

左耳朵耗子：疫情下的远程办公，聊聊我的经验和实践

陈皓 InfoQ 2020-01-31

作者 | 陈皓（左耳朵耗子）

疫情下的远程办公，跟你聊聊我的经验和实践。

我于 2016 年成立 MegaEase (<https://megaease.com/zh/>)，从早期 8 个人，直到今天有 20 来个人，我们从一开始到今天都是在远程工作。因为我很喜欢《Rework》这本书，写这本书的公司叫 37signal（现名 basecamp），这家公司在发《Rework》这本书的时候，整个公司只有 16 个人，分布在全世界 8 个城市，这种 Geek 的公司文化很吸引我，所以在我决定创业的时候，我就止不住地想成立这样一家能够远程工作的公司。于是远程工作的公司文化就成为了 MegaEase 的基因。

下面我分享一下我们公司远程工作的文化和其中的一些问题，最后还会分享一个我们的远程工作协议。

我们在早期的时候，8 个员工来自 5 个城市，现在的 20 来个员工来自 8 个城市 2 个国家。虽然我们使用“共享办公室”，但是本质上，我们的整个文化是远程工作的文化。

在 2017-2018 年度，我们公司产品商业化以来，公司早期的 8 个工程师在远程工作的状态下成功支持了得到老罗的跨年演讲活动，以及其它几个客户。一方面验证了用户愿意付费购买我们的产品和服务，另一方面也有一些不错的收入，客单价都在百万左右。还记得当时有几个投资人并不相信我们是一家连办公室都没有的公司，而且 8 个人还分布在 5 个城市，觉得我们是个骗子公司（哈哈）。

在过去的一年，我们通过我们的产品和服务帮助银行、电信、互联网等公司进行了他们系统架构的改造和升级，让复杂和高门槛的分布式技术和架构可以被更多企业所掌握。这说明，远程工作是没有问题的。实际上远程团队、远程工作真的不新鲜，Github 上有个 Repo 维护着一个支持远程工作的公司列表，还有一个跟远程工作相关的 Awesome 索引。

<https://github.com/remotetech/remote-jobs>

<https://github.com/lukasz-madon/awesome-remote-job>

当然，自从我创业以来，我身边就一直有好多不同的声音质疑远程工作。听过他们的理由后，我能够理解他们的疑虑和困惑，管理的确是一件很复杂的事，因为要面对的是极为复杂的人，所以，有这些疑虑也是正常的。下面是我的一些经验和分享。先说宏观管理，再说微观实践。

宏观管理

我发现很多人比较质疑远程工作的原因，更多是对宏观的管理上有疑问。所以，我还是想先说一下宏观管理，这其实并不分远程办公还是集中式办公，如果能够解决好些这管理上的根本问题，远程不远程也就都无所谓了。只不过，这些问题在“远程办公”的场景下更突显罢了。

努力找到好的人

团队管理的头等大事是找人，没有之一。很多人都会跟我说，你的这种远程团队需要很好的人。是的，没错，人很关键。远程团队需要的人的一般需要有这些特质：

能独挡一面的人。这样交给他的事能独立完成，没有路能自己找路，这样可以节省很多管理成本。

沟通能力很强的人。一方面，他能把模糊的事变清楚，另一方面，他能有效地说服他人。不然就会非常扯皮和消耗时间。

能自管理和自驱动的人。不能自管理和自驱的人，会增加大量的管理和教育成本。能自驱动的人，都是对所负责的事情有认同的人。

如果你仔细思考一下，你会发现，这样的人是任何一家公司所渴望的人，和远不远程无关。只不过，如果是远程团队的话，你会被逼着要招到这样的人。

招到这样的人，你团队的执行力会非常的强悍。招不到这样的人，你只能为他们不能自管理和自驱而招“经理”；不能写出好代码而招“测试”；不能很好沟通而招“项目经理”；不能独档一面，而要把好的人安排给他们当“教练”，而好的人则会被累死.....

这个时候，你就需要计算一下了，是花时间精力在教育不好的人，还是花时间精力找好的人？无论远不远程，聪明的管理者都会选择后者。这也就是为什么 Amazon 的 Bezos 会说：“我宁愿面试 50 个人一个人都招不到，我也不愿意降低我的面试标准”。

设定共同的目标和使命

对于远程团队来说因为见不到面，所以缺乏交流和沟通。所以，需要团队里所有人能够对要做的事有一个统一标准的认识。也就是共同的目标和使命的认知。知道要什么、不要什么，知道取舍，知道 trade-off。这些东西都是需要团队一起达成共识的。如果没有这样“Same Picture”的目标和使命，就会出现很多不必要的误解和冲突。另外，因为团队和业务也在迅速发展，所以，也需要不断地调整和沟通。这都需要领导者花费时间统一目标和使命。

老实说，无论远程不远程，一个团队都需要有共同的目标和使命。没有共同的目标，就算是集中在一起办公，也一样没有效率。

倾向使用小团队

因为沟通成本的问题，远程团队更倾向使用小团队，但并不是说小团队会限制整个公司的规模。《人月神话》说过，只有小团队才能驾驭复杂的系统。Amazon 的 Two Pizza Team 文化（团队只能是两张披萨就能喂饱的规模），就是把整个系统拆成“微服务”架构，这样可以导致整体效率的巨大提升。表现在，可以并行开发，专注于一个功能更利于解决复杂问题，可以更容易的运维，可以更容易的规模化.....

我工作的这 20 多年来经历过很多家公司，尤其是创业的这几年来，看过的公司更多了（50+ 以上）。我发现，人数越多的团队，基本上来说就更偏劳动密集型。劳动密集型的一个特征就是，大家整天在想，得整点什么事给这么多人，好让他们忙起来。而人数少的团队，因为人不够，所以每天都在想，什么样的事更重要，什么样的事可以自动化，怎么做更有效率.....小团队和大团队的关注点就这么不一样了，所以做出来的事也就不一样了.....

当然，并不是说劳动密集型有什么问题，就像《软件团队的两种管理方式》一文所说的一样，远程团队更倾向于“电影工作组”式的每个人都是 leader 的知识密集型的团队。

微观实践

在远程工作中，我们需要有很多的微观操作来让大家能够更好地进行远程工作。因为远程工作也有一些问题（但是方法总比问题多，不是吗？）

文档驱动

首先，远程的问题就是沟通不方便。集中办公的话，一群人可以在白板上进行讨论，然而远程工作这个事就变成很复杂了。所以，当要讨论什么事的时候，需要发起人先写一个文档，然后大家在这个文档上进行讨论（我们通常使用 GitHub 的 issue，Pull Request 或 Google Doc）。另外，写文档的好处太多了，除了给后人有一个可以追溯的东西，更重要的是，写作是一种深度思考，当你把你脑子里想的东西写下来的时候，你就会发现你的思考更多了。所以，文档驱动是我们团队非常重要的事。

自动化和简化

自动化和简化是我平时追得最多的东西了，从软件的 Unit Test, Functional Test, Performance Test 一直到用 Kubernetes 进行自动化部署，我要求的就是从一提交完代码后就自动化的上线。我们玩的是 Amazon 的“单分支”代码管理的玩法，一旦代码 merge 上 master，就会直接上线（当然需要通过灰度）。因为远程团队如果没有自动化的工具，那么，就会导致整体效率的下降。

Owner 文化

这个太重要的了，但是，这并不是在说，如果一个事没有 Owner，就会像“三个和尚”那样，事情就到了没人管的地步。这是因为很多人在工作中都是比较 nice 的，比较 nice 的人通常来说都不好意思跳出来对别人发号施令。所以，Owner 文化就是要求每件事都要定义一个 Owner，而这个 Owner 是有权对其它人发号施令的，其他人也有义务要配合他。当然，Owner 的权利越大，责任也会越大！

Review 文化

Review 文档是一种把知识或是想法传递出去的方式。我们在实践过程中，需要大家把好的想法写下来，这需要包括问题背景、目标、可选的方案（这些方案需要有引用和数据，不能

是拍脑袋)，还需要有 Pros/Cons 的比较，然后再发起讨论。这样，事情在一开始就做好，那么就可以让大家的讨论更加地有效率。很多人以为开会讨论有个议题就行了，其实不够，有效率的开会讨论需要的是议案，而且还是高质量的议案！

目标承诺

我们需要每个人承诺自己的工作目标，这个完全由每个个体来自发完成。一般来说，每个人自己给自己制定的计划最好是在 1-2 周内。

自我管理

我们的实践是没有审批制度，无论是休假、报销还是出差，完全是自己自由安排，但需要告诉团队（除非在一些关键时期没法休长假，需要整个团队全力以赴）。千万不要撒谎和作弊，一旦发现，直接开除就好了。这个是基于好人更多的原则制定的（没有必要为了少数的坏人一刀切后让所有人痛苦）。

闲聊和自行见面

见面和不能见面是一件非常不一样的事，在一起工作时，人和人是会有感情的，因为会有闲聊。远程的时候，则只有工作了。所以，我们鼓励团队人员间的私聊、闲聊，互相讲讲自己的经历和过往，同时，也鼓励员工自行出差到对方的城市见见跟你一起工作的人，公司报销差旅费。

知识分享会

我们每周都有知识分享会，一次只讲半个小时，不贪多，就讲一个小的知识点。然后，团队中的一些人还主动使用 Google Form 来收集分享的反馈信息。

就地奖励文化

我们默认上是没有年终奖的，只有就地奖励文化。也就是说，你做的事挣钱了，利润中有 70% 公司拿走，剩下的 30% 团队的人就地分掉。这样会让团队里的每个人都会想怎么挣

钱，除了可以把精力放到那些能够让用户付费的地方上，更重要的是让团队成员了解业务。当然，如果公司没有挣钱，但是员工工作的不错，我们还是会给年终奖的。不挣钱的主要责任是我的，而挣钱的主要功劳是团队的。

外包支持性的工作

一些支持性的工作尽可能地使用外包，比如：HR、行政、发工资、财务、员工持股、测试人员、定制化开发.....这样可以让你的团队更小，更高内聚，更利于远程。

异步编程

如果一个项目是从零开始的，对于一个团队来说可能会无从下手，这需要有个人把代码的框架和结构给组织好。然后其他的人进入把坑填了，这样的效率会高很多。另外，不见面的结对编程，完全可以使用异步的方式进行，这其实就是多人干同一个 pull request 的方式。有 GitHub 这样的协作工具，远程编码变得很方便。

远程工具

关于我们的远程工具，我们主要是使用：

开发环境

AWS。因为我希望团队在使用 AWS 的时候能够被潜移默化。

协作工具

Github。我们所有跟软件开发的工作都会在 GitHub 上，我们重度使用 GitHub 的 pull request 和 issue，也会使用 GitHub Project 里的看板和 Wiki。

Google 全家桶。我们重度使用 Google，包括 Google Group、Google Driver、Google Docs。

通讯工具

语音沟通主要是使用 Zoom，因为 Zoom 不但可以支持几十人在线，还可以云录制。如果小范围交流的话，一般使用微信语音。

工作沟通主要是使用 Slack，Slack 作为一个信息集散地，可以分频道，可以分 thread 讨论，微信是个渣。

吹水群。我司的吹水群主要是 Telegram，因为比微信好太多了.....

你会发现，我们的工具有好些都是在墙外的，是的，因为墙内的同类工具实在是太难用了。而且，我倾向于让大家用上最先进的工具，这样我们团队中的每个人的品味才会被这些好的工具潜移默化。

远程工作协议

下面是我们的远程工作协议（无删减），这是每一个远程工作人员需要同意并做到的协议（其中有 Amazon Leadership Principles 的影子），目前在 v1.3 版，未来还会更新，我现在把它晒出来，也希望得到大家更好的建议！

MegaEase 远程工作团队协作协议 v1.3

Principles

0) Ownership & Leadership

每个人都是 Owner，都是 Leader，如果看到团队或是项目有问题的时候，不要等，也不忍，请马上说出来，并给出相应的方案，自己跳出来召集开会，及时调整。不要闷在那里，自己憋！

1) Initiative

每个人个都必需是主动的，都需要自己发起要做的事，或是自己要认领要做的事，如果发现自己没有事情了，需要学会主动发现问题，主动找到可以 improve 的地方，创新来源于此。没有路要学会自己造路！

2) Objectives Oriented

每个人都是产品经理，也都是项目经理，每个人都必需把自己的工作和我们大的目标连接在一起，知道什么是重点，重点的东西就是两件事：一）从用户的角度出发；二）从产品的角度出发。这意味着我们要随时观察整个产品的样子，而不只是自己这一块东西。

3) Insists on High Standard

举法其上，得乎其中，举法其中，得乎其下，举法其下，法不得也。我们要坚持用高的标准要求自己，对于高标准的目标不妥协，但是在实施路径和策略上可以妥协。

Practices

0) Online

工作的时候必需在线。如果不在线了，需要说一下不在线的时长。目前我们工作的事宜在通讯工具上采用 Slack，如果需要请假，如果不是紧急情况，需要提前一天在 MegaEase 的 Slack #random 频道中提前说明。如果是紧急情况，也需要提前在 random 频道中告知大家。

1) Documentation Driven

面对面交谈、电话语音、微信、Slack 虽然是比较实时的反馈工具，但是只有文档是可以把重要信息结构化的，而且写文档其实比起前面的方式来说是更为深度的思考，因为会让你自己审视自己的想法。所以，对于一些重要“功能”、“流程”、“业务逻辑”、“设计”、“问题”，以及“想法”，最好都以文档化的方式进行。请使用 GitHub 的 wiki、project、issue 这些工具或是使用 Google Doc。

2) Design Review

对于一些重要的问题或是工作（每个人都能够判断什么是关键问题和工作），需要先把自己的想法 share 出来，而不是先实现。

一个好的 Design 文档需要包括如下项：

- Background。交待这个事的背景、需求和要解决问题。
- Objectives。说明这个事的目标和意义。
- Alternative Solutions。给出多个解决方案，并能够进行 Pros/Cons 对比。
Reference——方案需要有权威引用支持；Data——方案需要有相关数据数据支持。
- Conclusion。结论是什么。

3) Simplification & Automation

简化和自动化是软件工程所追求的两大目标，简化不是简陋，简化是对事物一种抽象和归纳能力，能够提升软件的复用能力和扩展性。自动化是工程能力的重要体现，一方面远程工作中自动化的能力可以让整个团队更高效地协作；另一方面，自动是规模化的前得条件。所以，我们要无时无刻地思考如何简化和自动化现有的事情。

4) Review & Re-factory

无论是代码还是工作都是需要反思和重构的。反思是进步的源泉，项目告一段落时，出现问题时，都应该召集团队做集体反思，把好的东西坚持下去，把不好的东西优化掉，这样才能进步和改进。但是任何的优化措施都是可执行的。

5) Milestone Commitment

对于一个项目，每个人都需要有自己的 milestone 计划，这个计划最好是在 2 周以内，1 周内是最好的，而且要承诺到。

6) Evidence Driven

任何讨论和分析都要基于权威的证据、数据或是引用。在我们做设计的时候，或是有争论的时候，说服对方最好的方式就是拿出证据、数据或是权威引用。比如：我的 XX 设计参考了 TCP 协议中的 XX 设计，我的 XX 观点是基于 XX 开源软件的实现.....如果争论不休就停止争论，然后各自收集和调查自己观点的佐证。

7) Demo Day

把自己做的东西跟团队做一次实时的演示。这样有助于开发人员从产品角度思考自己的工作。除了演示产品功能，还可以演示算法、设计甚至代码。

8) Effective Meeting

会议主要处理三件事：提出议案、发现问题、共识结论。

会议不仅仅要有议题，最好还有议案。

会议期间不解决问题，只发现问题，和跟踪问题。

会议必需要有共识和结论，如果不能达到共识和结论，那就当成问题处理，由问题的负责人跟进问题。

关于周会或是临时性的团队会议（私下讨论不属于会议），会议组织者需要在事前收集**会议议题**，其中包括如下分类：

- 项目类：需要事先有项目进度计划表（任何分项最好控制在 1-2 人周内）
- 方案类：需要事先写好相关的方案和设计才能讨论（参看 Design Review 章节）
- 问题类：需要事先写好相关的问题和解决提案（参看 Design Review 章节）
- 决策类：需要事先写好整事的前因后果以及利弊分析信息类：需要事先写好相关的事宜说明

组织者需要在周五的时候发出会议**议题收集**，其中包括：

- 自己知道的项目的进度跟进（需要相关的项目负责人准备相关的项目计划）
- 方案和问题类的需要各个项目负责人提出来，并有相关的设计文档可供 Review
- 信息类和决策类的事宜可以写在 Google Doc 上，也可以写在 Team 的 Issue 里
- 其它负责人可以在会议上加入自己团队的东西，或是要求其他团队提供更多的信息。

9) 1-2-3 Escalation 遇到问题的时候，自己一个人处理 1 小时内没有思路，请找他人小范围讨论，如果与他人 2 小时内没有结果，请上升到团队范围，如果在团队范围 3 小时内没有思路，我们就需要借助外部力量了。

A) 3PS Update

每个人每天在签到的时候，不要只是一个 check-in，而是需要更 meaningful 的说一下今天的工作内容，在每天工作完全的时候，希望简单的说一下当天的工作总结。这里的 practice 是：3PS – Plan, Priority, Problem, Summary, – 你的计划是什么？优先级是什么？遇到了什么问题？一天的工作摘要。

B) Disagree and Commit

在我们开发的时候，团队的成员都会有自己的风格，必然会对同一个问题产生较大的争议（Disagree），我们鼓励有争议，但是是在团队的决议作出之前。一旦团队形成决议，团队的成员就必须支持这个决议，并在这个方向上做出贡献。

但是关于决议的形成过程肯定充斥着各种争论，对于这些争论，我们可以按照下面的 Guideline 来处理争议：

- Owner 要负责对重大的讨论推进，尽快形成结论。
- 在决议过程中，要有纪要，要更新到 Github 相关项目的 Issue 或 Pull Request 里，并且要让整个团队知道，信息平等很重要。
- 不要妥协，坚持高的标准。第一标准是工业标准，第二标准是国外的大公司标准（如：Google, Facebook, GitHub, AWS...），第三标准才是国内的标准。
- 哪怕再复杂，只要是标准，就可以说服用户。用户再无理，也不可能反对工业级的标准。
- Release 出去的东西，只要被用户用上了，要改就难了，所以要谨慎而果敢。

作者介绍

陈皓，网名“左耳朵耗子”，MegaEase 创始人 & CEO，资深技术专家，骨灰级程序员，QCon2020 全球软件开发大会讲师，极客时间《左耳听风》专栏作者。本文首发自酷壳 CoolShell，InfoQ 获得陈皓授权发布。

今日推荐

极客时间是 100 万人在用的一站式 IT 学习服务平台，内容覆盖计算机从业者的全职业周期知识技能。利用新技术和模式，提供便捷、经济、丰富的学习方式。

【极客时间企业版】专注于研发团队成长服务，具备多种团队在线学习场景选择，企业灵活掌握团队学习仪表盘，打造学习型组织，帮助团队实现从优秀到卓越的进化。扫码了解详情

▽



极客时间 企业版

服务升级

极客时间企业版

助力企业打造卓越研发团队



学习报告统计等新服务
扫码了解



阅读原文 文章已于2020/01/31修改

喜欢此内容的人还喜欢

如果你还在crud，你应该试试刻意练习了

三太子敖丙

深入理解JDK动态代理

安琪拉的博客