

ThoughtWorks®

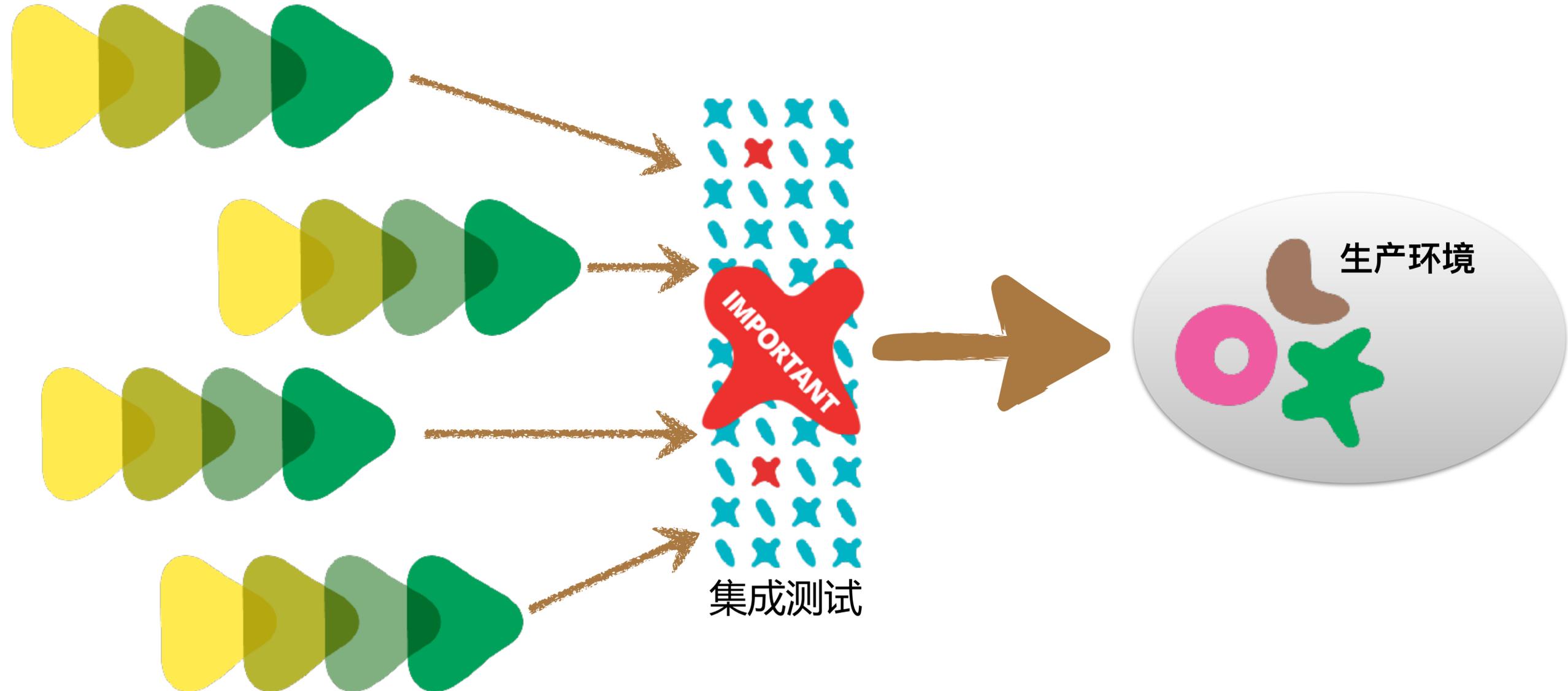
testwo 自由开放分享
测试窝

IT大咖说
知识分享平台

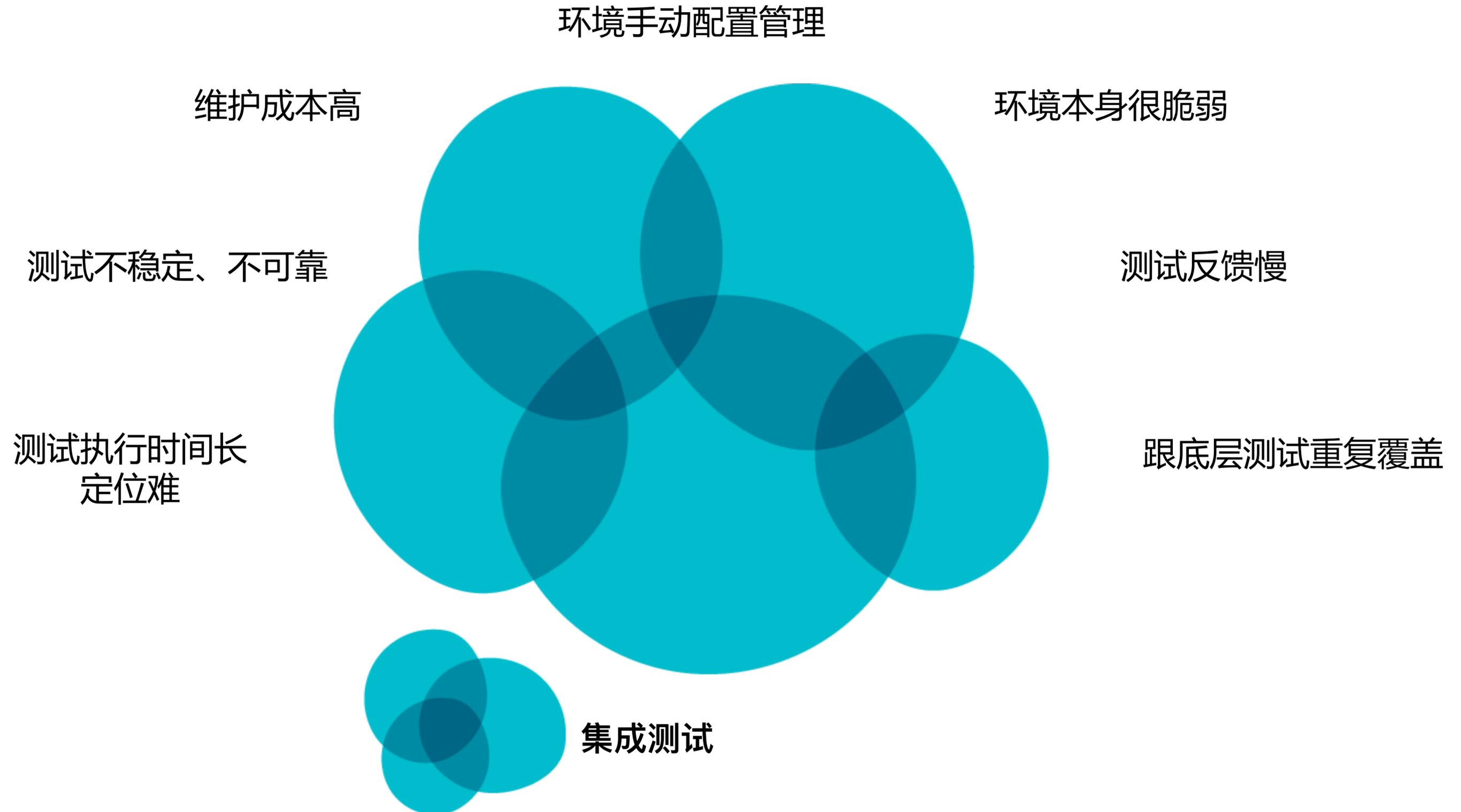
Testwo 测试沙龙

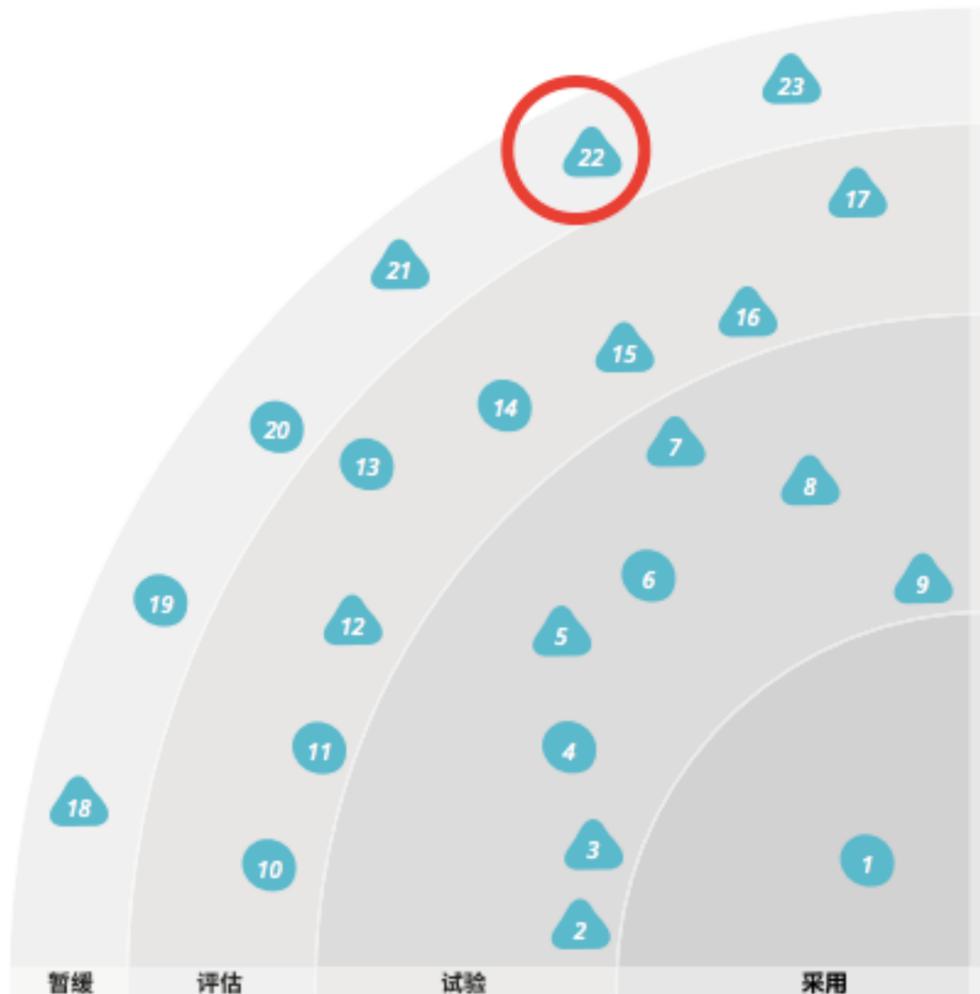
系统级集成测试的断舍离

ThoughtWorks 林冰玉



但是，集成测试成为了持续交付的瓶颈





<https://www.thoughtworks.com/radar>

暂缓

- 18. A single CI instance for all teams
- 19. Anemic REST
- 20. Big Data envy
- 21. CI theatre **NEW**
- 22. Enterprise-wide integration test environments **NEW**
- 23. Spec-based codegen **NEW**

企业级应用的整体的季度或月度的版本发布被认为是该领域的最佳实践, 在部署到生产环境之前维护一个完整的环境以进行测试是非常有必要的。这些**企业集成测试环境**通常称为 SIT 或预生产环境) 是当下持续交付常见的瓶颈。环境本身很脆弱而且维护成本很高, 而这些环境通常存在一些需要由单独的环境管理团队手动配置的组件。在预生产环境的测试给出的反馈慢且不可靠, 而且会重复测试那些在隔离的组件上已经测过的功能。我们建议企业以增量的方式为关键组件创建一条通向生产环境的独立途径。其中涉及到一些重要的技术包括**契约测试**, **将发布与部署解耦**, **专注于平均恢复时间和生产环境下的 QA**。

契约测试

发布与部署解耦

专注平均恢复时间

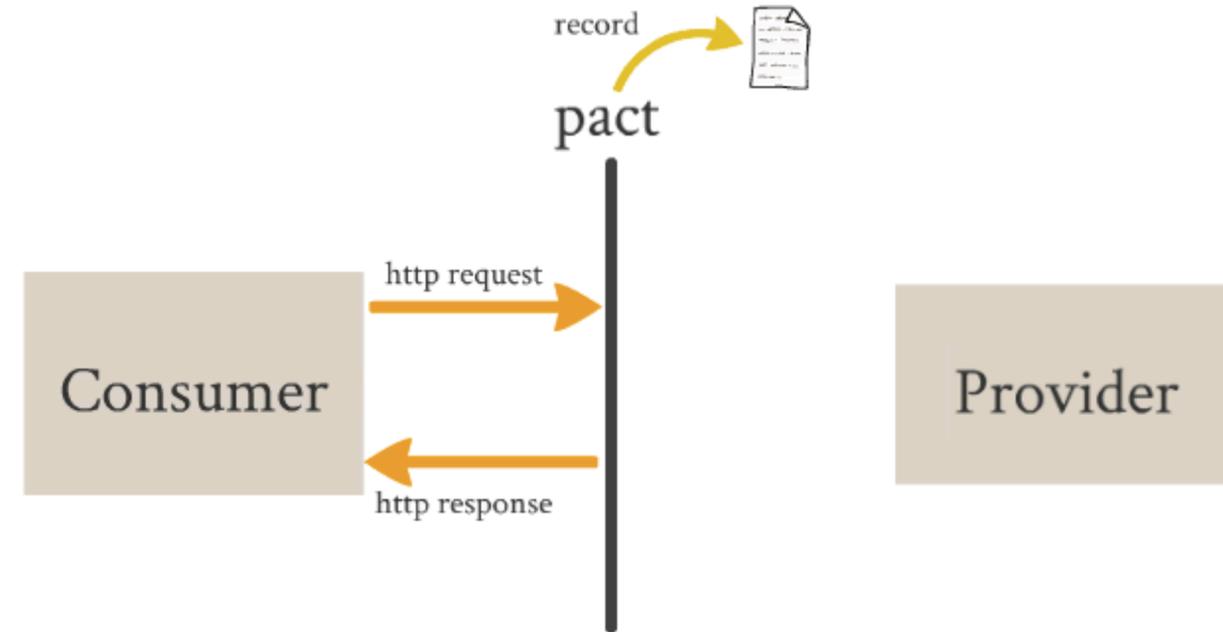
生产环境下的QA

Techniques

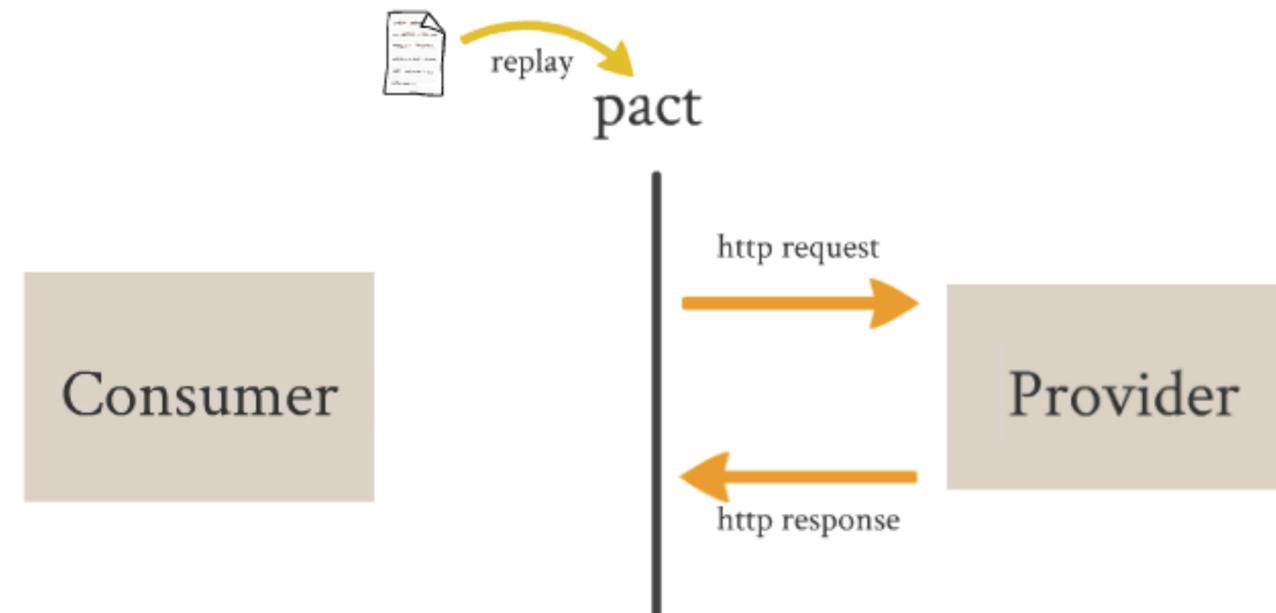
Consumer-driven contract testing

- 微服务测试的重要部分
- 使得独立部署成为可能
- 是一个技术和一种态度，并不需要特别的工具
- 实践重于框架
- 框架：Pact

Step 1 - Define Consumer expectations



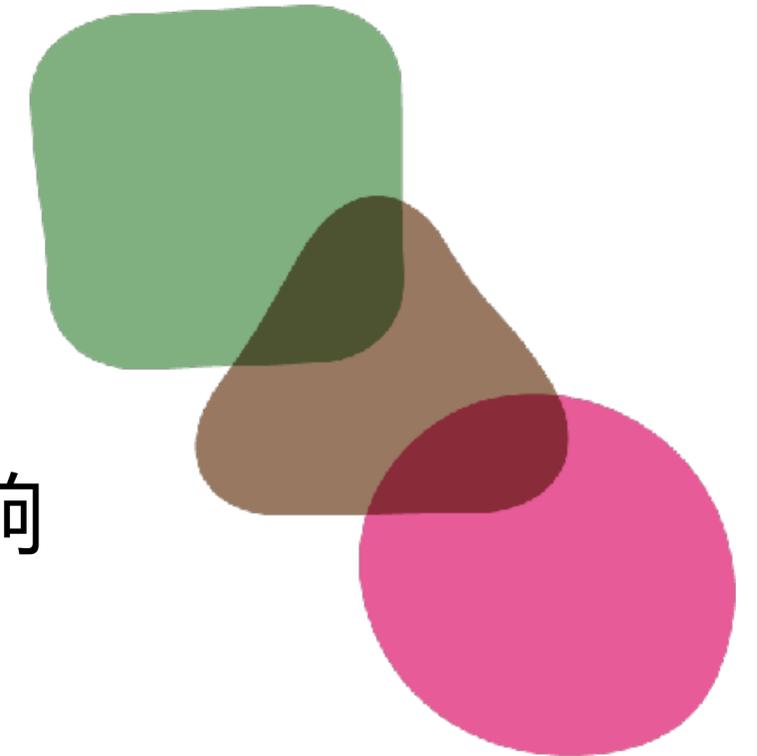
Step 2 - Verify expectations on Provider



Techniques

Decoupling deployment from release

- ❑ 部署：部署组件或基础设施到生产环境
- ❑ 发布：发布新加的/修改的feature给用户，有业务影响
- ❑ Feature toggle
- ❑ Dark Launch（+后台监控）
- ❑ 更频繁的部署，业务对要发布给用户的feature是可控的



Techniques

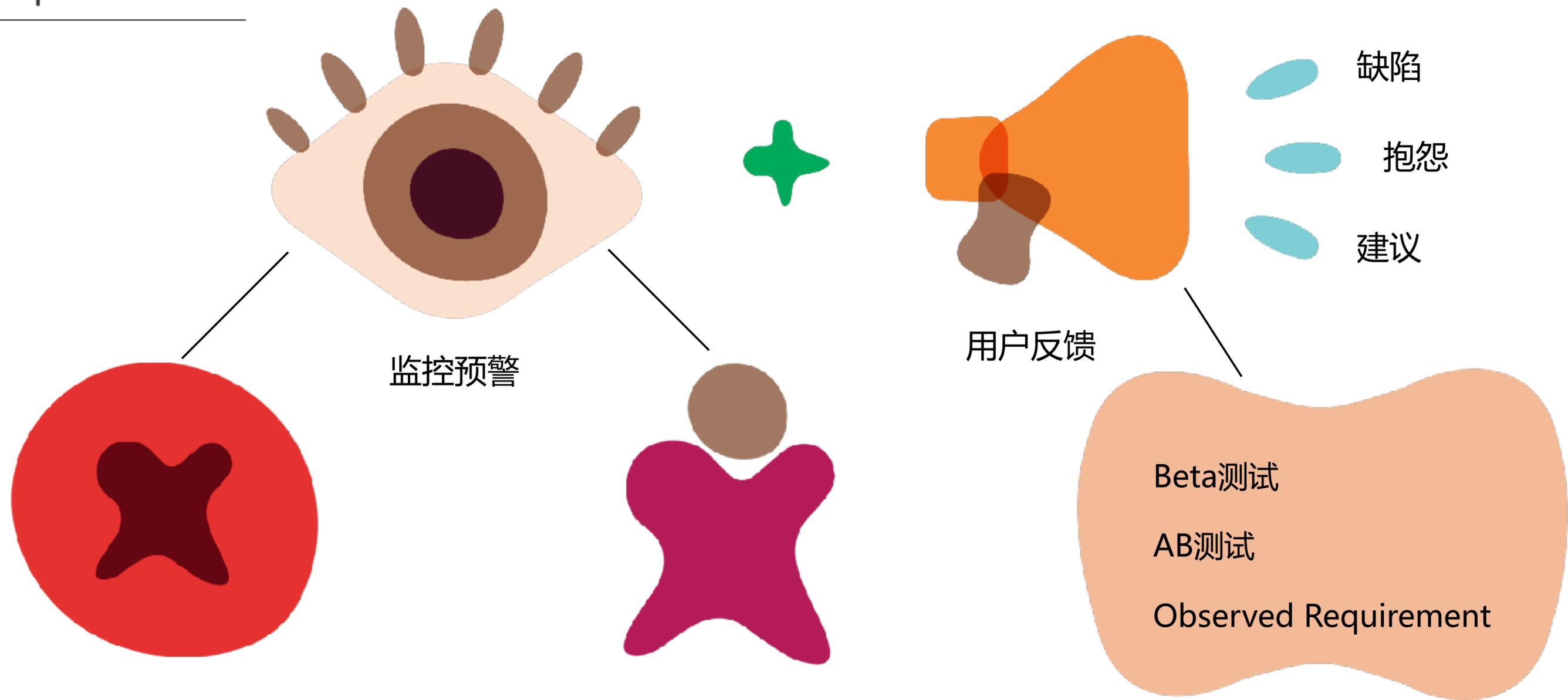
Focus on mean time to recovery

- 平均失败间隔
- 平均恢复时间
- 监控技术
- 持续交付的快速响应
- 减少预生产环境复杂的测试
- 对不断增加的安全攻击的响应非常重要



Techniques

QA in production

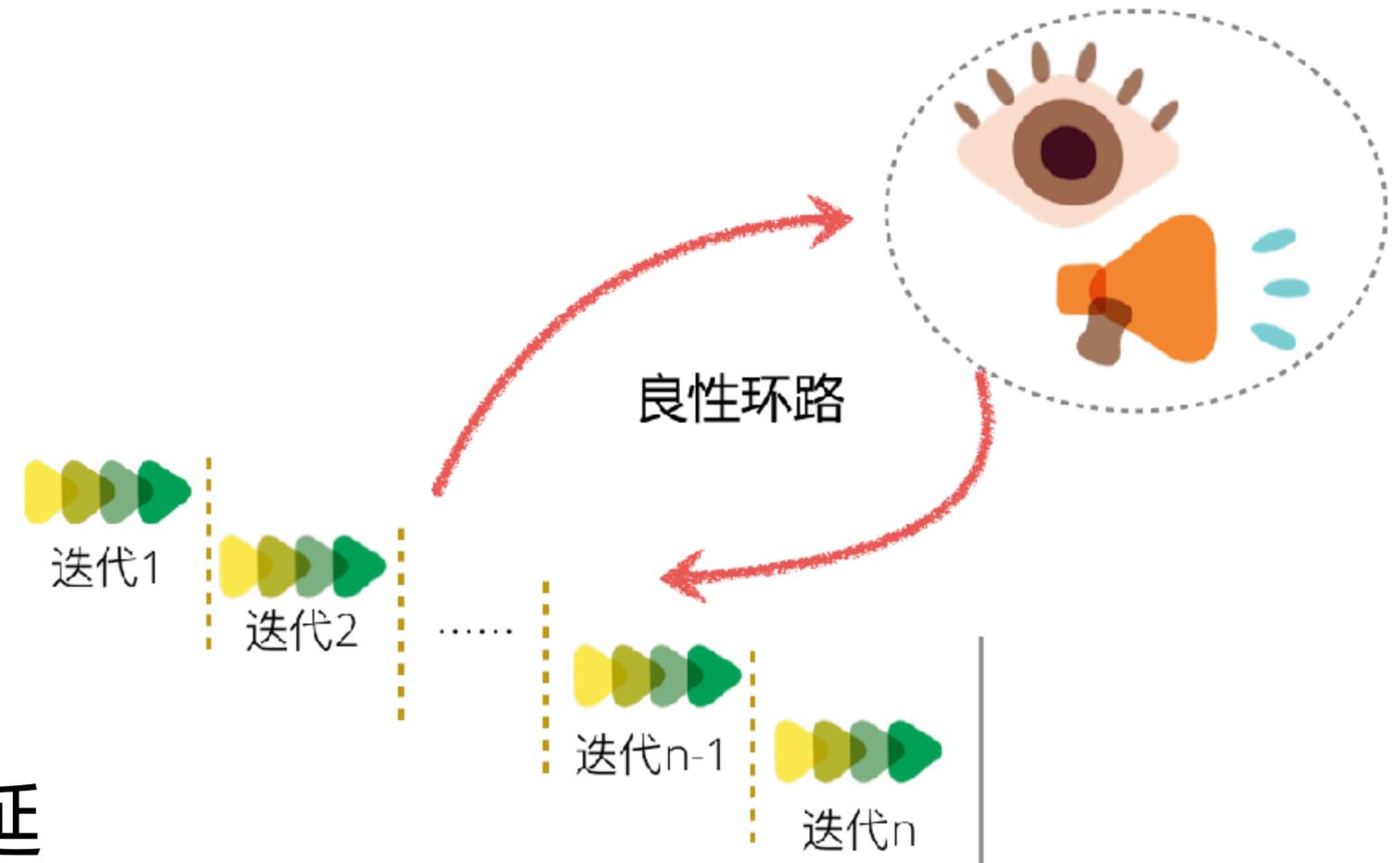


[QA in production](#)

[Observed Requirement](#)

Techniques

QA in production



- ❑ 不是预生产环境下的QA的简单后延
- ❑ 跟预生产环境下的QA相辅相成，形成良性环路才能做好
- ❑ 有别于Ops，QA在其中充当分析者和协调者的角色

- 用契约测试来保证API的提供端和消费端之间的契约关系
- 持续的部署，通过feature toggle的方式保证feature的完整发布
- 不要害怕生产环境的failure和error，要利用这些信息，做到快速恢复
- 通过生产环境下QA的各种方式来提高系统质量、优化业务价值

增加契约测试

加强日志监控

契约测试

发布与部署解耦

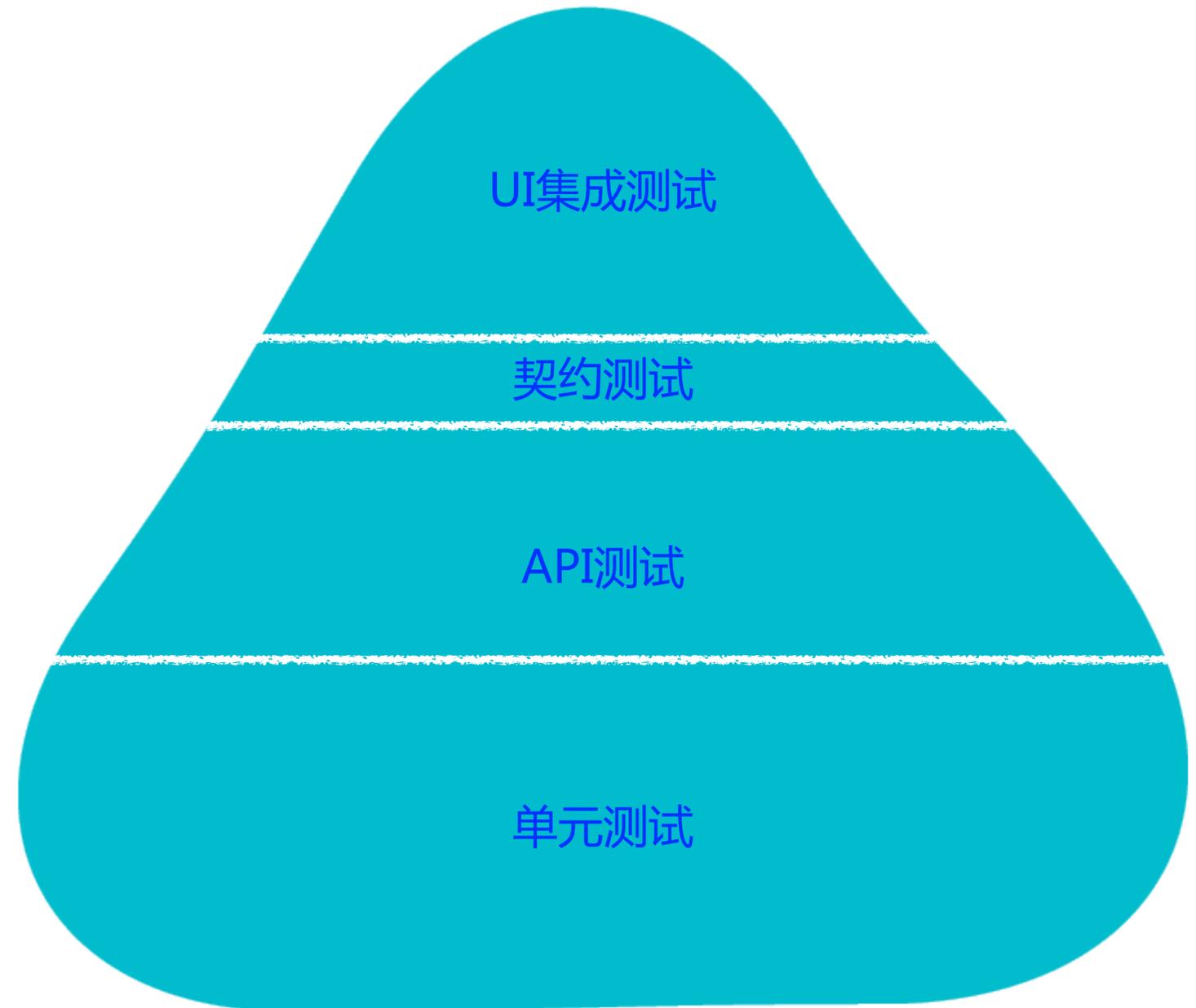
专注平均恢复时间

生产环境下的QA

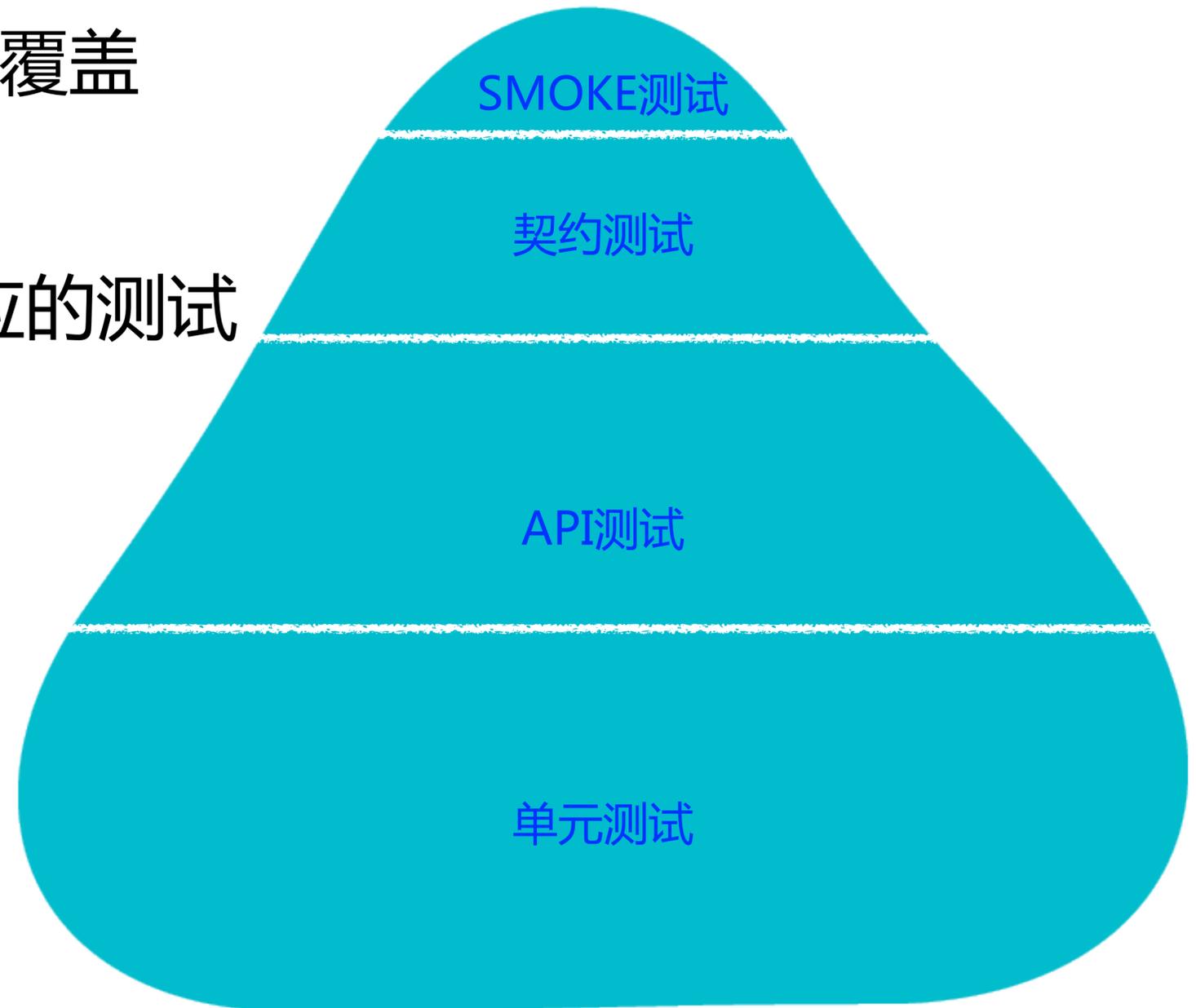
- 离岸交付
- 敏捷开发，4~5周一次发布
- 内部员工系统+外部用户系统
- 用户遍及全球
- 项目已经历7年
- 生产环境每天新增错误日志达几千条
- 集成测试成为持续交付瓶颈



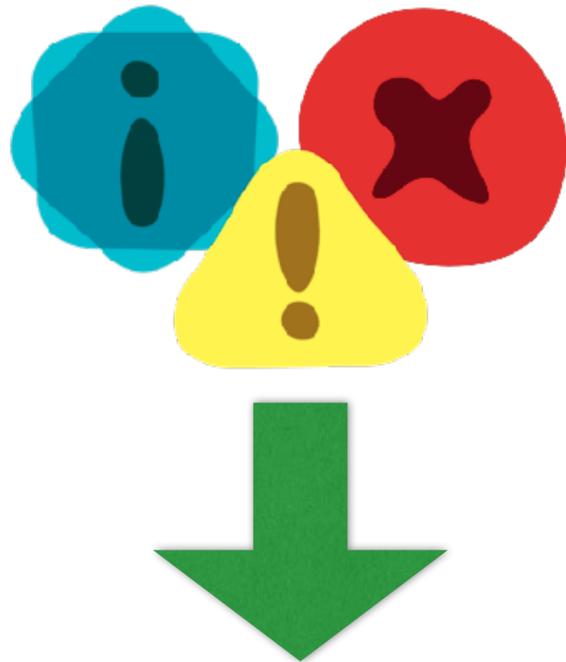
- 自动化测试组成如右图
- 测试覆盖不够优化，维护成本高
- 执行时间长，反馈慢
- 环境不稳定，频繁“随机挂”
- 很少发现真正的程序问题



- 从CI上摘掉原来的集成测试
- 测试下放，用单元测试和API测试来覆盖
- 增加契约测试
- 测试、生产环境发现问题，添加对应的测试



□ 工具：Splunk



搜索条件

搜索结果集

punct	count	percent
//_ N- _	8	57.142857
//_ N- _	2	14.285714
//_ N- _	2	14.285714
//_ N- _	2	14.285714

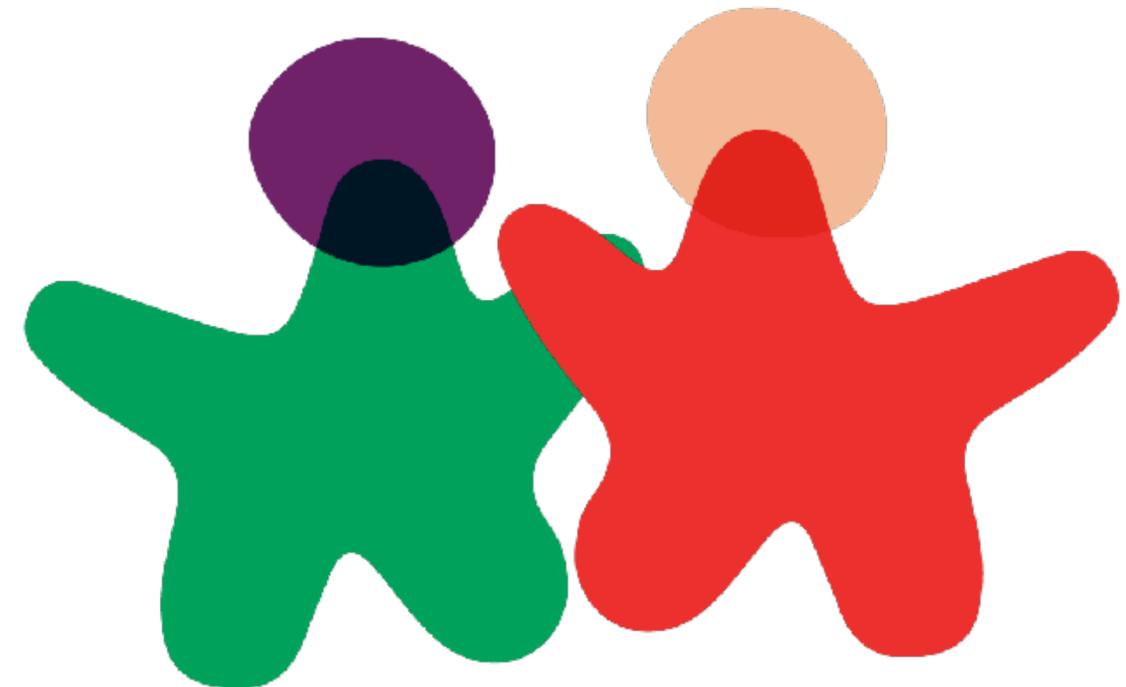
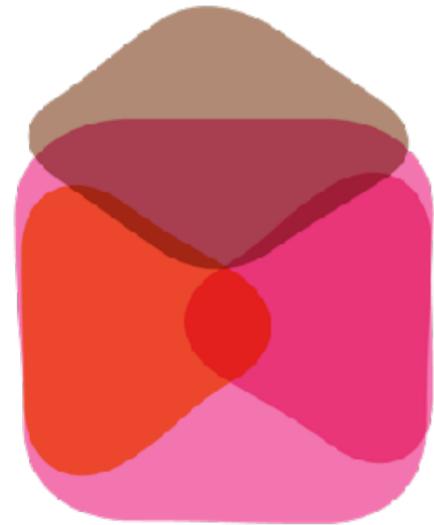
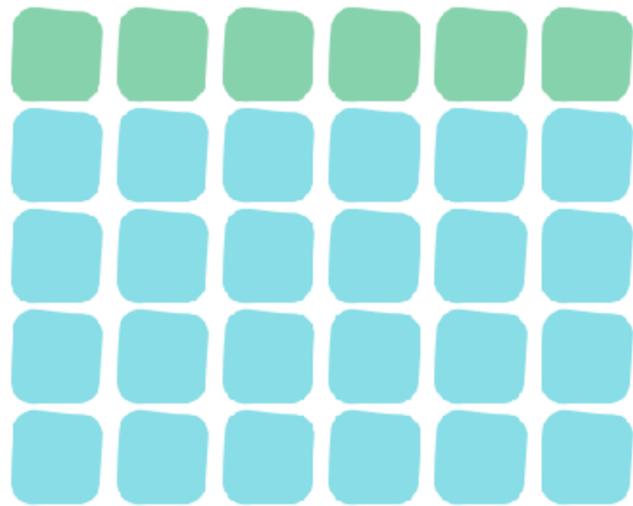
展开每条结果

详细错误信息

Time	Event
6/13/16 6:01:00.062 AM	2016/06/13 06:01:00.062 Error ... handler sync error: Cannot insert duplicate key row in object '...' with unique index '...' type'. The duplicate key value is (...). The statement has been terminated.
6/13/16 6:01:00.047 AM	2016/06/13 06:01:00.047 Error ... handler sync error: Cannot insert duplicate key row in object '...' with unique index '...' type'. The duplicate key value is (...). The statement has been terminated.
6/13/16 6:00:00.406 AM	2016/06/13 06:00:00.406 Error ... handler sync error: Cannot insert duplicate key row in object '...' with unique index '...' type'. The duplicate key value is (...). The statement has been terminated.

□ 数据采集机制

- 设置监控面板（ Dashboard ）：错误、性能等
- 设置预警邮件提醒
- 主动查找错误日志
- 同样的机制应用于测试环境，以提前发现问题



splunkalert	Splunk Alert: Perf Alert with High Cost APIs in
splunkalert	Splunk Alert: Perf Alert with Slow APIs in
splunkalert	Splunk Alert: QA - Perf Alert with Slow APIs in
splunkalert	Splunk Alert: QA - Perf Alert with High Cost APIs in

APIGroupUrl	cs_method	AvgTimePerRequest	NumberOfRequests	Cost
http://[redacted]	GET	53803.169426	2839	152747198.00
http://[redacted]	POST	427.595800	337311	144232767
http://[redacted]	GET	645.858341	164112	105993104
http://[redacted]	GET	390.020806	232246	90580772.1

- 发现系统功能问题，提供对应解决方案，优化业务
- 发现安全、性能等非功能问题，修复或优化
- 发现日志记录的不足，规范日志记录
 - 统一日志输出路径
 - 统一日志记录格式
 - 清晰定义日志级别
 - QA参与日志评审



日志格式实例

```
{ [-]  
  DataSize: 1  
  Method: POST  
  MilliSeconds: 2402  
  Order: 12  
  ReqId: [REDACTED]  
  StatusCode: 200  
  Time: 2017-06-02T08:34:18.16883117  
  Transaction: [REDACTED]  
  Type: performance  
  Url: https://[REDACTED]  
  UserId: [REDACTED]  
}
```

- 断：不再添加集成测试，采用Smoke测试来检查系统关键功能
- 舍：原有的集成测试全部废弃掉，简化CI Pipeline，加速出包
- 离：通过增加契约测试、日志监控等其他方式来做好产品的质量保证

增加契约测试

加强日志监控

契约测试

发布与部署解耦

专注平均恢复时间

生产环境下的QA

- 集成测试是非常重要的，但往往是持续交付的瓶颈
- 不主张构建复杂的集成测试环境，而是采用新的技术如契约测试、发布与部署解耦、关注平均恢复时间和生产环境下的QA
- 集成测试的断舍离需要团队有持续、递进、稳定的交付能力，用户不会受此影响
- 集成测试及其环境往往是费了很多心血，要做到断舍离不是一件易事，需要平衡和取舍



SUMMARY

谢谢大家

2017.06.24

ThoughtWorks®