

09 | 信息流通：让团队高效协同，让产品准确击中目标

2019-09-11 葛俊

研发效率破局之道

[进入课程 >](#)



讲述：葛俊

时长 17:12 大小 15.76M



你好，我是葛俊。今天，我来和你聊聊团队的信息流通问题。

研发过程中的信息流通，指的是各种跟研发相关的信息在工具、团队成员之间的流动。这些信息大到公司战略，小到 Bug ID，是一个团队达成共识、高效工作的重要因素。比如，你是否也曾遇到下面这些研发过程中的不顺畅问题呢？

最终产品背离用户需求。我曾提到，一个常见的低效能问题是，研发团队生产出来的产品与最初的产品设计差别很大，甚至需要完全返工。

前后端沟通不顺畅。后端修改 API 阻塞了前端的工作，或者后端实现了新的 API 前端不知道，继续使用旧的 API 造成浪费。

信息孤岛。在公司内部找信息反而比在互联网上找信息还要难，需要到处找人问，还不一定能问到。

信息难以溯源。团队成员不知道怎么寻找问题的源头，也不知道软件包发布了哪些功能。

工作干扰大。开发工作常被实时聊天工具、电话等打断，刚整理好的思路又得重新梳理。

实际上，这些问题都是信息流通不顺畅造成的。那么今天，我们就来看看如何做到信息流通，从而让开发人员更顺畅地生产出高价值的产品。总结来说，就是要从以下三个方面入手：

首先，从“人”入手，建设共享文化，鼓励共享行为，使信息共享与团队成员利益一致，从而让大家愿意共享。

然后，在流程和工具方面，针对与研发相关的信息，设计并实现对应代码、文档的共享，以及信息在流水线上的自动化流动。

最后，在沟通技巧上下功夫，掌握高效沟通的原则，根据场景选择合适的沟通方式和工具。

接下来，我们分别看看如何实施吧。

团队成员愿意共享是有效沟通的前提

人是首要的，不解决人的意愿问题，流程、工具再好，也用不好甚至用不起来。所以，让团队成员愿意共享，是有效沟通的前提。那，怎么才能调动起团队成员共享消息的意愿呢？

首先，要让团队成员了解信息的重要性。软件开发本来就需要多人协作，沟通的重要性不言而喻。但，有相当一部分开发者，尤其是初级开发者，因为工作经验欠缺等原因不愿意去和别人沟通，甚至意识不到信息流通的重要性。

所以，作为开发人员，我们要主动去克服不愿意沟通的倾向，比如在对需求有疑问时提醒自己，如果现在花一小时的时间去沟通，很可能会节省自己后面三天的时间；而作为团队管理者，我们更要在团队内强调沟通的重要性。

但，了解的信息沟通的重要性，只是实现顺畅沟通的基础。更重要的是，我们要在**团队内部建设机制，来鼓励共享的行为，从而形成共享的文化。**

简单来说，就是让信息共享和每个成员的利益紧密联系起来。这样做，可以帮助我们解决最终产品背离用户需求，以及前后端沟通不畅导致返工等低效能问题。这一类问题的共同点在于，负责整个产品或者 API 的成员没有紧密合作。在按职能划分的团队，尤其容易出现。因为职能部门的成员，在职能线上向上汇报，最关注的往往是职能部分的完成情况，而不是整个产品的进展。

一个比较有效的解决办法是，**按照产品或者功能划分团队，让团队成员直接对产品或者功能负责**，Facebook 和 Spotify 等很多高效能公司使用的就是这种方法。

一个团队负责一个产品或者功能，构成上一般不超过 15 个人，包括产品经理、UI 设计、数据科学家、前后端开发者。团队的工作重心就是把产品和功能做好，这也就意味着只有把产品和功能做好了，每个人的绩效才会好。

在这种情况下，沟通就和自己的利益紧密相关，所以大家会通过主动沟通去推动产品和功能的开发。比如，产品设计得慢了，开发人员就会去催促询问，看有没有自己能够帮得上忙的地方。

但是，改变公司的组织架构往往阻力大、推进缓慢，我们还是先立足现状，去**解决按照职能划分团队的沟通问题**。一个比较有效的办法是，使用虚拟团队。我以前在微软的 Office 团队第一次见到这种方式，后来又把它用在了其他公司，效果非常好。

具体的办法是，**给每个功能设计一个虚拟团队**，包括产品、设计，以及相关的开发人员。之所以说是虚拟团队，是因为它并不存在于公司的正式组织架构里。我们明确这个团队的职责，就是要做好某个功能。当时我采用的办法也很简单，就是给每个虚拟团队建立一个专门的聊天群，来沟通这个功能的相关问题。而我每周会收集每个虚拟团队的工作进展情况，以此来推动产品维度的进展。

说到这儿，你可能会有一个疑问，**我对非开发部门没有掌控权怎么办**？事实上，我在推动这个虚拟团队的时候，只负责管理前后端开发人员，对产品等团队没有管理权。但因为这个虚拟团队，和其他团队不存在利益冲突，而且流程很轻，所以并没有任何反对声音。而在虚拟团队中，我负责管理的前后端开发人员会主动发力，促进整个虚拟团队的顺畅沟通，效果非常好。

有了主动沟通的意愿，我们就可以开展下一步工作了，即设计流程和使用工具，推动研发信息的高效沟通。

设计流程和使用工具，推动研发信息高效沟通

这一步的关键在于确认研发流程中的重要信息，然后针对性地设计合适的流程，并选用恰当的工具。

在我看来，对提高研发效能起到关键作用的信息，主要分为 4 种。

第 1 种信息是，战略目标相关的信息

这一类信息的处理原则是尽量公开。只有当团队成员清楚公司以及团队目标时，才能更容易地把自己的目标与之对齐。或许，这就是 OKR 最近几年特别流行的原因吧。如果你想深入了解 OKR，并在团队中落地，可以参考我的朋友黄勇开设的《黄勇的 OKR 实战笔记》课程。

Facebook 每年都会召开一个员工大会，详细列举公司的战略目标；每周还会有一个 Q&A 会议，每个员工都可以参加，马克·扎克伯格（Mark Zuckerberg）会亲自出席，并回答员工提出的各种问题。这些举措，都促进了员工对公司战略和目标的深入理解。

第 2 种信息是，代码相关的信息

这一类信息的处理原则也是尽量公开。代码是最直接的参考，是最实时的文档。Facebook 基本所有的代码仓，都对全部开发人员公开。更进一步地，我们不但可以阅读其他团队的代码，还可以主动去修改、提高他们的代码，只要修改得当，发出的 PR 就会被接受。

在国内，由于 IP 保护不力等客观原因，绝大部分公司不愿意把代码对所有员工公开，这也可以理解。不过在这种情况下，我还是建议在不泄露核心机密、不影响核心业务的前提下，尽量扩大代码公开的范围以及受众人群。

选择共享代码的工具和方式，基本原则是方便查找，并在进行开发工作的主要入口（比如 IDE、命令行工具、Web 浏览器）提供接口。

以 Facebook 为例，他们使用 Phabricator 的 Diffusion 子功能，方便团队成员在网页上浏览和查找代码仓的历史；在代码搜索方面，他们自研了一个内部工具，对主要代码仓的代码进行几乎实时的索引，开发人员通过网页浏览器、命令行，以及 IDE 的插件使用代码搜索功能。

高效的代码浏览和搜索对研发顺畅很重要，推荐你加大在这方面的投入。

第 3 种信息是，研发过程中用到的各种文档

这些文档，包括产品设计文档、开发设计文档、测试文档、部署流程文档等。确保这一类信息的高效流通，比较有效的原则是，通过统一的工具，方便大家添加、修改、查询这些文档。

在 Facebook，每个产品团队可以选择自己的文档管理方式。总的来说，绝大部分团队主要使用公司统一的 Wiki 来进行松散的文档管理，既方便添加、修改，也方便搜索。而对于那些比较正式的文档，有些团队会使用类似 Quip 的共享文档系统进行管理。

关于文档管理，有两点值得强调：

第一，文档的管理流程，由每个小的功能团队决定。就比如，我刚才提到的 10 个人左右的小团队，因为他们的共同目标是尽快开发好产品，所以会主动寻找一种适合自己团队的文档管理方式和工具。

第二，绝大部分 Wiki 和 Quip 管理的文档都是全公司公开的，所以团队之间也可以很方便地找到其他团队的相关文档，避免信息孤岛的问题。

第 4 种信息是，各种标识信息

整个研发流程中，在各种工具之间流动着多种标识信息，包括任务工单、代码提交号、版本号、代码审查 ID、测试用例 ID、Bug ID 等。在我看来，**管理这一类信息的有效方法是，各种工具通过提供 API，做到服务化，形成工具之间的网状连接**，以方便开发人员在工具上快速拿到需要的各种信息。

接下来，我们看一个热修复的具体案例。开发人员写好热修复的代码提交后，需要去发布工具网站填写这个提交的 Commit ID，申请进行热修复。热修复发布工具根据 Commit ID，找到对应的代码审查 ID 以及 Bug ID，自动显示到这个网页上。同时，这个工具还会自动找到这个热修复提交所依赖的、还没有上生产的其他提交，并显示出它们的信息。

有了这些信息，开发人员就可以方便地检查、确认自己的这个热修复可以提交，然后点击确认，正式申请热修复。最后，热修复发布工具自动 cherry-pick 这些提交，把它们都部署到一个热修复环境上进行验证。

整个流程中涉及的服务和工具包括：代码仓服务、代码审查服务、Bug 管理系统、测试验证服务、发布上线服务等。这些工具具有信息高度互通以及自动化的特点，使得开发人员和

运维人员能够以最快的速度拿到各种信息，避免繁琐而且容易出错的手工操作，从而尽快部署热修复。

这也就解决了信息溯源难的问题。

通过对研发流程中重要信息的管理，以及对应的高效沟通方式，我们就实现了研发信息流通的顺畅性，从而实现了团队的高效协作。最后，我们再来看看在具体的工作中，团队成员之间有哪些沟通方式和技巧。

沟通方式技巧

高效沟通是个很大的话题，涉及方式、技巧等内容。落到研发团队沟通的具体场景中，我认为高效沟通的首要原则就是，**根据沟通需要达成的任务的实时性、方便追溯性，以及对别人的干扰程度，选择合适的沟通工具。**

面对面、电话、实时聊天工具、邮件、具体任务工具中讨论等不同的沟通方式，在实时性、方便追溯性、对他人干扰程度等方面各不相同。我将其总结为了一幅图，供你参考。

	实时性	方便追溯性	对他人干扰程度
面对面	最好	无	大
电话、IM	好	差	大
邮件	中	中	小
任务工具的讨论功能	中	好	小

图 1 各种沟通方式的特点

可以看到，不同的沟通方式之间差别很大，我们应该根据具体场景去选择不同的沟通工具。但在现实工作中，有一个不太好的趋势，就是大家都倾向于追求沟通的实时性，大多使用即时聊天工具和电话。这个情况在国内尤其严重，很多公司完全使用即时聊天工具，比如微信、钉钉等，基本上放弃了邮件等其他方式。

使用这种方法，最主要的好处实时性好，因为每个人都希望自己的问题能够马上得到反馈。但，短暂的好处却带来了长远的利益损失，主要表现在以下两个方面。

问题难以追踪。在聊天群里讨论问题时，问题只要一多马上就乱了，很难找到之前的相关内容。

对他人干扰巨大。一个极端情况是，我见过一个公司，有数据显示开发人员平均每 8 分钟就会被打断一次。可想而知，这种情况下的开发效率自然很低。

解决这个问题的办法很简单，就是针对不同的情况，要求大家使用合适的工具去沟通，并逐渐形成习惯。

具体来说，首先，**对某个任务进行讨论时**，最好是直接使用任务工具中的讨论功能，同时通过 @ 的方式来告知相关人员。而一般的工具都可以配置出现 @ 时，自动发邮件通知对方。所以，如果不是紧急任务，都可以采用这种方式。

其次，在询问问题的时候，可以分以下 3 种情况，选择沟通方式和工具：

如果问题不紧急，尽量使用邮件，避免使用即时聊天工具和电话。为了确保大家能够及时回复邮件，可以在团队内制定一个规定，要求检查邮件的频率，比如说每天至少检查一次或者两次。

如果问题紧急，则直接使用即时聊天工具、电话。

如果要讨论的问题比较复杂，或者是非常紧急，则直接选择电话，或者面对面沟通。

在我看来，选择合适的沟通方式及工具，不仅可以大大减少员工间的互相干扰，还可以提高问题的可追溯性，对团队以及个人的研发效能提升帮助非常大。

小结

在今天这篇文章中，我从人、流程、沟通方式和工具这 3 大方面，和你推荐了沟通顺畅，进而提高效能的方法。我们再一起回忆下核心知识点吧。

实现高效沟通首先要解决的，就是团队成员的意愿问题，让他们愿意沟通。我们可以将沟通与团队成员的利益挂钩，在团队内部建设机制，来鼓励共享的行为，从而形成共享的文化。

然后，针对研发流程中流动的各种信息，我们要做好分类，针对性地设计合适的流程，并选用恰当的工具，最大程度地共享给团队成员。比如，公司的代码、战略目标、文档、流程等信息，尽量公开；再比如，使用统一的工具，方便团队成员添加、查询有效信息。

最后，在平时的沟通中，要权衡实时性、可追溯性以及对其他成员的干扰程度。根据场景，选择合适的沟通方式和工具，提升团队和个人的研发效能。

开发者大多在沟通上有所欠缺，有时甚至认识不到沟通的重要性。事实上，信息的缺失，对研发效能影响非常大。试想一下，如果没有搜索引擎的帮助，你写代码的效率会下降多少呢。所以，在信息沟通上多花些时间、精力，对团队和个人效能的提升都大有裨益。

另外，在硅谷的互联网公司，大家都很有意被打断的情况。如果你在没必要的环境下，使用聊天工具或者电话跟同事沟通的话，他会非常生气。

在 Facebook 时，有些同事会直接告诉其他人，如果我戴上耳机，就表示不希望被干扰，除非紧急情况，否则不要找我当面沟通问题。虽说这种做法有些极端，但我非常希望国内公司也能够尝试类似的工作方式，因为它能让大家安心开发，在高效产出的同时，更多地享受心流的快乐。

思考题

在工作中，你见到的信息沟通的最大问题是什么，在今天的文章中能找到合适的解决方法吗？如果没有找到，你还有什么建议的解决方法吗？

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 08 | DevOps、SRE的共性：应用全栈思路打通开发和运维

下一篇 10 | 答疑篇：反对996并不是反对奋斗

精选留言 (9)

写留言



旭东

2019-09-12

心流 对于大部分公司都是可遇而不可求

展开

作者回复：是啊，如果环境不允许，只能在自己能控制的范围内提高。比如改变手机使用习惯。



1



小名叫大明

2019-09-11

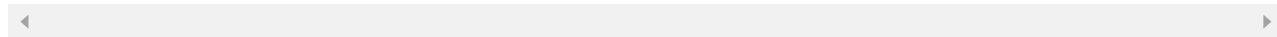
作者讲的信息流通，沟通，共享等一系列的问题我基本上都吃过亏，也见过了好的。

希望自己以后可以做到。

这个人做到也许很难，一个团队如此，慢慢就改变了。

展开 ▾

作者回复: 第一条“团队成员愿意共享是有效沟通的前提”，一般需要管理层的支持



刘晓光

2019-09-11

阿里的同学不来说说，每天上千条钉钉信息怎么破？

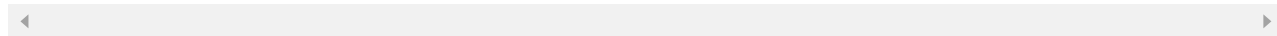
其实兰留给开发人员专注写代码的时间很重要，比如产品和运营都睡了的时候。

展开 ▾

作者回复: > 其实兰留给开发人员专注写代码的时间很重要，比如产品和运营都睡了的时候。

哈哈：)

可以尝试建一些专门的聊天群作为支撑群，让开发轮流在里面oncall，这样每天有一个开发收到大量干扰，其他人可以专注开发多一些。



李双

2019-09-16

学习

展开 ▾



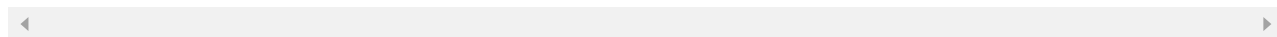
吕哲

2019-09-13

有理论有实践，很有感触。很好的课程！

展开 ▾

作者回复: 多谢支持！有用就好：)





日拱一卒

2019-09-13

在工作中，你见到的信息沟通的最大问题是什么，在今天的文章中能找到合适的解决方法吗？如果没有找到，你还有什么建议的解决方法吗？

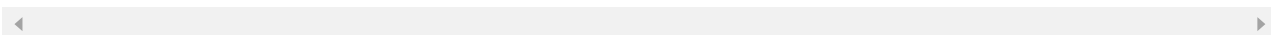
我在平时工作中会遇到的两个和信息沟通相关的问题有：1. 沟通不主动，一个大的项目会涉及到分散在全球各地的多个团队，每个团队在项目之外可能还有自己的诉求，并且对...

展开 ▾

作者回复: 你的思路都很棒。的确，首先是人的问题，人后是流程，然后是工具。

你提到的“传统软件公司”，如果上层没有认识去改变，我们能做的只能是从自己能有控制的部分来做。

关于你讲的“故意下线”，之前我在的一家公司有团队使用过，叫“静默时间”，效果不错。这种操作（以及邮件定时检查两次），关键在于定义一个沟通的协议，让大家逐步接受逐步遵守。自己控制范围内比较好搞，外部团队就只能慢慢来了。



旭东

2019-09-12

一个公司连研发电脑硬件配置好点的意识都没有，还谈什么DevOps...

提高对研发的重视度，多点工程师文化。才能切实提高研发提高效率的可能性，很多领导都是996的想法，搞人海战术，为了自己业绩竟能将加班做文化，加班时长做绩效

展开 ▾



小名叫大明

2019-09-11

不得不说我所在的MT开发团队这方面做得很好，很幸运

作者回复: 赞！！能不能介绍一些你们的实践经验给大家听听？



王军虎

2019-09-11

对他人干扰巨大。一个极端情况是，我见过一个公司，有数据显示开发人员平均每 8 分钟

就会被打断一次。——这不就是说的菊花厂么
展开 ∨

