

# 技术主管

管理层之外的领导力 威尔·拉森

© 2020 威尔·拉森

# 技术主管

[Staff（职员）](#_bookmark1)

[致谢](#_bookmark2)

[序言](#_bookmark3)

[前言](#_bookmark4)

[概述](#_bookmark5)

[Staff（职员）](#_bookmark6)

[What（什么）](#_bookmark7)

[头衔重要吗？](#_bookmark8)

[Operating（运营）](#_bookmark9)

[做重要的事](#_bookmark10)

[写作](#_bookmark11)

[管理技术质量](#_bookmark12)

[与权威保持一致](#_bookmark13)

[要领导，你必须先跟随](#_bookmark14)

[学会永远不犯错](#_bookmark15)

[为他人创造空间](#_bookmark16)

[建立一个同行网络](#_bookmark17)

[向高管汇报](#_bookmark18)

[获取](#_bookmark19)

[晋升材料](#_bookmark20)

[查找](#_bookmark21)

[Staff 项目](#_bookmark22)

[获取](#_bookmark23)

[保持可见性](#_bookmark24)

[决定更换公司](#_bookmark25)

[找到合适的公司](#_bookmark26)

[面试](#_bookmark27)

[协商你的offer](#_bookmark28)

[故事](#_bookmark29)

[米歇尔](#_bookmark30)

[拉斯·卡萨·威廉姆斯 - Mailchimp 的 Staff Engineer（职员工程师）](#_bookmark31)

[Keavy McMinn - Fastly 高级首席工程师](#_bookmark32)

[Bert Fan - Slack 高级工程师](#_bookmark33)

[凯蒂](#_bookmark34) [Ritu Vincent - Dropbox 的高级工程师](#_bookmark35)

[里克](#_bookmark36) [尼尔森·埃尔哈吉 - 前 Stripe 公司职员工程师](#_bookmark37)

[戴安娜](#_bookmark38)[丹](#_bookmark39) [乔伊·埃伯茨 - Split公司的高级职员软件工程师](#_bookmark40)

[达米安·申克尔曼 - Auth0 首席工程师](#_bookmark41)

[德米特里](#_bookmark42) [斯蒂芬·万 - Samsara 的高级工程师](#_bookmark43)

[结束](#_bookmark44) [资源](#_bookmark45)

[附加](#_bookmark46)

[哪里](#_bookmark47)

[管理 Staff-plus 工程师](#_bookmark48)

[设计](#_bookmark49)

[Staff-plus职业阶梯](#_bookmark50)

# 技术主管

## 致谢

这本书的完成离不开大家的共同努力，尤其是在充满混乱的2020年，大部分工作都是在那一年完成的。要感谢的人太多了，真不知道从何说起，但其实只有一个地方可以开始：那就是分享他们故事的人们。

感谢[米歇尔](https://blog.michellebu.com/), [卡萨](https://www.linkedin.com/in/raskasawilliams/), [基维](https://keavy.com/), [伯特](https://bert.org/), [凯蒂](https://sylormiller.com/), [里图](https://www.linkedin.com/in/rituvincent/), [里克](https://www.linkedin.com/in/kineticrick/), [纳尔逊](https://blog.nelhage.com/), [戴安娜](https://diana.dev/), [丹](https://blog.danielna.com/), [乔伊](https://medium.com/%40jkebertz), [达米安](https://yenkel.dev/), [德米特里](https://d-d.me/site/presentations/)和[斯蒂芬](https://twitter.com/stpnwn)。我也同样感谢那些为[staffeng.com](https://staffeng.com/)贡献故事的人，他们的故事没有收录在这本书中。每一个故事都是独特的声音，非常值得一读。

[坦尼娅·赖利](https://noidea.dog/)的序言非常精彩，我正热切地等待Tanya出版关于Staff工程师的权威书籍，以取代这本。封面插图由[露西亚娜·格拉](https://lucianaguerra.com/)创作，让人想起古老的海图，水手们在航行中也面临着与目前Staff工程师角色一样的不确定性。TechWriters Discord上的Greglas是这本书拥有合理字体和格式的唯一原因。 [TechWriters Discord](https://techwriters.dev/)上的社区提供了无尽的建议和支持，特别感谢Gergely、Shawn和Uma。[劳雷尔](https://twitter.com/laurelcodes) 对整本书进行了文字编辑；如果你在读完两章不一致的 capitalization 后没有把它扔进壁炉，那要感谢她。

很多人审阅了各个章节，特别感谢[席德](https://www.linkedin.com/in/siddharthsarda/)、[加里](https://twitter.com/LamGary)、[帕特](https://patkua.com/)、[格尔盖伊](https://blog.pragmaticengineer.com/)、[皮特](https://thepete.net/)和[汤米](https://twitter.com/tommychheng)。我也感谢所有写推荐语的人，以及二十多位[贡献了拉取请求的人们](https://github.com/lethain/staff-eng)。

## 序言

我买的*An Elegant Puzzle*，Will Larson的第一本书，附带了一份焦虑。不是说这本书不好——它充满了见解，我推荐它。但这是一本为*管理者*写的书，而我是在办公室的管理者读书会上读的。我不是管理者；我是一名首席工程师。我不确定我是否应该在那里。

对于选择了“技术领导”职业道路的工程师来说，培养技能往往会让他们觉得自己身处错误的房间。随着我们行业的成熟和对越来越大的问题的处理，越来越多的公司认识到，需要那些“见过世面”的工程师来推动技术战略，领导跨团队和组织的项目，并通过建模良好的工程实践来提高所有人的水平。但是，在任何学科中蓬勃发展都意味着找到可以学习的资源和社区——这可能需要在技术领导岗位上发挥一些创造力。

哦，我们有很多关于技术的书籍、聚会和会议，但是从“高级”级别开始（甚至更早），技术技能是不够的。成功通常意味着解读业务需求，传达清晰的方向，化解迫在眉睫的危机，说服团队就权衡达成一致，或者仅仅是产生良好的影响。工程书架上没有太多关于这些主题的内容。相反，工程师们阅读商业或管理书籍，挑选出可以应用于技术决策、架构等主题，以及可以在没有直接权威的情况下发挥作用的技巧。目前，经理读书会通常是我们拥有的最好的学习社区。（而且，需要明确的是，我们很感激被邀请！请继续邀请我们，经理朋友们。）

高级工程师缺乏资源是更大的问题的一部分：很容易忘记这份工作到底是什么。晋升到“高级软件工程师”以上的工程师可能会发现自己很孤独，因为他们需要在一个未明确定义的新角色中摸索，努力理解“影响力”这个神秘的概念，以了解他们是否在做正确的事情，并努力适应以季度或年为单位而不是以冲刺为单位的反馈循环。

这不是故意的疏忽：经理们只是并不总是知道如何支持他们最资深的工程师。当你的报告应该就最重要的问题向你提供建议，而不是反过来时，你如何确定你的报告正在做正确的事情？你应该期望那些将成为组织其他成员榜样的工程师具备哪些技能和行为？而且，不可避免的问题是，他们应该编写多少代码？

没有公司通用的职业阶梯，或者职位头衔根本不一致，这无济于事。这就是为什么当Will推出staffeng.com并将各种角色封装到“staff-plus”集体中时，会让人感到如此欣慰。好吧，我们仍然有截然不同的头衔，但至少我们现在有一个词来描述我们所谈论的这类角色。我已经听到“staff-plus”的使用，而以前我们需要费尽口舌来描述比高级级别更资深的工程师。

staffeng.com 在科技圈立即走红。我从未见过一个全新的网站如此迅速地成为某个主题的权威资源。Will 定义并描述了 staff engineer 的角色，提供了关于如何成为一名 staff engineer 的明确建议，并剖析了他所谓的“org‐level chiropractics”，这是胜任这项工作所必需的。他擅长于抓住一个熟悉但模棱两可的主题，围绕它画一个清晰的圈，并解释我们其他人一直在看但一直没有完全看到的东西。伴随这些建议而来的是我们一直需要的社区的开端。许多访谈（与 staff engineer、首席工程师、高级首席工程师、架构师、技术主管、技术顾问等等——我告诉过你头衔令人困惑）展示了通往 staff 角色的各种途径以及更高级的职位；大多数读者都能够找到一个感觉熟悉并且，我希望，可以实现的道路。

这就是我们现在需要在 staff 工程方面进行的写作类型，我很高兴 Will 以书籍形式提供了它。就像*An Elegant Puzzle*一样，你即将阅读的这本书提供了基于真实生活经验的清晰、务实和实用的领导力建议。但这一次，staff-plus 工程师*是*听众。好吧，是听众之一。无论你是一名正在弄清楚自己应该做什么的 staff engineer，一名正在职业道路之间做出选择的中级工程师，还是一名想要为自己的

最资深的工程师创造成功的经理，你都会在这里找到智慧。

软件工程的风险每年都在增加，而且这种情况不会很快停止。正如 Will 所说，“就像《老雷斯的故事》中老雷斯为树木代言一样，staff engineer 为他们公司的技术代言。”我们对榜样工程师的技能和行为的要求将直接影响我们编写的代码、我们部署的算法、我们做出的决策以及我们认为可以接受的模式。

我很高兴技术领域的领导者终于有了这种指导。我希望这只是为我们所有在经理读书会中感到有点格格不入的人提供的众多书籍和资源中的第一本。

坦尼娅·赖利

Squarespace 首席工程师。

## 前言

当人们问我关于写我的第一本书，《优雅的难题》时，我说我花了十年时间写了一半，另外六个月写了另一半。创作过程有时充满挑战，最终成品中有很多我想改变的地方，但创作本身是一次个人高光时刻。作为一名作者，你应该警告潜在的作者不要写书，但我没有这样的警告，甚至对自己也没有：我想再写一本书。

问题是，写什么书？关于工程管理，我可能还有更多要说的，但现在肯定没什么可说的了。我做经理的时间比做开发人员的时间还长，而且还有其他人更适合写关于有效开发的书。我希望有一天能写一本关于基础设施工程的书，但我正试图在未来几年内减少对基础设施的思考。

最终，我回到了两个核心问题。今天有什么领域在挑战我？我相信一本书可以朝着积极的方向推动技术行业的议题是什么？符合这两个标准的一个问题是 Staff 工程师的角色。在大多数职业中，人们随着资历的增长越来越确定自己的角色，但我的经验是，许多工程师在达到他们的第一个 Staff 角色后会失去方向感。眼睁睁地看着人们追求 Staff 工程师的角色十年或更长时间，然后发现他们讨厌这项工作或觉得自己没有能力成功，这真是令人恐惧。

除了亲自深入研究寻找和胜任这些 Staff-plus 角色的主题之外，我还认识到不同的人在达到这些角色时有非常不同的经历。我合作过的一些最有才华的人都在努力通过高级工程师级别。每次他们尝试时，都会遇到系统性的障碍，将他们的晋升推迟一个绩效周期。

当我开始写这本书时，我的第一步是列出章节和大纲。盯着那个大纲，我很快意识到我无法独自完成这本书。这导致了一系列的访谈，在那里我得以向

那些达到他们第一个 Staff 工程师角色的人的非凡故事，以及他们在晋升后如何在这些角色中运作的故事学习。这些故事，加上我作为经理支持、晋升和招聘 Staff-plus 工程师的经验，慢慢地变成了这本书。

我希望《Staff Engineer》能帮助你完善你对技术领导力的愿景，以及你如何朝着这个愿景成长。

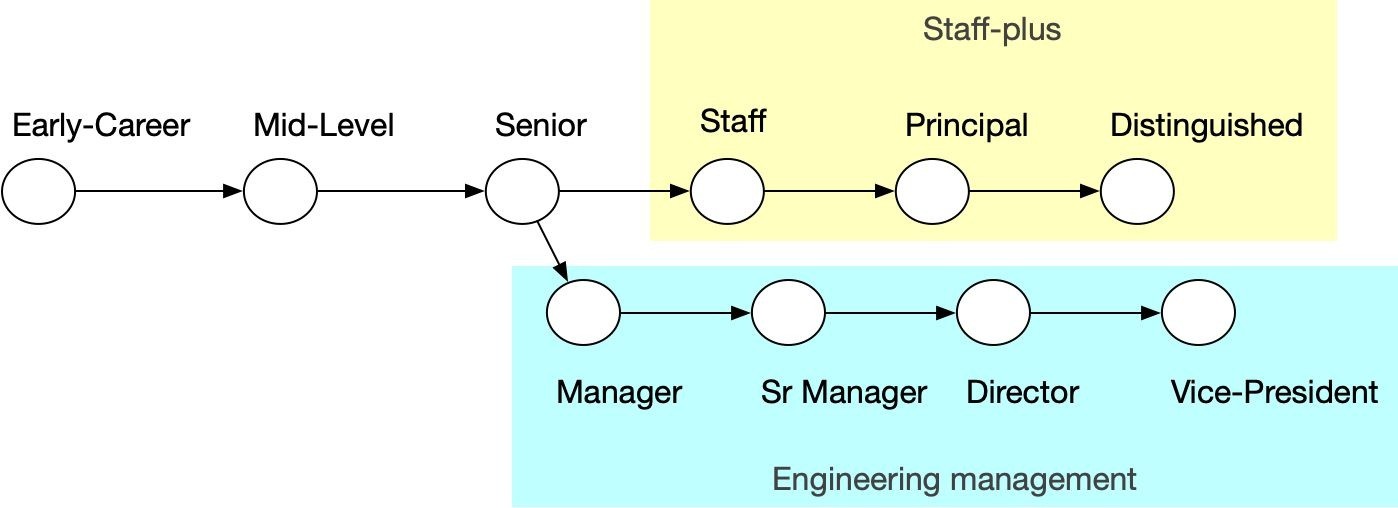
# 概述

****

[在大多数技术公司，您将在五到八年内达到软件工程师的职业级别--高级软件工程师。在这一职业层次上，公司的职业阶梯不会要求你为下一次晋升而努力；进一步晋升是一种例外，而不是期望](https://lethain.com/mailbag-beyond-career-level/)

在过去的几年里，我们看到了一系列书籍揭示了工程管理职业道路，比如卡米尔·富尼耶的《经理之路》，朱莉·卓的《成为经理》，拉拉·霍根的《弹性管理》，甚至我自己的《优雅的难题》。工程管理职业并非易事，但有地图可以帮助导航。

如果你想在不成为工程经理的情况下提升你的职业生涯呢？许多公司会兴奋地告诉你，他们有两条软件工程职业道路。工程管理是第一条道路，第二条是技术领导。技术领导道路上的人员头衔包括Staff工程师和Principal工程师。这里第二条道路的存在本身就是进步的标志，但要使其既易于访问又具有影响力，还有很多工作要做。



典型的双轨工程职业阶梯

本书标准化了最常见的头衔顺序：从高级到Staff，然后是Principal，最后是Distinguished。它使用术语Staff-plus作为Staff、Principal和Distinguished头衔的总称。许多公司只有这些头衔的子集，随着团队的成长慢慢增加，但只有一种技术领导头衔的公司几乎总是使用Staff。少数公司使用另一种顺序，但它们是少数。

围绕技术领导职业道路存在着普遍的模糊性，使得回答关于Staff-plus角色的看似简单的问题变得困难。如果你是一名高级工程师，并且想成为Staff工程师，你应该发展哪些技能？仅仅技术能力就足够了吗？大多数人是如何进入这些角色之一的？你的经理在帮助你前进的过程中扮演什么角色？你会喜欢成为一名Staff工程师吗，还是会为了达到一个不适合你的角色而努力多年？本书将帮助你回答所有这些问题。

鉴于围绕这些角色的广泛困惑，我知道我不能仅仅依靠自己的经验来写这本书。行业内十几位以上的Staff-plus工程师慷慨地分享了他们达到和在Staff-plus角色中运作的经验。他们的智慧创造了一些比我独自写作更丰富、更广阔和更有远见的东西。

如果你已经在Staff-plus职位上，我希望这些文章能激励你在管理轨道之外的领导者之旅。如果你的目标是这样一个

职位，我希望这本书能为你的追求提供务实的帮助。

你可以从头到尾阅读这本书，但如果你愿意，请随意跳跃。没有正确或错误的阅读方式。

其章节包括：

 **概述** - 对Staff工程师角色的调查，它在不同公司之间的差异，以及为什么这个头衔很重要

 [**以技术主管身份运作**](#_bookmark9)- 如何在头衔的另一边完成工作

 [**获取**](#_bookmark19)- 如何在你目前的公司获得Staff-plus的角色

 [**切换**](#_bookmark25)- 何时以及如何通过更换公司来支持对Staff-plus头衔的追求

 [**故事**](#_bookmark29)- 从Staff-plus工程师那里收集的关于他们做什么以及如何达到他们角色的故事

 [**资源**](#_bookmark46)- 如果您正在寻找更多信息，可以找到模板和进一步阅读的集合

每家公司都对Staff-plus角色有自己的理解，因此某些部分可能与您的经验不符。如果是这种情况，请接受引起共鸣的部分，并丢弃其余部分！

## 技术主管原型

大多数[职业阶梯](https://lethain.com/perf-management-system/) 为公司内部的Staff工程师定义了一套单一、统一的期望。每个人都受益于明确的角色期望，但职业阶梯是一种更适用于人群而非个人的工具。对于Staff-plus工程师来说尤其如此，他们的职业阶梯通常掩盖了隐藏在单一称号背后的几个不同的角色。

我与越多的人谈论他们在公司的Staff-plus工程师的角色，他们的经验就越好地聚集到四种不同的模式中。大多数公司强调其中一种或两种模式，并且只有在拥有数百或数千名工程师的公司中才存在一种模式。少数公司没有任何技术领导模式，并将所有经验丰富的工程师推向工程管理。在文学中，反复出现的角色模式被称为原型，例如“英雄”或“骗子”，原型术语对于标记这些常见的Staff-plus工程师变体很有用。

我遇到的Staff-plus角色的四种常见原型是：

 技术主管 指导特定团队的方法和执行。他们与一位经理密切合作，但有时他们与一个专注领域的两到三位经理合作。一些公司也有一个 技术主管经理 职位，它类似于 技术主管 原型，但存在于工程经理级别，并包括人员管理职责。

 架构师 负责关键领域内的方向、质量和方法。他们结合了对技术约束、用户需求和组织层面领导力的深入了解。

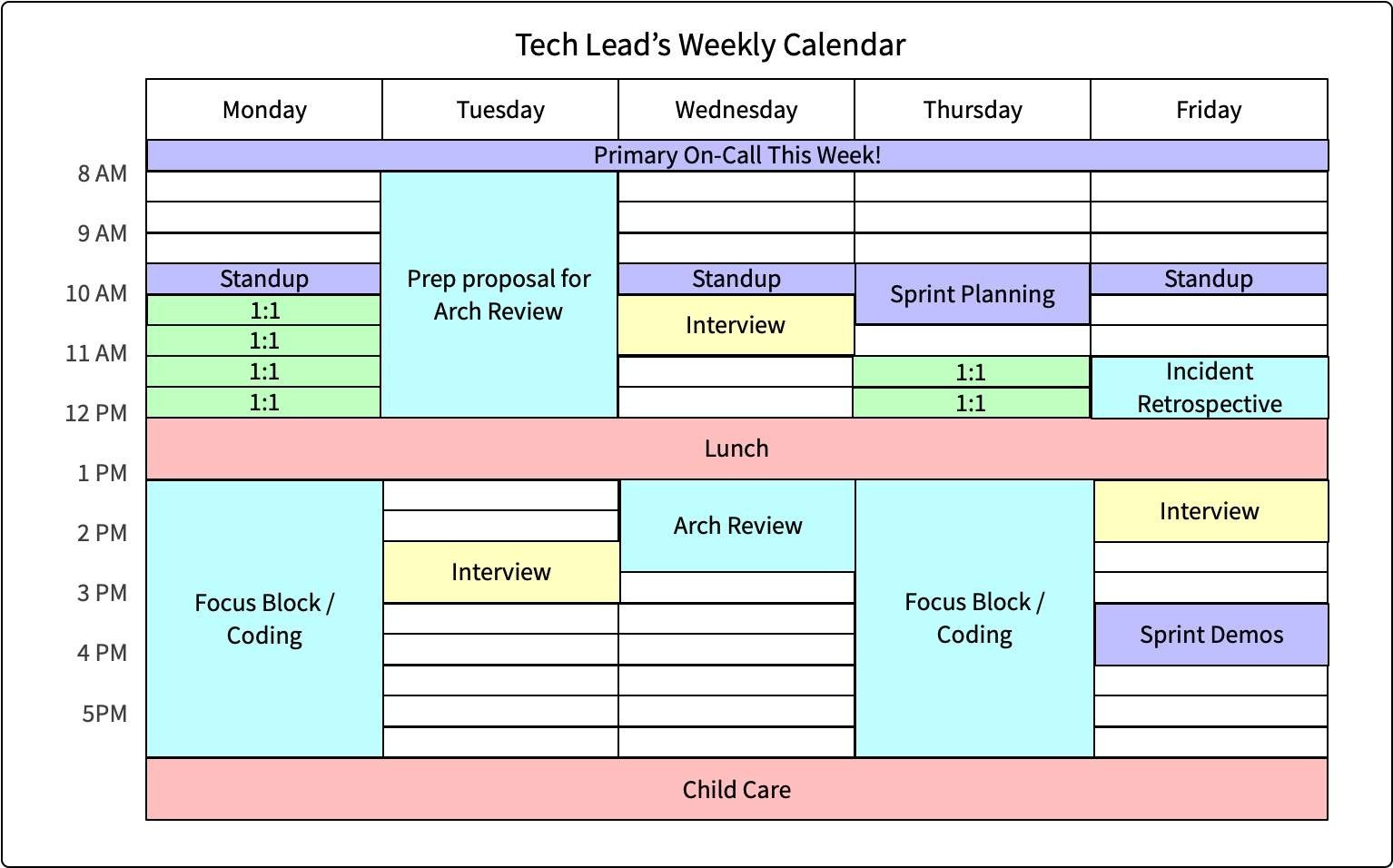
 问题解决者 深入研究任意复杂的问题，并找到合适的解决方案。有些人长期专注于特定领域。另一些人在组织领导的指导下，从一个热点跳到另一个热点。

 得力助手 延伸了高管的注意力，借用他们的范围和权力来运营特别复杂的组织。他们提供

为大型组织的领导者提供额外的领导力带宽。

这个分类法更注重 *实用性* 而不是 *完整性*，但到目前为止，我能够将我交谈过的每一位 Staff-plus 工程师都归入这些类别之一。诚然，有些人比其他人更容易分类。

### 技术主管



技术主管原型示例日历

*以技术主管原型为特色的故事*：[戴安娜](#_bookmark38)，[丹](#_bookmark39)，[里图](#_bookmark35)

*技术主管* 是最常见的 Staff 原型，并在方法和执行方面领导一个团队或一组团队。他们擅长确定复杂任务的范围，协调团队解决这些任务，并在过程中解除他们的障碍。*技术主管* 通常掌握团队的背景信息，并维护团队成功所需的许多重要的跨团队和跨职能关系。他们是团队产品经理的密切合作伙伴，也是在路线图需要调整时第一个被呼叫的人。

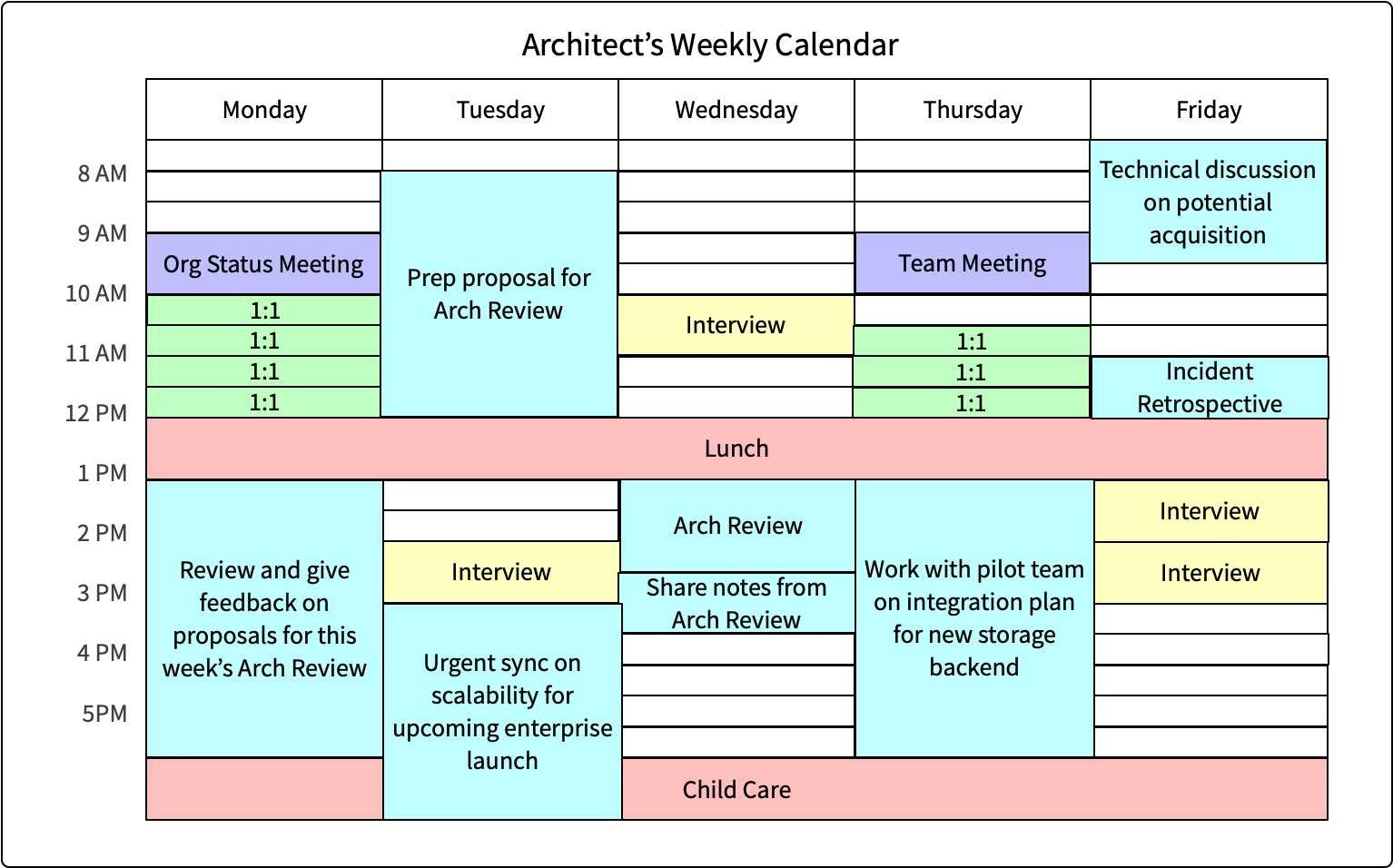
在他们职业生涯的早期，他们会实施团队中最复杂的技术项目，但在这个阶段，他们会默认将这些项目委派给团队中的其他成员。他们这样做既是为了培养他们的队友，也是为了承认随着*技术负责人*的编码量减少，团队的影响力会增长。虽然他们的编码量减少了，但他们仍然是定义团队技术愿景的人，并且会介入以在团队内部就复杂问题达成一致。

对于很多人来说，技术主管角色是他们作为架构师的首次体验。

一些因素共同促成了这一结果。首先，技术主管角色往往在对团队有强烈概念的公司中早期发展起来，这在使用敏捷方法的公司中很常见，而且大多数公司在某个时候都会尝试敏捷方法。另一个因素是，技术主管的日常工作与您作为高级工程师已经做的工作最相似，使其成为一个相当直观的过渡。最重要的是，一个组织大约每八名工程师需要一名技术主管，这使得它比其他原型更为常见。

有些令人困惑的是，一些公司使用技术主管作为头衔，而另一些公司则将其用作角色。在这个原型列表中，技术主管是作为架构师运作的一种方法，但执行技术主管角色而没有达到架构师级别工程师的预期影响是很常见的。实际上，您会发现非架构师工程师的行为与每个原型都相似。成为一名架构师不仅仅是一个角色。它是角色、您的行为、您的影响以及组织对所有这些事情的认可的交集。

### 架构师



架构师原型示例日历

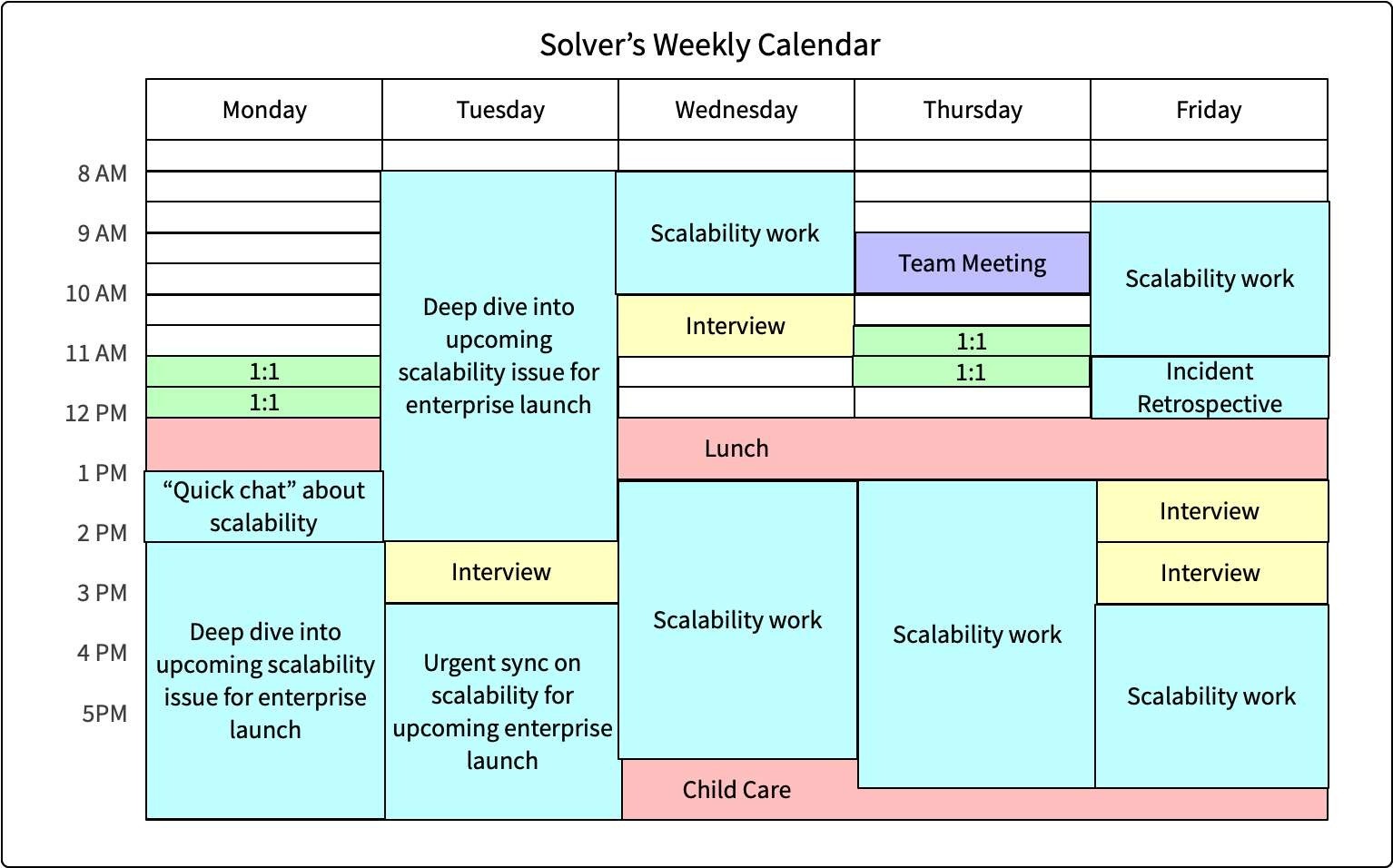
[*故事*:](#_bookmark32)[乔伊](#_bookmark40)[,](#_bookmark32)[凯蒂](#_bookmark34)[,基维 麦克明](#_bookmark32)

架构师头衔在许多公司已经不再流行，但是对于在架构师级别以上运作的人来说，架构师角色仍然活跃。架构师负责公司内特定技术领域的成功，例如，公司的API设计、前端堆栈、存储策略或云基础设施。对于一个领域来说，要值得拥有一个架构师，它必须既复杂又持久地成为公司成功的核心。

有一种有害的先入之见，认为架构师会孤立地设计系统，然后将他们的设计传递给其他人来实现。在某些情况下确实会发生这种情况，但是背诵这种刻板印象会诽谤我采访过的架构师。有影响力的架构师会投入精力来保持对业务需求、用户目标和相关技术约束的深入理解。他们利用这种洞察力来识别和倡导在其关注领域内的有效方法，并通过他们通过始终如一的良好判断力所获得的组织权威来做到这一点。

架构师角色往往在相对较大的公司、具有异常复杂或耦合代码库的公司以及正在努力偿还他们在最初冲刺以适应产品市场时所产生的技术债务的公司中发展。一些公司推动架构师深入代码库，而另一些公司则明确期望架构师不得编写代码：这两种模式都适用于某些公司。

### 解决者



Solver原型示例日历

*以Solver原型为特色的故事*：[伯特](#_bookmark33)，[纳尔逊](#_bookmark37)

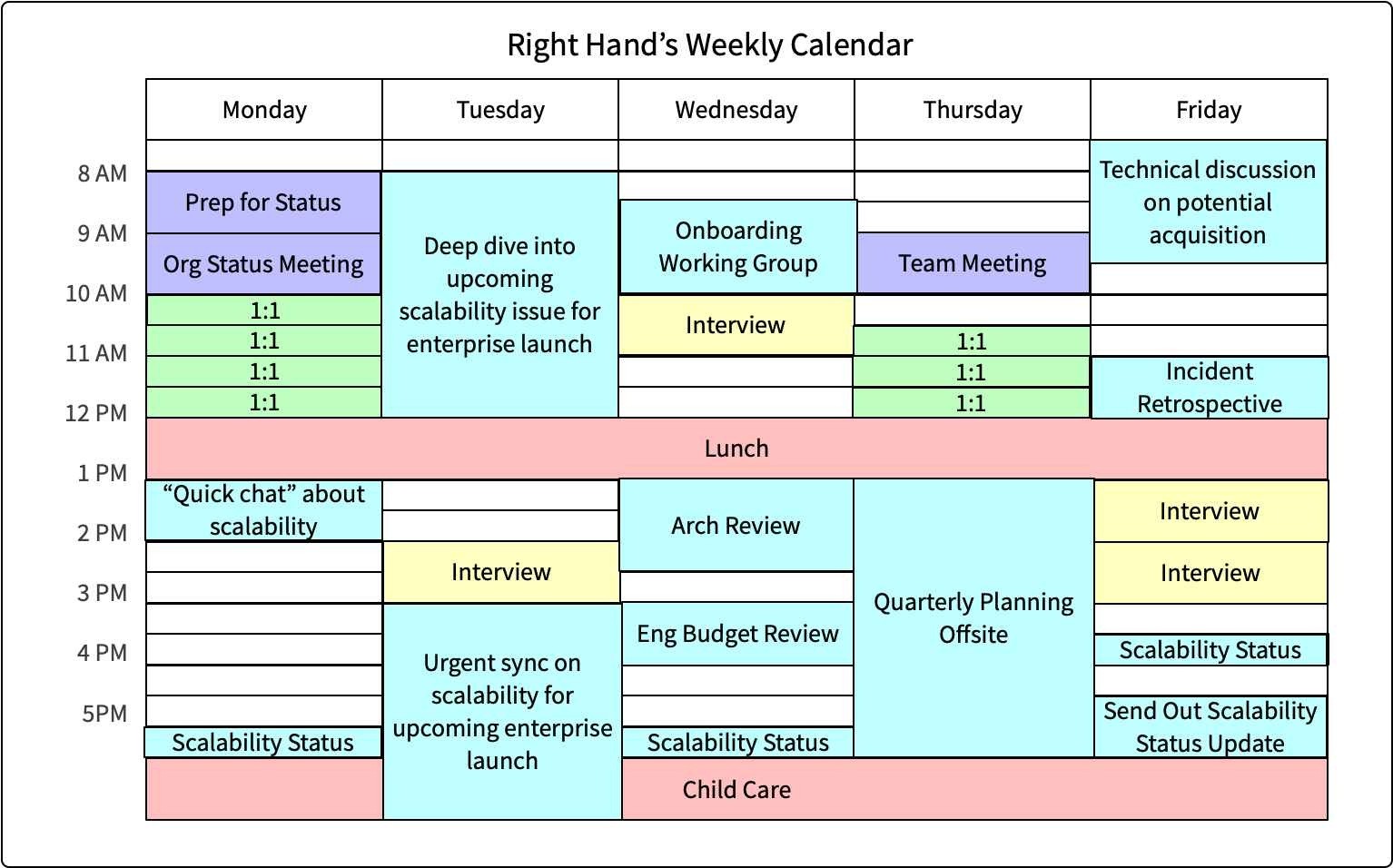
Solver是组织中值得信赖的代理人，他们会深入研究棘手的问题，并持续工作直到问题得到解决。担任此角色的人员会被调去解决组织领导层认为至关重要的问题，这些问题要么缺乏明确的方法，要么具有高度的执行风险。

大多数Staff级别职位都需要大量的组织协调，而Solver通常处理已被确定为组织优先事项的问题，因此他们需要做的组织层面的调整相对较少。另一方面，他们通常在问题得到控制后就停止工作，这可能会产生短暂的感觉，并且需要巧妙地处理，以避免激怒那些留下维护“已解决”问题的团队。

[那个*求解器*在那些认为自己是的公司里最为常见个人，而不是 团队，作为计划和所有权的基本单位。在这样的公司里，它是](https://lethain.com/weak-and-strong-team-concepts/)

公司直到那些公司变得足够大或存在足够长的时间，以至于积累了自己的技术债务。

### 右手



右手原型示例日历

*以右手原型为特色的故事*：[米歇尔](#_bookmark30)，[里克](#_bookmark36)

右手是所有原型中最不常见的，它出现在组织达到数百名工程师时，类似于在没有直接管理职责的情况下担任高级组织领导者。Rick Boone将他的角色比作《权力的游戏》中的国王之手和《白宫风云》中的LeoMcGarry，以高级领导者的借用权力运作。然而，借用权力伴随着与该领导者的方法、信念和价值观保持深度一致的义务。[里克·布恩](#_bookmark36)[将他的角色比作国王之手 在《权力的游戏》中和里奥 麦加里 来自](https://westwing.fandom.com/wiki/Leo_McGarry)[保持 深度一致 与那位领导人的方法、信念和价值观保持一致。](https://lethain.com/staying-aligned-with-authority/)

担任此角色的人员会参加其领导的员工会议，并通过解决领导面临的重要问题来扩大其影响力。在此层面解决的问题绝非纯粹的技术问题，而是涉及业务、技术、人员、文化和流程的交叉。*左右手*通常会深入火场，修改方法，将执行权委托给最合适的团队，然后跳到组织中其他地方的下一个火场。这些角色的乐趣在于你只处理重要问题。悲剧在于，当这些问题得到解决时，你总是在处理下一个问题。

### 哪个适合你？

当您考虑哪种原型适合您时，首先要反思哪些类型的工作能激发您的活力，然后考虑您的公司有哪些职位空缺。

所有公司都需要能够胜任技术主管角色的工程师，这使得它成为获得您的第一个Staff工程师角色最容易实现的原型。强调个人所有权而非团队所有权的公司通常会尽早培养问题解决者。另一方面，在严格的冲刺或敏捷方法下运营的公司往往会较晚培养这种角色，甚至永远不会培养。在最近涌现的快速增长的科技公司中，架构师和右手角色通常在组织分别达到一百名和一千名工程师时出现，并且在此之前根本不存在。具有其他文化基因的公司通常会更早地培养它们，或者有时永远不会。

在这些角色中取得成功需要保持投入；了解什么类型的工作能激发您的活力至关重要。技术主管和架构师倾向于多年来与同一批人一起解决相同的问题，从而培养出紧密的团队意识和共同目标。有时，他们的重点将是公司的首要任务，有时他们会进展顺利，以至于高管们忘记了他们团队的存在。

问题解决者和右手在各个紧急情况之间奔波，通常与他们在任何特定一周内合作的人进行更多的事务性互动。他们与高管的优先事项紧密结合，并可能因解决领导层最紧迫的问题而获得认可。另一方面，虽然他们名义上会与其他同事组成一个团队，但他们的团队关注领域通常几乎没有重叠，而且他们的社区意识往往有限。

对于每种原型，您都会发现喜欢它并觉得它非常有价值的人，以及觉得这项工作令人绝望的人。虽然朝着适合您的原型努力很重要，但同样值得记住的是，在您三十年或四十年的职业生涯中，您将有足够的时间[四十年](https://lethain.com/forty-year-career/)

体验每种原型。

## Staff工程师实际上做什么？

*Staff-plus工程师的角色很大程度上取决于团队的需求以及特定工程师的优势。根据我的经验，Staff-plus工程师的职责会随着时间的推移而变化。不过，通常情况下，他们的主要重点是从事对公司具有战略价值的项目/工作，同时推动技术设计并提升团队水平。-* [*戴安娜·波亚尔*](#_bookmark38)

任何在聚会上被亲戚围住并被要求解释软件工程师实际上做什么的人都知道，解释这项工作可能具有挑战性。随着时间的推移，您可能已经为您的亲戚创建了一个令人信服的答案，但是当他们的同事凑过来问：“Staff工程师是做什么的？”时，许多人的脑海一片空白。

最直接的答案是，技术专家仍然在做很多让他们作为高级工程师获得成功的事情：建立关系、编写软件、协调项目。然而，这是一个具有误导性的答案。技术专家确实也在做这些相同的工作，但之前这些是他们工作的核心，现在它们是辅助任务。他们的日常安排因原型而略有不同，但在所有原型中都有一个共同的基础：设定和编辑技术方向，提供赞助和指导，将工程背景注入到组织决策中，探索，以及 Tanya Reilly 所谓的“做粘合剂”。[通过 原型，但所有原型都有一个共同的基础：设定和](#_bookmark6)[坦尼娅 莱利 称之为](https://noidea.dog/)[做粘合剂](https://noidea.dog/glue)[.](https://noidea.dog/)

### 设定技术方向

*当我可以促进为一个领域设定技术愿景并让人们朝着这个愿景前进时，我感觉最有影响力。我认为我们都会同意，我们希望我们的代码在架构上比现在更好，或者在某些方面有所改进。然而，我发现人们常常对想要变得更好有一种模糊的感觉，但并不清楚他们想要的东西是什么。我喜欢帮助团队就他们试图达到的目标达成共识（实际上，即使我们永远无法达到目标也没关系），并提出一个实现目标的大致计划。- Joy Ebertz*[*乔伊·埃伯茨*](#_bookmark40)

[洛拉克斯](https://en.wikipedia.org/wiki/The_Lorax)就像《老雷斯的故事》中受欢迎的儿童读物里的老雷斯为树木代言一样，技术专家为他们公司的技术代言。技术不能为自己说话，需要有效的倡导者。成功推进技术的人是务实的、深思熟虑的，并且更关注长期进步的趋势，而不是将每个单独的决定视为成败攸关的危机。可以把它看作是兼职的技术产品经理。

有些 Staff-plus 工程师明确受雇领导 API 设计等特定领域，而在其他情况下，他们会发现自己需要编辑和调整广泛领域的方法。所有角色都有一个共同点，那就是在现实中，确定技术方向更多的是要了解

### 指导和赞助

*在我目前的职位上，当我赞助的人发布他们已经交付工作的公告时，或者当我看到我帮助塑造或改变了一个工程团队对一个重要主题的理解模型时，我感到精力充沛。是这些团队，而不是我，在日复一日地努力构建和支持他们的技术。我根据他们的进展来衡量我的影响，更重要的是，根据进展的方向以及他们的工作与公司目标的对齐程度。- Michelle Bu*[*米歇尔·布*](#_bookmark30)

有一种流行的英雄式领导的愿景，它以非凡的生产力个人为中心，他们的决定改变了公司的未来。这些叙述大多是由公共关系团队故意设计的，以创造一个好故事。通过培养你周围的工程师，你更有可能改变公司的长期发展轨迹，而不是通过个人英雄主义。培养你周围的人的最好方法是积极实践指导和赞助。

有时人们会在他们的职业阶梯中看到对指导的要求，并试图机械地勾选那个框，这很可惜，因为指导是高级职位中最有价值的活动之一。分享你的经验和建议，并建立持续的关系以了解接受者的背景，是一项具有高度影响力的工作。最有效的技术专家会将适度的指导与更多的赞助结合起来：直接在天平上施加压力，以帮助提升和支持你周围的人。如果你还没有读过，Lara Hogan 写了一篇关于赞助和指导之间区别的经典文章，What does sponsorship look like?[赞助是什么样的 ？](https://larahogan.me/blog/what-sponsorship-looks-like/)

### 提供工程视角

*我在更高级别的工程讨论中占有一席之地，这些讨论发生在个人项目和团队之上。我们有定期的员工工程会议，在会上我们讨论跨团队的问题，这些问题在本质上既有技术性的，也有非技术性的。-*[*丹·娜*](#_bookmark39)

高效的组织会简化日常决策。一个很好的例子是审查潜在企业客户合同的流程。早期，会有一些合同签署后，产品和工程团队不支持，感到不舒服。在发生几次这种情况后，流程将在审查步骤中包括更多的利益相关者，并且随着时间的推移，合适的人将在合适的时间出现在合适的地点。

即使是那些擅长做出日常决策的公司，在出现意外决策时也常常会感到吃力。这种决策既有时效性又很重要，甚至在做出决策之前，很难将合适的人员聚集在一起。经常发生[组织结构调整](https://lethain.com/running-an-engineering-reorg/)[到](#_bookmark23)

Staff-plus工程师通常是那些意外地被拉[进入](#_bookmark23)

房间里参与这类决策的人。这使他们有机会在仍然可以改变结果的情况下，将工程背景和观点注入到决策中。这些在关键决策上的短暂投入影响很大，将使您能够注入工程视角，否则这些视角将被忽略。请记住，您代表的是整个工程部门的利益，而不仅仅是你自己的利益。

### 探索

*在我目前在孵化器中的角色中，我整天都在进行原型设计，但在我之前的技术主管角色中，我做了很多不同的事情。-* [*里图 文森特*](#_bookmark35)

[爬山算法](https://en.wikipedia.org/wiki/Hill_climbing)是一种简单的优化算法。想象一下，你站在某座山上，想要到达山顶。你转过身，环顾四周，找到附近最高的点，然后走到那里。一旦你到达那里，你再次转过身，环顾四周，从你的新位置找到附近最高的点，然后去那里。如果你一直这样做，你就会到达你所在的山的顶部。但是，想象一下你在一个雾天尝试这样做。因为你看不了很远，你可能会到达附近最高的点，然后意识到有一个更高的点就在视线之外。

爬山法无法解决所有问题，但它非常有效，以至于许多公司都在努力采取其他方法。这可能是一家以消费者为导向的公司，难以支持企业交易，或者是一家成熟的公司，难以与规模较小的竞争对手的发布节奏竞争。甚至可能出现这样的情况：你目前的业务非常有价值，以至于[很难确定新的优先级 商业，即使有价值的业务增长率正在下降](https://en.Wikipedia.org/wiki/The_Innovator%27s_Dilemma)

从长远来看，公司要么学会探索，要么逐渐消失；这不是一个可以忽略的挑战。简单地分配一个已经掌握了爬山法的团队[到 做 是](https://lethain.com/how-to-invest-technical-infrastructure/)

这也不一定是一个业务问题；它可能是公司系统难以解决的任何模棱两可的重要问题。它可能是将你的基础设施成本降低一个数量级。它可能是确定一个需要六个月而不是三年的多区域战略。它可能是解决突然意识到你的主数据库只有三个月的剩余磁盘空间，并且你无法升级到更大的尺寸（根据我的经验，在快速增长的初创公司中，这是一个非常常见的问题）。

这是公司所做的最具回报也最具风险的工作。要被信任从事这项工作，需要大量的组织信任，包括得到业务部门足够的尊重，这样即使你失败了，那也是问题的反映，而不是你的问题。

### 成为粘合剂

[坦尼娅·赖利](https://noidea.dog/) 写了一篇精彩的文章，[成为粘合剂](https://noidea.dog/glue)，它捕捉了成功的Staff工程师的另一个核心要素：做那些需要的，但往往是不可见的任务，以保持团队前进并交付其工作。这并不光鲜，但高影响力的组织通常会有一名或多名Staff工程师在幕后工作，以加快[最重要的工作，并确保它完成](#_bookmark10)。

### 但是你还会写软件吗？

如果不就Staff工程师在聚在一起时提出的第一个问题发表看法就结束对Staff工程师角色的讨论，那是不礼貌的： “你还有时间写软件吗？” 答案当然是，这取决于情况！

[拉斯·卡萨·威廉姆斯](#_bookmark31) 说：“我仍然定期贡献代码——当然比我团队中的其他工程师少；但重要的是，我坚持“手到键盘”的工作，以确保我的技术战略（和其他宏观层面的决策）能够从我团队其他成员的实际经验中获得信息。”

[凯蒂](#_bookmark34) 说：“我是一名前端架构师，但到目前为止，我最近写的主要内容是SQL，因为我正在做大量的数据分析。我一直在

查看我们的性能指标，以找出需要改进的领域，以及修复哪些问题对提高性能和业务指标最有影响。我会在这里和那里编写一些JS或PHP代码，但主要目的是帮助团队解决问题或运行与性能相关的小型实验。”

[乔伊](#_bookmark40) 说：“你越资深，你的工作就越与代码无关。当然，与人事经理不同，你仍然有非常技术性的倾向，甚至通过负责人，你可能至少还会做一些编码。但是，你的职位越高，你的工作就越是关于指导和培养你周围的人（以及更广泛的范围），通过建立公司公开的技术品牌来建设你的团队，注意到可以改进或纠正的更大的技术趋势，帮助为你的团队或公司设定技术愿景，并倡导为技术债务项目提供资源。”

大多数人会写一些，有些人不写，但没有人写得像他们职业生涯早期那样多。偶尔会有一周完全是编码，但这些不会是常态，如果这种情况发生得太频繁，通常表明你正在[做一些舒适而不是重要的事情](#_bookmark10)。即使你写的不多，你也会阅读*大量的*同事的代码，并进行相当数量的代码审查。

### 缓慢但有回报

Staff-plus 工作的一个共同主题是时间框架更长。在你的职业生涯早期，很容易迷恋上软件开发快速的反馈周期——编写、测试、发布、重复——而你在这个级别所做的大部分工作，都将用需要数周、数月甚至数年的反馈循环来取代它。当你第一次承担 Staff-plus 角色时，这些较长的时间框架会让人感到非常沮丧。作为一名 Staff-plus 工程师，有些日子感觉自己一事无成是很正常的——坚持下去！

影响和个人成长都存在于那些较长的时间框架中，虽然我交谈过的每个人都希望他们偶尔能有更多的时间来编写代码，并且承认有些日子担心自己没有取得多大成就，但他们没有人后悔转型到目前的角色。

## 头衔重要吗？

如果你安全地依偎在高级工程师[职业级别](https://lethain.com/career-levels-and-more/)的舒适怀抱中，你可能会想是否应该追求 Staff 头衔。这是一项相当大的时间和精力投入，同时也需要相当多的运气。这项投资值得你付出时间吗？

答案当然是，有可能！通常伴随 Staff-plus 头衔而来的三个一致的优势是：

1. 允许你绕过非正式的资历衡量标准，
2. 方便进入“房间”，
3. 增加当前和职业薪酬。

潜在的第四个优势是，有些人发现这个头衔赋予了更多选择你所从事项目的自主权，但另一些人发现，这种自主权的增加被对业务相应的责任增加所吞噬。

### 非正式的资历衡量标准

当我与[纳尔逊·埃尔哈吉](#_bookmark37)谈论晋升到Staff级别是否允许他承担新的工作时，他回答说：

*关于“允许”的问题很有趣，可能不是一个完全正确的问题，因为关于谁能获得什么样角色的官方政策很少。大多数事情都依赖于更非正式的资历衡量标准。*

许多科技公司将自己描述为追求精英管理，定义为创造条件让有才华的员工自然而然地晋升到顶层。鉴于没有任何被广泛接受的个人价值衡量标准，这些公司开始依赖尼尔森恰当地称之为“非正式资历衡量标准”的东西。虽然这些衡量标准被认为可以客观地评估想法，但它们的纯粹非正式性成为一种广泛的偏见载体，并且经常将信心与能力混为一谈。

从重新确立个人能力的过程中解放出来，这一点经常被提及

作为Staff头衔的一个关键优势。这些非正式的衡量标准并非我与之交谈的每一位Staff-plus工程师都提到过，但那些不符合公司对经验丰富的技术专家刻板印象的个人经常提到它们。

[基维](#_bookmark32)

*当你拥有头衔时，你就不必花费那么多精力将你的资历摆在桌面上。它有助于为他人设定背景。从一开始你就更受尊重，这一点非常明显。*

Staff-plus头衔允许你将之前花费在证明自己上的精力重新投入到你所评估的核心工作中。如果你发现你没有花费太多精力来证明自己，那就太好了！也许你在目前的公司工作了足够长的时间，并且已经证明了自己足够多次，以至于这不再是一个问题。如果你*确实*发现你的时间被转移到证明和反复证明自己上，那么这个头衔将为你重新投入使用的时间带来相当大的回报。

### 身处其中

Staff-plus头衔的另一个常见优势是“身处其中”。丹·纳将其描述为：[丹·娜](#_bookmark39)

*我在更高级别的工程讨论中占有一席之地，这些讨论的层次高于单个项目和团队。我们有定期的员工工程会议，在会上我们讨论跨团队的问题，这些问题本质上既有技术性的，也有非技术性的。举个假设的例子，在这种类型的会议中，我会很乐意提出我认为工程入职流程中的不足之处。*

对于任何重要的决定，都有做出核心决定之前的准备阶段，以及之后的一切。在更高级的职位上，你通常可以在相对容易整合的时候提供意见，否则你的反馈可能不会被采纳——尽管它非常有价值——因为相关的推广或实施已经进展得太远。

### 薪酬

小公司的薪酬往往是临时性的，增长来自于与你的经理直接谈判。在这样一家公司晋升到Staff-plus职位甚至可能不会带来相应的薪酬增长。然而，大多数公司在达到一到两百人时，会为每个职位引入薪酬范围。这些薪酬范围通常会确保你的薪酬随着职位而增加。

任何公司薪酬最高的职位往往是执行和高级管理职位。随着公司的发展，他们通常会在管理和工程职位之间建立薪酬映射，这样达到Staff-plus职位（有时是Sr Staff或Distinguished职位，而不是最初的Staff职位）将显著提高你的薪酬。

即使你目前的公司对Staff-plus工程师职位的薪酬与高级工程师职位的薪酬没有太大区别，[*有些公司会这样做*](https://www.levels.fyi/)。在你的职业生涯中，你可以选择朝着这样的公司发展，并且拥有Staff-plus的头衔将有意义地增加你一生的收入。

### 接触有趣的工作

许多人担任Staff-plus职位，认为这将使他们能够接触到最引人注目或最令人兴奋的工作。在某种程度上这是真的，但这取决于Staff原型在你公司中最普遍。例如，解决者通常确实可以接触到最有趣的工作。相反，如果技术负责人以这种方式运作，可能会破坏他们的团队。[Staff（职员） 原型 哪些在你的公司最普遍。例如，*问题解决者*](#_bookmark6)

在我交谈过的人中，获得有趣工作最有效的方法是一开始就被聘用去做，例如[瑞图·文森特](#_bookmark35)被聘用启动Dropbox的产品孵化器，以及[基维](#_bookmark32)被聘用设计Fastly的API战略。

这并不总是奏效。有时，有趣的工作会清晰可见，但仍然遥不可及。你将不得不为了满足业务需求，而无法追求个人感兴趣的项目。在早期的职位上，你也许可以

偷偷地将这类项目添加到你的待办事项中，但现在你有责任树立良好的行为榜样。即使在项目对公司最有利的情况下，你也经常会决定将这个机会让给另一位工程师，因为他们会比你从中受益更多。

### 不同，而非更好

即使头衔很重要，但你不一定非要追求这个职位。即使你喜欢Staff+头衔的特权和福利，重要的是要认识到它们是建立在一个非常不同的工作之上的。[米歇尔·布](#_bookmark30)在给追求Staff头衔的人的建议中捕捉到了这一点：

*如果你更专注于达到Staff级别，而不是让自己去做那些能激发你能量的工作，那么很容易最终陷入一个你不想要的职位。成为一名Staff+工程师，尤其是一名广义的Staff+工程师，与成为一名高级工程师是非常不同的工作。重要的是退一步，思考一下这是否真的是你想要的工作。*

高级职称的优势是真实存在的，对于某些人来说，这些优势将他们的职业生涯从以生存为特征转变为具备成功所需的必要先决条件。然而，许多人发现，他们的Staff角色所提高的期望消除了过去让他们兴奋的工作。在你的职业生涯中，很少有选择是没有后果的，而这并不是其中之一。

### 物质上的，但不是魔法

你偶尔会遇到一位工程师，他认为获得某个头衔是他们与重要成就或机会之间唯一的障碍。这些人可能会表达出沮丧，例如，“如果我拥有Staff头衔，我就可以决定我们团队的技术栈。”

提高组织权威确实为解决问题提供了新的工具，但在管理良好的组织中成功地保持组织权威需要大量的细微差别和克制。如果你有一个问题，并且认为你的头衔是唯一阻碍你的东西，我想向你保证，专注于发展你的方法和技能将会更加

比头衔更有影响力。一旦你接近悬崖边缘，头衔会帮助你越过，但它永远不会像你期望的那样发挥那么大的作用。

这条规则唯一不变的例外是，女性和少数族裔通常确实发现，一旦她们获得Staff+头衔，她们花费在证明自己上的时间和精力会大大减少。这个头衔并没有为她们解锁新的能力，但它确实减轻了她们在整个职业生涯中一直背负的一些重量。

# 以技术主管身份运作

****

*有人给我的最好的建议之一，也是我确保传递给其他Staff工程师的建议是，有一种误解认为你成为一名Staff工程师，然后你就可以控制你所做的工作，每个人都会听你的，并按照你想要的方式去做。这与实际发生的情况完全相反！-*[*凯蒂·赛勒-米勒*](#_bookmark34)

许多工程师专注于 Staff-plus 职业道路，因为工程经理的道路有太多的会议，或者需要与其他同事进行太多的协作，而且，天哪，如果你以这种心态开始 Staff-plus，你会感到惊讶的。虽然 Staff 工程师的角色通常被定位为高级工程师之后的连续步骤，但它确实是一个不同的角色，你将越来越多地花时间做你以前很少做或根本不做的工作。

在 Staff-plus 职位上存在一个显著的学习曲线，最初会让大多数人绊倒。部分挑战在于，你所做的很多工作都有一个慢得多的反馈周期。这种延迟的反馈最初会让人感到非常沮丧，因为你用指导、关系建立和战略的不均衡进展取代了发自内心的编码REPL。

本章是关于克服这种学习曲线，学习如何作为一名 Staff 工程师运作，并找到这个角色中个人感到满足和组织上具有变革意义的部分。

### 主题

在本书的访谈中，以及我本人领导和指导 Staff-plus 工程师的经验中，一些主题不断涌现，成为个人发展的基石。它们并不是你在这个角色中会做的*一切*，但它们是你最有可能产生巨大影响或意外做出限制职业发展的举动的地方。

1. [**做重要的事**](#_bookmark10)以充分利用你的工作时间，特别是随着你在职业生涯中不断前进以及生活中的承诺不断扩大。
2. [**撰写工程策略**](#_bookmark11)以指导你的组织如何通过其架构、技术选择和组织结构来支持公司的业务目标。
3. [**管理技术质量**](#_bookmark12)以保持公司架构和软件的质量，随着时间的推移，它会不断增长和调整。
4. [**与权威保持一致**](#_bookmark13)以长期保持有效的领导者。技术领导角色依赖于来自另一位（通常是管理层）领导者的代理授权，而持续获得该授权取决于保持一致、值得信赖和可预测。
5. [**要领导，你必须先跟随**](#_bookmark14)。对事物应该如何运作有一个生动的认识是一种强大的领导工具，但学习将你的愿景与你的同伴和领导的愿景融合起来也至关重要。
6. [**学会永远不犯错**](#_bookmark15)从正确转向理解和沟通。停止花费你的社会资本来修复因冲突而破裂的关系，并学会与具有不同优先事项和观点的人合作。这还带来了额外的好处，即抱怨你的人会减少。
7. [**为他人创造空间**](#_bookmark16)，以便你的团队比你的贡献更强大。
8. [**建立一个同行网络**](#_bookmark17)，以审查困难的决策，并在你的角色权威开始缓和反馈时，给你诚实的反馈。

[一位敏锐的读者会注意到在员工 工程师实际上做什么？ 这个主题列表中缺少的是：第一个是“指导”](#_bookmark7)

淡而无味的重述更好。对于指导和赞助，花一些时间阅读 Lara Hogan 的[赞助是什么样的？](https://larahogan.me/blog/what-sponsorship-looks-like/)，对于做粘合剂，花一些时间阅读 Tanya Reilly 的文章，这篇文章孕育了“[成为粘合剂](https://noidea.dog/glue)”这个短语。

当你刻意练习这些领域中的每一个时，你将慢慢地从一个新晋的 Staff 工程师发展成为一个值得信赖的组织领导者。也就是说，这些并不能涵盖你所做的一切。有时你会发现你的角色与工程总监的角色惊人地相似，有时又会觉得与你职业生涯中以前的工作非常熟悉。

这种广泛的职权范围是描述这些角色具有挑战性的部分原因。如果你专注于某个缺失的特定主题，请查看[学习的额外资源](#_bookmark46) 附录。

## 做重要的事

*我开始使用“充满活力”这个词来代替“有影响力”。“有影响力”感觉是以公司为中心的，虽然这很重要，但“充满活力”更注重内在。找到充满活力工作是我在 Stripe 工作这么久的原因，追求有影响力工作。-*[*米歇尔·布*](#_bookmark30)

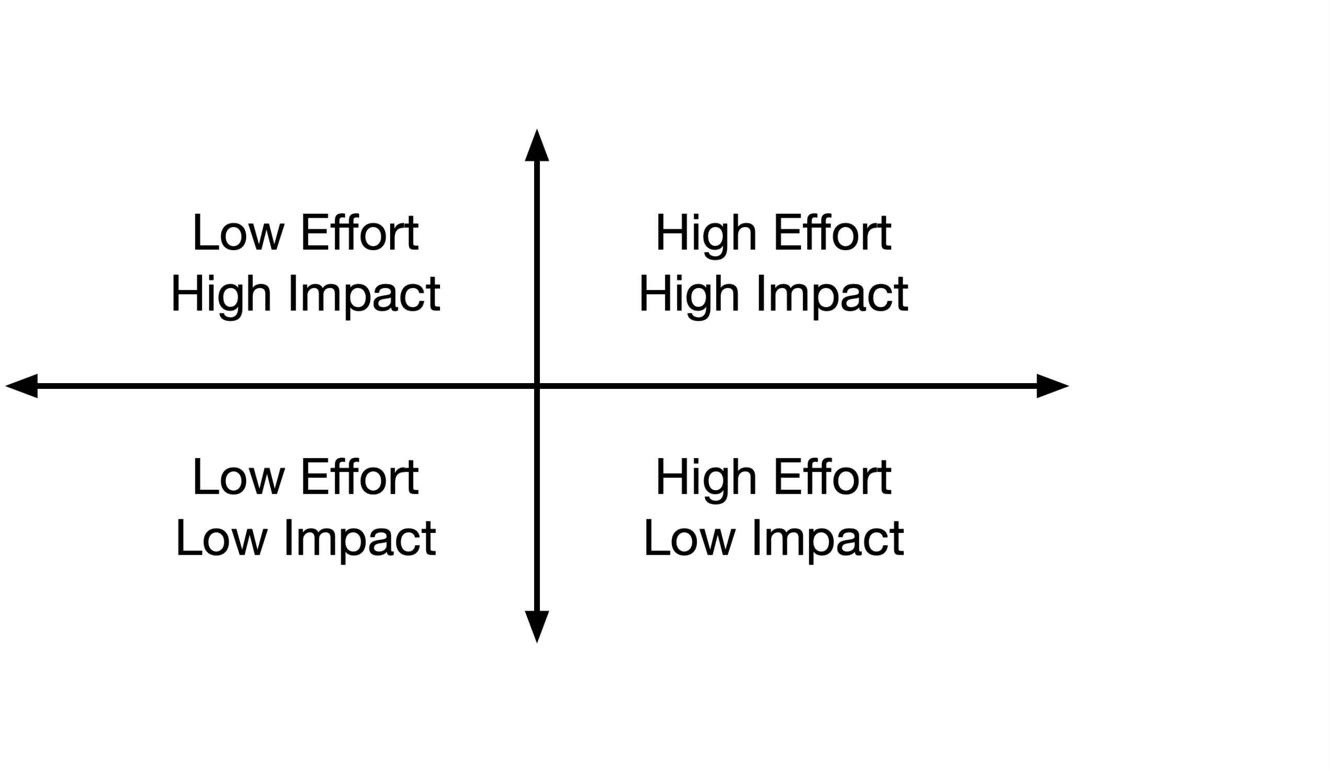
我们都有有限的生命时间，在这有限的倒计时中，我们投入了一部分时间到工作中。即使对于那些最专注于事业的人来说，你的生活也会充满工作之外的许多事情：支持你的家庭、孩子、锻炼、成为导师和学员、爱好等等。这是一个丰富生活的标志，但一个副作用是，随着你职业生涯的深入，你工作的时间将变得越来越稀缺。

如果你在职业生涯中不断进步，那么即使你可用于工作的时间减少，对你的影响力的期望也会不断增长。你可以尝试少睡觉或剥夺自己感受完整所需的非工作活动，但你最终会发现，你的工作对你的牺牲保持着冷漠的漠视，而不是奖励它。只有通过[调整你的 职业生涯与你的生活 你能长期维持自己吗？](https://lethain.com/forty-year-career/)

的确，自我调节成为一项持续、成功的职业生涯的核心挑战：越来越高级的职位要求你完成越来越多的工作，并且用越来越少的时间来完成。这两个约束条件之间的界限会随着你的晋升而变得越来越窄，但如果你采取一种深思熟虑的方法，它仍然是可以行走的。

首先，讨论一下容易出错的几种常见方式：*吃零食*、*梳妆打扮*和*追逐幻影*。然后我们将进入正题：如何*做*你从事真正重要的事情？

### 避免吃零食



高影响和低影响象限，按高努力和低努力划分

[避免“吃零食”](https://hunterwalk.com/2016/06/18/the-best-startups-resists-snacks-im-not-talking-about-food/)Hunter Walk 建议大家在确定工作优先级时避免“吃零食”。如果你在一个运转良好的组织中，在某个时候，你将会耗尽那些既有高影响力又容易完成的事情。这让你面临一个选择，要么向右转移到困难但高影响的事情上，要么向下转移到容易但低影响的事情上。后一种选择——容易且低影响——就是 Walk 所说的吃零食。

当你很忙的时候，这些零食会给你一种成就感，从而在心理上给你带来奖励。不过，你不太可能从做这些事情中学到很多东西，其他人可能也同样有能力完成它们（*而且*对于其中一些事情来说，这可能是一个很好的发展机会），而且与做一些影响更大的事情相比，存在巨大的机会成本。

花一些时间在零食上，让自己在更大的成就之间保持动力是可以的，但你必须诚实地对待自己，看看你花了多少时间在高影响的工作上，而不是在低影响的工作上。在高级职位上，你更有可能自己决定自己的工作，如果你没有刻意跟踪你的工作，你很容易发现自己几乎没有做任何高影响的工作。

### 停止梳妆打扮

“吃零食”是做容易且低影响工作的广泛类别，

有一种特别诱人的吃零食的子集，我称之为“梳妆打扮”。梳妆打扮是做低影响、高可见性的工作。许多公司将高可见性和高影响力混为一谈，以至于他们无法区分梳妆打扮和影响力，这就是为什么看到一些公司的高级工程师花费大部分时间做一些价值可疑，但在公司会议上经常得到认可的工作并不罕见。

如果你从短期来看待[职业发展](https://yenkel.dev/posts/how-to-achieve-career-growth-opportunities-skills-sponsors)，那么为了适应你当前组织在评估影响力方面的病态而进行优化是最佳途径：大胆前进，光荣地梳妆打扮。但是，如果你正在考虑发展自己，以便在你的[当前角色在复杂性方面不断增长](https://lethain.com/growing-with-your-company/)或跨多个组织时取得成功，那么在有价值的工作和自我成长之间取得平衡就显得更为重要。

这也是选择公司工作时要考虑的一个重要因素！深入了解一家公司的价值观，并确保它与你预期的个人成长相符。如果一家公司的领导层完全由那些将精力集中在表演性紧急事务或效忠行为上的人组成，那么当你在公司取得成功取决于这些活动时，不要感到惊讶。

更糟糕的是，要成为一个成功的献媚者，需要对你实际影响的批评几乎刀枪不入，如果你的精力被转移到献媚上，你真正的工作将会受到影响。通常这意味着你需要成为一位高级领导的虚荣心雇佣，或者以一种公司认为领导者应该有的样子来展示自己。如果那不是你，那么你试图用你的良好判断力来换取公司成功的尝试最终还是会失败：你会被追究缺乏真正影响力的责任，而那些符合公司对领导者外貌期望的人会以某种方式向上爬。

### 停止追逐幽灵

许多人会认为，公司作为理性的优化者，会避免在低影响、高投入的项目上花费太多时间。不幸的是，情况并非总是如此。令人惊讶的是，一位新的高级领导加入公司后，立即推动一项从根本上误解当前挑战的战略转变，这种情况非常普遍。他们之前情况的幽灵[从根本上转变的策略 误解.](https://lethain.com/grand-migration/)

对他们理解新公司的情况有着坚定的掌控，以至于他们错误地将熟悉的事物视为本质。

作为一名高级领导，你必须控制住你的自负，以避免大规模地投资于毫无意义的工作。当你在招聘过程中，被反复告知你被聘用来解决一些根深蒂固的问题时——你是新聘请的救世主，这可能会非常具有挑战性。当然，你的直觉是正确的！在改变现状之前花时间去理解现状，总是会以结果回报你的努力。

我最近与某人进行了一次讨论，他认为新的高级领导人即使怀疑这些努力会失败，也会故意推动重大变革。这些变革使组织越来越依赖于新的领导人，并且还确保任何进展顺利的事情都直接归功于新的领导人，而不是他们的团队。如果这就是你领导的方式，请知道你很糟糕，并花时间提升自己，直到整个公司的福祉和成功对你来说比被认为至关重要更重要。

### 生存问题

既然你已经完成了吃零食、献媚和追逐幽灵，现在是时候从另一个方向开始思考了：你应该做什么？寻找重要工作的第一个地方是探索你的公司是否正在经历生存风险。公司在一个永恒的迭代淘汰赛中运作，在未来的成功和生存到未来成为现在之间取得平衡。如果你即将输掉其中一轮，那么始终关注那里。[迭代 锦标赛,](https://lethain.com/iterative-elimination-tournaments/)

资金耗尽，就像我在Digg的经历一样，可能是最明显的问题，但并非每个生存问题都是财务问题，比如Twitter的失败鲸鱼稳定性挑战或适应Covid-19大流行带来的变化。[就像我在Digg的经历一样](https://lethain.com/digg-v4/)[Twitter的失败鲸鱼 稳定性挑战 或适应由Covid-19大流行引起的转变。](https://www.theatlantic.com/technology/archive/2015/01/the-story-behind-twitters-fail-whale/384313/)

如果你的公司正在发生可怕的事情，那么你就应该积极参与其中。如果这个问题得不到解决，其他一切都将变得无关紧要。

### 在有空间和关注的地方工作

存在性问题通常不是投入精力最有效的地方，但当四面楚歌时，效率并不是首要考虑因素。你应该涌向存在性问题，但如果一个问题不是存在性的，那么你应该怀疑是否要将你的精力投入到每个人都已经关注的地方。人们经常追逐领导的首要任务，但由于有这么多人希望在那里产生影响，因此通常很难产生有意义的影响。

相反，最有效的工作场所是那些对你的公司很重要，但仍有足够的空间来实际开展工作的地方。哪些是未来将变得至关重要的优先事项，你可以在提前做好出色的工作？哪些领域做得还行，但在你的支持下可以做得更好？

有时你会发现一些工作值得关注，但组织却无法关注，通常是因为其领导不重视这项工作。在一些公司，这是开发者工具方面的工作。在另一些公司，这是包容性工作。在大多数公司，这是粘合性工作。[粘合工作](https://noidea.dog/glue)

几乎总是有大量的空间来做这种无人关注的工作，因此你将能够在此方面取得快速的初步进展，这感觉像是一个值得投资的好机会。但在某个时候，你会发现这项工作需要支持，而对于一家旨在忽视或贬低的工作来说，获得支持是相当具有挑战性的。你早期的胜利将慢慢被冷漠和不协调所侵蚀，你最初的影响将被时间的流沙所吞噬。

这是否意味着你不应该做包容性工作？不，这不是我希望你从中得出的结论。有时，一个组织不关注的领域非常重要，以至于你会希望倡导它开始关注。教会一家公司重视它不在乎的东西是你所能做的最困难的工作，而且它经常失败，所以你应该尽可能少做，但不能更少。作为一名高级领导者，你拥有一项超越最大化公司认知影响的道德义务，但重要的是要认识到你所面临的困难，并相应地安排你的努力。

### 促进成长

一个经常投资不足（例如，有很多工作空间）同时又具有高度杠杆作用的领域是发展你周围的团队。招聘有很多人参与其中，通常是在优化招聘渠道方面，但入职、指导和辅导在许多公司完全被忽视，尽管它们至少与招聘对你公司的工程速度一样有影响力。[招聘漏斗](https://lethain.com/hiring-funnel/)[与招聘对公司工程速度的影响一样大](https://lethain.com/productivity-in-the-age-of-hypergrowth/)

如果你开始每周花几个小时来发展你周围的团队，那么很可能这将成为你的遗产，远在你技术规范和拉取请求被遗忘之后。

### 编辑

令人惊讶的是，许多项目只需一个小小的改变就能成功，只需一个快速的修改就能开启新的机会，或者只需一次对话就能达成共识。我认为进行这些小的改变、快速的修改和简短的对话就是*编辑*你的团队的方法。

凭借你的组织特权、你在公司建立的关系以及你从经验中获得的预见能力，你通常可以通过投入最小的努力来改变项目的结果，这是你能做的最有价值的工作。

它特别有价值，因为它快速、简单，对你和你帮助的人都具有高度的激励作用，并且如果做得好，影响巨大。（而且，如果做得不好，会非常消极，所以你的方法很重要！）

### 完成事情

一种特殊的编辑方式是帮助完成一个无法自行结束的项目。通常，你会遇到一位职业生涯早期的才华横溢的工程师，他们已经在做这项工作，但无法获得支持或弄清楚如何将他们的项目重新调整为可完成的工作。令人惊讶的是，指导队友如何将项目调整为可完成的项目，然后利用你的特权来消除正确的摩擦点，会将一个为期六个月的苦差事变成一个为期两周的冲刺，并且产生几乎相同的影响。

我们[仅](https://www.amazon.com/dp/B078Y98RG8/)，并且让一个项目完成

这条线是它从风险转变为杠杆作用的神奇时刻。花在完成工作上的时间总是值得的。

### 只有你能做什么

最后一类重要的工作是你独一无二能够完成的工作。当然，有些工作你比其他人做得更快或更好，但更重要的是，如果你不做，这些工作根本不会发生。

这项工作是你非常擅长的事情和你真正关心的事情的交集。它可能是[写作](https://lethain.com/magnitudes-of-exploration/)，人们*实际上会遵循*，它可能是说服一位优秀的候选人加入，它可能是改变你首席执行官关于如何偿还技术债务的想法，它可能是[正在精心设计一个具有辨识度的API](https://increment.com/apis/api-design-for-eager-discering-developers/)。

无论它是什么，如果你不做，有些事情就不会发生，这是你最大的机会去做一些重要的事情，而且随着你职业生涯的深入，这个类别会变得越来越窄、越来越深。

### 为什么重要

假设你正在面试一个新职位，而你已经工作了二十年。面试你的人会理解你对之前任何项目或公司的真正影响吗？不，我保证他们不会。相反，你会发现自己被一系列令人惊讶的主观标准评判：你积累的声望、你拥有的头衔和你工作过的公司、你的幕后声誉，以及你在面试过程中的表现。

你无法逃避主观的面试方式，但你可以通过做有价值的工作来有意识地积累专业知识。事实上，这是你职业生涯中唯一可行的长期赌注：专注于重要的工作，做能发展你的项目，并朝着重视真正经验的公司前进。

## 编写工程战略

*我有点觉得写关于工程战略的文章很难，因为好的战略非常枯燥，而且写起来也很无聊。而且我认为当人们听到“战略”时，他们会想到“创新”——*[*卡米尔·富尼耶*](https://twitter.com/skamille/status/1328763503973429250)

很少有公司了解他们的工程战略和愿景。这种不确定性的一个后果是，业界认为这些文件很难编写。在某些对话中，你可能会觉得自己在谈论一些神秘的东西，但这些只是平凡的文件。现实情况是，好的工程战略是枯燥的，而且编写一个有效的战略比编写一个糟糕的战略要*容易*。

要编写工程战略，编写五个设计文档，并提取出相似之处。这就是你的工程战略。要编写工程愿景，编写五个工程战略，并预测它们在两年后的影响。这就是你的工程愿景。

如果你无法抗拒在过程中加入你最精彩的想法的冲动，那么你可以将它们包含在你的预备工作中。把你所有最好的想法写在一份巨大的文档中，删除它，并且永远不要再提及它们。现在这些想法已经从你的脑海中消失了，你的头脑已经为接下来的工作清空了。

持久有用的工程战略和愿景是迭代的、自下而上的组织学习的成果。因此，所有的学习都有助于你所在组织的战略和愿景，但你的贡献不必如此抽象。即使你没有直接负责这项工作，也有一些实际的步骤，*你*可以采取这些步骤来推进你所在组织的战略和愿景，从*现在*开始。

### 何时以及为什么

在深入探讨创建有效战略和愿景的秘诀之前，一个好的起始问题是：“我应该在何时以及为何实际创建它们？”战略是主动调整的工具，它能使团队快速且自信地行动。

战略使每个人——不仅仅是被授权的少数人——能够快速做出自信的决策，否则这些决策可能会花费他们一周的讨论时间。战略也是缩小你众多可能未来的基石，足以让你写出一个现实的愿景。如果你意识到你已经重复讨论了三四次相同的问题，那么就该制定一项战略了。当未来太过模糊，无法确定值得进行的投资时，就该制定另一个愿景了。如果这些问题听起来都不熟悉——现在就去做其他工作，稍后再回来。

### 编写五个设计文档

设计文档描述了你在特定项目中做出的决策和权衡。你的公司可能会称它们为 RFC 或技术规范。也可能出现更奇怪的名称；优步令人困惑地称它们为 DUCKS，直到后来他们统一使用 RFC。一个好的设计文档描述了一个具体的问题，调查了可能的解决方案，并解释了所选方法的细节。有很多格式可供选择；一些可以开始思考的地方是Design Docs, Markdown, and Git、Design Docs at Google和Technical Decision- Making and Alignment in a Remote Culture。[标准化于RFC](https://blog.pragmaticengineer.com/scaling-engineering-teams-via-writing-things-down-rfcs/)[设计](https://caitiem.com/2020/03/29/design-docs-markdown-and-git/)[文档、Markdown和Git, 谷歌的设计文档，和技术决策- 在远程文化中进行决策和协调.](https://multithreaded.stitchfix.com/blog/2020/12/07/remote-decision-making/)

一个给定的项目是否需要设计文档取决于个人判断，但我发现了一些有用的规则。对于任何其功能将被许多未来项目使用的项目，都应该编写设计文档。对于任何对用户产生有意义影响的项目，也应该编写设计文档。对于任何需要超过一个月工程时间的工作，都应该编写设计文档。

一批五个设计文档是编写有效战略的理想素材，因为设计文档具有糟糕战略所缺乏的东西：基于现实的详细细节。同一团队中两个善意的工程师很容易以不同的方式解释一个抽象的战略，但是当你在实施一个具体的解决方案时，很难保持不一致。

在编写时的一些建议：

**从问题开始。**问题陈述越清晰，

解决方案就越明显。如果解决方案不明显，则花更多时间澄清问题。如果你无法清楚地表达问题，请将你所拥有的东西展示给五个人，并询问他们缺少什么：新鲜的视角总是能看到真相。

**保持模板简单。**大多数公司都有一个设计文档模板，这是一个很好的遵循模式。但是，这些模板通常会扩展到服务于太多的目标。过度加载的模板首先会阻止人们编写设计文档。首选最小的设计文档模板，该模板允许作者选择最有用的部分，并且仅坚持对风险最高的项目进行详尽的描述。

一起收集和审查，独自撰写。你个人不太可能拥有所有相关的背景信息，从而撰写出关于特定主题的最佳设计文档。在深入研究这个过程之前，从具有相关观点的人那里收集意见，特别是那些将依赖你的设计文档输出的人。然而，对于将这种协作过程带入设计文档本身的编写，要持怀疑态度。大多数人更擅长写作，而不是编辑。这意味着通常情况下，将一份集体文档编辑成清晰的文字，比确定一位作者来撰写一份清晰的文档更难。广泛收集观点，但独自撰写。只是要小心，在你与他人一起审查之前，不要爱上你所写的东西。

**宁要好，不要完美。**写出一份好的文档并让别人看到，总比为了稍微好一点而拖延要好。在对别人的设计提供反馈时，这一点尤其重要；很容易陷入期望他们的设计和你最好的设计一样好的陷阱。特别是当你变得更资深时，强迫每个设计都达到你自己的最佳作品的标准是有害的。专注于推动设计变得更好，而不是固执于你自己的最佳作品作为相关的质量标准。

写出优秀的设计文档需要大量的实践。如果你想改进你的设计文档，我最好的建议是，在你完成实施后重新阅读你的设计，并研究你的实施偏离计划的地方——是什么导致了这些偏差？哦，当然，继续写更多的设计文档。

### 将这五个设计文档综合成一个策略

在你的组织编写了五个设计文档后，坐下来一起阅读它们。寻找在多个设计中出现的有争议的决定，特别是那些难以达成一致的决定。我最近的一个例子是，陷入关于 Redis 是否适合作为持久存储还是仅作为缓存的争论。与其在每个设计文档评审中从零开始，如果我们回顾一下我们最近关于使用 Redis 的决定，反思我们是如何做出这些决定的，并将它们写成一个策略，岂不是更容易？

好的策略会指导权衡，并解释该指导背后的理由。糟糕的策略会陈述一项没有解释的政策，这会将它们与制定它们的背景脱钩。没有背景，你的策略会迅速变得难以理解——他们为什么会做出这个决定？——并且随着底层背景的变化而难以适应。在思考如何编写自己的策略时，可以阅读一些有趣的策略，例如A Framework for Responsible Innovation和HowBig Technical Changes Happen at Slack。[负责任的创新框架 和如何 Slack 的重大技术变革是如何发生的.](https://slack.engineering/how-big-technical-changes-happen-at-slack/)

如果你是[好战略，坏战略](https://www.amazon.com/dp/B004J4WKEC)的皈依者——这本书完全改变了我对策略的看法——那么你会注意到这个策略的定义是“诊断”和“指导政策”部分，将“连贯行动”推迟到设计文档中。

我写策略文档的最佳建议是：

**从你所在的地方开始。**在制定策略时，我们很容易被我们所处的固有巨大模糊性所麻痹，但你只需要投入并开始写作。等待缺失的信息是行不通的：每个缺失的文档都有其充分的理由。你写的任何东西都需要改变，如果你写了一些特别糟糕的东西，你会很快意识到需要改变它。你现在所在的地方永远是最好的起点。

**写出具体内容。**一直写到你开始概括，然后停止写作。如果你不能具体说明，那就等到你写了更多的设计文档。具体的陈述会产生一致性；一般的陈述会产生一致性的错觉。

**要有主见。**好的策略是有主见的。如果不是，

如果他们固执己见，就不会为决策提供任何明确性。然而，仅仅固执己见是不够的。你还需要展示你的工作。

**展示你的工作。**在成长过程中的数学课上，你必须展示你的解题过程

才能获得全部学分。在这里也是一样，你必须展示你观点背后的理由。展示你的工作可以建立对文档第一个版本的信心，但更重要的是，通过展示你的工作，你可以让其他人修改和扩展你的工作，因为底层环境会发生变化。

你写的一些最好的策略，当时可能觉得太明显而不值得写。“我们应该什么时候编写设计文档？”是一个值得写的策略。“我们对哪些用例使用哪些数据库？”是一个值得写的策略。“我们应该如何将我们的迁移从单体架构分阶段过渡到服务？”也值得写。当我们抛弃将策略作为卓越表现的想法时，我们可以开始编写更多的策略，并且可以更随意地编写它们。如果最终没有被使用，你总是可以在以后弃用它。

### 将五个策略推断为一个愿景

当你收集到更多的策略时，要推断各种策略如何相互作用将变得越来越具有挑战性。也许你的一个策略是[运行 更少 和](https://www.intercom.com/blog/run-less-software/)

选择你最近的五个策略，推断它们在未来两到三年内的权衡结果。当你编辑这些矛盾并将这些线索编织在一起时，你就写下了一个工程愿景。最终版本将给你[坦尼娅](https://twitter.com/whereistanya)所说的[稳健的](https://leaddev.com/technical-direction-strategy/sending-gifts-future-you)，这使得更容易理解你现有的策略如何相互关联，并简化了编写经得起时间考验的新策略。

对于一个有用的愿景，需要关注以下几点：

**写出未来两到三年的内容。**公司、组织和技术变化得足够快，以至于对未来的过度思考是令人担忧的。如果你写一个在六个月内过期的愿景，那也是行不通的——你实际上会在六个月的时间窗口内写多少策略？尽量关注未来两到三年；如果你是一家相当成熟的公司，你可以稍微扩大这个范围。

**立足于你的业务和你的用户。**有效的愿景立足于服务你的用户和你的业务。这种紧密的联系使愿景与你的领导团队的核心价值观——用户和业务保持一致。糟糕的愿景将技术复杂性视为一种自我辩解的存在理由——这种观点永远不会被你公司的领导层所认同。

**保持乐观，而不是大胆。**愿景应该是雄心勃勃的，但不应该是大胆的。它们应该是可能的，如果可能的话，应该是最好的版本。写下如果每个项目都能按时完成且没有重大挫折，你能完成什么。不要写你认为在无限资源下可能实现的事情。

**保持具体和明确。**愿景越具体就越有用。笼统的陈述很容易达成一致，但无助于调和相互冲突的策略。要比你觉得舒服的程度更详细一点。愿景中的细节通常是说明性的，而不是声明性的，它能让你品尝到未来的味道，而不是提供具有约束力的承诺。

**保持在一到两页的长度。**现实情况是，大多数人不会阅读长篇文档。如果你写的东西有五六页长，读者会在没有读完的情况下就开始放弃（或者会非常快速地浏览，而不会关注细节）。强迫自己写一些简洁的东西，并通过链接到其他文档来引用额外的上下文，供那些想要完整细节的人参考。

在你完成愿景的写作后，人们通常采取的第一步是在整个工程组织中广泛分享它。愿景背后有很多工作——每个策略有五个设计文档，一个愿景有五个策略——当你完成时，很难*不*感到兴奋。如此兴奋以至于很容易感到沮丧，因为对你的策略的反应几乎总是很平淡。反应平淡有几个原因。首先，你的愿景的核心受众是编写策略的人，这是一个相对较小的群体。其次，一个伟大的愿景通常是如此*显而易见*，以至于它带来的更多是无聊而不是兴奋。

不要以最初的兴奋程度来衡量愿景。相反，通过阅读两年前的设计文档和上周的设计文档来衡量它；如果有了显著的改进，那么你的愿景就是好的。

## 管理技术质量

*当我可以帮助改进一个善意的、解决了实际需求的提案时，我感到特别有影响力，但起草该提案的团队缺乏经验或背景来编写一个好的计划来抓住这个机会。在这种情况下，拥有一个结构良好的计划可以帮助大大缩小范围，同时获得大部分价值，从而更快地展示影响力。-* [*德米特里·彼得拉什科*](#_bookmark42)

如果说工程师、工程经理和技术主管们有什么可能达成一致的事情，那就是存在技术质量危机。一种容易识别的诊断和治疗方法是：我们的工程师没有优先考虑质量，我们需要雇佣更好的工程师或重新培训我们现有的工程师。当然，如果你觉得更舒服，你可以随意用“产品经理”或“高管”来代替“工程师”。这是一个引人注目的叙述，有一个明确的恶棍，并且它方便地将责任从工程领导身上转移开。然而，就像大多数将责任转移到权力最小的人身上的叙述一样，它既没有帮助，也是错误的。

当你接受技术质量低是由于糟糕的决策造成的这一前提时，你就会开始寻找糟糕的判断，而公司里的某个人一定是罪魁祸首。是前任首席技术官吗？是那个带着紧张微笑看着你的工程师吗？是所有人吗？如果都不是这些人呢，更奇怪的是，甚至也不是你的错呢？

在大多数情况下，低技术质量不是危机；它是预期的、正常的状态。工程师们通常在做出质量决策时会做出合理的决策，而成功的公司会随着规模的扩大、转型或向上转向企业用户而提高他们的质量标准。在一个运转良好且成功的公司里，你之前的大多数技术决策都不会达到你当前的质量阈值。弥合当前和目标技术质量之间的差距，不是失败，而是有效的工程领导的常规、必不可少的部分。

### 问题

作为一个工程领导团队，你们的目标是在保持适当的技术质量水平的同时，尽可能地将精力投入到核心业务中。你们必须平衡多个时间范围内的质量，而这些时间范围通常具有冲突的需求。例如，为了赶上下周的截止日期，你将做非常不同的工作来完成那个关键的合作，而不是构建一个平台，以便在下个季度更快地启动十倍。

正如贵公司的技术质量标准会随着时间的推移而变化一样，您管理技术质量的方法也会同步发展：

1. 修复**导致直接问题的**热点
2. 采纳**已知可以提高质量的**最佳实践
3. 优先考虑**在软件更改时保持质量的**杠杆点
4. 调整**组织如何更改软件的**技术向量
5. **衡量技术质量**以指导更深入的投资
6. 启动一个**技术质量团队**来创建质量的系统和工具
7. 运行一个**质量计划**来衡量、跟踪和创建责任制

当我们深入研究这个方法工具包时，请记住选择最便宜、最直接且可能有效的工具。技术质量是一场长期的游戏。没有所谓的胜利，只有学习和赢得继续玩下去的机会。

### 登上楼梯

深入研究手头的挑战，直到找到一个值得解决的普遍性问题，这其中有一种特别的乐趣。然而，同样重要的本能是快速解决当前问题，然后继续处理下一个紧迫的问题。

在考虑为你的团队和组织做出正确的质量改进时，通常最有效的方法是从最轻量级的解决方案开始，只有在早期努力在规模的压力下崩溃时，才转向大规模的解决方案。如果你不能让团队采用正确的代码检查，你试图推出一个全面的质量计划注定要失败。虽然后者在规模上可能更有效，但它们要困难得多。

执行。

所以，先做简单的事情！

即使它不起作用，你也会从未能推出简单的东西中学到更多、更快，而不是未能推出困难的东西。然后你就能更快地进入改进后的第二个迭代。随着时间的推移，你将朝着全面的方法发展，但没有必要着急。不要为了企业级协调的风险而放弃早期组织的轻松、快乐和纯真，而没有适当的需求。

将这些阶段呈现为要攀登的线性阶梯是很方便的，但现实中的组织很少这样使用它们。你更有可能修复一个质量热点，推出一个最佳实践，开始运行一个架构审查，废除该架构审查，然后回到热点修复一段时间。过早的流程增加的摩擦比价值更多，并且很快就会暴露自己是无效的。如果某些东西不起作用，尝试让它工作一段时间，然后庆祝它的消亡。

### 热点

当面对质量问题时，第一反应通常是识别出一个必然需要流程解决方案的流程失败。如果部署导致中断，那是因为作者没有正确遵循代码测试流程，所以现在我们将要求每次提交都进行测试——这将教会那些懒惰的开发人员！

有一个关于萨班斯-奥克斯利法案的老笑话：它并没有降低风险；它只是清楚地表明了事情出错时该责怪谁。不幸的是，这个笑话在没有幽默感的情况下适用于许多组织推出流程的方式。问责制有其作用，但更重要的是理解手头的问题并尝试直接解决它，而不是创建流程驱动的问责制。

流程的推出需要人们改变他们的工作方式，你不应该轻易地进行。与其寻求流程改进，不如从戴上性能工程师的帽子开始。衡量手头的问题，确定问题发生的主要区域，并精确地关注该区域。

之前未经测试的部署示例可能会受益于向部署工程师直接反馈，让他们改变测试习惯。或者，也许您最好承认您的软件设计容易出错，并采用《软件设计哲学》中描述的“消除错误”的方法。[软件设计哲学](https://www.amazon.com/Philosophy-Software-Design-John-Ousterhout/dp/1732102201)

如果您遇到开发速度问题，可能是优化测试运行时间，将您的 Docker 编译步骤移到RAM 磁盘上，或者使用《软件设计 X 射线》中描述的技术来找到需要改进的特定文件。[内存盘](https://en.wikipedia.org/wiki/RAM_drive)[软件设计X射线](https://www.amazon.com/Software-Design-X-Rays-Technical-Behavioral-ebook-dp-B07BVRLZ87/dp/B07BVRLZ87/)

[系统](https://lethain.com/systems-thinking/)系统思考是我在职业生涯中遇到的最具变革性的思考技术。不过，有时它也可能像海妖一样引诱你修复当前可能最好丢弃的系统。当然，您可以推出一个新的培训计划来教您的团队如何编写更好的测试，但或者，也许您可以直接删除 98% 的测试失败发生的那一个测试文件。这就是优先处理热点的非凡有效性，也是为什么它应该是您用来提高技术质量的首要技术。

在某个时候，您可能会发现您的组织产生质量问题的速度快于您修复热点的速度，那时就应该开始采用最佳实践了。

### 最佳实践

我曾经在一家没有团队计划流程的公司工作。随着时间的推移，工程主管对无法预测目标日期越来越沮丧，并强制我们使用[Scrum（一种敏捷开发框架）](https://en.wikipedia.org/wiki/Scrum_(software_development))。在强制执行后，一位经理在 wiki 上写下了 Scrum 流程。有人宣布我们正在使用 Scrum。经理们告诉他们的团队使用 Scrum。任务完成！

当然，没有人开始使用 Scrum。每个人都继续做他们以前做的事情。承认错误是很尴尬的，所以工程主管宣布采用是一项重大胜利，没有人忍心说不同的话。

这个悲伤的故事反映了许多公司试图推广最佳实践的方式，这也是最佳实践声誉不佳的原因之一。理论上，

组织应该在修复质量热点之前采用最佳实践，但我建议在热点分析之后再采用实践。采用最佳实践需要一定程度的组织和领导成熟度，这需要一些时间来培养。

[当好的 进化了 而不是强制执行。研究其他公司如何采用类似的措施](https://lethain.com/good-process-is-evolved/)

同样重要的是限制并发流程推广的想法。如果你试图让团队同时采纳多个新实践，你就是在与自己争夺他们的注意力。如果你正在考虑恢复或修改其中一项新实践，这也会使以后更难评估影响。这有点严厉，但我逐渐相信，你应该在任何给定时间将自己限制在单一的最佳实践推广上。将你所有的精力集中在一个实践的成功上，而不是将资源分散在少数几个实践上。

一次只采用一种新的实践也能迫使你认真思考应该优先考虑哪一种。选择你的下一个流程听起来很容易，但通常不清楚哪些最佳实践是真正意义上的最佳实践，哪些只是熟悉的或著名的。真正的最佳实践必须有研究支持，而关于这个主题的最佳研究来源是[加速](https://www.amazon.com/dp/B07B9F83WM/)。

虽然《加速》的所有建议都是数据驱动的，而且非常好，但我发现早期采用最有帮助的少数建议是版本控制、基于主干的开发、CI/CD和生产可观察性（包括开发人员负责他们编写的系统），以及以小的、原子性的更改进行工作。我有很多其他的实践想要倡导（谁没有花费一个职业生涯来倡导[更好](https://increment.com/documentation/why-investing-in-internal-docs-is-worth-it/)），但我不再像以前那样信任我的直觉了。

当你被太多的热点问题压得喘不过气来时，就会从修复热点问题过渡到采用最佳实践。下一个转变，从最佳实践到杠杆点，发生在你发现自己想要采用

一个新的最佳实践，而你正在进行中的最佳实践还没有奏效。与其[增加](https://lethain.com/limiting-wip/)，不如继续使用下一个工具。

### 杠杆点

在“热点问题”部分，我们讨论了如何利用性能工程师的思维方式来识别需要修复的正确问题。优化对于你已经存在的问题非常有效，但它有意地不适用于未来：性能工程最糟糕的罪过是将精力投入到未经证实的问题上。

然而，当你观察软件如何随时间变化时，在少数几个地方，额外的投资可以长期保持质量，既可以防止严重的质量问题，又可以降低未来质量投资的成本。

我称这些为质量杠杆点，而三个最具影响力的点是接口、有状态系统和数据模型。

*接口*是系统之间的契约。有效的接口将客户端与封装的实现解耦。持久的接口暴露所有底层本质的复杂性，而不暴露任何底层偶然的复杂性。令人愉悦的接口是[热切地辨别，辨别地热切](https://increment.com/apis/api-design-for-eager-discering-developers/)。

*状态*是任何系统中最难改变的部分，而这种对改变的抵制使得*有状态系统*成为另一个关键的杠杆点。状态比其他系统更快地变得复杂，并且具有一种惯性，使得以后改进它的成本相对较高。当你将围绕安全性、隐私和合规性的业务义务纳入其中时，更改你的有状态系统变得更具挑战性。

数据模型是接口和状态的交集，将你的有状态系统的能力限制在你应用程序认为合法的范围内。一个好的数据模型是严格的：它只暴露它真正支持的东西，并防止无效状态的表达。一个好的数据模型能够容忍随时间的演变。有效的数据模型甚至一点也不聪明。

当你在工作中发现这些杠杆点时，花额外的时间有意识地处理它们。如果是接口，则针对模拟实现集成六个客户端。如果是数据模型，则表示六个真实场景。如果是有状态的，则执行故障模式，检查一致性行为，并建立类似于生产场景的性能基准。

把你学到的一切都融入到一个技术规范文档中，并在你的团队中推广。收集同行的行业反馈。即使在你开始实施之后，也要倾听现实的声音，并保持对变化的开放态度。

投资于杠杆点的隐藏力量之一是，你不需要完全的组织一致性来做到这一点。要编写技术愿景或推出最佳实践，你需要那种支持，这就是我建议从杠杆点开始的原因。但是，如果你已经耗尽了杠杆点带来的可访问的影响，那么可能是时候转向推动更广泛的组织一致性了。

### 技术向量

有效的组织将其大部分努力投入到共同的愿景中。如果你将每个技术决策绘制为网格上的一个向量，那么这些向量指向同一方向的越多，你随着时间的推移所取得的成就就越多。相反，我合作过的一些最令人印象深刻的工程师创造了具有非凡幅度但方向不一致的向量。最终，这些工程师在试图领导他们的组织时损害了他们的组织。

[架构师](#_bookmark6)一种确保技术方向一致的万无一失的解决方案是将所有相关的决策交给同一个人，并在他们的头衔中包含架构师。这种方法效果很好，但难以扩展，而且架构师的决策质量会随着他们远离在实际流程中处理实际代码的实际工作而降低。在另一个极端，你可以允许每个团队做出独立的决定。但是，一个允许任何工具的组织是一个拥有统一不受支持的工具的组织。

用于对齐技术向量的基本工具包括：

**提供直接反馈。**当人们遇到不一致时，第一个答案通常是流程变更，但相反，首先要简单地向你认为不一致的个人提供直接反馈。就像他们错过了你的背景一样，你也错过了他们的背景，一次快速的谈话通常可以避免多年的不必要的过程。

**从技术规范到战略再到愿景改进你的工程战略。**[技术规格，到战略，到愿景](#_bookmark11)

**将你的方法封装在你的工作流程和工具中。**对清晰愿景的文档记录是有帮助的，但有些人就是不会学习你的文档。深思熟虑的工具创造的工作流程，比培训和文档更能培养习惯。例如，配置一项新服务可能需要访问一个网站，该网站要求你添加指向该服务技术规范的链接。另一种方法可能是阻止部署到生产环境，如果该服务没有建立随叫随到的设置，并且当前有人随叫随到，而且该人员还必须启用他们的推送通知。

**在入职培训期间培训新的团队成员。**改变人们已经形成的习惯是相当具有挑战性的，如果你试图让人们采纳新的做法，这会令人沮丧。然而，如果你在人们加入时就让他们朝着正确的方向前进，那么这种习惯性的动力将有助于保持一致。

**使用*康威定律*。**康威定律认为，组织构建的软件反映了他们的结构。如果你的组织结构不良，这将导致紧密耦合或混乱的软件。然而，如果你的组织设计是有效的，它也是一种质量的力量。

[**策划技术变革**使用架构审查, 投资 策略, 和](https://lethain.com/magnitudes-of-exploration/)[采用新工具的结构化流程](https://slack.engineering/how-big-technical-changes-happen-at-slack/)[. 大多数](https://lethain.com/magnitudes-of-exploration/)

无论你使用哪种方法来调整你的技术向量，这都是

一项需要花费数月甚至数年才能完成的工作。你不可能写完愿景文件，组织就能立即支持它的辉煌。更有可能的是，它会堆积灰尘，直到你投入精力来建立支持。

大多数公司可以将上述从热点修复到向量对齐的技术结合起来，形成一种成功的技术质量管理方法，希望这对你来说也是如此。然而，许多人发现这些方法还不够，你需要采取更重量级的方法。在这种情况下，第一步始终是测量。

### 测量技术质量

[加速](https://www.amazon.com/dp/B07B9F83WM/)在软件工程中，测量的愿望通常超过了我们的测量水平。Accelerate 确定了衡量速度的指标，这些指标对于定位流程和工具问题非常有效，但这些指标在代码合并后才开始。你如何衡量代码库的质量，以便识别差距，提出行动计划，并评估你为改进所做的努力的影响？

有一些过程测量与有效的变更相关。例如，你可以测量每个拉取请求中更改的文件数量，因为较小的拉取请求通常质量更高。你还可以测量代码库中每个文件的代码行数，假设非常大的文件通常难以扩展。这些都可能很有帮助，我甚至建议测量它们，但我认为它们充其量只是代码质量的代理测量。

我的经验是，有用地测量代码质量是可能的，这归结为制定一个非常精确的质量定义。你对质量的定义越详细，衡量代码库就越有用，对希望提高其工作领域质量的人来说就越有指导意义。这种方法在Building Evolutionary Architectures和Reclaim unreasonable software中有详细描述。[构建进化架构 和收回不合理的 软件.](https://lethain.com/reclaim-unreasonable-software/)

一些具有代表性的组件，可以考虑将其纳入您的质量定义中：

代码中静态类型的百分比是多少？有多少文件有关联的测试？

您的代码库中的测试覆盖率是多少？

模块之间的公共接口有多窄？

有多少百分比的文件使用首选的 HTTP 库？

端点是否在冷启动后 500 毫秒内响应请求？

有多少函数具有危险的写后读行为？或者对主数据库实例执行不必要的读取？

有多少端点在单个事务中执行所有状态变更？有多少函数获取低粒度锁？

有多少热点文件存在于超过一半的拉取请求中被更改？

您可以不同意某些属性应该存在于您的代码库的质量定义中：您的定义应该特定于您的代码库和您的需求。重要的是制定一个精确的、可衡量的定义。在制定该定义的过程中会有分歧，而且您必然会随着时间的推移而改变该定义。

在你完成了定义之后，这是一个工具可能真正具有挑战性的领域，而工具是有效指标的必要条件。工具的复杂性是在实践中采用这些技术最大的摩擦点，但是如果你能克服它，你就能解锁一些非常棒的东西：一个真实的、动态的质量评分，你可以随着时间的推移进行跟踪，并用它来在你方法中创建一个清晰的对齐，而概念上的对齐无法做到。

在定义和衡量了质量之后，你的下一步是在投资*质量团队*或*质量项目*之间做出决定。一个专门的团队易于协调，其带宽是可预测的，并且通常是更容易的起点。

### 技术质量团队

一个*技术质量团队*是一个致力于在你的代码库中创造质量的软件工程团队。你可能会称这个团队为开发者生产力，

开发者工具或产品基础设施。在任何情况下，该团队的目标都是在你的公司的软件中创建和保持质量。

这与有时被称为质量保证团队不同。虽然两个团队都对测试进行投资，但技术质量团队的职责范围更广，从工作流程到构建到测试到界面设计。

当你启动这样一个团队时，从三到六人的固定团队规模开始。拥有一个小型团队会迫使你毫不留情地优先考虑他们对影响的路线图，并确保你将保持对可实现目标的关注。随着时间的推移，这个团队将积累需要扩大投资才能维护的系统，Jenkins集群就是一个常见的例子，并且你会希望[尺寸](https://lethain.com/sizing-engineering-teams/)作为更广泛的工程组织的职能。经验法则在这里很棘手，但也许每十五个产品工程师就有一名工程师从事开发者工具工作，此外还有你的基础设施工程投资。

这些团队很少有产品经理，通常是一个或多个Staff+工程师，以及工程经理合作伙伴来填补这个角色。有时他们会聘用技术项目经理，但通常是在他们进入运营*质量项目*时，如下一节所述。

当启动和运营这些团队之一时，一些成功的要素是：

1. **信任指标胜过直觉。**你应该有一种方法来衡量每个项目。质量是一个复杂的系统，在这种地方，你的直觉很容易欺骗你。同样，随着你在公司变得越来越资深，你的经验将不再反映大多数其他人的经验。你已经知道粗糙的边缘，如果你发现一个新的边缘，你将是第一个排队获得帮助的人，但大多数其他人不会。指标让你保持诚实。
2. **保持你的直觉新鲜。**代码和流程会随着时间的推移而变化，而你离开构建产品功能的时间越长，你的直觉就会变得越迟钝。大多数人发现团队嵌入和团队轮换是保持你的直觉相关的最佳方式。其他人则监控聊天中的问题，以及与产品开发人员进行健康的1:1讨论。

最优秀的人会同时做到这两点，并随时掌握他们的指标仪表盘。

1. 倾听并向你的用户学习。有一种流行的观点是“品味水平”，这意味着有些人天生就知道什么是好的。在设计有效的质量投资的人中，存在巨大的差异，但这不是一种天生的技能。最优秀的人专注于深入理解他们的用户想要完成什么，并且优先考虑用户需求而不是实施约束。

你的工具的采用率和可用性比原始能力更重要。一个难以使用的强大工具只会吸引少数高级用户，但大多数人会忽略它。放慢速度，把这些细节做好。隐藏所有[偶然的](https://en.wikipedia.org/wiki/No_Silver_Bullet)。观察一位工程师第一次尝试使用你的工具，不要帮助他们。改进不足之处。再做十次！如果你没有对你的工具进行用户研究，那么作为一个质量投资团队，你*注定要失败*。

1. **少做一些事情，但要做得更好。**当你为整个工程组织构建工具时，你做得好的任何事情都会加速整个组织的发展。你做得不好的任何事情，包括几乎完美但有太多粗糙之处的事情，都会拖累所有人。虽然做少数最重要的事情通常比做许多平庸的项目更有贡献，但在你试图将工具和工作流程推广到整个组织的情况下，这一点尤其如此（组织过程中的限制仍然适用！）。
2. **不要囤积影响力。**集中式质量团队和他们支持的团队之间存在着根本的紧张关系。通常情况下，集中式团队倾向于采用全局最优的方法，但这会严重影响在非典型领域或工作负载上工作的部分团队。一个典型的例子是一家公司用JavaScript编写其后端服务器，并且不允许其机器学习工程师使用Python生态系统，因为他们不想支持两个生态系统。另一个例子是一家公司标准化使用REST/HTTP2/JSON用于所有API，而一个特定的团队想要使用gRPC。这里没有完美的答案，但重要的是要建立一个深思熟虑的方法

[那 平衡 这 福利 的 探索 反对 这 福利 的 标准化.](http://lethain.com/magnitudes-of-exploration/)

一个使用上述方法的成功的技术质量团队，其生产力*无疑*会高于相同数量的工程师直接从事产品工程工作。实际上，折现的开发者生产力（本着[现金流量折现](https://en.wikipedia.org/wiki/Discounted_cash_flow)的精神）是衡量这种团队影响的理论上正确的方法。仅仅是理论上，因为这样的计算主要是在评估你的自信心。

即使你非常成功，你仍然会有一堆你想要承担但没有带宽完成的高影响力工作。组织不会做出纯粹理性的团队资源决策，你可能会发现你缺乏带宽来完成重要的项目，并且也无法获得批准来雇用更多的人加入你的团队。

当你的团队有比你能承担的更多的高影响力工作时，这是一个好兆头：如果你不选择要承担的项目，那么你考虑得不够广泛。这意味着如果你有一个待办事项列表，你不应该一定尝试扩大你的技术质量团队。但是，如果你发现有一些关键的质量工作你无法完成，那么可能是时候探索启动一个*质量计划*了。

### 质量计划

一个*质量程序*根本不是计算机代码，而是一个由专门的团队领导的倡议，旨在维护整个组织的技术质量。质量程序承担了实现组织目标软件质量水平的广泛职责。这些相对来说并不常见，但你可能遇到过的类似情况是，一个事故程序负责公司的事故回顾和补救。

运行高质量程序的技术组成部分是上面讨论的那些，所以在这里我们将重点关注如何有效地管理程序。你的第一步是找到一位技术项目经理，他可以共同领导该程序并操作其机制。虽然你可以在没有技术项目经理的情况下，在组织程序的信息方面取得相当大的进展，

但这是一个陷阱。你会被在一个大型组织中独自驱动程序的协调开销压垮。

[运营组织项目是关于一个已经被广泛讨论的话题 书面的，但核心方法是：](https://lethain.com/programs-owning-the-unownable/)

1. **确定一个项目发起人。**如果没有一个有权力的发起人，你无法改变一个组织的行为。组织的行为方式是因为这是他们当前约束条件下的最佳解决方案，如果没有有影响力的人的支持，你无法改变这些约束条件。
2. **生成可持续的、可重复的指标。**通常情况下，运行一个项目的人每周要花费四个多小时手动维护他们的数据集。这是行不通的。你的数据会有漏洞，你将无法在后续步骤中将你的数据与自动化集成，而且你将耗尽精力来完成真正改变的工作；刷新指标仪表板没有内在价值。
3. **为每个受影响的团队确定项目目标，并为他们实现这些目标提供明确的路径。**你的项目必须为每个受影响的团队确定具体目标。例如，减少他们测试中的测试不稳定或更快地完成事件修复。然而，至关重要的是，你要提供成功的路线图！很多项目要求其他团队参与，但没有提供关于他们如何完成自己部分的明确指示。项目负责人是主题专家，不要把你的策略卸载给每个团队，让他们独立地重新发明。
4. **构建工具和文档，以支持团队实现他们的目标。**一旦你为团队确定了实现项目目标的明确路径，就要弄清楚你如何帮助他们做出这些改变！这可能是提供“黄金示例”，说明事情应该是什么样子，或者是一个示例拉取请求，将代码中具有挑战性的部分重构为新的模式。这可能是提供一个测试脚本来验证迁移是否正确。这可能是自动生成转换提交以进行测试、验证和合并，而无需工程师自己编写。尽你所能避免每个团队都必须深入理解

你试图取得进展的问题领域。

1. **创建一个目标仪表板并广泛分享。**一旦你将你的项目目标传达给每个团队，提供仪表板，帮助他们了解他们当前的状态、他们的目标状态，并为他们（希望）在前进道路上提供强化反馈。最好的仪表板既是每个团队工作的记分卡，也为每个团队提供关于下一步应该关注哪里的线索。

你的仪表板应该支持三个不同的缩放级别。完全缩小级别可以帮助你评估你的项目的影响。完全放大级别可以帮助单个团队了解他们剩余的工作。介于两者之间的第三个级别可以帮助组织领导者追究他们团队的责任（并支持你的项目发起人提出具体的、明确的要求来追究这些领导者的责任）。

1. **针对那些落后于目标的人，发送程序化的提醒。**人们都很忙。他们不会总是优先考虑你的项目目标。或者，他们可能在改进你所要求的方面做得非常出色，但后来又因为过时的做法而倒退。使用提醒来引导团队关注他们应该为你的项目目标所做的下一步工作。记住，注意力是一种稀缺资源！如果你用提醒邮件或消息浪费了人们的时间，他们就不会关注下一个提醒。
2. **定期与你的赞助人一起审查项目状态。**项目试图在与团队目标不自然一致的组织优先事项上取得进展。许多团队都在努力摆脱本地优先事项，以完成全局优先事项。因此，与你的赞助人一起审查你的总体进展，并引导他们关注那些优先考虑项目工作的团队，这一点至关重要。有效地利用你的赞助人来弥合不一致的优先事项，对你的成功至关重要。

[在这 适用于程序迁移工作.](http://lethain.com/migrations/)

如果你把所有这些步骤都做对了，那么你就是在运行一个真正伟大的项目。这可能会让你觉得工作量很大，而且哇，确实如此：很多项目都出了问题。以下是

导致项目失败的三个主要原因：

1. 纯粹从流程的角度运行，并与你试图完成的现实脱节，
2. 纯粹从技术的角度运行，认为你可以跳过倡导你的目标和倾听你试图激励的人的基本步骤，
3. 试图以一个人的身份涵盖这两种观点——不要单打独斗！

一个糟糕的项目很像一个效率低下的非营利组织：目标是正确的，但很少有资金到达预期的目标。无论你决定如何衡量技术质量，在运行你的质量项目时，最重要的事情始终是记住，项目不是目标。目标是创造技术质量。组织项目规模庞大，并积累了巨大的动力，即使在它们停止工作后，惯性也会推动它们前进。保持你的项目足够精简以便取消，并保持足够的自我批评以便在它停止推动质量创造时取消。

### 从小处着手，慢慢增加

当你意识到你的实际技术水平已经大大落后于你的目标技术水平时，自然的第一个反应是恐慌，并开始推出大量的技术和解决方案。不可避免地，把所有的配料都倒进锅里不会有什么好结果，更糟糕的是，你甚至不知道该保留哪些部分。

如果你发现自己在技术质量方面遇到困难——而且我们经常都会遇到——那就从小处着手，并不断迭代，直到它能够正常工作。然后添加另一项技术，并对其进行迭代。慢慢地构建出真正有效的东西，即使这意味着要承受行动不够快的指责。当涉及到复杂的系统和相互依赖性时，快速行动只是一种表面现象。只有有条不紊的行动才能完成任务。

## 与权威保持一致

*在我的职位上，我们经常连续几周不在同一个房间里，但我仍然必须像我是他的直接代理人一样运作。所以我走进一个房间，心想：“如果马修在这里会怎么做？他会想问什么问题？他对这个问题有什么指导？”因为我不能总是跑回去向他澄清，所以必须对他世界观有深刻的理解。这对于我保持作为他的代表所需的非常深的信任，并有效地执行他的战略和愿景至关重要。人们需要确信，如果马修在那里，我总是会给出与马修相同的答案。-*[*里克·布恩*](#_bookmark36)

一个常见的误解是，权威会让你变得强大。许多渴望担任更高级职位的人认为，他们最终可以按照自己的方式做事。他们认为，头衔本身就创造了灵活性和自主性。他们认为，阻碍他们的摩擦会爆发成一群蝴蝶，消散在风中。

现实情况要微妙得多。

头衔带有被称为组织权威的那种权力，而那种权威是由更大的组织权威借给你的。被授予的东西也可以被收回，而保持组织权威取决于与授予者（通常是你的直接经理）保持深度一致。为了在Staff-plus职位上保持有效，你必须学会与组织权威保持一致的艺术。

### 超越安全网

放弃你剩下的期望，即公司旨在帮助你走向成功。现在*你*是负责帮助公司、你的团队和你的经理走向成功的人之一。

大多数成熟的科技公司都成功地创建了一个可预测的晋升

从职业生涯早期加入公司的人员到获得高级工程师职称的晋升渠道。获得Staff职称的过程通常比之前的职称更复杂，但通常在你的工程经理的支持下完成。在整个过程中，你可能会对你的经理指导你的发展并为你的持续成功提供安全保障感到满意。在达到Staff职位后，你的安全网将不复存在，或者充其量，安全网会很短，以至于你完全有能力跳过它并进入等待着的深渊。当你进一步进入高级Staff和杰出工程师职位时，情况将越来越如此。

Staff-plus 角色是领导角色，而在领导角色中，支持你走到今天的支持系统将会逐渐消失。通常情况下，你会突然发现，你需要协调你周围的一切，以取得自己的成功。

### 在总统的意愿下服务

[里克·布恩](#_bookmark36)当Rick Boone描述他在 Uber 担任基础设施副总裁战略顾问的角色时，他将自己的角色比作《权力的游戏》中的国王之手和《白宫风云》中的Leo McGarry，后者经常说：“我遵从总统的意愿。”在这两个例子中，权力都来自于与更高权力的紧密联系，这对于 Staff-plus 角色来说是一个很好的思维模式。这可能与之前主要通过个人行动和长期影响积累权力的角色相比，是一个困难的转变。[国王之手](https://awoiaf.westeros.org/index.php/Hand_of_the_King)[利奥·麦加里](https://westwing.fandom.com/wiki/Leo_McGarry)

如果你和你的经理已经合作多年，那么在这段时间里，你已经进行了一种微妙的、潜移默化的协调。在其他情况下，一位新的高管会加入，他们熟悉如何支持这些角色，并会带来他们希望如何合作的明确计划。然而，这两种情况在很大程度上不受你的控制，因此，制定你自己的与经理向上协调的方法非常有价值。

为了与你的经理协调，需要关注的一些领域是：

 永远不要让你的经理感到意外。没有什么比让你的经理感到意外更能迅速摧毁信任了。指导一个大型组织通常涉及

同时处理多个项目和问题，而意外会威胁到杂耍者的节奏。重大或频繁的意外也会让人质疑领导者是否真正对他们的组织负责。一般来说，每次你让你的经理感到意外时，都应将其视为一个需要从中吸取教训的事件，并努力防止重蹈覆辙。

不要让你的赞助人感到意外。大多数人对他们的经理抱有极高的期望，例如，他们总是会记得传递与你当前工作相关的信息。经理们会努力做到这一点，他们中的一些人在这方面做得很好，而另一些人则不太擅长。如果你的经理在这方面做得不好，你当然应该给他们反馈，但你也应该采取积极的行动来促进信息的流动。这可能是每周的电子邮件更新或你团队频道内的 Slack 帖子，分享你本周的重点。在 1:1 会议期间，挖掘反馈！询问你是否应该关注其他领域，以及你当前的优先级如何与你的经理的优先级保持一致。如果你们继续让彼此感到意外，那么确定你将使用的控制措施来一起合作。[每周电子邮件更新](https://lethain.com/weekly-updates/)[识别控件](https://lethain.com/identify-your-controls/)

**为你的经理提供背景信息。**如果第一步是避免用你自己的行动让你的经理感到意外，那么下一步就是帮助你的经理不要被更广泛的组织所意外。如果团队对一项新政策感到沮丧，或者你的内部工具无法满足需求，请主动将这些信息反馈给你的经理。明确表示你*不是*给他们带来一个需要解决的问题，而是传递你认为有用的信息。意见是有帮助的，但如果你能找到数据，数据会更有帮助。

有时你会听到有人贬低一位同事，说他们非常擅长“向上管理”。当然，存在一些破坏性的向上管理方式，例如有人控制信息以隐藏问题或歪曲情况，但从根本上讲，向上管理是关于增加你和你的经理之间的带宽并减少摩擦。与你的经理建立一种深思熟虑的伙伴关系，将比在他们没有达到你的期望时感到失望更有用。

### 在不产生过多摩擦的情况下施加影响

作为一名领导者，在发展你自己的观点方面成长的一部分是如何

世界应该运转起来，如果没有这种视角，你就无法达到 Staff-plus 的级别。对事物应该如何运作有一个清晰的认识，可以提高你的判断力，并使你能够积极主动地采取行动。当你达到下一个领导阶层时，你越来越需要将你的愿景与更高级别的组织领导者的愿景融合起来。

你解决这个问题的第一个方法可能是用另一位领导者的愿景取代你自己的愿景，这种方法对某些人有效，但对许多人来说，这意味着放弃了促成他们作为具有强烈判断力的积极主动的领导者取得成功的视角。相反，我建议你提高对你所秉持的价值观与组织运作所依据的价值观之间差异的认识，并找到一种倡导这些价值观的方法，而不会被[被踢出房间](https://lethain.com/getting-in-the-room/)。

人们只能以如此快的速度改变，而组织是由人组成的。如果你在方法上深思熟虑，随着时间的推移，你将能够极大地影响你组织的领导者，但只有当你学会始终保持紧密一致时，你才能获得这段时间。

## 要领导，你必须先跟随

*这是关于采取全球化的思维并将其应用于本地。这意味着将你团队的（技术）倡议/路线图与工程部门范围内的技术战略对齐；并且有意识地决定何时偏离该路径，以满足你团队直接利益相关者的需求。这意味着与你团队的经理合作，采纳其他团队在招聘、入职和生产运营方面的成功实践；并分享你团队中有益于其他团队的实践。这意味着从公司范围内的业务/产品战略中获取背景信息，并将其转化为它如何影响你团队的直接项目 -* [*拉斯·卡萨·威廉姆斯*](#_bookmark31)

多年前，我当时所在的公司聘请了一位新的工程总监，首席技术官在谈论为什么这位新总监是一位了不起的雇员。这位新总监最关键的成就？对领导力和管理之间的区别做出了有史以来最好的解释。事实证明，这不是评估雇员的一种特别有效的方法，但这是一个有趣的话题。

定义领导力和管理是一个被过度涉足的领域，很难再添加更多内容，但大致来说，管理是一种特定的职业，而领导力是一种可以在任何职业中展现的方法。

在过去的几年里，我对领导力的看法有了一些演变，现在主要集中在两个具体的属性上。首先，领导者对事物应该如何运作有一个足够完善的看法，这样他们就可以依靠他们对事物现状和事物应该如何运作之间的区别，来识别主动的、一致的行动，以缩小差距。其次，他们足够关心这个差距，以实际尝试那些缩小差距的行动。

如果你只看到差距而不采取行动，你可能是一个有远见的人，但你是迟钝的。如果你在没有明确目标的情况下采取行动，许多人会认为你是一个领导者，但你的影响将是随机的、武断的和低效的。将两者结合起来，再加上一些运气，很可能会让你的职业生涯走得很远，而这些是

与我合作过的那些成功过渡到Staff-plus工程师或高级管理职位的人所共有的特征。

但这种领导力只能带你走这么远，就我个人而言，我花了*数年*的时间才明白为什么我的领导方式在刚加入公司时为我带来了如此多的早期成功，但随着时间的推移，我的贡献却逐渐被削弱。我慢慢学到的教训是，在你学会如何追随之前，你不可能成为一个有效的长期领导者。

我认为这是过去几年我学到的最重要的一课：最有效的领导者花在追随上的时间比领导的时间多。这个想法也出现在“[第一个追随者创造了一个领导者](https://www.cornerstoneondemand.com/rework/ted-talk-tuesday-how-start-movement)”的观点中，但有效的领导者不会将世界划分为领导者和追随者的二分法，而是与周围的人一起进出领导和追随者的角色。

有很多方法可以将这种方法付诸实践。

1. 清楚地了解你真正的优先事项是什么，不要让自己被所有出现的事情分散精力。如果有些事情你不同意，但只是在很小的程度上，让其他人带头解决它。这里一个有用的问题是，“我们在这里所做的事情在六个月后对我有意义吗？”如果不会，那就抓住机会去追随。
2. 迅速支持其他正在努力改进的领导者。即使你不同意他们最初的方法，一个值得信赖的人领导一个项目几乎总会得到一个好的结果。如果一个值得信赖的人正在领导一个项目，而你仍然不舒服让他们继续前进，考虑一下为什么你缺乏影响他们的能力，以及你是否不擅长给予反馈。
3. 使你的反馈明确地不具有阻塞性。这可以将代码审查评论分类为“可选的细节”，但也可以编写详细的反馈，但将其传递给某人，并提到你希望分享你的观点，而不是必须改变他们的方法。

如果这是你一直在努力的事情，我对此表示同情。我也为此挣扎过。当你拥有足够强大的世界观来领导时，你将开始在你周围聚集其他人，他们依靠你来维持那个世界的物理定律，并且容忍任何偏离你的愿景的行为会让你觉得你让他们失望了。

但这是让你达到一个成功水平但阻碍下一个成功的缩影：持续的增长需要学会将你的世界观融入你周围人的世界观中，加速你周围的整体进步，即使这意味着容忍偏离你的愿景的弯路。

你独自完成的事情远不及你通过创造领导者所能完成的事情。要成为一个伟大的领导者，花时间学习如何追随。

## 学会永远不犯错

*我提出了我认为对我们最有利的情况，人们可能会对此表示不同意。而且，你知道，他们经常这样做。我更多的是引导和影响，而不是说：“我有权告诉你该怎么做。”我从未见过这种风格奏效。-*[*凯维·麦克明*](#_bookmark32)

大多数人都和那些认为自己永远不会犯错的人一起工作过。在每次讨论中，他们都会凑上前，挺起肩膀，以一种强硬的方式扮演决策者的角色。他们会继续争论，直到他们的观点获胜或时间耗尽。他们通常是对的，但他们的正确方式会让人感到窒息。随着他们在公司任期的增加，他们可能会认为自己变得非常有说服力，但通常这是一种以同事的无奈为特征的说服方式。

我合作过的一些技术领导者找到了一种永远不会犯错，同时又不主导会议的方式。在正确的同时，为他人创造空间。对我来说，一直体现这种方法的人是[富兰克林](https://twitter.com/thisisfranklin)，我经常看到他通过致力于为每个人找到最佳结果、愿意放弃自己的初始立场以及默认总是存在额外的背景信息可以将看似冲突的观点统一起来，从而化解激烈的讨论。

要成为一名高级技术领导者，你必须对技术和架构建立深刻的见解。要作为这样一位领导者运作，你必须培养同样深刻的实用主义和对技术宗教的不可知论，以保持对自己的怀疑。这可能会让人觉得自相矛盾，但这是你每天都需要走的路。

### 倾听、澄清并观察现场气氛

很多时候，你会看到工程师们充满信心地进入讨论，认为自己的观点是正确的，并且希望让房间里的其他人同意他们的做法。这种心态使每次会议都变成一场零和博弈。即使在他们的做法被同意的“最佳情况下”，他们也没有机会

向房间里的其他人学习，而且房间里的其他人也不太可能带着活力离开。

最有效的工程师在每次会议中都会以达成共识、了解房间内的需求和观点，并确定需要采取哪些措施来统一方法为目标。他们将每次会议都视为项目更广泛背景以及他们与房间内人员关系中的一轮。如果房间准备好同意并推进解决方案，他们就会让团队采用该方法。如果房间没有准备好，他们不会强迫它发生。

要擅长这一点，你需要掌握三种方法：通过提问倾听、明确目的以及了解如何观察现场气氛。

**通过提问倾听**是一种积极倾听的形式，其目标是了解房间内其他人的观点。以良好的意图提出好的问题可以开启对话，为其他人提出自己的问题创造空间和安全感。提出好的问题是为了学习，而且它们是具体的。它们可以使对话更加清晰。它们使回答者免于捍卫自己立场的义务。在一次可能引起争议的会议中，在你分享你的观点之前，先问三个好问题，你就会看到你周围的房间发生变化。

好的会议始于明确的目的和议程，但许多会议并不符合这个定义，特别是临时讨论。如果你发现自己参与了一个目标不明确的对话，那么**明确目的**。花点时间问问你对团队希望完成的目标的理解是否正确。最好的方式是将其包装成一个澄清性的问题，例如，“确认一下，我们的目标是决定是否将项目发布推迟两周？”

请注意，如果过于频繁地定义目的，可能会产生干扰。在这种情况下，它不会帮助澄清对话，而是会造成对话的混乱。在大多数情况下，如果其他人已经尝试过了，尽量避免使用它。多次重构失败的会议几乎总是以安排另一次会议告终。

最后，在每次会议中，你都必须**掌控全局**。通常，人们会对谈话感到沮丧，并试图强迫达成一致，这给讨论带来了很大的压力，不太可能得出好的结论。如果房间里的人意见分歧太大，那么就确定一个能够花更多时间一起深入研究的小组，或者确定一个合适的当事人，在房间外进行升级。如果抽屉里的东西太多了，就不要试图把它硬塞进去。

### 如何练习

如果这些行为对你来说不是天生的，那也没关系；练习的机会无处不在。对文档的每一条评论都是一个机会。每一次会议都是一个机会。每一个拉取请求都是一个机会。

每周开始时，选择你希望在即将参加的会议中明确使用的一项技能。如果你有一个特别困难的会议要召开，花一些时间在脑海中或与同事一起练习，你如何利用这些方法来促进前进，尽管存在挑战。

### 混蛋

上述方法在大多数情况下都有效，但并非总是如此，一个值得注意的例外是当你与一个混蛋打交道时。在这种情况下，混蛋是指那些拒绝给予团队同意、不愿意妥协或不听取意见的人。这样的人还没有认识到，他们的职业生涯更多地取决于是否容易参与，而不是技术上是否正确。

处理混蛋最有效的两种方法是：

1. 在会议中包括一个他们无法成为混蛋的人（比如他们的经理或首席技术官）
2. 在会议前大力投入与他们保持一致，这样他们会感到被倾听，并且不太可能扰乱讨论

这两种方式都可能让你觉得是在浪费时间，但它们往往是最有效的，特别是当你与一个混蛋的互动不频繁时。如果这个人是你负责的领域内的，或者你经常与之共事，那么

你有一套稍微不同的义务。在这种情况下，尽可能友善地向他们提供反馈，同时保持诚实。再给他们一次机会。记录这两次反馈，如果你没有看到任何改进，那么通过面对面或视频讨论，将你的担忧（包括具体的记录）告知他们的经理。

认识到你的头衔所赋予的权力可以保护你免受许多这类人的侵扰也很有用，所以无论你遇到的人有多混蛋，他们对你的态度可能都比对其他人好。如果这种行为对你来说只是有点过分，那么对其他人来说可能就更恶劣了。

### 它如何提供帮助

这种方法之所以强大，是因为更复杂的项目往往因人际冲突而脱轨，而不是因为技术复杂性，而这是一种可重复的方式，可以将紧张关系转化为伙伴关系。它让人感觉很慢，因为它可能需要更长的时间才能开始，但最终它很快，因为你更有可能在没有中断的情况下完成工作。

此外，作为一名高级领导者，长久的职业生涯与保持人际关系和取得杰出成就同样重要。你会看到很多人一时风光无限，但后来却缺乏支持，无法取得进展。如果你想避免这种命运，就要学会永远不会犯错，并且永不停止实践。

## 为他人创造空间

*在这一点上，我花更少的时间倡导特定的技术或项目，而花更多的时间授权他人倡导他们认为重要的技术和项目。我还努力成为人们可以寻求反馈的知识和支持来源，尤其是在跨领域的产品决策以及向组织其他部门展示想法方面。-* [*米歇尔·布*](#_bookmark30)

衡量你作为一名Staff-plus工程师长期成功的最佳标准之一是，你周围的组织越来越多地受益于你的贡献，但不依赖于你的贡献。因为很多人都是通过成为组织的“百事通”而达到他们的第一个Staff-plus职位的，所以从必不可少到可有可无的转变可能很困难。

这种转变需要学会刻意为周围的团队创造空间，并且归结为积极地让他们参与讨论、决策，并最终用赞助来代替重复那些让你晋升到Staff的成功经验。

### 讨论

当你专注于最大化你的个人影响力时，一个好的讨论是指能够快速结束，并得出合理的答案，参与者之间达成一致，且参与者之间有积极的感受。当你开始考虑创造空间时，一个好的讨论的定义会大大扩展！

这个更广泛的定义取决于让更多的人参与进来，并在没有你个人太多贡献的情况下达成一系列好的决策。在这个新的世界里，一个好的讨论是指最终你发现你不需要参加的讨论。当你做出关键贡献时，为之感到高兴，然后思考一下接下来需要发生什么，才能让其他人做出同样的贡献。

伴随着思维模式的转变，我发现了一些技巧

有助于在讨论中创造更多空间：

将你的贡献转移到提问上。提出正确的问题[有助于避免失误](https://lethain.com/learn-to-never-be-wrong/)，同时也让更多的人更容易做出贡献

如果你看到会议中有人没有参与，就把他们拉入讨论。最好一次只拉*一个人*进入讨论。如果一下子向所有人开放，或者甚至试图一次拉两三个人，就会让人感到困惑

成为做笔记的人。这有助于消除人们认为做笔记是“低地位”的偏见，同时也让原本要做笔记的人可以腾出手来做出更多贡献。而且，这也能让你有事情可做，而不是只顾着发言！

如果你意识到讨论中缺少了应该在场的人，那就成为将他们拉入下次会议的人。与会议协调人沟通，让他们知道 включить 他们的价值所在

当你越来越忠实地遵循这些原则时，你在会议中的发言会减少，而你对组织的影响力会增长。

### 决策

在你的职业生涯中，大部分时候，成功在于做出正确的决定，但你需要一段时间才能意识到，在某个阶段，做决定本身并不是工作。[里图 文森特 很好地描述了这种转变，](#_bookmark35)

*正是在那个项目中，我的经理帮助我理解了作为技术主管的最初冲动是行不通的。最初，我在想，“我会把它分成二十块，分配出十八块，把最难的两块留给自己”，但我的经理推动我把困难的部分委派给团队，以拓展和发展他们。*

另一方面，很难将你的[判断](#_bookmark49) 转移给其他人，尤其是在复杂的决策方面。幸运的是，可以采取渐进式的方法，将越来越复杂和重要的决策转移到你的更广泛的团队。

**写下来。**有一个广为人知的关于天才的模型，概括在

费曼算法中：“1）写下一个问题。2）非常努力地思考。3）写下解决方案。”这种对天才的神秘看法既难以接近又令人沮丧。它也是不现实的，但如果我们不写下我们的思考过程供他人参考，人们很难知道它是不现实的。通过写下找到答案的过程，以及答案的理由，我们周围的人可以开始从我们的决定中学习，而不是仅仅被它们指导

**尽早传播**，并在你对某个决定形成定论之前这样做。大多数人很难从一个已经形成的观点中退回来，通过尽早收集反馈，更容易整合反馈，并让人们参与到决策过程中，这样他们除了最终结果之外，还可以看到你的思维轨迹

**将风格与实质分开**，并停止对其他人的决定提供风格反馈。如果一条反馈意见不会对项目的成功产生有意义的改变，那么可以考虑不提供。如果它有用但不是关键，可以私下提出建议，而不是把会议拉到你的轨道上

**不要试图**[**展示**](https://lethain.com/showing-value/)。一些资深人士觉得他们需要对所有事情都发表意见，以证明他们的资历。另一些人则要求每个决定都必须完全反映他们曾经做过的类似决定。这两种情况都将不安全感置于影响之上，并阻止其他人成长为领导者

**改变你的想法。**对你的同事表示尊重的最大标志之一是倾听他们的意见，然后在事后改变你的想法。如果高级领导不改变他们的想法，那么很快每个人都会将虚张声势与成功联系起来

让你周围的人参与到你所做的决定中，并分享你的决策方法，这是培养你周围团队的一个有价值的组成部分，但是让这些决定*成为他们的*呢？

### 赞助

通过让人们参与到你的讨论和决策中，你让他们参与到你的工作中。这是一个成长、参与和向周围人学习的好方法，但在某个时候，你必须采取下一步行动。

不要让他们参与你的工作，而是让工作成为他们的。

最后一步是[赞助](https://larahogan.me/blog/what-sponsorship-looks-like/)其他人去做那些让你获得Staff+级别的工作。当重要的工作交给你时，你的第一个问题应该是，“谁能胜任这项工作并从中获得成长？”看看你是否能让他们领导这项工作，然后与他们合作，为他们的成功搭建项目框架。你的方法是什么？他们可能需要考虑哪些初步的担忧？他们应该与哪些利益相关者尽早讨论这个问题？

当你发现新的重要工作时，比如发现你的工具或流程中的一个缺口，想想还有谁可以完成这项工作，然后和他们坐下来，让他们把你计划写的提案整理出来。然后像支持你自己的提案一样，支持他们的提案。

重要的是，当工作变成他们的时，你必须让它成为他们的。提供建议、给出意见、提供背景信息，但最终，赞助包括允许他们采取你不会采取的方法。结果可能不太好，他们会从中吸取教训——就像你从职业生涯中的错误中吸取教训一样。结果可能非常好，然后你反而会学到一些东西。

虽然赞助应该成为你解决问题的默认方法，但不应该是你唯一的工具。大多数Staff+工程师认为，直接参与一些项目对于保持他们对软件、工具和组织如何在实践中运作的了解非常重要。如果你需要一个经验法则，可以写一份赞助日志，确保你每个月至少赞助别人几次——如果你发现自己赞助的频率低于这个数字，那就深入挖掘是什么在阻止你。

相反，如果你回顾过去，想不出过去几个月你直接参与过什么工作，那也值得纠正。

### 如果你不这样做会怎么样？

如果你通过成为公司领导的关键人物的“首选人物”巩固了Staff+级别的地位，那么你就已经了解到，为组织领导解决一个紧急问题是通往成功的可靠途径之一

认可。如果你已经成为技术远见者，你的想法渗透到公司的架构路线图中，那么你就已经体会到掌控公司技术未来大门是多么的强大。

很难放弃这些。

[然而，这种模式的最佳情况是一家暂时蓬勃发展直到 那个人离开。更常见的糟糕情况是一家](https://www.amazon.com/dp/B0058DRUV6/)

要成为一家真正成功的公司的长期领导者，唯一的方法就是不断为他人创造空间，让他们获得认可、奖励以及让你走到今天的工作。这可能会出乎意料地令人不舒服，但别担心：无论如何，总会有新的工作等着你。

## 建立一个同行网络

当我与越来越多的高级工程师交流职业建议时，最一致的建议是建立一个从事类似工作的个人同行网络。不是每个人都强调这种方法，但超过一半的人提到了它，对于那些提到它的人来说，这往往是他们的第一个也是最强烈的建议。

[里图](#_bookmark35)

*对我来说，最有影响力的是拥有很多我认为是导师的人，通常是朋友、前任经理和与我共事过的人。我经常与过去与我共事过、了解我并且我信任的人一起吃月度午餐、咖啡聊天和晚餐。正是这些关于职业挑战和成长的对话让我达到了我职业生涯的现在的位置。*

[凯维·麦克明](#_bookmark32)Keavy McMinn提到她的网络是获得诚实反馈的重要途径：

*我首先想到的是找到你的同行或支持网络。就像管理一样，你爬得越高就越孤独，因此找到仍然会挑战你，并且你可以一起集思广益的同行非常重要。即使他们在你类似的工作领域，甚至在不同的公司，也没关系。*

[纳尔逊](#_bookmark37)Nelson Elhage也分享道：

*对我来说，培养一个由其他高级工程师组成的良好个人网络也非常有价值。我会和他们非正式地聊聊我们正在做和思考的任何事情。当你有人脉关系时，你可以非常坦诚地了解人们遇到的问题和他们正在考虑的解决方案。*

虽然知道你应该建立一个网络很有帮助，但有些人很难弄清楚如何去做。在建立网络的各种策略中，有两种[策略](https://lethain.com/meeting-people/)

最常见的策略是：容易被找到和内部建立联系。

### 保持可见性

在Staff-plus工程师中，对社区的潜在需求非常大，因此建立网络的\*\*最简单\*\*方法是让自己容易被Staff-plus工程师找到。一种有效的方法是参与围绕Staff-plus工程本身的讨论，比如Joy Ebert'z的《高级Staff软件工程师实际上做什么》或Keavy McMinn的《在技术领导力道路上茁壮成长》。虽然已经有很多人写下了他们对Staff-plus角色的看法，但每个人都带来了新的、有价值的视角。在这个话题上，你也有发言的空间。[高级职员软件工程师是什么 确实如此 或凯维·麦克明（Keavy McMinn）的](#_bookmark40)[在技术领导道路上茁壮成长](https://keavy.com/work/thriving-on-the-technical-leadership-path/)[.](#_bookmark40)[他们写下了自己对 Staff-plus 角色的看法，每个人都带来了新的、有价值的视角。这里有空间](#_bookmark46)

如果写作不是你的强项，那么你也可以发出自己的声音，在技术会议上发言是另一种在更广泛的社区中提高知名度的有效方式。Keavy McMinn描述了她参加会议演讲的动机，[凯维·麦克明](#_bookmark32)

*主要是，我喜欢在会议上遇到的人。后来，演讲者网络为我带来了工作机会。*

如果这些都让你觉得风险很高，那么即使是创建一个Twitter账户或加入几个相关的Slack频道（例如，Rands领导力Slack中的#staff-principal-engineering）也是一个好的开始。[*#staff-principal-engineering*在兰德斯（Rands） 领导力 Slack) 可能是一个好的开始。](https://randsinrepose.com/welcome-to-rands-leadership-slack/)

### 内部网络，也是如此

与专注于公开演讲和写作不同，[凯蒂·赛勒-米勒](#_bookmark34)的网络建议是在你当前的公司内部建立你的内部网络，

*建立人脉，建立人脉，建立人脉，建立人脉……你必须真正了解你在和谁说话，并确保你与多个团队和多个小组建立联系，以利用这些网络。*

虽然很容易认为人脉拓展只发生在外部，但通常在你已经所在的公司里更容易进行，在那里，它

在你的工作过程中，半自然地、半刻意地发生。这种方法还有一个额外的好处，那就是直接改善你的日常工作。从长远来看，这些人最终会离开并遍布整个行业，从而引导你更广泛的人脉。当你在一家规模相当大或声望很高的公司时，这种方法非常有效，而当你的公司规模变小或声望降低时，效果会稍差。

### 环境网络

在没有提到发展个人人脉的人中，大多数人提到建立一个基于与行业书籍保持同步并在社交网络（尤其是 Twitter）上关注行业领导者的环境学习网络。

[戴安娜](#_bookmark38)的评论是

*我广泛使用 Twitter，但我主要是一个消费者，关注科技领域的很多人。我通常关注那些我在会议上看到过演讲的人，或者我合作过的人，我发现他们的内容与我相关。这里有几个，没有特定的顺序：*[*卡米尔·富尼耶*](https://twitter.com/skamille)*，*[*拉拉·霍根*](https://twitter.com/lara_hogan)*，*[*乔希·威尔斯*](https://twitter.com/josh_wills)*，*[*维姬·博伊基斯*](https://twitter.com/vboykis)*，*[*大卫·加斯卡*](https://twitter.com/gasca)*，*[*朱莉娅·格雷斯*](https://twitter.com/jewelia)*，*[*霍尔顿·卡劳*](https://twitter.com/holdenkarau)*，*[*约翰·奥尔斯波*](https://twitter.com/allspaw)*，*[*查丽蒂·梅杰斯*](https://twitter.com/mipsytipsy)*，*[*西奥·施洛斯纳格尔*](https://twitter.com/postwait)*，*[*杰西卡·乔伊·克尔*](https://twitter.com/jessitron)*，*[*莎拉·卡坦扎罗*](https://twitter.com/sarahcat21)*，*[*橙皮书*](https://twitter.com/orangebook_)*。*

[达米安](#_bookmark41)

*我尝试在 Twitter 上关注那些我认为正在做有趣事情的人，以及我可以向他们学习的人。有很多人在做有趣的事情，也有很多东西要学习！我想到的一些名字：[包括*[*Aphyr*](https://twitter.com/aphyr)*，*[*坦尼娅·赖利*](https://twitter.com/whereistanya)*和*[*大卫·福勒*](https://twitter.com/davidfowl)*]。*

如果以这种方式建立人脉的想法让你感到不舒服，那么建立一个环境网络可能是在正确方向上迈出的良好第一步。也就是说，你会发现个人人脉更有影响力，并且找到一种真实的方式来建立一个人脉，是在你的职业生涯的漫长过程中，在高级职位上取得和保持影响力的重要一步。

### 质量胜于数量

一位同事曾经告诉过我一个故事，讲的是有人决心在业务拓展领域闯出名堂，他们会从旧金山飞到纽约，带着一份他们想见的人的名单。他们会寻找那些人在当晚可能出现的推文和Foursquare签到，然后去那里，买一杯饮料，假装偶然遇到他们。在一个顺利的晚上，他们会尝试以这种方式结识六个或更多的新人脉。

不用说，你不应该这样做——这完全是对界限的侵犯。此外，这样做甚至没有意义：在建立同行网络时，数量并不重要。相反，专注于与你真正信任、尊重和受启发的人慢慢建立联系。这才能创建一个真正强大的网络，帮助你解决遇到的最困难的问题和最棘手的情况。

最后，如果你已经读到这一段，并且真的想建立一个人脉网络，但又不确定如何开始，我将分享对我这个内向的人来说有效的方法，我曾努力寻找一种真实的方法。找到一个你尊重的人，给他们发一封简短的1-2段的电子邮件或私信，提出一个具体的建议请求。如果他们回复了，感谢他们，并在六到十二个月后发送另一个问题。如果他们随后向你寻求帮助或提问，尽你所能提供帮助。如果他们没有回复，不要担心；直接不作评论地继续前进。这种方法出奇地有效，最糟糕的情况也完全可以接受：他们只是永远不会回复。

## 向高管汇报

你是否曾向公司高管汇报一项关键的工程计划，兴致勃勃地走进会议室，却沮丧地离开？也许你只讲到第二张幻灯片，就被无关的问题打断了讨论。也许你完成了整个演示，却只听到人们说“干得好”，然后就离开了，没有任何有用的讨论。事后，你不太确定发生了什么，但你知道情况不太妙。

在你的职业生涯早期，你可能不会经常与公司高管互动。当然，如果公司规模足够小，你可能会，但这并不是常态。然而，随着你职业生涯的深入，你的影响力将越来越受到你有效影响高管能力的影响。虽然与权威保持一致是影响高管的先决条件，但你还需要培养一些新的沟通技巧。[停留 与权威保持一致 是影响高管的先决条件，还有](#_bookmark13)

### 为什么这很难

每个人在职业生涯的某个阶段都与糟糕的高管共事过，但大多数高管并不糟糕。几乎所有的高管都非常擅长*某些事情*；只是通常那*某些事情*不是你正在与他们沟通的主题。当你把对你所在领域的不熟悉与有限的讨论时间结合起来时，沟通就成了一个挑战。

不过，这些都是普通的沟通挑战，而与高管沟通可能会出乎意料地困难，原因在于一个不太明显的原因：高管已经习惯于以一种特定的方式来消费经过预处理的现实。

任何一位高管几乎总是非常擅长一种消费信息的方式。他们觉得以那种特定的方式消费数据最舒服，而且围绕他们的沟通系统也经过优化，以那种方式与他们沟通。我把这看作是预处理现实，而以错误的方式为给定的高管预处理信息将会

经常造成误解，而参与者都无法完全解释。

例如，一些高管在模式匹配方面有着非凡的天赋。他们在任何演示中的第一反应都是提出一系列详细的、看似随机的问题，直到他们能够根据自己之前的经验进行模式匹配。如果你试图向这位高管做一个结构化的、学术性的演示，他们会感到厌烦，而且你将浪费大部分时间来展示他们不会吸收的信息。另一些高管会无视你说的任何没有与特定数据或数据集相关联的内容。你可能会充满信心地进行演示，因为你知道你的数据在附录中，但他们会越来越不相信你的提议，认为它缺乏支持。

在大多数其他情况下，沟通不畅造成的更多是延迟，而不是错误。不过，在与高管沟通时，你通常不会有第二次机会在做出相关决定之前讨论给定的主题。在讨论之前进行投入，以避免事后的遗憾。

### 如何有效沟通

与高管有效沟通的基础是首先清楚地了解你与他们沟通的原因。你可能习惯于与人们沟通以改变他们的想法或告知他们关于你的项目，但这里可能并非如此。当你与高管沟通时，几乎总是以下三种情况之一：计划、汇报状态或解决分歧。

尽管这些是不同的活动，但你的目标始终是从高管那里尽可能多地获取观点。如果你开会是为了改变他们的想法，你可能会显得不够灵活。开会是为了了解你如何与他们的优先事项保持一致。你会显得很有策略性，并且可能会带着足够的信息离开，以便调整你现有的计划，使其在高管新提出的重点或约束范围内工作。

提取他们观点的最佳方式是编写结构化的文档。写作迫使你全面思考你的信念和数据。结构确保你让读者专注于重要内容。Barbara Minto，

[金字塔原理](https://www.amazon.com/Pyramid-Principle-Logic-Writing-Thinking/dp/0273710516/)她的金字塔原理是关于有效商务沟通的最有影响力的著作，她也是结构的忠实拥护者：

*控制你呈现想法的顺序是清晰写作所必需的最重要的行为。最清晰的顺序总是在给出被总结的各个想法之前给出总结性的想法。我再怎么强调这一点也不为过。*

有很多结构可行，但我特别建议每个文档的开头段落都遵循SCQA格式：

**情境**：相关的背景是什么？*示例*：在产品功能发布方面，我们已经落后于竞争对手两年了。去年，我们将工程团队扩大了一倍，但发布的功能却比前一年少。

**复杂性：**当前情况的问题是什么？*示例：*我们计划今年再次将我们的工程团队扩大一倍，但根据去年的经验，我们认为这将进一步降低速度，同时显着增加我们的组织预算。

**问题：**要解决的核心问题是什么？*示例：*我们是否应该继续推进今年将工程团队扩大一倍的计划？

**答案：**你对提出的问题的最佳答案是什么？示例：我们应该停止招聘六个月，专注于整合我们现有的团队。根据当时的进展，我们应该更新今年剩余时间的招聘计划。

在许多讨论中，一个结构良好的开篇段落足以引发重要的对话。虽然在这些情况下，你可能不会讨论你的文档的其余部分，但编写文档的过程仍然是完善你想法的重要一步。

相对较少的人对他们的整个文档采用正式的结构，但至少有一种流行的格式，有些人觉得有价值：来自前面提到的书中的明托金字塔原理。首先，将你的提案集思广益成一系列支持你答案的论点。一旦你把它们都写下来，就把它们分成相关的论点。将这些组塑造成三个顶级论点，最多有三个子论点

支持每个顶级论点。递归地应用这种方法，确保每个论点总结其最多三个子论点。按重要性降序排列每个组中的论点。到那时，你就完成了。

虽然我个人觉得 SCQA 立即有用，但我承认，当我第一次尝试遵循金字塔原理时，它给了我与盯着粗野主义建筑相同的情感反应。通过实践，它在我身上成长起来，但我仍然建议大多数人首先采用 SCQA 作为核心实践，并且只有在收到你的演示文稿难以理解的反馈时才完全采用金字塔原理。

在你写完结构化文档后，从你的同事和利益相关者那里收集反馈。在演示之前与利益相关者保持一致，有时称为[协商](https://blog.toyota.co.uk/nemawashi-toyota-production-system)，在减少意外方面非常有效。你的一些同事应该有向高管演示的经验，并且会对改进有有用的反馈。

对于演示本身，设置一个明确的议程，但不要专注于死记硬背。与执行领导层的精彩会议的定义是参与讨论，而不是解决议程上的每个主题。有些人会认为这是一个有争议的立场，更喜欢通过其行动项目来衡量每次会议，但这忽略了这些会议中通常更有价值的关系建立和发展方面。

### 要避免的错误

即使你为你的执行演示做了出色的准备，这些事情有时还是会出错。没有什么能让你避免所有糟糕的情况，但你可以避免大多数经常导致这些会议失败的反模式。

**永远不要对抗反馈**。高管提出关键反馈意见，但又没有合适的框架在当下沟通是很常见的。你希望他们无论如何都要给出反馈，而不是憋在心里，可能之后就忘了。如果你表现出抵触反馈，他们就会开始吞回评论，而你从会议中获得的相对较少。

专注于收集反馈；在有更多时间之后再考虑是否同意。如果需要做出你不同意的决定，那么你应该注入一两个可能改变他们想法的相关数据，但之后就放手。通过反思反馈并在之后改变他们的想法，比在会议中继续反驳更有效。

**不要逃避责任或问题。**很多人试图向领导隐瞒问题，但这总是会适得其反。成功人士将告知高管视为一种解脱：一旦问题摆在桌面上，你就可以着手解决它，而不是隐瞒它。如果高管在会议中嗅出了问题，尤其如此。接受反馈，不要逃避。通过同意他们的观点并在之后提供更多数据，你将建立更多的信誉。通过与他们争论，你会损害你的信誉。

**不要提出没有答案的问题。**给新领导人的一条常见建议是“永远不要带着问题去找你的经理，而不带着解决方案。”这通常不是一个好建议，但如果你向高管提出一个没有提议答案的问题，那么在他们脑海中，他们会想是否需要聘请一位更高级的领导来补充或取代你。除非你有一个提案供大家支持，否则你无法在房间里达成一致。

**避免学术风格的演示。**你在学校里学到的关于主题的演示方式或多或少是向高管演示的完全错误的方法。*明托金字塔原理*如果你遵循它的准则，它会引导你朝着正确的方向前进。

**不要执着于你偏好的结果。**人们常常会过于关注他们想要的结果，以至于他们花费精力来抵抗清晰、不可避免的迹象，表明事情不会那样发生。对于做出“错误”的决定感到沮丧很容易，但记住你遗漏了很多背景信息是有帮助的。没有永久的决定：几乎每个决定在未来两年内都会被多次重新考虑。

向高管演示可能会让人感到害怕，而且这可能弊大于利。

如果你想把它归结为一个简洁的提示：提前将草稿发送给参加会议的高管，并询问他们需要更改什么。如果你倾听并应用该反馈，你会在进行过程中弄清楚其他部分。

# 获得你所在职位的头衔

****

*我听过的最好的建议是，通常达到Staff级别是运气、时机和努力的结合。-*[*伯特·范*](#_bookmark33)

大多数科技公司都有一个“职业级别”，这个级别通常是大多数人能够达到的最高级别。在大多数公司，高级工程师是职业级别。虽然你可能会因为没有足够快地从入门级工程师晋升到中级工程师而被解雇，但大多数公司并不期望你从高级工程师晋升到资深工程师。在中级工程师岗位上工作六年？啊，那是个问题。在高级工程师岗位上工作二十年？当然，没问题。

除了对进步的期望消失之外，一旦你达到职业级别，公司的晋升系统通常会阻碍你进一步的进步。有时，已经拥有资深工程师头衔的人会保护自己的声望不被稀释。在其他情况下，由于团队健康或预算方面的考虑，组织可能不愿在单个团队中拥有多名资深工程师。但是，我认为最大的摩擦来源是工作性质的变化。资深工程师不是更好的高级工程师，而是已经转变为履行[员工原型](#_bookmark6)之一的人。

即使在你已经掌握了成为资深工程师的先决技能之后，仍然会有最后一个障碍：让你的公司授予你资深工程师的头衔。对于某些人来说，这个过程相对来说不是什么大事，可能比预期的时间长一两个周期，但最终会成功，而对于另一些人来说，这可能根本不会在他们目前的公司发生。我调查的资深工程师中，约有三分之二是在他们已经工作的公司晋升获得的头衔，其余三分之一则通过更换公司来获得该头衔。

如果追求这种角色是你的目标，那么将晋升到你的职业级别视为一个机会，重新调整你应对职业生涯的方式。从那时起，就没有标准的道路可循。晋升和绩效系统将不再围绕及时晋升而设计，有时会带有守门人的感觉。

为了更进一步，*你*必须更加有意识地控制自己的进步，本章分享了一些工具，这些工具对那些已经取得进步的人有效。

寻找你的足迹

### 如果你一直依赖你的经理来指导你的职业生涯，那么向自我指导的职业生涯的过渡可能会感觉相当突然。有很多关于管理你的软件职业生涯的书籍，但大多数都侧重于从你的第一份工作到你成为高级工程师。很少有关注于管理你超越高级工程师头衔的职业生涯，这正是本章的重点：

你的晋升资料包是你揭开资深工程师晋升神秘面纱、优先考虑正确的个人发展以确保你实现目标，并激活你的内部赞助者和网络以支持你的进步的基础工具[管理你的软件职业生涯](https://www.learninpublic.org/)

人们普遍认为，要晋升到资深及以上职位，需要成功完成一个资深项目。本节讨论了大多数资深工程师没有资深项目的事实，但也描述了如果你所在的公司确实需要资深项目，该如何处理。[**晋升材料**](#_bookmark20)

٣١٢٩٣٥٨٤٣٩[**员工项目**](#_bookmark22)

工程师们经常抱怨他们没有“参与”决策的制定，而且他们通常是对的：确实有一个房间，而他们不在里面。但较少被承认的是，你可能没有理由进入那个房间。本节将介绍如何进入房间，以及如何留在那里[**如何进入房间， 以及如何留在那里**](#_bookmark23)

最后，如果你的公司的领导不知道你是谁，你就不会得到晋升。你如何在内部变得可见，而又不占用所有的资源？[**在内部变得可见**](#_bookmark24)

坚持运用这些技巧，你就会朝着Staff的头衔前进，尽管即使是最好的计划，如果你在错误的公司执行，也会失败。[如果你在 错误的公司.](#_bookmark25)

### 机会分布不均

在追求Staff职位时，你会遇到的一个令人不快的现实是，在任何给定的公司，机会的分布都是不均匀的。如果你的公司领导认为基础设施工程本质上比产品工程“更复杂”或“更有杠杆作用”，那么机会将集中在基础设施团队中。如果你在一个强调交付功能的组织中工作，那么修复你导致的中断比防止未来的中断更容易获得奖励。如果你在你公司的总部工作，而不是在分布式办公室工作，你的工作将更引人注目。[分布式办公室](https://lethain.com/how-to-start-distributed-engineering-office/)

许多公司认为，他们有既得利益来假装机会是均匀分布的，即使事实并非如此。这使得人们很难相信这些动态的存在，但随着你收集更多的数据，趋势会变得清晰。

一旦你认识到这些挑战，你就必须评估它们的可修复程度，以及你希望优先考虑你的精力在哪里。将你的方法与这些不成文的潮流对齐，而不是重新规划创造它们的河流，要简单得多。如果你选择解决不平等的原因，首先要找到一位支持这项事业的高级赞助人。你只能在内部赞助的情况下改变一个系统。

### 你应该尝试管理吗？

大多数达到Staff-plus职位的人不会花时间在工程管理上，但有些人会。很容易将此视为一个关键的、改变人生的决定，但这可能有点想多了。如果你想尝试管理，你应该这样做。大多数公司都明白，管理并不适合所有人，并且很乐意让你轮换回工程职位。

那些尝试管理的人会获得更广阔的视角，即使他们回到软件工程职位，也能帮助他们。这是丹·娜的经验，[丹·娜](#_bookmark39)

*我仍然喜欢交付代码和管理团队，我认为在高水平上同时做到这两者的能力对于长期的工程成功至关重要。Charity Majors有一篇关于这个话题的精彩博文，我推荐阅读：“工程师/经理钟摆”。Charity认为“经理职业道路与工程师职业道路”是一种错误的二分法，花时间在两种角色之间交替可以让你在两种角色上都做得更好。这与我自己的经验相符。我是一个更好的经理，因为我知道在一个计划不周的项目中成为一个IC有多么糟糕，我是一个更好的IC，因为我知道何时以及如何在一个项目进展不顺利时发出警报。*[*这个*](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)

[里图](#_bookmark35) 分享了一个类似的观点，

*我的确会*[*摆锤*](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/) *工作，因为我对职业阶梯两边的很多事情都感兴趣。我对培养人才感兴趣，我真的很喜欢招聘工作，我是那种真正喜欢面试的工程师，我喜欢了解团队是如何成长的。但我也真的很喜欢编写代码，在花了一些时间进行管理之后，我想回到代码中，稍微研究一下。*

有些人尝试管理，最终却讨厌它。[乔伊·埃伯茨](#_bookmark40) 不太喜欢工程管理，

*实际上，我在Box工作期间管理了大约一年半的时间，发现我讨厌它（你可以在我的*[*博客 关于那个话题)。也就是说，我发现实际上有很多重叠之处*](https://code.likeagirl.io/why-i-left-management-the-engineering-technical-track-vs-management-track-abef5b1d914d)

即使Joy讨厌她的管理经验，她也觉得这可能对她的长期职业生涯有所帮助，

*如果我没有经历过管理，我可能会更快地晋升到Staff职位。也就是说，我不后悔这样做，我学到了很多关于人们如何思考、组织如何运作以及更大的*

*项目是如何确定优先级的。所有这些都一直在帮助我完成IC轨道上的工作，并且可能帮助我进一步晋升为高级Staff。虽然我确实认为当我晋升到Staff时，速度明显放慢了，但对于下一个级别，我实际上不太确定——我认为如果没有它，我很有可能会在Staff职位上待更长的时间。所有这些都说明，即使我没有走最直接的路线，我仍然学到了很多对我的长期发展有帮助的东西。*

对于考虑这种转变的人，我给出的最后一个警告是，人员管理不仅仅是最大限度地提高你晋升到Staff工程师职位的轨迹。作为一名经理，你将对你所支持的人产生深远的影响，如果你带着错误的动机去做这件事，你会后悔的，但远不及你的团队会后悔的。如果你有动力帮助你的团队成长和成功，那就去做吧；如果你只是为了自己而做，那就不要做。

### 半渗透边界

作为最后的警告，Staff+头衔是领导职位。如果现有的领导团队不认为你是潜在的成员，那么获得领导职位将是一个独特的挑战。这意味着，不幸的是，那些有*看起来像*他们已经是现有领导团队成员的特权的人，更容易实现过渡。

如果你读完本章，并且越来越沮丧地发现你已经做了这里的所有事情，那么你可能正在经历结构性的劣势。在我交谈过的女性中，大约有一半人不得不更换公司才能获得Staff职称，而在与男性讨论时，晋升摩擦的出现频率较低。

不要忽视这些经历——它们是真实的，很多人都感到受阻——但也要抱有希望，无论你如何定义自己，以及你希望如何规划你成为资深工程师的道路，都有很多成功的榜样。

## 晋升材料

有些人认为他们的晋升材料是达到资深及以上职位的顶峰，但我见过很多人通过采取相反的方法获得成功：早在他们认为自己可能晋升为资深工程师之前就开始撰写他们的第一份资深晋升材料，就像他们可能使用一份[炫耀文档](https://jvns.ca/blog/brag-documents/)一样。通过这种方式，你的材料就变成了实现你目标的地图。

你的公司很可能会有自己的晋升材料格式，并且*最终*你需要将你的材料翻译成那种格式，然后才能将其提交给内部晋升委员会或流程，但没有必要着急。你将花费更多的时间依靠它作为指南，而不是作为正式审查的正式文件，因此要为前者进行优化。

对于走向你的资深及以上晋升，一个有用的通用模板格式是：

你的资深项目是什么？你做了什么？项目的成果是什么（包括一个[明确定义的目标](https://lethain.com/goals-and-baselines/)）？是什么让这个项目变得复杂？保持非常简短，然后链接到支持性设计文档

你通过哪些高杠杆方式改进了组织？

你的项目的可量化影响是什么？（你增加了1000万美元的收入吗？你是否将年度客户支持票据减少了20%？）

你指导过谁，通过哪些成就？

你为组织做了哪些粘合工作？这些粘合工作的影响是什么？[粘合工作](https://www.slideshare.net/TanyaReilly/being-glue)

哪些团队和领导熟悉并支持你的工作？他们看重你工作的哪些方面？一句话概括，如果可能，请包含数据（例如，调查数据）

你是否存在真实或被认为存在的技能或行为差距，可能会阻碍你的发展？对于每一个差距，你将如何解决这个问题？每项一句话概括

花一些时间自己写出这些答案是有用的，但晋升到领导岗位并不是一项单人活动——这是你只能

在团队的支持下完成的事情。我推荐的迭代你的晋升材料的方法是：

1. **回答你为什么要这样做**。很多人选择不追求Staff级别；你应该有一个理由说明这对你很重要。如果你没有，你很可能会发现自己身处一个你不喜欢的角色。

[米歇尔·布警告说](#_bookmark30)，*“我对工程师的第一个建议是，他们应该避免以导致他们从事不喜欢的工作的方式进行模式匹配。我深深地被我所做的工作所激励，与团队合作解决抽象建模和设计问题。经过多轮反馈后，一次又一次地尝试需要一定的毅力。说实话，这并不适合所有人。如果你更专注于达到Staff级别，而不是专注于让自己从事能激励你的工作，那么很容易最终陷入一个你不想要的角色。”*

1. **调整你的期望。**晋升，尤其是在这个级别，是经过几个季度、半年甚至几年的积累。避免期望立即见效
2. **让你的经理参与进来。**在下次与你的经理进行一对一会议时，带上*晋升材料*，并告诉他们获得Staff晋升是你的目标。与他们一起审查空白的材料，并询问他们缺少什么、应该强调什么，以及他们是否建议在工作流程中添加步骤。你的目标是确保他们知道你对这件事感兴趣，并征求他们对你的方法的指导。

[里图·文森特建议](#_bookmark35)，*“人们经常来问我，‘我下一步应该做什么才能达到Staff级别？’我告诉他们的一件事是，要非常坦诚地与你的经理沟通你对职业生涯的期望。我早期在一对一会议中犯的一个错误是告诉我的经理我认为他们想听什么，而不是我实际的感受。”*

1. **撰写*晋升材料*。**花一个小时坐下来，把你的所有想法放在第一个版本中
2. **编辑*晋升资料包*。**等两天，重读你的*晋升资料包*

并针对内容、清晰度和上下文进行编辑

1. 与同事一起编辑晋升材料包。与几位值得信赖的同事分享您的晋升材料包以获得反馈，最好是已经担任Staff+职位的同事。同事通常比您更擅长识别您的优势和贡献，而且他们比您的经理更了解您的工作
2. 与您的经理一起编辑晋升材料包。与您的经理分享您的晋升材料包并请求反馈。特别要求关注需要弥补的差距。询问您是否可以在接下来的1:1会议中花时间讨论可以通过哪些项目和机会来弥补差距并使材料包更强大
3. 定期与您的经理一起审查晋升材料包。在您的职业生涯和以绩效为导向的1:1会议中，继续与您的经理一起审查晋升材料包。您和您的经理都应该利用它来引导您随着时间的推移展示晋升标准。如果您的直接经理发生变化，这一点尤其重要。维护此类文档并在不同的经理之间进行审查将有助于减轻经理变更后经常发生的晋升进度损失

如果您有条不紊地遵循这些建议，那么您将在获得晋升提名之前很久就整理出您的第一个Staff晋升材料包。从那时起，您将使用该材料包来集中您的注意力和与经理的合作，以实现该目标。它不一定会让您快速到达那里，甚至可能无法在您当前的公司让您到达那里，但它会将您的精力集中在能够推动您实现目标的开发和工作上。

当最终到了撰写正式材料包的时候，这将是一个将您收集的内容编辑到官方模板中的问题，而不是一个翻阅多年努力的档案过程。希望在晋升过程中一切顺利，并获得Staff职称。[出错 在晋升过程中，然后是Staff头衔。](http://lethain.com/promo-pathologies/)

## 找到你的赞助人

*拥有赞助人也绝对重要。我和我的经理关系非常好，我和我的越级经理关系也很好。我认为这也在其中发挥了重要作用。- Ritu Vincent*[*瑞图·文森特*](#_bookmark35)

正如我与更多试图达到他们第一个Staff+职位的人交谈时，大多数人都会遇到类似的挑战。许多人错误地评估了自己的影响力，并且还没有完成在该级别上运作的工作：Staff工程师不仅仅是一个更快的资深工程师。然而，有一大批人已经完成了这项工作——他们在他们的组织中可见，并且已经整理出了一个强大的晋升材料包——但仍然在努力让这项工作得到认可。[在其组织内可见](#_bookmark24)[并且汇集了一个强大的 晋升材料——但仍在努力使这项工作得到认可。](#_bookmark20)

这些人经常对他们的影响力和他们所获得的认可之间的差距感到沮丧，并向他们的经理和同事寻求关于缩小差距的反馈。他们被告知[完成一项员工项目](#_bookmark22)或[为他人创造空间](#_bookmark16)。对于那些尚未完成工作的人来说，这是一个很好的建议，但对于那些已经完成这些工作的人来说，这些复选框是一种干扰：他们真正缺少的是一个愿意推动对他们现有工作认可的赞助人。

人们通常会通过学校等在生活中评估我们的其他系统的角度来看待晋升系统，但这错误地将绩效评估定义为一项单独的活动。无论你的公司是临时晋升还是使用校准流程，晋升都是一项团队活动，正如时任 Slack 的朱莉娅·格雷斯 (Julia Grace) 在我找工作时给我的建议：“不要独自玩团队游戏，你会输的。”[使用校准过程](https://lethain.com/perf-management-system/)[朱莉娅·格雷斯](https://twitter.com/jewelia)

### 寻找你的赞助者

在指导你晋升的团队中，最重要的成员是你自己。第二重要的人是你的组织赞助者。拉拉·霍根 (Lara Hogan) 曾详细撰写过关于赞助的文章，但大致来说，这个人会在有影响力的论坛上为你的工作发声，并在倡导有限的资源（如加薪预算）时为你争取。[关于赞助的详细描述](https://larahogan.me/blog/what-sponsorship-looks-like/)

虽然你可能会有各种各样的赞助者，但在晋升的背景下

[晋升材料](#_bookmark20)——尤其是晋升到 Staff-plus 职位——这几乎总是需要你的直接经理。他们会是将你起草的晋升材料变成公司格式的人。他们会是在校准会议期间为你的晋升辩护，因为其他人会深入研究你的资历。他们也会是必须与你坦诚地谈论你在成为一名强大的晋升候选人之前仍然存在的差距的人。

虽然你总是需要你的直接经理参与作为你的赞助者，但你可能需要额外的赞助。如果你的经理以前从未将任何人晋升到 Staff-plus 职位，他们很可能会感到惊讶或在此过程中犯错。投入精力与你的管理链条上更进一步的人建立关系。你不需要花太多时间与你的越级经理相处，但如果他们对你的工作影响不够熟悉，以至于在两个月后的会议上都记不起来，你就不太可能晋升到 Staff 职位。

### 激活你的赞助者

激活你的赞助者的第一步是明确分享你的目标。“我希望被认可为 Staff 工程师”是一个很好的开始。瑞图·文森特 (Ritu Vincent) 提到这是她为寻求 Staff-plus 职位的人提供的最佳建议，[瑞图·文森特](#_bookmark35)

*人们经常来问我，“我下一步应该做什么才能达到 Staff 职位？”我告诉他们的一件事是，要非常公开和诚实地与你的经理谈论你对职业的期望。我早期在与经理一对一交流时犯的一个错误是，我告诉我的经理我认为他们想听什么，而不是我实际的感受。*

一旦他们确定了他们的赞助者，许多人就认为他们的工作已经完成：这取决于赞助者来完成繁重的工作。这通常会失败！赞助者是拥有比部署该资本的带宽更多的组织资本的人，当你为他们调整好各个部分时，他们会最有效地帮助你。询问你的赞助者，你如何支持他们的赞助。拥有你的职业生涯不仅仅是要求东西。它*是*关于那件事，但更多的是促进那些事情的发生。

与你的赞助人一起审查你的晋升材料是促进这种对话的好方法。重点关注如何询问差距所在，而不要促使你的赞助人编造答案。大多数人忘记了他们可以用“我不知道”来回答问题，如果你强迫他们回答他们不确定的问题，他们反而会编造无用的答案。如果你一直得到诸如“从事更大、影响更大的技术项目”之类的答案，那么你就是在用错误的方式、错误的问题或错误的人提问。

一个起始提示是，“如果我这次晋升没有成功，可能的原因有哪些？” 另一个值得问的问题是，“为了让自己成为更有力的候选人，我能做的最有效的事情是什么？” 也就是说，最好的问题是非常具体的，并且为回答者做了很多工作。想想回答这些问题有多难，相比之下，像这样的问题：“这个季度我完成了 API 重构，我认为这可以证明 Staff 级别的工作，但是进度延误了很多，最终让我们的产品经理感到沮丧，因为他们的工作被搁置了。我怎样才能更有效地处理这个项目？” 后一个问题更容易给出有用的答案，即使回答者不太熟悉项目的细节。

最后，请记住，激活你的赞助人不是在晋升前做一次性的交易。随着时间的推移建立关系，并在他们需要你的支持时努力帮助他们。[与他们的倡议保持一致](#_bookmark13)。假设他们需要人们加入一个工作组，自愿参与并努力工作。这些人有很多事情要问他们，而且他们很清楚那些在晋升前才出现的人。我曾经有一个同事很少来办公室，但*总是*在晋升决定前一周来办公室。人们注意到了。

### 如果它不起作用怎么办？

如果你发现自己和你经理相处不好的情况，这和*喜欢*彼此不太一样，那么你不会晋升到领导岗位。你的经理对你的影响和你被认为的影响有太直接的影响，以至于这种情况不会发生。同样，你可能和你的经理关系很好，然后他离开了公司。你几乎不会*注定失败*，但你的晋升时钟可能会

在你与新经理建立关系时重新设置。（有时，情况会相反，你的新经理会努力向你证明自己，通过代表你进行倡导。）

如果你在与你的经理发生摩擦后立即尝试更换团队或公司，你就会欺骗自己。公司通常不允许调动，除非你的经理批准，所以你可能会烧毁一座你站在上面的无路可去的桥梁。更重要的是，你将失去发展与你没有立即产生共鸣的人一起工作的技能的机会：这不是一个有趣的技能，但领导力*总是*涉及影响和与那些有冲突的目标和风格的人建立关系。

如果你已经花了六个月的时间积极地试图让这种关系发挥作用，那么现在可能是探索调动团队，甚至[考虑切换 公司. 这是众多案例之一，在这些案例中，拥有...非常有帮助](#_bookmark25)

## Staff 项目

*没有明确的期望，也没有任何地方将其列为正式要求，但人们理解你将完成一个 Staff 项目才能获得晋升。我想不出任何没有包含一个非常强大的项目的 Staff 晋升，通常是一个多人项目，工程师是技术负责人。-* [*瑞图·文森特*](#_bookmark35)

围绕达到 Staff-plus 角色的一个流行的反复出现的想法是，首先，你需要成功完成一个“Staff 项目”。这是一个被认为是足够复杂和重要的项目，完成它的人已经证明了自己是一名 Staff 工程师。然而，无论这个想法多么流行，如果你正在追求 Staff-plus 角色，重要的是要刺穿这些项目的神话，并专注于那些走在你前面的人的经验。

关于高级工程师项目的简短回答是，大多数工程师并没有将其作为晋升到高级工程师职位的一部分来完成，尽管有很大一部分人确实完成了，特别是那些通过在自己成长的公司晋升而获得该职位的人。对于那些没有完成项目的人来说，通常要么是因为他们在较长一段时间内积累了成功的记录，而没有一个单一的顶点项目，要么是因为他们更换了公司以获得该头衔。

我们将深入探讨高级工程师项目的几个不同角度：

1. 没有完成高级工程师项目的人
2. 完成了高级工程师项目的人，包括那些最终没有按计划工作的人
3. 识别和处理你的高级工程师项目，深入了解混乱的细节。

### 不需要高级工程师项目

当我问人们是否做过高级工程师项目时，一些回答非常简洁：

[乔伊·埃伯茨](#_bookmark40)，*“我实际上并没有真正做过高级工程师项目。”*

[戴安娜](#_bookmark38)，*“不，我没有被分配‘高级工程师项目’，这也不是 Slack 晋升过程的一部分。”*

[有些人甚至对整个员工晋升项目概念持怀疑态度。纳尔逊 埃尔哈吉 说，](#_bookmark37)

*我本能地对这种员工项目的想法有点警惕，部分原因是，我见过的员工工程师的典型形象之一是那些不一定自己运行大型项目或做大事的人。但他们只是非常有效的专家和协调者，使整个工程组织运行得更好。*

也有像丹·纳或达米安·申克尔曼这样的人，他们通过工程管理绕道到达这个职位。达米安描述了绕过Staff项目的情况，[丹·娜](#_bookmark39)[达米安·申克尔曼](#_bookmark41)

*我没有。因为我在Auth0的成长方式，我有点“跳过了那部分”。作为一家初创公司的总监，我有机会在技术上领导许多大型、关键的举措，但没有具体/明确的“staff/principal项目”。*

从这些故事中可以清楚地看出，任何告诉你必须完成一个Staff项目才能达到Staff-plus头衔的人都是错误的：有很多途径可以在不做Staff项目的情况下达到Staff-plus头衔，其中在工程管理部门工作一段时间是很突出的。[在工程领域工作过一段时间 管理 其中最突出。](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)

### 需要Staff项目

然而，同样真实的是，许多公司要求或非正式地强制执行Staff项目以进行内部晋升，因此，许多人确实将Staff项目作为其角色转变的一部分。

[里图](#_bookmark35)Ritu Vincent描述了她在Dropbox的经历，

*我肯定有一个Staff项目。很久以前，Dropbox最初是一个消费者产品，人们下载并安装在他们的机器上。*

*当我们推出Dropbox for Business时，有人要求您的个人和工作Dropbox帐户同时工作，包括能够在它们之间切换而无需注销并重新登录。最初的实现是在巨大的时间压力下编写的，它运行了多个Dropbox进程。一个用于您的个人帐户，另一个用于商业帐户。我的Staff项目是使单个Dropbox进程可以在多个用户登录的情况下运行。困难的部分是该项目从内核一直延伸到用户界面。我必须了解Dropbox系统的每一层。最初，我们认为这将需要六个月的时间，但最终花费了十八个月。它占据了桌面客户端团队的大部分资源很长一段时间。*

[拉斯·卡萨·威廉姆斯](#_bookmark31)Ras Kasa Williams加入了一个正在进行的项目，后来他成为了该项目的负责人，该项目成为了他的Staff项目：

*我以高级工程师的身份加入了Mailchimp。我立即被添加到一个项目团队（包括一名工程总监和两名首席工程师），旨在构建Mailchimp的第一个内部自助分析平台。该项目的一个关键方面是高效并以高水平执行。更好的是，或者更糟的是，拥有另外两名首席工程师意味着对我期望可能不会那么高。但我能够立即加入并开始为项目的核心方面做出贡献，而他们几乎没有提供任何帮助；到最后，我是团队中的主要贡献者之一。我最终将被正式任命为技术负责人，以帮助继续指导该项目的工作，因为它被吸收到我当前的工程组数据服务中。*

很少有公司会书面记录他们的 Staff 项目需求。它们更常见的是一种“软性门槛”，在晋升会议上被提出，有时会让经理和准 Staff 工程师都感到惊讶。发现这些需求最可靠的方法是你的“必成”晋升没有获得批准，但这并不是什么乐趣。几乎同样可靠且不那么令人沮丧的方法是依靠以下策略：[维护和 获取关于你的晋升材料的反馈 远在您晋升之前](#_bookmark20)

### 为什么你应该做一个 Staff 项目

有时很难确定守门和评估之间的精确界限，而 Staff 项目的前提就存在于那个模糊的领域。承担一个范围巨大的项目，驾驭这种模糊性，并成功交付它，是区分那些达到 Staff-plus 影响的人的有效方法，但也很明显，许多人在没有完成这样的项目的情况下也获得了 Staff-plus 的角色。

我的建议是，虽然你可以在没有完成 Staff 项目的情况下获得 Staff-plus 的角色，但它们是发展自己作为工程师的特别宝贵的机会。通过这种项目，你将*个人*得到拓展和成长，这是其他类型的 Staff 级别工作无法做到的。

[基维](#_bookmark32) 描述了她的 Staff 项目如何帮助了她，

*我从未听说过它被赋予一个名字，但我理解这个想法。我确实领导和架构过那种类型的项目——解决棘手的工程问题，对公司产生重大影响——几次，但不幸的是，它们并没有导致我被晋升。但它们确实促进了我的职业发展。这些项目给了我经验、知识和信心来不同地定位自己。甚至可以公开参加会议演讲，或者知道“我做过 X，并且可以再次做 X。”*

虽然每个项目都不同，但有一些典型的特征可以说明为什么它们在拓展你作为工程师方面如此有效：

**复杂且模糊** - 在你职业生涯的早期，你会得到定义明确的问题，但随着你深入其中，你将越来越多地遇到定义不清或未定义的问题，而 Staff 项目通常会从一个范围界定不清但复杂且*重要*的问题开始。你的项目可能仅仅从断言你公司老化的单体架构正在阻碍产品开发开始。从这个宽泛、不明确（且可能错误）的陈述中，你必须确定一种可行的具体方法。

**众多且意见不一的利益相关者** - 最简单的项目始于围绕问题和解决方案的组织一致性，但是

你的 Staff 项目很可能两者都没有。它可能是一个管理层认为存在生存风险的领域，但许多工程师认为它已经足够好了。它可能是一个每个人都同意存在问题，但在方法上存在强烈派系分歧的领域，例如，在追求服务战略或重新投资于现有单体架构之间存在分歧。

**具名赌注，失败很重要**- 这将是一个重要的项目

高层领导在全体组织或全体公司会议上谈论它就足够了。这意味着人们会密切关注你的工作，任何失败都会非常明显。成功也会非常显眼！

如果你符合这些条件，这可能是一个员工项目。这些项目可能会让人非常紧张，但也正因为如此，它们在培养你方面非常有效。

### 获得参与员工项目的机会

虽然决定承担一个员工项目是第一步，但你仍然需要获得参与这些项目的机会，这取决于你的管理链是否足够信任你，愿意押注你的成功。

这归结为三个因素。

1. 首先是学习与你的领导团队保持一致，其中一些策略在进入会议室和与权威保持一致中有所描述。[进入房间](#_bookmark23)[和保持一致 权威.](#_bookmark13)
2. 其次，你需要被认为具有解决当前问题的技术能力，这需要[保持可见性](#_bookmark24)。
3. 第三，你无法完全控制，那就是你的公司迫切需要解决一个员工级别的问题，这可能需要一些耐心。

### 你应该追求员工项目吗？

总而言之，如果你希望在当前公司获得晋升，并且之前没有担任过员工或管理职位，那么你可能需要追求一个员工项目，以在该级别上确立自己的地位。在其他情况下，你可能不需要。

无论如何，值得记住的是，无论这些项目是否是必需的，它们也是你能找到的最具挑战性的工作，并且是能让你得到锻炼和发展，成为一名更优秀的工程师的工作。在短期内追求头衔，避免这些项目可能是最佳选择，但在长期追求自我成长方面，它们是不可替代的。

## 进入房间，并留在那里

我从工程师那里听到的最常见的挫折之一是，他们没有参与重要决策的制定。他们不理解公司的决策，并且缺少或忽略了重要的背景信息。资深工程师经常将进入“房间”作为他们级别的主要好处，而头衔确实增加了你参与影响你的决策的可能性。

然而，重要的是要记住，没有一个可以进入的“房间”。进入正确的房间不是一次性需要面对的挑战。进入房间将是一个持续的、迭代的职业挑战。这意味着值得擅长它！

在你的职业生涯早期，它可能是一个与你的技术主管和产品经理举行的冲刺预计划会议。之后，它可能是一个季度计划会议，一个[架构审查](https://lethain.com/scaling-consistency/)，[绩效校准](https://lethain.com/perf-management-system/)，工程领导团队或执行团队。总会有另一个房间要进入。要达到高级水平，你不仅要有效地*进入*，还要有效地*留在*这些权力房间里。

### 进入房间

要进入房间，你需要：

**为房间带来一些有用的东西……**这可以是关于关键项目的细节，来自关键团队的背景信息，与房间目的相关的主题专业知识，在以前的公司运行类似项目或团队的经验，与关键相关客户的关系，或其他完全不同的东西。

**……房间里还没有的东西。**仅仅拥有可以带给房间的有用的东西是不够的。它还需要是一种房间里还没有的观点。小组比大组运作得更好，因此运营论坛通常会为了效率而牺牲冗余和代表性。要被包括在这些房间里，你需要带来一些与当前成员不同的东西。

**房间里的赞助者。**这些房间的空位有限，并且必须作为一个整体运作良好。要进入房间，你需要有人赞助你的会员资格。你的赞助者正在将他们的社会资本分配给你的加入，他们的同伴会根据你在房间里的行为来评判他们。这些房间通常混合了资历级别，因此通常情况下，你的赞助者的经理会在房间里根据他们赞助*你*的决定来评估*他们*。

**你的赞助人需要知道你想在那里。**你的赞助人可能在许多不同的房间里，并且可能梦想着离开他们身后的大部分会议。他们不一定会认为你想参加任何特定的会议，事实上可能会认为你根本不想在那里。确保他们知道你是否想被包括在内。

你如何为房间带来有用的东西，将取决于你和你想进入的房间的具体情况：没有任何单一的模式可以遵循。与你情况相似的人是否已经在房间里，也取决于你的具体情况，在某些时候，唯一的选择是等待或寻找另一个房间进入。

另一方面，有时增加你对房间价值的最简单方法是降低包括你的成本。以下是一些效果良好的方法：

**与你的经理保持一致。**人们会根据领导者的团队与其宣布的方法的一致程度来评估领导者。如果他们已经宣布转向持续部署，但他们的团队却在呼吁发布列车，那么人们就会怀疑谁在领导谁。如果你与你的赞助人高度一致，你更有可能被推荐进入房间。如果你*特别*一致，他们更有可能把自己的座位让给你，不再参加。

**为团队优化。**Stripe 过去的一个运营原则是“为 Stripe 优化”，这种为他人广泛优化的心态可以建立信任和对你判断的信心。

**清晰简洁地表达。**学会简洁地表达：随着你发展出精简的语言，你将能够在更少的时间内贡献更多的想法。学会清晰地表达：如果人们不理解你的提议，那么它

无论它有多好，都无关紧要。请记住，被理解是你的义务，而不是其他所有人理解你的义务。

**减少摩擦。**很容易陷入将每次讨论都视为阻止即将发生的灾难的最后机会的陷阱。在这种心态下，每次讨论都接近紧急状态，情绪高涨。这些讨论通常会花费时间来排解挫折感，而不是取得进展。如果你以能够有效地处理困难对话而闻名，你更有可能参与其中。

**做好准备。**有些公司把他们的工程师当成孩子，认为即使是非常资深的工程师也不会阅读议程、进行预读或为讨论做准备。容忍的事情和受到奖励的事情之间存在相当大的差距，如果你在每次会议前花时间整理你的想法，你就会脱颖而出。同样重要的是跟进你承诺的事情。

**专注并投入。**一旦你进入房间，一定要出现并参与。保持专注和投入。无论你还想做什么，它都会等待。

**自愿承担低地位的任务。**如果有人需要做笔记，请举手。如果有人需要跟进行动项目，请随时提供帮助。优先考虑发挥作用，尤其是在它不是最令人兴奋的工作时。

要进入这个房间，你必须同时处理分子和分母：不断发展独特且有用的视角，同时在会议的约束条件下更有效地传递这种视角。

### 留在房间里

进入房间是你的第一个障碍，但第二个障碍是留在房间里。最重要的是继续做那些让你进入房间的事情：将重要的背景信息带入房间，展示一个完美的自己，简洁明了，灵活变通。

有一些模式会持续地把你踢出房间：

**误解房间的目的。**每个房间都有[它自身的用途](https://www.tablegroup.com/books/dbm/)，如果你试图利用房间来对抗现有的

团队的意图，你就会制造摩擦。对于一个特定房间的功能，外部的看法（“他们在领导团队会议上做出所有决定”）与房间自身对角色的看法（“我们不做决定，只是提出问题供讨论”）往往相差甚远。花时间去了解房间的运作方式，并尊重这种意图融入其中。

**固执己见。**随着房间级别越来越高，他们必须讨论非常敏感的话题（薪酬、裁员、晋升、收购等），而且他们每周一起工作的时间是固定的。如果你固执己见，你就会制造摩擦，减慢讨论速度，并阻碍团队取得进展的能力。

**拒绝同意。**有效的团队是由愿意[不同意但承诺执行](https://en.wikipedia.org/wiki/Disagree_and_commit)的个人组成的。你通常可以通过拒绝同意，直到想法朝着你的方向发展，来迫使团队接受你的观点，但团队的步伐会慢下来，你可能会被排除在外。

**耗尽房间里的氧气。**有些是欢迎所有想法的头脑风暴讨论，有些是你已经转入运营模式以解除项目执行的时刻，你必须了解房间里正在发生什么。通常，这来自于[一种展示的冲动 价值, 但请记住，你之所以能进入这个房间，是因为你所拥有的东西](https://lethain.com/showing-value/)

**让你的赞助人难堪。**记住，你之所以能进入一个房间，是因为房间里有人支持你加入。

**表现不稳定或不定期出现。**位置有限，主持会议的人会优先考虑那些经常出现的人。

话虽如此，我认为人们很容易陷入过度担心留在房间里的状态。有时候，你最好考虑一下这个房间是否是一个值得你投入时间的地方。

### 离开房间

重要的是要记住，虽然有无限个房间可以进入，但没有

一个房间是真正能完成工作的地方。如果你能有选择性地决定待在哪些房间里，你就能发挥最大的影响力。虽然我见过很多人因为不被允许进入他们执着于的某个房间而感到不满，但我从未见过有人后悔过过早离开一个房间。如果任何一个房间感觉没有用处，就离开这个房间。在离开的时候，推荐其他人来接替你留下的机会。

## 保持可见性

*当人们，特别是女性和非二元性别者，向我寻求建议时，他们可能期望我谈论如何成长为一名技术领导者，但当我说“你可能已经具备了技术能力，你需要做的是提升你在公司的声誉”时，他们会感到惊讶。无论好坏，没有良好的声誉，你无法晋升到Staff级别。-* [*凯蒂·赛勒-米勒*](#_bookmark34)

[伯特·范](#_bookmark33)对那些试图达到Staff-plus级别的人的最佳建议是：“通常达到Staff级别是运气、时机和努力的结合。”时机是一种特殊的运气，所以你可以进一步简化为运气和努力。

如果你很幸运，那么你就不必刻意追求Staff-plus的职位。你已经在从事公司最重要的工作，有一位关心你职业发展的、位置优越的经理，并且在公司总部工作。如果你一开始什么都没有，那么晋升将是一个相当大的挑战，但不要妄自菲薄：很容易低估自己在获得好运中的作用。

提高运气最有效的方法之一是在你的组织内提高可见性。当然，也有非常快速、非常负面的方法来提高你的知名度，所以我将稍微修改一下这个说法。你的目标是以做好事而闻名，同时尽量减少你为此消耗的组织带宽。

### 为什么可见性很重要

[凯蒂·赛勒-米勒](#_bookmark34)凯蒂·赛勒-米勒描述了可见性是晋升为Staff的关键因素

*我还没有充分谈论的一点是沟通和透明度。晋升为Staff的一个重要部分是确保你的工作是可见的，人们知道你的名字，并且你拥有良好的声誉。*

Staff-plus 职位是领导职位，公司通过授予你这样的职位，是将你纳入其领导团队。该团队的现有成员希望确信他们正在扩大队伍，吸纳他们信任的人，如果他们不了解你，他们就无法信任你。

如果你在公司内部的可见度不高，这很可能会被认为是小圈子或把关行为。相反，如果你在内部广为人知，这可能会让人觉得是为了保持对担任领导职务的人员的一致期望和标准所必需的代价——如果你不熟悉他们的工作，又如何保持一致性呢？

有趣的是，简单地反思一下，具有包容性的组织是如何减轻验证人们是否适合加入其领导团队的负面把关影响的。答案是，他们设计了一些机制，以确保潜在领导者的全部范围都能接触到那些将评估他们是否适合担任领导职务的人。相反，包容性较差的组织会在无意中将访问权限集中在那些最积极地自我宣传的人身上。

### 内部可见性

[这个工作 物质 致贵公司及公司领导。这条路也是最](#_bookmark10)

但有时这还不够，还有其他一些策略：

撰写和分发更多长期存在的文档，如架构文档或技术规范

 领导（并在较小程度上参与）公司论坛，如架构评审、公司全体会议和学习圈

在 Slack 上为你的团队和同伴的工作加油你也可以通过电子邮件而不是 Slack 来加油

[分享每周笔记](https://lethain.com/weekly-updates/)每周与你的团队和利益相关者分享你的工作笔记，以便其他人可以在感兴趣时访问你的笔记

为公司的博客投稿

参加，甚至可能主持，你团队或组织的办公时间

找到合适的活动组合，这些活动能够发挥你的优势，没有过多的工作量，并且对你来说是真实的。如果你从未宣传过你的工作，自我推销可能会让你感到尴尬。你永远都不想完全失去那种尴尬感——克制是有帮助的——但你必须适应其中的一些。

### 高管可见性

要晋升到领导岗位，最重要的一种内部可见性是高管可见性。使用[晋升材料](#_bookmark20)，你将与你的经理建立可见性，但更进一步是有帮助的。尤其有价值的是找到机会与你经理的经理建立关系，但所有积极的、在该层面的可见性都会对你有所帮助。

这些人往往在批准晋升到Staff-plus职位的会议室里，他们很少支持那些他们不了解其工作的人。

### 外部可见性

用外部可见性工作来补充你的内部可见性工作是有帮助的。有很多成功的没有外部存在的Staff-plus工程师，但是[许多人发现，外部可见性有助于他们的职业发展](#_bookmark17)。

与完全专注于内部相比，建立外部存在的一个优势是，有更多的空间为自己创造一个利基市场和名声。内部努力往往最终会与你的同行竞争注意力，而外部努力则不会。

说到如何为自己和你的工作创造这种可见性，可以像Keavy McMinn或Dan Na一样在会议上发表演讲，像Michelle Bu一样参加播客，像Katie Sylor-Miller的ohshitgit一样将问题转化为网站和书籍，或者像Stephen Whitworth的High Growth Engineering一样创建邮件列表。[凯维·麦克明](#_bookmark32)[丹·娜](#_bookmark39)[米歇尔·布](#_bookmark30)[，将一个问题变成了一个网站和一本书，比如凯蒂](#_bookmark34)[赛勒-米勒的*糟糕，Git*,*高 增长工程*.](https://highgrowthengineering.substack.com/)

### 你*应该*专注于可见性吗？

你总能在你的组织内获得更高的知名度，但在某种程度上，提高你的知名度可能会减少其他人为自己创造知名度的机会。内部知名度并非严格意义上的零和博弈，但它受到那些你想看到你的工作的人的注意力的限制。

我的建议是使用[晋升](#_bookmark20)来确定缺乏知名度是否会阻碍你的晋升过程。如果是这样，努力清除这个门槛，但不要走得太远。知名度是一种短暂的货币。学习和发展自己是一种永久的货币；一旦你完成了清除前者悬崖的最低限度，就专注于后者。

# 决定更换公司

****

*我在Fastly被聘为首席工程师。所以，老实说，对我来说，最大的因素是更换公司。我所做的工作类型并没有发生巨大的变化，但更换公司最终使我获得了这个头衔。-* [*凯维·麦克明*](#_bookmark32)

我的父亲是一位经济学教授。在他二十多岁完成博士学位后，他开始在一所大学教书，在那所大学获得了终身教职，并在四十多年后退休。在科技行业工作，这听起来像一个童话故事。

拥有四十年业绩记录的软件公司很少，而[四十年](https://lethain.com/forty-year-career/)只效力于一家公司的员工就更少了。过去有一种说法，许多工程师在每家公司工作一到四年，以最大化他们的股权授予，然后跳槽到下一家公司。如果这种情况曾经发生过，那么对于那些渴望或达到Staff-plus职位的人来说，这肯定不是常见的行为。

相反，一般来说，只要情况支持他们的成功，这些人就会留下来并因留在给定的公司而获得奖励。如果这些情况发生变化，他们往往会在不久之后离开，或者花一段时间耗尽情绪储备后离开。

从高级工程师职位晋升到Staff-plus职位需要数年时间来建立知名度和社交信誉，这使得如果你觉得你*离晋升只有一步之遥*，就很难离开。离开，感觉就像是从头开始。

再说，正如[凯维·麦克明](#_bookmark32)所描述的那样，人们通常通过加入一家新公司来获得他们的第一个Staff-plus头衔。即使你拥有所有的内部信誉，有时离开才是最有效的前进道路。

什么才是适合你的正确决定？

在深入探讨之前，我想先承认两种截然不同的跳槽经历：一种是拥有特权的灵活性，另一种是受到严格的限制。你的居留许可可能取决于工作担保的签证。你可能需要赡养一个大家庭。你可能被限制在雇主稀少的地理区域。这些建议主要针对前一种情况，对于那些在技术职业生涯中足够深入以追求Staff职位的人来说，这种情况更为常见。如果这不能反映你的情况，你绝对应该忽略它。

### 为什么离职有效

最了解你优势的公司是你目前的公司，他们也最有可能给你Staff+的职位。然而，实际上授予这个职位取决于太多的环境因素，以至于在实践中并非如此。

如果你的团队非常资深，那么可能很难证明你在Staff工程师级别的贡献，因为它被归功于你的同事。你的经理可能预算有限，没有空间再增加一名Staff工程师。你可能缺乏内部赞助人。你的公司可能根本不需要额外的Staff工程师。这些都可能意味着，即使你应该得到晋升，但在你目前的公司你也不会得到晋升。

相反，当你面试新的职位时，你可以一直面试，直到找到一家能够授予你头衔的公司。你也可以故意选择在早期阶段的公司面试，这些公司可能会更看重你的经验。面试过程还会带来一位自动的赞助人——招聘经理——他们的激励机制在面试过程中与你的激励机制最为一致。

技术面试对于成功的预测是不一致且不可靠的，这对行业和公司来说都是不利的，但如果你决心获得Staff+的职位并愿意进行广泛的搜索，那么这对你有利。面试创造了进行“偏差套利”的机会，找到一家不成比例地重视*你的*公司。这可能是一家重视有会议演讲曝光度的人、你设计API的经验，或者[你关于编译器的博士论文](#_bookmark42)的公司。

类似地，有时你会陷入一家公司的困境，你的声誉阻碍了你的前进。也许你在指出包容性问题后被贴上了“难相处”的标签。也许你在午餐时让一位有影响力的主管感到难堪，他们正在阻止你的晋升。一家新公司让你摆脱了这些包袱。

是的，当然，你是否真的能在科技行业*真正*地摆脱任何东西，这始终是一个悬而未决的问题。有时会感觉有点小圈子。如果你在科技中心、大型公司工作过，并且工作超过十年，那么你几乎肯定与面试你的人有共同的联系。

如果你在一家公司有过糟糕的经历，也许你的经理是个恶霸，或者你正在经历自己生活中一个充满挑战的时期，这会让你觉得有一片乌云笼罩着你的未来。也就是说，就像一般的面试过程一样，推荐信和背景调查是非常随机的。如果你需要任何进一步的证据，可以看看那些在知名公司一次又一次被雇佣的连续骚扰者。

### 离开前要尝试的事情

如果你计划因为缺乏兴趣、兴奋感、支持或机会而离开，那么至少先探索一下内部环境是值得的。这让你在获得更换公司的诸多优势的同时，还能保留你的人脉网络。根据你公司的规模和增长速度，这可能不是你的选择，但有些人每两到三年在同一母公司内转换角色，并且发现

这是一种保持投入和学习的有效方式。

另一方面，如果你正在考虑因倦怠或精疲力竭而离开，有时可以协商带薪或无薪休假，在那里你可以花几个月的时间来恢复精力，通常与转换内部角色相结合。这在较大的公司中更为常见。（如果你想知道，不，你的同事休育儿假不是“休假”或“度假”。）

### 没有工作就离开

说到倦怠，如果你特别倦怠，值得考虑在没有找到另一份工作的情况下离开你的工作。有一个相当简单的清单来确定这是否适合你：

你的签证支持这样做吗？

你是否有足够的经济保障，至少一年不工作？

你是否在一个高密度就业市场远程工作，或者你对你的下一份工作地点有弹性？

你面试表现好吗？

你能否向问你为什么在没有找到工作的情况下就离职的人，清晰地阐述一个连贯的理由？

是否有人可以提供关于你工作的积极评价？

如果所有这些都是真的，我不认识任何一个后悔休假的人。但是，请记住，只有那些休了六个月以上长假的人才觉得获得了新生。休假时间较短的人也很感激，但通常只是部分恢复。如果你真的要休假，我强烈建议你把你的经历写下来。即使你不分享你写的东西，它也会帮助你处理这些经历。

### 冒险一试

如果你在目前的公司几乎要晋升到Staff级别，那么绝对有另一家公司会给你Staff的头衔。至于你是否喜欢在那里工作，或者在晋升后是否能得到支持，这很难预先确定。

更难预先确定。如果你的内部声誉受损，或者你一直处于晋升的边缘，但却成了不断变化的标准的受害者，那么如果头衔对你很重要，你应该认真考虑换工作。在某个时候，你必须听听你现在的公司在告诉你什么。

相反，如果你对目前的角色感到满意，只是对头衔不满意，那么可以考虑是否可以更有意识地追求晋升，而不是离开。许多人在晋升到Staff级别以上的道路上遇到了瓶颈，使用诸如[晋升材料](#_bookmark20) 之类的技巧可以帮助你摆脱困境。如果你已经尝试了所有方法，认真对待自我发展，但仍然无法达到目标，那么可能是时候离开了。

也就是说，很容易过度思考这些事情。很少有人会讲述他们十年前留在或离开某个工作的故事。

## 找到合适的公司

获得Staff级别以上职位只有几个魔法：在转换角色时谈判争取头衔，或者找到一个支持性的环境来“原地烘焙”，同时通过一位有权力的赞助人建立你的内部信誉，他会为你辩护。这两种魔法中最重要的试剂是选择合适的公司来执行它们。

好消息是，你正在申请一家新公司，虽然你可能需要投入*几周*的时间来确定你是否能在那里获得Staff职位，但你不需要投入*几年*的时间。另一方面，如果你正在寻找一家可以加入并在其中成长的公司，那么你将踏上为期数年的未知组织之旅。这是一个令人望而却步的决定，选择适合*你*的公司将对你是否能获得Staff级别以上的职位产生相当大的影响。

### 找到一个能充分重视你的地方

如果你的目标是职业发展速度，那么加入一家公司，无论出于何种原因，这家公司都非常重视你的优势。例如，Fastly 重视 Keavy 的 API 设计经验，而 Stripe 重视 Dmitry 在编译器方面的工作。如果存在一个与你的特定能力相匹配的缺口，而这个缺口限制了他们的成功，那么你对公司的影响将是独一无二的。[Keavy的API设计经验](#_bookmark32)[Dmitry在编译器方面的工作](#_bookmark42)

运作良好的组织会因为你的优势而重视你。不太好的公司会因为你的身份而重视你。例如，一种将攻击性视为领导力的文化确实会提拔和重用最具攻击性的人，但这对他们的文化和团队是有害的。有时你会发现一家公司对你的净价值评估是合适的，因为它重视你的附加价值，而不重视你认为有价值的贡献；这看起来似乎可行，但总的来说，这是一种令人沮丧的处境。

### 精英主义者和程序主义者

当你试图确定一家公司来进行 Staff 转型时，

在你的决策中，有很多公司价值观需要考虑。其中一个特别重要的是理解公司的领导层是否从根本上认同以例外为主的“精英主义”世界观，还是以一致性为主的“程序主义”世界观。很少有公司完全存在于这个连续体的某一端，但大多数公司都严重倾向于一个方向。

当然，人们不会用这些术语来描述自己。第一种风格几年前会称自己为精英主义。现在这个词已经不受欢迎了，他们会避免使用它，但他们的核心信念仍然完好无损。这种风格在硅谷尤其常见，它以例外驱动为主，并集中精力来款待一小群珍视的个人。一般来说，在这些公司里，如果你能与他们认为的高潜力模式相匹配，你就会非常成功。如果你不符合，你可能会遇到困难。

你会发现另一种类型的公司认为一致性驱动公平。他们是那种[执行政策，而不是例外情况](https://lethain.com/work-policy-not-exceptions/)的人：他们设计时钟，启动它，观察它滴答作响，并进行偶尔的维修。这些公司往往更注重结构驱动，而不是直觉驱动，这可以为更多的人创造更广泛的机会，但他们也可能是僵化的官僚机构，沾沾自喜地看着他们的机器碾压一个人。

不可避免地，精英主义者和程序主义者都将他们的世界观视为一种道德立场，并且根据你是谁以及公司的领导是谁，你将会有截然不同的体验。

在面试过程中探索的一些方法，以帮助区分这些思维方式：

拥有严格的薪酬范围并*实际遵守这些范围的公司*往往由程序主义者管理。那些愿意避开其范围的公司是精英主义者。

那些为了让个人加入而设立一次性职位的公司，往往由精英管理主义者运营。那些按照计划好的职位招聘的公司，则是程序主义者。

那些采用临时或非结构化面试流程，希望获得

你的“感觉”，特别是对于高级职位，往往由精英管理主义者运营。更多的结构意味着程序主义者。

那些临时进行新的、没有规则的面试的公司，往往由精英管理主义者运营。那些严格评估你，即使他们不让你发光，往往是程序主义者。

精英管理主义公司和程序主义公司在把你推向Staff-plus职位的几率上，都没有天生的优势。相反，这取决于你的身份和公司领导层人员的身份。根据这些因素的匹配程度，你可以估计你在追求Staff-plus职位时会遇到的支持和阻力程度。

### 原型

大多数公司只招聘一到两个[员工原型](#_bookmark6)，即使他们都使用相同的头衔来描述他们所招聘的职位。如果你想弄清楚特定公司的偏好，最有效的方法是联系他们现有的一些Staff-plus工程师，了解他们所做的工作。大多数公司不会刻意考虑他们支持哪种Staff工程师，所以直接询问的效果往往不如预期。

规模足够大的公司最终至少会有一些人在每种原型中运作，但这需要很长时间才能实现，通常*在*他们的工程组织扩展到*数千*名工程师之后。

### 增长

如果你只在快速增长、成功的初创公司工作过，那么你可能不会想到，在Staff-plus工程职位上可能会缺乏额外人员的空间，但对于增长较慢的公司来说，没有足够的工作或预算来支持更多领导职位的人员，这种情况非常普遍。对于尚未达到产品市场契合度的公司来说，这也是一个常见的限制——当公司需要保持高度一致，同时频繁变更时，领导职位有限——对于那些恰好在销售以开发者为中心或技术基础设施产品的公司来说，可能存在例外。

如果你加入一家快速增长的公司，新的Staff（职员）机会将会自然而然地涌现。在增长较慢的公司中，你可能需要等待某人腾出他们目前的职位，另一个职位才有可能出现。这并不是说如果你不想加入一家快速增长的公司就应该加入——它们通常压力很大，并且运行着长期过时的流程——这只是另一个需要考虑的因素。

### 赞助

在新公司获得Staff-plus（高级职员）的职位需要公司内部有人相信你，并且愿意克服相当多的组织摩擦来为你争取到这个头衔。晋升到Staff-plus需要一位相信你的经理和管理链，以及他们愿意克服*更多*的摩擦来为你争取到这个头衔。如果没有公司内部有权力的领导者愿意将他们的组织资本投资于你，你就无法获得领导职位。

在寻找一家可以追求Staff-plus职位的公司时，一个重要的部分是确定一家你能够拥有有效赞助人的公司。在当前公司之外进行面试通常能有效地为你找到赞助人：你未来的招聘经理往往有充分的动机向你提供Staff职位。同样重要的是，你的时间投入很高，但与在一家公司工作两年后才意识到你永远无法实现目标相比，仍然*相对*较低。

[飞楔](https://en.wikipedia.org/wiki/Flying_wedge)最容易找到的赞助人是以前和你一起工作过的人。一位高级领导加入一家公司，然后带上他们以前的同事的飞楔模式是一种众所周知的、理所当然地令人鄙视的模式，它依赖于这种内置的推荐人作为赞助人，但如果适度使用，它不一定是 токсичным。

这也是拥有外部存在和网络能够极大地帮助你搜索的地方。那些看过你的演讲、阅读过你的博客文章或赞同你的推文的人更有可能在面试过程中以及后来的晋升讨论中成为你积极主动的赞助人。[一个](#_bookmark17)

### 持久性

特别是如果你在职业生涯的早期，并且正在寻求晋升到Staff职位，那么考虑公司的持久性非常重要：当你希望你的工作最终能晋升到Staff-plus时，五年后这家公司还会存在吗？

更微妙的是，你还必须考虑你未来的Staff赞助人的寿命。有一些优秀的工程领导者正在创造公平获得Staff-plus职位的机会，但这些机会可能会迅速变成一个价值观绿洲，一旦赞助领导离开公司或改变角色，它就无法维持自身。[价值观绿洲](https://lethain.com/values-oasis/)

通过确保你加入的公司的商业模式确实有效，并且为与组织最高领导层在价值观上保持一致的领导者工作（这样，即使他们离开，你仍然与潜在的赞助人保持一致），从而降低持久性风险。

### 节奏

在长达四十年的职业生涯中，总会有一些时候，你休息充分，渴望一个具有挑战性、能让你全身心投入的角色。也会有一些时候，你感到精疲力竭。如果你接受了一个需要你付出超出目前承受能力的角色，你会伤害自己。在担任技术领导角色时，尤其重要的是要确保公司的节奏期望与你能够提供的相符，因为你会被部分地评估为公司节奏的榜样。[四十年的职业生涯](https://lethain.com/forty-year-career/)

### 以及其他一切

领导岗位的职位搜索比典型的软件搜索要慢得多，需要几个月而不是几周，而且试图加快速度很少奏效。当您评估一家公司是否有可能成为您达到Staff-plus职位的有效场所时，您还必须像在典型流程中那样，对其他所有方面进行评估。

深入挖掘你的耳语网络，寻找有毒的问题和个人。确保这项使命是你能够持续支持和参与多年的。寻找那些你能在工作中学习的人。如果你发现这些Staff标准把你拉向一家你原本担心的公司，那么你很可能会后悔这个决定。

## Staff-plus岗位的面试

当您决定面试高级工程师职位时，您大致知道会发生什么。您会[刷新你的简历，完成*破解编码 面试*,](http://www.crackingthecodinginterview.com/)

如果您能以类似明确的期望开始Staff-plus面试流程，那就太棒了，但大多数公司都在努力应对他们的Staff-plus面试环节。这可能与您获得的高级工程师职位的面试完全相同。这可能是一个工程经理环节，并添加了一个编程问题。也可能完全是其他的东西。

适应不确定性是Staff-plus角色的核心部分，因此一个特别乐观的人可能会将Staff-plus面试的状态视为展示您技能的好机会。如果您不那么乐观，您可能会觉得有点沮丧，但一点准备工作可以大大提高这些面试的可预测性。

### 划清界限

您职业生涯的工程领导阶段可能会持续二十年，但如果您从角色的角度来考虑，您如何度过这段时间可能会归结为四到五个关键决策。每个决策都是稀缺资源，您应该有意识地分配它们。在开始面试流程之前，花一些时间完善您愿意参与的流程类型的标准，并反思[合适的 适合你的公司.](#_bookmark26)

在面试过程中，您会收到一些信号，表明该公司不太清楚如何面试Staff-plus工程师。因为大多数公司的Staff-plus面试流程都很平庸，所以您不应该自动退出运行不佳的流程，但您应该考虑哪些

这些信号代表着你不想跨越的界限。

在Staff-plus职位上，很多人划出的一条线是他们不愿意进行面试编程。这通常意味着他们在许多公司用来评估职业生涯早期候选人的算法问题方面速度较慢或犯更多错误。不练习的人采取这种立场是因为他们已经决定，一家关心快速编程的公司很可能会滥用其Staff-plus工程师。这是一条你想要划出的线吗？也许吧，自己决定。

### 调试过程

在你划定界限之后，接下来，你需要弄清楚你正在面试的公司使用的实际面试流程。询问这些问题可能会让公司重新考虑你的候选资格，但这总是可以向招聘团队和招聘经理询问有关你的面试流程的更多细节。在Staff-plus级别，如果你不要求更多细节，这几乎是一个值得关注的点。公司希望你成功，而了解流程是准备的重要组成部分。

在开始面试之前，需要了解的三个最重要的事情是：

1. 面试形式有哪些，包括他们评估什么？
2. 是否有任何面试需要特别准备？
3. 面试官是谁？

一旦你回答了这些问题，剩下的就是准备工作了。记下你希望如何处理不同类型的问题。为任何演示面试准备材料。简要研究面试官，以便根据他们的背景调整问题。

这也是调试你是否处于正确流程中的时刻。如果你的面试小组主要由职业生涯早期和中级工程师组成，那么很少会产生Staff-plus的职位；面试小组将无法评估你的优势，而且人们通常会抵制比自己更高级的职位。如果没有深入探讨你之前的成就，也没有演示机会，那么同样很难证明支持Staff-的专业知识。

加量优惠。

如果面试的流程不对，但你非常擅长该流程所评估的衡量标准，那么你仍然有可能获得Staff-plus的职位。但是，如果你对这些附带的衡量标准不太自信，那么请礼貌且建设性地提出你的担忧，或许可以引导招聘人员参考[围绕以下内容提供的资源 设计员工级别以上的面试流程。不要让惯性把你拉入](#_bookmark49)

了解职级评定在公司发展过程中的哪个阶段进行尤其重要。有些公司会发布带有特定职级头衔的职位，这样你就可以直接申请你认为合适的职级。如果你希望获得高级工程师的职位，那么就申请Staff工程师的职位。然而，许多公司将这些职位作为临时头衔，并在之后确定；其他公司则非常严格。唯一的办法就是询问。

对你的面试过程进行更多的控制可能会感觉非常不自然，而且理论上，你可能会因此错过一些机会，但这是一个好的结果：你的目标是找到最好的领导机会，而不是第一个可用的机会。

### 好好收尾

即使你顺利通过了面试过程，也要始终[协商细节](#_bookmark28)，并记住好好收尾。向你的推荐人简要介绍职位的细节。向面试官发送后续邮件。接受提供的交流机会，并提出经过深思熟虑的问题。在这种情况下，只要你花时间走完最后一英里，它就是最容易的。

## 协商你的offer

早在2012年，Patrick McKenzie就写了[薪资谈判](https://www.kalzumeus.com/2012/01/23/salary-negotiation/)一书，此后该书成为软件工程师薪资谈判的事实指南。这是一篇很棒的文章，也是关于如何协商任何offer（包括Staff-plus offer）的入门读物：如果你以前没有过多考虑过这个话题，就从这里开始。

在你的职业生涯的大部分时间里，你得到的offer都是以一种相对公式化的方式产生的。也许他们有一个薪酬计算器，或者他们根据你之前的薪酬来确定，但这是公司的系统在驱动这些数字。然而，存在一个offer谈判发生转变的阈值，公司愿意进行定制化的offer，而不是系统驱动的offer，但你需要以某种方式直觉到你已经跨越了该阈值——没有人会告诉你。

这种定制化的offer*开始*在薪酬方面具有更大的灵活性，尤其是在你offer的股权部分具有更大的灵活性。也就是说，它不仅仅是原始薪酬。它还涉及到公司通常没有灵活性，但愿意为高级领导者做出私人例外的其他方面。

例如：

他们的标准合同可能会给予他们在离职后三个月内行使既得权益的权利，但他们可能愿意将一位杰出工程师的行权窗口延长至五年。

他们可能在运营上没有准备好支持每个人的提前行权，但可能愿意破例来完成一份高级工程师的录用。

他们可能愿意提供递延补偿计划，以支持具有税收优惠的支付时间表（这在上市公司中很常见）。

如果他们使用的是计量休假而不是无限制休假，他们可能会接受额外的休假天数。

他们可能会支持围绕在家工作、工作时间或

在他们目前不支持雇佣人员的州或国家/地区注册公司方面的灵活性。

很难知道一家公司的某个职位是否超过了门槛，也很难询问公司里其他的Staff-plus员工，因为他们很可能没有谈判这些方面（也许是因为他们是从内部晋升的，从未获得机会）。一般来说，如果一家公司有超过20名但少于500名员工，那么他们不太可能为你做太多的定制工作，除非你担任的是准执行职位——他们根本没有这样做的运营能力。然而，如果一家公司有数千名员工，但只有十几名与你同级别的人，那么他们很可能愿意谈判定制条款。

重要的是要有策略地谈判你想要的东西。如果你坚持所有商务旅行都要乘坐头等舱，你*可能*会得到它，但它也可能会传递一个关于你的优先事项的信息，而公司并不喜欢。

无论你试图谈判什么，都要花时间用叙述的方式来表达你的请求为什么对你很重要。例如，我曾与一位高级候选人合作，他将延长行权窗口的要求放在了刚买了房子，以及最近在离职后行使了前一家公司的股权的背景下。这比仅仅要求更多的现金补偿要容易接受得多。正确的叙述可以完成谈判，而且不会产生关于你的动机的负面信号。

# 故事

****

在我出版第一本书后不久，我意识到我不是那种喜欢阅读别人对我作品评论的人。最好的情况是，我感觉自己好了一会儿，但通常我只是感到难过。在那段我还在阅读评论的时间里，我读到一条评论说，这本书对于大多数人来说太以硅谷为中心了，无法从中受益。

这句话在我开始集思广益，最终写成这本书的时候一直萦绕在我的脑海中，我希望避免以自己的经历为中心，而忽略所有其他人的经历。更重要的是，我清楚地认识到我的职业生涯建立在一套特定的观点、运气和特权之上，我希望这本书能成为那些在行业中有不同经历的人的有用指南。

总而言之，这本书最精彩的部分很大程度上依赖于行业从业者的坦诚和富有洞察力的访谈，我很感激能够将这些访谈纳入最后的章节中。即使你到目前为止没有从这本书中获得太多，我希望你能在这些职业故事中找到一些独特的东西。

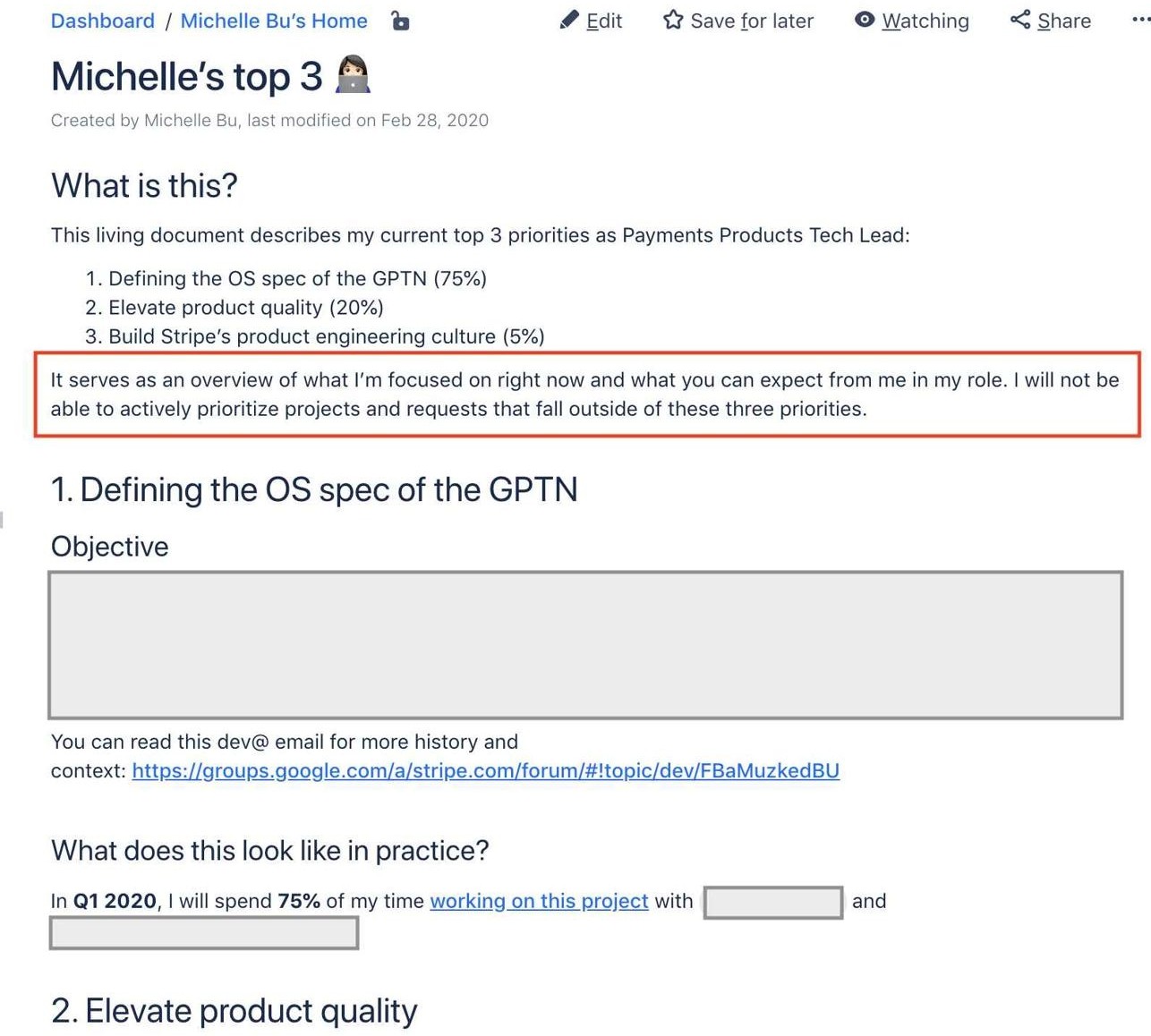
## Michelle Bu - Stripe支付产品技术主管

*这个故事录制于2020年4月。在她的*[*博客*](http://blog.michellebu.com/)*、*[*Twitter*](https://twitter.com/hazelcough)*和*[*领英*](https://www.linkedin.com/in/michellebu/)*上了解更多关于Michelle的信息。*

#### 请简单介绍一下您目前在Stripe的职位：您的头衔是什么，您和您的团队通常做什么样的工作？

我是Stripe的支付产品技术主管，直接与我们的首席产品官合作。我支持关键的倡议，并致力于缓解整个组织中紧迫的问题。我通常将80%的时间花在一两个大型跨组织的设计项目上。剩下的20%的时间，我用来审查和支持整个组织的技术和产品设计（特别是API设计）。

*我一直保留的“前3名”文档的示例：*



我管理着两名嵌入到高优先级领域的工程师。这既有助于我扩大我的影响力，也让这些工程师有机会涉足Stripe的许多领域。目前，一位正在研究核心支付API，另一位则专注于改善集成体验。我仍然按照IC梯队进行评估

—计划是每次的报告人数不超过几个。

##### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

Stripe的大多数Staff-plus工程师都在特定的团队工作。也有一些Staff-plus工程师同时拥有技术主管的头衔，并承担更广泛的职责

在特定产品领域或技术领域中的项目。

Stripe 有两种 Staff-plus 工程师：一种是范围深的，另一种是范围广的。

范围广泛的工程师通过从事模糊的、跨组织的的项目来产生影响。他们倾向于积累许多不同领域的背景知识，并在整个组织中的许多项目中发挥支持作用。这种类型的 Staff-plus 工程师在我们产品工程团队中最常见。

范围深的工程师往往是特定领域的主题专家。他们领导雄心勃勃的多年项目。这种类型的 Staff-plus 工程师通常可以在我们的产品基础设施和系统团队中找到。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

随着我进入目前的支付产品技术主管角色，这种情况对我来说已经发生了变化。（提供一些背景信息，支付产品由 20 多个团队组成。我们负责大部分面向用户的 API 和 UI 库。）

我已经开始使用“充满活力”这个词来代替“有影响力”。“有影响力”感觉是以公司为中心的，虽然这很重要，但“充满活力”更注重内在。找到*充满活力的*工作是我在 Stripe 工作这么久的原因，追求有影响力的工作。

当我在一个团队中直接工作时，当我能够直接与用户互动时，我感到最有活力，无论是帮助 #stripe IRC 频道上的用户，还是设计和发布用户可以无缝集成的 API。

在我目前的职位上，当我赞助的人发布他们已经完成工作的公告时，或者当我看到我帮助塑造或改变了一个工程团队对重要主题的理解模型时，我感到充满活力。是这些团队，而不是我，在日复一日地努力构建和支持他们的技术。我根据他们的进展来衡量我的影响，更重要的是，根据进展的方向以及他们的工作与公司目标的契合度。

最近记忆中的一个具体例子是，另一位 staff-plus 工程师和我对我们常见的 API 形状进行了分类：将一些标记为流程，一些标记为引擎，一些标记为配置等。这项工作的目的是为现有 API 的分类以及讨论和设计新 API 构建一个共享的思维模型和词汇表。人们在看到这些类别一次后就开始有组织地使用它们！正是在这些时刻，我感觉我正在通过传播有用的思维模型和想法来创造杠杆并扩大我自己的影响力。

我花时间在我们的几个评审论坛上，比如 API 评审，但通常这些论坛更像是代码评审。它们发生在设计过程的后期，因此它们在防止不良结果方面做得更好，而不是与团队合作来引导好的结果。当我可以为产品团队的工程师提供设计出色 API 的工具时，我感觉更有影响力。

##### 你能想到你作为Staff-plus工程师所做过的，在你达到这个头衔之前你无法做或不会做的事情吗？

我在Stripe工作了很长时间（从2013年开始！）。虽然由于我的任期，我一直拥有一定的影响力，但我的支付产品技术主管的角色（以及我直接向CPO汇报的事实）确实改变了人们与我互动的方式。我现在在工作中肯定感到更孤独了（并且正在积极适应这种新常态）。

有一件事需要逐渐适应，那就是现在大家期望我对我们讨论的任何事情都有自己的看法！当我还是一个直接在团队工作的staff级工程师时，这种情况并不经常发生。我记得在角色改变后不久的一次会议上，我比平时安静了一些，因为我有点累。后来我听说，演示者担心我不喜欢他们的提案，因为我什么也没说。这是我第一次意识到人们*期待我*发表意见并支持他们的想法！从那以后，我一直小心翼翼地在会议期间保持参与并提供反馈，即使只是明确地说我还没有完全形成任何意见。

有点让人感到困惑的是，有些人比我没有担任如此显眼的职位时更认真地对待我的意见，并且对我更好。以前，有

人们不合作或会无视我的意见的情况。我认为经历过这些是件好事。我足够自信（并且受到组织的足够信任），可以对他们的合作提供强烈的反馈，以便我可以确保类似的事情不会发生在像我这样的其他人身上。我现在担心我正在失去对这些互动发生在哪里的可见性。

##### 当您花更少的时间编程时，您如何保持对其他工程师的开发经验的同理心？

我担任新角色只有一年，所以我觉得还没有太脱节。也许这会随着时间的推移而改变。我以前是一个较小领域的技术负责人。在那个角色中，我编写了少量软件，并为我团队的一些“运行轮换”做出了贡献，我们在其中对传入的请求进行分类并修复紧急错误。

为了保持我在新角色中的背景，我花了很多时间与直接参与执行的工程师和PM进行一对一的交流。仅本周，我就进行了12次30分钟的1对1交流。我还关注Stripe报告的每一起事件。（我们有一个Slack群组，您可以加入该群组，以便自动被邀请到每个事件的Slack房间！）事件的嗡嗡声特别有用。通过阅读每个事件的详细信息，我能够估计我们的系统的现实与我每天思考的理想化架构/产品之间的距离。我想知道工程师遇到的问题的形态，他们陷入的失败深渊，以及开发环境如何或没有支持他们摆脱这些深渊。我将自己视为工程师向领导层的倡导者，因此对我来说，深入了解我们目前的现实非常重要。

##### 您是否花时间倡导技术、流程或架构变更？

在这一点上，我花更少的时间倡导*特定的*技术或程序，而更多的时间授权其他人倡导他们认为重要的技术和程序。我还努力成为人们可以寻求反馈的知识和支持来源，尤其是在跨领域的产品决策以及向组织其他部门展示想法方面。

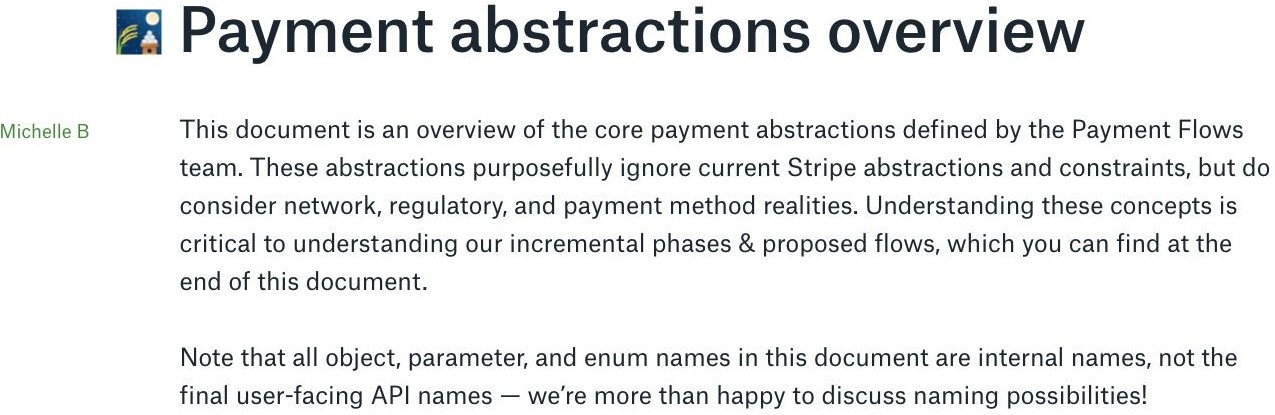
我确实会参与一些项目，在这些项目中，我会明确地思考理想化的架构和接口。然而，归根结底，迁移到任何理想化状态都将由各个团队完成，因此他们*需要*感受到一种主人翁意识和自主权。我花大量时间与实际做出日常决策的工程师和产品经理进行直接对话。理想的结果是，我们能够在方向上保持一致，然后他们能够在自己的团队中倡导我们的北极星，并做出良好的本地决策。

对于我现在正在进行的项目来说，这样做要困难得多，因为它涉及到为许多团队定义理想化的架构和接口

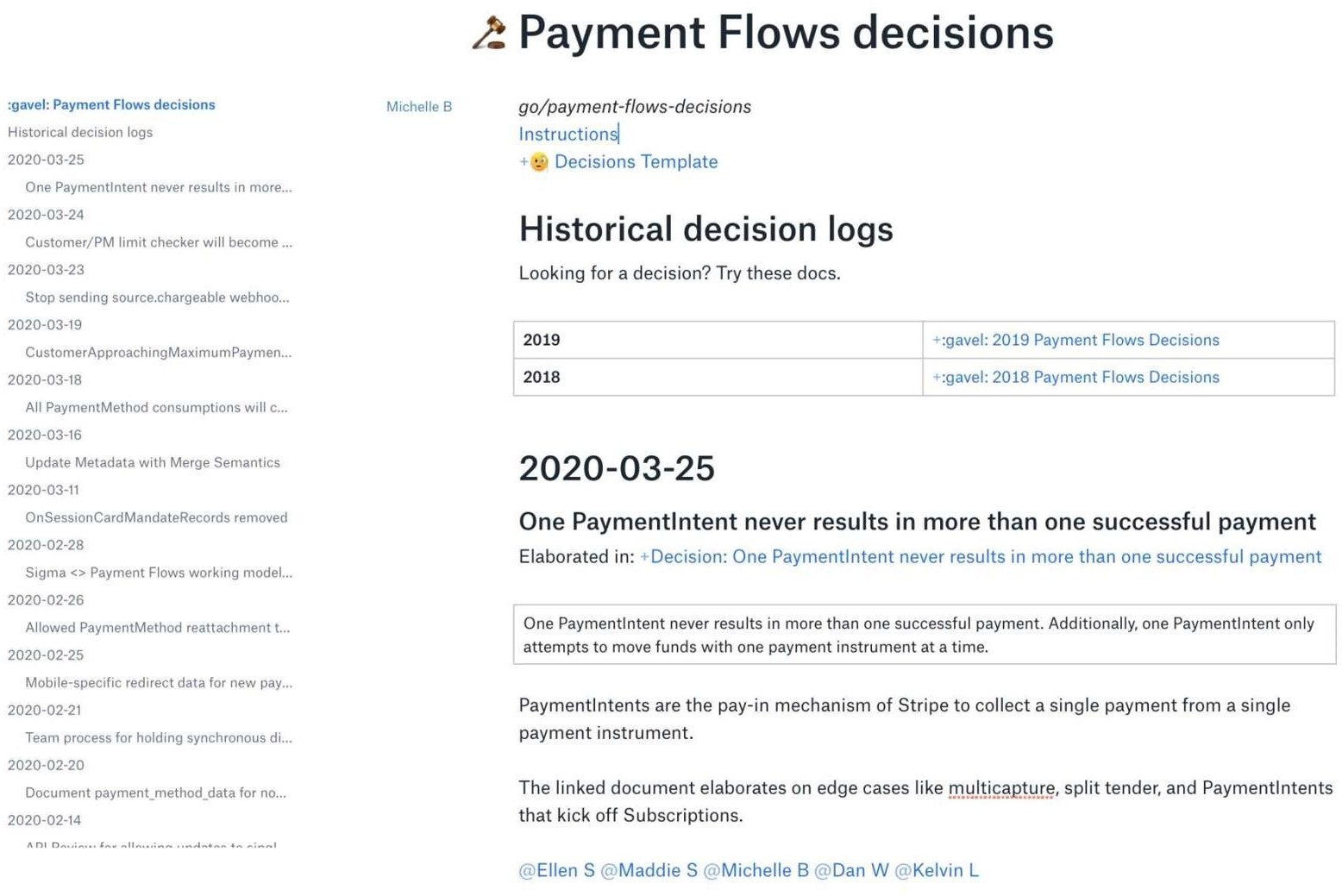
——基本上是所有从事支付工作的团队！我还没有找到一种可扩展的方法来让每个人都参与进来。即使是编写文档（最可扩展的信息分发方式！）也很困难，因为不同的团队（根据定义）从不同的角度来看待接口，因此对问题和解决方案的非常不同的框架会引起每个团队的共鸣。我们目前的方法是将对我们文档的审查视为用户测试：观察团队中的个人阅读文档，看看他们的光标在哪里移动，他们对什么做出反应等等。到目前为止，这种方法效果很好！

设计[支付意图 API](https://stripe.com/docs/payments/payment-intents)，这是对我们备受欢迎的 Charges API 的重新思考，以适应不断变化的支付领域，这是一个类似的跨领域项目，我之前参与过。这个愿景花了两年时间才完全被公司里的每个人接受。即使有了这种组织上的支持，我们仍然没有充分发挥其最初理想化设计的潜力。但这并不是一个错误！我们专注于在验证设计的同时为用户提供增量价值。我希望任何足够雄心勃勃的设计项目即使在我不再是团队成员时也能继续进行。使这项工作发挥作用的一个重要部分是把*所有事情*都写下来。

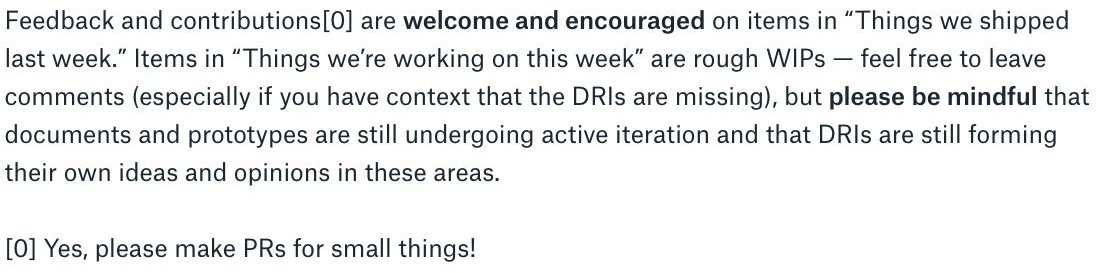
我们创建了一个规范文档，定义了我们理想化的抽象概念。即使在今天，该团队的成员仍然使用这些抽象概念作为北极星：



如果两个人问了同样的问题，我们会立即将其添加到我们保留的常见问题解答中。我们非常认真地对待每个人的反馈和问题，并将举证责任放在自己身上。最后，我们努力在工作中做到完全透明，甚至创建了一个决策日志，公司里的任何人都可以使用它来跟踪我们的进展。决策日志中的每个条目都简洁地描述了一个产品或技术决策，记录了谁参与了该决策，并链接到详细的支持性技术设计文档，这些文档通常包含完整的问题陈述和对备选方案的评估。



总的来说，我发现对于雄心勃勃的设计项目，极其透明，但也要明确你是否准备好接受反馈，这对于关心这个话题的人来说效果很好。以下是一些你可以在我领导的一些项目的（公开）笔记文档顶部找到的措辞：





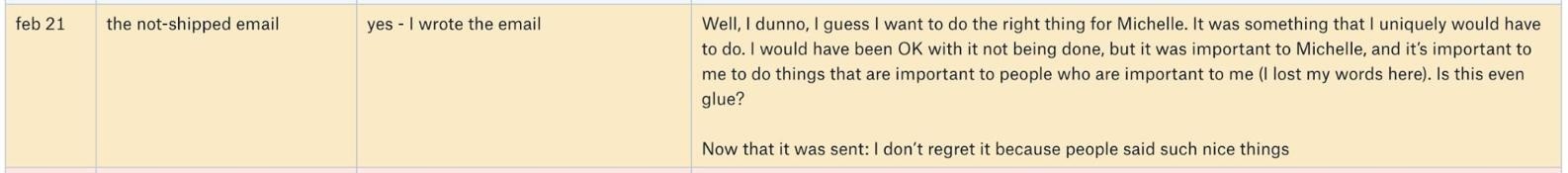
#### 赞助其他工程师是你角色中的一个重要方面吗？

是的，这是我最喜欢的部分之一！我非常关心与我一起工作的人——他们是我感到精力充沛去工作的主要原因。

对我来说，赞助的一个重要部分是为个人贡献者创造空间，让他们做他们关心的有影响力的工作。我很幸运，在我目前的职位上，我不必花时间积极证明自己有能力，所以我可以花大量时间在支持项目的角色上，并提升他人。我很少觉得我必须“邀功”或让我的名字明确地出现在我帮助过的项目的署名上（尽管当它发生时总是一种美好的感觉）。对于更开放的项目，有时对我来说，把我的名字借给这个项目是有用的。例如，我最近启动了一个产品质量指导计划，我在其中扮演更多的协调员角色，选择学员，将他们与导师配对，并偶尔审查他们的工作。我做的*几乎*没有导师们在项目中做的那么多，但我们能够启动这个全组织范围的计划，因为我赞助了它。

日复一日，我发现自己可以作为一个“橡皮鸭”来帮助那些想要获得关于如何驾驭一个复杂项目或解决技术分歧的建议的人。我发现这项工作——帮助他人取得进展而不直接参与——特别有意义。

最后，我总是牢记一份名单，上面记录着那些在各自领域表现出色的人，并在与他们的兴趣相符的可见机会出现时，为他们奔走呼吁。不过，这里需要把握一个平衡。我了解到，人们有时很难对我说“不”。最近，我请我合作的一个团队中的一位工程师发送一封关于她所做的一些出色工作的电子邮件。她在发送邮件后告诉我，她最初并不想这样做，但她也不想对我说“不”。然后她向我展示了她在她的“拒绝日志”中的相关条目：[“否 日志”:](https://twitter.com/amyngyn/status/1224160724072558594)



#### Stripe是你第一家全职工作的公司，而且你现在还在Stripe。你晋升到Staff级别的路径是什么？

我大学毕业后就加入了 Stripe。实际上，我加入 Stripe 后的成长曲线比我同期的其他工程师要慢。与那些雄心勃勃的大学毕业生相比，我最初四年的发展轨迹比较缓慢。我认为这部分原因是我编程的时间不长（我 2011 年才上了第一门编程课，2013 年就加入了 Stripe），还有部分原因是我在 Stripe 的第一个大型项目是一个为期 1.5 年的重写项目，但最终被取消了。

当Stripe首次引入职级时，我已经在公司工作了两年半，职级是L2，我们期望应届大学毕业生在6-18个月内达到这个职级。说实话，我当时非常失望，因为我的同龄人已经达到了“高级工程师”的级别。那时我已经做了很多有影响力的工作，启动了大部分新加入的产品工程师，并且在事故发生时总是挺身而出提供帮助。我非常努力地工作，做了很多有影响的事情，甚至在我主要项目之外！他们还想让我做什么？我应该*不*帮助别人吗？

回过头来看，基于职业阶梯，L2的评定是100%公平的。我努力工作，加班到很晚，因为我天生就比别人慢一点，写不出好的代码，这是因为我缺乏经验。我还没有为软件开发打下良好的基础，主要是因为我还没有足够的实践！我所做的有影响力的工作是有价值的，但也不是只有我才能做到的。当时，我是一个表现出色的L2。

在早期，我花了很多时间学习产品和支付领域，而不是编写代码。我花了很多时间在IRC上帮助我们的用户（开发者）进行集成。我做了一些较小的任务（错误修复、小功能、小问题的临时解决方案），这些任务在技术上没有挑战性，但对这些用户来说很重要。这类工作并不总是能促进工程师的成长（尽管我确实微调了我的调试技能——我现在是一个非常出色的调试员）。我还通过在Slack和工单上过度提供帮助，以及帮助团队找到我们用户的最佳解决方案，与其他团队和其他工程师建立了关系。在我最初的两年里，我帮助启动了大多数新加入的产品工程师。随着时间的推移，我开始以非常关心我们的用户和成为产品知识的源泉而闻名。（“用户至上”仍然是我最喜欢的Stripe运营理念）

原则。）

后来我意识到，在那段时间里，虽然我没有有效地朝着成为一名软件工程师的技术方面发展，但我实际上是在学习一些关键技能，这些技能让我能够非常迅速地从高级工程师晋升到骨干工程师，再从骨干工程师晋升到我目前的职位（总共只用了3年时间！）。事实上，我非常确定正是我在最初几年建立的关系，才使得一个被取消的1.5年的技术项目没有让我觉得对我的职业生涯造成太大的挫折。

[Stripe Radar](https://stripe.com/radar)当我开发最初版本的Stripe Radar和Stripe Elements时，我缓慢而有条不紊地构建我的技术基础。我坚信，只要我认真思考我的技术差距，认真填补我正在进行的项目中的这些差距，并用让我走出技术舒适区的项目来挑战自己，我就可以有机地建立和实践我的技术技能。那些软技能，公司内部的联系，用户至上，深入理解产品——这些技能花了更长的时间来学习，并且在我建立技术基础后，最终帮助我加速了晋升为Staff的道路。[Stripe Elements](https://stripe.com/payments/elements)

#### 你是否有员工项目？

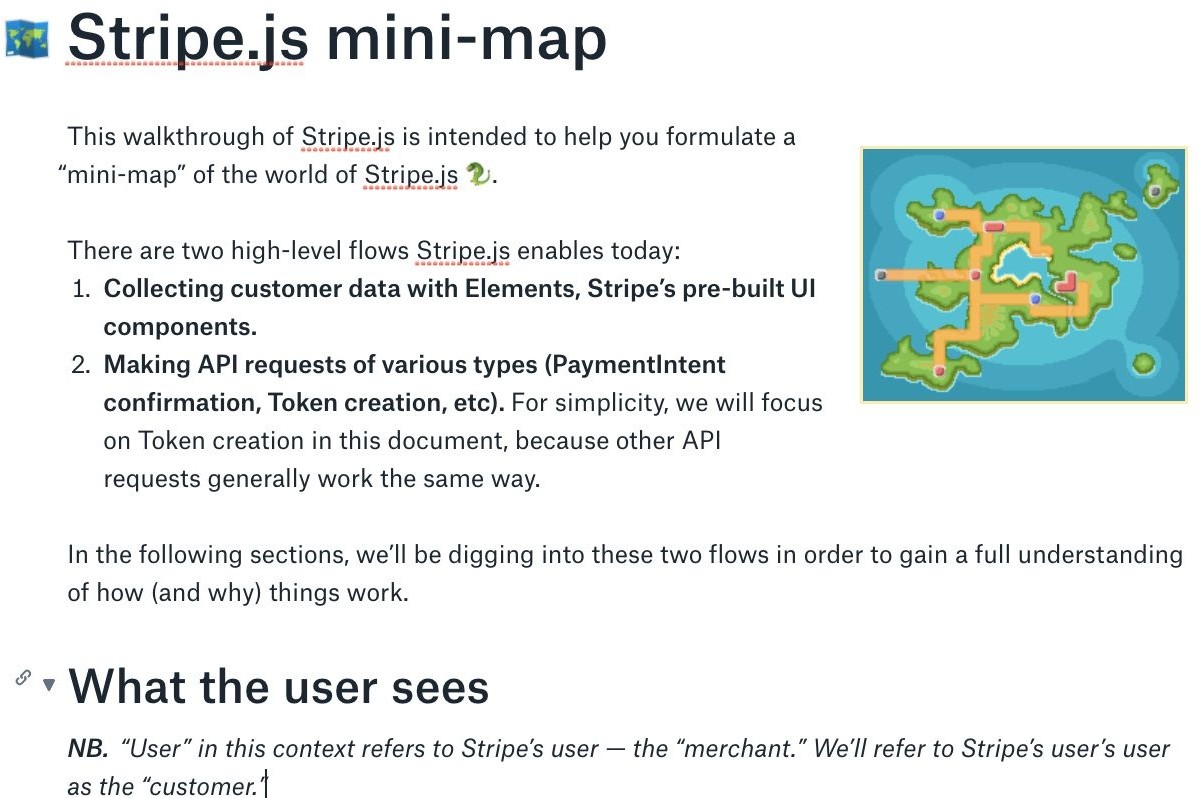
我曾在 Stripe 产品的几乎每个组件上都有广泛的工作经验。随着时间的推移，我参与的项目通常会分拆成各自的专门团队。在我担任高级工程师期间，我参与的两个项目特别有资格被认为是 Staff 项目：[Stripe Radar](https://stripe.com/radar) 和[Stripe Elements](https://stripe.com/payments/elements)。

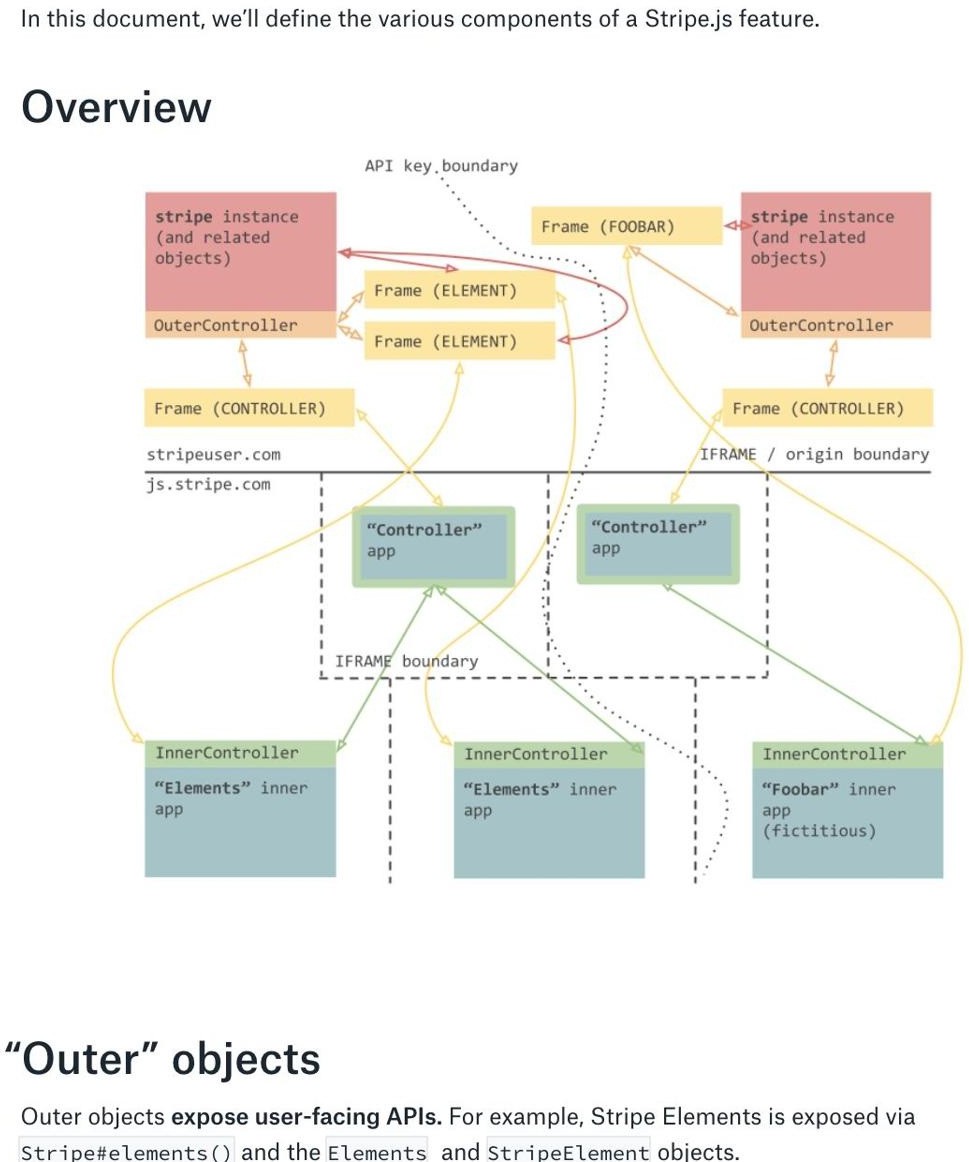
在 Radar 项目中，我们从零开始构建了一个全新的产品，在构建什么和为了尽快向用户推出产品而可以安全地取消什么方面进行了周全的权衡。当我们在 2016 年 10 月推出时，这是我们有史以来最顺利的产品发布之一。此后，它已成为一个非常成功的产品。

在 Stripe Elements 项目中，我构建了基础设施，从头开始设计了最初的 Card Elements API，并在不到 3 个月的时间内投入生产。这只有通过广泛的内部测试才有可能实现。在构建 Elements 时，我创建了三个具有不同设计

框架和设计（质量参差不齐）的小型电子商务商店，以测试其定制 API 的限制。从那时起，数十名工程师已成功地在代码库中进行开发，它是新的 Stripe Checkout 的所在地，最重要的是，我们对其原始 API 设计几乎没有后悔。随着 API 产品的扩展以及我们更多地了解开发人员如何在实践中使用它们，API 的重大更改总是可以预期的。我们很好地验证了最初的 API 设计，以避免这些重大更改，同时仍能快速发布。

为了确保新工程师能够顺利上手一个涉及大量 IFRAME 恶作剧的相当复杂的产品，我编写了大量文档。我发现讲述故事对于教导人们为什么事情需要这样运作非常有效：





现在回想起来，这两个项目的产品架构自发布以来总体上都保持得很好。当时，除了实施这些产品之外，我还需要在产品发布后等待一段时间，才能让产品选择在用户中证明其价值，并让技术选择在内部工程师中证明其价值，因为他们正在加速发展。

在工程师内部加速发展。

我认为这是产品中 Staff-plus 工程师的一个重要标准：不仅要构建可以发布的东西，而且要使其顺利推出，并在随着时间的推移继续成功和发展，并且尽可能少地做出令人后悔的选择。在产品开发过程中，总是会有一些妥协和功能被取消，尤其是对于新产品。Staff 产品工程师会深思熟虑地做出这些产品和技术选择，扮演各种不同的用户角色，以做出最佳选择，并为未来的工程师彻底记录粗糙的边缘。

##### 你是否需要整理一份晋升材料？

当我晋升为 Staff 时，我很幸运有一位非常积极地支持我晋升的经理。说实话，当时我并不真正理解如何以正确的方式撰写自我评估。我为来年想要学习的内容撰写了自我反思的发展计划，而不是记录我的工作的影响和范围。我的经理实际上完成了大部分工作，在他的评估中写出了我的影响。

还有其他一些事情帮助了我。首先，在那段时间里，我一直和同一个经理一起工作。如果你更换经理，那么你的经理会失去背景信息，这会将建立连续性的工作推到你身上。其次，我的经理管理着一个相对较小的团队，并且能够花大量时间跟踪和理解我正在从事的工作的细节。如果我向一位支持比如 10+ 名工程师的经理汇报，我可能需要在自己的晋升材料上投入更多精力。

##### 对你来说，达到Staff级别最重要的两三个因素是什么？

回想起来，对我来说一个可能令人惊讶的重要因素是（并且现在仍然是）我的冒名顶替综合症。它使我非常乐于接受反馈；学习和成长，并对与我的工作有任何关联的事情承担责任。它使我主动寻求对所有事情的反馈，从我对PR的评论的有效性到我如何主持一次特定的会议。如果有什么东西坏了（无论是技术上的还是组织上的），我都会感到不安，并且有内在的、深刻的动力去了解它并修复它。Stripe的任何

产品都不是“不是我的问题”。这发展成了两种超能力，对于Staff-plus工程师来说，这两种超能力可能比技术超能力更重要：

1. 真正倾听并同情他人。
2. 深入关心解决所有类型的问题。

当然，冒名顶替综合症是一把双刃剑。它经常让我感到害怕和不自在——早期我总是觉得我会因为不够快或不够有效而被解雇。随着时间的推移，我对自己的优势越来越有信心，但坦率地说，这花了我*lot*的时间以及我的经理和公司领导的积极强化。

##### 在产品工程部门工作是否比在基础设施工程部门更难达到Staff-plus级别？

我认为情况确实如此。我也认为在Stripe更容易一些，因为Stripe的核心产品是基础设施。这意味着在产品工程中有很多机会参与需要考虑规模、稳健性、迁移路径和精心设计的接口的项目。

如果你在一个主要构建UI的团队中，那么达到Staff级别肯定会很棘手，因为UI产品本质上更具临时性，并且允许更多的迭代和实验。要作为UI团队的工程师达到Staff级别的影响力，你需要能够创造杠杆作用。你可以通过构建精心设计的组件库、实验框架等来实现这一点。

作为一名产品工程师，建立杠杆作用的另一个方面是创建流程和系统来管理“产品债务”。人们经常谈论“技术债务”，但同样重要的是支持旧版本产品所造成的“产品债务”，并且大规模产品工程的许多困难都与管理产品债务和产品漂移（即需要相互操作的产品随着时间的推移朝着不同的方向发展）有关。我相信，公司产品债务的积累确实会在一定规模上在产品工程中创造出具有Staff复杂性的角色。

##### 你还记得有什么关于晋升到主管职位的建议特别

***对您有帮助吗？***

我没有得到太多*泛泛而谈的*有用的建议。我一路上得到过一些好的针对具体情况的建议，但这些建议总是与手头的情况联系在一起。

对我来说，最有用的通用学习是适应不确定性。高级职位上持续的成功取决于你适应和随着组织需求变化而发展的能力。

##### 给想成为资深工程师的人的建议？

一些注意事项：

我认为我特别幸运，遇到了我所拥有的那些经理。

我的兴趣一直与公司最重要的事务保持一致。（在这一点上，我有点不清楚我的个人兴趣（即，开发者产品、指导）是否一致，或者随着时间的推移，我是否将自己与对公司重要的事务保持一致。我*觉得*像是前者，但无论如何，我觉得我一直对我的工作非常感兴趣。）

我可能是公司里最受关注的产品工程师之一，所以工程师们有时会看到我在做什么，并试图模仿我的模式，成为一名 Staff-plus 工程师。这感觉很棒，我很幸运能够成为像我这样的人的榜样。

话虽如此，我对工程师的首要建议是，他们应该避免以导致他们从事自己不喜欢的工作的方式进行模式匹配。我对我所做的工作充满活力，与团队合作解决抽象建模和设计问题。经过多轮反馈后，一次又一次地尝试需要一定的毅力，说实话，这并不适合所有人。如果你更专注于达到 Staff 级别，而不是让自己从事能让你充满活力的事情，那么很容易最终陷入你不想要的角色。成为一名 Staff+ 工程师，尤其是一名范围广泛的 Staff+ 工程师，与成为一名高级工程师是非常不同的工作。

相反，去追求那些让你充满活力的事情，即使从表面上看，它似乎不能让你达到高级工程师的水平。作为一个成功的高级工程师，一个重要的组成部分是能够识别和确定具有影响力的全新工作范围，并说服其他人相信它的价值和影响。如果你所做的工作能让你充满活力，那么实际上更容易实现这一点，因为你会更喜欢深入思考你的工作！

##### 对于刚开始担任Staff-plus工程师的人，你有什么建议吗？

你作为Staff-plus工程师的工作是针对你的团队和组织的，重要的是避免采纳不适用于你情况的建议。例如，当我进入目前的职位时，许多其他的Staff-plus工程师都在专注于撰写个人章程，描述他们在未来1-2年内想要完成的事情。这种方法可能对深度工程师很有效，但对于像我这样需要快速响应组织变化和产品战略转变的广度工程师来说，效果并不好。

##### 你有没有考虑过工程管理？

我现在确实管理着两名工程师。也就是说，我并没有做很多传统经理会做的事情。我没有像招聘经理那样参与招聘，也没有经历其他经理会经历的那种绩效管理情况，因为被选入我团队的工程师已经是高绩效者。

我非常关心Stripe：当我看到任何不合适的地方，我都会感到焦虑，并想修复它。在一些组织中，我认为这可能会引导我走向工程管理，而不是我目前的职位，我很感激管理不是我唯一的选择。我的优势和兴趣在于产品工程和API设计与执行，我能够在我的职位上每天都发挥这些优势。

##### 你从哪些资源（书籍、博客、人物等）中学习过？

我喜欢阅读小说，我从伟大的文学作品中学到了很多关于世界的知识。我不太喜欢阅读非虚构的商业或技术书籍。当

涉及到学习与我的工作更直接相关的主题时，我珍惜我与同行的关系。我的同行会给我有价值的即时反馈，并帮助我理清我脑海中一直存在的答案。

Stripe还有一个名为“实践中的领导力”的项目，所有经理和一些高级工程师都会参加。该项目包括一个关于[适应性 领导力 这特别有帮助。从那以后，我应用了我所学的框架](https://www.amazon.com/dp/B004OC071W/)

我从来都不是一个向单一导师寻求建议的人。相反，我遵循我过去所说的“弗兰肯斯坦”式的，建立你自己的导师的方法，类似于Lara Hogan在她关于[培养一名管理者 百兽王将我与一位导师配对的项目从未真正让我感到](https://larahogan.me/blog/manager-voltron/)

我大部分时间都在处理那些没有简单、通用答案的困难、具体的问题。弄清楚正确的方法需要大量的背景情况，而局外人对此不会有太多的了解。

最近我读过并喜欢的一些非小说作品：

[第四稿，约翰·麦克菲](https://www.amazon.com/dp/B06X18NHC1/)：我大部分时间都在工作中写作，并且经常遇到写作障碍。但克服这一点非常重要，因为一篇好的书面沟通是传播思想和提升自己的最有效手段。

[创新公司，艾德·卡特穆尔](https://www.amazon.com/Creativity-Inc-Overcoming-Unseen-Inspiration-ebook/dp/B00FUZQYBO/)：这本书的语气确实让我感到惊讶，但这里有很多关于如何大规模培养创造性工作环境的知识。随着我们的产品组织和产品工程职能的增长，我经常思考这个问题。

[即兴，基思·约翰斯通](https://www.amazon.com/Impro-Improvisation-Theatre-Keith-Johnstone/dp/0878301178)：我将我的超能力（尤其是在公司发展壮大时）视为能够快速学习和适应，所以我喜欢阅读关于不同形式的学习和教学的书籍。这本书是关于学习如何表演/即兴表演的，并推动了关于教育的传统隐喻和叙事。

## 拉斯·卡萨·威廉姆斯 - Mailchimp 的 Staff Engineer（职员工程师）

*这个故事录制于 2020 年 7 月。在*[*领英*](https://www.linkedin.com/in/raskasawilliams/)*上了解更多关于卡萨的信息。*

#### 请简单介绍一下您目前的角色：您在哪里工作，您的头衔是什么，以及您和您的团队通常从事什么样的工作。

我是 Mailchimp 的 Staff Engineer（职员工程师），在数据服务工程组工作。数据服务可以被视为公司内数据工程的所在地。我们小组构建的系统主要支持我们的数据科学和分析师团队（即产品分析师、财务分析师、市场分析师等）。

我一直担任该小组的技术负责人之一，专注于构建可扩展的数据处理管道，为我们的内部分析平台提供支持，并支持关键商业智能计划的推进。我还承担了团队中很大一部分工程经理的职责（尽管我不是正式的工程经理）。在这个职位上工作了近 2 年后，我正在积极地将其过渡给另一位工程师。

#### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

在 Mailchimp，一旦你成为一名 Staff Engineer（资深工程师），你就是“工程领导层”的一员。由于正式的工程职级体系是几年前才实施的，因此对于“成为一名 Staff Engineer 意味着什么？”或“成为工程领导层的一员意味着什么？”的答案可能会有所不同。

在我看来，这关乎全球化思考。这意味着与其他成员合作，了解公司范围内的业务/产品战略，并将其提炼成工程范围内的技术战略，以支持产品、营销以及我们在其他职能部门的同事的执行。这意味着合作改进招聘、入职、跨团队沟通和生产运营等流程。这意味着合作发展

整个部门的技术和社交技能。

这关乎将全球化思考应用于本地。这意味着将您团队的（技术）计划/路线图与工程范围内的技术战略对齐；并有意识地决定何时偏离该路径，以满足您团队直接利益相关者的需求。这意味着与您团队的经理合作，采纳其他团队在招聘、入职和生产运营方面的成功实践；并分享您团队的实践，这些实践对其他人有益。这意味着从公司范围内的业务/产品战略中获取背景信息，并将其转化为它如何影响您团队的直接项目。这意味着有意识地为您的团队的个人贡献者创造机会，以提高他们的技能，获得知名度，并与公司内的其他人建立联系。

当然，我并非总是能做对。但对我来说，这是一种成功的运作模式。

##### 您每天如何安排时间？

如前所述，我一直在数据服务部门担任技术负责人之一（并承担了许多工程经理的职责）。

作为技术负责人，我负责定义并负责执行我团队的技术战略和方法。我努力将这一战略建立在对我们内部客户（或业务）的价值之上，并在需要时有效地在公司内沟通。一年几次，我会反思取得了哪些进展，以及业务内部是否发生了任何需要调整该战略的变化。我仍然定期贡献代码——当然比我团队中的其他工程师少；但重要的是，我保持“手不离键盘”的工作，以确保我的技术战略（和其他宏观层面的决策）能够从我团队其他成员的实际经验中获得信息。对我来说，一个重点是由于我设定的技术方向，培养和发展团队中的其他工程师。更具体地说：我努力帮助我的队友（通过指导或辅导）了解如何进行技术决策，以及它应该如何以它所解决的客户/业务问题和它所提供的价值为基础；我努力帮助他们了解如何推动工程项目

从问题陈述到生产发布/长期运营；我努力帮助他们了解针对不同受众的适当沟通实践（即工程同行、工程管理层、非技术利益相关者等）。总的来说，我努力帮助他们达到一种自给自足的水平，使他们能够在没有我在场或参与对话的情况下，以符合我的技术战略、方向和方法的方式进行操作和沟通。

我们小组没有产品经理。但是，围绕理解内部客户需求和利益相关者管理等方面的需求仍然存在。最近为我的团队聘请的工程经理负责此事；但我们分担了很多责任。我做了很多与内部客户聊天的工作，回答一些快速的一次性问题，了解对新工作的需求（即新数据集），推荐前进的道路，设定我们何时能够执行新工作的预期等。

我们小组也没有项目经理。但是，同样，这些需求是存在的。我坚信授权每个团队成员负责他们分配的项目的项目管理职责（即向利益相关者提供状态更新等）。我做了很多关于项目管理策略的指导，例如主动沟通风险/障碍，以便我可以帮助解除它们的阻塞，以及如何不断交付增量价值以保持势头。

我花了很多时间来构建被动的内部客户和业务背景。我会偶尔阅读我没有维护的仓库的合并拉取请求；我会阅读公开分享的技术规范和提案；我会偶尔参加数据科学和分析团队的各种内部演示，他们会在会上展示他们完成的项目或正在探索的想法。所有这些都是小策略，但它们有助于我了解情况，并且是我团队季度规划活动的一个关键输入。

作为一名技术负责人，我是工程技术负责人小组的成员。这是一个由工程部门所有技术负责人组成的正式小组，旨在分享背景信息，讨论想法，并帮助推进整个工程部门的技术路线图。这个小组有时会进行一些临时的对话和工作。还有一个为所有Staff、Senior Staff和

首席工程师准备的定期电话会议，旨在发现和讨论问题，在需要时分配负责人和行动项目，并普遍建立彼此之间的社区。我们与谷歌有着密切的合作关系；他们是我们的云服务提供商。偶尔，我会花时间与我们指定的合作伙伴团队讨论我们遇到的挑战、我们的计划、解决方案的正确方法、有用的正式培训等等。

随着我从技术负责人角色过渡出来，我每天的时间安排肯定会发生变化。但我真的不确定那会是什么样子。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

如果我能结束一天的工作，并且感觉我帮助我的同事们摆脱了困境并保持了势头，这总是一种有益的感觉。

这可能意味着帮助我的数据服务团队成员思考解决复杂技术问题的方案以及权衡，例如提供即时价值与长期可持续性。或者帮助他们思考如何深入研究一项新技术，以及它是否适用于解决我们的三大问题之一。

这可能意味着帮助另一个团队的同事思考他们在过去一个季度中的可交付成果，这些可交付成果是否提供了真正的业务/客户价值，以及如何围绕这种影响构建一个叙述，以为晋升辩护。

这可能意味着确定一项短期的“组织工作”陷入停滞，因为没有达成共识，鼓励一位参与其中的同事承担做出决定的责任（在获得相关方的适当反馈后），并推动其完成。

这种感觉我一直贯穿我的职业生涯；我享受我的同事的成功，就像我享受自己的成功一样。但担任技术负责人无疑提升了这样做的重要性。它从我喜欢做的事情，变成了我喜欢做并且对我的团队的健康也很重要的事情。

##### 你能想到你作为一名Staff-plus工程师所做过的，在你达到这个头衔之前你无法做到或不会做的事情吗？

如前所述，一旦你成为 Mailchimp 的 Staff Engineer，你就是 Eng Leadership 的成员。

值得注意的是，Staff+ 工程师通常对自己的时间有更多的自主权和掌控权。当他们抽出时间处理主要职责之外的工作时，遇到的阻力较小。这当然是我个人的经验。

我也获得了更多自然的机会来指导、帮助和支持团队之外的更多同事。这当然是我在晋升为 Staff 工程师之前就在做的事情；但晋升后明显增加了。这绝对是我一天工作中很有意义的一部分，所以我很欢迎。

行业内的同行经常认为头衔并不重要。但我百分之百不同意这种说法；我个人的经验以及在多家公司的工作观察都表明情况并非如此。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

从某种意义上说，我想是的。

我以高级工程师的身份加入了 Mailchimp。我立即被加入到一个项目团队（包括一位工程总监和两位首席工程师），旨在构建 Mailchimp 的第一个内部自助式分析平台。

这个项目的一个关键方面是**高效并高水平地执行**。不管好坏，有另外两位首席工程师意味着对我的期望可能不会那么高。但我能够立即投入并开始为项目的核心方面做出贡献，几乎不需要他们的指导；到最后，我成为了团队中的主要贡献者之一。我最终被正式任命为技术负责人，以帮助继续引导该项目的工作，因为它被吸收到我当前的工程组，即数据服务中。

这个项目的另一个关键方面是在**整个公司范围内的高度可见性**。该项目团队的工作被归类为公司层面的倡议。这意味着高层管理人员的高度关注；当然，也伴随着相关的压力。但整个项目团队能够始终保持良好的势头，并最终成功构建了分析平台的初始迭代版本。此外，我的经理和团队的首席工程师有意创造机会让我展示项目团队的工作；我最终在工程全体会议、公司范围内的全体会议上进行了展示，并在工程招聘活动中共同发表了一次技术演讲。由于该项目的可见性以及 Mailchimp 的总体文化，我能够在加入公司后很短的时间内与公司各个级别的工程师合作，并与来自其他部门的分析师互动——这是大多数人需要一年左右的时间才能获得的机会。

因此，这是做好工作并与更高级的工程师一起高效工作的结果；而且那些高级工程师（和其他人）有意让我获得在整个公司范围内的可见性和访问权限，即使他们是该项目的技术负责人。

重要提示：晋升直到我担任技术负责人并交付了大量价值后才到来。但这个项目绝对是导火索。

***你还记得在晋升到Staff职位时，对你特别有帮助的建议吗？***

首先是我经理[马克·赫德伦德](https://twitter.com/marcprecipice)的建议。他告诉我用第三人称写我的绩效评估。他的意思是，在给别人做评估时，你更可能赞扬别人，而减少批评。这很简单，但对我来说非常有用。奇怪的是，它在帮助我理解如何围绕我所做的工作以及它为企业提供的价值构建一个连贯的叙述方面发挥了作用。

其次是[丹·麦金利](https://mcfunley.com/)的建议，他是“Staff项目”中的一位高级工程师，我之前提到过。当我向他征求意见时，他直接指出了我的优点和缺点。他指出，我在公司内部建立关系的能力是一项重要的技能，因为工程中的人际关系/

社交方面不会消失；事实上，它们是完成任何事情的关键。

第三个建议来自[科达·黑尔](https://codahale.com/)，他是“Staff项目”中的另一位高级工程师，我之前提到过。他谈到了扩大你的影响力。具体来说：

*首先，有效地为其他工程师设定技术方向；其次，指导他们，并在你为他们构建的工作中培养他们的才能。*

这个建议是我思考自己作为技术负责人角色的核心；比如，真正有意识地为团队创造机会，扩展、锻炼他们的技能，并学到很多东西。

#### 对于刚开始担任Staff工程师的人，你有什么建议？

不要认为你能解决工程部门的所有问题；根据我所看到的，这会让人精疲力竭，而且你会很快变得厌倦。慢慢地积累这些东西。希望你是因为已经在Staff级别工作而被提升的；所以你不应该做任何非常不同的事情。继续做让你获得这个头衔的伟大工作。一旦你觉得你已经适应了，就从那里开始扩展。

沟通和构建叙述是关键。一定要写……很多。在思考问题或想法时，把它写下来（即使你不打算分享它们）。通常，如果我不能用一个连贯、简洁的段落来概括一个问题陈述或想法，这意味着我需要做更多的研究和/或我很难说服别人这是一项值得的投资。此外，书面文字可以让你更有效地扩展你的想法和围绕它们的讨论；比安排与每个可能的人开会来推销这个想法容易得多。

尽量让你的经理的工作更轻松。不要只是给他们带来问题；还要给他们带来（如果可能的话，多个）针对该问题解决方案的建议/提议，并征求他们的反馈。这样，经理就不必为你解决问题的所有工作（他们可能已经有很多事情要处理了）

盘子），他们有机会借鉴自己的经验来帮助你排除你的建议。有趣的是，让别人提供你的解决方案为什么不好的反馈，可能比让他们自己提出一个完整的解决方案要容易得多。

开始有意识地思考如何为你直接合作的其他工程师创造机会，以提高他们的技能，并让他们有机会接触到公司里他们通常无法接触到的人。

尽早建立关系。这在你需要花费一些社会/政治资本来对某件事采取立场时会有所帮助。例如，如果你第一次与某人接触时，你们正在为招聘问题的对立解决方案而争论，那么你一开始就处于劣势。需要明确的是，你不应该总是“取悦他人”。但是，如果你事先建立了关系，你与特定人员的工作关系以及在未来工作中进行富有成效的合作的能力可能会大大提高。

坦率地说，尝试应用所有这些东西是很困难的；而且这只是一家之言。把它当作餐厅的菜单；选择与你产生共鸣的东西并尝试应用它。然后，通过与下一个试图晋升为Staff的人分享你的经验来传递下去。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

我经常被问到这个问题。

我的确有一个成为CTO的目标；但这更多的是一个方向性的东西，以确保我不会对我的职业发展感到自满。在Mailchimp的背景下，晋升为首席工程师对我来说就足以实现这个目标。我已经在目前的级别上交付了真正的、切实的商业价值；所以我认为我可以继续这样做，并通过坚持个人贡献者的道路而继续感到满足。

此外，我一直在为我的团队履行Eng Manager的许多职责（但不是正式的Eng Manager）。所以，很多事情我都会

如果我转到管理岗位后想开始做的事情，我已经能够付诸实践并获得经验。所以，我确信这满足了我暂时的任何渴望。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

与各种技术和非技术受众进行关于技术的口头交流是一项技能。这是我每天都在努力培养和磨练的技能。我关注的一个人，从视觉上学习如何正确地做到这一点是[凯尔西·海托华](https://twitter.com/kelseyhightower)。另一个是我大学里几门软件工程课程的教授。他很少来上课；但当他来的时候，我通常学到的关于软件开发的知识比我在任何其他学术环境中都学到的要多。他们都知道如何很好地解释事情，并根据听众进行调整。

我偶然发现了一些博客，它们帮助我磨练和微调了我开发技术方法和策略的方式。第一个是Pete Hodgson的“实现架构战略”，它提出了一个框架，用于在功能交付和基础架构工作之间实现可持续的平衡。第二个是James Cowling的“垫脚石而非里程碑”，它是关于通过大型架构计划交付真正的价值。[*Pete Hodgson 的“实现架构战略”*](https://blog.thepete.net/blog/2019/12/09/delivering-on-an-architecture-strategy/)[*“Stepping James Cowling 的* 这是关于通过大型架构交付真正的价值](https://medium.com/%40jamesacowling/stepping-stones-not-milestones-e6be0073563f)

## Keavy McMinn - Fastly 高级首席工程师

*这个故事录制于 2020 年 3 月。在她的*[*博客*](https://keavy.com/)*、*[*Twitter*](https://twitter.com/keavy)*和*[*领英*](https://www.linkedin.com/in/keavy/)*上了解更多关于 Keavy 的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是[Fastly](https://www.fastly.com/)的高级首席工程师。Fastly 是一个边缘云平台，提供像 CDN 这样的服务。我在 OCTO（首席技术官办公室）工作，它由大约十名直接向首席技术官报告的首席或杰出工程师组成。OCTO 的每个成员都有自己的关注点，而我的关注点是担任 API 负责人。

#### 在您的公司，Staff-plus 工程师做什么？您如何安排您的时间？

在 OCTO 中，有几种截然不同的首席和杰出工程师。也有直接在工程团队中工作的首席工程师，而不是在 OCTO 中。在 OCTO 团队中，有些人从事互联网标准或学术研究，有些人进行深入的技术研究和原型设计，有些人帮助孵化一个团队来构建全新的东西。在我的 API 负责人角色中，我与更广泛的工程组织密切相关。

我们都在做不同的事情，但我们有一个共同的目标，那就是对事物采取整体的、长期的和系统范围的观点。我们还试图发现并帮助解决工程中可能被忽视或遗漏的事情。我们的首席技术官支持我们的工作，但不确定要从事的项目，这取决于我们自己。

我从未用百分比来考虑我的时间。我的一些工作分阶段进行，这周做的事情多一些，下周做的事情多一些。我大量的时间都花在写作、研究和与人交谈上。我会定期与构建 API 的团队和经理们开会。我会花时间把一个长期的战略分解成小块，进行研究，

并撰写关于此的提案。然后我必须在公司内部推广该提案。最近较少编写代码，但在其他阶段，我会构建演示或工具来支持更广泛的工作。编码工作仍然非常令人愉快。

##### 作为一名资深工程师，你觉得在哪里最有影响力？作为一名资深工程师，你做了哪些你在职业生涯早期不会做的事情？

作为一名普通工程师，很难抽出时间。你必须在定期安排的项目的时间和节奏限制内工作。作为一名负责人，你有信任、时间和空间去尝试一些事情。

当你拥有头衔时，你不需要花费那么多精力来展示你的资历。它有助于为他人设定背景。你从一开始就更受尊重，这一点非常明显。你还可以接触到高管，所以你可以更早地获得信息，并且可能在会议上拥有一席之地来影响事情。

##### 你是否花时间倡导技术、实践、流程或架构变更？

我被专门聘用来为API的战略方向设定方向。其中一部分是指导我们所做的技术方向和选择。我把它当作一种协作练习。你知道我会做研究的繁琐工作，我分析所有信息，提出权衡方案，并提出建议。我会考虑所有组织和工程团队的背景。

我提出我认为对我们来说最好的方案，人们可能会不同意。而且，你知道，他们经常这样做。我更多的是引导和影响，而不是说，“我有权告诉你该怎么做。”我从来没有见过这种风格奏效。

对于有争议的决定，我会与来自不同相关团体的代表会面。我会与一群工程师会面，告诉他们我打算推荐什么，并问“你们对此有什么看法？我遗漏了什么？”我还会与管理层和产品方面，以及可能还有法律、文档、安全或不同的人员会面，具体取决于项目。我也做过

相反的方向，即首先展示事物，然后进行通话以获取反馈，而不是仅仅等待人们在文档中留下评论。

##### 你倡导过什么？

在目前的工作中，我一直在努力倡导的一件事是API变更的设计文档。因此，在任何人编写任何代码之前，在进行更改的成本较低时，他们会写下用户工作流程以及可以启用这些工作流程的接口是什么样的。有时，事实证明，看似简单的事情实际上非常困难，特别是当团队不习惯锻炼这些能力时。

我的倡导方式是提醒我们自己，是什么痛点让每个人都感到，从而促使我们尝试做出改变。我们不是为了理论、代码美观或崇高的担忧而追求完美。我把它带回到“这些是每个人都说他们感受到的痛点，而这是一种你知道最终会有助于解决这些问题的 approach。”

我帮助其他人关心同样的事情，比如我将开始与更多的工程师一起进行 API 审查。在进行审查时，我尝试帮助教导其他人我在看什么，并在整个过程和对话中给予鼓励。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

这在我目前的角色中并不是一个重点。在 GitHub，我意识到我的资历和任期赋予了我特权，并赞助了一位工程师。我会给他越来越多具有挑战性的工作，鼓励他质疑任何不清楚或让他对我的工作感到好奇的事情，并为他争取更多的责任和认可。

##### 你是如何建立组织信任的？

在 Fastly，我从一开始就获得了信任。当我加入时，我被聘请来从事特定的工作。我记得我问，“这有一个时间表吗？”以及他们对战略的看法，并被明确告知他们

希望*我*去弄清楚并告诉*他们*。所以绝对有很多信任和责任。

当你建立这种信任而不是被聘用时，有利有弊。当你建立信任时，你会同时建立大量的背景知识，这就是我在 GitHub 的工作方式。虽然，我发现在我目前的工作中，没有背景知识，以全新的视角来看待问题实际上非常有用。这使得当人们认为“哦，好吧，也许我们一直都是这样做的”时，很容易提出质疑。不被过去束缚可以是一种解放。

##### 有一种流行的观点认为，成为一名 Staff 工程师需要完成一个“Staff 项目”。你是否有过“Staff 项目”，如果有，那是什么？

我从未听说过它被命名，但我理解这个想法。我确实领导并架构过那种类型的项目——解决棘手的工程问题，对公司产生重大影响——几次，但不幸的是，它们并没有促使我得到晋升。但它们确实促进了我的职业发展。这些项目给了我经验、知识和信心，让我能够以不同的方式定位自己。甚至可以公开演讲或知道“我做过 X，并且可以再次做 X。”

##### 公开演讲或公开露面对达到您目前的水平重要吗？

是的，我认为这对我职业发展来说总体上是一个巨大的因素。我认为这不是必须的，但我认为它肯定会有帮助，它对我很有帮助。我的第一次会议演讲是被邀请去做的——因为组织者认为我有一个有趣的视角可以分享，来自艺术背景。这很可怕，最初我想拒绝。但我母亲说服我答应了。所以公共演讲一开始与其说是一种深思熟虑的策略，不如说是一种稍微偶然的策略。

主要是，我喜欢在会议上遇到的人。后来，演讲者网络为我带来了工作机会。

##### 你是如何第一次获得Staff或Principal头衔的？哪些因素对你达到这个头衔贡献最大？

我作为首席工程师被Fastly公司聘用。所以，老实说，对我来说，最大的因素是更换公司。我所做的工作类型并没有发生巨大的变化，但更换公司是最终使我获得这个头衔的原因。

有人强烈主张专门聘用我，我确信这有帮助。他们不是我以前直接合作过的人，但他们熟悉我的工作。

##### 远程工作是否影响了你的职业轨迹？

就我所知没有。我一直是一名远程员工，我确信这在能够进行意外的对话方面起到了作用，但我已经这样做了很长时间了。你会刻意努力进行对话并建立关系。而且，我的公司在很大程度上是分布式的。我可以想象，如果远程办公是少数，或者公司没有完全接受分布式办公，这可能会成为一个潜在的问题。

##### 你是否得到过关于晋升为Staff的建议，这些建议对你特别有帮助？

没有，我得到了一些不好的建议。这太老套了，但就像“这很棒。现在你必须再次证明它。”有些建议会促使人们朝着英雄的方向发展，比如认为你需要发明一些不寻常或神奇的东西才能获得资格。一个人可以朝着很多不同的方向发展。工程阶梯常常助长这些信念。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

我首先想到的是找到你的同伴或支持网络。就像管理一样，你爬得越高，就越感到孤独，因此找到仍然会挑战你并且你可以一起集思广益的同伴非常重要。即使他们在你相似的工作领域，甚至在不同的公司，也没关系。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

我尝试过一次，但没有获得好的体验。我意识到这根本不是我的热情所在。我对工程管理非常尊重，不会为了正确的理由之外的任何理由去做它。正确的理由是支持其他人。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

[充满激情的程序员](https://www.amazon.com/Passionate-Programmer-Remarkable-Development-Pragmatic-ebook/dp/B00AYQNR5U/)对我来说，会议是一种资源，而且还能与成熟、低调、优秀的工程领导者和工程师一起工作。Chad Fowler 和他的书《热情程序员》名列前茅。Dave Thomas 是另一位，我刚开始学习 Ruby 时经常参加他的研讨会，他的书《实用程序员》也是一本伟大的书。[务实的程序员](https://www.amazon.com/Pragmatic-Programmer-Journeyman-Master/dp/020161622X)

## Bert Fan - Slack 高级工程师

*这个故事录制于 2020 年 5 月。在他的博客、Twitter 和 Linkedin 上了解更多关于 Bert 的信息。*[*博客*](https://bert.org/)[*Twitter*](https://twitter.com/bertrandom)[*领英*](https://www.linkedin.com/in/bertrandom/)

#### 请简单介绍一下您目前的角色：您在哪里工作，您的头衔是什么，以及您和您的团队通常做什么样的工作。

我是 Slack 平台团队的高级工程师。我很幸运能在我们推出 Slack 应用目录后不久加入 Slack，因此我有机会帮助 Slack 平台发展成今天的样子。[Slack 应用目录](http://slack.com/apps)

很难概括我所做的工作，但目标始终相同：使开发人员能够在 Slack 之上进行构建，从而使我们客户的工作生活更简单、更愉快和更高效。这方面的例子包括构建新的平台功能、提高 API 性能、编写文档以及与合作伙伴、内部集成商和第三方开发人员合作，以确保他们可以构建有影响力的软件。

#### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

Slack 的 Staff-plus 工程师所做的工作差异很大，具体取决于您所在的公司部门、团队的组成和规模以及业务对您的需求。Staff-plus 工程师通常是项目的技术负责人，这意味着他们帮助编写技术规范、从各种利益相关者那里获得反馈、与设计和产品部门密切合作以决定要构建什么，并领导项目的技术实施。他们还指导其他工程师、改进我们的面试流程和工程文化、开发工程流程和工具，并为重构和技术债务提供技术指导。Staff-plus 的全部意义在于使其他人能够更好地工作 - 成为一种力量倍增器。

我的角色包括我提到的所有内容，但我倾向于较少关注特定的技术，而更多地关注技术所实现的功能。所以我会

可能会花时间制作几乎肯定会被丢弃的概念原型，或者收集围绕特定用户流程的使用指标，以更好地了解如何改进系统。我经常在 Slack 平台上构建应用程序，以保持我对开发人员体验的真实了解，并积极尝试在其他人的平台上构建，以了解哪些有效，哪些无效。与公司其他工程师相比，我编写的代码进入生产环境的要少得多，我对这一点完全没问题。

##### 你能想到你作为一名Staff-plus工程师所做过的，在你达到这个头衔之前你无法做到或不会做的事情吗？

去年，我尝试了很多不同的产品创意，其中一些已经发展成为我们平台的实际功能。我之所以有机会这样做，部分原因是，我通过我成功交付的其他项目建立了信任，但拥有 Staff-plus 的头衔似乎让我在选择从事的工作方面拥有更大的灵活性。

不管好坏，我也参加了很多以前从未参加过的战略和规划会议。如果你一直想从领导的角度了解香肠是如何制作的，那么也许你应该考虑一下你是否真的想知道香肠是如何制作的。

##### 当您花在实际开发上的时间越来越少时，您如何与事物的实际运作方式保持联系？

我发现最好的方法是与公司各地的工程师定期进行一对一交流，并花大量时间倾听。如果你花时间建立关系，让工程师觉得他们可以对你坦诚，你就能了解到很多关于工程现状的信息。作为一名资深工程师，你对别人的收入或下次晋升没有实际的影响力，所以如果你让自己容易接近，工程师们可以更坦诚地对待你。

##### 您认为达到资深工程师级别最重要的两三个因素是什么？您加入的公司、您的地理位置或您的教育背景对您的职业道路有何影响？

我来自一个相当优越的背景——我大学毕业时获得了

计算机科学学位，没有债务或学生贷款，这在一定程度上给了我灵活性，让我可以离开一份工作，而不必担心如何支付房租或是否能找到新工作。我利用这种灵活性，在我想去工作的地方变得更加挑剔和更有策略。

我承认大多数人没有这种选择，但对我来说，我认为从事有意义的事情很重要——那些我个人使用过的，我认为对世界产生积极影响的事情。我相信这些选择已经得到了回报，因为像这样的公司会吸引志同道合的人，他们会继续去其他公司，希望这些公司也能以同样的方式保持一致。这不是一个精英统治，你的人脉很重要。我得到过工作，因为我在公司网站上像其他人一样申请，但也得到过工作，因为我笨拙地给一位多年未联系但尊敬的经理发了邮件，并且知道他们也尊重我作为一名工程师。我们所处的行业有很多选择来支配你的时间，如果你足够幸运，有灵活性和特权来选择，那么你不定期评估你所从事的工作，就是在亏待自己。

也许在某个时候，你会成为那种工程师，当你在 Twitter 上宣布你开始一份新工作时，以前和你一起工作过的人会创建一个日历提醒，提醒四年后你完全获得股权，这样他们就有最大的可能性挖走你，但在那之前，你还是要给那些你想再次合作的人写一些尴尬的邮件。

##### 你还记得在晋升到Staff职位时，对你特别有帮助的建议吗？

我听过的最好的建议是，达到资深工程师级别通常是运气、时机和努力的结合。以下是我观察到和亲身经历过的一系列事件：

1. 与你的经理建立一种他们完全信任你，你也完全信任他们的关系。坦诚直接地告诉他们你想要什么。建立这种信任需要你成功地完成他们要求你做的事情。
2. 因为你的经理信任你，当他们听到对公司有重大影响的项目时，他们会支持你领导这些项目。或者，你必须自己找到或创建一个项目，并倡导它的实现。这要困难得多，但仍然是可行的。
3. 成功交付项目。
4. 该项目对公司具有重大影响。
5. 因为你成功交付了一个对公司具有重大影响的项目，所以很容易支持你晋升为Staff（高级工程师），你的经理也很乐意这样做。

希望你能明白运气和时机是如何影响这个简单计划的——如果你和你的经理相处不好怎么办？如果你的经理离开公司或得到晋升怎么办？如果你在公司的一个没有有趣项目的领域怎么办？如果这个项目注定要失败怎么办？如果项目成功了但没有产生影响怎么办？

这些都是有可能发生的，我无法给出任何通用的建议来克服这些问题，除非有时候你永远不会得到晋升，你应该诚实地面对自己，并认清自己所处的境地。在这种情况下，有时候获得晋升的唯一方法就是离开公司，去做别的事情。你可能会在职业生涯的后期以比你离开时更高的职位回到公司，但就像一段失败的恋情，你重新审视它，你仍然想待在那里吗？或者是否有太多的包袱让它无法运作？

##### 对于刚开始担任Staff工程师的人，你有什么建议？

这在工程领域有点像一个老生常谈的笑话，但很多人进入这个行业是因为他们不喜欢与人交谈，但要胜任Staff Engineer（高级工程师）的工作，你可能会花很多时间与人交谈。我认为你可以在职业生涯的早期通过专注于编写越来越好的代码来取得进步，但在某个时候，你应该将重点转移到更好地与他人合作上。信任他人，并给予他们做出技术决策的自由（即使是你不同意的决策！），

理解他人的动机，学会给出困难的反馈，知道何时选择你的战斗——这些都是有用的技能。

如果你还没有这样做，试着成为人们想与之共事的工程师。在每家公司都有少数工程师，如果你离开你的工作，你会试图绕过一份非招揽协议，再次与他们合作。成为其他人的那些工程师之一，它将为你的职业生涯打开许多扇门。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

在我职业生涯的早期，我曾经告诉我的经理，我担心我迟早要进入管理层，他们说了一些类似这样的话：“别担心！在我们公司，我们有两条不同的职业道路，管理层道路和IC（个人贡献者）道路，并且在IC道路上，有与最高管理职位相当的职位，所以你永远不必为了在公司晋升而转到管理层。”虽然技术上是正确的，但现在对我来说显而易见的遗漏是，在很多公司，管理层道路比工程道路要清晰得多。

你在工程阶梯上爬得越高，你所能效仿的例子就越少，而且这些例子似乎越来越遥不可及。当你开始深入研究时，你可能会意识到，有人在他们的公司被收购时获得了这个头衔，或者他们是某种编程语言或框架的作者，或者他们为公司带来了数千万美元的收入。

我的很多同事出于各种原因进入了管理层，我怀疑其中一个原因可能是我所描述的更明显、更可靠的晋升途径。但我坚信这不应该是你主要的动机。如果你有开放的日历，看看你经理的日程安排以及他们每周进行的 1:1 会议的数量。那是你喜欢的日程安排吗？编写代码的愿望并不是非黑即白的，因为有编写代码的技术主管经理职位，也有从不编写一行生产代码，并将大部分时间花在 Google Docs 或 Dropbox Paper 上的 Staff-plus 工程师职位。但在我的职业生涯中，我从未不得不

开除某人，拒绝晋升，或者写绩效评估——我知道我宁愿站在哪一边。

## 凯蒂·西勒-米勒 - Etsy 的前端架构师

*这个故事录制于 2020 年 8 月。在她的*[*网站*](https://sylormiller.com/)*、*[*领英*](https://www.linkedin.com/in/ksylor/)*和*[*Twitter*](https://twitter.com/ksylor)*上了解更多关于凯蒂的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我在 Etsy 工作，Etsy 是世界领先的手工商品在线市场。我们让卖家可以把他们的商品拿出来出售，并卖给世界各地的人们。我们真正专注于为人们提供独特、特别或手工制作的商品，作为大型商店那种没有人情味的替代品。

我目前在前端系统团队，这是一个产品基础设施团队，负责我们的前端架构——包括我们的 PHP 视图渲染框架，尽管我没有非常积极地参与团队目前正在做的工作。在过去的几个月里，我一直专注于 Web 性能——以咨询身份处理所有与性能相关的事情——改进我们的监控和报告系统，识别需要改进的领域，并随时为产品团队提供与性能相关问题的帮助。

我认为，Web 性能是许多公司要么忽视要么不关注的东西。当我刚开始在 Etsy 工作时，由于像 Lara Hogan 这样的人，我们拥有出色的性能文化，但由于几年前的组织变革，我们不再有 Web 性能团队，我认为作为一个组织，我们安于现状，降低了 Web 性能的优先级。现在，我们正在把它重新摆在首位，因为行业内围绕如何定义和衡量“良好”性能的方式发生了很大变化，特别是对于 SEO 而言。谷歌真的在推动 Web 性能成为公司关注的搜索排名标准的一部分。所以这绝对是一个首要考虑的领域，特别是对于零售商来说。

#### 在你的公司里，“正常”的 Staff-plus 工程师是做什么的？你的

##### 角色是那样吗？还是有所不同？

关于我们如何看待“staff engineer”的角色，有趣的是，我们把一个人可以成为资深员工的两种不同方式都放在一个叫做 staff engineer 的桶里。但实际上，有两种不同的桶。

有人可以通过成为特定主题领域的专家，真正承担起技术负责人的角色，从而在他们的角色中变得非常资深，他们正在推动他们的团队或组织的技术方法和路线图。然后还有另一种成为资深工程师的方式，这些人扩大了他们的工作范围和重点，以便他们思考具有交叉性的问题，他们正在推动跨多个团队运营的系统和实践的创建。我认为第二个桶就是一个架构师。这并不是说你不是一个主题专家，而是说你的影响力范围大于作为一个特定团队的技术负责人。

在Etsy，我们有几个资历级别：高级工程师I和II，然后是Staff I和II，然后是高级Staff，这被认为相当于总监级别的角色。从技术上讲，我是Staff Engineer II，我也是这样看待自己的，但我的具体角色是前端架构师。这意味着我不只是负责我的团队在做什么，我还要负责关注Etsy在前端领域所做的一切。未来会是什么样子？我们需要解决哪些问题？我们将如何实现目标？我会考虑所有这些，并倡导能够在公司层面帮助我们实现目标的技术方法。

##### 作为一名架构师，你花多少时间进行软件开发？

是的，这很有趣。我是一名前端架构师，但到目前为止，我最近一直在写的主要内容是SQL，因为我正在做大量的数据分析。我一直在查看我们的性能指标，以找出需要改进的领域，以及修复哪些最具影响的问题可以提高性能和业务指标。我会在这里和那里写一些JS或PHP，但主要是为了帮助团队解决问题，或者运行与性能相关的小型实验，或者如果有一些重要的事情需要完成但其他人没有

有时间。

我确实发现我的速度变慢了，而且随着我的日程安排被会议填满，我需要更长的时间才能真正找到专门的专注时间来编写代码。所以我不认为你希望我再写太多代码了！我更专注于识别机会领域，然后尝试将其作为我的团队或其他团队应该做的工作来推销。

##### 您每天如何安排时间？

我想说50%的时间在开会，剩下的时间每天变化很大。有时我用另外50%的时间写文档，有时我在SQL中做大量的数据分析，有时我在Slack上与多个团队和角色的人交谈。有时，当项目落到我头上时，我的会议负担会增加一些，我会联系其他团队以了解更多关于他们正在做的事情，或者试图影响他们做出改变。变化很大。

##### 我发现这些角色的人经常难以量化他们的工作，你有没有找到任何有用的方法来衡量你的影响？

我很高兴听到这个，因为这确实是我非常非常挣扎的事情。我总是有很多不同的项目和讨论在同时进行，而且我有一种不好的倾向，会被最新的事情吸引住，或者让我的注意力游移，所以我必须非常认真和有意识地组织我的工作和笔记。我总是关注我可能做的所有事情，并选择对我来说最有效或最重要的事情去做，这可能非常困难。

直到我担任架构师的角色，我才意识到我多么依赖sprint和JIRA看板，以及完成一个ticket并将其移动到“完成”列的仪式，以此来检查自己，并知道我正在完成我需要完成的事情。现在我没有那种团队环境来帮助我组织我的一天，我不得不更多地依赖我自己的待办事项清单，而且我仍在努力改进我的系统。

有一件事肯定有帮助，那就是确保我一直在跟踪

在我每天完成的所有不同任务中——记录会议、电子邮件、Slack讨论等等。然后，当我和我的经理进行正式的季度目标检查时，我会回顾我所有的笔记，并意识到，哇，我帮助工程师修复了六个不同实验的性能问题，或者我影响了这个团队在新功能上采取更好的方向，或者我给了那个工程师帮助他们的反馈。这些都是小事，在当时感觉没什么，但加在一起就显示出真正的影响。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

我最喜欢做的事情是发现以前从未解决过的新的或独特的问题，想出一个疯狂的想法来解决这个问题，然后我那些才华横溢的同事们会采纳这个想法，并真正地将其付诸实践，从而创造出令人惊叹的东西。这始于从人们的工作中获取大量信息——看到这个团队在做 x 方面有问题，而另一个团队在做 y 方面有问题。然后，你将所有这些信息与你的经验以及整个行业正在发生的事情混合在一起，让它在你的大脑中沉淀一段时间，直到最终一切都变得清晰，你意识到这两个问题背后更深层的原因是 z，所以你提出了一个解决这个问题的计划，而这个问题真的很难解决。

在我转到架构师角色之前，我的团队负责我们的设计系统组件。更改或修复共享组件的问题非常困难，因为我们没有每个组件的标记和模板的单一来源。公司里的每个人都没有重复使用相同的模板文件，而是将 HTML 复制并粘贴到许多不同的地方。因此，当我们必须更改组件时，很难找到所有需要更新的地方，因为组件的各个部分分散在不同的地方进行管理——有时在 JavaScript 中，有时在 Mustache 中，有时在 PHP 逻辑中。

所以我有一个疯狂的想法：如果我们扩展我们的自定义 PHP 框架，以便在 Mustache 中启用可重用的模板块来表示我们所有的组件，并且我们能够像在 React 应用程序中那样轻松地将它们组合在一起，那会怎么样？我出去做了一个概念验证并写了

一份项目提案，并将其提交给团队。然后团队真正地接过了这个球并开始行动，他们构建了支持这个组件系统的基础设施，结果比我自己做的任何事情都要好得多，也更强大。

我真正喜欢的部分是发现问题并创造性地思考我们如何解决它，然后四处推销这个提案，并让其他人参与到工作中来执行它。

##### 从事前端工作的人员是否可以像开发人员生产力或基础设施角色的人员一样为公司创造杠杆作用？

是的，绝对可以。我个人只认识少数其他专门从事前端的 Staff Engineer，而且我认为前端作为一种技能在行业中的价值不如我认为的那么高。我非常幸运，能在像 Etsy 这样的公司找到一份工作，这家公司倾向于通过拥有计算机科学基础知识来招聘“全栈”工程师——我学习的是计算机科学，并且我在整个堆栈中都有工作和理解的经验。但实际上，我的热情和我的重点一直是前端，因为它就在你的用户面前。我希望看到更多的公司重视前端，因为我相信我们带来了宝贵的技能和独特的思维方式。

至于成为一名 Staff Engineer，我认为一个好的 Staff Engineer 的品质超越了你所从事的堆栈。最终，Staff Engineer 需要能够将工程决策视为一系列权衡，而清晰地表达这些权衡是一种你可以从堆栈中的任何角度获得的技能。

我还认为 Staff Engineer 应该对所有与其专业相邻的领域有广泛的了解。对我来说，在前端工作，我投入了大量的时间和精力来理解营销、业务目标、用户体验、视觉设计、服务器上的视图和业务逻辑层、我们如何将代码发送到浏览器、浏览器如何将所有这些代码转换为网站，以及用户如何与之交互。在所有这些不同领域拥有专业知识使我更容易看到我的技术决策的更广泛影响，并更好地理解这些权衡。

特别是，对用户有同理心是所有类型的工程师都需要培养的一项重要技能，而且我认为在许多不明白他们实际上确实有用户的基础设施或开发人员支持组织中，这项技能可能会被低估！我在前端基础设施部门工作，我们真的努力将自己视为产品工程师——只不过我们构建的产品是供其他工程师使用的系统。所以我们有客户。我们有用户。当我们考虑我们构建的系统的 API 时，我们是在为用户设计我们的 API，我们需要了解我们的用户——也就是产品工程师——才能做好这一点。

我个人认为，倾向于前端的人非常适合担任技术专家，因为他们非常习惯于不断思考用户以及用户将如何与他们构建的东西互动。用户同理心是前端人员带来的超能力。

##### 当你自己做的开发工作减少时，你如何保持同理心和对公司开发现实的认知？

建立人脉，建立人脉，建立人脉，建立人脉。一对一的交流尤其重要，因为我是一名全职远程工作者。当然，现在每个人都在远程办公，但作为一个不在完全分布式团队中的远程工作者，你必须真正了解你在和谁说话，并确保你与多个团队和多个小组建立联系，以利用这些网络。

在Etsy，我们非常幸运，我们有一些不同的员工资源小组，供员工在公司内部进行联系。我积极参与技术领域中边缘化性别身份的ERG（MAGIC），这很棒，因为该社区的成员在工程部的每个部门工作。远程员工社区也是如此。我抽出时间指导更年轻的员工，定期进行1:1的交流，并参与Slack讨论，以培养和发展这些联系，因为了解整个组织中正在发生的事情对我们有很大帮助。我还尽量确保我特别与产品工程部门的工程师交谈，因为产品工程师是我们的客户群。

我正在努力改进的一件事是与管理者建立更多的联系。长期以来，我在个人内部拥有非常好的网络

贡献者轨道，并且在过去的几个月中，我一直在努力扩大我的网络范围，以包括更多的工程经理。很多时候，我所做的工作需要“没有权威的影响力”，我不是自己做决定，而是试图影响他人的决定，很多时候，管理者是最终做决定的人。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

我非常幸运能在Etsy与Lara Hogan合作几年，她谈了很多关于赞助的事情，作为一名女性，我在技术领域受益于赞助，并且看到了赞助的价值。我肯定投入了大量的时间和精力。

大约一年半前，我的同事Andy Yaco-Mink，另一位技术专家，和我注意到，产品团队之间没有真正好的沟通方法来分享他们正在做的事情，或者与从事产品基础设施的团队联系。为了尝试解决这个问题，我们提出并开始了一个每月一次的会议，我们称之为产品工程会议。这是一个开放的论坛，人们可以在这里提出问题，分享他们的工作，庆祝胜利，以及我们基础设施人员分享我们正在做的事情。

我认为我们没有完全预料到的是，这也是创造赞助机会的好方法。每个月，Andy和我都要弄清楚人们正在做什么，这些事情值得更广泛地分享。哪些实验已经运行并获得了有趣的结果？谁在那里做着应该分享的酷事？然后我们会联系这些团队的工程师，说：“你应该来参加会议，谈谈你一直在做的事情！”这真的很简单。只有五分钟。非常随意，但这是获得公开演讲经验的好方法。

从那以后，我们有几个人来参加会议并发言，然后继续在公司全体员工会议或当地聚会上发言。至少有一个人最终在一次大型会议上发表了他们演讲的扩展版本。我们也听到有人说，在会议上发表演讲是他们用作晋升材料中领导力证据的东西，这

这是一种令人惊叹的感觉！

##### 你最初在你现在的公司获得了 Staff engineer（高级工程师）的头衔。晋升为 Staff 的过程是怎样的？

我被聘为高级工程师，因为当时我们不直接招聘 Staff 职位，尽管我们后来改变了这一政策。在加入 Etsy 之前，我已经在行业工作了近十年，但主要是在规模较小、名气较小的公司。在加入 Etsy 之前，我已经担任前端技术负责人五年多了。因此，我非常适应作为导师和领导者的角色。我已经花了很多时间与管理层、产品和设计部门密切合作，并制定路线图和执行计划。总而言之，我觉得我已经完全掌握了技术负责人的角色。

但是，当我来到 Etsy 时，其规模比我之前见过的要大得多。工程部门比我之前工作过的任何工程部门都要大几个数量级。我有很多东西要学习，关于如何在真正大的规模下运作，以及这与你在小公司时有多么不同。我学会了更加注意查看数据：我必须自己去学习基础统计学，才能理解实验框架。

但从一开始，我就一直在寻找我们可以改进的地方。我进来后说：“嘿，我们没有做这件事。我们应该做这件事。”例如，我注意到大家一直在以任何旧的方式编写设计系统的 JavaScript 组件，所以我说：“让我们为此制定一个框架和一个标准样板。”这是一件很小的事情，对我来说也很明显，但它大大改进了我们的实践。我认为，让一个人晋升为 Staff 的关键在于注意到问题并主动采取行动解决问题，而不是放任不管。

总而言之，在晋升为 Staff 之前，我在 Etsy 工作了不到两年。我当时的经理是全新的，不了解我的过往记录，所以我们非常密切和协作地一起整理我的材料。我听说过经理主导和 IC（个人贡献者）主导这两种极端情况的经历，我很高兴我最终成为了晋升过程的重要组成部分。尤其是在远程办公的情况下，我认为除非你积极主动，否则你的很多工作可能会被忽视，因为它发生在 Slack 上、在 pull request 中或

文档中，而不是在经理们倾向于运作的公开场合。你永远是你自己最好的倡导者，但作为远程办公者更是如此。你必须付出很多努力来确保你的成就广为人知。

##### 您认为达到资深工程师级别最重要的两三个因素是什么？您加入的公司、您的地理位置或您的教育背景对您的职业道路有何影响？

我已经讨论过这方面的一些事情：创造力、积极性、同理心等等。我还没有谈论足够的是沟通和透明度。晋升为 Staff 的一个重要部分是确保你的工作是可见的，人们知道你的名字，并且你有一个良好的声誉。

我很幸运能在一个构建前端基础设施的团队中工作，因为我们自然会向工程部门的每个人发送大量关于我们所做工作的电子邮件，所以我们获得了很高的知名度。但基础设施中更重要的部分是客户服务——帮助那些在你的 Slack 频道中提出问题或需要解决问题的人。在回到学校完成计算机科学学位之前，我曾在服务行业工作过几年，我总是试图在我与同事的每一次互动中，借鉴我从那段经历中学到的关于客户服务的经验：保持可用性，保持谦逊，并专注于真正倾听和理解人们的需求。当你真正关心帮助我们的同事时，这一点就会显现出来。

**你认为有些公司特别擅长培养 Staff engineer 吗？**

说实话，我不太了解 Etsy 以外的公司 Staff engineer 的运作方式，所以我完全有偏见！我认为 Etsy 擅长培养 Staff Engineer，因为我们拥有强大的内部文化，重视技术卓越，同时拥有无责文化和在更大的世界中做好事的愿望。我认为这导致了真正聪明和善良的人在 Etsy 工作，而智慧和谦逊的结合使人们成为伟大的 Staff engineer。这种环境会自我滋养，创造出好的榜样和想要效仿这些榜样以获得晋升的人。所以我认为总的来说，我们拥有一批很棒的人

在Etsy工作或曾经工作过，并为Staff Engineers树立良好的实践榜样。

我认为重要的是要记住，有很多规模较小、不太知名的公司也有很棒的人才在做Staff engineering类型的工作，但他们不被称为Staff Engineers，而是以其他方式被认为是技术方面的领导者。在许多公司，强大的技术负责人会成为管理者，甚至可能没有Staff Engineer这样的职位概念。在大公司里，人们很容易把Staff这个头衔当成最终目标，但请记住，职业发展的方式有很多种。

##### 你还记得在晋升到Staff职位时，对你特别有帮助的建议吗？

有人给我的最好的建议之一，也是我确保传递给其他Staff Engineers的建议是，人们有一种误解，认为你一旦成为Staff Engineer，就能掌控自己的工作，每个人都会听你的，按照你想要的方式去做。但事实恰恰相反！你为了获得晋升而努力了很久，然后你成为了Staff Engineer，突然间，一切都变得模糊不清。你从解决相对明确的问题，转变为负责找到正确的问题，然后想办法说服人们解决这些问题很重要。你将面临与你迄今为止的职业生涯中完全不同的挑战。

##### 你对追求Staff职位的人有什么建议？

在我获得晋升之前，我曾两次被提名晋升为Staff Engineer；第三次才成功。我认为最终促成我晋升的原因是我在我的主管那里有一个很好的支持者。所以我的建议是，确保你发展你的关系网，并开始与你的主管或副总裁会面，因为这些人是在会议室里决定你是否获得晋升的人。不是你的同事，也不是你的经理，而是另一群人，所以你希望他们知道你的名字和你的工作。在晋升讨论中，你希望他们会想，“哦，她发了那封关于这个项目的全公司范围的工程邮件。”或者“我经常在Slack上看到他

回答人们的问题。”或者“他们在那次会议上发言了，不是吗？”

当人们，特别是女性和非二元性别者，向我寻求建议时，我认为他们期望我谈论如何成长为一名技术领导者，但当我说“你可能已经具备了技术能力，你需要做的是提高你在公司的声誉。”时，他们会感到惊讶。不管好坏，没有良好的声誉，你无法成为Staff。我认为人们希望或期望这是一个精英体制，但事实并非如此。有很多因素会影响到获得Staff级别的职位。

##### 对于应对更高级职位带来的不确定性和模糊性有什么建议？

重要的是培养大量的自我认知，以了解你追求某件事是因为你想要，而不是因为它对组织有益。这可能很难做到。你必须准备好放弃你珍视的东西，做出改变，尝试新的事物。如果你采取的方法不起作用，不要强行推进。

我也非常喜欢[Dan Na 关于克服摩擦的演讲](https://blog.danielna.com/talks/pushing-through-friction/)，因为当你成长为一名技术领导者时，你会不断地体验到这一点。我经常思考“无权影响力”这个概念，因为当你是一名Staff Engineer时，你的工作是弄清楚团队或组织需要做什么，使组织围绕这个目标保持一致，并弄清楚如何在没有人员配置或最终决策权的情况下让人们做到这一点。这需要大量的毅力，你必须运用大量的所谓的非技术技能来推动事情向前发展。

#### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我已经说了很多名字，特别是Lara Hogan，Dan Na。我非常喜欢Julia Evans所做的一切，我很幸运能和她合作一个项目。Ryn Daniels曾经在Etsy工作，她在职业发展方面写了很多博客。Tanya Reilly对我来说是一个很大的鼓舞，她也是一个厉害的职场妈妈，同时也是一位受人尊敬的技术领导者。在前端领域，Nicole[劳拉](https://larahogan.me/)[丹·娜](https://blog.danielna.com/)[朱莉娅·埃文斯](https://jvns.ca/)[林](https://www.ryn.works/)[坦尼娅·赖利](https://noidea.dog/)[妮可](http://www.stubbornella.org/content/)

沙利文、珍·西蒙斯和伊森·马科特对我来说都是巨大的灵感来源，仅举几个例子。我真的很喜欢读卡米尔·富尼耶的《经理之路》。我从来没有走过管理路线，所以这有点像一个黑匣子，任何能让你深入了解管理世界的知识都是有帮助的，因为作为一名高级工程师，你几乎就像一个没有人员方面的经理。[珍](https://jensimmons.com/)[伊桑](https://ethanmarcotte.com/)[这个](https://www.amazon.com/dp/B06XP3GJ7F/)

## Ritu Vincent - Dropbox 的高级工程师

*这个故事录制于 2020 年 3 月。在她的 Linkedin 上了解更多关于 Ritu 的信息。*[*领英*](https://www.linkedin.com/in/rituvincent/)

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是 Dropbox 的一名高级工程师。实际上，我曾经是 Dropbox 的一名高级工程师，后来离开加入了一家不同的初创公司，然后最近几个月又回到了 Dropbox。我回来是因为 Dropbox 内部出现了一个非常有趣的机会，可以启动一个内部孵化器。我们正在努力促进公司内部的创新。Dropbox 在文件同步领域已经成为一个强大的品牌，但现在开始出现很多竞争，所以我们需要做更多的事情，并扩展到新的产品。孵化器直接与 CEO 合作，是一个非常小的团队。

我在 Dropbox 工作了足够长的时间，与这里的很多人建立了非常紧密的联系，所以当人们向我推荐这个职位时，听起来真的很有趣。我也做了几年的经理，有点想再次编码。把这些结合起来，我就回到了 Dropbox。

孵化器有两个部分。

第一部分是一个更经典的孵化器，公司各地的工程师都可以提出想法，并获得资金加入该计划，并尝试展示产品与市场的契合度或其他形式的进展，以便每隔几个月继续获得资金。目标是让成功的项目毕业到他们自己的业务线中，尽管我们还处于早期阶段。

第二部分是永久成为孵化器一部分的工程师，他们总是从孵化器内部产生想法，并以很大的自主性运作。我是那个永久“侦察”团队中的两名工程师之一，我们计划在明年扩大规模。这与我以前做过的任何事情都非常不同，这就是为什么我想报名参加。这对我来说是一个巨大的范式转变。老实说，最初的几个月真的是

既有趣又令人沮丧，因为当你的主要目标是快速尝试大量的新想法时，很难衡量明显的成果，其中许多想法都不会有任何结果。我不得不学会以更长的时间尺度来思考影响——不是从我今天交付的东西来看，而是从我未来可能影响公司交付的东西来看，

##### 在你的公司里，Staff-plus 工程师是做什么的？

我想说，Dropbox的Staff Engineer有两种不同的类型。一种是技术负责人，他们负责大量的协调工作，为团队设计工作，并花时间推动项目。另一种则更像是一位专家。

当我最初成为Staff时，技术负责人绝对是我的角色，当时我带领一个由大约八名工程师组成的团队，推动了一个为期十八个月的项目。这个项目有很多依赖关系，很多棘手的部分。我必须控制围绕项目的沟通，并弄清楚如何将项目的各个部分分配给团队，既能帮助他们成长，又能完成项目。

[这位专家在特定领域具有很深的专业知识，例如吉多·范 罗苏姆,](https://en.wikipedia.org/wiki/Guido_van_Rossum)

#### 专家是否主要来自外部招聘？

有一些专家来自行业，比如Guido和ML团队中许多非常有经验的人，但很多专家最终都是自己培养的。这可能与Dropbox相对较晚推出职称有关，这让人们有更多的时间来深入了解我们的技术。

#### 您每天如何安排时间？

在我目前在孵化器中的角色中，我整天都在做原型设计，但在我之前的技术负责人角色中，我做了很多不同的事情。

我在写代码，但写得不多，可能只占我时间的20%。我是

桌面客户端领域的技术负责人，花了很多时间协调和提供项目指导。我还花了很多时间与招聘部门合作，这是我感兴趣的事情，而不是必需的。

例如，我参与设计了专门的面试流程，主持了汇报和候选人筛选。我还做了很多关于多元化倡议的工作。这就是我在职业生涯中多次尝试工程管理的原因之一，因为我喜欢参与组织发展。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

我真正自豪的一件事是参与了我们工程级别的重大改革。早在2017年，我是少数被选出来参与工程师级别更新的个人贡献者之一，其他大多数人都是总监或经理。我感到自豪，因为新的晋升阶梯影响了Dropbox中从事工程、产品和设计的每个人。

深入思考公司发展如何改变角色和职责，这真的非常有趣。我们开始引入来自各种不同背景的人，并且我们希望能够以一种健康的方式奖励每个人。这与我日常的职责非常不同，并且在很大程度上将我推出了舒适区。

我也为我的员工项目感到自豪，这个项目在技术上非常复杂。那个项目也给了我一个机会去帮助团队中的很多人成长。多年以后，曾经离开公司的工程师会给我发邮件，告诉我他们现在变得多么自信，或者他们从那个项目中学习到了多少。

也是在那个项目中，我的经理帮助我理解到，我作为技术主管的最初想法是行不通的。最初我是这样想的：“我会把它分成二十个部分，分配出十八个部分，然后把最难的两个部分留给自己”，而我的经理推动我把那些困难的部分委派给团队，让他们得到锻炼和发展。

##### 你是否花时间倡导技术、实践、流程或架构变更？

作为一名技术主管，我花了很多时间来倡导变革。我会参与很多不同的架构和技术讨论，即使是在那些不直接属于我的专业领域的讨论中，因为人们似乎信任我的直觉。我知道有很多工程师拥有惊人的技术直觉，但没有技术专家（Staff Engineer）的头衔，但这个头衔确实使拥有这种直觉变得正式。

我确实更倾向于让将要负责一个项目的团队来做出最终决定。如果我脑海中有一个非常明确的“正确决定”，我会尝试引导团队朝着那个决定前进，而不是直接进去说“这是正确的决定”。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

我绝对认为自己是一个赞助者。执行是我工作中最有价值的部分之一——我喜欢创造东西——但我一直喜欢帮助人们成长。当我看到我非正式指导或在一个项目中帮助过的人去做一些伟大的事情时，我感到非常自豪。

作为一名技术专家，尤其是一名女性技术专家，我觉得很多人都在仰望我。在管理职业道路上似乎有更多的榜样，所以我试图把成为榜样作为我的责任的一部分，而不是仅仅埋头写代码。我的意思是，我可以仅仅埋头写代码，那也很棒，但我想要帮助其他人，尤其是那些患有冒名顶替综合症的人。

我经常遇到有人来找我说：“我不知道如何迈出下一步。”或者“我不知道如何成为一名技术专家，所以我打算去做一名经理。”我想尽力帮助他们找到自己的道路。作为一名技术专家，我认为能够被看到，并且能够让人们来问这些问题，是这个角色中很重要的一部分。

##### 作为一名高级工程师，你做过哪些在你职业生涯早期无法做到或不会做的事情？

没有什么是没有头衔就做不到的，但这个头衔确实给了我

自信。除了头衔之外，另一件让我感到自信的事情是意识到其他人也在与冒名顶替综合症作斗争。后者是我在与一位我认为是我共事过的最自信的工程师的一次关键对话中学到的，当我和他谈论这件事时，他说：“我质疑我做的每一件事。我回家后会为我那天早些时候说过的话而苦恼，以及它是否很愚蠢。”

正是那次谈话和职称的结合，促使我相信自己有能力胜任高级工程师。它们共同给了我信心去要求更难的项目，或者要求我的经理给我更多项目去做。

##### 你在 Dropbox 晋升为高级工程师的过程是怎样的？

在我加入 Dropbox 后不久，他们推出了外部职称。在第一个职称评审季，他们只授予了极少数工程师高级职称。那时他们实际上还在调整职称。在第二个评审季，我获得了高级职称。

到了第二个评审季，我已经担任技术主管一段时间了，我的经理和我都认为我显然已经在高级工程师级别上工作了。在评审周期之前，我们确实回顾了新的职业级别定义，以找出任何差距，但总的来说，过程非常顺利。

##### 你认为达到高级工程师级别最重要的两三个因素是什么？

对我来说，一个重要因素绝对是知名度。部分原因是我做了很多超出正常工程职责范围的事情。

例如，我曾在一个夏天帮助招聘部门运营实习生项目。在项目期间，我与不同团队的大量实习生导师合作，由于 Dropbox 往往有非常庞大的实习生班级，这最终意味着我在几乎整个公司都获得了知名度。招聘工作也很有帮助。如果你每个月主持几十次招聘总结会议，并推动招聘和评估对话，那么你将有机会与工程部门的每个人互动。我还帮助进行入职培训，向新员工进行核心工程演示。

有一个赞助人也绝对重要。我和我的经理关系非常好，我和我的越级经理关系也很好。我认为这也在其中发挥了重要作用。

##### 这项工作，在一些公司会被称为“粘合剂工作”，是否直接受到重视？

这项工作在 Dropbox 受到领导层的高度重视。领导层和许多非常资深的工程师都积极参与这些工作，尤其是招聘，这不被认为是粘合剂工作。话虽如此，单凭这一点并不能让我晋升为高级工程师。这是一个在文化影响力和展示技术实力之间找到良好平衡的问题。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

虽然没有明确的期望，也没有在任何地方列为正式要求，但人们都明白，你需要完成一个员工项目才能获得晋升。我想不出有哪个员工的晋升没有包含一个非常强大的项目，通常是一个多人项目，工程师是技术负责人。

我确实有一个员工项目。 早些时候，Dropbox 最初是一款消费者产品，人们下载并安装在他们的机器上。 当我们推出 Dropbox for Business 时，有人要求您的个人和工作 Dropbox 帐户同时工作，包括能够在它们之间切换而无需注销并重新登录。

最初的实现是在巨大的时间压力下编写的，它运行了多个 Dropbox 进程。 一个用于您的个人帐户，另一个用于商业帐户。 我的员工项目是让单个 Dropbox 进程可以在多个用户登录的情况下运行。 最困难的部分是该项目从内核一直延伸到用户界面。 我必须了解 Dropbox 系统的每一层。

最初我们认为这需要六个月的时间，但最终花了十八个月。 这在很长一段时间内占据了桌面客户端团队的大部分资源。

##### 您是否愿意分享一条对您达到 Staff 级别特别有帮助的建议？

在我的职业生涯早期，我的本能是要求那些我觉得自己能够很好地执行的项目，而不是那些更模糊、会促使我成长的项目。 我得到的建议是，让自己走出舒适区，并要求团队中困难的项目。 要达到 Staff Engineer 级别，你必须知道和做到比你目前所知道的更多。 重要的是，始终超越你正在做的事情，不要害怕要求你认为对你来说太难的事情。

这与冒名顶替综合症有关，你可能不想尝试任何事情，除非你绝对确定自己会擅长它。 但你必须适应你可能会崩溃和失败的事实。 没关系，你必须尝试。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

人们经常来问我，“我下一步应该做什么才能达到 Staff 职位？”我告诉他们的一件事是，要非常公开和诚实地与你的经理谈论你对职业的期望。我早期在与经理一对一交流时犯的一个错误是，我告诉我的经理我认为他们想听什么，而不是我实际的感受。

他们会问我对某项工作是否感兴趣，我会想他们为什么要问，他们*想*让我接受这项工作吗？ 所以我会说我感兴趣，即使我并不感兴趣。 或者他们会问我一个项目进展如何，可能进展得很糟糕，但我会告诉他们一切都很好，以避免让他们失望，而不是说我需要帮助。

在某个时候，我意识到你的经理实际上是在你的团队里。 他们正在寻找一种方法让你成长、提高效率、快乐并成为你能成为的最好的工程师。 与你的经理建立有效关系的方式，包括让他们赞助你，就是对他们超级诚实和开放。

当我成为一名经理时，这一点对我来说变得特别明显，

因为我*希望*我团队中的每个人都成为 Staff Engineer 并获得晋升。 我想找到晋升他们的理由，并与他们一起努力。

***你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？***

[摆锤](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)我确实会进行一定程度的钟摆式发展，因为我对职业阶梯两端的很多事情都感兴趣。我对培养人才感兴趣，我真的很喜欢与招聘部门合作，我是那种真正喜欢面试的工程师，我喜欢了解团队是如何成长的。但我也真的很喜欢编写代码，在管理一段时间后，我想回到代码中，稍微研究一下。

一旦我开始指导和管理，我发现自己对职业发展的思考方式完全不同了。钟摆效应帮助我看到了很多不同的视角。作为一名经理，你对诸如员工人数和绩效评估等事情负有非常明确的责任。高级工程师的职责非常模糊，并且因公司而异。围绕高级职位的这种模糊性导致许多人横向转到管理岗位，而他们本来可以更快乐地留在工程师岗位上。这就是为什么向人们提供更多关于高级职位的信息非常有价值。

#### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我读了很多书，但我的阅读非常休闲。对我影响最大的是有很多我认为是导师的人，通常是朋友、前任经理和与我共事过的人。我经常与过去与我共事过、了解我并且我信任的人一起吃午餐、喝咖啡和共进晚餐。正是这些关于职业挑战和成长的对话让我达到了现在的职业高度。

## 里克·布恩 - Uber基础设施副总裁的战略顾问

*这个故事录制于2020年4月。在*[*领英*](https://www.linkedin.com/in/kineticrick/)*上了解更多关于里克的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是Uber基础设施副总裁的战略顾问，这意味着我是基础设施领导团队的一员，与工程总监和整个组织的项目经理一起。Uber的基础设施工程大约有700人，分布在六个子组织中，例如处理我们数据中心和服务器的Metal、Storage、Developer Platform等等。我与副总裁一起处理诸如技术战略、文化战略和特殊项目之类的事情。

战略顾问是一个范围广泛的角色，例如，我可能会从事以下工作：

评估我们在未来两年的技术需求

帮助确定未来六个月路线图中创新的优先级

深入研究没有明确所有者的重要领域，并帮助简化正在进行的相关项目

了解工程师在大型组织变革之前或之后的感受

与需要就某事达成一致但意见相差甚远，并且似乎存在沟通问题的两个团队交谈，弄清楚如何帮助他们找到有效的前进道路

这只是一个非常非常广泛的角色，融合了工程、文化、心理学、组织设计和战略。我用两种方式来描述它，都来自流行文化。第一个就像是《权力的游戏》中的国王之手，这是我能想到的最好的类比。第二个是《白宫风云》中的利奥 麦加里，他总是说：“我为总统服务。”在我的角色中，我说我为基础设施副总裁服务。[汉德](https://awoiaf.westeros.org/index.php/Hand_of_the_King)[*权力的游戏*这就是我能想到的最好的类比。第二个是里奥 麦加里 来自*白宫风云*，他总是说：“我服务于](https://westwing.fandom.com/wiki/Leo_McGarry)

虽然现在只有我一个人，但之前在基础设施副总裁战略顾问这个职位上有两个人，我们会根据我们对项目的自然亲和力来分配工作。她通常更关注与经理和领导力相关的项目，而我更关注IC和工程项目——尽管我们仍然设法在两个领域都做事情

[战略顾问的角色有点非同寻常；它是由马修 门格林克 在他担任基础设施副总裁职位后不久创建的。对我的](https://eng.uber.com/core-infra-2018/)

在Uber的基础设施组织中，这是一个特别有价值的角色，因为它是一个非常非常广泛的组织，我帮助充当了对所有这些的综合看法。

#### 这个角色与TPM角色相比如何？

这是一个有趣的问题，因为前几天我一直在思考幕僚长和我自己的角色之间的区别。在基础设施领导团队中，我们有战略顾问和项目经理，过去，我们也曾有人担任幕僚长。

在我看来，项目经理是一个组织范围内的运营角色。他们在高层工作，确保Infra内的主要项目和领域正在沿着常规节奏推进和评估，运营工作+倡议等。幕僚长的角色是确保整个领导机器运转良好——所有参与运行和领导Infra的人员、团体、信息传递等都在有效运作。

我的战略顾问角色更多的是获取广泛的领域知识，包括技术和文化，深入了解个人和组织层面的问题，然后融入工程敏锐度。由此，我将综合出一套建议或见解，并将其传递给组织领导或整个领导团队。日常工作中，绝大多数

我的工作是直接与组织主管和项目经理一起完成的——向组织主管提供建议，然后，在他的投入和批准下，与项目经理合作将它们变成现实。

##### 您如何看待与您的赞助人保持一致的重要性？

这很有趣，因为这种一致性是这个角色的关键——几乎是必需的。马修和我在原则、价值观、世界观、对情商的重视、执行方法和理念上非常一致。在很多方面我们都是一致的，以至于这几乎是一种共生关系。

与赞助人保持一致对于发挥作用至关重要，但这不仅仅是战略顾问和副总裁之间不带感情的联系。这也关乎里克和马修作为个体之间的联系，并确保这种联系是合适的。

在我的职位上，我们经常连续几周不在同一个房间里，但我仍然必须像他的直接代理人一样行事。所以我走进一个房间，心想：“马修会怎么做？他想问什么问题？他对这个问题有什么指导？”因为我不能总是跑回去向他澄清，所以必须对他世界观有深刻的理解。这对于我保持作为他的代表所需的非常深的信任，并有效地执行他的战略和愿景至关重要。人们需要确信，如果马修在那里，我总是会给出与他相同的答案。

这也意味着我必须真正理解他的目标、意图、价值观和原则，以确保我准备好用我的声誉和信誉来推动它们前进。通常，我的部分职责包括向工程师倡导或翻译他的愿景和/或实施方案，有时在补充背景信息并不总是已知的情况下。当我这样做时，我必须确保我不仅理解他所做事情的逻辑和价值，而且我自己也相信它——否则，倡导就会变得困难，更不用说虚伪了。

这是我刚开始担任这个角色时非常挣扎的事情。马修

会 постоянно告诉我，“你是我的代表；你应该随意使用我的名字和角色来推动和执行事情。”这对我来说很困难，因为我以前从未担任过这样的角色。以前我总是用自己的名字和声誉来工作，而现在我是在副总裁的庇护下工作，以及随之而来的一切。随着时间的推移，我已经学会了如何谨慎地使用这把锤子，因为你不想过度使用它。

我还了解到，我有时必须让人们知道我戴的是哪顶帽子。我喜欢指导别人，但有时人们不确定他们得到的是为组织和公司利益工作的战略顾问，还是为个人及其职业生涯利益工作的导师；我尽量让他们知道我在特定对话中扮演的角色。如果我与我指导的人会面，他们可能想得到关于更换团队，甚至离开组织或公司的建议，他们想知道我给出建议的视角是什么。

##### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

我认为最大的区别在于，其他高级工程师主要从事技术工作。他们是领导者，所以他们确实会进入情商、沟通、协作、冲突解决、宣传等领域，但仍然有 80% 的日常工作是由技术问题驱动的。

而对于我来说，可能会有几周我专注于一个关于群体心理学或组织设计的项目。技术问题并不总是驱动我日常工作的纯粹焦点——尽管它们总是存在，即使只是在背景中。

##### 既然你开发的少了，你现在如何了解实际情况？

当我还是工程师的时候，我可以被动地做到这一点，因为你身处代码之中，试图提交代码，处理配置和操作服务的摩擦等等。但这种方法现在行不通了，因为我不太接触代码了；所以现在，获取这些数据和意识需要一个主动的过程。

我做的一件事是继续坐在我以前的团队旁边，这样我就可以听到他们的工作。也许他们会抱怨某个服务的稳定性，或者我们的工具存在差距，听到这些对我有帮助。

我还经常向人们询问他们的开发者体验。我脑海中记着一些善于发现问题并对方法提出反馈的人，我经常与他们联系。有时这些联系更结构化，实际上是征求意见的调查，有时只是一个快速的消息来问候一下。

我还告诉人们给我发送非关键路径且没有严格时间限制的工作，我尝试利用这个机会来保持编写实际代码的新鲜感。但我必须小心不要进入我们实际产品的关键路径，因为我知道我没有足够的带宽来维护未来的代码。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

这个特定角色的特别之处在于，它本质上是与副总裁的内置指导。当我刚开始时，他问我：“你五年后想做什么？你的目标是什么？”当时，我真的没有明确的答案。长期以来，我的观点是，在当今时代，能够编写代码让你在就业保障和发展轨迹方面处于人类历史上最好的位置之一，这对我来说似乎就足够了。

随着我花时间思考我的目标，我真正得出结论的是，我喜欢成为其他工程师，特别是其他少数族裔工程师的可见榜样，并帮助优步或更早职业生涯中的人们。我特别喜欢帮助那些刚入行，可能仍然有点害怕这个行业的人。这是驱动我的一个重要部分，这个角色帮助我意识到并承认了这一点。以前我不认为这是一个有效的目的，但我意识到，如果这是你所爱，如果你对此充满热情，那么你必须去追求它。

对我来说，指导的另一个重要原因是，在我的一生和

职业生涯中，我有六个人被我视为关键导师。他们每个人在不同的时期都对我的生活产生了巨大的影响和影响——如果没有他们过去和持续的指导，我就不会成为现在的我。我非常感谢他们，并且始终意识到他们对我的指导有多大。所以，我总是认识到导师的力量，并希望确保我可以为他人提供这种力量。有时，导师甚至不知道他们的言语或行为如何改变你，他们可能产生的连锁反应，甚至在几年后。所以，我总是尽量让自己可以作为导师帮助他人，因为你永远不知道什么时候你可以对某人产生那种改变人生的影响，或者如何产生这种影响。这可能只是正确的词语，正确的视角，你在正确的时间给他们的正确的推动。

我总是告诉人们，“说真的，如果你需要我，就来寻求帮助。”这是我所做的最令人兴奋的事情之一，我尝试通过几种不同的方式让自己可以提供帮助。

其中一种方式是，我每个月都会为工程部门的大部分新员工开设一门 Engucation（将工程和教育融合为一个词时会发生什么）课程。这门课程叫做“经验+问题”，实际上只是一个他们可以随意向我提问关于优步的任何问题的地方——技术、文化，任何问题——我会尽可能坦诚地回答。最后，我会告诉他们我的电子邮件，并欢迎他们与我联系。之后，很多人会联系我，我会给他们关于职业、在优步工作或任何其他方面的建议。其他时候，我会在办公室里遇到一些人，他们会向我寻求建议。

我希望作为一名黑人工程师被大家看到，向其他人展示我们*是*存在的，并且这是*可以*做到的。一旦我意识到这是我一个重要的动力，我就知道我必须提高公开演讲的能力，因为这是我作为榜样扩大自身影响力的一个重要途径。公开演讲曾经让我感到恐惧。我过去*讨厌*公开演讲。但因为它是一种接触大量人群的关键方式，我告诉自己我必须学会喜欢它，从那以后，我学会了成为一名有效的公开演讲者，并且实际上爱上了它。现在它是我做的最令人兴奋的事情之一——就像过山车一样；每次我做的时候，我都会感到紧张，但这是一种令人兴奋、有趣的紧张，而且我在做的时候会感到巨大的刺激。

##### 你是否考虑过建立你的外部品牌？

我有几个朋友花时间建立他们的外部品牌，其中一个现在又开始重新投入。他意识到他在优步（Uber）的工作非常紧张，以至于他把外部工作推到了一边。

我在这方面比较被动。当我参与一些被公开报道的事情或者我做一个公开演讲时，我会在领英（LinkedIn）上发布一个链接，但我根本不写自己的内容。我考虑过这样做，而且我对这样做很感兴趣，但我没有。我倾向于通过说话来思考，所以以这种方式写作需要大量的准备来组织我的想法，到目前为止，我还没有花太多时间在外部做这件事。

##### 你最初在你现在的公司获得了战略顾问的头衔。你是作为战略顾问被雇佣的吗？如果不是，你晋升到这个职位的过程是怎样的？

[罗布 庞库努斯 担任相同的角色，当他决定离开时，他被](https://www.linkedin.com/in/rob-punkunus-3791273/)

在那之前，马修（Matthew）和我已经有过几次积极的互动，我们开始意识到我们有相似的观点和价值观。例如，有一次，我们收到了一大堆匿名提交给我们的问答会议的恶意评论，看到我们的文化朝着这个方向发展，我真的很困扰。我在一次问答环节中站起来发言，要求大家找到一种更具建设性的方式来表达他们的担忧，我认为这引起了马修（Matthew）的共鸣。

当他第一次建议我担任这个角色时，我有很多冒名顶替综合症。我试图让他撤回这个提议，认为这不会是一个好的选择，但最终我还是接受了，并且从那时起一直担任这个角色。

#### 你成为战略顾问最重要的两三个因素是什么？你加入的公司、你的地理位置或你的教育

##### 对你的职业道路有什么影响？

[苏珊·福勒的博客文章](https://www.susanjfowler.com/blog/2017/2/19/reflecting-on-one-very-strange-year-at-uber)除了罗布的建议之外，最重要的因素是做与马修的价值观相符的可见工作。我参与的一个项目是加入工作组，以了解和改善2017年SRE的文化。工作组早在苏珊·福勒的博客文章发布之前就已经计划好了，而且我们的第一次会议恰好在她发布后的三天。我真的认为文化工作组做了一些伟大的工作，我和其他组成员都为此感到非常自豪，在超过18个月的时间里，我们以一种有意义的方式真正改变了一个百人组织的文化。

此外，我一直对文化和人类心理学+行为领域的事物个人非常着迷。在我的职业生涯中，在我工作过的公司里，文化+群体心理学通常是将组织从优秀变为卓越的隐藏因素。我已经通过阅读行为经济学、行为科学等方面的书籍和论文来满足我个人在这方面的求知欲，因此这种自然的兴趣帮助我逐渐走向了现在的方向。

#### 你还记得在晋升到Staff职位时，对你特别有帮助的建议吗？

在我的整个职业生涯中，人们总是告诉我，我的影响力比我意识到的要大得多，潜力也更大。我从来没有听过这些话，对我来说，以及对许多其他工程师，特别是少数族裔工程师来说，我们花了很多时间怀疑自己。我们很容易只看到不好的一面。我们可能没有意识到，当我们在会议上充满激情地谈论某件事时，人们真的在倾听我们。有人一直告诉我，我没有意识到我所产生的影响，我的观点不仅有效，而且实际上对组织有影响力，这真的很有帮助。

另一个有帮助的事情是有导师。特别是，我喜欢那些具有建设性对抗性的导师。我的意思是，他们把我扔进那些让我非常害怕的事情中，但他们确信我已经准备好了。他们帮助我超越了我认为对我来说可能的范围。这些人通常是我一起工作过的经理，但我们能够互相

互相学习。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

这可以追溯到我如何根据对组织心理学、文化、指导等方面以及技术的广泛兴趣而走到今天的位置。我从来都不是一个纯粹的工程师，只是整天沉浸在代码中。我从来都不是那样的人，我不得不接受这一点。

对我来说，重要的是追随我的激情。最近，这与指导有关，但也与机器学习等其他事情有关，这一直是我的爱好。我喜欢机器如何产生模仿人们思维方式的见解——这是我对技术

+心理学的兴趣的完美结合。

所以我怀揣着这些激情，然后当出现将这些激情与公司需要的某些东西结合起来的机会时，我会抓住它们。例如，我在Uber的前一个团队正在生成关于车队利用率的见解，用于容量规划，这是一个很好的机会，可以将我对机器学习和站点可靠性的兴趣结合起来。[将我对以下方面的兴趣结合起来 机器学习和站点可靠性.](https://www.youtube.com/watch?v=9ool1BQybaE)

小公司让你有机会做许多不同的事情，但达到一定规模的公司也让你有独特的机会专注于你的激情，这对我来说使我能够在保持影响力和激情的同时，永远不必整天坐在键盘前敲代码。

#### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

这是我有时会思考的事情，即使现在我还在思考。它在我的可能性清单上，并且在我的整个职业生涯中，人们都会问：“你有没有考虑过转到管理层？”

我现在想专注于成为一名高效的、具有大局观的高层领导。最终，我想培养人员管理技能。

也许在不久的将来也会做。吸引我的是人类的行为让我非常兴奋，而人员管理是一个很好的机会来花时间在这上面。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

你知道，在我职业生涯的前三分之二时间里，我非常喜欢阅读尽可能多的技术内容。我过去常常整天在 YCombinator 或我的 RSS 订阅源上阅读关于分布式系统、可靠性等的内容。现在，我更喜欢阅读关于行为经济学、行为科学、人类心理学、组织战略等方面的书籍。在这些领域里，我非常喜欢[丹尼尔·卡尼曼](https://en.wikipedia.org/wiki/Daniel_Kahneman)、[蒂姆·哈福德](https://en.wikipedia.org/wiki/Tim_Harford)、[丹·艾瑞利](https://en.wikipedia.org/wiki/Dan_Ariely)。还有一些很棒的播客——[魔鬼经济学](https://freakonomics.com/archive/)、[选择学](https://www.schwab.com/resource-center/insights/podcast)、[隐形大脑](https://www.npr.org/podcasts/510308/hidden-brain)。

[此外，去年我开始整理一个关于人脑的书籍阅读清单 和行为 我与任何对该主题感兴趣的人分享。](https://docs.google.com/document/d/1WIqIYuSGfyoU_ZO-xZMDXfaaUmnG2tmnkHiGZQ7pvqg/edit?usp=sharing)

我仍然在 Reddit 上关注[r/linux](https://www.reddit.com/r/linux/) 和[r/programming](https://www.reddit.com/r/programming/) ，它们已经取代了我的 RSS feed，让我可以发现新的阅读内容。

## 尼尔森·埃尔哈吉 - 前 Stripe 公司职员工程师

*这个故事录制于2020年4月。想了解更多关于纳尔逊的信息，请访问他的Twitter和博客。*[*Twitter*](https://twitter.com/nelhage)[*博客*](https://blog.nelhage.com/)

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我最近在Stripe工作。他们做在线支付处理，这是一个发展非常快的初创公司，大约有两千人。工程部门大约有六百人。我离开的时候，严格来说我没有头衔。如果我再待两个月，我就会成为一名Staff Engineer，因为经过多年的内部辩论，他们终于推出了头衔。

我最近所在的团队叫做支付架构，这是一个由三到四个相当资深的工程师组成的团队。支付是Stripe产品的核心，我们负责支付代码库。我们特别关注代码库的金融基础设施层，并构建我们需要的数据模型和抽象，以支持Stripe当前和期望的所有产品线。

我们研究了代码结构如何适应组织结构，包括如何在快速增长的组织内构建代码，该组织正在增加团队、产品、国家和支付方式。我们的架构能够支持在多个办公室和时区之间分散所有权，这一点尤为重要。

我们围绕代码质量和代码架构推动了很多举措，并进行了一些实施和重写项目。对于每一项举措，我们都制定了指标和目标，让团队承担这些目标，然后为团队提供工具，帮助他们迁移到新的标准。

#### “支付架构”团队是一个永久团队，还是更像一个项目团队？

两者兼而有之。它不是一个具有狭窄项目或范围的超级战术团队

关于它的任务。但它也不太可能作为一个团队永远存在下去。我们正在采取一种实验性的方法来发展我们的架构，目标是在我们前进的过程中修改和更新我们的方法。我们希望这个团队最终会把自己从工作中解放出来。

##### 在你的公司里，Staff-plus 工程师是做什么的？

很难很有信心地说，因为Stripe才刚刚引入职称。谁是Staff Engineer并不公开，但你可以根据那些从事最重要、最有影响力的事情的人来判断谁是高级工程师。

有一些明显的Staff Engineer原型。一种是从事深入的技术项目，可能是在规划或构建新的基础设施。在支付架构团队之前，我致力于构建[冰糕](https://sorbet.org/)，这是我们的静态Ruby类型检查器。我和另外两位高级工程师花了大约一年的时间从头开始构建它，这是一个深入的、高度杠杆化的技术工作原型的很好例子。

还有一些Staff Engineer花时间处理交叉项目，充当架构师和项目经理的结合体，将组织的不同部分整合在一起，以解决一个大问题。通常，这些问题与我们当前的架构或组织结构不太一致，因此需要许多不同团队之间的协作。

还有一些Staff Engineer与一个团队或一小群团队合作，他们是团队愿景的守护者。他们会确定团队正在朝着什么方向发展，以及他们希望在一到五年内达到什么目标。他们会在整个组织中工作，以建立和分享这一愿景，然后努力实现它。

#### 您每天如何安排时间？

这在支付架构角色和Sorbet角色之间看起来非常不同。Sorbet更像是一个“埋头苦干”的项目。在支付架构方面，仍然有一些编码工作，因为我们有一种特定的方法，我们既想尝试，又想演示我们的想法

正在推动。

我也做了相当多的项目管理工作。比如管理任务跟踪器，主持每日站会，弄清楚谁需要帮助或谁被阻止了。我还花时间在公司和工程组织中充当沟通纽带，特别是与那些对我们正在构建的工具和模式感兴趣的团队交谈，并为他们提供建议。

在那项工作中，我花时间参加各种会议，研究技术战略，并且每周还花相当多的时间编写关于我们发现的问题的设计文档，并推广我们认为可以解决这些问题的架构形态。最后，我努力向领导层和其他团队解释和推销这些想法，以此来设定议程，并倡导对它们的投资和优先考虑。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

当然，最容易追踪影响的一个是[冰糕](https://sorbet.org/)，在两年内，一个三人团队将Stripe从一个动态类型代码库转变为一个基本上是静态类型代码库。这影响了公司所有六百名工程师在他们的编辑器和开发环境中的日常体验。

话虽如此，很难知道这是否真的是最具影响力的项目。有一种更模糊的观点认为，架构战略工作从长远来看将更具影响力。

#### 作为一名Staff-plus工程师，你做了哪些你在早期职位上无法或不允许做的事情？

关于“允许”的问题很有趣，可能不是完全正确的问题，因为关于谁获得什么样的角色，几乎没有官方政策。大多数事情都依赖于更非正式的资历衡量标准。

但即便如此，Sorbet和支付架构团队都是相对雄心勃勃的项目。例如，Sorbet需要将三名高级工程师从更具体的项目中抽调出来。启动它们需要高水平的

组织尊重和信任，才能获得许可和支持，将团队从他们现有的工作中抽调出来，让他们用一年的时间来从事这些项目。

##### 你是否花时间倡导技术、实践、流程或架构变更？

这在计划过程中有点季节性。优先排序最终意味着人员配置，而人员配置决策发生在计划期间。

计划季节是一个特别紧张的时期，但我或多或少一直在考虑整个工程范围内的优先排序。这可能是注意到许多工程师遇到的问题，或者看到一些减慢团队速度的事情。这是一个持续不断的线索，我一直在思考，并且会定期成为一个紧迫的优先事项，我会花时间倡导创建一个团队或解决一个问题。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

我没有花很多时间从这些角度明确地思考这个问题，我也想不出明确的例子来描述我当时在做什么。我做过的另一件相关的事情是帮助启动我没有参与的团队。例如，一些团队会成立来接管一个曾经是我能力范围内的系统，我会与他们密切合作，提供背景和建议。

***在收购 Ksplice 后，您首次在 Oracle 获得了架构师头衔。您获得架构师头衔的过程是怎样的？***

[Ksplice](https://en.wikipedia.org/wiki/Ksplice)我不记得收购前 Ksplice 是否有职位头衔。收购后，我在 Oracle 工作了一年，职位是架构师，我认为当时这是他们最高的个人贡献者级别。那里肯定存在一些收购职位头衔膨胀的情况。如果我不是通过收购加入的，我不知道我是否能达到那个职位。

#### 在 Ksplice 被 Oracle 收购并且你成为一名架构师之后，你每天所做的工作与收购前相比是否发生了变化？

##### 收购？

我大致上做着相同风格的工作。改变的是，我花了更多的时间与 Oracle 内部的 Oracle Linux 组织进行沟通。我专注于弄清楚我们的产品如何与他们的产品集成，并让他们尽快了解我们的技术，以便他们能够使用它。我之前花时间培训新员工，但那速度比在 Oracle 慢得多，在 Oracle 的情况是，“我们把你扔进这个 400 人的组织，现在培训他们是你工作的重要组成部分。”

##### 你认为达到高级工程师级别最重要的两三个因素是什么？

我所走的具体道路很大程度上取决于我早期加入 Stripe。我大概是第 30 号员工。但我所做的事情，我认为与其他人所做的事情并不完全相同，是我试图在 Stripe 建立非常广泛的背景和认知。当只有 15 名工程师时，这相对容易做到；那时没有那么多事情。

但随着公司的发展，我花了很多精力试图了解*一切*工程方面发生的事情：团队之间的互动，扩展的痛点。我试图拥有一个异常全局的视角。这帮助我知道哪些问题是重要的，特别是哪些是更高层次的重要问题。如果我知道组织的目标是推出一个特定的产品，我就会有视角看到为什么它会很困难，是因为之前的这些架构决策，或者这个下游系统目前无法胜任这项任务。

随着组织变得非常庞大，看到那些更高层次的依赖关系变得越来越困难，试图保持一个广阔的视野和系统层面的视野对此有所帮助。它还帮助我将团队联系在一起，使我成为信息和想法的路由器，以及提案的发起者。

许多团队陷入只关注他们所在的世界，而对他们的内部客户如何与他们集成缺乏了解。这种情况发生是因为他们从未在他们支持的内部客户团队工作过。我帮助团队了解其他团队如何真正使用他们的系统，并将他们与组织中其他

他们应该收集的观点，

随着组织的成长，保持所有这些背景信息是很困难的，但对于那些在公司规模较小时没有开始建立这种全局背景信息的人来说，这甚至更加困难。通过尽早开始，相对于那些稍后开始并试图逆向工程架构和组织依赖关系的人来说，你拥有巨大的竞争优势。

#### [凯维·麦克明](#_bookmark32)当我和凯维·麦克明交谈时，她提出的一个有趣的观点是，有时能够看到事物而没有完整的历史背景是有帮助的。你是否发现你的背景让你更难前进？

当然。我注意到自己在与团队进行对话时，已经准备好向他们讲述过去七年里，每次有人尝试他们正在做的事情以及为什么没有成功的故事。我需要刻意地回顾这段历史，并问自己，“这些信息对他们有什么帮助或相关性？”

有时这些信息没有用。另一方面，如果有人试图做这件事，结果却碰得头破血流，那么可能仍然存在一些非常棘手的技术问题。指出这些障碍可能有一些价值，但同样，有勇气再次尝试也很有价值，因为几年过去了，我们已经成为了一个不同的组织。

#### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

我本能地对这种员工项目的想法持谨慎态度，部分原因是，我见过的员工工程师的典型形象是那些不一定自己运行大型项目或做大事的人。但他们只是非常有效的专家和引导者，使整个工程组织运行得更好。

也许我最接近员工项目的事情是，我因为参与一个名为“数据模型 Stripe 发布计划”的项目而获得了最终晋升。我领导了这个为期六个月的计划，让一群团队协调处理一些项目，以解决我们数据模型中的弱点，并推进数据模型

以一种有抱负的、变革性的方式。

我认为这在某些方面并不是一个很好的员工项目例子。首先，我们做了很好的工作，但由于多种原因，它远没有任何人希望的那么具有变革性。其中一些原因在我控制范围内，而另一些原因是问题太难了，组织没有足够的资源在六个月内真正解决它们。

虽然那个项目不一定比我在其他方面做的工作更好，但它是一个非常引人注目、备受瞩目的角色。它创造了知名度，并以重要的方式提高了我在公司的地位。

##### 你能分享一条对你成为一名员工工程师有帮助的建议吗？

我学到的一个教训是专注和优先级的重要性。当你拥有我之前谈到的广泛的组织背景时，这一点尤其重要。在任何时候，你都很容易识别出三十件你想做的事情。

偶尔你可以把这三十件事中的每一件都向前推进一点。在一段时间内，这是有成效的，但你需要小心。如果这些事情没有得到处理，而你认为应该处理，那么你最好一次选择其中一件，真正集中精力，而不是同时在许多不同的项目上都推一点。

一个很大的区别在于是否已经有团队在处理这三十件事。如果已经有团队在处理它们，但方向不是你认为有效的方式，那么你可以通过去帮助这三十个团队，解除他们的障碍，从而获得很大的 leverage（杠杆作用）。

最终你不得不说，“我希望我能做所有这些事情，但我不会做所有这些事情。今年我将选择一两件事来做，并且我会故意忽略其他的，即使我认为它们是主要问题。”

##### 对于刚开始担任 Staff（高级工程师）的人有什么建议吗？

***工程师？***

有一件事是，我非常相信康威定律的首要地位，它能指导组织的技术架构。

另一个是建立并投资于你与工程领导层的关系：经理、主管和副总裁。我认为这可能与组织结构有关，但肯定在 Stripe，这些人通常拥有很多隐性权力，因为他们是遇到问题时显而易见可以求助的人。他们对人员配置和优先级排序也有很大的影响力。

与他们保持良好的关系非常重要，这样你就可以用你的想法影响他们，而且你也可以了解他们看到的问题。你需要知道他们的动机是什么，以及他们认为存在但你没有意识到的问题。与领导层更好地保持一致，可以使很多事情变得更容易。

对我来说，另一件非常有价值的事情是估算。我发现能够查看一个系统，并习惯于估算这个东西每秒有多少 GB，或者这些数据需要多少存储空间，真的很有价值。你不必做到完美，得到最接近的 10 的幂通常就足够有用了。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

我考虑过，但不是很认真。我对自己的了解很清楚，至少目前，我不会真正喜欢这项工作。我认为我会发现所有的互动都不是一种可持续的消磨时间的方式。我偶尔希望我对它更感兴趣，因为我确实认为这是一种获得很多权力的方式，但我很幸运地有足够的自我意识来相信，我认为是正确的，我不会喜欢它，因此也不会擅长它。

##### 你从哪些资源（书籍、博客、人物等）中学习过？

我经常被问到这个问题，因为我拥有异常广泛的通用知识，但我不知道它从何而来。我

对计算、软件和架构非常好奇。我阅读很多不同的东西，而且我花在阅读软件工程 Twitter 上的链接的时间可能不太健康。

对我来说，培养一个由其他高级工程师组成的良好个人网络也非常有价值。我会和他们非正式地聊聊我们正在做和思考的任何事情。当你有人脉关系时，你可以非常坦诚地了解人们遇到的问题和他们正在考虑的解决方案。

我主要是通过我专业上认识的人的朋友网络，或者追溯到我上学的时候，来引导这件事的。这不是我事后才去追求的。

我偶尔会阅读一些技术论文，但这不是我主动去做的事情。通常是当有人引用它，或者在其他情况下出现时才会阅读。我绝对不会刻意地系统地跟踪或回顾最近的出版物。但我确实认为，对所谓的“基础文献”有一个不错的了解是非常有用的。

## Diana Pojar - Slack的Staff数据工程师

*这个故事录制于 2020 年 4 月。在她的*[*博客*](https://diana.dev/)*、*[*Twitter*](https://twitter.com/podiana)*和*[*领英*](https://www.linkedin.com/in/dianapojar/)*上了解更多关于戴安娜的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是 Slack 的 Staff Data Engineer（资深数据工程师），也是数据平台团队的技术负责人。我于 2016 年 2 月加入 Slack，是数据工程团队的首批工程师之一。我深度参与了构建许多工具和基础设施，以使数据可用于长期分析。我加入时，团队刚刚决定使用 Thrift 作为日志记录格式。如果有人想获得见解，他们必须在生产 MySQL 数据库的只读副本上安排 cronjobs。

Slack 数据工程团队的目的是使公司中的任何人（数据科学、工程师、产品经理等）都可以访问数据，以便他们可以计算见解、推动业务决策或构建新功能。数据平台团队专注于构建大规模运行的服务和框架，以授权每个需要在数据仓库中处理或使用数据的人。我们团队拥有的一些东西包括：数据发现服务，该服务公开任务、表、列沿袭和通用元数据，事件日志记录结构以及使用事件并在数据仓库中的原始表中公开它们的管道。

#### 在 Slack，Staff-plus 工程师是做什么的？你每天如何安排你的时间？

Staff-plus 工程师的角色很大程度上取决于团队的需求以及特定工程师的优势。根据我的经验，Staff-plus 工程师的职责会随着时间的推移而变化，但通常他们的主要重点是从事对公司具有战略价值的项目/工作，同时推动技术设计并提升团队水平。

我见过 Staff-plus 工程师分为两大类：更注重深度（专家）或更注重广度（通才）。

对于第一类，更注重深度的人通常是特定领域的专家，他们的大部分时间都花在编写代码或编写技术设计文档上，以寻找其专业领域的解决方案。公司面临着独特的挑战，需要主题 matter 专家来推动这些极其困难问题的技术解决方案。例如，在 Slack，随着公司的发展以及我们的系统需要扩展和执行，有一位首席工程师，他的主要重点和热情是检测和修复性能问题。

那些专注于广度的人通常与领导团队更紧密地合作，影响组织或公司范围内的技术愿景，改进流程和文化。由于他们的广度，他们更灵活，可以根据公司的优先事项和需求在工程组织的各个领域工作。

[技术领导力](https://slack.engineering/technical-leadership-getting-started-e5161b1bf85c)就我个人而言，目前我更享受和专注于广度，以及我如何分配我的时间，这很大程度上取决于我的团队和组织的需求。我想说的是，今年到目前为止，我大约 50% 的时间花在技术领导上，以及与人们讨论我们应该关注的更大的技术投资，而 50% 的时间则专注于指导、代码审查、编写代码、处理突发事件和修复关键问题等等。这个比例每个季度都会变化。

#### 作为一名 Staff-plus 工程师，你觉得最有影响力的地方在哪里？作为一名 Staff-plus 工程师，你做了哪些你在早期职位上不会做的事情？

就我个人而言，我感觉最明显的是，在我晋升/职称变更后，那些以前没有和我一起工作过的人对我的信任和尊重有所增加。拥有这个头衔与一个人影响组织/公司路线图和优先事项的能力密切相关——基本上你可以进入“事情发生的地方”。

我可以参与构建对公司直接成功产生影响的事情。倡导这样的项目并参与其中，在早期的职位上是不可能实现的。

我也能够提升其他更初级的人，让他们发出自己的声音。拥有 Staff+ 的头衔带来了一些其他人没有的特权，我试图利用这一点来帮助提升我的团队/同事。

##### 你是否花时间倡导技术、实践、流程或架构的变革？你倡导过什么？

实际上，我花费大量时间倡导技术解决方案、流程、架构或文化变革——不仅仅是编写代码。我经常参与许多需要构建依赖于数据工程工具和服务的团队的技术设计审查过程。除了参与倡导技术项目外，我的一个重点领域是改进文化或流程变更。

我内心深处的一个领域，我相信我在我的组织中发挥了重要作用，是关于事件管理和分析。我参与了公司的弹性团队，以改进我们的事件分析流程，但对于我的数据工程组织，我非常积极地推动我们的一般随叫随到期望和结构，同时也采用了公司的事件响应结构。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

赞助实际上对我来说是一个重要的领域，因为我专注于与许多与我合作的人建立良好的关系，并且我坚信我们需要互相提升。通过我获得 Staff Engineer 的旅程，并与我自己的冒名顶替综合症作斗争，我有机会与那些赞助我并对我的成长产生巨大影响的优秀人士合作。随着时间的推移，与我合作过并一直是我的导师和榜样的人有Josh Wills、Stan Babourine、Bogdan Gaza和Travis Crawford。[乔希·威尔斯](https://www.linkedin.com/in/josh-wills-13882b/)[斯坦·巴布林](https://www.linkedin.com/in/stanb/)[博格丹·加扎](https://www.linkedin.com/in/bogdangaza/)[特拉维斯·克劳福德](https://www.linkedin.com/in/traviscrawford/)

指导和培养我周围的人对我来说一直很重要，而且身处 Staff+ 的角色，你拥有其他人没有的那种特权和权力，我尽力利用这一点来帮助和提升我周围的人。

#### 你最初在 Slack 获得了 Staff engineer（高级工程师）的职位。你是以 Staff 的身份被雇佣的吗？

##### 工程师？如果不是，晋升为 Staff 的过程是什么？

我加入 Slack 时是一名中级工程师，一年后我获得了高级工程师的晋升。作为一名高级工程师，我有机会参与多个对整个组织/公司产生影响的项目，其中许多项目直接关系到我们公司业务指标的计算方式，这对于公司准备上市至关重要。

在担任高级职位两年后，我的经理告诉我，我的工作水平已经达到了更高的层次，他认为有充分的理由为我争取晋升，并计划推荐我。在 Slack，Staff+ 工程师的晋升需要准备一份晋升材料，其中要用清晰的细节和可衡量的指标来说明一个人达到了某个水平。主要的关注领域是：技术质量、影响力、协作和执行力。我们一起合作撰写并填写了晋升材料所需的所有必要细节。作为一名个人贡献者（IC），我强烈建议，如果可能的话，与你的经理合作，共同撰写这份文件：这应该是一项团队努力。材料准备好后，晋升材料将由一个特殊的晋升委员会进行评估，该委员会由来自整个公司的一些领导和 Staff+ 工程师组成。

##### 您认为达到资深工程师级别最重要的两三个因素是什么？您加入的公司、您的地理位置或您的教育背景对您的职业道路有何影响？

当我回顾并思考我作为一名初级工程师时对此的感受和想法时，晋升为资深工程师的主要因素实际上是相信**你能做到**，并且不要让冒名顶替综合症得逞。

总的来说，我一直努力在职业选择上做到非常有目的性，而且通常每年都会花一些时间来思考我正在做的事情以及我想关注的成长领域。我发现这非常有价值，因为它让我退一步评估我目前正在做的事情，询问我是否还在当前的环境中成长，并思考新的机会。

所以在2015年底，当我决定离开Twitter时，我发现

Slack 开始组建他们的数据工程团队。能够从头开始构建和设计系统、服务和框架对我来说非常令人兴奋。加入 Slack 一个新成立的团队是一个独特的机会，这无疑有助于我晋升为 Staff Engineer（高级工程师）。它给了我机会参与对整个组织或公司产生影响的项目。例如，我参与的第一个大型项目将生产 MySQL 数据库上大约 25% 的负载转移到数据仓库，为公司节省了数百万美元。

影响我成为首席工程师的另一个关键因素是我周围的人，我很幸运我的团队中有出色的榜样和导师。当我加入 Slack 时，我是一个非常资深的团队中的第 4 个人（其他所有人都是高级员工），这促使我渴望证明自己并表明我属于这里。在每个项目中建立指导、可见性和技术质量的记录也有助于我走向首席工程师的道路，我没有把我的工作仅仅看作是一份工作，而是把大量的热情投入到我们试图解决的每个项目或问题中。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

不，我没有被分配“员工项目”，而且这也不是 Slack 晋升流程的一部分。有一个职业阶梯，描述了每个级别的总体期望和影响范围，对于 Staff+ 级别，这种范围开始从组织范围的影响扩展到公司范围的影响。

我通常总是尝试挑战自己，并且一直在寻求推动变革并影响我的组织。我认为我参与过的最具影响力的项目，也是促成我晋升为 Staff Engineer 的项目，是参与思考和实施关于我们公司业务指标（例如：年度经常性收入 ARR）如何计算的技术设计，以确保该过程可靠、可扩展，最重要的是，可重现。这是一项关键举措，因为 Slack 正在完成一家上市公司的准备过程。

##### 你还记得在晋升到 Staff 级别时，有什么建议对你特别有帮助吗？回过头来看，有没有更容易晋升到 Staff 级别的途径？

我感觉非常有帮助的一点是，要理解 Staff+ 工程师的工作和责任不仅仅是编写代码。基本上，让你达到高级工程师水平的东西并不能让你达到 Staff+ 级别。重要的是要了解公司内部以及整个行业对这个角色的期望，因为公司之间存在一些差异。

与你的经理或更资深的同事合作，寻找能够挑战你并扩大你工作范围的项目。对我来说非常有帮助的一点是，我开始投入更多精力来发展我的领导力和沟通技巧。我也开始以不同的方式来构建和思考某些事情，当我开始感到压力或不确定自己的能力时，这通常表明我正在成长，并且偶然进入了一个提供大量成长机会的领域。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

晋升到 Staff 工程师会带来很多责任，你应该始终是你同事的坚定拥护者。作为一名 IC（个人贡献者），我认为执行和亲力亲为总是“容易”的事情，而困难的事情实际上是在你的组织中推动变革和产生影响。

我认为在你作为 Staff 工程师的不同时期，你可能会发现自己专注于不同的事情，这是可以接受的，也是预料之中的。对于 Staff 工程师应该做什么，没有一个明确的定义。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

实际上，这是一个我每隔几年就会问自己的问题。每次我进行自我反思并思考这个问题的答案时，目前的答案都是否定的——我不想成为一名经理。我太喜欢编码了，而且我坚信，要成为一名成功的经理，你不应该编写代码，而应该完全专注于培养你的团队。我太喜欢参与技术决策和思考技术解决方案了，以至于不想放弃这种亲身实践的经验，即使当你担任更高级的职位时，你

花在编码上的时间会减少。

不担任工程经理并不意味着你不能影响和帮助他人成长。作为一名 Staff+ 工程师，你需要许多核心管理技能，即使你不是经理，而且我发现阅读管理书籍非常有帮助。我实际上认为这两个角色，即使它们在独立的平行轨道上，也比人们想象的更接近。

有可能在某个时候，这个问题的答案可能会改变，这是可以接受的。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我广泛使用 Twitter，但我主要是一个消费者，关注科技领域的很多人。我通常关注那些我在会议上看到过演讲或者和我一起工作过的人，并且我发现他们的内容与我相关。以下是一些，排名不分先后：[卡米尔·富尼耶, 拉拉·霍根, 乔希·威尔斯, 维姬·博伊基斯, 大卫·加斯卡, 朱莉娅](https://twitter.com/jewelia) [格蕾丝,霍尔顿,约翰,查丽蒂,西奥·施洛斯纳格尔, 杰西卡 乔伊·克尔,](https://twitter.com/jessitron)[莎拉·卡坦扎罗](https://twitter.com/sarahcat21)[,](https://twitter.com/jessitron)[橙皮书](https://twitter.com/orangebook_)[.](https://twitter.com/jessitron)

我也喜欢阅读（我每年读大约50本书），而且从去年开始，我总是尝试在我的Goodreads账户上为我读的每本书留下一个迷你评论，但这里有一些我觉得有用的书：[我的 Goodreads 账户](https://www.goodreads.com/user/show/11950463-diana-pojar)

[谢谢](https://www.goodreads.com/book/show/20487821-thanks-for-the-feedback)[激进坦诚](https://www.goodreads.com/book/show/32809138-radical-candor)

[这个 改变](https://www.goodreads.com/book/show/34616805-the-manager-s-path)

[领导力与自我欺骗：走出盒子](https://www.goodreads.com/book/show/18966789-leadership-and-self-deception)

[这个 永远](https://www.goodreads.com/book/show/29342515-the-coaching-habit)

[首先， 不同地](https://www.goodreads.com/book/show/30109687-first-break-all-the-rules)

[这个 获得真正的幸福](https://www.goodreads.com/book/show/36752952-the-courage-to-be-disliked)

[给予](https://www.goodreads.com/book/show/16158498-give-and-take)

[错误 错误的决定和伤人的行为](https://www.goodreads.com/book/show/9530608-mistakes-were-made-but-not-by-me)

这并不是一份详尽的清单！

## 丹·娜 - Squarespace 的Staff Engineer和团队负责人

*故事记录于2020年3月。在Dan的*[*博客*](https://blog.danielna.com/)*、*[*Twitter*](https://twitter.com/dxna)*和*[*领英*](https://www.linkedin.com/in/danielna/)*上了解更多关于他的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是Squarespace的Staff工程师。Squarespace是领先的一体化平台，用于构建精美的在线形象：网站、域名、在线商店、营销工具、安排预约等。我还担任国际化平台团队的团队负责人，该团队负责构建和维护Squarespace产品中国际化的基础原语。工程师使用我们拥有的工具和库来创建本地化产品。

#### Squarespace的Staff-plus工程师是做什么的？你如何安排你的时间？

我认为在实践中，Staff-plus工程师的日常职责各不相同，这取决于你的具体角色以及你的职责在组织中的映射方式。

我作为团队负责人的职位意味着我完全负责团队的产出，无论是从业务角度还是技术角度。在业务方面，我花大量时间与公司内不同的团队和职能部门会面。这些利益相关者包括产品、战略、客户运营等。我希望确保我尽可能多地获得输入，以验证我团队的路线图反映了我们公司最重要的优先事项。

在技术方面，我经常发现自己正在审查技术文档，或者在白板上为我团队正在进行的工作确定范围。我的角色已经演变为更少的手动编码工作，更多的是提出关于架构决策和部署策略的探究性问题。一个讽刺之处在于，作为一名Staff

工程师，我实际编写的代码比我作为非Staff时要少得多。这绝不是普遍适用于所有角色的，但在我的团队的背景下，关闭vim并在更具战略/监督作用的角色中运作是我利用时间的最高杠杆。我很幸运，我的团队已经由优秀的工程师组成，因此我特定的代码贡献对我们的产出影响较小。

但是Squarespace的许多Staff-plus工程师不是团队负责人，并且编写大量代码。其他人则专注于工程流程和文化。总的来说，我认为Staff-plus工程师的职责具有高度的背景性。

##### 作为一名Staff-plus工程师，你觉得最有影响力的地方在哪里？作为一名Staff-plus工程师，你做了哪些你以前无法做或不会做的事情？

我可以在高于个人项目和团队的更高层次的工程讨论中占有一席之地。我们有定期的staff工程会议，我们在会上讨论跨团队的问题，这些问题本质上既有技术性的，也有非技术性的。作为一个假设的例子，在这种类型的会议中，我会很乐意提出我认为工程入职流程中的缺点。将工程入职等主题归因于特定团队可能很困难，但缺乏正式的所有权并不会降低其重要性。我认为Staff-plus的一个关键责任是愿意拥有所有有助于（或阻碍）工程产出的事物，包括技术战略和文化。

关于已经改变的事情，我的头衔在日常对话的一开始就赋予了我很高的可信度。虽然我不提倡一种重视头衔胜过想法的文化，但如果我说它没有帮助我升级或推动我以前可能难以解决的问题，那我就是在撒谎。

##### 你是否花时间倡导技术、实践、流程或架构的变革？你能分享一个影响你组织的故事吗？

我并没有真正从类别角度考虑倡导。我主要只是希望我们的工程团队和产品尽可能做到最好，并解决一些

经验告诉我我可以帮助改变的事情。一些例子：

当我刚加入公司时，我们正处于员工人数的巨大增长期，我注意到除非你碰巧一起参与某个项目，否则很难了解其他团队的任何人。因此，我创建了一个 Slack 房间——#connect-engineering——它使用一个机器人每两周随机配对工程部的两个人喝咖啡。这个房间已经配对人们喝咖啡两年多了。

根据个人经验，我知道工程领导职位可能会让人感到孤立，与同事交谈时，我能听到一些孤独感。因此，我和一些同事[创造 管理读书俱乐部，向团队负责人和工程经理开放。](https://blog.danielna.com/starting-an-engineering-management-book-club/)

公平地说，这两个例子都不需要 Staff-plus 的头衔。但我确实认为，成为一名有效的 Staff-plus 工程师的一部分是关心和解决文化差距，就像关心和解决技术差距一样。

#### 你第一次在 Squarespace 获得了 Staff 工程师的头衔。晋升为 Staff 的过程是怎样的？

我被聘为高级软件工程师 II（比 Staff 低一个级别）。我很幸运地加入了一个从事高影响力项目的团队，我能够立即为之做出贡献。该项目最困难的部分涉及一个我已经熟悉的问题领域——跨代码库的广泛而彻底的更改——我提出、原型设计并最终交付了一个替代架构，我认为这将更好地帮助公司取得成功。这成为了我们的前端翻译系统，我在我们的工程博客上写了关于它的文章：[构建前端翻译系统](https://engineering.squarespace.com/blog/2018/building-a-system-for-front-end-translations)。

我还负责围绕新翻译系统的沟通和教育工作，在内部会议上介绍该架构，并发送有关项目状态的相关电子邮件。将这种技术贡献归类为

一些有意义的文化倡议——其他的内部演示，#connect-工程等——我的经理有一个很好的晋升理由，并得到了工程主管的同意。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

我觉得在职业阶梯上晋升就像一个累加练习，迫使你关心比以前更多的事情。关心更多的事情很难。

举一个简单的例子：实习生关心他们可以在三个月内构建的某个功能的某个小方面。团队的全职工程师关心该功能的整个生命周期。团队负责人/经理关心构成产品的整套功能。主管关心其组织拥有的整套产品。等等。

阶梯上的每一级都意味着你关心另一个抽象层，此外还要关心你当前级别之下的所有层。

我觉得Staff Engineering（资深工程师）的角色也很类似，因为你要离开特定技术领域的舒适区，进入更通用的问题领域：工程。作为领导者，你要离开潜在的技术舒适区，进入影响工程产出的挑战系统领域。哪些是阻碍工程团队发展的最大问题，而这些问题又不在团队所有权的范围内？这些现在是你的问题，此外还有你技术领域的所有问题。

因此，虽然Staff是一个令人向往的头衔，但它也包含着重大的额外责任。你现在是一名领导者，无论你是否愿意。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

我认为赞助是任何高级角色的关键责任，并且对任何工程组织的成长都至关重要。我想“赞助”的定义各不相同，但对我来说，一种切实可行的方式是为以下方面提供机会

曝光。例如：

让资历较浅的团队成员有机会在更广泛的会议上拥有并展示他们的工作。

联系一个刚刚发布了一个很棒功能的团队，为我们的工程博客写一篇文章。

鼓励我在#connect-engineering咖啡馆遇到的，拥有独特经验或视角的人做一个内部演示。

确保会议不被少数活跃分子的观点所主导，并征求房间里每个人的意见。

在大型 Slack 聊天室公开表扬那些做了很棒的事情但大家都没看到的人。

[Lara Hogan发表了一篇关于实践中赞助的精彩文章：什么是赞助 看起来像什么？](https://larahogan.me/blog/what-sponsorship-looks-like/)

#### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

是的，我仍然积极地考虑它。我知道把这两个阶梯看作是互斥的更方便，但我不这么认为。

我仍然喜欢编写代码和管理团队，而且我认为在高水平上同时做到这两者的能力对于长期的工程成功至关重要。Charity Majors 在这个话题上有一篇很棒的博文，我推荐阅读：“工程师/经理钟摆”。[这个 工程师/经理钟摆”。](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)

Charity 认为“经理职业道路 vs 工程师职业道路”是一种错误的二分法，花时间在两种角色之间交替可以让你在两者上都做得更好。这与我自己的经验相符。我之所以能成为一名更好的经理，是因为我知道在一个计划不周的项目中成为一名 IC 有多么糟糕，而且我之所以能成为一名更好的 IC，是因为我知道如何在项目进展不顺利时发出警报。

我认为构建软件最重要的战略技能之一是能够趋同于务实的决策。我反复看到的一种失败模式是，当产品经理提出业务需求，而工程师提出技术上的反对意见时，双方都不愿意让步。The

能够同情两方面的动机并化解这种紧张关系是完成任何事情的唯一途径，而培养这种同情心的最好方法就是坐在两个位置上。

为了具体回答这个问题：在加入 Squarespace 之前，我之前的职位是工程经理。我喜欢当工程经理，但我想保持我的技术技能，所以我接受了一个 IC 职位。然后我被提升为 Staff。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

在工程领导力的背景下，有两本书脱颖而出。

[我最喜欢的工程领导力书籍是高产出管理 作者：安德鲁·格鲁夫我每年都会从书架上拿起它，然后](https://www.amazon.com/dp/B015VACHOK/)

[在领导力的人性化方面，我真的很喜欢拉拉·霍根的书：坚韧 管理我有幸在以下公司开始了我在纽约的科技职业生涯](https://resilient-management.com/)

就非书籍而言，我订阅并喜欢阅读“[非理性繁荣](https://lethain.com/)”，威尔·拉森在其中定期撰写关于工程管理的博客，具有高度务实和战略的视角。我最近还发现了并喜欢阅读马蒂·卡根的“[洞察博客](https://svpg.com/articles/)”，主要是因为产品领导力是我不太熟悉的领域，并且有兴趣了解更多。

我的榜样是我多年来密切合作的一些了不起的同事

。我在Etsy与[丹尼尔·埃斯佩塞特](http://www.danielespeset.com/) 并肩工作了四年，学到了很多关于将技术执行与文化影响相结合的知识。我学到了很多东西，比如看着拉拉为我们的工程团队争取并实现同工同酬。我从现在的同事那里学到了很多，比如[坦尼娅·赖利](http://noidea.dog/) 制定并改进我们的工程流程，以适应我们不断增长的规模。我最受那些我亲眼目睹有勇气为了公司变得更好而做出改变的人的启发，尽管他们在此过程中遇到了任何摩擦。

## 乔伊·埃伯茨 - Split公司的高级职员软件工程师

*故事记录于2020年3月。在她的*[*博客*](https://medium.com/%40jkebertz)*、*[*Twitter*](https://twitter.com/jkebertz)*和*[*领英*](https://www.linkedin.com/in/joyebertz/)*上了解更多关于乔伊的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是Split.io的高级职员软件工程师，负责我们所谓的COE团队的后端工作。Split是一个功能标记和实验框架。我们专注于使我们的客户能够在CI/CD中分离部署和发布，并启用A/B测试。我的团队负责我们Web应用程序的大部分主要业务逻辑，包括从数据存储到API的所有内容。有一个单独的团队专注于实验方面，包括所有详细的统计数据，因此我们能够更专注于主要平台。

#### 在Split，一名Staff-plus工程师做什么？你如何安排你的时间？

我仍然有点新，所以我仍在努力定义我的角色，这也是更高级别角色的魅力所在。今天，我仍在适应中，所以我可能将大约一半到四分之三的时间花在我的特定scrum团队的任务上，就像这里的任何其他工程师一样。在剩下的时间里，我参与对话并与其他工程师合作，以定义我们许多更长期的架构和战略，包括我们未来的API和平台战略，我们希望如何开发我们的授权框架，分解和解耦我们的构建等等。我最近还接管了我们后端部门的领导，现在与另一位工程师共同领导它，我们正在努力制定后端技术愿景，确定技术项目的优先级并领导标准讨论。如果这还不够，我还会继续定期在我的博客上写作并在会议上发言。

#### 作为一名Staff-plus工程师，你觉得最有影响力的地方在哪里？作为一名Staff-plus工程师，你做了哪些你以前无法做或不会做的事情？

当我可以促进为一个领域设定技术愿景并让人们朝着该愿景前进时，我感到最有影响力。我认为我们都会同意我们希望我们的代码比现在更好地架构或以某种方式改进。但是，我发现人们通常对想要变得更好有一种模糊的感觉，而没有清楚地了解他们想要的东西是什么。我喜欢帮助团队就他们试图达到的目标达成共识（实际上，即使我们永远无法达到目标也没关系），并提出一个如何实现目标的大致计划。这样我们都在朝着同一个方向前进。清楚地了解我们想要什么使我们能够与产品部门合作以确定其优先级。即使我们从未将整个事情确定优先级，但知道如何到达那里，我们就可以慢慢地进行更改，这将引导我们朝着那个方向前进。例如，如果我无论如何都要接触一个文件并且可以进行一些调整，使我更接近该愿景，我就会这样做。如果不知道该愿景，这些调整将永远不会发生。仅有愿景是不够的，我们需要每个人都理解该愿景并将其内化。我刚才提到那些小小的改变的力量在于，如果每个人都将它们作为日常编码的一部分。突然之间，我们所有人都在朝着一个共同的目标努力。

我认为现在和我刚入行时最大的不同在于我的主人翁意识和责任感。我一直都愿意反驳或推动改进。然而，当我刚入行时，我常常认为某些事情是别人的问题。现在，所有的事情都是我的问题。我可能会选择不优先处理某些事情，因为我认为它不如其他事情重要，或者我可能会选择将问题委派或转交给其他人，但我仍然认为这是我的问题。我再也不会认为别人会处理某些事情。我仍然坚信要选择我的战场，我不会处理所有的事情——那太多了。然而，我也不会认为别人会处理，所以如果某件事值得完成，那么要么由我来完成，要么由我把它转交给其他人。

##### 你是否花时间倡导技术、实践、流程或架构的变革？你倡导过什么？你能分享一个影响你组织的故事吗？

是的。所有这些。在我目前的职位上，我会说这是我工作的重要组成部分。虽然作为一名工程师，我也在一个 Scrum 团队中做出贡献，但我会说很多

我的工作是留意我以前见过的陷阱或更大的问题模式。我认为我的工作是提高所有工程的效率——无论是通过技术、架构还是流程。但是，我不应该为了改变而改变。这些年来，我倡导了很多事情，从重写我们的电子邮件通知系统到重新思考测试，再到重做几个授权框架。

对于某些事情，比如电子邮件的全面改革，我没有做什么大的或伟大的事情，我只是每次产品部门想要添加通知时，我都提醒他们我们的系统已经准备好崩溃了，在我们修复它之前，我们真的不能再添加任何东西了。当我反驳时，我周围的工程师也意识到他们可以反驳。起初，产品部门大多选择不添加更多的通知，但最终他们决定修复系统。在这种情况下，这主要是一个向他们解释系统风险的问题，并坚持我认为在保持我们的系统运行方面正确的行动方案。

对于其他事情，比如授权框架，我的任务是找到一个解决方案。在这些情况下，即使人们想要一个新的/更好的解决方案，你仍然需要说服他们你选择了正确的东西。对于极其复杂的系统，人们常常会认为他们发现了你遗忘的东西（也许他们确实发现了），所以尽早和经常寻求反馈，并仔细记录和沟通你选择的内容和原因，以及你考虑过的其他内容以及你没有选择其他内容的原因，这一点非常重要。人们需要感到被倾听，他们需要知道你充分考虑了他们的担忧。他们也想了解你的思考过程，但更重要的是，他们想了解你做了彻底的研究，而不仅仅是选择了第一个出现的东西。事实上，当我在审查别人的设计时，这是我真正寻找的东西之一——还考虑了什么？

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

是的。一旦你担任任何更高级的职位，这总是你角色的一部分，假设你选择利用它。由于我还在 Split 工作，我还没有太多机会在这里这样做，但我相信这将会改变。

有时赞助是大事——推荐人们领导项目或管理团队，但很多赞助是小事——鼓励那些对自己有点不确定的人，向他们通常无法接触到的更高级的人展示他们的成就，找到将你的工作委派给那些可以从中获得成长机会的人的方法。我认为有可能成为一名没有赞助的高级工程师，但我不确定是否有可能成为一名没有赞助的伟大的高级工程师。赞助是我们能够培养周围人的最有效的方式之一，我会说培养他人是我们角色中最重要的方面之一。

##### 你第一次在 Box 获得了 Staff engineer 的头衔。晋升为 Staff 的过程是怎样的？

在 Box，我们会提交一份晋升案例，其中概述了根据工程评估标准，我们如何在下一个级别上运作。我们的经理也会提交他们的推荐信，这两份材料会提交给由经理和 IC（至少比我们申请的级别高一级）组成的晋升委员会。他们审查案例，召集经理回答任何问题，然后提出建议。我们的副总裁可以更改任何决定（尽管据我所知，这种情况从未发生过）。如果答案是否定的，会给出反馈说明原因，你可以用额外的信息撤销决定，或者下次再试。申诉有时会通过，所以如果你不同意反馈，值得一试。我喜欢这个过程，因为它允许对我们的成就最有了解的人来撰写，并且允许你即使你的经理不同意也可以申请晋升。另一方面，我不喜欢这个过程，因为它微妙地歧视那些缺乏自信和不擅长自我推销的人。这也导致经理们在启动晋升流程时的主动性降低（让工程师来找他们说他们想要晋升，而不是主动建议）。

##### 您认为达到资深工程师级别最重要的两三个因素是什么？您加入的公司、您的地理位置或您的教育背景对您的职业道路有何影响？

我想说我的地理位置可能没有太大的影响。教育

在我刚入行的时候，教育在帮助我获得面试机会方面起到了很大的作用，但从那以后（至少是直接地）作用就小了很多。我想说，对我来说，三个最重要的因素是公司、知名度和机会。

我认为在很多公司都有晋升的机会。然而，我发现在一个快速发展的创业公司工作真的对我很有帮助。当我加入 Box 时，工程团队大约有 30 人，当我 8 年后离开时，已经有几百人了，但大部分增长都在前半段。这让我能够进入一个较小的工程环境，在那里我可以真正了解环境、人员和代码。然后，因为我们正在成长，对于那些有动力和愿意接受挑战的人来说，有很多领导机会和技术挑战。因为我们成长了，机会也随着我成长。与此同时，我周围也有足够多的人让我学习（我之前在一个非常小的创业公司工作——2-4 人，在那里根本没有这种机会）。

通过知名度，我的意思只是找到某种方式让自己被人们所知。我一直在现场工作，我发现这让事情变得更容易一些，但我认为即使你远程工作也是可能的（尽管可能更具挑战性）。如果你做了非常出色的工作，但没有人知道，那么当晋升的时候，你就会被忽略。此外，随着你变得更资深，你的一部分工作变成了指导和教导他人，并帮助你的公司创建一个技术品牌——所有这些在定义上都是可见的。知名度可以采取多种形式，但对我来说，我想说有几件事做出了贡献。我非常活跃于我们的 Slack 讨论论坛，尽可能地回答人们的问题。我还做了很多博客，并在内部和外部做了一些演讲。最后，我积极参与我们的女性科技团体，这让我能够与整个工程部门的各种人建立联系。

最后，机会。这些机会看起来也可能大相径庭。对我来说，有一个特别有帮助的机会——我加入了我们的 API 标准委员会。一开始我有点犹豫，因为我不认为自己是 API 方面的专家，但在阅读了几本关于 REST 的（简短）书籍，以及我之前在各种 API 上的工作之后，我对它有了相当扎实的掌握。这个小组的强大之处在于它横跨工程部门的许多团队，这让我有机会与许多不同的工程师合作（并给了我我刚刚

提到的）知名度。它还让我能够清楚地指出我在影响他人和成为为质量而战的人方面所做的事情。我们在那里的项目对整个工程部门产生了广泛的影响，并让我能够从整体上思考一些事情（在本例中是我们的 API）。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

我实际上并没有一个 Staff Project。在我晋升的时候，我大约在 6 个月前从管理岗位转回来，所以我提到了我管理期间的一些领导经验。当时，我（从技术方面）领导着 Box 的一个小团队，负责一个跨公司合作项目，该项目涉及了解另一家公司的开发团队的需求，并弄清楚如何在满足他们需求的同时尽可能少地构建。我是一个工程范围内的 API 工作组的成员，负责建立和维护我们的 API 标准，并且我还有几个正在进行中的副项目。我想说，所有这些都对我的晋升的各个方面做出了贡献，并共同帮助我证明我可以展示他们期望的所有方面。

##### 你还记得关于晋升到 Staff 的任何特别有用的建议吗？有没有更容易晋升到 Staff 的途径？

我得到的其中一条建议是放大我的优势。我们所有人都有优点和缺点，我们花很多时间谈论“需要改进的领域”。很容易让人觉得晋升的最佳方式是消除所有这些缺点。然而，如果这真的是我们薄弱的领域，那么即使只是稍微有所进步，也可能需要大量的工作和精力。显然，你仍然希望确保你没有任何真正糟糕的领域，但假设你已经做到了这一点，那么就专注于放大你的优势。你如何将你擅长的事情变成你的超能力？另一件需要考虑的事情是，你如何利用你擅长的事情来弥补你薄弱的事情？例如，我是一个巨大的内向者，不太喜欢与我不认识的人交往。我不擅长与陌生人建立联系。然而，我擅长写作并且喜欢写作。我利用我在公共博客上的写作来结识我原本不会认识的人，并更广泛地获得曝光。事实上，我确信我从中获得的

比我去参加很多很多的聚会所获得的要多得多。

另一个更具策略性的想法是与我们在 Box 撰写晋升案例的过程直接相关。有人向我提出了几点建议

- 首先，在我确定自己准备好晋升之前，就把晋升案例写好。这能让你看到可能存在的差距，并为你提供非常具体的工作方向。（或者你可能会感到惊讶，并意识到你比自己想象的更早准备好晋升）。第二点是要非常清楚这些差距在哪里。当晋升委员会阅读晋升案例时，所有的案例都会非常积极。没有人会在晋升时说任何负面的事情。因此，委员会不会寻找负面的东西，而是会寻找没有说的事情。空白点在哪里？什么似乎被回避或一带而过。从这个角度审视你的案例——可能缺少什么？你正在掩盖什么？一定要努力解决这些问题。最后，讲述你的故事。

我们的晋升案例有带有明确问题的模板，但是，尤其是在更高的层级，每个人都不一样，我们也不希望他们一样。不要只是回答问题，首先想想你的优势是什么。你的超能力是什么？你的故事是什么？然后弄清楚如何将这个故事融入到提示中。如果你在其中包含你最好的优势，你将会有一个更好的整体案例。

如果我没有在管理层兜圈子，我可能会更早晋升到 Staff 级别。话虽如此，我不后悔这样做，我学到了很多关于人们如何思考、组织如何运作以及大型项目如何确定优先级的知识。所有这些都一直在帮助我以 IC 身份完成我的工作，并且可能进一步帮助我晋升到 Senior Staff。虽然我确实认为这在我晋升到 Staff 级别时明显放慢了速度，但我实际上不太确定下一个级别——我认为我很有可能在没有这些经历的情况下在 Staff 级别待更长时间。所有这些都表明，即使我没有走最直接的路线，我仍然学到了很多对我的长期发展有帮助的东西。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

你越资深，你的工作就越与代码无关。当然，与人事

经理不同，你仍然有非常技术性的倾向，甚至在 Principal 级别，你可能至少还会做一些编码。然而，你级别越高，你的工作就越是关于指导和培养你周围的人（以及更广泛的范围），通过建立你公司的公共技术品牌来建设你的团队，注意到可以改进或纠正的更大的技术趋势，帮助为你的团队或公司设定技术愿景，并倡导为技术债务项目提供资源。它变得更多的是关于看到更广泛的事情并让其他人加入。突然之间，沟通、领导力和说服力变得比以前更加重要。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

实际上，我在 Box 工作期间管理了大约一年半的时间，发现我讨厌它（你可以在我的[关于那件事的博客文章 主题）。 也就是说，我发现两者之间实际上有很多重叠之处](https://code.likeagirl.io/why-i-left-management-the-engineering-technical-track-vs-management-track-abef5b1d914d)[在我工作期间学到了很多 管理 而且随着我晋升到 Staff，这段经历实际上帮助了我](https://medium.com/box-tech-blog/no-regrets-my-time-in-management-wasnt-wasted-140b40ded0e6)

#### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我不太倾向于追随任何特定的人，而是向我周围的几乎所有人学习并从中获得灵感。我将在这里列出一些，但老实说，我会说我从各个级别（包括许多比我年轻的人）的无数不同的人那里学到了东西。

我有一个经理，每次我带着问题去找他时，他总是会反过来问我，问我应该怎么做。到了后来，我都能听到他告诉我直接给某人反馈，或者告诉我如何解决问题，而我甚至不必和他说话。他真的教会了我，虽然作为一名经理，他愿意支持我，但如果我自己能做到，我将学到最多，成为最好的自己。他

教会了我对一切负责。

作为对比，我还想提一下我曾共事过的一位首席工程师，他后来告诉我，我不需要尝试自己做所有的事情。在我学会承担责任之后，我开始忘记我不是孤身一人。当然，我听人们谈论过授权，但听听或者在冲刺任务中想想是一回事，而授权让某事优先处理，或者授权弄清楚团队的技术愿景，或者授权跟进一项倡议，则是另一回事。

我曾经和另一位同事一起工作，她有时会让我非常恼火，因为她解决问题的方式和我的截然不同。我认为显而易见的事情，她会要求澄清；我认为大家都心知肚明的事情，她会要求详细解释。然而，她也是我合作过的最聪明的工程师之一，和她一起工作让我意识到，不同的风格不仅可以同样出色，而且有时将两种冲突的风格结合在一起，可以产生比我们任何一个人单独工作更好的结果。她能发现我认为显而易见的事情中的漏洞，虽然她有时让我很恼火，但我们完成了一些了不起的事情，我也因此变得更好。

## 达米安·申克尔曼 - Auth0 首席工程师

*故事记录于 2020 年 8 月。在*[*博客*](https://yenkel.dev/)*、*[*Twitter*](https://twitter.com/dschenkelman)*和*[*领英*](https://www.linkedin.com/in/damianschenkelman/)*上了解更多关于达米安的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是[Auth0](https://auth0.com/)的首席工程师，Auth0 是一个身份即服务平台。我在系统架构组工作，该组目前有三名首席工程师。我们与不同的团队合作开展战略计划，并塑造 Auth0 的[技术战略](https://yenkel.dev/posts/achieving-alignment-and-efficiency-through-a-technical-strategy)、架构决策和指导方针。

[在撰写本文时，我正在与一组身份和访问管理 管理 （IAM）团队合作，担任一个大型新产品功能的技术负责人，以及](https://auth0.com/learn/cloud-identity-access-management/)

#### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

在工程部门内部，我们按领域组织（目前是身份和访问管理、开发者体验、服务管理和平台）。Auth0 的 Staff Engineer 是可以在技术上领导领域内团队的人员。Staff Engineer 通常是领域内单个团队的成员，同时也能积极地为整个领域的计划做出贡献。

Principal 是我们晋升阶梯中的下一个级别。Principal Engineer 可以在特定团队中（深度），也可以与多个团队合作，他们的范围涵盖整个组织（广度）。今天，我以“广度模式”运作。这意味着既要从事具体的计划，也要定义技术战略、我们平台的技术选择，并领导设计和架构工作组（又名 DNA）。

DNA 有 6 名成员（3 名常任 Principal，以及 3 名每 6 个月轮换一次的 Staff/Senior II）。该工作组制定决策和指导方针，以帮助推动

Auth0 的技术朝着特定方向发展（例如，避免语言泛滥，以便我们可以构建一次库，人们可以轻松切换团队），并与团队合作进行大型计划的技术审查。

我的角色最大的不同之处在于，因为我在公司工作了 6 年多，其中 3 年多担任工程总监，所以我拥有“最广泛的范围”。我与产品和平台团队合作开展计划，并且经常与公司的其他部门合作：参与与重要潜在客户的对话，与我们的法律团队合作制定合同语言，或与市场营销部门合作。

##### 您每天如何安排时间？

这种情况差异很大。典型的一周涉及很多会议，所以我正在尝试一种新方法：将会议集中安排在周一、周三和周五。周四完全封闭，周二只处理紧急事务。因为我们是远程办公，所以所有会议都在 Zoom 上进行。

在开会日，我会有以下循环会议：\* 1:1 会议：与我的经理（工程副总裁）或团队经理或技术负责人进行沟通。这些对话对于了解他们的最新情况和了解他们的挑战非常有帮助。我觉得如果过于脱离这些，会影响我有效完成工作的能力。\* 团队会议：工程领导层、设计与架构工作组。

还会举行非循环会议。一些示例主题可能包括：

我正在技术领导的特定项目

帮助一组团队弄清楚如何启动某件事进行同步设计审查

在周四（以及尽可能多的周二），我花时间思考：

当前的计划以及它们的进展情况

我们将来可以/应该做什么（下个季度，明年）撰写文档、指南、博客文章

（不经常）进行 POC 和/或编写小型工具

##### 作为一名 Staff-plus 工程师，您觉得自己最有影响力的地方在哪里？一个具体的故事会很棒。

[扩大卓越](https://www.amazon.com/dp/B00EGMQIDG/)最大的影响来自于能够帮助实现“人员规模化”，从而积极地影响内部尽可能多的人的工作。《规模化卓越》这本书提供了一个容易理解的类比：规模化是一场地面战，而不是一次性的空袭。这需要大量的时间和耐心，但要达到你的目标，你需要使整个公司在目标以及如何实现目标方面保持一致。

作为首席工程师，我尝试寻找我认为从长远来看能为尽可能多的人指明方向的机会/差距。将我们产品交付部门的约 200 人围绕某个主题统一起来，比我自己为某个问题编写解决方案更有价值。前者影响更大，可扩展性更好。

#### 你能想到你作为一名Staff-plus工程师所做过的，在你达到这个头衔之前你无法做到或不会做的事情吗？

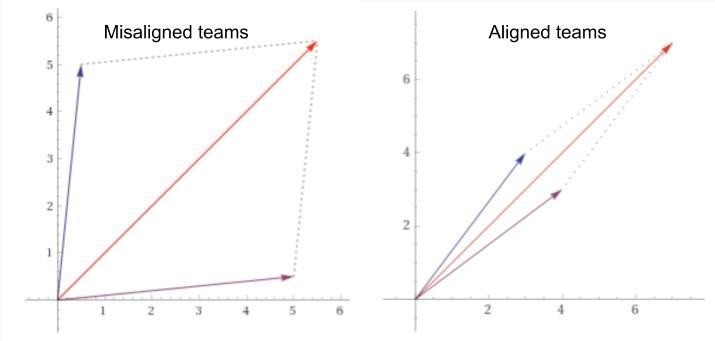
在成为首席工程师之前，我是 Auth0 的工程总监。最有趣的是，作为首席工程师，当提供反馈时，人们的防御心理会减少很多，而且在 1 对 1 的交流中，他们似乎也更加开放。我认为这可能与你作为首席工程师并不“代表组织的一部分”有关。

在这方面，作为一名个人贡献者感觉好多了。

#### 你是否花时间倡导技术、实践、流程或架构的变革？你倡导过什么？你能分享一个影响你组织的故事吗？

快速发展的公司面临的一个常见问题是，通常存在某种“缺乏清晰度”的情况。在我们的案例中，对于未来会发生什么存在很多困惑，这使得我们在做出技术决策时变得缓慢且效率低下。团队不确定是否应该使用某种特定的技术，因为他们不知道该技术在未来是否会得到支持；他们不确定是否应该以某种方式构建某个

特定产品，因为他们不知道这种方法是否符合我们的长期技术战略，等等。自然，这导致了很多效率低下的问题。



我们认为我们需要一个长期的方向，解释如何处理当今问题的技术实施，以及如何弥合我们最初的情况与未来愿景之间的差距。更准确地说，我们需要一份书面的技术战略，详细说明我们应该做什么和不应该做什么，才能在长期内取得成功。

在与很多人交谈后，我了解到他们都接触过不一致的信息和谣言，这让他们害怕做出决定，例如：“我听说公司将来会选择 X”或“我听说我们的平台团队将不再支持这项特定的技术 Y。”很多困惑是由一个特定的谣言引起的，即我们将支持某个客户需求及其技术影响。人们一直在听到这件事，但从未宣布具体的计划。我写下了这些问题，将所有的点连接起来，旨在将这些信息转化为知识。我意识到我们需要短期和长期的解决方案。

**短期：**我们必须填补与一些更紧迫和短期事项相关的不确定性空白。团队需要做出技术决策，并且不能

等待一个成熟的技术愿景和路线图。我们还意识到，一旦我们有了长期的愿景和决策，自然就需要审查特定例外的决策。我组建了“设计和架构”（DNA）小组，该小组还编写了指南和建议，包括“批准的”技术选择，以指导团队做出不需要审查的独立决策，并建立了 RFC 审查流程。

**长期：**我想出了一系列我认为公司需要做出决定的主题。我根据两个不同的受众——高管和技术人员——调整了我的演示文稿。对于高管受众，我开发了一个简洁的演示文稿，应用了非技术性的类比和解释，并提供了可操作的解决方案。技术演示文稿更加详细，包括许多技术术语。我使用了[协商](https://en.wikipedia.org/wiki/Nemawashi)（一种非正式的过程，通过与相关人员交谈、收集支持和反馈等方式，悄悄地为某些拟议的变更或项目奠定基础），并与我的工程副总裁、其他高管、我的同事和其他高级领导分享，以在正式做出决定之前获得支持。更具体地说，我主动询问人们的想法和意见，以确保获得支持，这样，当我们开会讨论我们的决定时，他们就不会是第一次看到这些想法。我们最终见面，讨论了权衡，并达成了一系列决定。所有决定都记录在一个决策日志中，并且我们以书面形式指定了具体的负责人来推进这些决定。

#### 当您花在实际开发上的时间越来越少时，您如何与事物的实际运作方式保持联系？

这包括两个方面，一是跟上总体技术发展，二是跟上Auth0的动态以及工程团队当前的“状况”。

为了与Auth0相关的事宜保持联系，我主要做以下几件事：\* 内部：通过Slack以及与一些技术负责人和工程经理进行一对一交流，密切关注情况。这有助于我直接了解他们遇到的挑战，并找到模式或得出全局解决方案，而不是局部方案。\* 外部：与客户/潜在客户交谈，了解他们如何使用产品，阅读提及Auth0和

身份领域的推文和新闻。

我觉得自己在技术方面并没有像我希望的那样“保持联系”，但我确实在努力。我们行业每个月都有很多重要的新事物出现，很难跟上。接受会议和不亲力亲为意味着我可能无法像我希望的那样了解情况，这一点很重要。一旦我接受了这一点，我就可以开始优先考虑有价值的事情。

我读书，抽出时间做一些POC（概念验证），或者阅读关于特定主题的博客/论文，并要求领导特定的项目，以了解我们是如何开发的，即使我不经常编码。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

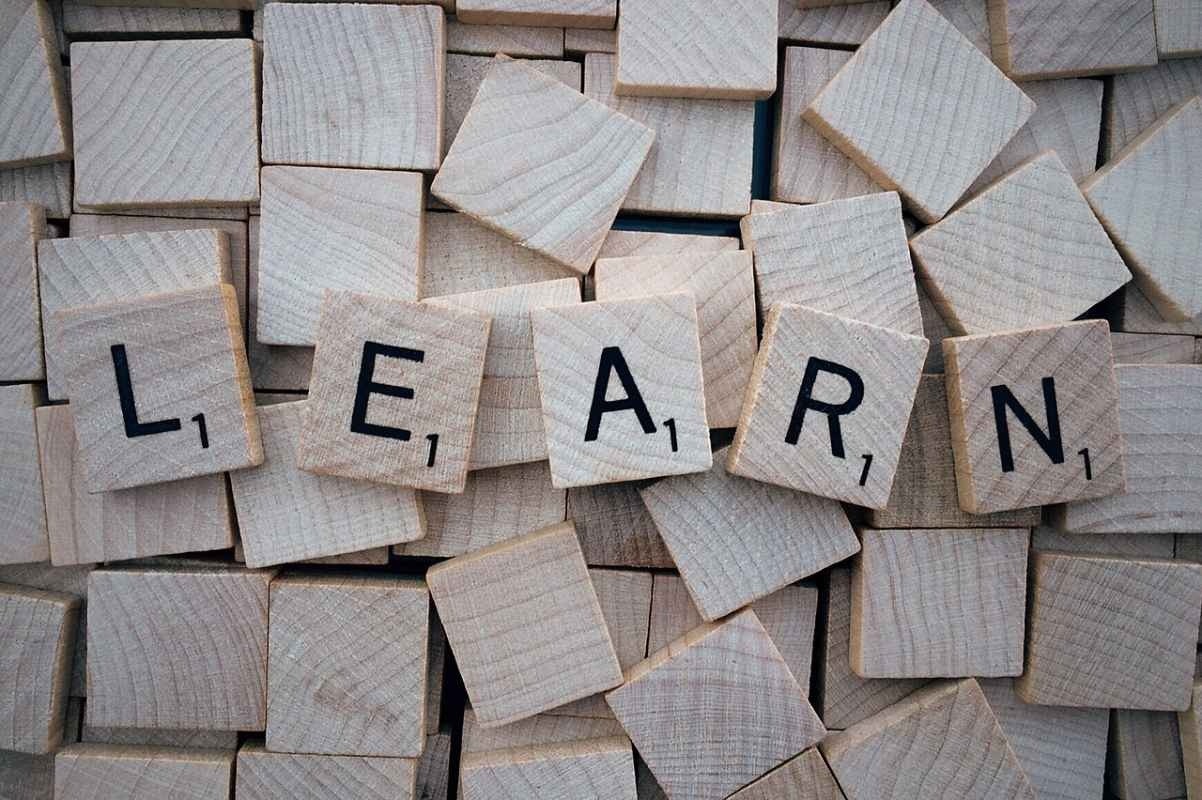
是的，很多！我是我们工程领导团队的成员。我们每周开两次会，讨论有关组织的话题。这与密切关注情况以及参与有关中期计划的会议一起，帮助我了解（有时会提出）可能存在的机会。

每当发生这种情况时，我通常会推荐我认为会从该机会中受益的人的名字，解释为什么我认为他们会擅长它，并且，如果有帮助，如果他们有任何感知到的技能差距，我会主动指导他们。

##### 您最初在Auth0获得了首席工程师的头衔。您是作为首席工程师被雇佣的吗？如果不是，晋升为首席工程师的过程是怎样的？

我的故事比较特殊。我于2014年5月加入Auth0，是第五位工程师，大约是第十位员工。当时没有头衔，没有晋升阶梯，什么都没有。大约在2015年，我开始指导几位新员工并与他们进行一对一交流。到2015年底，我正在进行我的项目，并且还在领导其他人，协助招聘等。2015年末[马蒂亚斯·沃洛斯基](https://twitter.com/woloski)，Auth0的CTO和联合创始人，正在寻找一位领导工程团队的人，他问我是否愿意担任工程总监。

我非常荣幸能够以一种最大化学习和解决难题机会的方式来对待我的职业生涯。这是帮助我做出决定的主要原则。当他向我，一个住在阿根廷的25岁年轻人，提供领导一家“硅谷”的、远程优先的、呈指数级增长的公司的工程组织的机会时，我自然而然地说了“是”。我从来没有想过“我想管理”，这只是因为我想学习和解决难题而发生的。



事情进展得很顺利。我学到了很多关于建设团队、组织、领导人员等方面的知识。因为我是最早的工程师之一，构建了很多系统，而且我喜欢技术方面的对话，所以我也在这个职位上做了很多技术领导工作，包括与产品和平台/基础设施团队合作。在2019年初，作为平台总监，我开始觉得我学习的速度不如以前快了，而且我想要比仅仅在我们的平台上工作更广阔的范围。在与当时的Auth0工程副总裁克里斯蒂安·麦卡里克多次交谈后，我意识到我接下来想要接受的挑战将是成为Auth0的技术领导者之一。我于2019年8月转任首席工程师。[克里斯蒂安·麦卡里克](https://twitter.com/cmccarrick)

##### 在你晋升为主管的过程中，最重要的两三个因素是什么？你加入的公司、你的所在地或你的教育背景对你的职业道路有何影响？

我喜欢塞内加的一句名言：“当准备遇到机会时，就会发生幸运。” 晋升为主管需要做对一些事情，但也需要很多运气。我想指出一些让我晋升为主管的关键因素，并展示运气在其中扮演的角色。

第一份工作：在阿根廷，在上大学期间开始工作是很常见的。高中毕业后，我在一家名为Southworks的优秀公司找到了一份工作。这个地方的两个关键点是：[Southworks（公司名）](https://www.southworks.com/)

该公司致力于使用尖端技术进行项目开发，这为我提供了许多磨练学习技能的机会

该公司主要作为微软美国的远程供应商，这意味着不仅技术技能受到重视，而且我们还可以练习沟通、期望管理和其他人际交往技能。

我高中毕业后就能从事软件工作的原因是，我十一岁时就开始告诉妈妈我想“制作电子游戏”，我的父母找到并支付了一所教授编程的高中。

运气如何发挥作用：我当时正要接受另一家公司的工作，这时我高中时的一个朋友告诉我，她的哥哥在Southworks工作，他们正在招聘初级人员。他很好地向我推销了这家公司，我决定暂停另一个机会，看看我是否能进入Southworks。

Auth0：我是Auth0最早的工程师之一，多年来，我参与了其产品和基础设施的许多部分，这使我能够轻松地帮助他人并为各种主题提供有价值的意见。担任工程总监也帮助我了解了我们业务的许多方面，这有助于我成为一个更有效的贡献者。

运气如何发挥作用：任何初创公司的成功都需要在

许多不同的时间点都需要大量的运气。如果Auth0没有像现在这样发展，我就不会有机会学习我所学到的东西，也不会有今天的成就。这一点尤其重要，因为我住在阿根廷，那里的软件行业比美国小得多，而且大多数公司都没有双轨制。

团队运动：我小时候和青少年时期打篮球，我早就意识到，通过得分赢得比赛比输掉比赛得很多分感觉好多了。这以两种方式塑造了我的工作方式：\* 它促使我经常帮助团队成员，看看我们如何作为一个团队取得成功 \* 它促使我学习和做一些“弥补差距”的事情，这帮助我培养了领导能力和人际交往能力，这在我职业生涯中成长时非常有用

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

我没有。因为我在 Auth0 的成长方式，我有点“跳过了那部分”。作为一家初创公司的总监，我有机会在技术上领导许多大型、关键的举措，但没有具体/明确的“员工/主要项目”。

我认为，最接近“员工项目”的是我在 2017 年和 2018 年领导的工作，旨在提高 Auth0 的可靠性和可扩展性，领导一些项目为我们的一些主要客户提供更高的 SLA。

##### 对于刚开始担任 Staff 工程师的人，你有什么建议？

Staff 在不同的地方意味着不同的事情，所以我给出的第一个建议是与尽可能多的人交谈，以明确他们所在地方的期望。

我要告诉人们的下一件事是要有耐心。他们之所以能达到现在的地位，可能是因为他们技术相当精湛并且取得了成果，但是随着你在阶梯上不断攀升，你的工作成果需要时间才能显现。你可能同时在做更多的事情，并且它们的影响具有更长的时间跨度。你现在也在影响更多不同角色的人，有时他们需要更长的时间才能“看到”你可能清楚地看到的事情。耐心、逐步影响他人和教导他人会带来长期的回报。

最后：习惯于把事情写下来并重复给别人听。写下想法、计划、推理和标准是你扩展自己的方式。当你记录下某件事时，你可以让任何人在将来轻松访问和阅读它，这更容易参考。这比“只是谈论它”要好得多：它扩展性更好，并且还可以减少被误解的机会。重复也是必要的，因为仅仅发布书面文件是没有用的，所以你必须与人们分享你的想法。举办 AMA、棕色袋子会议和其他会议来解释你的想法非常有价值。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

我没有计划这样做，但是当有机会成为总监时，我抓住了它。但是，我的想法是[这里有一个钟摆](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)你可以在两条路径之间来回摆动。这有多容易将取决于公司以及 Staff/Principal 的技能有多专业，但我认为这是可能的。

现在，我非常有兴趣继续发展我的技术技能和领导技能，因为我认为这将带来最有价值的学习和挑战。

#### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我尝试在 Twitter 上关注我认为正在做有趣事情并且我可以从中学习的人。有太多人在做有趣的事情，而且有很多东西要学习！我想到的名字有：

[Aphyr](https://twitter.com/aphyr)Aphyr与Jepsen合作以及关于分布式系统的一般内容的工作非常出色。[Jepsen（公司名）](https://jepsen.io/)

[坦尼娅](https://twitter.com/whereistanya) 有一些非常好的内容，比如[Squarespace 的 RFC 流程](https://engineering.squarespace.com/blog/2019/the-power-of-yes-if) 和[成为粘合剂](https://www.youtube.com/watch?v=KClAPipnKqw)。

[大卫·福勒](https://twitter.com/davidfowl) 分享了很多关于.NET Framework和ASP.NET内部结构的内容，我觉得很有意思。这里还有一段他分享自己如何成为[ASP.NET 架构师](https://channel9.msdn.com/Shows/Careers-Behind-the-Code/Becoming-the-ASPNET-Architect-with-David-Fowler)的视频。

在Auth0，我和[乔恩·阿利](https://www.linkedin.com/in/jon-allie-b250296) 一起工作，他是一位出色的工程师和人。他追求简洁，能非常清楚地解释事情，并且非常

考虑到他所知道的，他很谦虚。

我还没有找到很多专门针对高级“个人贡献者”的书籍或类似内容（写一本可能会很有趣）。我最近读了软件架构基础，它在描述这个角色方面做得相当不错，同时也理解了它的细微之处和灰色地带。[基础 软件架构 在描述这个角色时做得相当不错](https://www.oreilly.com/library/view/fundamentals-of-software/9781492043447/)

一些关于管理的书籍对于获得组织意识以及帮助指导、一对一交流、招聘很有用，这些都是员工级工程师通常会提供帮助的事情。在高产出管理中，Andrew Grove将“精通技术的管理者”定义为“可能不直接监督任何人，但即使没有严格的组织权力，也能影响他人工作的人”，这听起来很像员工级工程师。我强烈推荐管理人性，因为它分享的故事既容易又有趣，而且有助于对管理者产生同情心，这对于员工级工程师来说很重要。高效能人士的七个习惯也是一本对员工级工程师有很多好处的书。[高产出管理](https://www.amazon.com/dp/B015VACHOK/)[管理人类](https://www.amazon.com/Managing-Humans-Humorous-Software-Engineering/dp/1484221575/)[这个](https://www.amazon.com/dp/B00GOZV3TM/)

[加速](https://www.amazon.com/dp/B07B9F83WM/)加速是另一本很棒的书，它以一种有用的方式将公司的成功与工程实践和成果联系起来，从而影响利益相关者，尤其是在高管层面。

## Dmitry Petrashko - Stripe基础设施主管的技术顾问

*这个故事录制于2020年5月。在Twitter或Linkedin上阅读他的演示文稿，了解更多关于Dmitry的信息。*[*Twitter*](https://twitter.com/darkdimius)[*领英*](https://www.linkedin.com/in/darkdimius/)[*演示文稿*](https://d-d.me/site/presentations/)

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是一名员工级工程师，也是Stripe基础设施主管的技术顾问。

我目前的团队是Stripe基础设施团队，负责Stripe的基础设施服务，包括计算、网络、存储、数据库、数据工程、性能与效率、可观测性服务和开发者工具。我们的工作使Stripe工程师能够专注于产品。

我“成长”的团队是开发者生产力团队，该团队构建在Stripe产品开发过程中使用的流程、工具和核心库，包括测试框架、代码检查器、类型检查器、构建工具、用于逐步推广的库等等。我最初是该团队的工程师（当时还是一个单独的团队），最终成为该团队的支柱技术负责人。

#### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

Stripe的Staff Engineer不是一个角色，而是一个对应于对影响、沟通、人员和项目领导技能期望的级别。Staff Engineer担任不同的角色，我目前的角色是技术顾问（TA）。在这个角色中，我与基金会负责人Rahul Patil紧密合作，目标是提前研究关键主题，深入研究关键问题（设计、代码、分析），集思广益技术行动项，协助紧急技术跟进，为数据收集编写代码等。这个角色的目的是扩展Rahul的带宽和战略思维，但并不直接做出技术决策。

作为该角色的垫脚石，我也曾是支柱技术负责人。随着我们拥有更多的PTL，期望也得到了更好的定义：

PTL帮助他们的团队做出技术决策，这些决策能够与其他团队的技术决策良好地协同工作。Stripe的团队自己做出大多数技术决策，但经验丰富的PTL可以帮助微调这些决策，以实现更好的结果。当团队在技术问题上无法达成一致时，PTL也可以作为仲裁者。

PTL指导Stripe的技术方向，为要解决的最重要的问题提供意见，并为解决这些问题设定高层次的技术方法。

PTL通过向其他支柱技术负责人代表他们的组织来帮助他们的组织，并将其他地方做出的技术决策带回他们工作的团队，以创建一致性。

PTL为其他工程师创造机会来承担有影响力的项目，并帮助他们取得成功。

在PTL角色中，我过去常常与开发者生产力负责人和团队内部的经理紧密合作。我们交流背景信息，并朝着共同的目标努力。

这两个角色（PTL和TA）的相似之处在于，它们都与工程经理合作，并分享对我们用户的需求以及我们可用于解决这些需求的工具的见解，而EM对Stripe范围内的非技术约束（例如，资源约束）有更好的理解。

#### 您每天如何安排时间？

在一个完美的一周里，我会在周一、周三和周五参加会议或工作组：无论是 1:1 的会议还是团队会议，合作制定短期和长期的计划和战略。在我理想的一周里，周二和周四会用来独自编程。但实际上，根据当时团队的需求，我最终可能会有更多的会议或更多的时间来编程。如果我正在努力启动一个新项目，我通常会先用一周的时间减少会议：专注于项目简报，思考设计、可交付成果/里程碑以及安全/可靠性影响；然后用一周的时间来推广这个提案

在公司内部走动并处理反馈。

虽然有时可能觉得很难找到时间编写代码，但我认为这很重要，因为它使我能够与工程保持紧密联系，并成为业务需求/优先级排序与PTL所需的工程约束之间的桥梁。

##### 作为一名 Staff-plus 工程师，你觉得在哪里最有影响力？

首席工程师，尤其是支柱技术负责人，经常帮助确定新项目的方向。当我可以帮助改进一个善意且能解决实际需求的提案时，我感到特别有影响力，但起草该提案的团队缺乏经验或背景来编写一个好的计划来抓住机会。在这种情况下，一个结构良好的计划可以帮助大大缩小范围，同时获得大部分价值，从而更快地展示影响力。或者，另一种情况是，看到手头的提案解决了比团队最初预期的更多的用例，并将项目重新聚焦于团队不知道的用例，这将带来更大的业务影响：在这两种情况下，我都感到通过授权其他工程师而产生影响力。

##### 你能想到你作为一名Staff-plus工程师所做过的，在你达到这个头衔之前你无法做到或不会做的事情吗？

不，Stripe希望Staff徽章不要成为进入新机会的门槛，我相信我们在这方面做得很好。PTL角色也是如此。我们选择擅长代表他人意见的工程师担任PTL职位。甚至在我成为PTL之前，我就觉得之前的PTL，[保罗·塔扬](https://paultarjan.com/)，总是确保我的观点被表达出来。

#### 你是否花时间倡导技术、实践、流程或架构的变革？你倡导过什么？你能分享一个影响你组织的故事吗？

我被专门聘用来在Stripe中引入Ruby的类型检查。这包括与尼尔森和保罗一起，架构和实现类型检查器Sorbet，并发展围绕使用它的文化。[纳尔逊](https://nelhage.com/)[保罗](https://paultarjan.com/)[冰糕](https://sorbet.org/)

在Sorbet的早期，我们根据Stripe最需要的用例仔细选择要添加的功能。我相信我们已经成功地用一个类型系统覆盖了Stripe的大部分用例，同时保持了简单性：很容易得到一个类型系统和一种提倡复杂性和精英主义的文化，我很高兴我们的努力避免了从无类型到另一个极端。

今天，作为技术顾问，我倡导那些将产生巨大影响的变革，最常见的是在Stripe的可靠性、可扩展性、安全性和生产力方面。这可以是改变数据分片/存储的方式，也可以是改变我们处理变更管理的方式。但最大的不同在于，与我在Sorbet项目中待了多年不同，我希望找到/培养一个人来接管这个项目——在组织认可之后，并且有一个有明确里程碑和受控风险的计划。我将与推动这些重要项目的人员保持频繁的检查，目的是帮助减轻这些风险并发现更快交付项目的机会，因此我的参与仅在项目的早期阶段可见。

##### 当您花在实际开发上的时间越来越少时，您如何与事物的实际运作方式保持联系？

当我是PTL时，我每周至少有几天可以编写代码。我与团队中的其他工程师密切合作，我们不断地互相学习。

作为一名技术顾问，我编写代码的机会不如担任PTL时多。我主要在[黄色代码情况](https://www.usenix.org/conference/lisa18/presentation/kehoe)下编写代码。但这个角色的成功取决于拥有良好的洞察力和深刻的工程理解。为了做到这一点，我会与我们的内部客户交谈，并随时掌握我所支持的团队系统的设计，特别是故障阈值和故障模式。

在我的角色中，了解客户的需求非常重要。一个有用的资源是开发者生产力团队组织的Stripe范围内的工程师调查，我们希望从中找到哪些是阻碍我们的工程师提高生产力的最大因素：也许自上次调查以来，某些工具变得缓慢，或者某些用例的用户群增长了

那并没有得到很好的支持。虽然这项调查很少发现我们没有意识到的事情，但它是一个很好的相对优先级排序工具：我们可以比较有多少人抱怨某些事情，并据此确定优先级。

此外，在 Covid-19 引起的封锁之前，我过去常常在 Stripe 随机加入餐桌共进晚餐。我会问 3 个问题：

你在做什么？是什么让它变得困难？

基础设施团队如何提供帮助？

这在两个方面成为一个很好的工具：1) 将我与我的用户联系起来，帮助发现他们的需求；2) 通过进行类似于“是的，我同意我们可以通过做 X 来帮助你，现在，让我们一起看看我们应该停止做什么来为这件事创造空间”的讨论，来减轻尚未获得支持的团队的不满，在这种讨论中，一个人经常会发现，虽然他们希望我们解决他们的痛点，但他们不希望以牺牲我们降低当前项目的优先级为代价来解决。

当我从 PTL 的角色过渡出来时，我创建了一个名为 DevProd Assembly 的小组，该小组聚集了开发者生产力团队的领导者。该小组的每个成员都应与 2-3 个产品团队建立高度信任的关系，每月采访他们，并与其他 Assembly 成员汇总反馈。

#### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

虽然赞助其他工程师不是 Staff Engineer 的必需条件，但我认为它有助于您取得成功，因为它通过为他人创造机会并帮助他们取得成功来帮助您产生更大的影响。

是的，我已经帮助确定范围、启动和降低风险的多个项目，同时也帮助培养一个人在我完成下一件事时接替我。

指导和赞助之间也存在区别，我两者都做。

指导是关于帮助人们成长并产生影响。赞助是关于帮助一个人进入一个可以展示他们交付更大影响能力的位置。在与我的团队合作时，我尝试帮助人们从事一些超出他们舒适区的项目，并且我在其中赞助他们，然后，我可以指导他们以帮助项目成功。

##### 你最初是在现在的公司获得Staff工程师（Staff engineer）的头衔。你是作为Staff工程师被雇佣的吗？如果不是，晋升为Staff工程师的过程是怎样的？

我不是作为高级工程师被雇佣的。我不得不两次晋升才能达到 Stripe 的高级级别。这两次晋升都很相似：Stripe 在员工已经在下一个级别上运作了相当一段时间后才会晋升，这会调整人们的期望，即他们应该继续在该级别上运作。

##### 你认为达到高级工程师级别最重要的两三个因素是什么？

按重要性降序排列：

1. 专注于对业务和公司的影响。
2. 保持协作：通过参加会议/工作组，你应该帮助实现更好的结果。
3. 技术知识。

对我个人而言，在成为高级工程师之前，我需要擅长的领域是第二项。我已经产生了影响，并且被认为是寻求技术建议的人。我需要提高我的沟通技巧和协作技巧，以便我可以建设性地帮助团队以外的人，他们可能是第一次见到我，并且尽管他们的项目背后有良好的意图，但可能没有最好的计划来交付它。

一个帮助我的技巧是在会议结束后立即在私人聊天中寻求反馈，尤其是在那些进行得不顺利的会议之后。这让我了解到我做了什么可能导致其他方在这些会议中感到不舒服，并且在少数情况下，真正询问如何才能做得更好有助于改善会议的结果

已经进行得很糟糕的会议。

##### 你工作过的地方和你的教育对你的职业道路有什么影响？

**公司：**我很欣赏 Stripe 有这么多产生影响的机会，这确实帮助了我。

**教育：**我碰巧获得了一个非常实用的博士学位（关于如何构建快速且可维护的编译器），几乎完全转化为适用于我工作的知识：帮助公司扩展工程规模。而且，虽然它对我很有帮助，但我认为其中有很多运气成分：我碰巧在正确的时间加入了正确的实验室（当 Scala3 诞生的条件变得成熟时，我在实验室待了足够长的时间，不算太“菜鸟”，但又足够早，还没有完全确定我的研究方向）。我不确定是否应该建议其他人攻读博士学位：从我的角度来看，实际上，我的许多朋友如果在 Stripe/Google/Facebook 构建系统，用同样 4 年多的时间，也能学到同样多的东西。如果你想了解数据库是如何工作的——你可能不仅会在研究数据库的实验室学到，也会在对数据库有最高需求的公司学到，这些公司有团队致力于改进数据库。也就是说，博士学位对我来说是一个改变地点的好工具。

**地点：**我从瑞士来到美国加入 Stripe。我从俄罗斯来到瑞士，加入计算机科学领域最好的博士项目之一。我从乌克兰来到俄罗斯，加入前苏联最好的大学之一。在每一次搬迁中，我都觉得自己是在进行地理套利：我试图逃离我属于最好的地方，去到我属于普通的地方。在其中一些地方，我认为我不是最佳人选。通过加入那里的人并向他们学习，我成长了很多。我很难说美国与欧盟哪个在职业发展方面更好，但我可以肯定地说，根据我的经验，改变地点帮助我成长了很多。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

回想起来，这对我来说是一个非常难回答的问题。这是因为：据我所知，Sorbet 足够大，可以成为我的 Staff 工程师项目，但是

由于 Nelson 和 Paul 在那里，我们彼此密切且快速地合作，因此很难将项目的成功归功于特定个人，而不是整个团队。

在对项目进行第一次绩效评估时，我们三个人都收到了反馈，说我们应该有更好的方法来确定哪些影响是哪些个人造成的。虽然我很想说，我们在下一次绩效评估中没有遇到类似的问题是由于有意的行动，但我认为事实并非如此：我认为项目只是自然而然地进入了一个更大的阶段，因此我们不需要“在相同的 10 个文件中快速迭代”，自然而然地导致我们拥有更清晰和更大的所有权领域。

我成为了“内部架构/子类型”负责人，以及“与用户沟通”负责人，而 Paul 成为了“更改代码以使类型检查器喜欢它”负责人。Nelson 显然更了解 Stripe 的其他系统是如何工作的，因此帮助将该工具与它们集成。所有这些都发挥了我们的优势：我之前有类型检查器的经验（这是我的博士学位的内容），Paul 在程序化代码修改方面拥有巨大的技能，而 Nelson 不仅在系统方面非常博学，而且在 Stripe 工作的时间足够长，也足够早，几乎了解 Stripe 的每个系统。在项目的这个阶段（稳定、推广），所有这些都成为了巨大的领域，因此更容易让一个人成为某个领域的直接责任人 (DRI)，其他人偶尔提供帮助。

在 Sorbet 之后，我还在短时间内（6 个月）交付了其他几个有影响力的项目，我相信这最终促成了我获得 Staff Engineer 级别，但是，如果我必须选择一个，我仍然会选择 Sorbet，因为它具有广泛的项目范围：包括技术和文化。

##### 你还记得在晋升到Staff职位时，对你特别有帮助的建议吗？

1. 在学术界与 Martin Odersky 和 Ondrej Lhotak 合作帮助我理解了复杂系统如何协同工作，以及如何清晰地解释这一点。
2. Brian Goetz 帮助我理解了一个简单而强大的系统背后需要付出多少努力才能经受住广泛采用和设计。
3. Paul Tarjan 向我展示了调整我的沟通方式以促成所有相关方的建设性成果的重要性。

##### 对于刚开始担任Staff工程师的人，你有什么建议？

至少在 Stripe，Staff Engineer 们在非常不同的领域工作。确保你与你的汇报链条就你应该实现的影响以及为了达成该影响你允许妥协的事情达成一致。清晰地沟通你正在做的妥协以及原因。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

过去每次我考虑这个问题时，都会问自己和周围的人“这是否是一种带来更大影响的方式”。到目前为止，每次的答案都是“似乎不是”。

话虽如此，我发现即使在个人贡献者的角色中，从优秀的管理者（在我的例子中，是詹姆斯·艾里、斯科特·麦克维卡、威尔·拉森、克里斯蒂安·安德森和肖恩·奥沙利文）那里学习一些管理技能也能带来巨大的好处。

## 斯蒂芬·万 - Samsara 的高级工程师

*这个故事录制于 2020 年 9 月。在*[*Github*](https://github.com/stephen)*、*[*Twitter*](https://twitter.com/stpnwn)*和*[*领英*](https://www.linkedin.com/in/stephenwan/)*上了解更多关于斯蒂芬的信息。*

#### 请简单介绍一下您目前的角色：您的头衔、您所在的公司，以及您的团队通常做什么样的工作？

我是[萨姆萨拉](https://www.samsara.com/)的一名高级工程师。

我四年前加入这家公司，当时公司成立大约一年，我们有 50 名左右的员工。如今，我们公司有超过一千名员工，工程团队遍布湾区、亚特兰大和伦敦。

当我刚开始工作时，我们还没有组建真正的团队，我们十名左右的工程师负责处理任何出现的问题。九个月后，我们的人数增加了一倍，并围绕当时的一些核心产品组建了产品团队。在转到我们新兴的基础设施团队以启动我们的前端基础设施团队之前，我曾短暂地领导过一个产品团队。多年来，我逐渐向下转移到堆栈中，也在我们的后端和可观察性系统上进行扩展。

今天，我在我们的基础设施与平台（I&P）团队工作，大部分时间都在开发人员体验团队中，该团队构建工具以保持我们的全栈开发工作流程高效。

#### 在贵公司，“普通”的Staff-plus工程师做什么？您的角色是这样吗，还是有所不同？

我们的大多数 Staff+ 工程师都在专业领域，要么在 Web 基础设施领域，要么在设备固件领域。我想这让我成为了大多数人，但我们的 Staff 工程师所做的工作各不相同，所以很难声称在这个角色中是“正常的”。

在考察 IC 和管理路线之间的相似之处时，Staff 被认为是相当于总监的角色。Staff 工程师可以选择参与许多通常为管理者保留的流程。我们被邀请参加跨工程总监会议，至少在 I&P 中，我们参与其中

路线图规划和管理同步。最近，我们引入了技术骨干工程师来参与一些晋升评估会议。

这种访问权限的级别让人感觉像是两边都站稳了脚跟。这个角色与高级职位有很大的不同，它不太像一个“个人”的软件贡献角色，但它也与我们更注重人员管理的管理路线截然不同。

[威尔的员工原型](#_bookmark6)在威尔的技术骨干原型中，我认为我的角色介于问题解决者和技术负责人之间。对我来说，乐趣的一部分在于每6-12个月跳到一个不同的角色，与不同的人一起深入到系统的不同部分。

#### 您每天如何安排时间？

每天的情况都非常不同。现在，我尽量把周二和周四安排成会议日，这样我就可以在一周的剩余时间里有专门的专注时间。

我的会议日通常包括与我密切合作的人员进行一对一会议以及员工会议。我还会花一些时间与个人一起进行代码和设计审查，或者进行更开放的设计讨论。

在其他日子里，我的专注时间花在调查模式上，试图找出当前的问题，并为未来的项目奠定基础。哪些系统需要投资？我们的团队执行情况如何？我们小组应该为哪些即将到来的变化做好准备？这段时间花在更广泛意义上的“塑造”上。[塑造](https://basecamp.com/shapeup/1.1-chapter-02)

回顾过去，这种关注点与我担任技术骨干之前的角色截然不同。我不再在个人层面直接参与项目或团队的工作，而是将时间集中在更广阔的视角和更长的时间跨度上。

值得注意的是，我很难保证一次花超过一天的时间来编写代码。当我们考虑工程路线图带宽时，我不会被计算在内，但我确实尽量每周至少留出一天的时间来编写一些代码。

在微观层面，我保留一份名为“斯蒂芬在做什么？”的文档，

它以小时为单位驱动我的工作。有一个主要部分是关于本周的，还有一些关于未来几周的提醒。每个星期一，我都会重新开始——上周未开始的项目会被删除，很少有项目能保留到下周。



斯蒂芬在做什么？

这种默认删除的意图最终帮助我保持专注，不会感到过于分散。很长一段时间，我试图整理一个备选清单，列出要做的事情，但它主要起到了让我感到压力的作用。大多数备选项最终都会被删除，而且一个月后仍然不完整。

##### 你是否花时间倡导技术、实践、流程或架构的变革？你倡导过什么？你能分享一个影响你组织的故事吗？

是的。具体的技术或方法每个季度都在变化，但倡导最终占据了我大量的时间。在较小的方面，我帮助撰写了关于我们如何尝试处理设计文档、代码审查或代码所有权规则的文化文件。

举一个更大的例子，我花了几个季度帮助我们的产品团队采用服务级别目标（SLO）。

当时，我们拥有一套相当完善的功能和客户群，但我们对正常运行时间的衡量仍然很原始。在中断期间，很难了解对客户的影响，因为我们缺乏区分和定义来沟通（“有多少百分比的客户受到影响？是读取和写入都受到影响吗？这是中断还是现有错误？”），即使我们有很多指标和仪表板可以查看。

虽然引入 SLO 肯定需要新的工程工作，但我估计我在该项目上花费的大部分时间都用于编写文档、与人交谈以及与团队进行咨询式工作。我们希望人们能够端到端地理解可靠性目标：如何定义目标，如何在中断期间讨论它，如何在系统中衡量它，如何随着时间的推移跟踪它，以及当它变得不健康时如何做出反应。这种深度最终需要大量的消息传递和按摩才能深入人心。

与我们大多数大型“迁移”一样，让许多团队采用 SLO 是迭代进行的。我们首先在一个提供高接触支持的团队中试用了我们的新工具，然后才弄清楚为组织其余部分广泛传播的消息。我认为我的一个关键作用是能够与工程师讨论新工具的具体使用方法，同时说服主管级别的人员相信，用 SLO 进行沟通是值得他们努力的。

作为一名资深工程师，这种模式在我的大多数项目中都适用。我的角色最终是在各团队之间进行交易，以在整个组织中推广变革。

##### 当您花在实际开发上的时间越来越少时，您如何与事物的实际运作方式保持联系？

我每周都会花一些时间进行编程和代码审查，即使只是为了修复一个小错误。我尽量花时间参与其他 IC 每天都要经历的相同仪式——代码审查、浏览文档、处理中断情况等。

当然，这不足以在我的脑海中保持一个高保真模型——跨越太多团队发生的事情太多，无法跟踪。其余的都是有意识地寻求反馈并听取他人的第一手经验。

我还试图帮助将反馈循环融入到我们的组织中。我帮助我们开始进行半年一次的开发团队调查，其中混合了关于我们的技术系统和工程文化的问题。这些调查的回复对于从基层了解组织的感受非常有帮助。

##### 你如何赞助其他工程师？赞助其他工程师是你角色中的一个重要方面吗？

是的。我一直在努力有意识地放弃我的状态，退后一步，让其他人建立专业知识。

在组织层面，我认为有一些方法可以从结构上赞助他人，推动其他工程师担任主题专家。举个例子，去年年底我参与引入了一个新的分布式追踪系统。我们的核心网络应用程序由许多不同的后端系统驱动，多年来，这些系统之间的数据流变得越来越难以理解，页面性能也受到了影响。我们需要一个工具来摆脱困境。

我之前曾参与过我们性能工具的早期迭代，对这些系统的了解很大程度上停留在我的脑海中。在新项目中，我的一个具体目标是让更多人尽快上手。仅仅在系统设计中奠定技术基础是不够的：我的项目团队成员必须在未来几年内掌握该领域的专业知识。

实际上，这意味着我将花费更多的时间与

即将成为追踪系统负责人的人讨论和配对，而直接贡献代码或设计的时间则大大减少。当我们与一个产品团队进行beta测试时，我会推动另一位工程师负责销售宣传，或者弄清楚演示，或者让团队加入。

如今，追踪系统已被广泛使用，并由我们的SRE和可观测性团队全面管理。当时与我合作的人现在是解决性能问题的首选团队。

在更个人的层面上，总有一些小地方我可以帮助推动其他人成为焦点。赞助机会可以从小处着手。特别是如果我和一位职业生涯早期的人一起工作，我可能会建议他们承担更多新系统设计中未知的模块，或者为新的文档撰写草稿，或者在我们的全体会议上演示我们的成果。

这种小小的推动可能就是某人开始行动所需要的全部，但有时我认为它也能很好地转化为指导和配对的机会。有些事情（比如第一次构建幻灯片）在做几次之前可能会感觉难以处理。在那里，赞助和指导 दोनों 方面最终都感觉很有影响力。

而且感觉需要付出一些努力才能达到正确的状态。我们希望培养人们担任更多职务，让他们能够拥有更多自主权并做出系统决策，但我们也希望这些决策的制定方式以及我们努力的方向保持一致。这是一项艰苦的工作——最终需要投入大量的精力，既要避免把关，又要取得令人满意的结果。

##### 你最初是在现在的公司获得Staff工程师（Staff engineer）的头衔。你是作为Staff工程师被雇佣的吗？如果不是，晋升为Staff工程师的过程是怎样的？

当我加入公司时，我们没有IC职称。当我们在2019年初引入职级制度时，我被评为Staff级别。

我认为作为团队的早期工程师，我有一个很大的优势。这段历史对我理解我们过去的决策非常有帮助——了解我们已经遇到的陷阱，并帮助将新项目顺利落地。

在每个成长阶段，我们都会增加更多的人员和管理层级

并且组织运作方式会有一个“重新学习”的时期。随着时间的推移，团队的范围会缩小，只能看到拼图的一小部分。我脑海中拥有大部分的工程历史，这不仅帮助我将各个部门的碎片连接起来，而且让我在与我停止直接合作的组织部门保持个人联系方面抢占了先机。这种广度自然而然地让我能够弄清楚什么对组织最有影响力。

##### 您认为达到资深工程师级别最重要的两三个因素是什么？您加入的公司、您的地理位置或您的教育背景对您的职业道路有何影响？

我的背景有点不太传统——我学的是电气工程而不是计算机科学，而且在完成学位之前就辍学了。这段经历迫使我在经验上更加自学成才，但也让我产生了许多冒名顶替综合症。早期我因为没有合适的资历而在很多软件面试中失败了。在我的职业生涯早期，这种冒名顶替的感觉让我非常想尽可能多地学习，以弥补我害怕自己不知道的东西。

在我辍学前的那个夏天，我在Stripe实习。我记得，也许是透过玫瑰色的眼镜，我感到那里的工程文化让我充满活力：对客户体验的高度关注以及对构建技术来实现这一目标的兴奋。这段经历最终对我希望工作场所的感觉产生了很大的影响。

后来我离开学校后，开始在一家较小的创业公司全职工作，在那里我真的不知道自己在做什么。在那段时间里，业务有点漫无目的，但我很幸运能与有思想、资深的工程师密切合作，他们有指导的爱好。在那里工作最终让我在我想学习的东西上有了很大的灵活性，这对我有好处，但可能对公司不利。

作为最后一点背景，我在高中时曾在电脑夏令营工作过几个夏天，教中小学生基本的计算机知识。这种教学心态无疑让我对人们如何理解和与计算机系统互动有了更多的同情心。

到我加入Samsara时，这些经历让我清楚地知道我想要的工作感觉——成为一名早期员工让我有影响力来塑造那里的方式。

拼图的最后一块是我在Samsara的头三年。我很幸运在那段时间里能与这么多有思想的合作者一起工作。我可以很容易地追溯到我今天拥有的许多工作习惯、思维模式和举止到那些人。我无法想象如果没有他们的影响，我的职业生涯会走到今天这一步。

##### 有一种流行的观点认为，成为 Staff 工程师需要完成一个“Staff 项目”。你是否有过 Staff 项目？如果有，那是什么？

不，我没有指定的Staff项目。回顾过去，这些年来有一些项目可能积累起来相当于一个大型Staff项目，但这不是我们在级别划分中明确谈论的事情。

作为一个概念，我对那种单一关注的项目持怀疑态度，并担心它们可能会让人们陷入[英雄心态](https://lethain.com/doing-it-harder-and-hero-programming/) ，而我们真正想要重视的是能够建设组织而不是承担组织的工程师。我更乐于看到随着时间的推移的迭代改进和持续执行：一种深思熟虑的工程的记录。

我很高兴萨姆萨拉似乎同意这个评估。我们的职业道路文件最终更多地谈论持续执行，而不是单个大型项目。

#### 对于刚开始担任Staff工程师的人，你有什么建议？

我想到了几件事。

*习惯多说话*。我认为高级工程师和技术专家角色之间的一个重大转变最终会集中在人身上：协调相互竞争的优先事项，消除误解，使人们在同一页面上保持一致。即使他们通常没有直接下属，技术专家也在一个系统中工作

既包括技术，也包括人：最大的影响将来自于影响两者。

*尽力不要精疲力竭*。当我过渡到技术专家角色时，我很容易陷入一种心态，即我对发生的一切负责，并且必须将我的注意力分散到太多的事情上。我花了一段时间才意识到，这个角色并不意味着我必须更加努力地参与到所有事情中，而是我需要通过组织中的其他人来指导变革。信任大家，指出问题，并期望他们解决问题。

##### 你有没有考虑过工程管理？如果有，你是如何决定走Staff工程师这条路的？

早在2016年，我记得我和我的经理进行了一次关于追求个人贡献者（IC）还是管理道路的初步对话。当时，我仍然觉得我的职业生涯还处于早期阶段，并希望继续投资于我的核心技术经验。

我每年都会重新评估这个决定，最终得出了同样的结论——我还没有完成在技术方面的亲力亲为。在那段时间里，我所做的很多工作都集中在为公司的人们构建开发体验。这些努力最终促使我做了更多类似技术专家的工作，我自然而然地从那里发展而来。

##### 你从哪些资源（书籍、博客、人物等）中学习？你在该领域的榜样是谁？

我倾向于珍视那些用简单的英语谈论复杂话题的文学作品，包括小说和非小说。

我记得读到小说家[村上春树](https://en.wikipedia.org/wiki/Haruki_Murakami) 先用英语写他的第一部小说，然后再把它翻译回日语，以此来塑造他的表达风格。他指出，“我只能用简单、简短的[英语]句子写作。这意味着，无论我脑海中闪过多么复杂和众多的想法，我都无法尝试将它们按原样写下来。语言必须简单，我的想法必须以一种容易理解的方式表达。”

编写软件是一个完全不同的领域，但感觉这种观点符合我真正重视的关于沟通的某些原则：头脑中有一个理解是一半的战斗——能够表达这种理解同样困难和有价值。

我喜欢阅读那些深入探讨技术领域的博客和论文。这些年来，我经常回顾的一些包括：

[Bob Nystrom的博客文章](http://journal.stuffwithstuff.com/category/language/) 关于编程语言

[Vyacheslav Egorov的博客](https://mrale.ph/) 关于编译器和V8内部原理（Chrome的JS引擎）

[Brandur的](https://brandur.org/articles) 关于各种系统主题  Nelson Elhage的 [意外的二次方](https://accidentallyquadratic.tumblr.com/)

[Vicki Pfau的博客](https://mgba.io/tag/debugging/) 关于开发GameBoy Advance模拟器

[fail0overflow的博客](https://fail0verflow.com/blog/) 以及关于控制台架构和漏洞利用的讨论

[Bungie的工程出版物](http://halo.bungie.net/inside/publications.aspx) 关于构建和制作最初的Halo游戏

作为一个轶事，在我职业生涯的早期，我对编程语言的内部原理产生了浓厚的兴趣，并拿起了一本编译器教科书（“龙书”）来学习。这是一本很难读完的书。也许和教授以及一些同学一起读完是合理的，但对我来说，从阅读中获得一个可行的心理模型确实很困难。后来，我发现了Bob Nystrom的[工艺](https://craftinginterpreters.com/) ，它采用了一种更实用的方法，感觉就像一股清新的空气。

我也是阅读代码库的忠实粉丝。在我职业生涯的早期，我记得调试一个棘手的React问题，其中一些回调没有按照我预期的顺序发生。阅读文档没有帮助。放入打印语句也不够。我当时的导师让我阅读一些源代码，以便更好地理解发生了什么，这真的让我大开眼界。我不仅修复了一个错误，而且对React的工作原理有了更深入的了解。

那真是一个转折点。能够快速深入并跳过不熟悉的代码，真的感觉像是一种超能力，并为我提供了更大的模式匹配库，用于软件设计的不同方法。最近最喜欢的是阅读[esbuild](https://github.com/evanw/esbuild/blob/master/docs/architecture.md)的设计和代码，这是一个超快的javascript

打包器。

[BART 的历史，这个施乐帕克研究中心的历史，和概述 现代日本文化概述。在每个领域，我都发现了历史和背景](https://www.amazon.com/Making-Common-Policy-Institutional-Studies/dp/0822955105)

# 结束语

****

通常，当一群Staff-plus工程师坐下来聊天时，他们会花时间对行业中糟糕的技术领导力发展道路表示同情。这是真的；它确实很糟糕。但不真实的是，这是不可避免的。行业对技术领导力的看法是每个人在Staff-plus工程师角色中运作、渴望或管理时所采取的集体日常行动的总和。通过改变我们的习惯和方法，我们将改变这个行业。

通过提升自己，我们让行业变得更好。制定一个计划，赞助你组织中的其他人。综合一份战略文件，花时间让其他人感到被倾听。建立一个同行网络，并邀请更多人加入这个网络。当你做这些工作时，你的工程组织将从你的榜样中学习。

你的经理和管理链也将从你的榜样中学习。许多经理从未与有效的Staff-plus工程师合作过。相反，他们通过之前的灾难来看待这个角色。与他们合作，了解他们的优先事项和顾虑，以建立一致性。让自己承担与你的经理相同的压力。克服他们对可能出错的恐惧，并为可能做对的事情创造兴奋感。

谈论害怕Staff-plus角色是否感觉很奇怪？确实如此，但这是不可避免的。在大多数公司，阻止采用有影响力的Staff-plus角色的主要因素是害怕创造一个专制的架构师。这种恐惧是由他们之前与无能和管理不善的技术领导者的经历所驱动的。只有当我们集体补充

良好的Staff-plus榜样的地下水时，这种恐惧才会消退。

我希望阅读《Staff Engineer》能给你一张地图，让你朝着有回报的Staff-plus职业生涯前进，或者激励你继续走你已经找到的道路。软件技术行业仍处于起步阶段，它将变成什么完全掌握在我们手中。

# 资源

****

## 关于Staff-plus工程的更多资源

我采访过的所有一级工程师都不是孤军奋战。他们中的大多数人要么通过贪婪地阅读，要么通过建立强大的同事网络来实现这一目标。本节是推荐资源的集合。

### 你的网络

几乎一致地，Staff-plus工程师最有价值的学习资源不是书籍、博客、演讲或论文，而是他们的同行和导师网络。如果你只有一个小时来提升自己作为工程师的能力，那么最好的选择是建立一个与你角色相似的人脉网络。[b](#_bookmark17)[.](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

如果你正在寻找一个Slack社区，那么在Rands Leadership Slack中的#staff-principal-engineering是一个相当活跃的房间。[#staff-principal-engineering在](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)[R](https://randsinrepose.com/welcome-to-rands-leadership-slack/) [是一个相当热闹的房间。](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

### Staff-plus工程师做什么？

[是 通过](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)[定义卓越工程师 杰西·弗拉泽尔 著](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[我在 Heroku 如何以 Staff 工程师的身份工作 艾米·昂格 著](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[否 通过](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)[与塔尼娅·赖利一起了解细节](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[关于 通过](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)[关于成为高级工程师 约翰·奥尔斯帕夫 著](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[Staff 工程 萨姆·克莱曼 著](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[蓬勃发展 通过](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)[高级工程师的工作是什么？ 作者：Julia Evans](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[What（什么） 我的任务 和第二部分：角色的心态和关注点 作者：Joy Ebertz](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

[在 GitLab，Staff 级别意味着什么？ 作者：Charlie Ablett](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

### 成为一名资深工程师

[成长之路](https://elpha.com/posts/4j56np6p/becoming-a-staff-engineer-interview-with-kristina-fox-staff-ios-engineer-at-intuit)

Intuit工程师 作者：Kaya Thomas

[关于](https://blog.coinbase.com/on-becoming-a-senior-technical-leader-14106f1383b8)作者：Jesse Pollak[关于职业中期和管理者](https://www.ryn.works/blog/on-mid-career-and-managers)作者：Ryn Daniels

[如何](https://www.quora.com/How-does-one-become-a-Staff-Software-Engineer-at-Google-What-might-a-new-grad-entering-the-company-do-to-grow-their-career-to-reach-that-level)[工程师/管理者钟摆](https://charity.wtf/2017/05/11/the-engineer-manager-pendulum/)

[关于工程级别需要了解的事情](https://charity.wtf/2020/09/14/useful-things-to-know-about-engineering-levels/)作者：Charity Majors

### 以 Staff-plus 工程师的身份运作

[成为粘合剂](https://noidea.dog/glue)作者：Tanya Reilly

[计算机是可以理解的](https://blog.nelhage.com/post/computers-can-be-understood/)作者：尼尔森·埃尔哈格

[代码和系统的有效思维模型](https://medium.com/%40copyconstruct/effective-mental-models-for-code-and-systems-7c55918f1b3e)

[“我不会从这里开始。” 如何进行重大的技术变革](https://noidea.dog/blog/getting-there-from-here)

[迁移：](https://lethain.com/migrations/)[关于职业中期和团队动态](https://www.ryn.works/blog/on-mid-career-and-team-dynamics)

[在组织支线任务中生存](https://noidea.dog/blog/surviving-the-organisational-side-quest)

[系统](https://blog.nelhage.com/post/systems-that-defy-understanding/)[团队目标](https://svpg.com/team-objectives-overview/)

[技术决策](https://medium.com/%40copyconstruct/technical-decision-making-9b2817c18da4)

[技术研究与准备](https://keavy.com/work/technical-preparation/)

[这个](https://blog.coleadership.com/behind-the-scenes-tech-leadership/)

[理解项目管理将提升你的开发者工作](https://blog.danielna.com/understanding-project-management-will-improve-your-developer-job/)

[What（什么）](https://larahogan.me/blog/what-sponsorship-looks-like/)[从哪里开始](https://keavy.com/work/where-to-start/)

[设计文档、Markdown 和 Git](https://caitiem.com/2020/03/29/design-docs-markdown-and-git/) 作者：Caitie McCaffrey

### 技术规范

[A](https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs/)  [谷歌的设计文档](https://www.industrialempathy.com/posts/design-docs-at-google/)

[设计文档、Markdown 和 Git](https://caitiem.com/2020/03/29/design-docs-markdown-and-git/)

[记录架构决策](https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions)

[如何编写更好的技术设计文档](https://www.range.co/blog/better-tech-specs)

[远程文化中的技术决策和协调](https://multithreaded.stitchfix.com/blog/2020/12/07/remote-decision-making/)

[编写技术设计文档](https://medium.com/machine-words/writing-technical-design-docs-71f446e42f2e)

### 工程战略

[A](https://multithreaded.stitchfix.com/blog/2019/08/19/framework-for-responsible-innovation/)

[如何](https://slack.engineering/how-big-technical-changes-happen-at-slack-f1569d25ee7b)[关于起草工程战略](https://www.paperplanes.de/2020/1/31/on-drafting-an-engineering-strategy.html)

[定义技术战略](https://sarahtaraporewalla.com/agile/design/architecture/Defining-a-Tech-Strategy)

[交付](https://blog.thepete.net/blog/2019/12/09/delivering-on-an-architecture-strategy/)  [垫脚石而非里程碑](https://medium.com/%40jamesacowling/stepping-stones-not-milestones-e6be0073563f)

[通过技术战略实现一致性和效率](https://yenkel.dev/posts/achieving-alignment-and-efficiency-through-a-technical-strategy)

[这个](https://medium.com/ft-product-technology/the-difficult-teenage-years-setting-tech-strategy-after-a-launch-7f42eb94a424) 作者：安娜·希普曼

[学习拥有工程愿景](https://unwiredcouch.com/2018/01/03/engineering-vision.html)

工程战略的例子：

[减少运行的软件](https://www.intercom.com/blog/run-less-software/) 作者：里奇·阿奇博尔德

关于战略的其他方面也有很多很棒的资源，例如，马蒂·卡根关于[产品策略](https://svpg.com/product-strategy-overview/)的系列文章。

### 书籍

虽然我发现很多人不太读书，但当我向资深工程师询问他们最有价值的资源时，他们总是会提到一位私人导师或一本书。他们可能会提到与更具体的问题相关的博客文章和技术讲座，但他们受这种更大、更纸质的形式的影响最大。

一些被推荐的书籍：

[A](https://lethain.com/notes-philosophy-software-design/)作者：John Ousterhout

[加速： 和 扩展 高 执行 技术 组织 作者：Forsgren、Humble 和 Kim](https://www.amazon.com/dp/B07B9F83WM/)

[成为技术领导者：一种有机的解决问题方法](https://www.amazon.com/dp/B004J4VV3I/)作者：Gerald Weinberg

[构建进化架构](https://lethain.com/building-evolutionary-architectures/)作者：Ford, Parsons, and Kua

[逃离 价值 作者：梅丽莎·佩里](https://www.amazon.com/dp/B07K3QBWG1/)

[好的](https://www.amazon.com/dp/B004J4WKEC)作者：Richard Rumelt

[高产出管理](https://www.amazon.com/dp/B015VACHOK/)作者：Andy Grove

[这个 改变 作者：卡米尔·富尼耶](https://www.amazon.com/dp/B06XP3GJ7F/)

[人月神话](https://www.amazon.com/dp/0201835959/)作者：Fred Brooks

[这个](https://www.amazon.com/dp/B078Y98RG8/)作者：Kim, Behr, 和 Spafford。[充满激情的程序员](https://www.amazon.com/dp/B00AYQNR5U/)作者：Chad Fowler

[这个](https://www.amazon.com/dp/020161622X) 作者：Hunt 和 Thomas [弹性管理](https://resilient-management.com/) 作者：Lara Hogan

[软件 分析 作者：亚当·托恩希尔](https://www.amazon.com/dp/B07BVRLZ87)

[系统之美：A](https://www.amazon.com/dp/1603580557) 作者：Donella Meadows

如果你正在寻找更多推荐书单，包括我在 Irrational Exuberance 的最佳书籍中的书单。[非理性繁荣的最佳书籍](https://lethain.com/best-books)

### 演讲

我聊过的 Staff-plus 工程师通常认为，与听演讲相比，发表演讲对他们更有价值，但当然也有一些很棒的演讲。辛迪·斯里达兰（Twitter）是精彩演讲的最佳来源，特别是她撰写的 2019 年科技演讲精选、2018 年科技演讲精选和 2017 年科技演讲精选。[辛迪](https://medium.com/%40copyconstruct)[Twitter](https://twitter.com/copyconstruct)[2019年科技演讲精选, 2018年最佳 科技演讲, 和](https://medium.com/%40copyconstruct/best-of-2018-in-tech-talks-2970eb3097af)[2017年科技演讲精选](https://medium.com/%40copyconstruct/best-of-2017-in-tech-talks-8f78b34ff0b)[.](https://medium.com/%40copyconstruct/best-of-2018-in-tech-talks-2970eb3097af)

### 论文

相对而言，很少有 Staff-plus 工程师是计算机科学论文的狂热读者。然而，大多数人都熟悉一些基础论文，而少数花时间阅读论文的人往往从中获益匪浅。

如果你渴望加入经常阅读论文的行列，没有比 Adrian Colyer 的 the morning paper 更好的地方了，它会在每个工作日向你发送一篇计算机科学论文的摘要。如果你更有兴趣了解一些著名论文的基础知识，首先阅读 Peter Klein 的如何阅读学术文章或 S. Keshav 的如何阅读论文，然后阅读以下推荐论文列表：

[这](https://blog.acolyer.org/)地方比 Adrian Colyer 的 the morning paper，它会在每个工作日向你发送一篇计算机科学论文的摘要。如果你更有兴趣了解一些著名论文的基础知识，首先阅读 Peter Klein 的 How to Read an Academic Article 或 How to Read a Paper by S. Keshav，然后阅读以下推荐论文列表：[如何 如何阅读学术文章 彼得·克莱因著或](https://organizationsandmarkets.com/2010/08/31/how-to-read-an-academic-article/)[如何阅读论文](https://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf)[由 S. 提供](https://organizationsandmarkets.com/2010/08/31/how-to-read-an-academic-article/)

[Dynamo：](https://s3.amazonaws.com/systemsandpapers/papers/amazon-dynamo-sosp2007.pdf)  [关于设计和部署互联网规模的服务](https://s3.amazonaws.com/systemsandpapers/papers/hamilton.pdf)

[否](https://s3.amazonaws.com/systemsandpapers/papers/Frederick_Brooks_87-No_Silver_Bullet_Essence_and_Accidents_of_Software_Engineering.pdf)  [走出泥潭](https://s3.amazonaws.com/systemsandpapers/papers/outofthetarpit.pdf)

[这个](https://s3.amazonaws.com/systemsandpapers/papers/chubby-osdi06.pdf)  [Bigtable：A](https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf)

[Raft：](https://s3.amazonaws.com/systemsandpapers/papers/raft.pdf)  [Paxos 极简版](https://s3.amazonaws.com/systemsandpapers/papers/paxos-made-simple.pdf)

[SWIM： 弱一致性 感染式 过程 组 成员协议](https://s3.amazonaws.com/systemsandpapers/papers/swim.pdf)

[提示](https://s3.amazonaws.com/systemsandpapers/papers/acrobat-17.pdf)  [一团乱麻](https://s3.amazonaws.com/systemsandpapers/papers/bigballofmud.pdf)

[谷歌文件系统](https://s3.amazonaws.com/systemsandpapers/papers/gfs.pdf)

[CAP](https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed)  [收获、产量和可扩展的容错系统](https://s3.amazonaws.com/systemsandpapers/papers/FOX_Brewer_99-Harvest_Yield_and_Scalable_Tolerant_Systems.pdf)

[MapReduce：大型集群上的简化数据处理](https://s3.amazonaws.com/systemsandpapers/papers/mapreduce.pdf)

[Dapper，](https://s3.amazonaws.com/systemsandpapers/papers/dapper.pdf)  [Kafka：用于日志处理的分布式消息系统](https://s3.amazonaws.com/systemsandpapers/papers/Kafka.pdf)

[谷歌使用 Borg 进行的大规模集群管理](https://s3.amazonaws.com/systemsandpapers/papers/borg.pdf)

[Mesos：A](https://s3.amazonaws.com/systemsandpapers/papers/mesos.pdf)

可能找到高质量论文的最佳地点是Papers We Love，他们也会举办聚会来讨论论文。其他一些资源包括ACM SIGOPS名人堂奖项列表和Irrational Exuberance的论文集。[我们热爱的论文](https://paperswelove.org/)[美国计算机协会 SIGOPS 名人堂奖项列表 和](https://www.sigops.org/awards/hof/)[非理性繁荣的论文集](https://lethain.com/some-of-my-favorite-technical-papers/)[.](https://www.sigops.org/awards/hof/)

### 其他好东西

在为这些资源进行研究时，我发现了一些其他内容，它们不太适合放在上面的任何地方，但我认为它们很好，值得一看，尽管如此：

[测试 和测试 零件 作者：Cindy Sridharan](https://medium.com/%40copyconstruct/testing-in-production-the-hard-parts-3f06cefaf592)

[A](https://medium.com/%40copyconstruct/a-decade-in-review-in-tech-1cde76c9b43c)[虚构的怪物问题](https://blog.danielna.com/boogeyman-problems/)

如果您发现更多，请发给我！

## Staff-plus工程师在组织中扮演什么角色？

当我在设计工程组织的组织结构时，我会考虑很多“[组织数学](https://lethain.com/sizing-engineering-teams/)”，即每个团队应该有一名经理和六到八名工程师，以及每个经理的经理应该支持四到六名经理的指导原则。从这些数字中，您可以快速确定一个适用于您的组织的适当结构，并且效果相当好。它可能不是*完美的*，但它会*奏效*。

当我在设计多个组织时应用这种方法时，反复出现的边缘案例之一是决定资深工程师应该向谁汇报。他们应该像组织数学所规定的那样，向组织叶节点中的经理汇报吗？或者，作为您组织中的关键领导者，他们应该向更高级别的领导者汇报，以便更好地获取他们所需的的信息和授权，从而在他们的角色中脱颖而出？

在回答之前，值得描述一下你今天在公司里能找到的最常见的配置，特别是配置在员工级别以上的原型之间的差异：[员工- 加上原型:](#_bookmark6)

技术负责人通常向负责一个团队的经理汇报。较少情况下，他们会向负责两到四个团队的经理汇报。在这两种情况下，他们将在与该经理相同的范围内运作。例子：Dan Na 向国际化平台经理汇报。[丹·娜](#_bookmark39)

架构师通常向更高级别的经理汇报，通常是经理的经理。他们通常负责该经理职责范围内的横向切片，例如，数据建模。例子：Keavy McMinn 向首席技术官汇报。[凯维·麦克明](#_bookmark32)

问题解决者通常在具有“调整团队概念”的公司中运作，并且在这些公司中，报告层级通常不太明确或经过深思熟虑。最常见的是看到他们向团队经理汇报，但你会发现各种情况。另一种常见的模式是将这些人聚集到“首席技术官办公室”或“首席执行官办公室”，在那里他们向指导他们工作的执行官汇报。例子：Ritu Vincent 是一个向首席执行官汇报的孵化器的一部分。[弱](https://lethain.com/weak-and-strong-team-concepts/)[里图](#_bookmark35)

右手型员工向高级领导汇报，通常是负责一百人或更多人的经理，并以该领导的权力运作。例子：Rick Boone 向基础设施副总裁汇报；Michelle Bu 向首席产品官汇报。[里克·布恩](#_bookmark36)[首席产品官](#_bookmark30)

了解这些不同的原型通常如何在组织中以不同的方式汇报，有助于解读报告结构中看似随意的一些情况。

### 首席技术官办公室

关于“首席技术官办公室”概念的题外话，因为很多人在工作中没有遇到过。通常，首席技术官（有时首席执行官也会做类似的事情）会有两到八名直接向他们汇报的员工级别以上的工程师。这些人被视为高级领导，因为他们会获得一个问题或机会去追求，几乎没有管理支持，并且在必要时可以依靠高管的支持。

在这些办公室里，你会发现架构师、问题解决者和右手型员工的混合。

通常，首席技术官办公室在公司发展的后期才会出现，并且通常是作为解决现有组织问题的变通方法而引入的，这些问题很难解决，例如，员工级别以上的工程师和管理团队之间缺乏信任，或者首席技术官无法授权。如果你发现自己在公司发展的早期就想使用它，请问问自己是否在逃避一个你应该解决的问题，而不是将这个概念引入你的结构中。

### 但在实践中……

基于原型，通常每个工程师都有一个理论上正确的位置可以融入组织，但你会发现，在实践中，很少有组织能够完全将其实际报告结构与理论结构对齐。

有时这是由于您的管理团队对组织结构缺乏关注。在其他情况下，这是由于缺乏管理带宽来

支持处于正确位置的人员。例如，“正确”的经理已经在管理一个由 12 人组成的团队，无法有效地支持另一位工程师。另一种常见的情况是，结构变化过于频繁，以至于经理们不愿再次更换工程师的经理，尤其是因为经理变更通常会导致抽象的、折衷的绩效评估。

如果你发现自己向你认为不合适的经理汇报工作，那么与你的经理进行一次合理的谈话是可行的，但值得承认的是，许多经理会对暗示他们不适合你的经理做出防御反应。如果你的经理很成熟，并且你与他们关系良好，那就继续进行对话。如果不是，那么与他们的越级领导进行一次更抽象的讨论，了解 Staff-plus 工程师在他们的组织中向谁汇报，可能风险更小。

在你急于倡导变革之前，先问问自己，如果你的报告结构发生变化，你认为会有什么不同。报告结构是一种权威形式，通常，人们会高估权威对他们的帮助。当然，一个典型的陷阱是，那些从额外权威中获益最多的人——少数族裔和女性——是那些经理最有可能对变革的建议做出防御反应的人。

### 它应该如何运作？

如果你正在为你的管理团队设计一个关于他们应该如何随着时间的推移进行这类组织调整的提案，那么有几件事需要考虑。如果可能，**报告变更应立即发生。**延迟会迫使人们经历两次过渡，包括先前角色和新角色之间一个特别具有挑战性的中间环境。经历一次角色过渡的风险总是较低的，即使这意味着如果你的新经理因工作量过大而无法提供支持，你也同意减少经理的支持。

如果你不能立即进行更改，**始终为角色更改后向正确的经理汇报工作设置一个时间表**。如果你没有明确重新开放结构的时间框架，那么这种情况发生的可能性相对较小。

大多数公司都在努力建立这种组织基础设施，以充分支持 Staff-plus 工程师作为真正的领导者，因此你应该将此视为一个你希望在*多年内*取得进展的问题，而不是一个可以一夜之间解决的问题。如果你期望一个立即的、永久的解决方案，你可能会遇到一些动荡。

## 管理 Staff-plus 工程师

在获得关于[*StaffEng（高级工程师）*](http://staffeng.com/)的反馈时，有人要求提供更多关于管理Staff-plus工程师的内容。这不太符合主题——该努力的重点是Staff工程师本身，而不是公司或经理——但这是一个有趣的话题，也是一个值得的附录。

当然，管理Staff-plus人员的各个方面并非都仅限于该级别：有一些基本原则适用于管理任何角色的任何人，例如进行 有效的1对1沟通 或 给予反馈。对于这类事情，请阅读Lara Hogan的《弹性管理》 或Camille Fournier的《经理之路》。我在这里想探讨的是，管理Staff-plus级别的人与管理高级工程师有何不同。[做 有效的 1 对 1 会议 或](https://marcorogers.com/blog/my-approach-to-1-on-1s)[给予反馈](https://smallbigideas.substack.com/p/own-your-feedback-part-1)[关于这类事情，请阅读劳拉·霍根的](https://marcorogers.com/blog/my-approach-to-1-on-1s)[弹性管理](https://resilient-management.com/)[经理的道路](https://www.oreilly.com/library/view/the-managers-path/9781491973882/)

这些角色在不同的公司之间差异很大，因此管理Staff-plus工程师的某些方面将取决于[员工原型](#_bookmark6) 贵公司强调的内容以及[你的 Staff-plus 工程师应该如何融入工程 组织，但有些方法会对你有所帮助](#_bookmark47)

**赞助和支持多于指导。**如果你每天都在指导你的Staff工程师，那么你就是在错误的职位上使用他们。如果你没有每周给他们反馈，那么你就是在延缓他们的成长。如果你没有为他们的倡议提供赞助，那么你就会把他们的主动性训练出来。

**帮助他们重新定义成功的定义。**在一个高效的产品工程团队中工作是一个积极反馈的飞轮。你的产品经理欣赏你的工作。你的工程经理与团队互动。你的同事喜欢一起工作。你的用户喜欢你的产品。你的企业喜欢用户的采纳。相反，Staff工程师的反馈飞轮就不那么直接了。你花更多的时间来解决冲突。你工作的时间跨度更长。你代表着重要的优先级，需要降低一些业务或产品目标。很多人没有解决这种转变，一年后醒来讨厌他们的新角色，作为他们的经理，你可以帮助他们

认识到这种转变并找到补偿策略[到](#_bookmark30)。

**给予反馈。**重写他们对成功的定义——并保持他们成长——的一个特别重要的策略是给予频繁的反馈。如果他们选择了错误的战斗，告诉他们，并告诉他们*为什么*。如果他们优先考虑你不会做的工作，告诉他们，并告诉他们*为什么*。对于一个高绩效者来说，没有什么比不知道自己做得怎么样更让人感到压力的了！如果你不给予反馈，特别是关于他们最好的工作，他们会不断改变他们的方法，直到你给予反馈（通常让你后悔）。

**让他们知情**。作为一名经理，很容易忘记你比与你一起工作的工程师拥有更多的信息。现实情况是，大多数组织围绕着经理向其他经理传达关键信息来构建他们的信息流。如果你没有找到一种有意的、可重复的过程来与他们分享你的背景信息，你的Staff-plus工程师将会受到阻碍。有些人会在他们的一对一会议开始时这样做，这还可以，但我更喜欢在团队的聊天频道中随时发布，并将它们汇总到我的[每周电子邮件更新](https://lethain.com/weekly-updates/)中。

**让他们参与计划和优先级排序。**许多工程师对“正确的工作永远无法优先考虑”感到沮丧，解决这个问题最好的方法之一是积极地让更多的工程师参与到计划过程中。这在两个方面起作用。首先，他们更了解竞争的工作以及为什么这项工作很重要，其次，他们将能够更有效地倡导他们认为缺失的技术工作。

**就如何在独立行动的同时保持一致达成一致。**当你推动你支持的Staff-plus工程师走向领导岗位时，他们将开始更多地领导，这有时会让你感到惊讶。如果你合作的领导者*从不*让你感到惊讶，那么你授权得不够，但如果他们经常让你感到惊讶，那么[明确建立你的 控制.](https://lethain.com/identify-your-controls/)

**为他们创造思考的空间，同时又不会让他们脱离组织日常的实际情况。**许多担任这些角色的人都受到影响和“为企业做正确的事情”的强烈驱动，以至于他们会不顾一切地工作，而没有外部干预。如果你是他们的经理，那么“外部干预”就意味着你。如果你看到他们花费

花费太多时间救火和帮助解决紧急工作，与他们合作，为深度思考工作争取更多时间。相反，如果你发现他们只做深度思考工作，他们可能会失去对情况的了解，如果他们不调整这种组合，可能会失去同事和企业的尊重。

**提醒他们，他们是榜样。**就像他们对管理者所做的那样，组织中的工程师会观察Staff+工程师，以了解哪些行为和行动会得到奖励（以及容忍）。这是一项重大的责任，但也是一个巨大的影响力机会：通过践行积极的价值观，他们有机会在周围创建一个积极的组织。

[**尽量减少经理的溢出。**在追求效率高于 效能许多公司让他们的经理们陷入了惊人的数量之中](https://www.amazon.com/dp/B004SOVC2Y/ref%3Ddp-kindle-redirect?_encoding=UTF8&btkr=1)

**给他们未经提炼的问题。**这是一个高级职位，你应该给他们一个问题空间，让他们将其缩小为一个更具体的问题和解决方案。他们比你拥有更好的技术背景，如果你过于精确地指出你认为的问题是什么，你就无法从他们的判断中受益。精确地选择正确的问题至少与找到精确的正确解决方案一样重要，并且只有在你创造空间时才有可能。

**将领导空间让给他们。**当你管理Staff+工程师时，找到方法将你所有权的一部分明确地转移到他们的职责范围内。例如，你如何让他们对团队的技术质量负责，而不是你来做？这为你和Staff+工程师创造了杠杆作用和所有权感。

**欣赏他们。**优秀的Staff+工程师运作相当独立，因此当组织陷入困境时，很容易降低他们的优先级。忽视你最重要的人[是](#_bookmark10)——感觉很重要，但通常不是正确的优先事项。所以保持

你的1对1会议，并且通常记得支持他们，特别是如果他们不是那种会要求的人。

**建立并坚持与业务保持一致。**有些工程师尽管怀有一种技术工作比需要该工作的业务更重要的心态，但仍然取得了成功。这种心态通常是有害的，但当Staff+工程师持有这种心态时，尤其有害。这是一个更广泛组织的榜样，扩展他们的视野对于他们继续担任领导角色至关重要。与业务不一致的领导者会被公司削弱并最终被驱逐。

**让他们对整个角色负责。**虽然很少有人在技术方面存在重大弱点的情况下达到Staff+的角色，但我的亲身经历是，许多人在领导或行为方面存在重大挑战的情况下达到了这些角色。这些人获得了头衔，但往往徘徊在Staff的炼狱中，在那里他们被期望领导，但却被排除在大多数领导机会之外。他们被认为太不可靠或“参与成本太高”。你必须就他们的差距向这些人提供反馈，并让他们对整个角色的期望负责。不要让他们无限期地作为准领导者徘徊。也许他们最初是通过[职位膨胀](https://charity.wtf/2020/11/01/questionable-advice-the-trap-of-the-premature-senior/)获得了这个角色，所以你决定只是掩盖他们的差距而不是解决它——不要这样做，相反，制定一个计划来支持他们[同时转移了该责任 给他们.](https://hbr.org/1999/11/management-time-whos-got-the-monkey)

**让他们进入会议室，但不要将其视为一种地位象征。**人们经常沉迷于地位象征，而工程师特别关注的一个地位象征是“[在场](#_bookmark23)”。有时会议*是*工作发生的地方，但大多数例行报告会议都有太多人参加，你可以通过在各种会议中分摊出席人数，而不是为所有会议加倍，为你和支持的Staff+工程师创造大量的时间和空间。

从高级工程师到资深工程师的转变是一个重大的转变，它改变了所做工作的类型，而之前的转变通常只改变工作的范围。很多人都在努力适应这种转变，很多管理者也不确定如何帮助支持与他们合作的资深工程师。当然，这并不是一份完整的可以帮助支持他们的事项清单，但希望它能成为一个有用的起点。

## 设计资深工程师面试流程

当我们谈论设计资深工程师的面试流程时，首先要说的是，绝对没有人对他们的资深工程师面试流程充满信心。许多流程最终都在寻找一位解决问题*非常*快的资深工程师，但这并不能反映实际的角色。另一些流程则侧重于沟通技巧，这*是*该角色的一个关键部分，但肯定不是全部。一些公司甚至构建他们的流程来评估候选人是否*感觉*像他们现有高级工程团队的成员，将卓越与熟悉混为一谈。

即使没有人对他们的流程感到满意，仍然有一些集体经验可以纳入你设计流程的尝试中，这就是我们在这里要讨论的内容。我们将首先检查资深工程师面试流程中常见的失败模式，讨论你*确实*想在这些流程中测试的信号，最后讨论一些可用于评估这些信号的面试形式。

### 挑战

虽然一般的技术面试有点混乱，但对非常资深的候选人的面试本身就存在足够多的问题，需要专门解决。在考虑什么可能更有效之前，首先了解一些常见的失败模式是有帮助的：

**一个更优秀的高级工程师。**许多面试流程将资深工程师视为在所有方面都更优秀的高级工程师。他们速度更快。他们的沟通更清晰。他们在架构讨论中更细致。这源于大多数人对资深角色的不熟悉，并且通常会导致资深工程师在这些流程中表现不佳。特别是，大多数资深工程师的编程量少于高级工程师，因此在死记硬背的编程任务中，他们*更慢*而不是*更快*。你会发现一些资深工程师仍然是非常快速的程序员，但你不会发现这种速度与他们的影响力之间有太大的关联。

**一个更差劲的高级工程师。**相反，其他面试流程认识到资深工程师花费在编程上的时间较少，并且预计在一些更平凡的编程练习中表现会较慢。这使得资深工程师更有可能在流程中取得成功，但并不能获得关于是什么让这些工程师具有非凡影响力的信号。如果你不添加额外的面试来捕捉这些优势，那么在死记硬背工作上的速度降低可能会在一定程度上与资历相关，但它与其他许多因素的相关性更强。

**一个高级工程师，但他们会接受这份工作。**另一种失败模式是

那些努力招聘高级工程师的公司，决定在不改变角色期望的情况下抬高他们的头衔。在这些情况下，面试适合实际工作，但头衔不适合。由于公司和候选人都不完全愿意承认这种膨胀，该公司未来所有资深工程师的招聘都带有一层不确定性。

**像我们这样的人。**许多评估都侧重于Staff-plus候选人是否*表现出*与公司现有Staff-plus工程师相同的智慧和信心，面试汇报可能包括诸如“他们感觉会自然而然地成为团队的一份子”之类的陈述。这种方法更有可能锚定在半随意的特征上，比如他们如何展现自己的信心，而不是候选人的能力。

**不如我。**尤其是在招聘你的第一批高级工程师时，你经常会发现一些早期职业的面试官低估了候选人的优势，反而将重点放在候选人是否有能力胜任面试官目前的角色上。你可能会遇到一位令人印象深刻的高级候选人，但面试小组会怀疑他们是否有能力作为一名中级工程师茁壮成长。这种情况似乎最常发生在女性和少数族裔候选人身上。

**反向筛选。**某些类型的面试表明，作为一个组织，你不知道如何使用高级工程师：白板算法面试，面试官主要处于职业生涯早期，等等。许多高级流程导致最优秀的候选人在流程早期选择退出，这通常对招聘指标来说是有些隐形的。

**过于注重头衔？**在达到一定成就水平后，人们不太关心内部级别，通常是因为他们已经经济独立。这给新晋升到高级职位的人带来了一种特殊的压力

级别，以避免显得过于注重职业或头衔，这使得人们更难首次获得该头衔。

其中一些问题很难解决，另一些问题只要你记住它们就很容易解决，但所有这些问题都值得你在开始设计或重新构想你的高级面试流程时考虑。

### 信号

[最佳面试流程](https://lethain.com/designing-interview-loops/)最好的面试流程从你想要捕捉的信号出发，反向推导出面试的主题和形式，这意味着首先要回答的重要问题是：“哪些信号对于招聘成功的高级工程师来说是重要的？”

我建议关注的信号是：

**自我意识**。他们是否对错误负责？他们是否在以前较弱的领域表现出成长？

**判断力。**他们是否能够看到问题的各个方面来识别问题？他们是否能够驾驭广泛、模糊的问题？他们能否有效地调解争论权衡或设计的双方？他们能否降低执行难题的风险？

**协作。**他们是否能与他人良好合作？与经验不如他们的人呢？比他们更有经验的人呢？他们的经理呢？跨职能部门呢？高管呢？

**沟通。**他们是否是优秀的倾听者，能够理解他人提出的观点？他们是否能够清晰地表达自己的想法？他们是否能够以贵公司依赖的格式进行沟通（书面、口头等）？

**发展。**他们是否能促进周围人的成长？他们领导的领域的“组织人才储备”是增长还是萎缩？破损的系统和流程是否得到清理？

有趣的是，许多人不会认为这些是明确的技术技能。领域专业知识是他们成功的关键因素，但这是结合其他关键技能来运用这些专业知识，并且

这些行为将某人从一位资深的资深工程师转变为一名 Staff-plus one。

### 格式和结构

在设计面试流程时，需要问自己的两个关键问题始终是：

1. 这个人需要哪些任务和行为才能在日常工作中取得成功？
2. 我们如何让他们展示实际操作这些任务和行为？

大多数高级候选人变得越来越有外交手腕，*询问*他们关于工作的事情，永远不如*观察*他们做工作那么有帮助。如果指导是最重要的活动，不要依赖他们谈论指导，而是想办法*看到*他们指导别人。如果是架构，展示你当前的系统，并让他们提出问题，看看他们如何对他们不同意的决定做出反应——摆脱模棱两可的抽象。

除了典型的一对一讨论和编程面试之外，我发现以下一些面试形式和结构对于评估高级工程师的信号特别有效：

**结构化展示。**让候选人准备一个关于一个狭窄主题的20到30分钟的演讲，并向一群同行展示。这种形式在结构化思维、沟通、倾听和回答问题方面给出了很好的信号。根据你选择的主题，你可以获得关于你选择的一个或两个领域的强烈信号。这种形式在了解人们如何谈论他们的同行和同事方面尤其有效。

**代码审查。**准备一个拉取请求，并要求候选人提供反馈，重点关注同理心、清晰度和实用性。

**数据建模、接口和架构。**让候选人逐步完成一个系统的设计，通常侧重于使其不断演进以满足不断变化的需求。这些面试通常试图做太多事情：缩小你的关注范围，并在问题中添加层次，以便你可以继续深入挖掘那些取得重大进展的候选人。

**主题专业知识。**测试他们领域专业知识的面试。例如，前端工程师可能会与设计师和产品经理合作，讨论技术约束将如何影响拟议的设计和发布时间表。对于后端工程师，你可以为候选人提供一个损坏的软件或环境，让他们调试问题并找到修复方案。

**指导小组**。如果你的小组完全由职业生涯早期的员工组成，那么招聘Staff-plus候选人是具有挑战性的，但如果Staff-plus候选人没有展示出指导职业生涯早期员工的成功经验，那么招聘他们同样具有风险。让三到四个他们可能需要指导的人组成一个小组，并提出问题。观察人们如何将大致成型的框架问题转化为有用的讨论，可以特别深入地了解他们在新的角色中指导他人的能力。

如果这些形式还不够，那就开始四处打听！大多数公司都为他们的Staff-plus流程设计了定制的方法，你可以从讨论中学到很多东西。

### 如何整合

你可能会理所当然地认为，这篇文章会以一个预先编译好的面试流程结束，供你的公司用来评估Staff-plus工程师，我很遗憾让你失望了，但我认为大部分价值来自于思考对你来说重要的信号，并设计出能够以一种与你和你的公司产生共鸣的方式获得这些信号的形式。

无论你最终使用什么面试方式，都要测试它们，收集候选人的反馈，并不断改进它们，让它们变得更好！

## Staff-plus职业阶梯

现在公开分享了如此多不同的职业阶梯，所以在尝试设计自己的职业阶梯之前，没有理由不阅读六个不同的职业阶梯。

以下是一些特别值得阅读的：

[租金](https://docs.google.com/spreadsheets/d/1k4sO6pyCl_YYnf0PAXSBcX776rNcTjSOqDxZ5SDty-4/edit#gid%3D0)  [Kickstarter（众筹平台）](https://gist.github.com/jamtur01/aef437a79fee5a9cefdc#junioreng)

[Patreon（会员赞助平台）](https://levels.patreon.com/)

你可以在[progression.fyi](https://www.progression.fyi/)找到更多。Charity Majors 也写了一份关于[关于工程级别需要了解的事情](https://charity.wtf/2020/09/14/useful-things-to-know-about-engineering-levels/)的实用指南。

我认为重要的是要认识到，职业阶梯只对人群有效。它们很少，甚至从不对任何个人有效。这种效应在 Staff-plus 角色中变得尤为明显，这些角色通常只有几个人担任。阶梯是必不可少的，但不要陷入相信它们是事物实际运作方式的地图，而不是事物预期运作方式的神话。