

2019

代码度量

测试赋能开发

酷家乐·石塘

CONTENTS



PART 01
背景



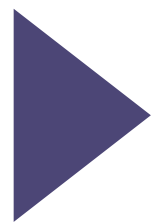
PART 02
架构



PART 03
落地



PART 04
愿景



PART 01

背景

酷家乐背景

用户

酷家乐服务家居企业数量超过14000家，总用户量突破2000万，设计师用户超过500万。

产品

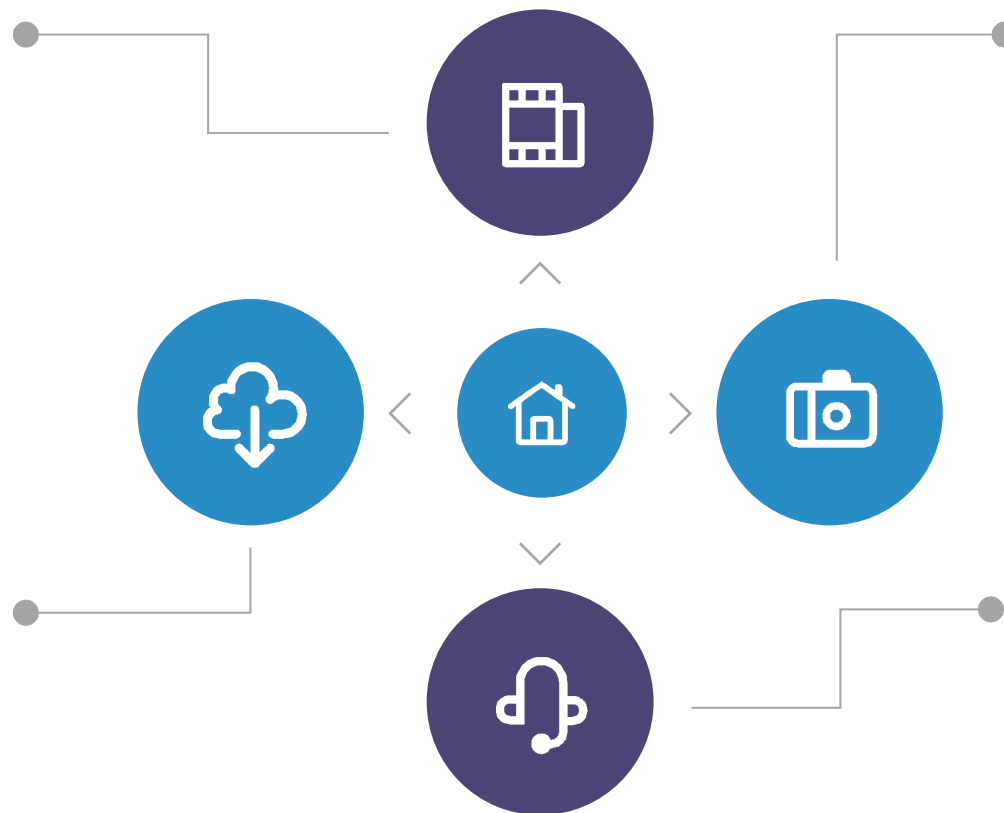
- 全屋硬装2.0上线
- 全屋定制对接生产
- BIM水电和施工图上线
- 国际品牌COOHOM
- 移动端国际化版本上线
- 智能云设计工具4.0
- 大运河计划

工具运营

平台日均渲染图量超150万张，全年用户总渲染效果图达2.21亿张，全年全景图浏览量近6亿次，全年浏览时长超过50亿分钟，产出家居设计方案突破6000万个

人员

600+ 到 1200 +



酷家乐背景

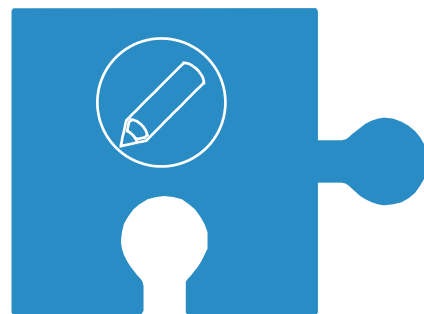
业务快速增长，人员急剧扩张，产品故障增多，代码质量下降



编码风格不同



CodeReview量大



单测覆盖下降



日志不规范

平台建设

如何解决当前痛点 → 代码度量平台建设



代码优化依据

- 代码规范
- 重构依据



衡量研发工作质量

- 可靠性
- 安全性
- 可维护性
- 覆盖率
- 复杂度



自动化

- 分析自动化
- 统计自动化
- 通知自动化



质量闭环

- 接入CICD，质量卡点



规范化

- 编码规范
- 组件引入规范
- 自定义规范



流程化

- CICD流程绑定
- 代码分析前置



代码度量平台

LITMUS

Dashboard

Analytics

Kanban

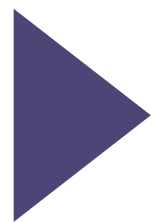
所有 收藏 请输入查询内容 业务组 敏感组 标签

新建Job

Job名称	业务/敏感组	分支	标签	最近执行时间	构建状态	质量分	操作
customservice-release-20190724-t-release		release/20190724-t-release	Java后端	2019-08-12 17:38:23	SUCCESS	40.00	
modeeditor-release-20190816-t-release		release/20190816-t-release	Java后端	2019-08-20 11:00:16	SUCCESS	45.00	
kms-moon-stable_stable_07_26_14_23_08		moon/stable_stable_07_26_14_23_08	Java后端	2019-07-29 11:09:36	FAILURE	0 --35.00	
drawings-frame-release-20190725		release/20190725	Java后端	2019-07-30 11:41:50	SUCCESS	80.00	
openapi-release-190726-beta1		release/190726-beta1	Java后端	2019-07-26 17:16:38	SUCCESS	40.00	
ue-service-release-20190726		release/20190726	Java后端	2019-07-25 15:59:04	SUCCESS	80.00	
customruleservice-master		master	Java后端	2019-08-20 20:19:03	SUCCESS	80.00 --5.00	
message-center-master		master	Java后端	2019-08-20 05:00:17	SUCCESS	80.00	
saas-account-master		master	Java后端	2019-08-20 17:59:34	SUCCESS	80.00	
moon-ui-chore-#9		chore/#9	Java后端			--	

< 1 2 3 4 5 6 ... 92 >

Litmus Design ©2019 酷家乐 测试架构组



PART 02

Litmus 设计

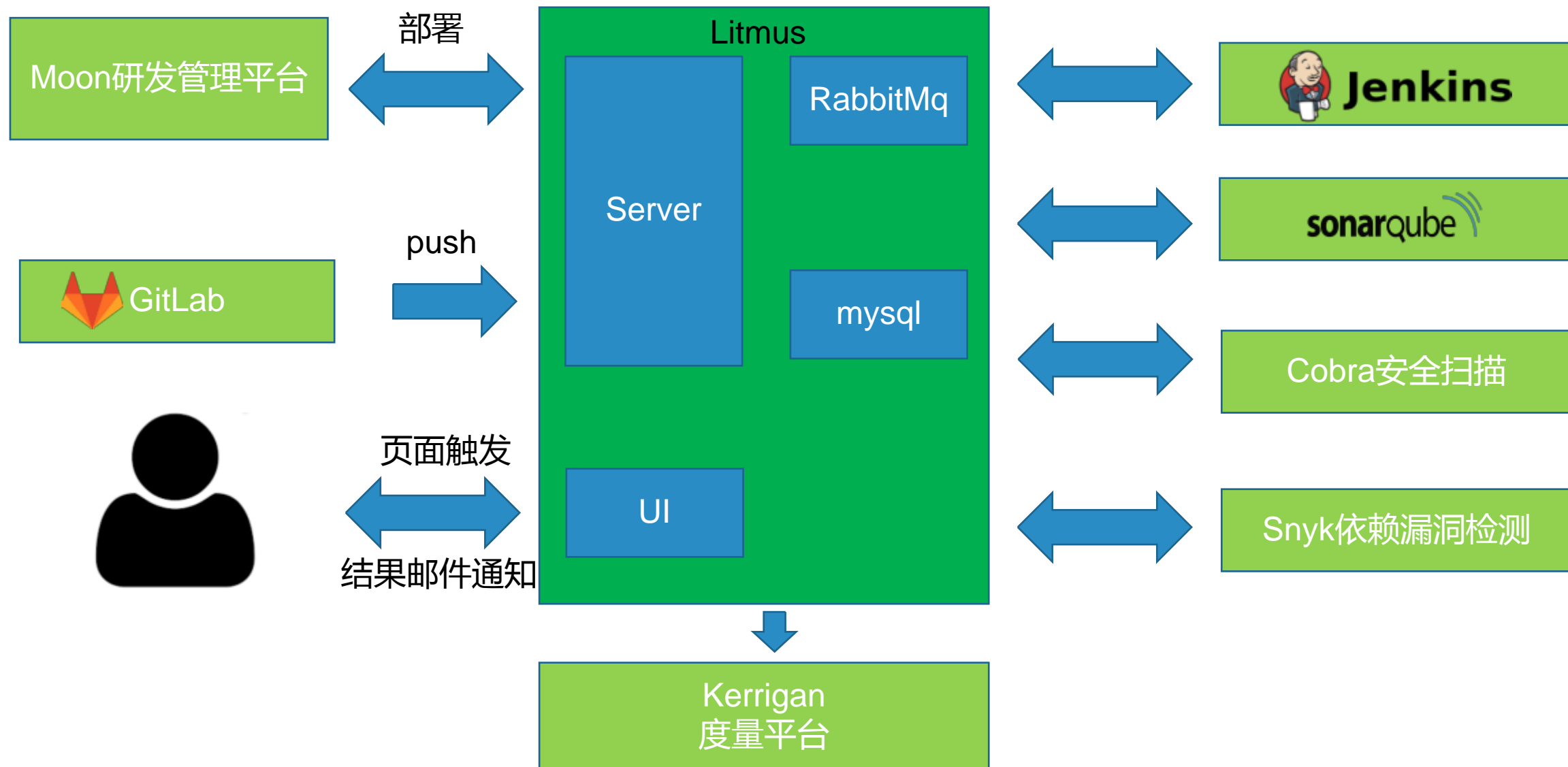
保证代码的高质量，高可用性，高可读性



平台设计

- 代码质量评估: SQALE (Software Quality Assessment based on Lifecycle Expectations)
- 编码规范: 阿里巴巴Java开发手册
- 代码扫描: SonarQube
- 代码安全审计: Cobra
- 三方包漏洞扫描: Synk

平台架构



统一度量

如何统一度量代码质量？

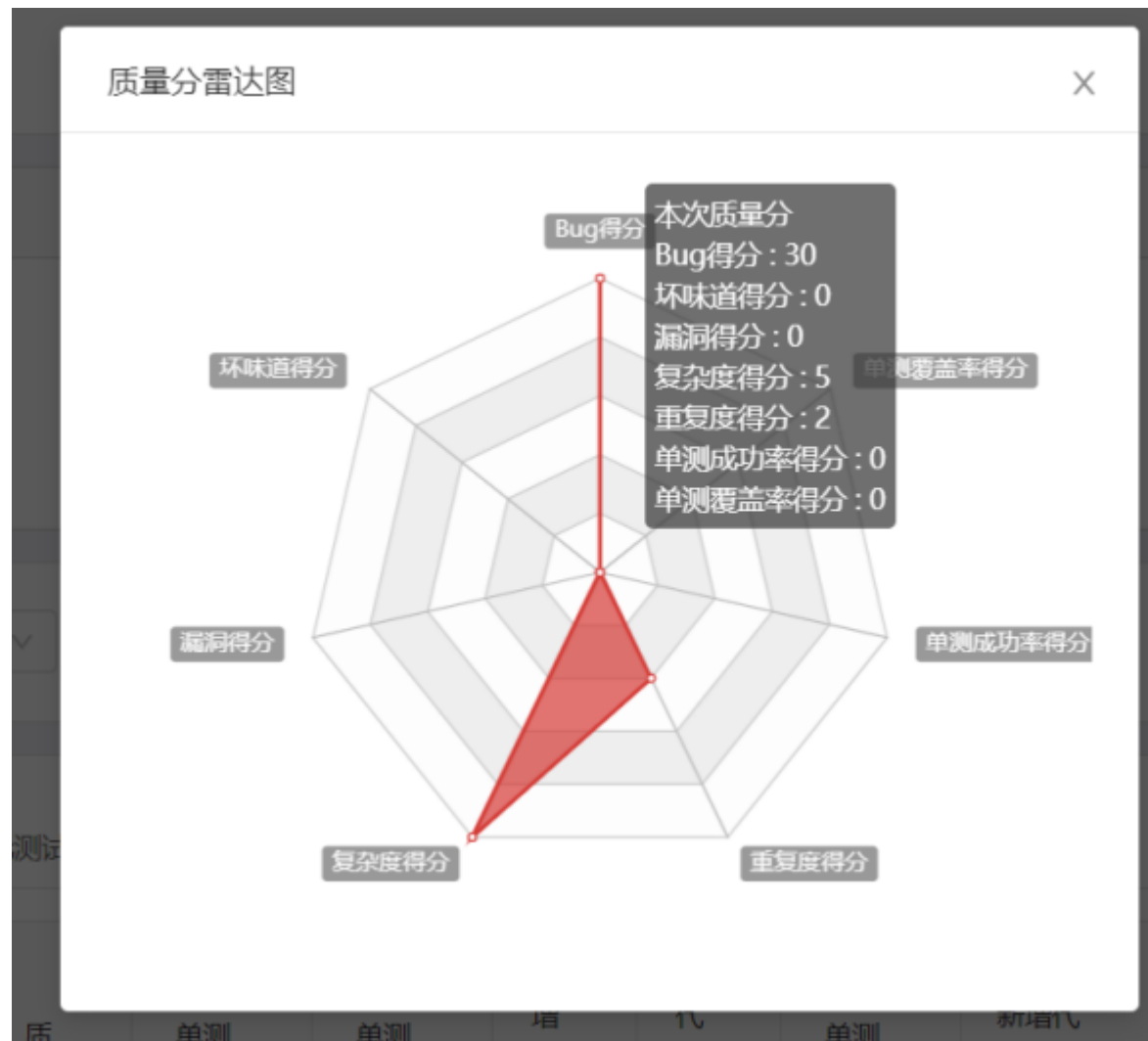
具体指标(违反规则数? 规则严重等级?)还是技术债(可读性, 扩展性, 可维护性, 复杂度)?

质量总分 = 单测通过率 得分 + 单测行覆盖率 得分 + 重复度 得分 + 复杂度质量分 + 坏味道质量分 + BUG质量分 + 漏洞质量分

- 单测通过率 得分 = 单测通过率 * 权重, 权重 5分
- 单测行覆盖率 得分 = 单测行覆盖率 * 权重, 权重 5分
- 重复度 得分 = 10 - 重复度, 权重 5分 ($\leq 5\%$, 满分, $\geq 10\%$, 0分, 每增加 1% 扣 20%))
- 复杂度质量分 = $5 * (1 - \text{千行复杂度} / \text{千行复杂度质量红线})$, 权重 5分 ($>$ 质量红线, 0分)
- 坏味道质量分 = $7 * (1 - \text{千行坏味道阻断数} / \text{千行坏味道阻断数质量红线}) + 2 * (1 - \text{千行坏味道严重数} / \text{千行坏味道严重数质量红线}) + 1 * (1 - \text{千行坏味道主要数} / \text{千行坏味道主要数质量红线})$, 权重 10分 ($>$ 质量红线, 0分)
- BUG质量分 = $20 * (1 - \text{千行BUG阻断数} / \text{千行BUG阻断数质量红线}) + 7 * (1 - \text{千行BUG严重数} / \text{千行BUG严重数质量红线}) + 3 * (1 - \text{千行BUG主要数} / \text{千行BUG主要数质量红线})$, 权重 30分 ($>$ 质量红线, 0分)
- 漏洞质量分 = $28 * (1 - \text{千行漏洞阻断数} / \text{千行漏洞阻断数质量红线}) + 8 * (1 - \text{千行漏洞严重数} / \text{千行漏洞严重数质量红线}) + 4 * (1 - \text{千行漏洞主要数} / \text{千行漏洞主要数质量红线})$, 权重 40分 ($>$ 质量红线, 0分)

数据驱动

质量分雷达分析图





数据驱动

质量排行榜，驱动代码质量提升

质量分最优 Top 10

质量分最差 Top 10

1	100	construct_drawing_front-master
2	99	summer-cache-master
3	99	texture-compressor-master
4	99	rtree-3d-master
5	97	tms-master
6	96	decoration-param-editor-master
7	96	captcha-service-master
8	95	coach-center-master
9	90	kuka-master
10	90	soa-master

单测覆盖率最优 Top 10

单测覆盖率最差 Top 10

1	96%	rtree-3d-master
2	92%	construct_drawing_front-master
3	86%	summer-cache-master
4	86%	parametric-engine-master
5	82%	drools-tools-master
6	81%	shape-translator-master
7	80%	brep-master
8	79%	math2-master
9	76%	i18n-diy-service-deploy
10	75%	report-service-deploy

问题总数最多 Top 10

问题总数最少 Top 10

1	0	AutoFilter-master
2	0	yuncore-master
3	0	snow-master
4	0	message-service-deploy
5	0	JOPanorama-master
6	0	cps-master
7	0	iphonejallodesigner-master
8	0	kuka-master
9	0	tools-common-master
10	0	cicd-master

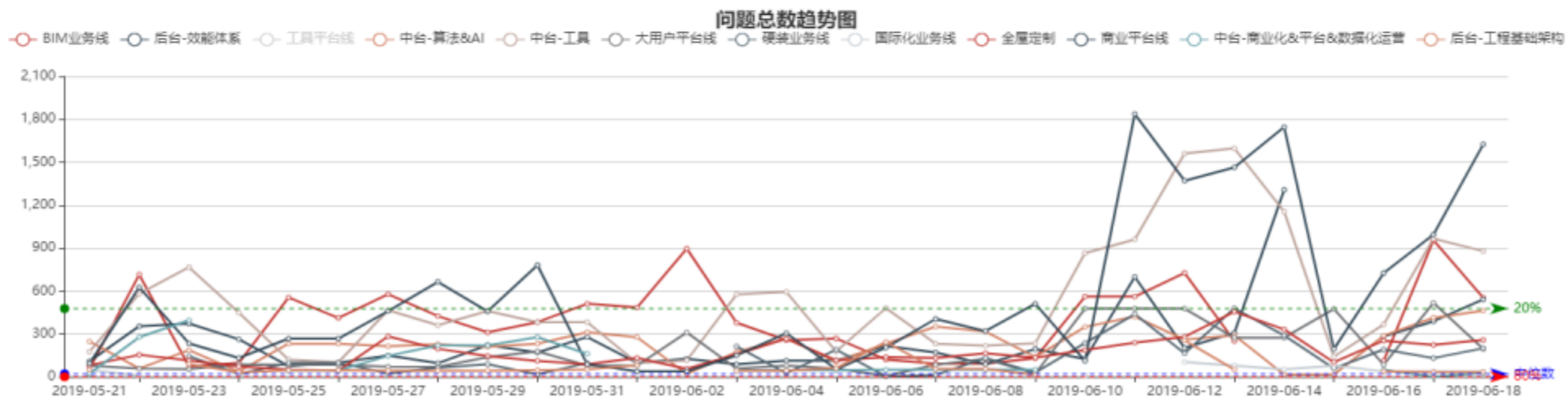
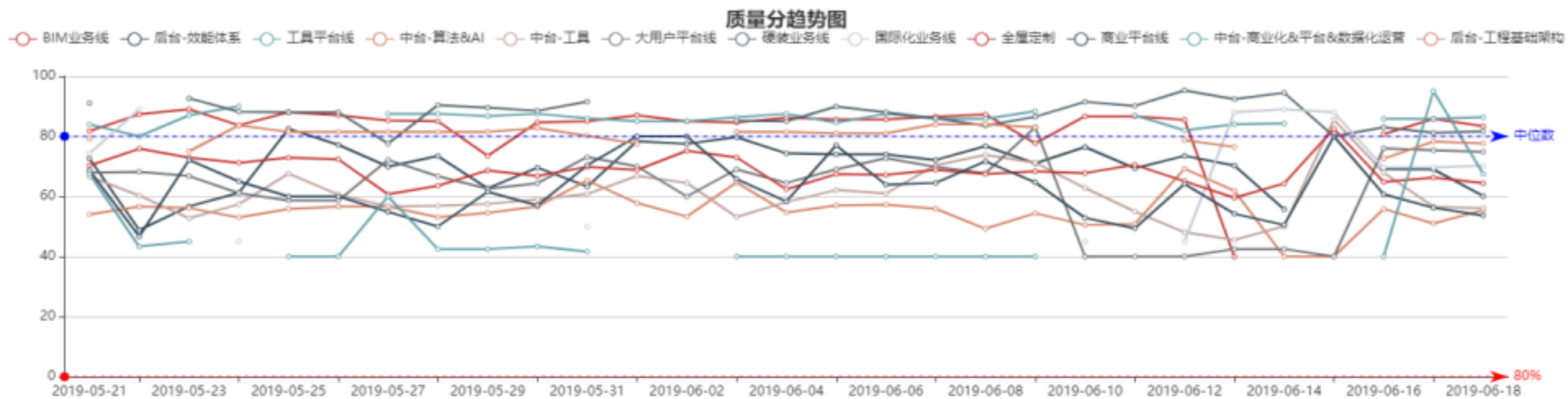
单测通过率最差 Top 10

单测通过率最优 Top 10

1	100%	tob-ms-master
2	100%	base-boot-master
3	100%	dcs-design-master
4	100%	plan-image-service-deploy
5	100%	parammodelservice-master
6	100%	parammodel-db-master
7	100%	plan-spider-master
8	100%	projectstorage-master
9	100%	revisioncontrol-master
10	100%	param-ceiling-service-master

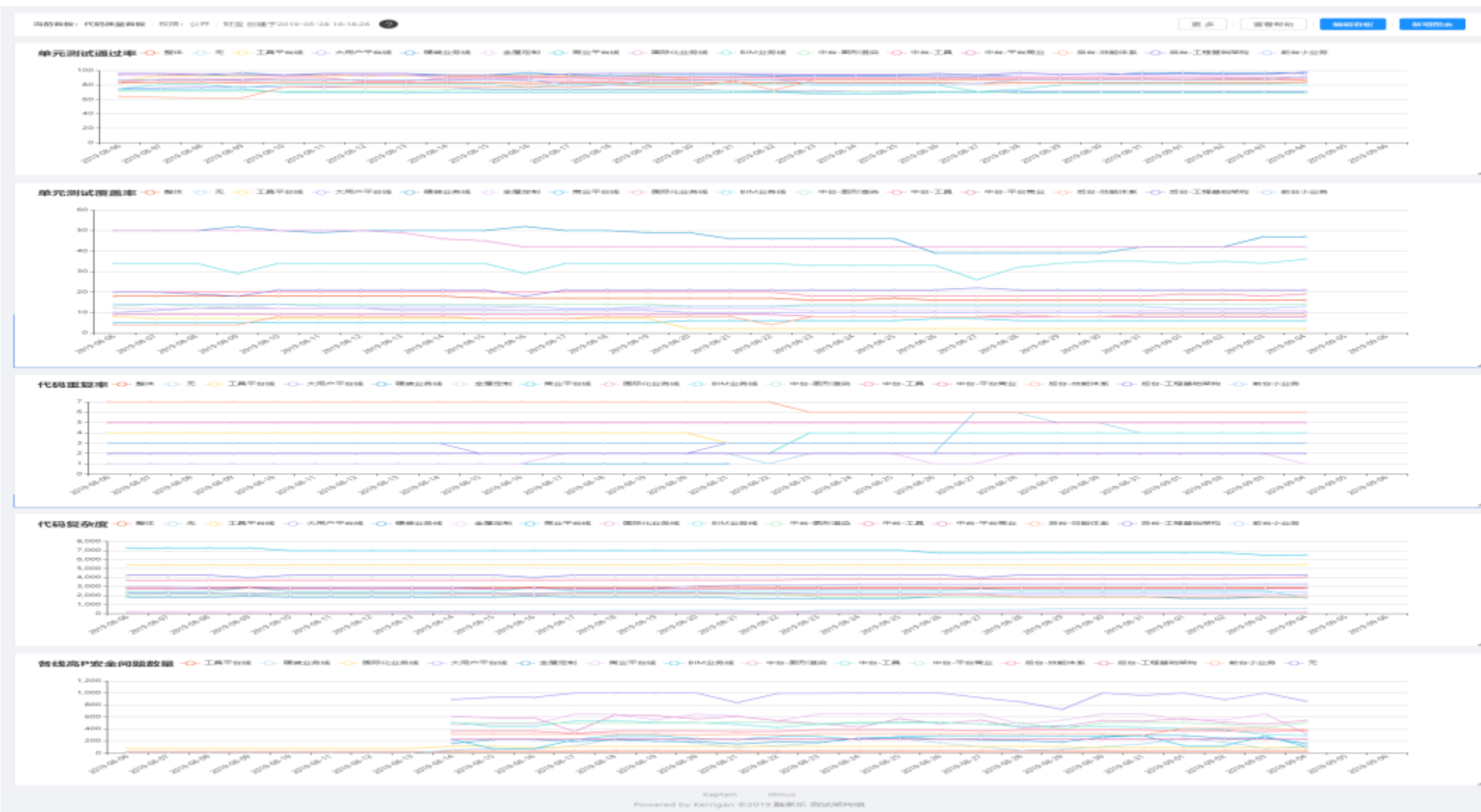
数据驱动

以业务线为维度，统计代码质量，驱动代码质量提升



数据驱动

以业务线为维度，统计代码质量，驱动代码质量提升

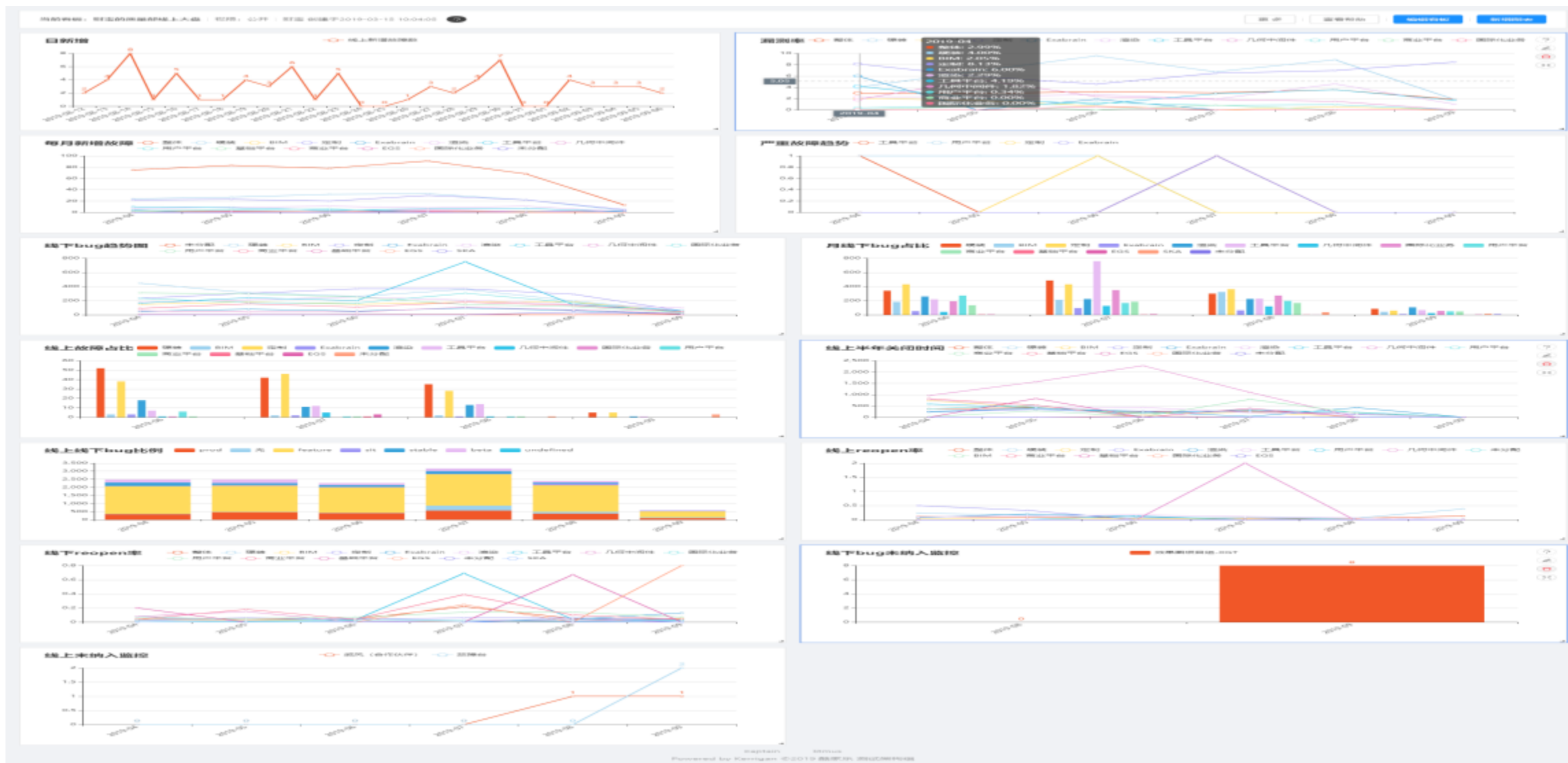


数据驱动

度量大盘：不度量，无改进



度量大盘：不度量，无改进





CD流程卡点

未达到质量要求的代码不允许部署上线

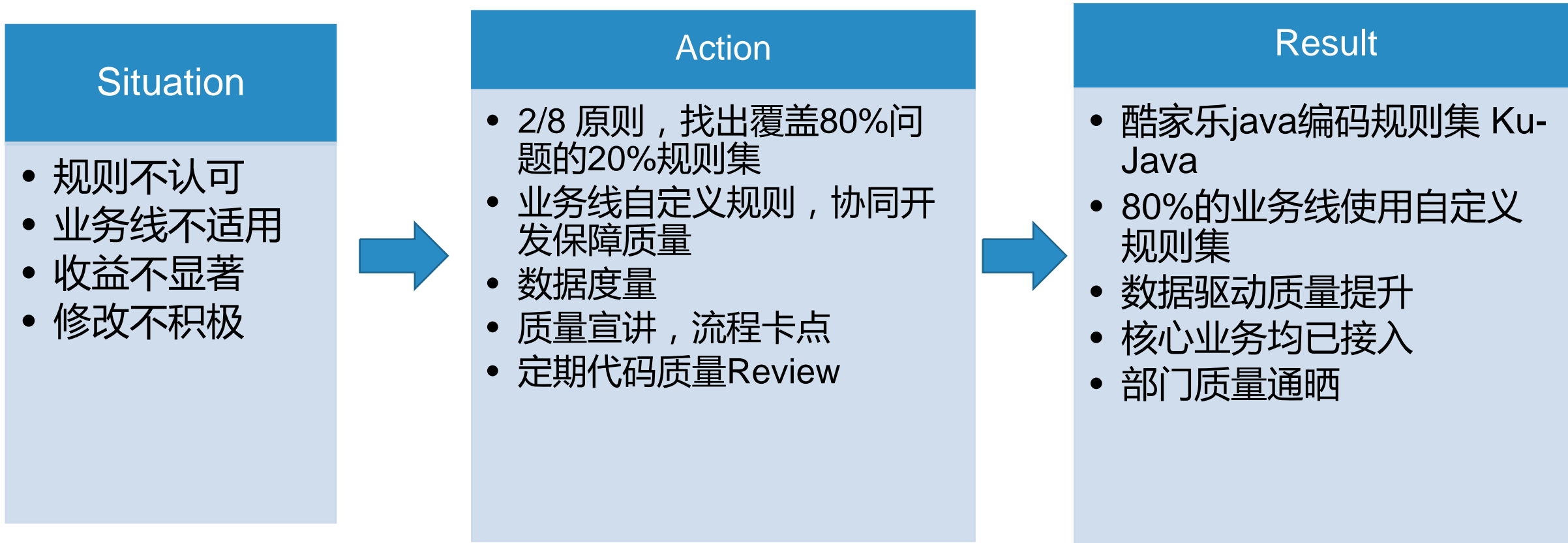
度量指标	度量结果	目标值	操作
单元测试相关	采集成功		
单测覆盖率	0.00%	≥30%	详情
单测数量	15	≥10	详情
单测错误数	0	≤10	详情
接口测试相关	采集成功		
接口测试通过数	392	≥200	详情
接口测试总数	401	≥380	详情
接口测试通过率	98.00%	≥95%	详情
接口测试行覆盖率	58.00%	≥50%	详情
代码质量相关	采集成功		
质量分	45.00	≥60	详情
违反编码规范	18	≤0	详情
静态扫描BUGs	0	≤0	详情
高等级安全漏洞	0	≤0	详情
二方包检测	采集成功		
SNAPSHOT包数量	0	≤0	详情
处于黑名单的包数量	0	≤0	详情
不在规定版本的包数量	0	≤0	详情

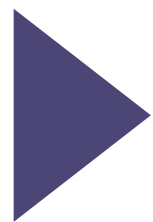


PART 03

落地

落地实践





PART 04

愿景

• 代码度量平台现状 •

代码度量统计

- 单测通过率
- 圈复杂度
- 行覆盖率
- 质量分

CICD集成

- 集成gitlab CI
- 集成Moon研发管理平台
- 流程质量卡点



编码规范

- 支持多种开发语言支持(java, go ...)
- P3C规则、FindBugs等规则

安全审计

- 支持多种开发语言(java, go ...)
- 支持多种漏洞类型(owasp top 10...)

• 代码度量平台成果 •

- 研发业务线100%接入代码度量平台
- 核心服务100%接入质量卡点
- 日均扫描次300+，发现问题1000+
- 高危三方依赖包检测，卡点研发升级版本
- 清除代码硬编码敏感信息
- 质量分平均提升30%

质量左移，赋能开发

- 扫描前置
- 简单问题自动HotFix
- 自动生成单测用例模板
- 编码提示最佳code snippet

Q&A





Thank You