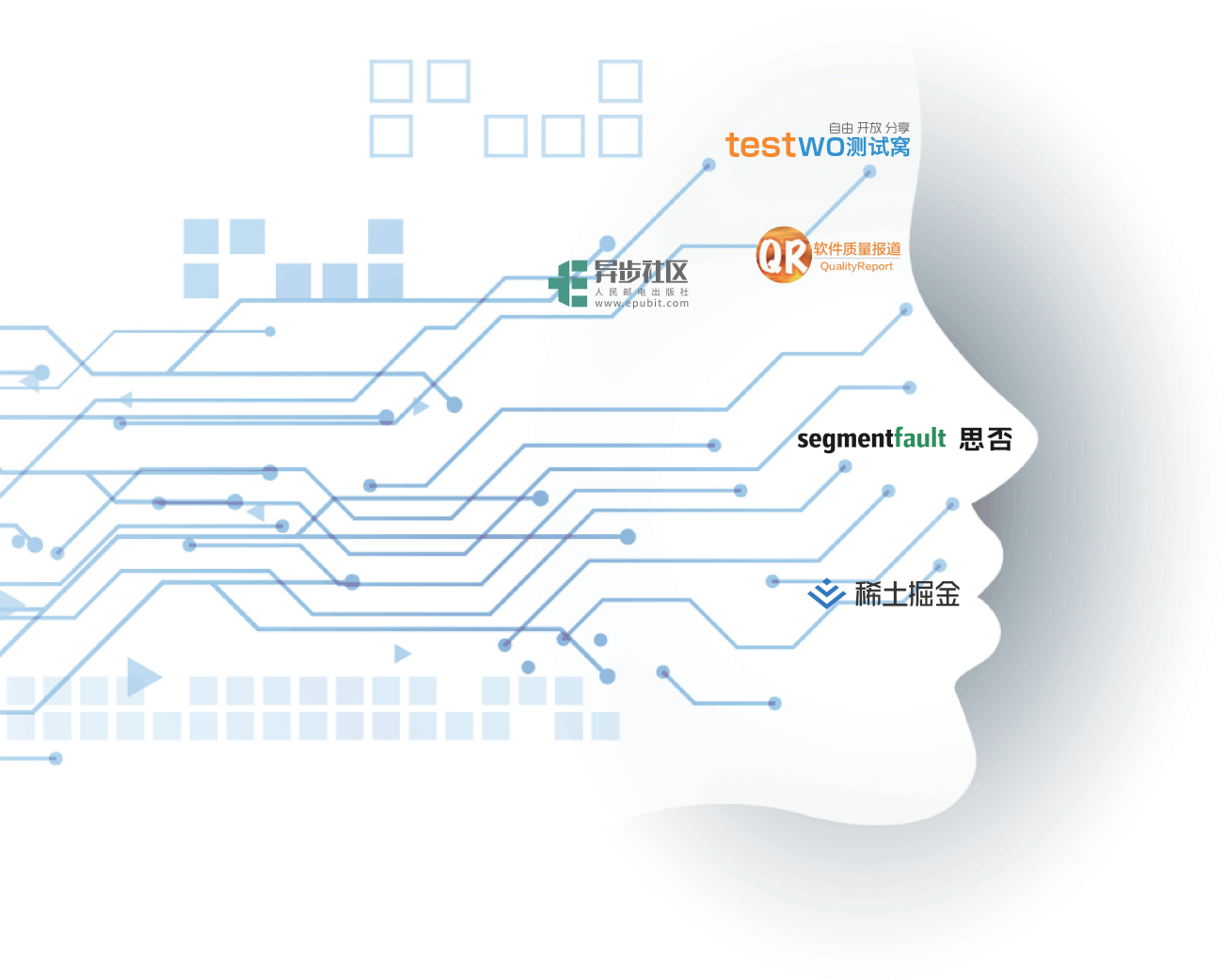


2023年 软件研发应用大模型

国内现状调查报告



自由 开放 分享
testwo 测试窝

异步社区
人民邮电出版社
www.epubit.com

QR 软件质量报道
QualityReport

segmentfault 思否

稀土掘金

科技生态圈峰会 + 深度研习

—1000+ 技术团队的选择



K+峰会

K+峰会

上海站

K+全球软件研发行业创新峰会

时间: 2024.06.21-22

敦煌站

CTO训练营

时间: 2024.10.17-19

香港站

技术规划训练营

时间: 2024.11.10-12



K+峰会详情

AiDD峰会

AiDD峰会

上海站

AI+ 软件研发数字峰会

时间: 2024.05.17-18

AiDD峰会

北京站

AI+ 软件研发数字峰会

时间: 2024.08.16-17

AiDD峰会

深圳站

AI+ 软件研发数字峰会

时间: 2024.11.08-09



AiDD峰会详情

2023年

软件研发应用大模型 国内现状调查报告

AI驱动软件研发
效能提升十倍、百倍

目录

01调查获得的一些结论	01
02参与调查的基本情况	02
03国内使用大模型的概况	04
04在软件研发生命周期使用LLM	08
05未来展望	14
附录	15
特别鸣谢	16



大模型正在成为人工智能的新范式，也助推了AIGC应用的新一波浪潮，由此掀起了软件研发领域的全新革命。大模型不仅可以提供强大的内容生成能力，还能够处理复杂的任务，为软件研发带来了巨大的潜力和机遇，将软件工程带入3.0时代。然而，目前国内对于大模型在软件研发中的应用现状尚未有系统性的了解和总结。所以有必要开展这方面的调查，帮助行业的从业者或企业的管理者，更加准确地了解我国软件研发领域对于大模型的需求和应用场景，为相关行业提供更加精准的解决方案。



为了推动我国软件研发领域的发展，作为软件研发行业的重要参与者——AI+软件研发数字峰会（AiDD）组委会，联合软件质量报道、测试窝、异步社区、思否、稀土掘金等多个社区、组织发起「2023年软件研发应用大模型」国内现状调查。通过这次调查，旨在全面了解我国软件研发领域应用大模型的情况，探索大模型在软件研发中的优势和挑战，为进一步推动大模型技术的应用与发展提供有力的支持和指导。



「2023年软件研发应用大模型」国内现状调查于2023年10月中旬开启，历时一个月时间完成数据收集。下面就基于调查获得的数据进行梳理和解析，汇编成本调查报告，希望不仅给每一位读者带来启发和思考，还能给每一个软件企业或团队带来重大的参考价值，能够指导他们完成下一年度AI战略的重构、相关AI部署和行动的规划。本调查报告于AiDD峰会深圳站（2023年11月24日）首发公布，其中难免有不妥之处，敬请指正！

01 调查获得的一些结论

1. 国内绝大多数（68%）软件研发团队是积极关注和拥抱大语言模型（LLM）的，开始使用或积极使用LLM，使用时间也超过2个月以上。积极投入较多人力和资金的企业也接近一半。
2. 软件研发团队最想借助LLM生成代码，提升研发效率，其次用于答疑解惑。
3. 超过1/3的软件研发团队是采用成本最低的方式：直接使用第三方LLM API。
4. 超过一半调查者是使用GPT3.5、GPT4.0，其次是国外开源Llama 2。国内LLM中，讯飞星火大模型、百度文心一言大模型、阿里通义大模型排在前面。
5. 有专业LLM团队的企业还比较少，只有28.1%，但如果不限于LLM，扩展到AI人才，那情况就好多了，接近73%。虽然LLM人才短缺，但还是有较好的人才基础。
6. LLM应用主要有七大困难，体现在关键要素上，如缺少人才、缺少算力、缺少高质量的数据等，同时目前应用水平低、存在安全/隐私担忧、数据数量不足、LLM部署能力不足等。
7. LLM在需求这个环节主要应用的顺序，非常巧合，正好形成了一个较完整的、敏捷开发范式的需求过程。
8. LLM在设计环节发挥的作用是最弱的，更多时候是作为顾问、知识库来使用，为设计提供知识服务、提供建议。
9. LLM在编程上的应用，“代码补全”、“函数级代码生成”分别排在第一、第二的位置（均超过55%）。其次，分别是代码修复、代码评审、遗留代码解释、代码优化等。代码采用率超过30%的比例达到了36%，效率提升超过20%的比重达到46%以上，LLM在编程环节获得良好收益的可能性有望达到80%。概括起来，LLM在编程上的应用值得去投入。
10. 使用基于LLM的编程AI助手，ChatGPT占了1/3，遥遥领先，其次是GitHub Copilot、自研的工具；国内工具前3名依次是CodeGeeX、Comate、CodeArts Snap。
11. LLM在测试应用上，生成测试用例遥遥领先（超过了代码补全），达到了62.9%。其次是生成测试脚本。测试效率提升明显的（超过20%）比重达到了43.7%，比开发略低（开发者46%）。
12. LLM在运维环节应用是最薄弱的，空白的比重占到42.3%，在运维日志分析上是最高的，也只有17.5%。
13. 展望未来，绝大多数人（超过90%）都是乐观的，希望人人都有一个AI助手、极大地提升研发效率、显著地提升软件交付的质量、降低开发的成本等等。

02 参与调查的基本情况

本次调研，参加的企业超过100多家，主要来源于互联网企业，占比超过40%，其次来源于大金融、信息与通信、教育和文化娱乐等行业，如图1所示。

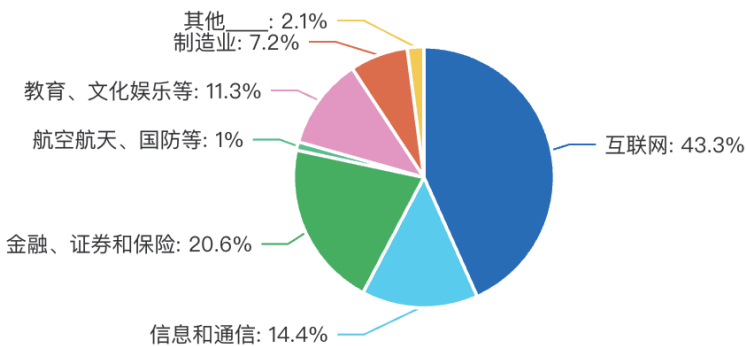


图1 参加调查的企业所处的行业

从参与调查的企业规模看，分布相对均匀，小型企业（300人以下）占的比重相对比较高，占了42.2%；中型企业（介于300人和10000人之间）占比接近41%，大型企业占比17.5%，如图2所示，基本符合实际现状，说明数据具有良好的合理性。

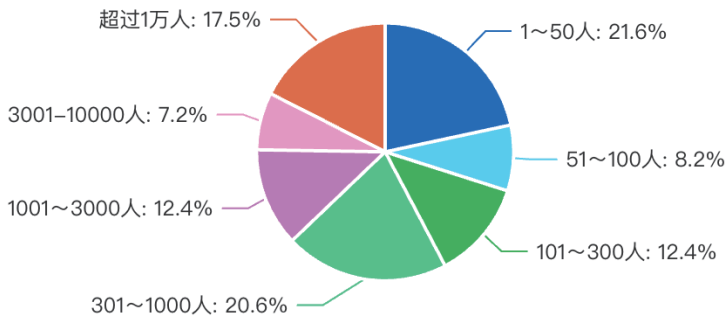


图2 参加调查的企业规模

软件研发团队主要用哪些编程语言，其结果基本在意料之中，Java、Python、C/C++、JavaScript遥遥领先，Go语言后来者直追而上。编程语言和大模型的训练也有关系，如salesforce公司就专门训练适合Python语言的代码大模型CodeGen-mono。如果要做类似的事情，Java、C++、Python等语言会被优先考虑。

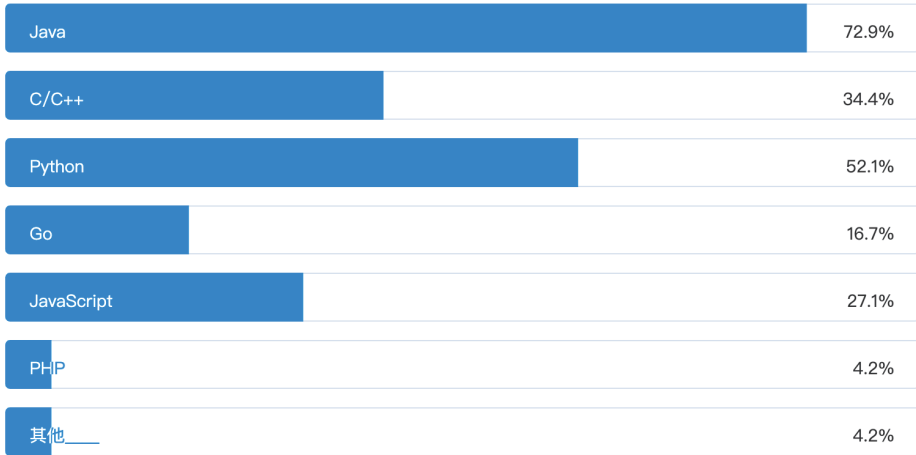


图3 调查者所使用编程语言的情况

03 国内使用大模型的概况

我们先看一下**国内软件研发团队对大模型（LLM）的态度**，绝大多数（68.7%）是拥抱LLM，关注、开始使用或积极使用LLM的，但也有1/4的团队或公司处在观望中，只是很少的团队（6.3%）不了解LLM，如图4所示。随着时间的推移，我们相信6.3%也会消失，大家都会主动去了解LLM的，那些观望者也会有所行动。

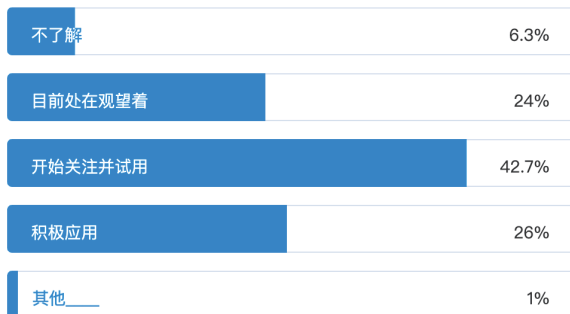


图4 调查者所使用编程语言的情况

态度决定一切，对待LLM态度是积极的，一般就会拥抱LLM，就会非常重视它，也乐意投入很大人力和资金，从而全面应用LLM。但有时心有余而力不足，缺少人才，公司效益不是很好，那就不可能会有很大的投入。我们从调研数据中看到，积极拥抱LLM并在LLM上投入很大人力和资金的企业数量还相对较少，比重不高，只有16.7%。如图5所示，重视并且投入较多人力开始研究和应用的企业不在少数，占了30.2%。这两类型企业加起来有46.9%，接近一半。再加上38.5%“投入少量人力试用”的企业，有不同程度投入的企业占了85%，这是一个很不错的数据。而不重视、非常不重视的企业只占少数，整体只有12.6%。从这个角度看，我们还是比较乐观的，说明有些观望的企业，也会投入少量的人力做些调查和试验。



图5 在LLM的投入情况

既然有超过三分之二的公司/团队拥抱大模型，使用情况如何呢？ 不了解或大部分观望者一般不会使用大模型，但少数观望者也会出于好奇尝试使用LLM，这不算在工作中正式试用。超过20%的公司/团队非常积极，在很早的时候就开始使用LLM，并且使用时长已经超过6个月，如图6所示。使用LLM超过2个月的比重占66%，接近2/3，算是积极拥抱大模型的，和上面对LLM态度的调查结果保持一致。

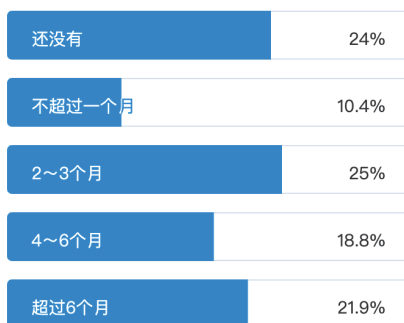


图6 国内团队使用LLM的时间

借助LLM发挥什么作用？ 可以看出软件研发团队**最想借助LLM生成代码，提升研发效率**；其次用于答疑解惑；然后就是用于代码检查、代码解释、生成注释行等以提升代码的质量，如图7所示。测试脚本和用例生成排在第4的位置，没排在第2，稍微出乎我们的意料，但正好有一半的团队会让LLM在测试上发挥价值。提升文档质量，LLM可以无处不在，相信未来，大家会有更多的使用。日常工作中，有接近一半人开始使用LLM来帮助我们生成文档、生成报告或润色文档等。而在需求、设计方向的LLM应用程度相对低一些，符合我们的预期。

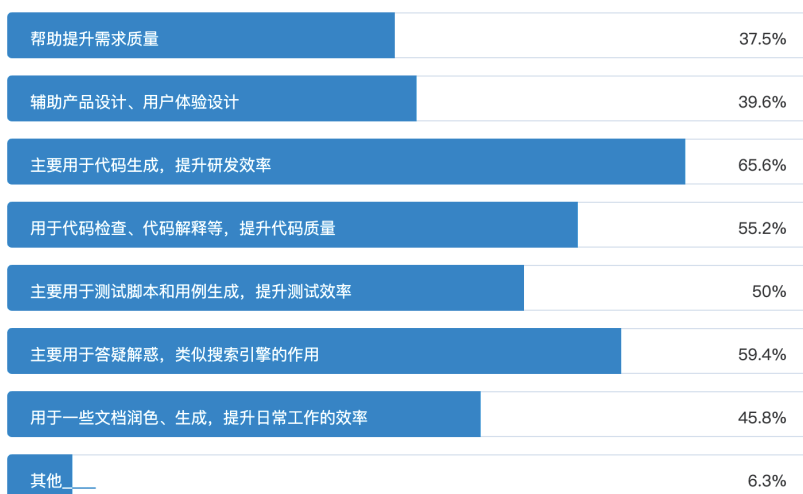


图7 国内团队想借助LLM发挥哪些方面的价值

如何使用大模型的技术呢？从图8可以看出超过1/3的软件研发团队是直接使用第三方LLM API，这种成本最低，适合中小企业。对于大型企业而言，可以训练基础大模型，但也不是每个大型企业都能做到这一点，所以这个数字（9.3%）低于直接调用第三方LLM API的比重（17.5%），但对大中型企业可以基于基础大模型（foundation model）训练（微调/精调 fine tuning）自己的领域大模型，这部分接近1/5（19.6%），还算比较高。更巧的是，领域大模型embedding特定的知识库和私有化部署正好都是12.4%，**是否意味着：一旦私有化部署LLM，那就必定会绑定自己的知识库？**

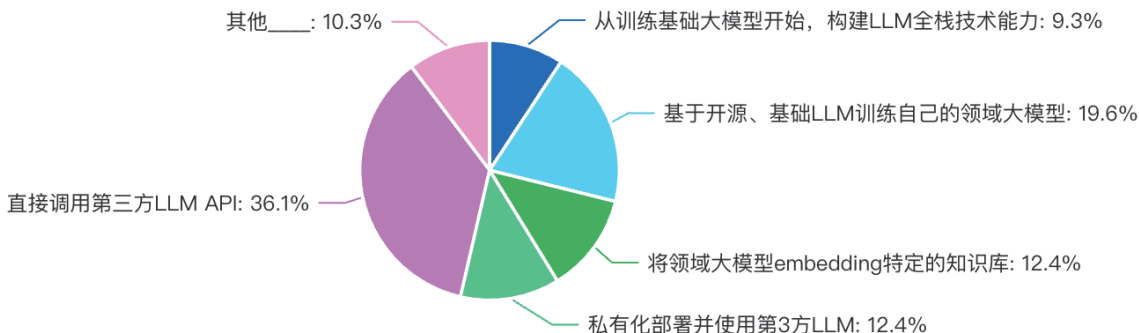


图8 国内团队使用LLM的方式

今年是大模型爆发的一年，国内呈现“百模大战”，那么大家主要使用哪种大模型呢？从图9可以看出，超过一半的企业或者团队使用的是GPT3.5、GPT4.0，在之前相当长一段时间，GPT3.5、GPT4.0是遥遥领先的，大家还是喜欢使用ChatGPT或调用OpenAI开放的API。其次，就是国外开源Llama 2。而国内LLM现状，讯飞星火大模型、百度文心一言大模型、阿里通义大模型排在前面。

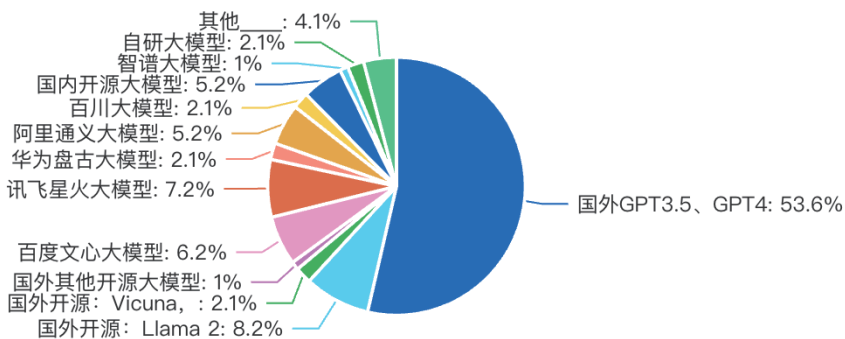


图9 国内团队使用哪些LLM

人们经常说大模型应用有三要素：数据、算法和算力，好的数据产生好的结果，但数据、算力是客观条件，而算法相对灵活，有着很大的挖掘和提升空间。对于大模型，算法更多体现在参数上，算法的设计影响了模型的参数数量、参数初始化方法、损失函数的选择、正则化方法等。这些因素直接影响了模型的训练效果、泛化能力和推理速度。而算法的设计最终依赖这方面的团队和人员。这次调查，在数据、算力等方面缺少一些问题，重点放在“人”之

上，毕竟人是决定性因素。

基于调查，有专业LLM团队的企业还比较少，只占28.1%，但如果考虑有LLM人才（虽然没有这方面团队）的企业，几者加起来可以达到44.8%，情况还不错，如图10所示。如果不限于LLM，扩展到AI人才，那情况就好多了，接近73%，超过2/3。有了AI基础，转型到LLM，相对比较容易。具备这样良好的基础，未来LLM的深度应用会更加普遍。



图10 LLM/AI团队现状

任何事情都不是一帆风顺的，特别是对新兴的LLM应用领域，遇到的问题还是比较多的。从调查结果看，LLM应用主要有七大困难（从高到低排列）：

- 1) 缺少机器学习和大模型训练人才
- 2) 缺少算力或云计算平台
- 3) 缺少高质量的（业务/研发）训练数据
- 4) 目前研发人员的应用水平不行
- 5) 安全、隐私担忧
- 6) （业务/研发）数据数量不足
- 7) 缺少大模型部署的能力

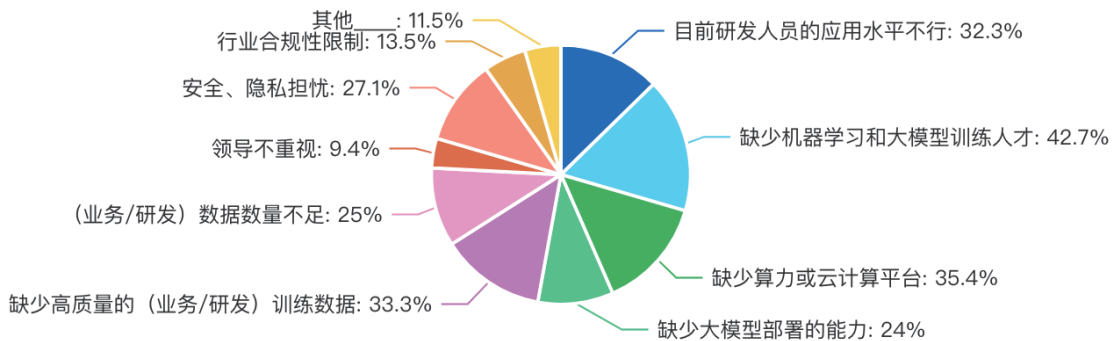


图11 LLM应用的困难因素

04 在软件研发生命周期使用LLM

了解了国内软件研发团队整体使用LLM的情况，下面接着分析在软件研发和运维的各个环节（需求、设计、编程、测试、运维等）是如何使用LLM的以及LLM发挥的作用、产生的效益。我们自然关注人们的使用方式，以及是否产生效益，如果产生良好的效益，那么就值得我们积极投入进去（假如之前还是旁观者）、投入更多的资源（假如之前已经是LLM的应用者）。

1. 在需求上应用LLM

虽然人们更为关注LLM在编程、测试方向的应用，但是今天我们可以和大模型进行自然语言的交互，可以逐步引导大模型理解我们的业务，帮助我们定义产品功能，甚至生成规范的需求文档。大模型生成文档的能力还是很强的，通过定义markdown等格式，可以生成文档模板所规定格式等文档，其次LLM对文档的内容进行总结、优化（润色）文档，所有这些能力都可以被应用到需求整理、需求分析和需求定义上。

根据调查（如图12所示），LLM在需求这个环节主要应用的顺序如下所示，很巧合，正好形成了一个较完整的、敏捷开发范式的需求过程：

- 1) 基于原始需求生成用户故事
- 2) 基于用户故事生成验收标准
- 3) 基于验收标准实例化需求
- 4) 检查/评审需求文档
- 5) 润色需求文档

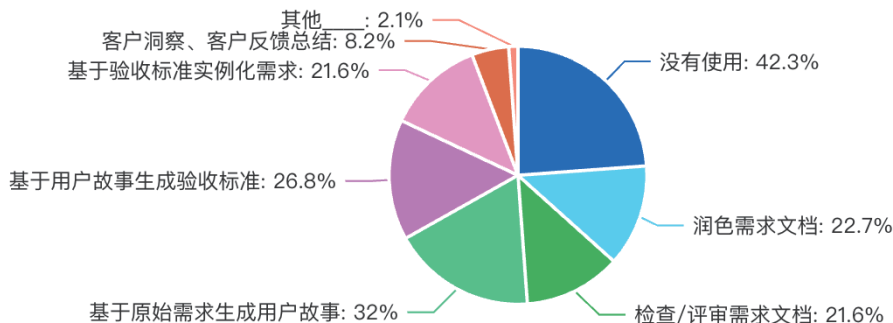


图12 LLM在需求上应用的状况

2. 在设计上应用LLM

从目前看，LLM在设计环节发挥的作用是最弱的，更多时候是作为顾问、知识库来使用，为设计提供知识服务、提供建议，如图13所示。其实，在GPT-4 Vision（多模态）发布之后，LLM在设计上能发挥更多作用。因为LLM可以看懂设计图了，可以帮助我们分析架构设计、UML设计图、UI设计等，指出其中存在的问题。如果设计图可以用一种特定的DSL语言（如XML、JSoN等）来表示，那么即使LLM画不出精准的设计图（如架构图、时序图等），也没问题，直接生成DSL语言，再导入特定的工具展示出来。



图13 LLM在设计上应用的状况

3. 在编程上应用LLM

在编程这个环节，正好是LLM用武之地，能够发挥很好的作用，所以在这个环节也设置了多个题目来了解更多的LLM应用的信息。首先看LLM在编程上的应用，从调查结果看，“代码补全”排在第一，略高于“函数级代码生成”，两者都超过了一半。其次，LLM在编程环节的应用，分别是代码修复、代码评审、遗留代码解释、代码优化，其差距不大，在30%左右，如图14所示。

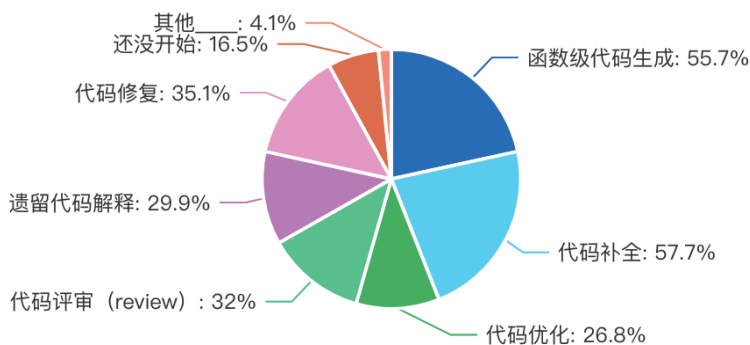


图14 LLM在编程上应用的状况

那我们看看效果如何？首先看看代码采用率，最好状态（超过50%的采用率）达到12.4%，是一个不错的成绩。根据百度最近发布的数据，代码平均采用率是40%，比较好的情况可以达到60%，这个结果也是可信的。如果针对某个产品线的代码数据进行fine-tuning，也许只有几百万行代码的数据，但效果可能很好，采用率超过50%，完全可能。

一般来说，超过30%的采用率，就算效果不错，那么这个占比达到了36%左右，刚超过1/3。采用率低于30%的占比超过一半，目前处在一种非理想的状态。还需要继续完善代码大模型，在fine-tuning上下功夫，获得更高的采用率。理想情况是达到70-80%的采用率，届时将是彻底解放大批码农。

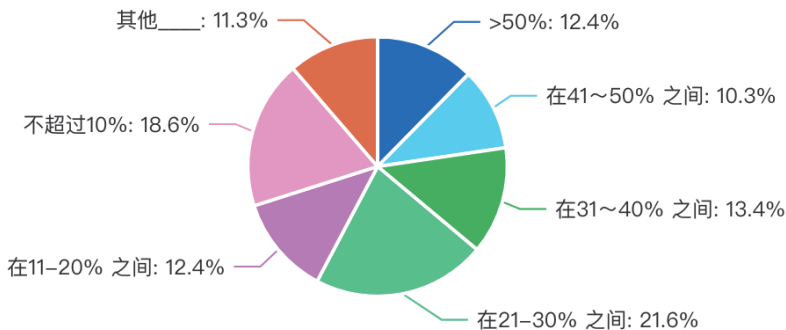


图15 LLM生成代码的采用率状况

衡量LLM在编程上的应用效果，我们还可以看另一个指标，那就是效率的提升。从调查结果看，如图16所示，提升一倍以上（超过100%）只有8.2%。按照GitHub copilot早期的统计数据，效率可以提升46%或更高。之后的GPT版本升级或做更多的精调，效率可以有更高的提升。而国内编程效率提升超过40%的团队，只有23.6%，占比还比较低，提升不明显的团队也占到20.6%。效率提升在10%~39%（假定平均提升20%）之间的占到44.3%。对大型企业，有几万员工，效率能提升20%，将是一笔非常可观的收益，例如假定1万研发人员，每人年人力成本（成本包含工资、三险一金、办公场地费用等）50万左右，一年可获得10亿的直接收益。如果算上交付周期缩短、产品竞争力提升，收益则更为可观。所以，即使效率没有提升50%、100%，对LLM的投入还是值得去做的，良好的收益预期能达到80%的可能性。

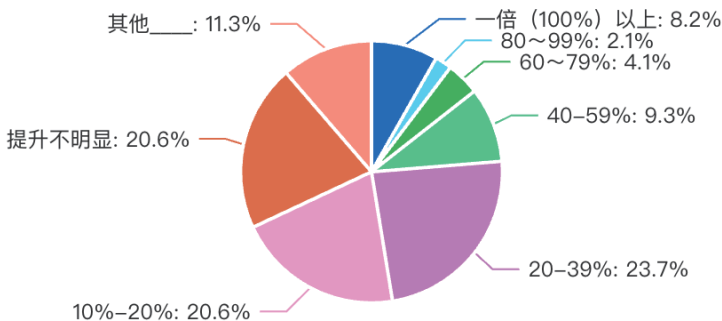


图16 LLM提升编程的效率

回到具体使用哪些编程副驾驶（copilot）或编程AI助手，可以看到chatGPT占了1/3，遥遥领先，如图17所示；其次是GitHub Copilot、自研的工具，其他各种工具占的比重都很低，国内工具前3名是CodeGeeX、Comate、CodeArts Snap。

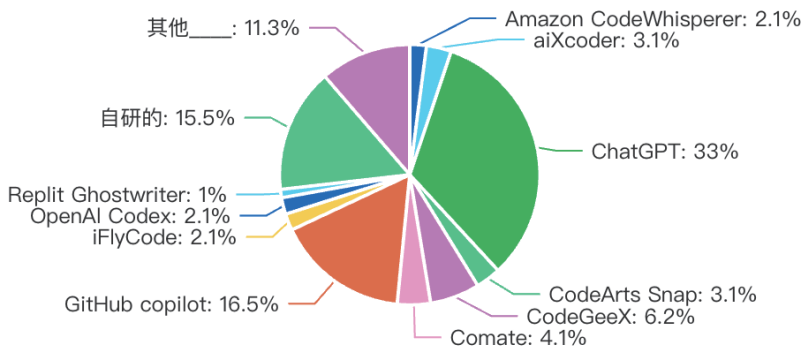


图17 基于LLM的编程助手被使用的统计结果

LLM在编程上的应用，最后一个调查项是在哪些AI框架上进行LLM模型训练，根据统计结果，（非常显著地）排在前3位的是：PyTouch、Tensor Flow和自研的，如图18所示。

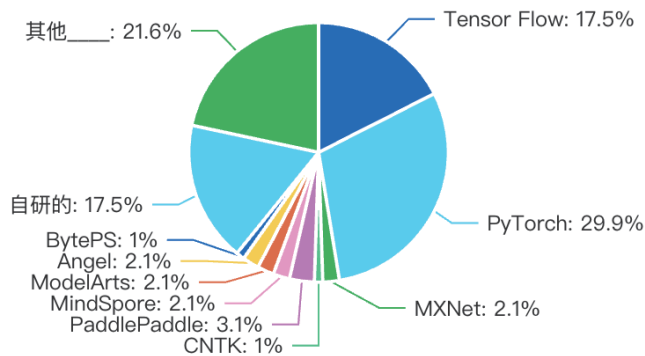


图18 模型训练框架被使用的统计结果

4. 在测试上应用LLM

软件测试环节也是LLM用武之地，仅次于编程环节。LLM在测试环节上的应用，生成测试用例遥遥领先，甚至超过了上面提到的代码补全，达到了62.9%。其次是生成测试脚本，达到了41.2%，如图19所示。其它应用相对比较少，而且占比也比较接近，更巧的是，最后三项（4~6）占比相同。

- 1) 生成测试数据
- 2) 评审/改进测试用例
- 3) 测试日志分析
- 4) 测试报告改进或生成
- 5) 异常数据分析
- 6) 评审/改进测试脚本

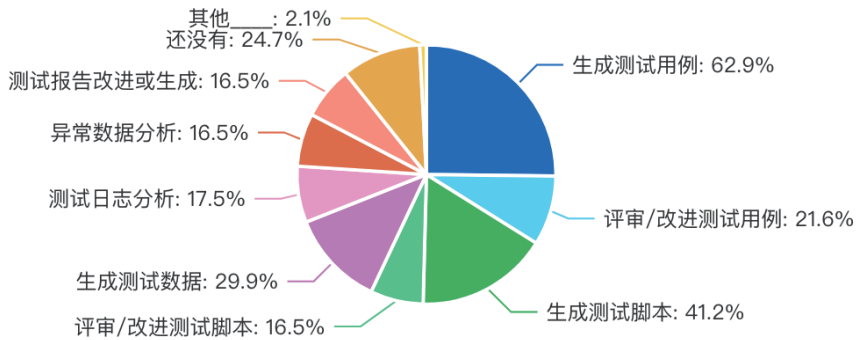


图19 LLM在测试上应用现状

那么LLM在测试上应用效果如何？提升显著的（假定超过40%）占比只有23.7%，目前还低于“提升不明显”的占比（25.8%），如图20所示。如果认为：超过20%就属于“提升明显的”，那么这个比重会上升近一倍，达到了43.7%。

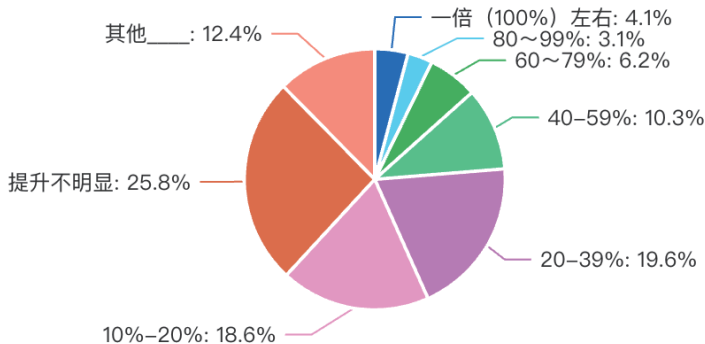


图20 LLM提升测试效率的统计情况

5. 在运维上应用LLM

根据调查结果，LLM在运维环节应用是最薄弱的。在运维方面使用尚且是空白的企业或团队，占到42.3%，在运维日志分析上是最高的，也只有17.5%，如图21所示。LLM帮助异常诊断、缺陷定位、生成数据统计图等应用很低，在10%左右。这部分的提升空间更大，需要我们有更多的探索、研究和实践。

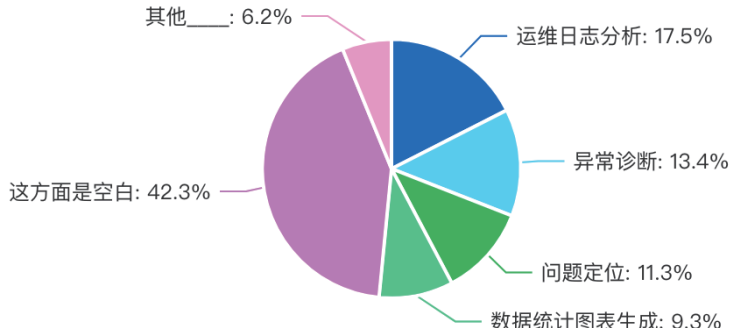


图21 LLM在运维上应用情况

05 未来展望

从革命性的自动化工具到创新的深度学习技术，AI正在不断改变着我们编程的方式，为开发者们打开了崭新的大门。展望未来，绝大多数人都是乐观的。持悲观态度的占比很少（只有2.1%），目前看不清楚的也只占到5%，绝大多数（超过90%）的人都对LLM寄予厚望。如果按照顺序，大家对未来的期望是：

- 1) 人人都有一个AI助手；
- 2) 极大地提升研发效率；
- 3) 显著地提升软件交付的质量；
- 4) 降低开发的成本；
- 5) 能够显著地缩短软件开发的周期；
- 6) 替换部分人员、缩减团队规模。

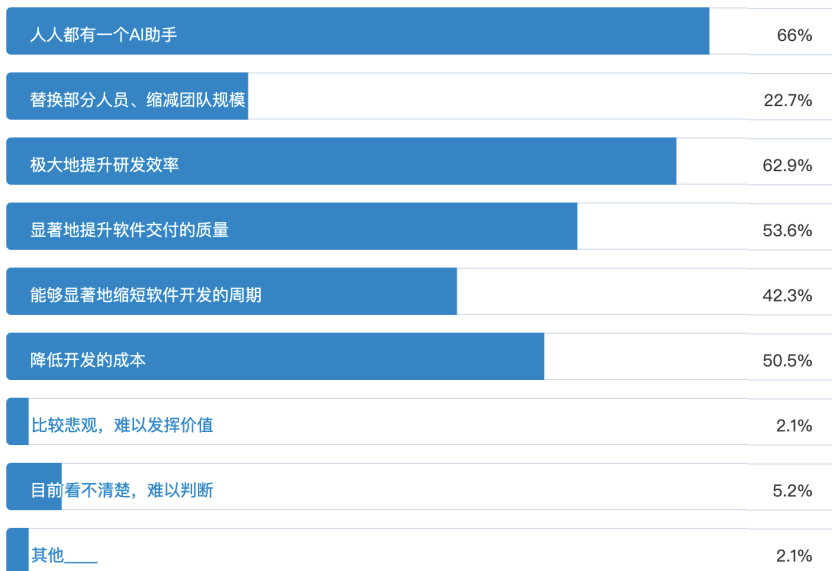


图22 LLM应用的未来展望

当我们回顾人类历史上最伟大的发明和创新时，往往会发现一个不变的趋势：科技的进步推动了社会的发展。现如今，我们正站在了又一个科技革命的风口上——人工智能（AI）。作为一项革命性技术，AI正逐渐渗透到我们的日常生活中，其中AI对于编程和研发效能方面的推动作用，无疑是一个备受瞩目的话题——它不仅关乎技术的前沿，更关乎科技对我们工作方式和生活方式的改变。因此，本次“软件研发应用大模型”调查报告的发布，将对行业发展起到里程碑式的意义。

LLM能发挥良好作用的应用场景清单

需求、设计、代码生成、测试、运维等全链路的提升

- **业务与需求**

需求分析和用户体验、需求评审、业务报表输出、材料整理、数据诊断、数据多样性分析、报表分析、数据治理、数据洞察、SQL数据库命令生成（BI，目前有成熟应用 NL2SQL）、市场调研、业务归因、统一产品研发业务的领域语言；

- **设计**

架构解读、架构分析、技术方案设计、功能描述细化、UML图的解析、架构内功能的初步效果验证；

- **编程**

代码补全、代码生成、遗留代码解释、代码注释、代码教学、代码重写、优化代码、缺陷自动修复、公司级别代码库、算法矫正、集成到代码提交门禁中；

- **测试**

单元测试用例生成、系统测试用例分析及自动生成（包括由详细的需求生成测试用例）、自动化测试数据生成、提高验收测试效率、问题定位、安全检查、系统安全性智能评估、业务全生命周期质量监测与智能评估、异常检测分析；

- **运维**

自动日志分析和总结、定期通知运维、运维报告自动生成、客户声音、智能客服版本发布（发布与需求的自动关联）、在线监控、、对系统状态进行监控并给出推荐设置、产品运营或软件运营分析；

- **文档及其它**

答疑或知识问答、启发（如通过多次询问，生成不同思路的回答，来增加视角，从中获取启发）、经验条目转化为知识、形成专家系统或个人知识库、产品介绍客服、文档优化、文档规范化、软件翻译、项目安排、流程调度、风险评估、多个 prompts 基于 context 的串联和整合。

特别鸣谢!

AI+软件研发数字峰会 (AiDD)



AiDD大会由K+峰会组委会和人工智能(AI)精英社区联合发起, 在高层次的技术委员会指导下, 聚焦AI驱动软件开发, 秉承“全心全意服务行业和国家高质量发展, 赋能企业和个人成长”的宗旨, 搭建一流的工业界、学术界的技术与实践交流平台, 帮助国内企业借助AI技术, 使计算机能够更深入地认知现实世界, 推动软件研发全面进入数字化时代, 软件研发效能得到十倍、百倍的提升。

官方网站: <https://www.AiDD.vip/>

《软件质量报道》公众号



本公众号致力于健康、安全、绿色的软件生态, 分享软件质量管理、软件测试的思想、方法、技术与优秀实践, 追踪软件质量领域的热点, 及时报道软件质量管理的成功案例或质量事故, 以及分享深度思考、有温度的技术文章等, 努力成为您工作中的朋友。



测试窝

自由 开放 分享
testWO测试窝

测试窝是以软件测试为主题的社区门户，作为开源非盈利的技术平台。一直坚持自由、开放、分享的理念。目前已成为最具影响力的软件测试工程师交流平台之一。

异步社区

 异步社区
人民邮电出版社
www.epubit.com

异步社区(www.epubit.com)是由人民邮电出版社创办的IT专业图书社区。异步社区于2015年8月上线运营，依托于人民邮电出版社20余年的IT专业优质出版资源和编辑策划团队，致力于优质学习内容的出版和分享，为读者提供优质学习内容，为作者提供优质出版服务，实现作者与读者在线交流互动，实现传统出版与数字出版的融合发展。

思否


segmentfault 思否

SegmentFault 思否是中国新一代技术问答社区。我们为中文开发者提供一个纯粹、高质量的技术交流平台以及前沿的技术行业动态。致力于凝聚集体智慧，推动技术进步。

稀土掘金

稀土掘金

稀土掘金是一个技术类内容分享社区，基于互联网技术的优质内容的发现、分享平台，包括iOS、Android、前端、后端、设计等分类下的每日精选内容。主要面对互联网公司的技术开发者、产品经理、运营等人群进行新技术的内容分发。移动端还有收藏、读过等对用户行为的记录，让用户能够更加方便的找到曾经阅读或收藏过的文章。



AI驱动软件研发
效能提升十倍、百倍



关注官方公众号



了解大会详情

NDD 峰会



了解大会详情

K+Talk



观看往期大咖对话



试听精选课程

