



GOPS 2019
ShenZhen



GOPS 全球运维大会

2019 - AIOps 风向标

大会时间：2019年4月12日-13日

指导单位：



主办单位：



泰康集团DevOps体系和工具平台建设之路

郭鸿

目录

- ➔ **1** DevOps建设背景
- 2** 研发痛点及措施
- 3** DevOps实施收益
- 4** TDS平台最佳实践

泰康保险集团作为一个世界500强企业，提供保险、资管、医养全方位服务，依托“活力养老、高端医疗、卓越理财、终极关怀”四位一体的商业模式，通过打造三个闭环，两大体系，依托科技驱动，极致体验，迈向大健康产业生态体系。泰康保险集团提出大健康战略，建设最大健康云，全力支撑泰康生态体系；因此对IT的支持提出了挑战。

建设最大健康云，全力支撑泰康生态体系！



外部环境

国家战略

国家发改委、财政部、工信部、科技部联合发起“2014国家云计算工程”。

国家级金融业务分布式平台及金融云研究课题。

金融行业

金融行业面临监管趋严和市场环境的变化，敏捷DevOps变革的时机已经成熟。金融业必须迎合移动互联网和智能化时代用户的需求，提升自己的能力。这一规律在过去几年中已经不断被证明。



国家战略



行业背景



金融行业



市场环境

行业背景

BATJ等互联网公司大规模使用DevOps工具链。

同业探索局限于边缘性的应用。

市场环境

互联网金融快速发展，金融系统架构亟需转型。

亿级客户规模、快速交付、按需弹性伸缩、成本可控。

内部困境

问题

业务线（**人寿、养老、资产、健康、医养、健投**）快速扩张，业务需求多而复杂

保险产品日趋互联网化、移动化、多样化

业务与技术创新加快

需求、开发、测试、运维等过程数据分散，精细化管理粒度较粗

需求



研发



创新



度量



挑战

传统需求管理已无法前驱和深入业务，需向产品经理转型

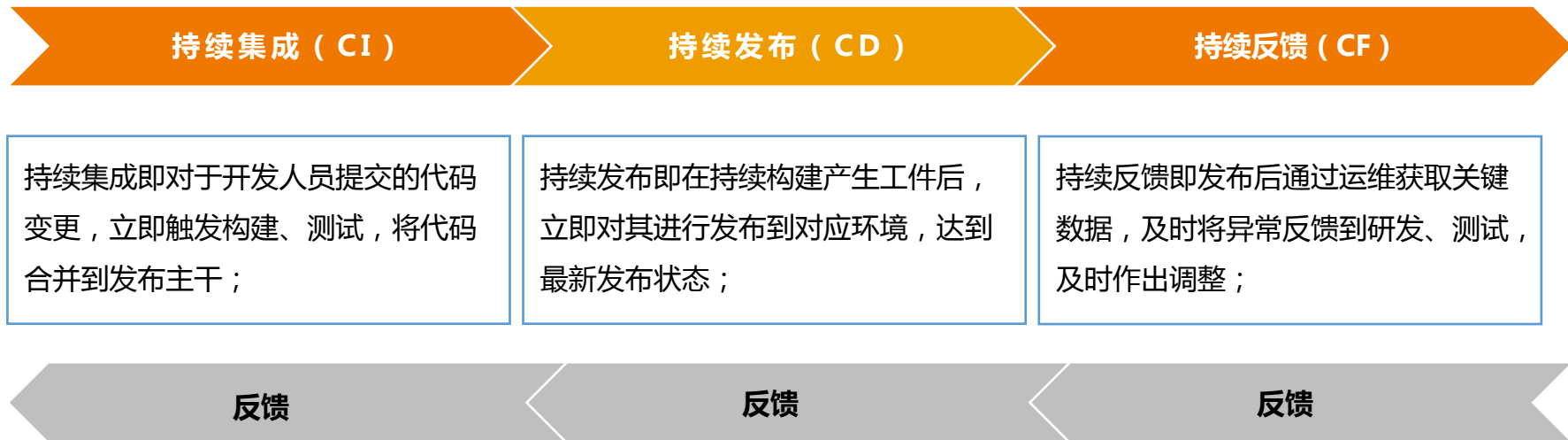
传统瀑布开发向敏捷开发转变

IT面临向微服务、敏捷、容器转变

持续交付过程信息可视化，传统管理向IT精细化管理转变

DevOps实施要点

DevOps 的要点在于**自动化和协作**，包括**持续集成(CI)**，**持续发布(CD)**及**持续反馈(CF)**三个关键概念，也涵盖了DevOps流程的三个关键阶段。



目录

1 DevOps建设背景

 **2** 研发痛点及措施

3 DevOps实施收益

4 TDS平台最佳实践

DevOps演进路线



泰康DevOps平台演进路线

研发痛点一 组织、人员、流程

组织

缺乏产品经理角色，无法前驱业务和深入业务，对研发效能重视度不够，缺乏自上而下的驱动力；

设置产品经理，测试并入研发团队，运维人员参与产品设计，成立流程效能组制定度量体系。

人员

研发团队=内勤+外包，能力参差不齐，疲于应付业务需求，对DevOps的认知和接受度较差；

成立集团/子公司DevOps专家组，制定DevOps实施方案和成熟度模型，建立社区文化，定期引入内外部培训，持续赋能。

流程

原研发流程复杂，工具多，各环节未打通，业务、开发、测试、运维存在严重部门墙，造成浪费；

引入精益思想，将传统的研发长流程精简成敏捷短流程，减少浪费和损耗。

成立公司级DevOps小组+人员赋能

组织与人员变革

人员赋能

转型专家小组



流程再造

原有流程

人员、流程、技术被“墙”阻断



需求 效率低

业务需求OA系统
BA需求ITSM系统和RDMS系统
SA需求通过RDMS系统



开发测试 反馈慢

开发人员RDMS系统和JIRA系统
测试人员使用RDMS系统和TMS系统



运维 故障多

运维人员ITSM系统

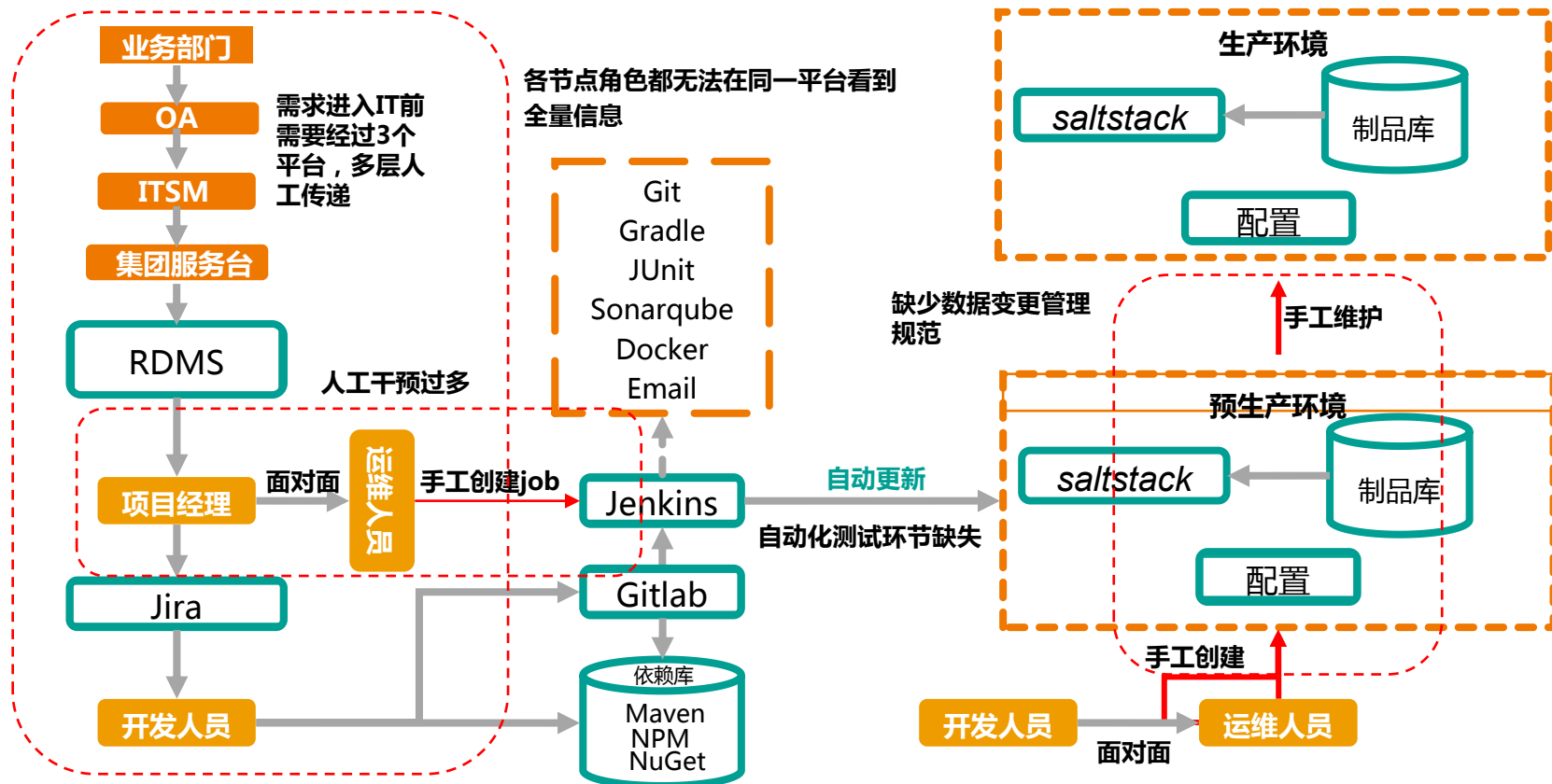


现有流程

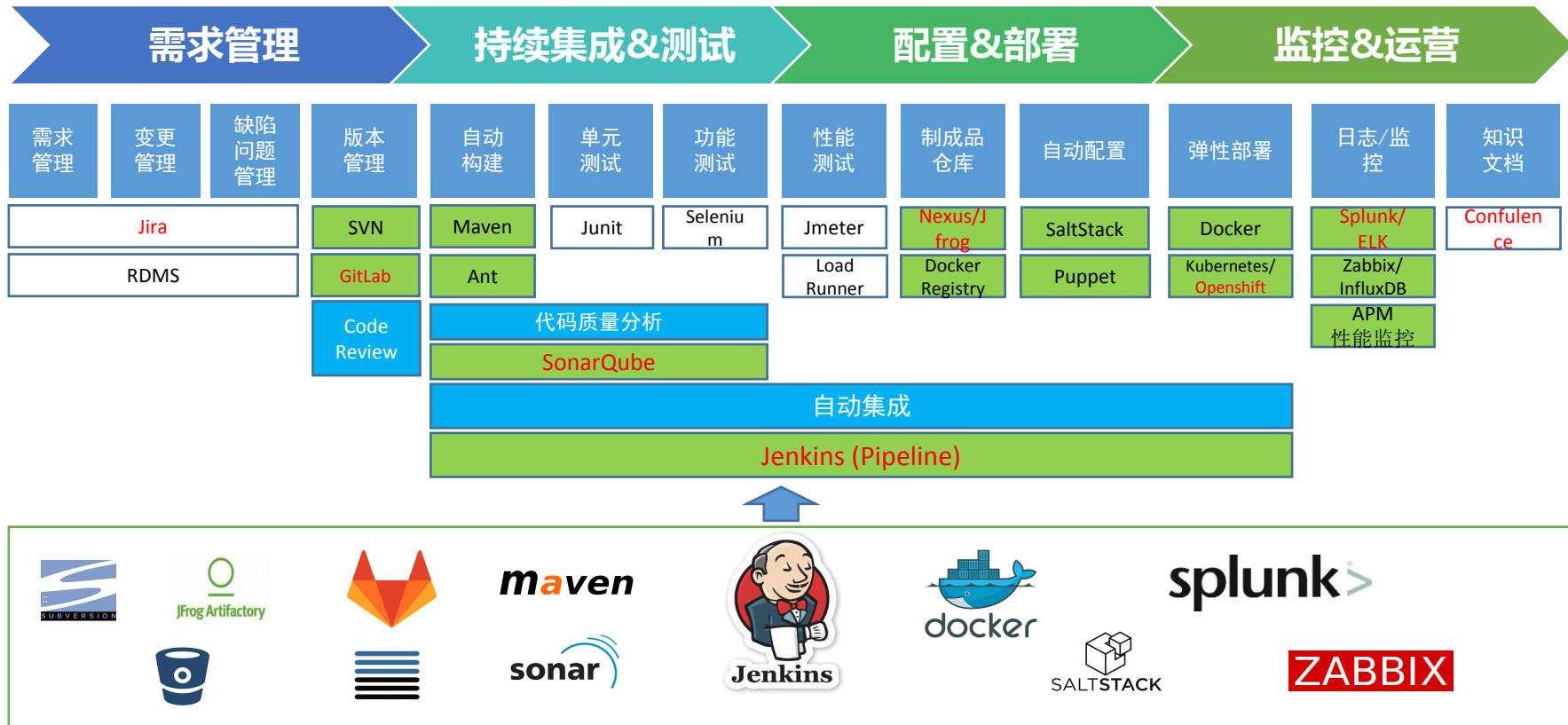
DevOps流水线满足业务快速、稳定交付的要求



痛点二 生产交付人工干预过多，自动化程度低



引入全流程工具链，提高自动化水平



设计以JIRA为源的全流程

Demo演示 / DEMOPS-9 系统需求 利用JIRA进行需求-用户故事-子任务-测试-代码-构建全流程关联

编辑 备注 分配 更多 将要做 正处理 已完成 管理

问题详情

类型: 需求 状态: TO DO
优先级: 重要 解决结果: 未解决
影响版本: 第一版本 修复版本: 无
模块: Demo演示开发组
标签: 根保康道

描述
点击这里添加描述

需求

创建父级 创建子级 链接父级 链接子级 需求树

DEMOPS-9 系统需求 需求拆分, 子集需求进度一目了然

DEMOPS-10 用户故事-1	↑	DONE	×
DEMOPS-11 用户故事-2	↑	TO DO	×
DEMOPS-12 用户故事-3	○	TO DO	×

测试用例

需求与测试用例、BUG对应, 真实展示开发质量

创建测试用例 链接测试用例

DEMOPS-13 测试用例-1	DEMOPS-9	↑	DONE	×
随机测试执行	DEMOPS-16			
DEMOPS-14 测试用例-2	DEMOPS-9	↑	DONE	×
随机测试执行				
DEMOPS-15 测试用例-3	DEMOPS-9	↑	IN PROGRESS	×
随机测试执行				
DEMOPS-17 测试用例-4	DEMOPS-9	↑	DONE	×
DEMOPS-18 测试用例-5	DEMOPS-9	↑	IN PROGRESS	×
DEMOPS-20 测试用例-6	DEMOPS-9	↑	TO DO	×
DEMOPS-23 测试用例-7	DEMOPS-9	↑	TO DO	×
DEMOPS-25 测试用例-7	DEMOPS-9	↑	TO DO	×

用户

经办人: 郭峰
报告人: 郭峰
投票: 0
关注人: 2 停止关注这个问题

日期

创建: 49 分钟前
更新: 15 分钟前

合作人

敏捷

在工作板中查看

缺陷

未找到相关缺陷。

Git 源代码

1 提交

比较代码 需求对应代码和合并请求清晰可追溯

分支

Create branch

DEMOPS-9-01 (后面有 2, 前面有 0)
DEMOPS-9-02 (后面有 0, 前面有 0)

Merge requests

Create merge request

DEMOPS-9 系统需求02 (OPENED)
DEMOPS-9 系统需求 (CLOSED)

标签 ①

问题链接

链接至

GitLab: Mentioned on commit - Update GreetingController.java D...

GitLab: Mentioned on merge request - DEMOPS-9 系统需求

GitLab: Mentioned on merge request - DEMOPS-9 系统需求02

需求对应开发代码一键跳转, 实时可见

Tempo

2019-03-01 - 2019-03-31

当前 报告 添加 Expense 记录工作日志

日期 时间 描述 已用

活动

所有 备注 工作日志 改动记录 活动 CI Builds Git 提交 版本

project-test-gh 索引 10 分钟前

apiuser 已提交 14 分钟前

代码完成提交、对比、提交构建结果实时查看

查看完整提交

Update GreetingController.java DEMOPS-9

project-test-gh master 779a333030b3f2e3e3f4dda1eb54dd176d37391 gitlab

所有 备注 工作日志 改动记录 活动 CI Builds Git 提交 版本

Jenkins返回构建结果并更新JIRA状态

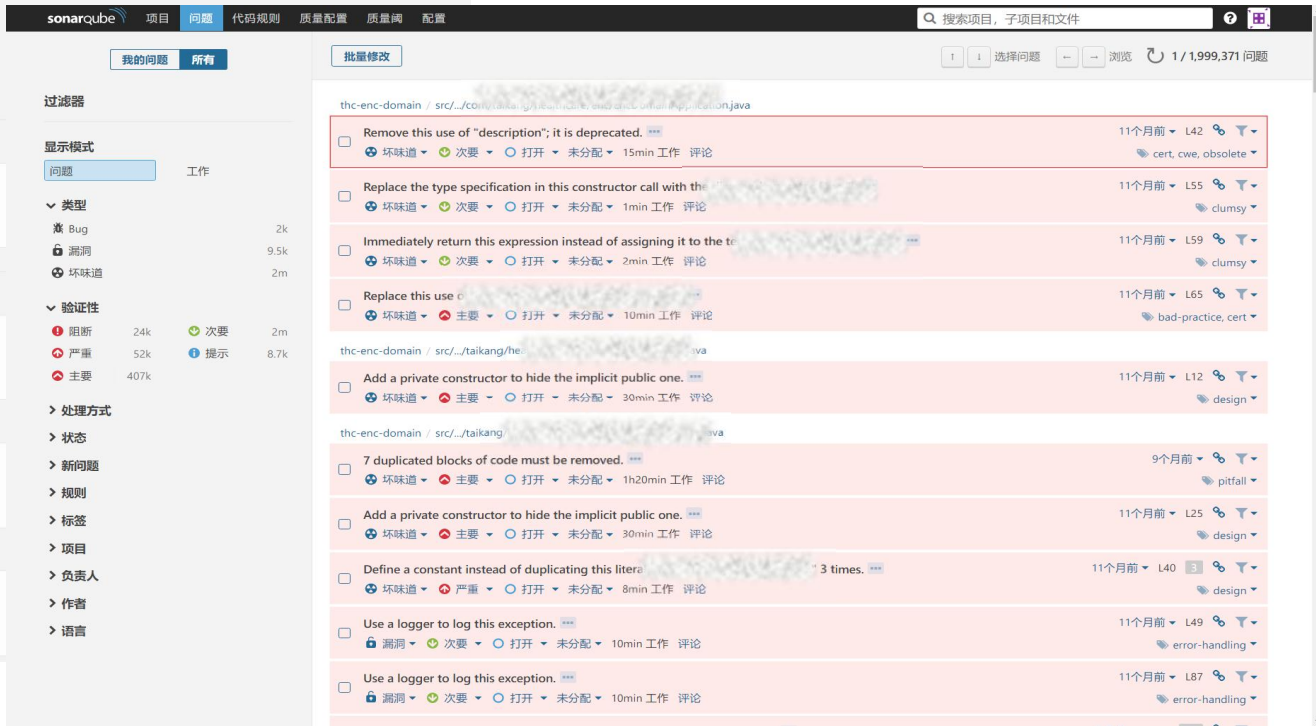
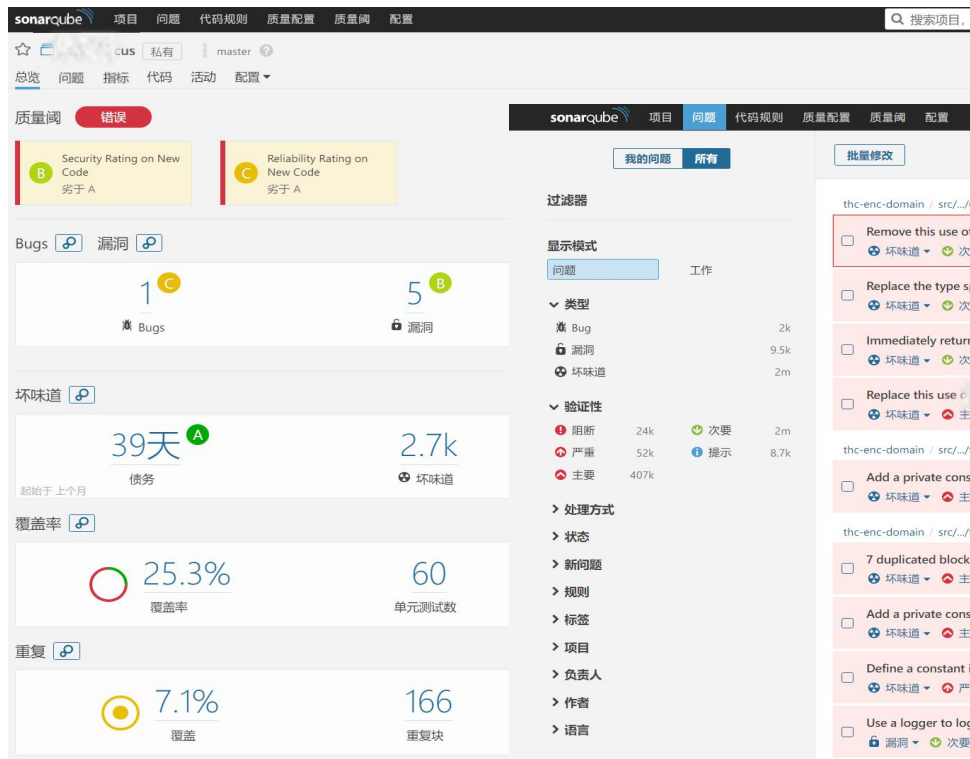
JIRA issue is updated

Related Issues: JIRA-23 Ran 2017-11-09 14:17 | Duration 12s
2 Total Tests | 2 Failed Tests | 0 Skipped Tests

JIRA-issue-update #93 Started by user guohong05
Related Issues: JIRA-23 Ran 2017-11-09 14:17 | Duration 21s
2 Total Tests | 2 Failed Tests | 0 Skipped Tests

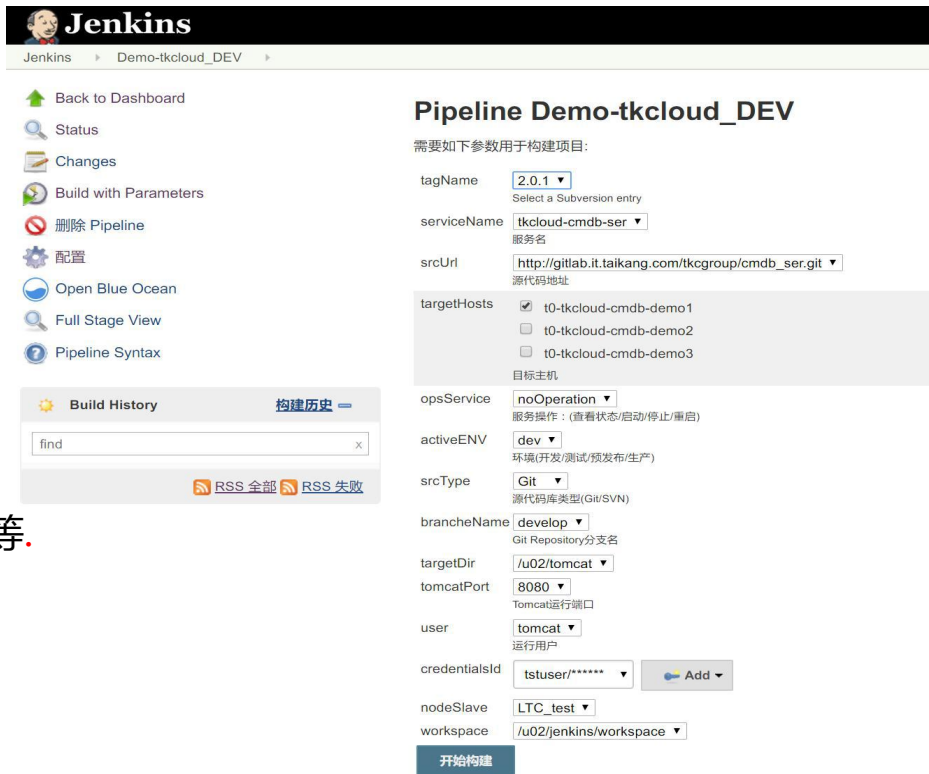
JIRA-all #19 Started by user guohong05
Related Issues: JIRA-23 Ran 2017-11-06 15:55 | Duration 14s
2 Total Tests | 1 Failed Tests | 0 Skipped Tests

引入Sonar静态代码扫描



建设以Jenkins pipeline为核心的CI/CD平台

- Job参数化构建
- 支持多主机灰度发布
- 服务(App)管理：start/stop/restart/status等
- 多环境支持：生产/开发测试
- 自动打包上传至Artifactory仓库
- 标准的Pipeline，由Groovy DSL编写；
- 基础设施即代码；
- 支持多种中间件：Tomcat、Jboss、SpringBoot、Nginx、NodeJS、Docker等；
- 支持自动化测试、压力测试、单元测试、代码检查等。



The screenshot shows the Jenkins web interface for configuring a pipeline. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build with Parameters, 删除 Pipeline, 配置, Open Blue Ocean, Full Stage View, and Pipeline Syntax. Below these is a 'Build History' section with a search bar and RSS links for '全部' and '失败'. The main content area is titled 'Pipeline Demo-tkcloud_DEV' and lists configuration parameters for a build job. The parameters include tagName (2.0.1), serviceName (tkcloud-cmdb-ser), srcUrl (http://gitlab.it.taikang.com/tkcgrou/cmdb_ser.git), targetHosts (t0-tkcloud-cmdb-demo1, t0-tkcloud-cmdb-demo2, t0-tkcloud-cmdb-demo3), opsService (noOperation), activeENV (dev), srcType (Git), branchName (develop), targetDir (/u02/tomcat), tomcatPort (8080), user (tomcat), credentialsId (tstuser/*****), nodeSlave (LTC_test), and workspace (/u02/jenkins/workspace). A '开始构建' button is at the bottom.

Jenkins

Jenkins > Demo-tkcloud_DEV

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Build with Parameters](#)
[删除 Pipeline](#)
[配置](#)
[Open Blue Ocean](#)
[Full Stage View](#)
[Pipeline Syntax](#)

Build History [构建历史](#)

find

[RSS 全部](#) [RSS 失败](#)

Pipeline Demo-tkcloud_DEV

需要如下参数用于构建项目:

tagName
Select a Subversion entry

serviceName
服务名

srcUrl
源代码地址

targetHosts ☒ t0-tkcloud-cmdb-demo1
☐ t0-tkcloud-cmdb-demo2
☐ t0-tkcloud-cmdb-demo3
目标主机

opsService
服务操作: (查看状态/启动/停止/重启)

activeENV
环境(开发/测试/预发布/生产)

srcType
源代码库类型(Git/SVN)

branchName
Git Repository分支名

targetDir

tomcatPort
Tomcat运行端口

user
运行用户

credentialsId [Add](#)

nodeSlave

workspace

[开始构建](#)

优化现有交付服务体系和流程

- **角色转变**：基础运维 向 平台运维转变
- **DevOps能力下沉**：加强子公司和集团各大业务线二级 (应用运维)梯队建设
- **资源交付以面向服务为目标**：收敛流程ITSM入口
- **统一化和扁平化**：统一的资源调配，统一的工作流程，统一的管理方法
- **服务手册文档化/产品化**：以公有云文档为模板，降低用户学习成本



基本信息

- 名称
- 负责人
- 备注信息
- ...



集群管理

- 开发环境
- 测试环境
- 生产环境
- ...



部署资源

- 程序包
- 配置包
- 文件包
- ...



构建信息

- 开发语言
- 开发框架
- 代码库地址
- ...



网络资源

- 负载均衡
- 内网域名
- 公网域名
- ...



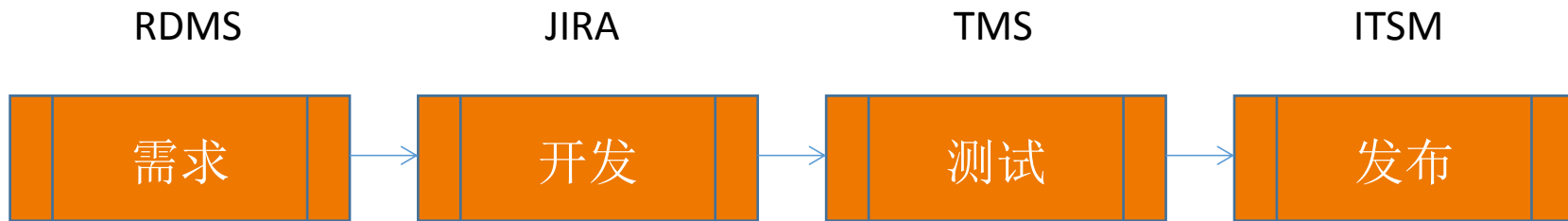
环境依赖

- 基础镜像
- 中间件
- ...

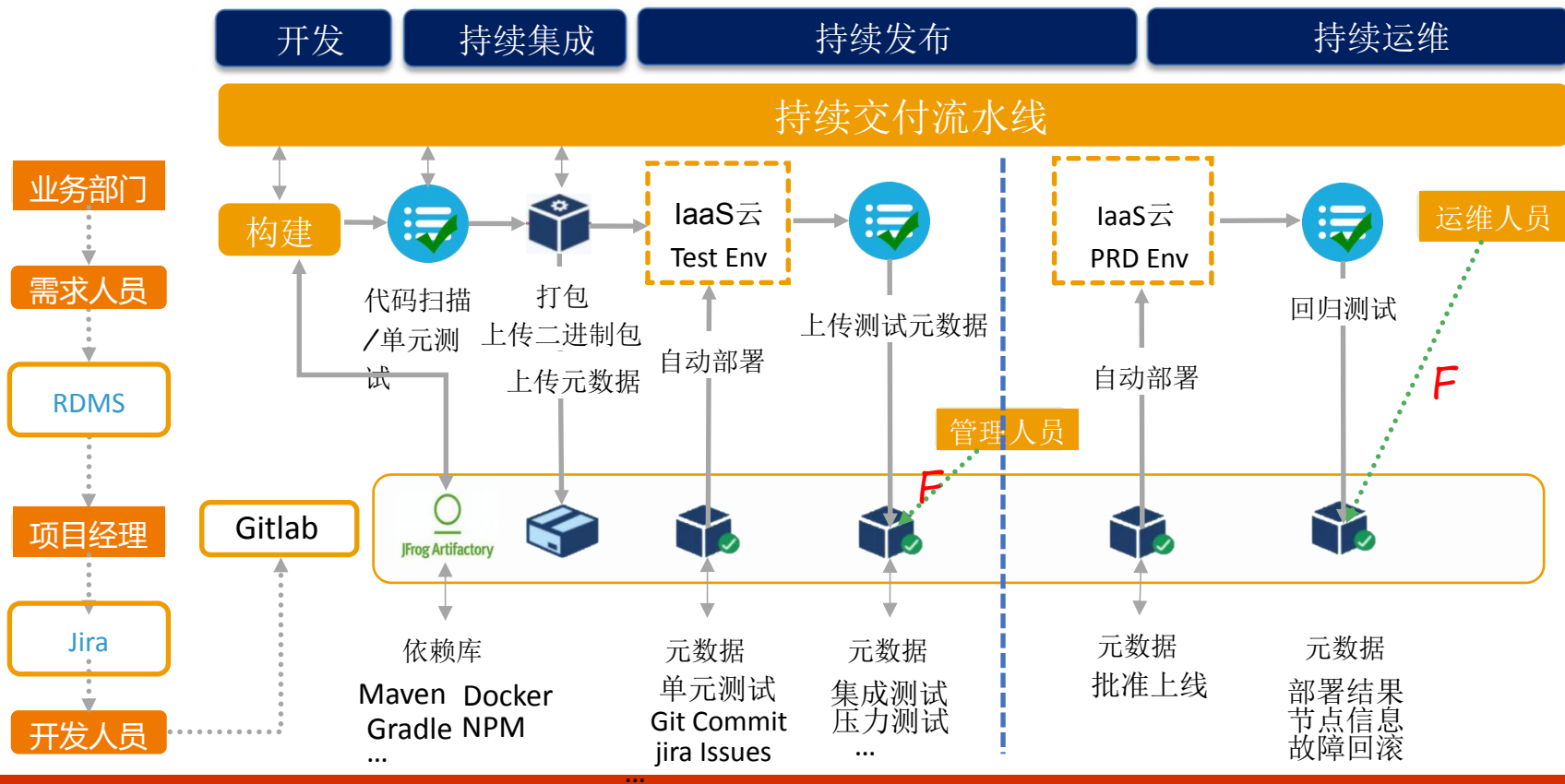
痛点三 缺乏质量关卡，上线问题频发

■ 测试点：研发过程研发人员单元测试、开发完成后提交测试部门手工测试：

■ 问题：缺乏有效的测试准入准出标准和上线规范



每个环节设置质量关卡

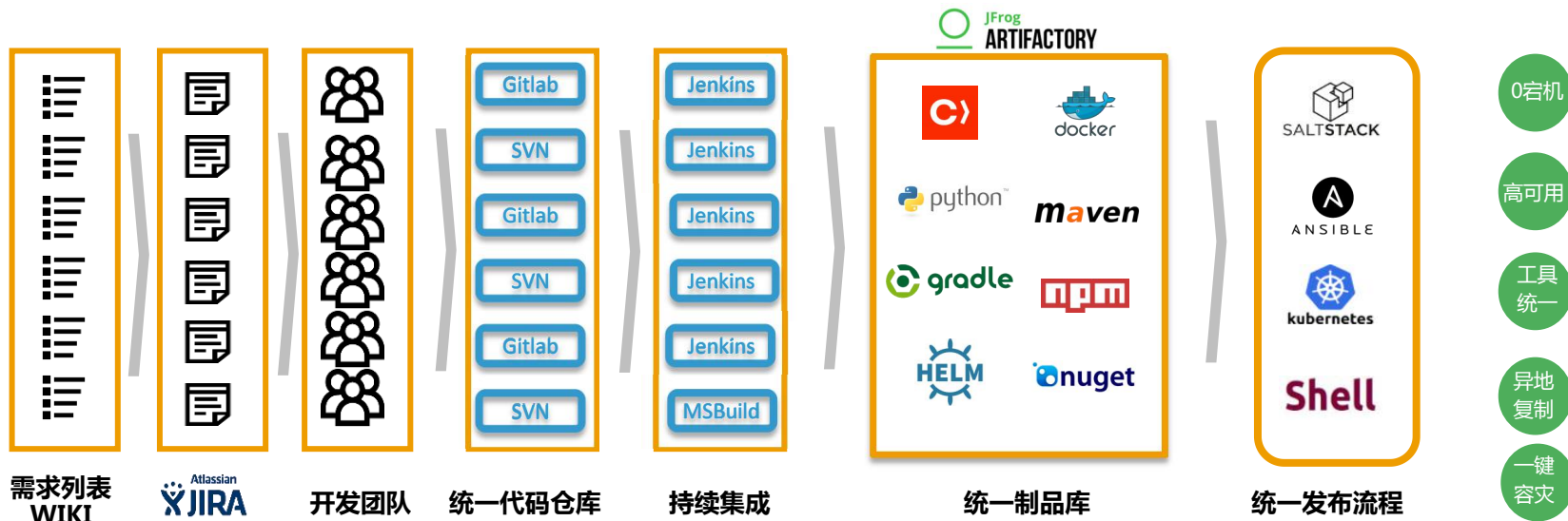


痛点四 版本管理缺失

- 代码通过测试的版本和发布版本不一致，导致上线后问题频发，开发、测试、运维互相推诿扯皮。
- 缺少统一的全语言制品管理，仓库运维成本高。

引入版本管理机制

从需求到发布保持版本号一致，引入Jfrog保证开发、测试、发布获取单一可信源
WIKI需求版本---JIRA发布版本---GitLab代码版本---Jenkins构建Job---Jenkins发布版本



痛点五 缺乏持续度量能力，团队质量把控力弱

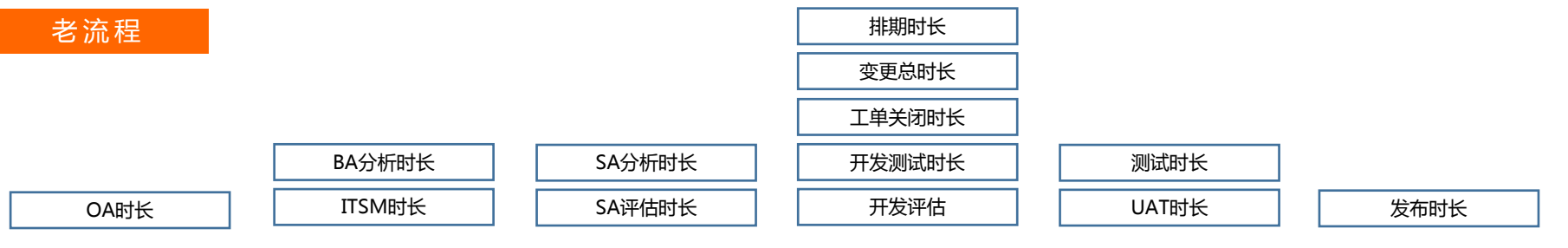
- 缺乏工程质量管理职能和角色
- 缺乏工程过程度量指标和体系

设计度量指标体系

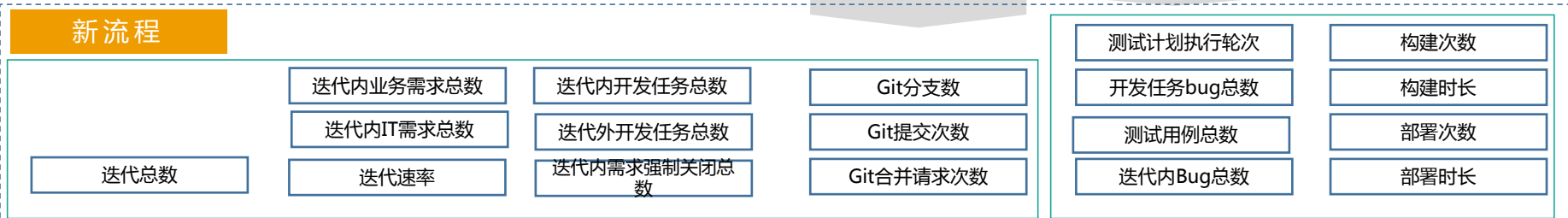
		线上缺陷密度	构建失败率	
交付吞吐率	开发交付周期	用例需求覆盖率	发布前置时间	
SA需求交付周期	代码重复率	测试通过率	发布频率	
BA需求交付周期	代码覆盖率	Bug有效率	部署成功率	故障恢复时间
需求	开发	测试	发布	运维

获取度量元数据

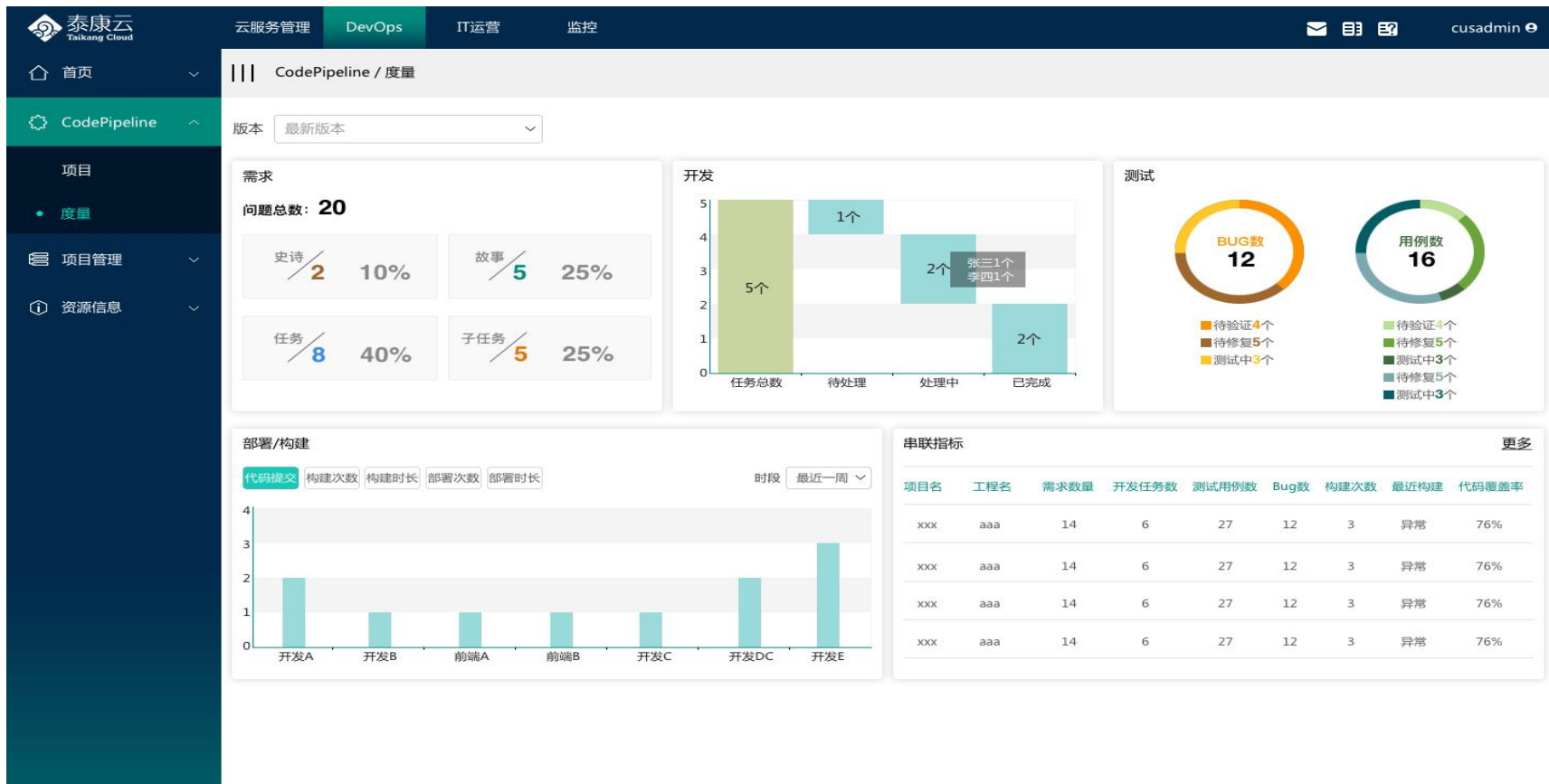
老流程



新流程



研发质量可视化



目录

1 DevOps建设背景

2 研发痛点及措施

 **3** DevOps实施收益

4 TDS平台最佳实践

一套敏捷作业框架

1.目的	4
2.适用范围	4
3.词汇表	4
4.角色权责	6
3.1 产品负责人 PO	6
3.2 Scrum Master	7
Scrum Master 服务于产品负责人	7
Scrum Master 服务于开发团队	7
Scrum Master 服务于组织	8
3.2 开发团队 Scrum Team	8
开发团队有以下几个特点:	8
5.主要流程规范	9
Scrum 敏捷开发的流程及团队架构	9
需求评审会	10
迭代计划会议	10
每日站会	10
迭代评审会议	10
迭代回顾会议	11
迭代协同会	11
完成定义	11
6.关键点检视	12
7.参考文献	12

敏捷作业管理框架目录

序号	事项	阶段	时间
1	需求沟通会	项目启动会前	2018/3/21
2	DevOps项目初期调研	项目启动会前	2018/4/3
3	准备开发环境及部署环境	项目启动会前	2018/4/4
4	泰康在线DevOps项目启动会	项目启动会前	2018/4/11
5	DevOps开班培训	项目启动会前	2018/4/11
6	项目实施前调研准备	项目启动会前	2018/4/13
7	对试点项目进行调研	项目启动会前	2018/4/18
8	JIRA敏捷项目培训	项目启动会前	2018/4/18
9	GitLab开发基础培训	项目启动会前	2018/4/18
10	DevOps项目培训	项目启动会前	2018/4/20
11	敏捷开发培训	项目启动会前	2018/4/24
12	需求故事冰山模型培训	项目启动会前	2018/4/26
13	第一试点车险理赔准备方案汇报	项目启动会前	2018/5/2
14	第一试点车险理赔准备迭代计划和用户故事	项目启动会前	2018/5/11
15	第一试点启动首次迭代	项目实施阶段	2018/5/14
16	第二试点核心系统沟通访谈	项目实施阶段	2018/5/17
17	第二试点启动首次迭代	项目实施阶段	2018/5/18
18	第三试点互联网开发团队沟通访谈	项目实施阶段	2018/5/21
19	第三试点互联网开发团队敏捷培训	项目实施阶段	2018/5/22
20	第三试点互联网开发团队JIRA和GitLab培训	项目实施阶段	2018/5/23
21	第三试点互联网开发团队启动会	项目实施阶段	2018/5/24
22	泰康在线DevOps项目中期汇报	项目实施阶段	2018/5/25
23	三个试点持续进行敏捷开发工作	项目实施阶段	2018/6/2
24	敏捷框架	项目收尾阶段	2018/7/20
25	DevOps项目试点团队总结会	项目收尾阶段	2018/7/31
26	DevOps项目总结报告正式	项目收尾阶段	2018/7/31
27	泰康在线DevOps项目总结汇报	项目收尾阶段	2018/7/31

组织过程资产目录及内容

- 01-项目启动会前资料
- 02-项目启动会后实施前
- 03-项目实施阶段资料
- 04-敏捷框架
- 05-DevOps项目试点团队经验总结

组织过程资产内容汇总

- 启动会前会议沟通纪要
- 项目启动会计划
- 项目启动前实施计划告知

01-项目启动会前资料

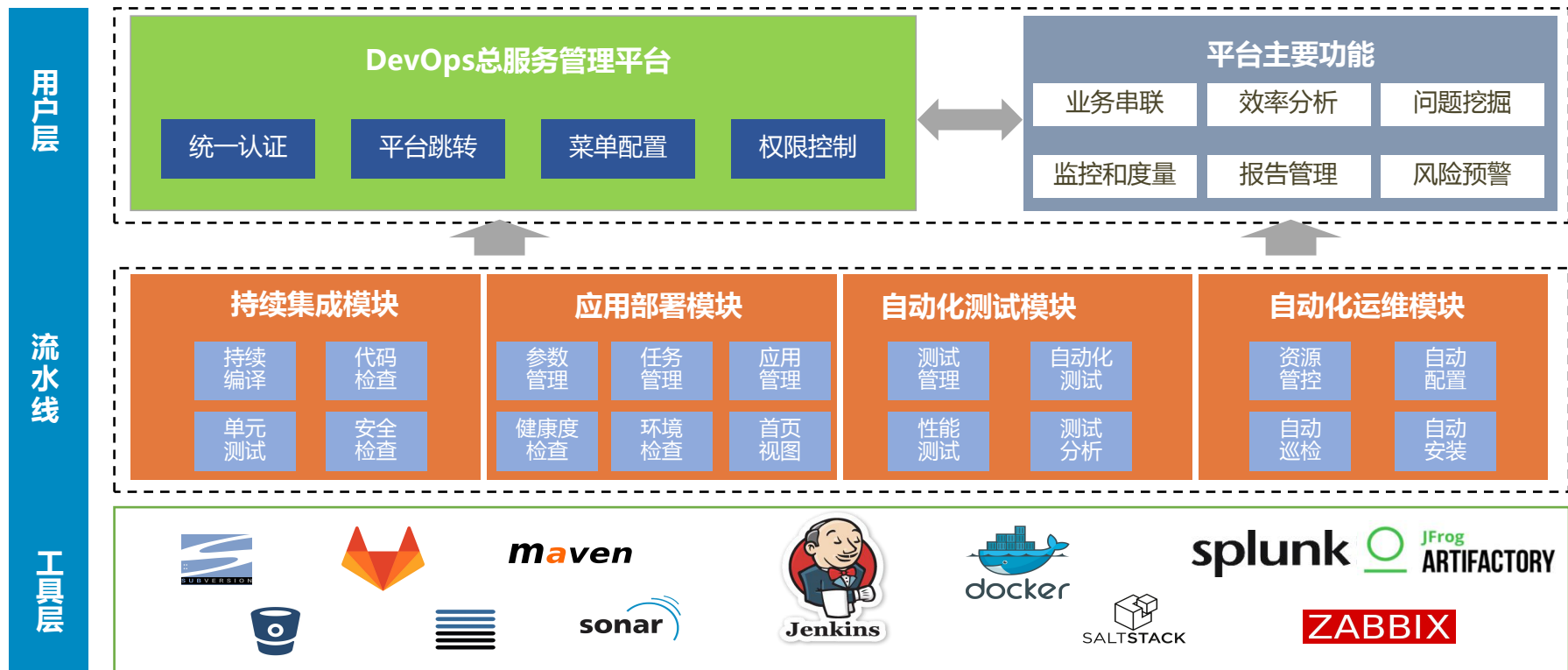
- 实施前培训资料
- 实施前沟通调研资料
- 会议纪要
- DevOps项目实施计划

02-项目启动会后实施前

- 01-第一试点 试点资料
- 02-第二试点 迭代资料
- 03-第三试点 团队迭代资料
- 04-汇报
- 05-培训资料
- 06-泰康集团看板使用模板
- 07-访谈
- 08-周报

03-项目实施阶段资料

一套工具链和管控平台



效能收益

1 协同层面

✓ 统一的流程和工具集成平台

降低需求、研发，测试，运维人员协同成本。

✓ 统一的跨团队信息共享

增强主动获取信息的沟通效率。

✓ 统一的反馈机制

自动化反馈机制，迅速定位问题，准确有效。

✓ 统一的度量数据来源

工具反馈准确的数值，进行精益管理与分析。

2 工具层面

✓ 软件构建

预期能力达到每天1-2次。

✓ 应用部署

持续交付后，随时可进行发布镜像部署到非生产环境。

✓ 代码分支合并能力

人工合并转为CI系统统一合并，减少人工干预。

3 质量层面

✓ 后续接入分层自动化测试

每次发布自动触发自动化测试，减少了人工干预，提高测试频率。

✓ 自动触发的分层质量扫描

对开发代码标准问题早发现反馈，减少部署阶段的质量风险。

✓ 流水线仪表盘发现问题

通过流水线与仪表盘的可视化展现，发现当前存在的问题，进行精益改进。

经验总结

- 两条腿走路，切忌一刀切
- 授人以鱼不如授人以渔
- 团结一切可团结的力量
- 持续推进，忌三天打渔两天晒网
- 自动化一切可自动化的环节

目录

1 DevOps建设背景

2 研发痛点及措施

3 DevOps实施收益

 **4** TDS平台最佳实践

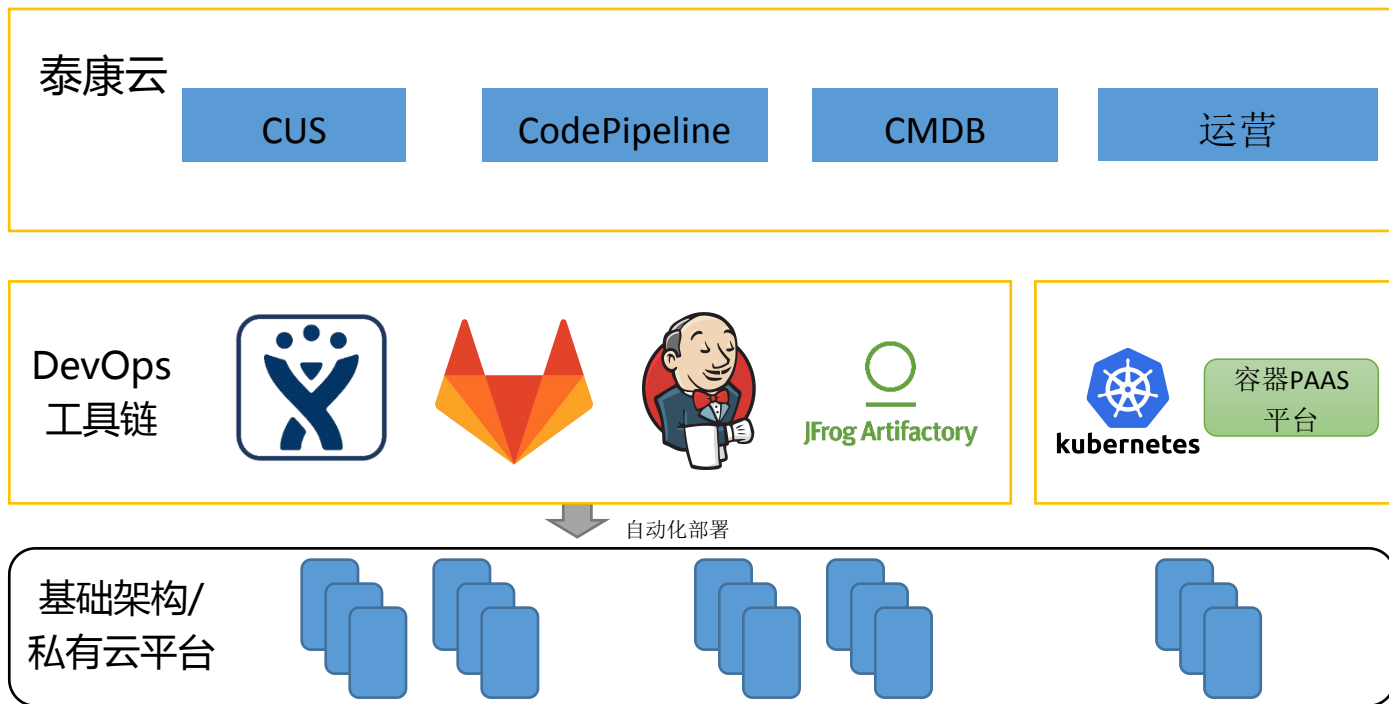
TDS (Taikang DevOps Service) 平台



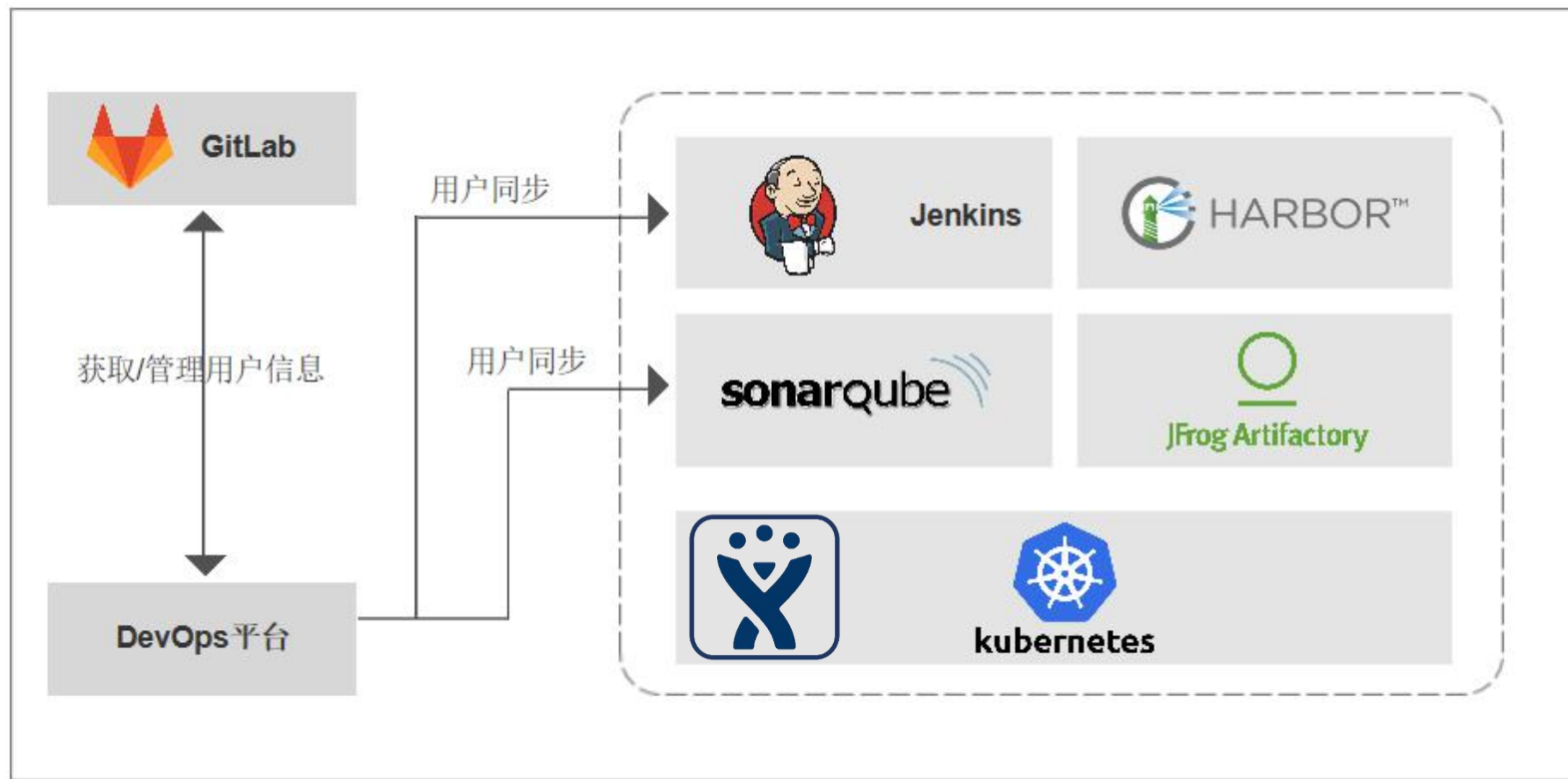
了解泰康云产品

 云服务管理 资源申请和管理 快速创建资源, 查看、授权、堡垒机登陆 业务与应用管理 业务与应用管理	 DevOps平台 SVN管理 SVN代码管理 应用发布 使用jenkins进行自动化应用发布	 监控与分析 监控平台 监控平台 分析平台 分析平台	 数字化运营 监控覆盖度 监控覆盖度 TED 自动化运维工具
---	--	---	---

平台架构



统一用户管理



CodePipeline命名规则

业务-应用-类型_环境

示例：

tkcloud-cus-service

tkcloud-cus-web

* 选择业务

泰康云运维服务管理平台



* 选择应用

tcmp-monitor-service(监控后端)



没有找到应用?

维护

* 选择环境

☒ 开发环境(development)DEV



测试环境(User Acceptance Testing)UAT



生产环境(production)PRD



模拟环境(Staging Server)STAG

项目(pipeline)名称:

tcmp-monitor-service_DEV

构建一个JAVA的软件项目

基于Maven/Ant来构建你的java体系风格的项目

平台易用性：演示项目/帮助

为降低学习成本提供了Demo项目，所有人可见

[项目地址](#)



CodePipeline / 项目 / 工程列表 / 工程详情

工程名称: demo-springboot-service_TEST 最新构建(#9), 8day5小时前

Jenkins Gitlab Sonar java maven SpringBoot

基础信息

- 项目名称: DevOps演示项目
- 业务名称: demo
- 应用名称: demo-springboot-service

构建部署

- 端口: 9999
- 执行用户:
- 部署路径:
- 目标主机:

最近构建

- #9 2019-02-19 09:15:40
- #8 2019-02-19 09:13:10
- #7 2019-02-19 09:11:18
- #6 2019-02-19 09:09:50
- #5 2019-02-19 09:08:38

CodePipeline / 项目 / 工程列表 / 工程详情

工程名称: demo-springboot-service_TEST

序号: #9 构建时间: 57秒

构建结果: Success 分支: r

构建人: itw_mengtao 主机: r

构建成功!

Start GetSlave GetCode Build Deploy end

Deploy

```
[demo-springboot-service_TEST] Running shell script
+ '[' -d /srv/salt/demo-springboot-service_TEST ']'
[demo-springboot-service_TEST] Running shell script
+ curl http://gitlab.it.taikang.com/devops/jenkins2.0-
scripts/raw/master/Springboot-nohup.sh -o Springboot.sh -s
[demo-springboot-service_TEST] Running shell script
+ curl http://gitlab.it.taikang.com/devops/jenkins2.0-
scripts/raw/master/nohup.powerboot.sh -o nohup.powerboot.sh -s
[demo-springboot-service_TEST] Running shell script
+ sed -i s#activeEnv=activeEnv=test#g /srv/salt/demo-springboot-
service_TEST/Springboot.sh
+ sed -i s#port=null#port=9999#g /srv/salt/demo-springboot-
service_TEST/Springboot.sh
[demo-springboot-service_TEST] Running shell script
+ sed -i s#javaOption=#javaOption='\' -Xms=512m -Xmx=512m\'\'#g' /srv/salt/demo-
springboot-service_TEST/Springboot.sh
[demo-springboot-service_TEST] Running shell script
```

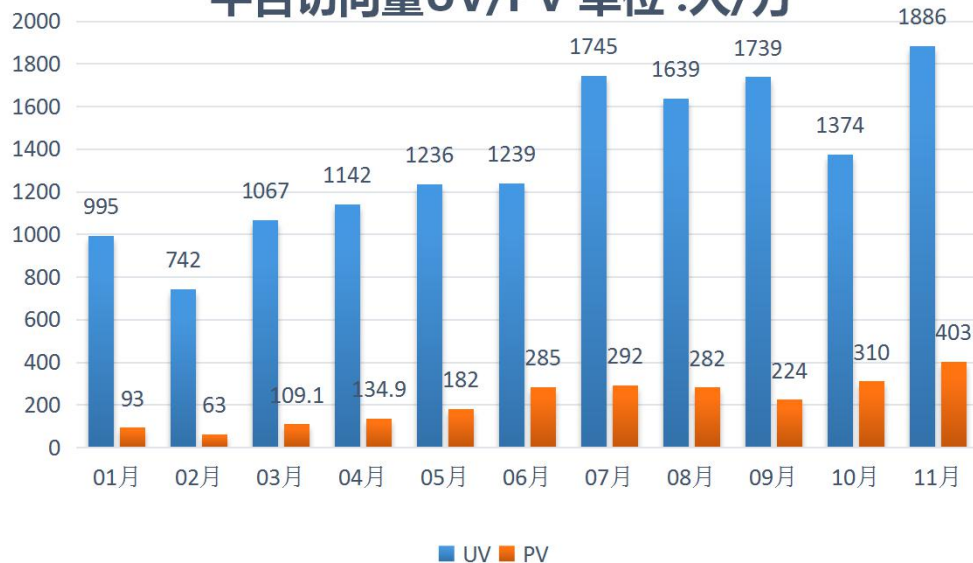
[帮助文档](#)

DevOps工具链平台



- 服务(应用)数量 500+
- DevOps Pipeline数量 2000+
- 快速增长中
- ...

平台访问量UV/PV 单位 :人/万



用户数 : 2000+

访问量PV : 450万+



Thanks

高效运维社区
开放运维联盟

荣誉出品