

快速迭代下的质量交付闭环

京东物流-运营研发部-质量部 王晓琦

下一代
软件研发
SOFTWARE
DEVELOPMENT

目录

CONTENTS

1

困难与挑战

2

我们的实践

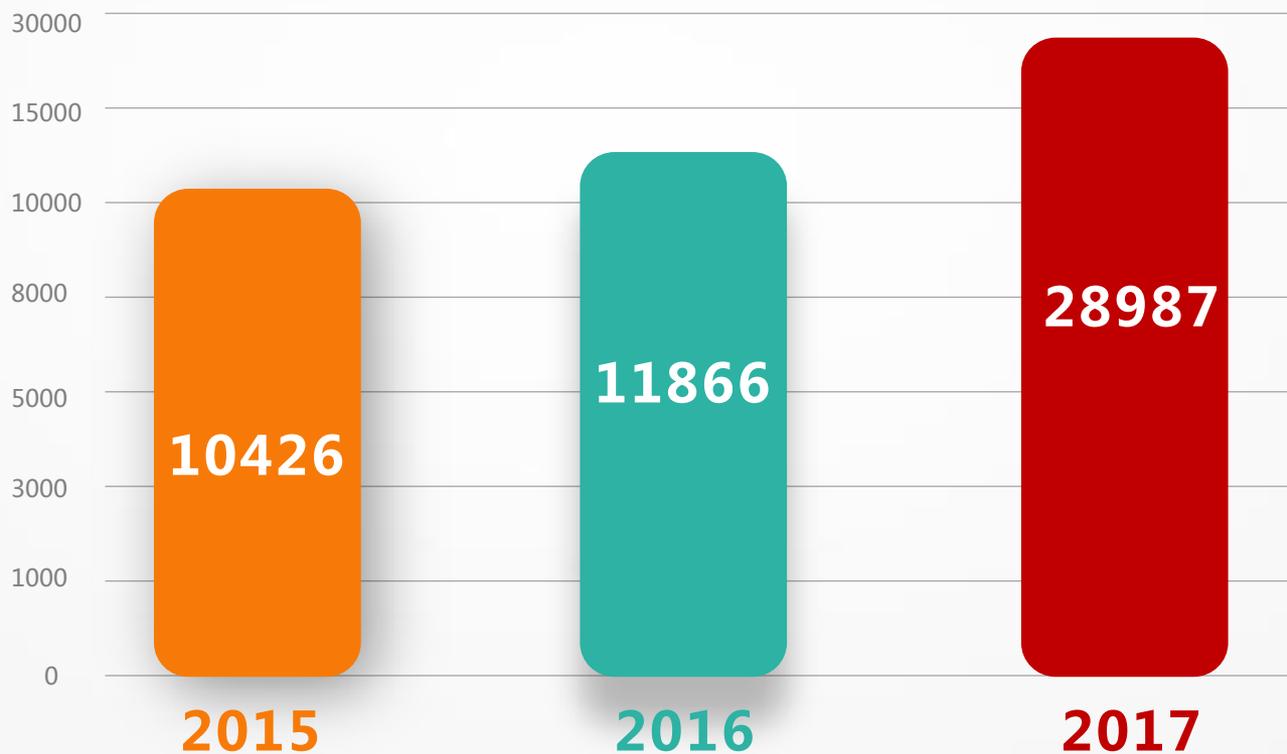
3

实践成果

4

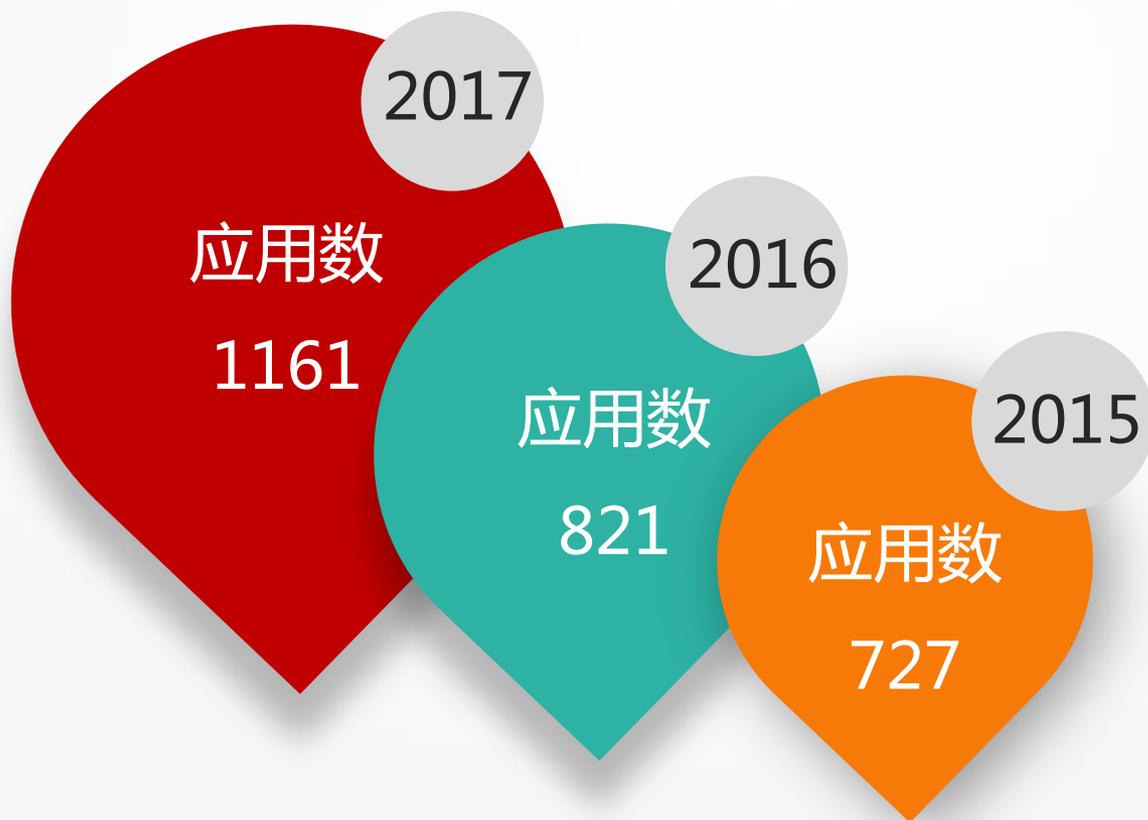
后续计划

➤ 业务驱动、迭代快



变更频繁、迭代快 如何保证质量？

➤ 业务系统环境复杂



环境部署及维护成本高

➤ 系统访问量

57亿次

结算页总调用量

1.3亿/分

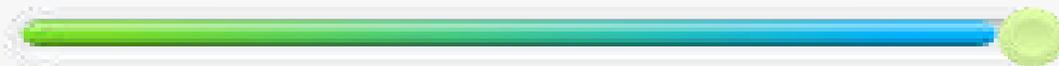
结算页峰值调用量

1400w/分

单个接口峰值

如何快速评测系统性能？

面对困难与挑战 Tester如何厚积薄发，成功逆袭！



目录

CONTENTS

1

困难与挑战

2

我们的实践

3

实践成果

4

后续计划

效率提升

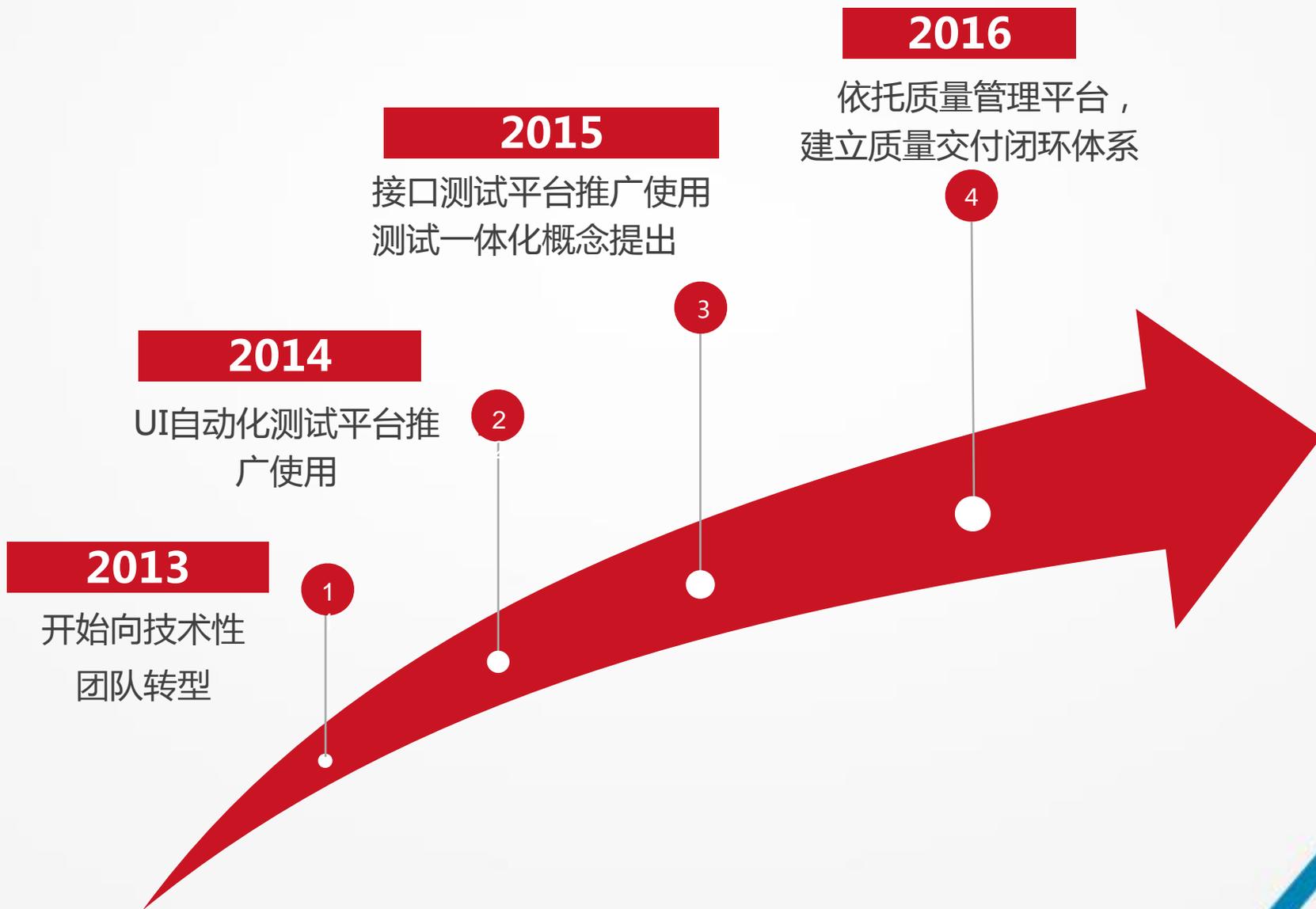
- 测试环境弹性化
- 应用依赖解耦
- 推行众测

质量交付闭环

- 规范提测流程
- 流水线作业

能力提升

- 测试转型
- 测试培训交流



效率提升

- 测试环境弹性化
- 应用依赖解耦
- 推行众测



测试环境弹性化

- 环境一致性
- 自动部署
- 环境可回溯

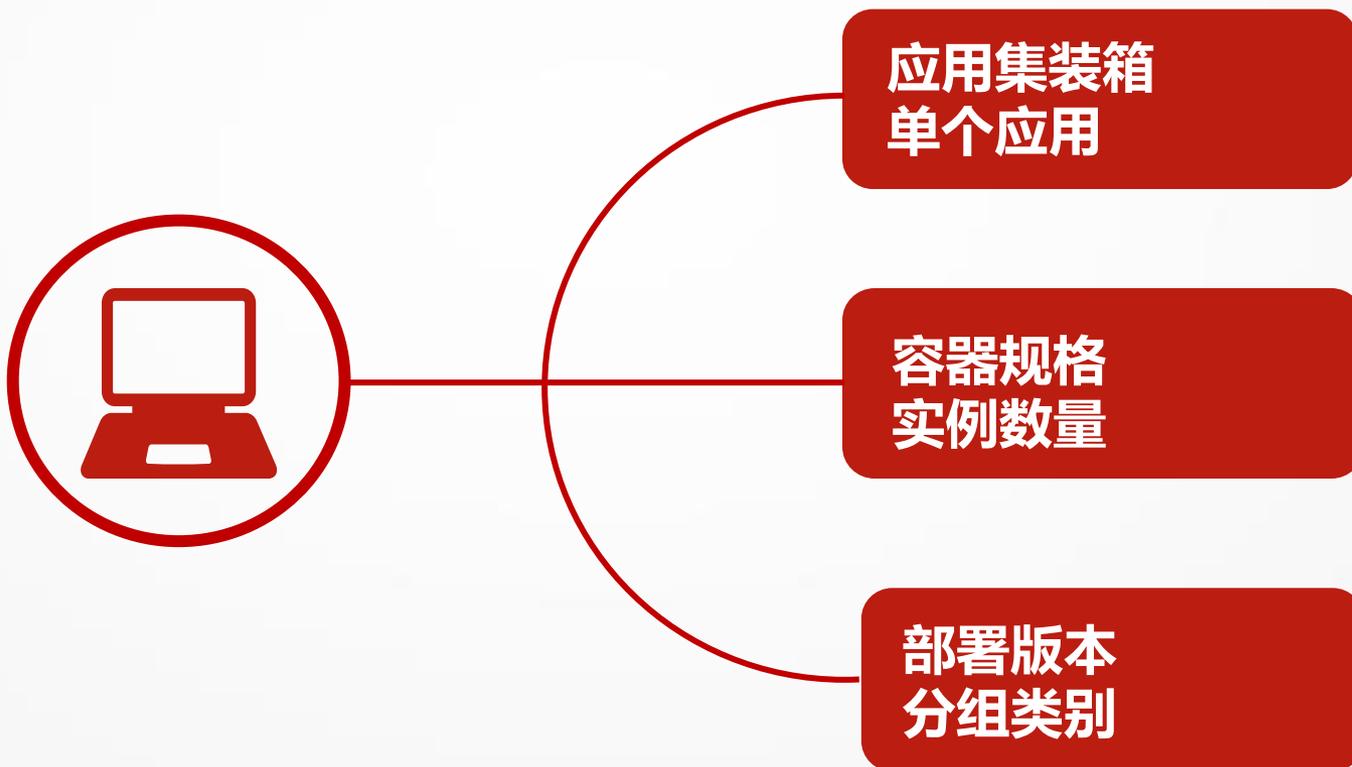
环境弹性化—环境一致性



- 运行环境**
 - Tomcat
 - JDK
 - 操作系统版本
- 应用版本**
 - 生产版本
- 基础服务**
 - JSF
 - JMQ
 - JimDB
- 数据库**
 - 数据库版本
 - 表结构
 - 索引及配置

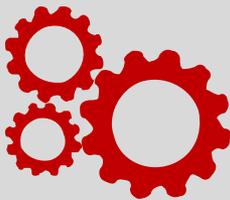
- 多应用一键部署
- 部署多套测试环境
- 配置文件统一管理
- 服务器资源弹性化

基于镜像的环境回溯

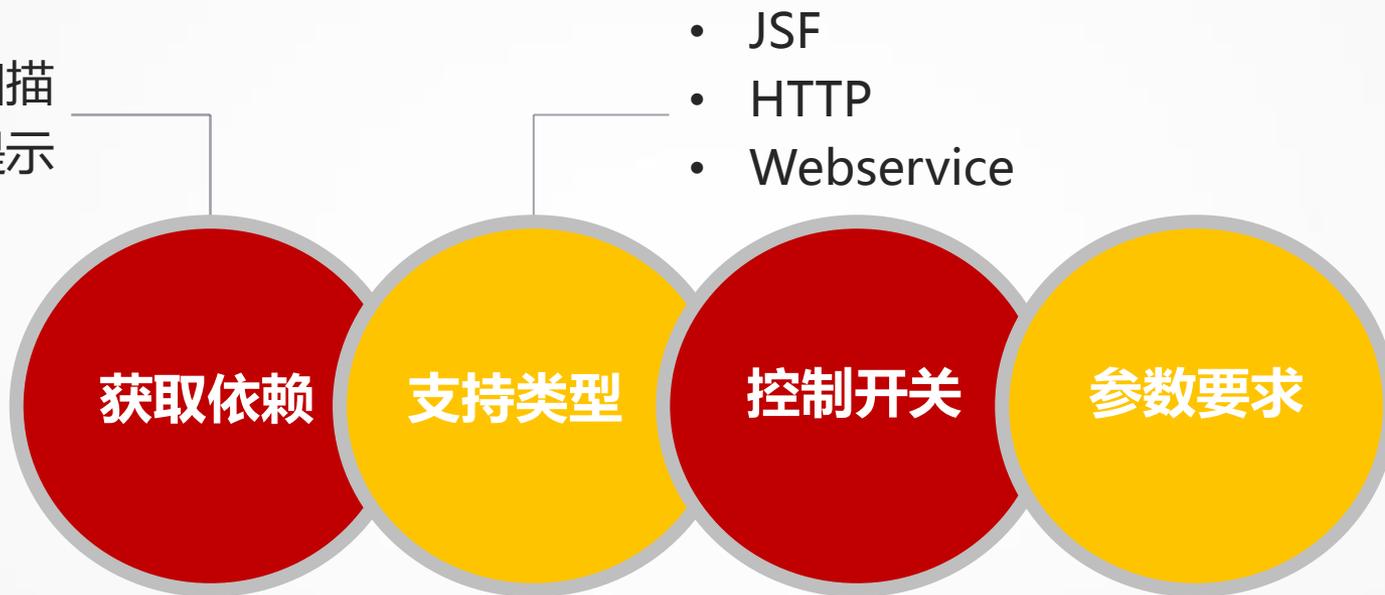


效率提升

- 测试环境弹性化
- **应用依赖解耦**
- 推行众测



- 依赖扫描
- 更新提示



- JSF
- HTTP
- Webservice

- Mock开关
- 超时开关

- 返回值参数化
- 超时设置

特点：

- 无需编码
- 无需部署
- 即刻生效



效率提升

- 测试环境弹性化
- 应用依赖解耦
- **推行众测**



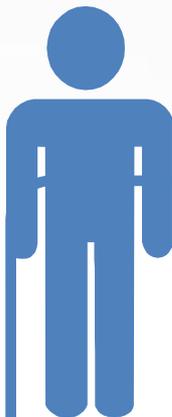
测试预备役

- 日常压测
- 安全清查
- 大促备战

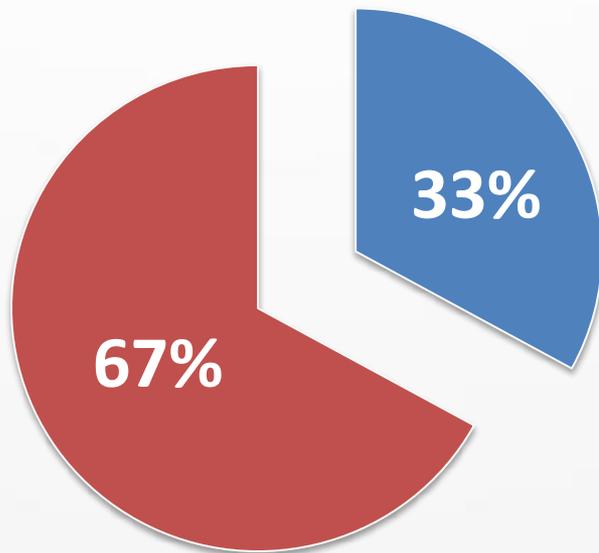


研发自测

- 线上压测
- 单元测试
- 接口回归



平台使用情况



质量交付闭环

- 规范提测流程
- 流水线作业



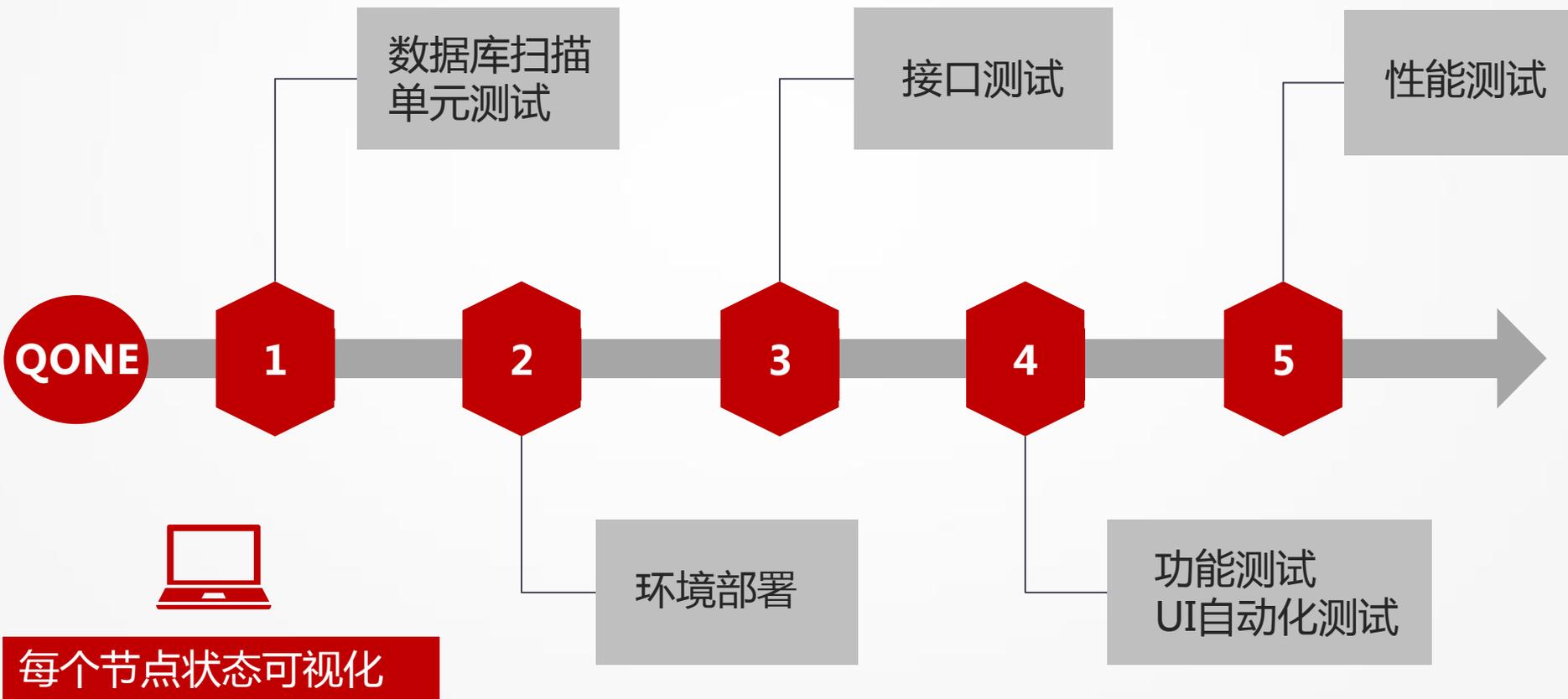
提测流程

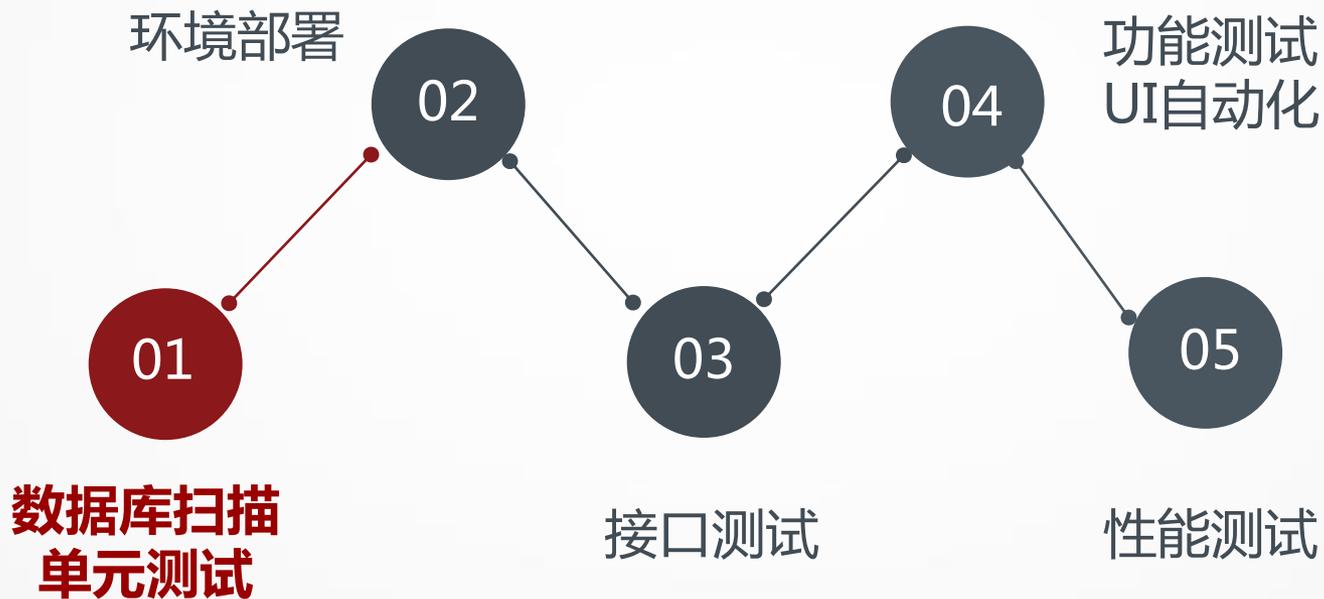


质量交付闭环

- 规范提测流程
- **流水线作业**







- 规范扫描

命名规范、数据类型规范、创建索引规范、逻辑删除

- SQL扫描

字段一致性、模糊匹配、字段隐式转换、多个子查询等

- 线上执行DDL加入测试审批环节

危险系数		规则	
严重	646	严重	字段命名规范(update_time字段除外)
中等	175	严重	所有表必须要有独立于业务的自增主键，统一命名为id，非空，类型为bigint
一般	0	严重	禁止在数据库中保存图片、xml等大资源
全部	821	严重	索引名必须全部小写，且以'idx_'开始
		严重	ts字段检查
		严重	is_delete字段检查

表名	危险	描述	执行人	更新时间
app_application	严重	表 [app_application] 的字段 [interfaceName] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [methodName] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [applicationName] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [createTime] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [updateTime] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [appName] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [appChName] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [appDomain] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [appOwner] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [sysId] 未定义为not null	测试	2016-05-03 11:22:34
app_application	严重	表 [app_application] 的字段 [appLevel] 未定义为not null	测试	2016-05-03 11:22:34



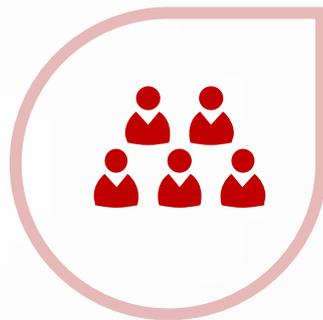
核心系统

0级系统核心模块



用例设计

测试与研发一同确定
测试范围



执行

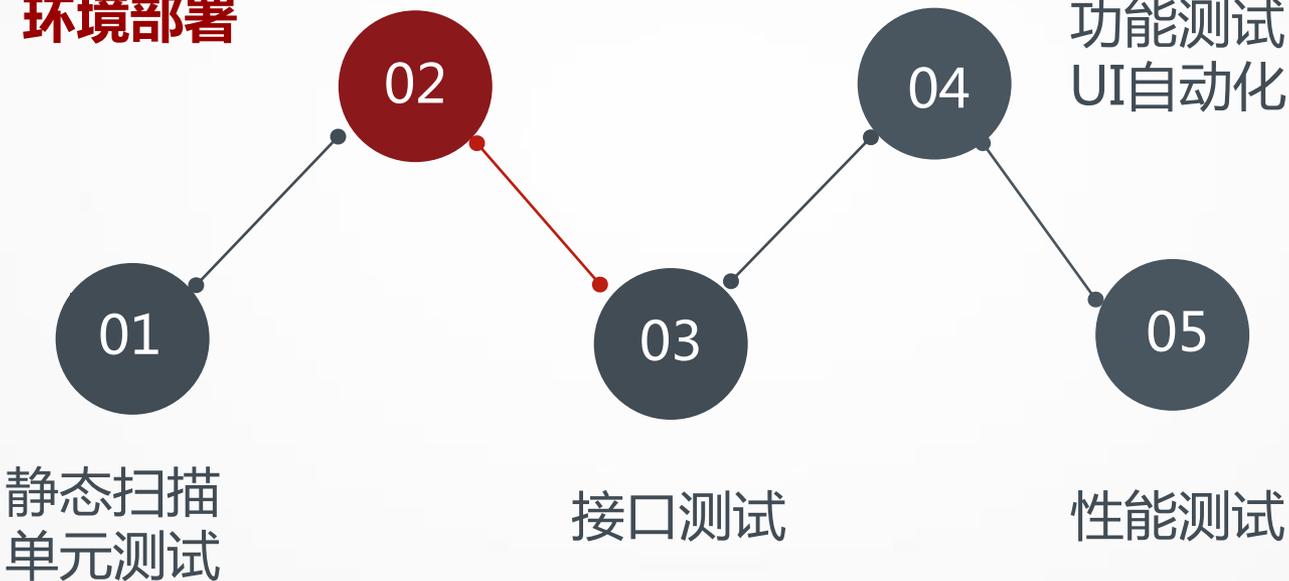
提测时触发执行



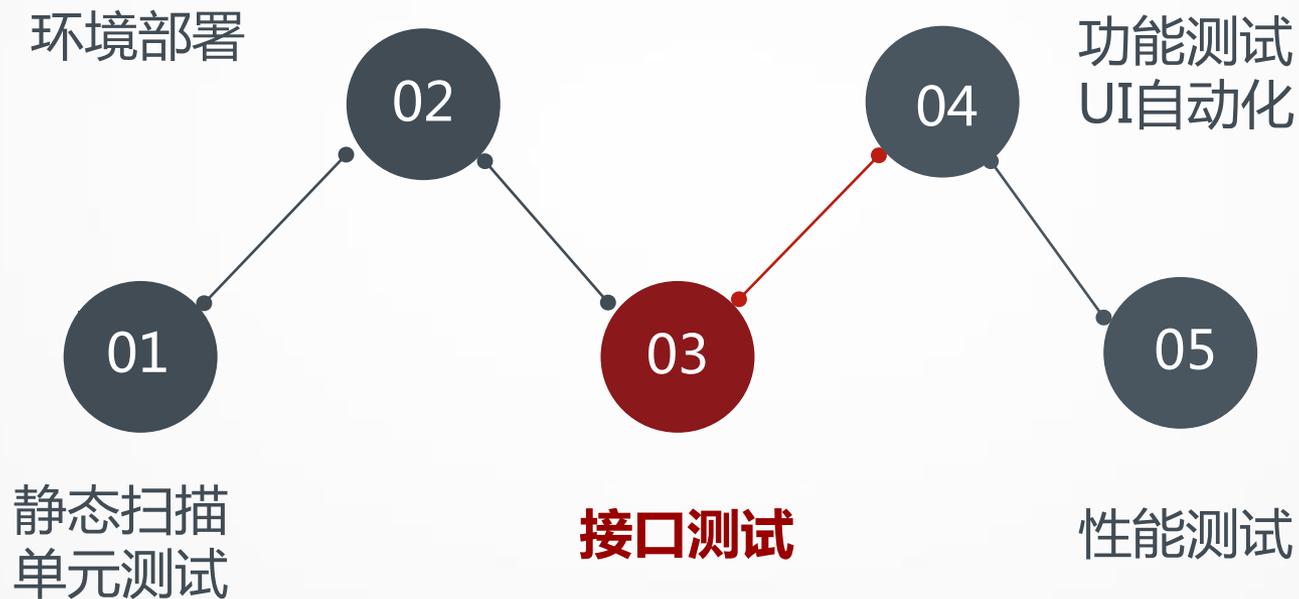
持续性

采取提测任务迭代方式

环境部署







JD.COM EasyTest 接口测试平台

[主页](#) [接口](#) [工具箱](#) [项目](#) [JSON格式化](#)

项目接口列表
测试接口类型: 接口名称:
项目ID:159 模块ID:119

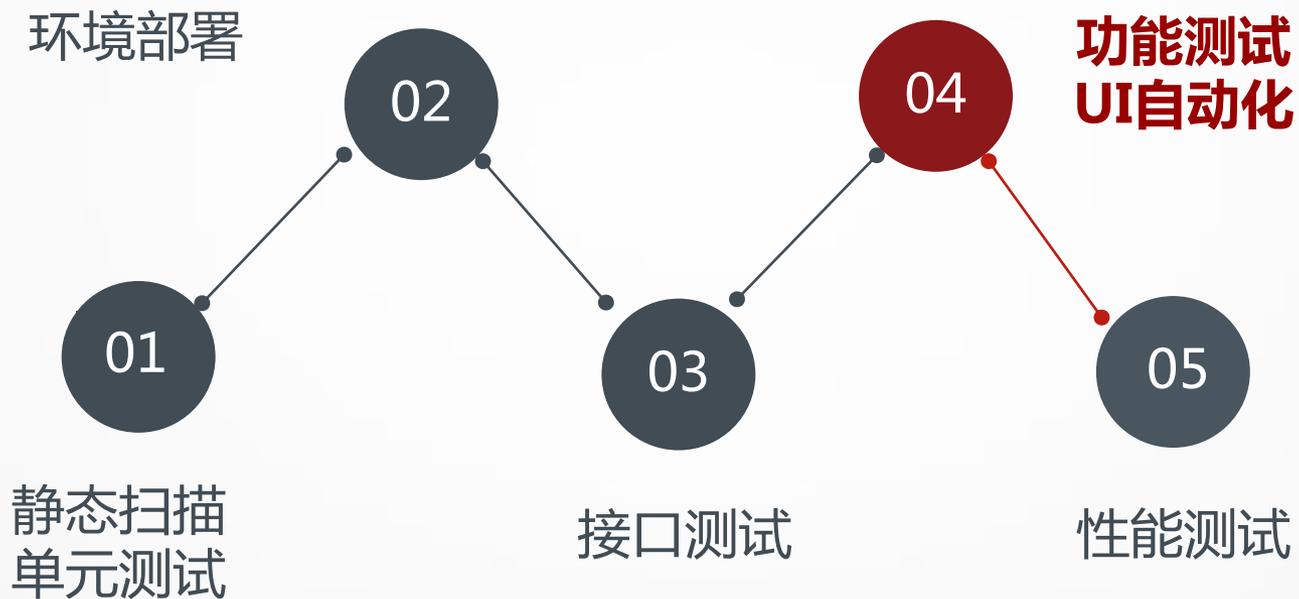
运行测试
查询
GIT路径扫描添加JSF接口

+ 新增
 ✎ 修改
 ✖ 删除
 ↶ 数据串前置
 ↷ 数据串后置
 🛡 维护测试数据

接口id	接口中文名称	接口名称	方法名	别名	编码格式	发布者IP	发布者端口	接口开发者
1	JSF接口	com.alibaba.fastjson.parser.AFastJsonParserImpl		AFSTEST-POP-1	UTF-8	192.168.158.79	9091	
2	单据处理数据	com.alibaba.fastjson.parser.AFastJsonParserImpl		AFSTEST-0.0.2	UTF-8	192.168.158.78	9090	
3	数据不一致			atomTest	UTF-8	192.168.144.98	22000	
4	wabf			test:0.0.1	UTF-8	192.168.195.145	22002	
5	顺序后的BUG			test:0.0.1	UTF-8	192.168.194.137	22001	
6	排序		queryComponent	test:0.0.1	UTF-8	192.168.194.137	22001	
7	无参数的			VTS-TEST	UTF-8	192.168.159.56	22000	
8	a	com.alibaba.fastjson.parser.AFastJsonParserImpl	readSMSCode	account-bt-test	UTF-8	192.168.146.64	22001	
9	wangxiabo	com.alibaba.fastjson.parser.AFastJsonParserImpl	getRealTimeStockCode	stock-t	UTF-8	172.24.7.75	22001	
10	异常的			O2O-VSAPI-PLATFORM:1.0.0	UTF-8	192.168.194.109	22002	
11	接口测试			eptd.member.test	UTF-8	192.168.195.147	22001	
12	张军的接口			userSdk	UTF-8	192.168.153.135	22000	

执行历史

- #2016-12-26 16:41:12
- #2016-12-23 14:27:17
- #2016-12-21 11:46:25
- #2016-12-21 11:46:13
- #2016-12-12 11:06:18
- #2016-12-12 11:05:25
- #2016-12-08 17:47:53
- #2016-12-02 14:40:54
- #2016-12-02 14:40:36
- #2016-12-02 14:40:20
- #2016-12-02 14:40:15
- #2016-12-02 14:39:59
- #2016-12-02 11:26:59
- #2016-11-30 09:31:47
- #2016-10-26 14:44:33



01

测试任务

提测需求与提测任务、
提测包相关联

02

测试用例

按提测任务，组装测
试用例

03

变更代码

根据Git地址获取两个
版本变更的代码

05

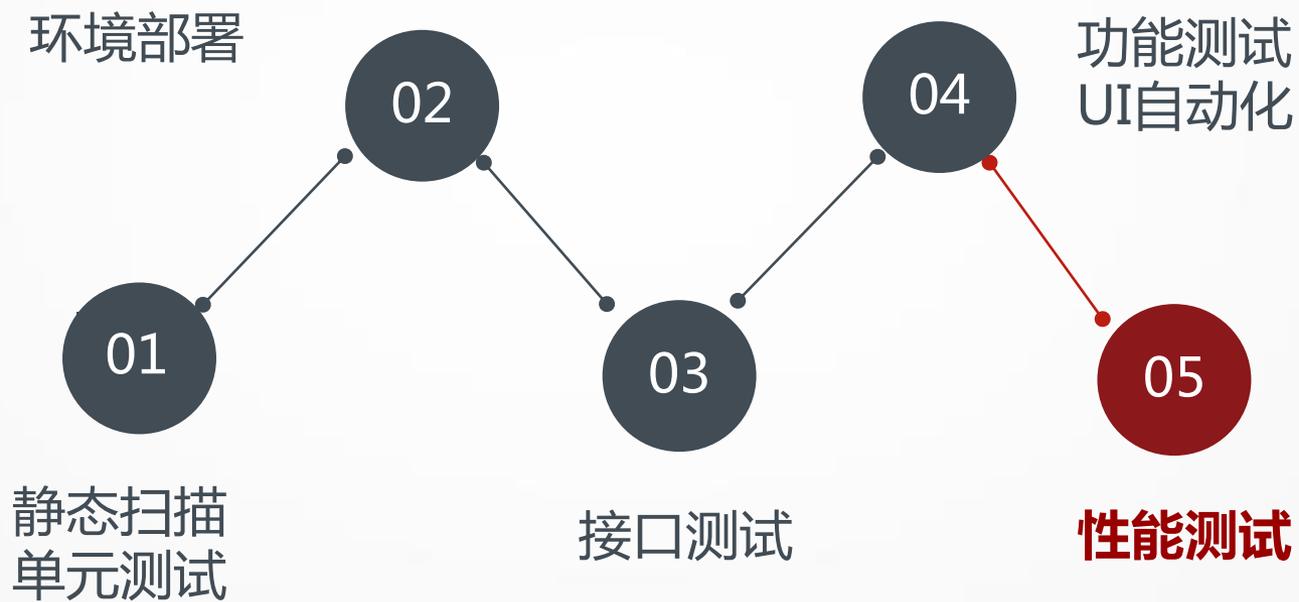
覆盖率

根据监控执行路径与变更的
代码匹配，获取执行覆盖率

04

监控

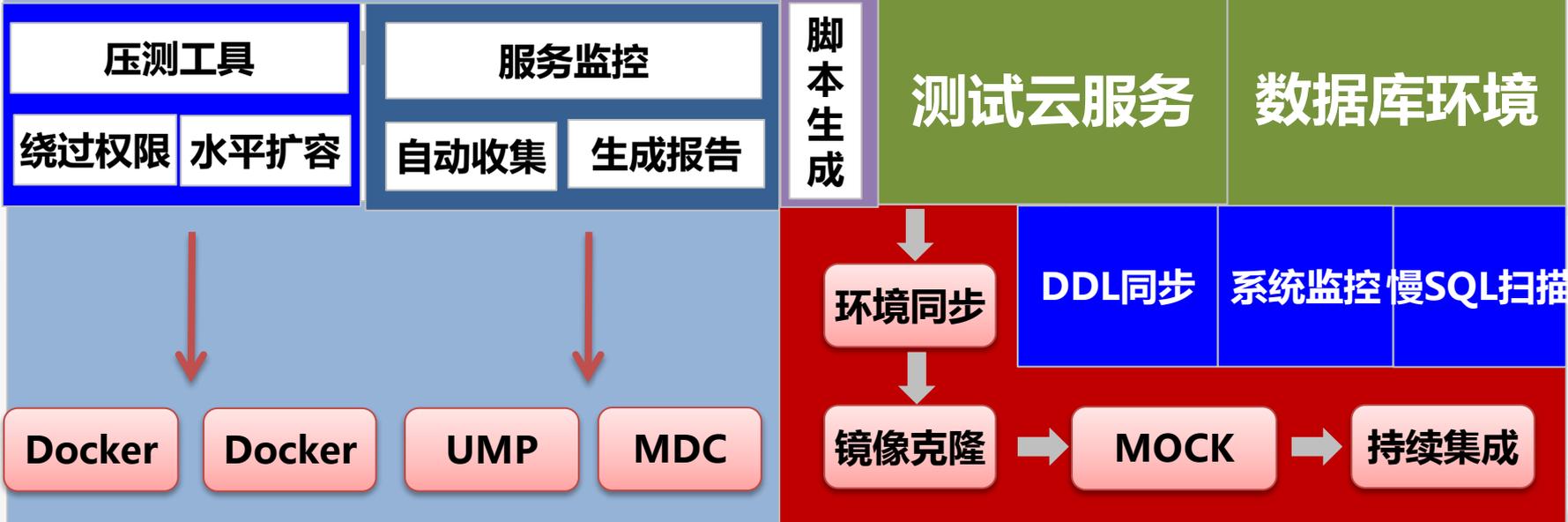
执行测试用例时，
监控应用执行路径



QONE质量管理平台

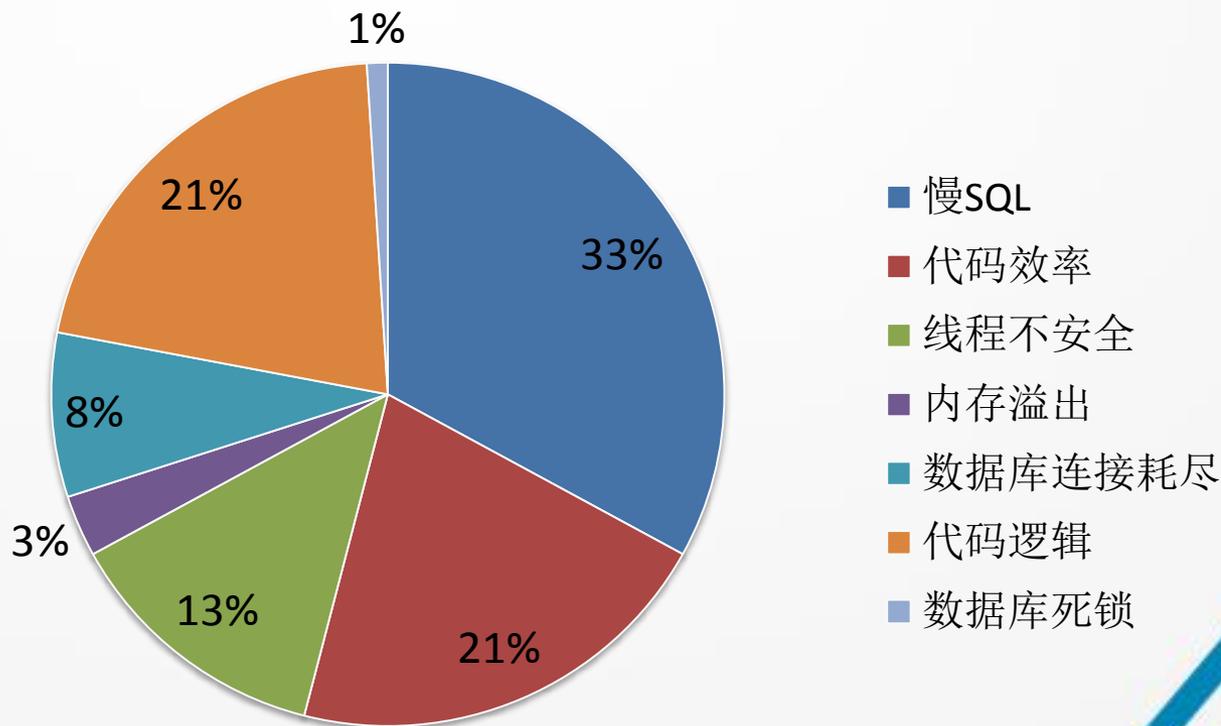
线上压测

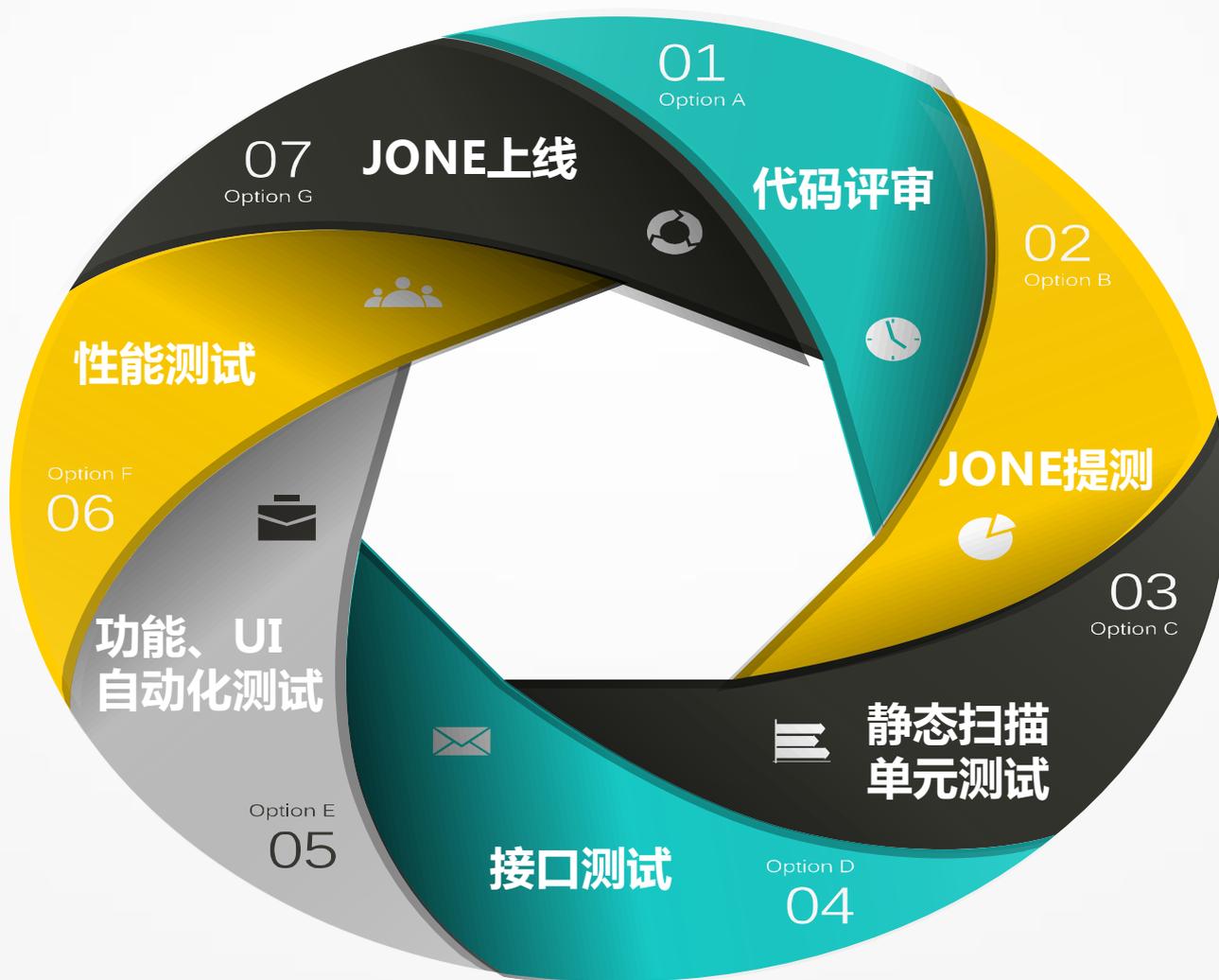
线下压测



- 核心业务线覆盖率达到 **90%**
- 性能测试常态化（**0级系统定期回归**）

缺陷类别





能力提升

- 测试转型
- 测试培训交流



- 测试到测试开发转型
- 测试技术培训
- 组织商城测试团队业务、技术分享

目录

CONTENTS

1

困难与挑战

2

我们的实践

3

实践成果

4

后续计划

执行任务数

7322

部署成功次数

22042

接入应用数

719



环境部署

- 一致性、可回溯
- 应用依赖解耦



线上压测

- 自动生成脚本
- 压力机水平扩容

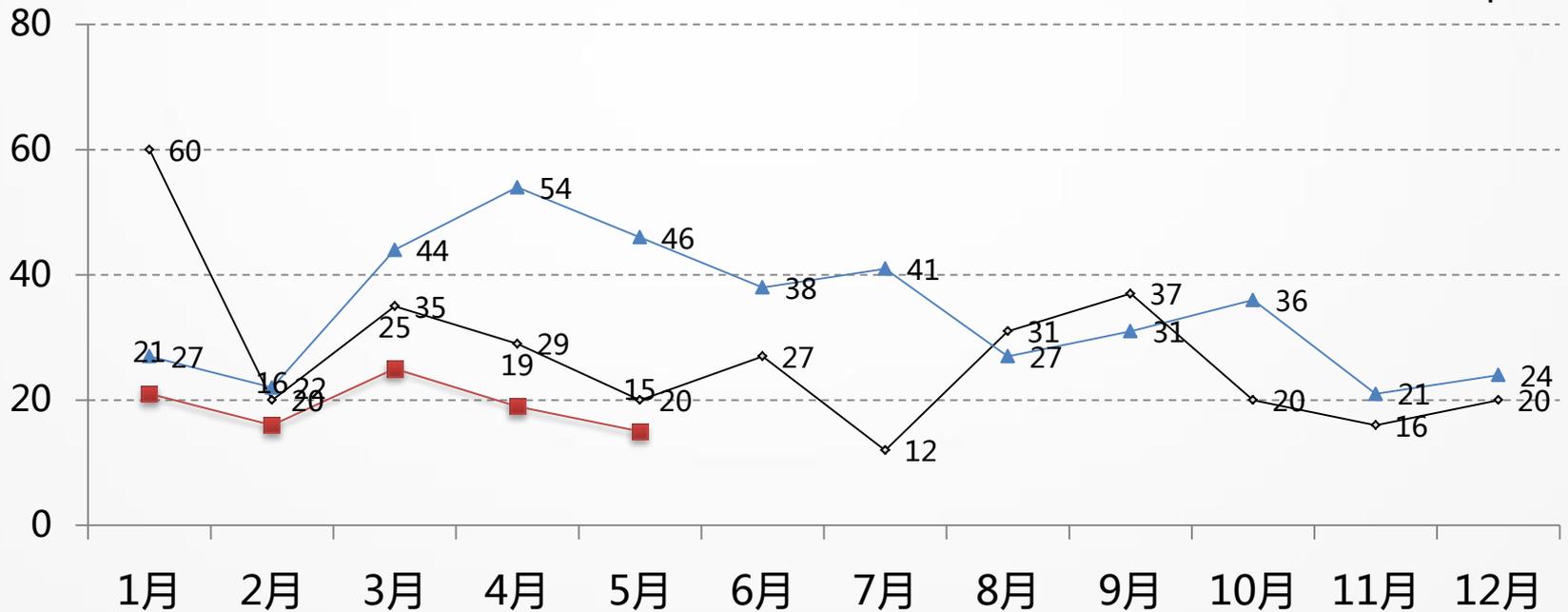


性能服务器成本

- 使用Docker
- 资源弹性化

COO-线上问题趋势

◇ 2015年
▲ 2016年
■ 2017年



目录

CONTENTS

1

困难与挑战

2

我们的实践

3

实践成果

4

后续计划

- 基于代码变更的回归测试
- 运营系统全链路压测
- 代码静态扫描
- 测试项目结果监控&分析
-

Q&A



谢谢！

