

# Startup Advice: AI GTM, Pivoting & How To Hire-nGLmpKi-jRU – 图文解说

一张关键画面对应一段完整解说，按原视频时间推进。

## 文章摘要

本期围绕《Startup Advice: AI GTM, Pivoting & How To Hire-nGLmpKi-jRU》展开，按时间顺序讨论创业者最难回答的两件事：卖给谁，以及如何获得注意力、把 AI 带进传统行业：三条 go-to-market 路径，以及如何防止人工服务吞掉自动化、先追求学习速度，再决定是否上 Enterprise；AI SDR 只能放大已经成立的销售方法、不要等下一次模型跃迁：先判断产品会被淘汰，还是会因新模型变得更强。以下内容保留完整问答、例子、数字、画面信息和限定条件，适合先读这段摘要建立框架，再结合每个关键帧逐段阅读。

## 01. 创业者最难回答的两件事：卖给谁，以及如何获得注意力 (00:00–00:38)



场景 1

**本节主旨** 创业者最难回答的两件事：卖给谁，以及如何获得注意力

### 完整内容

**Pete:** 创业刚开始时，最难回答的通常不是“产品怎么做”，而是两个更基础的问题：我到底要卖给谁，以及我怎样才能让他们注意到我。每个创始人都必须亲自解决这两个问题，它们几乎是创业者必须完成的两项“魔术”。

**Pete:** 我来到 YC 之后，花了几年时间重新学习，甚至要主动忘掉过去积累的很多经验。那些经验在成熟公司里可能有效，但未必适用于创业公司。

**Brad Flora**： 如果一个创业者有大量时间反复思考“什么时候该开始招聘”或“到底要不要做某件事”，这本身可能说明公司还没有忙到必须做决定的程度。Office Hours 的本期内容，就是直接回答 YC 社区提交的问题，重点讨论 AI 产品的 go-to-market、增长、转型和招聘。

## 画面说明

画面是四位 YC 合伙人的完整圆桌构图：Gustaf Alström、Brad Flora、Pete 和 Nicolas Dessaigne 同时出现在桌边。它不是某一个回答者的特写，而是整期 Office Hours 的讨论框架，适合作为读者进入后续问答的总览图。

## 02. 把 AI 带进传统行业：三条 go-to-market 路径，以及如何防止人工服务吞掉自动化（00:38–07:09）



场景 2

**本节主旨** 把 AI 带进传统行业：三条 go-to-market 路径，以及如何防止人工服务吞掉自动化

### 完整内容

**Brad Flora：** 第一个问题来自一家想把 AI、Agent 或 LLM 带进传统行业的公司。它的长期目标可能是完全自动化，但第一天不可能交付完整的自动化系统，因此关键是：在产品还不完整时，应该怎样进入市场？

**Pete：** 以会计行业为例，大致有三条路。第一条是做卖给会计师事务所的 AI 软件公司：先理解这个行业，找到一个既有价值、又能在创业前几个月或前半年做出来的细分环节，把这一件事做好，而不是一开始就覆盖事务所的所有功能。第二条是自己开一家全栈会计事务所，用 AI 逐步替代其中的工

作；这意味着还要处理报税、结账和其他不常见但必须完成的事务，因此需要会计师和相当多的人工流程。第三条是收购一家已有客户的会计事务所，再把 AI 引入其中；好处是客户已经存在，难点是要改变一家既有公司的文化，而且公司越大越难。

最常见的是第一条，第二条也有不少公司在做，第三条相对少见。**如果选择自己经营服务公司，不能只看收入，必须持续追踪“有多少工作已经被自动化”，这个比例应该随时间上升。**

**Brad Flora：** 这类公司的危险在于，客户工作一多，团队会忙着交付服务，却把自动化放到“以后再做”。软件背景的创始人往往更能观察完整工作流，找出最容易先自动化的环节；非软件背景的行业创始人可能更容易被日常执行拖住。

**Pete：** 一个常见失败模式是只自动化了 **20%** 的工作，却因为收入增长而过早扩张，继续招聘 20 个、30 个会计师，最后变成“带一点软件的会计事务所”。与其只看总体收入，更应该建立能约束组织的指标，例如让技术人员保持足够比例，避免非技术人员不断向技术团队提需求，反过来阻塞自动化。讨论中提到过大约 **30%** 的技术人员比例，但重点不是这个数字本身，而是让团队持续对自动化轨迹负责。

**Nicolas Dessaigne：** 可以把“不能继续增加人工”设计成 forcing function：例如只有一个会计师，业务增加后不能靠无限招聘解决，只能想办法扩大自动化。这个指标必须公开、持续被所有人关注，形成组织文化。

**Pete：** 对软件投资人来说，自动化率的增长轨迹比单纯的收入更能证明这是一家软件公司。收入增长很快并不难，但如果大部分工作仍然靠人工，就没有证明自己能用软件改变行业。一个有意义的早期客户，应该是行业里的创始人或决策者，他们真正被激励去采用新软件，愿意让团队和创业者一起把事情做出来；否则很容易陷入漫长的企业销售，却不知道客户到底会不会采用。

他们举了 Vessence 的例子：**两位创始人没有法律背景，却找到一家对 AI 软件非常感兴趣的斯德哥尔摩大型律所，直接在律所里工作几个月，用真实工作流做出 MVP。**对于缺少行业经验的团队，这是比闭门猜需求更可靠的进入方式。寻找第一批客户时，还要提前设计资格筛选问题，判断对方是否真的愿意成为早期采用者、是否有权推动软件落地。

## 画面说明

画面保留四人圆桌全景，重点不是某一位的表情，而是这段讨论的组织关系：传统行业客户、软件自动化、人工交付和团队约束同时存在。读者应把它与正文中的三条 GTM 路径和“自动化率优先于收入”的指标一起理解。

### 03. 先追求学习速度，再决定是否上 Enterprise；AI SDR 只能放大已经成立的销售方法（07:09–14:39）



场景 3

**本节主旨** 先追求学习速度，再决定是否上 Enterprise；AI SDR 只能放大已经成立的销售方法

#### 完整内容

**Brad Flora：** Enterprise AI 产品常常只有少数买家，销售周期很长，但投资人又期待快速增长。创业公司是否应该先去中型市场，换取更大的潜在客户池和更短的销售周期？

早期最重要的指标不是合同金额，而是学习速度：团队多快能知道客户想要什么、用户真正需要什么、哪些是尖锐的痛点、哪些只是客户暂时容忍的问题。如果一开始就追逐大型企业，除非创始人有特殊关系能缩短销售周期，否则反馈会来得很慢，团队会失去迭代、交付和与用户对话的机会。

**Nicolas Dessaigne:** 公司有两类。一类可以先从长尾客户起步，随着产品成熟、覆盖更多用例，再逐步向上销售，最后形成百万美元级别的交易；另一类解决的本来就是企业级问题，小公司根本没有这个需求，那就不能为了“先做中小市场”而假装存在一个不存在的客户。更实际的做法是找到“最小的、确实有这个问题的客户”，或者把产品范围缩小到企业里的一个团队、甚至一两个用户，让产品先在狭窄范围内产生价值。

**Pete:** 细分市场之后还要资格筛选具体买家。企业、中型公司和小公司都可能声称想买软件，但真正重要的是那个人是否有决策权，是否有动力推动购买，是否愿意和团队持续沟通。合适的中型公司买家可能行动很快；不合适企业买家则会让创业者陷入几个月的销售周期，最后仍然不知道产品是否有效。

下一位创业者通过视频提问：应该招聘 growth hacker、传播、销售或 SDR，还是尝试用一个 AI 员工替代这些岗位？画面中保留了提问者 **John Banner (Respaid, YC S23)** 的姓名条。

**Pete:** AI SDR 有价值，但通常是在一套已经能工作的销售流程上发挥作用。如果创始人自己还不会卖产品，把 AI SDR 当成“最后的救命方案”，通常不会奏效。创始人必须先回答两个问题：我卖给谁，以及怎样得到他们的注意力。答案明确以后，AI 才能帮助寻找这些人、触达这些人，并把已经验证过的销售方法规模化。

**Nicolas Dessaigne:** 如果 AI SDR 公司专门服务那些连自己的产品都卖不出去的创业公司，客户可能短期带来收入，却很快流失。更好的客户是已经有好产品、能自己完成销售、已经找到销售“魔术”的团队。AI 的作用是扩大那个可重复的流程，而不是替创始人发现产品价值。

**Brad Flora:** 这和招聘第一个销售人员的原则一样：除非创始人已经弄清楚注意力从哪里来、客户会提出哪些反对意见，否则通常太早。真正适合招聘销售的时点，是销售人员只需要执行创始人已经写好的 playbook。AI SDR 和新的 AI 岗位更是如此，创始人应该先亲自学习这些工作，理解营销负责人、销售人员和 SDR 的职责，再决定要不要组建团队。很多高流失不是岗位的人不够好，而是创始人对岗位的期待从一开始就错了。

## 画面说明

**主画面选择 John Banner 的视频提问并保留姓名条。**它比重复一张四人圆桌更能说明问题：这一段不是主持人抽象讨论“AI 是否能替代销售”，而是从一个真实创业者的具体招聘选择开始，再回到 YC 合伙人的原则性建议。

## 04. 不要等下一次模型跃迁：先判断产品会被淘汰，还是会因新模型变得更强（14:40–16:18）



场景 4

**本节主旨** 不要等下一次模型跃迁：先判断产品会被淘汰，还是会因新模型变得更强

### 完整内容

问题是：创业者现在是否应该大笔投入，换取一个暂时的 AI 优势，还是等待下一代模型，让产品未来更便宜、更普及？

**Nicolas Dessaigne：** 第一个判断是，当前正在做的东西，等 GPT-5 之类的新模型发布后会不会直接变得无关紧要。如果产品只是解决“下一代模型马上会解决的痛点”，继续投入可能没有意义。第二种情况是，新模型会让现有产品明显变好；这时等待未必是正确策略。

如果现在投入，团队会在真实开发和真实用户反馈中学习。等模型能力成熟后，只要把新模型接入，产品可能从第一天就更强。早期确实可能在某些能力上花了过多的钱，但这些过程中的学习不是浪费，它们让团队提前理解用户、流程和产品边界。

**Pete：** 他们今年已经看到过类似情况：某些内部工具在更强的 Claude Sonnet 模型出现之前几乎不能工作，模型能力升级后突然变得可用。Cognition 等工具也经历过类似变化，早期看起来勉强可行，模型跃迁之后效果明显提升。

因此，决策重点不是简单地“等模型”或“现在烧钱”，而是判断当前工作是否会被新模型抹掉，以及当前投入能否积累不会被抹掉的学习：客户需求、工作流、分销、信任和产品整合。如果这些学习会留存，就有理由现在开始；如果唯一资产只是暂时弥补模型能力差距的代码，则应该谨慎。

## 画面说明

画面是两位合伙人的双人近景。这里没有 PPT 或图表，关键视觉信息是两位回答者之间的接力：先由一人提出“会被淘汰还是会变得更强”的分类，再由另一人用模型升级案例补充。正文保留这两个判断分支，而不是把问题压缩成“应该现在投资”。

## 05. 有收入和客户也可能需要 Pivot：真正的领先指标是 Conviction（16:18–26:07）



场景 5

**本节主旨** 有收入和客户也可能需要 Pivot：真正的领先指标是 Conviction

### 完整内容

**Brad Flora：** 如果产品完全不工作，改变方向很容易理解；困难的是产品有一些 traction，却不够强，创始人不知道是否该转型。这里没有一个只靠收入数字就能运行的公式。

**Nicolas Dessaigne：** 一个例子是 Firecrawl。它现在帮助很多公司从网站提取数据，已经有大量 AI Agent 客户；但在此之前，团队做的是 Mendable，在文档之上提供问答，已经有数十万甚至数十万美元级别的收入和大客户。团队在为原产品工作时，发现自己需要一个爬虫，却找不到合适的工具，于是先为自己做了出来。和其他 AI Agent 创始人交流后，他们发现几乎所有这类公司都需要这个能力，原本只是大产品中的一个内部组件，反而比原产品更有价值。

他们没有一夜之间从想法 A 跳到想法 B，而是先在已有产品旁边实验这个组件，看到它真正起势后才决定把它变成公司。这仍然是一次很大的冒险：放弃已有收入和大客户，几乎从头开始，但新方向的需求和信念足够强。

同样的判断也发生在 Algolia 的早期：最初产品有一些收入，却销售困难、客户并不真正重视。继续做下去可以成为一家生活方式公司，但不是团队想要的公司，于是必须改变。能否 Pivot，通常来自很多次客户对话和创始人对同行网络的深度理解，而不是一个机械的 A/B 公式。

**Brad Flora：** 有些团队拿着几千美元 MRR 和几十个付费客户，就误以为数字上升代表产品被需要。他们真正应该做的是去找用户，问清楚谁最重视产品、为什么重视。如果每个用户描述的价值都不同，说明产品和用户之间的关系还没有组织好，需要聚焦到一两个明确用例，而不是只让数字继续上升。

**Pete：** Pivot 还是一个非常脆弱的阶段。创始人要先确认自己有继续承受不确定性的精力，因为过去几个月甚至几年的工作可能都要被放下。不要只带着一个新想法去问别人“这个是不是更好”；更好的方法是同时探索一组想法，允许丢掉其中几项，从比较和验证中建立 conviction。别人可以给主观意见，但不是目标客户，最终仍要通过用户验证。

一个秋季 batch 的团队一直在“要不要 Pivot”之间犹豫，原来有几千美元 MRR，却始终没有形成真正的业务，后来转向开源计费框架 Autumn。转型后收入反而更少，但团队重新有了 conviction；这可能比短期收入更能预测未来是否会成功。真正值得注意的领先指标，是创始人已经不再相信当前方向会成功。

随后问题转为“什么时候该放弃一个好的创业想法去寻找伟大的想法”。他们认为，伟大不是创始人在脑中给出的形容词，而是用户真实需要、愿意持续付费、每天都依赖它来解决重大痛点之后才成立。创始人要像擦拭石头寻找钻石一样，积极销售、快速做出最夸张但可测试的版本，持续寻找伟大的迹象，并诚实面对反对证据。真正优秀的创始人往往长期专注于寻找和验证伟大想法，却不会反复宣称“我有一个伟大的想法”。

## 画面说明

画面保留年轻创始人和胡须合伙人的双人讨论构图，代表“有 traction 但是否真正被重视”的对照关系。这个场景没有稳定图表，关键证据在对话中的案例、收入、客户价值和 conviction，因此不按镜头切换增加截图。

## 06. 技术难不一定是 Pivot 理由：降低范围，但不要降低客户学习速度（26:08–30:43）



场景 6

**本节主旨** 技术难不一定是 Pivot 理由：降低范围，但不要降低客户学习速度

### 完整内容

问题是：如果潜在客户已经验证需求，产品只要做出来就可能卖得很好，但技术上迟迟达不到可用门槛，创始人是否应该因为太难而转向？

**Nicolas Dessaigne：**直觉上反而可能相反。真正困难的技术问题会抬高进入门槛，别人不愿意尝试；如果团队有能力、勇气和足够强的 conviction 去做，困难本身可能构成最好的机会。他举了一个做微型制药工厂的团队：原先做的科学软件勉强能工作，却没有让市场兴奋；转向小批量制造药物的微型工厂后，技术和监管挑战一个接一个出现，但团队每次讨论都更兴奋、更确信这是值得解决的问题。这样的项目很难，却可能在成功后改变整个行业。

但“很难”不能成为闭门开发六个月的借口。

**Brad Flora**：如果团队确定需要半年才能做完整系统，可以先把范围砍小。Perfect Audience 早期知道自己需要连接多个广告交易平台、实时竞价和大量集成，根本没有能力一次性完成，于是先找到已有底层产品和 API，在上面做一个非常好的前端，先把它拿去市场；他们还做了定制计费系统，用这个较小的版本获得用户、咨询合同和真实反馈，等理解足够深、建立关系之后，再去招聘能完成底层难题的人。

**Nicolas Dessaigne**：复杂技术也曾发生在 Algolia 的早期。搜索引擎底层很难，团队花了大约六个月才把产品做得足够好；但如果重新来过，他会花更多时间和客户相处，即使产品还没有准备好，也可以通过观察客户生活来理解问题。技术难度不能成为不和客户说话的理由。

**Pete**：Optimizely 的网站编辑器也花了至少六个月，因为它要在任意网站上运行，并让非技术用户无需写代码就能创建 A/B 测试。团队没有一开始就做完整公开产品，而是先为自己做最简版本：一个 bookmarklet，在任意网站弹出文本框，让创始人直接写 JavaScript 并运行。这个“最差但能用”的版本已经足够拿到咨询合同，帮助他们接触客户，最终他们成为自己的第一批用户，再逐步把问题拆开。

核心原则是：**难题可以继续做，但要把系统拆成能验证的部分；先做自己能用的最小版本，尽早获得真实客户反馈，同时保持对长期技术门槛的诚实判断。**

## 画面说明

画面是双人近景，右侧的 Nicolas 正在解释技术挑战，左侧可见电脑和桌面。它对应的是“困难本身可能是壁垒，但必须用小范围版本进入市场”的核心张力；正文中的 Perfect Audience、Algolia 和 Optimizely 案例承担了比单纯人物镜头更重要的信息。

## 07. 什么时候该招聘：不是把人数当成功指标，而是提前识别正在断裂的工作（30:44–35:19）



场景 7

**本节主旨** 什么时候该招聘：不是把人数当成功指标，而是提前识别正在断裂的工作

### 完整内容

**问题是：**创业公司应该用什么指标判断何时开始招聘创始人之外的员工？

**Pete：**如果你有很多时间认真思考这个问题，通常还太早；如果“要不要招聘”每天都在脑中出现，也可能还太早。真正该招聘的时点，是业务已经忙到你连安排一次候选人面试的时间都找不到，团队正在突破正常工作节奏，工程、销售或 onboarding 中有具体环节开始崩溃，创始人和现有员工不得不持续超负荷才能完成眼前的工作。

这听起来像“已经太晚”，因为招聘到入职可能需要三个月。**因此要寻找更早的真实信号：某一个具体环节正在坏掉，或者根据已有趋势很快会坏掉，而不是把自己的愿望误认为增长指标。**创始人要诚实地区分“业务真的会到达临界点”和“我希望它会到达临界点”。

早期创业公司没有大公司竞争力，前几名员工往往来自个人网络：他们已经了解创始人和产品，愿意先见面，也更容易被说服加入，而不是通过冷启动招聘找到。**早期招聘信号因此可能没有想象中那么紧迫，因为创始人身边可能已经有愿意等待机会的人。**

**Nicolas Dessaigne**：招聘大致经历三个阶段。产品市场匹配之前只招少数关键人员；找到产品市场匹配之后，团队可能又因为犹豫而招得太慢；再往后，公司变大，反过来又可能招得过多。他回顾一家早期公司：九个人、约 **120 万美元收入**、还没有 AI 时，工作压力非常大，但那段时期反而是记忆中最好的阶段。这说明“辛苦”并不是自动证明应该扩张的理由。

**Pete**：招聘不是成功指标。它只是避免公司因为某个关键环节断裂而失败，或让已经存在的公司能够正常运转。很多创始人会把“我已经招到人”误认为公司正在成功，实际上新员工可能让沟通、管理和协调变慢。

对大多数 batch 中的创始人，他的默认答案几乎是“不需要招聘”。例外是 opportunistic hire：例如你最聪明、最信任、与团队高度匹配的朋友恰好毕业或离开上一份工作，此时即使公司还没到常规招聘节点，也可以因为这个特殊机会邀请他加入。**这里的关键词必须是“最聪明、最优秀、最适合”，而不是仅仅因为他曾在某家大公司工作，或者看起来不错但还没有足够把握。**

## 画面说明

画面选用年轻创始人与胡须合伙人的双人镜头，体现“什么时候开始招聘”是对创业团队现实工作量的讨论，而不是人数排行榜。**正文的核心是临界点、三个月招聘提前量、个人网络和 opportunistic hire 四个判断条件。**

## 08. 企业 SaaS 什么时候适合开源：把信任、合规和自托管变成销售能力（35:20–38:16）



场景 8

**本节主旨** 企业 SaaS 什么时候适合开源：把信任、合规和自托管变成销售能力

### 完整内容

**问题是：**企业 SaaS 产品开源有什么优势和代价，什么时候值得这样做？

**Nicolas Dessaigne：**最常见的开源 go-to-market 是开发者工具，因为开发者关心代码、愿意检查产品，也更容易相信和自己使用同一技术语言的创始人。但开源不只适用于开发者工具。

以 Medplum 为例，它做的是开源 EHR。它开源的主要目的不是追求开发者社区或把开源当成传播渠道，而是建立企业客户的信任，并可能把销售周期缩短一年。医疗等敏感行业的客户可能不希望把数据交给一个无法检查的黑盒；能够查看代码、理解系统、在需要时修改或自托管，会降低采购风险。

类似地，Twenty 是一个面向 CRM 的纯 SaaS 产品，客户并不一定是开发者，但开源依然能带来几层价值：客户可以在需要时扩展产品，可以确认产品内部如何工作，也可以在必要时检查代码。许多客户实际上永远不会读代码，但“我有能力读、改和运行它”本身就足以产生信任。开源还可以帮助处理合规和数据驻留问题，让客户不必把敏感数据完全交给一家刚成立的小型云服务商。

**Brad Flora：** AI 产品中的这种需求现在更常见。过去客户要求自托管时，创业公司常把它当成不可能完成的特殊要求；现在许多小公司已经能比较快地提供本地部署或自托管版本，客户问到后就直接做出来。这是企业软件进步的一部分。

但自托管不是免费的。**部署、升级、维护和支持都会增加成本，企业必须为这部分复杂度付费，产品价格也可能需要明显提高。**开源也不是所有 SaaS 的默认 go-to-market；它更适合隐私、敏感数据、合规要求高，或客户必须拥有控制权的销售场景。最终要问的是：客户是否因为“能看见、能控制、能自托管”而更愿意购买，以及额外的支持成本是否值得。

## 画面说明

画面是两位合伙人的双人近景，正文中的开源产品案例承担主要证据功能。候选帧中还短暂出现 Medplum 的产品页面，但本次路由严格采用 conversation + studio + speaker，**因此保留人物镜头作为主图**，并在文字中说明 Medplum 这一画面所代表的企业信任和自托管逻辑。

## 09. 结语：把下一道创业问题带到 Office Hours（38:17–38:25）



场景 9

**本节主旨** 结语：把下一道创业问题带到 Office Hours

### 完整内容

节目组感谢创业者提交问题，并邀请观众在评论区留下自己的问题。如果希望在未来一期 Office Hours 中得到回答，可以直接提交；本期关于 AI go-to-market、Pivoting、技术难题、招聘和企业 SaaS 开源的讨论到此结束。

### 画面说明

画面回到四位 YC 合伙人的圆桌全景，作为整期问答的收束。它与开场总览图形成首尾呼应，说明前文是一个连续的 panel 讨论，而不是由多个独立视频拼接而成。