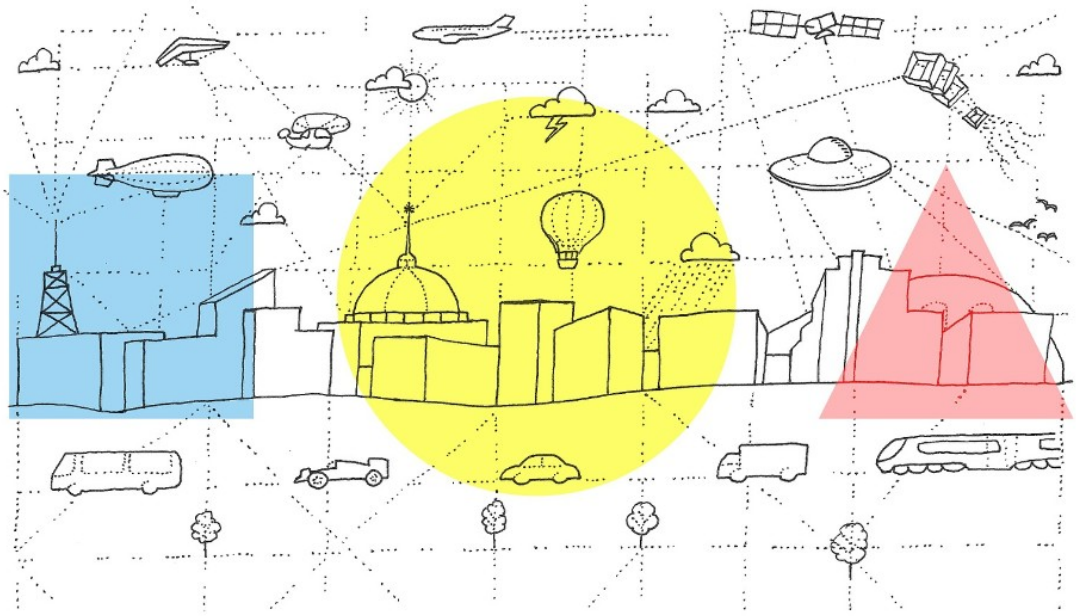
35 | Facebook工程师文化实践三大支柱之二拥有信息和权限

2019-11-13 葛俊



讲述：葛俊

时长 16:17 大小 14.92M

你好，我是葛俊。今天，我来和你聊聊，Facebook 工程师文化的第二大支柱：拥有信息和权限。

在上一篇文章中，我与你介绍了 Facebook 通过在入职、日常工作、转岗中的一些实践，来提高开发人员对所做工作的兴趣。有了兴趣之后，接下来就是提供空间让大家能够施展拳脚，在自己感兴趣的工作中自由发挥、充分发挥，也就是 Facebook 工程师文化的第二大支柱了。具体来说，第二个支柱包括让员工拥有信息和拥有权限两个子原则。

在我看来，Facebook 在这两个原则上的实践，对国内公司和团队有很大的参考价值。为什么这么说呢？因为**我接触到的国内公司管理都相对保守**，可以从与 Facebook 的具体实践对比中汲取经验，提升公司整体的研发效能。

在拥有信息方面，国内公司一般过于严格。信息的不透明，会降低员工的主人翁意识，并会因为信息不通畅导致工作效率和有效性的下降。所以，在我看来，国内公司还可以在信息方面做得更透明些。

在权限和信任方面，我接触到的国内公司，大都偏向于不信任的管理方法。这种方法难以让员工有主人翁的感觉，导致公司和员工互不信任。在这种情况下，又怎么期望员工能积极地为公司创造价值呢？

我始终认为，国内的开发人员和硅谷的开发人员一样，都有很强的创造性，能力也没什么大的差别；只要能给他们创造好的环境，提供信息并给予信任，就可以更好地激发他们的内驱力，从而提高公司和团队的研发效能。

接下来，我们就具体看看 Facebook 是怎么做的吧。

让员工拥有信息

作为知识工作者，拥有信息，是我们能够做出正确判断的一个必要条件。但信息的开放具有两面性，如果暴露过多的敏感信息，势必会对公司造成些负面影响。

面对这样一把双刃剑，Facebook 的态度是考虑风险，但并不是一味地避免风险，而是权衡公开信息的利弊，在确保基本安全的前提下尽量实现信息透明化。

接下来，我们一起看三个具体的案例吧。

案例 1：代码的共享

在 Facebook，几乎所有的代码都是全员共享，开发人员可以非常方便地查看这些代码。这，跟国内很多公司的情况差别非常大。

我在和很多国内公司的管理层谈到 Phabricator 时，对方的第一个问题通常就是如何管控代码仓的权限。比如，如何让前端开发人员看不到后端的代码，如何让后端开发人员看不到算法的代码，等等。

但其实，**代码的共享可以在开发、调测时节省很多沟通成本，大幅提高效率**。比如，某个前端开发人员发现后台 API 返回值不符合预期，就可以自己去查看对应的代码，即使他没能

自己找到问题，也可以在和后端开发人员沟通时提供有价值的信息。

但如果他看不到后端代码的话，就只能直接求助于后端开发人员，一方面会浪费后端开发人员的时间（有可能这个问题前端人员就可以解决），另一方面双方的沟通效率也会因为信息不对称而大打折扣。

国内公司大多对代码权限控制得非常严格，这跟国内的 IP 保护不力有直接关系。不过在我看来，国内的代码管理还是太过严格，甚至有些不分青红皂白就对所有代码进行严格控制的情况。这，不但会降低开发人员的工作效率，还会降低团队之间的信任。

所以综合考虑，**我们应该在可能的范围内，尽量放宽对代码的管控，对于那些即使泄露出去也没有太大关系的代码，要放松管控。**这样做风险不大，但收益却可能会很大。

案例 2：看板（Dashboard）

在 Facebook，很多信息都会展示在公司墙上的显示屏上，包括很多业务的实时监控数据，甚至还有少量在其他公司认为比较敏感而不愿意显示给员工的信息，比如某些敏感服务的实时数据。此外，很多团队还会制作自己团队的看板去显示需要关注的信息。

一些敏感数据泄露出去，的确会给公司造成些许负面影响，而且确实曾经出现过信息泄露。所以，有员工就问扎克伯格，**为什么要暴露这些信息**。

扎克伯格的回答是：这些信息泄露确实会造成少量负面的影响，但这些信息可以极大地方便开发人员的日常工作，能够更好给用户提供价值，比如看板上的某一条信息可能在你做某个决策时起到关键作用，所以权衡利弊，我们决定开放这些信息。

这个回答，恰恰佐证了公司认同信息对激发开发人员研发效能的巨大作用。

关于信息开放，在国内比较容易落地的部分是，把可以显示的信息用看板显示出来。比如，线上服务使用情况可以拉近员工与用户的距离，让员工更直观地感受到业务的价值，从而提高工作积极性；又比如，当次发布进展的相关数据，可以让开发人员对当前进展更有掌控感，对交付更有紧迫感。

我们在落地看板展示时，需要注意两个问题：

一是，不要做给外人看，而是应该用来激励内部员工；

二是，不要展示个人负面排名，比如 Bug 数，这种公开的负向激励方式，容易降低团队成员的积极性。

案例 3：使用 Wiki 来记录信息

Facebook 在公司内部大量使用 Wiki 来记录信息，很多对流程要求不严格的信息都用 Wiki 来记录，包括部分设计文档、团队成员列表、新员工入职手册、个人笔记，甚至有些团队的 OKR 也在上面。

这样一来，我们在日常工作中遇到问题时，首先就是到内部 Wiki 上查找，而且一般都能找到有用的信息。这就使得绝大部分员工愿意在 Wiki 上添加信息，包括一些笔记、心得等。于是，形成了一个良性循环，Wiki 系统的内容越来越丰富，对开发人员的帮助越来越大。

但，我见到的很多国内公司，很多信息都是口口相传，或者是藏到某些文档里，需要花费大量的时间在查找信息上。

在我看来，在公司内部用内部公开的 Wiki 来记录信息，风险比较小，而且可以大幅提升效能，所以比较容易落地。**作为管理者，我们可以考虑建立一个全公司的 Wiki，并采取措施鼓励团队进行知识方面的共享。**

这里的**一个技巧是**，把权限设置为默认公开，对需要进行权限控制的内容再特殊处理收紧权限。这跟默认保密，有需要再特殊处理放开权限的操作相比，看似相差不大，却是在传递支持信息开放的基本态度，效果非常不错。你也可以试试这个方法。

让员工拥有权限

在能做感兴趣的事，又拥有了信息之后，最后一条就是让员工能够拥有权限去做事儿。这里，我也与你分享三个具体的案例吧。

案例 1：对于商业软件，先购买再获取授权

Facebook 对商业软件的态度是，相信每个开发人员会从公司利益出发，权衡一个软件是否需要购买，如果需要可以先斩后奏。具体的案例，你可以回顾下第 1 篇文章中的相关内容。

其实，国内有些公司也已经采用类似方式了，员工可以在一定数额下自行决定是否购买。我在跟这些公司的员工聊到这个政策时，能够明显感觉到他们的开心和对公司的认同感。

一般情况下，员工不会擅自做离谱的决定去购买某一款软件，所以这个操作比较安全，你也可以试试。

案例 2：鼓励代码上的互相贡献

在 Facebook，除了几乎所有的代码对全员公开外，公司还鼓励开发人员互相修改和提高对方的代码。比如，前端开发人员发现后端 API 返回结果不符合预期时，如果他能识别出后端代码的问题，便可以去直接去修复，然后提交一个 PR 给后端开发人员，对方接受之后即可入库。

又比如，你发现某一个产品的某一个功能有 Bug，也可以直接修改，只要通过 PR 和测试，就可以上线。

这在内部工具方面尤其明显，也是 Facebook 的内部工具超级强大的一个重要原因。因为内部工具代码上线的风险比较小，所以很多人在看到工具的问题时，会上手去进行提高，甚至自己开发一些新工具。

一个具体的例子是，Phabricator 在开源之前，很多功能和改进都是非内部工具团队的开发者贡献的（你可以再回顾下第 15 篇文章中的相关内容）。

如果你要落地这个实践的话，我有两个建议：

一是，先从风险小的代码试点，鼓励互相贡献；

二是，对这种非本职却对公司有贡献的工作，应该体现在绩效考评中。

案例 3：提供宽松的容错环境

Facebook 对待事故的态度是，更关注吸取经验教训，以及将来如何预防，而不是追责。这样的容错环境，能够让开发人员放开手脚，最大限度地释放自己的创造力和潜能，保证公司的持续快速发展。

在第 33 篇文章中，我提到在持续进步方面，Facebook 提供的宽松容错环境，只要不是故意犯错或者重复犯相同的错，一般不会被追责。Move Fast and Break Things，就是一个有力的证明。

我刚进入 Facebook 时，曾见过一个事故：有一个开发人员错误操作，把公司最大的 SVN 代码仓给搞挂了；公司找来两个 SVN 专家，花了一天半的时间才把整个代码仓恢复。这给公司造成了很大损失，但这个开发人员并没有因此被裁掉。还有些人因为误操作造成线上事故，影响了 Facebook 网站的功能，但只要不是故意犯错或者重复相同的错，也基本不会被追责。

如果你想落地宽松的容错环境这个实践的话，我推荐你尝试 Facebook 的 SEV 复盘。我在国内公司实践过 SEV，效果不错。

你可以再回顾下第 4和第 20篇文章中关于 SEV 的相关内容，以及出现问题之后，到底应不应该追责、应该怎样追责的问题。

“软件先购买再获取授权”“鼓励代码上的互相贡献”，以及“容错文化”三个实践，我都在国内一个 50 开发人员左右的创业公司引入过，效果都还不错。引入时，注意些上面提到的问题就可以了。

Facebook 之外的落地经验

接下来，我再与你分享两个，我在其他公司采取的让团队成员拥有信息和权限的实践吧。

案例 1：促进信息上下流通的实践

在 Facebook，每周五都会举办一个内部的 Q&A，员工可以通过现场或者线上提问的方式，问扎克伯格以及高管们任何问题。扎克伯格和其他高管都非常坦诚，尽量把可以公开的信息都坦诚相告，对公司信息上下流通的效果非常好。

离开 Facebook 之后，我在一个团队内，采用类似 Facebook 的方式进行了 Q&A，使用尽量坦诚的、开放的问答形式，不说套话。这个团队的特点是，执行力很强，但氛围有些沉闷。在坚持了两三次 Q&A 之后，气氛逐渐活跃，大家逐步敢问一些敏感的问题了。

事实证明，开发人员的思维非常活跃，也都很喜欢用活跃的方式进行沟通，以获取到更多的信息。这样做的好处是，加强了员工对公司、对团队的主人翁的精神，进而提高整体的研发效能。

落实 Q&A 还有一个小窍门是，大家可以用匿名写纸条的方式去提问。这个方法，在国内效果更好，很多人更愿意用这种方式去问一些敏感的问题。

案例 2：信息共享的相关实践

这个实践是，我在一个创业公司，采用 Wiki 的方式来推动全公司范围内的信息共享，内容权限默认设置为公开，只有特殊情况才可以设置为只对某些人公开。

落地这个实践，的确是一个长期的过程。我见到，国内很多开发人员，都不太习惯做分享，所以**我推行 Wiki 时的两个方法，是坚持和以身作则。**

在推广 Wiki 的第一个月，90% 的新内容都是我写的！后来，大家逐渐看到了这种信息共享的好处，也看到了我的坚持，就开始主动向 Wiki 中增加内容。半年后，我的贡献降到了 10%。

小结

今天，我与你分享了 Facebook 工程师文化落地实践的第二大原则，让员工拥有信息和权限。

因为软件开发，是知识性工作，所以拥有信息是高效开发的前提。Facebook 在这方面的实践，包括代码共享、看板和公司范围内公开的 Wiki 等。

而让员工拥有权限和信任，可以让员工以主人翁的感受去施展拳脚，最大限度地激发其内驱力。这方面，Facebook 的实践包括，信任员工进行一定额度下的软件自行采购、鼓励互相贡献代码，以及建设宽松的容错环境等。

从我个人在国内推行文化的实践经验来看，国内大多数软件公司，是适合让员工拥有更多的信息和权限的。而且我也看到，国内越来越多的公司意识到了这一点，在逐渐放开信息和权限，从而提高员工的内驱力和团队的研发效能。

我相信，我们会逐步找到更合适自己环境的平衡点，打造高研发效能的公司。

思考题

信息开放的确会导致信息泄露。今年 10 月份，Facebook 出现了一次 Q&A 被录音并泄露的事件，你知道扎克伯格是怎么处理的吗？你还能想到更好的处理方式吗？

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！

上一篇　34 | Facebook工程师文化实践三大支柱之一做感兴趣的事

下一篇　36 | Facebook工程师文化实践三大支柱之三绩效调节

精选留言 (3)

写留言

石马

2019-11-15

Facebook内部的知识管理系统都用的哪些呢？我所在的团队对知识传承这块做得比较少，很多知识靠口口想传的多，用过media wiki，不太好用，主要内容输出上不太友好。

桃子-夏勇杰
2019-11-14

提供太多的信息，感觉也会让工程师信息过载。在一些特定的行业，例如，金融，对于权限管理还是非常严格的。如何在这样的环境下，开放必要的和有助于工程师协作的信息给大家，老师有这方面的经验么？

Geek_35ccc4
2019-11-13

您好，目前我们公司在推进代码加密，防止代码外泄，不知道这块像Facebook这些顶级的硅谷公司，是怎么处理的呢，基于什么考虑？期待解读

展开

作者回复: 主要采取放开权限+监控是否外泄的方式。不过因为他们的用户粘性才是最大的竞争力，代码反而不是最关键的。