



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2017

团队和工程管理的取舍

有赞 / 崔玉松

分享内容

1. DevOps的实践
2. 自研底层软件实践
3. 过程改进的实践
4. 团队构建的实践

故事背景

有赞是一个商家服务公司，
我们通过产品和服务，
帮助互联网时代的生意人管店、管货、管客、管钱，让生意更好做。



700+



60% 产品技术



杭州 深圳 北京

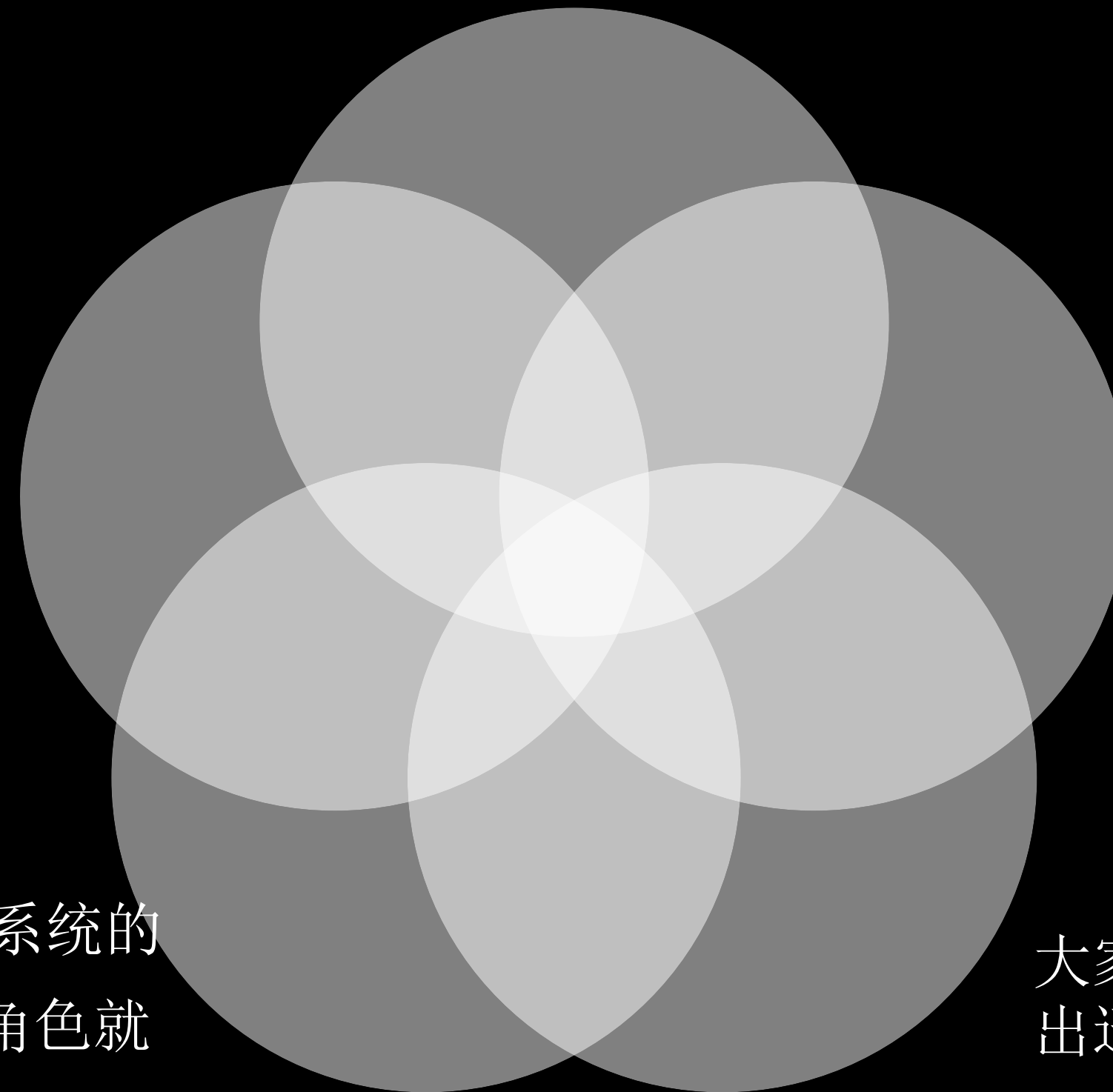
1.DevOPS?

我们以为的 DevOps

敏捷，不要跟更多的角色沟通

业务研发是万能的
肯定能搞定系统的事情

大家乐于干更底层的事情借机拓展自己的能力



运维就是抗机器、接网络、装系统的
使用公有云之后这个角色就

大家都是愿意接受告警信息并且能做出迅速响应的

不需要了

于是

技术团队过百人了，一个运维都没有

给每个人开账户登录生产环境，所有线上问题自己解决，包括系统部署

鼓励大家学习Linux运维相关技能，每个人都有sudo权限

招人的时候考核非常全面，对于Linux的要求是硬性的

一天发布几百次，每个人随时发布，随时修复问题

线上直接调试，快速还原场景，快速解决用户bug

线上环境乱七八糟，
各种版本，
各种路径，
各种依赖。。。

数据库的表被
清空了。。。

积累了两年的
BBS附件，
因为发布被删
光了。。。。

结果

业务上线了，
配套的监控，
好不容易上线了
监控响应不及时。。。

每个人有自己的想法，
在自动化工具上得不
到统一。。

填坑

严格的职能限制，
线上必须听运维的，
即使看起来不合理

成立运维团队，
全面负责基础环境
搭建和维护

收回sudo权限，
制定Linux线上
环境使用标准

每个核心业务都有专门的运维同学负责（告警会第一时间响应）

推动故障等级和
制度的标准化

强制具体的业务
制定稳定性指标，
而不是提高大家的意识

- 70%的工作其实不是研发过程，而是软件维护
- 70%的人看到告警之后不会马上采取行动
- 70%的运维同学天然对故障和告警敏感
- 没有完整工具链的支持，DevOPS就是瞎扯
- OPSDev 可能比 DevOPS更符合国情
- 现在还推崇DevOPS是因为系统不够复杂或者团队巨牛逼

总结

未来

正在挖另外一
个坑：SRE

继续剥离开发
的运维职责

术业有专攻、
谋定而后动

微服务是另外
一个巨大的坑，
一定会碰到
DevOPS一样的
问题

2. 自研底层系统？

● 我们以为 ●

很简单，一个月两个月就可以搞定某个系统

现成的系统或者对应的东西都有挺多的缺陷，代码也写得烂

不能完美的和已有的系统进行结合，还是需要一点开发

好像还有几个我们需要的特性没有被满足

开源文档不全，代码难懂，社区不活跃，说不定哪天烂尾了

于是

组织一个至少2人团队进行研发

需求调研、架构设计、技术评审，开始一二三期往前走

专注于解决对标物没有解决的问题，让自己更牛逼

停止类似软件的引入以减少后期切换的成本

过程里不停的和兄弟团队沟通需求，接受建议

结果

越做越发现，好像没那么简单

做的过程中发现了更多的需求

为了让自己更牛逼，做了挺多无用的功能

虽然理想很伟大，三四个月过去了，还在做基础功能

兄弟团队开始表现出不信任，系统里用各种方法绕过对这个需求的依赖

发现人力不够，要求加人，要求招聘

填坑

重新确定核心需求点

招募更好的工程师，
加速产品迭代

适当放开大家使用同
类产品，并制定好切
换策略

放弃高大上的想法，
回归先现在的业务满
足需求

加班加点急速填坑，
辛辛苦苦，还被埋怨

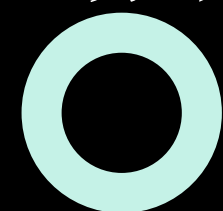
总结

- 基于一个成熟产品去研发，最保险，见效快，更从容
- 团队技能树不具备对应的产品要求，千万不要自研
- 直接面向开源去做自研，不仅代码质量好，文档也会更好
- 自研有很大风险，但能提升团队的技术追求和欲望
- 成熟开源软件不解决的问题，通常都是不重要的问题

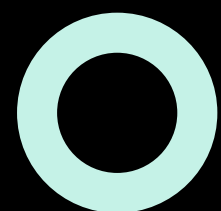
3. 过程改进

目标

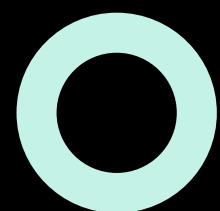
让横向协作更高效
比如跨团队，
跨项目，
跨业务单元



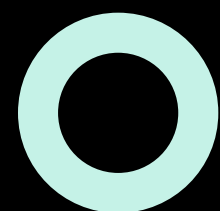
对于管理决策有数据支持，不能只拍脑袋



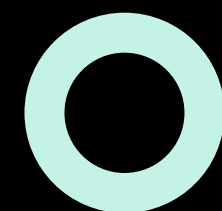
对于常规工作提供最基本的可参考实现路径



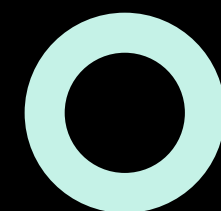
对多样化的生产方式提供实践方法和理论指导



保证生产过程有更多的数据留存



作为观察者，为整个软件生产过程提供可执行建议



过程

项目制是一个好方法，
建立柔性PM体系
(1:100)

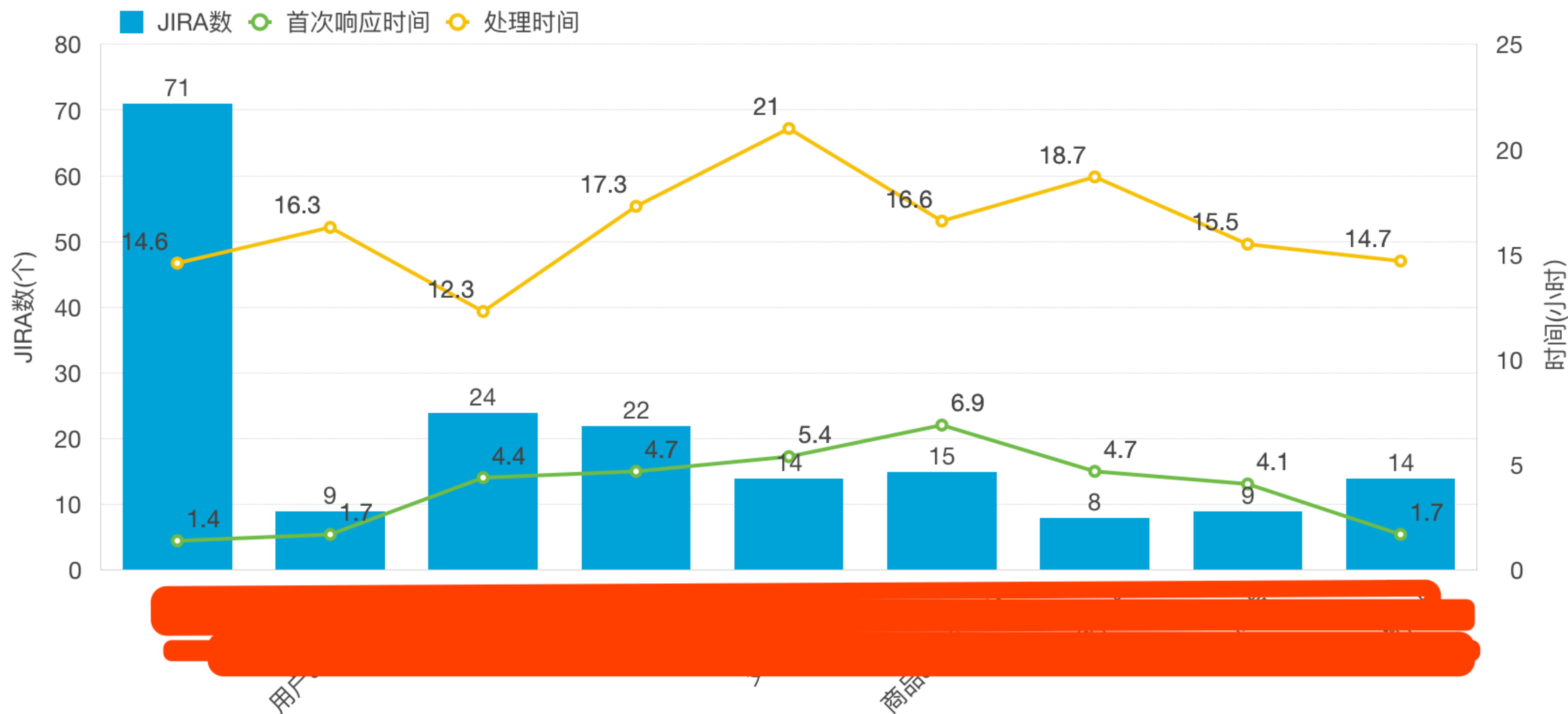
从需求方着手，重点
解决他们的问题

通过故障来推动改进，
建立故障制度

普及工具，帮助团队
使用工具来提高效率

人人都是PM，专职PM
负责指导，提升意识
和技能

线上问题:



近两周修复的Bugs中，除**电商技术**外，各团队平均**首次响应时间**都在**7h**内，商品&物流

首次响应和**处理速度**都比较快

4. 团队构建

几点经验

- 用极致的方法找到和你的团队味道一样的人
- 找到你的团队贡献最大的10%的人，并赋予他们权利
- 不要试图讨好你的团队，作为leader应该做正确事
- 万不得已的时候才增加岗位，否则只要增加角色就够了
- 周会是用来同步信息的，不是用来讨论具体问题的



关注QCon微信公众号，
获得更多干货！

Thanks!



主办方 **Geekbang**  **InfoQ**
极客邦科技