

DORA

2024

Google Cloud

加速:2024 年 DevOps 现状

Gold 赞助商

 catchpoint  chronosphere  DATADOG

Deloitte.

 Excella

 Gearset

 liatrion

 Middleware

 OPSERA

10

第 10 份 DORA 报告

目录

摘要	3	最后总结	83
软件交付绩效	9	致谢	85
人工智能:采用情况和态度	17	作者	87
探索 AI 对下游的影响	27	受众特征和企业特征	91
平台工程设计	47	研究方法	99
开发者体验	57	模型	113
引领转型	69	推荐的阅读内容	117
第 10 份 DORA 报告	77		

摘要

十多年来, DORA 一直致力于通过调研的方式, 来确定是哪些能力、做法和措施让技术驱动型团队和组织得以取得优异绩效。这是我们的第十份 DORA 报告。我们累计收到了超过 39,000 位专业人士的回复, 这些回复者来自全球各行各业不同规模的组织。感谢大家一路的大力支持和为调研做出的重要贡献!

为了收集数据, DORA 每年会面向全球从事技术及相关工作的专业人士开展问卷调查。调查中包含有关工作方式和成就的问题, 涉及组织层面和组织内的个人层面。

DORA 采用严谨的统计评估方法, 来厘清这些因素之间的关系及这些因素对团队和组织成功有哪些影响。

今年, 我们增加了调查方式, 对专业人士进行了深度访谈, 以此来获取更深入的洞见、进行三角验证, 并为我们的调研结果提供更多支持信息。请参阅[研究方法](#)一章, 了解详细信息。

今年, 我们调查的主要成就和成果如下:

减少工作倦怠	工作倦怠一种情绪、身体和精神上的疲惫状态, 是由于长期承受压力或压力过大导致的, 通常表现为有负面情绪、有疏离感和缺乏成就感。
沉浸度	沉浸度能衡量个人在开发工作中可以达到的专注程度。
工作满意度	工作满意度能衡量个人对工作的总体感受。
组织绩效	组织绩效能衡量组织方方面面的表现, 包括盈利能力、市场份额、客户总数、运营效率、客户满意度、产品和服务的质量及其达成目标的能力。
产品表现	产品表现能衡量产品的易用性、功能性、价值、可用性、性能 (如延迟时间) 和安全性。
效率	效率能衡量个人在工作中可以在多大程度上有效且高效地创造价值和完成任务。
团队绩效	团队绩效能衡量团队在协作、创新、高效工作、彼此支持及配合方面的能力。

主要调研结果

AI 的影响正在扩大

AI 的推广在软件开发领域掀起了一场范式转变。尝鲜者已经率先取得了一些喜人成果,但我们仍需保持谨慎。

采用 AI 能在以下方面带来好处:

- 沉浸度
- 效率
- 工作满意度
- 代码质量
- 内部文档
- 审核流程
- 团队绩效
- 组织绩效

但 AI 的采用也带来了一些不利影响。我们发现软件交付绩效有所下滑,AI 对产品表现的影响也还不明朗。此外,在个人层面上,随着 AI 采用的增多,个人在有价值的工作上花费的时间减少了。本报告的后面部分对这个不同寻常的发现做了进一步研究。

对于依赖 AI 所带来的影响,团队应继续实验并加深了解。

越信任 AI, 越采用 AI

开发者感觉使用生成式人工智能(生成式 AI)能帮助他们提高效率,信任生成式 AI 的开发者也会更多地使用 AI。这方面还有提升空间:仍有 39.2% 的回复者表示不太信任或完全不信任 AI。

以用户为中心有助于提高绩效

优先考虑最终用户体验的组织能开发出更优秀的产品,开发者的效率和满意度更高,也更少发生倦怠。

变革型领导力至关重要

变革型领导力可提高员工效率、工作满意度、团队绩效、产品表现以及组织绩效,同时有助降低员工发生倦怠的风险。

保持稳定的工作重点有助于提升效率和身心健康

组织频繁调整工作重点会严重影响效率并大大增加员工发生倦怠的风险,即便组织拥有优秀的领导者和优质的内部文档,并在软件开发中采用以用户为中心的方法也于事无补。

平台工程设计可提高效率

平台工程设计对提升效率和组织绩效有积极影响,但在软件交付绩效方面存在一些隐忧。

云服务可提高基础设施的灵活性

灵活的基础设施可提升组织绩效。但是,如果迁移到了云端,却没有发挥云带来的灵活性,则可能有害无益,还不如继续留在数据中心。要成功迁移,方法、流程和技术都需要改变。

高水平的软件交付绩效可以实现

绩效最好的团队在全部四个软件交付指标(更改准备时间、部署频率、更改失败率和失败部署的恢复时间)上都表现出色;而绩效最差者在这四个方面的表现也都很差。每个绩效类别中都有来自不同行业的团队。

应用 DORA 的数据洞见

要想借助 DORA 的数据洞见推动团队和组织改进,您需要评估自己当前的状况,找出要投资和改进的方面,并建立反馈环,以便跟踪行动进展。抱有持续改进的心态并付诸实践的团队更有可能获得最大收益。要投入资源,打造不断重复这一过程所需的组织能力。

您可以参考我们的调研结果,自行设计实验并做出合理假设。一定要通过实验来衡量变化产生的影响,发现对团队和组织最有效的做法。这样可以帮助您验证我们的调研结果。期待着您能收获不一样的结果。希望您能够分享自己的进展情况,以便我们都能从中汲取经验。

我们建议以实验的方式推动改进。

1. 找出您希望改进的方面或成果
2. 衡量基准水平或当前状态
3. 围绕可能有助于您达到理想状态的措施做出一系列假设
4. 达成共识并制定一项改进计划
5. 开展行动
6. 衡量取得的进展
7. 重复这一过程。改进是以迭代方式逐步实现的

DORA COMMUNITY



改进绩效不能仅凭一己之力！

我们可以从彼此的经验中学习：欢迎访问 DORA 社区网站 (<https://dora.community>)，在这个卓越的论坛中分享及学习各种改进计划。

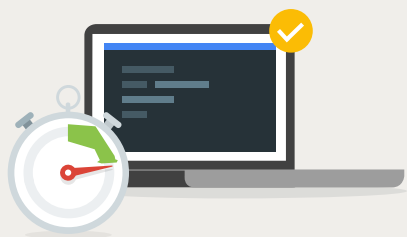
软件交付绩效

技术驱动型团队需要找到衡量绩效的方法，以便评估自己当前的进展、排定改进事项的优先顺序，并验证计划进展。DORA 反复验证了四个关键的软件交付指标。这四个指标能有效衡量软件交付流程的成果。



四个关键指标

DORA 的四个关键指标用来衡量软件更改的吞吐量和稳定性。这里所说的更改包括各方面的更改,包括更改配置和代码。



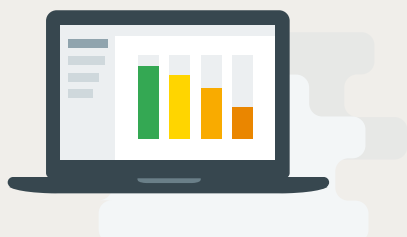
更改准备时间:

将提交或更改的代码成功部署到生产环境所用的时间。



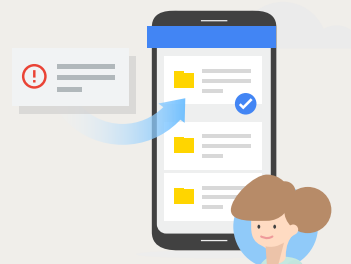
部署频率:

应用更改部署到生产环境的频率。



更改失败率:

更改部署到生产环境中后造成需要热修补或回滚的故障的百分率¹。



失败部署的恢复时间:

部署失败后所需的恢复时间。

我们发现这些指标往往同升同降,绩效较好的组织在所有四个指标上表现都好,绩效较差的组织则在所有指标上表现都差。

改进软件交付绩效的评估方式

长期以来,在对四个关键指标的分析中,更改失败率一直都游离在外。²虽然它与其他三个指标高度相关,但统计测试和统计方法不支持将它们合并成一个因素。对回复者回答更改失败率问题的方式进行更改后,这种关联有所加强,但我们感觉可能还有其他因素在发挥作用。

我们一直有个假设,即更改失败率可作为体现团队返工工作量的一个指标。交付失败后,团队可能需要通过引入新的更改来修复失败的更改。

为了验证这一理论,我们今年在调研中增加了一个关于应用返工率的问题:“过去的 6 个月里,在您开发的主要应用或服务中,大约有多少次部署是为了解决应用中用户可见的错误,并不在原计划中?”

分析数据证实了我们的假设:返工率与更改失败率相关。这两个指标共同构成了决定软件交付稳定性的可靠因素。

概念	
软件交付绩效	
因素	
软件交付吞吐量	软件交付稳定性
使用的指标	
- 更改准备时间 - 部署频率 - 失败部署的恢复时间	- 更改失败率 - 返工率

在此报告的分析中,我们多次使用软件交付绩效的概念和对应的两个因素。在描述软件交付绩效时,我们用到了全部的 5 个指标。

在描述软件交付吞吐量时,我们使用了更改准备时间、部署频率和失败部署的恢复时间。此因素衡量了执行任何更新、正常更改以及故障修复更改的速度。

在描述软件交付稳定性时,我们用到了更改失败率和返工率。这一因素衡量了部署直接导致额外的计划外工作量的可能性。



绩效等级

在每年的调查问卷中，我们都会询问回复者其开发的主要应用或服务的软件交付绩效。在分析他们的回答时，我们采用聚类分析的统计方法，识别相似回复，找出不同绩效等级。

为了与往年的聚类分析保持一致，我们仍对原始的四个软件交付指标执行聚类分析。

在分析软件交付绩效时，回复形成了四个类别。我们没有预先设定这些等级，而是让它们从问卷的回复中自然浮现。这样，我们可以大致了解每年所有回复者的软件交付绩效。

如下图所示，今年的数据中出现了四个不同的类别。

绩效等级	更改准备时间	部署频率	更改失败率	失败部署的恢复时间	回复者所占百分比*
卓越	不到 1 天	按需(每天多次部署)	5%	不到 1 小时	19% (18-20%)
高	介于 1 天到 1 周之间	介于每天一次到每周一次之间	20%	不到 1 天	22% (21-23%)
中	介于 1 周到 1 个月之间	介于每周 1 次到每月 1 次之间	10%	不到 1 天	35% (33-36%)
低	介于 1 个月到 6 个月之间	介于每月一次到每六个月一次之间	40%	介于 1 周到 1 个月之间	25% (23-26%)

*89% 的不确定性区间

吞吐量还是稳定性？

在这四个类别中，吞吐量和稳定性都是相关的。即便是在中等绩效类别（橙色）中（相比高绩效类别 [黄色]，吞吐量较低而稳定性较高），二者的相关性也依然存在。这表明有吞吐量和稳定性之外的因素影响了这个类别的绩效。例如，提高更改的频率可能对中等绩效类别的组织有益。

在提高部署频率和降低部署失败率之间，哪种更好？

这个问题没有统一答案，取决于要衡量的应用或服务、应用开发团队的目标，最重要的是应用用户的期望。

我们决定称速度较快的团队为“高绩效团队”，速度较慢但较稳定的团队为“中绩效团队”。这一决定突显了使用绩效等级存在的潜在弊端：对于团队来说，取得进步应当比达成某个绩效等级更为重要。最好的团队是实现了卓越级别的**改进**的团队，而不一定是卓越**绩效**团队。

软件交付绩效等级

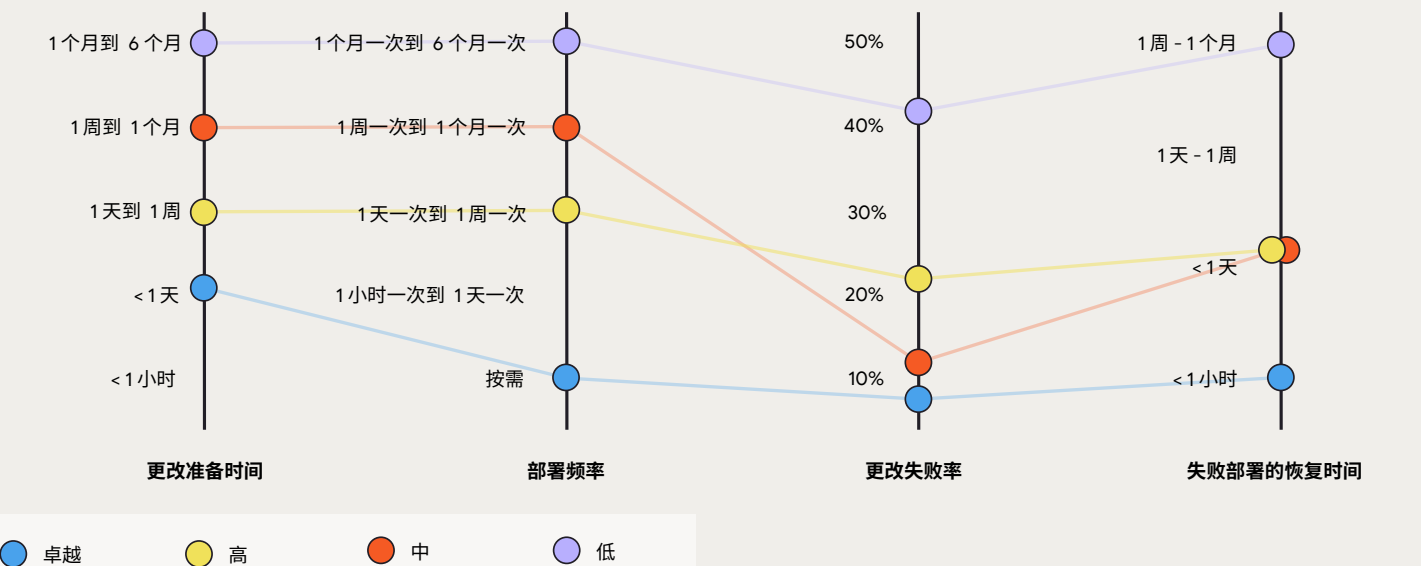


图 1: 软件交付绩效等级

卓越绩效团队与低绩效团队相比



如何使用绩效类别

绩效类别提供了今年调查回复者的软件交付绩效的基准数据。这些类别旨在鼓励大家相信，卓越绩效是可以实现的。

相较达到某个绩效等级，我们认为团队更应着力改进总体绩效。最好的团队是实现了卓越级别的**改进**的团队，而不一定是卓越**绩效**团队。

行业对绩效等级的影响不大

我们的研究几乎未³发现行业对软件交付绩效的影响：每个行业都有高绩效团队。这并不是说行业不存在特有的挑战，只是说在软件交付绩效方面，没有一个行业看起来特别困难，或是特别轻松。

使用软件交付绩效指标

每个应用或服务都有自己独特的背景。因此,我们很难预测某个更改对系统总体绩效的影响。此外,在组织内,几乎任何一项更改都会引发连锁反应。考虑到这种复杂性,我们如何使用软件交付绩效指标来指导改进?

首先,找出您想衡量和改进的主要应用或服务。然后,我们建议召集负责此应用的跨职能团队,衡量当前的软件交付绩效并达成共识。DORA 快速评估 (<https://dora.dev/quickcheck>) 可帮助引导交流并实施基准测量。您的团队需要了解什么在拉低绩效。

和团队一起完成价值流映射练习⁴是找出阻碍的一种有效方法。

接着,确定改进计划并就此达成一致意见。改进计划可集中改进 DORA 研究中提到的一项能力⁵,或者您的应用或组织的其他特殊需求。

有了计划之后,就要付诸行动了!为改进工作分配资源,并关注在过程中学到的经验。

在更改实施并产生效果后,就需要重新评估 4 大关键指标了。团队实施更改后,这些指标有什么变化?您得到了哪些经验?

重复这个过程将帮助团队形成持续改进的习惯。

记住:变革不是一朝一夕之功。需要采用迭代方法,打造学习文化,并实现快速流动和快速反馈⁶。

¹ 如果部署在生产环境中发布后,引发了可被终端用户感知的问题,则视为更改失败。相反,如果一个更改在部署到生产环境之前就被终止,则证明部署流程能够成功检测出错误。

² Forsgren, Nicole、Jez Humble 和 Gene Kim。2018 年。《Accelerate: The Science Behind DevOps: Building and Scaling High Performing Technology Organizations》(加速: DevOps 背后的科学:建立和壮大高性能技术组织)。IT Revolution Press。第 37-38 页

³ 《The 2019 Accelerate State of DevOps》(加速:2019 年 DevOps 现状) (第 32 页) 报告显示,零售行业的软件交付绩效明显更好。<https://dora.dev/research/2019/dora-report/2019-dora-accelerate-state-of-devops-report.pdf#page=32>

⁴ <https://dora.dev/guides/value-stream-management/>

⁵ <https://dora.dev/capabilities>

⁶ <https://dora.dev/research>

人工智能： 采用情况 和态度



要点

在我们的调查中,各个行业的大部分组织都在将工作重点转向更深入地将 AI 融入自己的应用和服务。与之对应的是,大部分专业开发者正依靠 AI 履行核心工作职责,并称 AI 帮助他们提高了效率。专业开发者普遍认为,要在当今市场保持竞争优势,就离不开 AI 的帮助。这一观点似乎成了组织和个人专业开发者采用 AI 的重要驱动因素。

简介

今年, AI 对整体开发工作产生了巨大影响。这一点很难忽略,因为大量热门新闻文章都是在报道 AI 的影响,无论是赞扬的¹、批评的²,还是表示担忧的。³在《加速:2023 年 DevOps 现状报告》中,我们只把 AI 当做影响绩效的诸多技术能力中的一项来讨论,⁴而今年我们要更全面地探索这一主题。

随着 AI 在专业开发工作中迅速从边缘走向普及,行业发展进入关键转折点。《加速:2024 年 DevOps 现状报告》的发布恰逢其时,可以帮助组织评估专业开发者对 AI 的采用、使用情况和态度。

发现结果

采用人工智能

关于 AI 采用情况的调查表明，人们越发意识到，AI 已不再是“未来趋势”，AI 时代已经全面来临，而且很可能会一直持续下去。

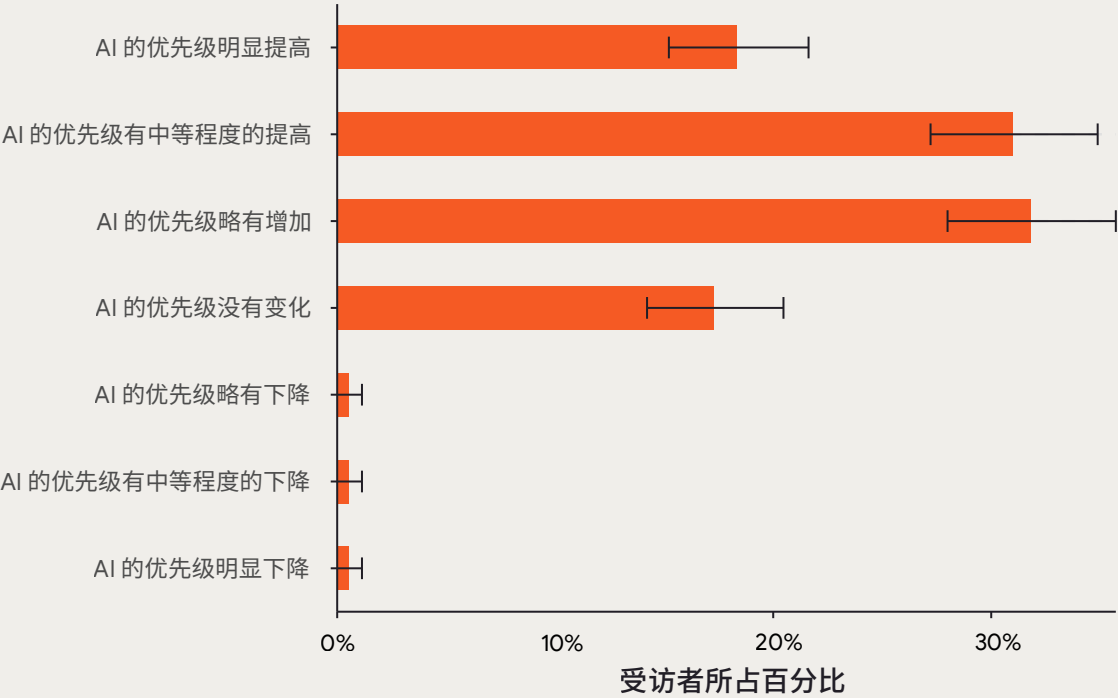
组织层面的人工智能采用情况

绝大部分回复者 (81%) 称其组织已将工作重点调整为进一步将 AI 融入自己的应用和服务。49.2% 的回复者甚至将这种转变的程度描述为“中度”或者“显著”。

值得注意的是，3% 的回复者称其组织正减少对 AI 的关注 (在本次调查的误差范围之内)。78% 的回复者相信，组织在改变工作重点后，会明确传达其 AI 采用计划。图 2 直观呈现了这些数据。

组织对 AI 优先级的调整

如
回



误差线代表 89% 的不确定性区间

图 2: 回复者对于组织提高/降低在应用和服务中整合 AI 的优先级的观点。

在我们调研的所有行业中,受访者在日常工作中对 AI 的依赖程度在统计数据上是相同的,这表明所有行业都在快速采用 AI。这有些出人意料。不同行业的监管约束度以及历来的创新步伐可能大相径庭,这些差异都会影响到技术采用的快慢。

但我们发现,与小型组织的回复者相比,大型组织的回复者在日常工作中对 AI 的依赖程度较低。这与以往的研究结论一致,即由于更高的组织复杂性和协调成本,大型企业适应技术变革的速度较慢。⁵



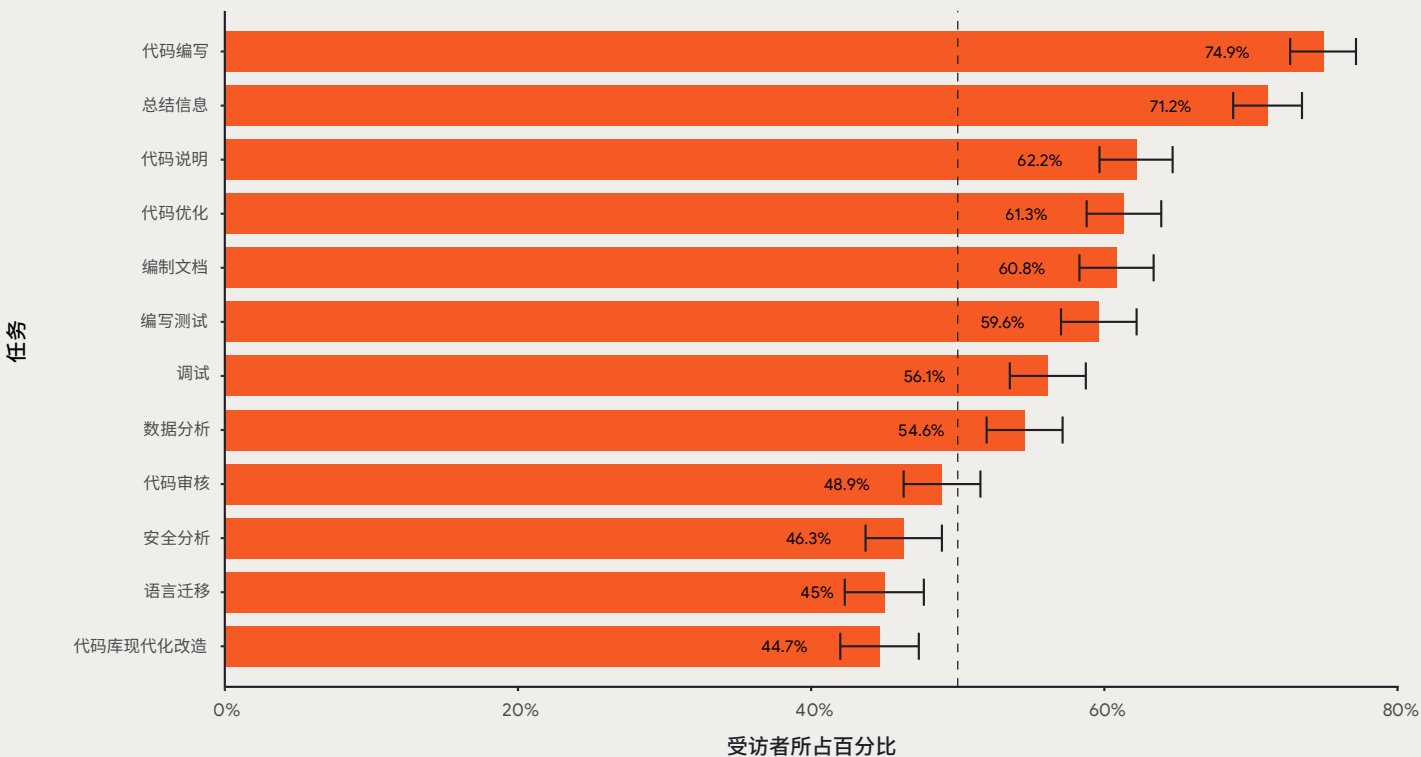
个人层面的人工智能采用情况

在个人层面上,我们发现 75.9% 的回复者会依赖(至少部分依赖)AI 来完成一项或多项日常工作任务。工作中涉及以下任务的大部分回复者都会依赖 AI:

1. 编写代码
2. 总结信息
3. 解读陌生代码
4. 优化代码
5. 编制代码文档
6. 编写测试
7. 调试代码
8. 数据分析

在调查回复中涉及的所有任务中, AI 在软件开发中最常见的用例是编写代码和总结信息。在工作职责中包含这两项任务的受访者中, 分别有 74.9% 和 71.2% 的人依赖 AI (至少是部分依赖 AI) 来完成的任务。图 3 直观呈现了这些数据。

各类任务对 AI 的依赖度



误差线代表 89% 可信区间

图 3: 依赖 AI (至少部分依赖 AI) 执行常见的 12 项开发任务的受访者比例

在日常工作中, 受访者与 AI 互动最常用的界面是聊天机器人 (78.2%), 其次是外部网页界面 (73.9%), 以及集成在 IDE 中的 AI 工具 (72.9%)。受访者通过内部网页界面 (58.1%) 和在自动化 CI/CD 流水线 (50.2%) 中使用 AI 的比例较低。

但我们知道, 受到与这些技术接触频率的影响, 受访者很可能意识不到自己在 CI/CD 流水线和内部平台中使用了 AI。因此这些数字可能被人为低估了。

我们发现, 与所有其他职位的受访者相比, 数据科学家和机器学习专家更倾向于依赖 AI。相反, 硬件工程师比所有其他职位的受访者更少依赖 AI, 这可能是由于硬件工程师的职责不涉及上述经常用到 AI 的任务。



采用人工智能的驱动因素

接受访谈的参与者多认为采用 AI 一方面是因为竞争压力,另一方面是为了满足行业标准,因为行业标准也在逐渐认可将掌握 AI 作为组织和个人的一项能力。

在一些受访者的组织中,使用 AI 本身就被视为“一个重要的营销亮点”(P3)⁶,可以帮助组织与竞争对手拉开差距。在得知竞争对手开始在其流程中采用 AI 后,一家公司甚至放弃了针对采用新技术的“繁琐组织流程”,因为他们急于采用 AI,并担心自己落后于竞争对手 (P11)。

在个人层面上,很多受访者认为其之所以采用 AI,是因为感到在软件开发领域,熟练运用 AI“似乎成了工程师必备的一项基本技能”(P9)。一些受访者认为开发者同行应该尽快在其开发流程中采用 AI,因为“这个领域实在发展得太快,几乎跟不上它发展的步伐…我认为,不使用 AI 很快就会被淘汰”(P4)。

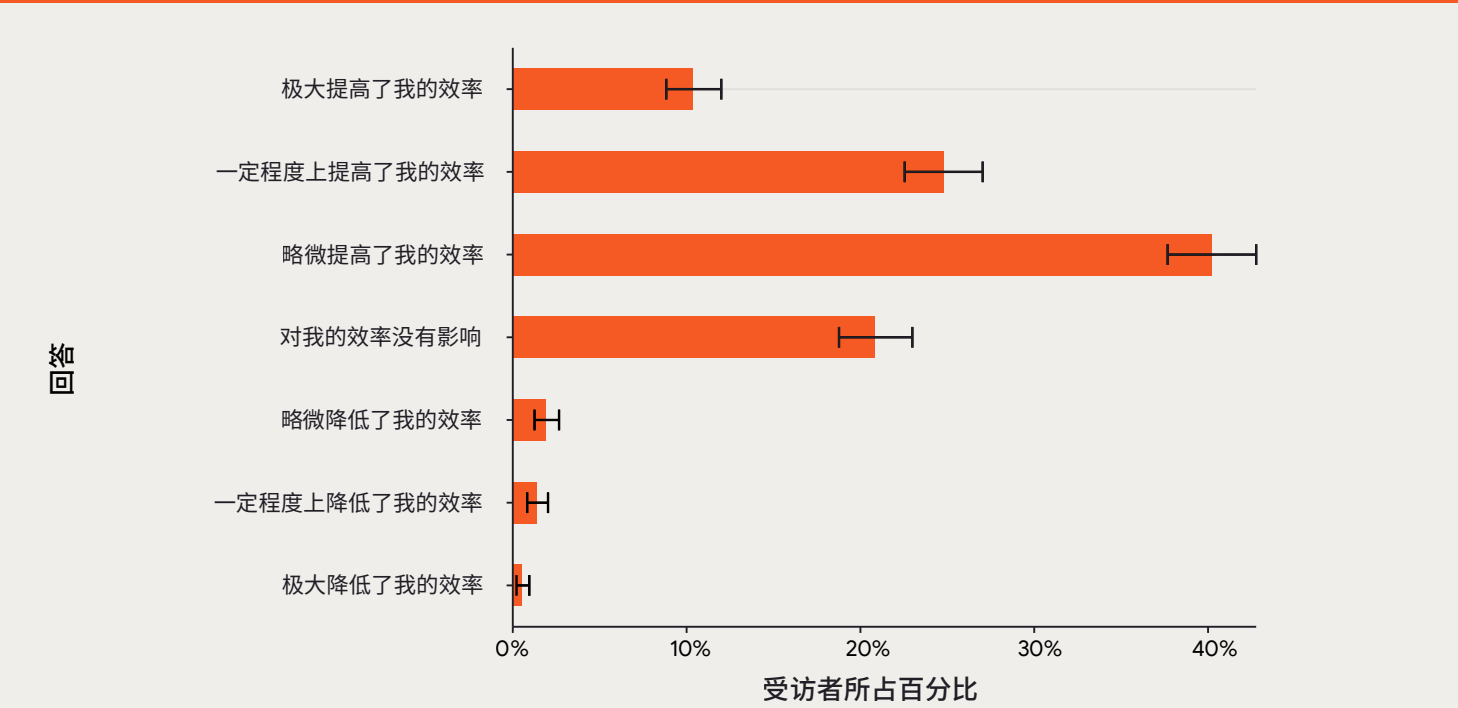
对人工智能的看法

人工智能可提高绩效

对于大量正在采用 AI 的组织和开发者来说,在开发工作中使用 AI 收效显著。在 2024 年初进行的调查中,75% 的回复者表示,他们在调查开始前的三个月中通过使用 AI 提高了效率。

值得注意的是,超过三分之一的受访者表示,自己感觉采用 AI 后效率得到了中等 (25%) 或极大 (10%) 的提升。有不到 10% 的回复者称 AI 对其效率产生了负面影响,即便这个影响非常轻微。图 4 直观呈现了这些数据。

AI 对效率的影响



误差线代表 89% 的不确定性区间
图 4:回复者如何看待 AI 对效率的影响。

在各种职位的回复者中,认为 AI 带来的效率提升幅度最大的是安全专家、系统管理员和全栈开发者。移动开发者、站点可靠性工程师和项目经理也表示效率有所提升,但与所有其他提及的职位相比,他们报告的效率提升幅度较小。

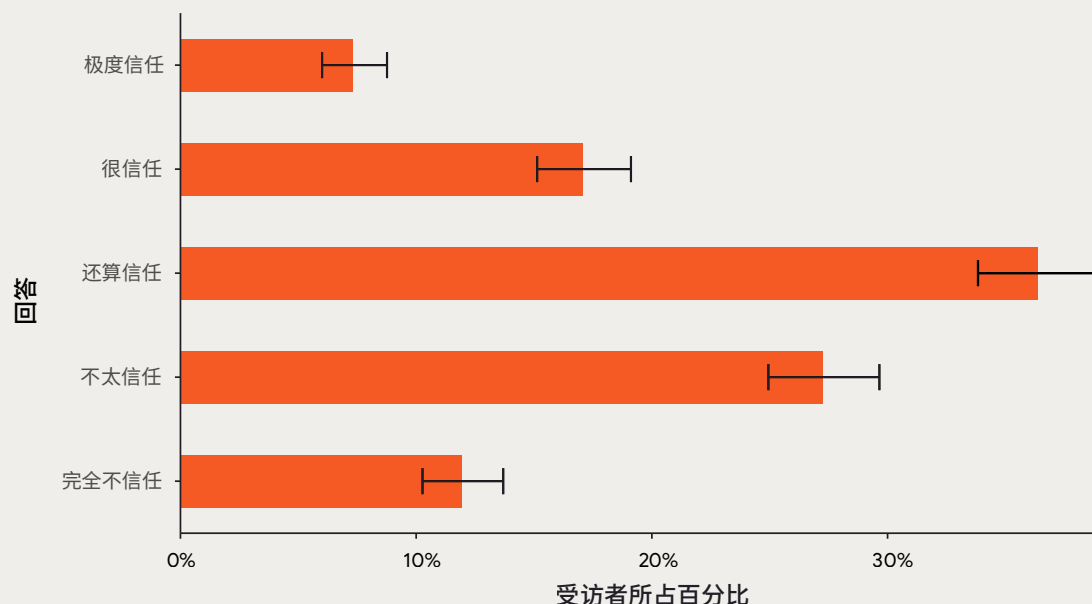
虽然我们曾怀疑, AI 在开发中的运用尚不成熟且需要一个学习过程,而这可能会限制开发者编写代码的能力,但我们的调研结果并不支持这一假设。只有 5% 的回复者表示 AI 在一定程度上限制了其编写代码的能力。事实上, 67% 的回复者表示,借助 AI 辅助的编码工具,他们编写代码的能力至少有一定程度的提高;而约 10% 的人表示, AI 让他们编写代码的能力有了“极大”提高。

对 AI 生成代码的信任度

对于能否在开发工作中安心使用 AI 生成的代码,受访者看法不一。虽然绝大多数回复者 (87.9%) 表示在一定程度上信任 AI 生成代码的质量,但总体信任度较低,有 39.2% 的回复者表示对 AI 生成的代码几乎不信任 (27.3%) 或完全不信任 (11.9%)。图 5 直观呈现了这些数据。



对 AI 生成代码质量的信任度



误差线代表 89% 的不确定性区间

图 5: 受访者对 AI 生成代码质量的信任度。

尽管调查显示开发者正在快速采用 AI、依赖 AI 并认为 AI 对绩效有积极作用,但大家对 AI 的信任度总体不高,这一点让我们感到很惊讶。值得注意的是,在我们的访谈中,许多受访者表示,在专业工作中,他们愿意或预期会对 AI 生成的代码进行调整。

一位受访者甚至用早期的 StackOverflow 来比喻评估和修改 AI 代码的必要性:“[当时]你总觉得 StackOverflow 上的人经验丰富,什么都懂。然而,当你直接复制粘贴他们的代码时,往往会爆发大量问题”(P2)。

或许因为类似问题已不新鲜,像 P3 这样的受访者认为其公司并不“担心有人直接从 Copilot 或 ChatGPT 复制和粘贴代码,[因为他们现有的代码质量保证流程]具有多重检查机制”。

我们假设,开发者不一定期望 AI 生成的代码准确到完全可以信任,这并不妨碍开发者认为 AI 生成的代码有价值。相反,似乎只要 AI 生成的代码基本正确,且能够通过调整来完善,开发者就可以接受。这样的代码足以吸引开发者广泛采用,同时也能融入到现有的质量保证流程中。

对 AI 未来的期待

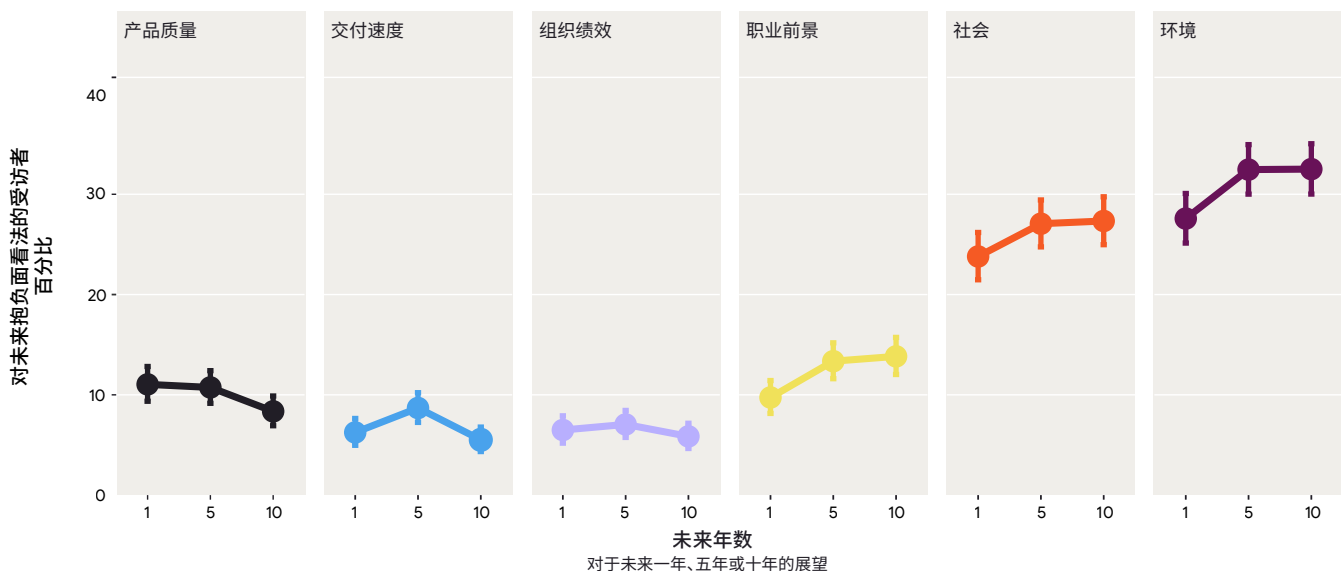
总体而言，我们的调研结果显示，AI 已经对专业开发者的工作产生了巨大影响，而且我们预计这种趋势将继续增强。尽管我们无法准确预测 AI 未来将如何影响开发领域以及我们的世界，我们还是让回复者设想了一下 AI 在未来一年、五年和十年内的影响，并与我们分享了他们的看法。

回复者表示 AI 对他们近期的开发工作产生了相当积极的影响，但他们对 AI 未来影响的预测却不如当前乐观。

乐观地看，回复者预计在未来的一年、五年和十年内，AI 将继续提高他们的产品质量。这与我们发现 AI 对专业开发者的绩效产生了积极影响这一结论一致。

但回复者也表示，综合来看，AI 预计将会对他们的职业前景、环境以及整个社会产生净负面影响，并且这些负面影响将在约五年后全面显现。图 6 直观呈现了这些数据。

AI 负面影响预期



误差线代表 89% 可信区间

图 6: 回复者对 AI 在未来一年、五年和十年的负面影响的预期。

与调查回复者相似,关于 AI 对未来的影响,访谈参与者的看法也比较复杂。一些受访者对于未来在监管政策尚未明确时可能面临的法律诉讼表示担忧,害怕“在规则明确后成为违规的一方”(P3)。

其他受访者表达了人们长期以来的担忧,并质疑:“它会取代人类吗?谁知道呢?也许吧。”(P2), 他们的同行则通过以前的例子打消了这种顾虑,“人们都说‘千年虫问题会让所有系统都崩溃!’因为当时它是个新事物。

[但是,] 没有什么工作最后被取代了。相反,还创造出了更多工作。我相信 AI 也会一样”(P1)。

AI 对世界的未来影响仍然未知。但是,今年的调查清晰显示, AI 已经为软件开发领域带来了一场无法忽视的范式转变。目前,专业开发者积极接受了这些变化。



1. <https://www.sciencedaily.com/releases/2024/03/240306144729.htm>

2. <https://tech.co/news/list-ai-failures-mistakes-errors>

3. <https://klyker.com/absurd-yoga-poses-generated-by-ai/>

4. <https://dora.dev/dora-report-2023>

5. Rogers, Everett M., Arvind Singhal 和 Margaret M. Quinlan. “Diffusion of innovations” (创新的融合)。《An integrated approach to communication theory and research》(一个整合的沟通理论和研究策略)。Routledge, 2014 年。432-44, Tornatzky, L. G. 和 Fleischer, M. (1990)。《The processes of technological innovation》(技术创新的过程)。马萨诸塞州列克星敦市: Lexington Books

6. (P[N]), 如 (P1), 是指访谈受访者的化名。

探索 AI 对下游的影响



要点

这一章研究了采用 AI 对从个人开发者到整个组织的影响。调研结果揭示了一个复杂的图景, AI 带来了显著的好处, 也产生了意想不到的负面影响。虽然采用 AI 可以提升个人的效率、沉浸度和工作满意度, 但同时也可能减少花在有价值的工作上的时间。

此外, AI 对代码质量、文档和审核流程产生了积极影响, 但并没有提高软件交付绩效。这一点令人意外。事实上, 采用 AI 在这方面似乎产生了负面影响, 对产品表现的影响也可忽略不计。

尽管存在这些挑战, 采用 AI 还是帮助团队和组织提高了绩效。本章最后呼吁批判性评估 AI 在软件开发中的作用, 并主动调整 AI 的应用, 以最大程度地发挥优势并减轻难以预见的后果。

AI 时代和 DORA

据估计,领先的科技巨头将在未来五年内投入大约 1 万亿美元用于 AI 的开发。¹这与“[人工智能:采用情况与态度](#)”一章中展示的数据相吻合,即 81% 的回复者表示其所在公司已将资源转向 AI 的开发。

AI 对环境的影响让成本构成变得更加复杂。部分估算数据表明,到 2030 年,AI 将会使数据中心的电力需求增长 160%。²训练一个 AI 模型的耗电量大致相当于“1,000 多个美国家庭一年的用电量”。³因此,超过 30% 的回复者认为 AI 将对环境产生负面影响也就不足为奇了。

除了开发和环境成本,我们可能还要承担采用 AI 的成本。

成本可能是多方面的,例如因效率下降而额外付出的成本、聘请专业人员等。采用成本也可能在社会层面出现。超过三分之一的回复者认为,AI 在未来十年内会对社会产生负面影响。考虑到这些成本,人们对回报的强烈好奇心也就显得顺理成章了。

大量的媒体报道、文章和研究都已经体现了这种好奇心,而其中的观点和数据至少在一定程度上是混杂不一的。



有人认为 AI 极大增强了人的能力,⁴另一些人则表示 AI 不过是帮忙做作业的工具,⁵也有人担心 AI 将威胁人类。⁶

在考察近端成果时,诸如成功完成特定任务的能力,AI 的正面效果相对明显。⁷但当考察远端成果时,例如团队的代码库,结果就不是那么明确和正面了。例如,有些研究表明,代码搅动率可能是 2021 年之前基准水平的 2 倍。⁸

弄清 AI 对下游的影响并不容易,这是意料之中的。影响与原因之间的关系越远,关联就越模糊和细微。

评估 AI 对下游的影响,就好像是量化投石入湖产生的涟漪。最容易统计的是距离石头入水点最近的波纹。距离入水点越远,石块的影响就变得越小,也就越难将水波归因于石块的冲击。

AI 本质上就像是一块投入到汹涌大海中的石头。理解 AI (或任何技术或实践) 引发的涟漪是一大挑战,因为其中涉及其他流程,而且整个环境在不断变化。业界一直难以采用一套系统化的衡量和分析框架,来了解 AI 的影响,这可能就是原因之一。⁹

我们的方法专门为解决这类挑战而设计。DORA 的目标是弄清楚一项实践有无效用。过去十年,我们研究了诸多实践对下游的影响,包括安全实践、变革型领导力、生成型文化、文档实践、持续集成、持续交付以及以用户为中心等。¹⁰

我们相信 DORA 的方法¹¹可以帮助我们了解 AI 的影响,特别是帮助我们探索 AI 在多种成果中的影响。



衡量 AI 的采用情况

要衡量采用 AI 所产生的影响, 第一个挑战是衡量 AI 的采用情况。我们认为, 要了解 AI 在开发工作流程中的地位, 衡量依赖度可能比衡量使用频率更有意义。您可能每月, 甚至每几个月才进行几次代码审核或者文档编写工作, 但这些任务对您的工作至关重要。

反之, 频繁使用 AI, 并不等于 AI 用在了重要或核心工作中。

有鉴于此, 我们请回复者回答他们的总体 AI 依赖度和在特定任务中对 AI 的依赖度。[上一章](#)详细展示并解读了调研结果。

借助因素分析, 我们发现, 我们的“总体 AI 依赖度”调查项与以下任务在 AI 依赖度上存在高度重叠:

- 代码编写
- 总结信息
- 代码说明
- 代码优化
- 编制文档
- 编写测试

这七个项目间的高度关联和一致性表明存在一个潜在因素, 即我们所谓的“AI 采用情况”。

AI 对个人的影响:既有明显的益处,也有一些潜在的不利

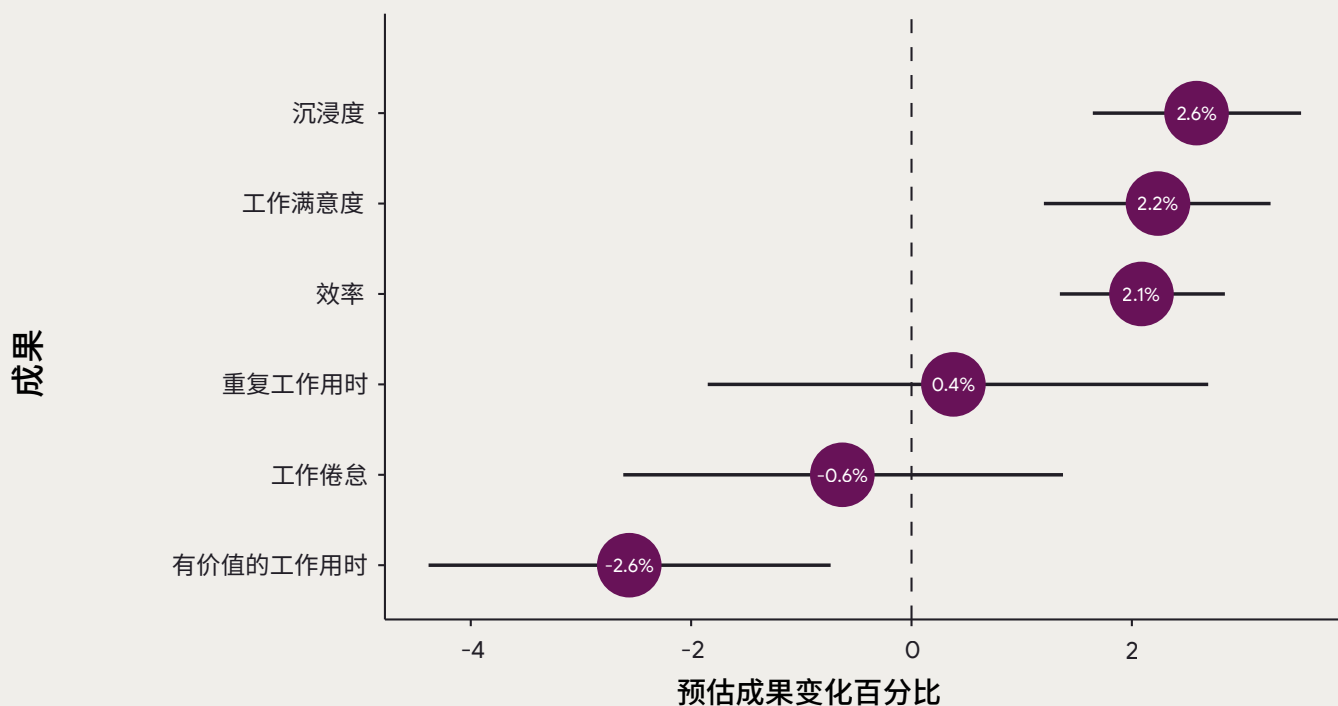
与往年一样,我们衡量了多种与个人成功和身心健康相关的构想:

工作满意度	用于概括个人对工作的总体感受的一项指标。
工作倦怠	一个从多方面描述倦怠的因素,包括生理、情绪和心理等维度及其对个人生活的影响。
沉浸度	这项指标用于衡量个人在开发任务中可以达到的专注程度。
效率	用于衡量个人感觉能否有效、高效地工作并创造价值和完成任务的一项因素,用分数表示。
重复工作用时	这项指标用于衡量个人用在长期价值较低的重复劳动及手动任务上的时间比例。
有价值的工作用时	这项指标用于衡量个人花在自认为有价值的任务上的时间比例。

我们希望了解, AI 的采用是否会影响回复者在这些问题中的答案。结果表明, 事实往往就是这样。

我们尽可能准确地估算了采用 AI 对个人成功和身心健康的影响, 并通过图 7 中的可视化图表进行了呈现。

如果个人将 AI 使用率提高 25%...



点 = 估算值

误差线 = 89% 的不确定性区间

图 7: AI 的采用对个人成功和身心健康的影响

明显的益处

采用 AI 为个人带来的好处总体上令人欢欣鼓舞,但也存在一些小问题。**目前比较明确的是, AI 可显著提升沉浸度、效率和工作满意度(见图 7)。**

例如,当个人的 AI 使用率提高 25% 时,效率可能会提高约 2.1% (见图 7)。数字看似不大,但这只是个人层面上的数字。设想下这个数字推广到数十名、甚至数万名开发者身上时,会产生怎样的效果。

这就是我们期望的效果。我们认为,这在一定程度上得益于 AI 能够整合来自不同来源的信息,并集中给出高度个性化的回答。如果是个人独立完成这样的工作,就需要花费大量的时间,还要在不同的工具之间切换,这样更不利于专注工作。

鉴于效率和沉浸度与工作满意度密切相关, AI 的采用能够提升工作满意度也就不足为奇了。

潜在的不利影响

这一点比较复杂。采用 AI 的一个价值主张是，让人们把更多时间花在有价值的工作上。也就是说，我们期望通过使用 AI 自动处理重复劳动、手动任务及繁琐任务，回复者能够将时间用于“更有价值的工作”。然而，数据表明，AI 采用的增加可能会产生相反的效果，即花在有价值的工作上的时间减少了，而用于重复工作的时间似乎没有变化。

回复者的身心健康指标，如沉浸度、工作满意度和效率，一向与花在有价值的工作上的时间存在关联。因此，看到这些指标都在上升，而花在有价值的工作上的时间却在减少，我们感到很意外。

要对这些规律做出正确的解读，就要研究清楚这个看似反常的结果，就像看电影时不能忽略与您的理解相悖的场景，看书时不能跳过与您的理解不符的章节。同样，要正确解读这些规律，就不能只看符合常理的那一部分。

尽管有无数种假设可以解释这些数据，但我们最终提出了一个似乎最简约的假设，来说明为什么采用 AI 会提升沉浸度、效率和工作满意度，但同时又减少了用于有价值的工作的时间，而没有影响用于重复工作的时间。

我们将这一假设称为“真空假设”。通过提高效率 and 沉浸度，AI 能帮助人们更高效地工作，更快地完成有价值的工作。

这样就出现了“真空”，多出了额外时间。AI 没有减少回复者工作的价值，而是加快了价值的实现。



什么是有价值的工作？

为了解释这些反直觉的调研结果，我们进一步探索了回复者认为哪些类型的工作是有价值的，哪些是重复无功的。

根据传统观念、以往的报告以及访谈中获取的定性数据，回复者认为与开发相关的任务（如编码）是具有价值的工作；而价值较低，甚至重复无功的工作通常包括与组织协调相关的任务（如开会）。按照这个分类体系，AI 更适合协助回复者处理他们定义的“有价值”的工作，而不是“重复”的工作。

我们从访谈的定性数据中发现，当主持人提问是否认为自己的工作“有意义”时，受访者通常会以工作对他人产生的影响来衡量工作的价值。

过去两年的 DORA 报告进一步证实了这一点，以用户为中心对提高工作满意度非常有益。

例如，在描述最近的职位变动时，P10¹² 表示之所以做出这个决定，是因为“这样我能影响更多人，做更有意义的事。”P11 也表示“如果你从零开始开发了一款软件或服务，看到它交付给用户或客户，你会很有成就感，你会对自己说‘太棒了！这是我开发的，很多人都在用！’”

开发工作的“意义”来自所创造的解决方案的影响力，而不是编写代码本身，这就解释了为何我们看到回复者花在有价值的工作上的时间变少了，但对自己的工作却感到更加满意了。

虽然 AI 让人们更轻松、快速地完成他们认为有价值的任务,但它并没有真正帮助人们处理那些他们不喜欢的任务。于是,重复的工作和工作倦怠依然存在,采用 AI 也无济于事。这说明 AI 并没有帮助我们摆脱冗长的会议、低效的常规流程以及其他诸多重复任务(图 8)。

但好的一面是, AI 也没有让情况变得更糟,也没有对回复者的身心健康产生负面影响。

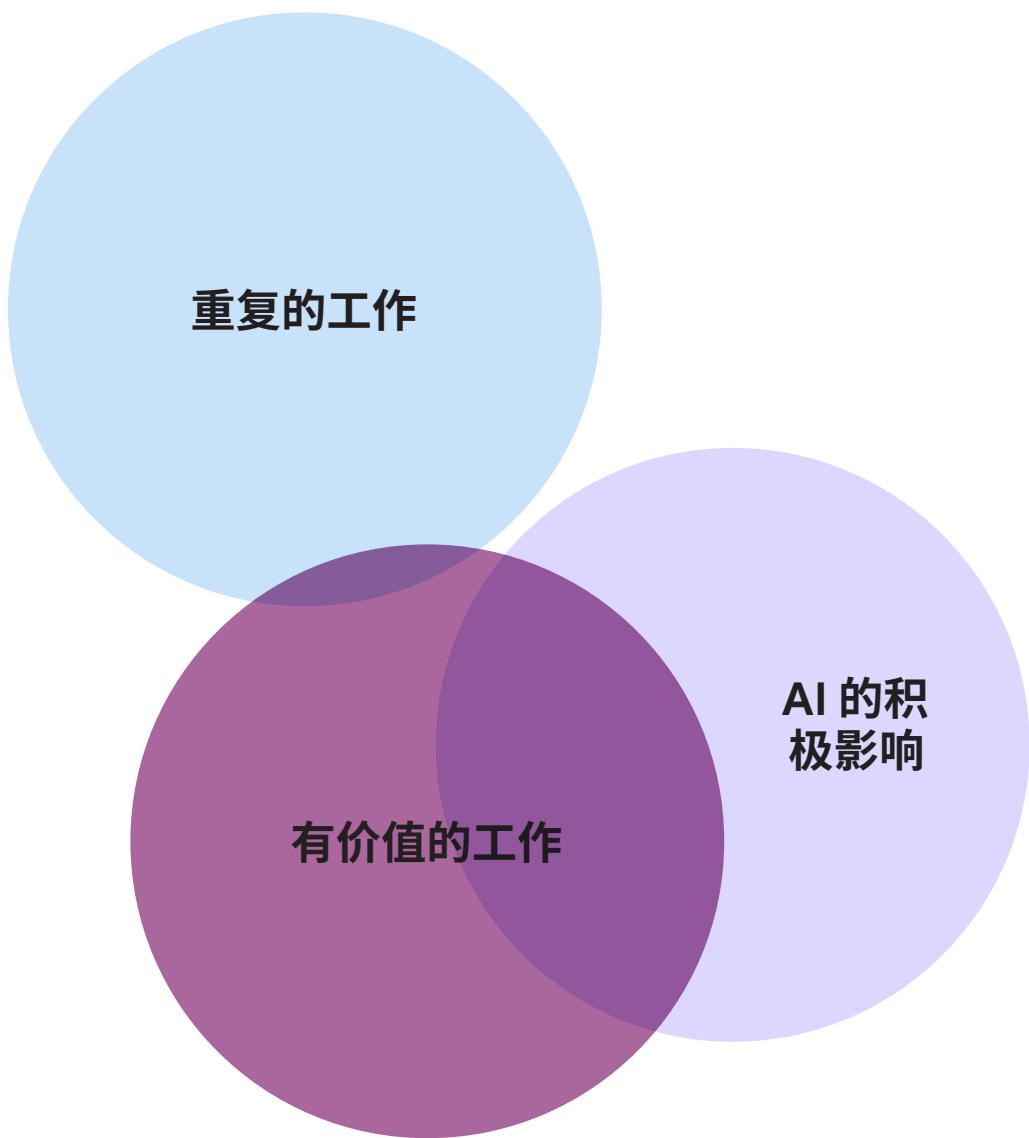


图 8:图中呈现的并非真实数据,而是我们的假设:AI 能帮助我们完成有价值的工作,但对重复工作没有帮助。

AI 对开发工作流程有积极影响

上一节探索了个人层面的成果。接下来要讨论的成果主要涉及到流程、代码库和团队协调。我们衡量了下面列出的指标：

代码复杂度	代码的精细度和复杂度在多大程度上降低了效率。
技术债	过去 6 个月中, 主应用或服务中现有的技术债在多大程度上影响了效率。
代码审核速度	为主应用或服务完成一次代码审核平均所需的时间。
审批速度	一般情况下, 对于主应用或服务, 从提出更改代码到获得批准可以用于生产所需的时长。
跨职能团队 (XFN) 协调	在多大程度上同意以下陈述: “在过去 3 个月中, 我能够与跨职能团队的成员高效协作。”
代码质量	在过去 6 个月中, 对主服务或应用底层代码质量的满意或不满意程度。
文档质量	对内部文档 (手册、自述文件、代码注释) 的看法, 考察方面包括可读性、易寻性、更新及时性及提供支持的力度。

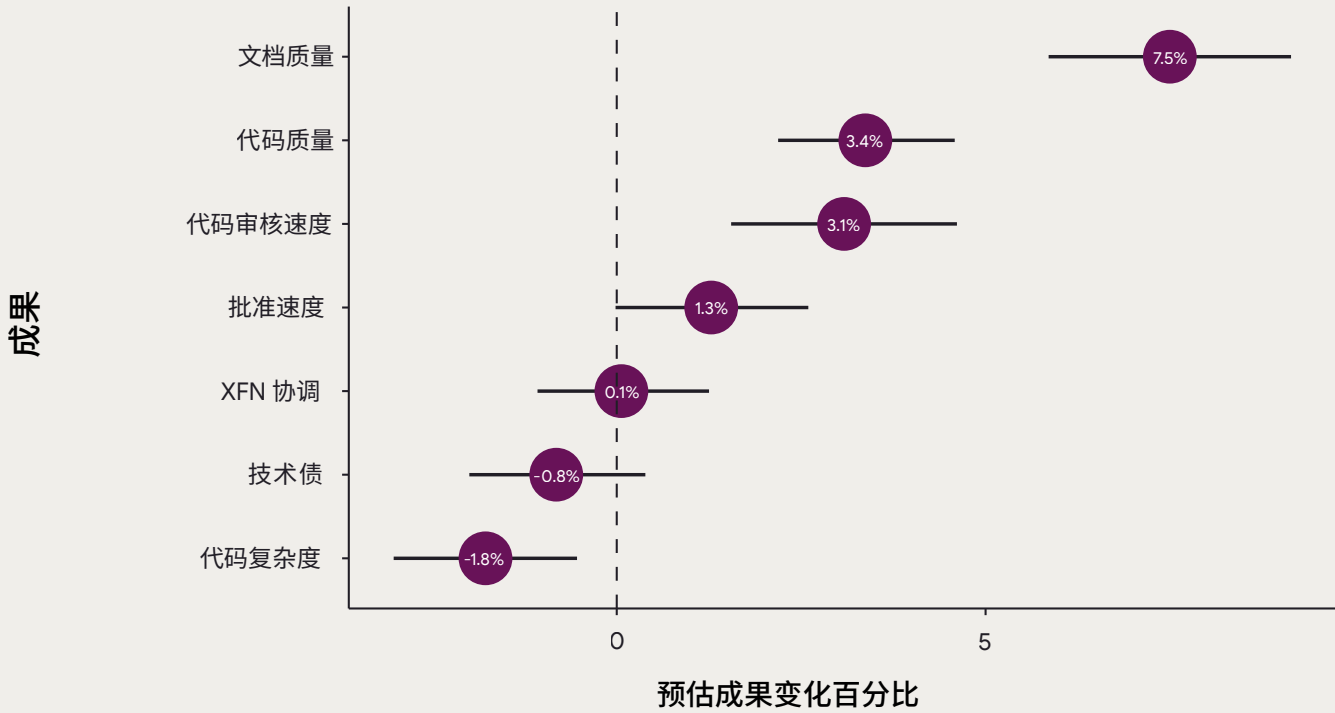
与之前一样,我们在这一阶段的目标是了解这些方面是否会因 AI 的采用而发生变化。我们尽可能准确地预估了将 AI 使用率提高 25% 后,相关成果的变化,图 9 用图表的形式进行了直观呈现。

从总体上看,提高 AI 使用率能带来积极的影响。本部分的主要结果如下。

AI 使用率提高 25% 的结果是…

- 文档质量提高 7.5%
- 代码质量提高 3.4%
- 代码审核速度提高 3.1%
- 审批速度提高 1.3%
- 代码复杂度下降 1.8%

如果 AI 使用率提高 25%...



点 = 估算值
误差线 = 89% 的不确定性区间
图 9: AI 使用率对组织的影响。

根据“[人工智能:采用情况和态度](#)”一章中的数据, AI 最常见的用途是编写代码。67% 的回复者表示, AI 可帮助他们完善代码。这里进一步印证了这一观点。AI 似乎可以改进代码质量并降低代码复杂度(图 9)。如果能结合对旧代码的重构, 高质量的 AI 生成代码可能会从总体上优化代码库。加上人们可以用 AI 生成更易于访问的优质文档, 代码库可因此得到进一步的改善(请参见[人工智能:采用情况和态度](#))。

优质代码更易于审核和获批。结合 AI 辅助的代码审核, 我们可以加快审核和批准速度, 这一点已在数据中充分体现出来(图 9)。

当然, 代码审核和批准速度提高, 不等于代码的审核和批准流程更好、更完善。速度的提高有可能是因为我们在流程中过于依赖 AI 的帮助, 或者有点过于信任 AI 生成的代码。这方面与图 9 中呈现的规律并不矛盾, 但也仍有待进一步研究。

此外, 我们还不清楚代码和文档质量的提升, 究竟是因为 AI 生成了这些内容, 还是因为 AI

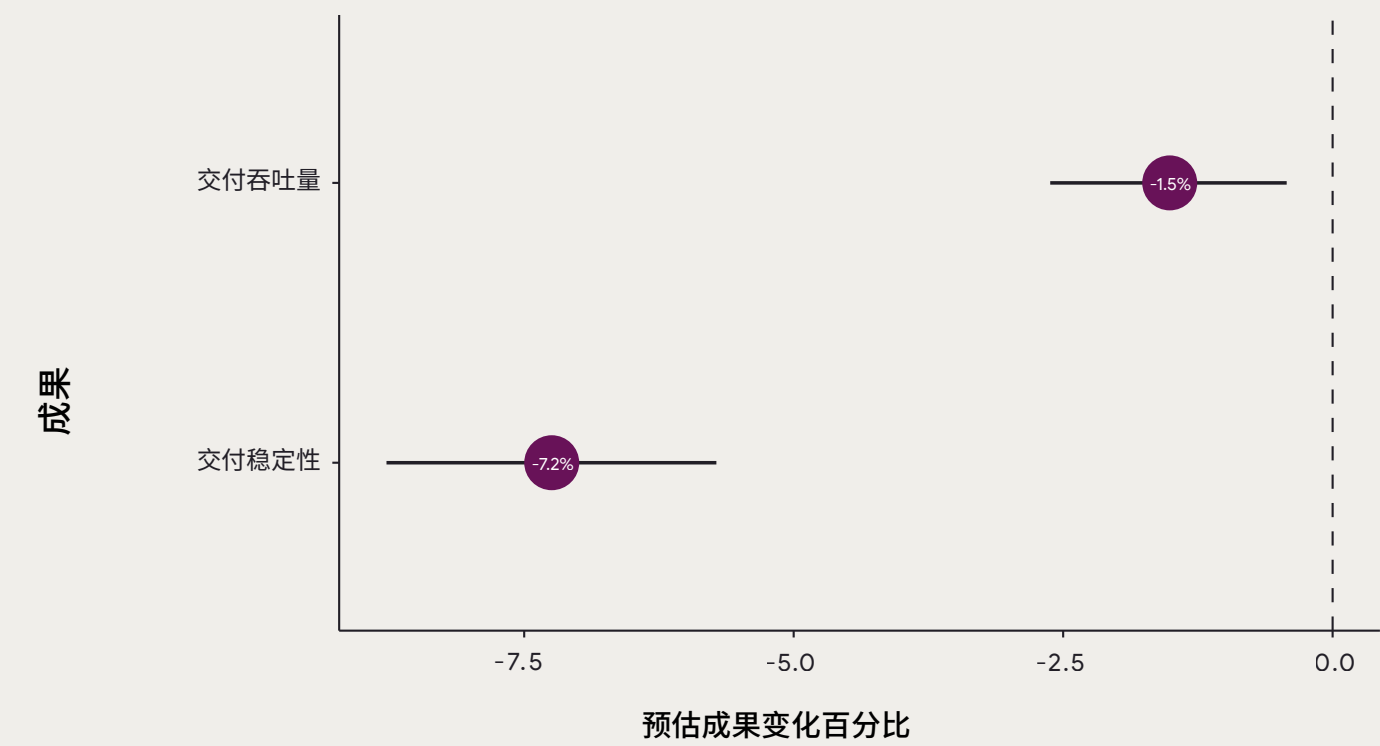
增强了我们从原本被视为劣质的代码和文档中获取价值的能力。会不会是因为在使用 AI 的过程中, AI 的强大功能足以帮助我们理解代码和文档, 从而让我们将优质代码和文档的门槛降低了一些呢? 这些规律的出现可能是两方面共同作用的结果, 因此在解读这些规律时, 不能只从一面去理解。

在这些规律中, 可以明确的一点是, AI 能帮助人们从所依赖的文档中获取更多信息, 从使用的代码库中获取更大价值。AI 还有助于减少代码审核和批准流程中代价高昂的瓶颈问题。目前尚不明确的是, AI 具体是如何做到这一点的, 以及这些好处是否会继续积极影响下游, 例如改进软件交付。

AI 正在损害交付绩效

过去几年, 我们看到软件交付吞吐量和软件交付稳定性的指标开始出现一定程度上的脱节。尽管吞吐量和稳定性之间仍然存在关联, 但新证据表明, 这两个因素的表现足够独立, 需要分开考察。

如果 AI 使用率提高 25%...



点 = 估算值
误差线 = 89% 的不确定性区间
图 10: AI 使用率对交付吞吐量和稳定性的影响。

调研结果显示 AI 的采用对软件交付绩效产生了不利影响,这一点与我们的预期相反。我们看到交付吞吐量受到的影响较小,但可能是负面影响(AI 使用率每提高 25%,预计交付吞吐量会降低 1.5%)交付稳定性受到的负面影响较大(AI 使用率每提高 25%,预计交付稳定性会降低 7.2%)。图 10 直观呈现了这些数据。

在过去的研究中,软件开发流程的改进会相应地提高文档质量和代码质量、加快代码审核及获批速度的加快,降低代码复杂度,最终会改善软件交付。所以,看到 AI 改进了这些流程指标,却又似乎损害了交付吞吐量和稳定性指标,我们感到很惊讶。

根据往年的调研结果,我们推断,AI 为效率和代码生成速度带来颠覆性的范式转变,可能使得业界忽略了 DORA 的一项最基本的原则,即“小批量”的重要性。也就是说,由于 AI 让回复者能在相同时间内生成多得多的代码,因此更改列表的规模可能,甚至极有可能变大。DORA 的研究一贯表明,较大规模的更改通常执行速度更慢,且更容易引发不稳定性。

综上,我们的数据显示优化开发流程不会自动改善软件交付,至少在不遵循小批量和强测试机制等成功软件交付的基本原则的情况下是这样。

AI 对许多重要的个人和组织因素产生了积极影响,为高效软件交付创造了条件,这让我们感到乐观。但 AI 似乎不是解决一切问题的灵丹妙药。



高绩效的团队和组织使用了 AI, 但似乎对产品改进没有什么帮助。

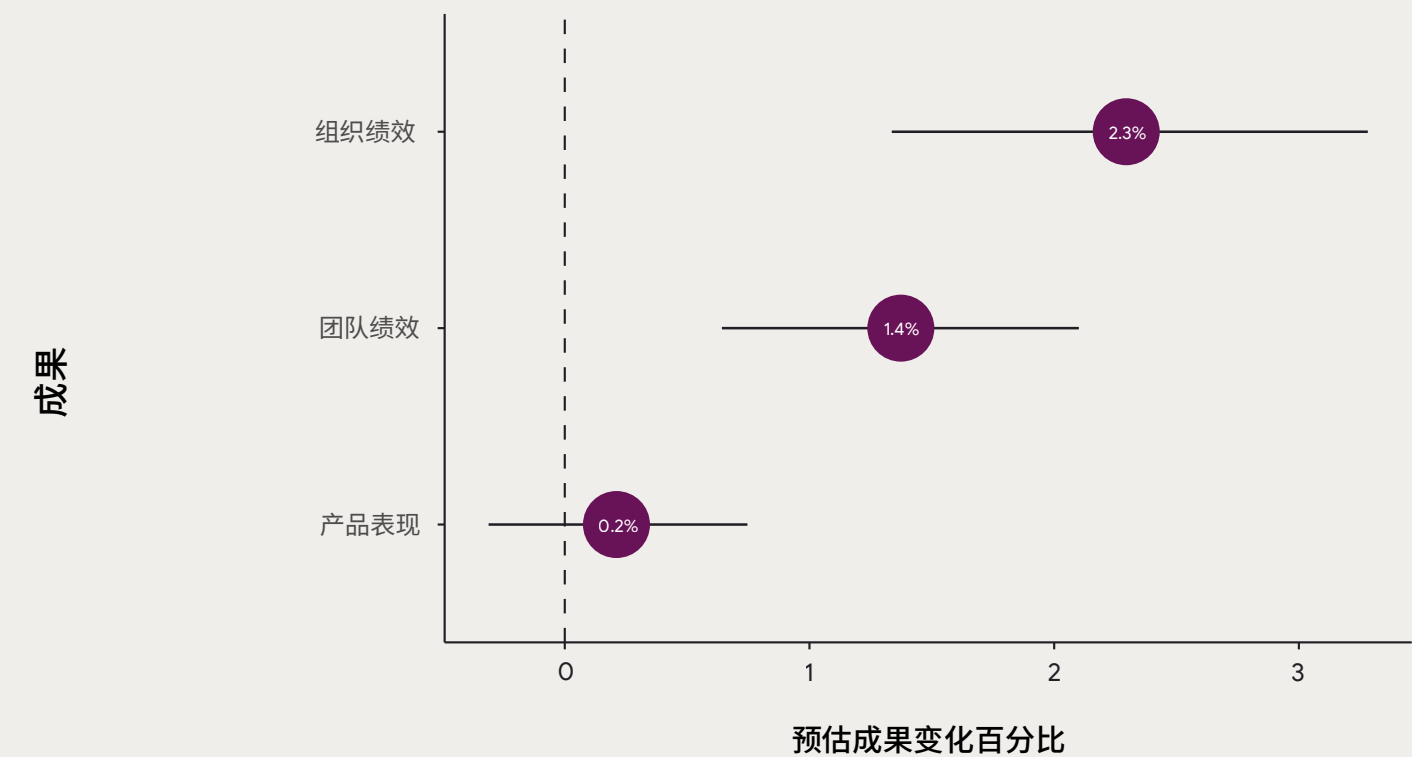
在这里, 我们研究了 AI 与最下游成果之间的关系:

组织绩效	这是一个因素分数, 它衡量了一个组织的整体绩效、盈利能力、市场份额、客户总数、运营效率、客户满意度、产品/服务质量以及实现目标的能力。
团队绩效	这个因素分数衡量了团队协作、创新、高效工作、支持彼此及调整适应的能力。
产品表现	这个因素分数衡量了产品的易用性、功能性、价值、可用性、性能 (如延迟) 及安全性。

这些成果与个人采用 AI 之间有何关系?分析这个问题非常困难且面临很多干扰,好比是在试图分析今天的午餐对组织今年的表现有何影响。

我们的逻辑是从微观层面(例如个人)跳到宏观层面(例如组织)进行分析。我们将在“研究方法”一章中讨论这种推论跳跃。现在,我们只看关联性:

如果 AI 使用率提高 25%...



点 = 估算值
误差线 = 89% 的不确定性区间
图 11:采用 AI 对组织绩效、团队绩效和产品表现的影响

AI 使用率每提高 25%，组织绩效预计提升 2.3%，团队绩效预计提升 1.4%。因此，采用 AI 对组织和团队的绩效似乎都有益(图 11)。但产品表现似乎与 AI 的采用并无明显关联。现在，我们可以转而尝试理解这些影响背后的原因。

我们推测，有助于改善团队和组织表现的因素与影响产品表现的因素有所不同。

团队和组织成功在很大程度上依赖沟通、知识共享、决策和健康的文化，而 AI 能克服这些方面的一些瓶颈，从而对团队和组织产生积极影响。

而产品的成功可能涉及更多因素。虽然优秀的产品与高绩效的团队和组织受一些相似因素的驱动，但前者与开发工作流程和软件交付的联系可能更加直接和紧密，而 AI 的引入对开发工作流程和软件交付的影响可能还不稳定。

部分原因可能在于，技术是优秀产品不可替代的构成要素。但要打造卓越的产品，还离不开艺术和同理心。对于那些认为一切问题都

可以通过计算解决的人来说，这一点或许难以接受。但产品开发中的某些要素，如创造力或用户体验设计，可能仍然(或永远)高度依赖于人类的直觉和专业知识。

组织绩效、团队绩效和产品表现之间存在必然关联，这是不变的事实。通过双变量相关性分析 (Pearson)，我们发现产品表现与团队绩效 ($r = 0.56$, 95% 置信区间 = 0.51 至 0.60) 和组织绩效 ($r = 0.47$, 95% 置信区间 = 0.41 至 0.53) 之间均存在中等程度的正相关关系。

这些成果相互影响，形成了明显的依赖关系。高绩效团队往往能够开发出更优秀的产品，但如果接手一个较差的产品，则可能会表现下滑。同样，高绩效的组织通过资源和流程滋养了高绩效的团队，但组织内部的困境也可能妨碍团队的表现。因此，如果 AI 的采用让团队和组织大为受益，那么我们可以合理推断产品也能从中受益。

AI 的采用才刚刚兴起，有些好处和弊端可能需要过段时间才能显现，这既可能是由于 AI 影响的内在特性所致，也可能是因为高效使用 AI 需要一个学习过程。

或许,事实很简单:我们正在探索 AI 如何帮助组织和团队,但尚未充分认识到它在产品创新和开发方面的潜力。图 12 尝试呈现了相关变化趋势。

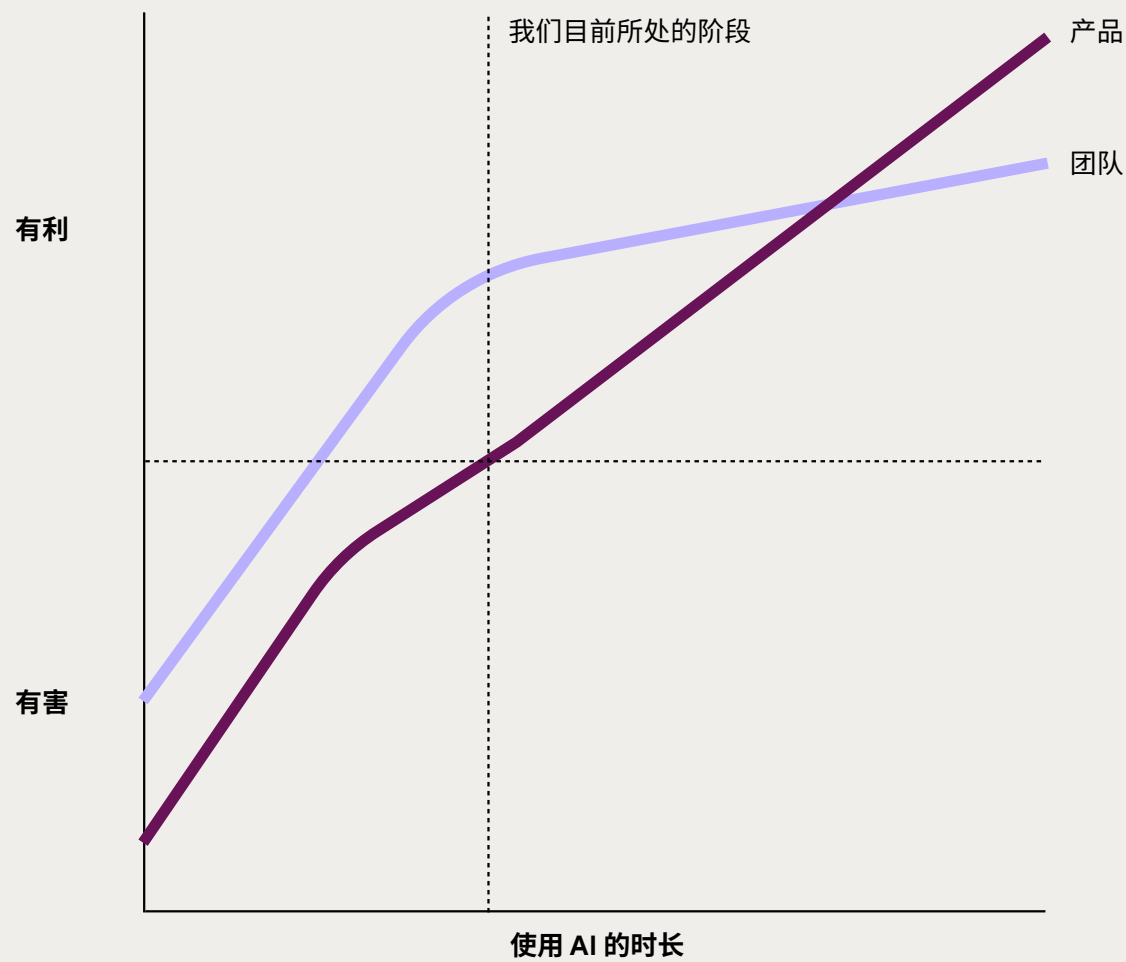


图 12: 不同学习曲线的示意图。这是为演示而进行的抽象概括,并非根据真实数据绘制。

接下来该怎么做呢？

我们希望了解 AI 的潜力，因为 AI 已经对个人、团队和组织起到了帮助作用。正在浮现的种种规律证明 AI 的影响并非空谈，而是正在发生。

有明确的证据支持采用 AI。但我们也很清楚，我们会面临大量潜在的阻碍和发展的阵痛，以及 AI 可能带来的各种负面影响。

以下是有关规划 AI 采用战略的一些建议：

大规模采用 AI 可能并不像按下播放按钮那样简单。一个周密、透明且灵活的战略可能会带来可观的收益。这样的战略需要领导者、团队、组织、研究人员和 AI 开发者共同制定。

领导者和组织需要想办法，优先在对员工最有帮助的领域采用 AI。

清晰定义 AI 的使命和使用政策，为组织和团队赋能。

向员工透明传达 AI 的使命、目标和采用计划。通过阐明宏观愿景和具体政策（如允许在哪些场景中允许使用 AI 编写代码、可以使用哪些工具等流程方面的问题），您可以减少员工的顾虑，并将 AI 定位为一个重要的工具，帮助员工专注于更有价值、更具成就感，以及更富创造性的工作。

创造一种围绕 AI 持续学习和实验的文化。

营造鼓励持续探索 AI 工具的环境, 允许个人和团队花时间去发现有益的用例, 并赋予他们自主制定行动计划的权利。通过在沙盒或低风险环境中进行实操, 建立对 AI 技术的信任。建议着力开发稳健的自动化测试系统, 进一步降低风险。实施一个衡量框架, 这个框架不应依据采用程度, 而是应根据有意义的下游影响来评估 AI, 即考察 AI 是否能促进员工成长、让使用贵公司产品的用户获益, 以及激发团队潜力。

认识到 AI 的利弊, 并利用 AI 的好处来获取竞争优势。

承认存在潜在的不足, 如花在有价值的工作上的时间减少、过度依赖 AI、在某方面获得的好处给其他方面带来问题, 以及软件交付稳定性和吞吐量受到的影响, 您便可以避开陷阱, 并在组织和团队层面引导大家正确利用 AI。深入理解 AI 的潜在优势和弊端, 将有助于您加速学习进程、为探索提供支持, 并将所学转化为行动, 获得真正的竞争优势。

毋庸置疑, AI 的采用非常值得兴奋, 但也存在着更多需要学习的地方。DORA 将继续保持关注, 并像过去十年那样, 尽力提供真实、准确且有益的观点。

¹. <https://www.goldmansachs.com/insights/top-of-mind/gen-ai-too-much-spend-too-little-benefit>

². <https://www.goldmansachs.com/insights/articles/AI-poised-to-drive-160-increase-in-power-demand>

³. <https://www.washington.edu/news/2023/07/27/how-much-energy-does-chatgpt-use/>

⁴. <https://www.gatesnotes.com/The-Age-of-AI-Has-Begun>

⁵. <https://www.businessinsider.com/ai-chatgpt-homework-cheating-machine-sam-altman-openai-2024-8>

⁶. <https://www.safe.ai/work/statement-on-ai-risk>

⁷. <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>

⁸. https://www.gitclear.com/coding_on_copilot_data_shows_ais_downward_pressure_on_code_quality

⁹. <https://www.nytimes.com/2024/04/15/technology/ai-models-measurement.html>

¹⁰. <https://dora.dev/capabilities>

¹¹. 需要说明的是, 这不是一种独创的方法。但在该领域, 此方法仍有一定的独特性。

¹². (P[N]), 如 (P1), 是指访谈受访者的化名。

平台工程设计



简介

平台工程设计是一门新兴的工程学科,在行业内获得了越来越多的关注,发展势头良好。Spotify 和 Netflix 等行业领军企业,以及《Team Topologies》¹(团队拓扑)等图书都为它贡献了热度。

平台工程设计是一门社会技术学科,工程师既要关注不同团队之间的社交互动,还要关注自动化、自助服务和流程可重复性的技术问题。包括 DORA 在内的多家机构早在多年前就已经在研究平台工程设计的概念。

一般来说,我们的研究重点是如何将软件交付给外部用户,而平台团队的成果通常是一套面向内部的 API、工具和服务,用于支持软件开发和运营生命周期。

在平台工程设计中,大量精力和关注都投入到了通过构建“黄金路径”来改善开发者的体验上。“黄金路径”是高度自动化的自助式工作流程。平台用户在交付和运营应用时,可以通过它与所需的资源交互。这些路径的目的是简化软件构建和交付的复杂性,使开发者只需关注代码本身。

通过黄金路径实现自动化的任务包括新应用的配置、数据库配置、架构管理、测试执行、构建和部署基础设施配置以及 DNS 管理等。

平台工程设计的一些概念,比如将某项功能向下移动到共享系统中(有时称为“下移”),²看似与“谁构建,谁运行”这类方法相悖。但我们认为平台工程设计是一种在整个组织中推广这些实践的方法,因为一旦平台具备了某个功能,团队基本就可以通过采用平台来免费获得该功能。

例如,如果平台具备执行单元测试并直接向开发团队反馈结果的能力,且无需团队构建和管理测试执行环境,那么持续集成平台功能就可以让团队专注于编写高质量测试。在这个例子中,持续集成功能可以在更大规模的组织中推广,使多个团队更轻松地提升持续测试³和测试自动化⁴的能力。

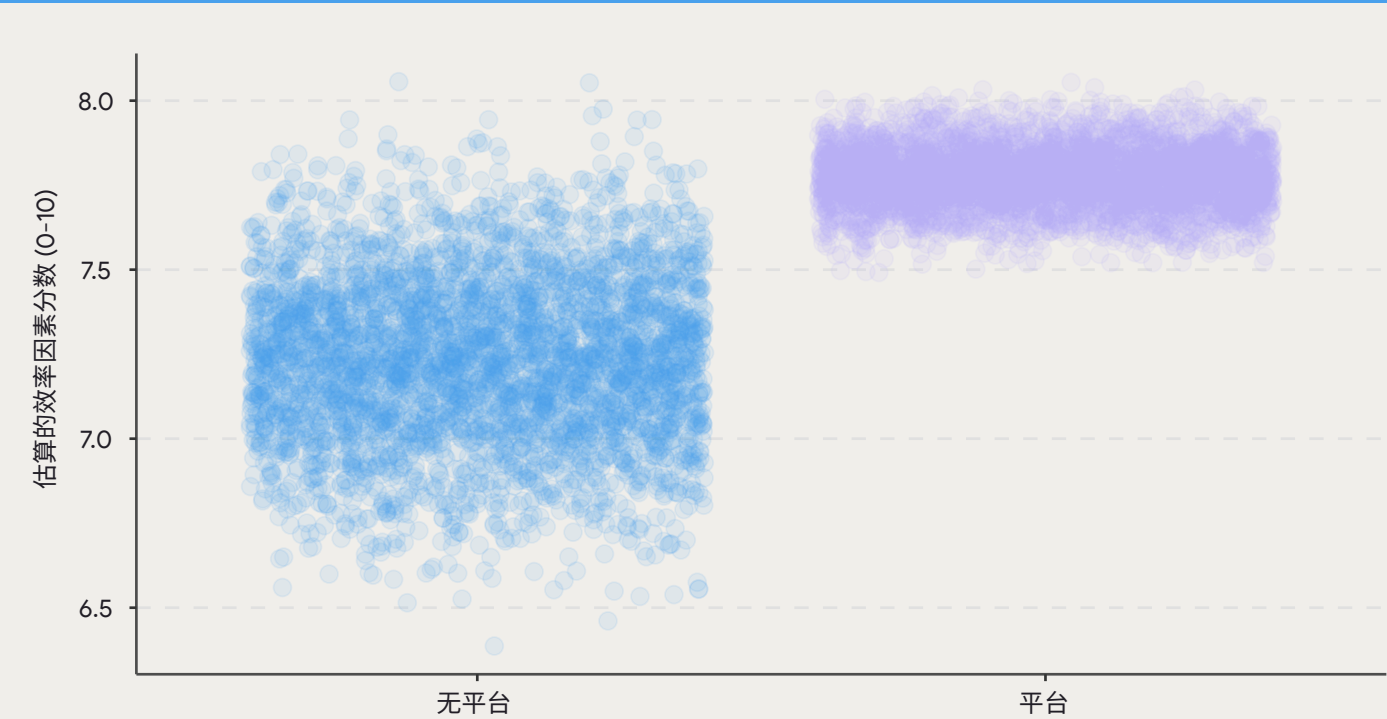


成功的关键在于在平台工程设计中坚持以用户为中心的理念(在内部开发者平台中,用户就是开发者),保持开发者独立性以及产品思维。这一点并没有让我们感到很意外,因为在今年和往年的研究中,以用户为中心已被证明是组织绩效提升的关键因素之一。⁵如果不采取以用户为中心的做法,平台带来的阻力可能大于助力。

在今年的报告中,我们尝试测试了平台和软件交付及运营绩效之间的关系。我们看到了一些正面的结果。在使用内部开发者平台

后,个人效率提高了 8%,团队绩效提高了 10%。此外,组织使用平台后,其软件交付与运营绩效提高了 6%。不过,这些收获也伴随着一些代价:吞吐量和更改稳定性分别下降了 8% 和 14%,这一结果令人意外。

在后面的部分,我们将更深入地分析这些数据、细微差别以及调查中发现的一些意外情况。无论您的平台工程设计计划是刚刚启动还是已执行多年,应用这些重要发现都能帮助您提升平台的表现。



每个点代表一个最可信的效率平均分估值,共 8000 个点
图 13:使用和不使用内部开发者平台时的个人效率因素。

平台工程设计的前景

由于平台工程设计对效率和生产力的潜在提升,内部开发者平台受到了软件开发者和IT行业的广泛关注。在今年的调查中,我们放宽了对内部开发者平台的定义,⁶并发现89%的受访者在使用的内部开发者平台。不同用户的交互模式差异很大。

这些数据点反映了行业对平台工程设计的广泛关注以及该领域的新兴态势。

总体上,平台发挥了积极作用。在使用内部开发者平台的情况下,个人的效率提高了8%,团队的绩效提高了10%。

除了效率,我们还看到使用平台对组织的整体绩效也有帮助:绩效提高了6%。总体而言,平台让组织能更快交付软件、满足用户需求并获取商业价值。

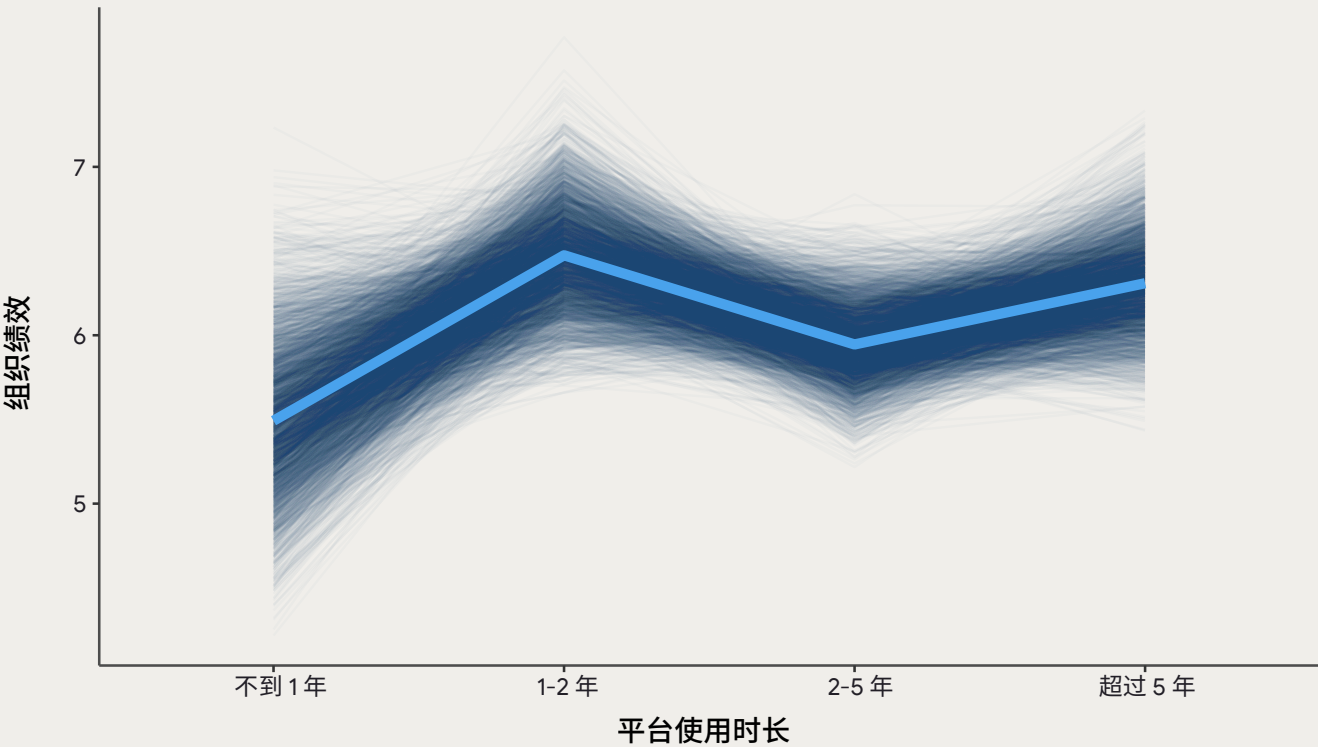


图 14:使用内部开发者平台时,组织绩效随平台使用时长变化的情况。

在结合平台使用时长来考察效率时,我们发现,在平台工程设计计划启动初期,绩效会有提升;随着平台使用时长增加,变得成熟,绩效会经历先下滑,而后恢复增长的过程。这种模式在转型计划中很常见,这些计划在初期会获益,但当好处一一实现后,就会面临新的挑战。

从长远来看,效率仍然是提高了的,这证明了内部开发者平台在软件交付和运营流程中的长期潜力。



主要调研结果 - 对开发者独立性的影响

在使用内部开发者平台交付软件时,开发者独立性对个人和团队的效率产生了显著影响。开发者独立性是指“开发者可在整个应用生命周期内独立完成任务,无需依赖支持团队”。

当平台用户无需支持团队即可独立完成任务时,团队和个人的效率均提升了 5%。这一发现印证了平台工程设计的关键原则之一,即支持自助式工作流。

对于平台团队而言,这一发现尤为重要,因为它突显了平台工程设计流程中的一个重要环节,即收集用户反馈。尽管问卷回复未揭示哪种反馈形式最有效,但常见的方法包括非正式交流、问题跟踪,以及持续的协同开发、问卷调查、遥测和访谈。

这些方法能有效判断用户是否可独立完成任务。调查数据还显示,缺乏对平台的反馈会带来负面影响。

次要调研结果 - 专门设立平台团队的影响

有趣的是, 专门的平台团队对个人效率影响不大, 但在团队层面带来了 6% 的效率提升。这种影响的不均衡性让我们感到意外, 这表明设立专门的平台团队对个人有所帮助, 但对团队整体影响更大。

团队通常由多名开发者组成, 他们职责不同, 技能也不同。与单个工程师相比, 团队的任务更加多样化。设立专门的平台工程设计团队有助于为团队中不同的任务更好地提供支持。

总体来看, 内部开发者平台的引入对效率有积极影响。

关键因素是:

以用户为中心的方法。它通过自助服务和可自主完成的工作流程保证了开发者的独立性。在平台中, 用户就是内部工程设计和开发团队。

像其他转型计划一样, 平台工程设计也会产生“J 曲线”效应, 效率的提升会通过持续改进趋于稳定。

意外的弊端

尽管平台工程设计对提升团队和个人的效率及组织绩效起到了很大作用,但它也有一个出人意料的弊端:吞吐量和更改稳定性有所下降。

我们意外地发现,不稳定的更改与工作倦怠之间存在非常有趣的联系。

吞吐量

我们看到相较于不使用平台的组织,使用平台的组织在吞吐量方面低了约 8%。我们对背后的原因做了一些猜想。

首先,更改必须经过一定流程,才能部署到生产环境,这可能导致更改吞吐量整体下降。通常,当使用内部开发者平台进行软件构建和交付时,系统与团队之间的“交接”次数会有所增加。

例如,当代码提交到源代码控制系统时,不同系统会自动接收它并进行测试、安全检查、部署和监控。

每个交接环节都有可能增加整个流程的时间,导致吞吐量下降,但总体上增强了完成工作的能力。

其次,对于那些“必须完全依赖平台来完成整个应用生命周期任务”的回复者来说,吞吐量下降了 6%。虽然这中间并不存在明确的因果关系,但其也可能与我们的第一个猜想有关。

如果在平台引入后,开发和发布软件所涉及的系统和工具数量增多,那么,强制使用可能并不适合特定用途的平台,或者流程中自然而然地增加了延迟,就难免导致效率下降。

为应对此类问题,在平台工程设计计划中,坚持以用户为中心并保持用户独立性至关重要。

更改不稳定性与工作倦怠

我们观察到,在使用内部开发者平台开发和运营应用时,应用更改的稳定性竟然下降了14%。这表明使用平台时,更改失败率和返工率显著增加。

更有趣的是,调研结果显示不稳定性和使用平台的双重作用会加剧工作倦怠。这并不意味着使用平台会导致倦怠,而是说不稳定性与使用平台这两个因素结合起来,会让工作倦怠问题更为突出。与吞吐量下降相似,我们尚未完全理解工作倦怠发生变化的原因,但对此有一些猜想。

首先,使用平台让开发者和团队更有信心推动更改,因为即使更改失败,也能迅速修复。在这种情况下,不稳定性升高未必是坏事,因为使用平台会鼓励团队实验和交付更改,而这难免会导致更高的更改失败率和返工率。

第二种猜想是,平台可能无法有效保证更改和/或生产环境部署的质量。

另一种可能性是,平台提供的自动化测试功能可以运行应用中的所有测试,但应用团队并未充分利用这一功能,而是优先考虑吞吐量并忽视质量,因此未能改进测试流程。在这两种情况下,错误的更改可能通过了流程,导致频繁返工。

第三种可能是,更改不稳定性和工作倦怠程度高的团队更乐于创建平台,希望借助平台提升稳定性和减轻工作倦怠。这是合理的,因为组织通常认为平台工程设计能减轻工作倦怠并提升持续交付小规模更改的能力。基于此猜想,采用平台工程设计或许标志着组织正面临工作倦怠和更改不稳定的问题。

在前两种情况下,平台导致的返工可能加重团队负担,也可能因此加剧工作倦怠。尤其是在第二种情况下,平台让错误更改得以应用,从而加重了工作倦怠。然而,在这些情况下,由于团队或个人能够持续发布更改和功能,他们可能仍然感觉自己的效率很高。而在第三种情况中,采用平台工程设计的组织本身就面临着更改不稳定性和工作倦怠的问题,他们将平台视为应对这些挑战的解决方案。

平衡利弊

虽然平台工程设计并非万能之法,但在软件开发和运营的整体流程中,它能发挥强大的作用。与其他学科一样,平台工程设计既有优势,也存在一定的局限性。

根据我们的研究,在启动平台工程设计计划时,您可以通过几项措施来平衡利弊,帮助组织在享受平台工程设计带来的优势的同时,有效监控和管理潜在的负面影响。

首先,优先构建既能保持开发者独立性又能支持自助服务的平台功能。在此过程中,需注意避免在应用生命周期的所有环节都强制要求只使用平台,以免影响开发者的独立性。

好的做法是,平台应为用户提供脱离平台工具和自动化功能的途径,以增强独立性。尽管这会在一定程度上加剧复杂性,但可以通过设立专门的平台团队来缓解这一问题。平台团队需积极与平台用户协作并收集反馈。

协作与反馈能够让平台计划更好地服务于用户需求,有助于平台的长期成功。从我们的数据中可以看出,反馈收集的方式多种多样,建议采用多种方法以尽量多收集反馈。

其次,密切监控应用更改的不稳定性,并尝试判断不稳定性是故意设计的,还是意外出现的。平台有潜力在不稳定的环境中发挥实验的力量、提升效率和规模化改进绩效。

然而,这种不稳定性在带来上述好处的同时,也可能加剧不稳定性和工作倦怠,因此在整个平台工程设计过程中必须密切关注并管理不稳定性。了解自身对不稳定性的容忍度至关重要。可以借助站点可靠性工程(SRE)中的服务等级目标(SLOs)和错误预算,帮助评估风险容忍度和平台在支持安全实验方面的效力。

虽然内部开发者平台高度关注开发者的体验,但很多其他团队(如数据库管理员、安全和运营团队)也在软件成功交付和运营中扮演着重要角色。

在平台工程设计计划中,应推动各团队之间形成以用户为中心和持续改进的文化,并确保他们的目标与组织目标一致。

这样一来,平台的功能、服务和 API 能更好地服务于个人和团队需求,支持他们更高效地交付软件和实现业务价值。



1. Skelton, Matthew 和 Pais, Manuel。2019 年。《Team Topologies: Organizing Business and Technology Teams for Fast Flow》(团队拓扑:组织业务和技术团队以实现快速流动)。IT Revolution Press。 <https://teamtopologies.com/>
2. <https://cloud.google.com/blog/products/application-development/richard-seroter-on-shifting-down-vs-shifting-left>
3. <https://dora.dev/capabilities/continuous-integration/>
4. <https://dora.dev/capabilities/test-automation/>
5. <https://dora.dev/research/2023/>, <https://dora.dev/research/2016/>
6. <https://dora.dev/research/2024/questions/#platform-engineering>

开发者体验



要点

即使有 AI 的协助,最终构建软件的也是人。开发者的工作体验是关乎组织成败的一个基本要件。

在今年的报告中,我们再次发现,如果开发者构建的产品与用户需求保持一致,员工和组织都能快速发展。当开发者抱着以用户为中心的心态构建软件时,他们效率更高,发生工作倦怠的可能性更低,也更有可能打造出优秀产品。

归根结底,软件是为人而开发。组织有责任优化环境,让开发者可以专注于打造可带来优秀用户体验的软件。我们还发现,在工作重心不会频繁变化的稳定环境中,效率出现了小幅度提升(这一点意义重大),员工工作倦怠感则大幅下降。

对于所开发产品的质量以及产品开发者的总体体验,环境因素有着巨大影响。

将用户放在第一位, 其他一切(几乎)自然会水到渠成

我们认为开发者的工作很有意义, 他们站在技术进步的前沿, 帮助塑造了我们生活、工作和与世界交互的方式。

他们的工作基本与人紧密联系在一起。这里的“人”是指他们开发的软件和应用的用户。开发者的工作环境往往更注重功能和创新, 但不太注重这些功能是不是能给产品使用者带来真正的价值。

针对上面的论点, 我们提供了令人信服的证据, 来证明将最终用户放在首位的软件开发方法对员工和组织都有益。

今年, 我们设计的问题着重于了解开发者是否:

1. 基于用户反馈重新审视功能并调整功能的优先级
2. 了解用户想要用某个应用/服务做什么
3. 相信以用户为焦点是企业成功的关键
4. 相信用户体验是企业的首要任务





我们的发现及意义

我们的数据明确显示，以用户的需求和痛点为导向的组织，能够创造出更好的产品。

我们发现，以用户为焦点可以提高效率和工作满意度，同时减轻工作倦怠的风险。

关键是这些好处不仅限于员工个人，还扩大到了整个组织。在往年的报告中，我们强调了高绩效组织能够快速可靠地交付软件。这意味着成功离不开出色的软件交付绩效。

然而，我们的数据显示还有一条通向成功的路径：

开发者及其雇主，乃至整个组织，可以创建以用户为中心的软件开发方法。

我们发现，当组织知道并理解用户需求时，软件交付的稳定性和吞吐量不再是产品质量的决定性因素。只要将用户体验放在首位，产品质量就差不了。

如果组织不以用户为焦点，不将用户反馈融入开发流程，则只能通过加倍努力确保稳定快速的交付来提高质量（见图 15）。

我们理解，一些组织可能本能地将重点放在功能开发和技术创新上。从表面上看，这种策略很合理。毕竟，与普通用户相比，开发者更了解技术细节。

但是，如果仅凭对用户体验的猜想开发软件，开发者很可能会创造出一些华而不实的功能。¹

当组织和员工了解用户的体验时，他们就更有可能会打造出能满足用户真实需求的功能。解决用户的真实需求会提高功能得到实际应用的可能性。

专注于为用户打造功能，您就能创造出受欢迎的产品。

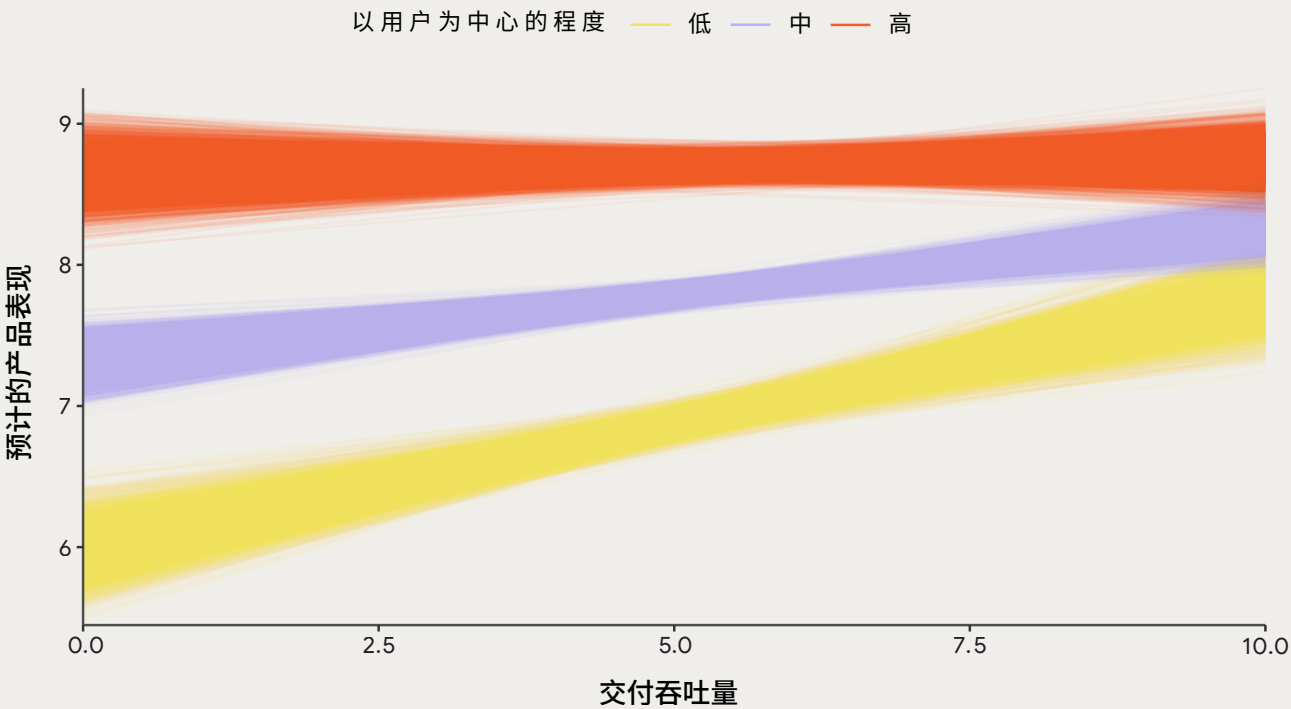


图 15:3 种用户关注度所对应的产品表现和交付吞吐量

为什么在软件开发中,以用户为中心是一种重要的理念和做法?

学术研究表明,从工作中找到使命感对员工和组织都有益。^{2,3}

例如,近期调查显示,93%的员工称他们很看重工作的意义。⁴另一项调查也得出了类似的结论:如果工作足够有意义,受访者平均愿意放弃未来总收入的23%。⁵

真没想到员工竟愿意做出这么大的牺牲。我们可以从中一瞥人们动力的来源,并知道人们想将时间花在有意义的事情上。

“如果每个人在公司的工作都能积极影响到公司外部的人,或者本地社区的人,那就太好了。但现实并非总是如此,也不可能每个人都有这样的机会。很多自动驾驶企业的宏伟愿景是让人们能在高速公路上边开车,边睡觉。我的目的不在此。我想帮助无法开车的人自由出行,去想去的的地方,做想做的事。”(P2)⁶

提供明确的方向:

采用以用户为中心的做法来开发软件,可以从根本上改变开发者对工作的看法。相较于交付天马行空的功能并猜测用户是否会使用,开发者可以根据用户反馈来确定优先构建哪些功能。

这种方法让开发者能确信自己开发的功能有存在的理由。他们的工作突然有了意义:确保人们在使用产品和服务时获得绝佳的体验。开发的软件与它的应用场景不再是割裂的。

开发者可以通过软件看到自己的工作产生的直接影响。

“我们作为一家公司,承受着交付压力。因此那些吸引人的好功能,或是关于想如何改进的讨论要点,在最近组织结构发生变化后,似乎一切都让位给了交付。我们更关注交付,而非质量。我个人认为这是一个大问题。”(P9)

增进跨职能协作：

即便是最有才华的开发者也不是孤军奋战。打造优质产品需要很多人的协作，这些人才能各异，优势互补。

在开发中采用以用户为中心的方法，让开发者能够在组织内进行跨职能协作。这样，他们的责任已经不再局限于单纯地交付软件。现在，他们成了团队的一分子，共同为使用产品的人创造卓越体验。

这种软件开发方法可以帮助开发者打破孤岛，统一认识、增进团队合作并创造更多向彼此学习的机会。解决问题的方式多种多样。我们不止要解决技术问题，还要想清楚如何以对用户最有益的方式来解决问题。

这种方法可以帮助增强员工互动，员工也会更积极地思考和学习，因而可以避免工作倦怠和由之带来的发展停滞的感觉。

组织可以做些什么？

根据我们的调研结果，我们建议组织多花点时间和资源去了解用户。了解应用/服务是为谁而开发，以及他们遇到的痛点是什么。我们非常相信这是一项值得的投资。

避免想当然地揣测用户，要在用户的使用场景中观察他们的行为，向他们提问，虚心听取他们的反馈并作相应调整。这样，开发者的效率会更高，产生工作倦怠的可能性更小，同时还能开发出更优秀的产品。

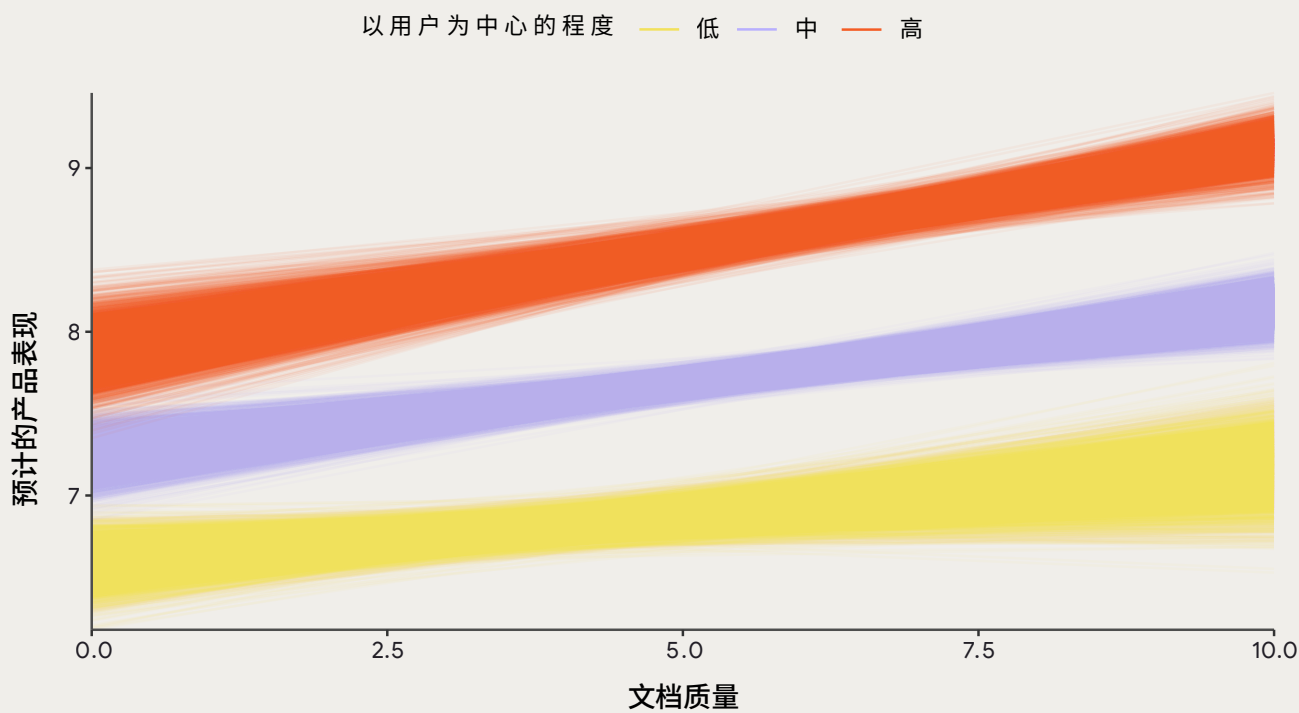
将优质文档与以用户为中心的软件开发方法结合起来会产生强大的作用。

那些关注用户的团队，其产品表现也更优异。如果在关注用户的同时提供优质内部文档，产品表现方面的提升会更大（见图 16）。这一发现与我们看到的行为一致，即文档会放大技术能力对组织绩效的影响。⁷

文档有助于将用户信号和反馈传递给整个团队，并最终体现到产品上。

我们看到，如果没有用户信号，内部文档不会对预计的产品表现有显著影响。但是，如果团队有优质的内部文档，则文档中包含的用户信号会对产品表现产生更大影响。

我们从 2021 年开始在调研中关注文档的作用，每一年我们都会看到优质文档的巨大影响。在今年的调研结果中，我们将内部文档对预估产品表现的影响添加到了列表中。



此图中包含 12000 条线，这些线来自我们为推断最可能的规律所做的模拟
图 16：以用户为中心的三个等级对应的产品表现和文档质量

文档文化

《敏捷宣言》认为“好用的软件比面面俱到的文档更重要”。⁷但我们不断发现优质文档是软件好用的重要条件。

“面面俱到的文档”可能只是一个用来掩盖不良实践的说法,而这些不良实践可能就包括文档相关工作。这样的文档包括那些单纯为满足冗长的常规流程而创建的文档,或者那些旨在掩盖管理层和员工之间的不信任而创建的文档。不健康的文档文化可能还包括只编写文档,但不对其进行维护或整合。

按照我们衡量优质文档的标准,这些文档的分数可能会较低。这类内容可能并不是为了真正的用户撰写的,在工作中用起来效果也并不好。文档内容太多和内容不够一样令人头痛。

文档是否优质取决于易寻性和可靠性等属性。需要记住一点,内部文档的主要用户是您的同事,甚至可能是您自己,比如未来您在完成特定任务时需要参考。⁸拥有良好文档文化的团队都会重视更好地服务于这些用户。这再次证明了需要以用户为中心。

我们在下面汇总了有助于创建优质文档的做法,遵循这些做法能帮助您在自己的团队中建立健康的文档文化:

详细记录重要使用场景。

开展技术写作培训。

明确说明所有权和文档更新流程。

在团队内部分配文档工作。

将维护文档作为软件开发生命周期的一部分。

删除过期或冗余的文档。

在绩效考核和晋升中考察员工在文档工作方面的贡献。

频繁更改工作重点的危害

我们都理解这样的感受,您在过去几个月一直在忙着开发一个新功能,您知道这个功能是用用户需要的,因此目标明确,干劲满满。这时候,领导团队突然决定改变组织的工作重点。现在,您不清楚您的项目是会被搁置、取消、合并,还是调整。

这种常见情况可能会深深影响员工和组织。为此,我们研究了组织经常改变工作重点产生什么样的影响。

我们的发现及意义

总体上,调研结果显示,如果组织频繁更改工作重点,效率会出现幅度虽小但后果严重的下降,而工作倦怠感会大幅增加。

我们的数据显示,工作倦怠感很难缓解。我们考察了能否通过强有力的领导、良好的内部文档,以及采用以用户为中心的软件开发方法,来抵消不断更改工作重点对工作倦怠的影响。

答案是:不能。即使组织做到了上面的几个点,如果工作重点不够稳定,员工仍然容易产生倦怠。

为什么工作重点不稳定不利于员工的身心健康？

我们猜想，组织工作重点不稳定会导致期望不明、降低员工的掌控感，同时还会增加他们的工作负担，从而加剧员工的工作倦怠感。

需要明确的是，我们认为问题不在于改变工作重点本身。业务目标和产品方向一直在变。组织能够灵活调整工作重点，这对组织来说可能是有益的。

我们认为，真正对员工身心健康产生负面影响的是工作重点变动的频率。工作重点不稳定所带来的不确定感，暗示着更改工作重点是一个长期存在的问题。

长达数十年的学术研究揭示了慢性压力对健康和福祉的严重危害。⁹我们看到关于慢性压力的研究与我们的调研结果之间存在相似之处。持续的不稳定状态会增加不确定感，并降低人们的掌控感。这些因素叠加在一起，极容易让员工产生倦怠感。

如果工作重点较为固定，会发生什么？

此次的调研结果有一点令人费解。我们发现，如果工作重点较为固定，软件交付绩效会下降。不光交付速度变慢，稳定性也有所下降。

我们推测，这可能是因为在工作重点稳定的组织中，产品和服务通常运行良好，所以变动相对较少。还有一种可能性是，由于工作重点稳定，发布次数减少，但同时，单次发布的规模却大于建议值。

即便如此，我们仍然认为这是一个意外发现。您认为，是什么原因导致稳定的组织工作重点反而拖累了软件交付的速度和稳定性？

为终端用户构建 AI 是一项稳定的工作重点,但却未能实现稳定交付。

把 AI 技术融入到终端用户的体验中可以作为组织的一项稳定的工作重点。这听起来像是在吹捧 AI,但我们的这一研究发现对 AI 本身没有多大意义。

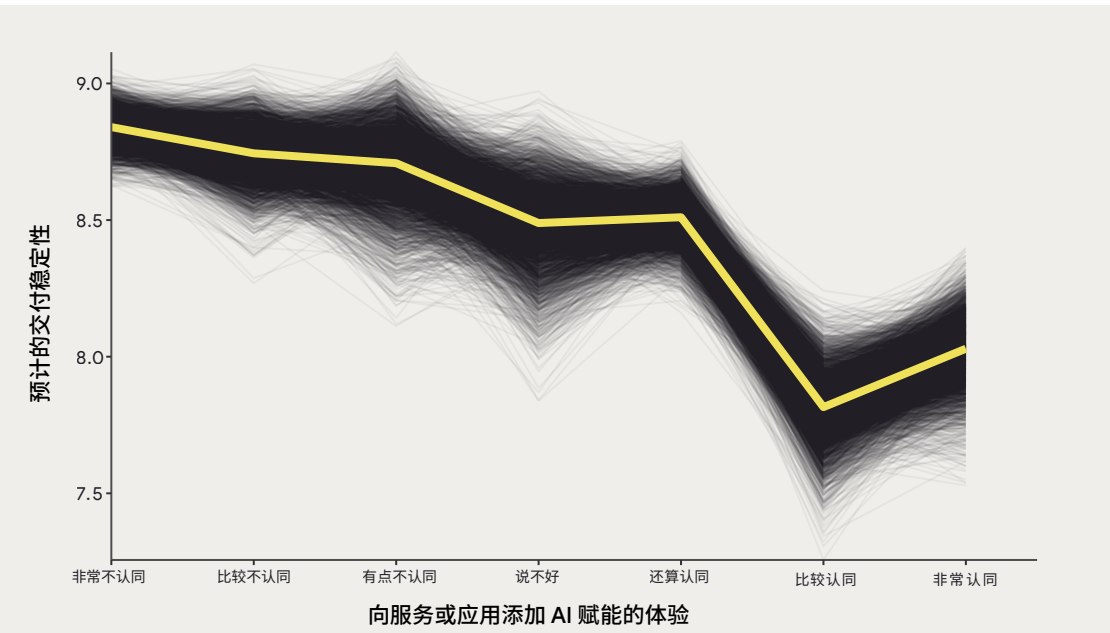
相反,我们相信加大力度构建 AI 可为组织提供明确的目标和方向,指引组织发展。这种明确性,而非 AI,才是让组织工作重点稳定下来的关键。

我们有必要强调这一点,因为它向我们展示了当新技术出现时,组织会面临什么。新技术会带来变化,而组织需要时间去适应。

这段时间,领导者们在摸索组织的最佳行动方案,工作重点可能会有变动。当尘埃落定后,组织明确了后续行动,工作重点会开始稳定下来。

工作重点稳定后,软件交付流程却不会马上稳定下来。我们的分析显示,在服务或应用中引入依托 AI 技术的体验将会伴随挑战 and 发展的阵痛。

我们发现,那些将资源转向 AI 的团队,其软件交付的稳定性比没有加大 AI 投资的团队下降了 10% 之多。下图直观地反映了这一挑战。



*为了推断出最可能的规律,我们做了 4000 次模拟,每条线都代表一次模拟
图 17:将依托 AI 技术的体验添加到服务或应用中对软件交付稳定性的影响

组织可以做些什么？

答案看似简单，但事实并非如此。根据我们的调研结果，我们建议组织专注于稳定工作重点。如果要抵消频繁更改工作重点给员工带来的倦怠感，这是一种可靠的方法。

我们的调研结果表明，频繁更改工作重点所产生的负面影响，会削弱优秀的领导者、良好的文档以及以用户为中心的软件开发做法所带来的积极作用。这让我们相信，除了保持稳定，组织为避免工作倦怠能做的并不多，但可以 (1) 稳定工作重点，并 (2) 尽力保证员工的日常工作不受工作重心频繁变动的影响。

-
1. <https://www.nngroup.com/articles/bridging-the-designer-user-gap/>
 2. <https://executiveeducation.wharton.upenn.edu/thought-leadership/wharton-at-work/2024/03/creating-meaning-at-work/>
 3. <https://www.apa.org/pubs/reports/work-in-america/2023-workplace-health-well-being>
 4. <https://bigthink.com/the-present/harvard-business-review-americans-meaningful-work/>
 5. <https://hbr.org/2018/11/9-out-of-10-people-are-willing-to-earn-less-money-to-do-more-meaningful-work>
 6. (P[N])，如 (P1)，是指访谈受访者的化名。
 7. <https://cloud.google.com/blog/products/devops-sre/deep-dive-into-2022-state-of-devops-report-on-documentation> 和《加速：2023 年 DevOps 现状报告》 - <https://dora.dev/research/2023/dora-report>
 8. <https://agilemanifesto.org/>
 9. 还有其他受众群体，例如管理者、监管者或审核者。
 10. Cohen S, Janicki-Deverts D, Miller GE. 《Psychological Stress and Disease》(心理压力与疾病)。JAMA。2007;298(14):1685–1687. doi:10.1001/jama.298.14.1685

引领转型

转型之路任重道远。今年,我们发现高绩效团队往往会优先考虑稳定性、以用户为中心、编写高质量的文档,并有优秀的领导者。我们的研究提供了一些有用的路径,可帮助您制定一个推动成功转型的策略。

我们发现,成功的关键是以持续改进的心态对待转型。我们的研究显示,表现较佳的团队了解阻碍其发展的变量有哪些,并将 DORA 指标作为基准,有条不紊地持续改进。取得长期成功需要在各个关键领域都表现出色,而十年的 DORA 研究为我们指明了四个具体且有效的方法,可以帮助您推动组织开始转型。



变革型领导力

在变革型领导模式中,领导者会唤醒员工的价值感和使命感,从而激励员工提高绩效,并推动大规模组织变革。

这类领导者从以下维度激励团队朝着共同目标努力:¹

愿景	他们对团队和组织的发展方向有着清晰的愿景。
正向沟通	他们肯定团队的价值;让员工以身为组织的一员为荣;并鼓励大家将变化的环境视为机遇。
智力刺激	他们鼓励团队成员从新的角度思考旧问题;并重新审视他们对工作的一些基本假设。
支持型领导力	他们在行动前会考虑他人的感受;行动时也会考虑其他人的需求。
个人认可	当团队成员表现优于平均水平时,他们会不吝表扬;并肯定团队成员工作质量的进步。

今年的调研结果表明,变革型领导力能有效促进员工效率的提高。我们看到,变革型领导力提高 25% 能使员工效率提高 9%。

变革型领导力能帮助组织提高效率,优秀的领导者还能:

- 降低员工工作倦怠感
- 提高工作满意度
- 提高团队绩效
- 提高产品表现
- 提高组织绩效

根据 2017 年的研究,上述领导力特质与 IT 绩效之间存在显著的统计学联系。高绩效团队的领导者在这五个特质上的得分都很高,而低绩效团队的领导者得分最低。此外,我们还发现变革型领导力和员工净推荐值 (eNPS, 即会向其他人推荐自己的公司的可能性) 之间有很强的关联性。

话虽如此,变革型领导力本身并不能直接提高绩效,而应被视为一种推动因素。

在提升技术和产品管理的能力和采用最佳实践方面,变革型领导力发挥着关键作用。具体的做法有:(1) 将权力和自主权下放给团队;(2) 向他们提供解决问题所需的数据和业务信息;以及 (3) 创建以价值交付而非功能交付为导向的激励机制。

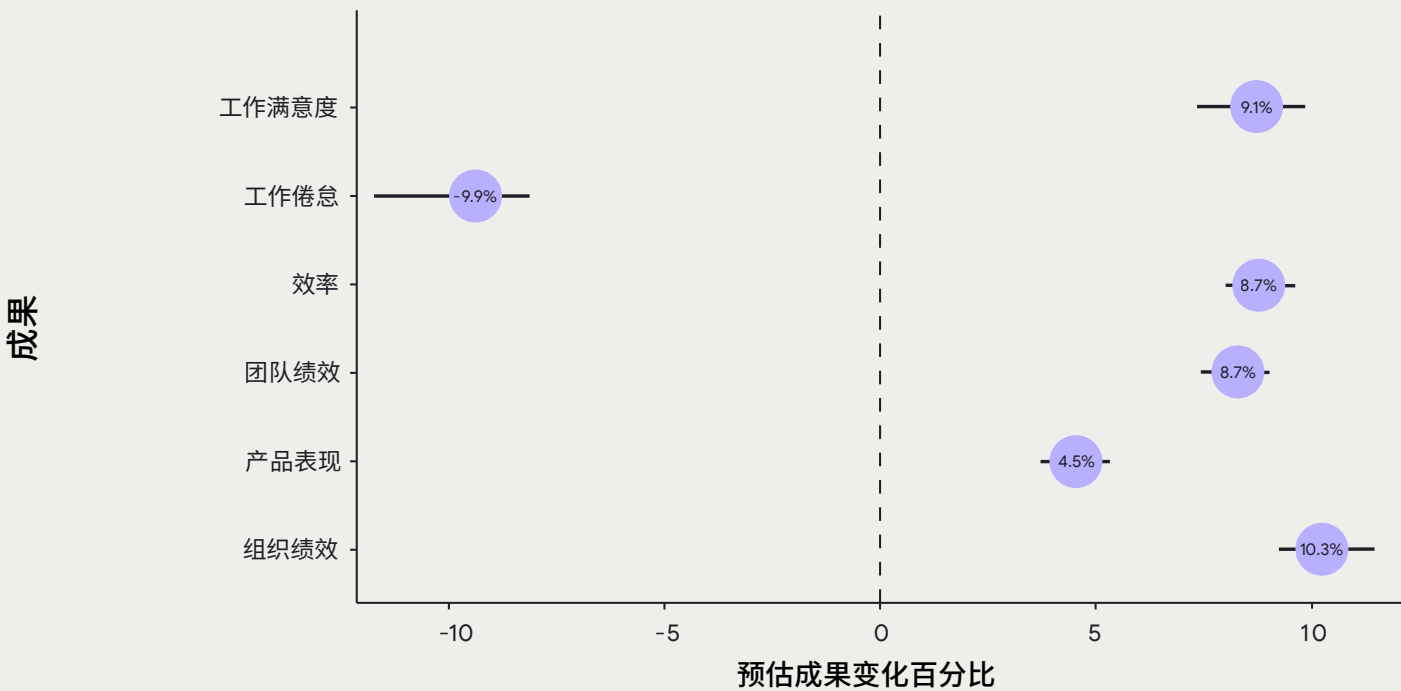
转型需要时间,也需要工具。领导层必须专门为改进工作分配资源。改进工作需要时间,也需要资金支持,在这方面,优秀的领导者发挥着关键作用。不应要求工程师在工作之余学习新知识和实施自动化,这应该是他们正常工作的一部分。

我们的调查帮助扭转了把 IT 视为成本的固有观念,让组织开始将 IT 视为推动业务成功的一项投资。在 2020 年,我们撰写了 DevOps 的投资回报率白皮书²。您可以利用其中的一些计算方法,来阐明投资 IT 改进所创造的潜在价值。

这项投资带来的回报不仅仅是金钱。根据我们 2015 年的调查,“组织对 DevOps 的投资与以下因素密切相关:组织文化;开发、运营

和信息安全团队实现共赢的能力;较低的工作倦怠水平;更有效的领导力;以及持续交付和精益管理实践的有效实施。”³我们建议专门为改进工作划拨一定的资源。

如果变革型领导力提高 25%...



点 = 估算值
误差线 = 89% 的不确定性区间
图 18: 变革型领导力对各种成果的影响。

坚持以用户为中心

今年的研究表明,拥有优秀的领导者并围绕用户需求构建软件的组织,往往能够开发出更出色的产品。这两项因素叠加在一起可以产生重要影响。当软件开发以用户为中心时,领导者可清晰地阐明愿景。

最终目标是让用户爱上我们创造的产品。正如我们在[开发者体验](#)一章中所讨论的,以用户为中心使产品功能有了存在的意义。开发者可以充满信心地构建这些功能,因为他们深知自己的工作能帮助提升用户体验。

我们发现,那些深切渴望了解并满足用户需求,且拥有收集、跟踪和回应用户反馈的机制的团队,其组织绩效更高。实际上,只要以用户为中心,即使软件不具备很快的部署速度和很高的稳定性,组织也能获得成功。在 2023 年,我们发现以用户为中心的团队比其他团队的组织绩效高 40%。⁴早在 2016 年,我们就发现以用户为中心的团队拥有更高的组织绩效。

今年的研究成果与之前的结果相符:以用户为中心的团队能够开发出更出色的产品。

有了这样的团队,不仅产品有了改进,员工对工作也更为满意,发生工作倦怠的风险也降低了。

快速稳定的软件交付让组织有更多机会经常开展实验和学习。理想情况下,这些实验和迭代都应⁵以用户反馈为基础。快速且稳定的软件交付让您能够进行实验、更加了解用户需求并在其需求未得到满足时快速反应。

如果在交付方面能兼顾速度和稳定性,您可以更轻松地根据市场变化和竞争进行调整。

请务必记得,您的内部开发者也是用户。组织通过内部开发者平台 (IDP) 来向开发者传递价值,然后开发者继续向外部用户或其他内部用户传递价值。

我们的研究表明,应该把 IDP 当成一款产品来开发。成功的 IDP 应该以用户为中心,并且能支持开发者独立工作。以这种方式部署 IDP 可以提高个人效率、团队效率,以及组织绩效。

成为以数据为导向的组织

直观呈现取得的进展是一项重要的能力。在过去 10 年里,我们已经证明了为何要成为一个以数据为导向的公司。DORA 的四个关键指标⁵已成为衡量软件交付绩效的全球标准,我们还确认了 30 多种可用于推动组织实现改进的能力和流程⁶。

指标的价值在于它们能够告诉您是否取得了进步。这四个关键的指标应该用在应用和服务级别,但不适合用在组织或业务线级别。您可以用这些指标来呈现您在持续改进方面的投入,但不能用它们来比较团队的绩效或是个人绩效。

这些指标也不应该用作评估您的应用或服务团队成熟度的模型。低、中、高或卓越绩效者的划分固然值得关注,但建议您谨慎看待这些标签,因为它们在您的转型过程中并没有太大价值。

由于我们的调查在不断发展演变,建议您跳出这四个关键指标的框架进行思考。现在大家普遍认为,用户反馈指标与这四个关键指标一样重要。我们认为,这是因为大多数团队已经制定了可行的解决方案来提高速度

和稳定性。因此,速度和稳定性方面的绩效普遍提高,由此带来的优势也就大大减弱了。

为了从全局角度看待转型,我们建议创建信息中心和可视化图表,将技术指标(例如四个关键指标和可靠性指标)和业务指标结合起来。这有助于弥合自上而下的转型工作和自下而上的转型工作之间的差距,还可帮助将您的战略方向、OKR 和员工目标与 IT 投资联系起来,从而更好地量化投资回报率。

我们认为指标是衡量卓越的一个必备条件,也有助于决策的制定。掌握的定量和定性研究指标越多,就越有机会做出合理、更明智的决策。人们对数据的价值和意义通常会有不同的看法。然而,基于数据本身做出决策往往比依赖个人看法或直觉更可取。



全力上云还是留在数据中心

自 2018 年以来,我们一直在研究 NIST 定义的 5 大云计算特征⁷(按需自助服务、广泛的网络接入、资源池化、快速弹性伸缩和衡量服务用量,也称为灵活基础设施)和组织绩效之间的关系。我们发现,与那些不太成功的团队相比,成功的团队更愿意利用灵活基础设施。

在去年的研究中,针对这个主题,我们得出了一个最令人惊讶的结论:如果使用云服务而不利用其五个特征,可能会有不利影响,组织绩效也可能会下降。

如果组织不愿意彻底改造其应用或服务,那么他们可能更适合留在数据中心。当然,要彻底转型,单靠采用工具或技术是不够的,往往需要在应用的设计、构建、部署和运行等方面全面采用新的范式。进行大规模变革时,从少量服务着手会更容易。我们建议采用迭代方法,帮助团队和组织在前进的过程中逐步学习和改进。

摘要

过去 10 年的经验不断告诉我们，转型是成功的必要条件。许多组织误以为转型是一个终点，实则它是一个持续改进的过程。⁸我们的研究明确印证了这一点：做不到持续改进的公司实际上正在逐渐掉队。相反，那些秉持持续改进理念的公司往往能够取得最大的成功。

在这个过程中，我们要提醒您的是，您可能在前进的道路上遇到一些困难，并可能感觉不适应。我们的研究表明，在 DevOps、⁹SRE，¹⁰以及今年新增的平台工程的实施过程中，组织最初会出现绩效下降，随后又会大幅提升（也称为“J 曲线”）。这是常见现象，而且只要您持续改进，情况就会好转，最终您将比刚开始时进步很多。

一想到要无休止地改进，您可能会感到气馁。追求完美的转型计划或设计很容易让人陷入困局。成功的关键在于撸起袖子，立即开始行动。组织和团队的目标应该是每天都比前一天进步一些。我们过去十年以及未来研究的目标是，帮助您提升改进的能力。

-
1. 《Dimensions of transformational leadership: Conceptual and empirical extensions》(变革型领导力的维度：概念和实证的拓展) - Rafferty, A. E. 和 Griffin, M. A.
 2. 《The ROI of DevOps Transformation》(DevOps 转型的投资回报率) - <https://dora.dev/research/2020/>
 3. 《2015 State of DevOps Report》(2015 年 DevOps 现状报告) <https://dora.dev/research/2015/2015-state-of-devops-report.pdf#page=25>
 4. 加速：2023 年 DevOps 现状报告 - <https://dora.dev/research/2023/dora-report/2023-dora-accelerate-state-of-devops-report.pdf#page=17>
 5. DORA 的四个关键指标 <https://dora.dev/guides/dora-metrics-four-keys/>
 6. DORA 的能力和流程 <https://dora.dev/capabilities/>
 7. 由 NIST 定义的 5 大云计算特征 <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
 8. 持续改进的旅程 <https://cloud.google.com/transform/moving-shields-into-position-organizing-security-for-digital-transformation>
 9. 《2018 Accelerate State of DevOps Report》(加速：2018 年 DevOps 现状报告) <https://dora.dev/research/2018/dora-report/>
 10. 2022 年 DevOps 现状报告 <https://dora.dev/research/2022/dora-report/>

第 10 份 DORA 报告



发展历史

DevOps 运动诞生于 2009 年两个主题相关的独立事件。当年 6 月, John Allspaw 和 Paul Hammond 在 Velocity 大会上发表了一个演讲, 标题为: “10 deploys per day: Dev & ops cooperation at Flickr” (每天 10 次部署: Flickr 的开发和运营合作)。¹ 几个月后, Patrick Debois 带领一支志愿者组织者团队, 在比利时根特举办了首届 DevOpsDays 活动。²

DevOps 社区渴望了解更多关于其发展状况的信息。当时就职于 Puppet Labs 的 Alana Brown 于 2011 年开展了一次问卷调查, 旨在深入了解 DevOps。这次调查帮助确认了一点: “以‘DevOps’方式工作正在成为 IT 界的新业务运作模式”。

随着这一运动继续蔓延到新的行业和组织, Alana 在此调查的基础上, 与 IT Revolution Press 合作, 在 2012 年组织了另一项问卷调查, 并在《2013 年 DevOps 现状报告》中发表了他们的调研结果。³

次年, Nicole Forsgren 博士加入了该调研团队, 让该项目更加科学严谨。《2014 年 DevOps 现状报告》⁴将软件交付绩效和组织绩效联系起来, 并发现“与 IT 团队绩效低的组织相比, 拥有高绩效 IT 团队的上市公司在三年内的市值增长高出 50%。”

到 2016 年, 年度报告的模式已基本确立, Forsgren、Jez Humble 和 Gene Kim 创立了 DevOps 研究和评估组织 (DORA)。同年, DevOps 现状报告增加了计算项目, 用于衡量团队为采用 DevOps 实践付出的投资。在 2020 年发布的《ROI of DevOps Transformation》(DevOps 转型的投资回报率)⁵白皮书中, 我们进一步扩大了这方面的工作。

Forsgren、Humble 和 Kim 合著的《Accelerate: The science behind devops: Building and scaling high performing technology organizations》(加速: DevOps 背后的科学: 建立和壮大高绩效技术组织)⁶于 2017 年由 IT Revolution Press 出版。这本书总结了该研究计划早年的成果, 并重点研究了推动改进的能力。

DORA 公司在 2018 年发布了一份独立报告: 《The Accelerate State of DevOps: Strategies for a New Economy》(加速: DevOps 现状: 新经济策略)。⁷除了这些报告, Puppet 的团队同年开始发布自己独立于 DORA 的系列报告。⁸

在 2018 年末, DORA 被 Google Cloud 收购,⁹这项中立的科学研究得以延续。今年, 第十份 DORA 报告出炉,¹⁰我们很高兴与您分享我们的发现。感谢您的阅读!

DORA 的重要洞见

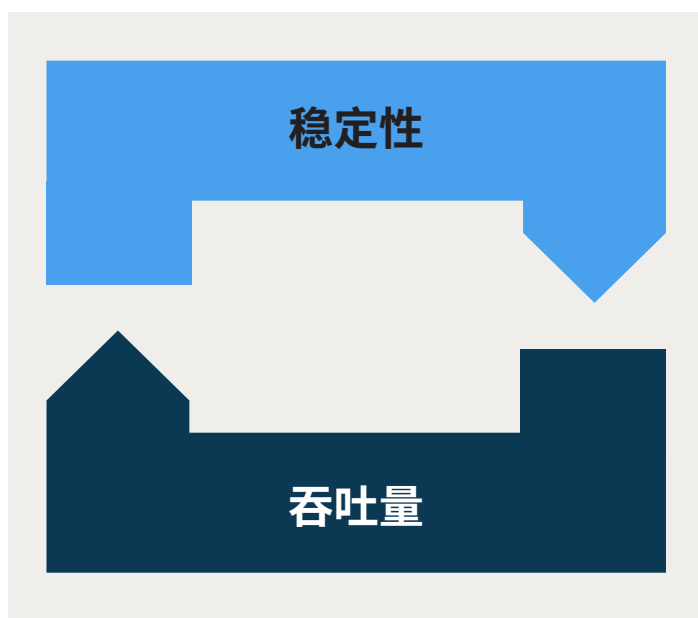
团队无需为了稳定性而牺牲速度

技术驱动型团队需要找到衡量绩效的方法，以便评估自己当前的进展、排定改进事项的优先顺序，并验证计划进展。DORA 找到并验证了四个有效的软件交付指标。这四个关键指标能有效衡量软件交付流程的成果。这些衡量软件交付绩效的指标已成为业界标准。

我们的研究已表明，更改的吞吐量和稳定性通常同升同降，各行业都有团队在这两个方面取得了高绩效。

团队可以通过多种方法衡量这四个指标，包括：

- 通过团队会议期间的讨论与反思
- 完成 DORA 快速评估 (<https://dora.dev/quickcheck>)
- 软件工程智能 (SEI) 类别的商用及开源¹¹工具
- 为团队所用特定工具量身打造的集成

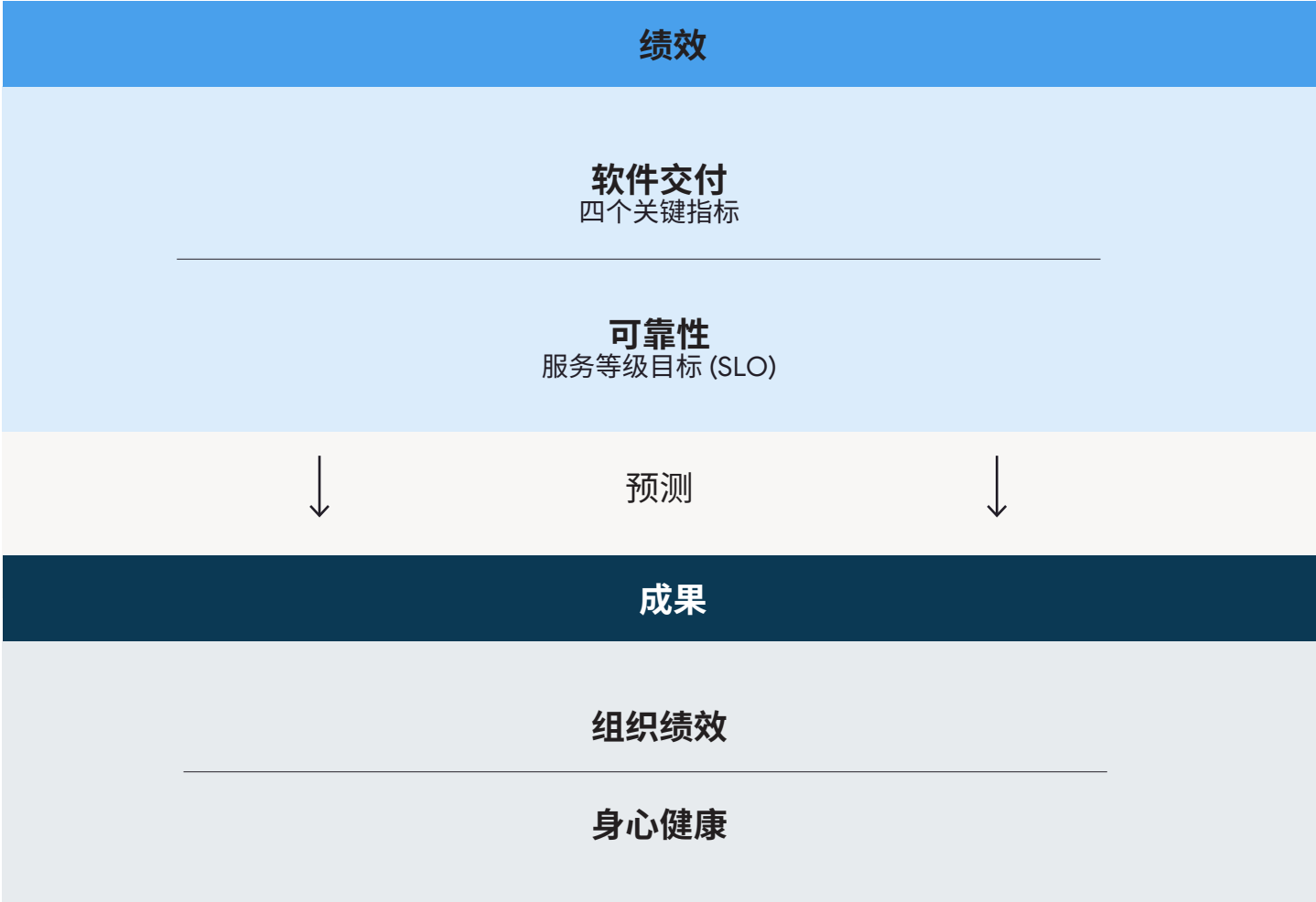


软件交付和运营绩效对提升组织绩效有帮助

DORA 通过四个关键指标来衡量软件交付绩效。DORA 在 2018 年开始研究运营绩效。这项指标可以衡量组织在实现软件产品或服务开发承诺方面的能力。

软件交付和运营绩效齐头并进,方能提升组织绩效和员工身心健康水平,从而取得最佳成果。

身处技术驱动型团队的从业人员深知,在交付过程中减少阻碍并满足用户对应用可靠性的期待至关重要。





文化是成功的基石

组织文化是预测绩效最准确的依据之一。我们不断见证高信任度文化在促进学习与协作氛围上的强大力量。例如，我们在 2022 年的研究表明，文化是预测组织应用开发安全实践的一项最主要的指标。¹²

文化影响着研究的各个方面，它包罗万象，且不断变化。多年来，我们借鉴了诸如 Westrum 的《组织文化分类学》¹³等研究，采用了多种不同的衡量方法。我们的身心健康衡量指标包括工作倦怠、工作效率和工作满意度。

不断改进

我们建议团队设置一个目标，并不断改进。要推动改进，既要有持续改进的理念，也要有相应的行动。为此，您需要评估自己的现状、确定改进工作的优先级，并建立可帮助您衡量进展的反馈机制。

采用实验性改进方法难免会有失败的时候，但无论结果如何，团队都能从经验中获得有意义的行动启示。

下一个十年

在过去十年里，我们相互学习，收获颇丰。感谢您参与我们的年度调查、加入 [DORA 实践社区](#)，¹⁴并将 DORA 应用于您的实际工作中。

随着技术领域的不断发展，DORA 将继续研究能帮助技术驱动型团队和组织获得成功的能力和实践经验。我们会继续优先关注技术的人文层面，并致力于发布中立的研究成果，助力您探索自己的改进之路。

我们过去发布的许多研究洞见仍具有价值，可以助力您采用新兴技术和实践；我们也期待与您一同探索新的发现！

我们会坚守 DevOps 运动的根本原则：文化、协作、自动化、学习，以及运用技术实现业务目标。我们的社区和研究受益于多元化角色的视角，包括那些可能不认为自己属于“DevOps”的人士。“DevOps”一词将不再是关注的焦点。

今年报告的重点是人工智能 (AI) 的应用及影响。读到这里，您应该已经知道 AI 的采用度正在上升，且该领域仍有广阔的实验空间。未来我们将继续深入研究此领域及其他新兴技术与实践。请结合我们以往的研究和最新发现，推动 AI 的采用并助力改善每位团队成员的体验。

1. 幻灯片 - <https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>, 录制内容 - <https://www.youtube.com/watch?v=LdOe18KhtT4>
2. <https://legacy.devopsdays.org/events/2009-ghent/>
3. <https://www.puppet.com/resources/history-of-devops-reports#2013>
4. 《2014 State of DevOps Report》(2014 年 DevOps 现状报告) - <https://dora.dev/research/2014/>
5. 《The ROI of DevOps Transformation》(DevOps 转型的投资回报率) - <https://dora.dev/research/2020/>
6. Forsgren, Nicole, Jez Humble 和 Gene Kim。2018 年。《Accelerate: The Science Behind DevOps: Building and Scaling High Performing Technology Organizations》(加速: DevOps 背后的科学: 建立和壮大高绩效技术组织)。IT Revolution Press。
7. 《Accelerate State of DevOps: Strategies for a New Economy》(加速: DevOps 现状报告: 新经济策略) - <https://dora.dev/research/2018/dora-report/>
8. <https://www.puppet.com/resources/history-of-devops-reports#2018>
9. <https://dora.dev/news/dora-joins-google-cloud>
10. 我们将 Forsgren 博士加入研究计划的那一年，也就是 2014 年，视为第一份 DORA 报告的发布时间，虽然 DORA 是在几年后才成立的。2020 年没有发布报告，因此 2024 年的报告是我们的第十份报告。
11. <https://dora.dev/resources/#source-available-tools>
12. 加速: 2022 年 DevOps 现状报告 - <https://dora.dev/research/2022/dora-report/>
13. Ron Westrum, “A typology of organisation culture” (《组织文化分类学》), BMJ Quality & Safety, 第 13 卷, 第 2 期 (2004), doi:10.1136/qshc.2003.009522
14. <https://dora.community>

最后总结

十年来, DORA 已成为行业研究、洞见和信息的可靠来源。随着行业不断采用诸如平台工程设计和人工智能等新的实践和技术, DORA 将始终伴您左右, 探索能助力团队改进的工作方式。感谢您与 DORA 一路同行。

复现我们的研究

今年报告的研究领域和调研结果较为复杂, 有些地方不太明确, 甚至有矛盾之处。因此, 建议您复现我们的研究。专注于单个团队或组织大大有助于加深理解。

在您的组织内开展实验

DORA 的调研结果能为您接下来的实验提供假设基础。详细了解您团队的运作模式, 并根据 DORA 研究计划的调研结果, 找出可改进的领域。

在您的组织内部开展问卷调查

借鉴本报告和今年的调查问题, ¹设计您自己的内部调查问卷。您的调查问卷可以围绕您的受众群体设计更有针对性的问题。²阅读[研究方法](#)一章, 详细了解我们的研究是如何进行的。请务必将您的调研结果付诸实践。

分享您的学习成果

从实验中得出结论后, 请在组织内分享。分享方式多种多样, 可以是面向广泛受众群体发布正式报告、通过非正式的实践社群分享, 或者在同事之间口头交流。不妨多尝试不同的方式, 了解哪种方式最契合您的环境和文化。这也是一个实验过程。



如何利用此研究？

欢迎加入 DORA 社区 (<https://dora.community>), 与其他成员分享您的经验并互相学习, 从他们的持续改进历程中汲取灵感。



1. 2024 年调查问卷 <https://dora.dev/research/2024/questions/>

2. 软件公司开展内部 DORA 问卷调查的经验 - <https://www.infoq.com/news/2024/08/dora-surveys-software-company/>

致谢

今年,我们发布了第十份 DORA 报告,这具有特殊的里程碑意义。我们衷心感谢所有研究人员、专家、从业人员、领导者和转型推动者付出的努力,感谢他们成就了这项工作并与我们一同成长。

自 Puppet Labs 和 IT Revolution Press 发布第一份 DevOps 现状报告以来,我们已取得长足进步。衷心感谢 DORA 创始人们为此铺平了道路。回首往昔,我们深感变化之巨,收获之多。

对参与今年报告发布的所有人员,我们深表谢意。引导和影响行业实践是一份巨大的责任,而各位的贡献功不可没。

从最初的探索到如今激动人心的 AI 时代,感谢所有参与这段旅程的人们。各位的支持和见解至关重要。让我们展望下一个十年,携手共同探索!

DORA 报告团队

James Brookbank

Kim Castillo

Derek DeBellis

Benjamin Good

Nathen Harvey

Michelle Irvine

Amanda Lewis

Eric Maxwell

Steve McGhee

Allison Park

Dave Stanke

Kevin Storer

Daniella Villalba

编辑

Seth Rosenblatt

本地化志愿者

Andrew Anolasco

Mauricio Meléndez

Marie-Blanche Panthou

Miguel Reyes

Yoshi Yamaguchi

Jinhong Yu

DORA 指南

Lisa Crispin

Steve Fenton

Denali Lumma

Betsalel (Saul) Williamson

现场顾问/ 专家

John Allspaw

Birgitta Böckeler

Sander Bogdan

Michele Chubirka

Thomas De Meo

Jessica DeVita

Rob Edwards

Nicole Forsgren 博士

Gene Kim 和
IT Revolution

Laura Maguire 博士

James Pashutinski

Ryan J. Salva

Majed Samad

Harini Sampath

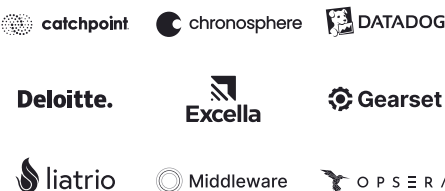
Robin Savinar

Sean Sedlock

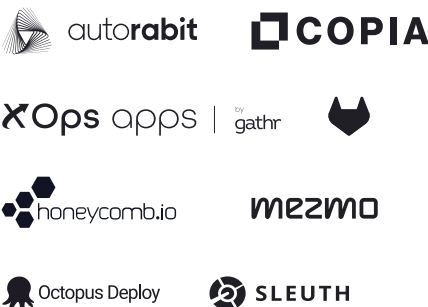
Dustin Smith

Finn Toner

Gold 赞助商



Silver 赞助商



作者



Derek DeBellis

Derek 是 Google 用户体验量化研究员和 DORA 首席调查员。他主要负责调查问卷研究、日志分析, 以及找出方法来衡量可证明产品或功能为用户创造了重要价值的概念。Derek 在之前发表的文章中探讨了人工与 AI 交互、新冠确诊对戒烟的影响、NLP 错误设计、用户体验在隐私保护议题中所扮演的角色, 团队文化以及 AI 与员工身心健康和工作效率之间的关系。目前他在进行一项业余研究, 主题是探索信仰和权力传播方式的模拟。



Kevin M. Storer

Kevin M. Storer 博士是 Google 开发者体验研究员, 负责领导 DORA 团队的定性调查工作。凭借在软件工程领域的专业经验, 以及在社会科学和人文领域的跨学科研究生教育背景, Kevin 自 2015 年起, 一直领导着围绕人展开的软件开发者研究。其研究的问题背景、参与者特征和研究方法多种多样。Kevin 的研究已在专注于人工智能、信息检索、嵌入式系统、编程语言、普适计算和交互设计等主题的顶尖科学期刊上发表。



Amanda Lewis

Amanda Lewis 是 [DORA.community](#) 开发主管, 以及 Google Cloud 的开发者关系工程师。她的职责是在开发者、运营人员、产品经理、项目经理和领导团队之间建立联系。她曾任职于电子商务平台、内容管理系统、观测工具开发团队, 以及开发者支持团队, 致力通过联系和对话来满足客户的需求, 实现更出彩的业务成果。她将自己的工作经验和同理心带入工作, 帮助团队了解及落实软件交付及人工智能实践。



Benjamin Good

Ben Good 是 Google 的云端解决方案架构师, 他热衷于通过云技术和自动化来改进软件交付实践。作为一名解决方案架构师, 他通过提供架构指导、发布技术指南和贡献开源资源, 帮助 Google Cloud 客户解决问题。在加入 Google 之前, Ben 曾在丹佛/博尔德地区的多家公司负责云运营, 并在此过程中推行 DevOps 实践。



Daniella Villalba

Daniella Villalba 是 Google 用户体验研究员, 专门通过调查问卷研究, 来了解提升开发者工作效率和满意度的因素。在加入 Google 之前, Daniella 研究过冥想训练的益处、影响大学生经历的心理社会因素。她拥有佛罗里达国际大学的实验心理学博士学位。



Eric Maxwell

Eric Maxwell 领导 Google 的 DevOps 转型实践, 他为全球的顶级公司提供改进建议, 帮助他们更快地为用户创造价值。在职业生涯的前半段, Eric 是一名一线工程师, 他建立了自动化程序来处理所有工作, 并设身处地地为其他从业者着想。Eric 参与制定了 Google 的云应用现代化改造计划 (CAMP), 是 DORA 核心团队的成员。在加入 Google 之前, Eric 任职于 Chef Software 并与同事度过了一段愉快的合作时光。



Kim Castillo

Kim Castillo 是 Google 用户体验计划经理, 自 2022 年起负责领导 DORA 计划幕后的跨职能工作, 包括监督研究运作情况, 以及 DORA 报告的发布工作。此外, Kim 还负责 Gemini in Google Cloud 的用户体验研究。加入 Google 前, Kim 曾在科技行业从事技术计划管理和敏捷辅导相关工作。Kim 主要通过社会心理研究, 探索其祖国菲律宾的非法处决、都市贫民窟发展, 以及社区韧性问题。



Michelle Irvine

Michelle Irvine 是 Google 的技术文档工程师, 她的研究重点是文档和其他技术交流形式。在加入 Google 之前, 她从事教育出版工作, 并为物理模拟软件撰写技术文档。Michelle 拥有滑铁卢大学的物理学学士学位, 以及修辞和传播设计硕士学位。



Nathen Harvey

Nathen Harvey 负责领导 Google Cloud 的 DORA 团队。Nathen 向一些出色的组织、团队和开源社区学习, 并将经验分享给大家。他是多份 DORA 报告的合著者, 并且是 O'Reilly 于 2020 年发表的《97 Things Every Cloud Engineer Should Know》(云工程师应该知道的 97 件事) 的撰稿人兼编辑。

受众特征和企业特征

哪些人参与了问卷调查

十多年来, DORA 研究计划一直致力调研是哪些能力、实践和措施让技术驱动型组织得以实现高绩效。我们累计收到约 39,000 位专业人士的回复, 这些回复者就职于各行各业不同规模的组织。感谢大家与我们分享真知灼见! 今年, 来自全球各行各业的超过 3,000 名专业人士分享了他们的经验, 帮助我们更好地了解了有助于技术驱动型组织提高绩效的因素。

今年的受众特征和企业特征问题采用 Stack Overflow 组织的研究。

超过 90,000 位回复者参加了 2023 年 Stack Overflow 开发者调查。¹虽然该调查未能覆盖所有技术从业人员, 但它已尽可能接近对开发者群体的全面普查。

由于该调查参与人数众多, 我们可以借此找出我们的数据中存在偏差的回答, 并了解我们的研究结果可以在多大范围内进行泛化。此外, 该 Stack Overflow 开发者调查提出的受众特征和企业特征问题都经过精心设计, 值得参考。

简而言之, 我们的样本与 Stack Overflow 的样本之间没有显著差异。这意味着我们有充分的理由相信, 我们的样本能够代表整体人群。

行业

我们请调查问卷回复者指明了所在组织主营业务所属的行业，共提供了 12 个类别供回复者选择。回复者人数最多的行业分别是：技术 (35.69%)、金融服务 (15.66%) 和零售/消费/电子商务 (9.49%)。

行业	受访者所占百分比
技术	35.69%
金融服务	15.66%
零售/消费/电子商务	9.49%
其他	5.94%
工业和制造业	5.49%
医疗保健和制药	4.60%
媒体/娱乐	4.26%
政府	3.89%
教育	3.66%
能源	3.03%
保险	2.39%
公益	1%

员工数

我们请调查问卷回复者指明了所在组织的员工人数，并提供了 9 档范围供回复者选择。其中员工人数最多的范围依次是：10,000 人(含) 以上 (24.10%)、100 至 499 人 (18.50%) 和 1,000 至 9,999 人 (15.60%)。

单位规模	百分比
1 人	2.0%
2 至 9 人	3.2%
10 到 19 人	4.3%
20 至 99 人	14.5%
100 至 499 人	18.5%
500 至 999 人	11.2%
1,000 至 4,999 人	15.6%
5,000 至 9,999 人	6.7%
10,000 人(含) 以上	24.1%

残障人士

我们根据 Washington Group Short Set (华盛顿工作团关于残疾情况的简短问卷) 指南, 从六个维度来确定残障情况。2024 年是我们连续第五年在问卷中统计回复者的残障状况。报告有残障的回复者比例从 2022 年的 11% 下降到 2023 年的 6%, 并在 2024 年降至 4%。

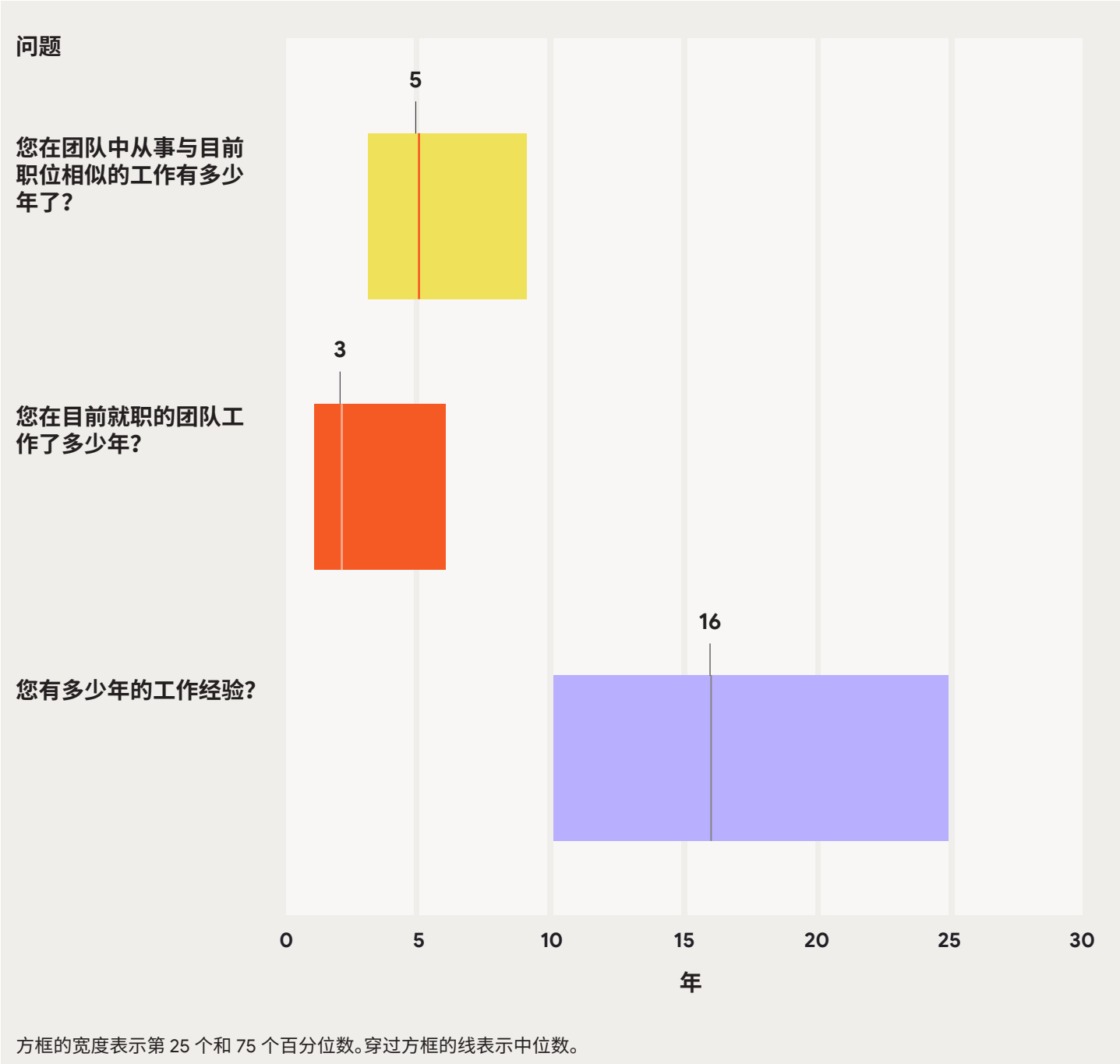
残障人士	占回复者的百分比
非残障人士	92%
至少有一项残障情况	4%
选择不回答	4%

性别

我们通过调查问卷中统计了回复者的性别。83% 的回复者表示自己为男性、12% 表示自己为女性、1% 选择自我说明, 还有 4% 拒绝回答。

性别	百分比
男性	83%
女性	12%
自述性别	1%
选择不回答	4%

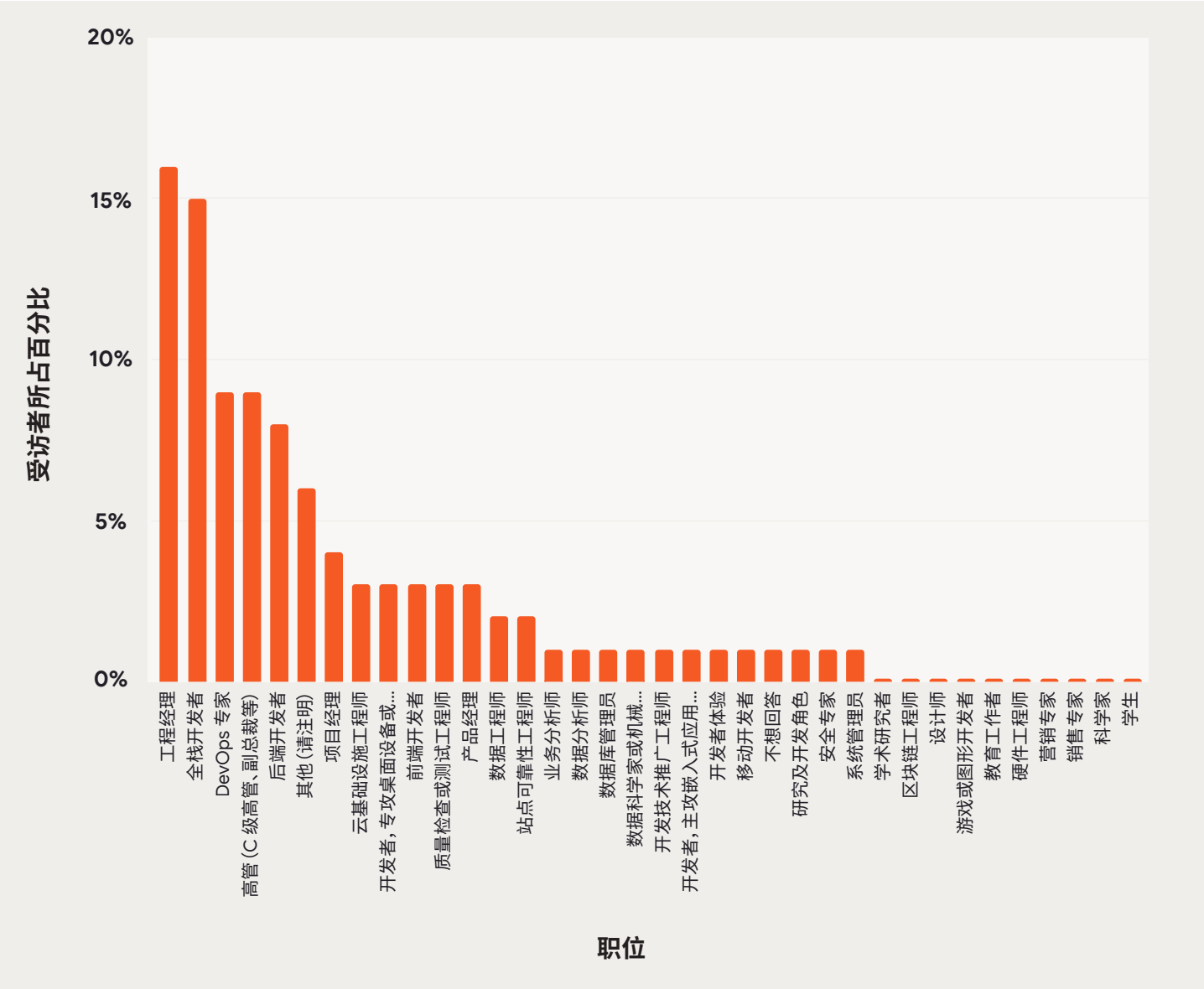
我们通过调查问卷统计了受访者在当前职位和团队中的工作年限。回复者的工作年限中位数是 16 年，担任当前职位的年限中位数为 5 年，在当前团队的年限中位数是 3 年。



角色

为了更有效地将受访者占比不高的职位纳入分析,我们将一些职位分在了一组。其他在我们的数据中占比较高的类别包括:

- 开发者, 占回复者的 29%。
- 管理者, 占回复者的 23%。
- 高级管理人员, 占回复者的 9% (比 2023 年增加 33%)。
- 分析类职位, 占回复者的 5%



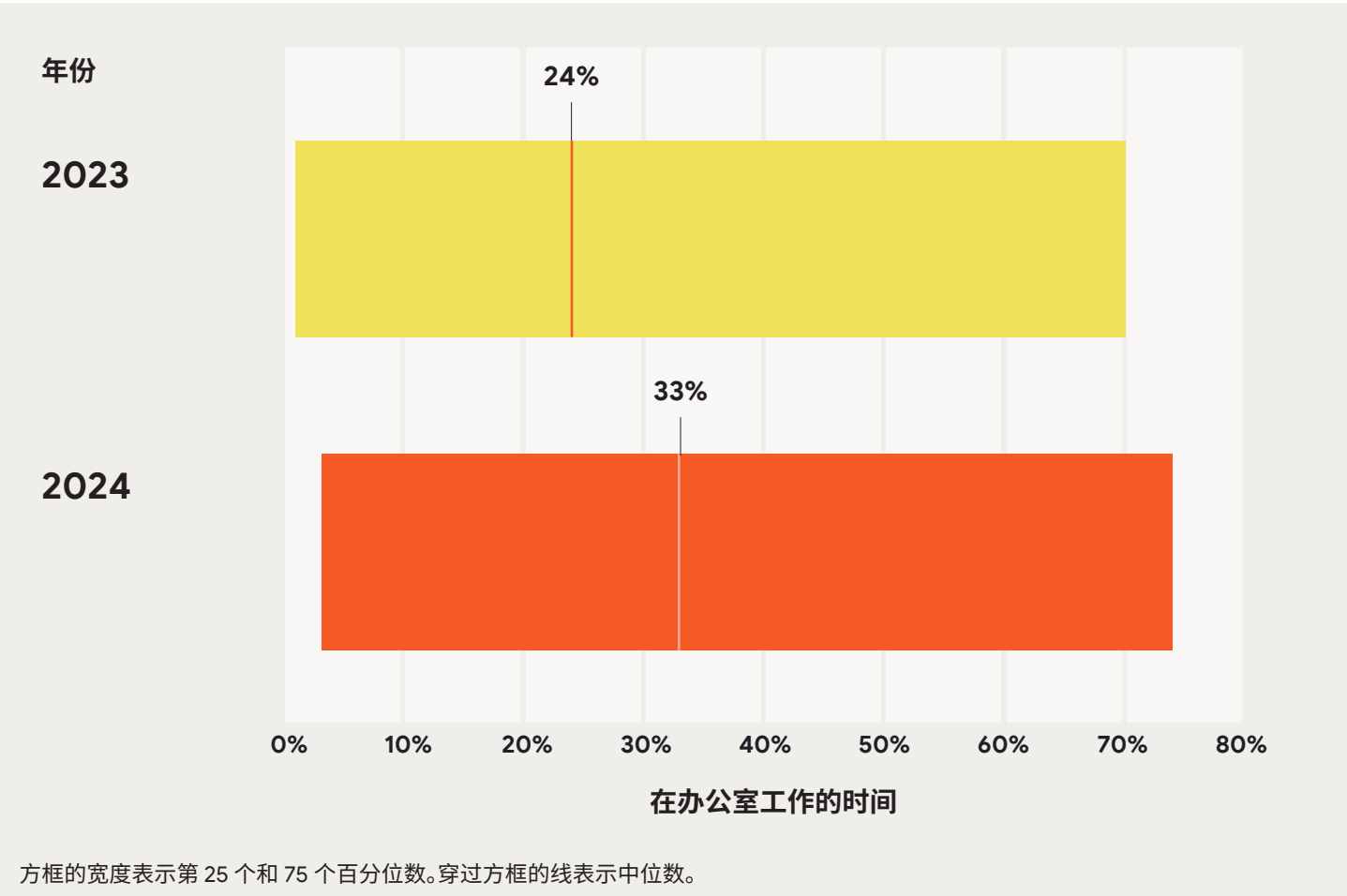
就业状况

我们通过调查问卷统计了回复者目前的雇佣形式。绝大多数 (90%) 回复者为组织的全职员工。

雇佣形式	百分比
全职合同工	6%
全职员工	90%
兼职合同工	1%
兼职员工	2%

工作地点

尽管又经历了一年的重返办公室 (RTO) 政策推动,但去年的模式基本没有发生很大变化,尤其是在两端的数据分布上。中位数值上升了 37.5%,这表明混合办公或至少定期到办公室办公的模式越来越普遍。



国家/地区

我们的回复者来自 104 个不同的国家/地区。看到全球各地的从业人员参与这份问卷调查，我们一如既往地感到荣幸之至。感谢大家鼎力相助！



国家/地区					
美国	意大利	新加坡	冰岛	卢森堡	危地马拉
英国	瑞士	阿尔巴尼亚	伊朗	尼加拉瓜	中国香港特别行政区
加拿大	阿根廷	格鲁吉亚	约旦	巴基斯坦	马耳他
德国	墨西哥	希腊	肯尼亚	秘鲁	毛里求斯
日本	葡萄牙	菲律宾	沙特阿拉伯	韩国	摩洛哥
印度	奥地利	匈牙利	斯洛伐克	斯里兰卡	尼泊尔
法国	罗马尼亚	塞尔维亚	斯洛文尼亚	突尼斯	巴拉圭
巴西	芬兰	阿富汗	泰国	安道尔	斯威士兰
西班牙	土耳其	阿尔及利亚	乌兹别克斯坦	巴巴多斯	阿拉伯叙利亚共和国
澳大利亚	保加利亚	埃及	安哥拉	伯利兹	中国台湾
荷兰	爱尔兰	印度尼西亚	亚美尼亚	贝宁	前南斯拉夫马其顿共和国
中国大陆	以色列	俄罗斯联邦	波斯尼亚和黑塞哥维那	玻利维亚	特立尼达和多巴哥
瑞典	比利时	乌克兰	多米尼加共和国	布基纳法索	乌拉圭
挪威	智利	越南	厄瓜多尔	科摩罗	委内瑞拉玻利瓦尔共和国
新西兰	哥伦比亚	孟加拉国	爱沙尼亚	科特迪瓦	
波兰	捷克	白俄罗斯	哈萨克斯坦	萨尔瓦多	
南非	马来西亚	哥斯达黎加	拉脱维亚	埃塞俄比亚	
丹麦	尼日利亚	克罗地亚	立陶宛	冈比亚	

种族和民族

我们通过调查问卷统计了回复者的种族和民族。其中人数最多的群体为白人 (32.4%) 和/或欧洲人 (22.7%)。

种族或民族	百分比值
白人	32.4
欧洲人	22.7
亚洲人	9.9
北美洲人	4.6
印度人	4.1
保密	4.1
西班牙裔或拉丁裔	3.5
南美洲人	3.2
东亚人	2.5
非洲人	1.8
南亚人	1.7
多民族	1.5
自述种族或民族：	1.5
东南亚人	1.4
黑人	1.3

种族或民族	百分比值
中东人	1.3
混血	0.4
中美洲人	0.4
我不清楚	0.4
北非人	0.4
加勒比人	0.2
中亚人	0.2
南亚人	1.7
民族宗教群体	0.2
太平洋岛裔	0.2
原住民 (如美洲原住民或澳大利亚原住民)	0.1

1. <https://survey.stackoverflow.co/2023/>
2. <https://www.washingtongroup-disability.com/question-sets/wg-short-set-on-functioning-wg-ss/>

研究方法

“研究方法”应如同一份菜谱,可帮助您复现我们的工作,并判断我们生成和分析数据的方式是否能得出有价值的信息。尽管篇幅有限,无法详尽阐述,但希望这部分能为上述工作提供一个良好的起点。

调查问卷设计

问题选择

在考虑是否将某个问题纳入调查问卷时，我们会考虑以下几个方面：

此问题…

- 是否能将我们的工作与之前的研究联系起来？
- 是否能反映行业希望实现的成果 (例如，高团队绩效)？
- 是否能涵盖行业正在考虑投入资源的领域 (例如，AI)？
- 是否能体现我们认为将有助于人们实现目标的能力 (例如，优质文档)？
- 是否有助于我们评估样本的代表性 (例如，职位或性别)？
- 是否有助于我们阻断偏差路径 (例如，编程语言或职位)？
- 是否能让绝大多数回复者以相当高的准确度作答？

为了决定是否将某个问题纳入我们的调查，我们会查阅文献、与 [DORA 社区](#) 交流、进行认知性访谈、开展并行定性调查、咨询主题专家，并举办团队研讨会。

调查问卷体验

我们非常注重提高调查问卷的易用性。我们进行了认知性访谈和易用性测试，以确保调查问卷遵循以下原则：

- 完成调查问卷所需的平均时间应较短
- 问卷应容易理解
- 在合理范围内，问题应比较容易回答。考虑到概念的技术性，这一点难度很大

数据收集

本地化

我们每年都会收到全球各地的回复。今年，为了让更多受访者可以轻松填写问卷，我们提供了英语、西班牙语、法语、葡萄牙语、日语和简体中文等语言的本地化版本。



收集调查问卷回答

我们通过多种渠道收集调查问卷回复,这些渠道分为两类:“人际网络”和“固定样本”。

“人际网络做法”是指使用我们可用的所有社交媒体渠道,让大众知道我们想请他们填写调查问卷。其中包括发布博文、发起电子邮件宣传活动,以及在社交媒体上发贴,我们还要求社区成员采取同样做法(也就是“即滚雪球抽样”)。

同时,我们采用“固定样本做法”来弥补人际网络渠道的不足之处。具体而言,我们在广大技术社区内招募传统上属于少数群体的受访者,并尝试从特定类型的行业和组织中收集到足够多的回复。

简单来说,固定样本做法能让我们控制招募对象,这是人际网络做法做不到的。此外,固定样本做法也让我们很容易就能确保接触到足够多的受访者,因为人际网络做法不一定能收集到我们进行分析所必要的回复。今年,我们通过人际网络渠道收集到了足够多的回复来开展分析,并使用固定样本做法来让参与者群体更加完整。

调查流程

今年,我们要调研的问题有很多,但时间有限。我们考虑的方案有…

- **设计一个非常长的调查问卷**
- **选择几个领域进行重点研究**
- **或为受访者随机分配不同的主题**

为了不放弃感兴趣的研究内容,我们选择将参与者随机分配至三个独立的流程。这三个流程之间有很多重合之处,但每个流程都深入探索了不同的领域。

以下是三个不同的研究主题:

- **AI**
- **工作单位**
- **平台工程设计**

调查问卷分析

衡量结果验证

我们希望通过这次调查问卷验证多个概念。描述这些概念的方式很多,其中一个观点是将这种评估概念的指标称为变量,这些变量是我们的研究所涉及的模型的构成要素。大致而言,我们可以从内部和外部来分析这些评估指标的效力。

为了了解评估指标的内部效力,我们会研究我们认为能代表某个概念的具体行为。例如,有人使用文档能解决问题即可表示文档是优质的。

我们大多数的变量是由多个指标构成,因为我们要研究的构想通常属于多层面。

为了了解变量的多层面性质,我们会进行测试,针对我们认为能代表该概念的具体行为,判断彼此共通程度。如果共通程度高(也就是有高度共同的方差),我们会假设这些行为背后有共同的概念,例如利益。

例如,以幸福为例,幸福是多方面的。我们预期人感到幸福时,会有特定的情绪、行为和

思考模式,同时假设幸福是产生特定情绪、想法和行为的基础。

因此,我们预期幸福感会与特定几种情绪、想法和行为同时出现。随后,我们会询问有关这些情绪、想法和行为的问题,并使用验证性因素分析,来测试这些元素是否确实同时出现。

今年,我们使用 lavaan¹ R 套件进行这项分析。Lavaan 会传回各种拟合统计资料,帮助我们了解某些构想是否确实反映了受访者提供的回复。

如果某个概念的指标无法共通,这表示我们尚未找到可靠的评估方法,因此可能需要修改或舍弃这个概念。

至于构想的外部效力,我们则是研究构想之间的拟合程度。我们会预期某个构想与其他构想之间有特定关系,有时甚至预期两者之间呈现负相关,例如快乐与悲伤。

如果快乐的衡量结果与悲伤呈现正相关,我们可能会怀疑所采取的评估方法或理论有误。

同样，我们可能预期两个构想之间呈现正相关，但并非高度正相关。例如，工作效率和工作满意度之间可能属于正相关，但我们不会认为这两者相等。如果关联性过高，我们反而认为这相当于评估同一个概念，也表示我们的评估方法不足以准确区分这两个概念之间的差异，或者现实中并没有我们假设的差异。

模型评估

我们以一套假设为指导原则，构建假设模型，试图简单地去理解一些世界运作的奥秘。我们会检验这些模型与我们收集的数据的拟合程度。在评估模型时，我们采用简约法。也就是从非常简单的模型²开始，然后不断提高复杂性，直到模型不再合理为止。

例如，我们预测软件交付绩效和运营绩效之间的相互作用将决定组织绩效。最初的简单模型并没有考虑这种相互作用：

$$\text{组织绩效} \sim \text{软件交付绩效} + \text{运营绩效}$$

考虑了这种相互作用的第二个模型如下：

$$\text{组织绩效} \sim \text{软件交付绩效} + \text{运营绩效} + \text{软件交付绩效} \times \text{运营绩效}$$

我们根据“Regression and other stories”（《回归分析与其他实例》）³，以及“Statistical Rethinking”（《统计反思》）⁴的建议，使用留一交叉验证 (LOOCV)⁵ 和渡边-赤池信息准则⁶，判断是否有必要额外提高复杂性。

用于因果推断的有向无环图

一个经过验证的模型为我们提供了开始思考因果关系时所需的信息。接下来，我们将探讨因果思考的挑战。

以下是我们尝试因果推理的一些原因：

我们认为问题本质上是一个因果性问题。您想知道做某事是否会导致某种结果。如果您认为不存在因果关系，也就不会投入资源去行动。

我们的分析结果取决于我们对世界的因果认知。回归分析得出的实际数值会因回归模型中包含的变量而变化。回归模型中包含哪些变量取决于我们对数据生成过程的理解，而这种理解本身就是一个因果性论断。所以，我们应对此给予清晰说明。

因果思考是求知欲的最终导向，也是我们花费大量时间的地方。我们常常思索世界万物如何彼此关联及其背后的原因。我们不必对生活中的万事万物都进行实验，也能运用因果思维对其进行思考。

因果思维对行动至关重要，而我们正是希望这份报告能帮助您做出行动决策。

我们能利用经验证的模型，来确定理解某种效应需要考虑哪些因素。简单来说，它使我们尝试将数据转换为 A/B 实验的形式，即构建两个完全相同的世界，然后只让它们出现一处不同。理由是，经过这样的操作，这两个世界后续出现的任何差异都应归因于这处不同。

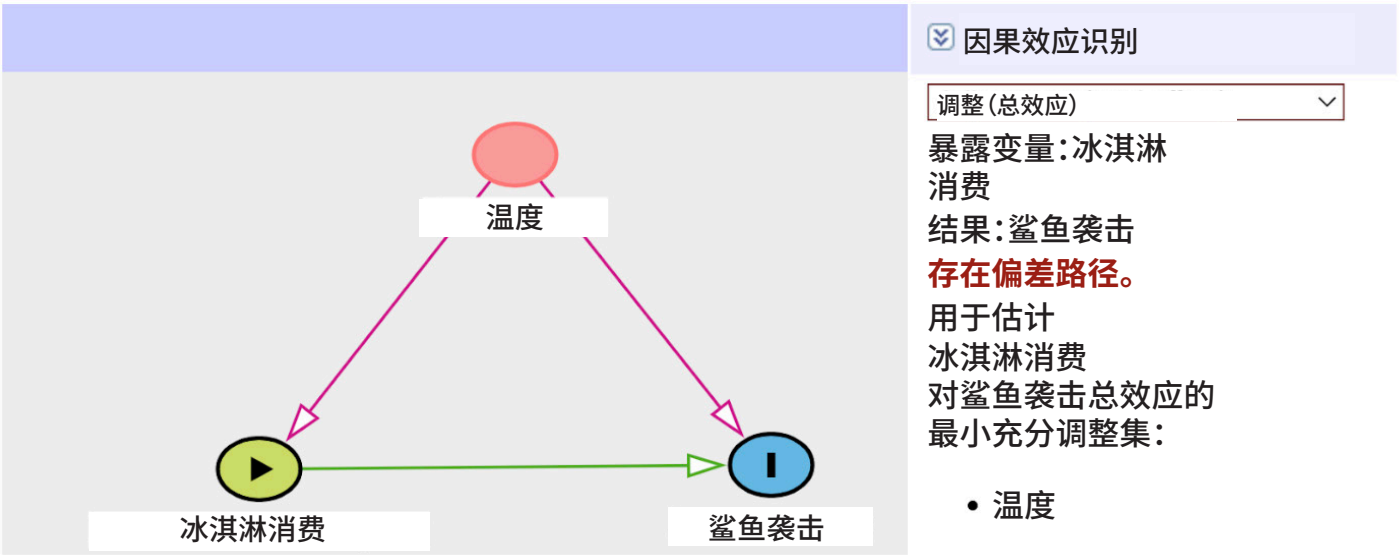
但是观测数据和调查数据是无法非此即彼地划分清楚的。参与者之间存在许多差异，这也就引入了干扰因素。我们的因果推断方法尝试将这些差异考虑进去，方式是模拟一个实验，保持所有其他因素不变，仅改变一个因素（例如，AI 的采用）。

我们来看一个经典案例，即冰淇淋“导致”鲨鱼袭击的现象。这个案例的症结在于，人们往往在炎热的天气吃冰淇淋，也往往在炎热的天气去海滩。人们倾向于吃冰淇淋并去海滩时的情况，与人们倾向于不吃冰淇淋且不去海滩时的情况是不同的。数据并未遵循实验的逻辑。我们遇到了一个干扰因素，那就是温度。

有向无环图 (DAG) 可以帮助您发现真实世界与实验环境的不同,并提供解决这些问题的途径,即尽可能地模仿实验,让除某个因素以外的所有因素固定不变。让我们看看在冰淇淋和鲨鱼袭击的例子中,DAG 如何指导我们。在这个例子中,我们想要量化冰淇淋消费对鲨鱼袭击的影响:

我绘制出模型,告诉工具我想要理解什么效应,然后工具会告诉我哪个因素会导致我对效应的估计产生偏差。在此案例中,该工具显示,如果不就温度做出调整,我就无法估计冰淇淋消费对鲨鱼袭击的影响。这种统计方法旨在使除冰淇淋消费外的所有因素都保持一致,然后观察鲨鱼袭击是否继续随冰淇淋消费而波动。

我们将在模型一章中概述这些模型。



图片来源于 <https://www.dagitty.net/dags.html>。

有向无环图告诉我们, 在分析特定效应时需要考虑哪些因素。

例如, 在分析 AI 采用对效率的影响时, 我们需要考虑哪些因素?

贝叶斯统计方法

我们使用贝叶斯统计方法来进行这项分析。贝叶斯统计方法具有诸多优点:

- 我们不需要再考虑显著或不显著的概念 (如果让 10 个人解释频率学派中的 P 值, 您会得到 10 个不同的答案)
- 我们想知道的是在给定数据的情况下, 假设成立的概率, 而不是在给定假设的情况下, 数据出现的概率。
- 我们喜欢将先验知识纳入模型, 或者至少明确说明我们对未知因素的了解程度⁷
- 我们不得不正视建模过程中的潜在假设
- 我们可以探索后验分布, 了解预测结果的大小强弱、不确定性, 以及模型如何以及在多大程度上理解了数据。最终, 它让我们很好地认识到可以从现有数据中了解什么, 了解不到什么
- 它是一个灵活的框架, 能够以统一的方式解决许多统计问题。

“模拟”指的是什么？

“模拟”并不是编造数据。我们使用贝叶斯统计方法算出后验概率，来获取“不同参数值的预期出现机率”。⁸“模拟”部分是指从这个后验概率中抽取超过 1,000 次，以找出对数据而言最可信的参数值

(平均值、Beta 权重值、Sigma 值、截距值等等)。

“将后验概率想成一个桶，其中装满 0.1、0.7、0.5、1 等等的参数值。在桶内，每个值的出现概率与其后验概率成正比，因此接近高峰的值会比接近尾部的值更常见。”⁹

这表示我们可以通过模拟的方法来找出数据的可能解读方式，以及了解不确定性高低。您可以

将每次模拟视为一个小型 AI 操作，这个 AI 只知道我们的数据和几项规则，且试图根据数据猜出值来填补空白 (参数)。只要模拟 4,000 次，即可针对指定参数得出 4,000 个小型 AI 的猜测值。

这些猜测值很有用处，您可以从中算出平均猜测值、89%¹⁰ 猜测值介于哪两个数值之间、多少猜测值在特定水准以上、猜测值之间的差异多大等等。此外，您甚至可以做一些有趣的事，例如合并多个模型的猜测值 (模拟)。

在这份报告中，您可以通过呈现许多线条或潜在值分布情况的图表，了解我们根据数据算出的最合理情形，以及不确定性高低。

与社区一起让研究结果发挥价值

我们的调研结果为技术驱动型团队和组织提供了一个颇具价值的视角，但我们需要通过对话和共同学习的方式让大家充分理解这些结果。与 DORA 社区互动能为我们带来多元化的见解，质疑我们的既有假设，并帮助我们发现解释和应用这些调研结果的新方法。

我们鼓励您加入 DORA 社区 (<https://dora.community>)，分享您的经验，向他人学习，并发现落实这些建议的多种方法。我们将共同探索如何最好地利用这些见解，并在您的组织内推动有意义的变革。

访谈

今年,我们在年度调查中加入了深入的半结构式访谈,以便对我们的定量调研结果进行三角验证,并提供背景信息和澄清不明之处。访谈提纲与调查问卷中的主题相呼应,每次访谈设计为约 75 分钟,通过 Google Meet 远程进行。

我们总共采访了 11 位参与者,他们的情况符合我们调查的标准。所有访谈都录制了视频和音频。访谈时长在 57 分钟至 85 分钟之间,所有参与者的访谈总时长为 14 小时 15 分钟。参与者的数据使用 P(N) 形式的标识符进行假名化处理,其中 N 对应于他们接受访谈的顺序。

我们使用自动化软件对所有访谈进行了转写。转写文稿使用我们的调查主题作为先验代码进行人工编码。本报告最终发布版本中出现的引文在收录前都经过了重新审阅和人工转写。在引述参与者的话语时,本报告的作者用方括号 ([]) 表示添加的文字,用省略号 (...) 表示省略的文字,并且仅在需要使引文表达清晰易懂时进行编辑。

结果中的推论跳跃

我们的目标是创建一个能够切实反映现实世界的模型,以便大家都能用它来帮助改善工作方式。我们知道有一些复杂因素被简化了。这正是模型的意义所在。Jorge Luis Borges 有一篇很短的故事,名为《On Exactitude in Science》(论科学的精确性),讲述了一个制作比例为 1:1 的帝国地图的故事。¹¹故事的荒谬之处在于,这样的地图完全没有意义(至少我是这么理解的)。我们进行简化的目的是希望大家有所帮助。

即便如此,我们仍然希望说明一些推论上的跳跃。

因果关系

根据 John Stuart Mill 的说法,要断定 X 导致 Y,您需要满足三个条件:¹²

- **相关性:** X 需要与 Y 共变吗?
- **时间先后性:** X 需要在 Y 之前发生吗?
- 是否已考虑偏差路径 (如上文 DAG 部分所述)?

我们有信心理解相关性,这通常是一种标准的统计程序。我们的调查只是记录了一个时间点的数据,因此时间先后只是理论上的推断,而非数据的一部分。

至于偏差路径,正如我们在上文讨论结构方程模型和有向无环图时所提到的,我们会尽

力去考虑偏差路径,但这是一种高度理论化的分析,与时间先后性不同。我们可以在数据中探索它的意义。

总而言之,我们没有进行纵向研究或适当的实验。尽管如此,我们认为因果思考是我们理解世界的方式,所以我们尽力运用新兴的因果推断技术来为您提供准确的估算数据。相关性并不意味着因果关系,但它确实体现了我们是如何思考因果关系的。

微观层面现象 -> 宏观层面现象

我们通常从个人层面的能力出发，观察这些能力如何与更高层面相联系。例如，我们将个人对 AI 的采用与某个应用或服务以及团队绩效联系起来。乍看之下，这并不那么容易理解。描述一个宏观层面的现象如何导致个体层面的现象通常更容易。通货膨胀（宏观）影响我是否购买鸡蛋（微观），这似乎比我不购买鸡蛋导致通货膨胀更合乎情理。

同样，组织的绩效（宏观）对个人的身心健康（微观）的影响也是如此。根据经验判断，组织对个人的影响可能大于个人对组织的影响。

那么，为什么我们还要费心强调个人行为会影响团队或组织绩效呢？我们做了一个我们认为并非完全符合逻辑的推论跳跃。具体而言，我们假设，如果数据达到一定规模，以下陈述倾向于成立：

当个人所在的组织或团队在做某事 (X) 时，该个人也做 X 的概率会更高。因此，个人在做某事反映出他的团队和组织也可能在做 X。当然，这里的干扰因素非常多，但规律最终会显现出来，我们也因此能够利用这一假设获得一些重要的能力。

为了更好地理解这一点，我们来看一个 DORA 之外的例子：想象两个平均身高不同的国家。在一个国家，人们的平均身高为 5’6”。另一个国家人们的平均身高为 6’2”。标准差是相同的。如果您从每个国家随机挑选一个人，您认为个子较高的人更有可能来自哪个国家？如果您像这样抽样成千上万次，那么从平均身高更高的国家挑出来的人的平均身高应该也更高。挑出来的这些人的平均身高将大致接近相应国家的平均身高。

$$p(\text{个人做 } X \mid \text{组织做 } X) > p(\text{个人做 } X \mid \text{组织不做 } X)。$$

虽然没有必要,但我们还是进行了一个快速模拟来验证这一点是正确的:

```
#R code

#set seed for reproducibility
set.seed(10)

#6'2 and 5'6
height_means = c(6 + 1/6, 5.5)

#constant standard deviation at 1/4 of
foot
std_dev = 0.25

#random draws
draws = 1000

#random draws from country A
country_a <- rnorm(draws, mean = height_
means[1], sd = std_dev)

#random draws from country B
country_b <- rnorm(draws, mean = height_
means[2], sd = std_dev)

#how of the draws represent the correct
difference
represented_difference = sum(country_a >
country_b) / 1000

#show results as percentage
represented_difference * 100
```

结果并不意外。1,000 次随机抽样中, 97.2% 的结果符合预期方向。当然, 非随机抽样、国家/地区之间的差异较小以及样本量较小都可能造成偏差。但核心观点依然成立: 宏观层面的差别往往会在微观层面体现出来。

1. Rosseel Y (2012).《lavaan: An R Package for Structural Equation Modeling》(lavaan: 结构方程模型专用的 R 套件)。Journal of Statistical Software, 48(2), 1–36. <https://doi.org/10.18637/jss.v048.i02>
2. 这还涉及了对潜在干扰因素的检验。
3. Gelman, Andrew、Jennifer Hill 和 Aki Vehtari。2021 年。《Regression and Other Stories》(回归分析与其他实例)。N.p.: Cambridge University Press。
4. McElreath, Richard。2016 年。《Statistical Rethinking: A Bayesian Course with Examples in R and Stan》(统计反思: R 与 Stan 中的贝叶斯方法教学与范例)。N.p.: CRC Press/Taylor & Francis Group。
5. Gelman, Andrew、Jennifer Hill 和 Aki Vehtari。2021 年。《Regression and Other Stories》(回归分析与其他实例)。N.p.: Cambridge University Press
6. McElreath, Richard。2016 年。《Statistical Rethinking: A Bayesian Course with Examples in R and Stan》(统计反思: R 与 Stan 中的贝叶斯方法教学与范例)。N.p.: CRC Press/Taylor & Francis Group。
7. 我们倾向于采用弱先验(持怀疑态度、中立且信息量少), 并且我们会检查结果是否未受先验条件的影响。
8. McElreath, Richard。《Statistical rethinking: A Bayesian course with examples in R and Stan》(统计反思: R 与 Stan 中的贝叶斯方法教学与范例)。Chapman and Hall/CRC, 2018 年, 第 50
9. McElreath, Richard。《Statistical rethinking: A Bayesian course with examples in R and Stan》(统计反思: R 与 Stan 中的贝叶斯方法教学与范例)。Chapman and Hall/CRC, 2018 年, 第 52 页
10. 我们选择 89% 可信区间是依据 McElreath 在《统计反思》第 56 页中阐述的观点。“Why these values? No reason... And these values avoid conventional 95%, since conventional 95% intervals encourage many readers to conduct unconscious hypothesis tests.”(为什么用这些值? 没有什么原因... 我们没有使用传统的 95% 置信区间内的值, 因为传统的 95% 置信区间容易使读者在无意识中进行假设检验。)我们提供的区间只是为了展示一个与模型和数据相符的合理“参数值范围”。
11. Borges, J. L. (1999)。《Collected fictions》(小说集)。Penguin。
12. Duckworth, Angela Lee、Eli Tsukayama 和 Henry May。《Establishing causality using longitudinal hierarchical linear modeling: An illustration predicting achievement from self-control》(使用纵向分层线性模型来建立因果关系: 一个根据自控力预测成就的例子)。Social psychological and personality science, 第 1 卷, 第 4 期 (2010): 311-317。

模型

过去,我们会构建一个大型模型,并使用多种结构方程模型技术(包括偏最小二乘法、基于协方差的方法、贝叶斯方法)进行验证。而在 2023 年的报告中,我们转而采用更小的模型,以便了解特定过程。

例如,我们建立了一个细致的模型来解析优质文档的作用。创建专门用于理解特定效应的小型模型¹具有重要优势:

- 能够更轻松地识别模型拟合度差的方面
- 模型中每个新增要素都会产生影响。模型越大,理解各变量之间如何相互作用的难度也越大
- 有助于防止您在分析时将可能产生虚假关系的变量作为条件²

我们是如何使用这些模型的？

我们要研究的问题有很多，但许多关键问题都可以归纳为以下形式：

如果我们实施 X, Y 会发生什么情况？

X 通常指代一种实践，如创建优质文档、采用 AI, 或投资于文化。

Y 通常指代我们期望实现或竭力避免的结果，可能发生在个人层面（如效率）或组织层面（如市场份额）。

我们构建、评估和使用这些模型³是为了解决这类问题。我们致力于准确估计实施 X 后对重要成果产生的影响。⁴在报告效应时，我们强调两个关键特征：

1. 我们对效应**方向**的把握程度，即多大程度上肯定该实践是有益还是有害
2. 我们对效应**大小**的把握程度。我们会估算某些实践的相对影响力，也会说明这些预估结果的不确定性程度。

以下是今年我们要研究的一些能力：

- AI 采用
- 平台使用
- 平台使用时长
- 变革型领导力
- 工作重心稳定性
- 以用户为中心

以下是今年的一些成果和成果组：

- 个人绩效和身心健康(如工作倦怠)
- 团队绩效
- 产品表现
- 开发工作流程(如代码库复杂度和文档质量)
- 软件交付绩效
- 组织绩效

我们侧重于这些成果是因为它们本身就具有价值。当然,这一点在某些成果上体现得更为明显。如果您发现组织绩效和团队绩效与软件交付绩效毫无关联,那么您可能会接受较低的软件交付绩效。

然而,我们希望即使组织绩效不依赖于个人身心健康,您依然会优先考虑员工的身心健康。

重复的模型

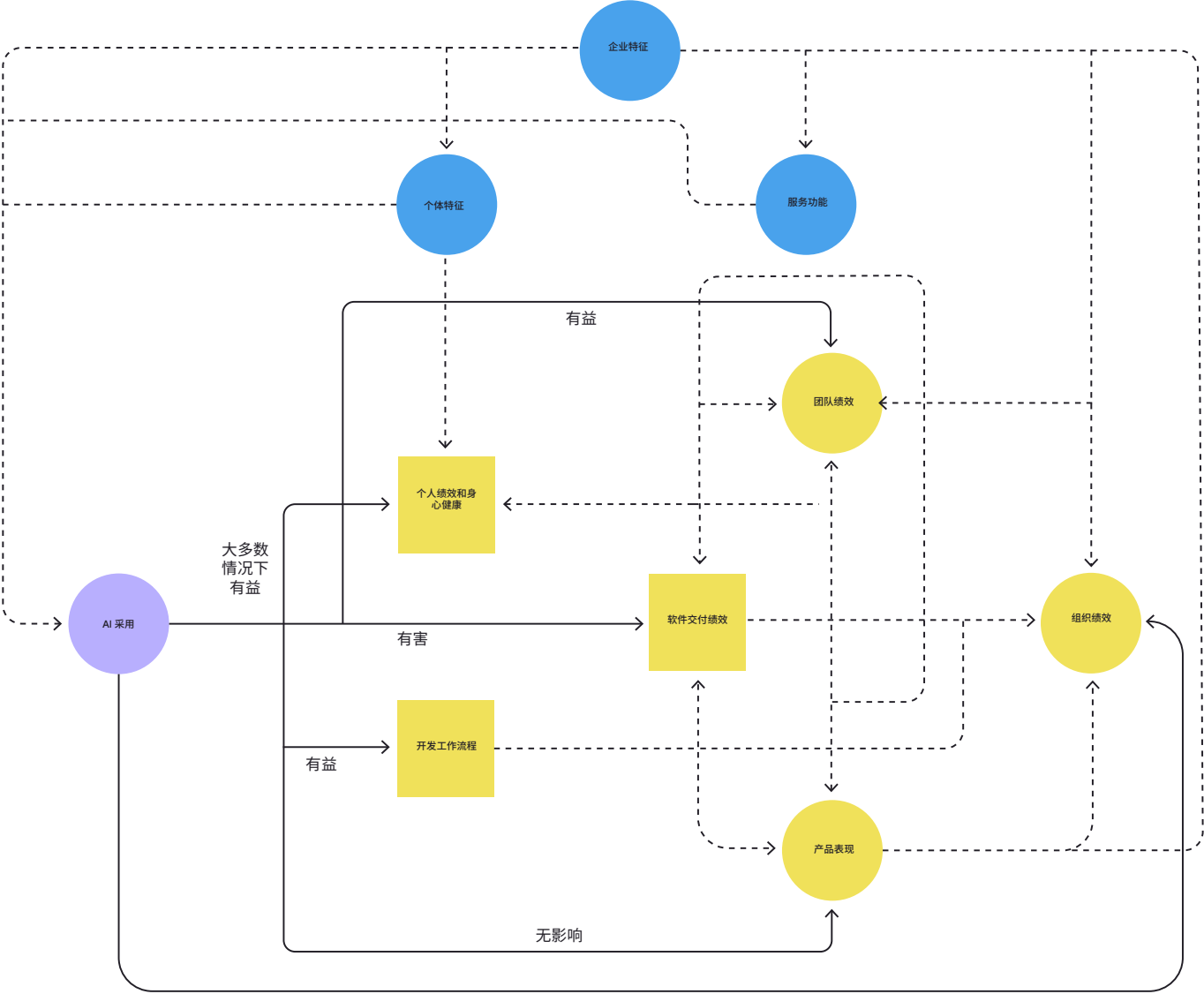
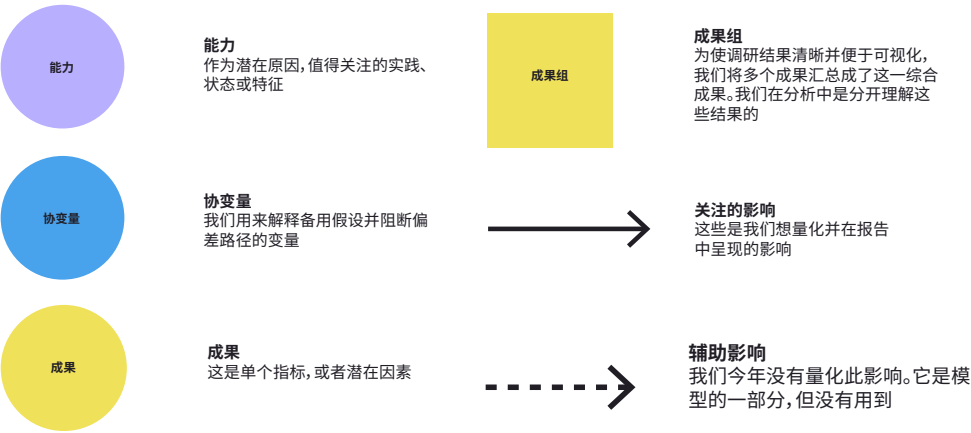
在过去三年里,我们制定和探索了许多细致的假设,尤其关注调节效应和中介效应。

今年,我们减少了对这些类型的假设的关注,而将更多时间放在尝试估算能力对成果的影响上。这意味着,每种能力所采用的模型大体一致。

例如,AI 采用的影响模型就与以用户为中心的影响模型非常相似。我们可以复制模型并更改能力的名称,但这可能对您意义不是很大。

因此,我们只展示 AI 模型。但请知悉,它是我们每个模型背后的框架或形式。如果您有兴趣运行自己的分析,在诸如 [DAGitty](#) 等工具中构建此模型应该能让您近似复现我们在分析中使用的回归结果。需要说明的是,为了提高可读性,这里所呈现的模型略有简化。此外,虽然不同能力对应的模型非常相似,其影响效应却各不相同。例如,您将在下文中看到,AI 采用通常会对软件交付绩效产生负面影响,而内部文档和以用户为中心等因素则恰恰相反。请参阅各章节,了解更多详细信息。

图例



1. Gelman 等人的《Regression and other stories》(回归分析和其他实例) 在第 495 页至 496 页中提供了一些很有启发性的重要建议：B.6 拟合多个模型，B.9 有针对性地进行因果推理，而不是将其作为大型回归分析的附带结果

2. 《Statistical Rethinking》(统计反思) 第 6 章对此内容进行了详细讨论。我这里尤其要强调的是碰撞偏差。

3. 请参阅“研究方法”一章中关于这些模型如何与有向无环图联系起来的讨论。

4. 我们在“研究方法”一章中简要讨论了因果关系。

推荐的阅读内容

加入 DORA 社区进行讨论、学习和协作, 以便改进软件交付绩效和运营绩效。

<https://dora.community>

完成 DORA 快速评估。

<https://dora.dev/quickcheck>

探索那些能够营造学习氛围, 并实现快速流动和快速反馈的功能。

<https://dora.dev/capabilities>

建立开发者对生成式人工智能的信任。

<https://dora.dev/research/2024/trust-in-ai/>

阅读图书:《Accelerate: The science behind devops: Building and scaling high performing technology organizations (加速:DevOps 背后的科学:建立和壮大高绩效技术组织)。IT Revolution。<https://itrevolution.com/product/accelerate>

阅读图书:《Team Topologies: Organizing Business and Technology Teams for Fast Flow》(团队拓扑:组织业务和技术团队以实现快速流动)。IT Revolution Press。

<https://teampatterns.com/>

DORA 研究计划出版物, 包括往年的 DORA 报告。<https://dora.dev/publications>

研究和报告相关常见问题解答。

<http://dora.dev/faq>

勘误表 - 查看和提交本报告的变更、修正和说明。<https://dora.dev/publications/errata>

查看这是否是 2024 年 DORA 报告的最新版本:<https://dora.dev/vc/?v=2024.3>

《加速:2024 年 DevOps 现状报告》由 Google LLC 制作,
已获得 [CC BY-NC-SA 4.0](#) 授权

