

YEAR IN REVIEW

Extra Geeks' Organization Networks

2015

EGO EXTRA GEEKS' ORGANIZATION
NETWORKS



CONTENTS 目录

01	关于EGO/About EGO	14	学习内容精选——人才招聘与管理
02	创始人致辞	22	学习内容精选——团队管理与协作
03	EGO版图	36	学习内容精选——企业架构与成长
04	EGO2015大事记	46	学习内容精选——个人成长与管理
05	EGO会员名单	54	学习内容精选——创业之路
11	会长致辞 杨卫华	63	会员感言摘选
12	会长致辞 范凯	64	加入EGO
13	会长致辞 陈懿新	65	极客邦科技介绍

关于EGO/ About EGO: EGO (Extra Geeks' Organization)

EGO是极客邦科技旗下高端技术人聚集和交流的组织，由InfoQ中国团队运营，旨在组建全球最具影响力的高端技术人社交网络，线上线下相结合，为会员提供专享服务。

EGO采用实名付费会员制。每一位申请加入的会员都必须经过EGO组织的严格审核，保证会员信息的真实性。会员皆为有技术经验的管理者，比如公司CTO、技术VP、技术背景的CEO等高端技术人。

目前已建立EGO北京、上海、深圳分会，十余个学习小组，广州、杭州、南京、苏州、武汉、西安等区域分分会也将于近期成立。

EGO核心价值观：

信任和尊重，渴望学习，勇往直前，打造印记，



酷。

EGO会员福利：

学习交流：/ Learning Exchange

独一无二的学习和交流体验，从月度分会学习活动到月度小组私密会议，会员间平等分享各自的实战经验，从工作到生活；

拓展人脉：/ Expand Contacts

认识国内外志同道合的技术同行，分享交流，拓展自己的人脉圈子；

获取信息：/ Get Information

获取最新最重要的行业信息及资讯，了解最新技术动态、新闻、社会热点，独家原创的专家观点及评论；

解决问题：/ Solve The Problem

身边的智囊团，遇到问题时能够第一时间通过EGO平台找到答案；

工作之外：/ Outside Of Work

在日益激烈的竞争中平衡工作与生活之间的关系，提高软技能。

“

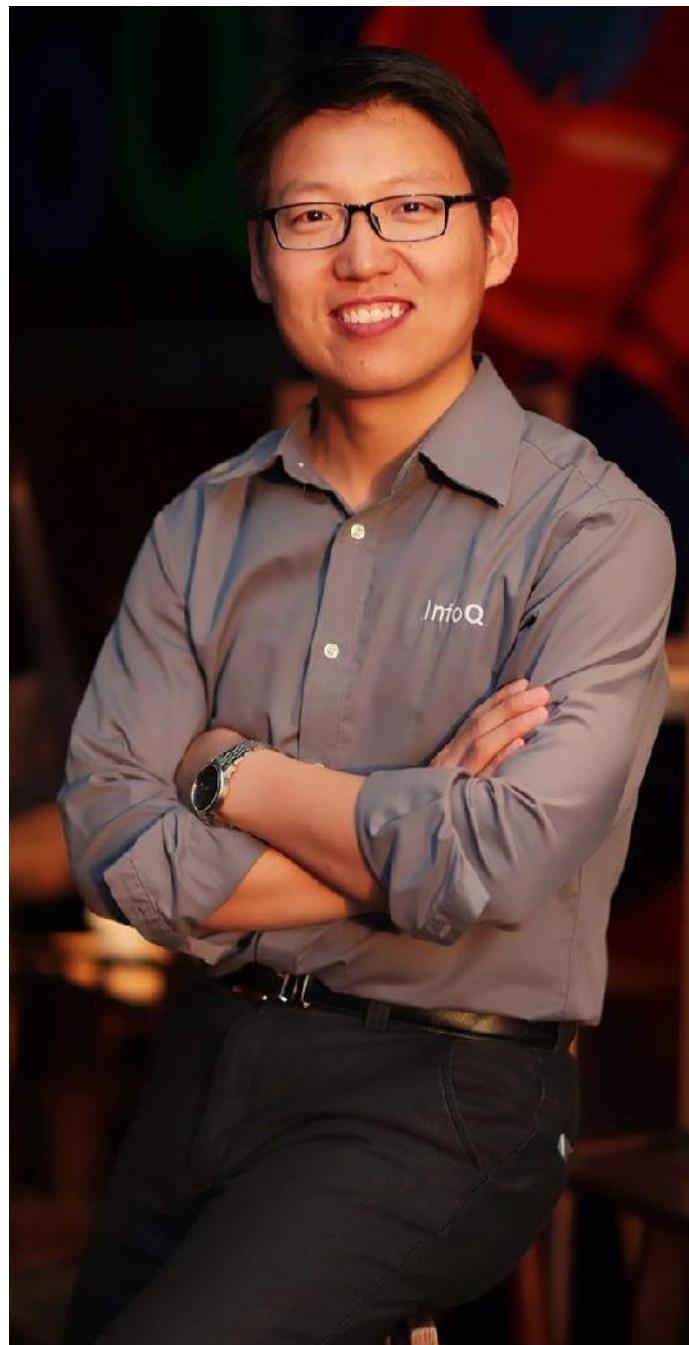
如果从EGO北京分会去年8月8日成立开始算，这个新生事物半岁还不到，但是已经让高端技术管理者侧目了。到目前为止，共有106名会员，马上就是108将。看上去人数不多，但是如果了解到这106位会员都是每个快速发展公司的CTO或者技术VP，甚至是某个创业公司的CEO，估计EGO这三个字母就变得沉甸甸的了。

从另外一个角度看，这说明技术人员，尤其是高级技术管理者是需要同行者的。这也是极客邦科技在InfoQ之后，推出EGO产品的最主要原因。每年有数千名中高端技术人员参与InfoQ举办的各种会议，数百名专家担任讲师，如何能够在会议之外，能够持续为全国各地的技术人提供高质量的交流机会，一直是我在琢磨的事情。在和很多的技术人沟通过程中，大家共同的反馈是人到高处，需要找到志同道合者，吐吐心声。所以，大家就可以看到EGO会员之间的沟通，尤其是小组之间的，都特别强调私密性，目的就是给大家提供一个安全的交流环境。

当然，对于EGO来说，连接只是第一步，终极目的还是为了引导大家学习和成长，这也是为什么每个月除了各小组自己的会议，当地分会还有面向全体会员的学习活动。在这些活动中，也期望给会员提供一个共同认识的机会。至于大家学习和成长之后的价值，显而易见，就不用赘述了。我认为，不论何时，人都是一个公司发展的关键因素，更不用提关键人物。EGO每一个会员成长的后面，都代表一个研发团队、一个企业在快速提升。这就是EGO存在的价值！

2016年，EGO会在现有的北京、上海、深圳等三地分会基础上，走进杭州、广州、武汉、西安、成都、厦门、大连等城市，在力所能及的范围内，把国内优秀的技术人连接起来。我们的愿景是成为全球最具影响力的高端技术人社交网络，但2016年的目标是先成为中国最具影响力的是：）

希望在EGO见到你，祝大家新年愉快，猴年精神抖擞，猴精猴精的！



极客邦科技 & InfoQ中国
创始人兼CEO 霍泰稳

“

最初产生EGO的想法时，我们想做的，就是打造一个最具影响力的高端技术人社交网络，把全中国的CTO、技术VP们聚在一起，给他们提供分享和交流的平台，帮助他们学习与成长。同时，中国技术开放日等活动帮助国内技术走出国门的同时，也让EGO与硅谷的技术人建立了联系。未来，EGO还将走出国门，真正实现“全球最具影响力的高端技术人社交网络”的愿景。

从2015年8月8日北京分会正式成立算起，EGO诞生还不到半年，但就在这短短的时间内，EGO已经在北京、上海、深圳建立分会，吸引到超过百名会员，举办分会活动16场，小组活动36场，也从线上到线下，把会员们真正连接了起来。

有些会员一见如故，私下成了非常好的朋友，常常相约一起跑步、健身；有些会员发现彼此业务相合，一下就建立了合作关系；有的会员知道同组有人没法参加分会学习活动，就会帮忙详细记录并整理学习要点，再分享给大家；有些会员虽然工作繁忙，但为了见到大家，还是坚持参加小组活动……

会员们有什么疑惑、遇到什么难题，发到群里，总会有人给出自己的观点、建议，甚至能引发大家的讨论；想说些什么，大家也都能理解，总能找到志同道合者……有时也会有很多不同的声音，但对高端技术人而言，不同观点之间的碰撞恰能闪现出更明亮的火花。

这些正是EGO想做的，不仅仅是线上浅浅的几句交谈，也不是线下泛泛的几次活动，而是真正建立起彼此之间的强关系，让大家真正成为能彼此相伴、共同成长的珍贵伙伴。

当然，EGO并不是加入之后，就能放松随意跟别人聊聊天的地方。我们的主旨是建立学习和成长的平台，会要求每位会员都热爱学习、乐于分享、互相促进、真正成长。因此，EGO也有着严密的规则，对会员、小组、分会等活动都有严格要求。

可能有些人会觉得EGO的规矩太多、太繁琐，但分享和学习从来不是一件轻松的事情，是需要有一个规则去督促的。EGO做的，就是建立起这样的规则，让会员们能在此之下自由交流、尽情分享。

“三人行必有我师”，EGO有着这么多CTO、技术VP层级的高端技术人，只要你想，又怎会没有收获呢？



**EGO联合创始人
李秋月 Pheope**

EGO 版图 TERRITORY (截至2016.01)



2016年计划新成立的分支:

广州分会、杭州分会、武汉分会、成都分会、西安分会、南京分会、苏州分会

EGO2015大事记



2014年
12月

2015年
4月22日

2015年
5月5日

2015年
8月8日

2015年
9月9日

2015年
11月26日

2015年
12月



EGO雏形初步确立

EGO首场活动

Facebook首位华人工程师赵海平先生做关于研发团队管理的分享

EGO迎来了第一位会员
——百姓网技术总监艾美女士

EGO北京分会正式成立
EGO正式扬帆起航

EGO上海分会正式成立
开启上海技术圈交流成长

EGO深圳分会正式成立
引领深圳技术人辩理论道

EGO会员数量突破100

(2016.01)

EGO MEMBERSHIP LIST

会员名单 / 按姓氏拼音排序 /

A. | 艾芙 / 技术总监 / 百姓网

B. | 白顺龙 / 技术总监 / 上海七牛信息技术有限公司

毕鹏 / 技术总监 / 点融网

C. | 蔡日炜 / 架构师 / 深圳市力天保利科技有限公司

陈超 / 技术总监 / 七牛云存储

陈钢 / 联合创始人 CTO / 深圳市早知道科技有限公司

陈靖华 / CTO / 北京基调网络股份有限公司

陈威 / 资深架构师 / Pivotal

陈锡言 / 技术总监 / 风行网

陈懿新 / 部长 / 华为中央软件技术战略部

程辉 / CEO / UNITEDSTACK(北京)科技有限公司

崔华 / 合伙人 技术VP / 小信用

EGO

MEMBERS LIST

会员名单



D. | 丁宁 / CTO / 广州荔枝网络技术有限公司

董海冰 / 架构研发总监 / 沪江网

段念 / CTO / 宜人贷

F. | 范凯 / 技术副总裁 / 丁香园

冯亮 / 技术合伙人CTO / 深圳极钛星华信息技术有限公司

冯是聪 / CTO / 北京明略软件系统有限公司

冯帅刚 / 单品网事业部技术总监 / 北京网库信息技术股份有限公司

G. | 龚银 / 研发负责人 / 一加科技有限公司

郭理靖 / CTO / 同仁堂国际

H. | 韩卿 / Product Lead of eBay ADI / eBay

郝飞 / CTO / 上海爱聊信息科技有限公司

何明 / 系统运维总监 / 北京亿玛在线科技

何小峰 / 技术中心主任 / 清华同方知好乐教育技术研究院

EGO

MEMBERS LIST

会员名单



何明 / 系统运维总监 / 北京亿玛在线科技

何小峰 / 技术中心主任 / 清华同方知好乐教育技术研究院

洪强宁 / 首席架构师 / 宜信

侯冬雯 / 技术总监 / 北京微梦广告有限公司

胡键 / CTO / 浙江中创物联科技有限公司

胡震 / 资深架构师 / 创业

华健 / 技术VP / 北京广联达梦龙软件有限公司

黄彬彬 / 首席架构师 / 上海脉创网络科技有限公司

黄慧攀 / CTO / 杭州又拍云科技有限公司

黄凯耀 / 主任工程师 / 华为

黄良懿 / 数据中心总监 / oppo

黄希彤 / 总监 / 腾讯云计算公司

J. | 蒋炜航 / VP / 宜信大数据创新中心

| 巨建华 / CEO / 青云极创

EGO

MEMBERS LIST

会员名单



林科 /CEO Founder/北京奥克巴尔科技有限公司

刘斌 /总监/易宝支付有限公司

刘超 /技术负责人/武汉闪达科技有限公司

刘建国 /CTO/果糖移动

刘如鸿 /助理副总裁/正大集团

刘天胜 /CTO/上海讯联数据服务有限公司

刘先军 /技术总监/普元信息技术股份有限公司

刘译璟 /首席架构师 研发总监
/北京百分点信息科技有限公司

刘颖 /架构师/腾讯云计算公司

刘智勇 /CTO/深圳萨摩耶互联网金融服务有限公司

陆栋栋 /CTO/喜马拉雅FM

K.

孔令欣 /CTO/点融网

孔令岳 /运营总监/腾讯

L.

黎明 /CTO/优维科技(深圳)有限公司

李道兵 /首席架构师&VP/七牛云存储

李国鹏 /CTO/北京风信科技有限公司

李锦国 /高级研发总监/找钢网

李军 /高级技术经理/百度

李鹏涛 /研发总监/京东商城

李申申 /CTO/知乎

李汐 /CTO/北京轻松筹网络科技有限公司

李晓东 /CTO/上海益盟软件技术股份有限公司

李志丹 /技术VP/杭州数云信息技术有限公司

梁宇鹏 /首席架构师/环信

廖益玄 /CTO/聚熵信息技术(上海)有限公司

林得苗 /CEO/北京鲜柚科技有限公司

林华根 /CTO/安徽超远信息技术有限公司

EGO

MEMBERS LIST

会员名单



M.	马鉴 / 联合创始人 / 北京白鹭时代信息技术有限公司
P.	庞律 / 研发总监 / VIPKID
Q.	邱模炯 / 平台开发总监 / UCloud
S.	史海峰 / 架构师总监 / 当当 孙宇聪 / CTO / Coding.net
T.	唐小浙 / CTO / 沪江网
W.	汪威 / CTO / 咸蛋家 汪长明 / 客户端架构师 / 深圳墨麟科技股份有限公司 王博龙 / CEO / 迈博 王葱权 / 产品部副总 / 普元信息技术股份有限公司 王栋 / CTO / 雪球(技术)开发有限公司 王慧星 / 副总裁 / 腾讯云计算公司 王立博 / 技术负责人 / 麻袋理财 王平 / CTO / Mobvista 王胜 / 首席架构师 / 汽车之家 王涛 / 联合创始人兼CTO / 广州巨杉软件开发有限公司 王望记 / 开发技术总监 / 北京六间房 王晓波 / 首席架构师 / 同程旅游 魏吉贺 / 技术总监 / 深圳市残友软件股份有限公司 吴海波 / CTO / 北京汇通天下物联科技有限公司

EGO

MEMBERS LIST

会员名单



X.

谢孟军 / 首席工程师 / Apple

谢秀鹏 / O2O高级合伙人 平台技术负责人 /
永辉云创管理有限公司

熊昌伟 / 运维总监 / 畅捷通信息技术股份有限公司

Y.

闫祥 / 副总经理 / 新秀动力文化传媒(北京)有限公司

杨德升 / CTO / 周伯通招聘

杨海朝 / CTO助理 / 去哪儿网

杨荣伟 / 技术VP / 眇客

杨威 / 技术合伙人 / 北京明略软件系统有限公司

杨卫华 / 研发副总经理 / 新浪微博

杨毓丞 / 技术总监 / 北京游谱科技发展有限公司

虞冰 / Co-Founder CTO / 蛋壳公寓

Z.

张海龙 / CTO / 同程旅游

张海涛 / 部门经理 / 北京国信创新

张虎 / 创始人兼CEO / 深圳市微智云科技有限公司

张新波 / 技术副总裁 / 杭州同盾科技有限公司

张雪峰 / CTO / 饿了么

张智宇 / 技术专家 / 阿里巴巴

章烨明 / CTO / 杏仁医生

钟忻 / 云平台总监 / 乐视云计算有限公司

朱炳芳 / 华南区总经理 / 正益移动Appcan

邹光先 / CTO / 广州爱拍网络技术有限公司

会长致辞

【北京 · BEIJING】

杨卫华 Tim Yang

EGO北京分会会长, 新浪微博研发副总经理

“

在 IT 领域尤其是技术领域, 知识的升级迭代速度非常快, 因此从业群体需要比其他行业更多更快的学习新知识及了解新资讯。因此有大量人群被这些垂直领域的意见领袖聚集在群、自媒体等社交渠道周围。

在线下, 也有另外一些不同的新的玩法。极客邦科技 (InfoQ 中国背后的公司) 今年成立了一个技术人学习组织 EGO, 会员主要以公司技术负责人为主。Tim 由于也是其中的会员之一, 因此有机会结识了很多新的会员。这些加入会员也是在自己的职业生涯某些方面存在困惑, 公司的工作太忙, 网上的信息太庞杂, 去校园进修又不太现实, 因此其实很缺乏高效的线下学习途径。

Tim 在与这些会员交流的过程中, 发现大部分人在他们相关的领域非常优秀, 也慢慢了解他们所属公司最近几年的发展历程, 了解他们所在的不同行业的发展情况, 也了解到他们在不同领域的奋斗、成长与得失。Tim 也很喜欢这种交流的过程, 一方面, 与这些行业优秀的技术人一起可以快速学到他们身上的优点及特长, 同时也可以通过他们了解到许多以前不了解的领域。

一次一个会员问了 Tim 一个非常好的问题, “你最近在这个组织得到的最大收获是什么?”

对于 Tim 而言, 如上所述, 最大的收获是跟不同会员的深度交流, 学习会员做事的方法与理念, 了解他们对于行业方向的判断、以及对于技术的态度及选择。

作为EGO北京分会的会长, 我见证了北京分会这半年来从成立到发展的整个过程, 参与了各次分会活动, 也帮助EGO面试了很多申请的会员, 并逐渐形成了一套申请规范, 这于我于EGO, 都是很有意义的事情。眼见着会员越来越多、交流越来越多、收获也越来越多, 祝福EGO在2016年发展顺利! 为高端技术人的学习和交流做出更多贡献!



会长致辞

【上海 · SHANGHAI】

范凯 Robbin

EGO上海分会会长, 丁香园技术副总裁

“

我觉得对研发人员来说, 不管是多做一些技术分享、技术交流, 还是在网上多写一些文章, 都能锻炼自己的表达能力, 能有更多的机会走出去跟别人沟通, 从而客观地审视自己身上的缺点, 有目的地去改正或弥补。

另外, 社区的影响力也与日俱增。原先的时候, 大家的技术通常都比较专用, 彼此之间缺少共同语言, 导致不同公司不同领域的技术人员很难直接交流。但从开源浪潮起来之后, 这种情况就改变了, 大家开始用同样的东西。同样的技术、同样的框架, 交流就有共同话语了, 这时, 你把你擅长的技术领域分享出来, 就能聚集更多的人, 你的影响力就能覆盖更大的范围。

特别是在现在这样一个激烈竞争的时代, 技术人仅仅坐在家里、公司里面钻研自己的技术已经不行了, 需要走出去学习、交流, 去看整个时代、整个行业是怎样快速往前发展的, 交流对技术人职业生涯发展起到的作用是越来越大了。

但是技术人、特别是高端技术人的交流组织很少, 我始终觉得这件事是有很大意义的, 以前就一直想做这样的事情, 但我一个人的话是势单力薄, 又不是职业的, 很难把它做起来。既然极客邦做了EGO这件事, 我可以参与进来, 大家合作把EGO做好, 也达成我自己的心愿, 这也是我为什么愿意加入并担任上海分会会长之职的原因。

希望在2016年能够有更多的有识之士加入EGO, 我们一起交流学习, 共同成长进步!



会长致辞

【深圳 · SHENZHEN】

陈懿新

Chen Yixin

EGO深圳分会会长, 华为中央软件技术战略部部长



“

深圳是一个高科技引领的地方,很多在北京创业的人最后会转战南方来深圳,因为深圳拥有自己独特的优势,一是在软件和技术领域的底蕴非常深;二是拥有非常全面的生态链,想做技术、做研究都能找到很资深的人。我相信在我们的努力之下,肯定能够让EGO走得很好、很伟大。

在这个技术创新井喷的时代,我们需要不断的学习和交流,EGO是技术人员的一个交流平台,大家在这里得到成长,收获技术与友情,祝愿EGO在新的一年越办越好!

美团论剑： 人才招聘与管理

【7月5日，EGO会员走进美团，
就人才招聘与管理的问题，与美团技术专家交流探讨。】

原文作者：EGO编辑 陈园园

怎么把人招进来

张锦懋（美团技术工程部负责人、技术委员会主席）：

我分享几点美团是怎么做的。首先，校招是可以去关注的一个点。现在美团每年需要招聘的人里面，40%-50%都是由校招解决的。当然，现在北京校招的竞争越来越白热化，可能更多的要往北京之外的城市去扩张。

第二个，美团有一个非常强的特点，就是通过内部员工的推荐去做招聘，虽然没有非常精确的数字，但总的来讲，至少招5个人里面有3-3.5个是通过推荐来的，举个例子，像邱剑就是这样加入美团的。

但是内部招聘这个东西要做好，有一点要去注意的，就是从公司CEO到事业部负责人、到一个小团队负责人，各层的人员都要把招聘当作一件优先级最高的事情去看待，特别是创业公司。

李鹏涛（京东商城研发总监）：

因为之前看《亮剑》，李云龙的方法就是放权，我们现在的做法也是一样，都分下去，就是让下面的人在符合公司条件的情况下自己去招人，你招到的就是你的。别的组招来了，还没招来的就会想办法，我们现在的人基本上全是这种方式招来的，还是挺好的，也不怎么依赖HR。

蒋炜航（原网易有道VP，现宜信大数据创新中心VP）：

挖人就是得经常自己上，比如说我们的产品总监，那是我大学同学给我推荐的，我最近招的一个有微软十几年一线工作经验的人，也是我们技术总监的同学给他推荐的。基本上改变上来就面试的方式，先吃饭、先忽悠，聊的差不多了再进行技术面试，这里HR就是辅助吧。还有一个方法，就是踢开HR直接找猎头，回复猎头也及时、勤快一点，像我五个小时就回了，所以下次他发简历就会给我发的勤一些，我的销售全是通过猎头招的。

陈锡言（风行网技术总监）：

我在招聘的时候，发现校招尤其难，有很多创业公司，不按套路出牌，就很难招到特别合适的人。我们主要还是使用内推的形式，去年这么做了效果还是很好的。我们估算过，通过内推，哪怕给出再高的奖励政策，最后的成本也比通过猎头要低很多。

刘建国（猫团动漫CTO）：

在我看来招聘不仅仅是筛几次简历、安排次面试，对于关键人才的招聘需要用更系统的方式。比方说，觉得谁合适，可以先请他过来参观认识一下，慢慢加深了解，也可以请过来做一些分享和交流，团队需要的时候，还可以找人

做一些场外指导,让他帮忙提提意见,也可以一起参加社交活动、技术会议什么的,机缘合适了就自然加入了。总而言之,在我看来这是一个很长线的事情,而不是几次联系、一次面试那样一个短线的事情。

另外,我觉得应该一开始就亮出我们的理想、梦想,虽然这可能十次里九次不管用,但一旦一次管用,招到一个认同我们梦想的人,这个价值就很高。

李军(百度高级技术经理):

我们前年急招一个APP消息推送的人,招了很久都没招到,我就自己到微博上去搜,搜发过这个技术相关帖子的人,然后关注他,给他发私信,最后还真成了,招到了一个有丰富相关经验的工程师。这样做虽然效率有点低,但当你要定向招某一个人、希望他有某种行业很聚焦的积累的时候,我觉得这个方式是行得通的。

招到人之后怎么办

张锦懋(美团技术工程部负责人、技术委员会主席):

每个阶段都是不一样的,在这个阶段,美团有越来越强的动力去推行一些代码规范、一些工具和平台的使用。因为在今天公司的这个规模情况下,这是一些非常重要的东西,而且我们也已经有了一些资源和人力去做这些事情。

但在公司比较早的阶段,我们是另外一种做法,因为那时候公司的人相对比较少,大家从一线的技术主管到公司的高层管理人员,每个人在做什么其实都是相对比较容易了解的,而且也经常能接触得到,所以那个阶段更多的通过文化的措施来做质量的管控:

蒋炜航(原网易有道VP,现宜信大数据创新中心VP):

招聘挺重要的,另外招到之后基本上就是一个抓新人,一个抓骨干。

抓新人的话,就是梳理一个比较有效的培养新人的方法。在mentor的培养下,从简单的任务做起,快速上手,逐步深入。

抓骨干的话,就是每次绩效评估,我跟技术总监都会比较看好某几个人,那我们就把所有资源都倾注在他们身上,包括我们自己的时间。就像线索管理一样,有抓有放。

史海峰(当当架构总监):

有一点说得比较好,确实不同的公司有不同的发展阶段和层次,所以,我们不能指望能把一个人培养到底,或者说指望他能在这儿干很多年。那就需要我们把这个职位或者技能需要的东西,做出相对成熟的、规范化的流程类的东西,把这些人培养出来后,能快速上手做我们所需要的工作。这样的培养体系相对来说是简单有效的,因为我们也不是为了培养人,我们是为了干活,员工的个人成长、技能提升一方面结合实际工作,一方面要靠自己。

刘建国(猫团动漫CTO):

我还是主张发展一个人,而不是用一个人,你可以尽可能用精力和资源发展他,虽然来一个走一个,但是十个人里可能有三个人回报这个公司,他可能自己不行,也会找人给你、会帮你介绍很多的信息。所以在我看来,从长线来看,一定要去用心去发展一个人,而不是把他当作一个员工。

(原文节选,完整版文章请查阅EGO微信公众号: egonetworks)

雪球CTO王栋谈招聘 认可团队与产品最重要

原文作者: EGO编辑 陈园园



EGO: 雪球是否有独属于自己的风格或文化?

王栋: 雪球并不总是说什么企业文化之类的事情,也没有一些slogan,当然工作上是有一些风格的,我们大概的风格就是工作中完全的信任和授权。比如我们的CEO就是完全授权给我管理技术团队,他只问我要结果,具体怎么实现、用什么样的技术实现、团队到底要怎么样管理,他都是不需要管的,都需要我自己去想。我对于团队的管理也沿袭了这种风格,最早我们定好一个大家比较能达成一致的目标和时间线,然后大家就可以顺着这个目标走。我们都不想过多的去直接干预大家,这样一是具体做事的人会感觉非常不好,同时也会减少他的主动性。

EGO: 这样完全授权的方式会不会对技术人员有比较高的要求?

王栋: 所以我们在招聘的时候就会特别注意这点,我们不能保证招进来的每一个同学都这样,但我们会希望,交给你的事情,你能主动的去思考、去理解、去沟通、去执行,开发过程中遇到问题能主动找方法去解决,实现之后能主动去做Review、做测试,这样一步一步,一直到上线,也会主动去监控服务、数据是不是正常……这一套如果你自己都不去关心的话,那就肯定没办法达到我们最基本要求的。

要求更高一点,在做这个事情的过程中,是不是有一个比较通用的东西,那这个通用的东西我们能不能把它抽象化,然后用到更多的地方。

所以我们对工程师的要求,就是积极、主动、主动去思考,不仅是思考技术上的东西,也希望大家积极去思考产品上的东西。

EGO: 那怎么找到这样合适的人呢? 会不会花很多的时间在招聘上?

王栋: 的确是,这几年,我花时间最多的可能就是在招聘上,比较密集的时候,可能要花60%-70%的时间在招聘上。一方面是约了工程师来办公室做面对面的沟通,另外对一些相对资深的工程师,我也会上门去找、或通过前的朋友推荐,或者大家一起去吃个饭,经常做一些交流,不断地去接触,这样也会花费很多时间。

EGO: 那您对招聘是否有自己的心得体会?

王栋: 有几个比较深的感受,在我们规模比较小、知名度比较低的时候,招聘其实是比较困难的,最终发现,现在留下来的,而且工作特别好的,其实都是对这个产品、团队特别认可的人。这些人可能会更多的去关注产品的愿景是什么样、怎样把自己的想法融入产品里面等东西,而且思考的内容可能比你还好、还要更扩展。这样,他们的主动性会特别强,能力本身只要不是技术特别差,我们认为有这个主动性,他就一定会成长。

稍微中期一点，我们稍微有一些知名度了，也希望招到能力更强，或者有过大型互联网公司背景的人，但我们更多的还是要看他对产品本身是不是感兴趣、愿不愿意很好的融入到团队里、愿不愿意去做一些脏活儿累活儿……因为在一个小的创业公司里面，我们没办法明确说，你只负责这一块，不用做其他任何东西，这是完全不现实的。

当然随着业务的不断发展，会有很多的外部曝光、融资等机会，会加强公司对优秀工程师的吸引。其实现在我们团队的主要的构成，就是有过一线互联网公司工作经验的工程师，可能没有特别耀眼的光芒，但都是有一定能力，又比较有潜力，又希望去做一个好产品的工程师。

有一点需要关注的是，如果一些工程师特别级别高，或者在社区里面特别有名气的，有时候反而不是我们这种公司比较好的选择。一是因为他们本身的期望会比较高，而且在一定程度上，他们的级别可能导致他们已经脱离一线代码工作，对我们创业公司来说，会需要所有工程师都会在一线，不管是查bug也好、写代码也好，可能并不需要那么多纯粹的管理者。

还有一个反面的经验，如果一个人其他方面都很好，但就某一方面有些缺陷，通常情况下，我们的任何一点点犹豫，到最后，犹豫的这个事情最后一定会变成现实，要不就是能力有问题、要不就是沟通有问题、要不就是入职很短的时间就离开了。

EGO：从业至今，您遇到的最大的挫折和困难是什么？有何收获？

王栋：第一个，我自己之前也有独立创业的经验，大概有2年左右，其实比较艰苦，人也特别少，而且那时是在04年左右，年纪也比较小，当然能力也欠缺非常多，只是凭一种感觉去做。当时人也招不来，产品却一直在做，压力也比较大，持续高强度的去工作，最后也没有一个特别好的结果，最后就不了了之了。从现在回头来看的话，当时如果能再成熟一点，这事而可能能做的更好一点。

后来加入了一个处在创业中期的公司，在那里，经历了从小到发展的比较好，然后又碰到瓶颈这个过程。最开始进去的时候，整个团队的气氛、士气都非常好，但不知道为什么，突然就碰到了一些问题，然后就是团队成员的流失、业务增长的停滞。但在那里真的学到非常多东西。

这两个都是印象比较深的，也都对之后做事情会有一些感触。一是，做事情你一定要有一个非常明确的目标和愿景；二是，一定要坚持，当你碰到困难的时候，不能放弃，一定是有各种各样的方法能够解决这些困难的；三是，要在任何时候都做好会随时碰到困难的准备，即使在你情况最好的时候，也要多思考如果碰到困难该怎么样去应对。

（原文节选，完整版文章请查阅EGO微信公众号：egonetworks）

阿里巴巴 是如何发展和激励人才的

原文作者: EGO编辑 陈园园

【张霞，福道诚壹咨询公司首席顾问，原阿里巴巴人力资源副总裁。先后工作于摩托罗拉、微软中国、Towers Perrin。2004年加入阿里巴巴，先后任职薪酬福利部总监、集团副总裁，曾负责B2B中国事业部人力资源、集团组织部管理等工作。2012年创办福道诚壹管理咨询公司。】

本文根据张霞老师在2015年12月19日ArchSummit全球架构师峰上的分享内容整理而成。

2004年我加入阿里巴巴，那时候也不知道它是不是个好公司，但很幸运在那儿工作了将近8年时间，看着他一路成长为今天这样。

2012年从阿里巴巴出来之后，我就跟创业者打交道，这4年我见过的各种各样失败的经验，写一本书都可以写好多。这么几年对比下来，我真的想告诉大家，如果你真的想做一个好公司，是有普适的规律可以学习。

CEO/CTO一定要学习，就好像程序一样，不能把自己写死，扩展性要很强。也就是，今天晚上学了半小时之后，明天的你跟今晚的不一样才行，要写出一个有成长性的程序。CEO/CTO要让自己成为这样的人，如果这样，一般你的公司不会太差。

今天主要讲一些互联网时代的人才发展与激励，是我的一些经验和心得，但有些东西还有待考证，不一定百分之百准确或适用于你。

创业都是人的事儿，所以要有共同的价值观

创业完全都是人的事儿，人搞定了，人心顺着往前走，业务就能很顺利的往前走。尤其是对年轻创业者而言，最难受的就是人。但是，不管跟你一起干活的是你的同学还是从外面招来的人，要注意的第一件事情就是，咱们是在做一件事儿，不是在处朋友，必须要有一颗公心。在一起共事的人都要有共同的价值观，都想成就一件事儿，而不是想你好我也好，这样才能把事情做好。

价值观是什么？你现在确定了公司要做这么一件事，希望所有人都围着这件事转，但是想让那么多不同的人在一起做事，你就一定要做价值观，要确定一套标准，在这个公司怎么做才是对的、是好的。

阿里巴巴在价值观上做得非常好。

第一, 要求他必须要积极融入团队。我们在招人时就会选择那些比较好融入团队的人, 而不是那些性格不好的, 除非他特别有才华。

第二, 要求他积极分享。我们会要求他在融入团队之后, 还要将自己的经验分享给别人, 而不能有自己的知识是专利、谁都不能说的思想。互相分享, 你才能更好的成长。

第三, 用正能量影响员工。公司会明明白白写清楚, 在这个公司, 哪些事情是被大家认可、哪些是不被认可的。

到最后, 价值观其实不是领导的洗脑工具, 而是我们所有人一起工作彼此约束的法则。

其实, 当把价值观实行四年、五年之后, 真正受益的是员工。一旦我们引进了跟价值观不太吻合的人, 或者他的行为跟公司价值观不吻合, 最先不舒服的是员工。他就会跟领导说, 公司现在的价值观不是我们以前的价值观了, 现在不如以前好了。

所有的使命、愿景、价值观, 如果你坚持五年, 它们就会变成你的文化。

不管你搞不搞, 文化总是会有的。如果你不搞, 你的公司很随性的就会形成一个文化。基本上CEO和核心团队是什么样的, 价值观、文化就是什么样的。团队就很容易出现揣摩领导的氛围, 就会形成一个没有正向文化的公司。但是如果你有意识的去做的话, 就会更理性的发展。

激励人才: 裁掉一半人, 让员工动起来

1. 把人用到极致

在职场里, 把人用到极致的时候, 其实是他最开心的时候, 也是最有成就感的时候, 尽管会很累。做人才盘点的时候, 凡是没有激情的同学, 我会跟他的负责人说, 肯定是工作量不饱和, 没事情做了。所以, 创业团队, 轻易不要加人, 先把现有的人用到极致。

举个例子, 有个团队说我的员工没有激情, 听说拓展能让人变得有激情, 他们就搞拓展, 搞了之后发现这些人不参与, 还是没有激情。所以他们又搞绩效体系、搞绩效考核, 想把绩效、职级体系做得更科学。我跟他开了一个电话会议, 告诉他不用搞拓展, 只要把人裁掉一半就可以了, 好多工作干不过来的时候不会没有激情。当然也有些人想过朝九晚五的生活, 会因此干不下去。

所以想要激励团队的每个人, 创始人作为领导者一定要学会闻他们的味道, 看他们的电脑屏幕上是游戏还是工作, 看大家每天晚上工作到九十点钟还是特别松散。如果一个创业公司早上10大家才陆陆续续进场, 下午7点不到人就走光了, 这个公司肯定没有什么前途。如果只是朝九晚五的工作, 你的公司离死也不远了, 因为你是创业公司。

2. 要有评价体系

互联网需要有评价体系, 这个人干的好不好得说一下。对于创业公司早期, 评价体系是可以没有的, 因为初始几个人需要在很短时间内把模式冲出来。但是超过半年、一年, 就必须有评价体系。

前几天我有一个客户，他有个干了十年的员工，是前两个加入公司的员工，但因为是自己人，这十年也是没有一个评价给他的。随着公司越来越大，CEO对他的工作不满意，他这几年做得也总是不开心、不满意。但是因为没有做评价，他其实也不知道不满意的到底是什么。现在公司做股票，我们帮他做股票分配，在一起工作十年的这个人提出来说规则是什么，这十年我给公司做了这么多贡献，你是怎么算出来我的股票的，跟我的贡献是什么关系。

现在去倒十年的老账你还倒得出来吗？所以评价体系过个一年半载一定要有，因为你需要去激励员工。大家到这儿来上班，不管短期利益还是长期利益，利益是最重要的事情。我在摩托罗拉工作的时候，股票市场还没有被炒的这么火热，还没有什么市值概念，所以我们那时候就是工资和奖金，最多加利润分享。现在这个时代，股票已经被资本市场当作激励成员工重要的手段。

各得其所的时代，打造平等开发的文化

1.各得其所

由于互联网时代的扁平化和信息透明，大家变得各得其所，你想想看有这么多人出来创业，不愿给公司打工了。我印象很深的，李琪是跟马云一起创办阿里巴巴的第一代COO，跟着马云立下了汗马功劳。我跟他聊天，他就说我这辈子从来没有想过给别人打工，我上大学就想着创业。那个时候创业的人还是很少，主要是像我这种给别人打工的，但是现在你们看有多少人。我从2012年开始给CEO讲课，都不止几千人了，像牛文文做的黑马创业领袖，每个班都能招到100个CEO，整个黑马会的会员都是CEO，也是一个很大的规模了。

所以，这是各得其所的时代。如果你的信息还是不透明，他心中有一些相干的事情在公司内不能实现，或者你不知道，或者你知道了也不给他做思想工作、不跟他讨论，他就一定会离开。因为现在的工作机会非常多，这也是为什么你要尽最大的努力让公司员工有各得其所的感觉的原因。

我认识一些阿里巴巴最基层的员工，会发现有些事情对于创业公司还好办，他们的团队可能才两三层，CEO、核心团队和普通员工，但是阿里巴巴一个3万多人的公司，层级非常多。所以高层希望打造的文化，基层人员就完全感觉不到了。

举个例子，我们一直认为公司内部换岗是一件简单的事情，公司那么多岗位对内对外招聘，你想换工作就递你的简历呗。但是发现员工不干，为什么？因为员工跟主管说，主管不同意，这还是个互联网公司，那么致力于打造平等的文化，但是到了最基层的员工那里，你会发现他得到的信息还是非常少的。

那怎么办？阿里巴巴就做了一个内网，把所有的信息都公开，比如公司所有关于加薪、晋升、轮岗的话题都可以讨论。

后期做得越来越好后，员工的胆子就不会那么小了。我印象很深的一件事，有个员工的主管给他打了不合格，他马上辞职不干了，临走前还在内网上写了给马云的最后一句话，那个帖子阅读1000多，变成了神贴。CTO、总经理、直属领导马上就跟员工开展一个完全开放的视频对话，全程播放给所有员工，问这件事情的前因后果是什么、员工有哪些不满意、主管有哪些不满意。等到透明之后，你会发现员工不需要你给他洗脑，公道自在人

心，有人觉得主管做得不对，也有人觉得员工做得不对，所有人都获得了学习。整个公司所赞成的东西会变得非常显性化，这就是互联网时代带来的开放价值。

2.各尽其才

我们希望利用互联网时代的文化，互联网是一种生产力，而生产力决定生产关系，它既然这么开放透明，那你封闭也没有用，因为总会有人知道并公布出去。未来，互联网的开放性会逐渐凸显出来，而在公司内部更容易做到这件事情。

既然一切是透明的，那么是你的责任你就必须把它做好，你偷懒不尽责，不仅是领导不认同，所有的员工都不会认可，这是普适的值观。所以，我觉得未来一定是各尽其才、人尽其用的社会。

现在我们做什么事情都是先拉一个群，领导、总监、普通员工所有人都在里面，全部公开透明，减少一对一的沟通。一对一的沟通往往会产生很多信息的不对等，造成各种猜测、揣测，导致节奏非常慢。

举个例子，很多人经常会有一个问题，工程师资源到底花到哪里去了？阿里巴巴原来有个工程师总监说某业务人员是防火防盗防某某。为什么？因为那个人跟工程师关系好，经常找工程师乐子，没有很好的分配任务。

我还有一个朋友，他的公司很随意，项目说上就上、说下就下，不开放也不透明，工程师一年又一年的时间都被荒废掉，没有做任何有价值的事情。

所以，一定要用这个时代的开放透明，让每个人都各尽其才、各取所需。

3.各取所需

现在公司每个层级的工资多少钱，大家都是知道的，像阿里P7是3万、P8是4万到5万之间。未来一定会走向透明，这是趋势。虽然不公开，有些矛盾就不会被公开化，但是职级是绝对公开的东西，是总监还是专家都会明确印在名片上，所以非常透明。包括股票，一上市，最起码前面那些人都是公开透明的。

所以，我们在激励人方面是完全公开化进行的，包括上什么课。现在互联网时代上课很容易，什么都可以学，机会非常多，成本也不是那么贵，像这次的球架构师峰会，大家都可以来学习，所以员工都是自己在培养自己。

当然作为公司，资源肯定比个体员工多，但是我们认为成长是自己的事情，你想成为总监、成为架构师，你就应该自己训练自己，让自己学习。所以最后就是各取所需，互联网时代学习那么容易，难道还让别人教你不成吗。所以我觉得创业一定要做到各得其所、各尽其才、各尽其忠、各取所需。如果做到这样，你的公司一定很棒。

如何在中国打造拥有谷歌工程师文化的团队

原文作者: EGO编辑 陈园园

【在2015年5月的Qcon大会上, 出门问问&Ticwear CTO雷欣从Coding、开发环境以及团队合作等角度切入, 分享了谷歌的工程师文化, 以及他和他的团队如何在中国打造公司的工程师文化。】

一、什么是谷歌的工程师文化

(一) Coding

在加入谷歌之前, 我曾在微软和斯坦福研究所工作过。这两个地方其实是非常不一样的, 微软是一种传统软件的开发模式, 而斯坦福研究所是一个研究所, 主要工作更偏向研究方向, 更看重创新等能力, 反而对代码质量等的要求不是特别高。到了谷歌之后, 我印象最深的是, 谷歌对代码质量的要求和追求到了一种近乎狂热的程度, 主要体现在几个方面:

1. Coding Style

谷歌内部有一套非常长的Coding Style Guide For Every Language。其中C++是谷歌的主要语言, 包括其他语言如Java、Python等, 不管什么语言, 谷歌都有一个很长的Style Guide。

2. Readability

在谷歌, 当你要想写程序的时候, 要先拿到所谓的Readability。假如你的代码质量不过关, 没有通过Readability考核的话, 那你的代码可能就没法Check In到代码库里面。

3. Unit Test

谷歌有非常完善的Unit Test机制, 也有十分强烈的Unit Test文化。大家都知道, 写软件其实跟搭积木很像, 如果下层基础不扎实的话, 那整个软件框架其实是很危险的。在谷歌, 差不多每做一次代码改动都会有一个Unit Test, 虽然这会让开发变慢, 但它也意味着高质量和可靠性。

4. Code Review

谷歌很重视Code Review, 基本上谷歌所有代码都会保证至少有两个以上的人对它Review后, 才会让你Check In。

5. LGTM (Look Good To Me)

最后, 你的代码写完之后, 你需要得到这样一个“LGTM”, 就是“Look Good To Me”, 得到这样一个Prove后, 你才能把这个代码提交上去。

(二) 高效的开发环境

1. 强大稳定的Base库

除了对代码质量的高要求, 谷歌的开发环境也相当强大, 非常高效。在谷歌, 成千上百万的代码用的其实都是同一个代码库, 大家共享一套强大稳定的Base库。

这点是我以前在微软的时候没有看到的, 在微软, 我们可能是每个小组都有自己的代码库, 而且每个小组都

有自己不同的代码规范和风格，因此，当更换团队或跨组工作的时候，不同小组之间的不同标准就会造成一些困难。

但在整个谷歌，所有的代码规范都是统一的，因此可以这样想，假如把代码质量看作一个函数，描述成一个高斯分布的话，那谷歌的总体代码是高斯分布的Balance，方差是非常小的，整体质量是非常高的。

2. 代码构建工具Blaze

代码的话，谷歌有一个构建工具叫Blaze，这是一个好东西。我们写代码的人可能都知道Makefile，但这个工具的话，我记得以前经常会花更多的时间在调这个Makefile，而不是写代码上。但谷歌的Blaze就让事情变得简单了，它可以让我们集中精力在代码上，进行快速并行编译，并且效率非常高。实际上，很多前谷歌人从谷歌出去后，想到的第一件事都是怎么重现谷歌的这个框架，怎么能够重新搭建这些东西，让写代码变得更加容易。

其实，很多公司都有类似的工具，比如Facebook有Buck，Twitter有Pants，国内有个更加山寨的Blade，实际上这些东西都是想要达成Blaze的机制。

谷歌还有其他很多好东西，譬如Protocol Buffer，就是一种数据交换的格式；Stubby，谷歌后台系统基于HTTP的RPC机制；还有Map Reduce、Big Table、GTest、GFlags、GLog等等。

总的来讲，这些东西能让开发更高效。

3. 敏捷的开发团队

谷歌的开发团队一般都是比较小、比较敏捷的。以前谷歌的CEO Eric Schmidt说过这样的话，“Google有影响力的项目都是2个人的小团队开发出来的，如搜索、广告、Gmail”。

一位在腾讯的前谷歌人也说过这样一句话，“3-5个人做不好的事情，30-50人做的更烂”。

我是比较倾向于认同这句话的，假如一个项目在初始阶段，没有靠那么两三个人把框架搭得非常好的话，你再招更多的人只会是添乱，增加沟通、交流的成本。一个很小的、2-3人的小团队有什么好处呢？你能实现快速开发、快速上线、快速迭代，然后再去推进这个产品的改进。

4. 一切自动化，数据驱动

谷歌还有一个特点是一切自动化，数据驱动。

我们以前在谷歌是做语音识别的，语音识别中其实有很多东西是需要人手动去参与、调解的，比如一些参数类的东西。当时，我们组里面就有人提出要做这么一件事情——当时谷歌有40、50种语言的识别系统，那我们做这样一套系统，我按一个键，它过一段时间就能把所有的、每一种语言的语音识别系统全都训练出来，然后全部自动推上线，中间过程中不需要任何人的参与。

当时就有人提出了这样的想法，这就有点将工程做到极致、做到最优的理念，这是非常重要的。

二、如何达到谷歌的代码质量

那我们怎么在一个小的Start Up中间达到谷歌的代码质量呢？我有一些建议：

1、坚持原则

一定要坚持原则，坚持Unit Test和Code Review。

Start Up的步伐、步调是非常快的，各方面的竞争也很激烈，可能你不快的话就会落后。在这种情况下的话，很多产品经理面对各方面的压力，可能会“我只需要这个产品快速出来，我根本不Care你里面是怎么做的，你的代码质量、代码风格，Whatever，我不Care。”

但实际上，我觉得这样是不太好的，因为，从公司的长远角度来讲，你把基础框架搭好之后，对以后的发展、稳定都会有很大的帮助。因此，在很大程度上，要坚持Unit Test、建立Code Review的机制，定要保证你的代码是有一定质量的。

2、拥抱开源

第二点是要拥抱开源。实际上，谷歌是一个非常大气的公司，它已经把很多很好、很酷的内部的东西都开源出来了，我们要做的话，就是需要把这些东西积攒在一起，然后搭建一套更适合于创业公司的开发环境。

3、鼓励更新、快速迭代

很多时候，我们会面临一个问题，代码的第一个版本出来之后，那我以后是在这个基础上修改呢，还是在一定阶段把它推翻了重来呢？

谷歌的话，它内部基本上都认为，没有什么代码是能活过超过2、3年的，所以，会对代码进行推陈出新。

实际上，在代码的推陈出新之间，会把以前没有考虑到的东西会做得更好，也会锻炼新人，然后学到更多东西。

三、我们怎么做产品

同谷歌一样，我们推崇“快速上线、持续迭代”，认为“天下武功，唯快不破”，所以我们一般都是以2-3个人为一个小组去集中攻坚一些比较难的问题，寻求快速的突破。

每周，我们都会进行产品的迭代，像我们最近做的智能手表操作系统Ticwear，每周的话我们基本上都是，周一到周三开发，提出Bugs和Features，然后周四进行内部测试和粉丝内测，然后周五就马上公测。

维持这样一个很快的节奏，实际上是非常累的，但是在这个过程中，你实际上锻炼了团队，你的产品也可以更快的满足用户的需求。

四、打造一支谷歌工程师文化的团队

1、招什么样的人

第一是要通才(Generalist)而非专才(Specialist)，这个是非常重要的，通常来讲，一个创业公司跟一个大公司的差别在于，创业公司的资源是非常有限的，每个人可能要做很多不同的事情，所以，对我们创业公司来说，一般想招的是通才(Generalist)，而不是专才(Specialist)。

大公司的话,因为它资源非常多,人也非常多,所以他更希望某一个人进来之后,能老老实实做一件事、做一个螺丝钉,把这个事情做得最深、最透,但是我们小公司的话,更多地希望员工能有更多的Coverage。

第二是一定要有分析和解决问题的能力。

第三就是执行力,小公司之所以能够在某些程度上做得比大公司更好,靠得其实并不完全是Idea,因为Idea是很容易被抄袭的,实际上拼得是执行力,假如你执行的不够好的话,那你就没有任何机会。

第四是要有一定的交流能力,就是说,虽然我们非常鼓励员工能有强大的单兵作战能力,但我们同时也希望在2、3个人这样的小团队中,他们能有非常高效的交流、沟通,去推进一些项目。

2. 面试流程

其实我们的面试流程基本上都是Follow谷歌的那一套,都是Phone Screen 再加上3轮Onsite Coding Interview,每一轮Coding Interview都是必须要在白板上写代码,其中就包括系统设计、算法、编码等。算法会多一点,可能会有几个,还有就是编码,我们希望说,一个算法,你想出来了,也能通过你写的代码呈现出来,理论上,你的代码写完之后,我放在编辑器里面,它没有错误,能够编译、能够跑通,这是我们的要求。

3. 培训

那人招进来之后,我们怎么去培训他们呢?当然会有一些基本的Coding Style的培训,然后我们会找一些导师,给他们设计一些比较独立的Starter Project,让他们能慢慢融入这个团队。同时,我们也会有一些Codelabs,在某些专题方面,教教大家怎么用我们搭建的一些模型。

平时,我们还会有各种各样的活动,比较大一点的有问问讲堂,我们会请一些比较知名的专家、学者来给我们分享一些技术、业界的最新的进展。同时,我们内部也会有一些技术分享,还会搞一些Hackathon,做一些Self Review、Peer Review,让大家更好地融入这个团队。

4. 在中国招人留人的困难

我实际上回来才一年多一点,就发现在中国招人和留人有很多困难。

A. Risk Taking。我发现,其实很大一部分中国年轻人承担风险的能力、Risk Taking的能力,相对美国来讲是比较低的。

我们在招人的时候,经常会碰到,有些人会说,“我爸妈认为大公司好,觉得BAT好”,或者说要有北京户口,这样的情况,这些我在美国的创业公司是很少见到的,所以这个也是我们比较头疼的。

B. Temptation。即使我们好不容易招到人了,那么怎么去留住他们也是我们比较头疼的地方。因为我们的要求比较高,所以我们招到的人、特别是后端的人才,素质都还是比较高的,那么,特别是在现在这个创业环境特别好的情况下,他们就会面对很多各种各样的诱惑,譬如说,我们公司至少有3个工程师走了,其中有2个到了别的公司当Director,另外一个到别的公司做CTO,他们都只是一般工程师啊。

所以这种情况下,你招人、到底要招什么样的人、完全招那些最强的人是不是最好的一种方式,也是值得我们去思考的一个问题。

赵海平: Facebook 研发团队的高效秘诀

原文整理: 徐川

【极客邦科技旗下的技术人社交网络EGO首场活动在北京国际会议中心举行, 阿里巴巴技术保障部研究员赵海平在活动上分享了Facebook研发团队的高效秘诀, 以下为正文。】

大家好今天主题是讲研发团队高效秘诀, 我想说, 我要讲的很多Facebook团队的特点, 都是跟它的历史是有关系的。我会讲(Facebook)这个团队怎么样的从小到大, 一点一点起来, 才慢慢形成自己的风格。这个团队我其实并没有系统的研究过, 我就简单的讲几个大的方面, 一个是公司文化, 还有就是整个内部组织结构, 最后可以讲一讲Facebook的人, 他的思维的方式是什么样子的。

Facebook公司文化: Hack

第一个讲讲公司的文化。这个文化特别明显, 叫Hack, 它不是说黑客, 而是很快的把事情攒出来, 攒出来之后不一定特别好, 第一个版本往往很差, 但是不要紧, 可以后续再迭代改进。在Facebook早期的时候, 总共就三四十人, 每个人负责的事情很多, 一个人要负责网页上的一大块内容, 你其实没有很多时间把事情想清楚再做产品, 你肯定是急急忙忙把它攒出来的。

于是慢慢形成这样一个情况, 做任何一个产品的时候, 第一个版本不在乎是不是做的好, 它可能漏洞百出, 但是我可以确定两三天或者三五天就可以上线, 做出来后放到网页上马上就得到反馈, 来自用户和周围同事的反馈, 或者你自己看出毛病, 然后你就知道朝哪个方向改, 然后第二个版本就好一点, 第三个版本更好一点。这个思路就是, 我不求这个迭代很快达到我最终想要的那个样子, 但要求每次迭代速度很快, 这样很快我就朝着正确的产品方向走了。因为很可能你花了很多时间精力, 坐那儿想了两三个星期, 觉得这个东西不错, 想的特好, 然后推出去, 结果用户发现这个东西不是他想要的。尤其我们在做面向用户的服务或者产品, 经常有一种现象, 你自己觉得不错, 自我感觉特别良好, 但是拿到外面, 用户反映完全不是那回事。Facebook就是这些年从小到大, 摸爬滚打中渐渐总结出这样的经验。

公司早期说Hack这个词很少, 那时候大家都忙着攒网页, 攒这个那个, 后来大家慢慢意识到Facebook的文化就是Hack文化。后来我们搬了很多次家, 到最后搬到现在这个最大的楼, 我们就在大的水泥地上写了四个大大的字母“HACK”。这个真的是一个公司多年文化的结晶。

即使到后来公司发展壮大, 有了团队区分的时候, 大家还是讲究这个, 要Hack, 不要担心自己的第一个版本不完美。因为人的思维永远是有局限性的, 你就是坐那儿想十天, 也不会说那个完美的程度达到十倍, 想十天可能达到两倍的完美, 这是不值得的, 万一你产品更改方向, 你不是白白坐那十天吗? 所以说Hack文化是一个特别有效的手段。尤其我觉得公司还小的时候这招还是挺管用的, 因为小公司没有那么多人力物力, 不

太容易把一个产品搞的那么完美, 所以你可以Hack一下。

我还想说Hack是可以传染的, 既然你Hack我也Hack。如果你Hack的话, 我不用Hack就比你慢, 这个人花三天时间就推出来, 我不用就要十天。一旦形成Hack文化, 它会慢慢波及公司各个方面, 大家都开始Hack。但是大家不要觉得Hack是不敬业, 或者对产品不追求完美, 它只是说把对产品完美的追求分成多步去走。像是说你看我们公司都是什么什么学校毕业, 都是能人什么的, 但是我们不求把自己做的东西搞的很神秘很高大上, 说要推出一个震惊世界的产品。初期的产品不用特别好, 只要你做出来就行了。

公司的文化在很大程度上就是这样一个词所总结的。其实我刚刚进公司的时候, 并不是说我完全接受这样的文化, 因为我是完美主义者, 我的东西我一定要做的超级完美才拿得出手, 但是跟这些同事一起共事, 渐渐就领略到Hack的高效, 它实际上有一个非常坚实的理论基础, 我们也很幸运, 在Facebook早年的时候, 就吸取这么一个公司产品开发的一个做法。这是公司文化上一个很大的特点。

组织结构: 去中心化, 避免依赖

再有一个我想讲一下这个公司的组织结构。Facebook说白了, 它就是一个非常去中心化的组织, 并且尽量避免对别人的依赖。一个团队只管自己的事情, 不但如此, 在做自己事情的时候, 尽量不依赖别人。这个也跟公司里边的人工作方式有关系, 很多小孩他们一直工作到夜里三四点钟, 中午睡到十二点, 下午一点才到公司。假如夜里三四点工作遇到问题, 你让他找谁? 所以他不愿意自己有依赖性, 不愿意他团队做的工作对公司有依赖性

这不意味着公司不愿意分享Code Base, 事实上Facebook只有一个Code Base, 任何人都有权限修改。有时候代码有可能会搞坏, 但是搞正确的可能性更大。如果说搞坏损失十万块钱, 但是搞对了对我们今后一直是有好处的, Facebook一直不断地在靠这个创造收入。所以最后平衡下来, 衡量下来还是要去改的。

在Facebook经常说的一句话, 就是我不怕失败, 我不怕把这个事情搞砸了。这个是真的, 在Facebook历史上搞砸的次数太多了。有一个高中毕业的学生, 被Facebook直接录用来了, 原因很简单, 非常有才华, 写代码又快又好。来Facebook看这儿不好那儿不好要改, 这么一个有激情的男孩儿, 你怎么让他不改呢? 那是不对的。所以他改了, 改了第二天就出事了, 整个网站都被他弄宕机了。我们来了以后都面面相觑, 原来那哥们儿把网站的header-cache改了。那时候大家把他嘲笑一顿, 说你怎么能够把自己的网站搞宕掉呢? 他自己也很害羞, 自己去喝闷酒。但是不会有有人说, 不应该这么做, 这个我觉得特别的关键。

你要鼓励他去失败, 这样的话他才会意识到自己是主人翁, 才会让每个人都觉得, 整个code base, 每件事情都是我来管, 这样的话我才对其他的团队没有依赖性, 在没有依赖性的情况下你的团队才走的最快。你自己应该很明确自己要干什么事情, 任何阻挡我把这件事情完成的东西都要排除掉, 这样的话才能真的很快把这个东西做出来。只有你养成这种习惯后才能影响周围的人这样做。只有大家都这样做的话, 公司才会很快速的发展。所以07-09年, Facebook任何的产品都是以飞速的, 甚至大家觉得是不是有点太快了的速度在发展。

不过, 每个人都可以改代码库造成一个问题, 最后我们的PHP的Code Base变得五花八门, 因为每个人都有自

己的想法和主张，就把整个代码改的乱七八糟的。这个时候就需要有一个Key Person，这个人非常强，他站出来说明不行，同学们，这个代码被你们改成这样了，我要重写一下。Facebook早期的员工真的是有这样有才华的员工，他可以花几天几夜时间把Code Base重新写一下。

所以在实行Hack文化的时候，一定要有特别重要的人，他让你在乱中还有一些静，能够重构和梳理公司的代码，因为Hack出来的代码肯定是很乱的。大家看到他把我的混乱的代码处理好了，特别感激他，然后下一次我就遵守他制定的规则。这种情况在Facebook屡见不鲜，总有人站出来说明这个不行，我要来树立一个章法。在树立章法的时候，也是一个民主的过程，不是说靠奖金什么的，而是靠技术，这样的话，在技术的引领下，很多工作就这样慢慢展开了。

当时我做PHP Profiling的时候，也有人提出其它的看法，要不然咱们干脆停下来把网站改成Java，也有说要不然就把PHP改了，这些路不是没走过，但是都不通。因为你做这些事情就对其他团队造成依赖性了，实际上你在央求别人停下来去做你让他们做的事情，这个在Facebook是行不通的。在后来2011年、2012年有个很厉害的人总结过一次，他跟整个Facebook的人说，我看了一下，Facebook所有成功的项目都有一个特点，就是真的不对其他任何团队，不对其他任何人有依赖性，只要是透明化项目，只要你这个项目做出来不需要太多其他人参与的，做出来对大家都是有益的，这些项目是容易成功的。在这么纷繁复杂的组织里，不是说他不认可你的想法，他有自己的本职工作，所以你不能依赖他，他说我很忙，把你这个事安排在下个月，渐渐这个事就泡汤了。

不过，我把这个（去中心化）讲的特别好，并不是说它没有弊端，其实有时候它是很有弊端的。举一个特别极端的例子，扎克（马克·扎克伯格）向大家抱怨，他说，你知道吗，本来我想做一件事，可是我找不着任何人帮我做这件事，因为大家手头都有自己的事情。他说我想做这个，为什么没有人帮我做这个事情。他就在那儿哭诉这个事情，我们大家都笑了，就是没人理他。这种事情在早年发生的特别特别多，到后来公司的人多了，公司有大的项目的方向，人也多了，可能扎克就能找到人帮他做这件事了。

思维方式：善于争论，用数据说话

第三个我讲的是思维方式，思维方式其实跟文化有一些类似的地方，Facebook的人有才华，他有自己的思想和主见，他对很多产品的方向，对很多事情都有很强的想法。强到什么程度，强到员工之间经常争论不休，这个在早年的时候特别严重，到2010年的时候，公司说不行了，同学们，这样争论太伤感情了，大家互相之间还是要有一份同情之心，毕竟我们在一起工作了这么些年，还是要有一份爱在这儿的。这从另外一个角度来解读，就是Facebook人之间的争论，有的时候真的特别激烈。

但是争论归争论，我们很明确的知道，将来怎么解决这个争论。怎么解决？就是用数据去说话。你可以有你的想法，你说这个网页必须按照你的设计，我也可以有我的想法，我说网页是这样的。你可以争论，但是是骡子是马拉出去遛遛。

最后就是大家争论太多了，发展出来一个非常完善的AB测试系统。所以看Facebook软件系统真正的功能性的code写的可能有20%、30%，但还有40%的code都是去帮助这百分之二三十去搜集数据，去印证它好还是不好。它花了我们很多精力到网页上搜集数据，去反过来看我们这个运行情况是怎么样的。无论是UI的，还是整个流程，还是整个非常大的产品也好，实际上都是这样的，都是去试出来的。这个试出来的背后多多少少都是有争论在里面，因为真的是大家谁也不服谁。

甚至公司的这种思维方式有时候会非常极端，极端到什么地步？Facebook的人这样认为，任何事情我都可以评论，我都可以发表我的见解。大家认为我们公司的愿景啊这些，无论任何的事情，我们都是可以拿出来讨论的。

最逗的一件事情是差不多2008年时候，那个时候公司开始完善了，公司说，员工们，其实我们真的是一个公司，我们是有假期的。我们有三个星期的年假——HR的人说三个星期的假是指三五十五天的假。他发邮件的时候是说三个星期的假，但是很多人都理解三七二十一天。等公司公布的时候大家就开始争论，说这个不对，应该按二十一天，有人说你太可笑了，明明是十五天。结果你猜Facebook后来做了什么？Facebook把这个HR炒了，说你连邮件都不会写，让大家有这么大的争论。与此同时，Facebook直到今天也是每个人有21天的年假。这个虽说是一个可笑的故事，就是这么一个事情被Facebook争论成这样了，更不用说有理有据的争论。

这个思维方式就是这样，他认为我跟你争论是OK的，我跟你争论并不是我不尊敬你。大家最常说的，就是我特别尊敬你的，但我不同意你的想法，我认为这个事情是这样，然后列出一二三四五。其实大家都在争论当中成长，达到共识，慢慢形成同样的思维方式。所以我觉得，如果说公司还小，真的可以花一些时间和精力去开发一批这样的工具，让公司内部员工可以很好的有一个地方去争论这些事情，而且不要怕大家争论，不要怕大家有不同的意见，这才是真正的公司员工，这是把心思放在你这个公司上的表现。

你可以去控制一下场面，让大家在争论的时候不伤和气，但是最后还是要鼓励员工百花齐放，百家争鸣的氛围。这个氛围其实Facebook到后来的时候是越来越弱了。要说2013、2014年大家都非常非常客气，就没有表面的冲突，别人要说一件什么事，要说有不同意见的人的话，他可能只是小小的，弱弱地问一句，再不然的话，就是底下很礼貌地讲一句话，你觉得这样好吗？其实这表面看起来很平静，但是没有把尖锐的问题拿到桌面上。

公司很小的时候，或者说需要一个高效研发团队的时候，你必须要把很多的问题拿到桌面来探讨，而且希望大家在争论的过程中，大家真正地意识到自己才是那个主人公，自己不是帮公司打工的那个人，自己正儿八经用大脑真正思考公司面临的问题。

创业公司要招强人

最后如果强调的话，就像我刚才提到的，Facebook早期的员工有一批人非常杰出，就像我刚才提到的那个高中毕业的小孩，他没有上过大学，但是他写代码的能力很强，是可以挑大梁的任务。早期至少有五六个人把这个团队的技术水平带到一个新的高度上，因此虽然用Hack文化，但代码质量并不特别差。所以在雇人的时候真的要做到宁缺毋滥，就是说我宁可花更长时间雇人，但是雇进来的人一定是一个特别强的人，因为他干的好，他会在无形中带动周围的同事。如果雇工作效率不是很高的人，这个团队就会跑偏了。所以刚开始的人还真是挺重要的。

我现在回想起来，我加入的时候有四五十人，这四五十人都是强手。现在即使回过头来，他们有人说，海平，你们这批人早期进来是Facebook的员工，是不是你们这帮人有运气？只是沾了早进公司的光，而后面进来的人比早期进来的人要强。我说这是毫无疑问的，后面会有更强的人，但是比例上来讲，后面的强人往多了算有20%就不错了，但是我们说早期的四五十人，强的至少在90%以上，这是一个很高的数字。90%的这些人甚至到六七年以后，能很轻松地负责一个大的团队产品开发。这些人具备快速学习的能力，他会迅速地把自己转化成所需要的那个领域的专业人士。早期的时候有了这批人，才能够把这个公司的文化给定住。如果说90%的人都是这样的话，那么剩下的10%的人形不成气候，而这90%的人把公司高效运转起来了，到了有五六百人的时候，有百分之五六十的人很强，以至于这个公司到后来越来越强。早期的员工在最后见面的时候，都有一种惺惺相惜的感觉，因为你打内心是很佩服他的，你知道他确实很强。我不知道别人怎么看我，我就是很自信，大言不惭地分享自己的感想，我们这批人互相见面，就觉得我们就应该是那批成功的团队，我们当年就是做了很强的事情。

所以早期做的时候就要抓住这点，使自己的团队有这样很强的人，你可以花很长时间去招一个人，或者接触的人稍微有点弱，就不要他了。说实话，一般的人可能做某些事情很熟练，看起来快，但很强的人给他一点点时间，也能把事做了，并且从整体上会做的更好。所以当公司开始起来的过程中，招到强人是很重要的。

（原文节选，完整版文章请查阅EGO微信公众号：egonetworks）

Uber首席系统架构师

Matt Ranney: 快速扩张 决定Uber今天的工程师文化

原文作者: EGO编辑 陈园园

【2015年10月16日, EGO“研发梦工厂: 番茄工作法+Uber团队管理实战案例”在2015QCon上海的分会场顺利举行, 现场邀请到了Uber首席系统架构师、Voxer联合创始人Matt Ranney, 为大家带来了Uber的工程师文化。】

我想给大家分享一下我们Uber的文化和管理, 这些在全球也都是关注度很高的内容。

这是我们近期的一些运营数据, 可以看到现在Uber覆盖全球61个国家340个城市, 每天完成3,000,000次乘车, 全球员工4,000人, 其中工程师1,000多人。

之所以需要这么多人、这么多工程师, 一个是因为Uber得发展非常迅速, 另外也是因为我们雄心勃勃, 我们想要成为全球领域的供应商, 因此需要在全球不同的市场提供服务, 而每个市场又有自己的独特性, 所以不管是业务、工程设计、还是技术服务, 我们都需要如此多的人员去支持Uber的快速成长。

当然要雇佣这么多人, 过程中我们需要去面试更多的人, 这就需要花费非常多的时间, 而且, 仅仅在一场比赛中, 我们很难了解这个工程师到底行还是不行, 所以每一个面试者要经过我们很多工程师的面试, 这样我们现在在职工程师就要花很多时间去面试这些新人, 这也成为他们工作的主要部分, 对他们来说也是一个挑战。

【Uber的文化】

促使我们快速向前奔跑的驱动力之一, 是希望能为城市带来更加便利的生活, 提高人们的生活质量, 这也是我们一直致力于做的。Uber一直在全球拓展业务, 发展非常快, 而这种快速发展也决定了Uber的文化。

1. MERITOCRACY

这意味着在所有的点子中, 我们会选择最好的那一个, 无论这个点子来自于谁, 也就是英雄不问出处, 只要是好点子就行。

同时, 在Uber, 我们鼓励员工挑战高层, 我们鼓励大家不要害怕去向自己的老板挑战, 只要你有好的方法、只要你认为你的想法能帮助到公司, 你都可以提出来。

我刚开始在Uber工作的时候, Uber缺少很多基础架构, 这个阻止了Uber的发展, 所以我不得不向高层去挑战, 一开始他们不高兴, 但是最终他们意识到这么做是正确的, 当然交谈过程并不是非常愉快, 但是我们还是要这么做。

2.LET BUILDERS BUILD

Uber的另一个文化是给员工一种真正的授权，尽量让真正做产品的人能有自主的权力，去开发这些产品、去发展公司，而尽量减少官僚主义。

3.ALWAYS BE HUSTLIN'

在Uber，我们倡导的并不是最快，而是要最有创意地去解决问题。你可以不找到一个问题的最佳解决方案，但你可以想各种各样的点子，最后用各种各样的创意去解决问题。

可以举一个例子，早些时候，世界上有些城市并不想要Uber进入他们的城市，所以他们会尝试去黑我们的系统。我们就想出了一个办法，我们包括很多工程师花了一个通宵的时间，另外我们还使用了其他的技术供应商来帮助我们重新打造另外一个版本的基础架构，通过这种方法防止他们来黑我们的系统，所以我说要很有创意的来解决问题。

4.CUSTOMER OBSESSION

也就是客户至上，我们客户是乘客和司机，我们所做的一切都是为了让他们的生活更加便捷、更加好，所以我们所做的一切决定都是为了他们，为了帮他们解决问题。

这几天我在上海使用Uber，我非常震惊的是，在上海Uber带给用户的体验要比在美国糟糕很多。我一直在跟我们的工程团队沟通，如何解决这个问题。我们如何才能让中国的用户拥有其他地方一样好的体验。所以在很短的时间里面，以及很少的邮件沟通之下，很多工程师都积极参与到这里面了，因为我们工程师都希望全球各地的用户都能够体验到Uber的好处。

5.SUPERPUMPED

这其实是我编出来的一个词，指的是一种态度，要超级有热情地对待工作，就像打了鸡血一样，在工作中要有非常高的积极性。

这里那我自己做一个例子，我每周要面试7位工程师，这是一个非常大的工作量，但是我还是非常兴奋地能做这些，因为我希望能尽量招到好的工程师加入我的团队，所以我每次面试的时候都非常兴奋。

6.BIG, BOLD BETS

就是大胆的下赌注的意思，虽然说Uber可以成为一个成功的公司、一种成功的模式，但是由于技术的发展是非常快的，如果我们没有办法做好规划，去应对未来5年技术的发展，哪我们公司就很有可能被颠覆。所以我们要尽最大的可能去对未来进行投资，这也是我们所做的大胆的赌注，赌未来的走向。

大家可以去了解，实际上Uber的投入非常大的，在技术方面的投入非常大，在自有车辆方面也投入了很多。我们所做的所有投入都是希望能够建立一个非常高效的运营网络，所以再过几年之后，我们相信很难有其他竞争对手和我们比拼了。

【Uber的工程师文化】

之前谈到的这些文化都是从Uber整体上来说的，都是公司层面的价值观，但我作为一个工程师，也特别希望从工程师角度来讲一讲Uber的工程师文化。

{Scalability、Availability与Cost}

这三点是我们优先考虑的几个问题，一是可扩展性，也就是近些年来我们一直专注于不断扩展公司的业务，如果我们不能扩展系统的话，就没办法满足来自于全球的需求。比如对Uber来说，一年当中最忙的一天就是新年之前的一个晚上，这一晚上的需求是一年中最高需求的两倍。

现在，Uber已经足够成熟，我们的系统已经能够应对这些高峰值了，所以我并不担心系统的可扩展性，我们现在就要考虑到可用性了，要想一想，在哪些地方、哪些城市能用到Uber，也就是说可用性有多高的比例，我们的服务是完全没问题的，是能够供应得上的。

第三点是成本，以后我们可能会考虑成本的问题，但目前为止我们完全不在意。其实我并不想把这一点让我们投资人看到，因为我不想让他们知道成本是我们最少考虑的，这并不是说成本不重要，但是Uber在这一阶段愿意做一些投入，我们更加关注系统可扩展性，而现在就是更关注到我们服务的可用性。

{POWER OF ENTHUSIASM}

重要的是这种热情，这关乎到我们到底该选择哪些技术去打造系统，因为我觉得最重要的是能够让我们的开发人员对未来技术是感到非常兴奋的、是非常有热情去做的，不是说他们觉得这个技术很无趣、并不想做。

比如在Uber，我们使用了Node.js的技术，可能很多开发人员觉得这并不是一个好的技术，但是我们却发现公司开发人员对Node.js非常喜欢，所以公司决定去拥抱这个技术、去使用这个技术。

{MULTIPLE TRACKS}

在Uber，对于工程师而言有两个发展方向，一个可以向技术管理的方向发展，另外他们也可以以个人技术的独立发展为发展方向。这点其实是很多软件公司所忽视的，但是Uber和其他几家公司非常重视这一点，对这两种不同发展方向是有同等路径的，技术人员可以自由选择自己想走的路。

其实管理人员的主要职责就是做管理工作，如果你想要继续编码，你可能更适合当个人工程师，因为作为经理，你要编写很多的文件，参加很多会议，这会是你工作的主要内容。

当当架构部总监 史海峰： 架构与管理是相通的

原文作者：EGO编辑 陈园园



【打造内部应用框架】

当当技术部现在是按照产品线划分的，一个产品线的产品、开发、测试都在一个部门，但像项目管理、运维、架构这些技术体系中公用的部分是独立的部门。架构部里主要分成三部分，一个是架构与规范，一个是性能测试，一个是基础应用系统研发。

我们花了比较多的精力在技术架构上，去年我们在Dubbo上做了二次开发，做了DubboX并且对外开源，业界反馈还不错，包括很多来面试的人都知道。

我们的技术体系、核心业务系统明确的方向是Java，去年年底，我们开始做一个基于Java的应用开发框架，DDFrame，用它去对接一些核心组件，包括SOA、作业调度、缓存、消息队列、数据库、配置中心等，现在已经发布了2.0版本。虽然受限于资源，进度比较缓慢，但我们一直在做这件事，未来也会慢慢完善这个框架，使其成为技术体系的核心。

【架构师并非必需品？】

我在上大学的时候学的就是计算机，但没学过系统架构方面的课，也没听说过架构师，之前做项目的时候也很少有专门的架构师角色。一般来讲，系统比较简单的话，并不一定要有架构师，当系统更为复杂，才需要有人在更高的视角上去关注整体性的东西，这也是系统规模不断发展的结果。

所以，我们可以认为，架构师并不是一个必需品，甚至不同公司架构师的职责都不太一样，并没有一个非常明确的定义，但整体来讲，架构师需要关注整个体系中方方面面的东西，还需要去解决一些关键性的技术难点，并需要有更为长远的考虑，这个是共识。

【架构师与工程师之别】

架构师与工程师之间的差别并不在于年纪，而是在于视角的不同以及各方面积累的差别。

首先是意识。作为架构师，不能仅仅关注怎么去实现一个功能，还得去琢磨为什么这么做、怎样才能做得更好、应该在什么场景采用什么样的技术方案等问题。另外还得去关注测试、部署、项目管理的方式等方面，甚至要去了解用户的需求、公司业务的需求。如果一直考虑这些事情，时间长了、经验多了，就会有比较好的整体概念或视图在脑中，综合素质会得到提升，明白功能只是最基本的，系统的可用性、稳定性、可扩展性更为重要。

其次是积累。IT技术一直在演进，全世界无数人的不断研究与实践成就了技术的提升与进化。作为技术人，需要关注当前最新的技术、架构、解决方案、技术理念等，理念可以用不同的技术来实现，也是一个不断进化的过程。而作为架构师，承担的是更重要的角色，他的决定会影响到更大的团队或体系，所以就需要有足够的相关知识和技能，以及足够广的视角，而这些都需要架构师平时不断的积累。我每年大概会看20多本书，

技术相关的大概一半，另一半主要是社科类、经济、历史、管理之类的，对提高架构思维很有帮助。

再次是责任感。之前提到，架构师承担的是更重要的角色，他的决定会影响到技术选型、系统架构、具体实现的方案甚至系统发展的方向，所以架构师需要有很强的责任感，要对技术团队负责，需要发挥自己的影响力，做很多沟通、协调、支持的工作。

最后是兴趣。在我看来，人是受限于他的性格、兴趣、天分这些因素的，会不自觉地去靠近他更喜欢、更擅长的方向。所以，到底是当工程师还是架构师，或者其他角色，也是要看兴趣的，有的人就是喜欢解决技术难题，就是喜欢具体的模块实现，不想牵扯太多精力去考虑其他的方面，那么也不见得非要当架构师，只是分工不同，业界也有技术专家、研究员这样的角色。

【架构与管理相通】

之前架构师大会上经常有人说，没有最好的架构，只有最合适架构。确实如此，作为架构师，很多时候，技术上的东西可能跟程序员差不多，但差别就在于能不能以更大更广的视角去看待问题，而非仅从自身角度出发。

视角变大之后，所要关注的东西就会变多，变量、变数也会更多，很难理想主义，很多时候都需要做出妥协或者说平衡，到最后就会发现，架构和管理在很多时候是相通的。

管理是在一个有限资源、确定时间点、明确目标的情况下，尽可能达成目标，过程中需要考虑轻重缓急，需要随时调整以适应现实变化，以完成目标为首要考量。

架构也是如此，需要考虑的是宏观上的方向性的问题，是各个部分之间的平衡关系，是如何配合才能达成最佳效果，而非仅仅是短期目标而达成，也不必纠结于细节的完美。

这些年，管理、架构都发展出了很多的理论，虽然行业、环境一直在变，但却也不能直接断定它们是不是合适，需要学习的是其中的思维方式，具体的问题具体分析。

【IT是条不归路】

有这样的感慨是因为IT行业发展实在是太快了，覆盖的领域也越来越广。前两天刚好面试了一个候选人，40多岁，传统IT领域的，能力很不错，在原来公司也做到了挺高的职位，但他不熟悉现在互联网主流的东西，面对的也是不同维度的需求，思路对不上，就很难符合我们的要求。每次见到这样的老大哥，心里都有很悲凉的感觉。

IT人的能力和价值是基于技术的，一旦跟不上技术进化的脚步，或者当初所选的领域成为夕阳领域，职业道路就会面临转折。这也是我之前换工作的原因之一，一直在原来的公司干下去的话，真的会失去竞争力，很有危机感。

尤其对我们这一代做IT的人来说，前面没有多少人走过这条路，没有借鉴之处，真的不知道十几、二十年后，我们的未来会是什么样？我们这些年的摸索，也是给后来者趟路，现在刚毕业的二十多岁的年轻人，就能看到未来的方向，知道自己十年之后大概会是什么样的，但我们真的一直都不知道。

不过，既然选择了这一行，就只能持续关注行业发展，不断提升自己，多学习、勤思考，努力走出一条路来。



邱剑：美团在云上踩过的那些坑

原文作者: EGO编辑 陈园园

我先简单介绍一下美团云。

其实美团上云是非常早，在2012年的时候，那时候美团规模还相对比较小，只有百台服务器规模的时候，就开始规划、开发我们的云平台了。



我们云平台整体框架是基于OpenStack做的，但是我们并没有完全用它，我们有我们自己的一些对于云平台的要求。所以当时我们就用了它的框架和少数的组件，一个是认证组件、一个是数字镜像管理组件、一个存储组件，其余部分都是我们美团自主开发的，基于美团自己的特点。

我们为什么迁移到云上，因为掌握了大部分的代码和平台，所以我们能够放心地去用它。

【业务如何玩转云平台】

那我们怎么去用这个云平台支撑美团的业务呢？

先看一下美团云平台的整体架构。(右图)



整体上，我们有一个公共的云平台的基础，在这个基础上有一个私有云，私有云是真正跑业务的，也是最大的；然后有一个办公云，这个部署在我们公司，就是给IT使用，IT可以自己去申请、去扩容的；然后还有一个美团云，这个不仅跑自己的业务，还有对外的产品形态。

美团所有的业务，包括电影、团购、外卖等全部都运营在私有云上，开发团队则在办公云上。那我们具体是怎么用这个云平台去跑这些业务呢？

【服务部署规范化】

使用云，最重要的就是可以加速服务的交付，可以加速自动化部署。

这点主要体现在，因为美团的特点就是小业务多，不像搜索这样大的框架，都是在比较均一的、统一的服务的框架在里面，但是像美团的业务比较琐碎，所以使用虚拟化的好处，就是说我们不需要把很多业务塞在一台机器上，而是形成一个业务，一台主机一个服务。

而且我们对于主机的命名做了规范，就是必须按照机房-业务组-产品-服务<序号>的方式明码，看到这个就知道是哪个组的，他们在做什么业务，出了问题找谁。

同时主机的名称就是访问这台机器的域名，所以不用管这台机器的机房，只要知道这个名字就可以找了。

【自动化一切】

回到云平台，我们就能够把整个事情都统一化、规范化，然后以前没有上云之前比较复杂的事情就变得更简单了。

比如说，上云之前，要上一个服务，给它选一个IP地址，给它配个云，主机还得装系统，装完系统还得做一些基础的配置，还有一些监控、日志、报警、权限管理等等，非常复杂。

我们有了云平台，我们在云平台基础之上就能把这些业务都打通起来，云可以管理这些网络，还会自动分配，不会担心访问不到，这样就把网络的问题解决了。

这样的话主机分配到IP地址之后，就自动把IP地址和主机名称关联起来，并且在DNS上注册上，这样就不用关心这台机器是哪个IP、在什么机房，我直接运营网络就可以了。

在主机的模板里我们把必要的基础软件安装上去，并且在机器启动后就能完成自动化运行，像这台机器是为数据准备的，它就会下发跟数据相关的配置，然后就会把相应的软件配好，环境运行好。

同时，说到业务相关的话，还有一些基础的配置，像监控、日志搜集、报警等等，这些都是自动加上的。

我们每台服务器都有登录的权限，像我是云计算组，我可能就不能访问数据组的主机，这样的话就通过IPA

来控制。这个的话也是主机发布之后，它就根据它的业务组，自动加载上去了。

这样的话，我们的业务线提交一个虚拟机的申请，它就会把这些全部自动化做好，这样的话他只需要关心说怎么保证业务正常运行就行了。

【代码沉淀最佳实践】

通过这样云上自动化的流程，我们就能非常快速地向业务提交我们的资源，但是在运行过程中会也遇到很多坑，同时也会有很多经验。

1. 性能优化

比如说，像虚拟机的性能，我们发现是可以通过一些手段提升的。比如，使用HugePage，通过测试使用一段时间发现没问题后，我们会把相应的配置放到我们代码来，和这些云平台的配置就一起生效了。还有CPU节点的绑定，这个也能降低虚拟机的满载的概率。

我们发现的这些的实践能够固化到代码里的就固化到代码里去。

2. 避免踩坑

同时还能避免重复踩坑，像我们以前遇到使用cgroup的时候经常出现BUG，就是当我们绑定cgroup是以进程绑定，这样就会有一个冲突，导致很多坑。发现这个问题的话，我们就在云代码平台里设置限制，代码加入之后就不会再出现这样的问题了。

我们就是通过一些实践，把它代码固化，然后以后接触这个的可能都不太知道这个事情，但是它已经沉淀到代码里面，也不会再发生这样的事情。

【问题与解决方案】

我们业务线使用云平台大概就是这样的一个流程，但是这个说的相对比较轻松容易，就是一切都能自动化，但是其实还有很多很多的坑在里面的。下面介绍一下，在我们使用的过程中遇到了哪些问题，和我们是怎么去解决的。

【性能】

很多人听到我们说，美团把自己的业务跑在云上，一个最大的问题，就是性能能不能满足虚拟机运行，所以最大的问题就是性能问题。

经过我们实践，虚拟机、虚拟化平台是能够满足一个生态环境的性能需求的。但是它并不是说所有情况都是

这样，而是分情况。

一般就是CPU密集型，这样的业务比较适合跑在云上，所以我们把一些典型的密集型业务，像JAVA前端等服务就跑在虚拟机里。像最近有一个网络协议站，就是一台物理机性能很多，但是无法完全用在物理机上，都用到了内核上，如果把物理机拆分以后，可以更好充分利用上物理机的资源。

我们通过虚拟机能够确保它的性能满足业务需求，但是不是通过代码把这个解决掉，更多的是依靠我们的SRE团队，他们对于虚拟机的性能监控、性能隔离，通过这些技术手段来实现的。

性能隔离，像权重隔离，有八核和四核，我们分配给八核高于四核两倍的权重，就是给它在竞争的时候能够跟它的核数相匹配的运算能力。

还有，美团在业务高峰的时候会出现比较严重的CPU-STEAL，就是说有些时间会被虚拟机占用了，后面我们发现这种情况后，就采用阈值隔离的方式，就是说不只给它设置简单的权重，也会给它设置一个最高的值，这样的话即使在高峰的时候，大家争抢也不会挤到别人的份额。

权重隔离方式和所有物理机、虚拟机运行的时候，能够充分利用物理机，因为这是根据权重来分配的，但是这样的话竞争的时候没办法保证性能，对于这样的话我们就设置一个阈值，同时，网络也会存在竞争，这时我们会做网卡限速。

发现这些问题的方式，是通过我们一个比较好的性能监控的实践，就是我们无论物理机还是虚拟机都作为zabbix host，根据虚实关系建立hostgroup，便于做同宿主机上的VM横向比较以及聚合计算。

【调度】

为了保证一个最好的性能，除了做好监控、做好隔离，最重要的一个就是在虚拟机刚刚分配的时候就要给它找到一个合适的宿主机分配上去。

这对我们调度存在很高的挑战，因为在刚开始阶段，调度比较复杂，像业务线提一个需求过来，他就会根据这个要求选择一个合适的物理机把它调度过去。

随着业务的发展，规模比较大之后，人工的方式就开始转变成自动的方式，自动的方式就需要在调度算法里把整个因素考虑进去，然后一定程度地满足业务的调度的需求。

比如说我们申请VM的时候，要求业务组声明是CPU密集型还是IO密集型，这样的话我们就能够按照不同业务的特点进行调度，充分利用物理机资源。

还有我们会把一些虚拟机做成不同池子，就是这些数据是给什么业务做的。这些都是调度要考虑的问题，这就导致调度的工作非常复杂，它要考虑很多因素才能找到合适的虚拟机。

为了解决这些问题，让它能自动化地根据自己的业务特点把数据调到合适的虚拟机上，我们就想了一个方法，叫基于标签的策略调度，这可以帮助业务线根据特点调度机器。大致意思就是我们会给每个虚拟机设置一系列标签，比如这台机器负载率比较高，然后我们会给虚拟机调度时指定相应标签的策略组合。

【迁移】

刚才说的调度，随着虚拟机业务的增长或下降，对资源需求的不一样，这个时候会存在，需要把很多数据调到一台虚拟机上，这种情况下需要做虚拟机的迁移。

刚开始做冷迁移，就是把机器停了，然后复制，这样效率比较低的。

然后我们SRE团队说就不迁移了，因为我们发布的数据很快，我们没必要迁移，这两台机器合在一起，我把这台机器重新发布一遍，把这台销毁，这个其实降低了业务下线的时间，虽然麻烦一点，但是也能解决问题。

当然这种比较适合业务发布比较规范，就是不需要太多人工操作的情况，它会很快发布一台新的，把这台销毁实现迁移。

后来，我们做了一个基于本地存储的虚拟机热迁移，虽然是本地存储，但是我们也希望热迁移，就是瞬间能迁到别的机器上。

这个的大概方式，就是我们能够通过一个远程的mount到本地，基于这个子盘做一个很薄的磁盘，同时设置一个VM，然后把这个VM迁到另一个VM，我们把这个磁盘挂过去，然后把数据迁移过去。

【物理机管理】

最后说一个，刚开始我们云平台管理是虚拟机，随着业务的发展我们虚拟机数量迅猛增长，物理机也迅猛增长，这个时候我们希望把物理机也管起来，所以我们把物理机管理也放在云平台上。

我们云平台实现了把物理机从机房上下就直接放到云平台，做到，物理机和虚拟机统一模板、统一API接口、统一流程、资产管理。

就是说我们虚拟机的模板部署到物理机上运行，通过这样的方式，我们以前在虚拟机上执行的自动化发布等等，在物理机上完全适用。有了这个系统的话，我们资产管理完全依赖云平台，就是信息都是在云平台上采集。

Uber高级工程师

赵磊：从PHP到Node.js，漫谈Uber技术架构的变迁

【2015年7月19日，EGO走进深圳，并邀请Uber高级工程师赵磊与大家分享Uber的系统架构变迁，以及是如何从PHP转到Node.js。】

【Uber的service】

基本上，Uber service分成两大部分，一部分是Real-time Service，一部分是Offline System。

【Real-time Service】

从乘客的角度，打开Uber APP，就可以看到周围的driver。这时候乘客点一个request，我们的client就会发送一个request到Real-time Service里，然后会发送一个offer或dispatch到driver方，等其中一个driver接受offer后，就开始一个trip。直到乘客下车，这个trip结束，整个trip都是在Real-time的系统里做的。
为什么叫Real-time，因为这个系统是绝对不能down掉的，如果down掉了，可能你在车上都不知道去哪儿，而且它的latency要非常低。等一个trip结束后，我们会把它放到一个Message Queue或者是Task Queue里。

【Offline System】

Real-time Service后面就是Offline System，它会去queue中获取task，然后做一些任务，如build a credit card等，乘客和driver之间可以互相read。这些都是在Offline System里做的，因为它们可以delay一段时间，work states等那几分钟可能并没有什么关系。

现在在Uber，Real-time system基本上都是用Node.js写的，而Offline System基本上都是用Python写的。

【Real-time system】

我今天主要介绍Real-time system，因为这是最复杂的，要保证high availability，而且我们的Real-time是一个city shared service，也就是说全球有几百个城市，每个城市会run在几个worker上，它前面有一个我们自己写的load balancer，任何一个request进来之后，就能看到你现在是在哪个城市，之后就能把这个request forward到相应的worker，然后那个worker就process这个request。Shared service有很多challenge，我们最近的很多努力主要是在service routing和服务发现上。

【service routing】

先讲service routing是怎么回事。你的系统现在其实变成一个micro service的架构, 做一个flow可能会需要很多的service来协作完成一个request。这里就会有service A发送request给service B的情况, 那A怎么知道B在哪呢? 这就是service routing, 我要知道B的host name或IP或port, 然后把request通过一个RPC发送给B。这里有两个选择, 一个是client routing, 一个是server routing。

<client routing>

先看一下client routing, 假设我要把一个request发送给B的worker, 那我首先有个sharding function, 就是可能你的keyspace user的费用太大, manage这么大的keyspace也会非常复杂, 所以就用sharding的方式把这么大的keyspace变成很小的shard, 比如说1个million。

然后就有一个global的shard map, 知道哪个shard应该在哪个worker上, 从client的角度考虑, 就可以通过sharding的方式找到这个user在哪个shard, 然后再在那个shard map里找到在哪个server, 就可以把这个request发送过去了。

这种方法比较常见, 比如说MemCache现在基本上就是一个client routing。如果你有个MemCache service, 假设有10个node, 想写一个key, 那怎么知道去哪个node呢? Run一个hash function, 用consistent hashing就能找到你要的node了。

但这里还有一个connection reuse的问题, 大家都知道TCP有一个three-way handshake, 这里就有一个round-trip time, 虽然one-trip time非常低, 但是如果你有很多request的话, 这个cost也非常高。有一个简单的解决方法, 做connection polling, 我可以一直maintain一个persist TCP connection, 当我发送request的时候, 就不需要做handshake, 直接用现有的TCP connection发送就可以了, 非常简单。

在client routing的架构里面, 要怎么做connection reuse呢? 比如我刚刚举的例子A-B, 如果你有100个A、1000个B, A可能会连到任何一个B, 所以每一个A的node都要1000个connection, 这时候的connection数量就是 $N \times M$, 也就是 100×1000 , 而且这只是A-B, 实际上A会有其他各种各样的downstream service, 比如说AC、AD, 所以你的connection数就是 $N \times M \times$ 你的整个service数, 会非常多。

所以, 我们现在正慢慢地把Uber的整个架构从client routing往server routing这个方向发展。

<server routing>

Server routing中, 虽然你的service是shard, 但是从client的角度来看它是non-shard, 这样就能把很多complexity都放进server里面, 从client来看是看不出来的。

那它是怎么做sharding的呢? Server routing中, server之间是互相认识的, 它们知道应该怎样去shard这个user, client只要把request发送给任何一个server, 它就会把这个request proxy给对应的node, 就是你的shard应该在的那个node。

这样的好处在于, server routing的架构里面, client是没有sharding function的, 否则, server和client都有sharding function的话, 你upgrade的时候就很难去改动。特别是对于connection reuse, 这种方法非常容易, 因为原本一个client要连到1000个server, 现在连到任何一个server就能做所有的事, 所以这种

情况下它一般会选一个很小的selection, 有时是3个, 那我的client只要保持3个connection到3个server node就ok了, 每一个request都能发送给它们中的任何一个。这时候client到server之间的connection基本上就非常少了, 比如说刚刚那个例子里有1000个server, 这样我只要保持3000个connection就ok了, 刚才是100K, 现在是3K, 会小很多。

另外, 如果你的系统里有多种语言, 比如有Java、有Python、有Go、有JavaScript等, 那你的client当然是越简单越好。同样的server, 如果你的client比较复杂, 像刚才说的要做sharding function上的routing的话, 你可能会需要把同样的逻辑在每个语言上都实现一遍。如果你是server sharding的话, client非常简单, 你很容易就可以把任何一个语言的client都纳入这个server, 这一点在Uber比较重要。现在, Uber有Node.js还有Python, 很多新的项目也在开始尝试用Go。

【service discovery】

再来讲service discovery是怎么回事? 我从A连到B, 怎么知道B的IP地址和port呢?

Uber用了server routing之后, 自己做了一个service broker或者说service discovery, 它有一个单独的服务, client本来是要talk to A、talk to B, 这时候它只要把request发送给这个routing service, 然后routing service会把它forward给B, 这样client只要认识这个routing service就ok了, 不需要它直接认识B。

首先, service在start时, 比如说你的routing service run了10个node在上面, 还有10个IP地址, 这10个基本上是static, 那你的server可以是dynamic, server在start的时候, 整个routing service说自己是一个service B, 我listen to IP和port, 那这个routing service就会记得B在这个point上, client也同样认识这个routing service, 这时候client不需要知道B在哪儿, B也不需要知道A在哪儿。

【从PHP到Node.js】

很早以前, Uber的dispatch是用PHP写的, 后来我们换成了Node.js, 当时做这个决定的原因主要有几点。

第一是因为downstream service, 一般来说, 做任何事时, dispatch system会把你的request发送给下面的downstream service, 拿到response, 再发送给下一个downstream service, 要做很多不同的RPC才能完成一个request。

在PHP里面, 基本上所有的IO都是sync, 任何一个downstream service, 都要先发送一个request, 需要等一段时间拿到response后才能进行下一步, 中间不能做任何事情, 这个latency可能很长, 等你的downstream service越来越多的时候, 你的整个的latency可能会越来越长。但是Node.js有一个好处, 它所有的IO都是一个sync, 你可以把所有的request都发送出去, 等它们拿到所有response的时候, 再一起process那些response, 这样, 你的overall latency会低很多。

第二是因为in memory state, PHP是一个完全面向web的语言, 当有request时, 它会开始一个新的thread,

这个thread会run你的handler, 你的handler会process这个request。等你的response结束以后, 这个thread就会结束, 而且你的handler不能run很长时间, 在里面你很难做到能够有cross request state, 一般大家都会把这个state放到一个external里面, 比如说MemCache。

但在dispatch system里, 我们还是想要有一些in memory Cache, 比如现在in memory里, 我们会知道这个城市里有哪些driver、有哪些rider。当我做dispatch时, 我知道这个rider周围的driver都在哪, 一旦这个driver发出offer以后, driver state就会变, 这个state我们就放在memory里面, 而用PHP的话, 这个基本上是没办法实现的。

第三是因为user sharding, 如我之前所说, dispatch system是city sharding, 如果用PHP的话, 一般PHP appliment都是一个比较简单的load balancer, 它把这种request发送给任何一个PHP node, 这里面是没有sharding的。我们如果做node service的时候, 可以再做一个简单的client side, 因为有a memory state, user必须在你的node上, 所以你之前需要有一个比较聪明的load balancer, 能够把你的request发到那个user应该的那个node上。

极光推送CTO黄鑫：技术人要建立自己的知识图谱

原文作者: EGO编辑 陈园园



【了解技术变革与技术组合】

对于架构,在我看来一定要特别深入地去了解技术本身以及技术所经历过的变革。比如说Redis,后来推出了Redis-cluster,大家都知道Redis-cluster能解决哪些问题,但当我再问一个深一点的问题,从Redis到Redis-cluster为什么经历那么久时间,推出这个新版本、做出这个新方案的时候,它有参照哪些相似的技术、有哪些优点和缺点、可能会有什么样问题。你会发现,这个问题几乎可以难倒95%以上的人。也就是说其实他们并没有深入地去理解一个技术的变革,这在我看来是一个很大的问题。

另外,当了解了这些技术之后,你还需要能对这些技术做出灵活的组合,因为没有任何一种技术能解决所有的问题,尤其在我们现在的时代,你的业务线非常多、你的数据非常大……所以你一定需要有这种组合技术的能力,能把若干个技术组合到一起去解决问题,这也是架构的时候非常重要的能力,对创业公司尤其如此。

【CTO的挑战: 技术、产品与商业】

我现在负责整个技术和产品研发团队的管理。大公司的CTO可能更像一个management,对具体的技术细节,他可能不需要特别了解。但对我们这样不到200人的创业团队来说,CTO需要承担更多的技术职责,除了传统上的管理以及把握公司整体技术前进的方向之外,还必须要完成某些技术的攻坚。比如我会需要跟技术团队一起讨论跨IDC要怎么做、某一个算法要如何设计优化、甚至会讨论到某一个页面怎样的体验是最好的等等,这些很细节的层面,也都需要我作为一个CTO做一些把控。

当我达到现在这样程度的时候,也会对自己提出更高的挑战,因为创业公司的CTO其实在一定程度上决定了这个公司的技术瓶颈和技术高度,是你在引领整个公司的发展,所以你就不能拖后腿。

别人提出一个技术方案的时候,你要能够及时响应,并且能够告诉他这个方案可能会有哪些问题。而当别人没有提出的时候,你需要做的是第一时间提出新的技术方案,并且安排人去做研究和尝试。这些都是对CTO更大的挑战。

作为CTO, 50%的时间做前沿技术学习, 30%的时间做对外交流, 20%的时间做竞品研究, 20%的时间做日常管理, 这是我认为比较合理的时间安排, 是的, 做CTO需要花费120%的时间在工作上。

另外, 在创业公司, 尤其是在云服务公司, 作为一个CTO, 你还需要去了解公司整个的商业模式。因为你比任何人都了解技术, 所以你一定要知道公司以后到底要朝着哪个方向发展, 比如说到底是要专注地去做推送, 还是要去做数据, 还是要做开发者平台, 还有如何把握彼此之间的平衡等, 这些都需要你去关注。你需要把技术和商业更好的结合起来, 这一点可能是我最近一年最大的进步。

也许有的人的目标就是想做一个纯粹的技术专家, 但对大部分人来说, 你的技术其实是为产品、为商业服务的, 这是我觉得很多人需要去做更多的思考、更多地走出去看业界的一点。

【建立属于自己的技术知识体系】

我的学习模式是, 刚开始, 我会把一门、两门技术学得非常深、非常透, 并且在脑海里形成鲜明的两个知识点。然后我会不停的去接触新技术, 但是每接触一个我都会跟之前的这两个技术点做比较、建立起关联, 相当于形成三个点, 而且这些点之间会不停地连成很多的边, 显示着这些技术彼此之间的优势与劣势、相似与区别等, 到最后就会在脑子里形成一张图。比如说Python支持这样的一个语法, 我会想那其他的语言有没有也支持这样的语法的, 其他的语言不支持的时候, 它到底是怎么实现这个问题的。这样你就会对比起来了, 而且你会发现, 你加的点越多, 你对之前的东西理解得越深。

所以说技术在很大程度上是相通的, 这也是我特别喜欢计算机的一个原因, 就是需要你不停的去学习, 然后就会觉得你学的东西原来越来越有趣。

这是我觉得非常重要的一个能力, 尤其现在全栈工程师非常火, 但是我面试的时候却发现, 很多人他的知识面很广, 但无论是他们技术的深入度还是技术点之间建立起的联系都是不够的。比如他说我会Java、Python还会C++, 但当我问它们三个之间有什么区别、共性、面对具体项目怎么选择的时候, 可能他就没法很好的回答, 只能说出一些很浅显的东西。

在我看来, 一个人不努力反而没那么可怕, 因为当他遇到感兴趣的事情时, 我相信也许他就会自己努力起来了。最可怕的是一个人其实很努力, 但是他从来不思考, 我身边就有这样的人, 他读了很多的书, 可能有上百本技术相关的书, 但是你会发现, 他看完就看完了, 不去想这些书背后的东西, 也不去深入思考这些书能产生的价值, 这是我觉得非常致命的一点。

(原文节选, 完整版文章请查阅EGO微信公众号: egonetworks)

你比Facebook工程差什么： 自我管理能力缺失？

原文作者：EGO编辑 陈园园

【在4月22日EGO的活动上，阿里巴巴技术保障部研究员赵海平做了题为“Facebook研发团队的高效秘诀”的分享。分享结束后，大家就团队管理、工程师文化等问题探讨至凌晨。小编就选取一些精彩讨论，与大家分享。】

一、良好的自我管理能力是出色工程师的必备能力

李云：听了海平的分享，相信大家对于工程师文化、Hacker文化、去中心的管理文化都很向往，但现实是，在中国，这样的文化很难生根发芽。原因在于，中美IT行业的专业化水平很不一样，请注意，专业化水平不单是指你的技术能力如何，更包含工程师的自我管理能力。

出色的工程师大多具有良好的自我管理能力，这多少能解释为什么早期的Facebook好像不需要管理。这些出色的人会因为良好的自我管理能力而在工作中体现出良好的方向感。而国内工程师自我管理能力的缺失，使他们很难通过团队力量去沉淀，从而抬高整个行业水准，这不仅解释了我们的专业化水准上不去的原因，还能很好地解释为什么中国的软件工程师是“青春饭”。

李军：非常赞同李云的观点，而且我自己对国内工程师professionalism普遍偏低深有感触。我觉得这是中美IT行业核心竞争力差距之一。不过我觉得，这只是因为国内的IT行业本身就年轻，而且尚处于丛林拼刺刀的阶段，每个公司都在生死线挣扎，根本无暇顾及员工能力全面提升和可持续发展。假以时日，随着行业的发展和管理人员水准的提高，这个现象会很快的好转。

我自己也在日常工作中致力于培养工程师的职业化精神。而且，从我团队的经验来看，大家接受度还是蛮高的，转变起来也很快。

刘诚忠：的确，优秀的工程师往往拥有良好的自我管理能力，而这些人遇到一起自然就会形成这种文化，打造出工程师文化来。换句话说，这些东西都是自底向上形成的，而不是刻意塑造的。

萧田国：自我管理应该是靠自己，类似于工程师的自我修养。我们的竞争力不应该仅仅是技术，技术本身容易被超越过时，竞争力应该在于长期工作历练形成的积累。但国内普遍过于看重技术，都想做“全国最好的石匠活”。

自我管理的难处在于，需要锻炼右脑（情商和直觉沟通），一般技术人厉害的是左脑（智商和逻辑推理）。大

家都喜欢做自己擅长的事情，于是左右脑越来越失衡，自己的危机感也就越发的重。

李云：没错，中国的工程师有一个天然的毛病，重技术轻管理（自我管理包含其中）。技术的进步其实是相对容易的，一个工程师真的沉下心来在一个领域钻研2、3年就很可能成为这个领域的专家了，但管理能力就没有那么容易成长，在大环境不具备良好的管理能力的情况下，这种能力的成长很慢。

我是在美国Motorola工作期间深有体会，那边的工程师个个都具备架构师的能力，这与他们的管理文化沉淀到了一定的水准有很大的关系。

二、团队管理的着力点在于营造能让工程师提升自我管理能力的环境

艾英：百姓网受Facebook影响非常深，王淮曾是我们的顾问，我们也崇尚move fast and break things，也是所有人share code，也自己折腾各种各样的工具来提升效率。但是，Facebook是40多人的时候还是那样hack，到300来号人的时候才需要管理，我们现在60来号人，已经有不少管理工作了，为什么团队“衰老”的速度会比Facebook快这么多？

李云：我认为管理最根本的着力点在于改善工程师的自我管理能力，项目实施、团队协作是达到这一目标的手段。但尴尬的问题在于，真正能帮助团队改善自我管理的管理者很少，因为在目前的行业背景下，大多管理者是在自己不具备良好自我管理能力的情况下成长的。只有个体的自我管理能力上去了，才有可能出现稳定的工程师文化，才有可能让团队运作高效。

管理检验的是“行”而不是“知”，有时我们写出来很在理的文章其实收效却并不好，原因在于我们的日常工作中需要根据自己的理解和理念去切实引导工程师发展，帮助他们改善。一个人影响另一个人，甚至一个人影响一个团队是很具挑战的事。

袁店明：即使60多个人，那么所做的事情也不少了，如何让60多个人中的每个人都清楚自己的职责，每个更小的交付团队之间的依赖和耦合是不是足够少，另外，即使不需要管理，那么激励和成长是否有效？同时，由于每个产品都不相同，因此，需要根据具体的情况进行分析，但总的来说，就是遵循面向对象的“高内聚、低耦合”。

LM：人多的时候可以划分成相互独立、依赖非常低的小团队，除了财务不独立，其他从开发到运营全部独立，每个小团队都像一家公司一样。对此，需要管理者非常深入地理解业务，根据业务本身的耦合度来划分团队。

李云: 之前海平说, Facebook成功也不意味着什么都好、什么都能完善地解决, 我认同这一观点, 每种方法其实都是有其适用范围的, 不是简单搬过来就了事, 得考虑产品的特点、团队的水准与特质等。在这一块我们仍需要探索, 而且是那种“量身定制式”的探索。另外, 探索的过程中我们一定得记住80/20原则, 即有80%的内容是可以用有效的科学方法去规避和减缓的, 而另外20%则是无法实现通过预测去解决的。80/20的比例不变, 但其总量会随着时间推移而增大, 承认那20%其实就是承认软件开发的复杂性本质, 这一点非常重要。

艾芙: 激励、成长、清晰的职责、自我管理, 都是很团队管理重要的关键词, 而“管理最根本的着力点在于改善工程师的自我管理能力”+“在大环境不具备良好的管理能力的情况下, 这种能力的成长很慢”=管理的着力点在于营造能让工程师提升自我管理能力的环境。

(本文由编辑根据明略数据技术研发经理刘诚忠、阿里巴巴集团UC浏览器电脑版技术团队主管兼软件架构师李云、北京触控科技有限公司运维总监萧田国、百姓网技术总监艾芙、敏捷与精益创业咨询师袁店明、百度高级技术经理李军等人的讨论整理成文。)

杨卫华谈技术人成长： 要善用社区提升自身影响力

原文作者: EGO编辑 陈园园



【影响力是优秀技术人稀缺的软能力】

身边每年有很多优秀的年轻人加入，包括很多刚迈出校园的同学，如何成为优秀的技术人，是这些年轻人经常思考及询问的话题。

对于刚进入职场的年轻人，我的建议是先在一个领域做得卓越，在没有合适的舞台之前，先做好身边的每件事，并且在部分领域做得专业及突出。通过这种方法形成优秀的做事习惯。当习惯将每一件事情做好之后，就会更多的得到公司及同事的信任，为自己赢得更多施展及成长的机会。在此基础上，寻找适合自己施展的舞台，让自己的特长能够很好的发挥。

我比较看重技术人的两个品质，一方面是通过技术达成的成就，成就是指通过技术完成产品，并通过产品影响并触达用户，最终能影响的更大范围的用户，虽然成功或成就不同人心中有不同的定义，我个人更欣赏这种通过技术服务更多用户的成就。

另外一方面，技术人还应当具备一定影响力，影响力是指通过自己的作品、言行、文字对社区、业界、同行形成推动及改变。业界需要通过正面的榜样、不同的思考，来推动业界更好的前行。影响力是优秀技术人稀缺的软能力。

【技术人要善用社区的力量展示自己】

优秀的技术人还要善于利用社区的力量，让自己的特长及成绩能够得到适当的展示。要多写作、多分享。写作首先可以整理思想，习惯写作的朋友可能深有感触，一旦提笔来写的时候，会发现很多自以为想得很清楚的问题，观点的完整性及逻辑皆有问题，需要去进一步查阅资料、统计数据以及与同行交流才能得到完整的、具有说服力的文字。写作通常是一个事例、数据与逻辑综合的过程，通过这一系列的写作过程，可以重新更深刻的理解及认识目标。

同时，通过写作分享自己的观点及感悟，可以更多的结识及影响同行。大部分技术人的工作环境中，能接触到的人有限，同事之间交流也是以工作任务为主，因此对于更广泛的领域，身边并没有合适的交流对象。但通过微博、公众号等方式将自己观点分享的方式，则有机会结识业内有相同兴趣的人，了解他们的看法及观点，以便延展思路，开拓视野。自己觉得很普通的一件事情，分享到微博之后会有一大群人过来

评论及讨论，日常工作的一些感悟整理出来放到公众号，可能会收到很多陌生人的赞赏，这些赞赏更多的是对分享的共鸣及鼓励。

在专业角度，通过文字和分享还可以更好的展示自己，让身边及业界更好的发现自己的特长，虽然优秀的人有多元化的自我展示方式，但在网络时代，通过社交渠道是技术人展示自己的优点和特长最有效的方法之一。

【高端技术人在EGO更好地学习和成长】

我们身边有大量有理想的技术人，他们通过工作和学习不断成长，当技术人发展到一定阶段，比如成为公司或者一个部门技术负责人之后，就会碰到一些发展上的瓶颈，靠原先的方法并不能非常好的解决，需要解决的问题在公司或部门范围并没有现成的成熟解决方法；对于行业的趋势，我们也缺少一些深度的同行参考意见。EGO通过吸引较多行业内资深会员的加入，可以帮助高端的技术人才弥补在自身环境的一些不足。通过业内大拿的小范围分享，以及内部小组讨论，帮助大家拓宽视野，更深度的理解行业特点，帮助技术人更好的获得学习及成长。

同时，高端的技术人还会经常面临一些重大的选择问题，比如平台的选择；技术方向的选择以及产业布局的选择等。EGO组织里面具有丰富行业背景的朋友，没有业务利益或者竞争关系，具有相似的视野，能够中立的帮助会员获得抉择的参考意见。

资深的技术人要继续前行，要么成为顶尖人物，要么跟随顶尖人物，在前行的路上，如果有成熟的跟随榜样，在某些程度会让我们避开很多很多弯路。在我们公司内较难找到好的榜样；业界领袖又离我们太远，EGO里面的行业精英可以更好的作为我们工作及处世的榜样，更快的获得成长。

作为EGO首届北京分会的会长，我希望自己能在学习目标、交流机制、人员准入门槛上发挥应用的作用，给大家做好服务，让每一个来到EGO北京大家庭的成员能够从身边每个人身上看到闪光点，通过EGO获得更多的价值。

（原文节选，完整版文章请查阅EGO微信公众号：egonetworks）

果糖移动CTO

刘建国：怎样带出一个 2.5倍速的研发团队



EGO：您在带领技术团队和管理方面有怎样的经历和心得？

刘建国：带人要用心，要学会授权。

在带人上，用心是最重要的，首先得看你是在利用一个人还是在培养一个人。不少公司的管理者把员工当棋子用，并不用心去培养和发展。在我看来，团队是做事情的基础，对团队、对人的培养肯定要花心思。每个团队、每个人的优势都不一样，所以要根据他们的不同的特征来做不同的训练，培养出富有不同风格的带头人。

另外要学会授权，不能所有的事情都自己做，尤其是自己擅长的事情，更要去带人做，强调自己的个人产出不是一个好的管理思路，不能做“超级rd”，一定要给你认为值得培养的人机会，自己控制好风险，别出大的乱子即可。

EGO：在创业公司当CTO和在百度有什么不同之处吗？

刘建国：差别挺大的。

在百度我主要是带技术团队，大公司对目标的规划都比较清晰，所以每个团队的职责也是非常清晰的，我所

要做的就是把涉及的战略目标做好拆解执行，并不断想办法提升产出效率、团队战斗力。虽然当时团队比较大，但是重视人才培养和梯队建设的话，整体的可控性是比较高的。

而我现在创业团队中，虽然规模小了，但要求和挑战反而变高了。由于招聘难，搭一个高素质团队就很有挑战；由于负责的是一个公司所有的研发资源，就得考虑公司所有的工作目标，并且要有良好的进度保证。所以你会发现，业务的快速调整变化和工程师招聘与磨合的长周期，这两个挑战同时出现在创业公司，这对CTO的判断力和解决问题的能力要求是很高的。

另外，在大公司，节奏、流程、做事方式都有较强的容错性，而在创业公司就得平衡好一个度，既要保证质量、稳定性，又得快速推进；既得提前储备人才，又得考虑财务预算和管理成本；既得做好技术的调研和提前布局，还得花精力兼顾团队的稳定；总之创业公司对“平衡”的艺术要求更高。

作为整个技术团队的负责人，需要视角更高一些，更考验综合素质，需要磨练和修炼的地方也就更多一些，所以近一年我感觉自己成长了好多：）

EGO：对于CTO的定位，您是怎么看的？

刘建国：CTO是一个管理职位，而且在我看来是一个三位一体的立体职位，深度上要对技术有良好的判断力，广度上能够涵盖研发目标达成的方方面面，时间线上还得能带出一个不断变强的团队。

作为CTO，技术方面的要有判断力、觉察力和领悟力，能觉察到我现在做的这件事情需要用到哪些方面的技术，将来业务发展可能会用到哪些方面的技术，更加深钻的东西可以交给高工去协助解决，但自己肯定要有一个大概的轮廓和广阔的视野，这是技术方面基本的素养和要求；只懂技术还不够，较好地达成研发目标，包含了对业务的理解和公司战略的拆解，还包括项目管理、多角色协作、流程建设，以及质量保证、效果评估、线上问题跟进机制等等一整套东西；以上还只是保证眼下事情能搞定，作为一个创业公司，一项很重要的积累就是团队的培养，放在时间线上就是要保证产出是长期稳定的，这就涵盖团队培养与发展的全方位的工作，诸如招聘、带新人、员工发展、梯队建设、氛围建设，还有心态、软素质的培养blabla。

综合来说，CTO所需具备的素质中，视角高度和软素质会更重要一些。

EGO：从业这么多年，遇到的哪些困难是您印象比较深刻的？

刘建国：那有很多，一直都有，首先吃过一个亏是说，千万不要认定自己一定不会干什么，别自我设限。我在入职百度的时候，从来都没想过要创业，而且我认定自己这一辈子都不会创业，结果去年我出来创业了，那就导致我在创业这件事情上有意识的积累不够，无论是业务上的还是人脉上的，只能从零开始，花费了很多的精力。这就是一个教训，环境是会变的，人也是会变的。

还有一个教训是这样的，来到创业公司之后发现，最大的挑战还是人才的招聘，但是在招人的时候无论怎样都不要纯靠刷脸把关系比较好的朋友、或不好拒绝你的人拉进来。因为这样做之后，就会发现自己有很大的包袱，压力很大，有时还会觉得对不起人家。也不是说完全不能刷脸，但还是要靠大家有共同的认知，比方说契合的兴趣点、对未来的判断、对风险的评估等，在此基础上，如果彼此之间之前认识的话就是非常良性 的状态了。

EGO：对技术人成长，您有何建议？

刘建国：对于工程师的发展路径，大家都说“T”型人才，那么是先广还是先深呢？即便志在全栈工程师，我的建议也是从深再广，先把一个领域做深做透，然后会发现很多技术是触类旁通的，再把它向广博扩展会很容易，这是一个常规的做法。如果上来就奔着全栈去，每样蜻蜓点水，没有精深的领域，从技术发展角度来说是非常受限的。

先做精深再做广博，技术是这样的，其实万事万物都是深度积累后往外扩散的，比如我们写代码甚至也能锻炼我们怎么跟人沟通，你写代码要不要考虑别人怎么接手，一旦考虑了，你就具备换位思考能力，表达能力也会得到锻炼，就看有没有这方面的意识了。

另外，一般来说技术是为产品服务的，所以技术人员要有意识地多和产品沟通，提升自己的产品意识以提升工作价值；最后，技术人员成长到最后，无论是带技术方向还是带技术团队，都是需要很好的规划、合作、沟通、目标导向等等偏软的素质的，需要有意识加强，不能说，我走技术路线，我就是要个性……，发展肯定会受限。

（原文节选，完整版文章请查阅EGO微信公众号：egonetworks）

酒仙网副总裁助理 宗刚：那些技术创业踩过的坑

原文作者：EGO编辑 刘硕

【本文根据宗刚在今年10月QCon上海站的演讲《那些技术创业踩过的坑》整理而成。】

如何正确选择业务方向：

如果你真要作为一个技术负责人创业的话，首先要想明白跟谁合作、做什么样的业务。

1. 找准用户的痛点

项目成功与否，其判断的标准之一就是看有没有找准用户的痛点，如果项目根本没有抓住用户的痛点，你就没有机会获得成功。

我第二次创业做的是社区O2O，其实用户的痛点很弱，说白了就是我现在不想去菜市场，那么下一个APP然后下单就会有人送菜上门。在这个过程中，人力成本很高而用户痛点不明显，而且大部分用户是老年人，对此你的盈利性很小甚至没有。

2. 有足够大的市场

在07年时，我有个朋友在做玻璃行业的信息化系统的创业项目，他做的是定制一套系统卖给用户，这个过程很累，同时它的市场很小。我觉得市场太小会限制你的成长速度，因此，在选择业务方向时要有足够大的市场。

3. 有竞争但没巨头

一个行业可以有适当的竞争，但是不要有巨头出现。今年4月份我接触到一个做巴士的公司，它刚刚融到一百万美金，想让我去做CTO，但是我没有去。虽然那时候还没有滴滴巴士，但我认为滴滴肯定要做巴士，它既然做这个领域那肯定要垄断。面对强大的对手，你拿一百万美金怎么烧都烧不过别人，所以如果创业，不要去碰这个领域，以及一些O2O的细分领域如58到家、美团、糯米等。如果选择做小品类的O2O，需要找到目前该领域存在的明显壁垒，否则你根本没有机会。

如何对股权结构进行合理的划分：

要想创业、建立一个团队，你就要把钱分清楚，否则后患无穷。如何把钱分清楚，这就需要合理的划分股权结构。

1、要有带头大哥

很多时候，包括我07年第一次创业的时候，我们的股权结构是平均的，平均的最终结果就是决策不了，所以创业时必须要有一个带头大哥。这个大哥提供资金或者他的业务很懂，至少要占一样。如果没有带头大哥大家都差不多，那么你戏不大。创业时必须要有一个带头大哥。这个大哥提供资金或者他的业务很懂，至少要占一样。如果没有带头大哥大家都差不多，那么你戏不大。

2、需建股权池

对于创业来说，股权池是需要建立的。一定要把公司的股权预留出很大的空间，不要杀鸡取卵，在一开始就把所有的股权分出去。你要为后面进来的优秀人才留有足够的空间，而且你的角色应该是多元的。

3、需要4年授权

创始人要勇于牺牲、不能独占股份，但是创始人的股权也不是一下子分出去的，他可能需要在4年内慢慢的把股权分散出去。

如何在技术上变“快”：

天下武功唯快不破。对于创业，我们在技术上也要讲求一个字“快”。如何变快？如何以最低的成本变快？又或者说如何死得更快？

1、技术外包是永远的痛。

我们在创业时往往会尝试技术外包，但是外包公司里基本上都是入行才一两年的人，他们很难做得好。我的第二个创业公司就是这样。当时用了技术外包，他们花了两个月做出来的系统很烂，最后我们又重新做了一遍。这导致我们错过了去年最好的融资窗口，最终没有融资成功。

2、对于人才，Leader不要抠

在天使融资之后，你会发现急需用钱的地方很多，所以要一分一分的抠钱。但对于人才，你还是不要抠，留住人才是关键。以前，我们用了两个低端的人做安卓开发，对于一些小的细节，他们用了两天也没搞定，最后我请了技术高的人过来，五分钟把所有的事情搞定了。所以对于人才，不要抠，不要因为一个月差一万工资而请低端的人。如果因此错过了一个月的时间，也许你的机会就错过了。

3、擅于用第三方的服务

之前面试一个做测试的人，他说他花了很多力气去实现了其他人已经做出来的功能，我觉得很奇怪为什么不用第三方的服务、非要自己花费这么大力气呢。

时间对于技术人来说很宝贵，它可能决定着你的项目成败、机会得失等，因此，我们需要抢时间。如何去抢时间，这就需要我们擅于用第三方的服务，对于做技术而言，我觉得能用第三方的服务就不用自己的东西，要与时间赛跑。

如何组建和管理研发团队：

想要创业，你需要组建自己的研发团队。这样才能事半功倍，避免出现使用技术外包时所面临的问题。

A+C策略

相比于大公司来说，创业公司较难招到很多A类人，一般多是C类人。对此，如何进行团队的管理和优化，这就需要使用A+C策略即让A类人把C类人带出来成为B类人。

如何招到A类人：内推+猎头+高级顾问等

A类人尽量去大公司招，因为在大公司，他的视角看的远、知道这个东西应该怎么去玩。而且他有很多业内的朋友，也能指导你技术上的正确方向、避免犯错。如果你招到的A类人一直在创业公司工作，他的视角可能要比大公司的差一个级别，这就意味着你要为他交学费，所以leader最好是BAT出来的人。

如何招到这样的人，方式多种多样。如果猎头可以招到或内部推荐团队可以找到A类人，赶快定下来。如果实在招不到可以找替代方式，如可以聘请高级顾问。现在大公司里面很多资源是闲余的，虽然他们水平很高但是很少有在外面赚钱的机会。对此，你可以聘请他当你的技术顾问，为期一周或者两周一次，到你的公司来指导技术问题或是方向性问题。

研发团队的管理——看板

招到合适的人之后，除了迭代计划和每日晨会，我们一般会采用看板的方式来进行管理。

我们在白板上画出这样的线、每天制作你的卡片。在卡片上写具体要做的工作项目以及对应的负责人。它是一个很好用的工具，可以使你快速明白了解。

（原文节选，完整版文章请查阅EGO微信公众号：egonetworks）

听云CTO

陈靖华：创业就像一场必须坚持到底的长跑

原文作者: EGO编辑 陈园园

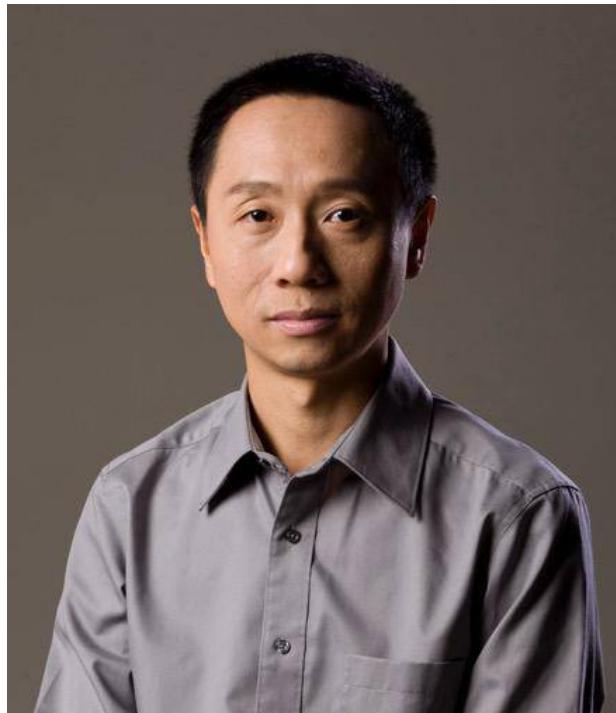
EGO: 您是什么时候加入听云的？目前在听云主要负责哪些方面的工作？

陈靖华：我是06年底的时候加入，是听云的创始团队人员之一。目前，我主要负责的就是技术方面的研发、运维、还有产品方面的工作，未来会更多的偏向产品方向的工作，我现在大概70%-80%的时间是在产品工作上。

EGO: 今年2月份的时候，听云完成了新三板上市，这对听云的发展是否有些不一样的影响？

陈靖华：登录新三板是对我们之前业务的一个认可，代表着我们SaaS服务的水平得到了资本和市场的认可。之前的投資商可能更愿意去做2C而不是2B的业务，这个事件，对2B的SaaS企业也算是一个里程碑。

对听云来说，登陆新三板更重要的是资本方面的认可，我们未来会在研发和技术上大力投入，用于新产品、新技术的研究，所以登陆新三板对我们的影响还是蛮大的，它能保证我们后续有持续的资金投入。上市之后，现在我们所有的业绩状况都是公开的，这是压力也是动力，促进我们更正向的发展。



EGO: 快速跟进新技术和服务的稳定性之间，听云是如何平衡的呢？

陈靖华：这个确实是很难平衡，但对我们来说更重要的还是考虑稳定性。性能监控的话其实有好几种技术，有几个维度的区分，考虑对用户性能的影响的话，一般会有主动式、被动式的区分。基本上所有的被动式，都是一种侵入式的方式，也就是说，我会把我的监测探针部署在用户的应用里面的，或者注入代码到用户的代码里面。这种情况下就可能会对用户应用稳定性和性能产生影响。这个影响的范围必须是非常可控的，同时在用户可以接受的范围内，否则就没有必要去做APM了。

我们现在做的一些新产品，就因为它是侵入式的，所以要求做的关于稳定性、兼容性等相关的测试的工作量就会非常多。我们会在内部做很多不同平台、不同版本、不同机型、不同服务器、不同操作系统的测试，包括它们彼此之间组合起来形成的多维的矩阵，矩阵的每一个点都需要测到。

我们每一个产品的迭代中，大概有60%–70%的时间，都是花在测试上，保证我们产品的稳定性是第一位的，然后才是去跟进用户的需求。

EGO：对于做产品和做技术之间的跨度，您是怎么看的？

陈靖华：跨度确实是蛮大的，困难也挺多的，对我来说，比较大的挑战可能还是UI、UE这些交互方面的东西。但相对2C的产品，我们公司的产品还是不太一样，最终用户人群基本上都是各个公司的IT部门或者是他们的研发、运维等技术人员，他们的思维方式我们相对来说还是比较了解的，我们能知道他们需要什么东西。所以，在听云做产品工作的，基本上都是从研发或运维转过来的，毕竟作为产品经理一定是从用户角度去考虑，怎么去帮助对方快速定位问题并解决问题，从技术转产品的话会有一定优势。也因此，听云很鼓励转岗，不论出于你个人的职业规划，还是你的个人能力和兴趣点考虑，我们都非常鼓励技术人员转到产品或销售、技术支持等相关的工作。

当然他们肯定也会遇到很多挑战，除了之前提到的UI、UE方面的，还有跟客户的交流沟通、了解他们的需求等很多新的问题，但其实这些都是可以想办法克服并解决的。之前提到，听云很重视大家的自我学习能力，只要学习能力够强，这些我觉得其实都没有问题。

EGO：在您看来，作为创业公司的CTO，哪些能力和素质是必需的？

陈靖华：创业公司其实有不同的阶段，一开始小的时候其实不需要什么CTO，反正基本上有事情就做，当时我就负责所有的跟技术相关的事，从前到后，从跟客户跑需求到编写调试代码、跑机房上架服务器，所有事都要做。

慢慢公司大了之后，除了技术能力之外，更多的会需要组织管理的能力，如何组织团队、搭建团队、培养团队架构等。特别是一般来说，一个团队中技术非常强的牛人是非常稀少的，那最重要的腰部、中层的力量是需要重点去培养的。所以，组织管理能力，是公司到了一定层面后，CTO一定要去重点关注和掌握的。

还有就是对整个技术团队激励的能力，如何去激励他们，让他们在技术上、学习上做更好的规划，这也是CTO能力中非常重要的一部分。这些团队方面的工作，可能比之前单纯做技术工作，更有挑战性。

当然,对我们来讲,因为听云本身是技术型的服务公司,提供的是技术服务,所以对CTO的来说,除了团队能力之外,还需要有更好的技术敏感度,能看到这个行业或者是整个互联网行业发展的一个趋势,同时还要重点去研究这些方面的技术,能做更好的产品路线图和公司发展方向的规划,技术会更多的参与到公司的决策中去。

EGO: 您平时的业务爱好有哪些?

陈靖华: 大概三年之前,在公司同事的影响之下,我也喜欢上了跑步,最近三年,基本上每天有时间就会去跑步。上上个月份刚刚完成了我的第二个全马,感觉还是比较兴奋的,因为比较顺利地完成了自己制定的完赛目标。

坚持跑步对我的整个生活、健康、工作带来了很多帮助,一方面在长跑的过程中我能释放很多工作中的压力,尤其压力大的时候健康就尤为重要。另外一方面,其实长跑,特别是马拉松跟做企业、创业都挺像的,你能看到目标在那,但是在途中你会遇到很多问题,会很痛苦,你需要的是要有一个坚定的心坚持把它跑完。

(原文节选,完整版文章请查阅EGO微信公众号: egonetworks)

EGO 会员感言摘选

Members' Words



“

杨卫华 新浪微博研发副总经理

对我而言，加入EGO最大的收获，是跟不同会员的深度交流，学习会员做事的方法与理念，了解他们对于行业方向的判断，以及对于技术的态度及选择。祝EGO在2016年越办越好！

范凯 丁香园技术副总裁

技术人需要走出去学习交流，去看整个时代、整个行业是怎样发展的。极客邦做了EGO，我很高兴参与进来，希望大家合作把EGO做好。祝EGO在新的一年里茁壮成长！

陈懿新 华为中央软件技术战略部部长

在这个技术创新井喷的时代，我们需要不断的学习和交流，EGO是技术人员的一个交流平台，大家在这里得到成长，收获技术与友情，祝愿EGO在新的一年越办越好！

梁宇鹏 环信首席架构师

一切都如期待的样子，EGO在半年内成功集结上百位京沪深三地的高端技术人，让人欣喜赞叹。值此元旦佳节，祝EGO在新的一年里继续茁壮成长，照亮技术人的前行之路！

艾美 百姓网技术总监

加入EGO，是源于自身对成长的渴望。交流，解惑，学习，共进……我们不孤单：）

黄良懿 OPPO数据中心总监

EGO 让思想在这里碰撞出新的火花, 经验在这里分享和印证, 点头之交逐渐变成了亲密伙伴。祝 EGO 越办越好, 帮助更多资深技术人进一步提升!

史海峰 当当网架构部总监

EGO提供了一个平台, 让大家相互交流, 分享经历、经验, 这些都是无形的知识。

刘建国 果糖移动CTO

大家在EGO一起探讨问题, 找到互相交流和陪伴的感觉, 毕竟创业的人都是孤独的。

郭理靖 同仁堂国际 CTO

大家面临的挑战是类似的, 而且有些问题不便公之于众, EGO为大家提供的线下私密交流的形式就会好很多。

蒋炜航 宜信大数据VP

在EGO的活动中, 通过几个小时的长谈, 大家深刻交流过去几年的工作经历, 这种体验真的是与众不同且让人难忘的。

陈靖华 听云CTO

EGO帮我打开了一个新的交流的通道, 大家很多人都是从技术转到管理上, 确实帮助很大。

陈锡言 风行网技术总监

加入EGO后能有更多学习的机会, 能够获得技术上的提升, 也希望能和更多更优秀的技术人有更多的交流。

邹光先 爱拍原创联合创始人&CTO

EGO就提供了一个很好的交流平台, 大家互相学习, 碰撞出火花, 也能找到志同道合、理念相合的人, 非常不错。

丁宁 荔枝FM CTO

通过EGO我其实能认识很多人, 能够学到很多东西, 这个平台真的是很好。



王 栋 雪球CTO

可以在EGO认识更多的业界的朋友，大家都是水平比较高的同学，可以相对比较平等的去交流，是一个非常好的学习的过程。

韩 卿 Apache Kylin Co-Creator

EGO能够把不同背景、不同行业的人聚集在一起，形成一个线下交流的组织，让大家互相学习、互相吐槽、互相成长，这正是我们现在非常缺乏的。

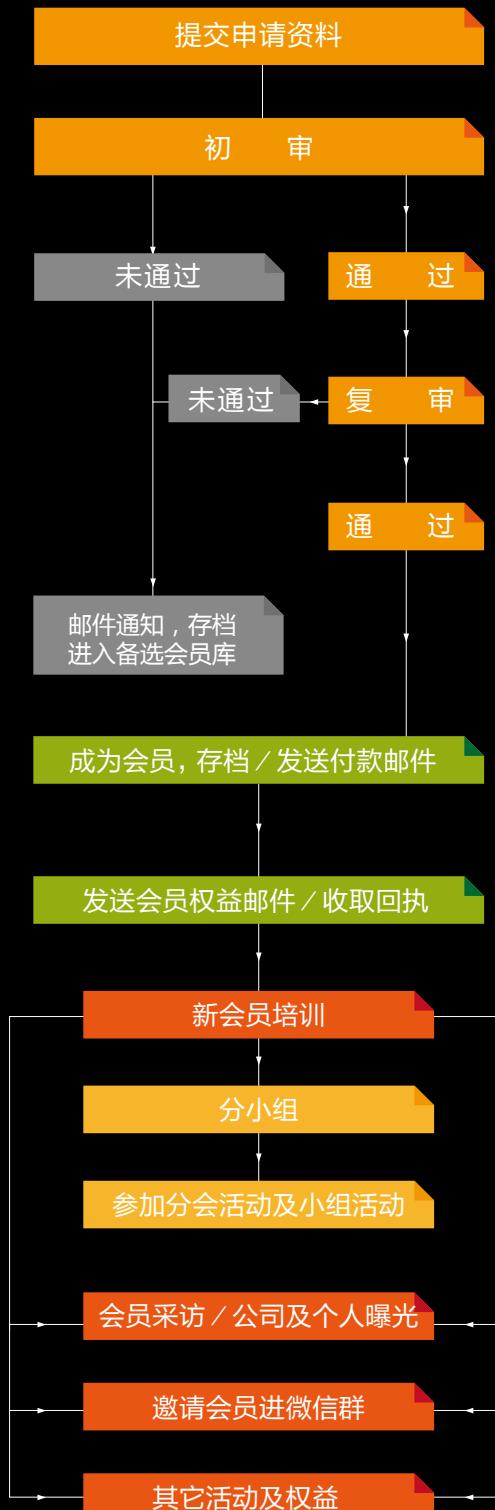
黄慧攀 UPYUN CTO

我们这些技术人，通过EGO这个平台，定期出去跟不同行业的人坐下来聊聊天，就能起到很好的补充作用。

王 涛 巨杉数据库CTO

我觉得EGO是一个很好的交流平台，也希望EGO能继续扩大影响力，覆盖人群越来越多，圈子越来越大，彼此之间的沟通交流也越来越多。

EGO会员申请流程



加入Join EGO

为什么加入EGO/Why Join EGO

★ 学习交流

私密独享线下小组交流, 顶尖技术专家、商业领袖分享

★ 拓展人脉

EGO年度盛会、企业间互访, 认识国内外志同道合的技术同行

★ 推广曝光

EGO会员专访及曝光, StuQ在线教育课程录制及推广, GIT企业培训顾问讲师身份自动获得

★ 获取信息

获取行业信息咨询, 独家原创专家观点, EGO解决方案智囊团

★ 会员福利

有限参与InfoQ中国主办的技术大会, 享受会员购票优惠政策, 会员间优质资源共享

★ 其他权益

EGO入会基本资格

EGO Membership Qualification

- ☆ 认同EGO理念, 遵守EGO规则
- ☆ 公司/团队中的技术决策者, 如CTO/首席架构师、技术总监等
- ☆ 热爱学习, 乐于分享, 追求成长, 相信技术的力量
- ☆ 在某一技术领域具有5年以上工作经验, 拥有2年以上技术团队管理经验, 曾管理最大团队规模在30人以上

EGO入会申请方式/How to Apply

- ★ EGO客服E小欧: 17091071219 (电话/微信)
- ★ EGO微信公众号: egonetworks
- ★ EGO微博: egonetworks



主办精彩活动包括：

- QCon全球软件开发大会,
- ChinaTechDay中国技术开放日,
- ArchSummit全球架构师峰会等。



EGO EXTRA GEEKS' ORGANIZATION NETWORKS



Extra Geeks' Organization Networks