



# 非互联网公司落地 DevOps 的方法和实践

---

快速发布 谁与争锋

**王青** (11年研发老兵, 曾就职IBM、HPE、爱奇艺)

JFrog 中国首席架构师

[wq@jfrogchina.com](mailto:wq@jfrogchina.com)

- 主导了JFrog 国内大型客户的包管理平台落地。
- 主导 HPE Cloud 云平台的架构和研发
- 主导 Rational软件生命周期管理平台的开发
- InfoQ 特约社区编辑, 文章多次发表在 InfoQ, DBAPlus, 高效运维, Docker 社区等技术社区



- 非互联网企业 DevOps 的案例
- DevOps落地方法
- Q&A

- 非互联网企业 DevOps 的案例
- DevOps落地方法
- Q&A

# JFrog 4000+全球用户



互联网 & 软件



+4,000  
Customers

科技 & 电信



银行 & 金融



+100  
New Logos  
a month

工程 & 航空



零售 & 消费



95%  
Retention  
Rate

教育 & 研究



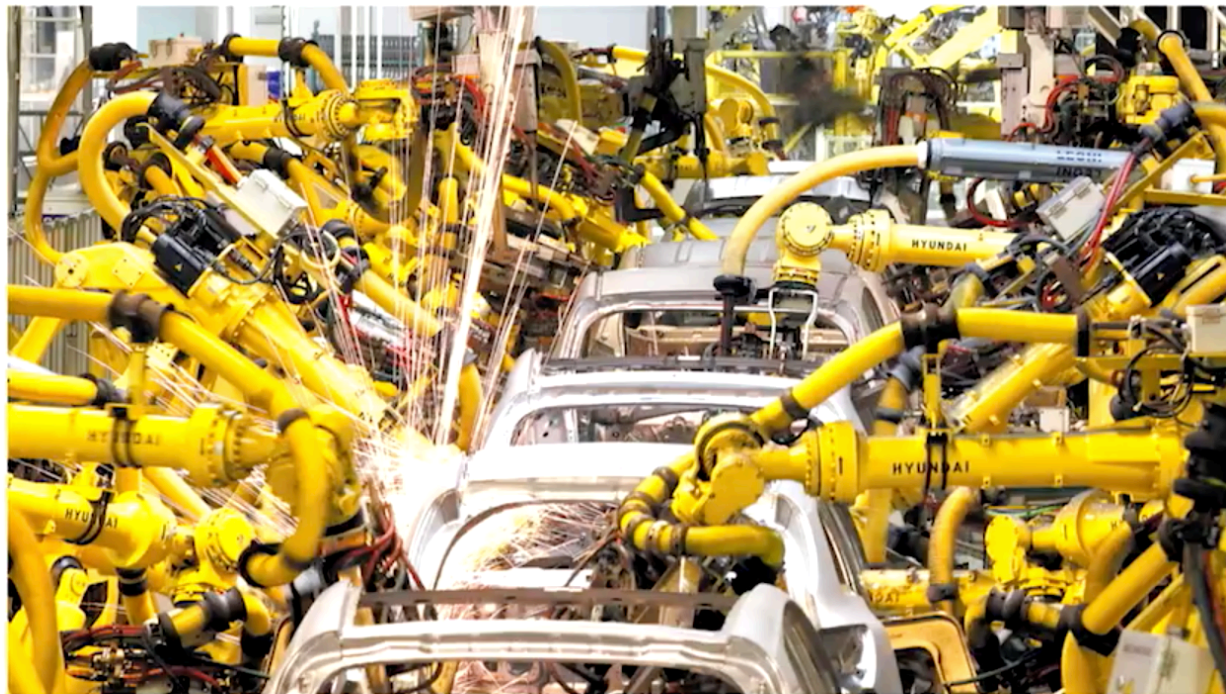
- 全球 52,000 员工
- 2016年净利润 – 4,976Million EUR
- 600研发团队



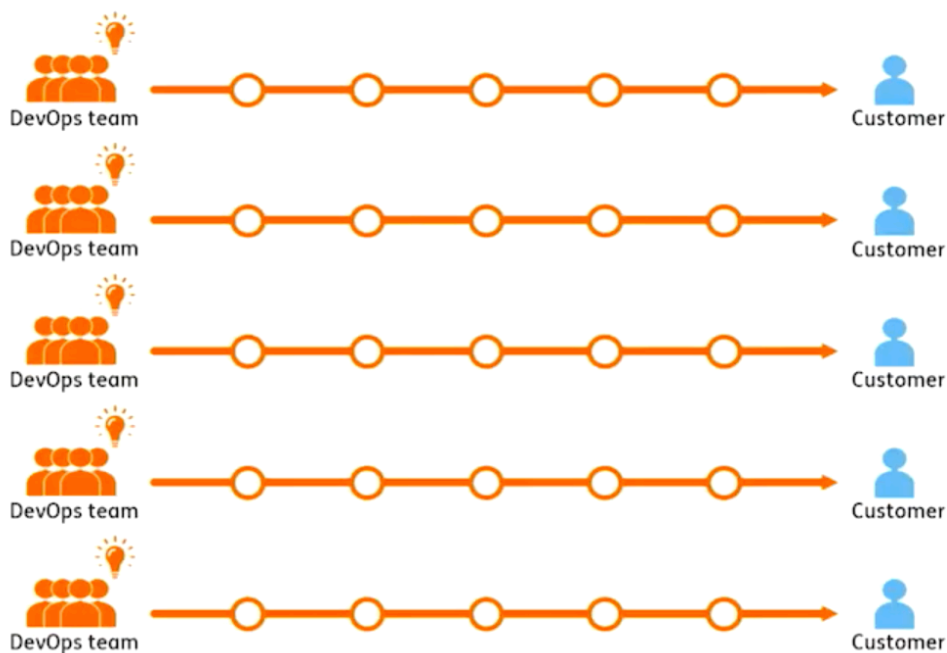
## How we were building software



## How we wanted to build software

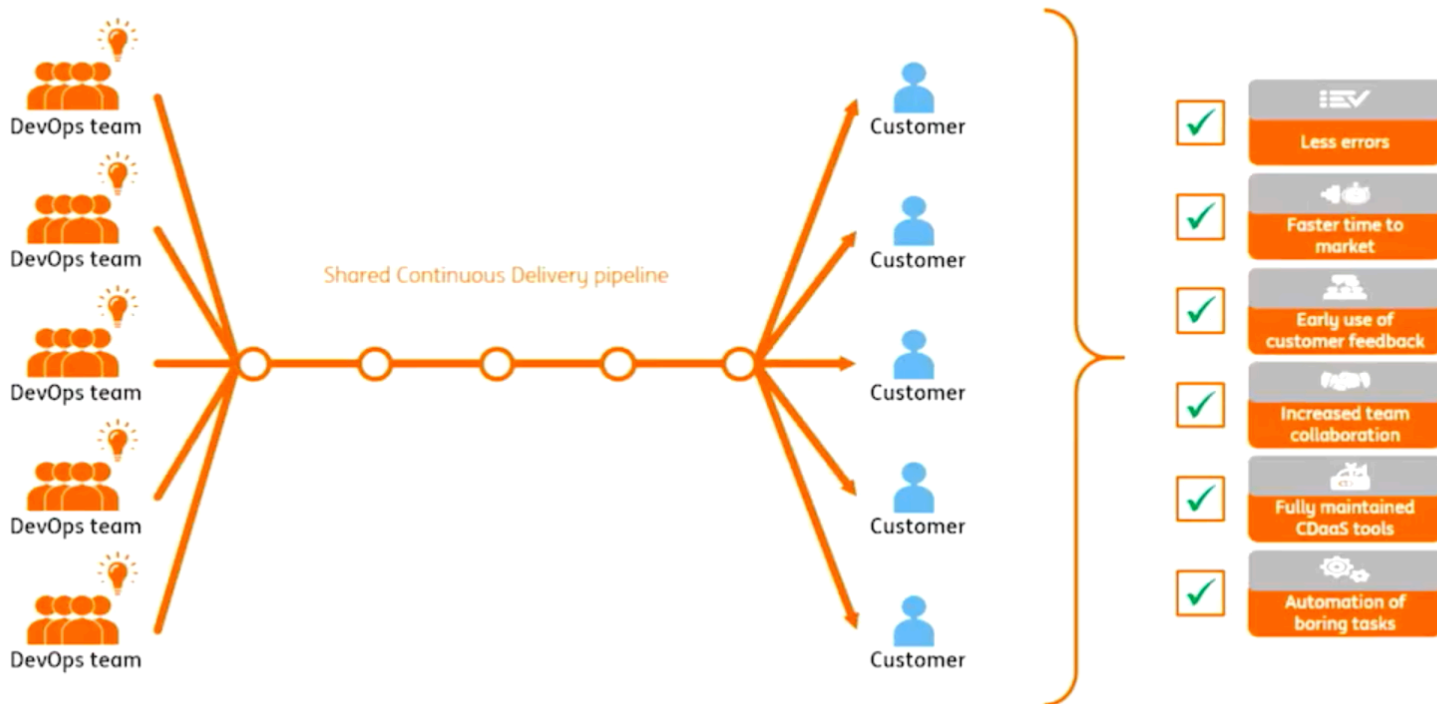


## Team specific processes often come with downsides

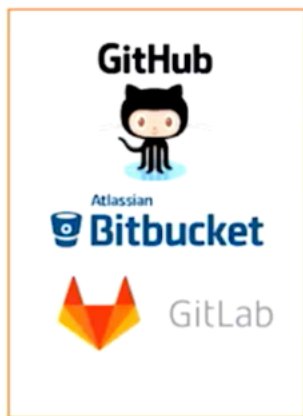


- ✘ Duplication of tasks and operation
- ✘ Little/no knowledge sharing and learning
- ✘ Often not fully automated
- ✘ Maintenance of own tools takes time
- ✘ Less agile
- ✘ Focus on processes instead of value creation

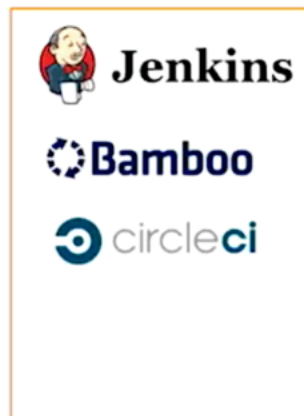
## By using a shared Continuous Delivery pipeline, ING gains significant benefits



## Build - Continuous Integration



Code



Orchestrate



Build

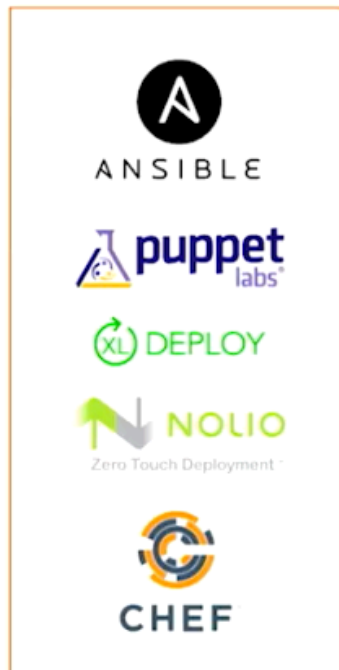


Check



Store

## Deploy - Continuous Deployments



## Artifactory

### Numbers

#artifacts	3.500.000
#binaries	2.500.000
#local repos	1800
#remote repos	39
#users	4000
#groups	2600
#permissions	900
#storage	11.5 TB
#growth	1TB / month

### Artifact types



## Automating Artifacts for CDaaS



- Team
- Artifact
- System
- Build
- Deploy



JFrog Artifactory



Maven™



## Tooling

- CDaaS portal
- LDAP
- Vault
- Script buildserver
- Deploy trigger



## One Portal to rule them all

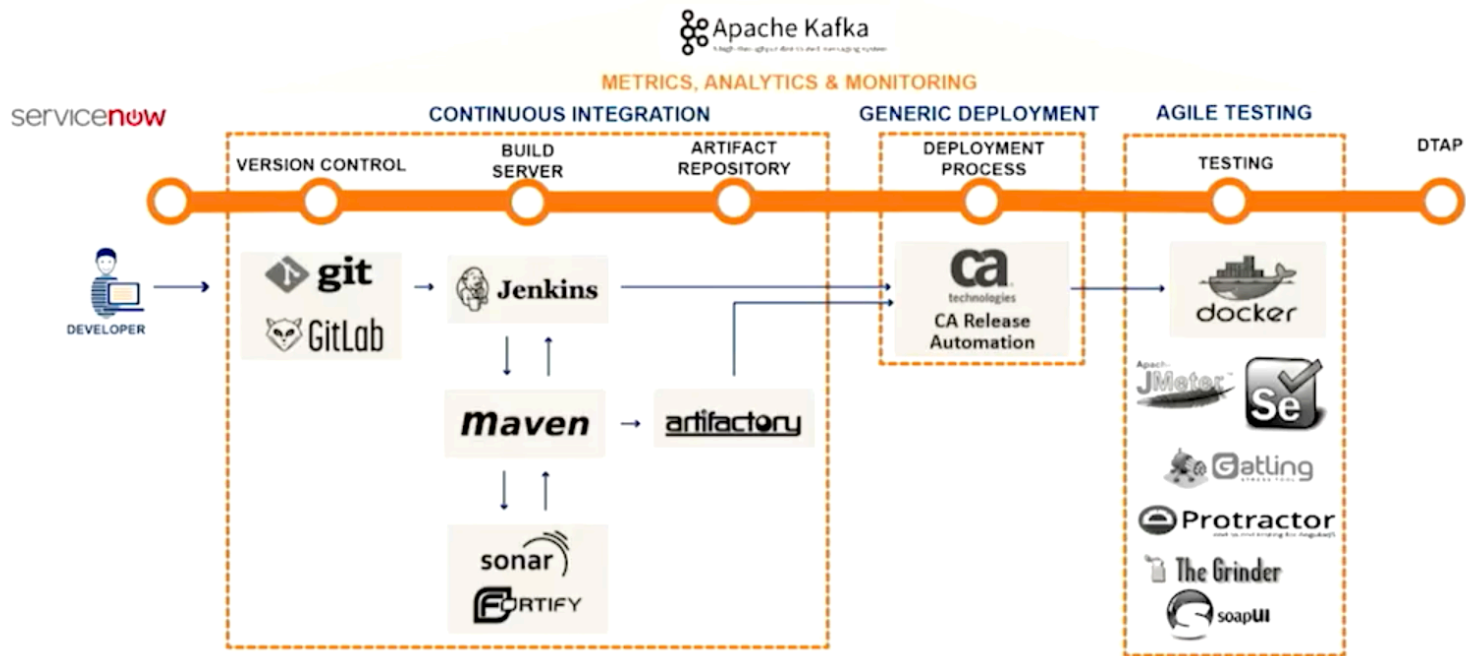


Search

### CD Portal

Continuous Delivery as a Service





One opinionated flow for all the (~600) teams

Going to production takes minutes/hours

Considerable reliability improvements

Relaxed atmosphere on the floor

- 支持了500种应用交付
- TTM 小于6周
- 减少50%的线上发布事故
- 发布次数12,000次/月



**ORACLE®**  
Docker的弹性伸缩



**Google**  
云原生容器 CI-CD for Google Kubernetes with  
Artifactory



**Adrian Cockcroft**  
DevOps的未来



**CISCO**  
思科如何在一年内实现0-5百万 artifacts的增长



**Raytheon Ca**  
**ORACLE® EMC<sup>2</sup>**  
JFrog企业客户讨论解决最紧迫的问题



**CARS.COM**  
使用Artifactory 和  
Docker完成不可变的基础设施



**Bintray**  
最佳实践  
Docker持续分发



**salesforce**  
Bintray如何挽救了我的婚姻：  
用Bintray 拓展web jars包

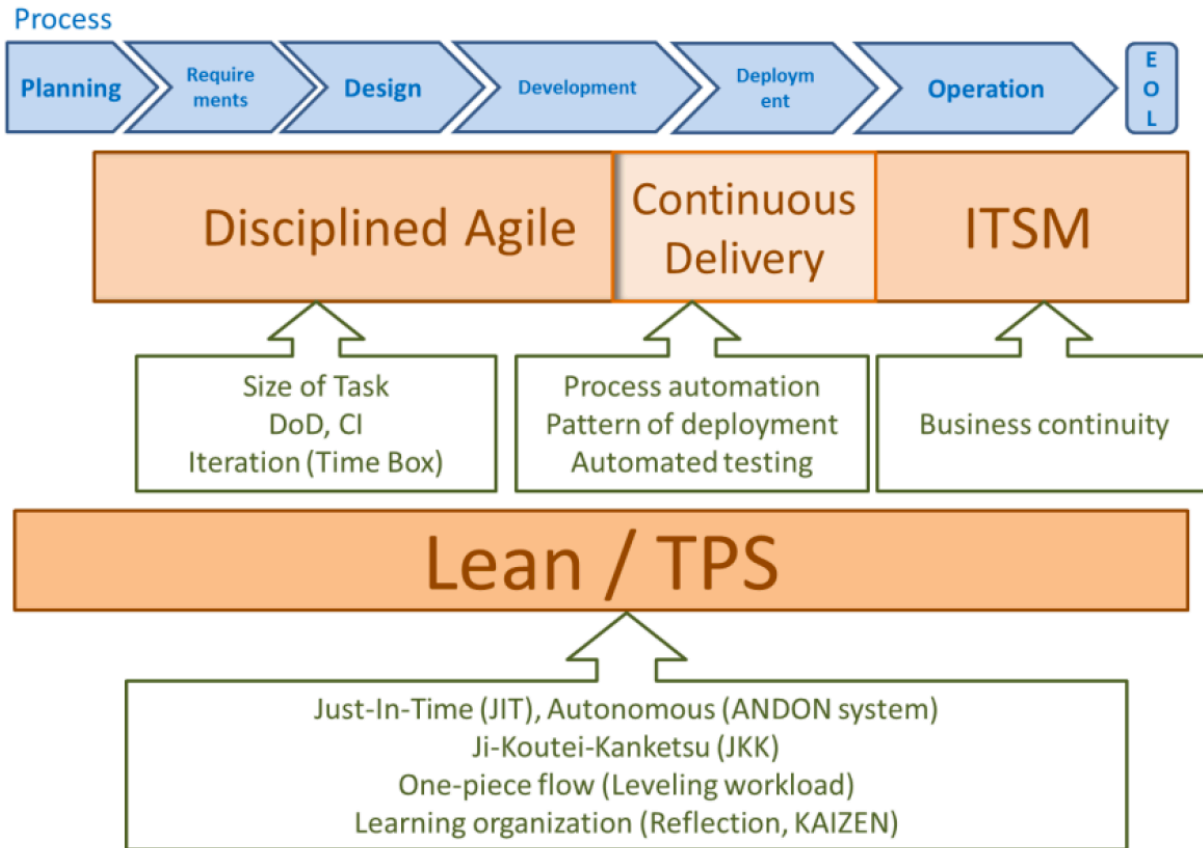


**ca**  
JFrog Artifactory如何作为一个全球性的CI 流程的核心

- 非互联网企业 DevOps 的案例
- DevOps落地方法
- Q&A



# 企业级DevOps 知识体系



# 企业级DevOps 知识体系



持续安全

## 安全需求设计:

- TFS / ALM Octane

## 安全静态扫描分析

- Sonarcube & Fortify

## 漏洞扫描分析

- JFrog Xray & BlackDuck & WebInspect & AppScan

## 漏洞扫描分析

- AppDefender & JFrog Xray



项目经理



产品经理



业务部门



架构师



测试团队

## 持续计划&定义

- 需求管理
- 项目管理
- 发布管理
- 资源管理
- 消耗管理
- 缺陷管理
- 敏捷过程管理
- 架构设计
- 安全设计
- 测试计划

## Software Tools:

- Jira / 禅道
- Confluence
- Viso
- Axure
- ALM Octane / TFS
- Jira Zephyr / TestLink



开发

测试

运维

## 持续集成

- 前后端开发
- 需求和代码关联
- 缺陷和代码关联
- 代码单元测试覆盖率
- 代码静态扫描
- 构建/打包
- 自动化测试
- 代码评审
- 容器镜像构建
- 制品库管理
- 质量元数据收集

## Software Tools:

- IntelliJ, Eclipse, VS
- Gitlab / SVN
- SonarCube Fortify
- Jenkins
- Artifactory
- Gerrit
- Junit / TestNG

## 持续发布

- 虚拟机发布
- 容器发布
- CMDB
- 自动化发布工具

## Software Tools:

- OpenStack
- Docker / Kubernetes / Openshift
- Device42
- Jenkins
- Ansible / Chef / Puppet



Dev Env



FT



SIT



UAT



Prod

## 持续测试

- 功能性测试
- 安全漏洞测试
- 开源组件漏洞测试
- 接口测试
- 集成测试
- 性能测试
- UI 自动化测试

## Software Tools:

- UFT
- AppScan
- JFrog Xray
- Newman
- Jmeter / LoadRunner
- Selenium / Apprium
- TestLink

## 持续运维

- 自动扩容/缩容
- 故障自愈
- 监控
- 服务治理
- 蓝绿发布
- 日志分析
- 告警

## Software Tools:

- Kubernetes / Openshift
- Zabbix
- Spring Cloud
- ELK
- Spinnaker
- Prometheus
- Grafana
- Istio

持续评估



发布速度 KPI



发布质量 KPI



研发生产力 KPI



安全 KPI

## Software Tools

- Grafana / Hygieia / Artifactory metadata

持续协作



项目管理协作工具



代码提交机器人



自动化部署机器人



故障告警机器人

## Software Tools

- DialogFlow / Slack / HipChat



# 明确DevOps 平台建设的需求和目标

# DevOps 平台建设的目标 ( 示例 )

一年计划

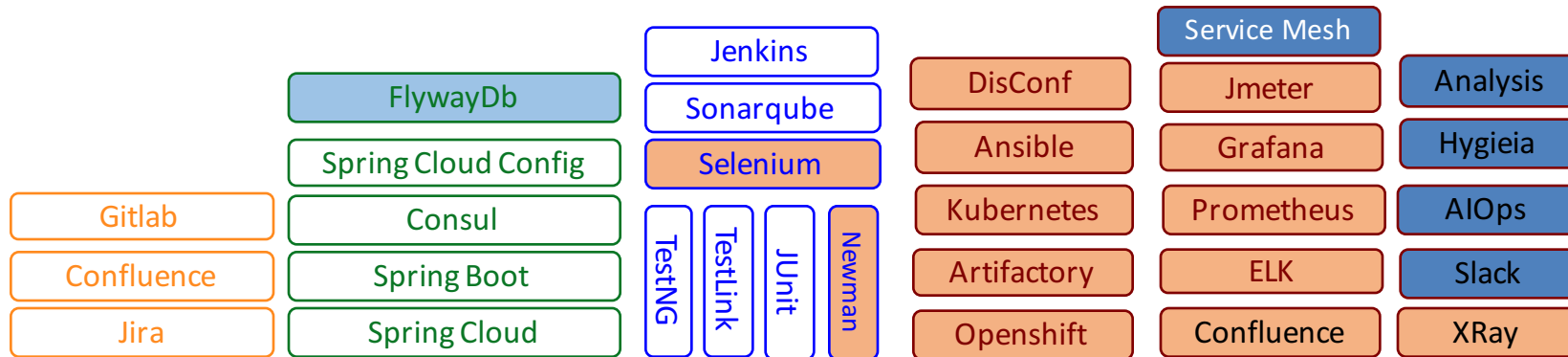


两年计划

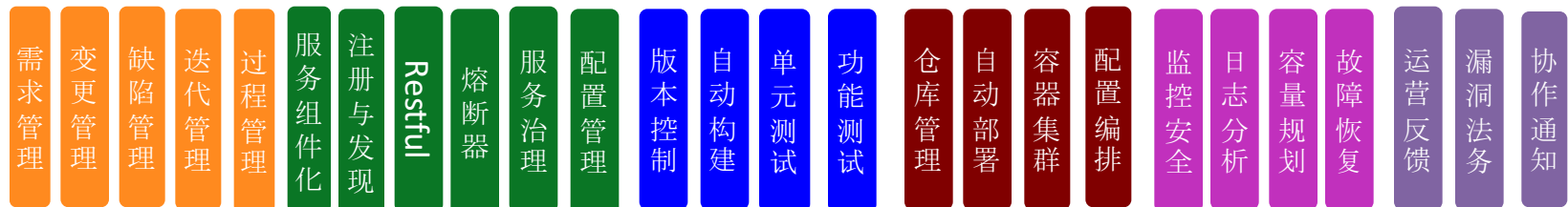


## DevOps 统一门户 & 可视化

工具



方法技术



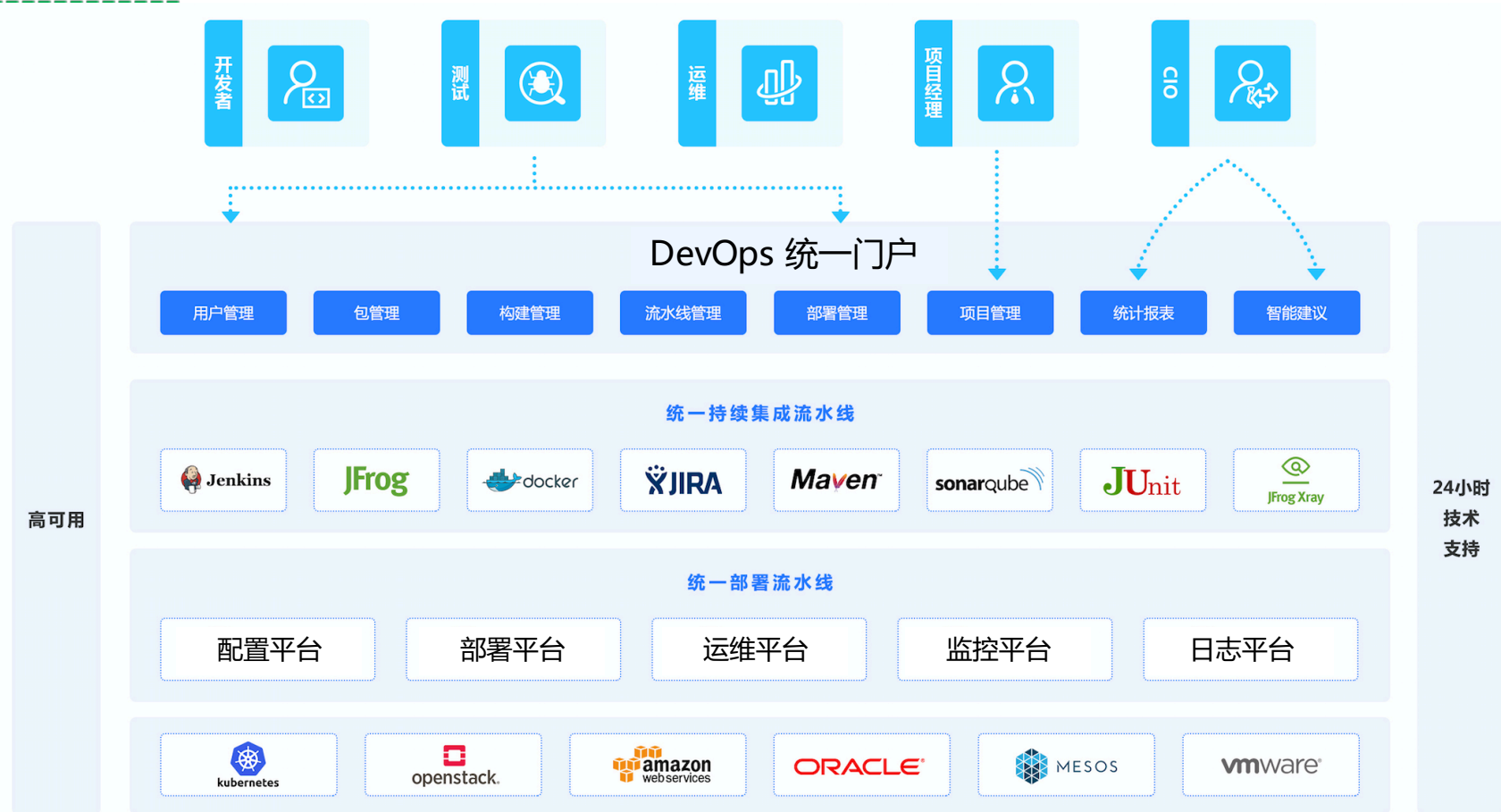
策略



精益思想

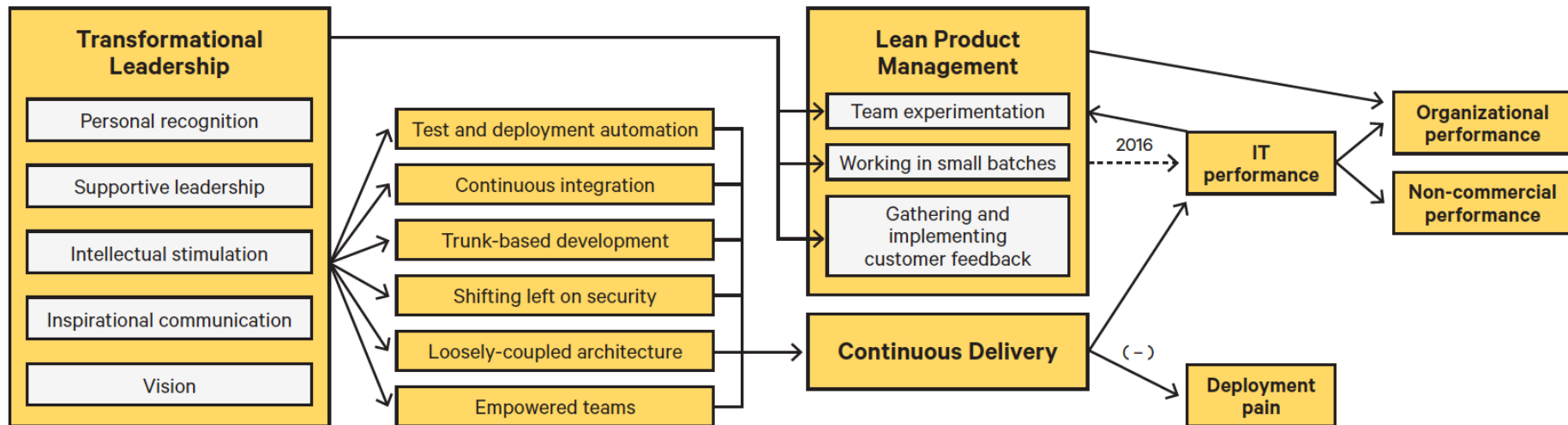


# DevOps 平台建设的方法



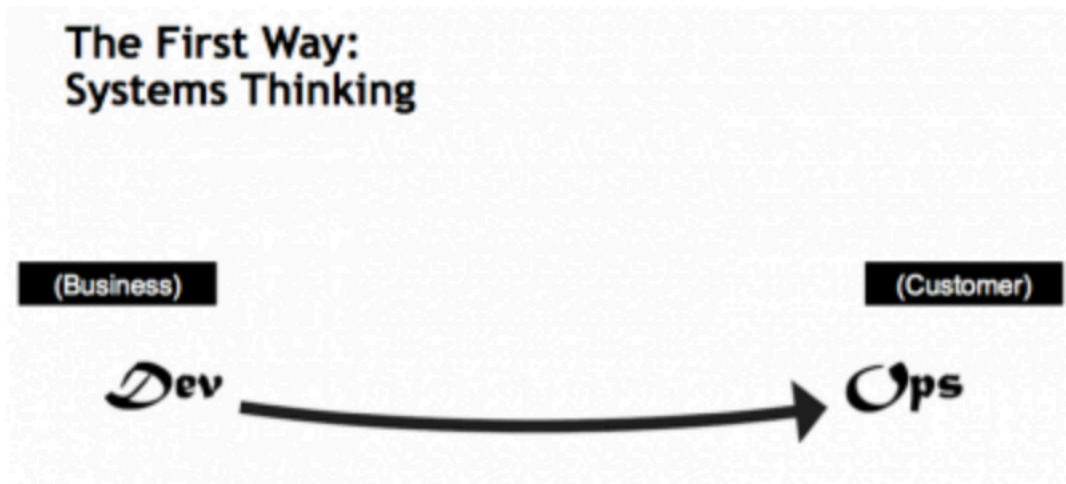
有了目标，如何开始落地？





## • Structured equation model (SEM)

- 领导力
- 精益管理
- 持续交付



From Gene Kim - <<凤凰项目>> <<DevOps Handbook>>作者  
<http://itrevolution.com/the-three-ways-principles-underpinning-devops/>

## Kanban board

QUICK FILTERS: Only My Issues Recently Updated

3 Backlog      4 Selected for Development      1 In Progress Max 1      2 Done      Release...

Expedite 2 issues

- SKP-2: Kanban boards are often divided into streams of
- SKP-1: Kanban cards represent work items >> Click the

Everything Else 8 issues

- SKP-3: Add work items with "+ Create Issue" at the top
- SKP-4: Work items are ranked in priority order (from top to
- SKP-10: Instructions for deleting this sample board and
- SKP-6: Work In Progress (WIP) limits highlight delays.
- SKP-7: ... so 2 work items violate the limit and cause the
- SKP-9: As teams develop with Kanban they get better at
- SKP-5: Work items flow through different stages from left to
- SKP-8: Filters at the top of the board allow you to quickly cut

Agile Estimates for JIRA panel

estimated values on Kanban Board

### Sample Kanban Project / SKP-1

Kanban cards represent work items >> Click the "SKP-1" link at the top of this card to show the Detail view - there are more on Kanban in the 'Description' section

Estimate: 5

Your Vote: No vote

10s 60s Show votes Restart

#### Details

Status: SELECTED FOR DEVE...

Component/s: None

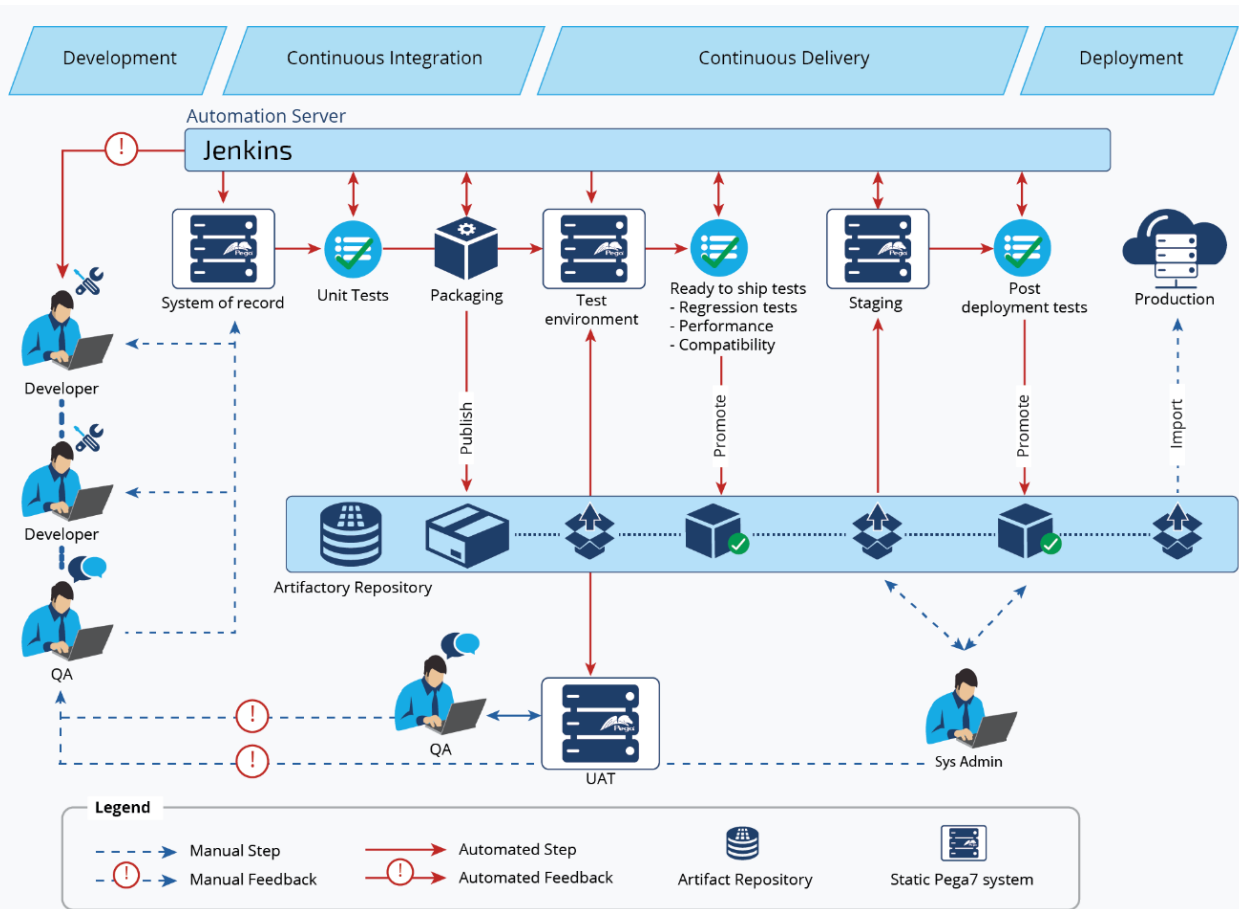
Labels: None

Affects Version/s: None

Fix Version/s: None



# 单团队持续交付模型



# 持续交付流水线



## Stage View

		Checkout MAVEN	Build Maven	Checkout Docker	Resolve	Build and Deploy	testing app	Promotions	distribute	deployment
Average stage times:		4s	11s	3s	453ms	10s	20s	10s	541ms	1min 8s
#33	Jan 30 20:46 3 commits	5s	28s	6s	662ms	30s	23s	34s	609ms	1min 9s
#32	Jan 16 20:40 1 commits	5s	8s	3s	376ms	7s	19s	6s	542ms	1min 12s
#31	Jan 16 17:15 No Changes	5s	10s	3s	529ms	15s	23s	10s	533ms	2min 45s
#30	Jan 09 20:45 1 commits	4s	17s	3s	407ms	11s failed				
#29	Jan 04 15:25 No Changes	7s	12s	4s	640ms	9s	20s	7s	539ms	1min 16s



# Git + Jira 和 Jenkins 集成

- Jira 创建任务

- Git

–Git commit –m “#JIRA-101 add fix for login form”

- Jira Jenkins Plugin

- ChangeSets in Jenkins



Changes

1. [PLUG-1 TEST 2015-10-01-01:46:17 \(detail\)](#)

Jira ID



Started by

Some issue to be done



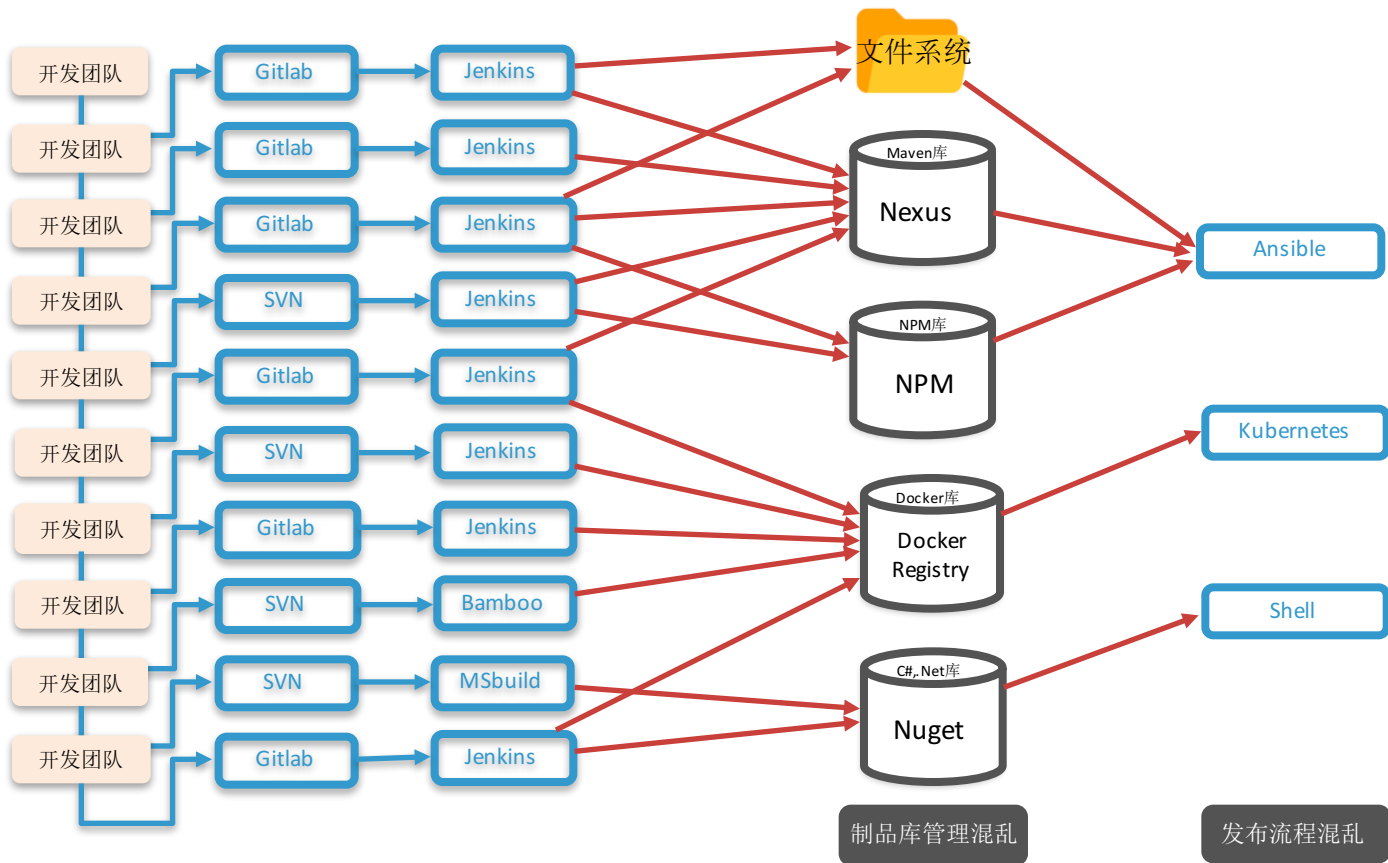
Revision: 5e537067e5c4b6281a4ffaa779a0a326c246603c

- refs/remotes/origin/master

The screenshot shows the JFrog Artifactory Build Browser interface. The left sidebar contains navigation options: Home, Artifacts, Packages, Search, Builds, and Admin. The main content area is titled 'Build Browser' and shows the path 'All Builds > mvn-jira-test > 6'. Below this, it displays 'Build #6' with tabs for 'General Build Info', 'Published Modules', 'Environment', 'Issues', and 'Licenses'. The 'Issues' tab is active, showing '1 Issue'. A table lists the issue with columns for 'Key', 'Summary', and 'Previous Build'. The row contains 'DEV-1', '接口调研', and an empty cell. A red box highlights the 'DEV-1' key, with a red arrow pointing to it from the text 'Jira Issue 链接'.

Key	Summary	Previous Build
DEV-1	接口调研	

# 痛点 - 制品库管理混乱，发布流程混乱



容易宕机

不支持高可用

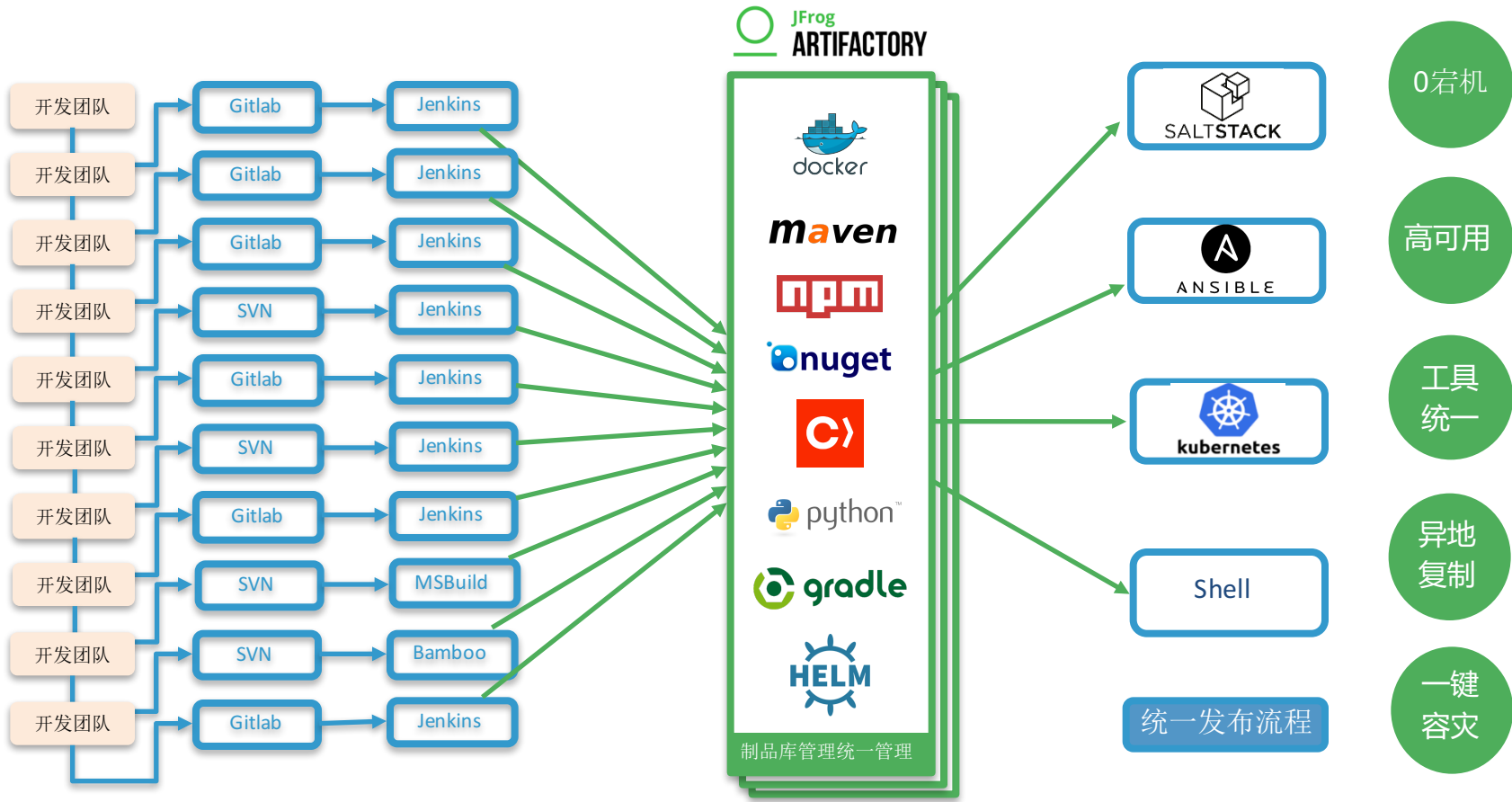
工具碎片化

不支持多地复制

不支持容灾



# 使用 Artifactory 管理所有语言的制品仓库



0 宕机

高可用

工具统一

异地复制

一键容灾



# Jenkins 实现自动化升级软件包版本



源码按需自动打tag, 并自动升级开发版本

Jenkins 3  [admin](#) | [log out](#)

Jenkins > svn-test > Artifactory Release Staging [ENABLE AUTO\\_REFRESH](#)

### Artifactory Pro Release Staging

Last built version

### Module Version Configuration

One version for all modules

Release version

Next development version

Version per module

Use existing module versions

### VCS Configuration

Create VCS tag

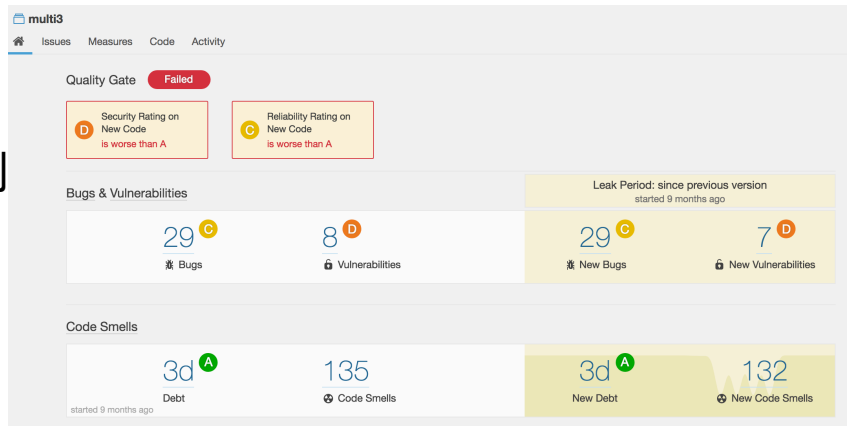
Tag name/URL

Tag comment

Next development version comment



- 使用 TestNG 进行单元测试
  - 研发需要花30%的时间进行测试用例的编写，每个方法有3-4个用例覆盖
  - 每次 Ci 构建必须跑 UT
  - UT 失败即停止流水线
- 使用 Sonar 进行代码扫描
  - 定义质量关卡，Bug 级别达到 A 级别
  - 代码重复拷贝达到 A 级别
  - 测试覆盖率达到 C 级别以上

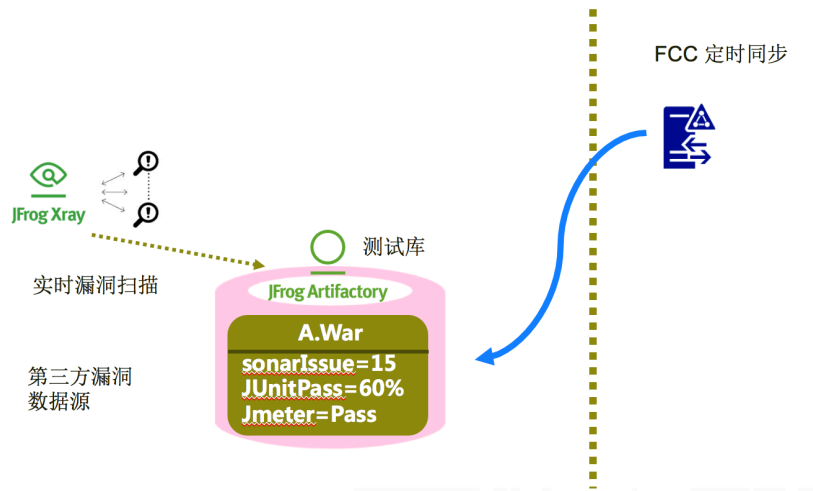


- 使用 Selenium/Appnium 实现 Ui 自动化测试
  - 测试人员负责编写 Selenium 的 case , 比如模拟登陆操作
  - 页面可交互的元素需要定义统一 Id命名规范 , 供 Selenium xpath调用 ,
  - 保证页面布局变化不影响 Selenium 的 case。
  - 集成 Selenium 的 case 到 Jenkins 流水线

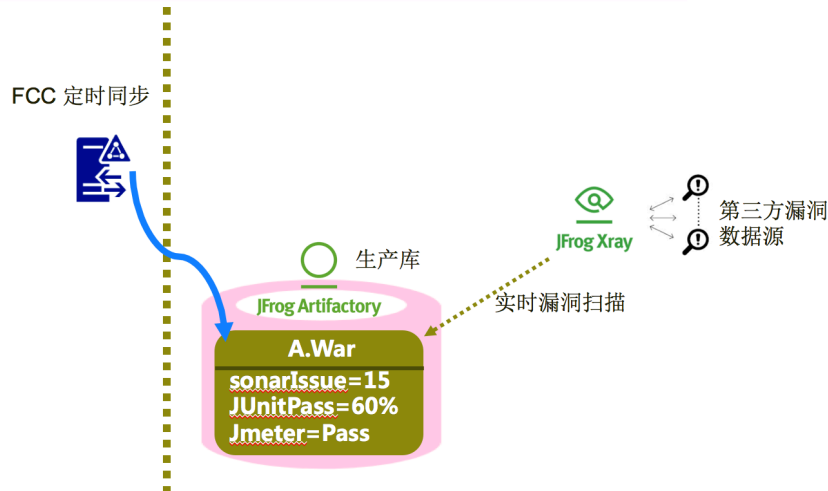
<https://github.com/postmanlabs/newman>



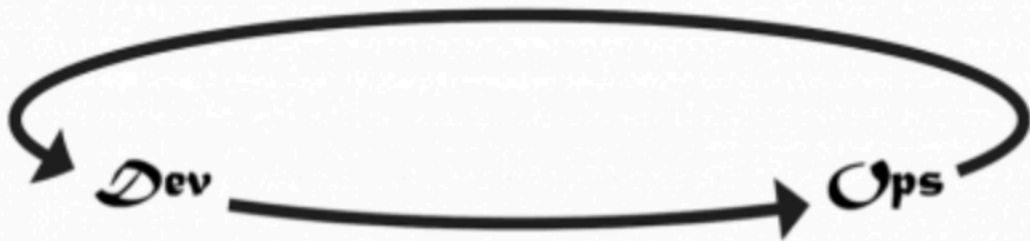
## JFrog Xray 测试库漏洞扫描



## JFrog Xray 生产库漏洞扫描



**The Second Way:  
Amplify Feedback Loops**



# 反馈数据指标



- 第三方开源组件扫描通过率
- 静态代码覆盖率
- 部署频率
- 发布时长
- QA 验证时长
- 部署成功率
- 需求覆盖率
- 功能的使用频率反馈
- 故障修复时间





## Pipeline must have 16 gates

- 全公司统一流水线
- 16个质量关卡
- 每个阶段元数据  
跟包绑定

Source code version control 源代码版本控制

Optimum branching strategy 优化的分支策略

Static analysis 静态分析

> 80% Code coverage 大于80%的覆盖率

Vulnerability scan 漏洞扫描

Open source scan 开源扫描

Artifact version control 制品版本控制

Auto provision 自动化分配资源

Immutable servers 不可变服务器

Integration testing 集成测试

Performance testing 性能测试

Build, Deploy, Testing automated for every commit 每次提交进行构建,部署,自动化测试

Automated Rollback 自动化rollback

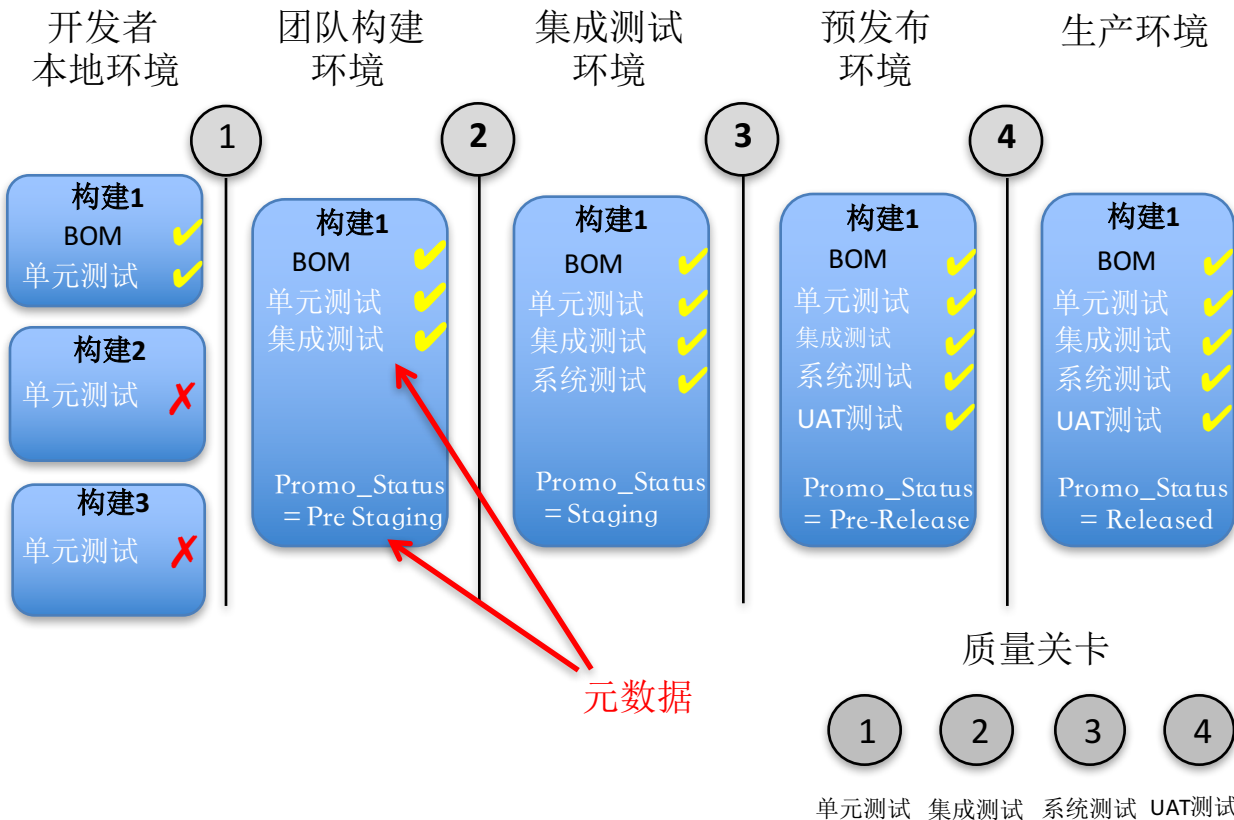
Automated Change Order 自动化变更工单

Zero downtime release 零停机发布 (蓝绿部署、金丝雀发布等)

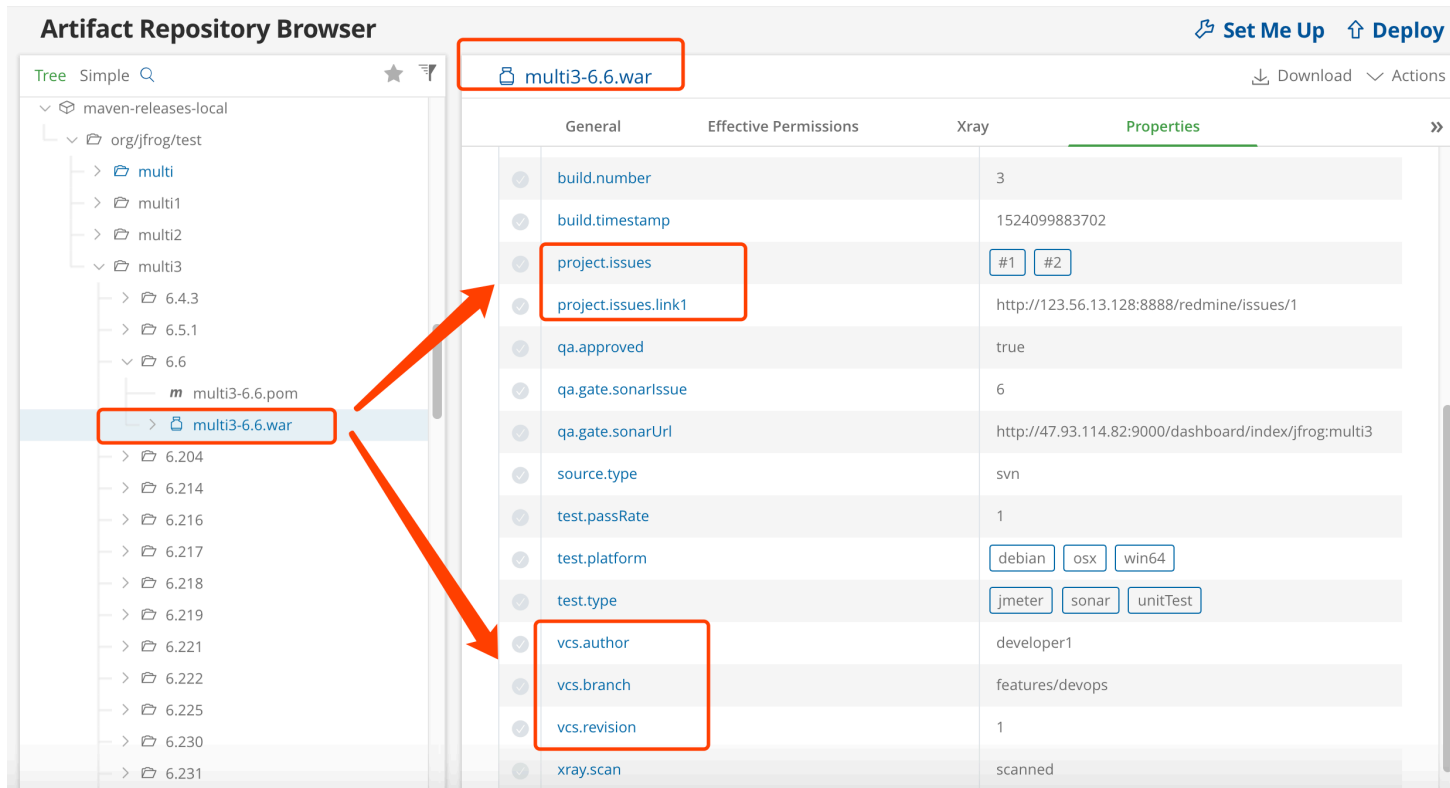
Feature Toggle 功能开关



# 发布包晋级 (Promotion)



## 二进制包作为软件生命周期记录的媒介

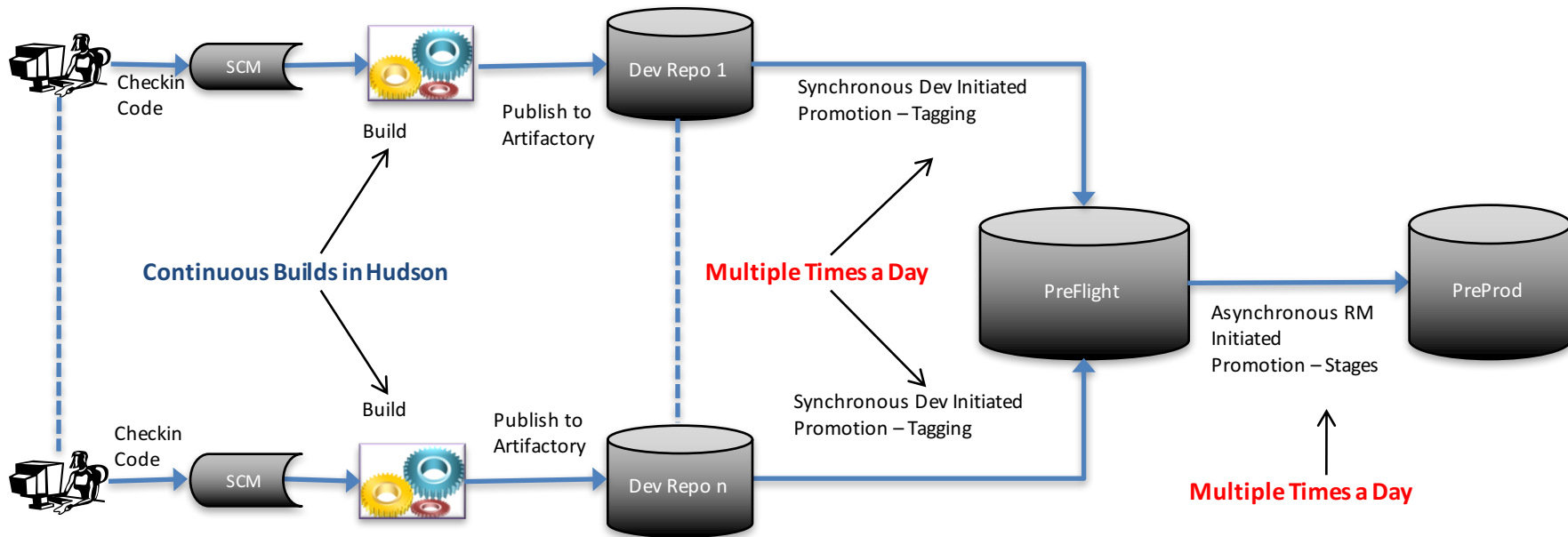


The screenshot displays the 'Artifact Repository Browser' interface. On the left, a tree view shows the artifact path: `maven-releases-local/org/jfrog/test/multi/multi3/multi3-6.6.war`. The main panel shows the artifact details for `multi3-6.6.war`. The 'Properties' tab is active, displaying a table of metadata.

Property	Value
build.number	3
build.timestamp	1524099883702
project.issues	#1 #2
project.issues.link1	http://123.56.13.128:8888/redmine/issues/1
qa.approved	true
qa.gate.sonarIssue	6
qa.gate.sonarUrl	http://47.93.114.82:9000/dashboard/index/jfrog:multi3
source.type	svn
test.passRate	1
test.platform	debian osx win64
test.type	jmeter sonar unitTest
vcs.author	developer1
vcs.branch	features/devops
vcs.revision	1
xray.scan	scanned

Red boxes highlight the artifact name `multi3-6.6.war` in the tree and the `project.issues`, `project.issues.link1`, and `vcs` properties in the table. Red arrows point from the artifact name in the tree to these specific property rows.

# 多团队协作模型

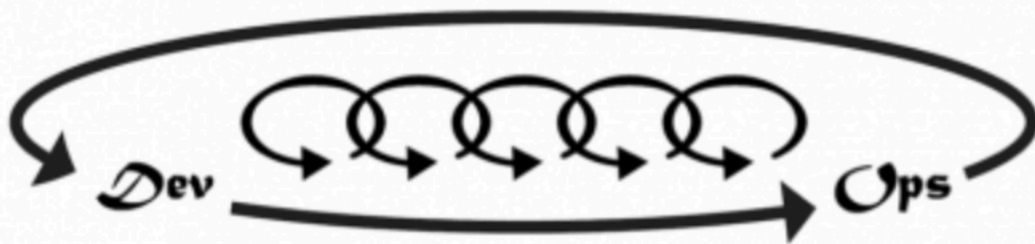


- Artifactory AQL
- Ansible Playbook

```
items.find(  
  {  
    "repo": {"$eq": "maven-releases-local"},  
    "name": {"$match": "multi3-6.*.war"},  
    "@test.type": {"$eq": "unitTest,sonar"},  
    "@test.passRate": {"$eq": "1"},  
    "@quality.gate.sonarIssue": {"$lt": "7"},  
    "@test.status": {"$eq": "done"},  
    "@qa.approved": {"$eq": "true"}  
  }  
) .sort({"$asc" : ["repo","name"]}).limit(1)
```

```
[root@iZ2ze5cznrx04hfon02wn6Z ansible]# cat deploy-war.yml  
- hosts: production  
  vars:  
    - warRemotePath: /root/tools/apache-tomcat-8.5.29/webapps  
  
  tasks:  
    - name: Download WAR to server  
      shell: /root/tools/jfrog rt dl maven-releases-local/org/jfrog/test/multi3/*.war /root/webinar.war --sort-by=created --props test.  
status=done --sort-order=desc --limit=1
```

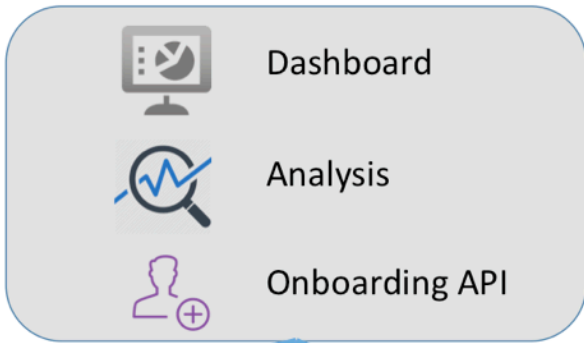
**The Third Way:  
Culture Of Continual Experimentation And  
Learning**



# DevOps 平台 (服务企业内部多团队)

基于开源工具链的DevOps流水线，赋能研发团队

模版化 Jenkins 任务  
统一接入  
跨团队数据分析  
可视化报表  
可扩展容器化管理



JIRA 集成  
GIT 集成



构建

sonarqube JFrog Xray

JUnit Apache JMeter

测试以及  
代码扫描

JFrog Artifactory  
Snapshot库

JFrog Artifactory  
RC/生产库

ANSIBLE CHEF

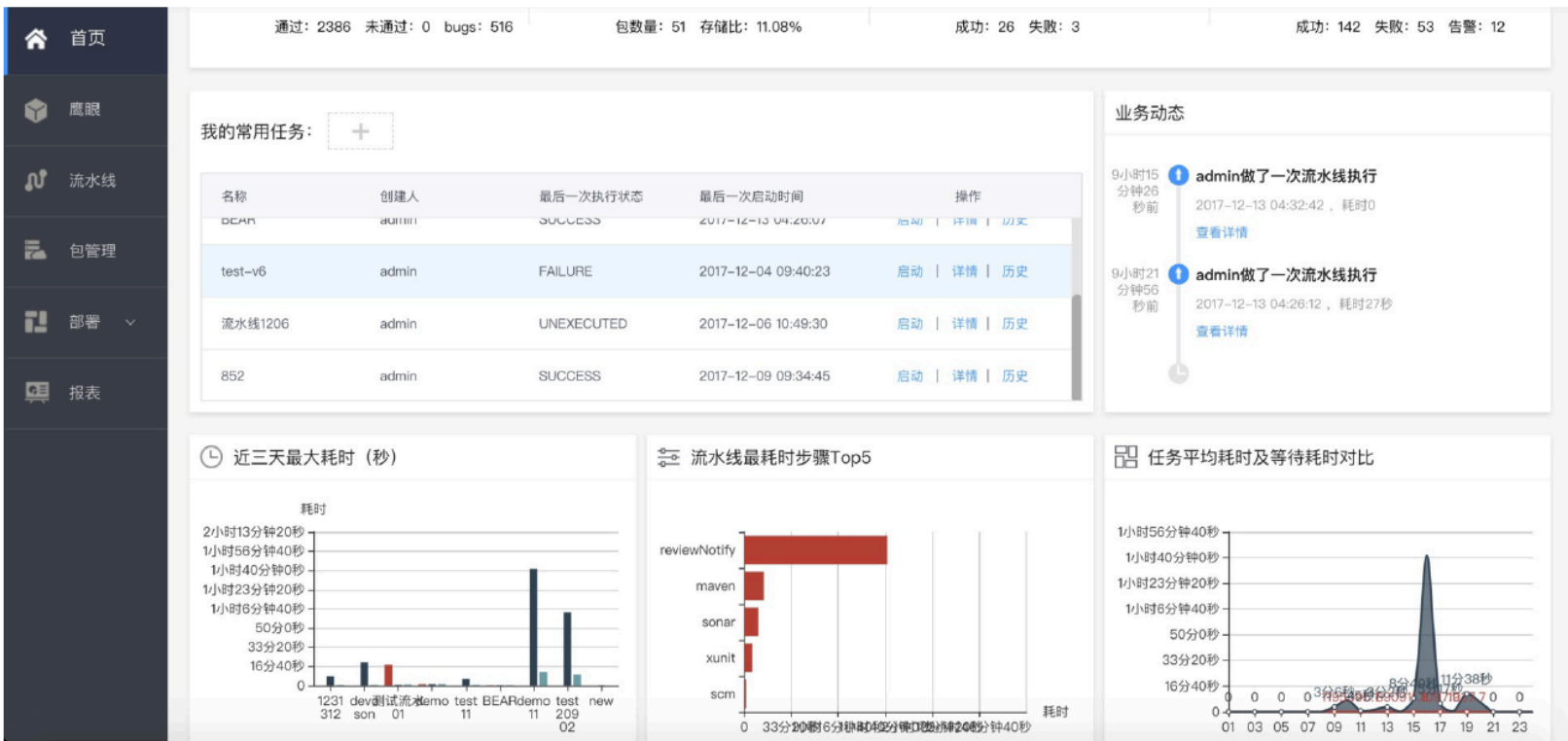
SALTSTACK puppet labs

部署工具

预发布/生  
产环境

软件可复用的交付流水线







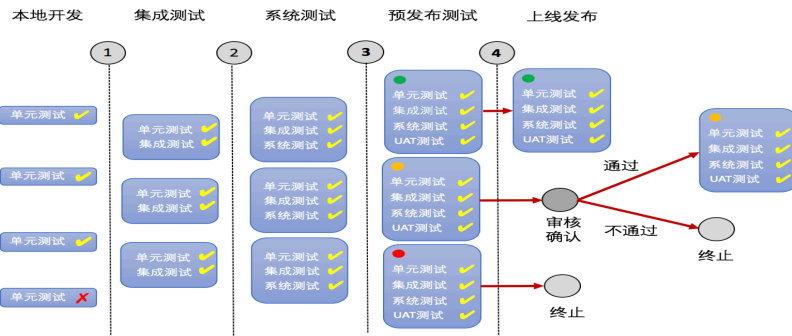
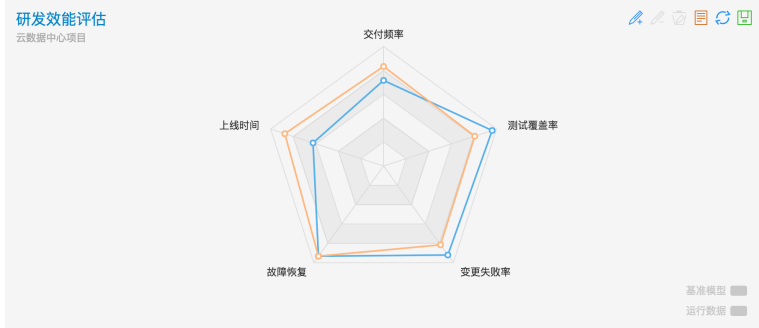
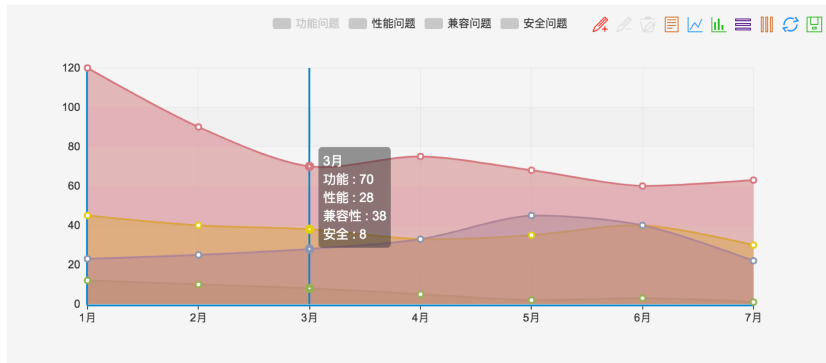
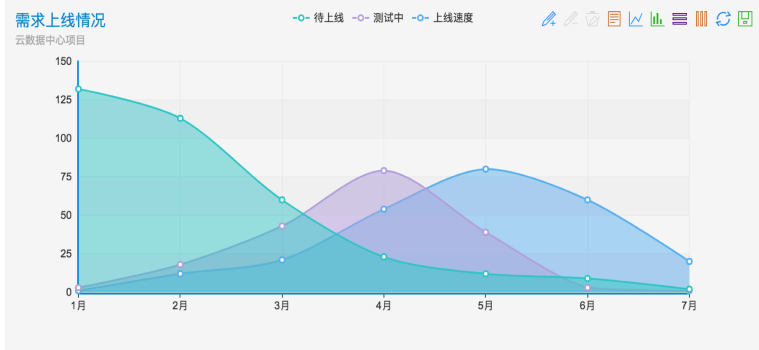
微服务流水线01

1 开始 0个参数	2 步骤1 1个任务	3 开始 0个任务
全局参数	脚本	本流水线共有: 1个stage 0个任务
添加参数	添加任务	

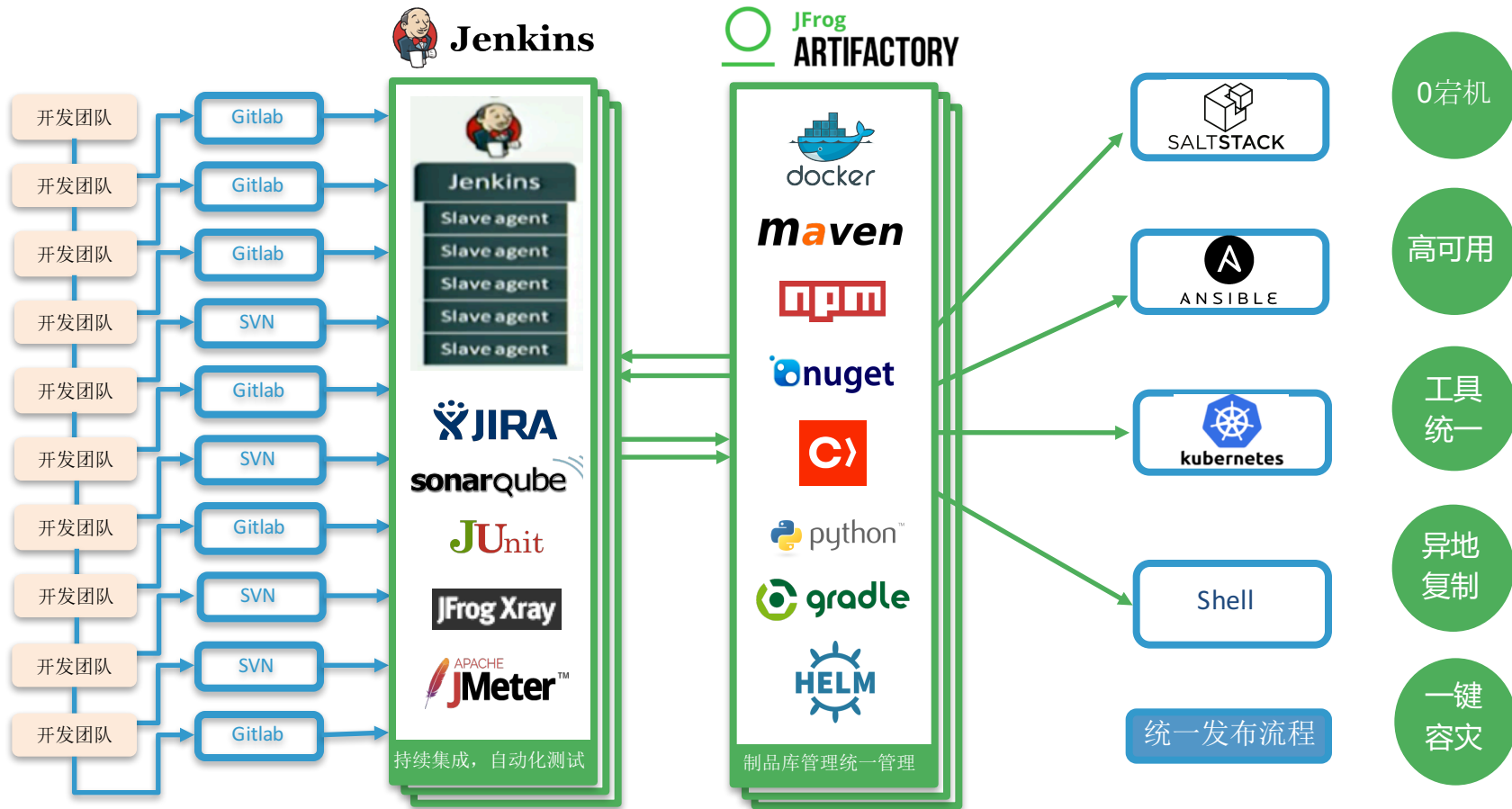
保存 取消



# 持续反馈 - 发布质量可视化



# CI/CD 统一流程管理





JFrog 2018

DevOps 技术交流群



免费试用地址:

<https://www.jfrogchina.com>