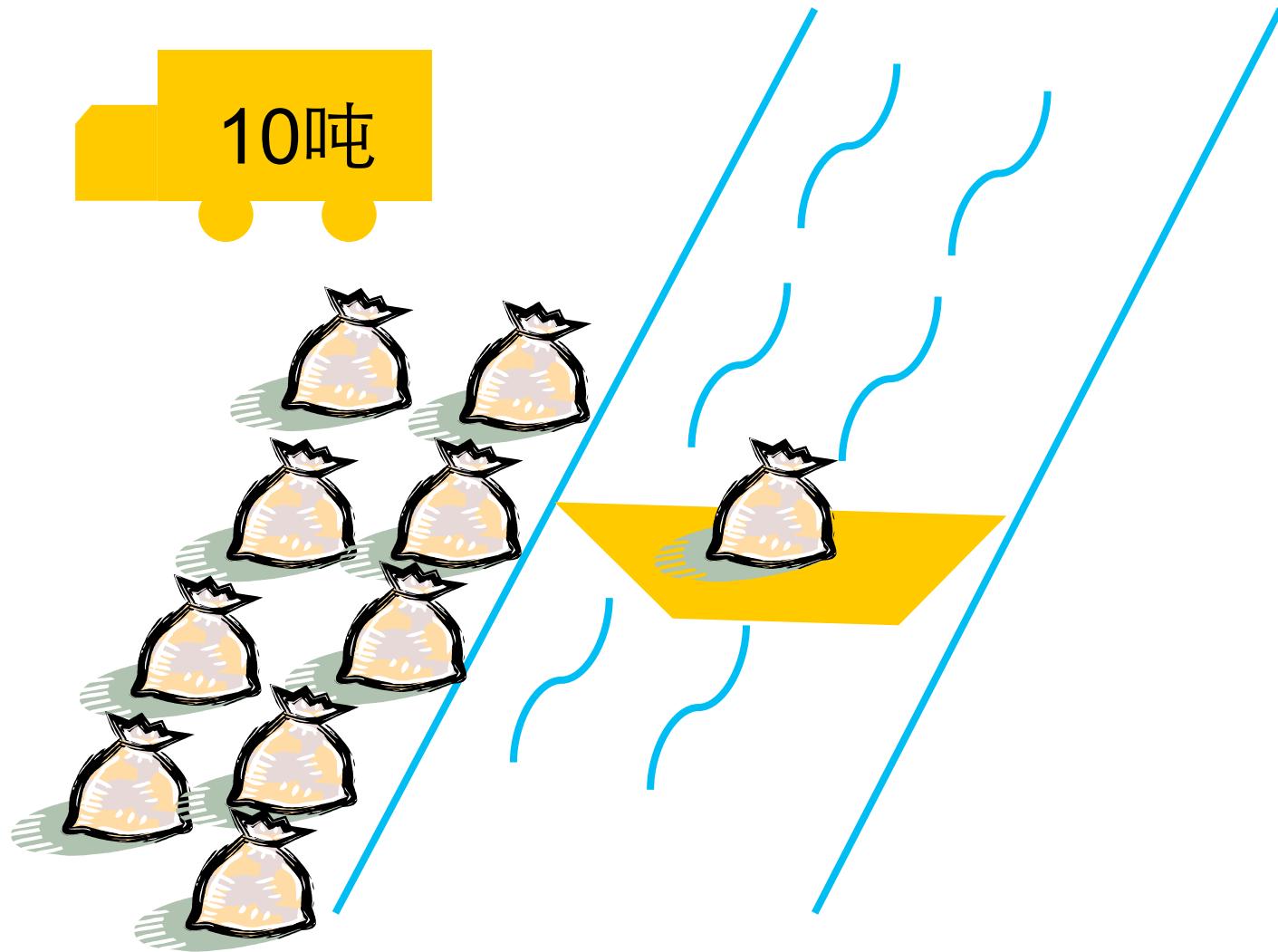


3000万代码行敏捷 项目的每日发布

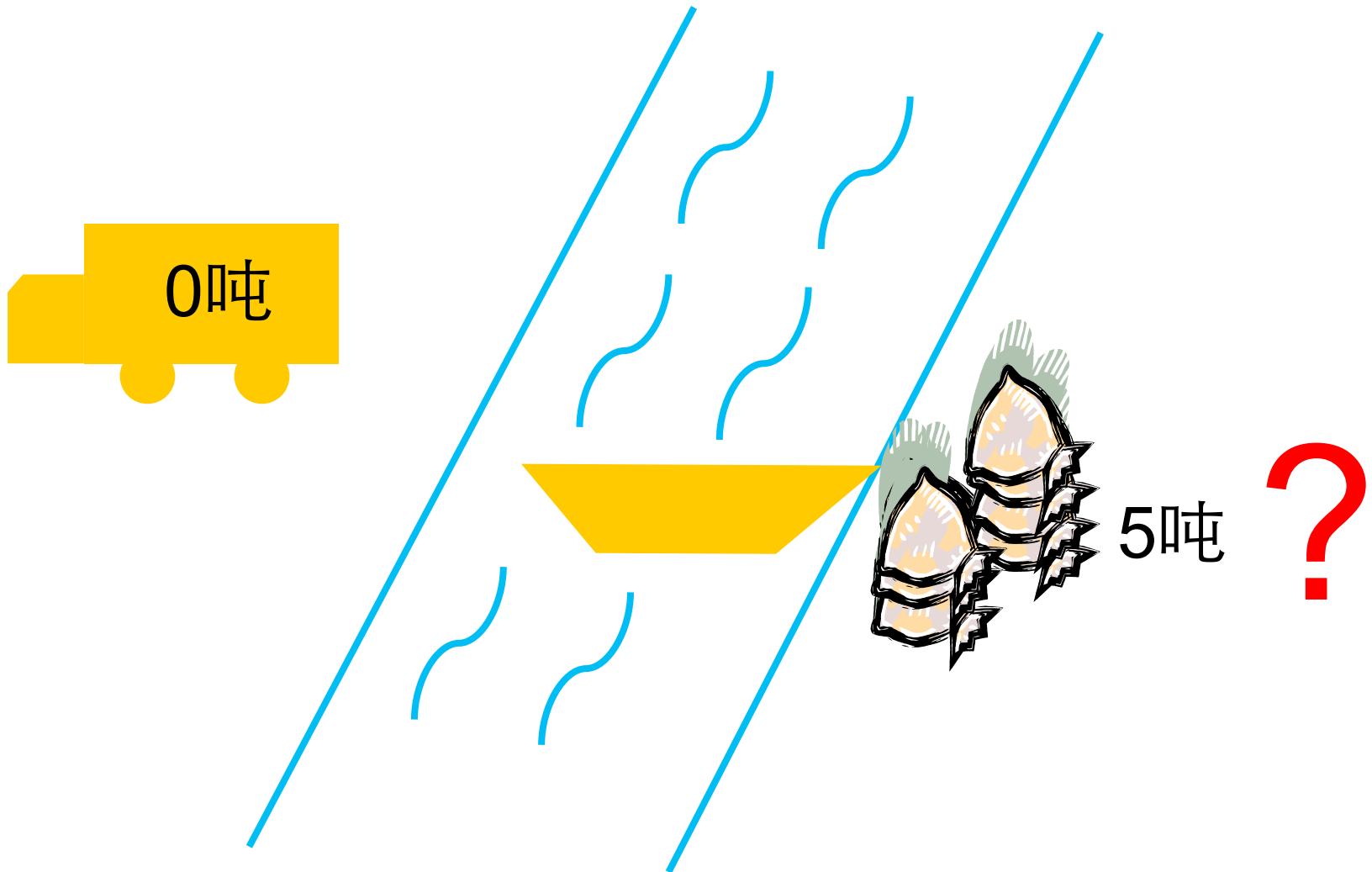
秦之远

诺基亚西门子持续集成**Product Owner**
<http://www.weibo.com/loveatbupt>

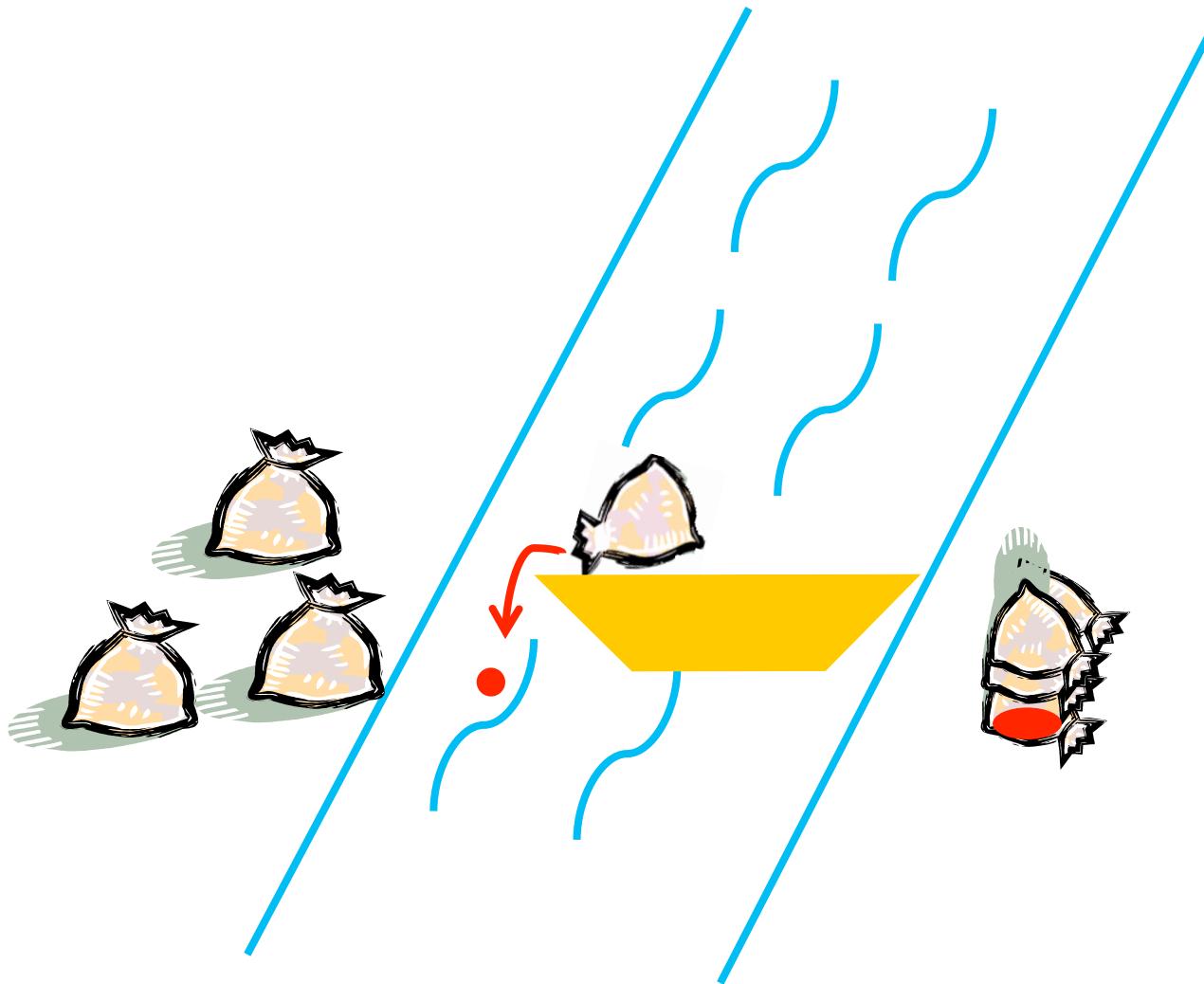
迭代开发与持续集成



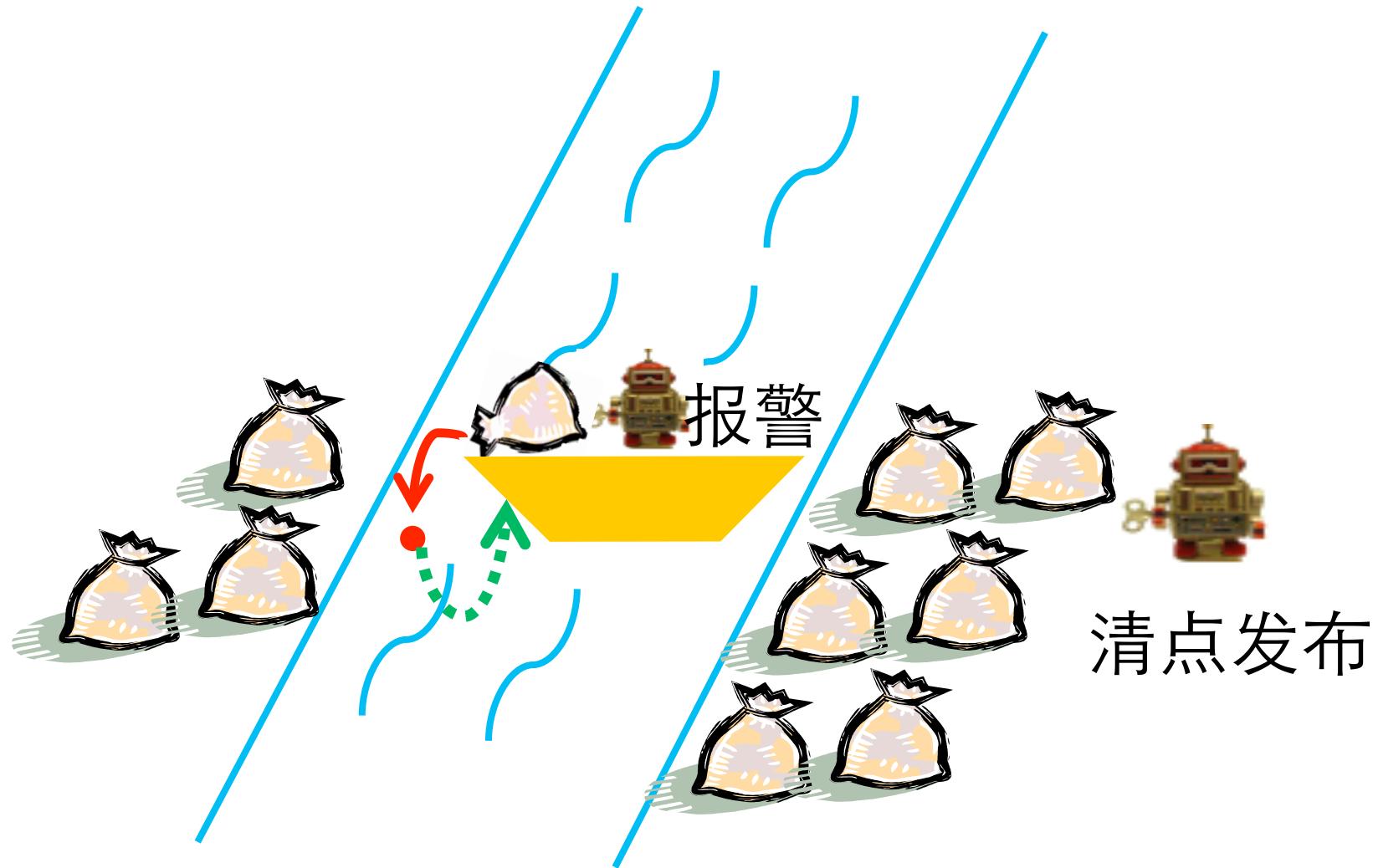
迭代开发与持续集成



迭代开发与持续集成



迭代开发与持续集成



持续集成基础 软件配置管理 (SCM)

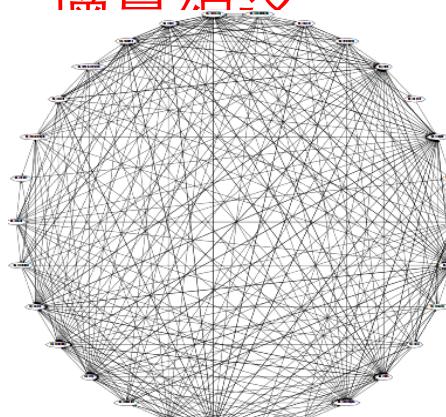
SCM与持续集成团队

基础1

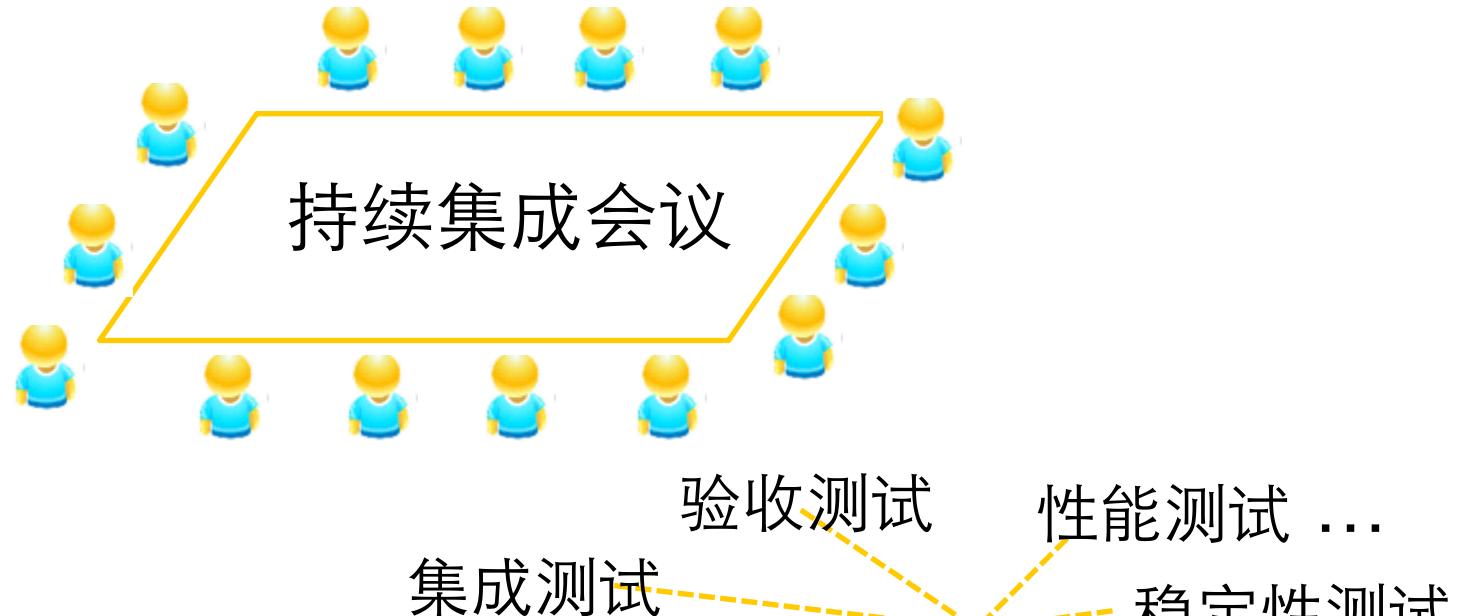
1. 代码仓库
2. 版本控制
3. 强大的编译服务器
4. 高性能编译器
5. 自动化发布和部署
6. 产品编译架构和模板

基础2

1. 支持单元测试
2. 代码分析
3. 并行编译(集群编译)
4. ~~项目管理~~



持续集成基础 测试自动化



单元测试

优点

成本低

缺点

测试环境无关

优点

不可替代

缺点

成本高
想象力差

持续集成基础 单元测试自动化

- 盲目追求单元测试覆盖率



三种不同单元测试	覆盖率
{A=5,B=1}	100%
{A=5, B=1}, {A=5,B=-1}, {A=5, B=0}	100%
{A=5, B=1} coding {A=5, B=-1} coding {A=5, B=0} coding code refactory	100%

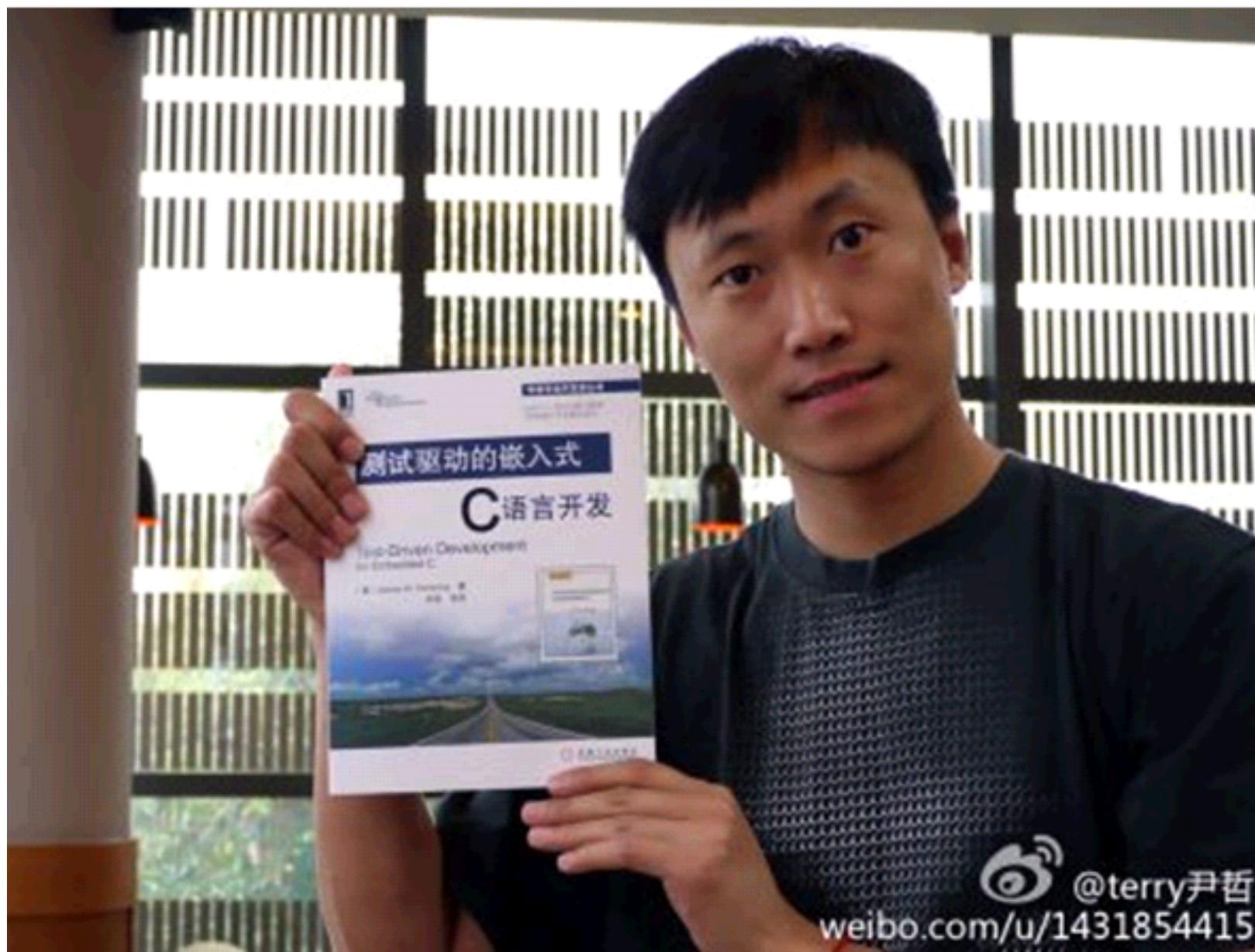
持续集成基础

单元测试自动化

- 昂贵的“单元测试”



测试方式	成本	效果
<ul style="list-style-type: none">• 桩函数• 让桩函数返回成功， 测试• 让桩函数返回失败， 测试	0.00001秒	代码逻辑全覆盖
<ul style="list-style-type: none">• 发消息， 测成功路径• “阻塞”内核， 测失败路径	0.1秒+内核崩溃风险	代码逻辑全覆盖



 @terry尹哲
weibo.com/u/1431854415

持续集成基础 功能测试自动化

Robot + Python

迭代式开发中，功能测试自动化必须充分预研究

基础1

1. 自动化测试工具
2. 自动化测试稳定性
3. 严格时间成本控制
4. 自动化测试编程规范
5. 有效的团队配合机制

基础2

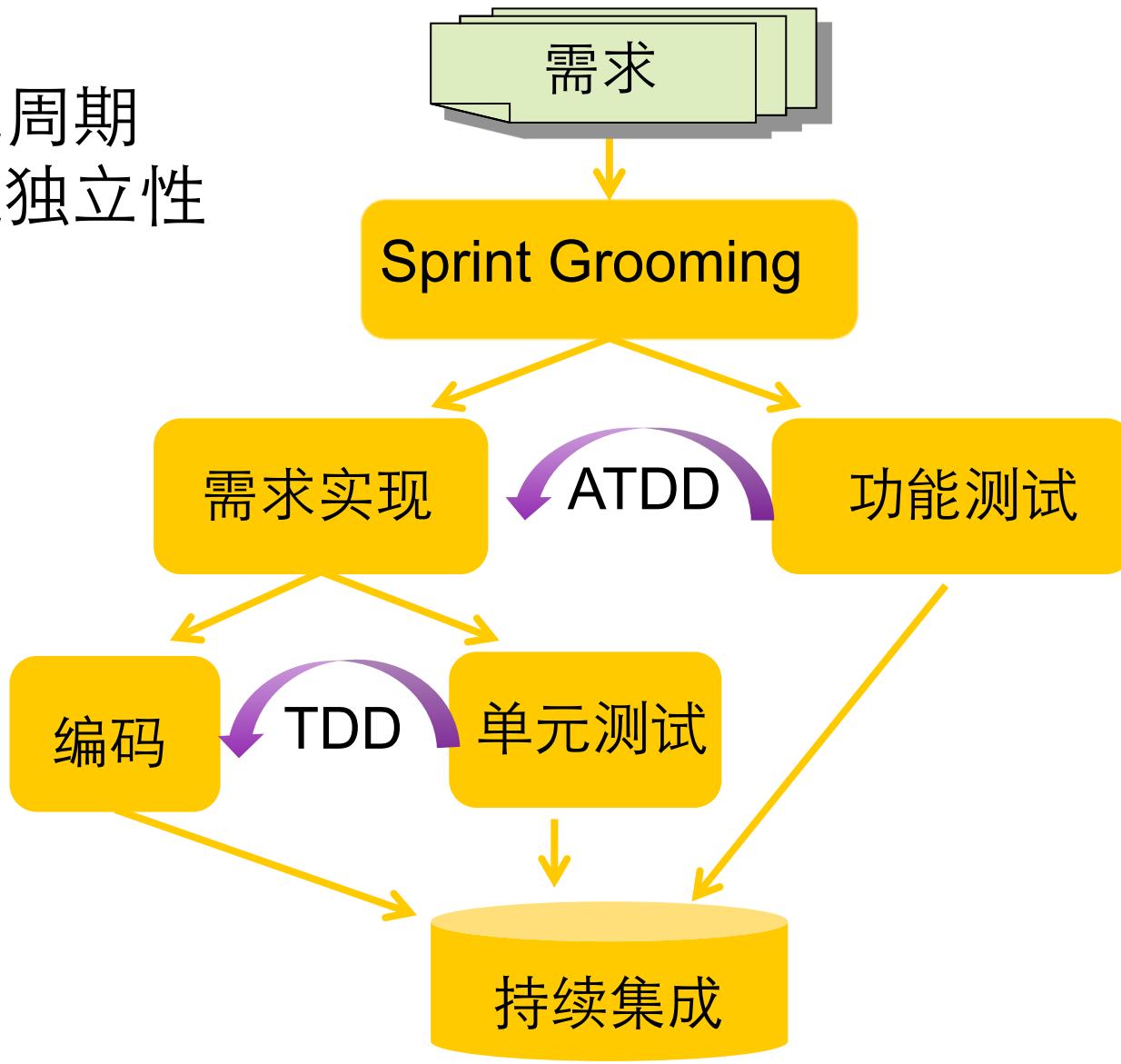
1. 自动化测试体系自身的模块划分
2. 自动化测试的静态分析
3. 自动化测试分类和管理
4. 自动化测试策略：
增量式功能自动化测试
PairWise测试
多任务测试框架
多场景测试框架
.....

持续集成基础 合理的系统模块划分

- 颗粒度
- 耦合度
- 详见分布式持续集成体系

持续集成基础 开发流程

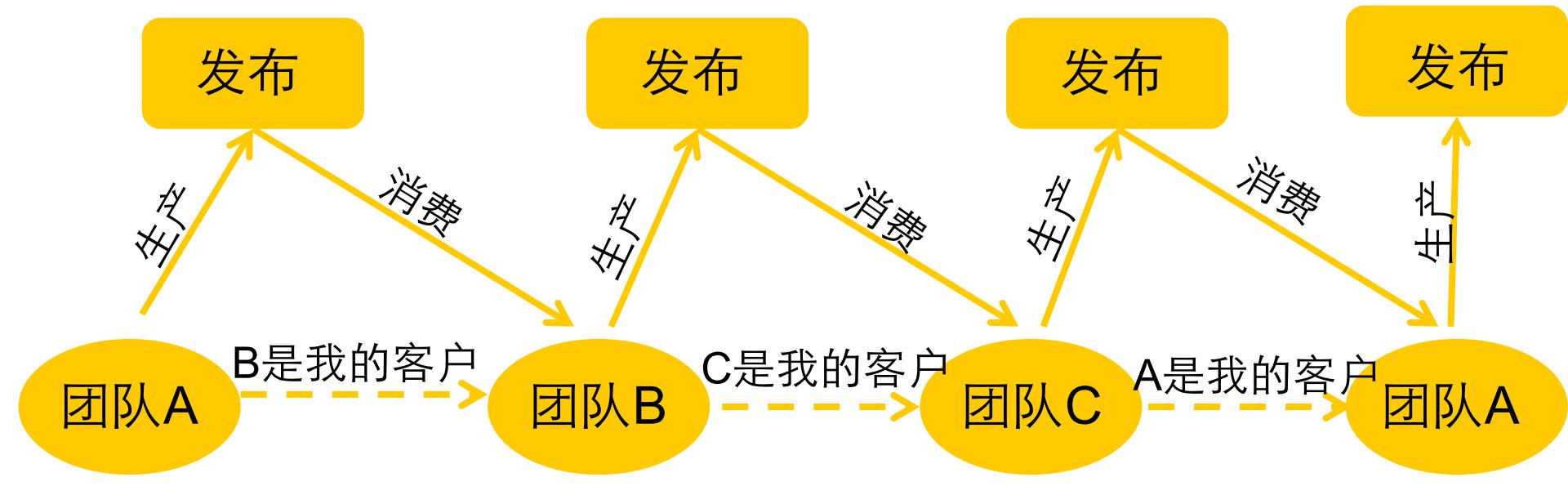
- 迭代周期
- 测试独立性



持续集成基础 工具

- 持续集成工具:推荐Jenkins
 - 免费
 - 界面简单
 - 插件多
 - 不支持批处理
- 需要具备基本的工具开发能

一套适合百人或小规模敏捷团队 简单持续集成体系



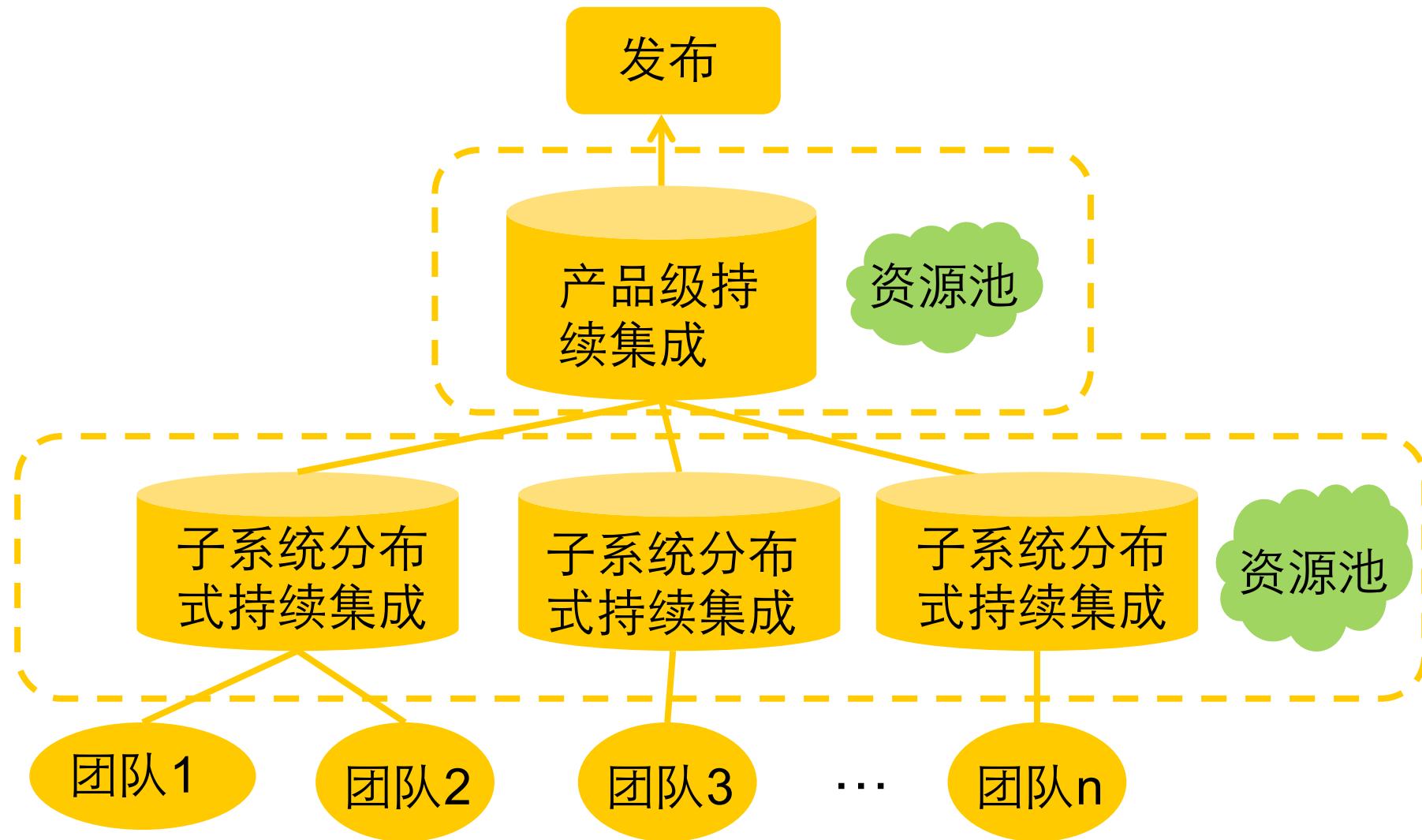
优点

- 培养团队持续发布意识
- 减少项目开发初期系统可测性不足带来的困扰，责任明确
- 持续集成系统自身架构简单

缺点

- 电信领域“消费”成本太高
- 敏捷团队越多，集成周期越长
- 集中式管理，沟通成本高

适合大规模敏捷团队 持续集成体系简介(L4模型实例)

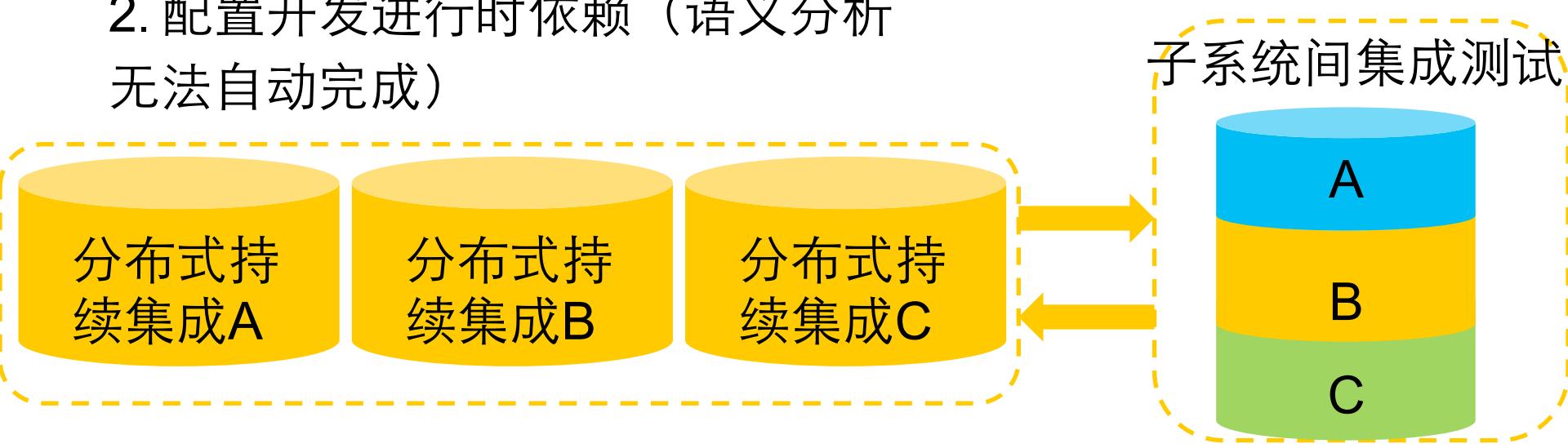


适合大规模敏捷团队 持续集成体系简介(L4模型实例)

- 简单：研发团队只需关注两件事
- 1. 研发人员只需频繁check in，无需任何“消费”成本。



- 2. 配置开发进行时依赖（语义分析无法自动完成）

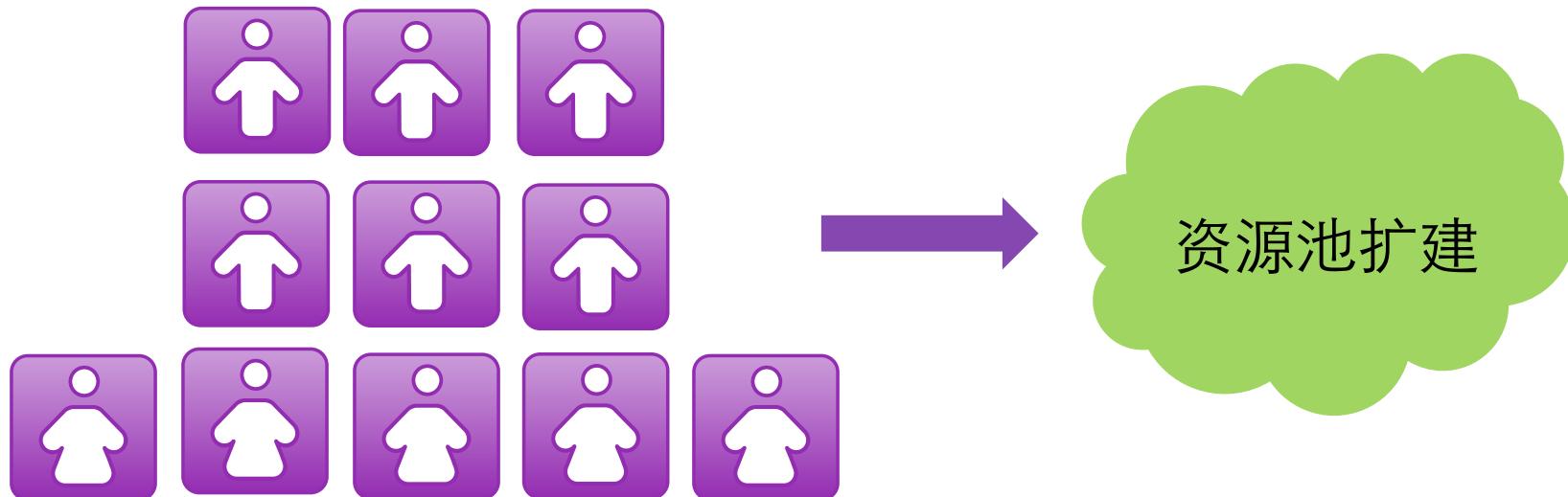


适合大规模敏捷团队 持续集成体系简介(L4模型实例)

- 易扩展

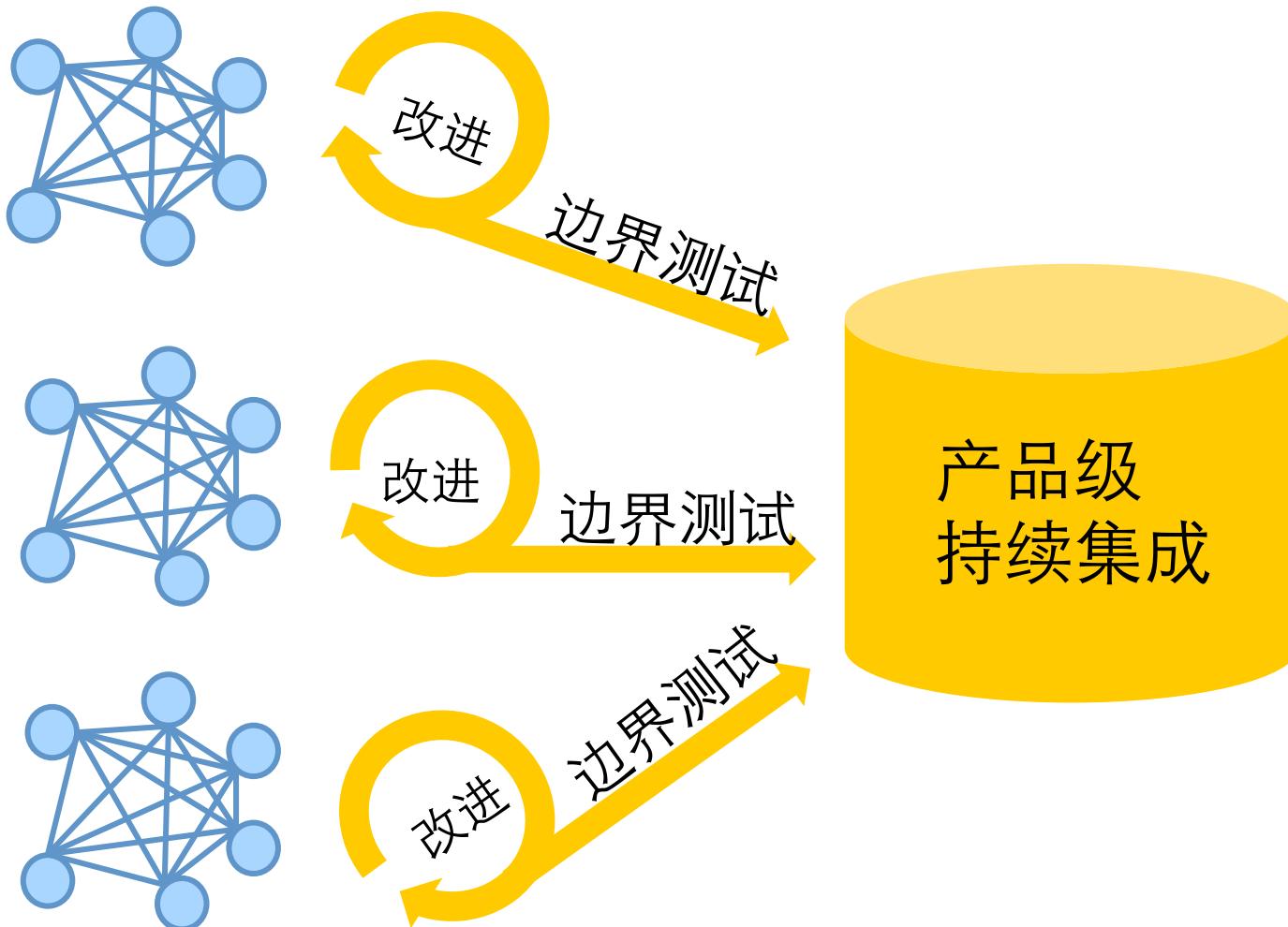
团队扩建只需扩展资源池，确保资源利用率最大化

新团队：



适合大规模敏捷团队 持续集成体系简介(L4模型实例)

- 自身推动团队持续交付转型



适合大规模敏捷团队 持续集成体系简介(L4模型实例)

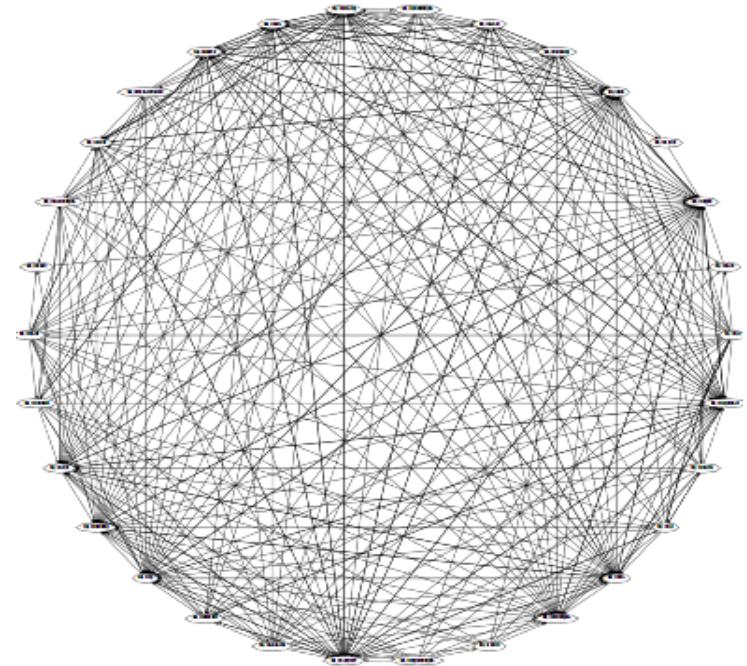
- 项目开发可见性



适合大规模敏捷团队 持续集成体系简介(L4模型实例)

分布式持续集成系统无法规避问题:

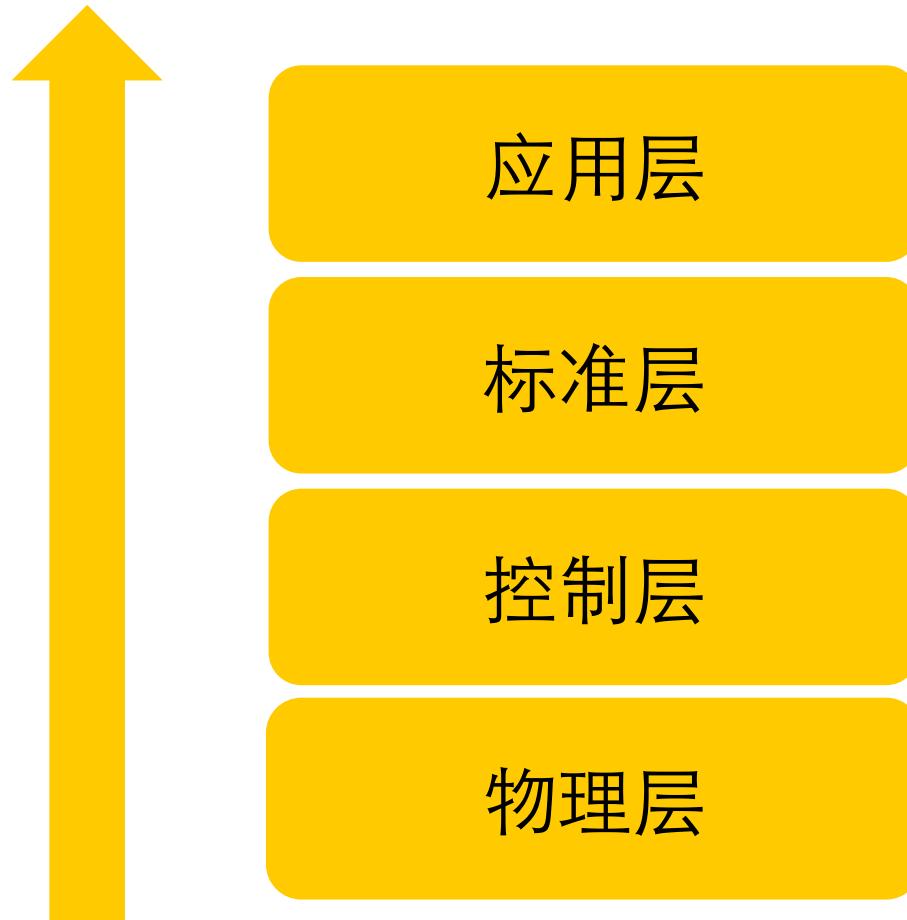
- 产品模块之间耦合性
 - 模块颗粒度
 - 模块耦合性
- 测试自动化各模块之间耦合
 - 用例与库
 - 系统各模块间用例



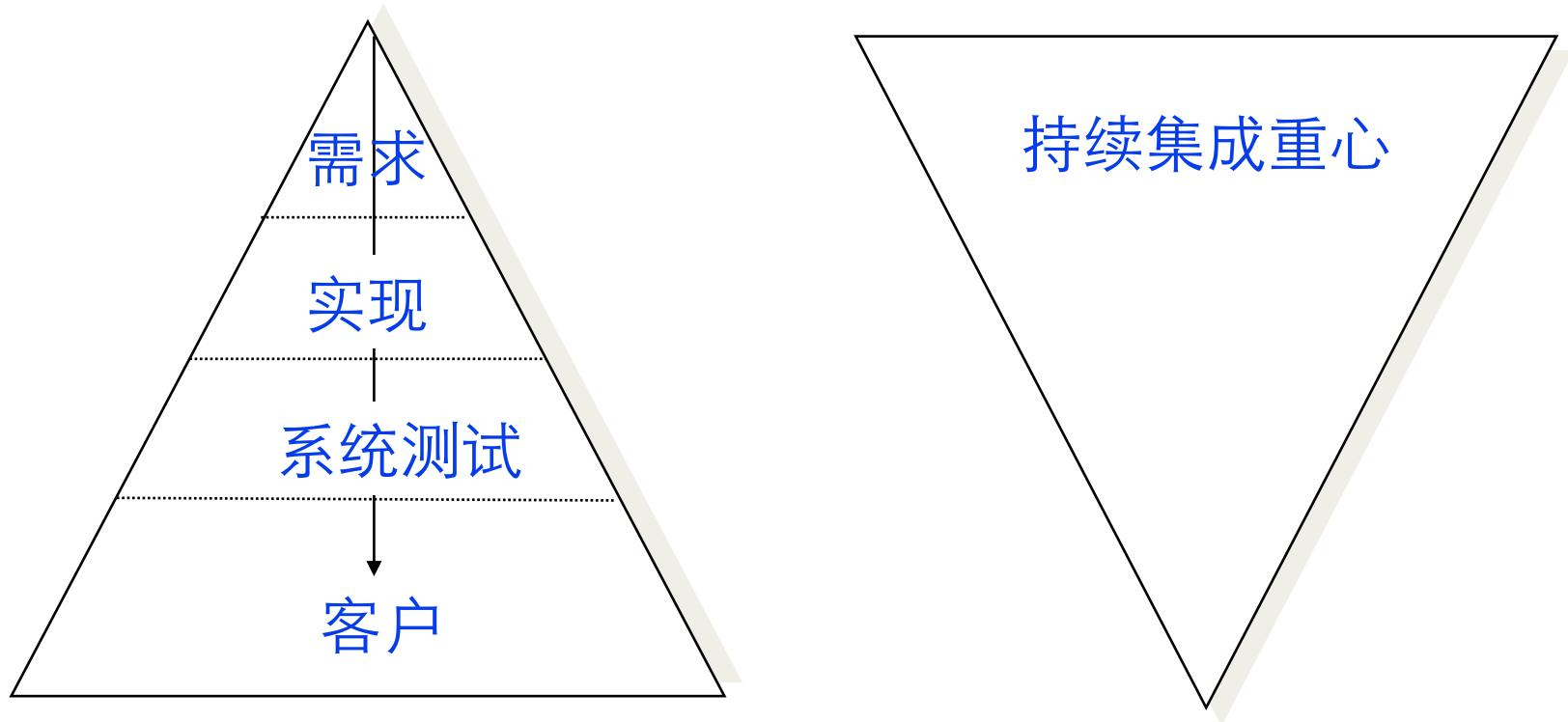
不合理的模块耦合性，破坏了持续集成的
“颗粒度”，频繁的引起构建失败。

持续集成系统L4模型

持续集成四层系统模型



产品质量与持续集成重心 理论模型

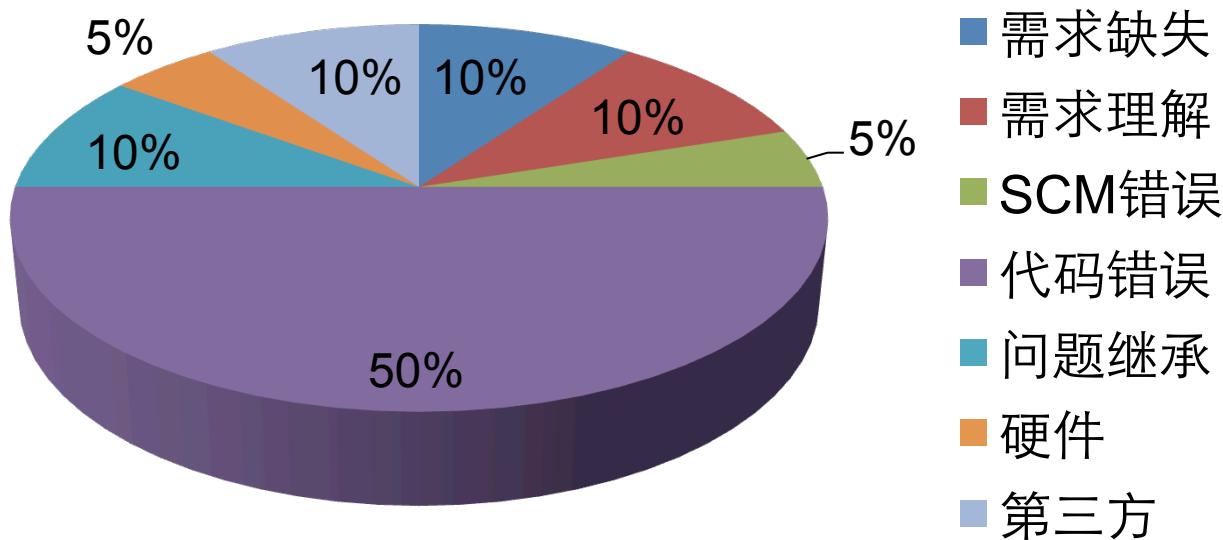


软件问题在各个不同阶段修复成本

产品质量与持续集成重心模型

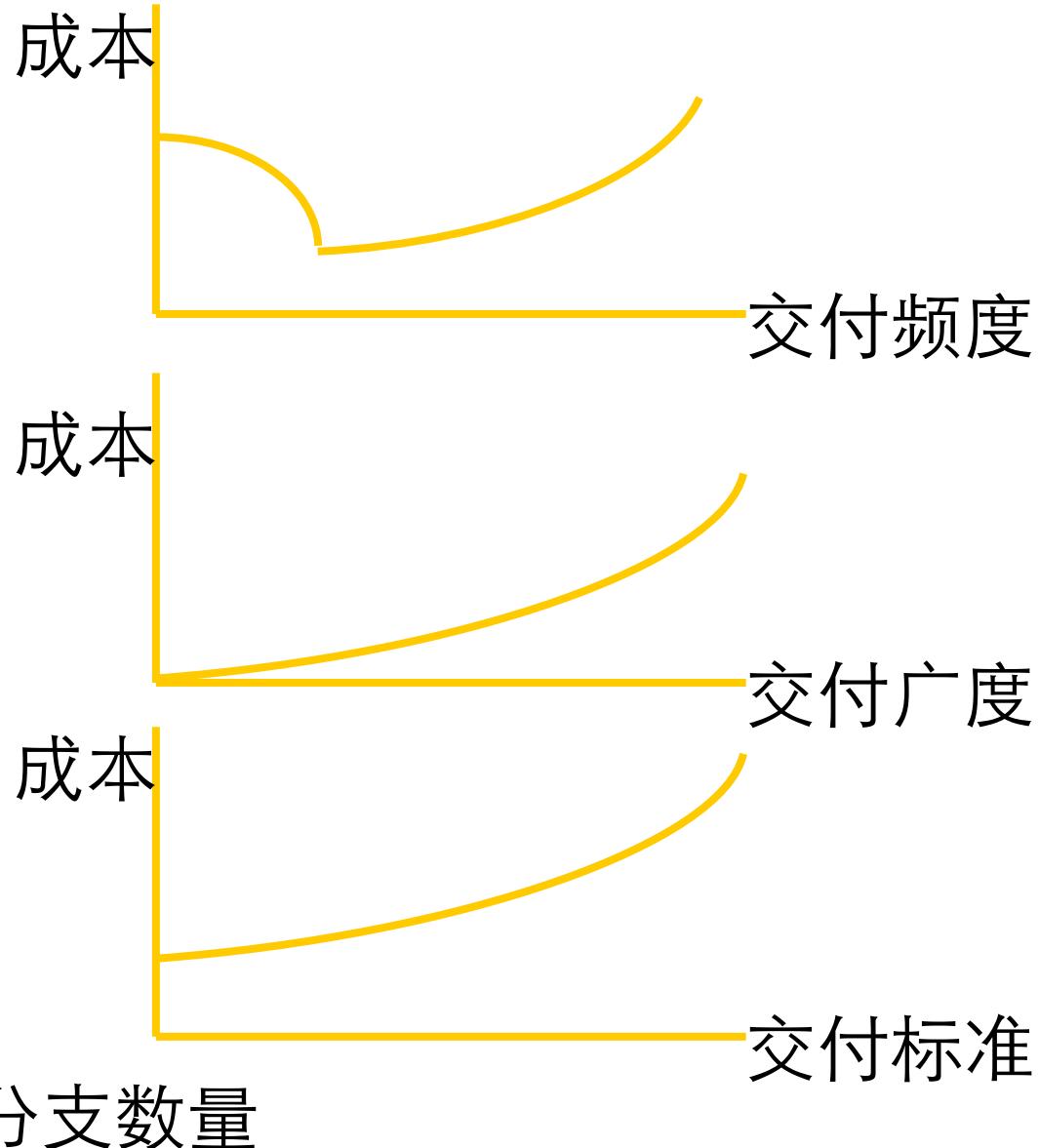
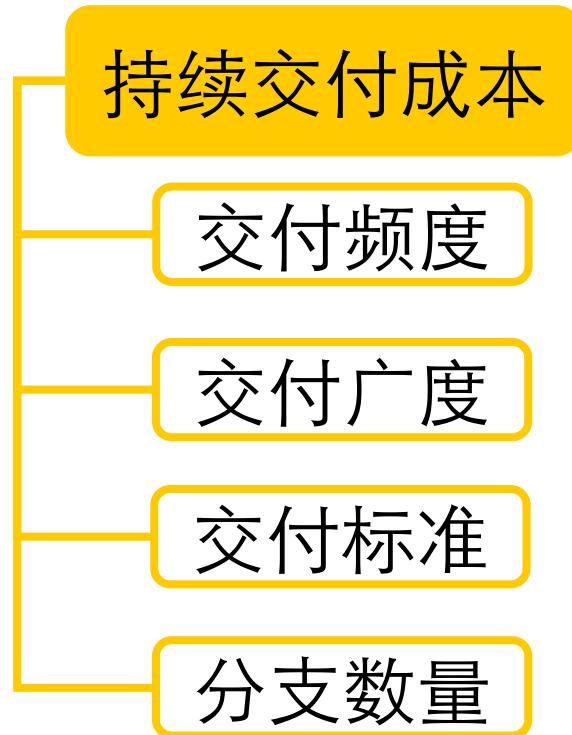
实际产品质量模型

质量模型

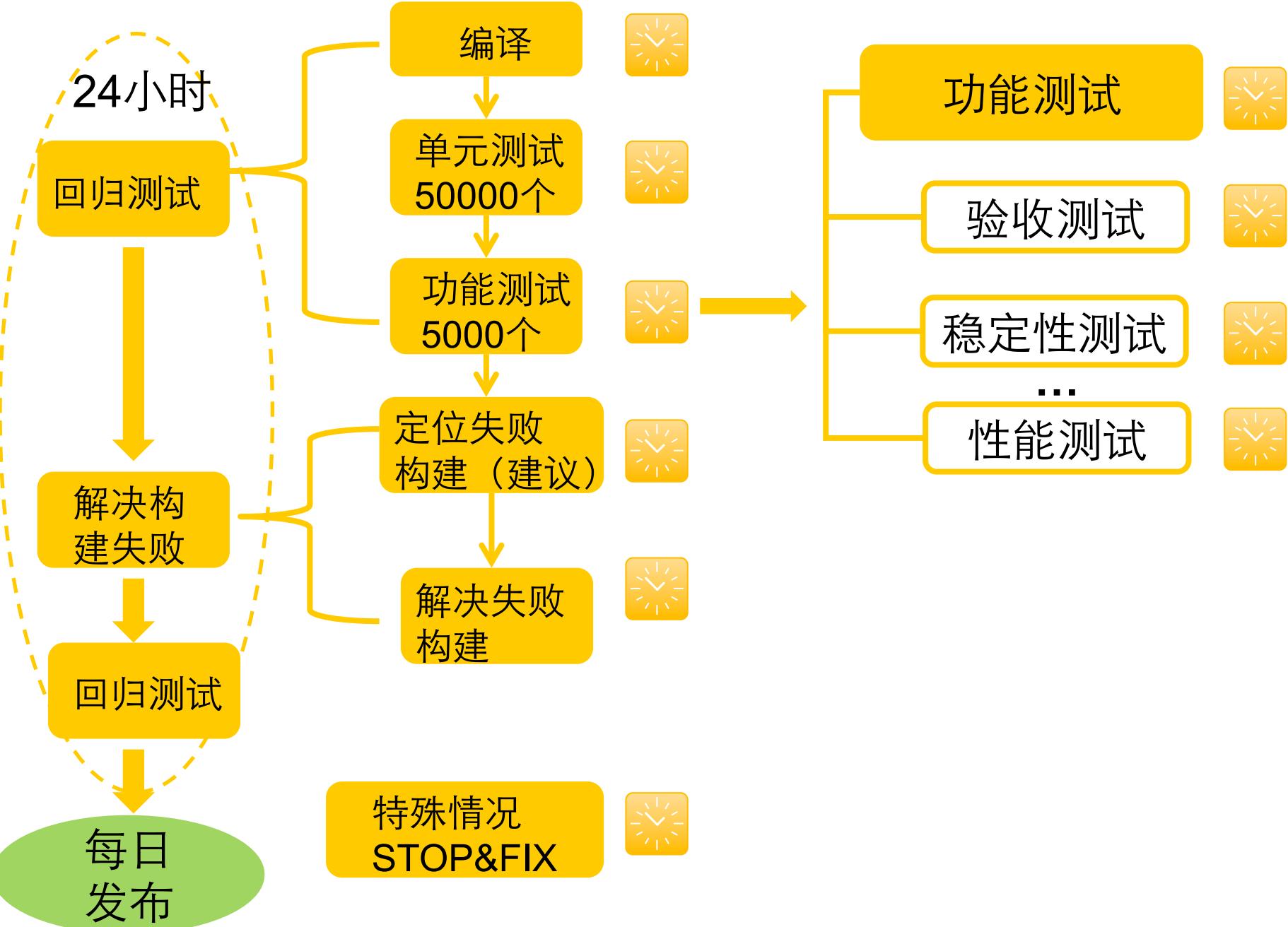


根据项目规模和具体产品问题爆发阶段，可以在薄弱阶段加强持续集成

产品持续交付的成本剖析



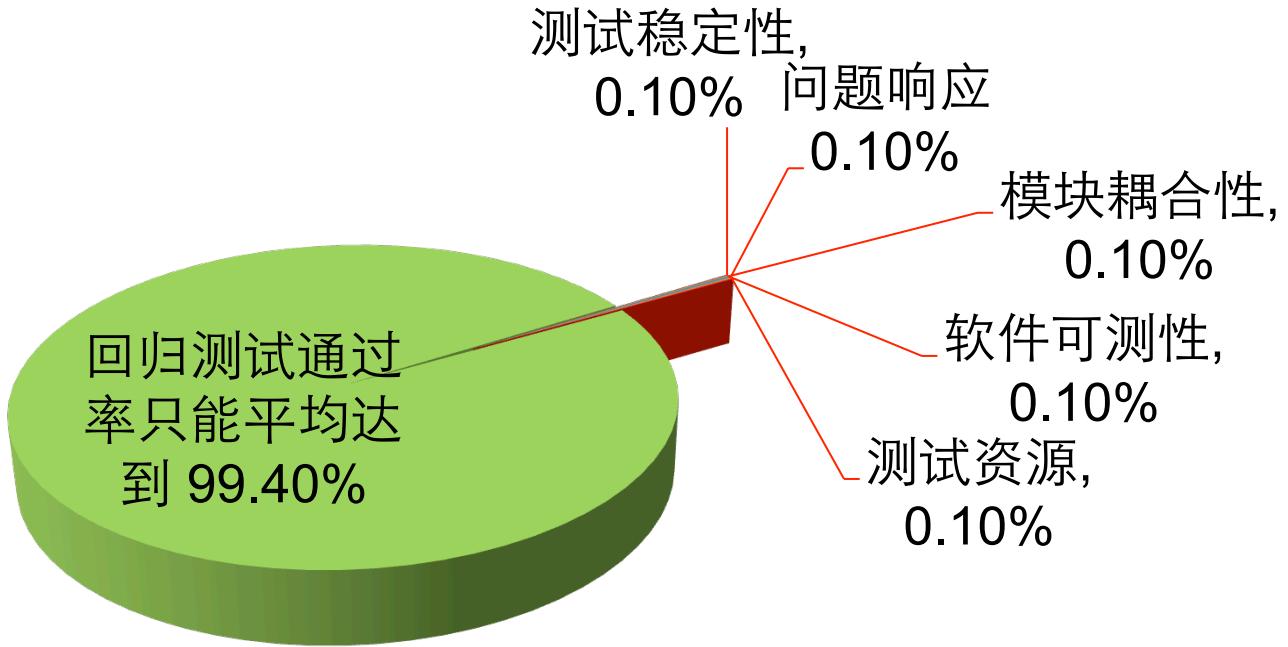
产品持续交付成本剖析实例



产品持续交付成本剖析

交付标准

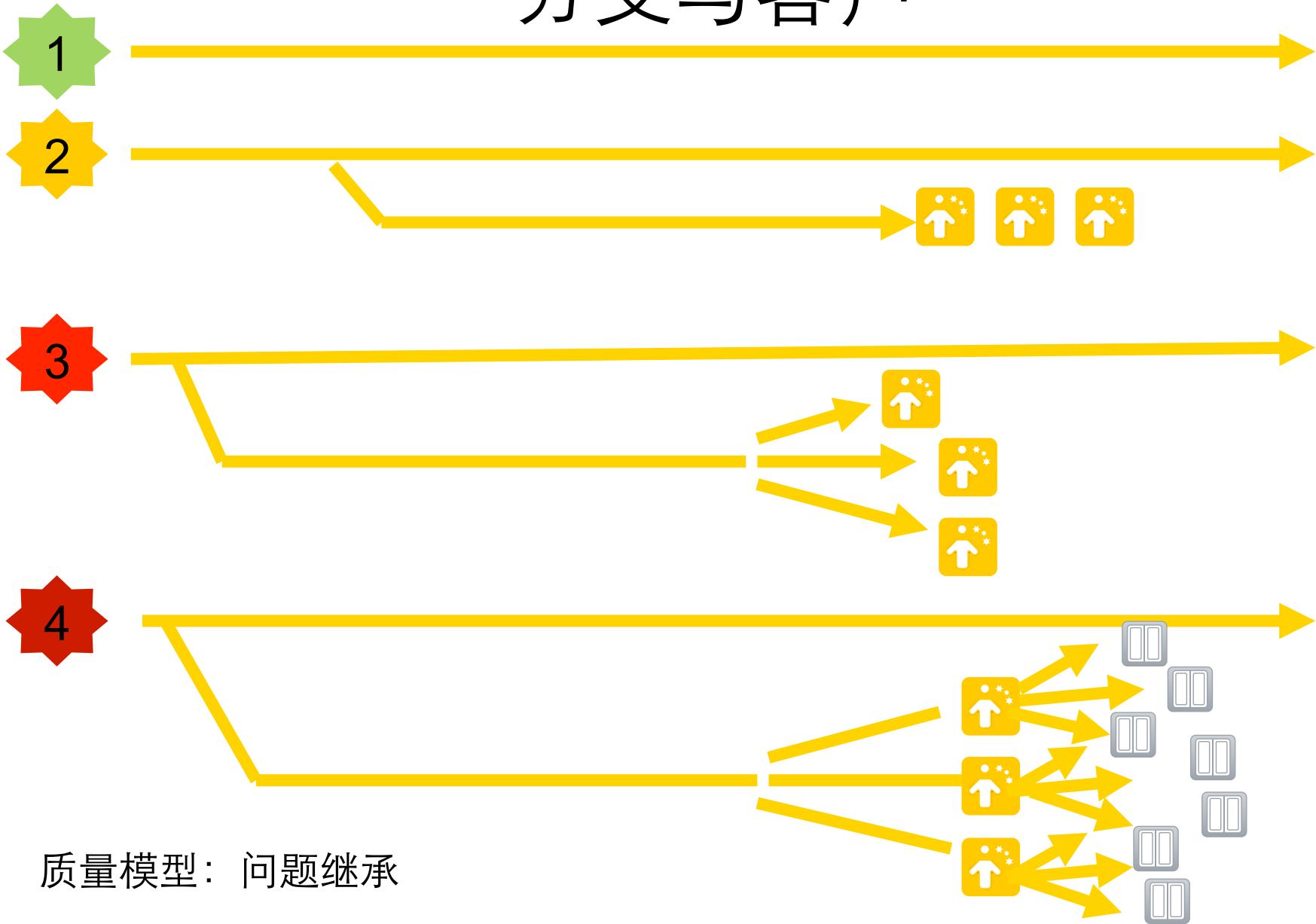
交付标准



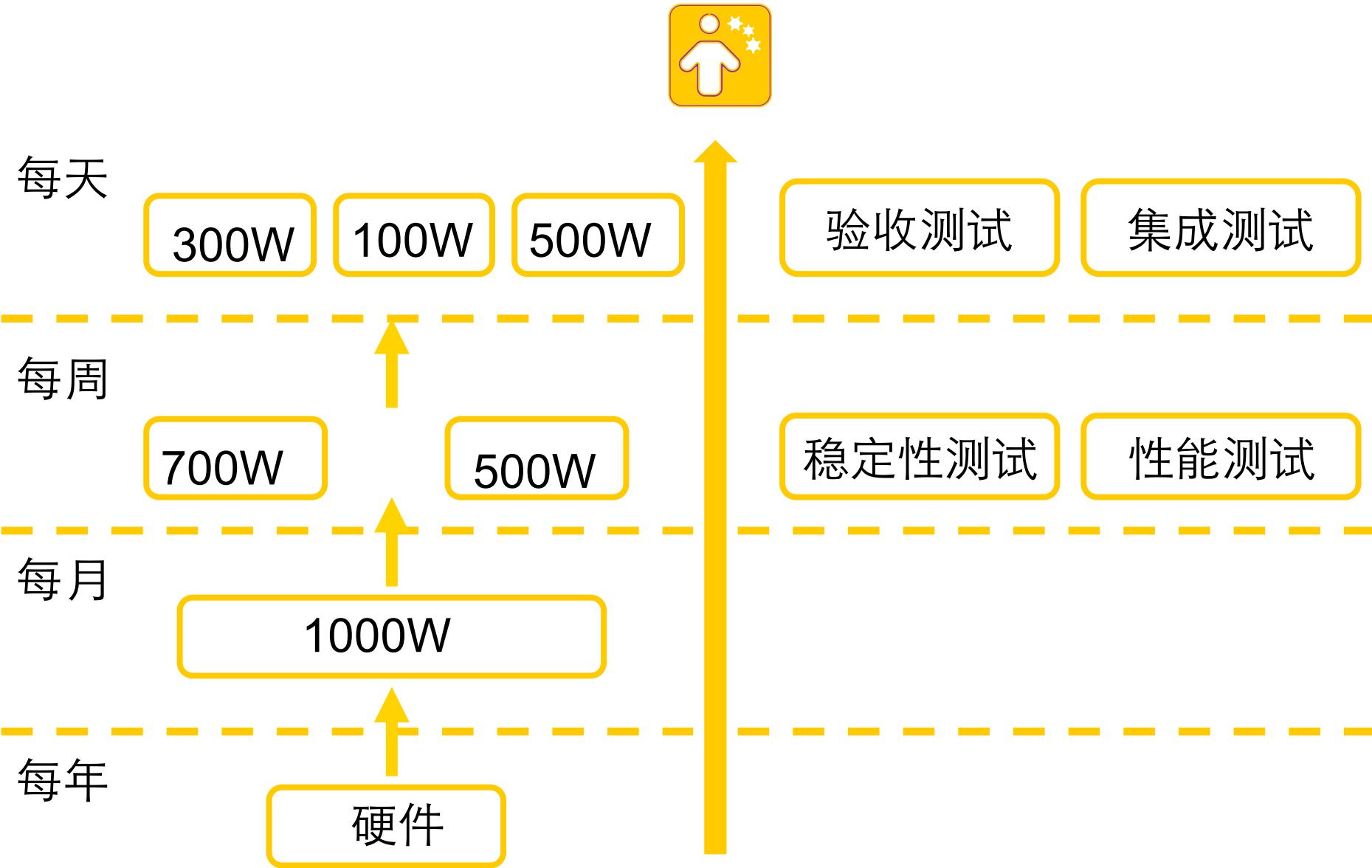
每天的99.4%保证了任意一次客户正式发布的100%

产品持续交付成本剖析

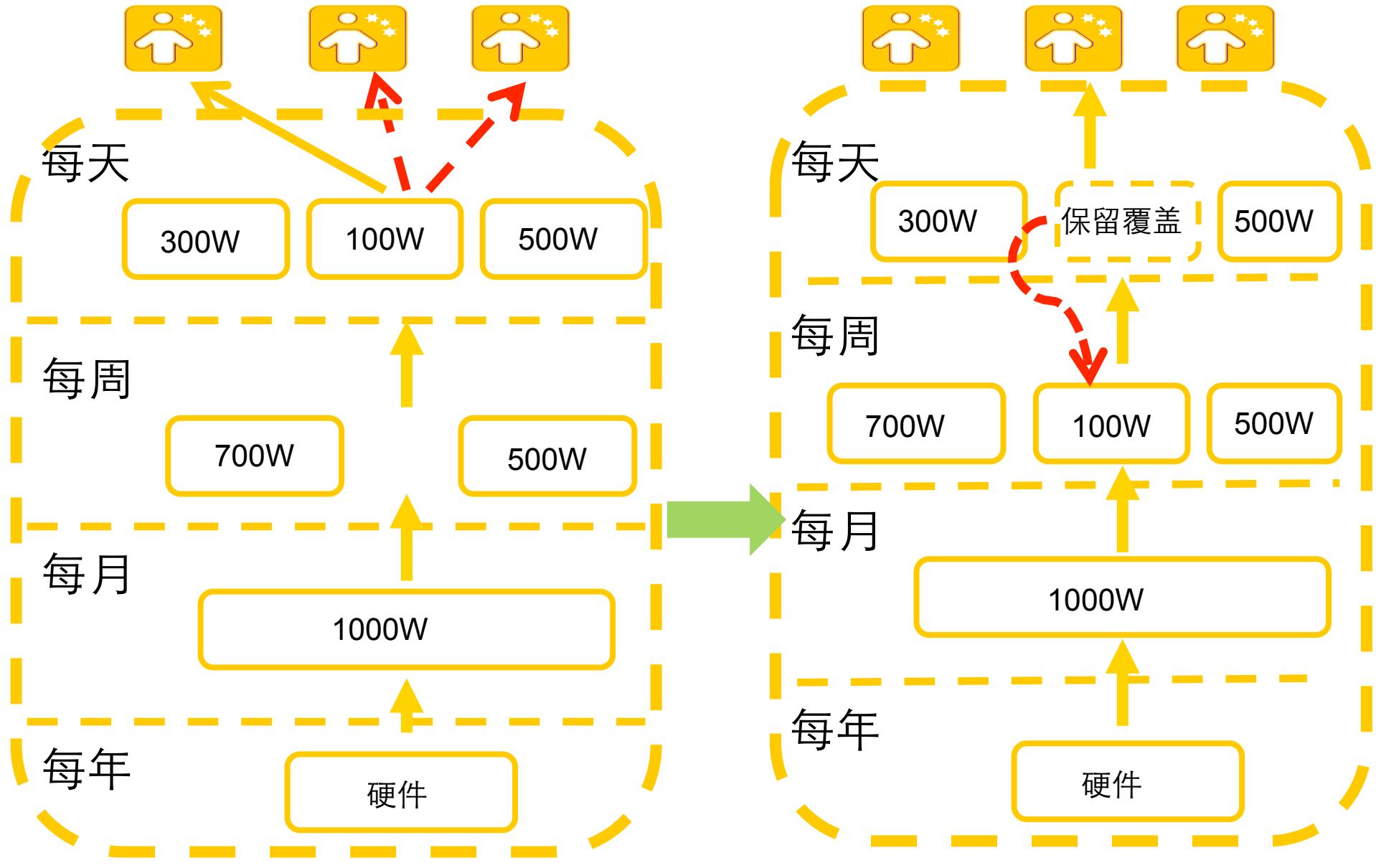
分支与客户



有限成本下的持续交付策略



有限成本下的持续交付策略



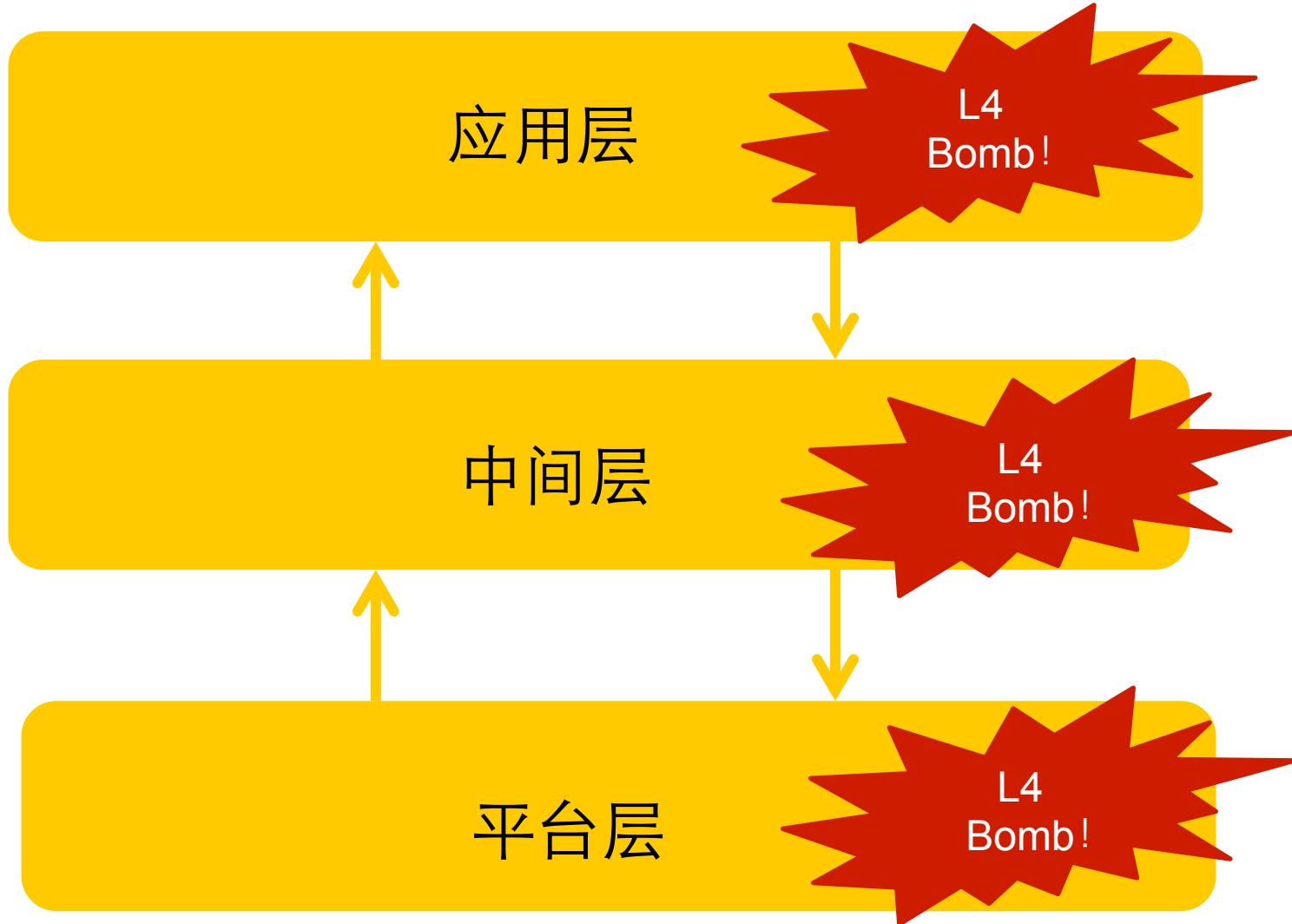
持续集成自身风险

- 信用危机

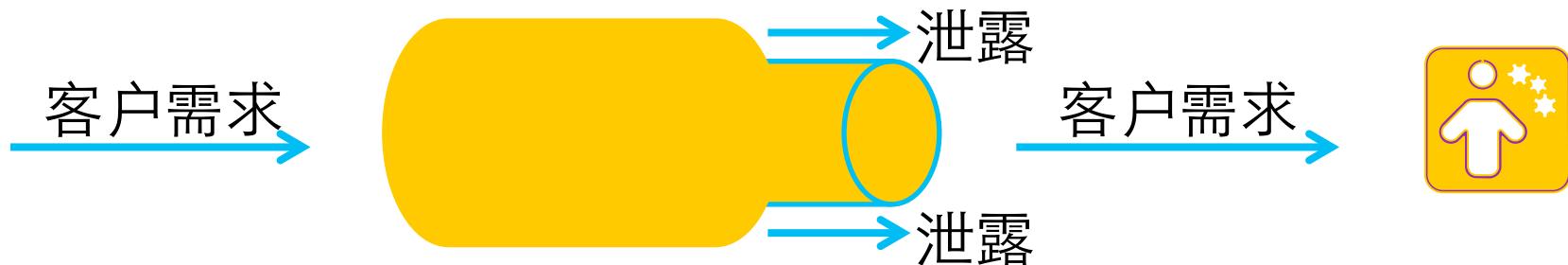
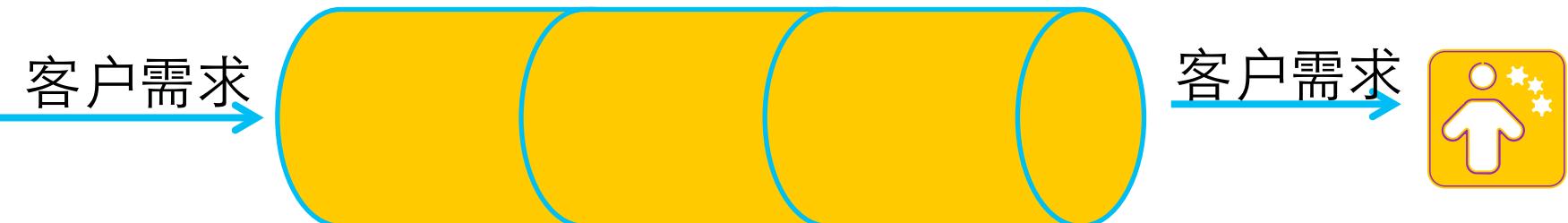
持续集成系统 > (功能测试自动化 = 单元测试自动化)



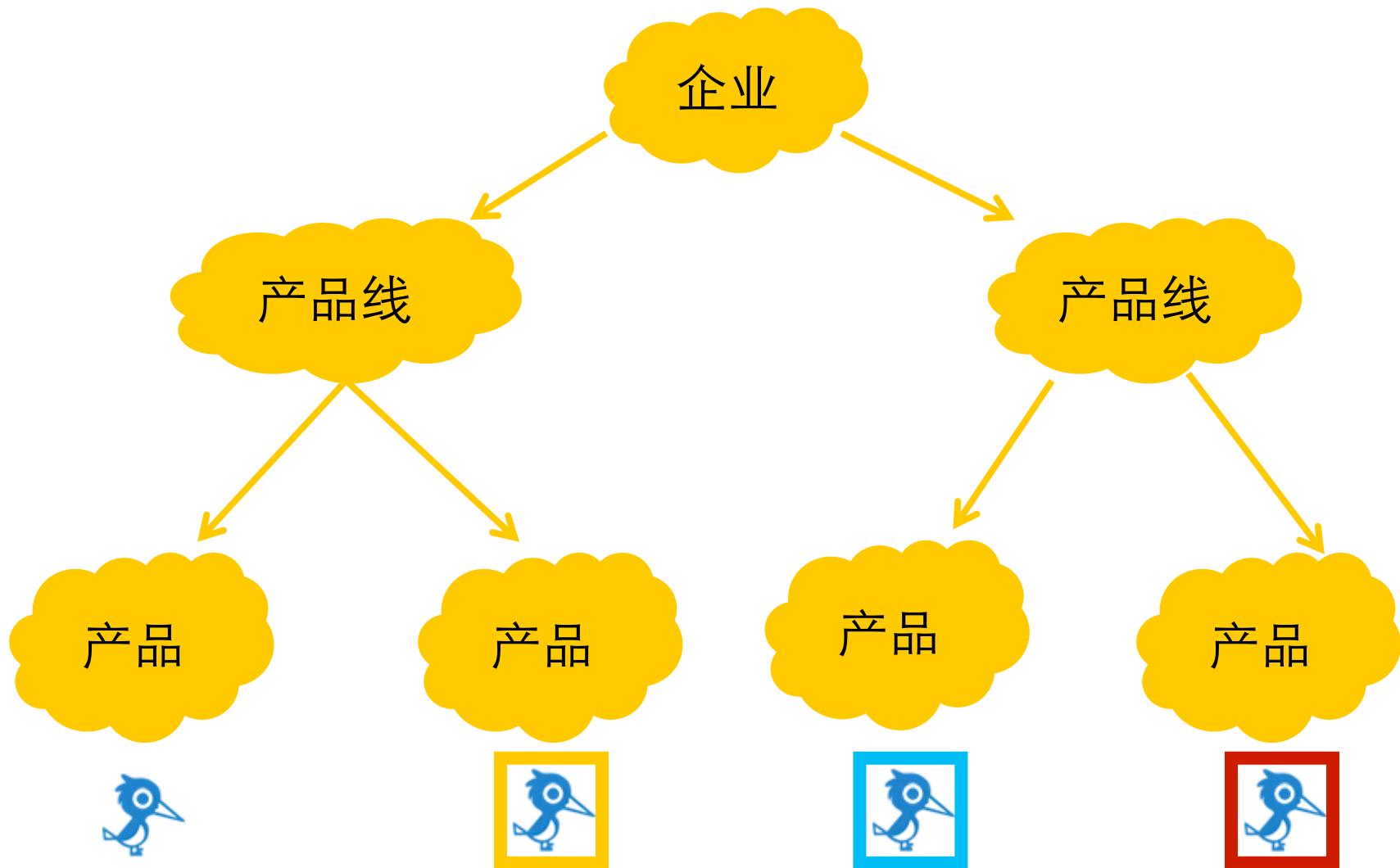
产品内部持续交付转型



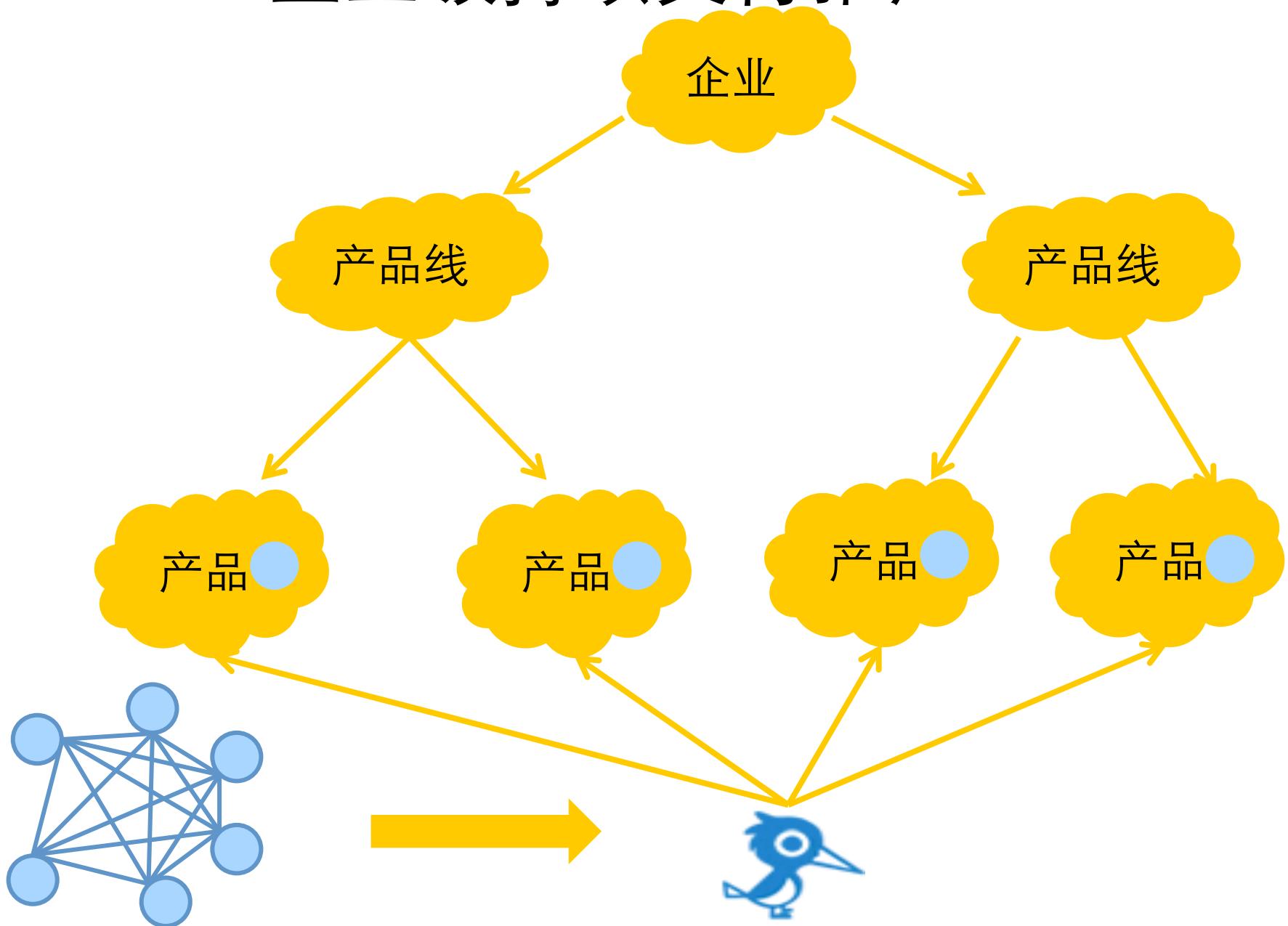
产品内部持续交付转型



企业级持续交付推广



企业级持续交付推广





Cugesoft



窦涵之

感谢敏捷公司Cugesoft
提供的Logo

持续集成团队的自身 持续改进与评价

- 确保产品持续发布
- 确保发现更多的问题

在实际操作过程中，持续集成团队应该建立自身的一套衡量标准来推动自我改进。

Q&A

谢谢！



ArchSummit

中国·深圳 2012.08

INTERNATIONAL ARCHITECT SUMMIT

全 球 架 构 师 峰 会

详情请访问: architectsummit.com

•3天 •6场主题演讲

•3场圆桌论坛 •9场专题会议

•国内外30余家IT、互联网公司的50多位来自一线的讲师齐聚一堂

主办方: InfoQ

战略合作伙伴: Tencent 腾讯

特别支持:



<http://architectsummit.com>



杭州站 · 2012年10月25日~27日
www.qconhangzhou.com (6月启动)

QCon北京站官方网站和资料下载
www.qconbeijing.com

