



● 大型互联网企业研发效能提升之道

王一男

腾讯DevOps产品专家



自我介绍



王一男

腾讯 — DevOps产品专家

前

百度 — 资深产品经理

开源中国 — 产品总监




提升研发效能的策略

研效提升的实践分享

什么是研效？

什么是研发效能？

什么是软件工程能力？



什么是软件工程能力？

百度软件工程能力的定义

使用系统化的方法，在保证质量的前提下，更高效率的为客户/用户持续交付有价值的软件或服务的能力。



百度如何定义工程能力

使用系统化的方法，在保证质量的前提下，更高效地为客户/用户持续交付有价值的软件或服务的能力

提供价值

- 系统设计，编写代码只是手段
- 从客户需求或商业价值出发，努力为客户、企业和社会创造价值
- 建立成本仪式，以最小成本换取最大价值

质量第一

- 低质量软件会增加整体研发成本
- 平衡好质量与效率，高效率，低成本，系统性保证质量
- 质量控制贯穿产品全流程交付过程

持续交付


- 软件的研发和维护是长期性的
- 软件/服务的价值提供是长周期的
- 摒弃“线快速粗暴开发，后加倍还债”

系统化方法

- 软件工程有大量优秀理念，方法和实践，是否使用科学方法论效果相差百倍
- 主动拥抱云计算，开源
- 基于优秀的基础设施和软件复用能力支持快速迭代

提升人效


- 人效贯穿全研发过程，决定交付的执行速度和人力成本，是决定产品成功与否的关键因素



腾讯如何定义研效

让每一位腾讯的工程师能够高效、快乐地工作





提升研效/工程能力的目的是什么

助力业务成功



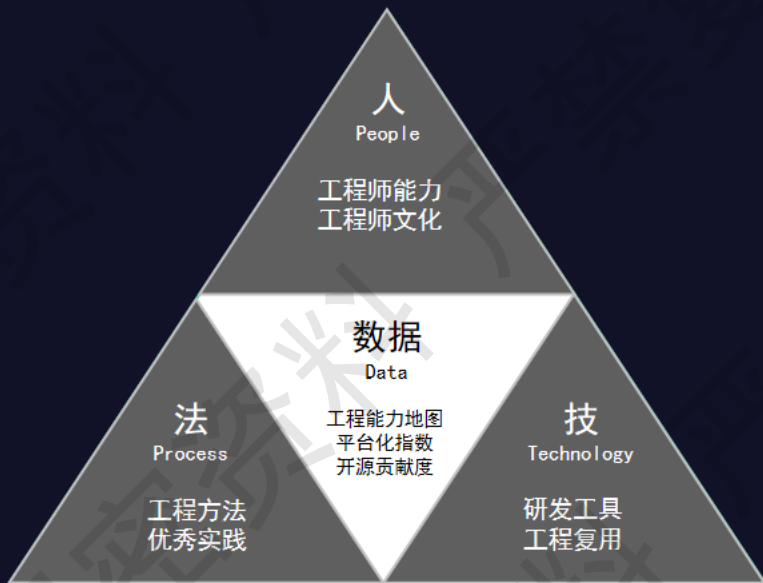
工程能力的影响模型

- 提高研发效率和产品质量
- 减少浪费
- 工程师素养提升
- 有序治理



提升研发效能的策略

研发效能提升的策略模型



提升研发效能的策略



人 (People)

- 招优秀的工程师
- 工程师能力培养
- 工程师文化建设
- 研效技术委员会

- 个人能力素质
 1. 需求把握
 2. 系统设计
 3. 编码能力
 4. 项目管理
 5. 运维能力
 6. 产品意识
 7. 客户服务意识
 8. 安全意识
 9. 质量意识



- 团队能力素质
 1. 研发过程管理
 2. 研发效率
 3. 成本意识
 4. 教育培养

ENGINEERS
DEVELOPMENT
工程师



研效技术委员会

为了深化开源协同，提高垂直领域开源协同质量及效率，公司技术委员会决定成立子领域委员会——研效技术委员会，统筹研效领域开源协同工作。

联系我们

提升研发效能的策略



法 (Process)

工程实践

管理实践

平台治理方法

开源方法

研发方法：百度方法+



内部开源：腾讯四大衡量维度

研发指标	运营指标	协同度量	社区运营
<ul style="list-style-type: none"> 代码规范 代码安全 代码度量与检查 文档质量 	<ul style="list-style-type: none"> 业务接入覆盖应用规模 Oteam服务水平 (SLA) 	<ul style="list-style-type: none"> Commit SourceAdd Merge Request Code Review Issue 	<ul style="list-style-type: none"> 需求响应 用户评价 文档建设 技术氛围 项目进展 课程 文章

*开源率作为参考值

工程标准：百度工程能力白皮书 - 工程标准V2.0





提升研发效能的策略



技 (Technology)

研发工具

管理协同工具

DevOps工具

工程复用

平台复用 (平台治理)

源码复用 (开源)



提升研发效能的策略



数据 (Data)

研发现场大数据

工程能力地图

工程师画像

工程复用大数据

平台化指数

开源贡献度

Java	54.15%
XML	42.97%
SQL	1.89%
JSP	0.55%
JSX	0.18%
luigi-service	55.07%
评审	138
Function定义	894
代码缺陷	104
善于发现缺陷	18



CR数量	变更行数	单位CR变更行数	人工评论数	同行评论数	CR参与人数	有效开发人数	CR参与度
76	90,606	1,192	130	1.43	3	4	75.00%
5,899	3,194,203	541	22,028	6.90	246	278	88.49%
1,864	3,296,827	1,769	5,224	1.58	122	153	79.74%
654	351,167	537	1,152	3.28	45	49	91.84%
7,342	4,232,421	576	49,385	11.67	476	529	89.98%
3,900	2,580,650	662	22,416	8.69	68	79	86.06%
1,063	550,704	518	3,981	7.23	71	104	68.27%
2,234	978,695	438	6,192	6.33	170	207	82.13%
467	359,821	770	2,343	6.51	24	28	85.71%
16,732	21,061,900	1,259	104,286	4.95	1,003	1,136	88.29%
10,265	10,160,900	990	37,322	3.67	762	922	82.65%
1,415	1,674,762	1,184	7,077	4.23	123	163	75.46%
1,404	78,412,451	55,849	5,208	0.07	125	156	80.13%
3,459	3,523,841	1,019	12,573	3.57	280	359	77.99%
26,827	29,130,882	1,086	101,620	3.49	1,872	2,202	85.01%
1	2	2	1	500.00	1	3	33.33%
203	274,506	1,352	195	0.71	25	27	92.59%



研效提升的实践

研发工具Infra化-提高研发效率和质量



Tools VS Infrastructure



研效提升的实践

代码Infrastructure

代码托管

代码协作

代码评审

代码质量保证

代码治理



质量是开发人员保证的，这种思想落地在代码工具中，增加了许多自动化检测工具



冲突检查 | 编码规范 | 缺陷检查 | MI 指数 (圈复杂度、注释...) | 重复文件 | 提交前构建流水线 (编译、UT...) | Review App | 人工评审



研效提升的实践

对代码质量的认识不同

一般公司：质量是测试人员保证的，代码扫描工具仅个别团队自己搭建使用

互联网大厂：质量是开发人员保证的，这种思想落地在代码工具中，增加了许多自动化检测工具

一般公司：这么多自动化检查，是否会降低效率？

互联网大厂：质量风险前置是提高整体研发效能、提高产品质量的最有效手段

**DevOps and
Agile
Fail Fast
Fail Early
Fail Often
Fail Safe**

研效提升的实践

数据驱动研效提升

工程能力地图

规范

落地

可见

DevOps TOC

项目团队



DevInfra

Engineering
Mapping

数据指标分类:

过程指标

研发基础数据

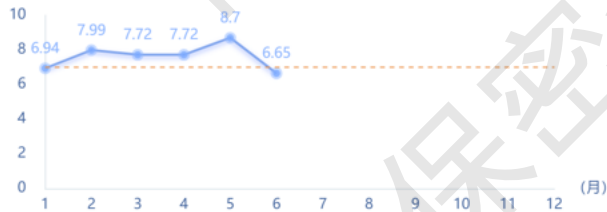
管理实践

工程实践

结果指标

速度

质量



2019年1月份，春晚红包项目平均每天

- 新增需求、任务卡片 1800
- 解决BUG 340
- 代码评审 1100
- 编译 11000
- 自动化测试 12000
- 发布版本包 900
- 上线部署 240

[illegible]

研效提升的实践

研效大数据平台建设

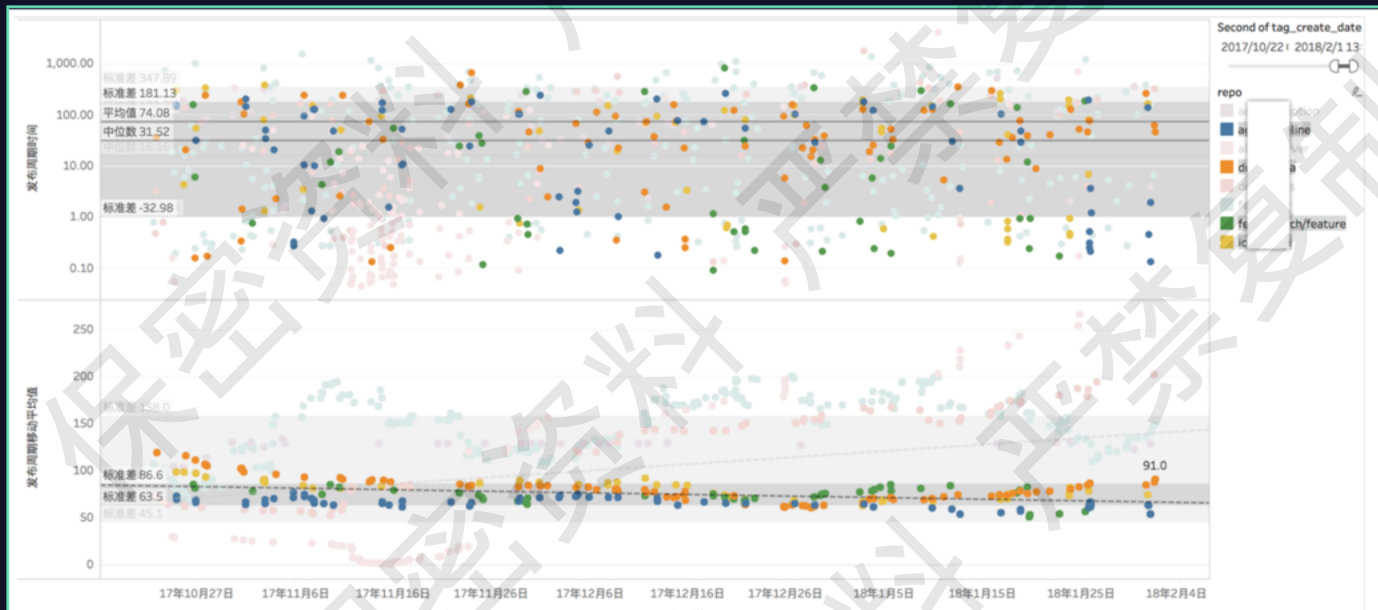
- 以工程数据为主（需求、开发、准入、测试、部署），实时度量团队的功能能力
- 数据采集、数据同步工具
- 数据存储服务
- 数据计算平台
- OLAP查询能力
- 数据治理服务
- 数据量大，准确度高，应用范围广




研效提升的实践

周期时间（CycleTime）的定量分析

工程实践对开发周期的影响



我们发现在选定的时间范围内（横坐标），推进工程实践落地的4个团队（图中高亮的颜色），其开发周期的移动平均值（下图的散点）成稳定下降趋势；没有推进工程实践落地的4个团队（图中背景部分），其开发周期主要成发散上升趋势。



研效提升的实践

周期时间 (CycleTime) 的定量分析

工程实践对开发周期的影响

分析结果：

- 团队采用的工程实践数量越多，其开发周期越短
- 工程实践做得程度越深入的团队，其开发周期也越短
- 团队人数越多，实施工程实践对缩短开发周期的作用就越大



什么是软件工程能力？

研发数据怎么用？

微观：

- 代码搜索
- 代码补全
- 缺陷预测

中观：

- 工程师画像
- 工程能力地图
- DevOptics

宏观

- 企业研发效能提升



源代码数据



缺陷数据



日志数据

总结

提升研发效能的策略

研效提升的实践分享

感谢聆听！