

The top corners of the slide feature decorative geometric shapes, including a network of blue dots connected by lines and a dark blue sphere with a grid pattern.

Gdevops

全球敏捷运维峰会

The bottom corners of the slide feature decorative geometric shapes, including a network of blue dots connected by lines and a dark blue sphere with a grid pattern.

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



7年



3年

高级开发经理/工程师



10年

架构师/主管



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



小米手机

红米

平板

电视·盒子

路由器

智能硬件

服务

社区

手机 平板 电话卡 >

电视 盒子 >

路由器 智能硬件 >

移动电源 插线板 >

耳机 音箱 >

电池 存储卡 >

保护套 后盖 >

贴膜 其他配件 >

米兔 服装 >

箱包 其他周边 >



路由器



路由器 mini



路由器 青春版



九号平衡车



净水器



水质检测笔



手环及配件



空气净化器



空气净化器 2



体重秤



摄像机



除甲醛滤芯



血压计



运动相机



WiFi放大器



智能家庭套装



床头灯



全部智能硬件

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?

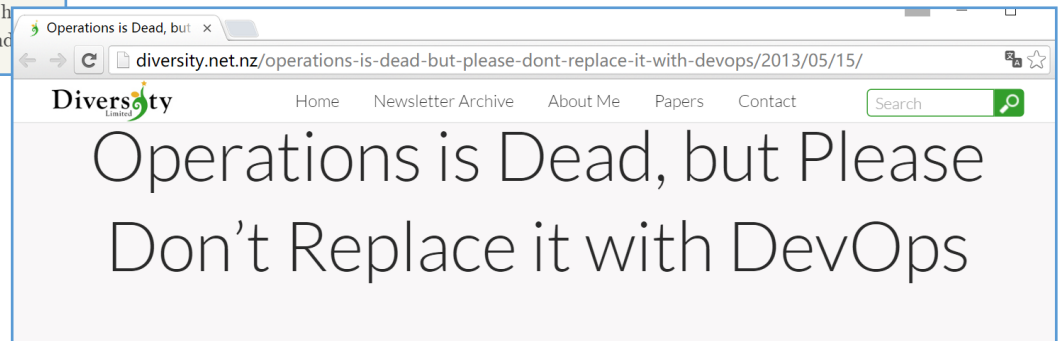
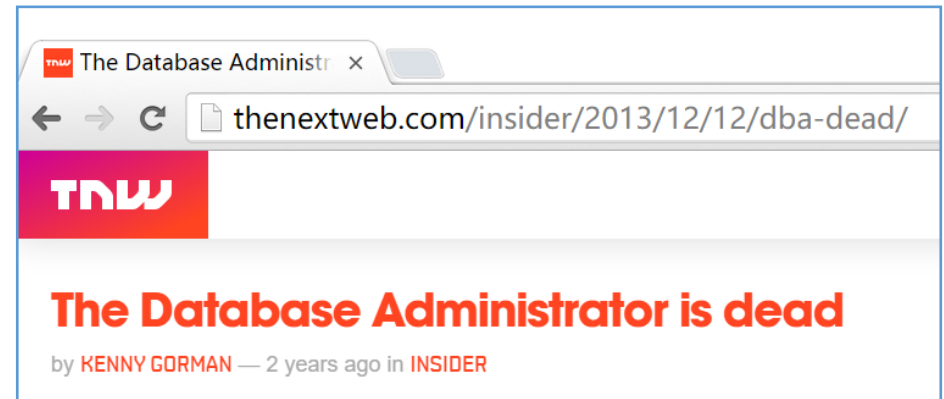
TDD is dead. Long live testing.

By David Heinemeier Hansson on April 23, 2014

Test-first fundamentalism is like abstinence-only sex ed: An unrealistic, ineffective morality campaign for self-loathing and shaming.

It didn't start out like that. When I first discovered TDD, it was like a courteous invitation to a better world of writing software. A mind hack to get you going with the practice of testing where no testing had happened before. It opened my eyes to the tranquility of a well-tested code base, and the bliss of confidence it grants those making changes to software.

The test-first part was a wonderful set of training wheels that taught me how to think about testing at a deeper level, but also some I quickly left behind.



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

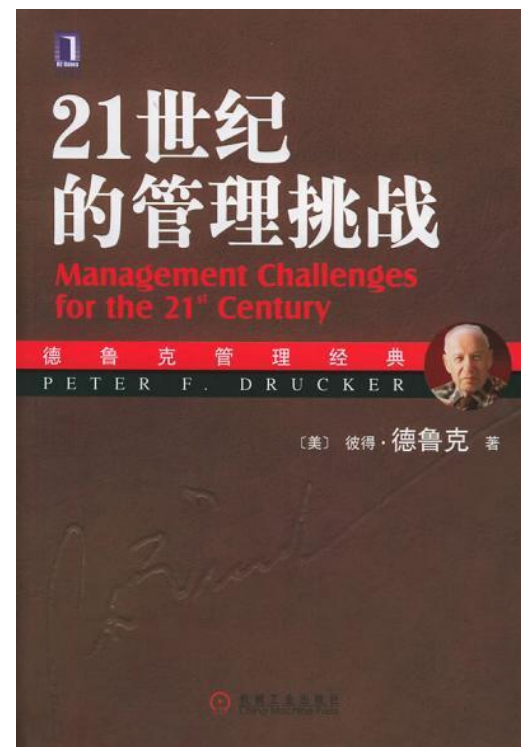
- 质量是用户体验，提供用户价值
- 质量是经济竞争性
- 质量
 - 代码覆盖率
 - 用例自动化率
 - 缺陷数量分布
 - 响应时间



软件 → 服务

知识工作者的生产率

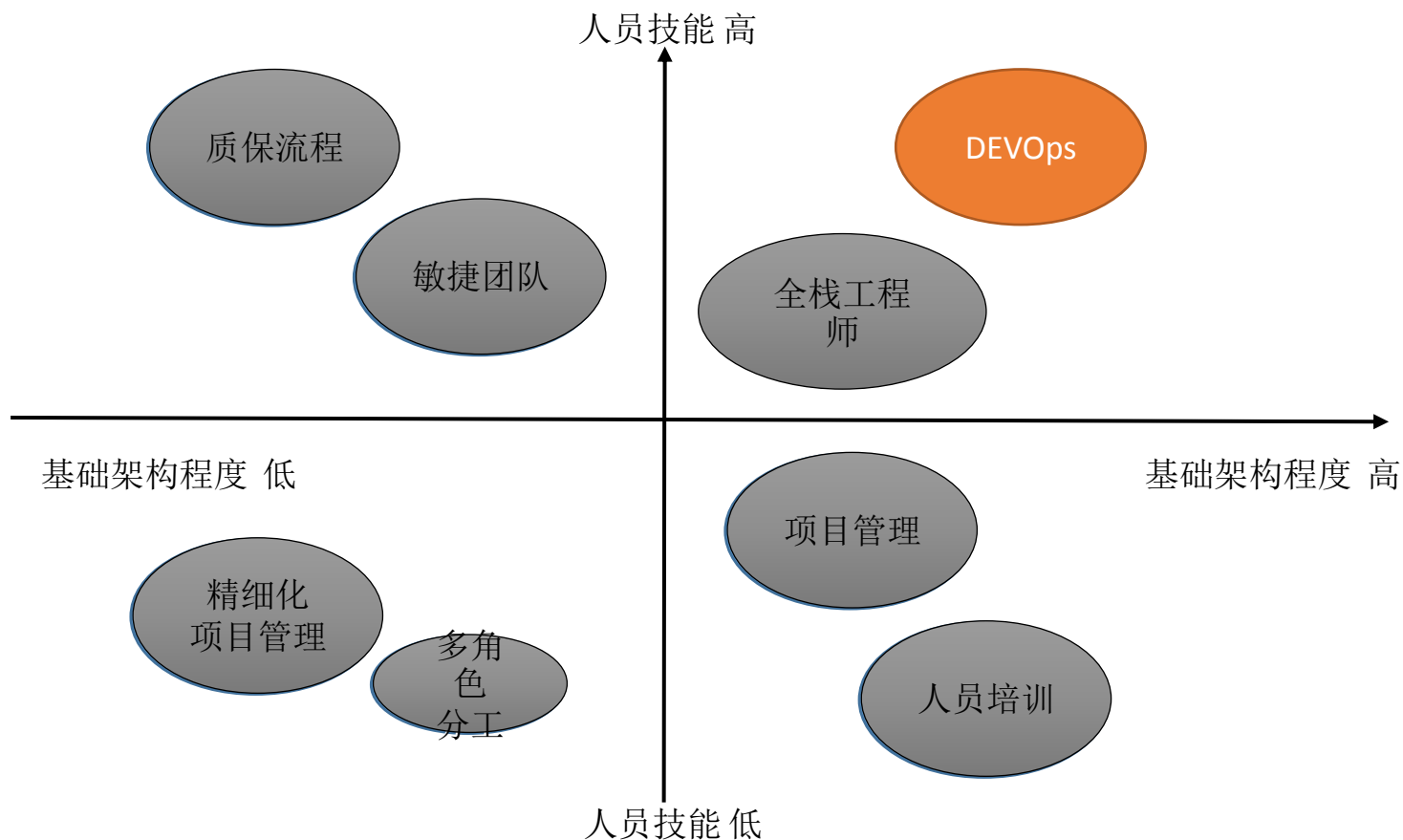
- 搞清楚知识工作者的任务最为关键
- 自我管理生产率
- 不断创新
- 不断学习和教导
- 质量重于数量
- 知识工作者为资产

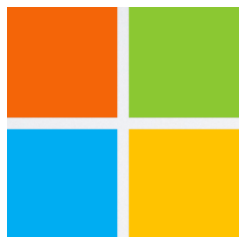


我理解的DevOps！

- 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



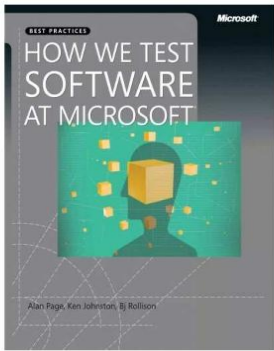


微软的测试 转型

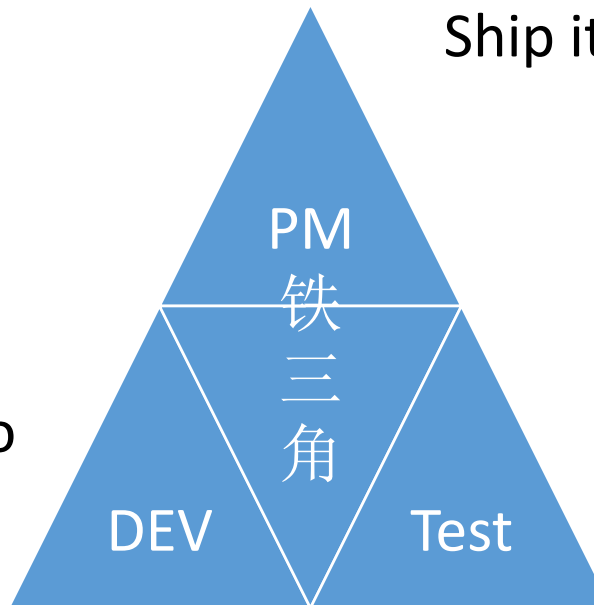
Combined Engineering

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！



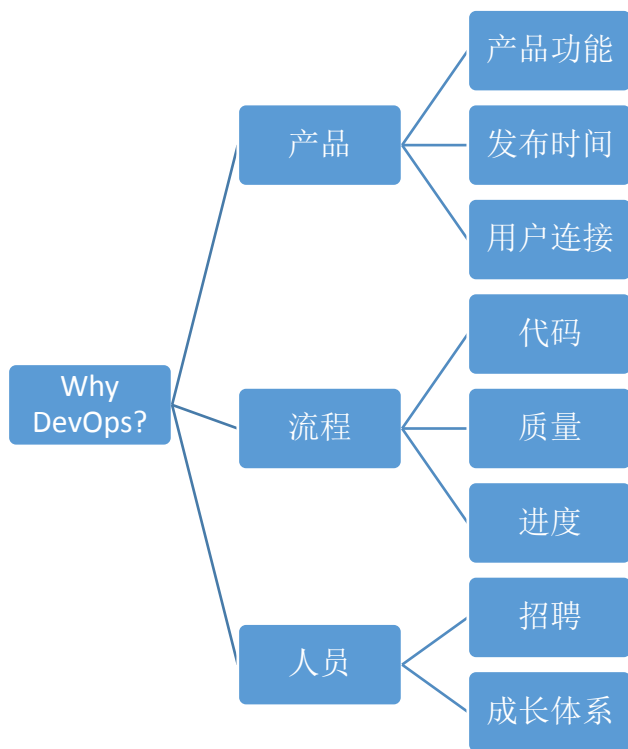
Implementation!



Ship it!

Quality Feedback!

测试转型前的一些问题



低MTBIAMSH

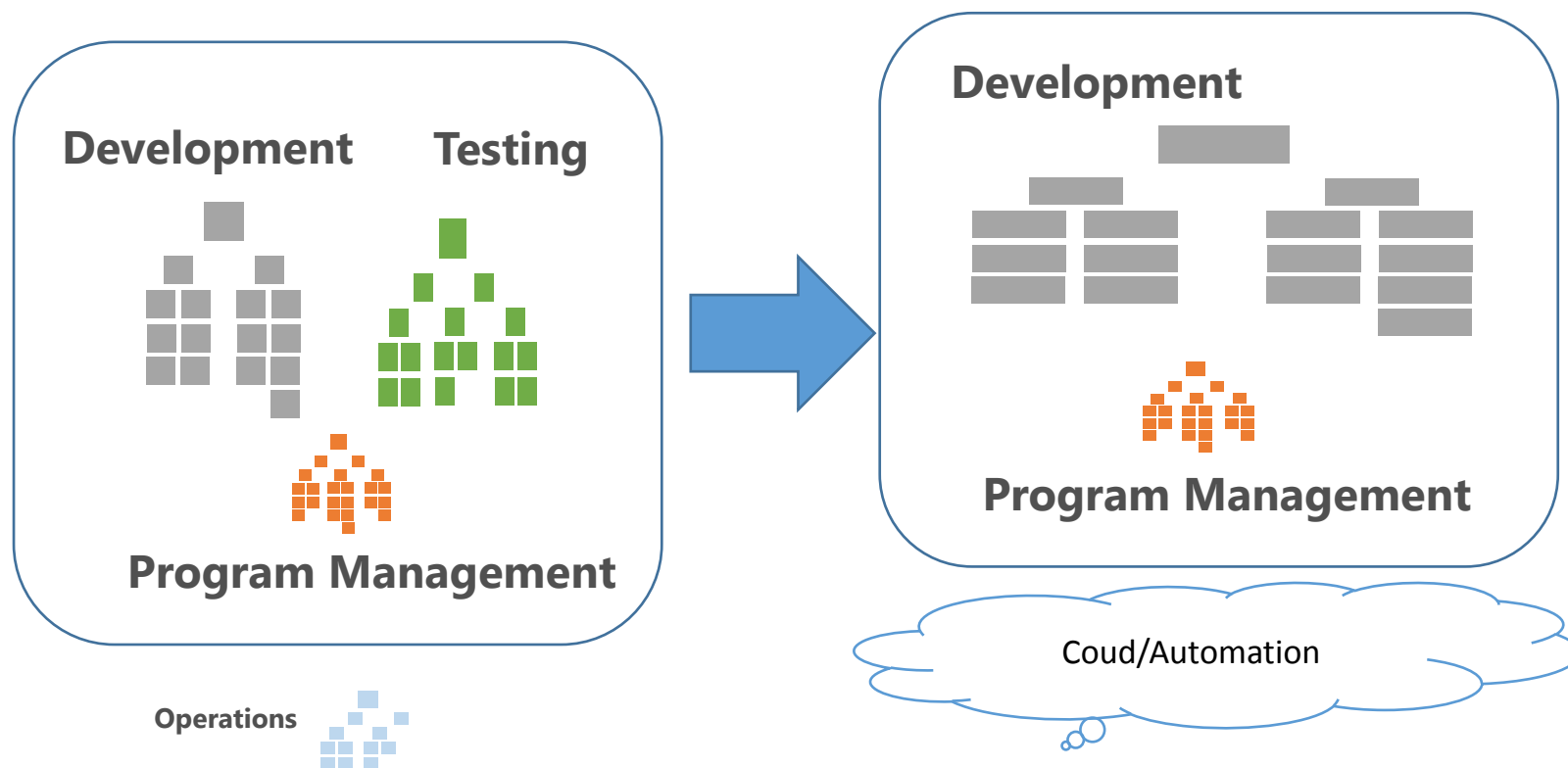
MTBIAMSH (Mean Time Between Idea And Making Stuff Happen)

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

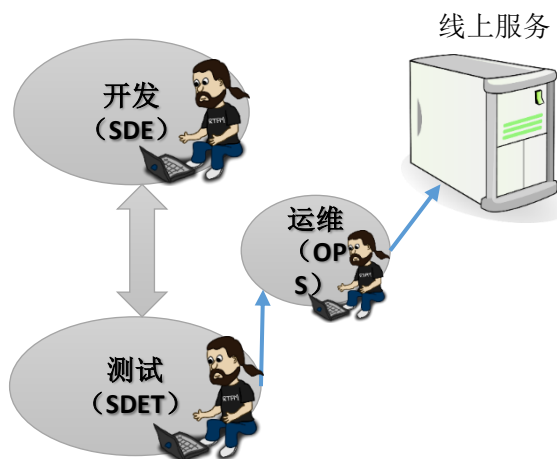
组织的变化：减少角色，管理扁平



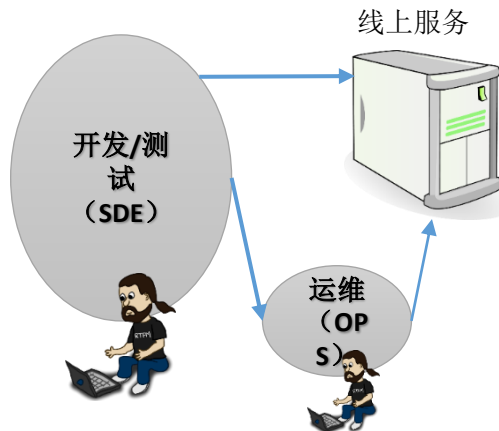
小组：

- 10-12人
- 自管理
- 12-18月
- 1个产品/其他都是开发

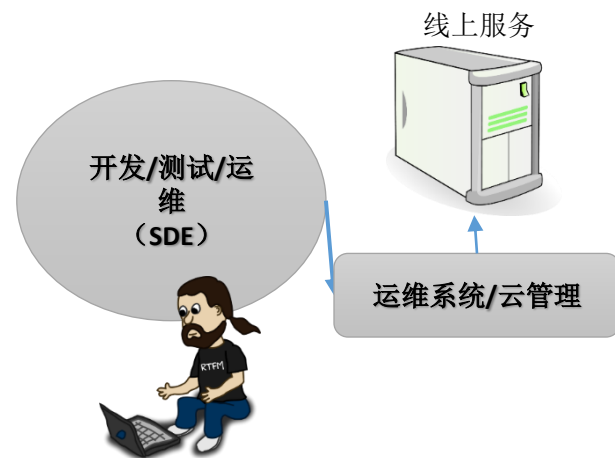
微软的DevOps之旅



阶段一（2008以前）



阶段二（2009~2013）

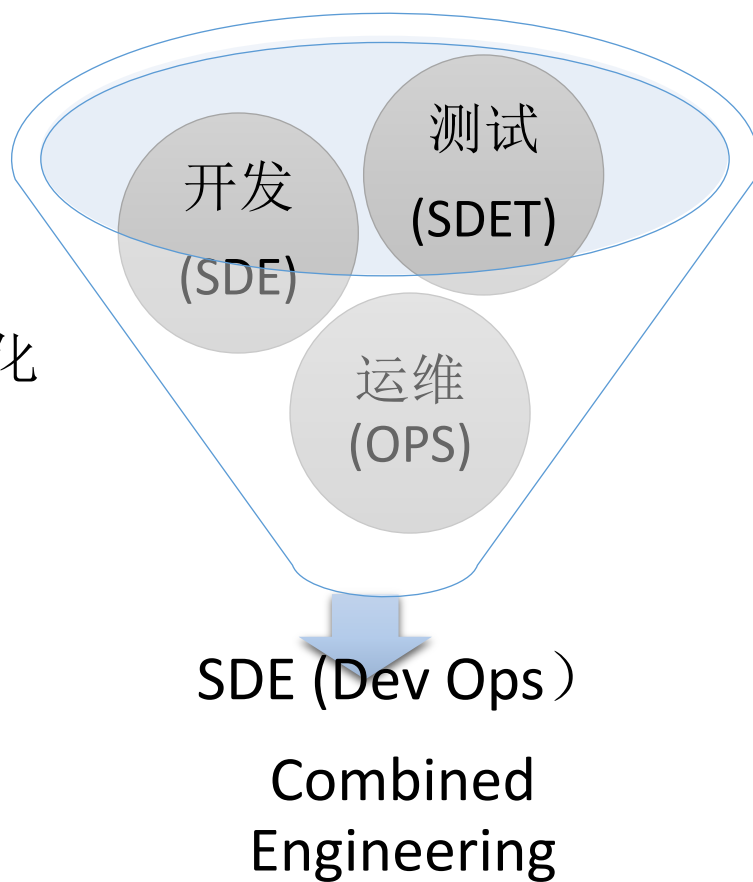


阶段三（2013~）

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



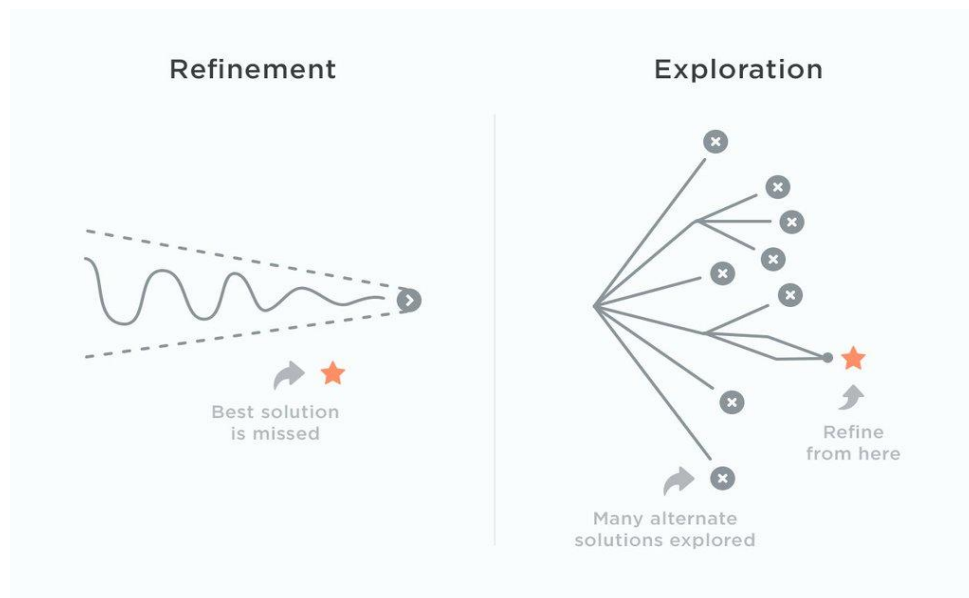
一些任务和角色的转移

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

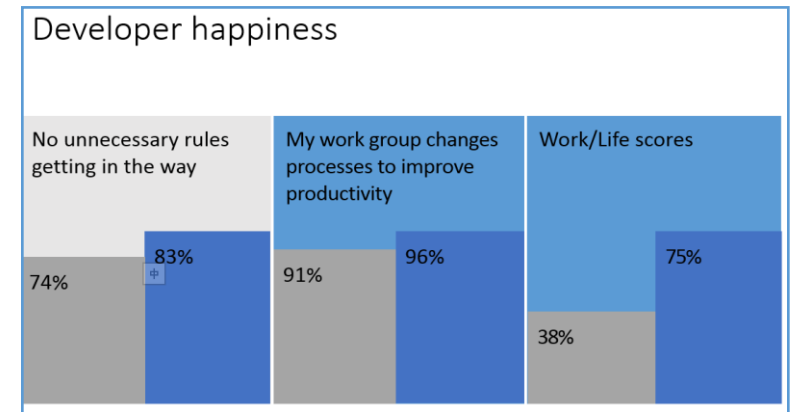
- 主动探索
- 数据驱动
- 用户为中心
- 极客文化
- Show me the code



越来越少的专业测试讨论 ☹

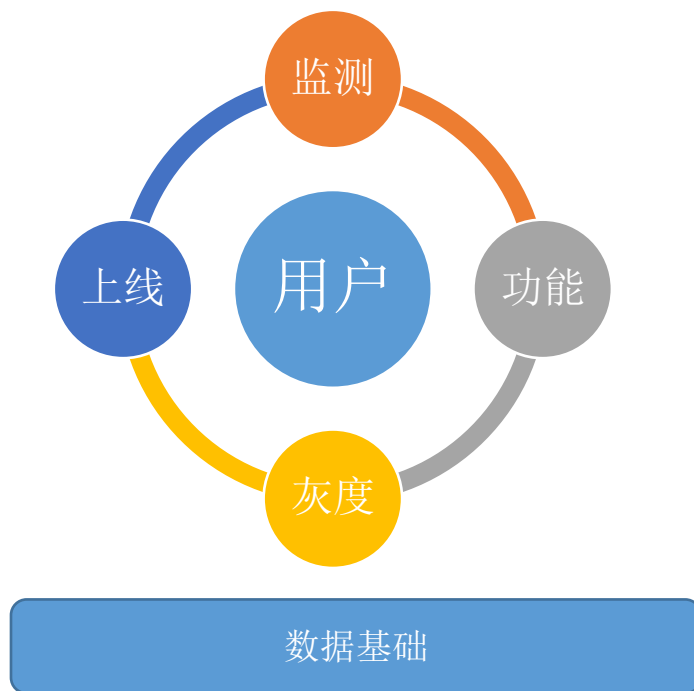
Combined Engineering 效果(类 DevOps)

- 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

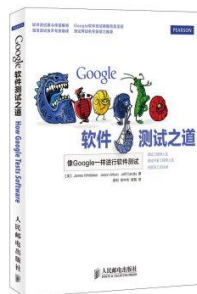


Combined
Engineering



No Testers!

- 单元测试
- 灰度为王



.0: 1

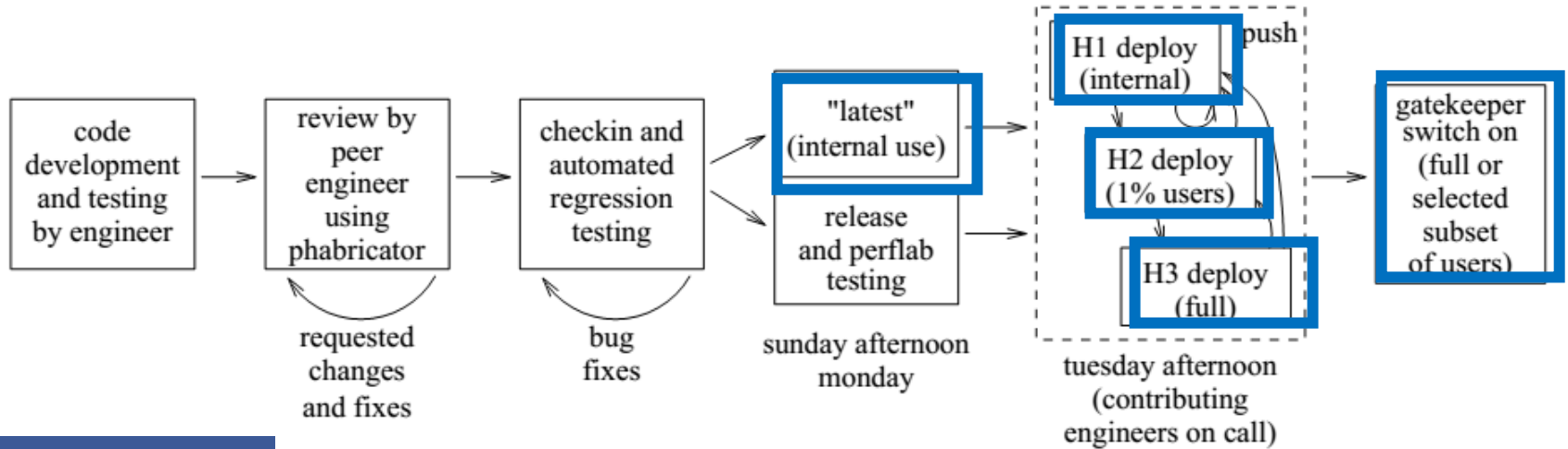


Engineering Productivity "We are small and elite Special Forces that have to depend on superior tactics and advanced weaponry to stand a fighting chance at success."



Testers focus on
Business Process

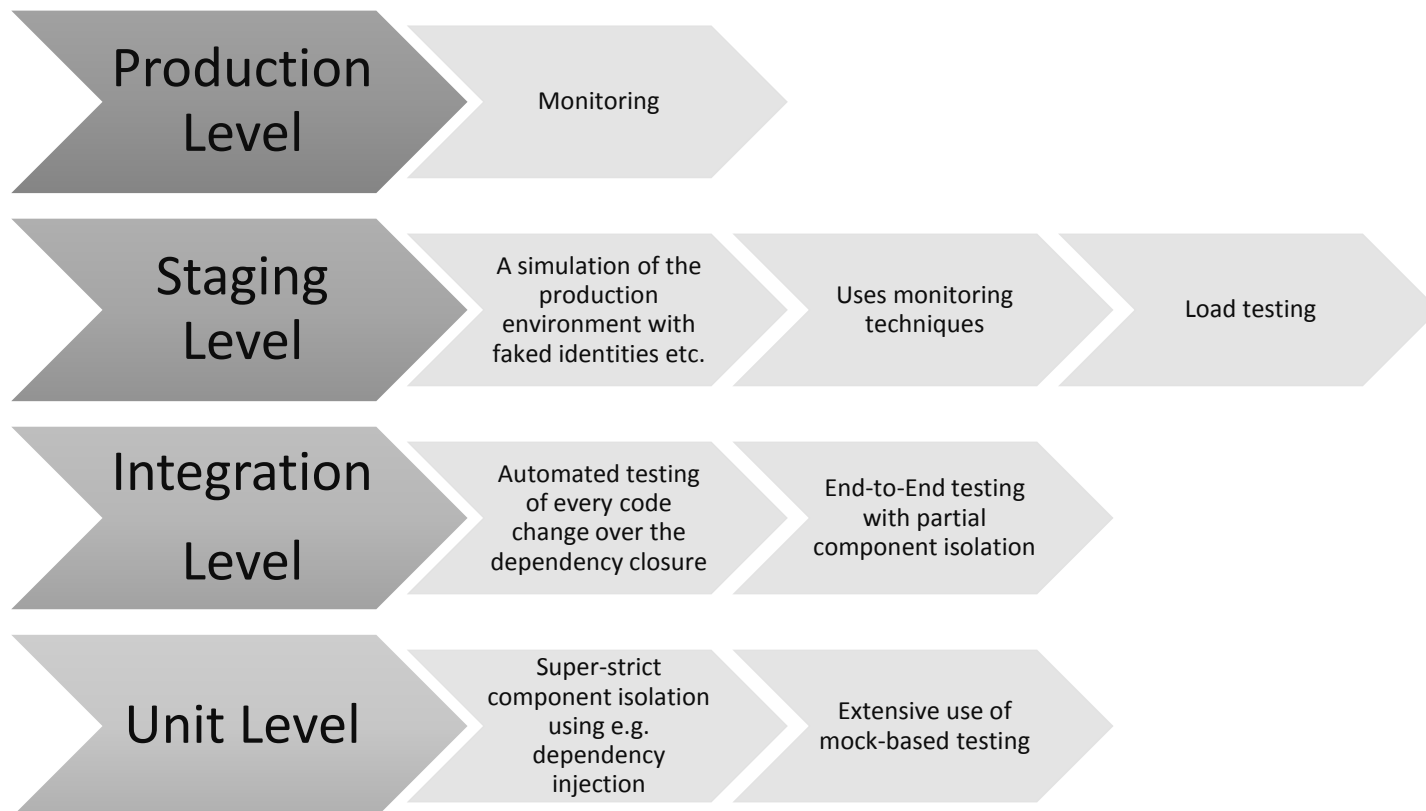
FaceBOOK: Staged Data Acquisition



facebook.

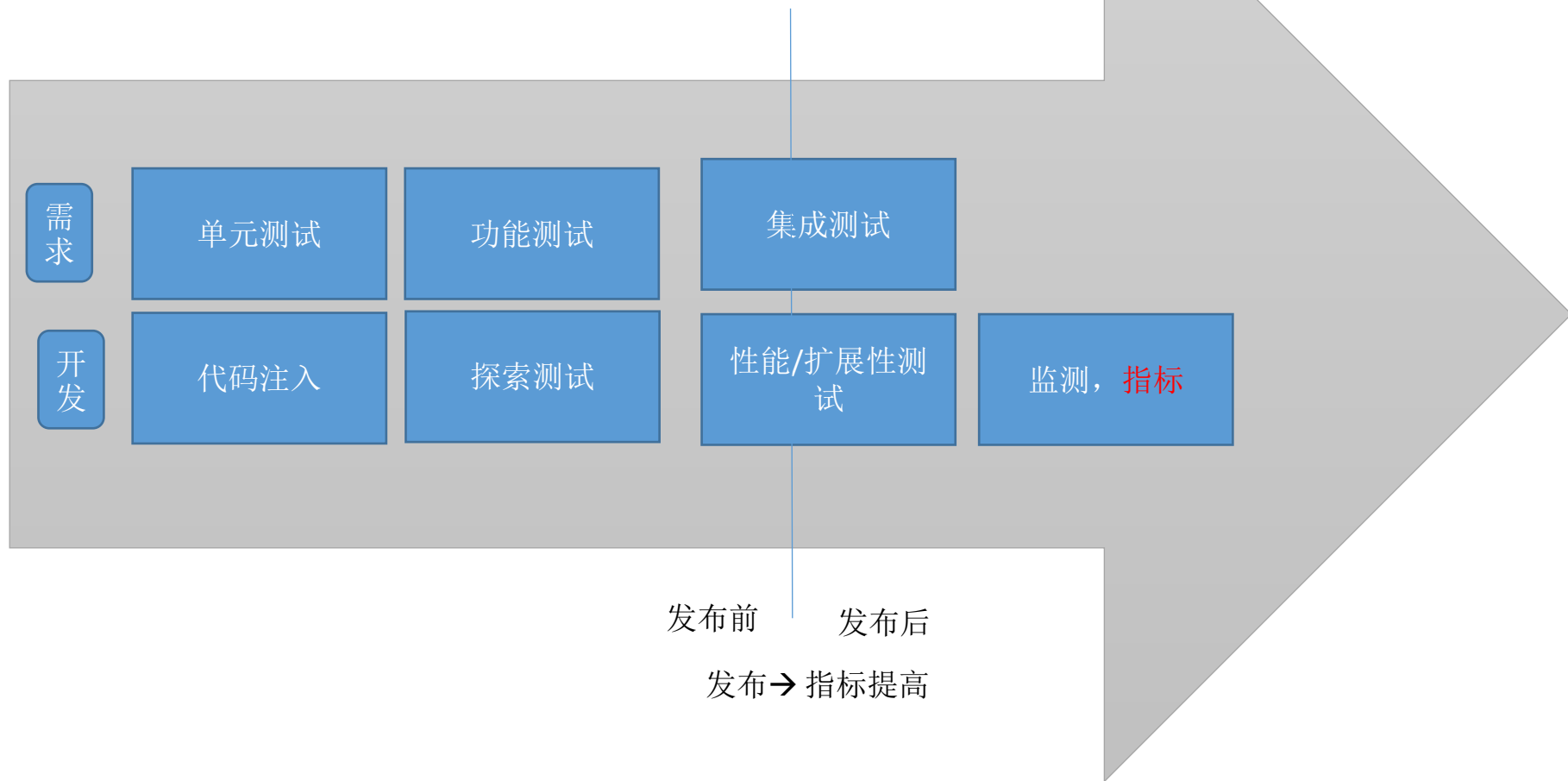
- Dogfood
- In prod, no users (except internal ones)
- Some servers in Production
- World-wide deployment
- Feature flags

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



让每个人都能享受科技的乐趣！

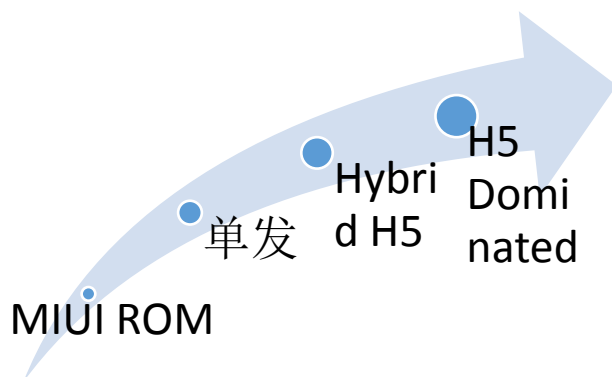
小米MIUI和应用的趋势



- 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



MIUI 第260周发布公告及更新日志

扫地僧 | 2015年10月30日

MIUI更新 第260周

【本周推荐】

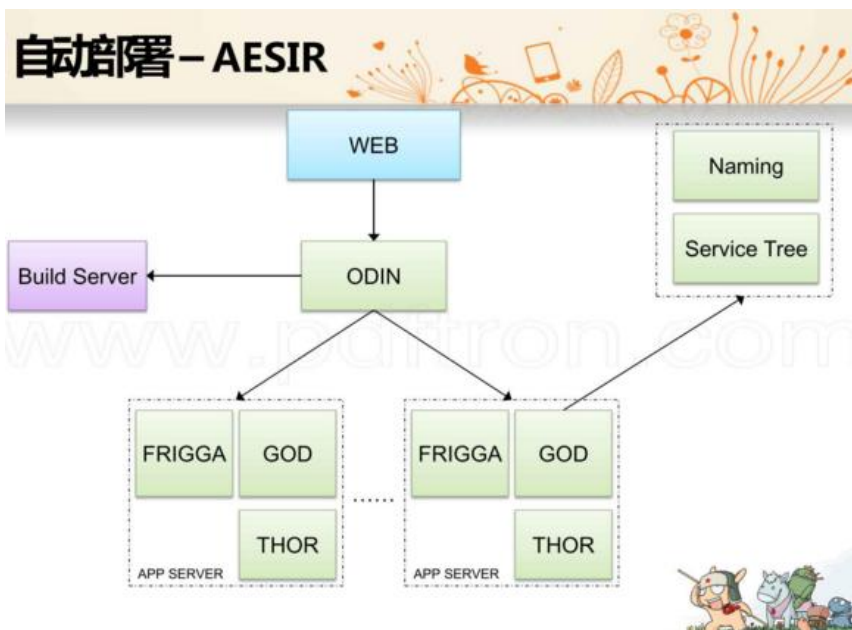
【更新推荐】

新增 新版全局搜索

新增 支持在安全中心中添加“网速详情”快捷方式，可查看各个应用的实时网速详情

查看全文

小米的自动部署



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

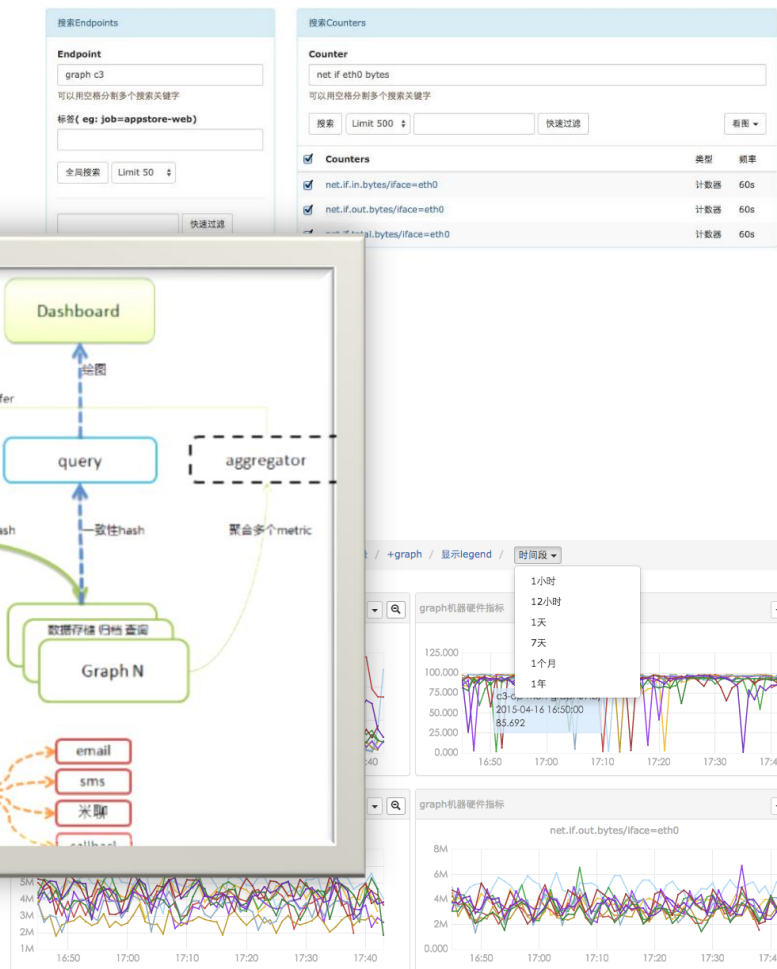
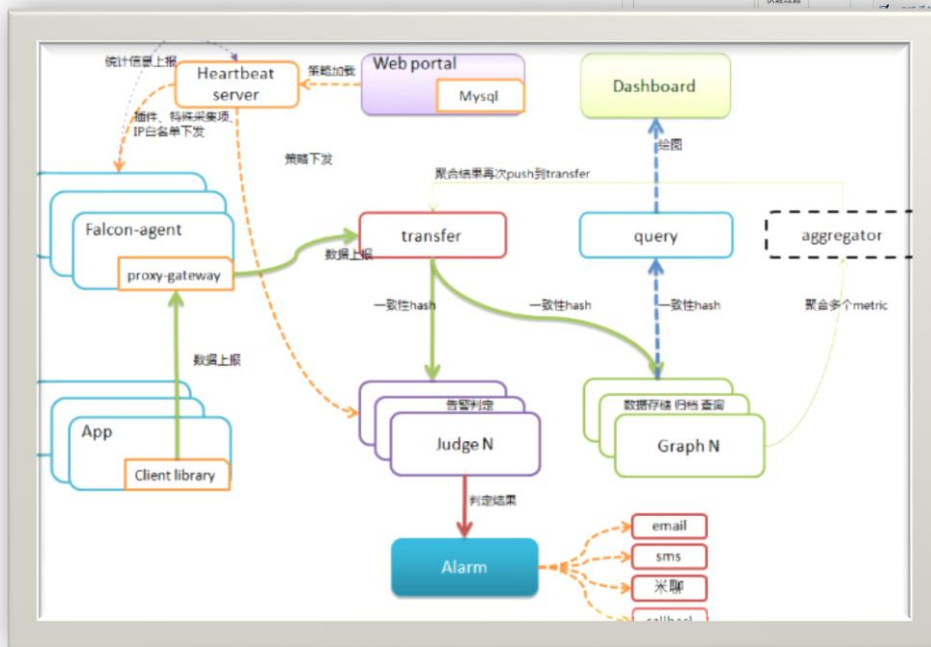
《运维自动化实践之路》[伏晔](#)

小米监控利器：Open-Falco

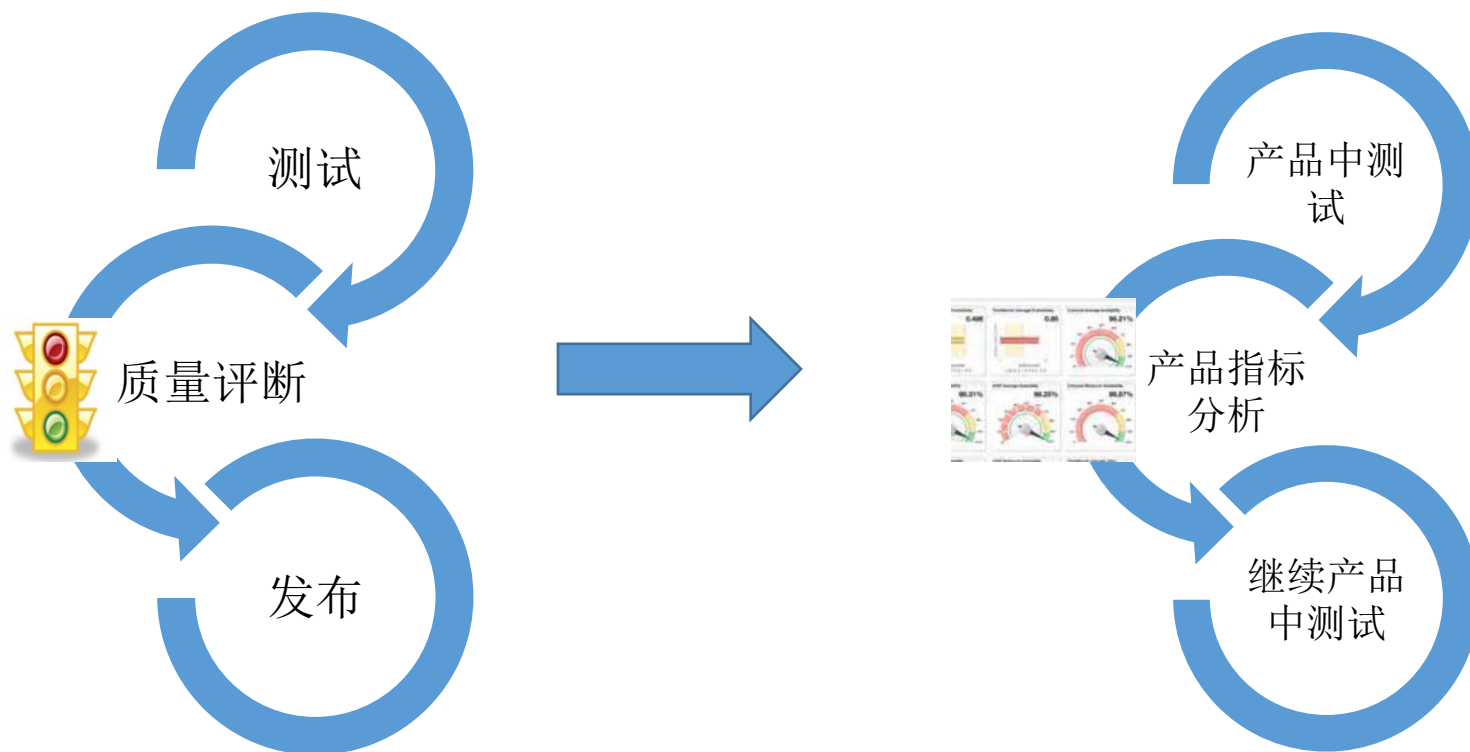
- 性能
- 易用性
- 扩展性

- 报警
- 可视化

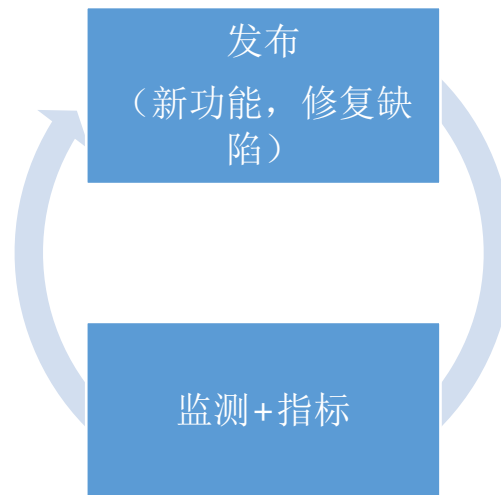
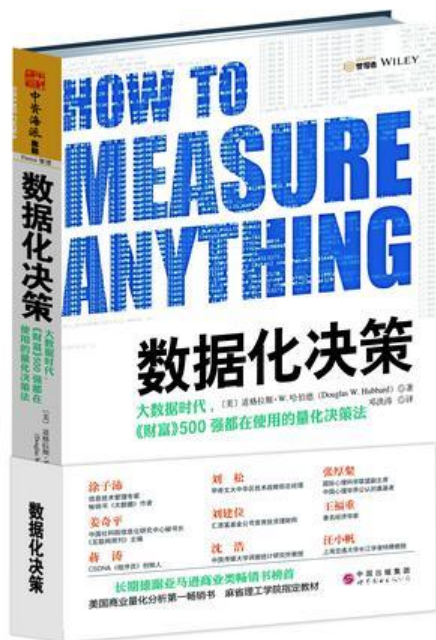
<http://github.com/open-falco>



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

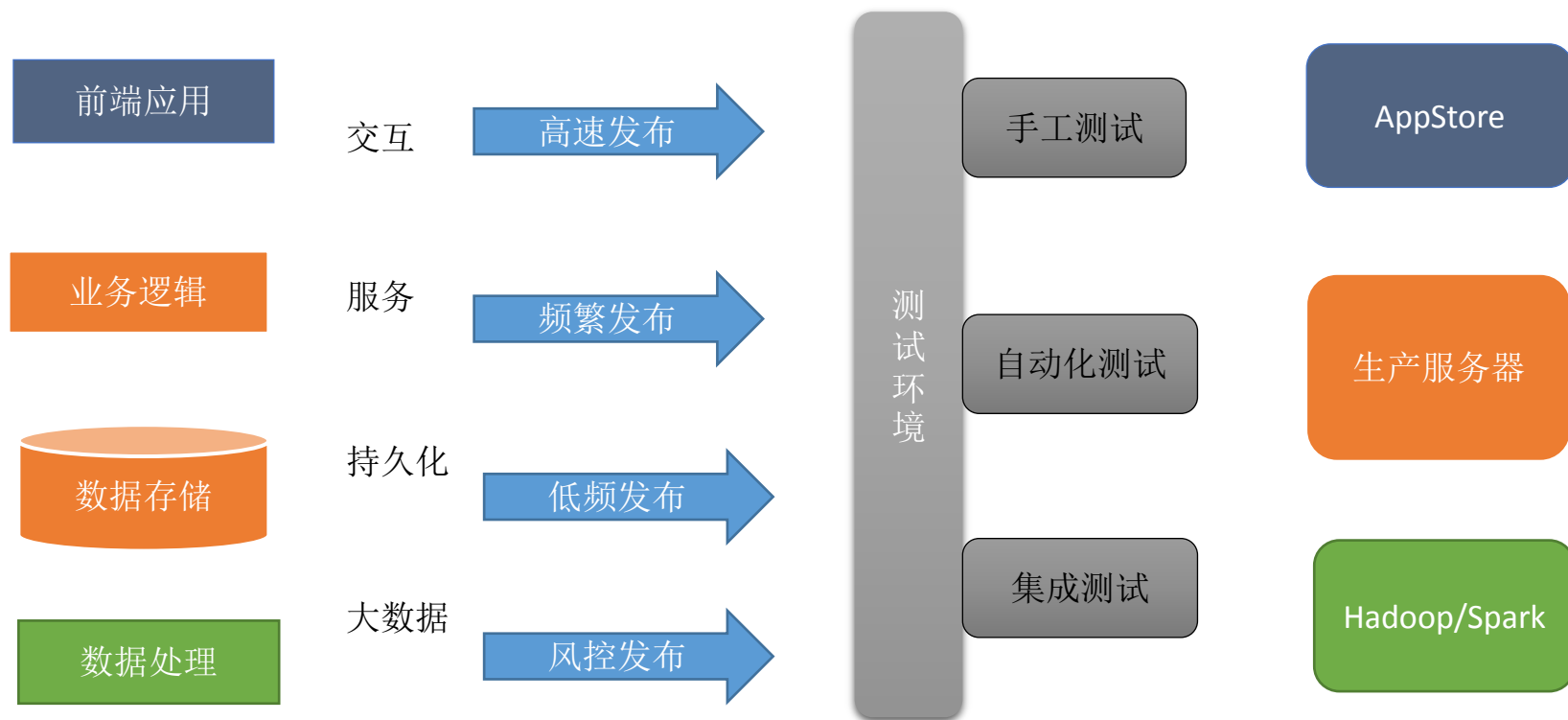
- TTD : Time to detect
- TTE : Time to Engagement
- TTM: Time to mitigate
- TTR : Time to Resolve



- 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

- 代码版本管理
- 分布式编译
- 自动化测试管理
- 反馈管理

部署服务

- 自动化部署
- 灰度发布
- 滚回管理
- 产品监控，报警
- 日志服务

共享服务

- 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

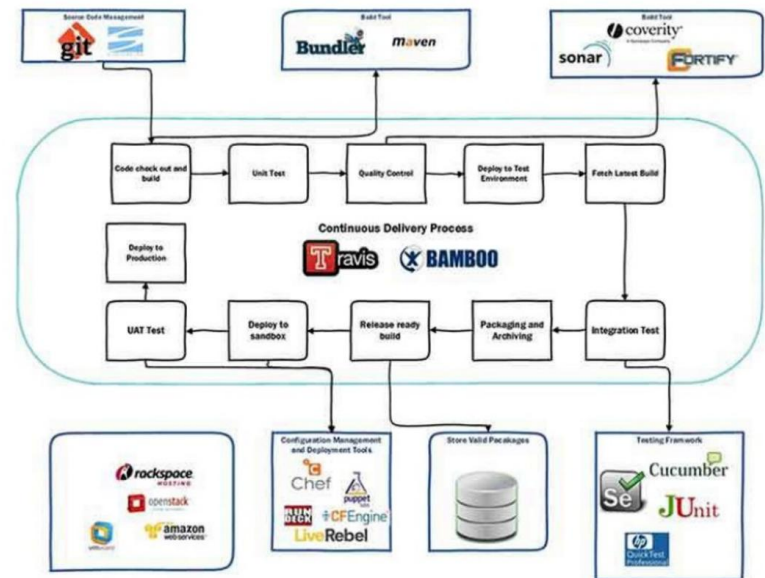
RUNDECK

CHEF

docker

STRATASIS

Apache Tomcat



Take it
away

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

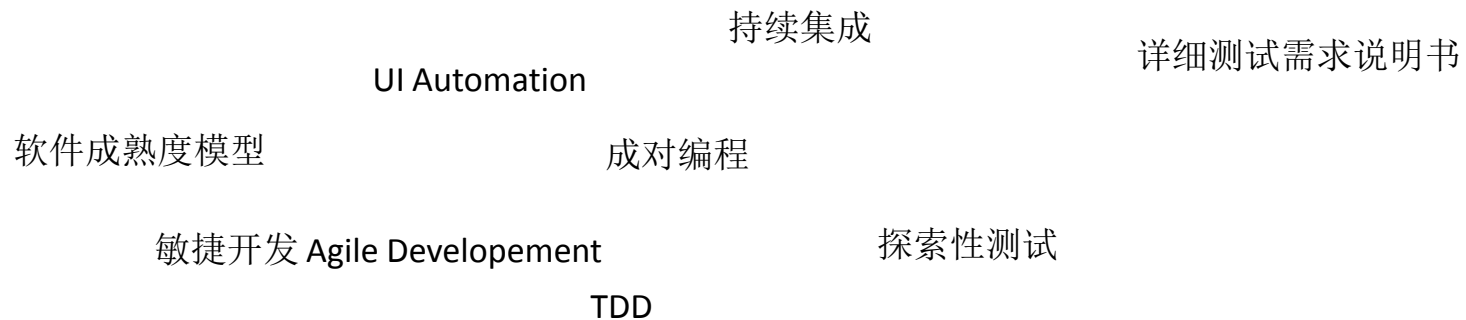
两个非核心问题

- 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？



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”！

The top corners of the slide feature decorative geometric patterns. On the left, there is a dark blue shape with white dots and lines. On the right, there is a similar pattern in a lighter blue shade. In the center, above the 'Gdevops' text, is a white outline of an upward-pointing triangle.

Gdevops

全球敏捷运维峰会

The bottom corners of the slide feature decorative geometric patterns. On the left, there is a dark blue shape with white dots and lines. On the right, there is a similar pattern in a lighter blue shade. In the center, below the 'THANK YOU !' text, is a white outline of an upward-pointing triangle.

THANK YOU !