新旧项目如何从测试角度推敏捷

宗刚

35代孙

宗泽（1060～1128）北宋末、南宋初抗金名臣

- 惠普QO非功能负责人
- 民企、创业、外企
- 开发、测试、项目经理、产品经理
- 敏捷实践者、高级程序员、DBA

议程

- 测试面临的困境
- 敏捷的核心
- 新项目从无到有敏捷实践过程
- 旧项目从无到有敏捷实践过程
- 敏捷推动的技巧
- 关键敏捷测试实践

测试面临的困境

版本快速上线，无法保证质量；

开发人员不自测，丢给测试的版本经常通不过冒烟测试；

老板老是以业界“最佳实践”10:1的开发测试比说测试人员应该更少。

想求助于敏捷，开发人员也不愿写单元测试而停滞不前，怎么办？

最后测试人员只能埋头干活，没有时间抬头看路。

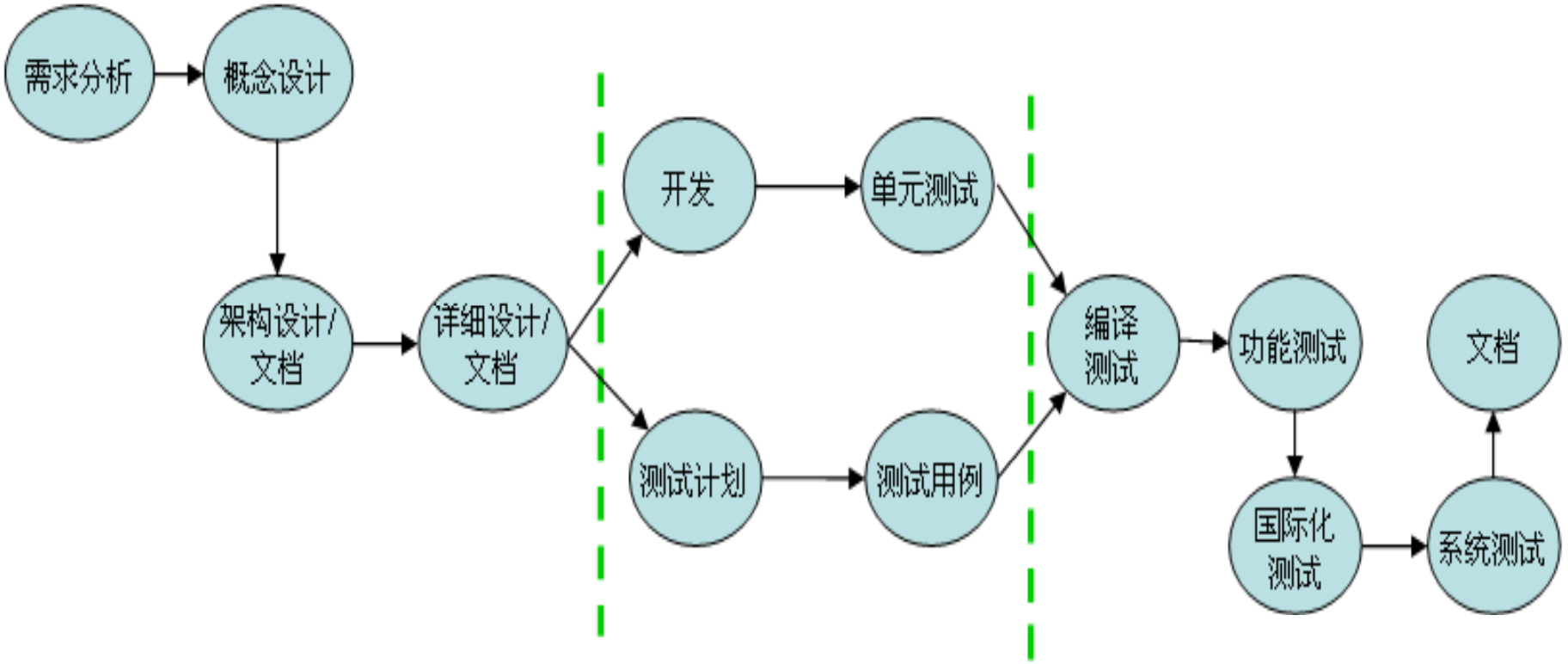
江湖中传说的敏捷

1. 所有好的都往敏捷靠，传统都是坏的
2. Best Practice 大集合？
   a) 结对
   b) 100%覆盖、100%自动化
3. 讲师背景不同，看问题的角度就不同
   a) 百度 vs 惠普
   b) 利益出发点不同，建立能力“壁垒”
4. 能力要求特别高
   a) TDD
   b) 结对

议程

- 测试面临的困境
- 敏捷的核心
- 新项目从无到有敏捷实践过程
- 旧项目从无到有敏捷实践过程
- 敏捷推动的技巧
- 关键敏捷测试实践

传统软件开发的模式


需求分析
概念设计
架构设计/
文档
详细设计/
文档
开发
测试计划
单元测试
测试用例
编译
测试
功能测试
国际化
测试
系统测试
文档

传统软件开发模式的前提假设

开始就存在着一套定义相当明确的需求；

改变是小型且便于管理的；

最后阶段系统集成会顺利进行；

我们完全可以按计划交付。

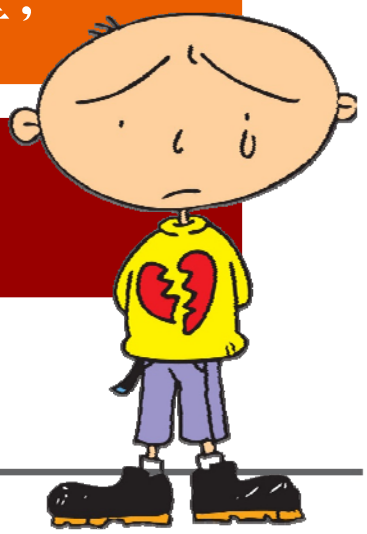
传统软件开发模式的挑战

系统有很多模块，无法集成，集成过程中大量的bug；

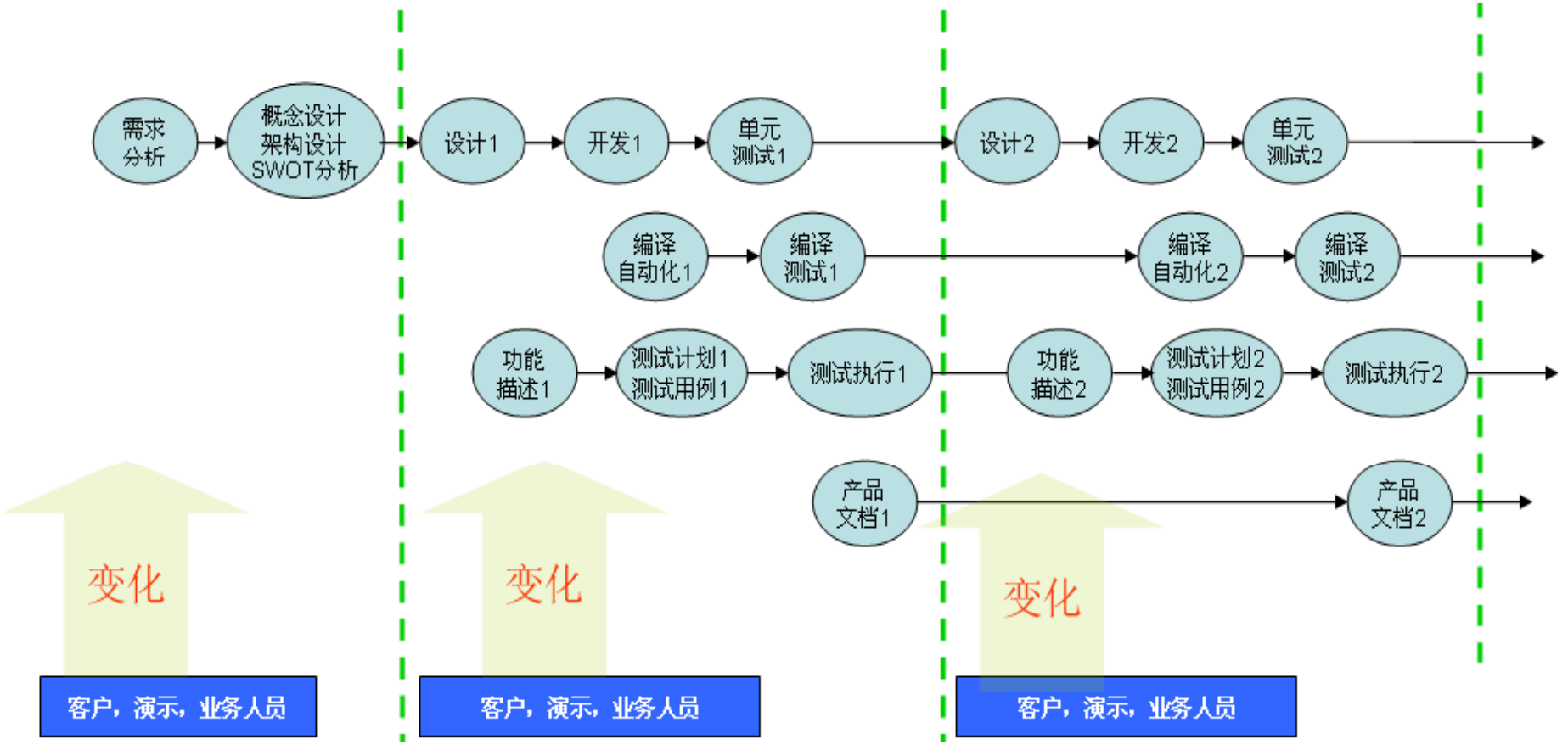
系统集成后不好用，不可用，无法满足业务需求；

变更实时发生，一个月50个版本，无法保证质量；

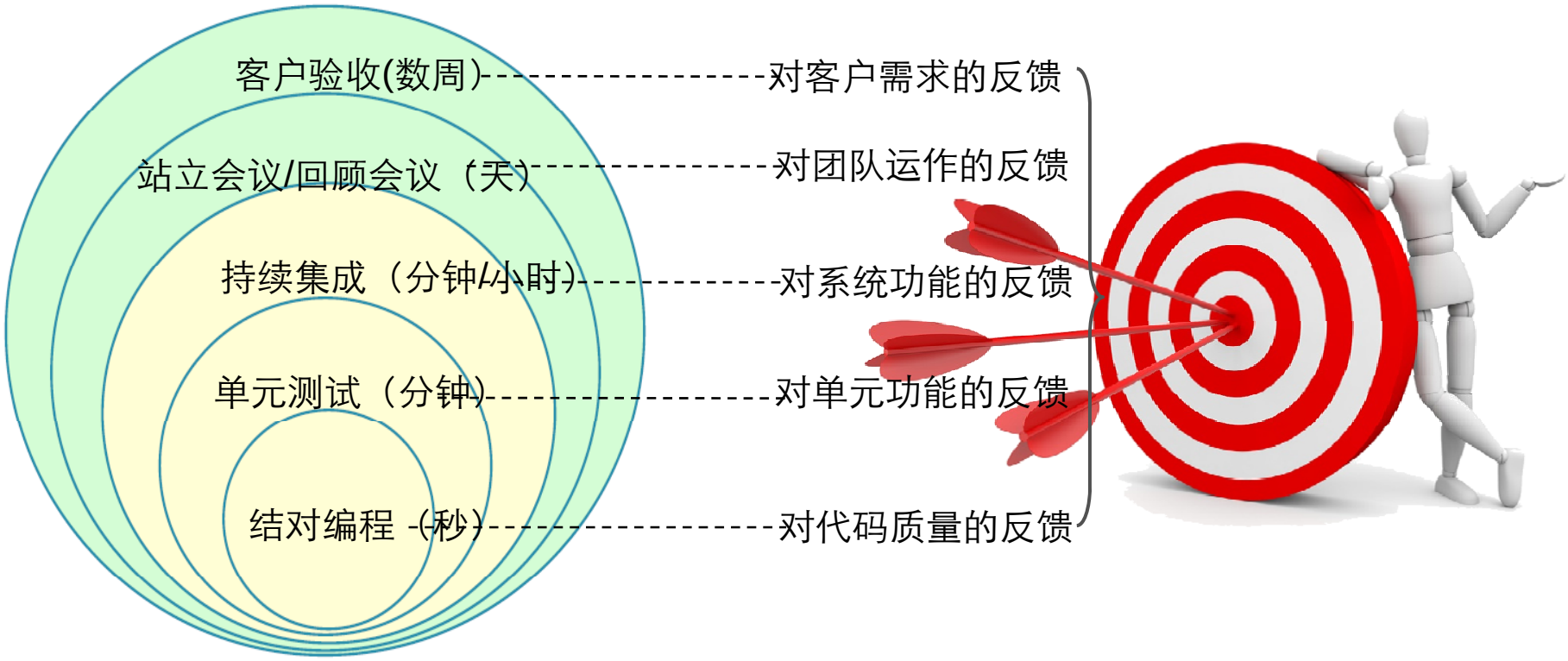
软件开发项目进度一推再推，缺乏透明性。



敏捷开发流程


需求
分析
概念设计
架构设计
SWOT分析
设计1
开发1
单元
测试1
设计2
开发2
单元
测试2
编译
自动化1
编译
测试1
编译
自动化2
编译
测试2
功能
描述1
测试计划1
测试用例1
测试执行1
功能
描述2
测试计划2
测试用例2
测试执行2
产品
文档1
产品
文档2
变化
变化
变化
客户，演示，业务人员
客户，演示，业务人员
客户，演示，业务人员

多层次反馈不断调整以逼近目标



议程

- 测试面临的困境
- 敏捷的核心
- **新项目从无到有敏捷实践过程**
- 旧项目从无到有敏捷实践过程
- 敏捷推动的技巧
- 关键敏捷测试实践

新开发项目背景

- 企业性质：外企
- 人员结构：1项目经理 2业务 1架构 13开发 1UI 1测试
- 开发测试比：13:1
- 人员能力：多数来自民企Leader
- 地域分布：法国 中国
- 技术：B/S架构 主要进行数据转换，界面展现少

项目痛点

- 测试Case用于将来几个月之后
- 各开发自己负责一块开发，各扫门前雪
- 所有代码放在自己的电脑上
- 将来是一个未知数？

如何保证将来的质量？

如何快速提升质量反馈？（步骤）(1)

第一步：

– 问题：开发不Checkin 代码

– 解决方法：SVNStat 每天至少签入一次

第二步：

– 问题：接口是否统一，能否编译通过

– 解决方法：Checkin代码自动化编译 Ant CruiseControl

第三步：

– 问题：手动部署麻烦，常因环境问题浪费时间

– 解决方法：自动化部署

如何快速提升质量反馈？（步骤）(2)

第四步：

- **问题：**如何快速验证签入的代码业务是正确的，常出A类Bug
- **解决方法：**自动化冒烟测试 DBUnit从1到2，邮件通知修改Bug

第五步：

- **问题：**冒烟测试经常通过不了
- **解决方法：**开发人员自测，自动化部署脚本通用于开发和测试环境

第六步：

- **问题：**有些Bug常出现，Bug缺少归类和分析
- **解决方法：**每周和开发人员一起分析Bug原因，增加Bug模块、Bug原因等字段，需求点和TestCase映射

如何快速提升质量反馈？（步骤）(3)

第七步：

问题：有一些模块Bug较多，有些人开发的Bug较多

解决方法：推动常出问题模块的开发人员单元测试，结对核心模块修改，提升一些开发人员的技能

第八步：

问题：代码质量反馈速度慢

解决方法：分不同粒度的自动化测试，代码签入就进行UnitTest，1个小时一次的集成测试，一天两次的全量测试

第九步：

问题：持续集成系统反馈速度太慢

解决方法：换Jenkins支持多台机器跑Case

本项目持续集成使用工具

– Tools：静态分析
- StatCVS　统计CVS的变化
- Cloc　代码行统计
- Metrics 各种衡量指标
- Emma 测试代码覆盖率
- CheckStyle 代码风格检查
- Jdepend 查找包之间的依赖
- PMD 查找是否有复制拷贝的代码
- FindBugs 查找bug
- Jtest 全面查找问题（商业软件）

– Tools：动态分析
- Junit 单元测试，集成测试
- DBUnit 数据库单元测试工具
- Junitperf 单元性能测试工具
- TPTP 白盒性能测试工具
- Diagnostics 白盒性能测试工具
- WebInspect自动化黑盒安全测试

其它一些实践

版本管理

- 主干开发
- 测试代码也是代码
- 配置都是代码
- 基础架构也是代码

估算与优先级

- 产品负责定优先级
- 技术负责估算时间
- 产品依据实施时间调整优先级

结对

- 业务开发结对
- 业务测试结对

项目成果

成果：

系统成功上线

开发测试需求和睦相处

产品团队和项目团队人员良性轮岗

不足：

UAT阶段系统才发现性能是大问题，花大量时间进行优化

议程

- 测试面临的困境
- 敏捷的核心
- 新项目从无到有敏捷实践过程
- **旧项目从无到有敏捷实践过程**
- 敏捷推动的技巧
- 关键敏捷测试实践

旧项目维护背景

- 企业性质：民企
- 人员结构：1项目经理 2业务 2架构 9开发 6测试
- 开发测试比：3:2【注】
- 人员能力：多数为毕业1-2年的人员
- 地域分布：中国
- 技术：C/S架构 主要应用于传统行业的数据计算

【注】：持续集成、工作分工、应用性质

项目痛点

- 老代码不敢动，一个函数40过个参数传入；
- 版本变更非常快，软件质量无法保证。

如何快速提升质量反馈？

止血优化

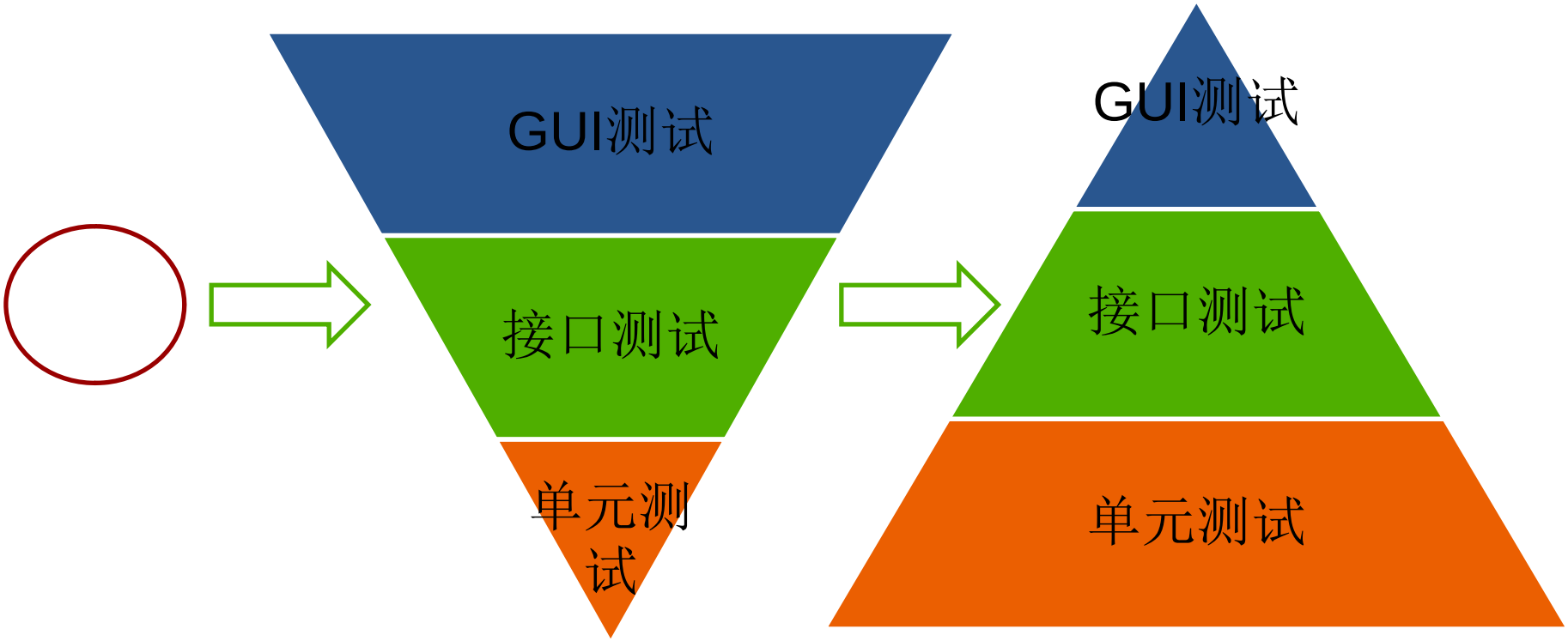
- 冒烟持续完善
- 复杂算法单元测试
- 自动化的三角图转化过程

结对

- 旧带新
- 核心模块设计编码
- 开发测试结对
- 业务与测试结对

估算与优先级

自动化的三角图转化过程



项目成果

成果：

持续成功维护40多个特性版本

开发测试需求和睦相处

不足：

系统在某客户现场出现严重性能问题，实施人员编故事。

议程

- 测试面临的困境
- 敏捷的核心
- 新项目从无到有敏捷实践过程
- 旧项目从无到有敏捷实践过程
- **敏捷推动的技巧**
- 关键敏捷测试实践

敏捷推动技巧

- 全局视图，痛点驱动
- 领导支持很重要，拿出数据说话
- 速赢与分步骤
- 80/20原则，别过度优化
- 不断反思，哪些地方可以进一步提升反馈

议程

- 测试面临的困境
- 敏捷的核心
- 新项目从无到有敏捷实践过程
- 旧项目从无到有敏捷实践过程
- 敏捷推动的技巧
- 关键敏捷测试实践

关键敏捷测试实践

A. 持续集成（首选实践）

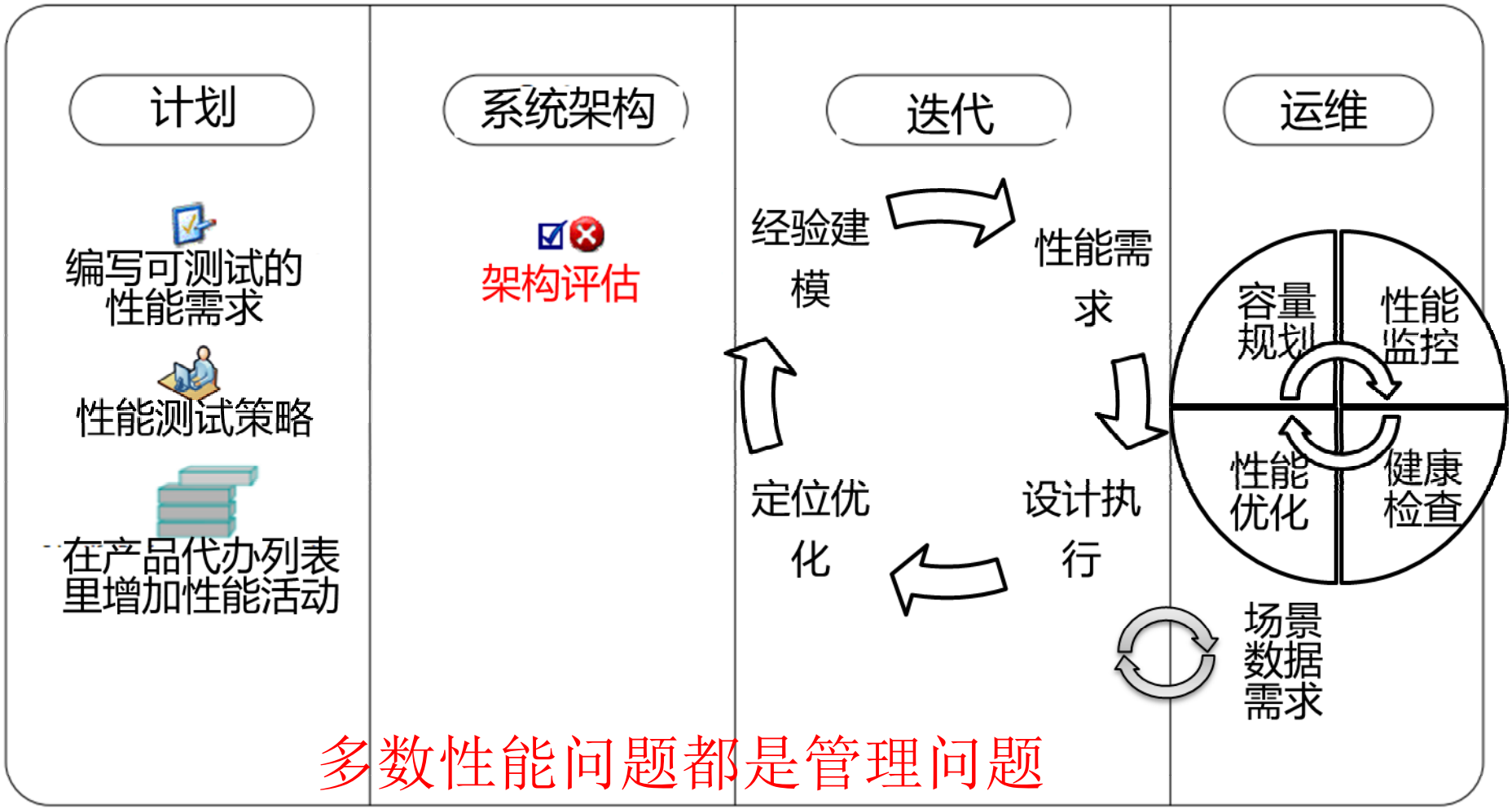
B. 冒烟测试再强调也不为过

C. 注重实效的单元测试

D. 有选择地结对

E. 研发与业务估算和优先级

全生命周期敏捷性能框架




站得高看得远，别把自己看成测试

人人都是产品经理（mini-CEO）

A. 工作的第一个项目：代码差 、工作1-2年的人员、抢占市场

B. 1个Bug引发的成本 ： 互联网 微软 银行

C. 自动化 ： 不注重技术投入？ ROI

D. 小团队最优不是最优：打架的绩效评估

E. 商业模式 竞争力？

JUST DO IT !