

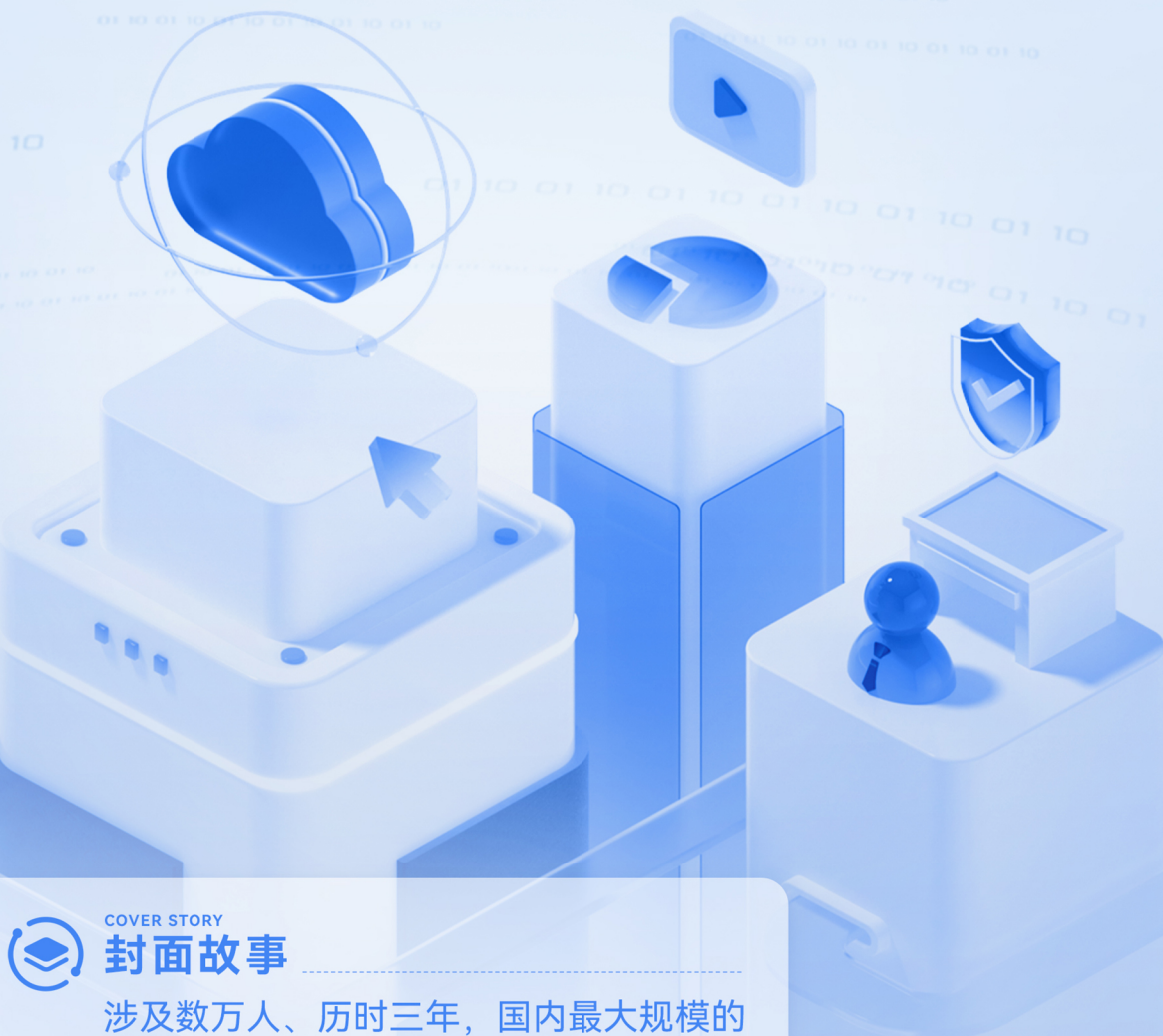
T O P T E C H T E A M

中国卓越

CHINA

技术团队
访·谈·录

2023年·第一季



COVER STORY

封面故事

涉及数万人、历时三年，国内最大规模的
云原生实践是如何打造出来的？

InfoQ 极客传媒

QCon

全球软件开发大会

广州站

50+ 国内外技术案例 两天一站式学习

大厂经验 | 实践案例 | 现场交流

专题出品人 陈炬
网易严选 数据中心负责人

专题出品人 朵晓东
蚂蚁集团 资深技术专家

专题出品人 罗庆超
阿里云 资深技术专家

演讲嘉宾 Luca Galante
平台工程社区核心贡献者

时间：2023年5月26日-5月27日

地址：广州·粤海喜来登酒店

热线：+86 18514549229

扫码查看详情>>



目录

封面故事

涉及数万人、历时三年，国内最大规模的云原生实践是如何打造出来的？	i
--	---

重磅访谈

从“幕后”走到“台前”，我们在阿里如何建设可观测体系？	1
大模型如何实际在行业落地：生成式大模型结合知识库，打造出 7*24 小时永远在线的超级员工	13
中国的“贝尔实验室”：我们的数据库从自己的第一行代码写起	31
我们不是野心家，走出大厂创业是时代使然	46

技术管理漫谈

可悲的现实，大部分技术领导者可能并不称职	60
如何为那些在裁员中幸存的人重建技术文化	74
架构师角色的演变：从发号施令到与团队合作	83

封面故事

涉及数万人、历时三年，国内最大规模的云原生实践是如何打造出来的？

编辑：Tina

采访嘉宾：邹辉、于广游、王涛

云计算的竞争旷日持久，表面看来格局初定，内里却在酝酿巨变。

具有先发优势的玩家，好不容易取得了看似不可撼动的地位，不曾想到有朝一日会中途被拉到同一起跑线，更换新的“CloudOS”重新出发。这个局面恐怕连 Kubernetes 早期创始人都会觉得不可思议。他当初只是想改变现状，在亚马逊的主导地位下，让谷歌取得一战之力。

2014 年，谷歌开源了 Kubernetes，红帽、腾讯、阿里、华为等国内外一众厂商开始力出一处，共同推进 Kubernetes 的采用。2017 年底，就连亚马逊也推出了 Kubernetes 产品，这也是 Kubernetes 成为标准化技术的最大信号之一。

这最终改变了整个云计算。

大家都开始基于 Kubernetes 技术生态去构建公有云产品，基于统一的标准以避免“深度绑定”，但这也让云原生行业严重同质化，因为各个云厂商所提供的功能和服务并没有太大的不同。对一些厂商来说，那些当年引以为豪的自研技术突破，那些树立在公司门口的纪念碑，那些专有性产品优势，都被抹平，这是一件非常残酷的事情。

同时这又是一些公有云厂商重塑格局的机会。锚定 Kubernetes 进行云原生改造，相当于给公有云更换“技术底座”，并由此构建出一些新的竞争力，从而赢取更多用户。

这场技术改造，难点在哪里？

对于腾讯来说，这不仅仅是一次技术“改造”，还兼带着腾讯全体系“自研业务上云”的战略任务。

在谷歌 GKE 之后，腾讯云于 2017 年推出了 TKE。但腾讯云对外服务时，还是会面临客户的质疑：“为什么腾讯自己的业务没有使用腾讯公有云，是不是不敢用？”

腾讯这次“云原生改造 + 上云”的价值就藏在客户的拷问中。腾讯在这二十年里，发展出了包括社交、音视频、游戏在内的多种业务，每种业务又都拥有海量的用户。全面上云腾讯不是第一家，但腾讯是拥有最复杂的业务场景的一家，在这个过程中，需要结合业务制定各种各样的技术方案，来满足不同的业务诉求。可以理解为，**每个业务的痛点都有局部最优解，而全面上云，则是在云上寻求通用最优解**。如果这些痛点都能解决，那这样的云服务是足够让大家敬畏的。

要运行这么多业务，云原生底座也不能有短板，必须承载得了微信、QQ、音视频、游戏等自研业务所有需求和核心能力，并最终将这些业务的技术积累和技术优势反向复制到公有云上，展现给外部用户。

除此之外，云原生改造还对组织能力提出了考验。

在移动互联网时代，腾讯发展出了自己的技术哲学：每个业务都有自己的技术团队，

每个团队都要打胜仗，这就要求“小、快、灵”，要有闭环。在自研业务上云之前，腾讯的每一个业务都有自己完整的技术栈，内部业务在一定程度上形成了“部门墙”效应。并且因为技术栈不同，员工从一个业务转岗到另一个业务，需要重新学习一遍技术，这跟换公司没什么区别。

根据财报数据，腾讯员工已超十万人，其中超过 7 成是技术人员，这是一次集体向云的迁移，就像一次“搬家”，但又不仅仅是将行李打包那么简单，它是将具有一二十年历史的不同特色的多个“大建筑”，制定“平移”方案迁移到新环境中继续安然运行，难度可谓前所未有的高。考虑到花费的时间、涉及到的人员规模、技术深度，这个项目可能是在世界范围内也很难找到的超级“软件工程”实践。这样的改造，过程中既有高层的推进、动员，也有执行层的博弈、妥协，最终实现了用一个点调动全局，让全公司的技术团队得到了一次很好的穿透对齐，让分散的技术能力得以统一。

有人说，评估腾讯云水平如何，应该参看自研业务上云后的整体水平和运转情况。

去年，腾讯自研业务初步完成了全部的云原生技术改造，腾讯云将所有的底层资源合并到一起进行统一管理和调度，自研业务上云规模突破 5000 万核，TKE 的在离线业务混部能力使服务器资源利用率从 30%提升至 65%，远远高于改造之前。2020 年，线上会议需求爆发，腾讯云组织了几十号人，花了 8 天紧急扩容 100 万核，创下了中国云计算史上的一个记录。而全部上云之后，放到现在这个阶段，利用一键扩缩容，腾讯会议再要去扩容 100 万核，那就是十分钟的事情。

所以，这次云原生改造的好处显而易见：对外，在垂直场景沉淀下来的技术能力，让腾讯云原生获得了差异化的产品能力，能真正解决用户在各种场景下的业务痛点；对内，让腾讯在云端整体的资源利用率有了一个大幅提升，这本身就是巨大的降本增效。

然而，这个过程却是经过了千辛万苦。

“像下一盘棋”

2018 年，云原生行业发展趋势初定。

随着云原生技术的兴起，腾讯内部几万研发人员的技术焦虑逐渐加深。早期腾讯积累了大量的技术架构理念，技术人员有非常强烈的自豪感，但是越是成功的组织惯性就越大，腾讯内部很多技术理念和流程还停留在上一个时代。据称，那时候腾讯内部讨论平台“乐问”上充满了技术人员的吐槽和争议。

除了“部门墙”的存在，每个业务部门为了应对突发的流量，在升级服务器资源时会留出资源缓冲区，当所有的缓冲区叠加在一起，就形成了大量的闲置资源浪费。所以，无论是从技术还是资源的角度来看，上云并进行统一的调度在当时已经是不得不做的事情。

2018 年底腾讯开了一次高层决策会议，决定将公司内部所有平台合到一起推行 K8s，开始进行彻底的技术更新换代。

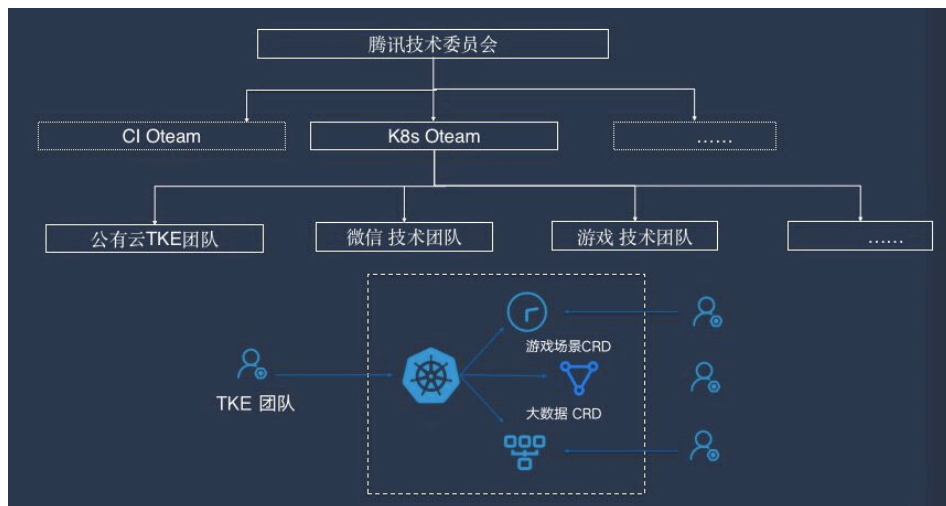
这个事情一开始由邹辉领导的 TKE 团队牵头。TKE 团队主要由一批资深技术人员构成，成员基本都在 30 岁以上，资历以 10 级、11 级为主，团队对成员的技术能力和业务理解能力要求很高。

决策已定，但是在执行过程中，尤其是 TKE 团队，前半年时间并不是真正的去做技术工作，而是跟腾讯内部几个事业群的平台技术团队去聊需求聊具体的改造方案，他

们发现还是存在很大的技术阻力。

腾讯云事业部门在 2016 年下半年的时候就启动基于 K8s 的 TKE 项目，但腾讯内部不同 BG 存在不同的路线，有的基于 Docker，有的基于 Mesos。现在要将所有东西都统一到标准的公有云 TKE 上去，其实内部技术团队难免会心生疑惑：你们是不是要过来抢我们的活？

为了减轻这些问题带来的阻力，当时腾讯没有采取调整团队人员和效仿建立技术中台的方式，而是制定了开源协同技术战略，把公司内部所有做相似事情的团队整合在一起，采取类似于外部开源运作的方式协同工作。这样既解决了技术浪费的问题，又可以去中心化，保持快速响应，还能更好地满足业务需求。腾讯内部把这种模式称为 OTeam。OTeam 挂在公司技术委员会下面。由这七八个平台组成的 K8s OTeam 就是一个典型的例子，它是腾讯首批三个开源协同项目之一。



在解决了技术团队的顾虑之后，腾讯从高层开始推进，说服自研业务团队上云，同时打通职级晋升体系，通过设置公司级的专项大奖、普及云原生知识、改造进度榜单晾晒等，从多个方面入手提高大家积极性，依照三年规划，有步骤地进行云原生改造和上云。

如何用好开源技术？

其实在上云决策制定之前，腾讯云已经花了两三年时间做了一个 TKE “原型”，也踩过了不少坑。

K8s 本身只是一个主要做容器编排调度的开源项目，TKE 底层是基于标准的 K8s，再在上面进行产品化，将 K8s 和网络能力、存储能力、日志监控等能力对应的网络产品、计算产品、日志产品、监控产品对接整合，给用户提供一个开箱即用的 K8s 产品，所以 TKE 对接了腾讯底层的各种 IaaS 产品能力。

2016 年腾讯开始做 TKE 的时候，国内都还没上 K8s 服务，业界比较好的产品设计也就是谷歌的 GKE，一切都是摸索着来。最开始，TKE 团队试图在云上提供一站式的 K8s 服务，将 K8s 的概念进行了一些简化，希望通过帮用户降低使用 K8s 的成本、让用户愿意直接接入 K8s，但最终发现这条路线是错的。

他们发现 K8s 不是直接面向终端用户的，而是面向一个企业内的 Infra 平台团队的。应该由 Infra 团队基于 K8s 构建自己的 PaaS 平台，提供给公司使用。“它是个 Kernel，是云的操作系统的内核、不是 PaaS。”于 2016 年加入 TKE 团队，一直负责 K8s 产品化相关工作的于广游表示。“我们没有意识到这样一个核心设计的本质。最开始，我们对它的理解有偏差，所以我们犯了一个错误，走了一些弯路。早期的时候为了面

向业务有一些改动，意识到错误后，在 17 年底、18 年初的时候就纠正了。后来才变成了我们现在 TKE 的形态，我们也因此做了一次产品改名，从 CCS 改名为 TKE (Tencent Kubernetes Engine)。”

到了 2018 年，腾讯启动开源协同之后，因为这七八个不同的容器平台团队，各自都有各自的优势，如果要融成一个标准 K8s 技术，该怎么做？TKE 要么选择都不接收、全部“作废”重来，要么选择将所有的历史包袱都背起来。K8s OTeam 在一起讨论之后，选择了后者。

这也是为了上云而做出的妥协。整个公司“像下一盘棋”，下棋是核心矛盾，往 K8s 里贡献不好维护的代码是当时的次要矛盾。据邹辉和于广游回忆，当时很快每一个团队都用上了 K8s，大家也都更加深刻地理解 K8s 了，理解到往 K8s 里面去改太多的逻辑，不是最优的方式。有了这个共识之后，K8s OTeam 团队在不更改 K8s 主线代码情况下，差不多用了一年时间，真的就把七八个平台所有的功能、核心技术特长全部融入到了 TKE 容器平台上。

大家在 K8s 基础上去添加功能，且无需向用户暴露 K8s 的基本概念，那么“零 K8s 基础”的用户也能快速部署应用并管理其监控、日志、服务注册在内的整个生命周期。后来腾讯创建了一套应用模型 Tencent Application Definition (简称 TAD)，直接使用应用管理平台，用户不需要去感知 K8s 细节，极大地降低了容器使用门槛。同时也引入了插件机制，复用了 K8s 的框架，可以像写 K8s 插件一样写 TKE 插件，方便第三方开发。

腾讯云原生底座的“养成”计划

相比公有云外部客户的业务，腾讯自研业务的体量更大，技术积累更深厚，测试标准也更全面和严苛，业务也千差万别。

一开始，K8s 很多能力不支持，业务很难平滑切换。在 2019 年之前，大部分业务还是基于虚拟机的方式去上云，因为自己的 IDC 物理机切到云上的虚拟机之后，这个过程业务基本上没有感知，整个架构和代码不需要任何的改造。但是业务上虚拟机违背了上云本质诉求，即希望利用云原生的快速弹性伸缩能力，和统一资源池的其他一些能力去提升各个业务团队的研发效能，所以最终 TKE 团队还是需要帮助业务从虚拟机切到容器化，并提供相应的产品能力。

TKE 平台在初期选择的更多还是一些**无状态**的业务，先让这些无状态的业务能够快速搬到云上完成改造。团队选择了一些平台的核心能力去解决业务痛点，比如说“发布”的问题。

在公有云场景下大家使用的是 K8s 基本的发布能力，比如基于滚动的发布。滚动发布过程可控性很差，遇到了问题后回滚，整个发布就会中断。腾讯自研业务需要满足**灰度发布**的要求，灰度发布对业务来说也是非常关键的一项能力。为了保证服务的质量，业务团队要求能够非常精准地控制发布频率、节奏和容错，做到发布过程一切尽在掌控之中。针对这样的需求，TKE 在自定义工作负载基础之上发布了一套灰度发布策略，业务可以指定要发布的 Pod，可以按照一定的百分比进行发布，也可以设置升级失败的比例来实现暂停或回滚。

同时 TKE 也给业务提供了一些虚拟机提供不了的能力，比如**动态路由**能力，在容器

销毁时，平台会将对应路由去掉，在容器起来后，平台会自动将容器加到路由中。使用虚拟机，业务需要自己去配置，使用容器之后，就不需要去管理业务的路由了，通过 K8s Operator 的机制已经实现自动化。如此一来，大家开始初步感知到容器带来的效率价值。

另外一个好处则是**弹性伸缩**和**健康感知**。之前使用虚拟机部署业务时，需要用户先购买虚拟机，再在虚拟机里去部署业务的包，再确认业务进程健康拉起运行，最后对路由进行管理.....这个流程在接入容器之后可以大幅简化，通过配置自动扩缩容的能力，或者手动触发，修改副本数后，后面所有的流程都是自动化的，可以做到秒级创建一批 Pod、自动感知实例健康状态并添加到服务路由里去，业务扩容非常丝滑。

还有就是**成本**上的优势，尤其是这几年，所有业务成本压力都比较大。容器在的优势是按量计费，Pod 销毁了就不收费了，计费粒度是秒级的，但虚拟机不一样，它的生命周期更重一些，弹性能力也比容器差，计费粒度也更粗。

此外，腾讯垂直业务场景也会给容器平台提出不一样的需求，为了满足这些需求，TKE 反之也给自己带来了差异化能力，这些最终都转变为了腾讯云原生产品的竞争力。

从垂直场景走出来的通用产品竞争力

从 2020 年下半年开始，腾讯游戏共有十多款产品陆续推动云原生改造，转向微服务架构。游戏是腾讯所有业务里软件结构比较特殊的一个，游戏服务的镜像一般比较大，有的甚至达到十几 GB。而我们每启动或更新一个容器，就需要将对应的应用程序从远端拉到本地的机器上启动，这么大的镜像，在部署的时候，开发对网络要求很高，源端就成了一个瓶颈。

为了解决这个问题，Oteam 团队在会议上讨论了很多次，商量出了一套“镜像分发系统”的解决方法，类似 P2P 下载网状结构，避免源端成为瓶颈点。据腾讯游戏介绍：“云原生架构里基于容器的快速扩缩容，是以分钟级、秒级来实现的，以前我们只能以十分钟为单位。”

提高镜像分发的效率，不仅仅是有益于游戏场景。在一些 AI 训练场景中，镜像甚至更大，几十 GB 也不少见，如果是需要发布成千上万个 Pod，那就需要几十分钟，甚至更长时间，所以现在这种解决方案同样也可以适用于大规模训练场景。

而在腾讯会议以及其他社交场景中，也有一些特殊要求，这种服务往往含有大量的会话信息，很多是长连接，有些业务还会大量使用共享内存，这些都属于有状态的服务。

无状态的容器扩缩容相对简单，但有状态的服务要去享受容器化的灰度发布、弹性伸缩能力，难度很大，需要对业务架构进行大量改造。因为业务不可能在短时间内做存算分离，把存储层下沉、上层逻辑层做成一个无状态的服务。所以容器就必须扛起这个责任，基于业务的这些特殊需求，在容器层适配有状态服务。

在**有状态的服务**中，如果在升级过程中对应容器的中断时间达到秒级，用户通话就会出现延迟和卡顿，所以在升级过程中就要保证 Pod 容器的中断时间控制在一秒以内。TKE 团队实现了一种自定义工作负载，将新版本业务镜像提前下载到 Pod 里，通过文件锁和容器状态探测机制来控制老版本和新版本之间的快速切换，将升级的中断时间控制在毫秒级别。

另一个不得不提的是原地**升级**的能力，比如说容器扩容的时候，不是通过销毁重建的方法扩容，而是通过原地无感知的提升扩容。比如一般公有云对 8G 的容器进行扩容，

会将其销毁重新创建一个 8 核 16G 的，这种对业务是有感知的。TKE 实现了更快速的原地升配，可以将 8G 的容器变成 8 核 16G，但业务对此是无感的，除此之外，还支持分批原地更新 Probe、Image 等能力。

另外，这几年腾讯会议经常遇到用户人数突然暴增的情况，比如每年 9 月 1 号秋季开学的时候，腾讯会议的用户量就会涨好几倍。腾讯会议应用程序内部有大大小小几百个模块，一个应用下面可能就包含几十个模块，运维人员需要做大量的紧急扩充容，手动完成一次对应用的扩缩容，针对这几十个模块进行操作，可能要投入很多的人力，需要很长的时间。

为了减轻运维负担，TKE 团队实现了基于 PCU，即同时最大在线人数，这么一个指标去做**一键扩缩容**的功能。比如说现在腾讯会议在第一天的同时最大在线人数 PCU 是一千万人，以此预测，第二天可能就是两千万人，那意味着腾讯会议的上下游整个链路基本上要扩容一倍。之前运维要去做这个事情，得去找整个腾讯会议几万个 Workload，然后对每个 Workload 将副本数扩一倍。为了提升这里的效率，腾讯自研了云原生全局一键扩缩容的产品能力，将整个腾讯会议关联的这些 Workload 构建成一个或者若干个业务拓扑，同一个业务拓扑内的 Workloads 支持等比例的一键扩缩容。

“我记得在早期的时候，腾讯会议这几百个模块，扩容几十万核可能要花个近半天时间，但我们把这个能力实现后，当大家再面对这种扩缩容场景时，20 分钟左右就能完成这几百个模块的共计几十万核的扩缩容。”

这种基于业务拓扑的全局扩缩容能力其实是一种普适性的大规模业务诉求，很多业务做活动都会基于一个北极星指标来进行容量评估。针对这个通用的需求，TKE 团队将之提炼成一个通用的产品能力，在 TKE 平台上形成了一个全局的（跨地域、跨集群）

基于业务拓扑的一键扩缩容的产品功能。

通过上面这些一个个的贴近真实业务的“小细节”，我们可以看出腾讯做云原生的思路是希望让用户的付出和痛苦最小、收益最大，尽量减少业务架构的改造，减少运维的压力。而且这些动态路由、无感升级等功能，王涛表示不仅仅是内部自研业务需要，“我发现很多外部客户平时都有类似的这种需求，他们也急需需要这样的一些产品能力，这也推动着我们将这些能力从内部推到公有云上去，提供给外部客户。所以，我们在腾讯自研业务上打磨的这些能力也变成了腾讯云产品的一个优势。”

深水区的那些痛

腾讯花了一年半的时间，将无状态业务搬到了云原生平台，几乎把能踩的坑都踩了一遍，为后续其他业务上云铺平了道路。这也证明了上云是可行的，给了业务团队更大的信心，后面就有更多业务滚雪球式地自发接入了。到了 2020 年底，上云的自研业务已经达到了三四百万核心的规模，平台也运行得非常稳定，所以 TKE 团队开始通过提升资源利用率、降低成本，来证明云原生确实能够给业务和公司带来很多实实在在的好处。

经过 2021 年整整一年时间，通过一系列的技术手段，团队把一些混部集群的利用率提升到了 65%。



同时在一些业务层面，一些**有状态的业务**，比如说像 Redis 数据库、中间件、一些大数据的套件，也做了原生改造，逐步搬到了整个云原生平台上来，腾讯内部数据库团队进一步开发了“云巢”云原生有状态服务平台。这个阶段差不多也用了一年的时间，最终到 2022 年，也就是到去年为止，整个腾讯内部的资源业务基本上完成了上云，整体资源达到了 5000 万核，3 年累计节省 30 亿。腾讯云包含了混部解决方案的开源项目 Crane 也经过认证，成为 FinOps 全球首个认证降本增效开源方案。

在这个过程中，TKE 团队在**调度层面**做了大量的工作。

在统一资源池中，资源分散在不同的 K8s 集群里，不同 K8s 集群的资源利用率参差不齐；资源需要在不同 K8s 集群之间流转，将闲置机器腾挪到繁忙的集群中，让每个集群的资源率都非常高，这个工作是特别困难的。

最开始，TKE 团队尝试优化每一个集群的资源利用率，同时通过在离线混部，把每一个集群中的额外的资源抽离到另外的算力平台中，进行统一的调度。这虽然缓解了很多问题，但随着利用率越来越高，干扰的问题还是会存在。为了解决这个问题，TKE

团队引入了新的统一调度方案，让 K8s 不再负责调度，只负责 Pod 的管理，真正去分配资源的时候，是将请求给到了 **Serverless** 调度器进行统一调度，解决资源使用不均的问题。

同时，因为 K8s 自带的原生 HPA 控制器，在这种大规模场景下，扩缩容会有非常大的性能问题，比如在业务流量的洪峰来临时来不及扩容，或流量出现抖动。所以腾讯将原来的控制器从 K8s 里剥离出来，**单独部署**，这样就可以进行单独的一些管理，如高可用、容灾等，同时对控制器里的内部实现逻辑做一些性能优化，来满足这种大规模场景下业务需要的秒级的弹性扩缩容的能力。

沉淀多集群管理能力

前两三年云原生行业都在“卷”单集群规模，通过优化 ETCD、API Server、Controller 调度器的性能，将单集群的节点规模做大，达到上万个节点。但最后瓶颈还是很明显，做到 5K 个节点集群跟一万节点的集群，本质上没有带来很大的业务价值，反而一旦单集群出现故障，爆炸半径会很大。

所以，在腾讯看来，一味地去突破单集群性能不是一个正确的技术路线。

最近整个社区，包括腾讯主要投入做多集群的管理。单集群做得更小，比如说两千个节点，甚至几百个节点就行；但是让更多的集群组合在一起，通过多集群的调度管理，让它看起来像一个集群，通过这种方式去扩展整个底层资源池的规模。

TKE 沉淀了各种多集群管理的能力，让上层的这种多集群管理能够去统一调度管理跨分区、跨地域的多集群资源。在此基础上，再重点解决了从面向集群到面向应用的调

度编排问题。

这在部署全球化的业务时非常有帮助。原生 K8s 是面向集群的一套编排调度系统，用户感知的是集群里面的 K8s 对象，没有提供基于可用区容量感知调度、副本分配策略决策这些调度能力。比如一个业务全网有一万个工作负载、五万个 Pod，分布在全球十七个地域，共八十多个集群。如果要对这个业务做一次全网变更，按照以前面向集群的方式效率非常低。

所以腾讯云原生团队抽象出了面向应用的能力，对跨集群应用的统一变更，提供了一个应用管理平台，用统一的看板跟踪发布是否正常。业务部署后还要能从视图看到部署的容灾是不是合理的，所以要有多地域的容灾检测。平台也可以根据用户定义好容灾部署策略进行巡检，出了异常可以自动告警。在面向全球化上，用户还可以利用全局一键扩缩容能力，对海外和国内的多集群进行等比例扩缩容。所有这些多集群编排能力都是基于腾讯云的 [Clusternet](#) 开源项目来建设。

进一步提升资源利用率，难度也不断加大

在过去三年多，腾讯统一了资源池，能够在一个大的资源池中调度虚拟机、容器和函数，最大化地利用物理机的资源。业界很少有这么大规模的资源池，当规模足够大，底层的环境足够复杂时，总会遇到一些别人遇不到的真实问题。

在不断提升资源利用率时，你会发现，这其中大部分的时间都必须跟内核打交道。

当利用率提升了之后，整个节点里面的内核资源抢占的问题会越来越严重。同一个节点上面部署了不同的业务，甚至上十个业务，这些业务都在一个节点上，利用率高的

时候会出现**网络带宽、内核锁**等各种各样的问题。每次遇到问题的时候，TKE 团队都需要和内核团队一起去分析，经常需要内核团队经常提供热升级补丁，或者在下一个升级版本中去做优化；然后为了减少“抢占”，也需要通过内核优化资源隔离能力；还需要完善内核资源的监控力度。比如说内存的分配时间，CPU 在队列里面的等待时间，这些很详细的内核稳定性指标，会由内核暴露出来，给到容器。容器结合这些内核的稳定性指标再去调度决策，以提升整个节点的稳定性。

在大规模资源池里追求极高利用率的场景下，还需要考虑几十万个节点的内核版本的管理，也就是说一定要把这么多节点的内核版本给收敛起来，不然太过零散，这些内核问题永远都处理不完，一定要有一套自动化收敛节点内核版本的机制。所以 TKE 团队做了一个基于业务无感知调度腾挪能力，去自动化升级节点内核的系统，可以在业务低峰期的时候，比如每天凌晨的时候，自动化地分批次挑选最合适的节点，升级这个节点的内核版本，逐步地、自动地将整个平台的节点内核版本收敛起来。

另外，因为腾讯是一家一二十年的老企业了，当将所有资源都合并到一起后，就会存在有机型代次差异的不同服务器硬件，而不同代次的机型，算力是不一样的，如果同一个工作负载的不同 Pod 位于不同代次的机器上，这就可能导致不同 Pod 的负载极其不均衡。为此，TKE 研发了基于机型的性能动态修改每一个 Pod 对应的路由权重能力。当一个 Pod 底层用的是些很老的机型的时候，会自动调低对应的路由权重；当 Pod 底层的机型比较新的时候，对应的权重会更大。通过这种方式，打平了不同 Pod 之间的负载，用户看到的每一个 Pod 的负载都是均衡的，最终达到对业务屏蔽机型差异的目的。

另外，不同可用区域的资源余量也不一样。为了解决资源问题，业务往往需要在不同可用区域之间来回腾挪。为了让业务更加充分地利用不同可用区域的资源，能够灵活

地不同可用区之间调度，甚至做到业务不感知可用区域的属性。TKE 应用管理平台提供模糊可用区的能力，彻底屏蔽 K8s 节点、集群、可用区的概念，让大部分业务完全不感知这些资源的属性，充分利用不同可用区的资源，同时让业务具备跨区域容灾能力。

云原生路线图

如果要总结腾讯云原生的特色，那可能主要有三点。

第一点是超大规模，这种体量规模至少在国内没有第二家。第二点是业务场景极其丰富，包括社交、音视频、游戏、支付、腾讯地图等业务场景。第三点，这些腾讯自研业务对稳定性、容灾要求非常高。

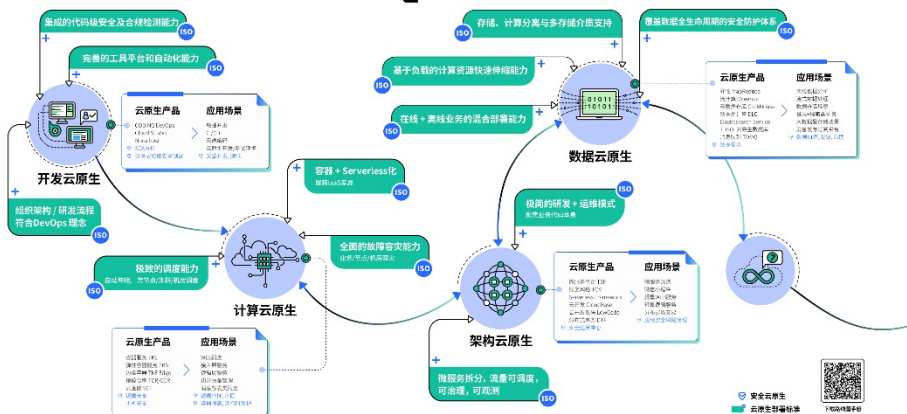
王涛总结说，“我们做这个事情最大的压力是要保证容器化之后业务的稳定性，如果不小心把一个集群搞挂了，或者出现大面积的节点宕机，影响业务运行，这个后果就非常严重。也就是说我们从一开始就理解到业务对稳定性要求极高，大家都是如履薄冰，做事情在细节上会考虑非常完善，因此 TKE 服务腾讯自研业务这么长时间，平台没有遇到过大的故障。”

如今 TKE 平台在腾讯内部已经成功承载了数以亿计的容器，支撑众多海量业务平稳运行，这个持续三年的改造项目，用邹辉的话来讲，它不是一锤子买卖，而是一个持续迭代、持续更新的过程。

腾讯根据自研业务以及目前一些外部企业使用 TKE 进行云原生改造的经验，设计了

一个五阶段的路线图，希望能够给其他企业带来参考。

「腾讯云原生最佳实践路线图」 让用云更简单



采访嘉宾简介

邹辉，腾讯云原生产品中心总经理。

现任腾讯云原生产品中心总经理，全面负责腾讯云容器、Service Mesh、函数计算等产品相关业务和团队管理工作。自 2010 年加入腾讯以来，先后带领技术团队负责腾讯内部多个高性能通信框架及缓存系统，服务于腾讯内部多个海量业务；2017 年开始从无到有搭建腾讯云容器产品，推动容器产品规模连续多年保持三位数以上速度增长，并取得行业领先地位。

于广游，腾讯云专家工程师，腾讯云云原生产品中心副总经理，主导了腾讯云容器产品从 0 开始的设计、研发和运营工作，并在腾讯云海量 Kubernetes 集群的治理和落

地过程中积累了大量的经验。腾讯自研业务全面云原生上云的主要参与者之一，在云原生领域有丰富的实践和思考。目前致力于 Kubernetes 在成本节省、Serverless、混合云等场景的探索。

王涛，腾讯云专家工程师，腾讯云容器平台负责人，9 年 K8s 生产经验，从 0 到 1 建设服务自研业务的 TKE 平台，全程支持了腾讯海量自研业务容器化上云，熟悉各种业务场景的容器化挑战和技术方案。

重 磅 访 谈

从“幕后”走到“台前”，我们在阿里如何建设可观测体系？

作者：凌敏

采访嘉宾：阿里云云原生可观测团队

IT 系统的运维监控最早出现在上世纪 90 年代。彼时，分布式架构正向传统的单体架构发出挑战，其带来显著优势的同时，也为系统开发和运维带来了新的难题。在这一背景下，IT 人员开始引入监控技术，观测主机上的应用运行情况，及时定位问题。

随着分布式系统、微服务、云计算技术兴起，IT 系统发生多轮演进，复杂的运维环境对监控提出了更高的要求。2018 年，CNCF 将可观测性引入 IT 领域，取代监控。可观测性也一跃成为云原生技术领域最热门的话题之一。

5 年后的今天，可观测性技术早已从早期的运维排查问题工具，逐渐进化成业务生产过程中的生产力工具。Gartner 更是将应用可观测性列为“2023 年十大战略技术趋势”，并表示“如果能够在战略中予以规划并成功执行，可观测性应用将成为数据驱动型决策的最强大来源”。

作为阿里巴巴集团最早的监控&可观测团队，云原生可观测团队早年打造了 EagleEye（鹰眼）作为分布式调用跟踪系统应用于阿里内部各业务线，随后将该工具进行产品化，结合云上客户的广泛需求，打造出了阿里云应用实时监控服务 ARMS。

那么，阿里云云原生可观测体系的建设背景与历程是什么样的？可观测体系建设的重

难点是什么？如何从内部自研走向产品化？2023 年，企业和开发者应该如何理解可观测性？在本期访谈中，InfoQ 有幸采访到了阿里云云原生可观测团队的多位核心成员，以期找到上述问题的答案。

阿里云云原生可观测体系建设历程

2010 年 4 月，Benjamin H. Sigelman 等人在 Google Technical Report 上发表了一篇名为《Dapper, a Large-Scale Distributed Systems Tracing Infrastructure》的论文，介绍了 Google 生产环境中大规模分布式系统下的跟踪系统 Dapper 的构建和部署经验。这篇论文正式揭开了分布式链路追踪的技术大幕，也为后来涌现出的包括 EagleEye 在内的分布式调用系统提供了灵感源泉。

分布式链路追踪 EagleEye 的设计与实现

2012 年，阿里的淘宝电商业务正处于高速增长期，为满足业务快速迭代的需求，支撑不断提高的交易量，阿里采用微服务架构对整个业务逻辑做了一次重构。微服务架构在性能、可维护性和可用性上带来优势的同时，也带来了四大难题：

- 故障定位难：一个简单的下单购买操作背后是由十几个甚至数十个微服务共同完成的，这些微服务又由不同的团队负责，微服务的过度协同带来的结果就是，一旦出现问题，需要十几个团队一起来解决；
- 容量预估难：在大促场景下，过去只需按照预估的流量与当前系统的单机压测容量做对比，再将所有的系统按比例去扩容即可，但在微服务架构下，每个系统在

核心链路中的参与度、重要性都不同，无法进行等比例的扩容；

- 资源浪费多：这也是容量预估不准造成的后果，同时，资源浪费多也会引发性能优化难的问题；
- 链路梳理难：复杂的微服务体系，让各个微服务系统的负责人很难梳理清楚每种业务的上下游细节逻辑对自身系统的影响。

“我印象比较深刻的是，当时淘宝已经迭代出了上百个应用，但却没有一个业务架构师能够讲清楚整个业务的系统架构是什么样子的。正是在这个时候，我们遇到了 Google 的《Dapper, a Large-Scale Distributed Systems Tracing Infrastructure》这篇论文，我们参考了 Google 的主要思想，在阿里内部做了落地实践。” 阿里云可观测技术负责人司徒放回忆道。

正是在这一背景下，EagleEye 应运而生。EagleEye 是一个以链路追踪技术为核心的监控系统，通过收集、存储、分析分布式系统中的调用事件数据，协助开发和运维人员进行故障诊断、容量预估、性能瓶颈定位以及调用链路梳理。2012 年，EagleEye 第一次发版，EagleEye 1.0 能够构建链路跟踪核心体系，提供调用链与离线报表服务。

当时虽未出现可观测性一词，但业界已将其分解为三个更具体的方向展开研究：Metrics（指标）、Tracing（链路追踪）以及 Logging（日志）。这三大方向也是此后 OpenTelemetry 协议定义的可观测性三大支柱。

最开始，EagleEye 主要关注 Tracing 领域，这也是业界比较早期的大规模 Tracing 实践。“EagleEye 解决了阿里当时微服务架构下的分析和诊断挑战，有很多阿里内部研发人员在应用它。” 司徒放说道。

2013 年上半年，EagleEye 打通了淘系所有常见中间件的调用数据，应用负责人能看到自己的系统在整个链路上的执行情况，为大促和单元化的容量规划、依赖分析提供了数据支撑，并具备了快速定位分布式系统故障的能力。

EagleEye 的第一个重要发展契机是淘宝的双十一大促，这是 EagleEye 的首次大规模应用。也是在这时，EagleEye 上线了实时链路大盘，为全链路压测提供压测透传和链路来源流量分析。

当时，为了提前做好系统容量的准备，阿里对线上系统进行了全链路压测，而全链路压测的底层其实是与 EagleEye 的 Tracing 能力紧密相关的。于是，EagleEye 从 Tracing 领域切入到 Metrics 和 Logging 领域，通过 Traces 去做流量级别的、精准的 Metrics 数据统计，用来分析上下游应用的依赖关系，提供一个全局流量拓扑。同时抽象了一个通用的实时日志处理系统，用来做通用的日志采集、计算、存储方案，进一步提升问题排查效率。

EagleEye 1.0 之后，团队陆续发布了 2.0、3.0、4.0 版本，围绕链路跟踪的核心能力，EagleEye 逐步构建了集监控、诊断、优化于一体的综合性服务平台：提供多维查询与实时报表以及数据分析能力；基于内存统计，提供精准实时报表与系统监控服务；基于链路追踪，提供单机诊断与业务定制化服务。

脱胎于 EagleEye，2017 年，云原生可观测团队打造出了应用实时监控服务 ARMS，并于 2022 年 6 月推出阿里云云原生可观测套件（Alibaba Cloud Observability Suite），打造云原生时代完整可观测数据生态与产品套件。

从内部自研工具到产品化

EagleEye 从最初的内部自研工具走向产品化，是一个自然而然的过程。

在自研体系下，EagleEye 的出现为阿里内部降低了成本和风险，提高协作效率。同时，在云原生变革的大趋势下，云原生可观测团队也越来越深刻地意识到，PaaS 类产品一定得是开放的，是基于标准和开源的。结合云上客户的广泛需求，云原生可观测团队将 EagleEye 进行产品化，打造出了阿里云应用实时监控服务 ARMS。

“阿里内部的技术栈是比较统一的，在内部业务场景中，我们采用完全自研的形态去支撑业务没有任何问题。但外部客户的技术栈百花齐放，要求所有人学习和使用我们的技术体系并不现实。于是，我们的整个可观测体系从自研转向了开源的 Prometheus。”司徒放提到。

Prometheus 最初是由前 Google 工程师在 SoundCloud 上构建的开源系统监视和警报工具包，自 2012 年创建以来，许多公司和组织都采用其作为监控告警工具。2016 年，Prometheus 加入 CNCF，成为继 Kubernetes 之后的第二个 CNCF 托管项目。

如今，Prometheus 已经成为云原生时代的可观测事实标准。根据 CNCF《Cloud Native Observability MicroSurvey》调查，84% 受访者在可观测技术栈中使用 Prometheus。

司徒放表示，转向 Prometheus 架构后，再去审视移动端、应用、中间件、云服务的基础设施监控等子领域，我们发现采用开放的体系可以让整个数据模型得到统一，整个实体关联也更加简单。“这和当时秦始皇统一六国后实行的书同文、车同轨、度同制道理是一样的。我们的整个可观测体系统一后，打通了很多数据孤岛。”

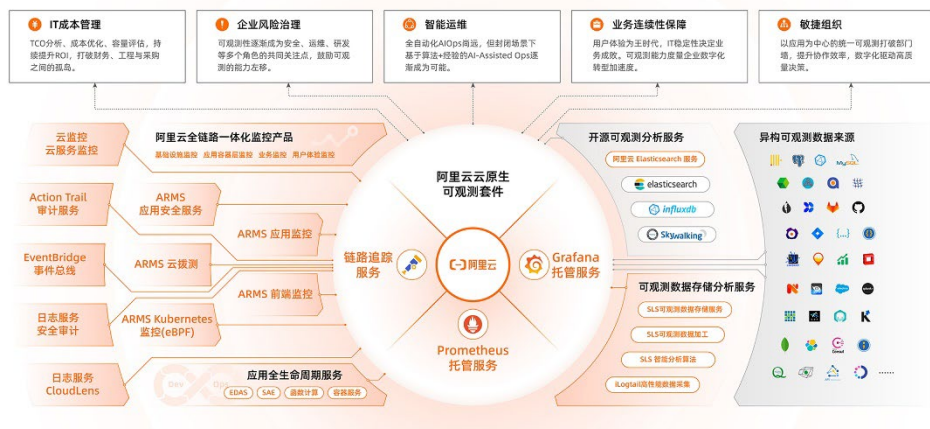
在转变的过程中，云原生可观测团队也面临着来自技术与思维的双重压力。

技术上，过去团队支撑内部业务应用，可观测工具能够保证运行稳定、快速响应业务需求即可。但可观测云产品的市场需求多元，对易用性、可扩展性，以及应用集成的能力要求较高，并需要满足数据安全标准。思维上，过去团队着重关注技术，但现在也需要考虑如何做好商业化产品，需要设计产品的差异化能力，以及盈利模式、用户增长。

“这些对于我们团队来说，都是过去没有经历过的问题，我认为这个过程也是一个很好的机会。能够把自己擅长的技术打造成一个可能会被全球各个企业广泛使用的产品，并与其他世界一流的产品展开竞争，对于程序员来说，这应该是最大的荣耀和梦想。”司徒放说道。

在 Gartner 发布的《2022 Gartner 应用性能监控与可观测魔力象限》中，Gartner 将阿里云定义为此魔力象限中的细分领域者，并给予了高度评价。

当前，阿里云可观测产品由核心产品应用实时监控服务 ARMS、Prometheus 监控，联合云监控 CMS、日志服务 SLS 共同组成，以公共云 SaaS 服务、混合云不同产品形态为不同规模、不同业务需求的企业提供开箱即用的可观测服务。其中，阿里核心容器调度（千万核规模）以及超过 50 款云产品，全面基于 Prometheus 构建可观测体系。



在提到与同类型产品的差异化时，阿里云可观测高级技术专家徐彤表示，阿里云可观测产品及解决方案的显著优势是上下游产品生态的支持，结合阿里云整个原生生态，用户可以在一个统一的平台上实现对云计算应用的全方位可观测，效率与便捷性大幅提升。过去，EagleEye 在阿里内部积累了大量运维场景经验，脱胎于 EagleEye，ARMS 也积累了丰富的行业应用与运维经验。

此外，阿里云可观测产品形态丰富，阿里云 Prometheus 监控托作为完全兼容开源 Prometheus 的全托管监控服务，提供开箱即用的 Grafana、智能告警等组件，并预置常见场景模板。用户无需关注系统搭建与日常维护，有效提升运维监控效率。在开源开放方面，阿里云可观测产品兼容业界通用的 OpenTelemetry 标准，支持多语言协议及 SkyWalking、Jaeger 平滑迁移。

“开放与集成是我们很重要的能力，我们坚持将开源标准和产品集成到平台，方便用户在现有产品的基础上进行拓展和优化。同时我们也在积极贡献开源社区，与业界共同推动云原生可观测生态发展。”徐彤介绍道。

下一阶段，阿里可观测产品体系的重点是数据的集中管理和关联，整个系统端到端的分析将更加全面、完整、统一，同时结合可观测下游服务与人工智能技术，从监管控一体化走向智能化运维。

云原生可观测团队：从“幕后”走到“台前”

从打造 EagleEye 应用于阿里内部各业务线，到打造阿里云应用实时监控服务 ARMS 服务更多企业，云原生可观测团队基于可观测行业趋势，将开源项目与商业产品相结合，帮助越来越多的企业获得完整的可观测能力，节省运维成本。

徐彤表示，随着当前千行百业以及云产品对可观测的重视度与日俱增，可观测在云原生体系中所扮演的角色也发生了变化，很多团队会主动找过来，一起建设自己的可观测能力。“在过去，我经常说我们团队是幕后无名配角，但现在，我们也会开玩笑说，自己已经慢慢地走到了台前男二号的角色。大家更重视我们，我们也会提供更多、更重要的能力，二者结合，越走越远。”

而 EagleEye 既是阿里云云原生可观测团队打造的第一个产品，也是云原生可观测团队最初的名字，其诞生背后的逻辑其实是客户需求和行业趋势决定的。

据司徒放介绍，企业做云原生数字化转型会先选择做容器化、微服务架构的改造，这也导致整个开发、运维体系以及协作模式发生翻天覆地的变化。在这一背景下，云原生团队开始进一步孵化出可观测团队。目前，云原生可观测团队主要由包括应用监控、链路跟踪、Prometheus 以及告警前端监控在内的技术团队，以及运营、解决方案等

团队构成。“在整个云原生大团队中，可观测团队的定位就是为我们的客户提供一套统一、高效、易用的原生可观测解决方案。在日常工作中，云原生可观测团队与容器、存储和网络、安全、解决方案等多个团队密切协作。”

作为可观测产研团队的重要组成部分，运营团队也在其中扮演着重要作用。

“云上产品和服务越来越走向 PLG (Product Lead Growth) 模式，运营也变得更加重要，作为足球比赛中‘中场’的角色，即衔接用户需求与产品服务。”阿里云可观测高级运营专家王希正认为，在市场侧，从应用性能监控领域到可观测，市场发生了很大变化，运营团队需要构建全新的技术内容，协助开发者快速找到解决自身问题所匹配的技术方案，并配套开发者体验场景及优秀的文档能力，让开发者能够自体验自生产自服务。在服务侧，可观测产品作为数据汇聚地，如何呈现数据从而辅助决策至关重要。运营团队需要提供丰富的模板，帮助开发者自助享受云上可观测能力，真正用起来、用得好。

作为“中场”，运营团队还需要拥有出色的数据洞察能力，发现不同业务场景下的主流用户需求及体验问题，驱动产品研发为用户提供更好的服务。“未来，B 端产品尤其是以 SaaS 交付方式提供的产品，产品驱动增长的 PLG 模式将成为主流。运营同学除了做好增长外，更多需要深入到自身产品的使用体验中，基于市场的洞察推动产品改进到一个领先的位置。这也会是整个团队的一大助力。”王希正总结道。

可观测力即生产力

2018 年，CNCf 将可观测性引入 IT 领域。5 年后的今天，可观测再次获得了广泛关

注，并被 Gartner 认定为“2023 年十大战略技术趋势”。Gartner 表示，可观测性使企业、机构能够利用数据来获得更加明显的竞争优势，在最恰当的场景挖掘出数据背后的战略价值，以便规划与决策战略方向而不是盲目的快速行动。因此，可观测性应用作为一种强大的工具，如果能够在战略规划过程中充分使用，这将成为数据驱动型决策最强大的支持。

阿里云可观测高级产品专家曹剑认为，与 5 年前相比，当前可观测市场正迈入一个全新的阶段——数据为王。

随着千行百业进行云原生架构转型升级，可观测数据量得到了指数级增长，数据之间的关联分析以及传输也带来了可观测数据模型的标准化，标准化也会反过来促进数据的上下游协作，并进一步催生出多个细分市场。比如，有些厂商专门做可观测数据的采集，有些厂商专门做可观测数据的编排，等等。而在安全、软件质量等场景下，新的可观测产品也瞄向了新的客群。“现在不单是运维同学在应用可观测产品，运营、市场以及管理者，都开始使用可观测工具。在这样一个数据为王的时代，谁能够把可观测的数据价值挖掘出来，谁就能够给用户提供更好的可观测服务。”

与此同时，随着国内云服务越来越成熟，开发者用云深度增加，可观测作为具有代表性的云服务，已经从早期的运维排查问题工具逐渐变成了业务生产过程中的生产力工具。

王希正表示，从近些年国内外可观测的发展来看，有一个很明显的趋势是，企业的可观测力即生产力：

从云产品视角看，当前云计算不再只提供资源服务，越来越多的云产品能够帮助企业

更高效、更易用、更安全的创建云原生应用，使业务研发模式发生深刻的变革，各个云的组件本身都需要提供开箱即用的可观测力。以 MQ 为例，它其实代表了企业业务数据的流向，云上的 MQ 通过 Prometheus+Grafana 提供的可观测能力可以快速对线上消息消费问题进行排查和定位，还能通过分析消息流量变化趋势、流量分布特点或消息体量，帮助客户更好的进行业务规划，这也驱动了可观测的发展。

从开发者及企业视角看，构建可观测力不止是运维或者 SRE 部门的事情，越来越成为企业业务决策的一个方向。对于产品业务部门来说，可观测能够将业务数据与 IT 性能与指标关联，是 PLG 产品的必备条件；对企业管理者来说，可观测产品有助于构建高效的 Issue to resolve 流程，为 IT 和业务提效。

“整体来说，驱动可观测增长的重要动因实际是当前越来越激烈的市场竞争以及高效的研发模式，从过去的月度发版到很多业务的周迭代，势必要引入可观测能力去主动发现流程与过程问题，而不是在问题出现后才去解决。”王希正总结道。

写在最后

2023 年，可观测性技术还将持续地发展与演进。

未来，AI 驱动的可观测将得到更多应用，并对数据进行了更加高效的整合，故障预测、异常检测、自动化业务都会逐步变得智能化，可以更好地降低成本。同时，随着低代码的应用更加广泛，企业运维门槛降低，可观测性技术也需做好迎接这一趋势的准备。此外，用户对安全隐私保护重视度与日俱增，未来的可观测产品设计中，也需要更加注重合规性与安全性，充分保护数据。而在实时性方面，随着应用程序以及基础设施

建设完善，可观测系统本质上等同于大型数据处理系统，如何更快、更稳定地处理数据也是从业者需要思考的课题。

徐彤表示，下一步，云原生可观测团队会重点关注几个方向：

- 可观测性技术演进：持续关注可观测领域的技术动态，并与开源项目以及行业标准相结合，优化现有产品。同时探索 AI 技术在可观测领域的应用，帮助用户实现更加智能的自动化运维。
- 可观测性体系化建设：当前的可观测产品主要聚焦在运维监控阶段，在整个软件生命周期的管理中位置偏后，需要结合其他技术继续往前走。
- 解决方案：每个行业在不同阶段都有自己的可观测解决方案诉求，需要深入了解各个行业的痛点以及诉求，为用户提供一站式的可观测能力。
- 持续运营：加强运营动作，帮助用户更好地理解和应用可观测性技术。
- 全球化：关注全球化的可观测性诉求。

嘉宾介绍

司徒放，阿里云可观测技术负责人，资深技术专家。

徐彤，阿里云可观测高级技术专家。

曹剑，阿里云可观测高级产品专家。

王希正，阿里云可观测高级运营专家。

大模型如何实际在行业落地：生成式大模型结合知识库，打造出7*24小时永远在线的超级员工

作者：刘燕

采访嘉宾：中关村科金·AI平台能力中心

对话式 AI 产品拥抱大模型

一个大胆的决定

自 2014 年成立以来，中关村科金就选择专注于企业服务赛道提供对话场景服务，聚焦生成式 AI 技术，包括领域大模型、大数据分析、多模态交互三大核心技术。如今基于这三大类核心技术，已形成了一套完备的技术体系，并构建了一个基础的技术底座——即生成式的得助对话引擎。

这些底座式的能力都由 AI 平台能力中心来提供支持的。这是一个在内部被定义为偏底层、汇聚“原子能力”的地方。这个能力中心，既要构建前沿的技术能力，又要快速响应前端业务系统的变化，构建标准化的产品组件用以快速落地。

从得助对话引擎上“长”出了三大类产品，包括数字化洞察与营销、数字化服务与运营、数“智”底座，基于这三大类产品陆续推出云呼叫中心、全媒体智能客服、智能

外呼机器人、智能质检、智能陪练、智能音视频等一系列服务，覆盖了用户服务的全生命周期，包括售前、售中、售后等环节，被广泛应用于金融、政务、零售、大健康、制造等行业。

“我们一直在探索对话式 AI 技术在企业服务赛道的创新应用，同时积累数据，比如对话场景的 KnowHow。此外我们一直紧密跟踪大模型的发展趋势，进行相关的技术更新和迭代，比如预训练模型如何在领域里做优化，为企业提供贴合实际应用场景的模型等。” 中关村技术副总裁张杰博士表示。

自 2018 年开始，预训练模型逐渐兴起，起初用的比较多的是判别式模型，例如 BERT 模型。近几年，预训练模型几乎是以爆发式的速度增长，参数规模逐年上涨。尤其是去年 11 月底，ChatGPT 火爆出圈成为革命式的事件，基于 GPT 出色的生成效果，很多传统的 NLP 任务都规划到了生成模型中。从 BERT 到后来的 T5，再到 GPT4，张杰团队观察到，**整个技术发展的趋势，在向一个统一范式的方向发展。**

“大模型+领域知识”这一路线，核心是为了利用大模型的理解能力，将散落在企业内外部各类数据源中的事实知识和流程知识提取出来，然后再利用大模型的生成能力输出长文本或多轮对话。但是这个方向上一直没有一个创新力强的产品出来。ChatGPT 发布后，让张杰和他的团队眼前一亮。

中关村科金算法团队负责人于皓告诉 InfoQ，他们在多个业务场景进行实际测试，验证了在一些特定的场景，经过精心设计的领域 prompt，大模型的效果会显著提升，特别是在新领域的模型泛化性能力方面表现优异。例如，以前用判别式的模型解决意图识别问题需要做大量的人工标注工作，对新领域的业务解决能力非常弱，有了这类大模型以后，通过微调领域 prompt，利用大模型的上下文学习能力，就能很快地适

配到新领域的业务问题，其显著降低对数据标注的依赖和模型定制化成本。

“以前我们主要用 BERT 技术体系解决实际业务问题，ChatGPT 出来后，我们重新研究了 GPT 整个发展路线，梳理好整个大模型的技术发展脉络后，**从 AI 技术发展角度研判，生成式大模型可能是未来通向通用性 AI 的一条可行性路线，于是大胆地做了一个决定——在产品中，积极探索应用生成式大模型解决实际业务问题。**”

2022 年，在中关村科金 AI 平台能力中心的主导下，公司的智能外呼、智能客服、智能质检、智能陪练等产品通过自研的对话引擎全面拥抱大模型，充分挖掘企业各类对话场景数据价值，帮助企业实现更加智能的沟通、成本更低的运营维护。

从传统对话引擎转向大模型对话引擎

“通过新一代的得助对话引擎，**我们正在从传统的对话引擎迈向大模型的对话引擎**，用一套对话引擎支持多种业务系统，业务系统会基于行业线进行拆分，在不同的行业线还推出了私有化和 SaaS 化的版本。”中关村科金资深 AI 产品总监曹阳介绍，通过一套技术体系对产品进行能力赋能具有很多优势，以前产品矩阵的底层有几百个定制化模型，运维起来非常麻烦，现在可以统一用一套大模型就搞定了。

于皓介绍，**新一代得助对话引擎的核心能力是：领域 CoT + 领域大模型 + 领域知识库 + 领域能力套件**。通过将领域的 know-how 转化为领域 CoT，使大模型具有更复杂的业务问题解决能力；通过外挂知识库的形式，把事实性知识的流程性知识都放在领域知识中台内，大模型用来做抽取、调度和生成，然后下游业务系统通过 API 获取结果，以保证业务知识的实时性、可靠性；通过领域能力套件，打通大模型和企业已有系统的融合，建立模型和企业已有业务系统的无缝链接，将大模型的能力充分释放

到各个业务系统中。

得助对话引擎主要特点是以大模型为中心，传统对话引擎相关的意图识别、对话状态跟踪和话术生成等任务全部由大模型进行判别和自主执行，为了保证整个对话过程的可用、可信、无害和可靠，中关村科金自主研发了领域 prompt 工程组件，可以有效将对话能力约束到领域边界内，使大模型可以在业务规范下，安全、可靠地完成对话任务。

在新一代得助对话引擎的设计中，充分考虑到实际业务情况，**轻量化部署本地化大模型是未来企业的强烈需求**，团队研发了**本地化大模型的快速优化套件**，主要包括领域知识的注入能力、领域 prompt 生成能力、领域指令自主生成能力、领域指令微调能力和领域规范行为对齐能力，可以帮助企业快速构建适合于自身业务场景的大模型，降低大模型在企业的落地门槛。

于皓介绍，在得助对话引擎架构的设计过程中，也充分考虑了 ToB 场景的特性。在 ToB 场景中，企业有很多领域知识，但这种领域知识基本都固化在各个知识库里，如何结合大模型的隐性领域知识和推理能力，与企业已有的固化好的显性知识融合起来，这是目前需要解决的一个大问题。

因此在设计引擎架构时，于皓的算法团队把知识库和大模型的基础推理能力相融合，结合企业固有的知识满足实际业务场景需求。针对解耦合的模式设计了交互式的推理能力，例如针对问答场景，首先要理解问题语义，抽取出问题中涉及到的实体关系，如果问题属于隐形知识范畴，就由大模型直接回答。如果涉及到业务中的显性知识，其可能存在于结构化数据库或者文档库等形式的知识库中，大模型需要利用领域 CoT 和领域能力套件自主性生成查询语句，根据查询语句到知识库里查出相关知识，把知

识提炼出来之后，再根据上下文生成对话的形式，将一个复杂的问题做思维链推理，提炼成简单的问题，借助于领域能力组件可以从知识库、业务系统或者互联网等资源检索信息，再借助于大模型的上下文学习能力归纳总结出答案，在这一模式下，重点在于大模型解决问题时的推理合理性、过程可控性和结果的可靠性。

大模型如何在领域落地，打造超级员工

随着基础大模型的不断成熟，中关村科金 AI 能力平台中心不断拓展得助对话引擎的应用场景，推出虚拟员工助手，帮助企业打造“超级员工”，在营销文案生成、客服问答、坐席助手等场景，助力企业营销服价值提升。这些超级员工就像是企业里的“超人”。

“超级员工”形成的技术路径

在基础大模型能力加持下，得助对话引擎帮助企业构建“超级员工”，需要经历“学、教、用”三步形成路径。

第一步：学，大模型在领域数据上的无监督学习。

大模型就像是一个智商较高、理解能力很强、过目不忘的“文科生”。中关村科金 AI 平台能力中心在这个底子很好的“文科生”基础之上，注入企业的领域知识，如各种培训材料、行业通用知识等，让大模型能够理解领域知识，成为一个具备领域知识的“普通员工”。

这一步的重点是让小模型从散落在各处的数据源中萃取出领域事实性知识。

中关村科金高级算法工程师罗华刚介绍，做到这一步的关键是要“及时动手自制”。ChatGPT 用的数据都是开源的，没有规范的具体领域数据。而如果要应用到领域里，则首先要用自己的领域数据。比如中关村科金积累了多个行业大量的对话数据，把这些高质量对话数据灌到大模型里训练，就可以让模型更符合领域分布，那么，它生成答案就会带有领域的知识，这种是隐性的知识。

对于事实性的动态知识，模型是难以把控的。比如针对具体某个金融产品，客户会询问利率，正确答案是 5%，但是大模型生成的结果可能是 6%。大模型善于理解用户的意图，使生成的回答符合逻辑，但并不能保证事实性。另外，随着时间的推移，这款产品的利率可能会降低，变为 4%，大模型很难及时跟进此变化。

事实性的动态知识尽管也可以通过训练融入大模型，但无法保证输出的正确性。如果要保证正确性，就会让这个模型过度拟合，这不符合训练模型的目的。因此对于事实性的动态知识，非固有的领域性知识，罗华刚团队参考了 GPT-4 提供的插件功能，来保证大模型实时输出的正确性。“我们将它作为插件，领域知识库/中台作为它的事实性或动态知识的存储地，大模型负责对给出的问题做语义理解，同时发挥中枢调度的功能，最终生成答案。”

第二步：教，从人类反馈中以小规模有监督学习的方式做微调。

“普通员工”依托专业的产品设计，不断和人类专家进行闭环反馈。基于人类专家的反馈，它能够不断地获得提升，逐步成为“超级员工”。

这一步的技术难点是，如何让大模型学到流程性知识。因为与事实性知识相比，流

程性知识更强调连续性和逻辑性。

为了让大模型具备流程性知识，罗华刚介绍，可以采用两种方式。一是无指导学习的方式，让模型自己去体会对话中存在的知识，将对话数据以及对话数据产生的目标（比如，营销场景下坐席与客户的对话数据，最终是否完成了销售任务）设为标签，作为一个监督性的任务进行微调。

第二种是有指导学习的方式，告诉模型怎么做，这具体有两种方法，一是采用思维链的技术，加入逻辑引导，比如针对一小段对话，加入逻辑分析，告诉模型如何分析这通对话，分析顾客的特点、顾客表现出的意愿等，输出一个更优质的对话告诉模型，这样的例子是比较好的，让模型再去学习这样的案例，大模型通过学习一个评判模型来评判一个对话逻辑。另一种方法可以考虑采用强化学习或 GAN 的形式，比如，机器人与机器人之间产生了对话，再去评判这通对话做的怎么样，通过这样不断地来回学习，使模型的能力越来越强。

第三步：用，在特定场景下以机器人或助手的方式应用。

成为具备领域知识的“超级员工”后，企业可以给它分配特定的任务，在具体的场景下进行应用。比如撰写营销文案，自动解答用户的问题，或辅助坐席去回答一些问题等。

这一步的关键点是产品设计，如何合理的为人类员工和数字员工分配任务，实现能力互补，并且让数字员工从业绩反馈中持续学习。

曹阳介绍，为“超级员工”分配任务，目前是机器+人工结合的方式。基于业务属性，会按照一套通用的框架和流程进行分工，到具体环节会采用单元模式的动态人工介入

的方式。

这种方式的核心是单元化产品逻辑，所有的业务单元都会抽象到一个整体的单元，系统会按照固定流程执行，流程中涉及到人工审核或验证环节。人工参与的程度可以根据业务问题的复杂度进行动态调整，简单的业务问题系统会进行自动化处理，复杂的问题交由模型处理后，人工再进行审核。针对不同的业务，也会动态化的控制人工参与度，比如，售前营销沟通场景中业务类型比较重要，且用户画像相对丰富，这个环节会有较强的人工介入，其他场景比较简单，动态控制的要求也会相应地降低。

应用案例解析：AI 落地商业空间更大了

“超级员工”的目标是帮助企业降本增效。目前，中关村科金打造的“超级员工”已在各个场景展开试点。罗华刚向 InfoQ 介绍了两个代表性的应用案例。

外呼机器人，让话术师告别“刀耕火种”时代。

一组应用数据显示，以前在一个新场景构建外呼机器人，大概需要 22-23 周时间，且需要非常熟练的话术师才行。但现在，借助一个构造好的领域大模型，只需大约 1-2 天时间就可以成功交付，能明显降低交付成本，加快交付效率。

与传统的智能客服相比，大模型进一步降低了开发和运维成本。以前，各种场景都需要算法工程师标注数据以训练特定任务的模型，因此开发成本较高。现在，大模型本身的通用性好，不再需要很多算法工程师标数据，可以直接拿过来用，有时稍微标几条数据就够了。企业部署外呼机器人、客服系统的成本会大大降低。原有 30 个话术师的工作量，现在 2 人即可完成，而且语义理解准确度从 85%提升至 94% 。

营销文案助手，赋能理财师撰写营销文案，原先 10 分钟一条营销文案，现在 10 秒即可完成。

当下的财富管理行业面临业务增速较快，但理财师规模和人才增速不足的挑战。理财师的专业要求高，其中，文案生成就是一项刚需工作，如果技能不够，就容易流失很多高净值客户。因此，理财师亟需借助智能助手工具，将一些繁琐的工作，如编写营销话术工作等交给机器完成，这样就能释放出更多精力放在拓展新客户等工作上。

针对理财场景，中关村科金研发了营销文案助手。它发挥大模型的语言理解能力，将产品的介绍文档、行研报告、权威媒体的财经新闻、专家观点等“灌进”大模型，大模型从这些非结构化文档中，抽取核心观点及关键信息，如新基金产品的发布日期、期限费率、收费政策、风险等级、利好政策、行业趋势等。这些抽取出来的要素，形成了新的领域知识库。

当理财师选择某一客户时，客户的属性就能从 CRM（客户关系管理系统）中关联出来。根据这些客户特有的属性，理财师就能了解其投资偏好。接下来进入营销环节，根据所处的阶段，大模型可以生成相应的营销话术，为保证生成的内容是准确、可控的，理财师最后对生成的内容进行审核和再编辑。

透过实际的落地实践，中关村科金 AI 平台能力中心发现，**拥抱大模型也明显加速了 AI 商业化的进程。**

“在探索大模型实践的过程中，我们尝试了多个对话应用场景，也和客户共创了一些有场景代表性的试点项目。”张杰坦言，“使用大模型以后，已有的对话产品中定制化建模的成本降低了，而且之前技术达不到要求的对话场景现在也可以做数智化尝试，

AI 落地的商业空间更大了。”

原来的智能分析比较简单，基本是机器根据录音做转出分析，或者机器按照已设定好的流程一步步往下执行，但不考虑用户的反馈，对用户的意图理解并不到位，客户体验不好，很难实现成单；客服质检也只能做一些简单的操作，比如关键词匹配等，且匹配度也有一定的提升空间；原来针对各项对话场景分析的粒度不够细，准确度也不够高。用上大模型升级后，对话的理解能力和智能分析效果有显著提升，成单率也取得了极大地提升。

大模型重塑生产关系

未来“超级员工”在企业里所承担的角色，一方面可以完成一些机械、重复的工作，另一方面，可以辅助人工，承担一些创造性的工作，减少员工的工作量。大模型强大的意图理解能力以及泛化性使其完成一些创造性的任务成为可能，这将对企业的生产关系带来重要的变革。

“中关村科金的愿景是希望通过对话式 AI 技术，重塑企业的生产关系。现在尽管大模型十分火爆，但在企业服务赛道，很少有人意识到大模型未来会对企业的生产关系带来很大影响。”张杰表示。

现在的企业生产关系是一个树状的架构，从上往下分别是董事会、职能部门、业务部门，一层一层往下是金字塔式的。现阶段在数字化转型中，企业开始将一些简单的体力劳动、能总结出规律的活动，写成具体的程序，通过自动化校对的方式来实现。也有些企业会训练模型，这些模型会以助手的方式辅助一线员工，员工下面一层也就变成了助手机器人。但整体来看，整个组织结构仍然是树状的，是人-人-机的架构。

未来，情况可能会发生变化。

张杰表示，“未来大模型带来的启示是，它不但能够替代一些简单的体力劳动，还能替代一些简单的脑力劳动，甚至包括那些能够从日志里总结出经验的脑力劳动。”

在张杰看来，未来企业的组织结构将呈现纺锤形，上层是人类经营者，中间层真正负责干活的是机器人，少数的业务专家会指导机器或与机器协同互补，是人一机一人的架构。“机器人在其中的角色并不完全是助手。最开始由于技术所限，它以助手的形态呈现；未来在具备自主学习能力后，它能够真正成为独立承担工作的数字员工，而且是成本非常低的员工。”

未来，随着大模型重构企业组织架构、重塑企业生产关系，可以释放出更多的人力，开展更具创造性的工作。但不可避免地，重塑生产关系意味着必然有一些人会被替代掉。张杰认为，从短期来看，大模型带来的影响是，一些不产生价值的、中间的职能岗位，可能会很快将被机器取代掉。长期来看，关于价值判断、规则制定、以及关乎人性和心理的工作，是大模型不能取代的。

颠覆对话场景：下一阶段企业数字化转型的重点

对于企业来说，数字化是一项“必修课”。最近几年，可以明显感受到，数字化转型逐渐往“纵深化”发展。而大模型技术的爆发，有望给企业的数字化转型进程带来“加速度”。

一个显著的变化是，前几年，企业数字化主要针对企业内部的交易数据和核心业务系统，对这些数据通过数据挖掘的方法进行建模，实现降本增效。

随着近些年大模型的高速发展，对话数据成为企业愈发重视的数据资源。无论在现阶段还是未来，无论是企业与外部客户沟通还是企业内部员工的培训和协作，对话都一直是最主要、最自然的交互形式。在这期间，会产生很多对话数据，包括线下营销和线上营销、文字沟通和电话沟通等场景。

企业希望充分利用对话数据、挖掘对话数据的价值，从而更好地服务于数字化的需求。这也是对话式 AI 技术解决方案提供商当下正在思考的问题。

采访中，张杰谈到了他的一个判断：“**对话数据，将是企业数字化转型下一阶段的重点。**过去，企业的**数据只是存了下来，并没有进行结构化的表示和挖掘，更遑论提取出智能服务。**随着大模型的到来，可以理解这些非结构化数据中蕴含的语义，进而挖掘出其中的智能服务。”

张杰表示，大模型能解决对话场景下数字化转型中存在的“最后一公里”的问题。以销售话术复盘场景为例，很多企业都在针对其做数字化转型，此前大都通过规整和挖掘订单、客户标签等数据的方式进行。但往往在“最后一公里”的时候，无法实现特别好的转化效果。

“最后一公里”是指，业务人员与客户在门店、连锁店、呼叫中心等线下和线上销售的场景交互时，通过对话的方式进行，这一环节没法做数字化转型；“最后一公里”的分析和挖掘也很难做到位，比如传统的客户呼叫中心在进行电话营销时，原先只能做简单的关键词质检，无法做更细粒度的分析。

大模型具备的超强语言理解能力，让“最后一公里”的销售过程实现数字化转型成为可能。具体的做法可以是，通过电子工牌或呼叫中心将销售过程录下来，采用 ASR 语音转写技术将录音转成文本；再通过对话文本挖掘出用户的意图；随着对话过程不断进行，大模型可以实时生成流程图谱，给销售提供对话建议，分析潜在的话题引导方向，提升销售人员的营销技能，提高成单概率和用户的留存率。由此，既能帮助企业通过智能对话服务实现降本增效，也能有效提升用户体验以及拓展服务外延。

在此前很长一段时间里，对话业务在对话式 AI 厂商的语境中基本指“客服”。曹阳表示，造成这一局面的原因主要有两个，一是客服在售中、售后等环节的业务较规范、标准，可在固定框架内让机器人回答相对固定的问题，实现对效率的追求和用户体验的平衡；另一个原因是技术的局限性。受到技术所限，客服基本围绕售中、售后环节服务，如果要实现从售中、售后向售前扩展则面临技术挑战。

得益于大模型带来的技术变革，对话可做的业务范围会得到极大地扩展，如从售中、售后向售前扩展。与此同时，售中、售后环节能否带来新的营收增长点等探索也会增加。此外，与以往集中在客服场景做数字化转型不同，现在很多企业希望在整个生产流程中所有涉及到对话的场景都进行数字化转型，包括人和人之间工作的协作、员工培训、私人社交等领域，未来将在企业和个人层面诞生更多的应用。

对话式 AI 及大模型发展技术趋势展望

对话式 AI 下一步重点是流程性问题

张杰认为，对话式 AI 下一步要攻克的一个技术难题是流程性的问题。对话有流程步骤，如何让大模型，基于一个特定的目的，探索出最佳的实践流程，这很关键。

目前，中关村科金 AI 平台能力中心正在对这一问题进行攻坚。让大模型不断从历史对话中总结话术流程，通过不断地总结完善，使其生成的话术流程更有针对性，可以应对不同客户的诉求。这样模型准确度更高，自动化程度更高，智能化程度也会更高。

程序员的归宿不是提示工程

大模型与之前的预训练模型的不同之处之一就是提示学习（Prompt Learning）。预训练模型需要微调，大模型往往需要提示。

提示学习是 2020 年出现的新概念，主要是为了解决预训练语言模型训练过程的任务和实际业务的任务之间不一致的问题。通过提示语，可以让预训练语言模型理解当前任务的类型，从而可以更好地完成任务。随着 NLP 技术的飞速发展，现在的提示工程已变得更为复杂，提示语通常包含任务指令、任务目标、行为约束、输出规范、资源清单、样例展示和思维能力提示等要素。

于皓的团队将提示学习作为大模型工程化方向上的研究重点，并在多个场景测试效果。团队根据不同的场景设计了自动化封装的 Prompt 工程的方法，一条思路是离散的提

示工程，另一种是连续的提示工程，比如用一个模型把对话自动生成向量，隐含提示工程，然后将提示向量融合到大模型向量中解决问题。

现在，给每个任务找到合适的提示语还是一个很大的挑战。测试中，于皓团队发现，不同的任务需要差异化的 Prompt 模版，从指令设计、样例选择、样例的顺序以及推理过程等细节进行 prompt 的优化微调，每一个环节都可能影响到 Prompt 在实际应用的场景效果。测试结果显示，在意图识别上，不同的 Prompt 的准确率能达到 2%~80% 的巨大差距。

关于提示学习的另一焦点话题在于，未来的提示学习工程师可能会比软件工程师多。对此，于皓认为，“Prompt 工程的确在现阶段非常重要，但至于说未来是不是程序员都成为提示工程师，我的观点是，Prompt 工程可能是暂时的一个中间过程。只是说，现在大模型的能力还没有达到基于人工设定的复杂任务目标去自主性进行任务分解，然后根据这些任务转化成一种它可以直接解决的细粒度的自然语言任务。现在大模型需要中间的提示工程师帮助它理解任务，然后转化成它可以直接执行的自然语言任务，这中间是一个适配的过程”。

未来随着大模型的能力向更高层级提升，会覆盖掉现有的 Prompt 工程。因此，于皓认为，**程序员的归宿不是提示工程，提示工程一定会被大模型的能力覆盖。未来大模型一定具备很强的交互能力，甚至实现人人都是“陪练师”。**每个人在日常工作中可能都会有大模型与之交互，在交互的过程中，大模型会不断提升其对领域的认知能力，增强大模型专业能力，逐渐成为 7*24 小时的“超级员工”。

算力难题下，如何选择适合自己的模型

大模型迎来爆发后，很多企业都在争相上车，但一个现实的问题是，大模型背后所需的算力成本极为昂贵。

以 ChatGPT 为例，其训练约需要万卡时的计算量。对于大多数的企业来说，做到这一点不太现实，全球可能达到这一量级的企业都极少。此外，还面临很多需要处理的技术难题，例如，数据质量差，训练过程中模型缩小导致最终训练结果不及预期等问题。

因此，罗华刚认为，**当大多数企业很难付出像 ChatGPT 这样的大模型所需的成本时，就需要考虑如何将模型调小以满足需求。**模型的规模会随着应用场景的复杂程度及数量发生改变，模型越大，提供的能力会越强；而领域越小，它需要的模型规模越小。因此，在企业自有资源允许的条件下，建议选择尽可能大的模型，使得模型的能力更强。在这种情况下，企业可以考虑用一些方法来降低训练成本，进一步细分到具体的任务场景下，采用比如 Self-Instruction 或 LLM+LoRA 技术。

Self-Instruction 是通过大模型的输入、输出来微调模型，比如，有人训练斯坦福的羊驼模型大概只用了 500 美元的成本，通过调用 ChatGPT 的接口生成一系列的 instruction，微调指令和输入、输出，最后自己的模型只有 70 亿参数，相对而言成本降低了 20 倍，用这样一个模型去拟合，最终它的效果可以接近 ChatGPT 的效果。

LoRA 是指在大模型插入一些小模型，微调时，大模型不动，只微调小模型的部分，也可以达到同样的效果。这种方案牺牲了模型的整体能力，提高了在特定任务上的能力，但这样做能够降低训练成本。

如果企业没有自己的大模型，也没有自己的领域大模型，可以考虑部署开源模型，这样也可以节省算力消耗的资源。

未来：领域模型成王者？

大模型混战之际，业界也在讨论和预测大模型的终局。张杰的判断是，未来会呈现出**基础大模型多家并存、领域模型百花齐放**的状态。

“基础大模型，特别是多模态基础大模型，未来应该只有几家公司做，因为做基础大模型需要大量的数据、算力和人才，这些组合资源极少部分企业能够承担得起。因此，未来一定是有数据、有算力、有人才的公司，更可能去构建出基础大模型。基础大模型未来会聚焦在提升多模态能力、挖掘复杂推理能力，以及构建应用生态圈。”

基础大模型如果想用在实际业务中，还有很多方向需要适配，例如在法律、医疗、金融、政务等领域，很多工作流程逻辑复杂，且对数据敏感性、业务可解释性要求高，基础大模型在这些场景无法直接商用。

这就给未来其他企业留下了空间。如何根据实际的业务，将大模型转化成一种具有高效的计算方式的小模型，小模型再根据专业知识做注入、指令微调、思维链提升、对齐等，使其更适配某一领域的规范约束。

作为对话式 AI 技术解决方案提供商，中关村科金需要思考的是，**如何发挥自身优势，在领域内如何积累数据，如何沉淀领域知识，如何将领域知识注入到大模型上，以此构建自己的技术护城河**。此外，在具体应用场景下，思考围绕对话和推理两种能力颠覆已有的产品体验，挖掘新的应用场景。

“可以畅想，未来大模型领域会是一个百花齐放的局面。个别头部企业会去做大模型，其他企业根据大模型的能力提升每个领域的中小模型，中小模型再结合领域的知识，变得更专业化，这是一个大趋势。”张杰如是展望。

采访嘉宾介绍

张杰，中关村科金技术副总裁，天津大学计算机专业博士。先后就职于华为诺亚方舟实验室、阳光保险、明略科技。在知识工程、自然语言处理等技术领域拥有丰富的理论和实践经验，出版技术专著两部，发表学术论文十余篇，发明专利一百余项，主持或参与国家级课题八项，获第十届吴文俊人工智能技术发明一等奖。主持开发了推荐引擎、知识问答系统、客服机器人、大数据风控系统、行业知识图谱等多项商业系统，累计销售额数亿元。

于皓，中科金算法专家，同济大学计算应用技术博士，先后参与机器学习、知识图谱和大模型领域相关项目数十项，具有丰富的项目实战经验；申请国内外技术发明专利 20 多项，获得第十届吴文俊人工智能技术发明一等奖、2019 年上海市科技发明一等奖、第四届中国保险业年度最佳突破奖等数十项奖项。

罗华刚，中关村科金高级算法工程师，北京大学计算数学专业硕士，研究方向为运筹优化、知识图谱、自然语言处理等。曾参与建设 HAO 图谱、知识即服务、图谱 CBB 等多项系统，撰写发明专利三十余篇，协助撰写专著《知识中台》，获 2020 年吴文俊人工智能技术发明一等奖、世界人工智能大会卓越人工智能引领者奖 Top30。

曹阳，中关村科金资深 AI 产品总监，拥有超过 10 年的 ToB 产品经验，曾任职于阿里、京东、字节跳动、shopee 等公司，主导多个智能客服产品，对 NLP、智能客服、CRM 相关的技术、产品应用、商业化有着丰富经验。

中国的“贝尔实验室”：我们的数据库从自己的第一行代码写起

作者：褚杏娟

说起做数据库，没人会觉得这是一件能够随便成功的事情。

1985，此前忙于推广 Ingres 商业化的 Michael Stonebraker 重返学术界，想要解决当时数据库存在的问题。到了 1988 年，Michael 所在的项目组才实现并运行了第一个 Demo 版本，次年才发布了 1.0 版本。

不过，这个项目优化到 4.0 版本后就被停掉了。1994 年，加利福尼亚大学伯克利分校研究生 Andrew Yu 和 Jolly Chen 用增加的一个 SQL 语言解释器替代了早先基于 Ingres 的 QUEL 系统，并创建了 Postgres95。1996 年，Postgres95 被重命名为 PostgreSQL。

简单的文字背后是两代人努力了近十年才有了雏形，此后又是二十多年的不断优化和改进。

在国产数据库热潮下，很多进入这个赛道的企业会选择基于开源做数据库，这是可以理解的：创业者从零开始研发不仅要经历不断试错、改进带来的更长产品周期，而且未来能否被市场接纳并盈利更是未知，期间的风险不言而喻。

不过，也有一些做数据库研发的人选择从第一行代码开始去亲手构建完整的数据库，深圳计算机科学研究院（以下简称深算院）的 YashanDB 便是其中之一。他们从零开

始，选择将多年的学术研究成果转化为工程产品，然后再投入到市场中检验和优化，让企业可以用另一套方式运行数据库。本期《中国卓越技术团队访谈录》，我们有幸邀请了 YashanDB 的研发团队来讲讲他们从无到有打造自研数据库的故事。

缘起

2018 年 11 月，深圳市人民政府批准建设“十大基础研究机构”，深算院便是其中之一。2019 年 4 月，深算院正式揭牌运营。当时，市面上缺乏一款从代码层面完全掌握主导权，并能与国际产品同台竞争的商业数据库。能不能做、可以做成什么样，成为深算院需要考虑清楚的问题。

经过调研和企业沟通后，深算院发现了一个问题：当时市面上的数据库普遍缺乏在关键业务场景下完全一比一平替甲骨文的能力，这成为深算院切入市场的好角度。然后，盘点了自身在数据库领域的积累后，深算院认为自己有能力做出这样的架构，甚至未来在某些场景可以超过 Oracle。如此，深算院便成立了专门的团队来做，这也是后来的 YashanDB 研发团队。

YashanDB 研发团队最开始不到十个人，但都是经验丰富的“老手”。像 YashanDB 研发总监欧伟杰有十年以上数据库内核设计与开发经验，YashanDB 解决方案架构师王义寅有着二十年行业从业经验。

当时，研究院的工作场地还没有落定，这些人就先在类似众创空间的场地里办公，两间办公室，一间也就容纳四五个人。“大家都是想把数据库当作毕生事业去经营的人。尽管办公室非常狭小，但丝毫不影响大家的奋斗热情，因为重要的是我们有一方天地

可以做自己感兴趣的事情。”欧伟杰回忆道。



YashanDB 研发团队初始成员，来源：YashanDB

现在，YashanDB 研发团队已经发展成为三百多人的大团队了。对数据库的情怀也延伸到了招人选择上，他们会更青睐看好数据库行业、又认为这值得长期投入的应聘者。

YashanDB 研发团队，或者说整个深算院，沿袭了贝尔实验室的模式：基础研究——技术开发——新产品生产——市场营销——信息反馈——产品改进。这与企业主导的研究团队和高校的研究团队存在本质上的不同。

企业主导的研究有一个很明显的特征，就是目标性很强。一个产品做出来后能否在短期内盈利决定了这个产品寿命的长短。这样的结果就是，企业没有足够的耐心做各种细节上的打磨。但是研究工作，特别是基础软件领域的研究，非常需要持续的技术积累，比如 Oracle 比较成功的 7.0 版本也是经过了十年时间的打磨才做出来的。

研发基础软件路阻且长，深算院做好了潜心研究以及更多耐心和时间打磨的准备。同时，深算院又设立了专门的基础研究团队，可以直接解决工业界工程实施过程中遇到的问题，并从中抽象出研究课题。

与高校研究相比，深算院拥有经验丰富的工程团队，可以把研究成果直接转化为工业级系统，这在大学很难实现。

那么，YashanDB 研发团队是如何将学术研究转化为商业产品呢？具体来说有三个阶段：学术论证、工程实现和市场验证。研发团队首先要论证学术课题的可行性，至少要做原型验证，然后再进行工程实现，之后再到市场中做具体场景的验证，否则产品能否支撑业务永远都会是一个未知数。

从学术理论到产品落地

纵观数据库五十年发展历程，从 E.F.Codd 提出数据关系模型、到 J.Gray 提出共享数据库的一致性和锁的粒度，再到 L.Lamport 提出 Lamport 逻辑时钟，数据库的发展一直是由原创理论方法驱动产业技术革新。

但数据库内核上的理论创新进展缓慢。YashanDB 解决方案架构师王义寅用了十年的

时间，从最初的实用数据库慢慢接近到数据库内核。但到了内核层面后，他突然觉得找不到方向了，因为很多理论还是三十年前的理论。“现在数据库行业需要的并不是应用创新，而是理论的革新。”王义寅说道。

深算院的基础研究部承担了理论创新的责任。一方面，基础研究部展开大数据领域的原始创新探索，专注在理论研究和突破上；另一方面，他们将原创创新成果带到实战场，希望用新技术实现弯道超车。

YashanDB 研发团队成立时，樊文飞院士的有界计算理论（bounded evaluation）和数据驱动的近似计算（data-driven approximation）理论已横扫计算机理论和系统大奖，这成为后来 YashanDB 研发团队的技术突破口。

樊文飞院士是国际上囊括了数据库理论与系统顶级会议最佳论文或时间检验奖的两位学者之一。有界计算理论是把大数据计算规约成小数据上的处理，近似计算则可在硬件规模投入有限的情况下，实现大数据精确高效查询。

樊文飞院士曾分享道，在数十亿条数据的实时查询场景下，91%的查询可以用有界计算来解决，并且 70%以上的查询效率可以提升 25 倍到 14 万倍，剩余 9%不具备有界计算条件的查询，可以通过数据驱动的近似计算理论来解决。

2019 年 4 月，YashanDB 研发团队的七八位工程师听了樊文飞院士的理论阐述，经过一番讨论后，锁定了有界理论工程落地的关键技术点。当时，YashanDB 研发团队考虑关系型数据库里的很多计算是低效的，即使增加硬件设备也达不到理想的提升效果。这与有界计算和近似计算的研究方向十分匹配。

因此，在樊文飞院士的带领下，一批青年科学家与工程专家聚集在 YashanDB 研发团

队，开启了自己新的一段数据库生涯。

原型验证

很多研究没办法落地的原因就是理论和产品之间存在鸿沟，没有办法落实到产品里。理论研究要进入生产环境必须要经过原型验证的考验。原型验证针对性更强，会选择优先验证某一方面的特性。2019 年年中，YashanDB 研发团队在第一时间对有界计算理论做了原型验证。

传统关系理论本身已经相当成熟，现有关系型数据库系统大都基于传统关系理论打造，而有界计算理论突破了传统关系。如何在现有理论框架之下，把有界计算理论融入到关系计算的模型中存在非常大的挑战。

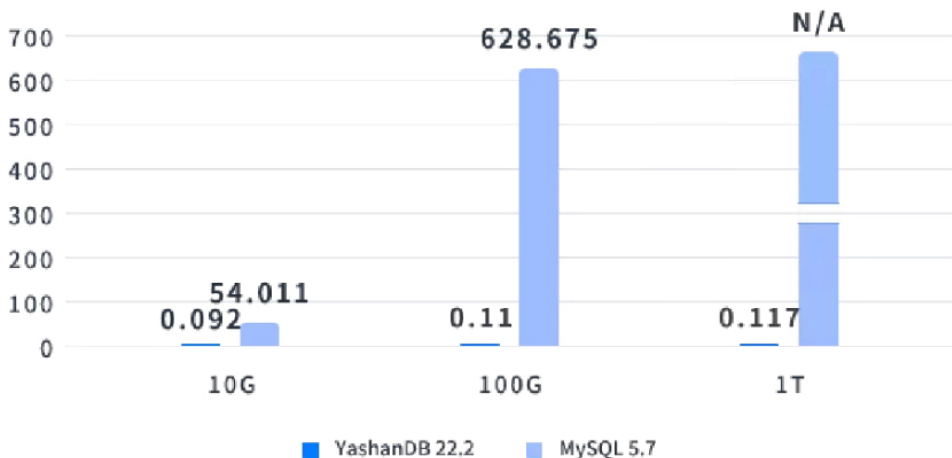
具体来说，一是如何与现有系统兼容，在不改变现有用户体验情况下，使用标准的 SQL 能力充分发挥有界理论的先进性；二是数据的实时变更，保证加速数据与源数据的一致性；三是如何让有界查询加速更好地服务于实际的业务场景。

为此，团队成员决定基于开源 PostgreSQL 从单机加速引擎开始验证。2019 年 9 月，BEAS（有界计算引擎）1.0 性能测试完成，结果让团队感到振奋：基于有界计算理论，实现在 AIRCA 数据集上将查询的响应时间平均缩短了 100 倍。2020 年 6 月，团队又将有界计算扩展至分布式系统中，到了 2.0 阶段，AC discovery 已经可以通过算法的方式实现数据语义的自动发现，代替用户操作，大大提高了效率。

在刚刚发布的 22.2 版本中，YashanDB 提供了有界计算能力，将大数据变小，实现在大数据分析时不需要访问全部数据，只需要取其中的小数据集就能得到想要的答案。

经实测，数据量从 10GB 增长到 1TB，YashanDB 响应时延维持亚秒级，性能提升千倍以上且未衰减，极大地节约了计算资源。

不同规模数据量下查询响应时间 (s)



硬件配置：2288虚拟机 (24核，88G内存，2T HDD)
软件版本：OS(CentOS 7)，DB(YashanDB 22.2、MySQL 5.7)
测试模型：AIRCA (美国航空运输局) 数据模型

“2019 年时，基础研究团队和产品研发团队前前后后讨论了将近两个多月，不断地发现问题、讨论解决问题，经过很多尝试后最终实现了它的验证。” 欧伟杰说道。

现在，研发团队和基础研究团队之间彼此支持，会定期交流，互通有无。比如研发团队在事务调度方面遇到性能瓶颈时，就会将问题提交给基础研究团队研究；基础研究团队也会将最新的研究成果同步研发团队，研发团队再做验证，如果结果得到学术界和研发团队认可，再继而转化成研发需求，成为产品能力的一部分。

因此，基础研究部做理论研究的方式大概有两种：一是该领域普遍面临的科学性课题，

二是将 YashanDB 运行后遇到的问题和挑战抽象成研究课题。

随着某一理论的成熟，基础理论团队的研究重点也会相应调整。比如把有界计算应用于图计算，去拓展它的应用范围。现在，团队开始主攻跨模计算方向，优化对多模数据的查询。

数据库要应对的场景非常多，一种理论方法不能只在某个场景有效。对于做学术研究的人来说，从数学层面去证明方案在各场景的有效性非常重要。从一定程度来说，数据库理论创新就是要用最严谨科学的方式，证明自身解决方案的适用范围。

工程实现

科研理论更多是在单点上实现最大化突破，而要做一个产品就得考虑方方面面的事情，包括易用性、可维护性等，维度更多、更复杂。如果说原型验证是一个点，那么工程实现就是一个面。

在花费了数月的时间做完原型验证后，团队在 2019 年下半年启动了全自研的系统开发，2022 年中正式发布 YashanDB 首个版本。

研发初期，YashanDB 研发团队首先就是做环境搭建，这是后来规模化协作平台的基础。得益于核心成员丰富的经验，研发团队用了不到一个月的时间就搭建完成了。

之后，研发团队将内核代码做了模块化划分，每个工程师会被分配到特定的模块后就专注于该模块的功能研发。做模块研发时，工程师先要梳理接口，搞清楚模块之间的交互方式，否则后面整个系统链条都会遇到问题。

“系统的实现节点有点像洋葱，它是一层一层的。我们肯定是从最核心、最内部、最基础的能力开始做，在这个基础上再不断地基于这些内核能力做特性增强。”欧伟杰表示。研发团队最开始的理念就是先用最小粒度的内核能力让系统跑起来。

“根据团队成员们的经验，在大多数场景下，一个数据库 80%以上的功能都不会被用到，反而可能只有 20%的功能是真正需要。因此，我们就优先解决关键矛盾，即先解决那 20%的功能，不常用功能的优先级相对较低，后续根据外部客户的需求再添加。”欧伟杰表示。

YashanDB 研发团队最先做了 SQL 引擎和存储引擎，SQL 引擎方面首要考虑其执行能力，存储引擎则更多考虑存储组织等。这些能力打通后，研发团队开始考虑做能力优化，比如在 SQL 引擎上增加有界计算支持来提高查询性能，存储引擎上考虑数据复制、事务处理等能力。

常规的开发模式是积累到一定规模后要先设计再验证，但 YashanDB 研发团队早期选择了开发迭代速度更快的方式，即在保持竞争力的前提下小步快跑，不断增强和调优。

“打个不恰当的比方，有点像在一路狂奔。团队给要增加的功能定下上线时间后，各模块的同学就完全沉浸在工作中了。”欧伟杰说道。这也促使早期 YashanDB 基本以周为单位进行迭代。

不一样的底层

YashanDB 与 Oracle 的表现形态接近，但底层基础完全不一样。用欧伟杰的话说就

是：车子的外形可能看起来差不多，但里边的发动机却有很大的差别。

在优化器方面，Oracle 的优化器经历了很长时间的打磨和不同场景的调优，毫无疑问是业内领先的水平，而刚刚起步的数据库很难找到所有场景，甚至很多场景都还没见过。YashanDB 研发团队的应对思路也是先从简单的着手，再逐渐衍生到复杂场景。

根据团队里从业多年 DBA 总结的经验，研发团队先把最常见、最基础的优化规则放到了自己的优化器里，这可以被认为是 RBO (Rule-Based Optimization, 基于规则的优化器)。在此基础上，团队做了完全自研的 CBO (Cost-Based Optimization, 基于代价的优化方式)，完成了第二阶段的工作。下一步，也是现在研发团队正在做的，就是基于机器学习能力积累各种场景应对能力，研发团队也希望可以借此实现弯道超车。

鉴于一旦使用开源产品，存储引擎的能力就会受制于开源，YashanDB 研发团队选择了全自研的方式。存储引擎本身是一个非常精巧、需要细致打磨的组件。在此方面，研发团队优先关注数据持久化和事务处理。

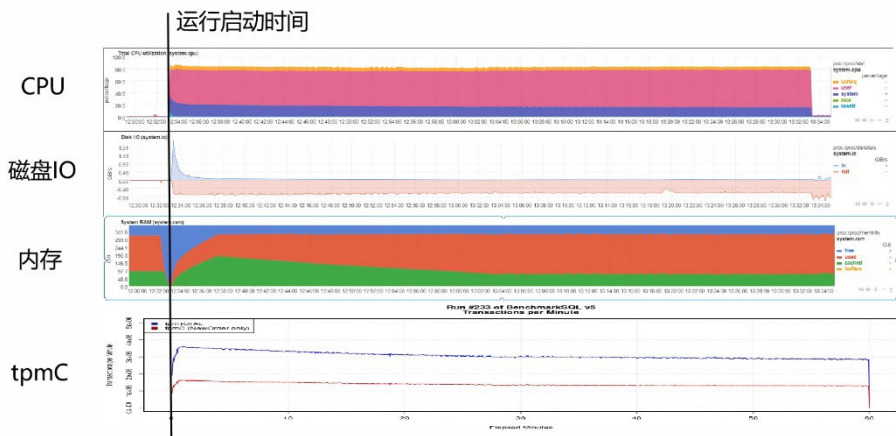
数据持久化与硬件资源紧密相关，如何最大化 IO 资源的使用、保证数据可以在最短时间内写到硬盘上是个挑战。事务则要保证无论怎么对数据增删查改，都要保证它的 ACID 特性（原子性、一致性、隔离性、持久性）不受影响。

对于事务处理，团队会更看重性价比。事务处理的粒度越小，并发性就越好，并发访问同一个事务单元的几率也越小。但是，如果事务处理的粒度非常细就会带来大量的管理成本。所以业界大部分的实现都在找事务并发粒度和管理成本的平衡点。

研发团队则利用了樊文飞院士提出的并发事务调度方式。当前，业界主要的事务处理

方式有 MVCC（多版本并发控制）、OCC（又名乐观锁）和 PCC（又名悲观锁）。实践中，大家更偏向 MVCC，但学术界多年来都在研究 OCC 并发控制思路。樊文飞院士则结合了 MVCC 和 OCC 的优势，使得在高并发场景下，系统不受核数改变的影响，而且整体成本可控。

面对大压力场景，数据库保持稳定的常用方法是让内存资源开销保持相对稳定，避免因资源大幅波动导致系统行为不可控。YashanDB 则结合了存储、事务处理能力，最大限度保证在线高通量场景下数据库的性能稳定。



在分析场景上，YashanDB 研发团队则采用了可更新的列式存储方式，对存储模型做了创新。传统数据库里，冷热数据管理可能是两个命题，挑战与解决方案等都不一样。YashanDB 改变了传统列式存储不支持更新和删除的特性，实现了同一个数据库中管理冷热数据，并且既支持实时数据分析，又支持海量冷数据的分析。

商业化摸索

对于 YashanDB 的定位，研发团队最开始瞄准的是新兴场景，主要聚焦在 OLAP 领域，商业目标与 Snowflake 类似。

到了 2019 年下半年，YashanDB 研发团队认为，基于开源改良/二次开发的产品并不是解决我国安全可控的可行技术路线，于是同步进入 OLTP 领域，目标是打造一个全自研的国产数据库，具备全面替代 Oracle 的能力，真正从核心应用层面进行国产替代。

“金融对数据操作正确性要求非常严格，这种场景是我们比较擅长的。”王义寅说道，“很多软件是在假设硬件非常可靠的情况下设计的，而我们设计的前提是硬件没有那么可靠。”

金融行业对高可用的要求非常高，而基础设施如存储、网络、服务器等本身存在故障失效和性能扰动。国产替代后，硬件扰动出现的概率有所增加，如果与软件适配性差，产生的业务影响很可能是致命的。YashanDB 研发团队给出的解决方案是增加一个主动监测机制，间隔特定时间自动检测一次并主动报错、修复等，包括硬件故障的主动修复。

对于企业来说，迁移面临最重要的问题是这个过程中是否存在显式和隐式的转换。如果存在大量的显式和隐式转换，则意味着需要做大量的适配，风险也会更高。

为使企业从 Oracle 迁移更加无缝，YashanDB 研发团队将存储的数据类型，比如字符串、日期、浮点类型等细节上都做了与 Oracle 精度完全一致，消除隐式转换风险。显式转换上，则在 SQL 函数、语法格式等方面与 Oracle 兼容。团队还提供了一系列

迁移工具，比如数据和应用校验工具等。

针对重点行业的核心系统国产化迁移，YashanDB 建立了一套完整的适配替代保障机制，完善故障预案、保障措施齐备，来确保操作充分验证且可回退，从而达到无感切换的目标。

具体来说，在业务上线前，YashanDB 通过充分调研数据库使用情况，对原有配套软硬件提前适配，并对关键 SQL 的执行计划进行性能评估，确保关键 SQL 的执行性能不低于替代系统。同时会进行关键负载场景下的业务长稳测试，保证系统环境的高可用。业务上线过程中，通过多种工具手段，YashanDB 会反复验证存在转换的操作，通过双平面长周期并跑，确保满足平滑切换条件。上线运行后，除了提供服务热线和提供驻场服务外，YashanDB 还提供” Bug 直通车”，凭借对数据库代码的完全掌握能力，即使触发软件错误也可以快速及时修复。

“我们会主动跟团队的人强调要从客户视角去思考问题。”王义寅说道，“技术团队容易说我的代码多好、技术多牛，但客户要的只是能否以最小成本获得最优的产品体验，这才是关键。”

“技术的本质体现在人上”

“没有技术积累做不出好产品，而技术的本质体现在人上。”欧伟杰说道。

YashanDB 研发团队能够快速发展的原因，除了 2019 年赶上了国产数据库发展的快车道外，还有就是现在进入数据库行业的人相比之前越来越多了。但事实上，国内数

数据库开发人才的缺口还是很大，真正能去做数据库内核的人没几个。

“为什么数据库这么少的人在做？因为大家都觉得这件事很难，都觉得这不是一件很容易成功的事情。”欧伟杰说道，“但是当你明明知道它很难，还愿意义无反顾去做的时候，必然是有某种激情或者是热爱来驱动自己。”在欧伟杰看来，兴趣难以持久，能让人长期坚持下去的是热爱。

“很多工程师，特别是一些专家不经过一定时间的积累是很难真正在这个领域有所建树的。国内外的很多大牛，年龄不要说 35，很多都是五字开头的，他们在随着数据库一起成长。”欧伟杰分享道。

但欧伟杰也坦诚，在招人方面，深算院的优势并不明显。当前，YashanDB 虽然已经商业化，但还需要继续深入。但欧伟杰表示，相对于很多企业要有经验的人，YashanDB 研发团队对零基础的同学会有更多耐心，通过进行相关培训来帮助新人度过陡峭的学习曲线。

面对越来越庞大的团队，YashanDB 研发团队的管理理念跟他们做数据库的思路很类似：化整为零，即把复杂的事情变成更小的粒度。团队以小组为基本单位，每组 5-7 人。“我们尽可能想让信息能在内部传递流畅，过深的层级可能导致大家只能看到自己眼前的一亩三分地。我们希望大家，尤其是校招的应届同学能够看到一个完整的体系，端到端地去思考一些问题，这对大家成长更有利。”欧伟杰说道。

结束语

在欧伟杰看来，一个行业要出现好的产品，至少需要两方面的因素：一是有一定程度的技术积累，二是被市场接纳。数据库是 To B 的生意，新产品就需要时间逐层突破。

未来国产数据库要做的就是真正将企业核心业务承载起来，这场竞赛中只有过硬的产品才能胜出。对于这个部分成员甚至已经有几十年数据库经验的团队，当前的重点是多活共享集群的打造，他们要直接对标 Oracle RAC，支持可扩展的、多活共享集群的能力，真正实现在核心业务上的高性能、可信赖。

“这对 YashanDB 来说也是一个里程碑，我们一定要打出漂亮的一仗！”欧伟杰说道。

嘉宾介绍

欧伟杰，武汉大学博士，深圳计算科学研究院 YashanDB 研发总监，10 年以上数据库内核设计与开发经验，曾负责分布式 NewSQL 数据库研发，多篇顶级会议论文及技术专利，熟悉 OLTP，HTAP 业务场景及前沿技术趋势，研发的产品服务全球数亿用户。

王义寅，YashanDB 解决方案架构师，毕业于南京邮电学院通信工程系。曾就职三大运营商、四大行、ICT 巨头，近二十年行业从业经验，主导产品有营账系统、分布式数据库、移动设备自研数据管理方案等，多项中美技术专利。

我们不是野心家，走出大厂创业是时代使然

作者：李冬梅

采访嘉宾：SelectDB 创始团队

Apache Doris 是一个面向实时分析的 MPP 分析型数据库，可提供亚秒级查询和高效的实时数据分析。2022 年 6 月 16 日，Apache Doris 从 Apache 软件基金会顺利毕业，正式成为顶级项目（TLP）。

2022 年年初，Apache Doris 的核心团队离开了大厂，共同成立了 SelectDB 公司，正式开启了 Apache Doris 的商业化之路。在商业产品上，SelectDB 推出了基于 Doris 开发的商业化产品，包括全托管的云上数仓服务和私有化部署的企业版软件两个版本。那么，几位创始成员为什么决定离开大厂出来创业？Apache Doris 发展路径发生了怎样的变化？SelectDB 产品的定位和技术侧重点与 Doris 社区版本有什么不同？近日，InfoQ 有幸采访到了 SelectDB 创始团队的多位核心成员，了解他们创业背后的故事、他们在 Doris 和 SelectDB 项目的技术实践与经验，以及这些工作背后的沉淀和思考。

离开大厂创业，是时代趋势使然

开源发展至今已经有 40 年历史，前 20 年是开源软件“野蛮”生长的粗犷式发展阶段，后 20 年进入到互联网时代后，不少新兴企业基于开源模式发展起来，开源软件

的商业化和生态趋于完善，开源软件逐步被广泛应用于生产中。

开源项目大多发端于大型互联网公司，这些互联网公司在快速的业务发展驱动下，研发了满足业务需求的技术并将这些技术开源出去。在这一阶段，项目的首要目的是满足自身公司业务的诉求。

云计算崛起后，在开源和云计算双重趋势的推动下，这些当初仅满足公司自身业务发展的开源项目迈入了产品化和普世化的进程中，甚至有许多项目走出大厂，独自成立了公司以商业化的形式继续运行下去。

以 Presto 项目为例，2022 年，从 Facebook 开源出来的 Presto 项目衍生出了 PrestoSQL，PrestoSQL 变为 Trino 后最终走向了商业化的道路，Starburst 就是 Trino 的商业化主体公司。

放眼全世界，近些年来全世界其他源自大厂的顶级开源项目都在走向独立创业的道路，比如 Kafka、ClickHouse 等等，这些趋势背后的核心推动力就是技术普惠时代的到来。

基于此，SelectDB 团队离开大厂创业显得不那么让人意外。

SelectDB 团队在接受采访时也表示，“这几年出来做 ToB 创业的企业越来越多，这不是单独某个人某个团队自己的选择问题，是一个新的时代在召唤大家。”

2000-2010 年是 PC 互联网时代，诞生了百度、阿里和腾讯等大企业。这个时代又鲜明地分成了两个阶段：即互联网基础设施的完善，和后面 WWW 的兴盛。互联网基础设施与 WWW 应用，互相促进。2010-2020 年，智能手机的出现推动了移动互联网时代的到来。移动互联网时代也分为两个阶段：即智能手机的普及和手机上大量

APP 的诞生。智能手机平台与智能手机应用，互相推动。

时至今日，我们已经走入了产业互联网时代。随着公有云基础设施的逐步完善，未来的十年，也将是云上各类软件服务的鼎盛时代，企业服务将会迎来巨大的发展空间，所以企业服务成为了很多企业花重金也要打下来的“战场”。

“以 TP 领域为例，源自大厂的项目可能更重视大规模扩展，所以系统实现更为复杂，但如若作为产品推向大众，会发现大多数企业可能仅需要单机就能满足，并且对部署和运维要求尽可能简化，那么面向大规模扩展这一特性就丧失了原有的优势。所以，自身定位一定需要转变。我们基于 Apache Doris 创业，是希望将 Doris 以及我们的商业化产品 SelectDB 朝着更加产品化、更加通用化的方向努力，例如易用性的改进、与上下游产品有更好的贯通、与云平台的结合、开箱即用的跨集群复制和备份恢复能力等，这些都是除核心特性以外需要重点优化的方向。我们不想错过这样一个充满机遇的时代。” SelectDB 团队如是说。

创业中遇到了哪些问题

创业第一道坎：组建一支强有力的团队

创业维艰，这条路并不好走。解决产品定位、招聘优秀人才都是创业初期较为棘手的问题。SelectDB 团队面临的第一个挑战就是如何组建高效执行的团队。相较于大厂自带光环，创业公司在人才招聘方面会显得尤为艰难。

幸运的是，得益于在开源社区过去的投入和对开源技术的坚持，SelectDB 团队很快就跨过了招聘这道坎。“开放、包容、合作的文化氛围对于吸引优秀人才具有重要的意义，因此很多热爱开源的技术工程师在获知我们招聘的第一时间就主动联系到我们希望加入”。

据 SelectDB 团队介绍，“在团队内部，习惯了开源协作模式的我们坚信顺畅的信息流通可以帮助团队获得更高的执行力，因此所有研发工作都采取了以 GitHub 为基础的协同开发模式。另外也设置了一系列沟通机制，进一步清晰团队职责、帮助每个人更清楚各自的分工、在目标执行时有更明确的方向，降低沟通成本的同时也保证了更高的协作效率，也有助于团队新人的融入”。

目前，SelectDB 团队已发展至 130 多人，其中 80% 为技术人员，包括研发、运维、质量、解决方案/技术支持，在开源和商业产品上的人力基本上是各占一半。

创业第二道坎：产品定位

人有了，接下来就是做产品。对于团队来说，他们要面临的挑战就是视角和思维方式的转变——从工程技术视角转向产品视角，研发产品时要更多地要去了解用户和客户想法，从场景驱动产品、产品驱动研发。

推动开源技术创新、繁荣开源生态、打造云原生时代实时数据分析领域的标杆产品，是团队的首要定位。其实在成立初期，SelectDB 创始团队已经将这一定位想得非常清楚了，接下来要做的就是如何推行和落实下去。

创业第三道坎：社区认同

此外，像所有从大厂走出来的开源项目一样，SelectDB 团队也面临着开源社区身份认同的问题。Apache Doris 作为最早从百度孵化出来的项目，如今由 SelectDB 团队进行商业化。无论是内部或者外部都会被问到为何投入如此大的人力物力来建设 Apache Doris 社区，而不是研发自己的闭源产品或开出新的分支。

面对这样的声音，SelectDB 团队解释称，正是因为当前时代具备无限可能性，使 Apache Doris 可能成为一个伟大的开源项目，使我们有机会打造出伟大的产品，因此也赋予了我们成立这家公司的意义。因此我们更应该把握时机，团结齐心，将这种可能性变成确定性，所以也更应该不遗余力地建设 Apache Doris 社区。

而在问到如何与市面上其他同类产品如何共存时，SelectDB 团队提到，“正是因为 Apache 项目的中立性、只会从属于 Apache 基金会，所有任何人都可以投身到社区的贡献中来。我们不是社区的拥有者，而是社区的推动者和贡献力量之一。同时市场上也有几家云厂商推出了基于 Apache Doris 的企业级数仓产品，他们与我们并非竞争关系、而是共赢关系，这种多样性也保证社区得以更加健康的态势发展下去”。

开源商业模式是伪命题吗？

在基础软件蓬勃发展的今天，开源正在吞噬着一切。但一直以来，围绕着开源展开的话题永远逃不掉开源商业化。开源商业模式到底是不是一个伪命题？开源项目想要获得商业价值，要具备哪些条件？

在 SelectDB 团队看来，开源商业模式并不是一个伪命题。「开源」本质是一种软件研发模式、通过源代码共享协作实现技术迭代，「商业模式」则是企业满足市场需求、实现收入或盈利的商业逻辑，如果只强调开源、不去思考如何获得商业收入，这样的企业最终可能深陷泥潭，开源与商业化的平衡与共存是每个以开源技术为基础的公司都必须要考虑的。

因此，SelectDB 团队更倾向于将「开源商业模式」定义为「企业以开源软件为基础来构建产品或服务、以实现收入或盈利的商业行为」。在这一定义之下，过去我们常提到的 Open Core、SaaS/云托管、订阅服务、License 授权等一系列开源商业模式都说得通了，无论是 MongoDB、Suse、Elastic、HashiCorp，所销售的产品或服务都是基于开源软件构建、都是为最终实现商业转化和收入目标而服务。

开源与商业化需要找到一个良性并存的方式，才能将开源推向另一个高度。在经历过长时间深入探索后，现在我们笃定这一良性共存的最佳方式就是将开源和云相结合，这也将是未来数据库演进的主要趋势。

一个开源项目想要获得良好的商业收益，一定要具备以下三个关键点：

- 被广泛认可的产品价值
- 繁荣、自治、良性发展的社区生态
- 开源与商业化的平衡与共存

就目前而言，Apache Doris 在前两关键点上的表现还是可圈可点的，而要想达到开源与商业化的平衡与共存，则需要更长的时间才能得到证明。

打造新一代实时数仓

在整个社会数字化转型加速的背景下，各行各业对大数据实时处理和应用的需求正快速增长，对分析时效性的要求也越来越高。如果用一句话来概括目前数据处理的痛点，那就是数字化时代不断涌现的数据分析诉求与降本增效的行业趋势之间的矛盾。

过去，当用户有查询报表需求时，一个 MySQL 或者 Oracle 就能轻松解决。但是随着时代与技术的发展，数据量呈指数增长、数据类型更多样化、数据场景更加细分，就需要引入 Hadoop、Spark、Elasticsearch、Presto、Clickhouse、Druid、HBase 等多个大数据技术栈来解决特定业务面临的问题，包括 Ad-hoc、联邦查询、固定报表、标签画像以及日志分析等。

多个技术系统的复杂性带来了繁重的开发维护成本，技术栈的简化与统一成为人们所追求的方向，而云计算技术的发展，为实现更高性价比和极简使用体验带来可能。

正由此，SelectDB 团队希望贡献自己的技术经验和工程力量解决行业痛点、构建云原生时代具行业普适能力的实时数据仓库。通过数据分析技术革新与云计算技术的结合也就成为 SelectDB 产品的突破口。

在产品设计上，SelectDB 提供两个版本，一个是全托管的云服务版本（SelectDB Cloud），一个是可以私有化部署的企业版（SelectDB Enterprise）。

那么，这两个版本之间有何区别？

据 SelectDB 团队介绍，SelectDB Cloud 面向的是有上云需求的企业。SelectDB Cloud 可以运行在国内和国外主流的公有云上，并且在多个云上有一致的使用体验。

多云一致并且体验一致是其区别云厂商数仓服务的一大特色。

SelectDB Cloud 对 Doris 进行了大量重构以便利用云的强大能力，提供更大弹性。存储与计算的分离，可以让存储与计算独立扩缩容；多计算集群的支持，可以在共享一份数据的基础上，可以提供物理隔离的多个计算集群；并且每一个计算集群都可以进行自动扩缩容，当前 SelectDB Cloud 支持手动和按时间设定的平滑扩缩容能力。

SelectDB Cloud 也针对数据库管理员运维提供了可视化的管理控制台，简化运维工作。

SelectDB Enterprise 版本则可以服务于希望私有化部署 SelectDB 的企业。

SelectDB Enterprise 版主要提供一个长周期支持的、稳定的 Doris 内核。开源的 Apache Doris 内核迭代比较快，新功能不断合入，企业客户在不断体验新功能的同时，也会担忧投入生产后的稳定性问题。所以，SelectDB 基于开源 Doris 提供了一个企业级的稳定内核，会在广大开源用户使用的问题反馈基础上、经过 SelectDB 专职测试团队测试和调优，并且 SelectDB 为每个稳定内核提供长达 12-36 个月的长周期持续维护，免除企业升级带来风险的担忧。这个内核完全可以与开源 Doris 内核互相兼容，企业随时可以从两个内核互相切换，不用担心被锁定到 SelectDB 的企业内核上。

同时，SelectDB Enterprise 版也会提供可视化的 Manager 功能。数据库管理员可以利用 Manager 管理多个集群，完成部署、升级、重启和配置等功能，同时可以诊断、监控和报警等。SelectDB Enterprise 版，也会提供跨集群复制和备份恢复等企业级功能。

无论是 SelectDB Cloud 还是 SelectDB Enterprise，在最初产品设计时 SelectDB 团

队都希望将这些产品打造成具有普适能力的数据库产品，至少要能够覆盖 80%的用户需求。

SelectDB 团队希望仅通过一个系统解决绝大部分问题，降低复杂技术栈带来的开发、运维和使用成本，最大化提升生产力。

新一代数据仓库的一些挑战

One Size Fits All，行得通吗？

但据墨天轮数据社区发布的《2022 年中国数据库行业年度分析报告》显示，截至目前国内共有 249 款数据库产品，单 2022 年就新增了 55 款产品，占比总数量的五分之一。

数据库市场百花齐放，各细分领域垂直产品经过不断打磨，已经可以在金融、制造等场景下独当一面了，想要通过一款通用的、普适化的产品链接到各种应用场景里，并不是件容易事。在软件行业，追求 One size fits all 的产品似乎都最终失败了。

对此 SelectDB 团队表示：

做 One size fits all 的产品，一定要加上一个大前提——符合高内聚低耦合的原则。也就是说，但凡是应该要高内聚的能力，我们需要做到 One size fits all；而那些低耦合的、不应该内聚的，即使有技术有能力做到一起，也不应该去做。很多 One size fits all 失败的原因都是违背了这个原则。比如 HTAP，大家觉得应该把 TP 和

AP 做到一个系统，也可能通过努力实现这一目标，但是 TP 和 AP 在很多企业没有实现一体化，有很多不是技术的原因。所以在现阶段 HTAP 我们更认可作为一个一体化的方案去推动更好，而不是试着去做一个单一的 HTAP 软件。比如，可以使用 MySQL 来做 TP，那么可以用 Doris 来做 AP，通过 Doris 自带的可以从 MySQL 秒级实时同步数据的能力，可以打造一个基于 MySQL+Doris 更好的 HTAP 一体化解决方案，这也是最近 AWS 在 Aurora 和 Redshift 推行 Zero-ETL 的原因。

此外，SelectDB 团队也强调，在考虑实现产品普适化能力的同时，一般而言不会以牺牲性能为代价，原因有三：一，良好的实现方案在设计之初就会最大程度去避免系统性能受影响，同时配合流水线上的性能测试保证不会有新功能的合入带来的性能回退；二，即使会对某些性能造成影响，也会对普适能力的提升和所影响的范畴之间进行权衡，看成本和收益比是否合理；三，数据库性能的优化是一个无止境的工作，影响性能的每个环节都有可能持续的提升空间。

也就说，在所有产品能力中，普适性能力并非是更高优先级，而是与系统的性能、稳定性以及易用性等多方面处于并列的位置，生产级别的数据库不能只考虑某一方面而忽视另一方面，用户的抉择也从来都是一个全面的思考过程。

从互联网走向传统企业，如何克服兼容性问题

自项目诞生之日，初创团队对于 Doris 的定位就是新式数仓。但多年之后，随着数据处理方式的演进，Doris 有了新目标——成为一个极具兼容性、全面拥抱云原生的实时数据仓库。

最初，Doris 来自互联网公司，在推广到传统数仓领域时，在 SQL 兼容性上遇到了一

些问题。离开大厂后，SelectDB 团队也针对这一问题进行了优化，目前已经有大量传统企业开始使用 Apache Doris 来替换以 PG 协议为主的传统数仓。由于 Doris 采用了 MySQL 兼容协议，从实际使用来看，很多企业也愿意向 Doris 的 SQL 来转移，毕竟 MySQL 的协议大家更为熟悉。在改写过程中，实际需要改动的也并不多。

但并不是说 Doris 与所有的传统数仓都能完全兼容。对于一些已经有大量的业务在传统数仓中、且复杂 SQL 迁移难度大的企业，建议采用逐步替换的思路：先迁移部分业务上来，另外再尽量提供一些工具帮忙自动改写，并针对其中难以改写的 SQL 在语法层做一些兼容。

走向云原生主要做了哪些工作

作为一款分布式的 MPP 数据库，Apache Doris 最初设计是部署在 IDC 物理机上。当将这个原生分布式的数据库迁移到云上并实现云原生，势必需要进行一次大的重构，以便深度利用云的能力。

首先，要完成存储与计算的分离。存储要全部放到对象存储上，而对象存储的性能又不满足实时数仓计算需要的性能，就需要采用计算节点的本地存储作为热数据缓存，如何设计缓存策略对系统性能有着重要影响。

其次，存储与计算分离后，用户可以针对一份数据同时启动多个计算集群，并且多个计算集群可以用来做多活，可以用来做负载隔离。SelectDB Cloud 提供了多计算集群可以共享同一份数据的能力，并且多计算集群都可以做到多写，而不仅仅是一个写多个读的模式。

SelectDB 团队称，“作为一个全托管的云数仓服务，如何保证客户的数据安全，并且方便的与客户在云上的 VPC 进行网络贯通，这些都需要重点去研发”。

未来规划和新方向展望

解决完兼容性和上云问题后，Doris 的下一步规划是怎样的？后续 SelectDB 团队在开源社区和公司层面的首要工作分别是什么？

SelectDB 团队表示，后续在 Doris 的工作重点就是核心功能 Feature 的研发，主要集中在高性能、混合工作负载、半结构化数据分析、数据湖分析、实时性与稳定性提升等方向上。

从具体功能上来看，包括支持更复杂 SQL 并具备全查询场景自适应优化的查询优化器、单节点数万 QPS 超高并发承载量的点查询优化、更灵活执行调度的 Pipeline 执行引擎、基于倒排索引的全文检索能力以及更高效的文本分析算法、根据写入数据自适应 Schema 的动态表等重要 Feature 都有 SelectDB 团队的工程师参与贡献。

除功能以外，对社区用户的技术支持和社区运营推广也是 SelectDB 团队投入的另一大方向。

目前，SelectDB 团队成立了一支专门的技术支持团队，在过去一年多的时间里为大量的开源用户提供免费的技术支持，沉淀了上百万字的内容知识库，后续这些内容也将逐步将开放在 SelectDB 开源论坛中。

而在公司向，SelectDB 团队的工作主要分三块：增强数据库的实时性、完善数据湖

分析、更多企业级特性的开发。

在增强实时性上，主要是优化实时数据导入、继续加强主键存储模型的实时 CRUD 能力，以便能够承载更高频的数据导入能力，提升数据可见性。

在完善数据湖分析上，虽然当前 SelectDB 已经可以直接查询 Hudi、Iceberg 和 Hive 等数据湖，但一些性能仍有待优化和提升。

针对很多企业需求比较大的企业级特性，尤其是跨集群复制（CCR）和更加完善的备份恢复，也是 SelectDB 团队下一步的重点工作。

技术管理漫谈

可悲的现实，大部分技术领导者可能并不称职

作者：许晓斌

编辑：孙瑞瑞

本文由 InfoQ 整理自阿里巴巴资深工程师 许晓斌 在 QCon 全球软件开发大会（北京站）2022 上的演讲《技术领导力实战》。

大家好，我是许晓斌，目前就职于阿里巴巴技术风险与效能部，负责运维与构建基础设施平台。在软件行业从业 15 年，包括微服务架构、DevOps、云原生等领域，软件管理工作 5 年。出过一本书《Maven 实战》，做 Java 的同学应该有不少人读过。

目前我在阿里巴巴带了多年团队，在实际工作中也有一些管理上的经验，但在准备 QCon 全球软件开发大会北京站的这个演讲主题时，在官网信息及材料中刻意隐去了自己的 title，我认为如果大家只是被 title 吸引而来，那其实并没有什么意义。个人认为，这个话题在国内也很少能有非常专业的分享。因为管理和领导是一种“专业”，它跟技术专业不一样，但它也是一种专业。众所周知，当下国内整体上是一个“业务为王”的时代，只要业务增长，管理、领导，甚至技术，做得好与不好都不重要。很多时候业务好和人的技术无关，和管理也无关，只是因为你恰好在风口，所以大家得意识到这一点。假设业务不增长了，该如何去量化一个管理者？如何去评估一个管理者做的好与不好？这是一件挺有意思的事情。

那这次演讲为什么讲管理？因为我见过太多糟糕的管理者了，包括我自己。

技术领导反模式

回顾一下我刚刚开始带团队的时候，可以用“稀烂”来形容。技术领导力其实非常重要，因为好的管理，它决定了公司的战略是否能够得到执行和落地；其次，它也决定了大量工程师的成长和发展。举个例子，如果你犯个技术错误，无非是一个故障；但是如果你犯管理错误，你对一个人的一年、两年甚至更长时间，会产生一个巨大的影响。所以这是一个非常关键的事情，需要重视。

但可悲的现实是，大部分技术领导者是不称职的。以下列举的几种错误模式在技术领域随处可见，基本都可以对号入座：

- **闷头干模式**

延续独立贡献者的工作方法，所有方案自己做，所有代码自己写。

- **路由器模式**

上级任务往下转发，任务结果收集汇报。

- **高压模式**

对上过度汇报，对下持续增加，辅以不科学的价值宣导。

- **不决策模式**

不对任何需求 say no，或者决策全部下放，并让下属承担决策后果。

- **有业务无工程模式**

高度关注短期业务交付，不管工程质量。

作为管理者，你可以通过参考以上反模式来反思一下自己是否称职。如果你是一个

“大头兵”，你可能感觉到自己老板不靠谱，时常被 PUA 到怀疑自己能力问题，但这些问题的根源并不是因为大家不想做好这件事情或者缺乏专业的理论，而是因为行业发展得太快了。以蓬勃发展的互联网为例，像阿里、腾讯等公司在短短十几年时间内从 100 人增长为 10W 人规模，这个时期产生了大量的技术管理者。然而，由于社会面缺乏既懂业务又懂技术、同时又懂管理的人才，只能通过内部提拔。但这些被提拔上去的人是因为管理做得好吗？大部分不是。他们成长晋升得很快，大部分是因为业务好或者技术好，但管理能力却不一定好。因此，这会导致这些人产生了一个错误的认知偏差，认为自己的成长和晋升速度很快，但其实本身的工程能力不足，实际的管理能力与其“总监”、“总裁”的 title 并不匹配。

事实上，很多时候业务的成功和他们的管理能力没有必然关系，换一个人也是同样的结果。业务的飞速增长是因为业务本身处于商业风口，或是因为商业战略、业务战略、运营战略的判断。

那么，如何进行管理呢？我自己也阅读了一些管理方面的书籍，并有一些实际管理工作中的经验和心得体会，可能不是最准确的，但应该还算靠谱。在本文中，我将删繁就简，探讨一些重要的事情。也希望能够通过本文得到一些反馈，帮助我更好地整理和思考。

人才

我自己带团队的时间大概有五年多，总结下来，如果说技术领导者只能做好一件事的话，就是做招聘，挖掘人才。



那为什么是招人最重要？这个观点也出自于我现任的一位 Leader，大约三四年前，我们一起在杭州的 EFC 地下食堂吃饭聊到了相关的话题。当时，他问我，你觉得做管理什么事情最重要？我当时没有想好，沉默了一分钟没有说话，然后他看我笑了笑说，你傻，就是招人。

如果你正在带领一个团队或正处于一个团队中，发现团队里有一些非常棘手的或是痛苦的问题得不到解决，通常都能够最终溯源到某个人身上。比如我反思自己在团队中做得最成功的事情，我能够溯源到我招了一位正确的人，或者是我培养了一位正确的人；反思我做得比较失败的事情，或者是让我头疼的事情，都是因为这个人不是我招聘的，或者是不得不塞到我团队里的。

往往常见的情况是，老板给了业务后团队立马招人扩大规模，你想了想明年可能又可以晋升了。但实际上，一定要给团队招一个正向的人，即与团队目标一致、文化一致，能力一致。如果团队里某个人的专业素养不能支撑住在团队生存的时候，他必然会进化出一种其他方面的能力帮助自己在团队里生存。比如他可能特别“会汇报”、特别会“写 PPT”、特别会“搞东搞西”的一些事情来帮助他自己生存，因为他的专业能力无法跟上团队，为了不被踢出团队，所以需要进化其他能力。所以我后来就养成了一个习惯，就是在招聘的时候会将候选人专业素质的要求提高，我宁愿招不到人，宁愿业务不做或是做得慢一些。

那重视人才意味着什么？

- 你每周花几个小时做招聘 / 面试 / 1-on-1 沟通？
- 你是否对每次面试都严格要求？会不会因为项目压力降低要求？
- 你能欣赏和你不同的想法和观点吗？
- 你有信心充分地授权，并敢于为此负责吗？

在招聘比较旺盛的时候，比如校招开始时，我每天平均会花几个小时的时间来做招聘和面试，和一些 1-on-1 沟通，并且不断地告诫自己，一旦招聘一个人，如果我很喜欢但是又有点犹豫，我就会判断可能哪里存在问题，就会放弃。在招聘过程中，首先我会非常的重视工程能力，比如会进行在线笔试，在线编码能力和历史代码等，一轮两轮三轮，不断地验证候选人的代码是否足够 Solid，讨论时是否足够 Opening，还会问一些明知对方不会的问题，去看对方是直接反馈不会，还是选择绕圈回答，比如有人绕了一圈后，最后结论是虽然我不会，但是我会怎样怎样。但其实我不需要对方去回答这些问题，我需要的是职业发展目标对齐，基本能力对齐，沟通简单，逻辑清楚，这几点要求我一直非常坚持。

到目前为止，我们团队在公司每年的 360 员工满意度调研中的分数都是不错的，这并不是因为员工进来之后我带领的多好，而是因为他们进入了合适自己的团队，并且能够开心地工作，这是我认为比较关键的。

下一步，就是如何带团队。

带团队肯定要定战略、做规划。那战略目标如何落地？阿里是用的 OKR，我每个两周或一个月，都会去看我的 OKR 进展并进行更新。很多人的规划经常变来变去，可能过去一个月后做的事情已经和规划完全无关了，但我今年的规划相比去年，有 80% 其实是一样的，这是多重原因决定的，一方面是因为我认为我做的规划比较科学清晰，另一方面是整个组织结构比较稳定，比如没有老板天天换等等，这两点其实非常重要，实际上也应该如此。

我们的业务并没有多大的变化，我们是做工程的，平时的工作就是写代码的时候做发布、做构建、做运维，以及访问各种云产品等，这些平台和业务是非常稳定的，并不会发生剧烈的变化。但如果你的规划发生剧烈的变化，证明你或者你的老板甚至公司的 CTO 并没有想清楚，这些是不应该发生的事情。

那作为管理者，如何去制定 OKR？关于 OKR 有很多相关的书籍可以学习，此处我根据自己的体感做了一些摘录和总结，主要为以下几点：

1. OKR 应该体现团队为谁服务（for who），即围绕价值阐述。很多人将 OKR 写成了一个指标，比如写“性能优化 10%”，那因此谁受益了？你是为谁服务的？比如写“构建速度提升 10%，让研发者在构建的时候得到更快的反馈”，这里写让谁发生了一个变化很关键，所以要围绕用户价值进行阐述。

2. OKR 应该体现聚焦（即取舍），资源有限，集中精力办大事。举个反例，很多人写 OKR，带了 3 个人，写了 8 个 O，20 个 KR，这样写并不知道他们在干嘛。我的团队规模大约三四十人，我的 O 应该是 4 个，专注这四个目标，明确地告诉团队我们做什么和不做什么，如果我一旦列 40 个 O，那所有人都会不明白到底需要做什么。其实也没有那么多需要做的事情，把每一件事做好是不容易的，很多人做事情是抱着“广撒网总能捞到一条鱼”的态度，但我们的目标其实是去“捞一条最好的鱼”，因此一定要做取舍，集中精力办大事。

3. OKR 应该要尽可能量化（不必 100%），用来校准方向，且量化不应被用来考核绩效。

量化的意思，即不要全是形容词，比如写“卓越的、先进的”，如何衡量是否卓越、是否先进、是否优秀？这很难说，所以需要去尽量量化。同时，量化也是个双刃剑，因为一旦量化之后，大家很容易陷入为了“数字”而去“做数字”的局面，所以作为管理者需要去和团队强调，量化的目的是为了让大家了解方向是否正确，进展是否偏离，而不是为了进行 KPI 考核，一旦强调是在考核，那“数字”必然会变得好看，但实际上毫无用处，还可能给团队带来负担。所以作为管理者一定要谨慎地对待量化，并在团队建立起好的共识。

4. OKR 的承接应该遵循 Single Threaded Leadership 原则。OKR 的负责人没有，或者 OKR 的负责人有一堆，都是错误的。在《亚马逊逆向工作法》一书中，有个观点非常好，即你关键的 O 需要有唯一的责任人，他只对这个 O 负责，并不用对太多事情负责，权责对等。

5. OKR 应该公开，且根据实际情况沟通调整。

反例：一线员工看不到主管的 OKR，看不到更高级主管的 OKR。

工程文化

接下来，讲一些被广泛低估的一件事，工程文化。

为何工程文化重要？

中国的互联网快速发展，导致大家产生了我们的工程能力赶超英美的错觉，但其实在很多方面还是会被打回原形。我们整个工程能力在性能和稳定性领域，其实和谷歌、微软等公司没有太大差距，因为我们分布式系统的用户量会要求我们的性能必须做到极致，否则便无法支撑。我们一直讨论的研发效能、代码质量等其实都是工程文化的问题，软件系统是极高复杂度的系统，工程文化一旦出现问题，复杂度失控，质量失控，会导致系统崩溃；其次，研发人员是知识工作者，是“手艺人”，良好的工程文化能激发他们的工作热情，反之则会消磨热情，增加稳定性风险。像阿里这种规模的公司，每月多人协同产出数以亿级的代码行，其实是非常复杂的，如果没有好的工程能力最后会无法维护。

除了效率问题，另外一个激励问题。工程师其实都想在专业领域做得很好，抛开只为达成功利目的的少数人，大部分人都是希望把自己的工作给做好，代码写得漂亮舒服，被人认可，这是大家都广泛认同的东西。毕竟，谁愿意在代码“屎”山上工作呢？所以我们需要去重视工程文化。

技术管理者与其他领域的管理者之间最大的不同，就是技术管理者除了招聘、做战略

规划之外，还需要关注团队的工程文化。那如何建设工程文化？以下是我的一些做法：

- 要求代码开放，要求 code review，要求 unit test
- 搭建 CI 看板
- 技术领导者每天参加 code review 绩效考核 / 晋升考核中纳入“技术素养”的要求
- 定义阶段性技术目标，降低系统复杂度（如：下线服务，架构治理）

如果作为技术管理者不亲力亲为，工程文化往往会被牺牲。举个例子，我的老板是研究员级别，P10 级别，他每天都会花不少时间看 code review，每天挑几个看一下并给个反馈，慢慢地就形成了一个比较好的工程文化。如果团队里有工程师有事没事删个几千行代码，那他一定是佼佼者，因为降低系统复杂度是比往系统里怼功能加代码难得多的事情，以上的这些方法都很关键。

案例：故障 Review 的重要性

所有的公司都会去做故障 Review，但是我并不能确定是否所有的管理者都会去仔细 Review 团队中的每一个故障和细节。因为从中可以看到这几个方面的信息：

1. 系统架构是否存在问题（例如：存在不合理依赖）
2. 研发流程是否存在问题（例如：代码提交没有单元测试覆盖）
3. 运维应急能力是否存在问题（例如：是否第一时间操作扩容）

基本上团队的每个故障我都会去看，最近看的比较少，因为故障比较少：），在新接一

些系统时故障比较多。为什么要去做这件事情？一是你会对整个技术架构有更深入的理解，因为多数故障是工程能力不足的症状表现；另外，通过观察团队对故障的处理和复盘，可以发现团队里优秀工程师，优秀的工程师在处理故障的时候，他对整个系统的全局有着清晰点认识，但如果是一个不熟悉系统的工程师，他会非常慌张，因为他不知道哪里出现了问题。因此，一方面是在看系统，一方面也是在看人，在这种过程中，看到优秀的工程师需要去给他相应的一些资源支持，鼓励他去做架构治理，去下线系统等等，这都是非常关键的细节之处。

出现故障之后，了解问题并帮助大家改进，我们宣扬 blameless，即创造安全感。如果将故障与人的绩效挂钩，那会造成相互甩锅的局面，没有人想着去改进，故障非个人主观情况造成的话，没有必要去进行追责，鼓励大家发现问题，去改进系统，给予好的正向反馈才是重要的，这也体现出管理者对技术的要求与态度。

建设开放透明的文化

团队文化的建设其实是润物细无声的一件事，很多人将文化建设停留在口头，或者理解的是一些团建聚餐活动等，导致大家认为它是“虚”的，其实不然。如果你的团队文化不注意建设开放透明的文化，那会发生一些反例：

- A 同学把自己写的代码设置为 private，他人不知道其工程能力，老板也不在乎，但是他非常会写 PPT 汇报。
- B 同学找 C，D 单独沟通获取了大量的信息后，和老板单独汇报（选取对自己有益的信息），促成了老板做出对自己有利的决策。

- B 同学就某个想法找 C、D 聊完后，包装成自己的观点，和老板单独沟通，给老板造成他能力强的印象。
- X 领导在一年中多次改变团队目标，但是未和团队解释这些变化的原因，导致团队士气低落。
- 晋升季的时候，B 同学被晋升了，但是领导没有向大家清晰的公开晋升标准以及 B 同学何以满足这些标准，导致团队各种猜测。

那有哪些建设开放透明文化的方法：

1. 开放团队所有的代码和文档
2. 关键决策公开群组 / 会议讨论，鼓励离线记录讨论
3. 公开晋升 / 奖励等标准，公开其过程
4. 公开团队和个人目标（如：OKR）

有很多团队中每个人都有自己的代码库，只有在彼此系统对接合作时才会了解，我要求团队必须强制开放代码，是因为这样做可以让所有人相互了解，形成一种同行的压力。在团队管理中，尽可能地给团队信息公开，信息透明，决策透明，避免私下沟通、信息差等问题存在，能够提高团队的效率和凝聚力。

另外，公开透明的环境会让“小恶”无处遁形，创造鼓励向善的行为十分关键；其次，在团队中给予充分的信息和规则，知识工作者自己可以做最高效的判断；最后，文化的建设需要时间，不可能一蹴而就。

达尔文进化论的启发

最后聊一下管理中的激励，大家讲管理经常会提到马斯洛需求层次理论，但这里我认为达尔文的进化论在管理和激励上给我们的启发会更大。

达尔文的进化论大家都了解，这里不再赘述，我简单提炼下几个核心的信息：

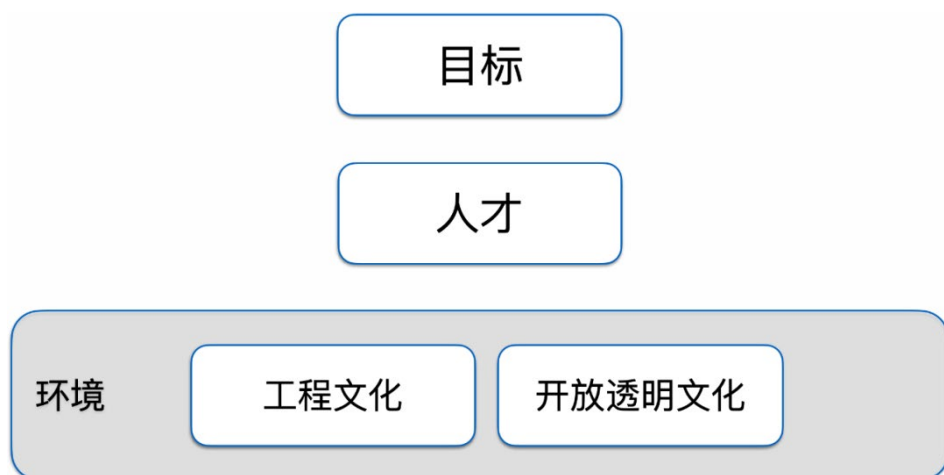
- 人类进化的 99%时间（约 200 万年前到 3.5 万年）都处于依靠狩猎采集的社会状态。在这漫长时间内进化出来的，追逐地位和尊重的心理特点，普遍刻在每一个人的基因里。
- 在狩猎采集阶段，面对力量和速度都远超人类的野兽，人类的核心竞争力是相互协作。
- 被集体所排斥（安全感缺失）意味着高概率的死亡。
- 被集体所尊重，获取更高的地位，意味着更多的物质机会和更多的交配机会。

我们今天所有的心理状态都是在进化过程中逐渐形成的，都旨在帮助我们生存，以上这些心理机制在今天的管理和激励上也有很大的借鉴意义，管理者需要意识到人类普遍对安全感的需求，对地位和尊重的需求。

总结一下，那如何在团队中营造一个充满安全感的工作氛围？首先，从微小之处让团队成员感受到自己的价值与意义，比如 1-on-1 的沟通，鼓励团队中的正向行为，公开场合明确目标，让团队成员知晓自己的决策过程等方式，让团队成员获得安全感和价值感。

其次，理解团队成员对于“名利”看重的心理机制，作为管理者可以尽可能的去为团队成员的成长提供资源支持，但固然无法满足每个人对于“地位”和“资源”的诉求。

然而，管理者可以做到对每个人的尊重，对其工作成果的尊重，不论是开庆功会还是群里发红包等形式，让团队成员感受到自己工作的价值和意义，这对促进团队成员的创造性行为是非常重要的。



定目标，找人才、建文化，这就是我们做团队管理比较关键的一些内容。有时候很多事情是很表面的，但实际上内心的一些机制、人的认知、人的心理，其实是起了一些决定性的作用的，我们是改变不了的，我们只能接受。在这些基础上，我们再思考可以去做哪些措施去不断地优化。

重新 Review 你的面试流程，是否有明确的标准要求？是否有严格的流程遵循？

和团队的关键成员安排一次 1-on-1 沟通，关注你和他 / 她是否有清晰一致的目标？

把团队所有代码设置成尽可能公开，至少团队内公开，每天至少花 2 小时 Review 代码。

思考并修订团队目标（OKR），和你上级讨论达成共识，在团队公开宣讲。

思考团队目标，如果必须要砍掉其中一个子项，你会怎么选择，写下你的思考。

整理团队的技术债务和技术风险，产出改进计划。

最后我认为很关键的一句话，做管理实际上你要在团队内建立一种氛围和文化，把每个人的善意都激发出来，我觉得这是非常值得做的一件事情，不一定是伟大，但是是非常有意义的一件事情。抛开我们对升职加薪那些功利的追求之外，每个做管理的人，不论你的团队是 10 人、20 人、30 人、50 个人还是 100 人，每个人产生的影响都是巨大的，这也可能是对技术更有价值的一件事情。

作者介绍

许晓斌，阿里巴巴工程师，技术管理者，目前负责阿里巴巴集团运维及构建基础设施平台，《Maven 实战》作者。

如何为那些在裁员中幸存的人重建技术文化

作者：Mehnaaz Abidi

译者：刘雅梦

策划：丁晓昀

一波裁员潮冲击了软件行业，并改变了技术文化的定义。本文探讨了多家科技公司的情况，以及为支持幸存下来的员工和不得不告别的员工而做出的不同选择。它为我们这些留下来的员工以及如何在我们的技术团队中重建文化提供了建议。

本文要点

- 2023 年重建技术团队文化的三大领域是团队自由、远程或混合工作以及保护多样性。
- 通过专注于在你的直属团队内而不是在整个公司内建立文化，从而找回动力。
- 投资自己；不要等着公司给你学习的机会。
- 一家公司的真正文化可以从他们如何对待离职员工和他们的前员工的网络中看出。前员工是更强大的未来雇员。
- 不要为成为公司的拥护者而感到压力，而是要成为团队的拥护者。

在 2022 年旧金山 QCon 和 2022 年 12 月 QCon Plus 会议上，我谈到了软件行业在新冠疫情后如何学习适应新的全球现实以及其对流量度量的影响。我的演讲名为“作为一个技术团队，如何在一个感觉像是《疯狂的麦克斯》电影的新现实中获胜？”我

分享了一些快速技巧，以帮助技术负责人在世界试图恢复常态之际，为混合和远程团队创建文化。我强调过，后疫情时代需要留住女性技术员工，以及将文化建设作为一项新的指标纳入流程框架的重要性。

自那次演讲之后，一波裁员潮冲击了软件行业，并改变了技术文化的定义。

看到一些公司采取非人性化的方式是残酷的，而另一些公司却为支持离职员工付出了额外的努力，这着实让我惊讶。许多人会因为找到工作而感到幸运和欣慰，同时又会为前同事的遭遇而内疚。

2020 年，《哈佛商业评论》(Harvard Business Review) 将此与人们质疑的幸存者内疚进行了比较，

“为什么我成功了，但他们没有？当我还在工作时，我将如何面对那些在经济困难时被解雇的朋友？雇主会不会再次裁员，然后像对待他们一样对待我？”

未来几年，对雇主的信任将是许多公司求之不易的回报。

从那以后，我采取了额外的步骤来了解多家科技公司的情况，以及为支持幸存下来的员工和那些不得不告别的员工而做出的不同选择。

在这篇文章中，我将重点关注那些留下来的人，以及如何在我们的技术团队中重建文化。

在经历了同事被解雇后，如何重新找回自己的动力？

当你对雇主的信任度很低的时候，这更有可能？

答案不在于去另一家公司，而在于你在当前的工作场能做什么。

人们很容易认为，新的工作场所将能解决现有团队中的一些文化不适应问题。在微软《2022 年工作新未来》报告中，他们提到寻找新工作的主要文化因素是不尊重、不包容、不道德、残酷和虐待。根据我的经验，很难通过星光熠熠的雇主品牌和几次面试来判断这些因素，在这些面试中，招聘经理会由于压力而必须展示公司最好的一面。因此，作为一名员工，我们找到文化契合度的最佳时机是改进现有的“团队”文化。

我强调的是“团队”而不是公司。我会将帮助你的邻居吃饱饭与解决世界饥饿问题相提并论。规模和范围对于在技术团队中成功构建良好的文化至关重要。你仍然可以安然入睡，因为你知道你的邻居是健康的，而这个国家还有更大的问题需要解决。

因此，在你的直属团队范围内认清你的影响范围。如果你是一名工程经理或工程师，请专注于你的团队，包括项目经理和设计师。如果你所在的是一个职能团队，比如架构，那么请把重点放在围绕管理者和直接下属的团队上。如果你是一个管理者中的管理者，比如高级工程总监，那么再次将注意力集中在直接下属身上，同时要求他们为自己的团队复制框架。不要试图同时为多个团队构建文化，这会让你不知所措。

既然你已经通过缩小范围减轻了自己和团队的压力，那么就开始了了解你是否：

- 信任你的团队成员吗

- 可以和他们一起开怀大笑吗
- 在团队中能就共同关注的问题达成一致吗
- 能想象和他们一起工作至少一年吗？

如果以上任何一个问题的答案都是肯定的，那么你就在这部《疯狂的麦克斯》电影中找到了你的安全网。你有一个团队，你可以与他们一起努力改善心理安全感，重建裁员后留下的破碎文化。对于我们中的许多人来说，工作和生活是相互关联的，我们在这些空间中建立了亲密的友谊。失去一个像朋友一样的同事，可能会引发内疚、愤怒、否认、后悔等等。所以从这里开始。给自己和团队成员足够的空间来谈论“他们的感受”。

如果你在裁员期间担任管理职位，那么在尝试之前建立你的诚意就变得至关重要了。例如，一位工程经理为了节省的公司成本，通过一封电子邮件解雇了一些终身自由职业者，那么当他尝试进行敬业度调查或试图谈论团队文化时，他们的团队会几乎没有什么回应。通过电子邮件辞退的方式，使得他们不再被团队视为一个能承受压力的人，而是视为一个头衔。工程经理在潜意识中设定了这样的信息：“我把我的团队视为资源，而不是人。在让我的团队成员离开之前，我没有给予足够的关心，甚至没有进行过交谈。”因此，团队将避免任何来自工程经理尝试建立的联系，直到他们能够证明他们对团队的关心是真诚的。

微软的报告还指出，在工作中感受到被关怀的员工在工作中快乐的可能性是其他员工的 3.2 倍。这并不一定要通过雇主，许多员工在经历了裁员过程之后，会发现任何此类努力都是假的。但真的是你从直接团队成员那里得到的关怀。

将你的精力、承诺、坚持和创造力集中在你自己的团队中。你会发现自己的动力在上

升，而无需伪装你对公司的承诺。不要因为要成为公司的拥护者而感到压力，而是要成为团队的拥护者。

在受影响的技术团队中，重建文化的三大重点领域是什么？

团队文化不是将一堆框架组合在一起，而是一个舒适的空间，在这里你的团队可以感受到彼此的联系、哀悼、治愈和庆祝。

让我们从技术团队对工作满意度的第一个要求开始——自由

从《摩登原始人》开始，在裁员之前，产品 x 技术 x 业务经常会出现分歧。许多公司都提出了可靠的优先级框架和技术。但优先级排序总是以某些团队感到受限为代价。因此，我们永远无法真正平衡高优先级和全面的工作满意度。我曾见过一些出色的功能，它们带来了非常高的用户留存，但背后的团队却认为这些不是正确的工作。

那么，我们如何在团队中培养更大的自由感，而不会被我们无法影响的优先级（尤其是在当前艰难的市场条件下）搞得精疲力竭呢？

找时间讨论一下团队的优先级。一旦你把压力/紧张从系统中释放出来，你就会发现在这些优先级中仍然有巨大的自由。我曾经和一个团队合作过，他们承受着巨大的压力，要“尽快”彻底重新设计他们的整个产品。我们经历了五个悲伤的阶段——否认、

愤怒、讨价还价、沮丧和接受。

当我们进入验收阶段时，奇迹发生了。团队在这个项目中找到了合作的共同目的，并找到了对产品进行不同思考的自由。但这种自上而下的消极要求已经早早地剥夺了他们发挥创造力的能力。在当前的市场条件和公司的被动性中，自上而下的优先级只会增加。

因此，一个团队可以从接受开始，并更早地找到发挥创造力的自由，或者像其他团队一样，经历五个悲伤的阶段，然后再达到同样的状态。

要求继续远程或在家工作

在遵守公司关于远程或混合工作政策的同时，每个团队都可以为各自的团队成员创造在非办公地点工作的灵活性。有自己的习惯，通过工作与生活的平衡来相互支持，尤其是对初级看护者来说。

团队可以随时建议他们的人力资源团队不要设定固定的办公室工作时间，而是让员工根据团队的需要选择工作时间。我看到许多人力资源团队已经认识到了这一点，并支持员工决定什么是对他们团队最有利的。但如果你的人力资源团队还没有做出决定，那么就保持团队的地位。迫使所有员工在办公室固定工作的唯一原因是象牙塔认为这将能提升公司文化。

不要因为担心失业而放弃你在家工作以努力提升自己知名度/价值的混合例程。我们不能让任何人破坏在加强工作与生活平衡方面取得的进展，尤其是由于大规模裁员引发的担忧。相反，通过提高团队内外的在线沟通技能来提高你的知名度。

一些对我有用的事情：

- **利用每封电子邮件作为自我介绍的机会：**在科技领域，当我们必须解释大的概念或让一个大的团队做出决定时，我们通常会写内部电子邮件。利用这个机会，让听众知道你是谁，以及他们为什么应该听你的。
- **通过平衡细节和清晰度来成为具有同理心的人：**多年前我在一篇文章中读到，“唯一比听一个你不知道他们在说什么的人说话更糟糕的事情就是听他们絮絮叨叨。”始终保持简单，并提前阐明你谈话的预期结果。询问他们是否需要了解更多信息，不要让太多的技术信息压倒他们，让他们不知所措。
- **自信并定期分享召回价值：**工程师们经常努力提高他们在团队之外的知名度。通过观察新闻频道的记者，以及他们如何用事实和正确的语气进行激烈的对话，我学会了自信。他们教会了我在压力下进行有意义的对话，展示自信的艺术，以及如何让房间里充满影响力。我的建议是找到有影响力的人，他们可以帮助你改善表现方式，并开始在有公司的虚拟形式（如团队演示、全体会议、公司员工大会、利益相关者更新等）中为自己创造知名度，定期撰写技术文章，并让公司为此感到兴奋。

是保护团队的多样性

对于许多科技领导者来说，招聘多元化的员工非常困难。多年来，我们一直在努力加强不同的社区，并将他们与技术角色联系起来。最近的裁员分钟内就销毁了所有这些努力。所以我们中的许多人现在都有责任保护团队中剩余的多样性。

我自己的关注点是更好地理解如何提高女性团队成员的心理安全感。新冠疫情对科技

行业的女性来说尤其困难，该行业的女性留存率大幅下降。许多因素，包括作为主要的看护人，都决定了她们要么暂停职业生涯，要么离开职业生涯。休完产假回来已经变得很困难了，现在随着大规模的裁员，复杂性增加了，他们甚至不确定自己是否还能回到原来的团队。

女性技术工作者比以往任何时候都需要更多的支持，尤其重要的是，我们要将重点放在更加个性化的挽留措施上，以支持她们。我听过一个故事，一名女性技术员工在分娩时收到一封被解雇的邮件，并在她生完孩子几分钟后阅读了全部的细节。她在这家财富排名前五的公司工作了9年。我们不能允许任何公司在任何国家、任何时期建立这种先例。作为一名女性技术负责人，我不能接受任何公司的这种行为。

我正在通过承担更多的责任来应对这一现实，以确保维护科技行业女性的利益。我很高兴看到我们行业中的许多女性同行也对我们的社区承担起了类似的责任。

我留给大家的想法是，公司文化不再是单个团队文化的总和。较高的团队 eNPS 和公司 eNPS 之间不会有因果关系。那些试图让裁员变得更简单的雇主，已经在科技行业产生了不信任的副作用，这种不信任将长期存在，并将影响工作的开展方式。对于一个可持续的未来工作场所，这些公司为每个团队提供空间来领导他们的文化重建至关重要。

一些员工会热情地参与其中，但其他人则需要时间来重建健康的工作关系。这将是一项耗时、富有挑战且容易遭受挫折的工作。然而，这将按照团队的步伐，在更坚实的信任基础上进行。最终，你不仅可以帮助你的团队从这次事件中恢复过来，还可以创建一个更健康、更高效的公司。

作者介绍

Mehnaaz Abidi 是 Urban Sports Club 的首席产品和技术官。在加入 Urban Sports Club 之前，她曾担任阿迪达斯（Adidas）的产品副总裁，并在耐克（Nike）、trivago、Tesco 等公司担任技术负责人。

原文链接

<https://www.infoq.com/articles/rebuild-tech-culture-layoffs/>

架构师角色的演变：从发号施令到与团队合作

作者：Leigh Griffin, Chris Foley

译者：明知山

策划：丁晓昀

不断变化的世界

与传统的科学相比，软件可以算是一门非常年轻的科学。但即使是在它的婴儿期，它的关键组成部分之一——架构及其形成方式已经发生了重大变化。架构蓝图——花几个月时间完成可以解决所有问题的完整设计——一去不复返了，也没有了由一人负责管理所有东西的场景。之所以发生这种模式转变，一部分是因为行业创造出了更好的工具，还有一部分是因为用户行为发生了变化。他们的交互模式从事务性服务转变为消费驱动型服务，将用户行为从记录系统转变为参与系统，用户现在有了更主动和及时的需求。软件架构需要随之一起演化并拥抱可用的工具才能满足这些新的需求。现在的架构更多的是关于决策而不是结构，更多的是关于对不断发生的变化做出响应而不是遵循规划，更多的是关于频繁交付而不是一次性大型交付。这对架构师所扮演的角色的影响是非常深远的。

在这篇文章中，我们将探讨共享架构的文化变化和架构师角色的演变。从之前依赖架构师的权威和独特视野，变成了在系统设计问题浮出水面时需要整个团队的投入一起解决。这导致了一种控制反转式的团队关系，向共享所有权转变的团队可能正在为融

合这种新范式做着苦苦的挣扎。

我们将分享我们是如何经历这一变化的。我们有超过 25 年在一个团队中担任多个职务的经验，从工程师、产品负责人到团队教练和经理。这些角色中的每一个都让我们能够与架构师接触，因此目睹了行业和架构师角色的演变。我们希望能够为那些在转变过程中苦苦挣扎以及那些希望进一步增强和推广他们的架构的人提供指导。

变化因素

职责的变化

传统的架构师有许多基本职责，其中之一就是关于应用程序的可伸缩性。架构师需要考虑许多不同的因素，确保能够处理系统的预期负载。这些决策包括：哪种语言最适合用来处理这种类型的应用程序？如何处理 I/O？阻塞还是非阻塞？数据库采用怎样的策略？需要多少个 CPU 核心？内存呢？存储呢？这些考量因素影响到了部署策略、特定硬件或芯片组的可用性，甚至是应用程序的部署位置。这些决策为我们提供了有关应用程序生命周期的整体概要、它的预期使用以及更新节奏和策略。

在现代环境中，开发团队通过使用工具减少了之前架构师需要考虑的问题。例如，自动伸缩功能解决了应用程序的计算资源消耗问题。Kubernetes 这样的编排平台让部署和处理突发负载变得非常简单，这些平台可以根据需要增加应用程序实例，并在流量减少时逐步减少实例。分析工具，从圈复杂度的静态分析到性能分析指标，再到 API 功能的可视化，现在已经在整个团队层面提供了丰富的信息。这些工具现在已经

出现在标准的工作规范中，这意味着架构师以前的专业知识自然而然地分布到了整个团队中，知识生成和数据见解远远超出了单个角色在团队中所能分享的程度。这意味着这个领域的一些所有权和责任已经转移到了整个团队，而不是在某个个体身上。共享所有权已经成为一种现象。现在，团队通常会根据行业标准、用户期望和公司内部的技术一致性来决定采用什么工具。

用户使用模式的变化

云计算（或 SaaS 文化和模型）的快速发展要求我们在如何发布、何时发布以及发布什么方面变得更加灵活。现在的重点是提供更健壮的服务和支持，让团队能够快速改变他们的关注点。功能的增加会带来更多的用户使用，了解用户的使用情况就成为开发和演化功能的关键决策。在以前，这一强化要素是一项长达数月关于稳定性、伸缩性和健壮性的思考，而如今已让位于实验性的意愿。

技术预览版功能（不要在生产环境中使用的警告通常会被故意忽略）可以让应用程序的演化与用户的需求同步，消除了用户与团队之间的脱节。在以前，这种关系通常由业务系统分析师或产品负责人等角色负责维护。现在，团队的用户意识更强了，有时甚至强过架构师。他们了解用户是如何与系统交互的，并通过遥测应用程序的见解知道用户何时与系统发生交互，了解用户需要什么、为什么需要以及如何需要。这为应用程序的开发带来了强大的多层面观点，因为现在整个团队——拥有不同的背景、技能和专业知识——可以为更大的愿景做出贡献，真正地从个体角色转变为主要由团队驱动并与用户需求进行协作的角色。

从代码层面来看，微服务的兴起最能体现其实际影响。随着所有权和需求发生变化，

应用程序需要处于能够独立演变的位置，允许一些服务尝试不同的东西、测试一个功能，能够为部分或所有用户打开和关闭某些功能。这创造了一个良性循环，这种开发方法催生了一套支持工具和服务，（如 API 网关）来管理服务合约，还有消息传递系统（如 Apache Kafka）和支持 Spring Boot、Flask 和其他特定语言框架的微服务。工具的可用和成熟反过来使得团队更容易自行选择微服务架构风格，从而进一步推动在工具上的投入。

对于架构师来说，他们不能再按照自己的蓝图来设计架构了。现代用户使用模式对灵活性的要求更高。架构师必须不断调整系统设计来满足快速变化的用户需求，必须促进架构向前演化。

思维模式的变化、机遇、挑战和现在的架构师需要掌握的新技能

在了解了这些变化因素之后，我们相信对于现代架构师来说，他们需要做出改变，需要应对挑战和抓住机遇，并练习和掌握新的技能。

软件架构一直在演化

软件架构是一个演化的旅程，有着不同的路线和影响，这是当今软件架构的一个基本原则。这种演化意味着我们需要根据了解到的东西改变我们的思维，而架构师在促进架构对话方面发挥着关键作用。以下引用了我们在 Red Hat 与一位首席架构师进行互动时说的两句话，反映了当今架构师的一些想法和担忧：

康威定律的影响是一个关键主题：系统的架构反映了组织的结构。

架构对话的另一个作用是提出一些没有人意识到的问题，或者每个人都意识到但不愿意谈论的问题。把这些问题提出来讨论是必要的。不是只关注如何解决和实现，而是更多地讨论它们，这样每个人都知道该怎样前进，或者可以随着设计的演化做出适当的调整。有时候，这些讨论最重要的输出是认识到有些问题在当时并不是问题。这种清晰度对每个人来说都很重要。它可以让人们专注于即将到来的任务，而不是被阴影所笼罩。换句话说，它消除了一些无法言喻的负担。

—— Emmanuel Bernard，Red Hat 杰出工程师和架构师

假设大多数人会倾向于同意这些想法，但他们的架构决策过程是否进化到与这种想法相匹配的程度？他们考虑到组织结构了吗？他们是否放弃了预先设计而引入了预先对话？任何变化的第一步都是先意识到，然后才是接受。

影响产品/服务变化的主要因素之一是用户互动和反馈，用户有可能是内部团队，也可能是付费客户。在现代市场中，反馈循环是持续进行的，架构师必须充分利用这个机会。与持续反馈相应的是期望更频繁或尽可能接近持续反馈的频率进行交付。这给架构师、团队和组织结构带来了挑战，因为持续交付很少能够独立实现，它通常需要在组织层面才能成功实现。

实现频繁交付的小型迭代很适合这种时间窗口。然而，当引入潜在的更大的功能块（例如架构重构）时，它可能不像人们期望的那么简单。这对架构师和团队提出了挑战，他们要能够频繁地交付组件，同时要确保服务能够高质量运行，能够满足 SLA 和质量期望。更重要的是，在开发的早期，在强制的演化发生之前提出解决方案路径会带来一种对强制的变更拥有所有权的感觉。

在很多情况下，设计决策时机可以与业务需求挂钩。及时地将业务需求与系统设计决策结合起来是架构师及其团队需要解决的真正挑战。传统的架构重点关注最终目标，现在却变成了“接下来需要做些什么来充分解决未来几个月的业务需求”。这可能会导致我们做出一些后续可能需要再次改变的决策，但它们在当时是正确的。我们通过构建产品来获得经验，我们的客户通过与我们的产品交互来获得经验，这为我们提供了紧密的反馈循环。这是系统架构的自然演变，提前了解并制定策略来处理最好的情况和最坏的情况是团队需要的关键技能。架构曾经被认为是一条笔直的道路，但它不是，或者说将来都不应该是。架构的演化是一条曲折的路，每一次转弯都为我们带来一个学习机会。

这并不是说架构师必须忽略架构的最终目标（他们曾经唯一的关注点）。在当前的环境下，产品负责人成为架构师非常重要的合作者，这意味着最终目标和愿景成为共享的经验。对于架构师来说，与产品负责人就产品/服务的愿景展开讨论、合作并达成一致是确保方向性的必要条件，即使这个方向可能并不总是非常明确。相反，这种讨论让产品所有者的愿景变得更加贴近现实，他们可能需要做出妥协，知道什么是可实现的、什么样的时间表是现实的。这个愿景成为技术决策过程的另一个重要输入。

架构不再是单个人的职责

架构师在软件开发当中扮演独立角色的情况一去不复返了。系统架构现在是一项团队运动。团队具备跨职能交付产品的能力，由为交付软件过程增加价值的人组成，其中仍然包括架构师。之所以会这样，正如前面所说的，部分原因是软件开发生态系统涉及技术、语言（不仅是开发语言，还有业务和技术语言）、体验（开发和用户）和利益相关者。没有哪一个人能够覆盖到所有这些方面。这种变化意味着架构师需要转变

思维方式。对于架构师来说，他们的工作成为了团队的一部分，这有很大的好处，但也存在挑战。

依靠他人并将职责移交给他人，这是一项需要掌握的重要技能。这包括在架构师和团队成员之间建立信任。他们必须共享技术方向的所有权，并信任同事能够驱动系统的某些方面或组件。架构演化已经从单一的自上而下的指挥转变为团队联合参与贡献，所有人都可以有不同的观点。

信任是双向的，有时候需要对判断意见加以保留，允许新的想法和见解不断涌现。架构师需要成为建立团队心理安全防线的主要人物。

团队层面的失败不应被视为无能，而应被视为把事情做好的机会。更频繁的交付节奏有助于实现这种模式，因为他们可以快速采取行动。

架构师需要接受架构设计已经从一个单独的个人职责演变为一个共享的团队职责的事实。接受了这样的事实，他们就可以利用团队可以提供的一系列好处。对于那些戴惯了传统架构师帽子的人来说，为了成为团队成员的一部分而降低自己的身份是一种挣扎。不幸的是，我们已经亲眼看到，一些架构师对团队在创新阶段提出的想法、建议或改进表现出了挑战的姿态。

这就导致了僵持的局面，随着时间的推移，团队慢慢变得沉默，他们知道自己的建议没有被采纳，而架构师无法调和自己的局限性，也就无法规划前进的道路。这变成了一场对抗头衔、不愿放弃控制权、承认自己认知和能力不足的斗争。在我们的团队中，这种行为持续存在——成为所有变更的审查者，这破坏了更有能力的团队成员的成长，减缓了交付的速度。我们已经在多个行业的一些公司中看到了这种模式，而在以云为

中心的环境中，技术栈的快速发展让这成为一个更大的挑战。

这意味着架构师不再是技术方面的唯一权威，因为改进的速度要求工具也不断改进，技术栈和方法的改变发生在每个开发者身上，更重要的是整个行业使之成为一场技术军备竞赛。如果架构师不愿意信任和授权他们周围的人，这将无意中导致支持、信任和实现技术栈改进失败。对于团队来说，这是一场注定失败的战斗，因为架构师觉得他们需要让自己的知识跟上变化的速度才能做出决策，然而，开发者每天都在积极地编码、调试、实验和学习，而且是在架构师无法达到的更深层次上。如果这种知识鸿沟未能被弥合，就会导致团队人员流失和丧失心理安全感。解决这个问题需要强有力的领导，更重要的是需要强有力的支持，帮助架构师克服他们正在经历的恐惧。架构师和整个团队都需要意识到，对团队来说最好的东西可能对个人并不是最好的，但通常会让个人获得最大的收益（在产品或服务的改进方面）。

为了确保团队能够达成共同理解而放慢发布节奏就是这方面的一个例子。之前的公司有一个主题专家（SME），他被授予了架构师的头衔，并几乎独自在设计方面推动功能的实现。他们的架构愿景是重构现有的面向组件的设计模式，使之成为一种更加模块化的基于插件的架构。这种想法源于他们在架构最佳实践方面的丰富经验，以及与客户深度联系。他们的愿景被认为是一个强大的概念证明，用一封电子邮件解释了它们的基本原理，并建议在出现下一波客户流失之前进行重构。不可否认，愿景、激情和能力是我们（工程师 Chris，工程经理 Leigh）想要挖掘的东西，但团队内部弥漫着一股不安的气氛。这是一个架构上的转变，整个团队对新技术知之甚少。在这种情况下，SME 将在下一次出现客户流失大约三个月后领导另一个项目，并且已经按照合同期待交付，这意味着将会产生延迟成本。

我们决定支持 SME 的愿景，但要求他们更详细地阐述这么做的好处（更好的互操作

性、更顺畅的客户集成、更容易的调试)，然后与客户协商我们的发布承诺。这为淡化个人主义提供了一个安全的基础，更重要的是让整个团队适应了技术和变化。结果是我们得到了一个可持续的架构，但更重要的是，其他五个开发者培养了可持续的技能，他们将长期参与这款产品的演化。毫无疑问，这给销售团队和客户端带来了压力，但在接下来的 18 个月里，开发团队获得的回报是显而易见的，因为他们正在协商新的业务功能需求。这是一个皆大欢喜的结局，但更广泛的接受度源于工程领导为团队和产品的进化创造安全感而进行的诚实对话。

只具备技术敏锐性是不够的

掌握技术知识一直都是架构师所必备的，而商业头脑和对市场的了解无疑增加了其重要性。但是，架构师需要做出的最大改变是对软件生命周期中涉及的所有人员进行指导。这听起来可能过于简单化了，但在日益增长的快节奏软件开发行业中，对于架构师来说非常重要。他们倾听和消化业务视角、技术需求、来自开发人员的需求以及管理层快速交付的需求的能力变得至关重要。架构师需要能够使用“强大的开放性问题”作为激发更深层次思考和引出不同观点。

“为什么”这样的提问方式带有评判的味道，例如，“你为什么要采取这种方法”。如果将问题改为“是什么让你决定采用这种方法”，会促使被问者解释他们的想法，而不是为他们的决定辩护，因为当被问及“为什么”时，他们可能会认为自己的决定是不正确的。这种简单的改变，以及使用开放、好奇的语言，可以在整个团队中创造包容性，更重要的是，创造一种支持的氛围，而不是一种被认为是挑战的氛围。

架构师已经成为一个通晓多种语言的人，除了传统的技术语言，他们还使用了商业语

言和从工程团队中获得最佳观点和想法所必需的语言。

实用技巧

经过总结，这里为架构师提供了 6 个实用的技巧，也为正在转变泥潭中挣扎的团队提供了 6 个实用技巧。

给架构师：

1. 成为帮助团队架构理解的导师，而不是障碍。公开主动地分享你的知识。
2. 为那些只有你自己感觉到但团队可能没有意识到的挑战寻求指导，来帮助你克服内心的挑战。不要独自承受，支持性的指导有助于你的角色演化。
3. 欢迎来自客户、团队和环境的挑战。这种反馈循环可能会让人感到筋疲力尽，但可以带来巨大的回报。
4. 用你的经验将对话引向你的专业知识告诉你会遇到的挑战。
5. 了解团队的动态、他们的优缺点、他们对工具的掌握，以及他们日复一日地构建应用程序的实际情况。帮助你在正确的时间组织你的输入，在哪里可以带来最大的价值。
6. 成为人际关系的建设者。培养你的软技能，建立起人际网络，从销售团队到产品负责人，从工程经理到技术中小企业。每天都要培养和维护这些关系。

给团队：

1. 为非领域专家总结使用工具的经验，带他们踏上理解之旅。

2. 利用架构师丰富的经验来洞察你可能有的想法、挑战或主意。他们现在是你团队的一员了。
3. 简单明了地表达你的想法、优点和缺点，准备好接受开放性和有挑战性的反馈，构建心理安全感。
4. 把头衔和自负留在门外，拥抱团队环境，向团队里的每一个人学习。在当今的软件设计当中，你对设计过程的影响是真实存在的。
5. 培养你的演讲、沟通和指导技能，并每天使用它们，在快节奏的团队中进行信息交流是至关重要的。
6. 尽你所能留住你的架构师。他们深厚的专业知识对团队的成长和壮大是无价的，不要让他们感到孤立，让他们觉得自己是团队的一部分，是未来解决方案的一部分。

结论

对于软件行业，更重要的是对于我们的用户来说，架构师的角色已经发生了根本性的变化。我们与用户互动的方式，我们构建、发布和支持软件的方式都发生了变化。这种变化对整个开发团队和他们之前的支持角色（如质量工程/质量保证）进行了赋能。现在，每个人都可以发声，关于系统如何随时间的推移而演化，他们有自己的意见和有效输入。与此相辅相成的是两个独立但相关的变化。首先是终端用户期望的变化，现在要求有更快速的反馈，通过不完美的服务来指导实现他们的需求以及何时实现。其次，出现了一套支持开发者日常工作的工具。这为以前只能由架构师解决的问题带来了解决方案，并允许开发者对性能、伸缩性和设计有更多的见解，并自然地渗透到团队中。架构师需要扮演的角色发生了变化。他们多年的经验和丰富的最佳实践

现在需要重新渗透到团队的日常流程中。这是一个提升整个团队经验水平的机会，为我们如何构建软件创建了一个更多样化的视图。实现这种改变可能很困难，它需要管理层和团队的支持。它还要求这个角色愿意面对挑战，提供比以往任何时候都更多的价值，为了团队、产品和客户的改善而放弃对头衔的渴望。

原文链接

<https://www.infoq.com/articles/architecture-architecting-role/>

优秀的产品背后，必定有优秀的团队做支撑。《中国卓越技术团队访谈录》是 InfoQ 打造的重磅内容产品，以各个国内优秀企业的 IT 技术团队为线索策划系列采访，希望向外界传递顶尖技术团队的做事方法/技术实践，让开发者了解他们的知识积累、技术演进、产品锤炼与团队文化等，并从中获得有价值的见解。

此前，访谈录嘉宾邀请主要是以 InfoQ 主动邀约的形式进行，现在我们决定**长期开放报名通道**。

如果你身处传统企业经历了数字化转型变革，或者正在互联网公司进行创新技术的研发，并希望 InfoQ 可以关注和采访你所在的技术团队，就请填写下方链接中的表单吧！期待你的报名。

