

如何通过数据运营驱动技术升级
王培安

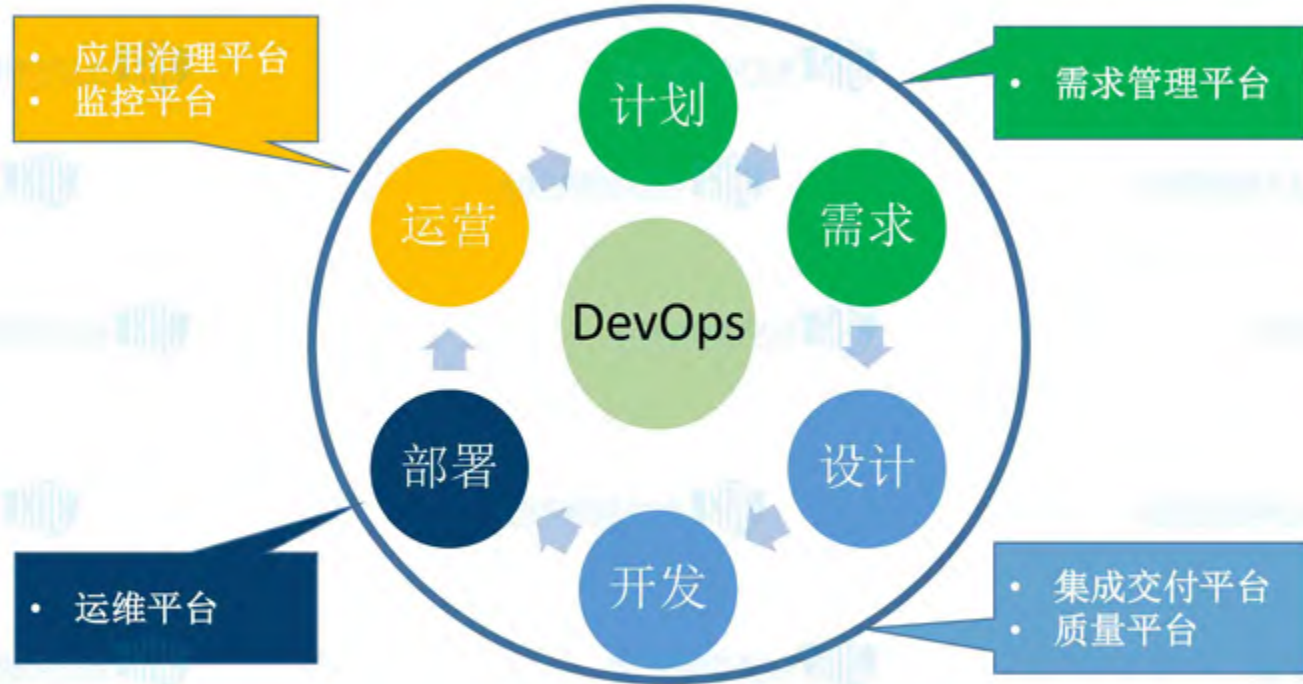


理想的DevOps模型，计划、需求、研发、上线、运营数据互通，快速反馈，不断优化



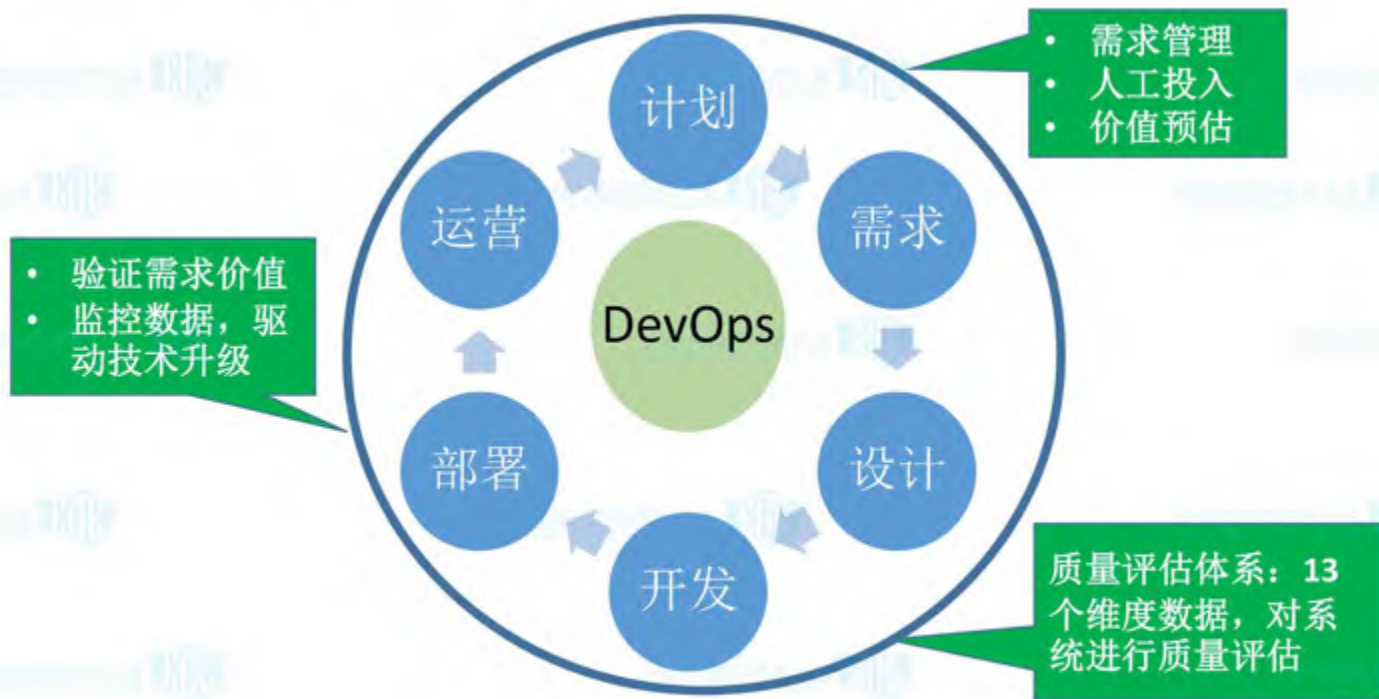
现实中，运营、需求、开发部署之间隔离的，不能形成一个完整的端到端的数据

找钢PaaS平台



找钢PaaS平台集成需求管理、集成交付、质量平台，运维平台、应用治理平台，监控平台，把DevOps各个环节打通

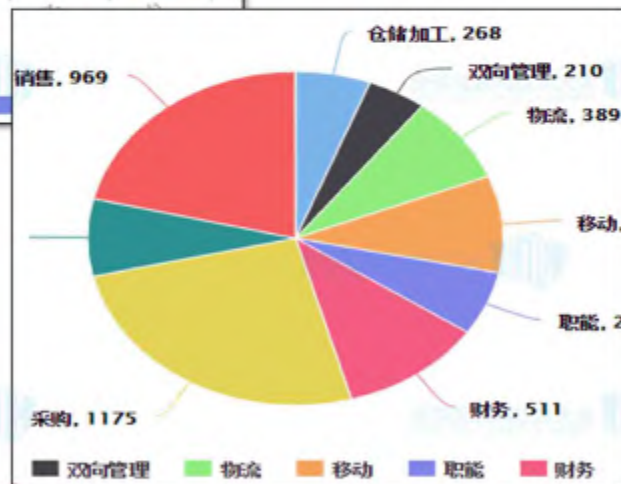
找钢PaaS数据流



项目价值、资源利用率、技术指标透明化，把产品、研发工作可视化，质量度量体系的数字化，推动技术升级

需求数据驱动

- 人效指标
- 上线数据



任务、资源、效率透明化，让工作可视化，并通过各类维度的度量体系将之数字化，为管理和决策做数据支撑

需求价值数据驱动

- 需求价值数据化？
- 满意度调查
- 产生交易量

- 通过数据化运营，对项目价值进行一定的度量，防止**盲目**的项目开发
- 通过项目价值数据化运营，**下线**无价值系统，**降低**运营成本

交付质量度量



集成交付平台



■ 持续集成

通过13个维度数据，从代码编写到系统上线，实现全流程的质量把控

■ 持续交付

设置关键质量指标的标准，如果达不到相应的指标，不能进行发布

■ 数据度量

明确的质量标准，清晰定义系统质量：优秀、合格、不合格

■ 多维度视角

开发、测试、经理、总监关注点不同，展示不同维度的数据

集成交付平台架构图

持续交付平台一站式服务

代码质量度量模型

持续集成微服务

Bug + 手动用例执行

代码评审

代码质量

合并代码

编译

单元测试

测试覆盖率

API自动化

UI自动化

安全扫描

性能测试

发布系统

线上监测

分支策略统一

测试环境策略统一

GitLab/Jenkins/Sonar/Gerrit/JIAR

质量度量维度

基本要求

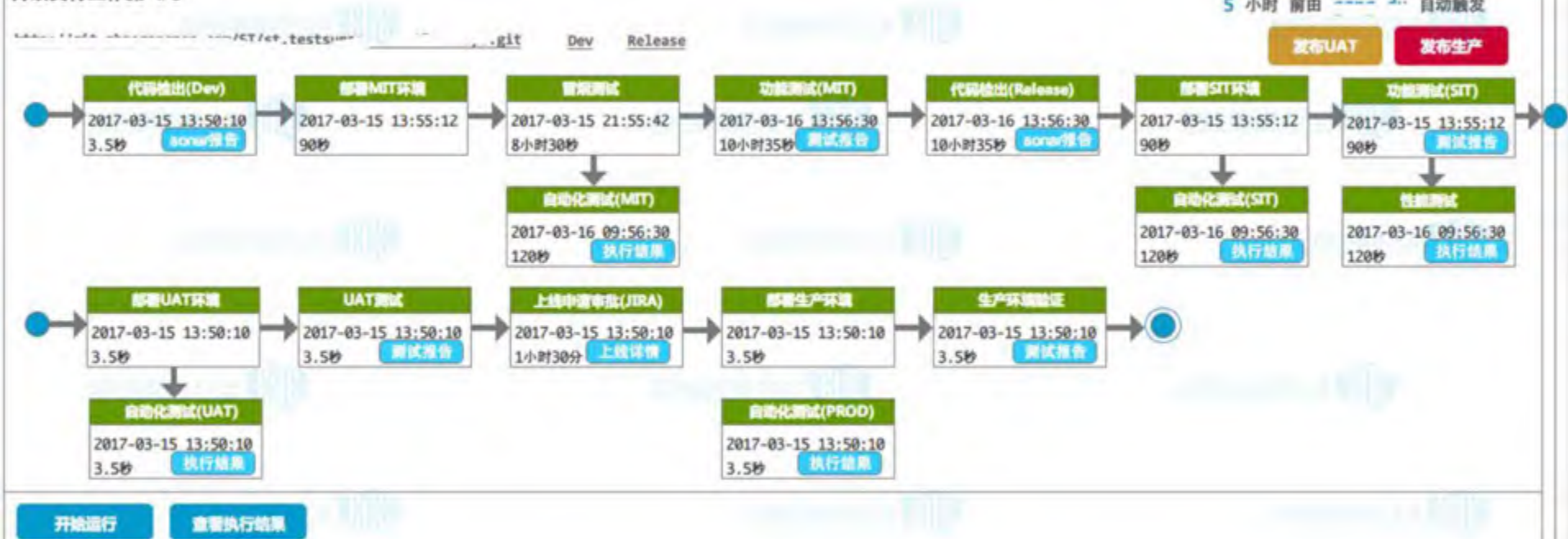
中级

高级

质量组成	评估维度	检查方式	描述	标准	评级标准
规范性	阻塞	自动	SONAR检查的阻塞问题	无新增	合格：必须
	严重问题	自动	SONAR检查的严重问题	无新增	合格：必须
	Code Review	人工，工具统计，追溯	通过Review工具，形成可视化、可追踪的代码review机制	所有代码提交必须经过code review	合格：必须
可读性	API 注释	自动	所有public都用JAVA doc格式注释	100%	优秀：必须
	代码重复率	自动	代码重复检查	小于5%	优秀：必须
接口、单元测试	覆盖率	自动	单元测试质量检查	服务端：60%	优秀：必须
	接口测试	自动	所有对外接口需要有测试用例覆盖	100%	合格：必须
	单元测试通过率	自动	单元测试通过率	100%	合格：必须
安全测试	通过安全行查扫描	自动	静态安全漏洞问题	安全问题：0缺陷	合格：必须
性能测试	通过benchmark测试	自动	性能基准测试	性能测试通过	合格：必须
编译	编译通过率	自动	编译通过率	99.99%	优秀：必须
发布	发布通过率	自动	发布成功率统计	99.99%	优秀：必须
用例执行	执行通过率	人工，工具统计，追溯	用例平台的测试用例执行情况	冒测：P1-P2级 MIT/SIT:P1-P4级 UAT:P1-P2级	合格：必须
Bug情况	线下Bug	人工，工具统计，追溯	JIRA的Bug统计	P1-P2	合格：必须
	线上Bug	人工，工具统计，追溯	JIRA的Bug统计		

项目开发全流程的展示

持续交付 workflow #8



通过13个维度的数据，从代码开发到系统上线全流程的质量度量，关键点不合格的，不允许发布

明确的标准

25
优秀

详情

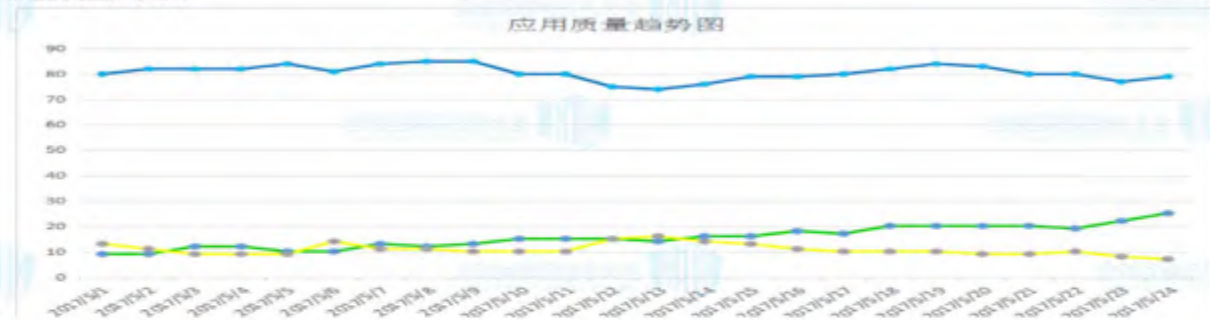
79
合格

详情

7
不合格

详情

质量度量趋势图



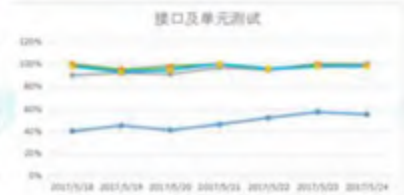
研发部: [下拉菜单] 研发站: [下拉菜单] 应用别名: [下拉菜单] 质量等级: [下拉菜单]

最后更新时间: 2017-02-08 15:45:08 ~ 2017-02-08 15:45:08

互调 重置

应用别名	质量等级	代码规范性		代码可读性			接口及单元测试			安全问题	性能基准	编译	发布	用例执行	bug情况	
		阻塞问题	严重问题	代码评审	API注释	代码覆盖率	UT成功率	接口覆盖率	接口成功率							
...test.ui	优秀	0	0	100%	100%	4.2%	65%	100%	100%	100%	0	优	100%	100%	100%	通过
...prage.dem.ui	优秀	0	0	100%	100%	3.7%	75%	100%	100%	100%	0	优	100%	100%	100%	通过

应用质量度量趋势图

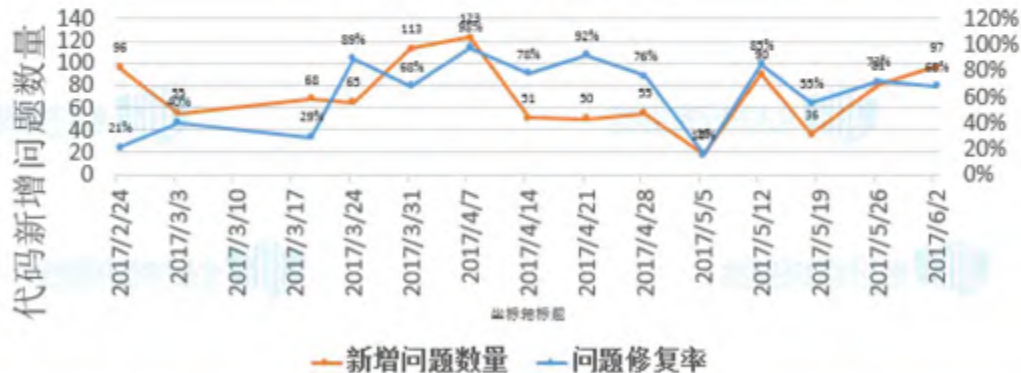


代码质量驱动技术升级

- 阻塞+严重问题修复
- 单元测试 + 代码覆盖率



代码新增问题统计



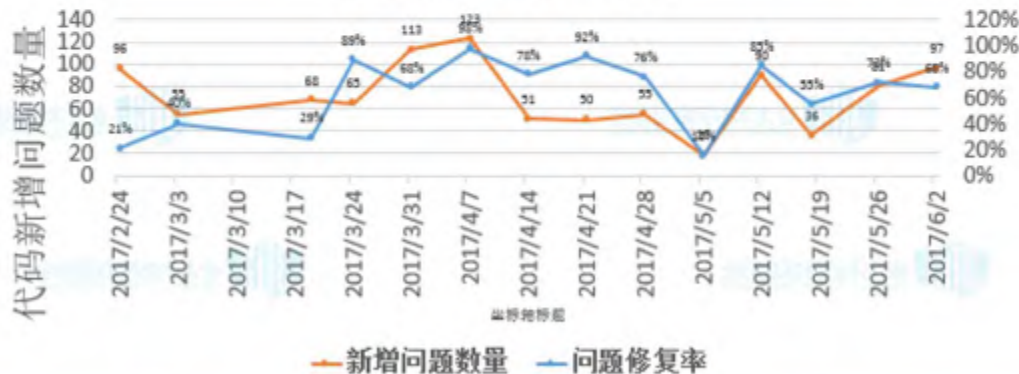
通过代码质量驱动，提升开发人员技术能力水平，保证系统上线前的质量

未修复问题排名

- 阻塞+严重问题修复
- 单元测试 + 代码覆盖率



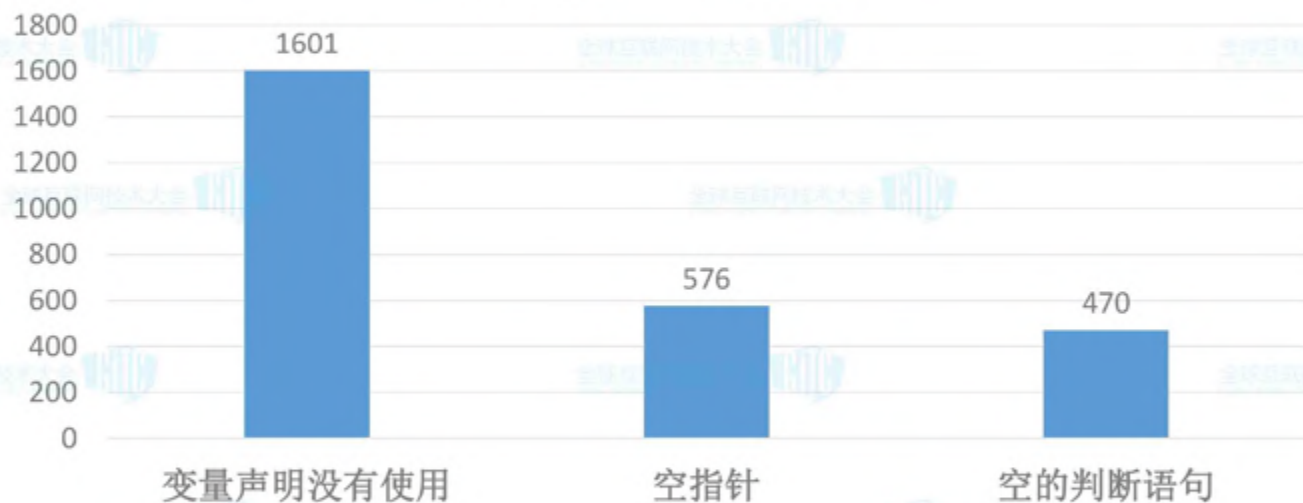
代码新增问题统计



通过代码质量驱动，提升开发人员技术能力水平，保证系统上线前的质量

常见问题分析

排名前三的问题



不断根据常见问题排名，针对性培训，驱动技术人员水平提高

测试用例覆盖

- 手动用例
- 自动化测试
- 性能测试

测试用例总数趋势图-12周 (2017-03-27~2017-06-18)



部门	统计周期	用例总数	P1	P2	P3	P4	总计	P1	P2	P3	P4	总计
1	2017-06-18	6191	778	2358	2591	428	6148	4	36	8	9	43
2	2017-06-18	13737	6048	3258	3497	753	13552	135	37	8	8	188
3	2017-06-18	9497	3893	3843	2119	599	9452	2	42	8	8	60
4	2017-06-18	5723	1254	2178	1639	487	5558	55	78	58	4	127
5	2017-06-18	8638	2427	3488	3421	298	9636	3	8	8	8	23
6	2017-06-18	11545	1315	3438	6727	1062	11543	8	8	2	8	22
7	2017-06-18	6728	1688	1853	2185	798	6635	76	8	53	4	93
8	2017-06-18	10167	888	5872	2824	481	10157	2	14	8	10	40

手工用例和自动化测试的统一，客观度量被测系统整体测试覆盖率

测试用例通过率

		测试用例执行 ×		测试用例执行统计 ×		↑ ← ↓ → 刷新	
研发部:		研发组:		统计周期:		统计时间:	
时间段		请选择开始时间 至 请选择结束时间					
查询		重置		帮助			
查询条件:		根据部门查询 > 所有研发部					
部门		统计周期	执行用例总数	执行用例总次数	执行质量		
					通过率		
1	研发部	2017-06-20	1061	1061	100.00%		
2	研发部	2017-06-20	29	32	100.00%		
3	研发部	2017-06-19	6	6	100.00%		
4	研发部	2017-06-19	88	88	100.00%		
5	研发部	2017-06-19	7	7	100.00%		
6	研发部	2017-06-18	1	2	50.00%		
7	研发部	2017-06-18	4	5	80.00%		
8	研发部	2017-06-16	186	187	100.00%		
9	研发部	2017-06-16	2	2	100.00%		

通过功能测试的执行通过情况，驱动技术提高基本功能开发完成质量

线上监控驱动技术升级



监控平台



■ 日志系统

从服务器的、网络设备的、应用的等各种日志，都能灵活的适配并统一存储起来，提供方便易用的查询能力，快速定位分析问题原因。

■ 诊断中心

各维度采集的数据都会及时上报给诊断中心，通过个性化的规则设置，能够自动的甄别各种异常和隐患征兆，并及时的通过多种渠道通知到干系人，及时处理。

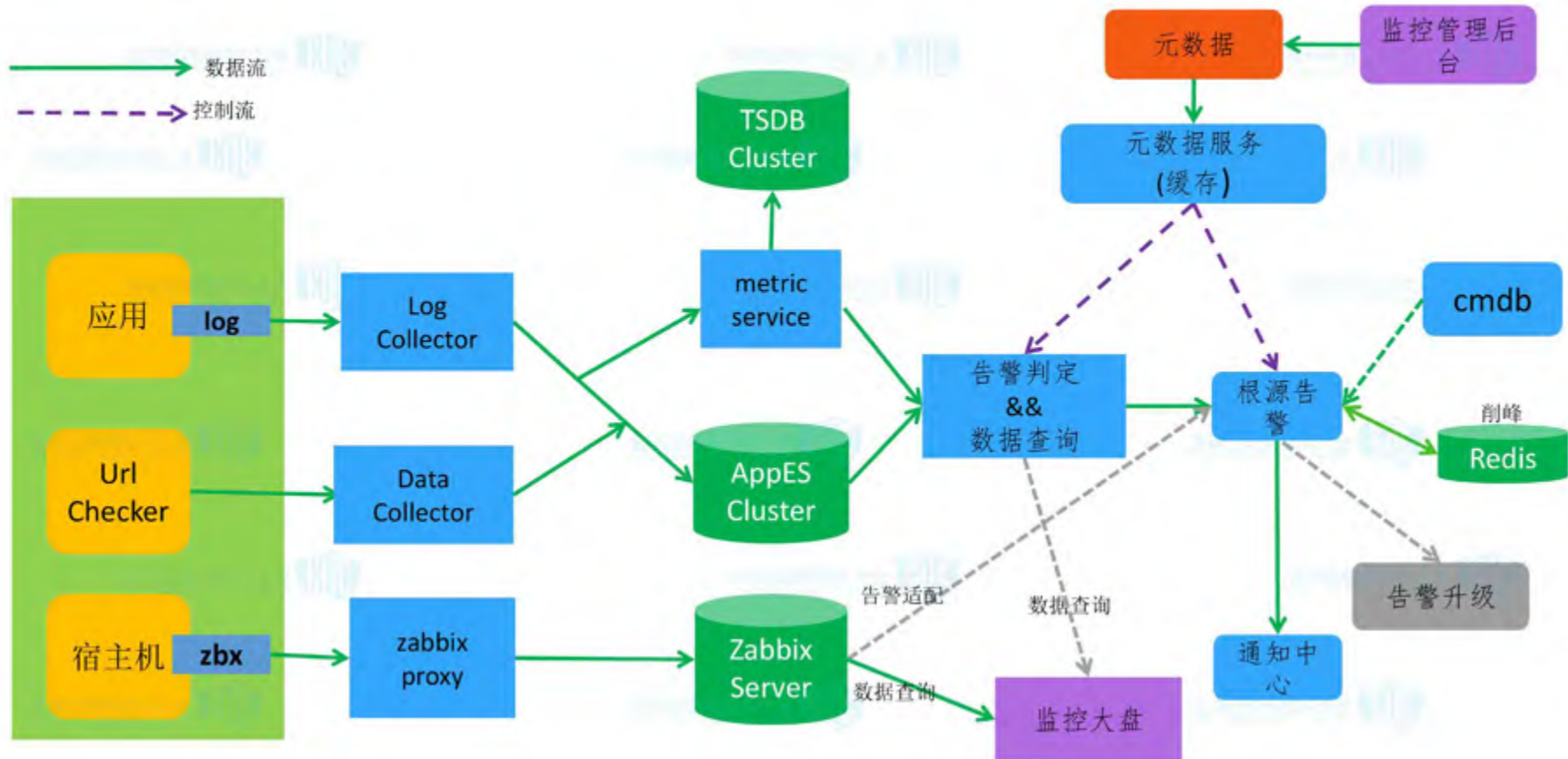
■ 全链路追踪

从用户请求开始，经过页面、接口、缓存、数据库再返回的完整路径的实时呈现，让任何隐患和故障都一目了然，快速精准定位到病灶，及时消除隐患。

■ 监控大盘

无论是运维人员关心的全盘信息，还是研发人员关心的应用信息，监控大盘都能满足不同层次的需求，并配置特定的呈现界面，让各类系统运行的状况都一览无余。

监控平台架构图

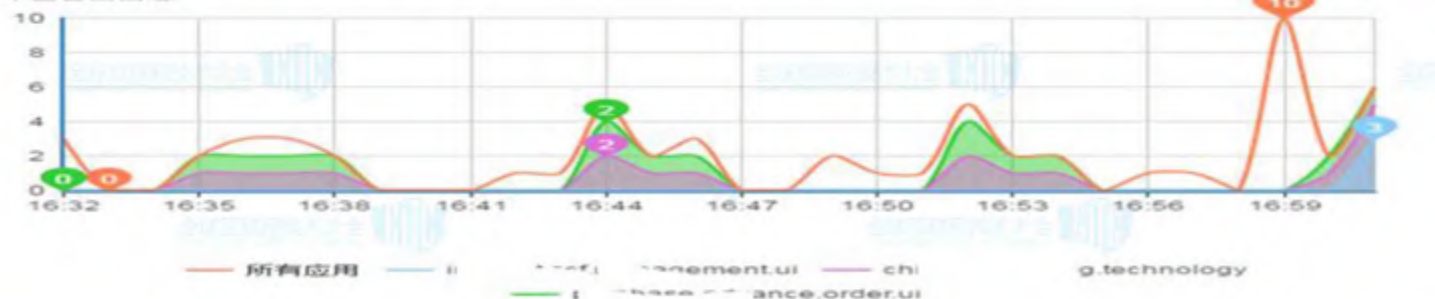


错误日志实时告警



报错日志趋势图 (30分钟)

不含警告日志

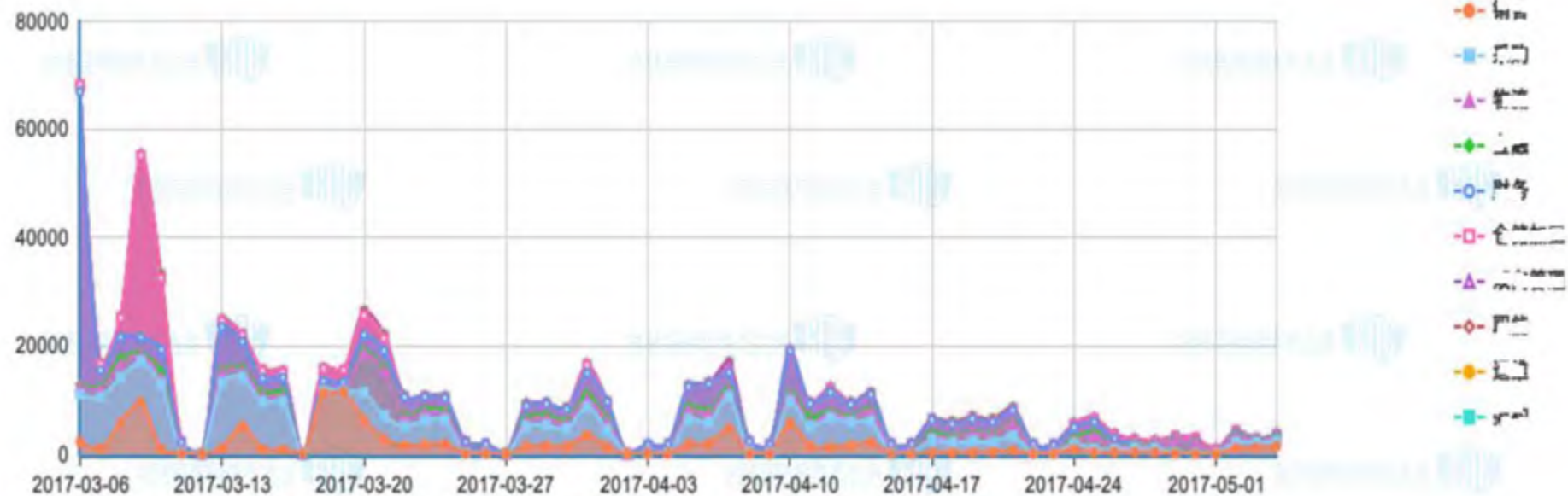


实时报错大盘，为快速定位系统故障和优化提供有利保障

错误日志历史统计驱动技术升级

业务线异常日志趋势图（按天）

不含警告日志



日志大盘历史统计数据，是将异常报错量比较多的系统、设备、应用都显示出来，方便运维、研发主动关注自己的系统健康度，并根据报错，不断优化

慢调用推动架构设计提升

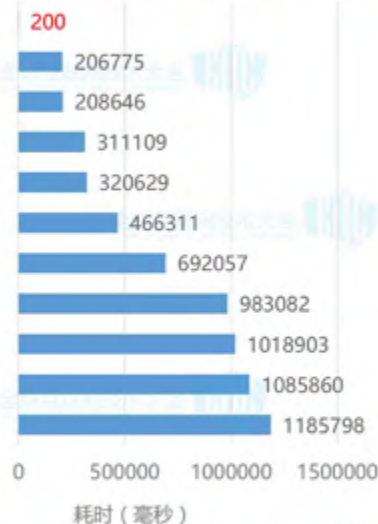
- 精准定位系统瓶颈
- 实时暴露慢调用的接口



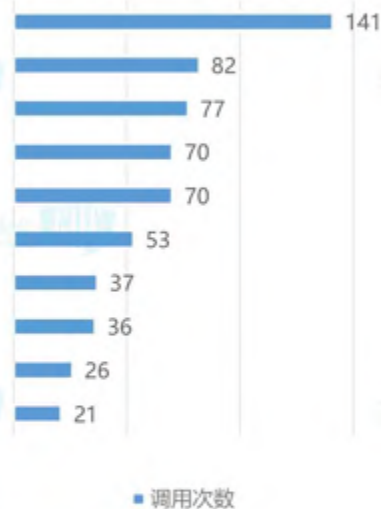
- 通过监控系统，实时暴露慢调用接口，推动架构设计上进行优化
- 服务调用和完整调用链信息，找出系统的瓶颈，定位到有问题（耗时）的方法

排名曝光驱动架构改造升级

XX月接口性能最差黑榜



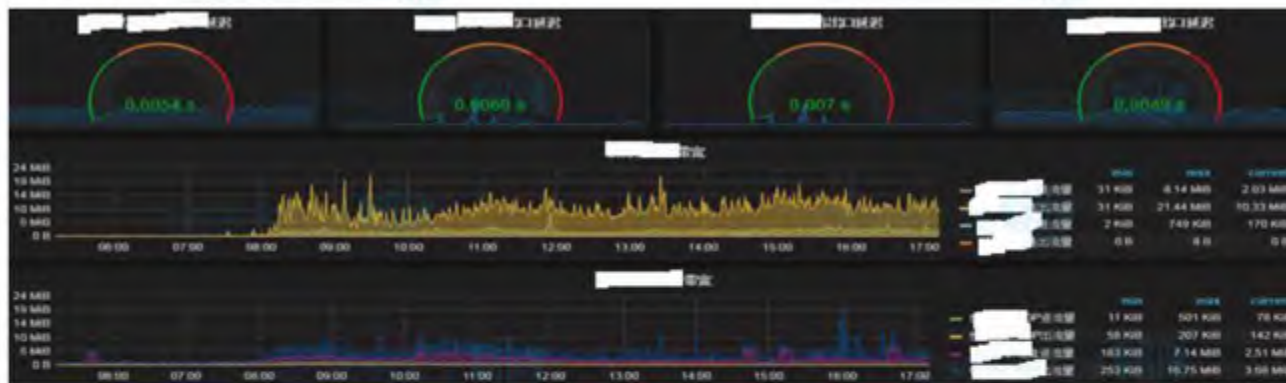
XX用调用次数黑榜



根据接口性能、接口调用次数排名曝光，驱动研发技术升级

基础监控驱动应用部署优化

- 实时监控基础资源瓶颈
- 多种维度展示物理资源信息

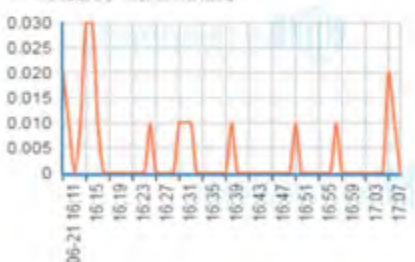


应用服务器【 ... 】

[GC信息](#) [更多信息](#)

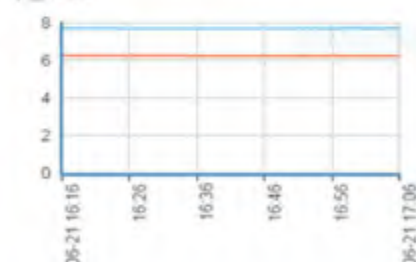
CPU负载

CPU负载大于50%为异常情况



空闲内存

单位: GB



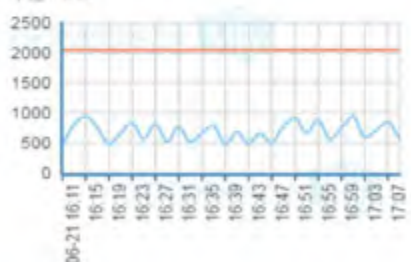
网络流量

单位: Kbps



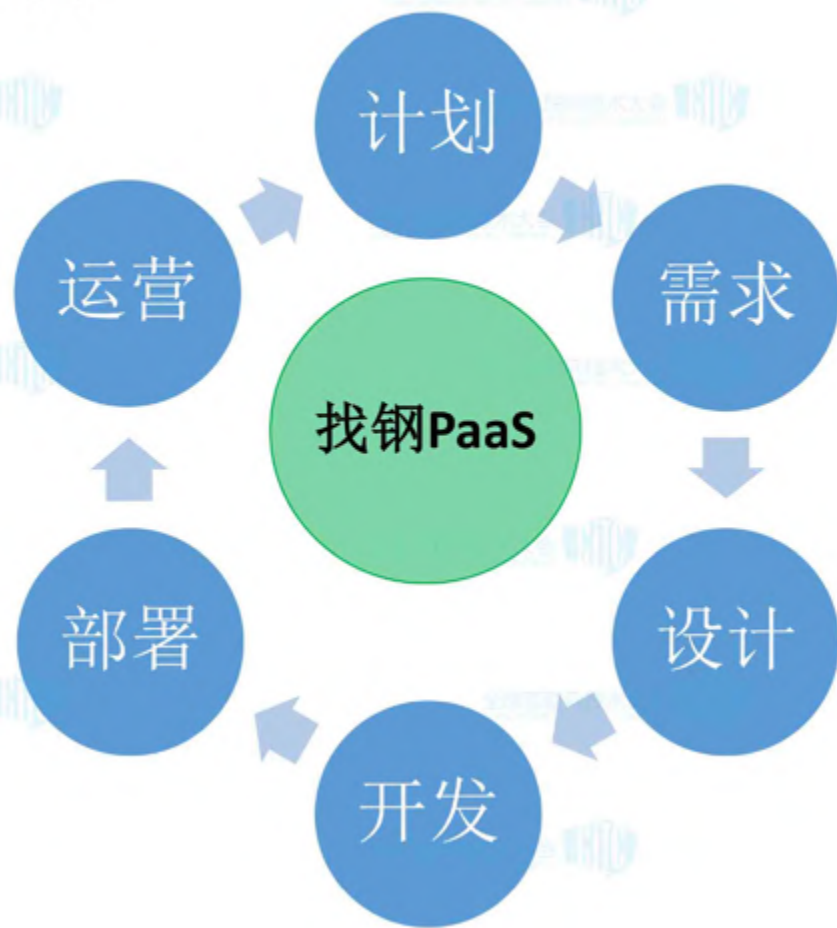
JVM内存使用量

单位: MB



根据应用维度、全局两个维度展示基础监控驱动应用部署架构优化

找钢PaaS平台



找钢PaaS平台架构图

PaaS协作平台

应用治理平台

服务治理

消息治理

配置治理

缓存治理

任务治理

数据治理

运维操作平台

配置管理

资源管理

应用管理

流程管理

数据库管理

网络管理

集成交付平台

持续集成

持续交付

包管理

监控平台

监控大盘

规则中心

现象诊断中心

告警中心

根源诊断中心

通知中心

安全平台

终端防护

系统防护

应用防护

网络防护

数据防护

访问防护

质量性能平台

手动测试中心

压测中心

自动化测试中心



Thanks!

Q & A