

## 31 | 业务目标和技术目标两手抓：怎样打造高效团队？

2019-11-04 葛俊

研发效率破局之道

[进入课程 >](#)



讲述：葛俊

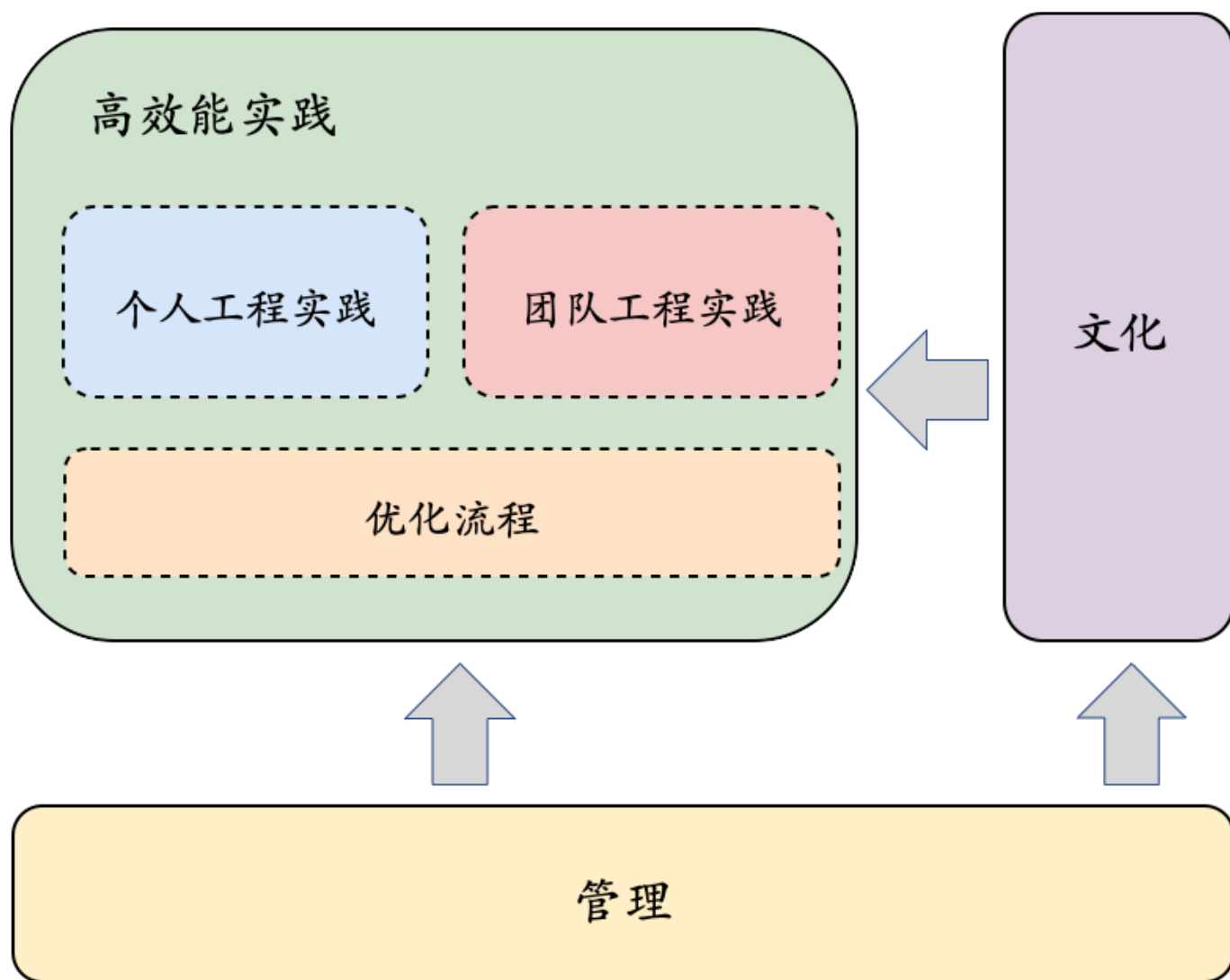
时长 17:58 大小 16.47M



你好，我是葛俊。今天，我来和你聊聊管理和文化。

今天这篇文章，是最后一个模块“管理和文化”的第一篇。在前 30 篇文章中，我已经从优化流程、团队工程实践、个人工程实践这三大方面与你介绍了很多原则、方法和具体实践。通过这些内容，相信你应该对软件研发活动的本质有了更深刻的理解，也对这条超级灵活的流水线如何提效有了新的认识。

但作为团队管理者，要提高团队的研究效能，掌握了这些原则、方法和实践后，还要通过管理和文化让它们真正在团队落地。管理是提高团队研究效能的基石，而文化是持久高效的保障。同时，管理又决定了文化，如下图所示。



所以，在接下来的几篇文章中，我会参考硅谷高效能公司的一些管理实践和原则，以及我在国内外公司的落地经验，与你介绍如何通过管理和文化来提高团队的研发效能。今天，我们就先从技术管理者的主要工作步骤出发吧。

在我看来，技术管理工作主要有如下 3 个步骤：

1. 寻找目标；
2. 目标管理；
3. 计划并执行去实现目标。

其中，计划执行又包括人、流程和工具 3 个方面。

## 第一步：寻找目标

在 [第 1 篇文章](#) 中，我曾和你分享研发效能的三要素是有效、快速、可持续性，其中第一条就是有效，也就是准确性。所以，要想建设高效的研发团队，技术管理者的第一项重要工作就是，作为舵手，为团队寻找方向和目标。

**首先，技术团队的根本目标就是业务目标。**毋庸置疑，业务目标是团队存在的意义，完成它是一切的基础。

业务目标的设定有两个层次：

弄清楚公司和上级对团队的预期，达到这个预期，是团队的基础目标。

在基础目标之上，还要给团队设定一个进取目标。我们需要分析公司的发展方向，以及团队的实际情况，找到那些既符合公司利益，又通过跳一跳就能够得着的目标。

除了业务目标之外，还有**另外一种技术管理者一定要关注的目标，即技术目标**。如果只关注业务，我们的行动就容易短视。有这么一句话我非常赞同：技术常常在短期被高估，在长期被低估。作为技术管理者，我们需要看清楚并坚持技术在长期可以发挥的巨大作用，在技术上持续投资，制定并完成合理的技术目标。

在我看来，**技术目标主要有两种：**

一种是关于偿还技术债的，这是处理已经形成的问题。关于技术债的处理，你可以再回顾下 [第 14 篇文章](#) 中的相关内容。

另一种是前瞻性的技术目标。作为技术管理者，我们要有灵敏的技术嗅觉，对即将出现的技术挑战，做一些预防和准备。

关于前瞻性的目标，我再和你分享一个我在 Facebook 时的例子吧。

2013 年左右，我们从公司开发者的使用数据观察到，由于代码仓的迅速增大，Git 对它的支持有些吃力，比如一些操作的速度越来越慢。于是，我们抽出人力去做调研，克隆了一个 Facebook 的代码仓，并按照当时的增长速度去模拟大量的提交进行测试。

结果发现，再过半年，许多常用的 Git 操作速度都将下降到不可接受的程度。于是，我们团队专门成立项目组来解决这个代码仓将会出现的性能问题。最终，我们在问题还没发生前就

把它解决掉了。这种具有前瞻性的技术目标，确保了公司的业务能够持续发展，给公司和团队带来的好处显而易见。

## 关于技术目标的设定，有两个常见问题：

一是，业务目标和技术目标的时间占比应该是多少？从我个人的经验看，80% 是一个合适的点。

二是，技术目标要不要立项？在我看来，这要视公司情况而定。如果你的主管对技术目标认可，那最好能够单独立项；否则，就把技术目标合并到业务目标中。比如，要实现某一个业务，我们必须重构某一个组件。这里，重构组件就是一个技术目标，无论采用哪一种方式，我们都需要持续关注技术目标。

## 第二步：目标管理

目标管理的第一步就是制定计划。这里，我先给出一个有效设定目标的原则：SMART 原则。SMART 是 Specific（具体）、Measurable（可衡量）、Attainable（可达成）、Relevant（与主目标有相关性）和 Time-bound（有明确的截止日期）这 5 个英文单词首字母的缩写。

一个目标可能被定义为“今年下半年实现用户讨论功能”，但很明显这个定义不够清晰。而另外一种定义方法是“今年 12 月 31 号之前，实现用户讨论功能模块，在主页以及至少一个其他页面使用，并且用户使用率大于 10%”。

可以清楚看到，第二种定义方式更具体，团队成员有更明确的方向，能专门针对具体的模块使用场景和使用率努力。并且，具体的数字和截止日期，可以帮我们更容易跟踪项目进展。

很明显，第二种定义方式正是使用了 SMART 原则中的具体（S）、可衡量（M）和有明确截止日期（S）三个原则。另外，这个任务还要是可达成（A）的，也就是既要有挑战性又可以通过努力达成，这样团队工作起来才会更有劲头。最重要的是，与主目标的相关性（R）是前提，只有这样的任务才有价值。

总结来说，SMART 有更明确、具体的目标，利于员工更加明确、高效地工作，完成与公司目标更加对齐的业绩，也为管理者实施绩效考核提供了目标和标准。网络上有很多 SMART 原则的相关资料，比如 [SMART 原则以及实际案例](#)这篇就还不错。



除了 SMART 原则，**OKR 是一个很有用的目标管理工具。**

我们经常会听到领导者（Leader）和管理者（Manager）这两个概念，但你有注意过它们的区别吗？两者经常混用，但实际上有一个本质区别：领导者告诉团队需要去哪里，而管理者告诉团队如何去到那里。

在我看来，**每一个管理者都应该努力成为一个领导者**，给团队目标，让团队成员自己找到达成目标的方法。而，OKR 正是帮助管理者做到这一点的工具。

其中，O 表示目标，是鼓舞人心的目标，可以使每个人保持一致和受到启发；KR 则表示关键结果，是达成目标需要注意的度量。在 Google 使用后，最近几年 OKR 很流行，效果的确很好。OKR 的内容比较多，如果你想要系统了解并在团队落地的话，推荐你阅读《[@黄勇的 OKR 实战笔记](#)》这个专栏。

这里，我想强调一下，**用好 OKR 的两个关键点。**

第一，使用 OKR 最重要的目的是，让全公司对齐目标，所以在实施 OKR 时，我们要随时留意这个目的。可以说，这是执行 OKR 最关键的原则。基于这个原则，我们可以扩展出很多实际操作。比如公司级别定义一个 O，每个团队和个人根据实际情况制定自己的 O 和 KR；所有人的 OKR 都透明，全公司可见，同时定时做回顾、调整和复盘。

第二，OKR 不是一个 HR 工具，不是绩效管理方法。绩效管理方法一般包括目标、度量和考核（激励 / 惩罚），与之相比，OKR 只包括目标和度量，是一个目标导向工具。

OKR 之所以在目标对齐上有很大作用，是因为团队成员可以发挥主观能动性，自己制定与公司目标一致的 KR。而一旦 OKR 跟绩效挂钩，团队成员承担风险的意愿和内驱力就会大大减弱，倾向于制定更容易实现的 KR，从而失去了目标导向的意义。

所以，OKR 不要跟绩效挂钩。那么问题来了，**使用 OKR 之后怎样进行绩效考评呢**？实际上，这也很简单，只要评估团队成员具体完成的工作对公司的贡献度即可，甚至可以 OKR 和 KPI 并存。比如，公司高层可以使用 KPI，用数字衡量绩效，而全公司都使用 OKR 进行目标对齐。

### **第三步：任务执行**

在具体的任务执行上，作为技术管理者，我们应该从人、流程和工具上来提高研发效能，高效达成业务目标和技术目标。

## 关于人：最关键的是调动起主观意愿

首先，把团队成员的利益统一起来，才会激发他们的主观能动性，自己想办法去达成目标。典型的例子是，为了消除职能竖井，采用全栈开发方式（你可以再回顾下 [第 8 篇文章](#) 中的相关内容）。

**统一团队利益的主要方法是，采用康威定律来组织团队结构**，使其与产品结构相吻合，让产品的成功成为团队一致的目标。

比如 Facebook，针对产品和功能，一般组织 10 人左右大小的团队，里面包括了前后端开发者、设计师、产品经理、数据科学家等。所有人的目标，都是如何把自己团队的功能、产品做好。小团队之间松耦合，有比较高的自主权，不同团队间主要通过目标的一致性来进行协调。这种方式不但使得大家的力往一处使，而且灵活机动、出产品很快。

把这种小分队的方法用到极致的是 Spotify 公司。他们在产品层面把各个功能模块隔绝开，某个功能出现问题，不影响其他功能正常使用。每个功能由 8 人左右的自主运营小组负责，称之为 Squad。Squad 的主管负责确认、沟通团队需要解决的问题，以及解决这些问题的意义，而团队成员的职责是自己决定如何解决这个问题，自主性非常强。

另外需要注意的是，**在把组织架构和产品对应起来的时候，还有一个重要原则：DRY**。对个人开发者说，DRY 是不要重复自己；而对公司或者团队来说，DRY 就是要建设针对基础设施、共享业务的平台，以避免重复造轮子。

以 Facebook 为例，Infrastructure 团队（基础平台团队）人数众多，话语权大，对公司的业务发展至关重要（我之前所在的内部工具团队，就是 Infrastructure 团队的一部分）。各项业务之所以能够快速开发、验证、上线、迭代，并实现高可用、高并发支持上亿日活用户，都跟 Infrastructure 团队的工作密不可分。

除了基础平台外，DRY 在业务方面最主要的表现是业务中台，也是最近非常火的话题。

调动人员的意愿，除了组织架构方面外，绩效管理和企业文化也很重要，我会在后面几篇文章中与你详细介绍这些内容。

## 关于流程：选择合适的方法论、原则、实践

关于流程需要注意的是，寻找符合软件开发行业特性的方法，并根据团队情况不断优化。**具体来说，我推荐使用先从全局的端到端流程入手，寻找系统瓶颈，然后再集中精力解决瓶颈，完成一轮优化。**

这样可以从全局出发，避免以偏概全，能够最高效地使用团队的人力、物力。在优化过程中，我们要尽量采用数据驱动的方式，用数据来寻找问题，并通过数据的比较来检查改良措施的有效性。

针对流程的优化，你可以再回顾下 [🔗第 4 篇文章](#) 中的相关内容；而关于效能度量，你可以再回顾下第 [🔗2](#) 和第 [🔗3](#) 篇文章中的相关内容。

另外，要提高团队的研发效能，作为团队技术负责人，需要保持技术判断力，包括技术选型的能力以及方法论的选择能力。你可以参考我在前三个模块中，给出的流程优化、团队工程实践及个人工程实践的一些高效方法论和实践。

## 关于工具：根据实践进行选择

完成了人、流程的工作之后，工具的选择就比较容易了：让团队**根据方法论选择适用的工具即可**。在前面的文章中，我对各个方法论的适用工具都做过详细介绍，你可以再回顾下相关内容。

这里，我再给出**团队选用工具的两个原则**吧。

第一个原则是，选择工具时要根据场景的复杂程度选择自建还是使用第三方服务 / 工具。对简单、单一场景，我推荐使用开源工具；而对复杂的系统和流程，可以考虑使用一些付费工具，比如 SaaS 产品，因为术业有专攻，这样比自建工具性价比更高。

通常情况下，针对小型创业公司，很多 SaaS 产品的价格不算太高，有些甚至免费。作为技术管理者，我们要考虑投入产出比，让开发人员把时间花在业务上可能才最合适。在硅谷，小公司使用付费软件服务的现象也很普遍，国内公司也慢慢有这个趋势了。

第二个原则是，工具虽然重要，但背后的方法论和原则更重要。比如 OKR，如果使用得当，使用一套简单的 wiki 系统就可以做起来；但如果概念不清、原则不对的话，即使引入

专门的 OKR 工具效果也不会好。所以，作为技术管理者，我们一定要花时间了解工具背后的原则。

## 小结

今天，我通过寻找目标、目标管理，以及如何执行这三步与你介绍了一些管理方法。

首先，最重要的一点是，技术管理者需要同时关注业务目标和技术目标。只有这样，才能够让团队持续发展。如果今天这篇文章你只能记住一个观点的话，我希望能是“业务目标和技术目标两手抓”。

在目标管理方面，OKR 是帮助团队对齐方向的不错工具。不过，我们使用 OKR 时一定注意，目的是对齐目标，与绩效挂钩的效果会大打折扣。

在具体的任务执行方面，从人、流程、工具三个方面入手，即想办法调动人的主观能动性，从流程全局入手把时间花在最需要优化的地方，以及根据具体方法论和场景复杂度选择工具。

最后，我觉得每一个技术人员，都应该花些时间去了解管理，原因包括两点：

对公司、团队的管理措施有更清晰的理解，可以帮助我们更高效的、有的放矢的工作；

管理是职业发展的一个方向，了解一些管理可以帮你尽早弄清楚这条路是否适合自己。

## 思考题

Facebook 提前预测并解决 Git 性能问题的例子中，我们最后使用 Mercurial 代替了 Git。你知道这其中的原因是什么吗？

感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎你把这篇文章分享给更多的朋友一起阅读。我们下期再见！



# 研发效率破局之道

Facebook 研发效率工作法

葛俊

前 Facebook 内部工具团队 Tech Lead



新版升级：点击「👤 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 30 | 答疑篇：关于价值导向和沟通

## 精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。