

# 头疼，技术leader怎么定量化KPI

58速运-沈剑

## 关于 - 我

- “架构师之路”作者，深夜写写技术文章
- 百度 - 高级工程师
- 58同城 - 高级架构师，技术委员会主席，技术学院优秀讲师
- 58到家 - 高级技术总监，技术委员会主席
- 58速运 - CTO
- 本质：技术人一枚



## 目录

- 技术部门KPI/OKR的**量化技巧**
- **底层技术团队**典型量化KPI/OKR**坑与实践**
- **中台技术团队**典型量化KPI/OKR**坑与实践**
- **业务技术团队**典型量化KPI/OKR**坑与实践**
- 总结

KPI , 还是 , OKR ?

今天不讲

不管是哪一种？量化很重要（SMART）

# 一、关于量化

## 别纠结SMART，能容易评估“完成”或“未完成”

- 今天气氛很热烈
  - 我很帅
  - 今天很热
- > 确定标准之前，不容易评估，即“不量化”



## 看一个底层技术部门的关键结果

- 来看几个关键结果
  - 数据库资源利用率的提升
  - 研发质量的提升
  - 研发效率的提升
  - 技术氛围的提升

## 看一个底层技术部门的关键结果

- 来看几个关键结果
  - 数据库资源利用率的提升 -> 数据库实例数减少20%
  - 研发质量的提升 -> 线上bug数减少20%
  - 研发效率的提升 -> 全部系统接入自动化上线平台
  - 技术氛围的提升 -> 每月进行至少3次技术分享
- 技巧：通过“**量化数字**”的增减，或者“**完成**”“**实现**”等动词，季度末容易判断关键结果是YES，还是NO

## 二、底层技术团队的典型目标与关键结果

## 底层技术团队的常见目标与关键结果

- 适合部门
  - 运维部OP
  - 数据库管理部DBA
  - 架构部
  - 基础服务部
  - 技术平台部
- 常见目标
  - 优化公司系统架构
  - 提升公司研发效率
  - 提升技术研发质量
  - 构建高效稳定组织力强技术团队

## 一些看似合理的关键结果

## 底层技术团队，关键结果，看似合理

- 关键结果
  - 完成“里程计算”基础服务的研发
  - 发布“内部sso”技术平台
  - 完成“数据收集中心”技术平台
  - 完成“图片上传下载”组件的自研
- 存在什么问题？

自high是没有意义的

## 底层技术团队，关键结果

- 关键结果
  - 完成“里程计算”基础服务的研发，完成速运业务线接入，降低里程库写压力80%
  - 发布“内部sso”技术平台，推动至少5个系统接入
  - 完成“数据收集中心”技术平台，推动并完成http访问，APP上报，业务日志的收集
  - 完成“图片上传下载”组件的自研，推动并完成至少3个上游的接入



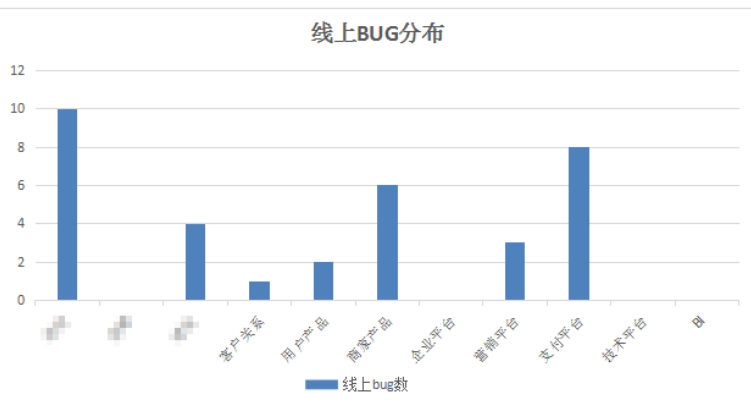
不只要“生产”，还要“销售”

质量相关，关键结果如何设定？

## 底层技术团队，关键结果，质量模块

- 关键结果
  - 报警率
  - 故障率（故障次数）
  - 监控覆盖度
  - 上线成功率
  - 请求时延
- 看似容易，需要系统和数据支持
  - 有报警数据么？
  - 有bug统计么？有bug分级制度么？
  - 有监控系统么？
  - 有上线成功率数据么？
  - 有相关日志数据与统计么？

业务线	线上bug数	占比
BI	0	0.00%
技术平台	0	0.00%
支付平台	8	23.53%
营销平台	3	8.82%
企业平台	0	0.00%
商家产品	6	17.65%
用户产品	2	5.88%
客户关系	1	2.94%
总计	34	100.00%



业务线	上线总数	一次性上线成功	一次上线成功率	结果
BI	43	41	95.35%	达标
技术平台	13	13	100.00%	达标
支付平台	108	102	94.44%	达标
客户关系	45	44	97.78%	达标
用户产品	39	39	100.00%	达标
商家产品	119	119	100.00%	达标
企业平台	37	37	100.00%	达标
营销平台	29	29	100.00%	达标
支付平台	46	44	95.65%	达标
技术平台	68	68	100.00%	达标
BI	36	36	100.00%	达标
合计	583	572	98.11%	达标

什么是线下BUG加权总分？分大权重和小权重，以大权重为例，线下BUG加权总分=最高级BUG个数\*9+严重级\*7+主要级\*5+次要级\*3+影响最小级\*1

级别	大权重	小权重
最高级	9	5
严重级	7	4
主要级	5	3
次要级	3	2
影响最小级	1	1

### 千行代码BUG率

业务线	GIT代码统计行数	线下Bug加权 (大权重)	线下Bug加权 (小权重)	千行代码故障率 (大权重)	千行代码故障率 (小权重)
BI	130034	184	110	1.42	0.85
支付平台	127173	726	437	5.71	3.44
客户关系	72742	381	231	5.24	3.18
营销平台	74344	736	449	9.90	6.04
技术平台	86402	317	195	3.67	2.26
企业平台	110240	874	524	7.93	4.75
用户产品	77587	1242	756	16.01	9.74
商家产品	180746	776	472	4.29	2.61
BI	83021	77	49	0.93	0.59
技术平台	28720	1683	1027	58.60	35.76
支付平台	153020	3195	1924	20.88	12.57

业务线	超过3人日项目总数	冒烟测试通过率超90%项目数	冒烟测试通过率超90%项目占比
BI	8	7	88%
技术平台	0	0	0%
支付平台	12	11	92%
客户关系	3	3	100%
用户产品	2	2	100%
商家产品	4	4	100%
企业平台	5	5	100%
营销平台	0	0	0%
支付平台	9	9	100%
技术平台	0	0	0%
BI	1	1	100%
总计	44	42	95%

质量需要数据支撑

改进质量，先有数据

# 底层技术部OKR实例



2016Q2-技术平  
台部-沈剑

## 三、中台业务研发团队的典型目标与关键结果

## 中台业务研发团队的常见目标与关键结果

- 适合部门
  - 企业信息化平台
  - 支付平台
  - 营销平台
  - 客户关系平台
  - 客服/回访平台
- 常见目标
  - 协助完成业务指标
  - 核心项目按时上线
  - 建立通用的中台系统
  - 提升效率...
  - 提升质量...
  - 组织氛围...



## 一些看似合理的关键结果

## 中台业务研发团队，关键结果，看似合理

- 关键结果
  - 完成A项目
  - 完成B项目
  - 完成C项目
  - ...
- 存在什么问题？

项目永远能完成（和完成程度有关）

某项目没完成，也一定完成了其他项目

## 中台业务研发团队，关键结果

- 关键结果
  - 完成A项目，实现某业务线的队长自动化
  - 完成B项目，提升客服工作效率10%提升
  - 完成C项目，确保大促项目顺利上线
  - ...

不只要“完成”，还要有“效果”

质量相关，关键结果如何设定？

## 中台业务研发团队，关键结果，质量模块

- 关键结果
  - 业务事故数
  - 业务慢SQL数
  - 业务非HTTP200报警数
  - 业务核心接口性能（处理时间）
  - ...
- 当然，也得有数据

# 中台业务研发团队OKR实例



2017Q2-信息技  
术部-沈剑



## 四、业务研发团队的典型目标与关键结果

## 业务研发团队的常见目标与关键结果

- 适合部门
  - 业务研发部门
- 常见目标
  - 业务目标
  - 技术驱动业务
  - 技术有效支撑

# 小心自业务子部门关键结果对立

## 看一个技术与产品的关键结果

- 技术侧关键结果

- 系统可用性99.99%
- 请求报警率低于1%%
- 项目延期率低于10%
- 完成SSO架构改造

- 产品侧关键结果

- 总PV提升15%
- 请UV提升15%
- 总日活提升15%
- 完成订单改版项目，订单转化率提升15%

## 看一个研发与测试的关键结果

- 研发侧关键结果
  - 千行bug率低于XXX
  - 提测延期率低于YYY

- 测试侧关键结果
  - 千行bug测出率高于XXX
  - 研发测试工期比大于ZZZ

同一个业务的不同team形不成“合力”

别为达成关键结果，行为与目标背道而驰

## 看一个业务团队的关键结果

- 业务关键结果
  - 业务订单量高于XXX
  - 业务GMV高于YYY
- 达成关键结果的手段
  - 刷单（开启自毁模式）
  - 疯狂补贴（烧钱烧得没有意义）



# 业务研发团队OKR实例



2017Q4-速运技术-沈剑

## 五、总结

## 总结

- **目标设定要量化**，要容易评估，通过“量化数字”的增减，“完成”“实现”等动词落地
- **底层技术团队**
  - 不能自high，成果要推广
  - 质量优化，要先有数据
- **中台业务研发团队**
  - 光列项目不行，项目要有效果
  - 质量要与业务结合
- **业务研发团队**
  - 子部门关键结果勿“对立”，要“合力”
  - 勿一味追求结果，让行为与目标背离
  - 业务目标最优先，技术团队支撑/驱动

谢谢！

“架构师之路” 公众号



GIAC | 全球互联网架构大会  
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



扫码关注GIAC公众号

2017.thegiac.com