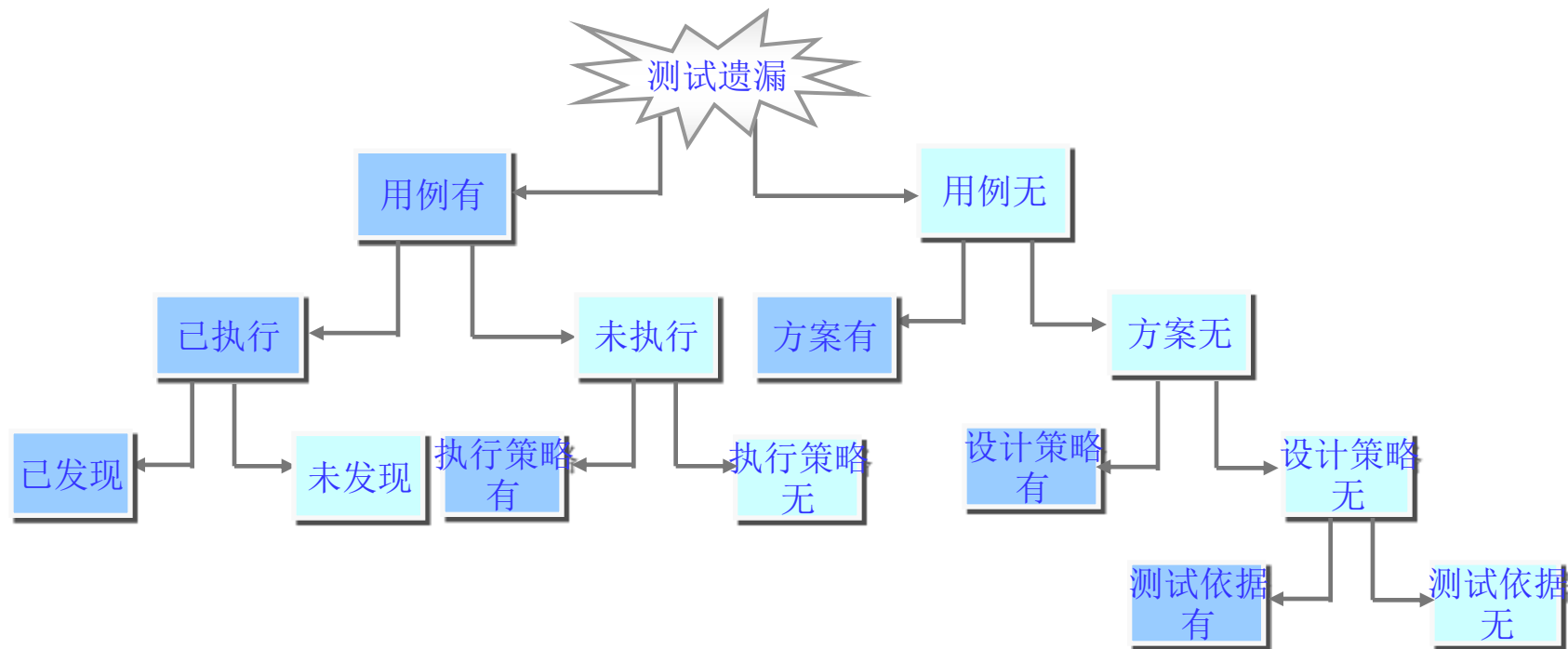
根据问题发生过程，还原客户的期望：



客户的真实期望是：问题及时发现、被处理，不造成业务失效。

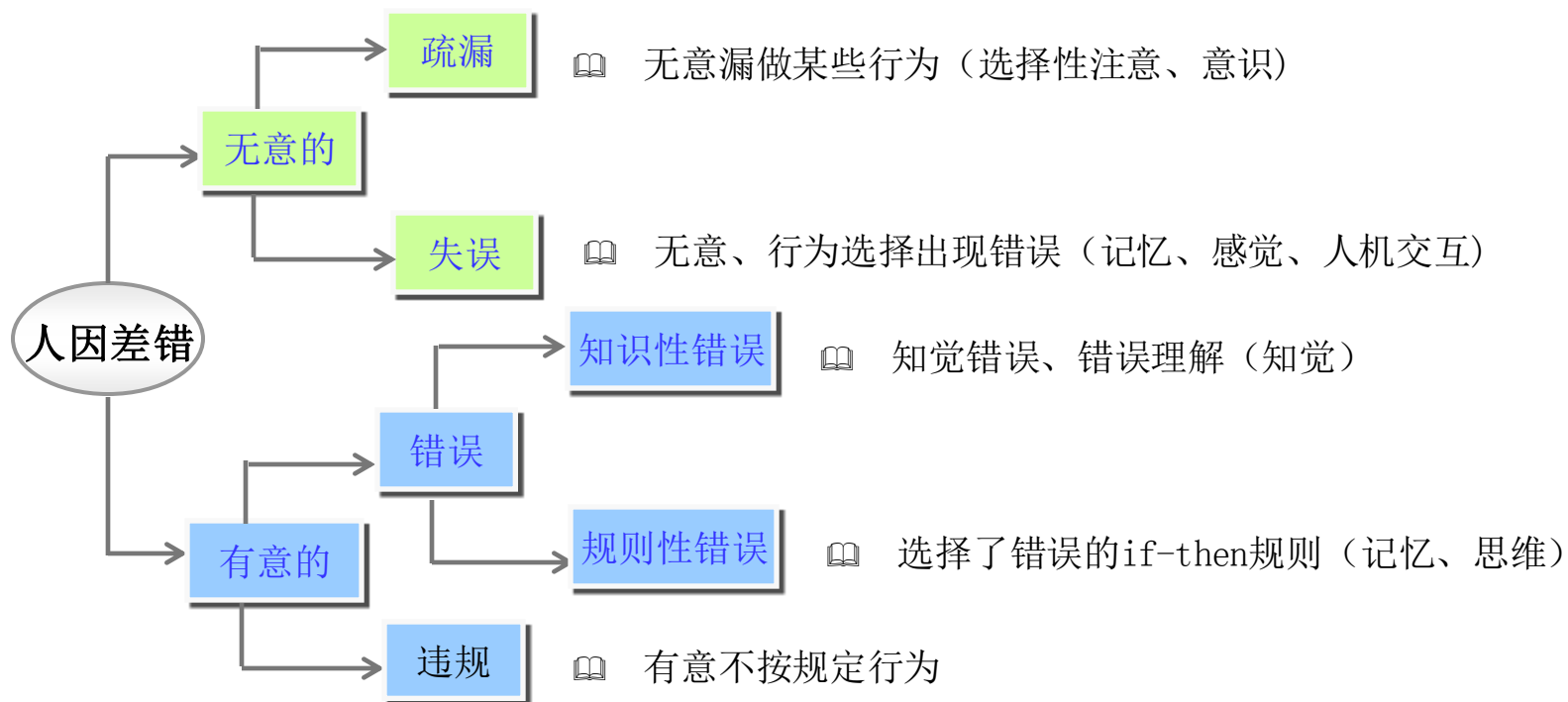
进一步设想：如果没有告警，以短信通知等其它方式确保问题被发现被处理是否也可以？

测试根因分析：测试流出原因



- 测试过程回放，从测试流程中寻找缺失的缺陷控制点。
- 找到测试流出环节后还可以进一步细化分析。例：“用例无”指的是：无这个用例 or 用例内容残缺不全，测试点/观察点缺失

测试根因分析：测试人因分析

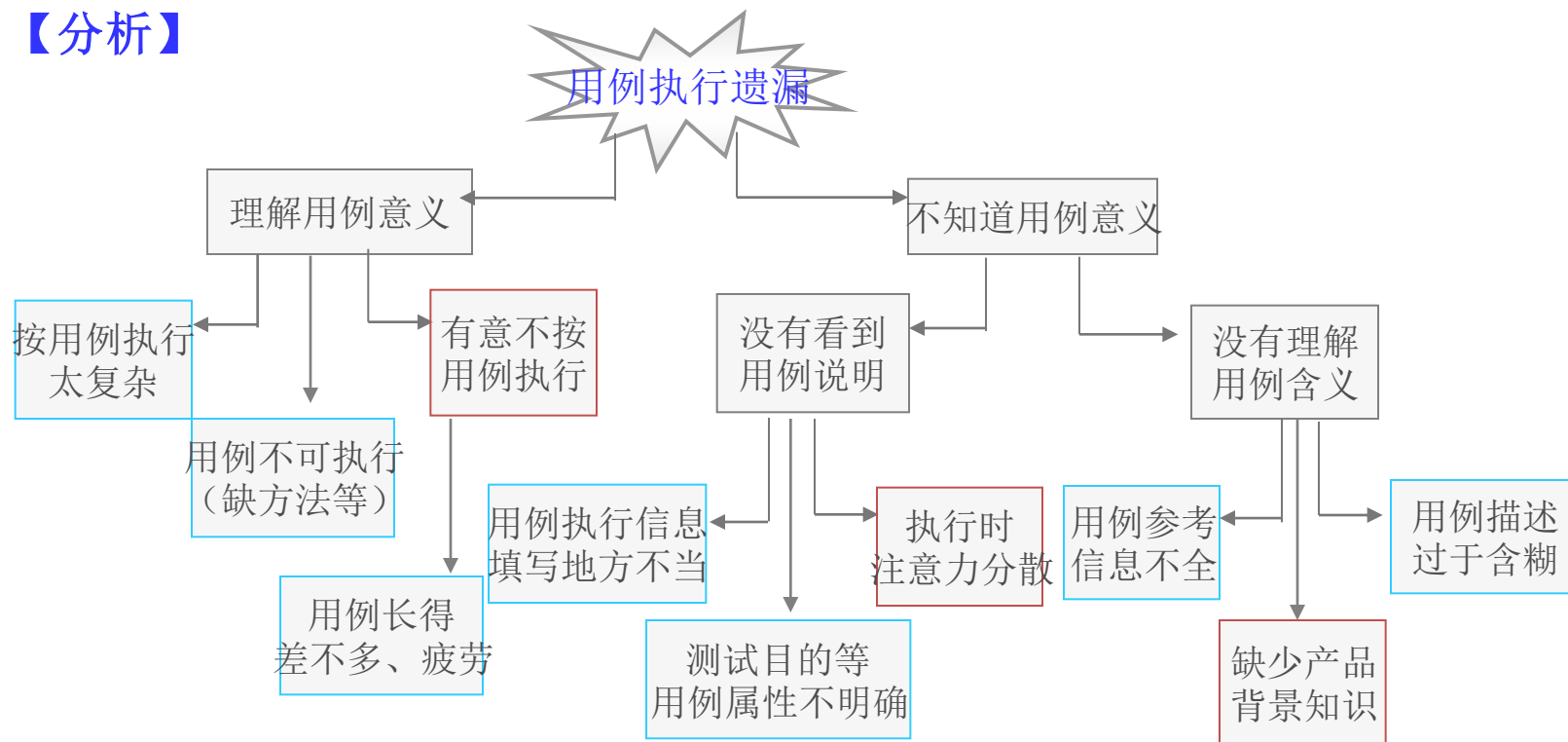


- 基于SRK模型（技能Skill, 规则Rule, 知识Knowledge）
- 违规主要体现在系统操维护层面

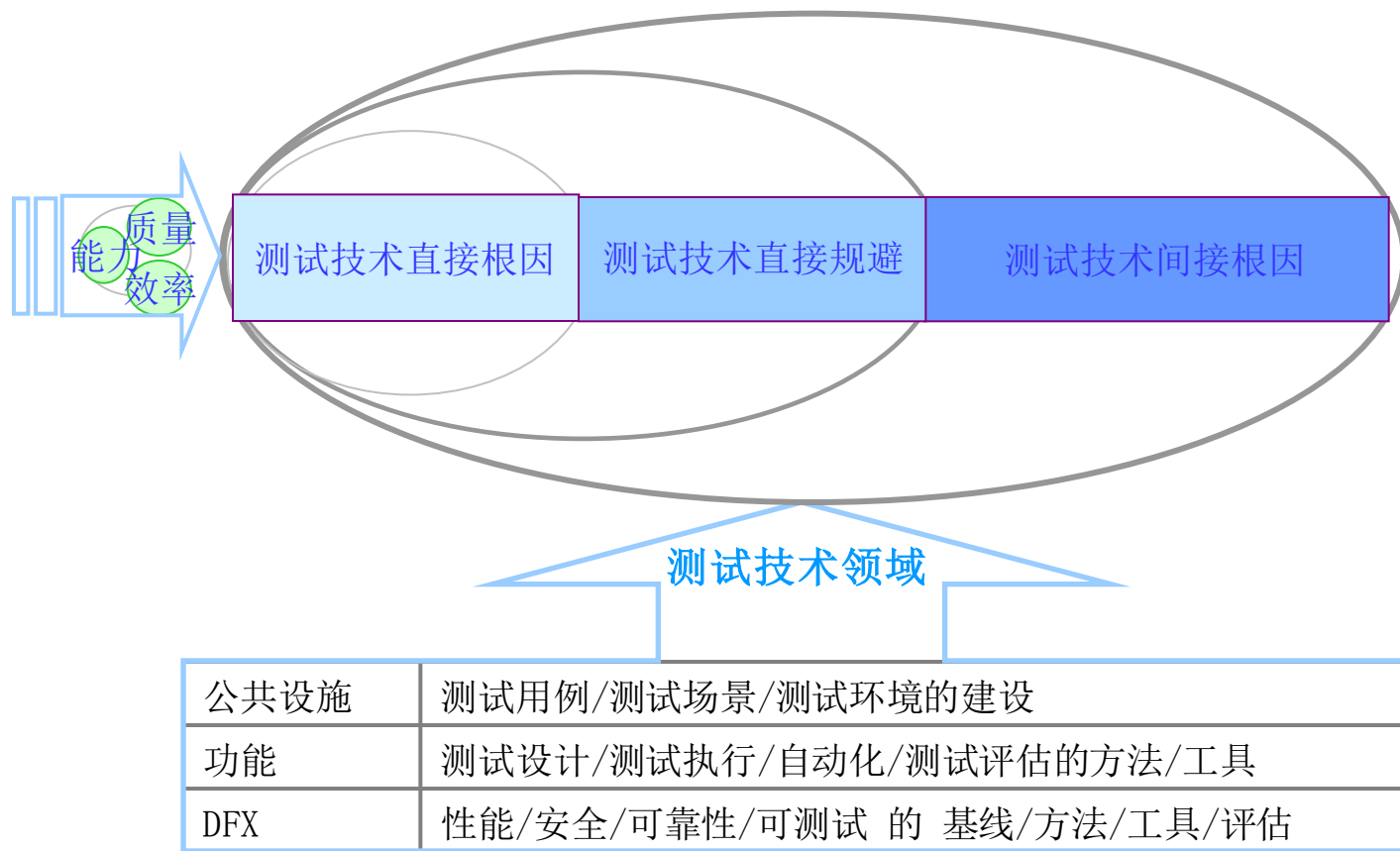
测试根因分析：测试人因分析

【示例】测试流出原因是功能用例执行的测试人因分析

【分析】



测试根因分析：测试技术原因

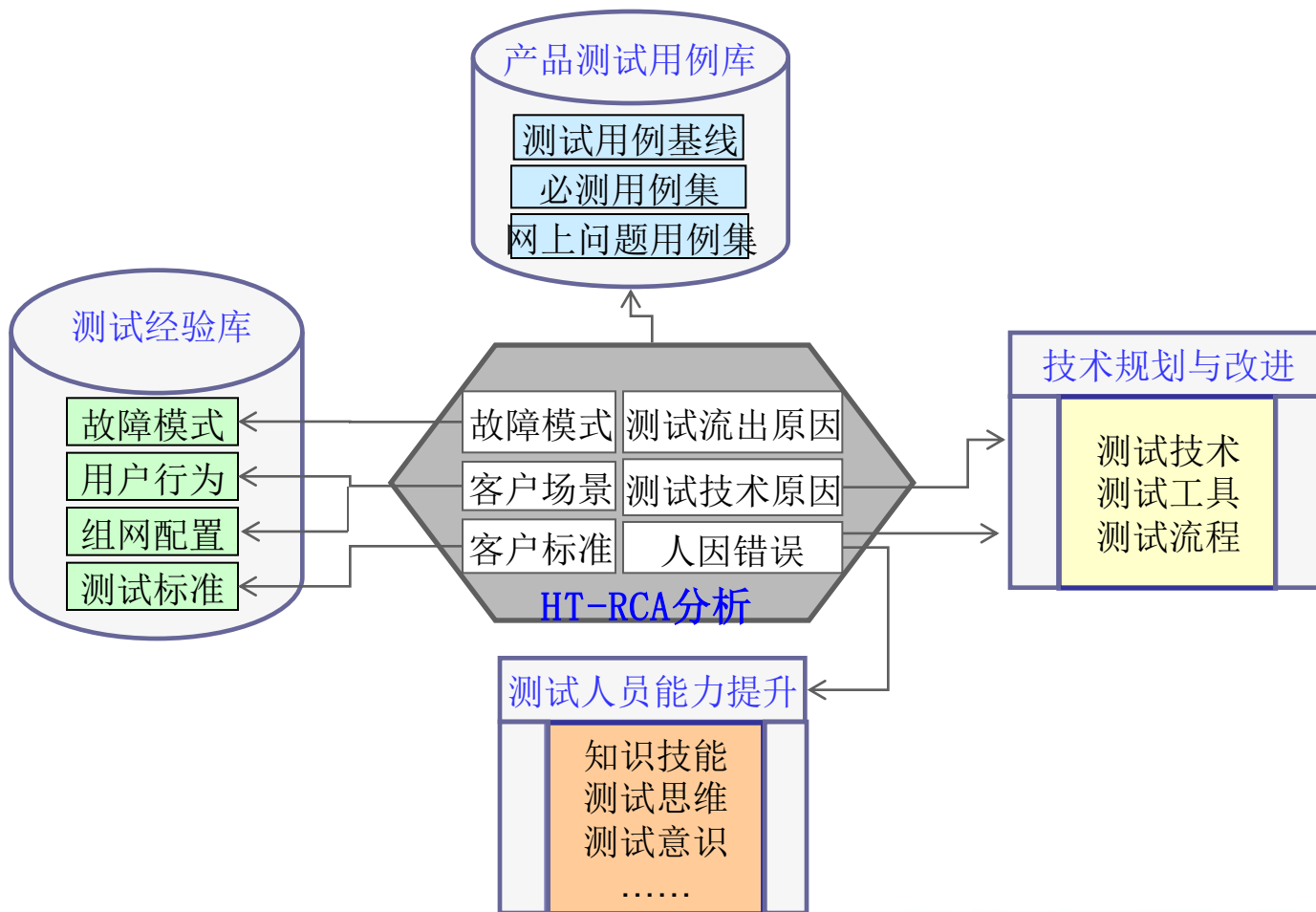


测试根因分析：测试技术分析

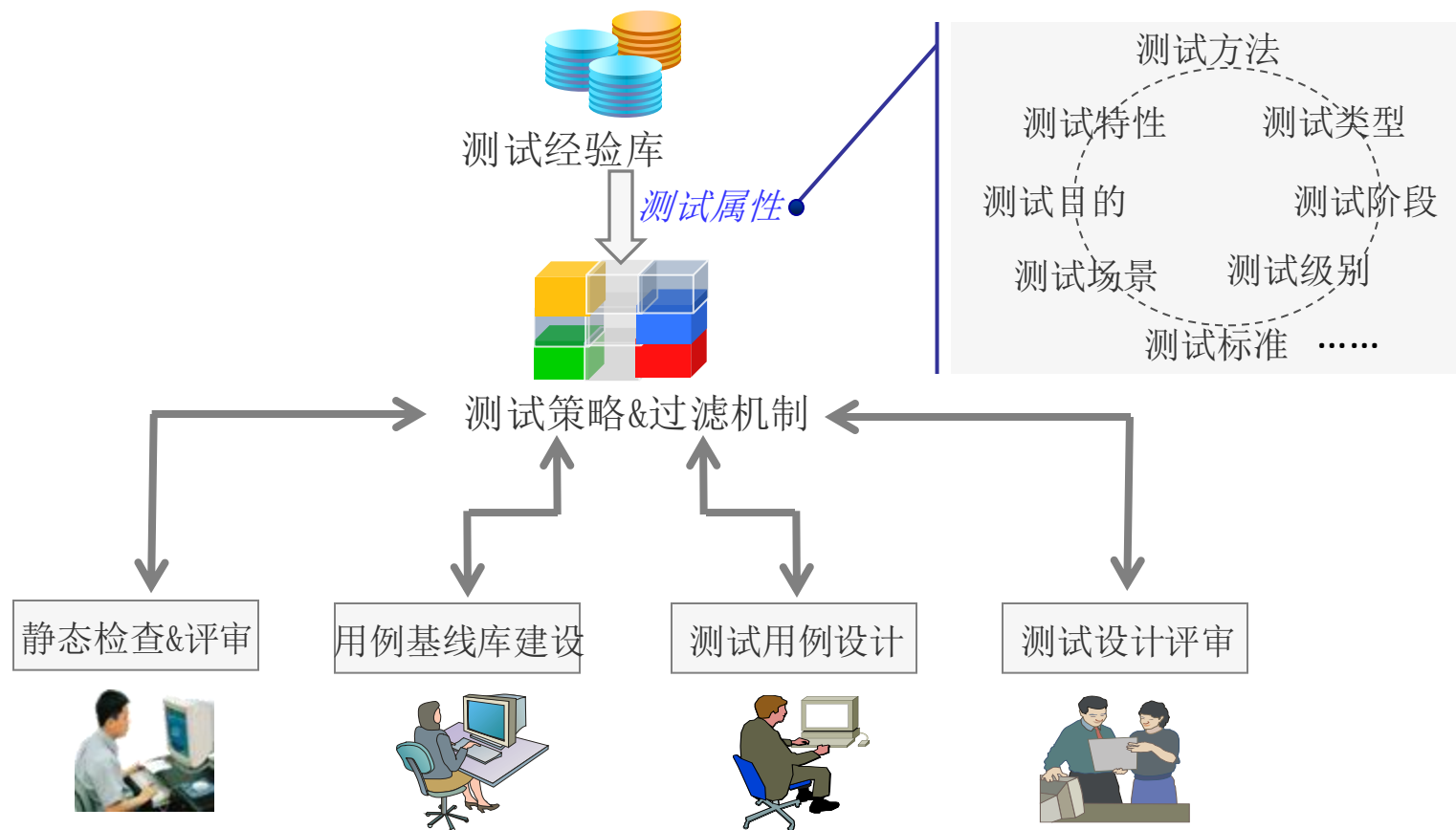
【示例】测试流出原因是功能用例执行遗漏的测试技术原因分析
【分析】

要素	了解&分析
操作	<ul style="list-style-type: none"> • 是否有办法构造测试场景? (能力) • 操作场景人工构造过程是否繁琐、易于出错? (质量) • 操作场景构造效率如何、是否相应工具支撑? (效率) • 执行工具是否易于使用、所构造场景是否满足操作需要? (质量)
配置	<ul style="list-style-type: none"> • 是否有办法、有手段进行测试条件配置? (能力) • 人工配置过程是否繁琐、易于出错? (质量) • 配置正确性是否易于检查? (质量) • 配置效率如何、是否有相应工具支撑? (效率)
观察	<ul style="list-style-type: none"> • 是否相应的输出信息都有手段可以观察到? (能力) • 输出信息通过人工是否易于观察和理解? (质量) • 是否观察点多、分散、是否有工具统一观察信息收集? (效率)
评价	<ul style="list-style-type: none"> • 是否所有获得的观察信息可以分析和比对? (能力) • 是否所获得的观察信息通过人工易于分析和比对、不易出错? (质量) • 结果分析和比对效率如何、是否有相应工具支撑? (效率)

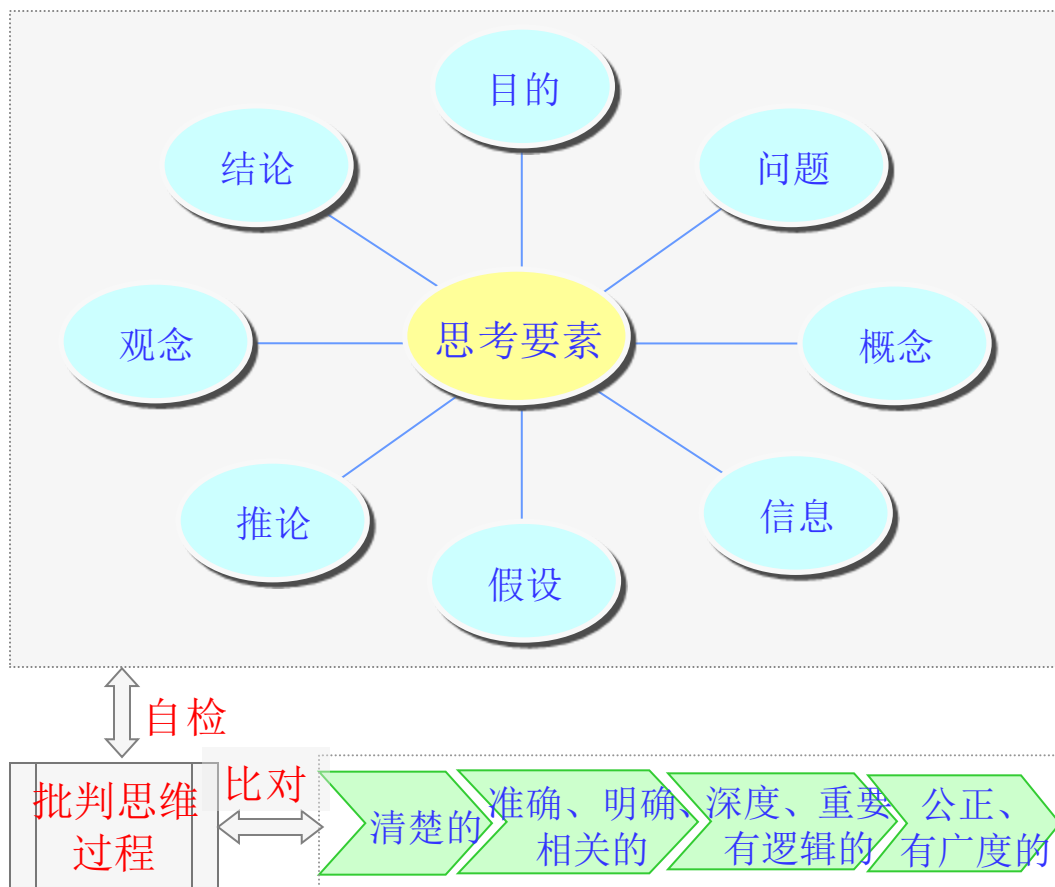
固化与改进：总体改进思路



固化与改进：经验固化方案



固化与改进：测试思维



应该把独立思考和综合判断的能力放在首位，而非获得特定知识的能力。

—爱因期坦

固化与改进：人因工程

