



GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

# 如何快速引入DevOps • 提升产品交付能力

曹水生 极策科技研发总监



# GIAC

## 全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



关注msup  
公众号获得  
更多案例实践

GIAC 是中国互联网技术领域行业盛事，组委会从互联网架构最热门领域甄选前沿的有典型代表的技术创新及研发实践的架构案例，分享他们在本年度最值得总结、盘点的实践启示。

2018年11月 | 上海国际会议中心



高可用架构  
改变互联网  
的构建方式

# 目录



1

团队背景&研发的困境

2

DevOps实施路径

3

DevOps实践试点和推广

4

团队的收获&思考





# 团队的背景

16.10

- 房产营销平台30个城市800个案场上线
- 园区招商运营平台上线
- CRM平台上线
- 大数据平台产品 X系列V1.0发布

17.06

- 客户大数据平台发布
- 客户经营平台URM上线

16.06

- 房产营销平台产品上线

16.12

- 集团房产数据中心建设完成
- 大数据精准营销产品发布
- 大数据平台产品 X系列V2.0发布

17.11

- 养老运营平台上线
- .....





# 研发的问题现象

问题

1

为了获取投资商的信任，我们定了一个几乎完不成的目标。

2

产品越来越多，交付周期越来越短，产品需求又变了.....

3

测试人员版本又没有按时发布，理由听起来都非常合理。

4

开发为了赶任务，接口设计、代码质量和规范打折扣。

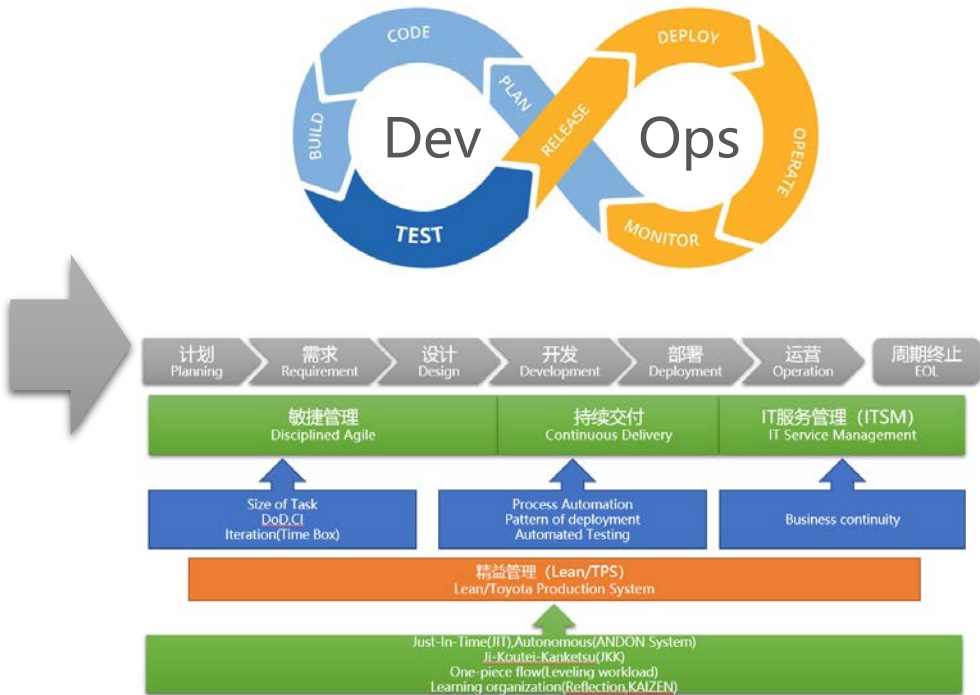
5

版本部署问题层出不穷.....参数忘改了，这个包的版本不对





# 解决问题的办法？



# 目录

1 团队背景&研发的困境

➔ 2 DevOps实施路径

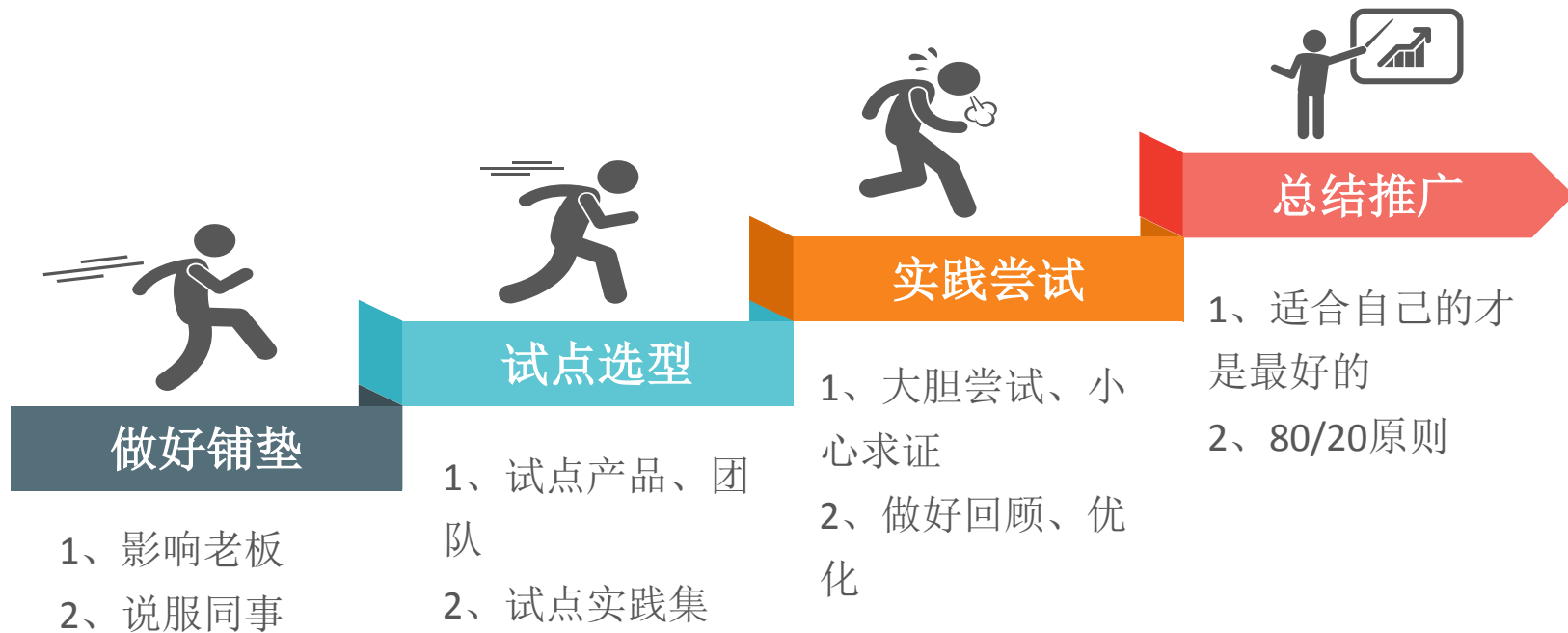
3 DevOps实践试点和推广

4 团队的收获&思考





# DevOps实施路径

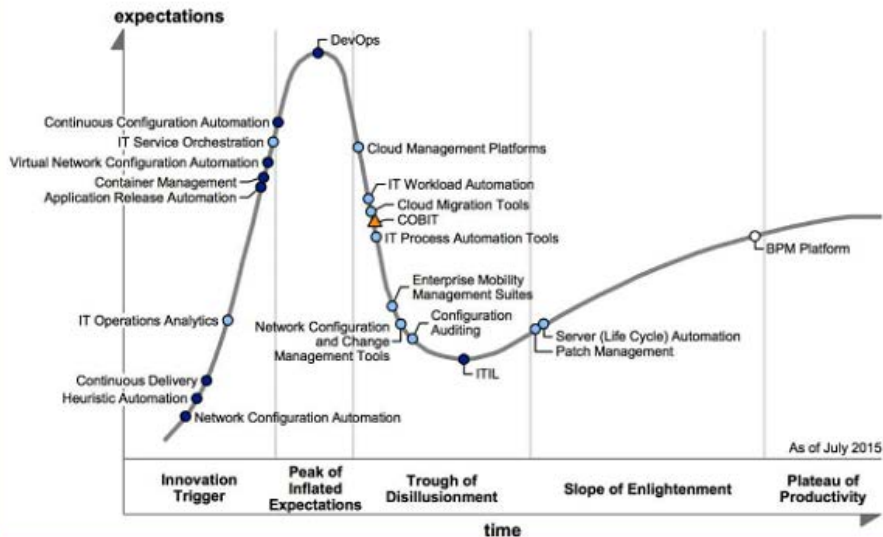




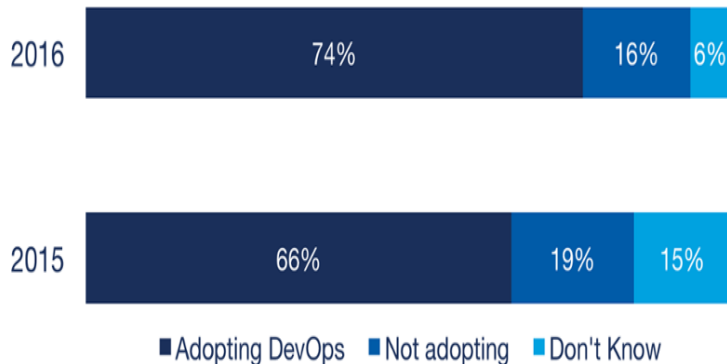


# 导入前准备-说服老板支持

Figure 2. Hype Cycle for I&O Automation, 2015



## DevOps Adoption Up in 2016



Source: RightScale 2016 State of the Cloud Report

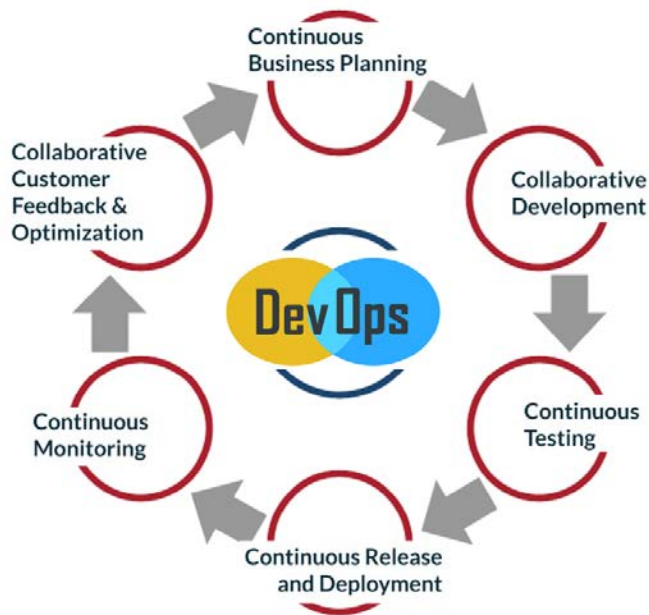
根据Gartner最新出炉的2015 I&O Automation报告，DevOps同样正处其技术发展曲线的最高点。

DevOps的采用已经在2015年-2016年大约增长了8%，而且这个数字预计将在2018年显著增长





# 导入前准备-说服老板支持



- 产品快速推向市场（比如，缩短开发周期时间和更高的部署频率）
- 提高质量（比如，提高可用性，提高变更成功率，减少故障，等等）
- 提高组织的有效性（比如，将时间花在价值增加活动中，减少浪费，同时交付更多的价值至客户手中）





# 导入前准备-说服老板支持

## 1个简单的产品1年人工打包部署时间成本？



每个产品  
工程数

4个

\*

每次打包  
部署

1小时

\*

每迭代交付打  
包部署次数

3轮

\*

测试服务  
器环境

\*

每月产品  
迭代版本

3个

\*

每年12个  
月

12月

= 9.8人\*月

147W

10个产品的成本？



# 导入前准备-说服团队主动参与

DevOps Virtuous Cycle of Technical Debt Repayment



- 产品经理不用总是担心无法按时按质发布版本
- 开发人员不用频繁编译打包
- 测试人员编写完测试脚本后，自动化测试看报告即可
- 运维人员不用人员部署各种包到服务器
- QA可以自动化代码规范检查，生成代码质量报告





# 导入前准备-说服团队主动参与

**DevOps开发工程师** 超级雇主

20k-40k / 深圳 / 经验3-5年 / 本科及以上 / 全职

高级 后端开发 Java linux SVN

1天前

职位诱惑:

职位描述:

工作职责:

负责持续集成和持续部署的相关工具的开发工作

工作要求:

大学本科及以上学历, 计算机或者相关专业;  
了解DevOps相关开发运维系统, 熟悉CI/CD研发流程。  
熟悉持续集成中的版本控制, 质量内建工具等工具。

未来职业发展  
必备技能, 能  
偷懒, 又能升  
职加薪!!!

**DevOps工程师 (高级)** 千万家门

30k-50k / 深圳 / 经验3-5年 / 本科及以上 / 全职

python 运维 linux OpenStack

职位诱惑:

职位描述:

工作职责:

负责内部PAAS建设  
推进 DevOps 理念落地



# 目录

1 团队背景&研发的困境

2 DevOps实施路径

➔ 3 DevOps实践试点和推广

4 团队的收获&思考





# 选择试点产品团队的要点





# 选择试点实践的技巧

适合团队的

最容易产生价值的

能解决重复性人工作业

容易操作推行

短期效果兼顾长期收益

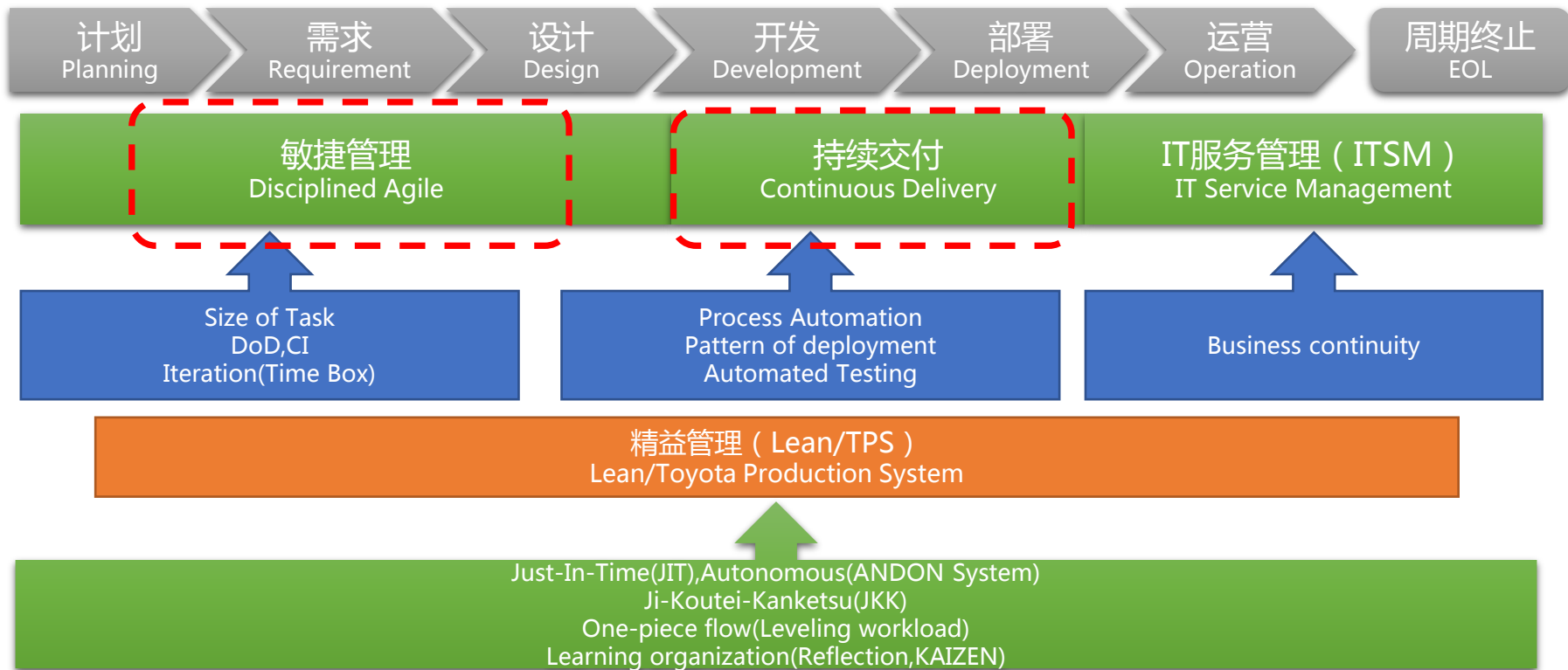






# 选择的实践

最容易产生价值的实践先做，增强团队的自信心





# 提倡的团队氛围



## 奖励持续的“改进冒险”：

每一个改进的尝试都是一场“改进冒险”。在冒险中，失败并不可怕，可怕的是失去了继续改进的动力。团队成员需要给当前的状况有建设性的建议和反馈，而不仅仅是抱怨和指责。

## 自动化一切：

自动化是 DevOps运动以及促进合作的基石。将测试、配置和部署自动化可以让人们释放出更多的精力，并专注于更有价值的活动，此外还能减少人为失误。

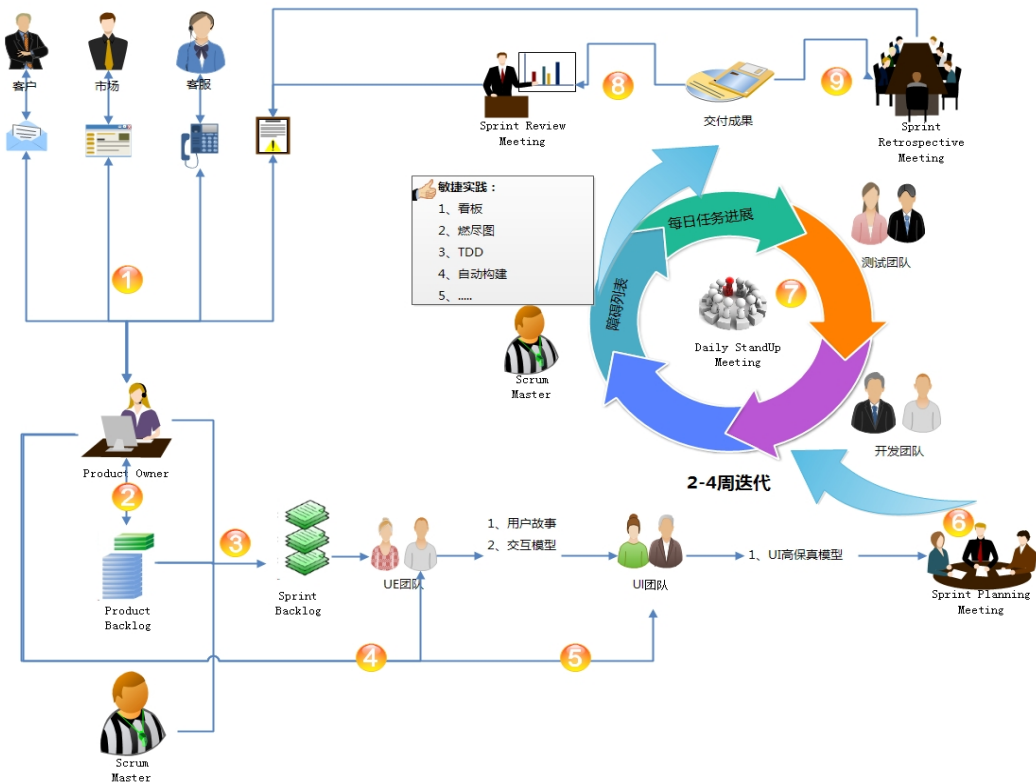




# 敏捷的实践-团队协作

敏捷开发协作流程

——20160426定稿



- 1 通过多种渠道获取产品需求
- 2 刷新产品需求优先级，维护产品需求。
- 3 产品经理和敏捷教练梳理迭代需求
- 4 1、PO给UE团队讲需求  
2、UE团队根据需求理解输出交互模型
- 5 1、UI团队根据交互模型设计并输出高保真模型
- 6 迭代规划会议。1、迭代需求澄清。2、任务分解  
3、评估工作量。4、认领任务。5、迭代计划（timebox）
- 7 1、组织每日站立会。2、跟进任务进展。3、扫清障碍。
- 8 1、showCase  
2、产品经理验收成果
- 9 迭代回顾会议。1、Top 3 Well。2、Top 3 Less Well。  
3、形成有效解决方案步骤，纳入下一代任务。

- 仪式感很重要
- 建立起团队的游戏规则
- 形似---神似
- 习惯的力量





# 敏捷的实践-团队协作

迭代计划 (TimeBox)\_Two Weeks

	第一周					第二周						
	星期一	星期二	星期三	星期四	星期五	星期一	星期二	星期三	星期四	星期五		
PO	迭代规划会议 1、迭代需求宣讲 2、分解任务 3、工作量估算 4、认领任务 5、迭代计划	少量需求澄清				下一代需求细化					1、迭代评审会议 (showCase) 2、迭代回顾会议 (Top3)	
		维护Product Backlog				协调UE完成下一代交互模型						
							协调UI完成下一代高保真模型					
					组织每日站立会，跟进任务进展，扫清障碍							
SM							下一代需求理解		下一代准备工作			
DEV		迭代需求开发					修复BUG					
TE		设计测试用例			评审用例	用例终稿			测试&回归			
UE							下一代需求理解，交互模型设计					
UI								下一代需求理解，高保真模型设计				

PO	Product Owner (产品经理)
SM	Scrum Master (敏捷教练)
DEV	Develop (开发团队)
TE	Test (测试团队)
UE	UE (UE团队)
UI	UI (UI团队)

找到团队自己的节奏才能发挥团队的威力





# 自动构建-分支管理策略

## GIT分支管理策略

Master: 确保每次都能即时向预发布和生产环境发版本

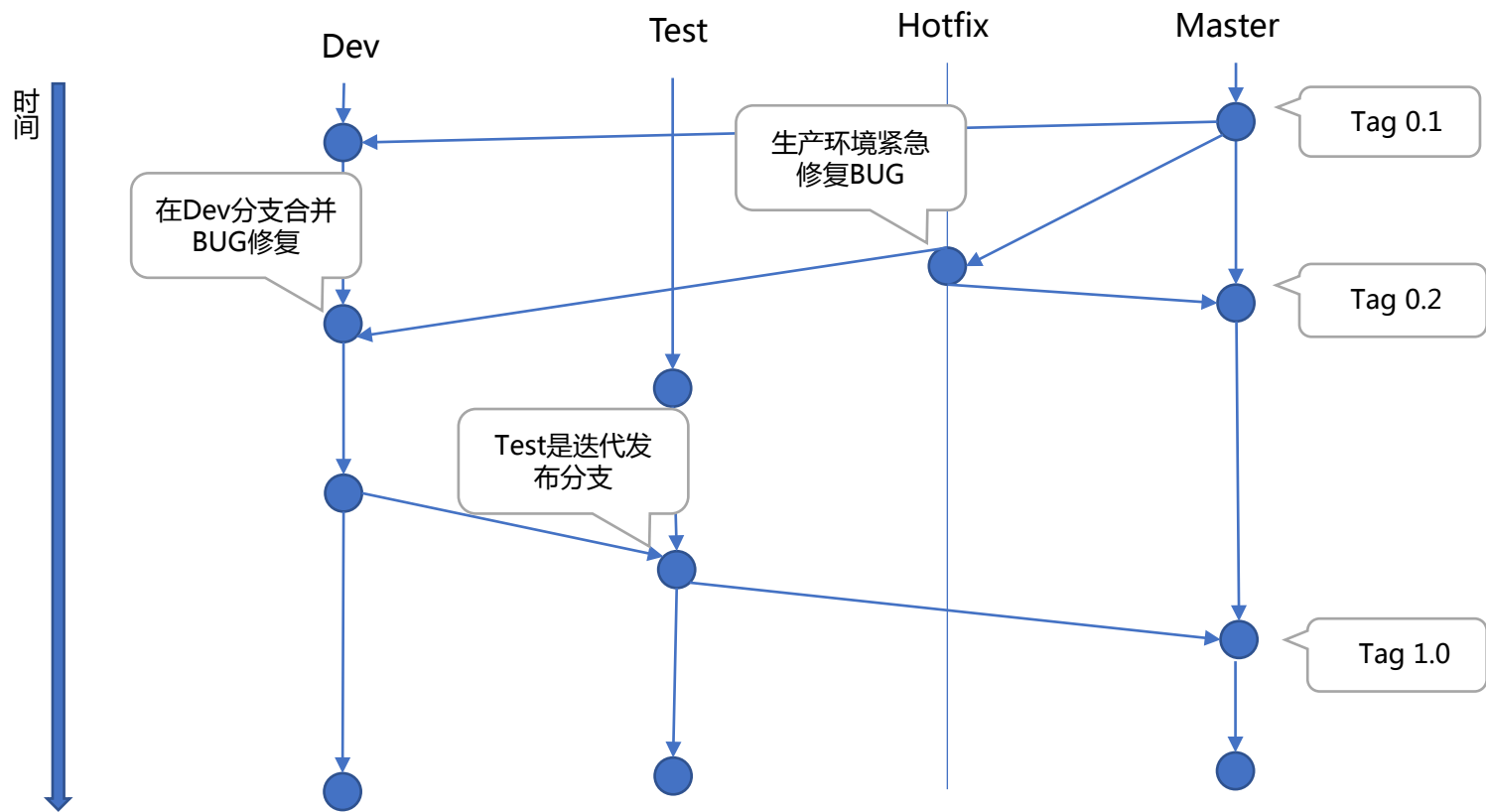
Test: 迭代交付测试分支，向测试环境发版本

DEV: 开发人员分支，向开发环境发版本





# 自动构建-分支管理策略





# 自动构建-分支管理策略

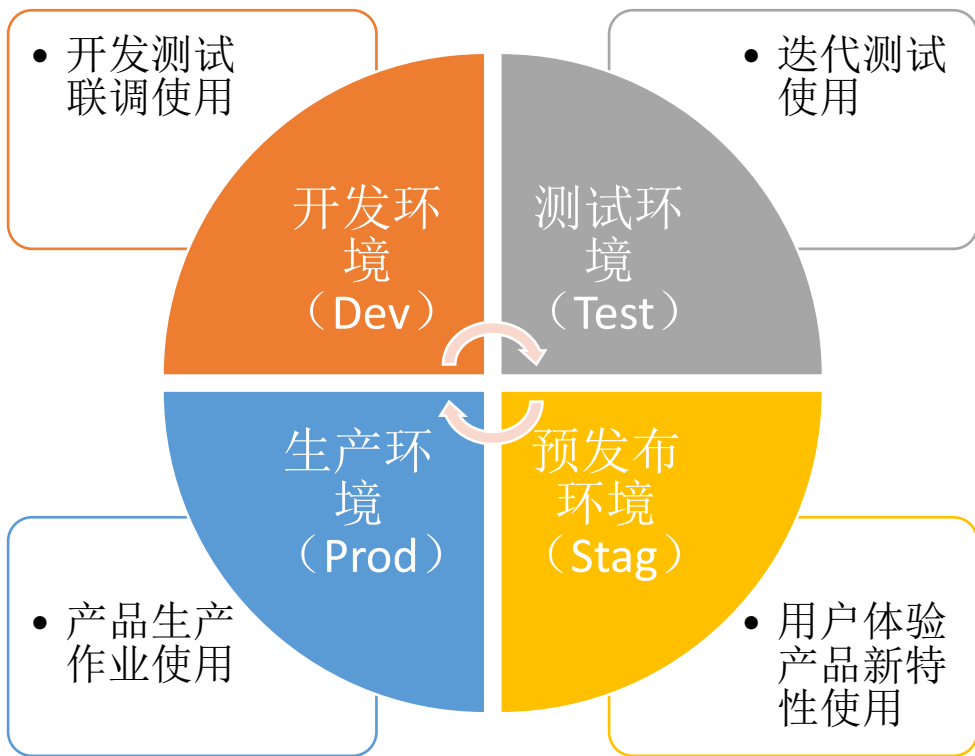
分支类型	命名规范	创建自	合并到	说明
dev	dev/*	master	test	新开发的需求
test	test/*	dev	dev 和 master	迭代新版本的发布
hotfix	hotfix/*	master	develop 和 master	生产环境中发现的紧急 bug 的修复
master	master/*	test/hotfix		预发布环境和生产环境发版本

代码版本分支是自动化构建的基础，开发团队必须遵守





# 自动部署-环境策略







# 自动部署-环境策略

分类	用途	目标代码来源	构建策略	Job责任人	Job权限配置
开发环境 ( dev )	主要提供给开发人员对代码进行测试	dev分支	轮询版本管理系统dev分支代码变更进行自动构建 手动触发构建	开发leader	开发leader可配置 匿名可执行构建
测试环境 ( test )	主要提供给测试人员对功能、性能进行测试	test分支	轮询版本管理系统test分支代码变更进行自动构建 手动触发构建	开发leader	开发leader可配置 测试人员可执行构建 匿名可查看
预发布环境 ( stag )	主要用于正式上线前的测试验证和产品验收	master分支	根据测试人员预发布通知，手动触发构建	DevOps工程师	DevOps工程师可配置 开发leader可查看
生产环境 ( prod )	提供生产作业的正式环境	master分支	根据测试人员上线通知，手动触发构建	DevOps工程师	DevOps工程师可配置 开发leader可查看





# 自动构建-job规范

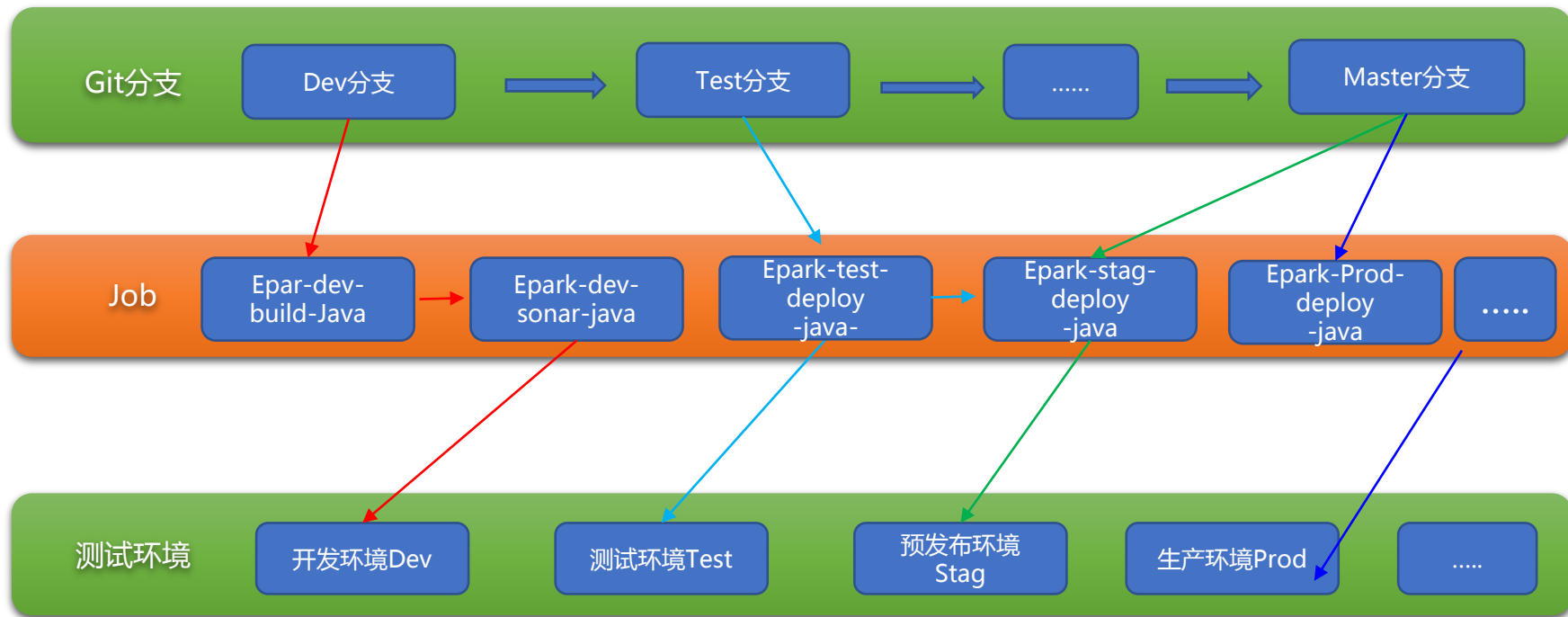
产品名称-环境名称-Job类型[-专业线] （其中“[-专业线]”为可选，必要时加以区分）				
产品名称	环境名称	Job类型及说明		专业线
XXX1	dev	codeCheckout	代码检出	Java
XXX2	test	build	构建	H5
XXX3	stage	sonar	代码静态检查	pc ( pc端H5 )
	prod	deploy	部署	Android
	-	codeMerge	代码合并	IOS
	-	apiTest	接口自动化测试	PHP
	-	...	...	-
示例： XXX1-test-build-Java				

能从几百个job中迅速找到你要的JOB有助于提高工作效率





# 自动部署策略规则





# 技术的构建部署方案

Java构建部署方案

H5构建部署方案

android构建部署方案

ios构建部署方案

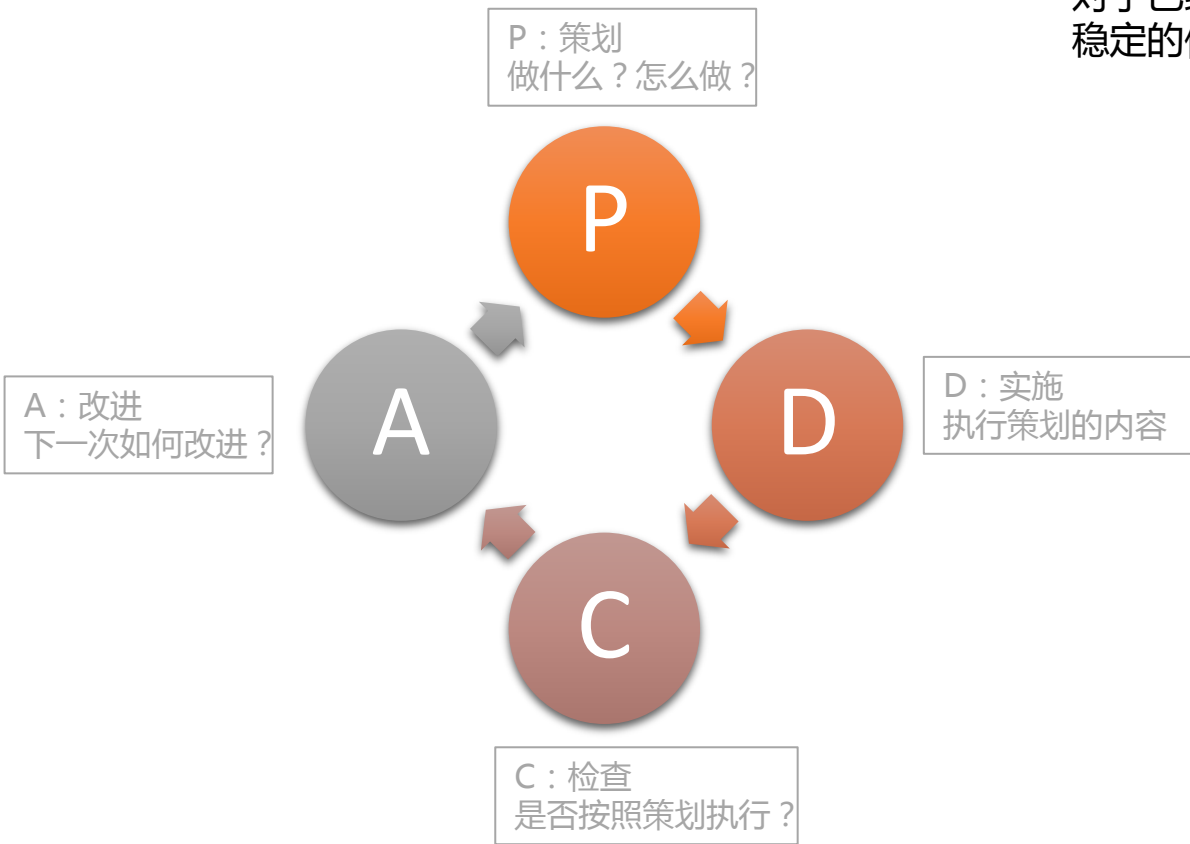
PHP构建部署方案





# DevOps实践试点-复盘

对于已经形成好的实践固化掉，产生持续稳定的价值



产品研发管理过程符合度Checklist					
迭代名称		产品经理		过程符合度	
迭代周期		开发leader		检查日期	
类别	序号	检查项		检查结果	完成情况说明
需求	1	需求相关文档是否及时输出（原型+需求描述+产品Backlog+必要的UML图）？			
	2	需求是否经过评审？是否提前预审（至少提前一天）？			
	3	需求是否纳入了待道中进行管理？			
	4	需求变更是否记录到需求变更清单并知会到全员？是否进行了影响评估？			
	5	需求变更是否在原型中进行了标识？			
	6	需求相关文档是否及时完成归档？			
计划	7	迭代计划会议是否按时召开？迭代计划是否及时（迭代计划会议开完当天）发出？			
	8	是否进行了任务分解与估算？			
	9	任务是否纳入了待道进行管理（包含预计工作量、责任人等必要属性）？			
设计	10	接口设计是否输出？是否经过评审？			
	11	数据库设计是否输出？是否经过评审？			
	12	方案设计（类图、时序图等）是否输出？是否经过评审？			
	13	设计相关文档是否及时完成归档？			
开发	14	每日站会是否按时、有效召开？			
	15	是否组织了代码检视？			
	16	每日任务状态是否及时刷新？是否记录了任务的实际工作量？			
	17	版本是否按时构建？			
	18	修复BUG是否分析了BUG的产生原因，并记录解决方案（修改逻辑）？			
测试	19	测试用例是否覆盖所有需求？			
	20	测试用例是否经过评审？是否提前预审（至少提前一天）？			
	21	需求变更时是否及时补充必要的测试用例？			
	22	测试用例是否纳入了待道中进行管理？			
	23	BUG是否纳入了待道中进行管理？			
	24	《迭代测试报告》是否及时（版本上线后半个工作日内）输出并归档？			
发布	25	版本是否按时预发布？			
	26	版本是否按时上线？			
回顾	27	《迭代发布报告》是否及时（版本上线后半个工作日内）输出并归档？			
	28	迭代回顾会议是否及时（新迭代开始前）召开（每个迭代都要求进行回顾）？			
	29	复盘分析、总结出TOP问题，并制定改进措施（改进措施应符合SMART原则）？			
	30	回顾总结是否及时（新迭代开始前）归档？			
检查结果说明：OK-满足、POK-部分满足、NO-不满足、NA-不涉及					





# DevOps实践推广



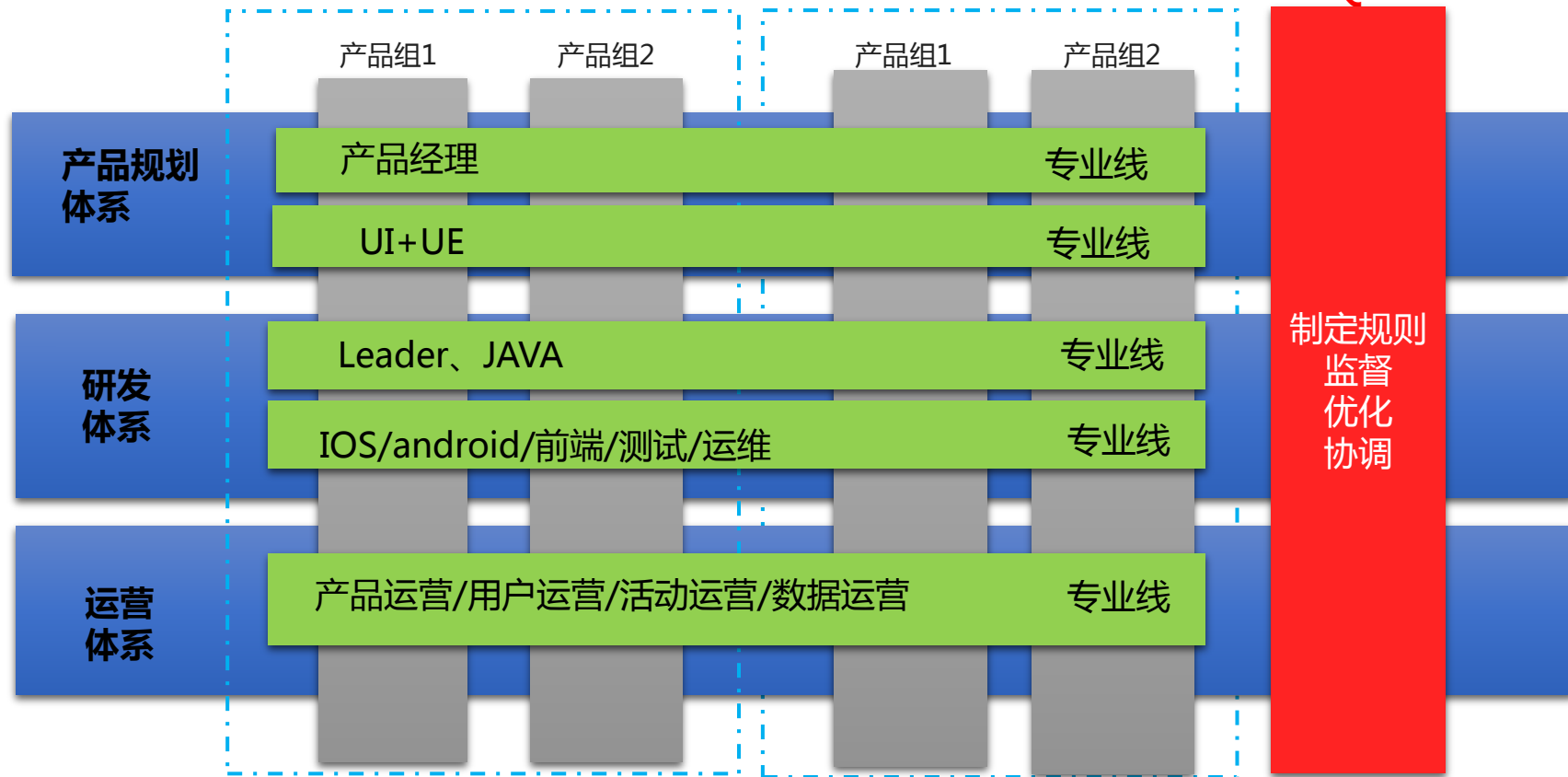


# DevOps实践推广

产品线一

产品线二

SMO  
QA





# DevOps实践推广



产品团队一



+



+



+

...



产品团队二



+



+



+

...



产品团队三



+



+



+

...

团队文化





# 目录

1 团队背景&研发的困境

2 DevOps实施路径

3 DevOps实践试点和推广

➔ 4 团队的收获&思考





# 团队的收获-大家的反馈

测试人员说：接口每天跑出来，看报告即可，不要再全量重复测试了。

开发人员说：不用频繁去编译打包了，重复工作交给电脑做，专注业务逻辑、技术架构、代码质量

运维人员说：部署全交给电脑跑，不会出现配置文件低级错误了，部署重复劳动少了。

产品经理说：交付时间点和交付质量更靠谱了。





## 团队的收获-算笔账

117人月, 176W

183

Jenkins中JOB个数

19731

构建成功的次数

3176

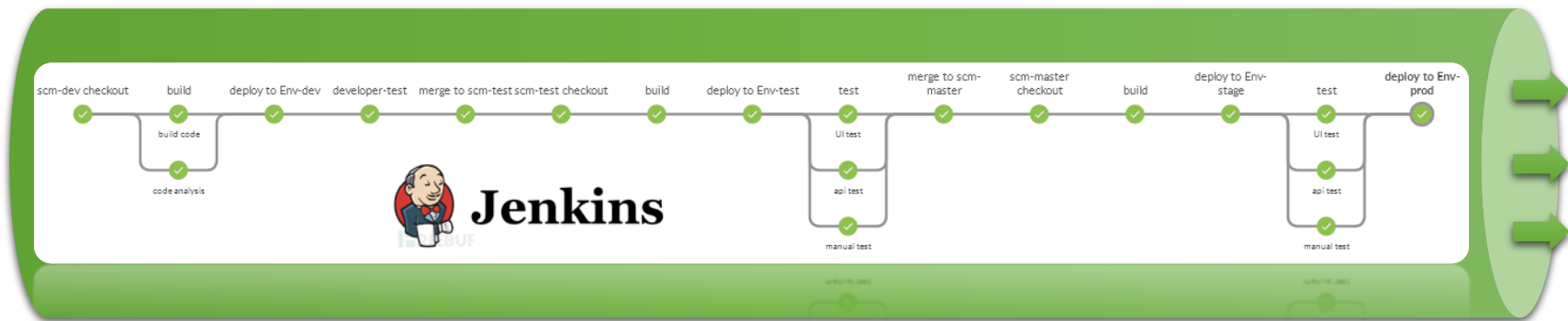
接口自动化测试执行次数

更大的价值:  
产品交付能力





# 团队的收获&思考-中小企业全开源部署方案



# GIAC

## 全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



关注公众号获得  
更多案例实践