Gdevops

全球敏捷运维峰会

移动时代DevOps转型之旅

演讲人：欧阳辰

我是谁？ >15年的软件研发老兵

欧阳辰

	开发主管	高级开发经理/工程师	架构师/主管
			 
7年	3年	10年	广告平台，大数据 1.5年

公众号：

www.ouyangchen.com

关于小米

MI (Mobile Internet)

2010年成立
2011年推出手机

Mi.Com

议程

- 什么DevOps
- 微软是如何做DevOps
- 小米是如何做DevOps
- 如何成功向DevOps转型
- DevOps的基础架构
- 结束语

我对DevOps的4个观点

- 竞争性软件公司终将采用DevOps
- DevOps不是银弹，是关于效率的文化，自动化的技术。
- 大量专职运维和测试将消失
- “靡不有初，鲜克有终”

一些你也许知道的事情：“测试已死”

“测试已死”

- Alberto Savoia 2011, GTAC

1) 做正确事情 VS 正确的事情

为什么微博是140个字？

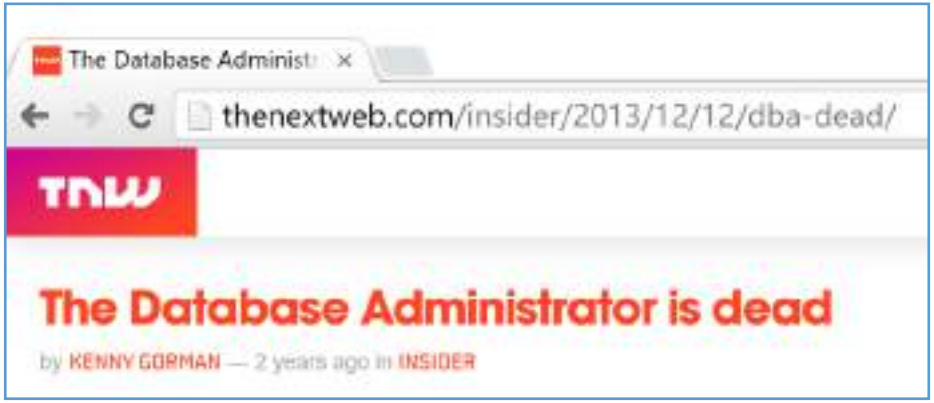
为什么搜索引擎总是返回10个结果？

2）构造原型 (Preto-type V.S Proto-type)

尽早失败，
常常失败

95% 移动应用不能盈利！
80% 创业公司失败

“消失”的技术和角色！DBA? Ops?



什么是质量？ 什么是测试？

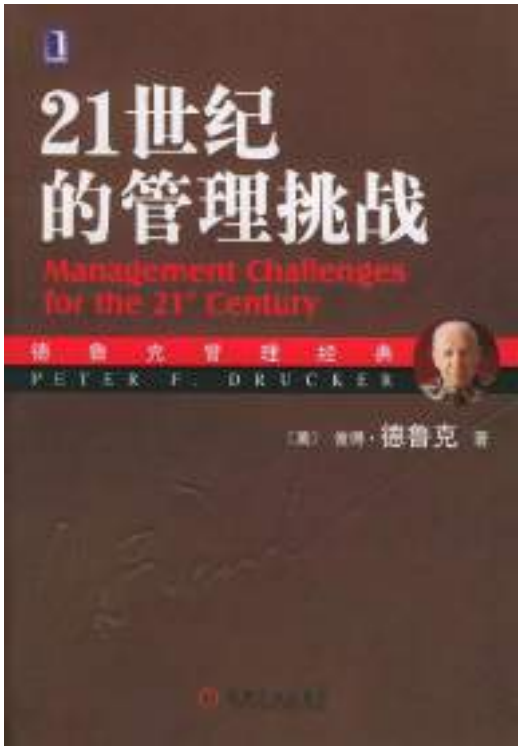
- 测试是为了发现程序中的错误而执行程序的过程
  - 《软件测试的艺术》1979
- 确保软件是保证用户需求的程度
  - IEEE
- 质量度量和缺陷预防
- 最大程度和最快速度满足客户的需求

- 质量是用户体验，提供用户价值
- 质量是经济竞争性
- 质量
  - 代码覆盖率
  - 用例自动化率
  - 缺陷数量分布
  - 响应时间

软件 → 服务

知识工作者的生产率

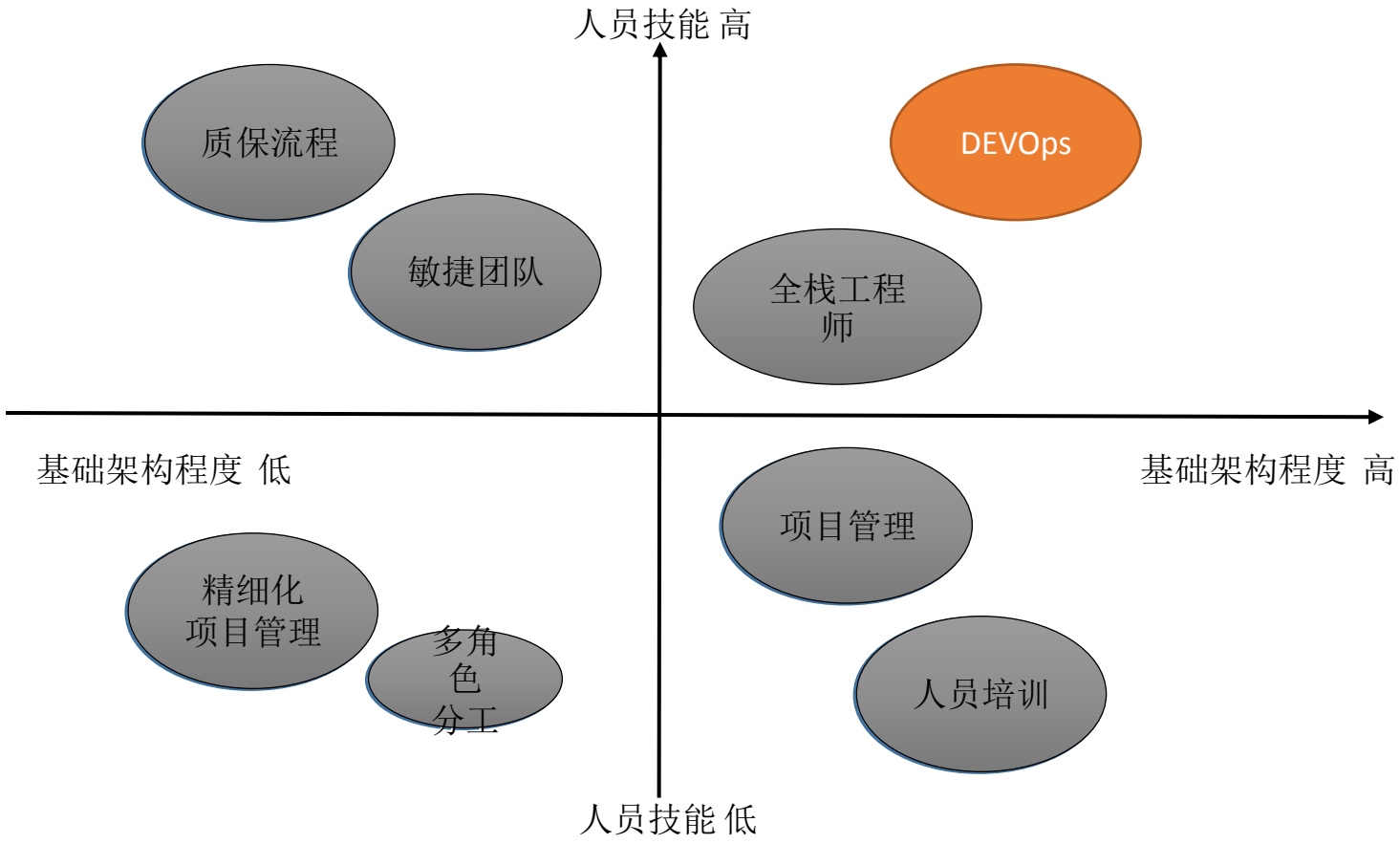
- 搞清楚知识工作者的任务最为关键
- 自我管理生产率
- 不断创新
- 不断学习和教导
- 质量重于数量
- 知识工作者为资产



我理解的DevOps！

- 所有人干所有活，使用工具而不是依赖他人
- 对代码和业务24小时负责
- DevOps是一种文化转型和演化
- DevOp是一个提升生产力的契机
- 敏捷解决的流程的问题
- DevOps帮助解决协作和融合的问题
- 都是提高发布效率的方法

DevOps的象限


人员技能 高
质保流程
敏捷团队
DEVOps
全栈工程师
基础架构程度 低
基础架构程度 高
项目管理
精细化项目管理
多角色分工
人员培训
人员技能 低

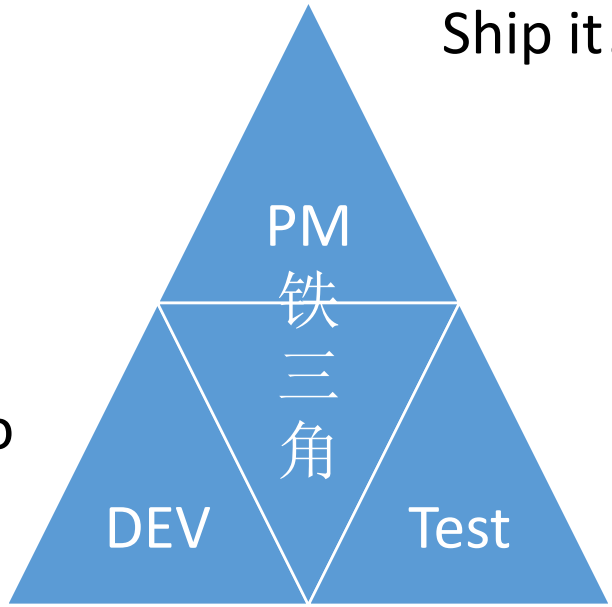
微软的测试转型

Combined Engineering

世界上最强的测试团队，最深厚的测试文化

2009年前的模型：铁三角的价值！

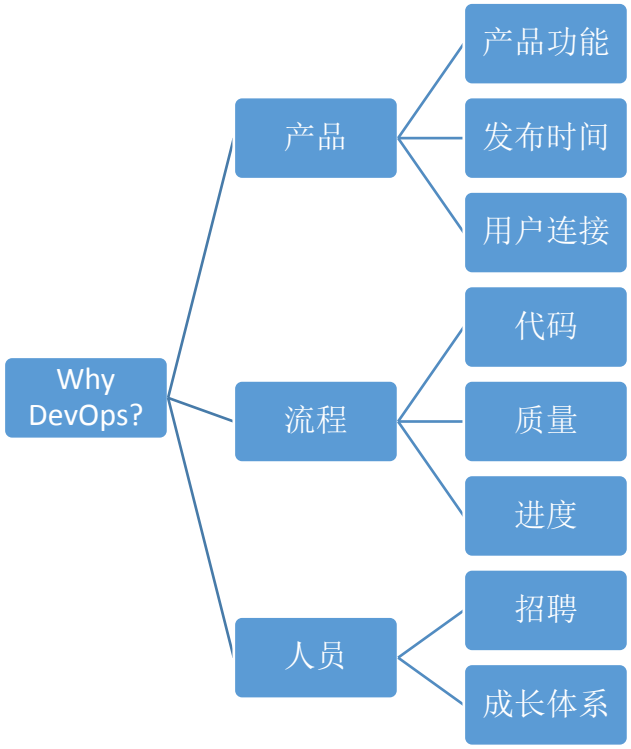
Ship it!



Implementation!

Quality Feedback!

测试转型前的一些问题



低MTBIAMSH

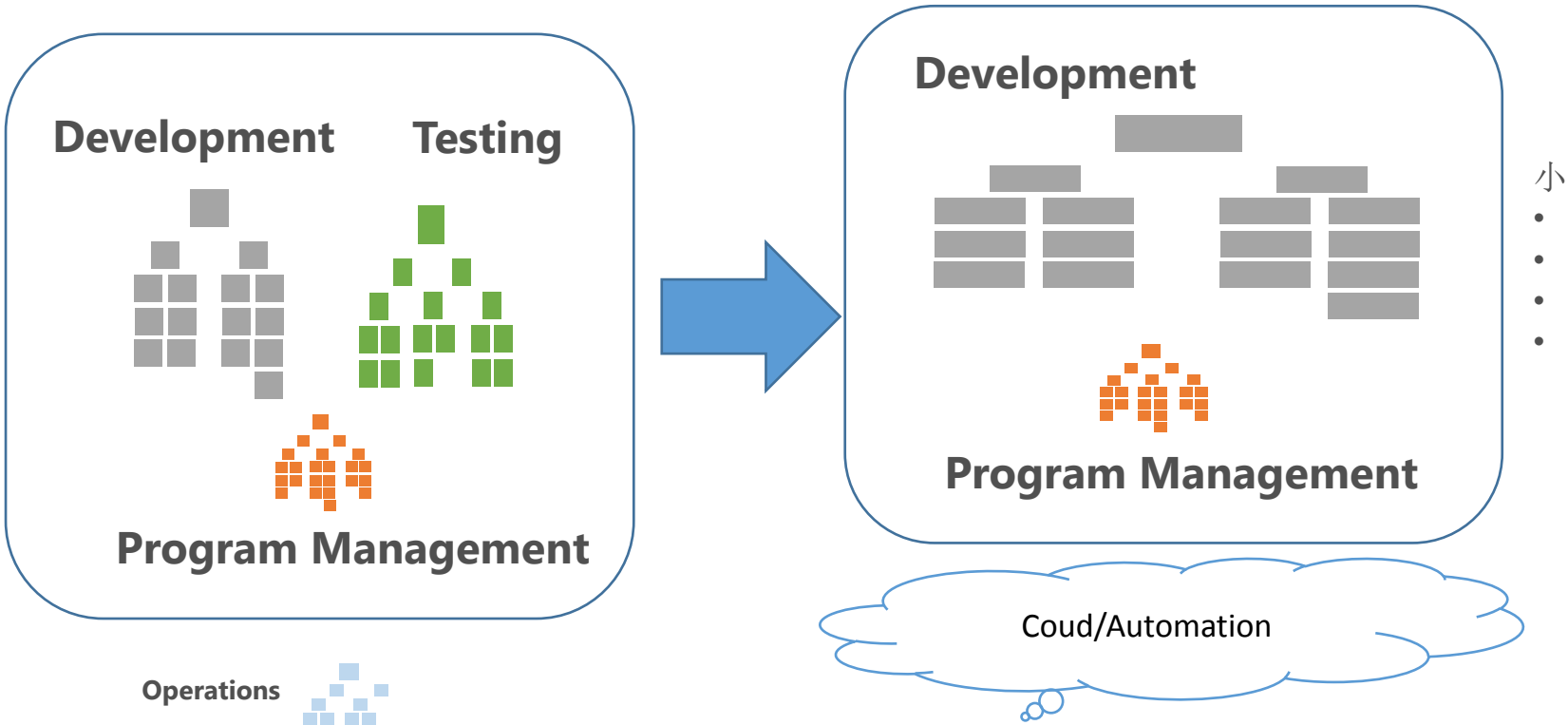
MTBIAMSH (Mean Time Between Idea And Making Stuff Happen)

不适应快速发布的流程

-One Page Spec
-Deployment hell
-婆婆太多

测试人员招聘难，成长空间不明

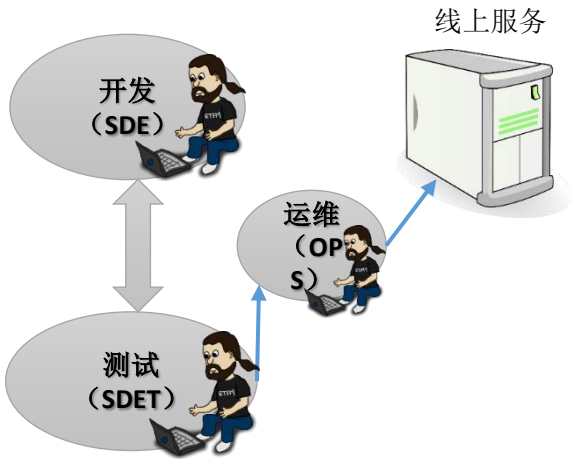
组织的变化：减少角色，管理扁平



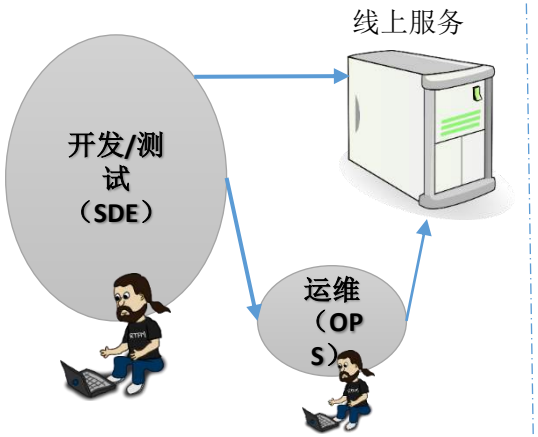
小组：

- 10-12人
- 自管理
- 12-18月
- 1个产品/其他都是开发

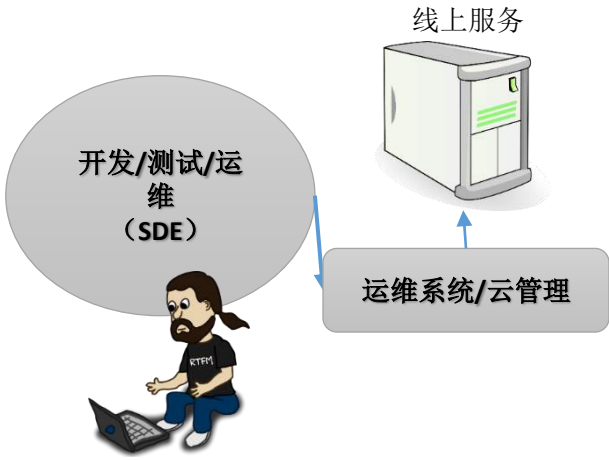
微软的DevOps之旅



阶段一（2008以前）



阶段二（2009~2013）

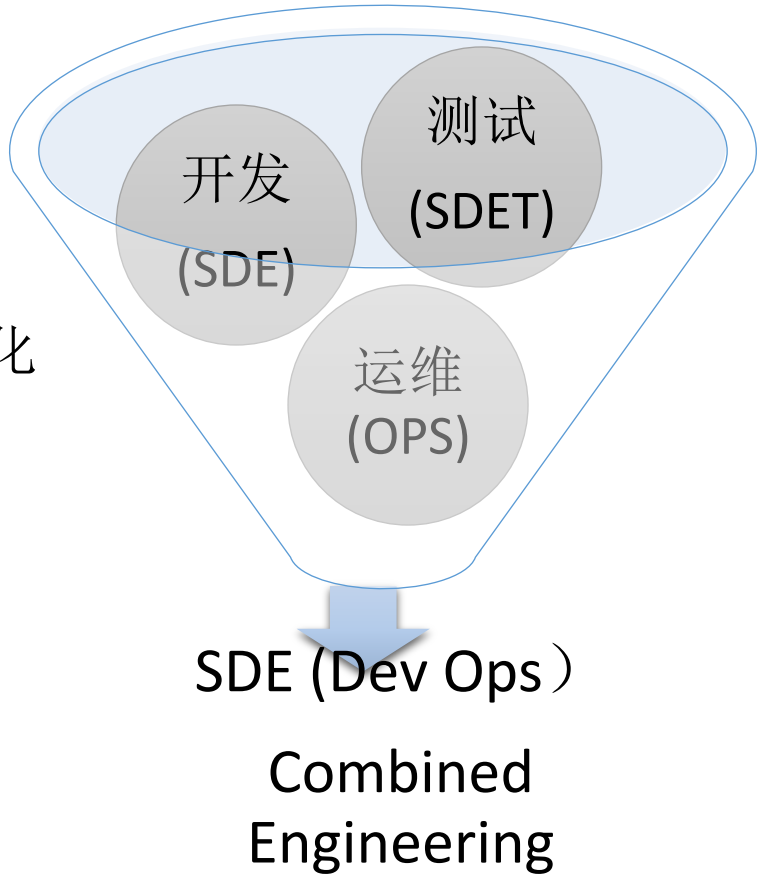


阶段三（2013~）

微软的DevOps 之旅：服务端开发

微软DevOps之旅：角色的融合

- 简化角色，单一负责
- 复杂操作标准化，自动化
- 可追溯的操作



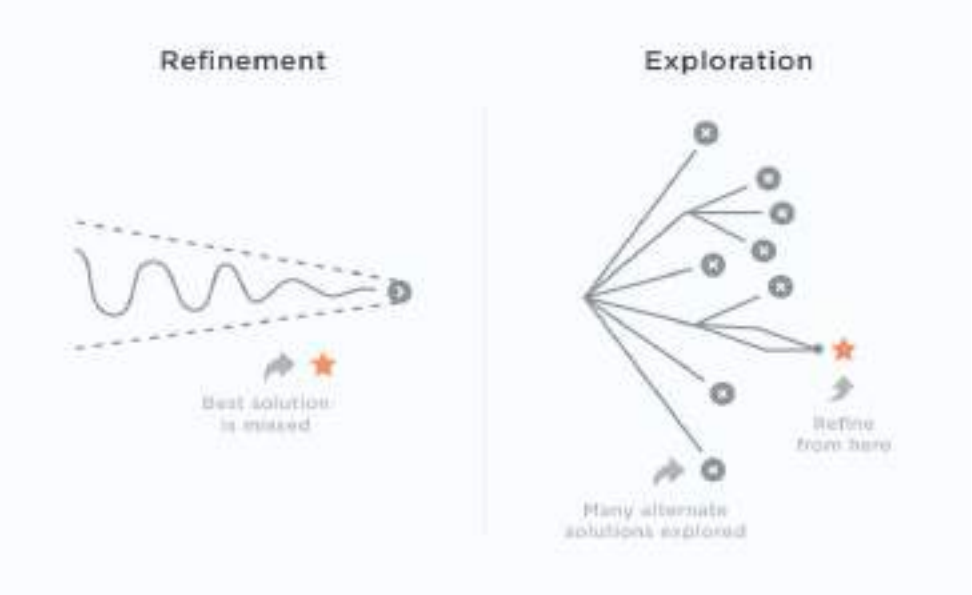
一些任务和角色的转移

OPERATIONAL CAPABILITY	PRE-DEVOPS TRADITIONAL OPS	NOW DEVOPS
Capacity Management	Ops	DevOps*
Live Site Management	Ops	DevOps*
Monitoring	Ops	DevOps**
Problem Management	Ops	DevOps*
Change Management	Ops	Dev**
Service Design	Dev & Ops	DevOps
Service Management	Ops	DevOps*

All DevOps

工作方式：服从计划-》主动探索

- 主动探索
- 数据驱动
- 用户为中心
- 极客文化
- Show me the code



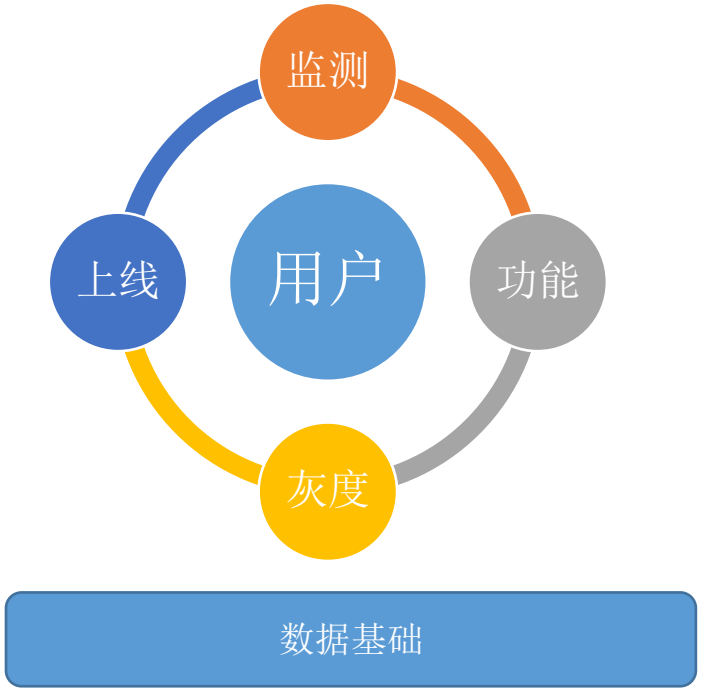
越来越少的专业测试讨论☹

Combined Engineering 效果(类 DevOps)

- 发布节奏明显加快
- “做和不做” → “如何更快”
- 生产力问题是焦点
- 线上产品的主人翁精神

微软的几个工程原则

- 线上事故第一优先，
  - Live Site First
- 敏捷的计划和执行
  - Adaptive and Agility
- 持续优化工程效率
  - Continuous Productivity
- 数据驱动的工程实践
  - Data Driven engineering practice



互联网公司的测试

.0：1

Combined Engineering

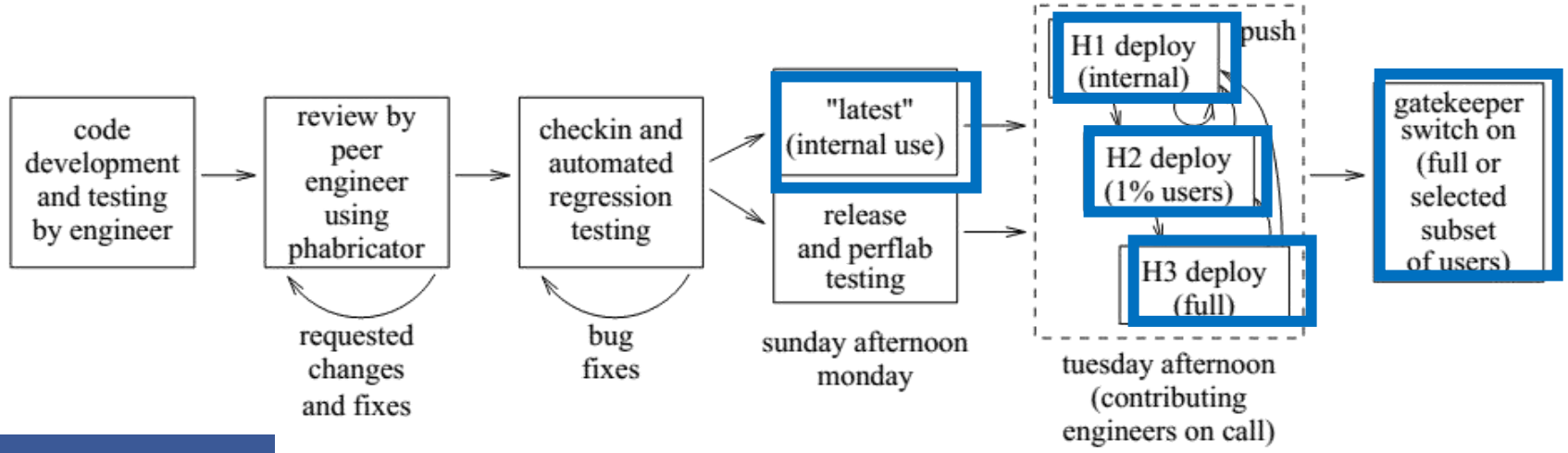
No Testers!

- 单元测试
- 灰度为王

Engineering Productivity "We are small and elite Special Forces that have to depend on superior tactics and advanced weaponry to stand a fighting chance at success."

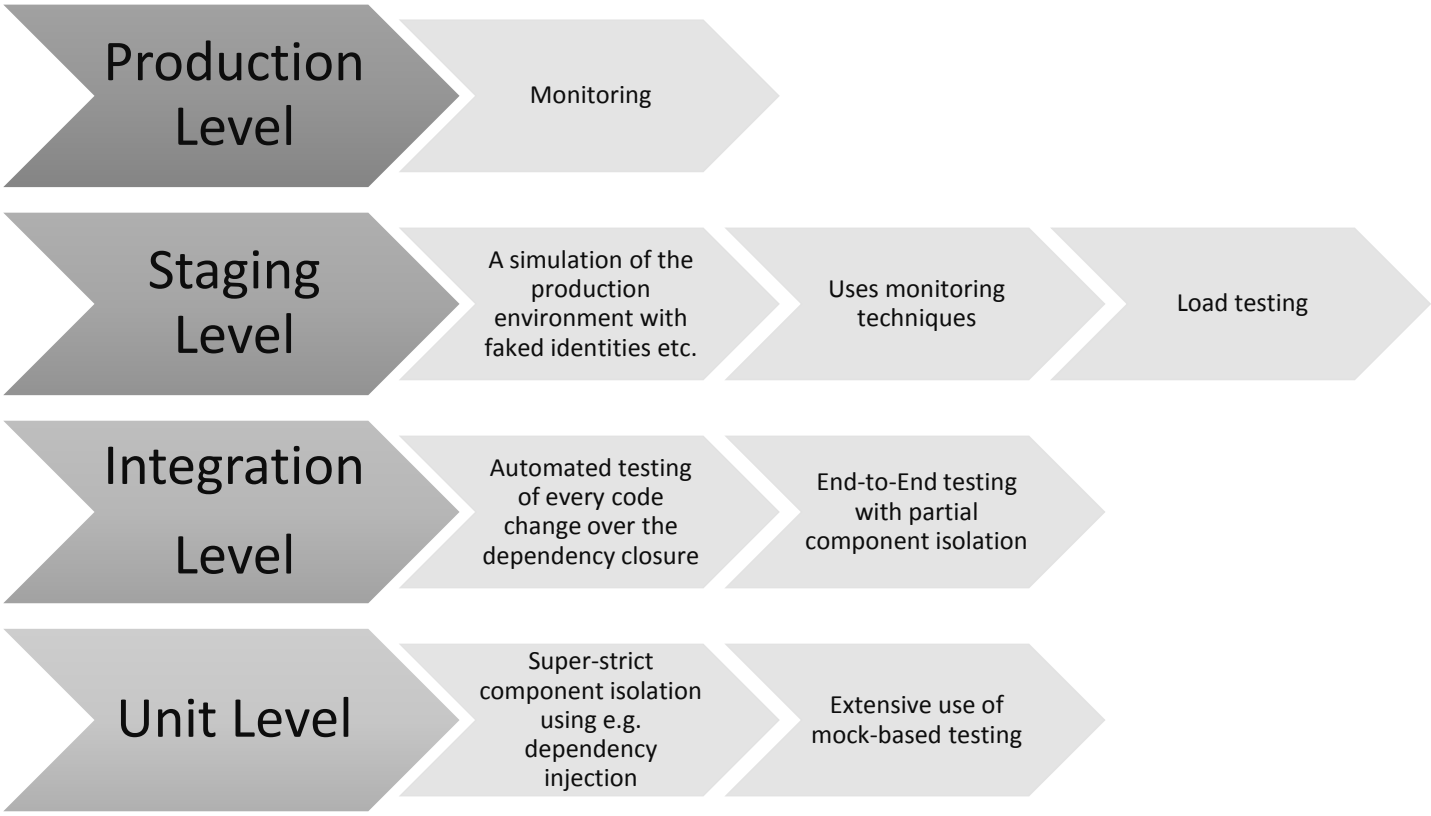
Testers focus on Business Process

FaceBOOK: Staged Data Acquisition



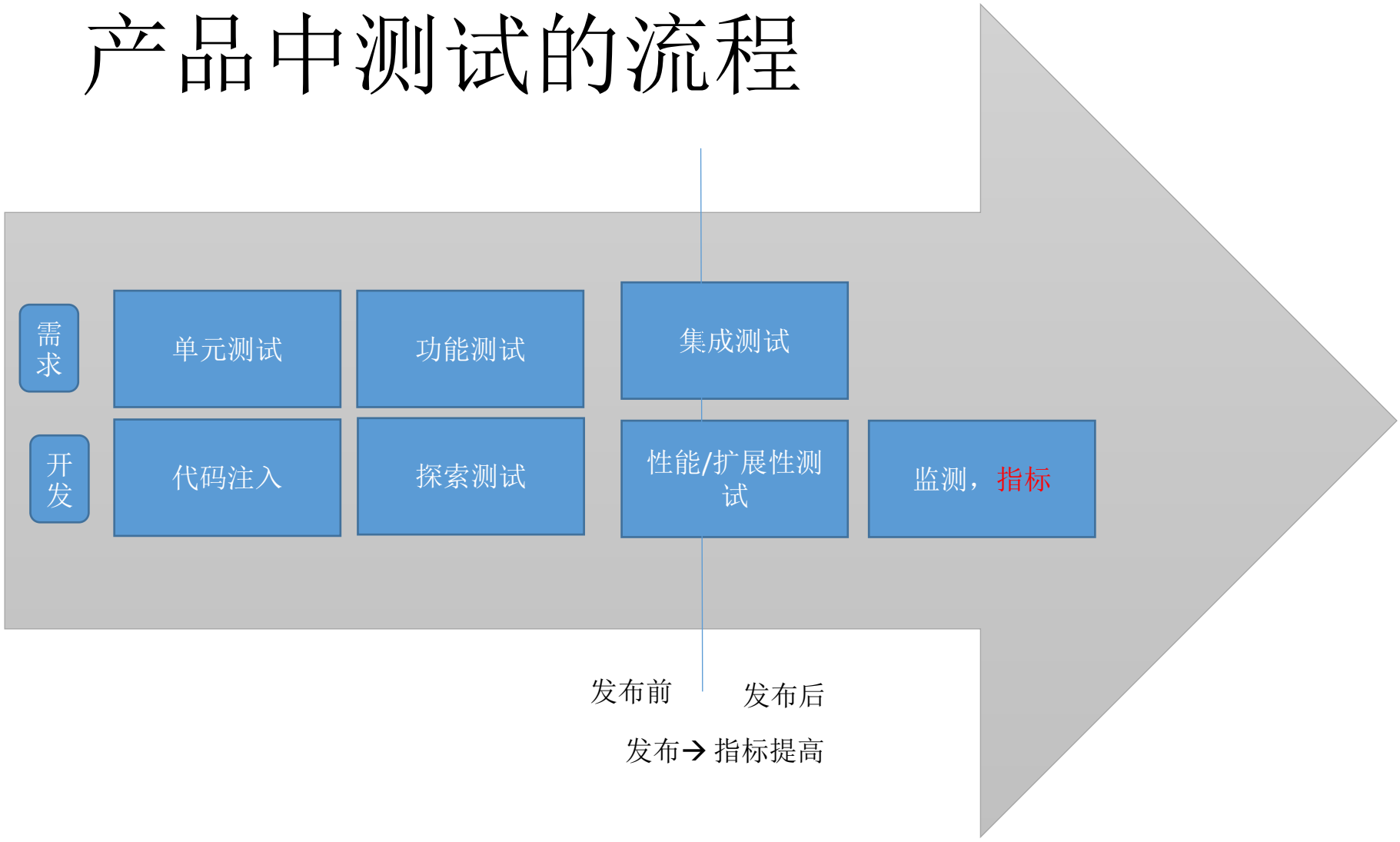
- Dogfood
- In prod, no users (except internal ones)
- Some servers in Production
- World-wide deployment
- Feature flags

Google Runtime Analysis and Testing



By Wolfgang Grieskamp @ Google(MSFT-X

产品中测试的流程



智能硬件+软件+服务

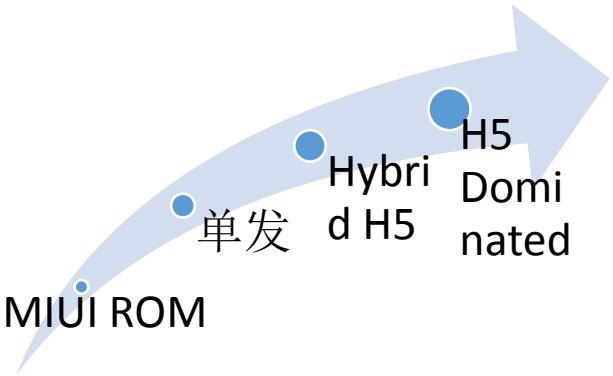
MIUI8

让每个人都能享受科技的乐趣！

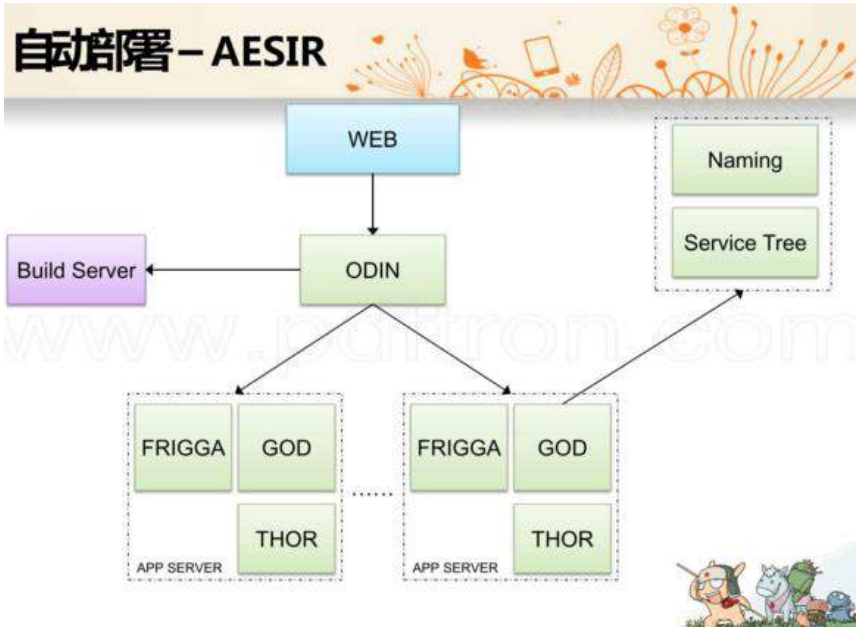
小米MIUI和应用的趋势

- 小米MIUI 的发布周期
  - 体验版：每日发版（几十万内测）
  - 开发部：每周发版（几百万公测）
  - 稳定版：每月发版（全体推送）

- MIUI应用的趋势



小米的自动部署

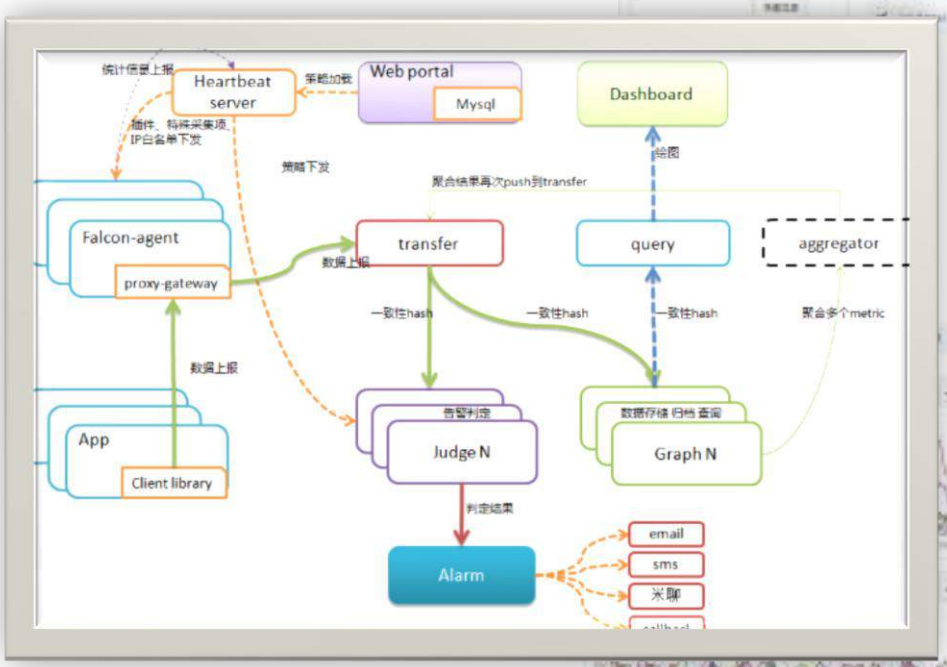


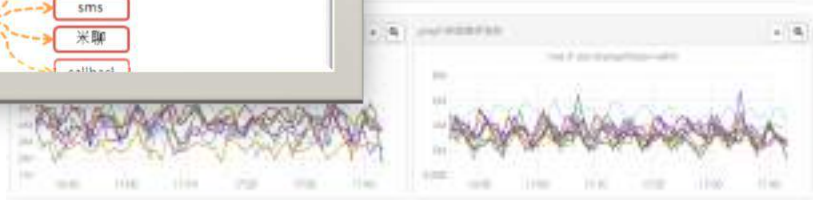
Deploy.YML-→ODIN.YML→Build/Package/Download-→FRIGGA-→Template Build -→ GOD

《运维自动化实践之路》伏晔

小米监控利器：Open-Falco

- 性能
- 易用性
- 扩展性

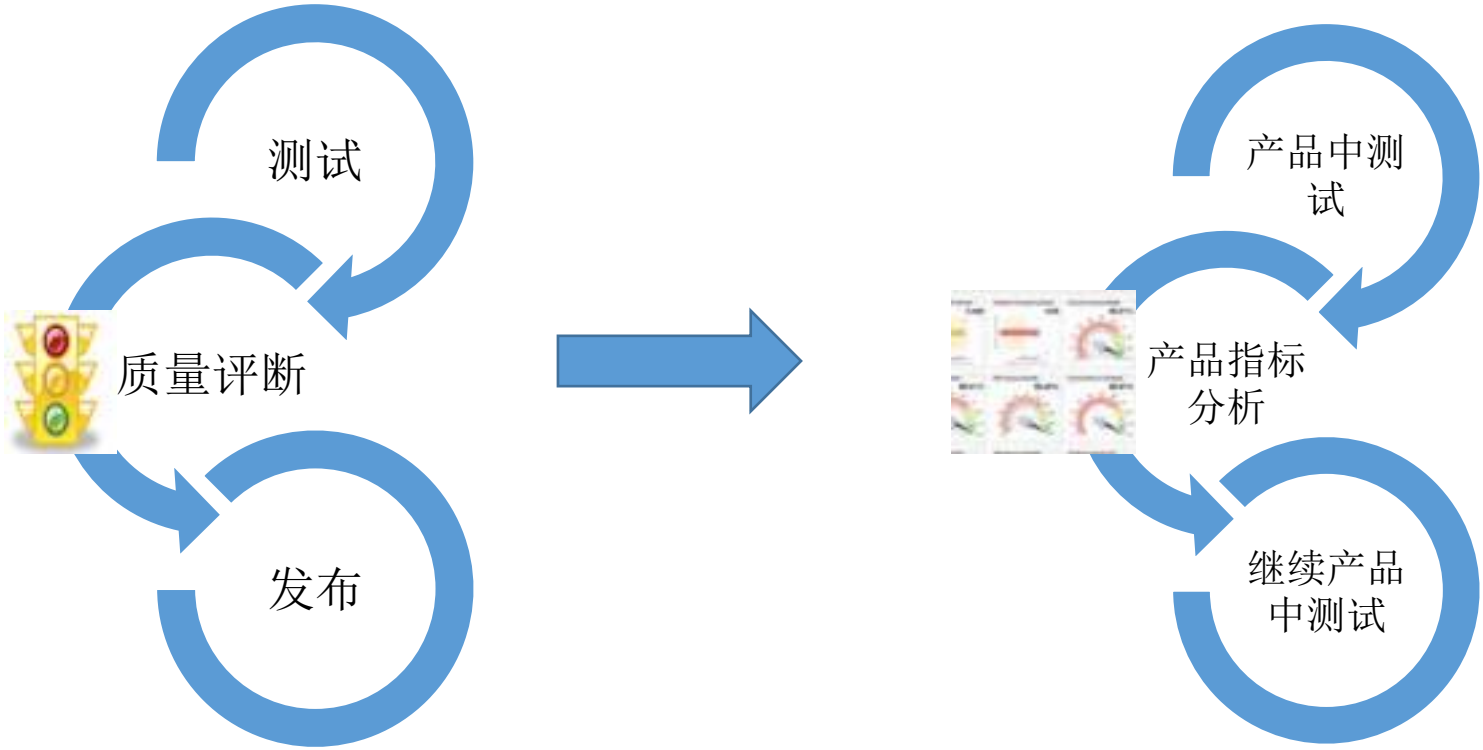




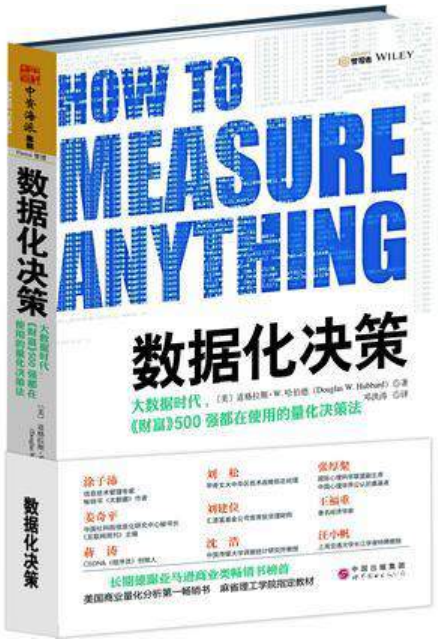
- 报警
- 可视化

http://github.com/open-falcon

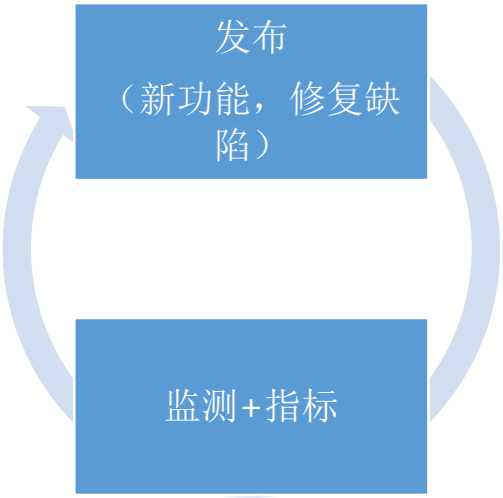
质量评判---》 质量提升



度量的力量

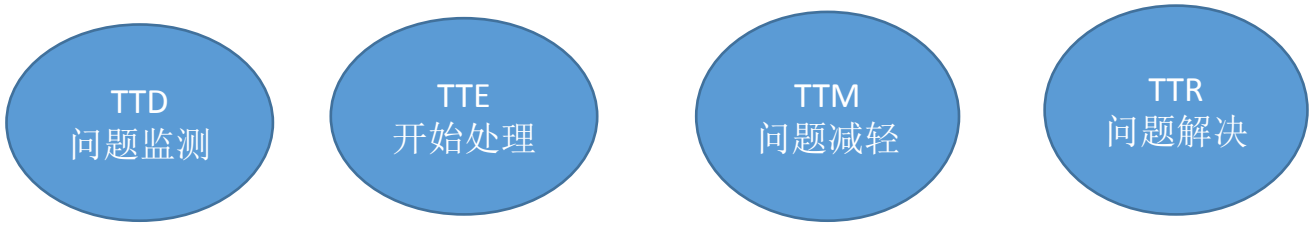


如何量化一切



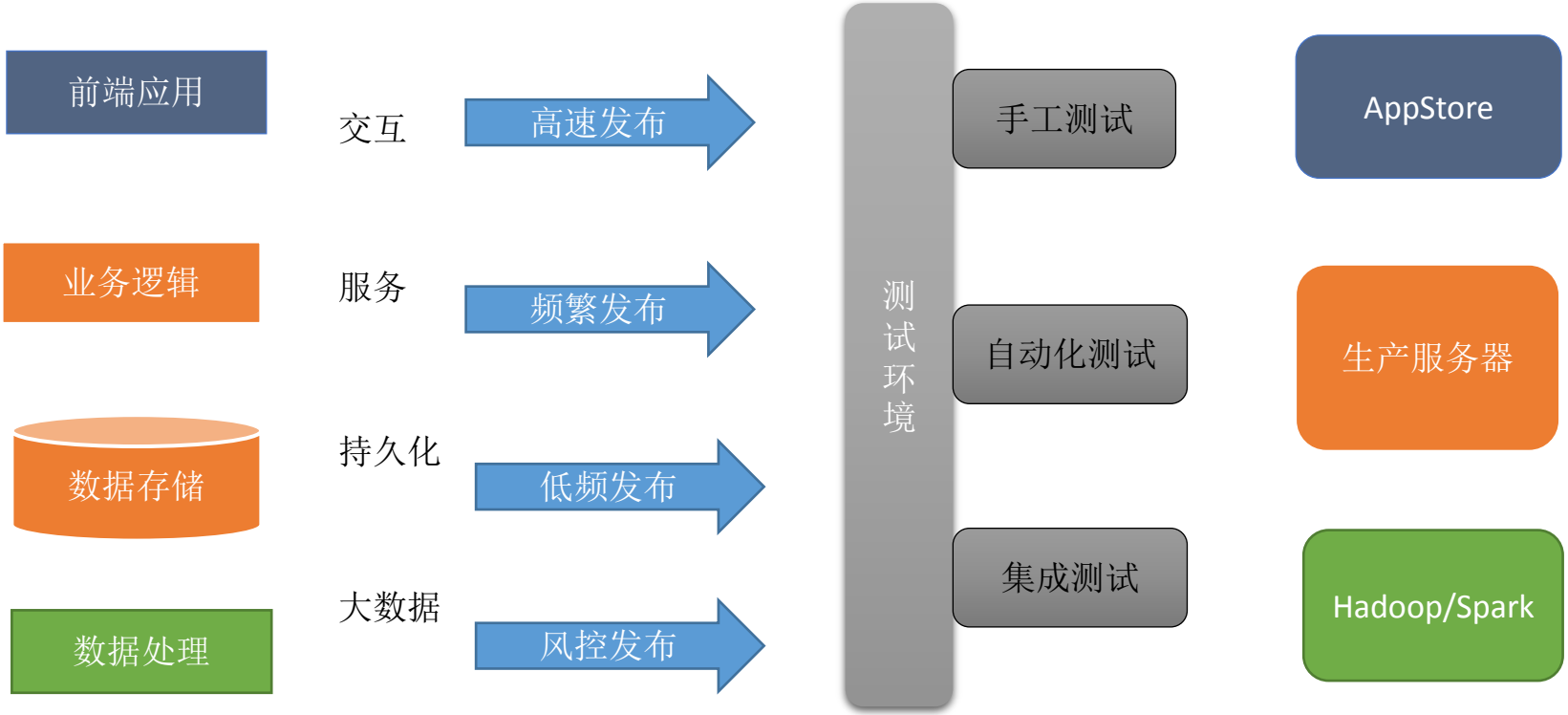
关于产品监控和线上事故的指标

- TTD : Time to detect
- TTE : Time to Engagement
- TTM: Time to mitigate
- TTR : Time to Resolve



- 自动检测
- 人工发现

风险，成本，速度，质量的博弈



DevOps:
基础架构和工具篇

找到要解决的问题，往往比找工具要难很多！

DevOps的基础架构

运维基础架构/云服务

持续集成	部署服务	共享服务
• 代码版本管理 • 分布式编译 • 自动化测试管理 • 反馈管理	• 自动化部署 • 灰度发布 • 滚回管理 • 产品监控，报警 • 日志服务	• 网络存储 • 分布数据库 • NoSQL • DNS/VIP • Zookeeper • 流服务

DevOps的技术支持

- 开发
  - 代码库(Git)
  - 代码评审(**Phabricator，Crucible)**
  - **集成测试/持续集成**（Jenkins)
- 部署
  - 应用部署（Puppet，Chef，Ansible）
- 监控
  - Zabbix
  - Open-Faclon
  - Jiankongbao.com
  - OpenTSDB

微软：Autopilot

- 自动修复机器
- 可编程的监控
- Everything is code

谷歌：Borg/Omega

Twitter:Mesos

sonarqube

Nexus

maven

Jenkins

JIRA

git

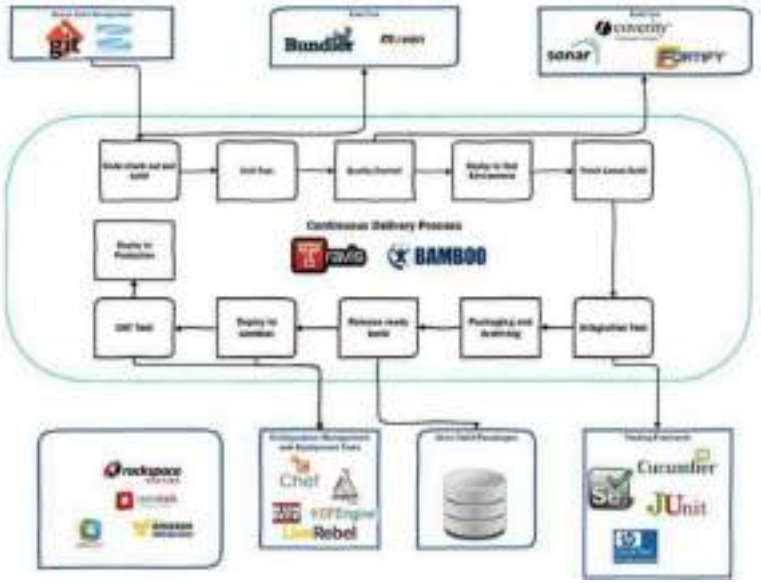
puppet labs

RUNDECK

CHEF

docker

Apache Tomcat



总结一下观点

我对DevOps的4个观点

- 竞争性软件公司终将采用DevOps
- DevOps不是银弹，是关于效率的文化，自动化的技术。
- 大量专职运维和测试将消失和减少
- “靡不有初，鲜克有终”

我理解DevOps的3大核心问题

三大核心问题

- 自动化发布：灰度发布(Staged Deployment)
- 自动化监控：产品监控和报警
- 自助PaaS（基础平台服务）:存储，容器，队列，认证等

两个非核心问题

- 持续集成CI（Build->Test）
- 持续交付CD (Ready for deployment)

DevOps和适应变化

- 动物学家达尔文:”世界上进化下来的动物，并不是那些最强大的动物，也不是那些最聪明的动物，而是那些最能够适应变化(Responsive to change)的动物”，例如说老鼠，人，蚂蚁等等。软件系统也一样，能够传承发扬的软件，能够快速适应变化化。

什么样的流程方法能够存活下来？

持续集成

详细测试需求说明书

UI Automation

软件成熟度模型

成对编程

敏捷开发 Agile Developement

探索性测试

TDD

道阻且长，行则将至

不忘初心，方得始终

个人主页：www.ouyangchen.com
知乎专栏：zhuanlan.zhihu.com/ouyangchen

“互联居”

广告架构，大数据和观察

“致力于中国互联网广告技术的繁荣”！

Gdevops

全球敏捷运维峰会

THANK YOU！