甲子光年
JAZZYEAR

2024

2024年AI代码平台及产品发展简报

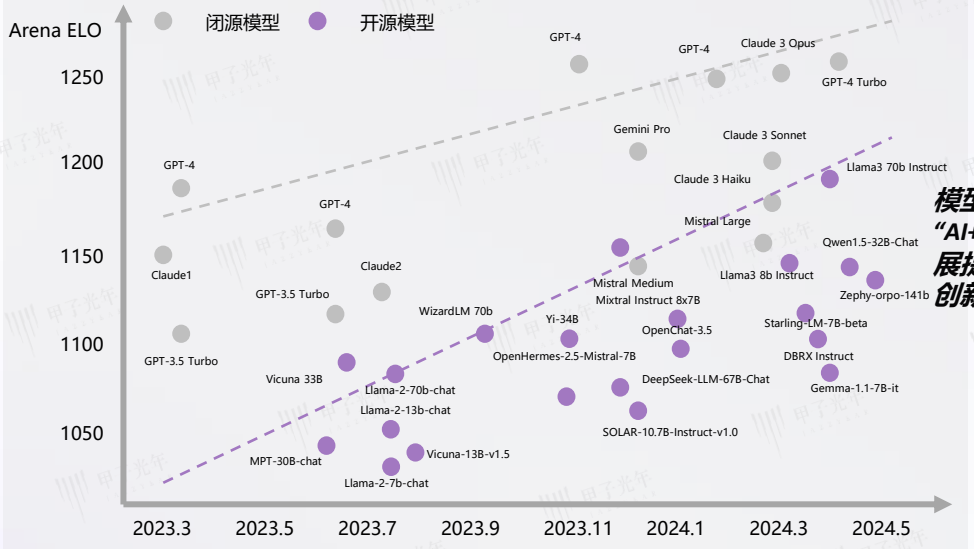
重构智能时代的开发模式，释放AI数字生产力

出品机构：甲子光年智库
智库院长：宋涛
撰写人：刘瑶 王艺霖 李哩莉
时间版本：2024.10

前言：生成式AI爆发后迎来商业加速，代码领域迎来创新及变革契机

- 随着人工智能技术的飞速发展，大模型（包括开源模型及闭源模型）的数量和能力都在显著增长，为生成式AI应用的爆发提供了坚实的技术基础。
- 代码及开发领域具备广泛的高质量数据、丰富的应用场景及多样的用户人群，因此生成式AI为代表的技术提供了生产力的创新空间和发展潜力。

大模型的数量及能力均不断上升，AIGC应用爆发具有技术基础



数据截止2024年5月

模型技术的蓬勃发展为“AI+”领域/场景/行业的发展提供坚实的基础及发展创新的土壤

代码结合AI迎来创新机会
开发领域亟待生产力升级

- ***高质量的数据***
- ***通用性的需求***
- ***广泛的使用人群***
- ……

目录

CONTENTS

甲子光年
JAZZYEAR

Part 01 机遇：AIGC引发的数字生产变化

Part 02 价值：重新构建代码开发的应用范式

Part 03 落地：智能时代AI+代码的先行者

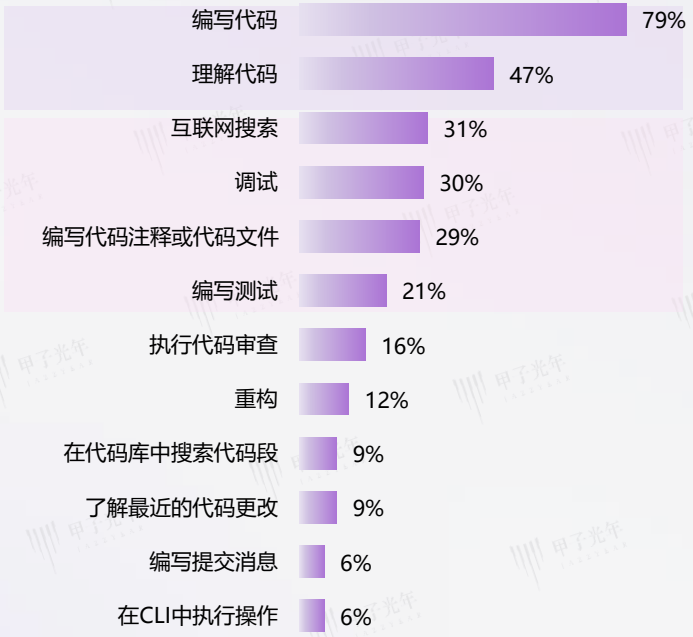
Part 04 展望：人工智能普惠时代的期待

需求破局：“AI+”的工作方式成为首选，释放代码工程的生产力

甲子光年 JAZZYEAR

- 编写代码，理解代码及互联网搜索、调试、写注释、写测试等工作是开发者的高频工作需求，因此解决围绕代码解决问题实际上解决开发者最高频刚需问题，随着生成式AI技术能力提升，更多的开发者都在尝试使用AI解决问题。

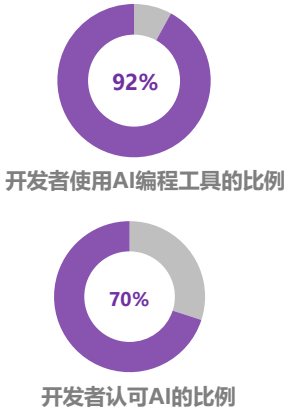
开发者最耗时的活动

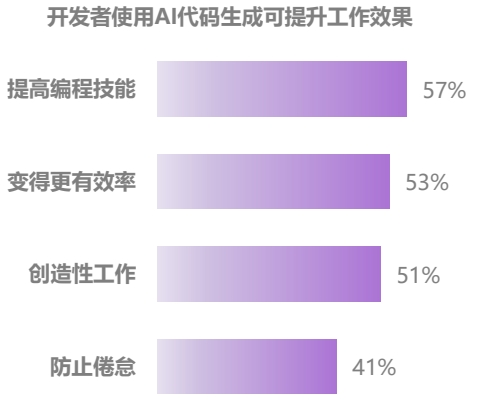


JetBrains，State of Developer Ecosystem Report（2023）

代码及开发工作不仅仅是“脑力活”，也是“体力活”

开发者会选择尝试AI编码提升自身的效率及技能水平



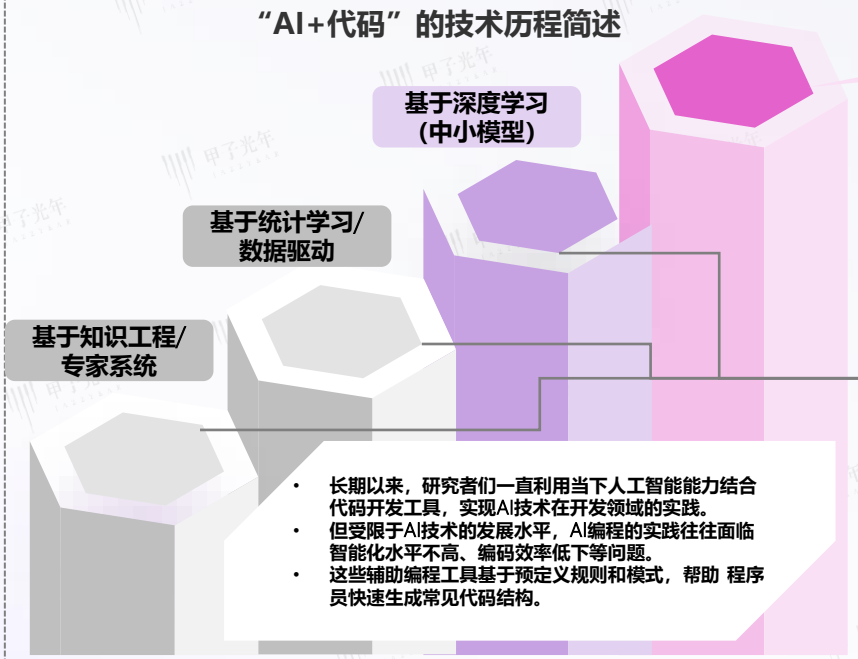


GitHub（2023）

技术升级：大模型赋予了“AI+代码”更多的创新空间

- AI生成代码的应用思路可以追溯到AI技术应用初期，但往往受限于当时的AI技术所体现的智能化水平。
- 大型语言模型（LLMs）的出现为深度学习带来了新的范式，尤其是针对语言类型的数字内容生成能力提升，因此也为“AI+代码” 进一步发展带来了更多创新空间。

“AI+代码”的技术历程简述



- **长期以来，研究者们一直利用当下人工智能能力结合代码开发工具，实现AI技术在开发领域的实践。**
- **但受限于AI技术的发展水平，AI编程的实践往往面临智能化水平不高、编码效率低下等问题。**
- **这些辅助编程工具基于预定义规则和模式，帮助 程序员快速生成常见代码结构。**

大模型为代码生成带来了“质”的改变

LLMs给“AI+代码” 提供了一个**突破性技术方案：**LLM带来了深度学习新范式，思维链和强大的自然语言理解能力，从而理解程序员的需求和意图，自动生成符合规范或者采纳率更高的代码片段或完整功能模块，从而让**创建广泛应用且实用的AI代码平台**成为可能。

并且推动了开发者的编程习惯和开发方式发生转变，虽然工作重心依然在代码编写上，但与AI的互动逐渐增多，编程习惯和开发方式开始发生转变。未来，开发者可能开始从代码的具体实现转向更高层次的任务管理和决策。

过往的工作主要集中在代码生成及搜索的特定能力，更加注重算法设计和训练策略。事实证明，**增强模型固有能力是推动智能代理进一步发展的关键因素。因此大模型对于泛语言形式的数字内容智能化水平的提升，成为推动AI辅助代码生成的关键。**

Copilot模式/IDE插件作为主导：
AI通过自动补全代码、生成代码片段等方式帮助开发者完成繁琐的编程任务。开发者依然在整个开发流程中起主导作用，AI的角色主要是提高编程效率和降低重复劳动。

产品形态的改变

Agent模式逐步被采纳，与Copilot模式/IDE插件共同组成丰富的产品功能：
AI不仅仅是代码生成的助手，还能够自主处理一些开发任务。AI在这一阶段表现出更高的自主性，可以减少人工干预。开发者的角色从编写代码逐步向监督和管理AI输出转变。AI的作用从简单的助手扩展到可以独立完成特定任务的合作者。

数据来源：公开资料，专家访谈，甲子光年智库整理，2024.10

产品红利：围绕开发项目的全环节和需求进行展开

甲子光年 JAZZYEAR

- AI代码平台的功能围绕具体的开发环节及场景展开，AI技术是代码平台的能力，当具体的产品功能服务依然围绕项目提质增效展开。

广义的AI代码平除了核心的产品功能外包括企业级开发能力（和个人开发能力做对应，强调其系统性的开发能力），另外包括用户体验的设计，服务生态的建设，另外包括与算力、数据存储、环境部署的适配。

狭义的AI代码平台往往指代的是核心产品的功能，也会被称为AI编程助手，主要指在代码开发过程中辅助编写代码的功能。

产品功能类型主要类型可以概括如下：

- 生成与搜索
- 修改与重构
- 注释与解释
- 审查与测试

AI代码平台的产品功能架构图示例

应用场景
金融 泛互联网 运营商 国央企 科技服务商 汽车 零售 工业

企业级开发能力

企业看板	知识增强	插件管理
用户管理	安全保护	多Agent管理
API管理	训练调优	成果共享
DevOps(需求分析、代码编写、调优对齐、数据回流)		

产品核心能力

代码生成	知识管理	注释生成
代码修复	智能推荐	代码解释
代码注释	代码翻译	智能问答
单元测试	风格检查	……

用户体验	服务生态
自然交互	多语言
自动/手动	插件管理
功能唤醒	Agent开发
自定义	低代码/高代码
风格学习	多模型管理
逻辑理解	多产品集成

数据存储、算力、网络等产品支撑

云服务/终端部署	算力异构调度	网络调度	数据存储/管理

数据来源：IDC等公开资料，专家访谈，甲子光年智库总结整理，2024.10

功能简析：技术能力和交互模式的融合升级

产品功能简析-解决贯穿开发环节的典型问题（1/3）

产品场景及功能类型示例		AI代码开发平台的产品价值
生成与搜索	代码补全/生成 智能搜索 ……	**提高了代码的质量和稳定性：**通过实时代码生成和补全，AI代码平台减少了开发者在编写代码时的手动输入，使得他们能够更快地实现功能。这种自动化不仅提高了编码效率，还减少了因手动输入而导致的错误。 **减少重复性的琐碎工作：**使他们能够专注于更有挑战性和创造性的任务。AI代码平台通过智能搜索和代码预测功能，帮助开发者快速找到所需的信息和解决方案，从而节省了大量搜索和调试的时间。这使得开发者可以将更多的精力投入到创新和优化代码结构上，推动项目的创新和发展。
修改与重构	代码修复 代码重构 ……	**更自然的交付体验：**“对话式”的调试方式，使得开发者能够以更直观和自然的方式与代码进行“对话”，从而更迅速地理解和解决问题。这种及时的反馈机制，尤其在软件开发的早期阶段，能够及时修复错误，避免后续的连锁反应，从而提升代码的整体质量和稳定性。 **优化和升级变得更加快速和高效：**智能化的重构不仅减少了人工重构的工作量，还降低了重构过程中引入错误的风险。自动化重构的实现，使得代码结构的优化和升级变得更加快速和高效。
注释与解释	添加注释 代码解释 代码翻译 ……	**提升代码可读性：**帮助开发者更快地理解和掌握复杂和陌生的代码。这种能力对于处理大型项目或维护遗留系统尤为重要，因为它减少了开发者在理解代码逻辑上的时间消耗，从而提高了开发效率。 **帮助开发者理解软件程序的结果和语义：**确保产出的代码符合企业的设计原则。这一功能对于保证软件质量和一致性至关重要，因为它减少了因误解设计意图而导致的错误和返工。 **优化现有代码的可移植性：**提高了代码的适应性和可维护性，还为企业节省了时间和资源，因为代码转换通常比从头开始编写新程序更加高效。
审查与测试	代码审查 性能检查 单元测试 ……	**提升单元测试的效率：**AI代码平台通过自动化功能，极大地提高了单元测试环节的效率。它能够辅助开发人员编写更多的测试用例，确保软件程序的有效运行。这种自动化不仅提高了测试的覆盖率和质量，还减少了开发人员在编写和维护单元测试上的时间和精力。这使得开发人员可以将更多的注意力投入到软件的设计和创新上，从而提高软件的整体质量和用户体验。 **人工驱动转为数据驱动：**通过收集和分析大量的代码数据，AI代码平台能够提供更加精准的测试建议和代码优化方案。这种数据驱动的方法，不仅提高了开发效率，还降低了软件缺陷的风险。

数据来源：公开资料，专家访谈，甲子光年智库整理，2024.10

功能简析：技术能力和交互模式的融合升级

甲子光年 JAZZYEAR

产品功能简析-解决贯穿开发环节的典型问题（2/3）

AI代码开发平台常见产品功能特点		未使用AI代码可能面对的问题
✓ **代码补全：** 在用户输入按键后自动提供后续代码的建议。用户可以将其当作参考，也可以采纳其为正式代码。 ✓ **智能搜索：** 可将对话问答功能与 IDE 进行了集成打通，其中的聊天互动界面可以让开发者通过自然语言来做提示词，探索和生成大段新代码，甚至是整个程序。并支持一键插入到编辑器中。	**生成与搜索**	x **手动记忆：** 程序员需要记住大量的函数、方法和库的名称及其参数，这在面对庞大的代码库或复杂的编程语言时尤为困难。 x **信息过载：** 在没有智能搜索的情况下，程序员可能需要在大量的文档、论坛和代码库中手动搜索解决方案，这既耗时又容易出错。
✓ **代码修复：** 针对编码过程中出现的代码问题，可以一键给出修复建议，助力开发者快速解决代码问题。 ✓ **代码重构：** 在不改变软件功能的前提下，主动提出建议给出选择，对代码进行结构上的调整。	**修改与重构**	x **定位问题：** 在没有智能辅助的情况下，程序员需要手动分析和定位代码中的错误，这可能非常耗时，尤其是在大型项目中。 x **识别重构机会：** 确定哪些部分的代码需要重构，以及如何重构，通常需要深厚的经验和对代码结构的深入理解。
✓ **添加注释：** 支持多种编程语言生成注释，只要选中代码段落，用按键选择生成注释，AI 编程助手就能一键生成方法注释及行内注释。 ✓ **代码解释：** 支持开发者将代码片段粘贴到聊天界面，并获得自然语言解释，还可以在 IDE 环境中突出显示需要解释的代码片段。 ✓ **代码翻译：** 将代码从一种编程语言翻译成另外一种编程语言，节省相应的时间。	**注释与解释**	x **时间消耗：** 编写清晰的注释需要额外的时间，程序员可能因为项目进度压力而忽视这一步骤。 x **理解差异：** 不同程序员对同一代码的理解可能存在差异，这可能导致沟通上的障碍。 x **语义保留：** 代码翻译不仅仅是文字的转换，更重要的是保留代码的语义和逻辑，这在自动翻译中很难实现。
✓ **代码审查：** 自动审查用户代码，支持对代码进行函数级检查功能，审查代码的可读性、可维护性、一致性以及是否遵循编码规范。 ✓ **性能检查：** 支持对代码进行函数级检查功能，可以对大篇幅代码做检测，检测代码性能，评估代码在运行时的性能，如处理速度、内存使用等。 ✓ **单元测试：** 根据自然语言提示或按键选择，为指定范围内的代码自动生成单元测试。	**审查与测试**	x **主观性：** 代码审查很大程度上依赖于审查者的个人经验和偏好，这可能导致不一致的审查标准和结果。 x **难以预测：** 在没有工具辅助的情况下，预测代码的性能表现可能非常困难，尤其是在复杂的系统或多线程环境中。 x **测试成本：** 编写单元测试需要额外的时间和努力，随着代码的更新和迭代，维护相应的单元测试也变得复杂和耗时。

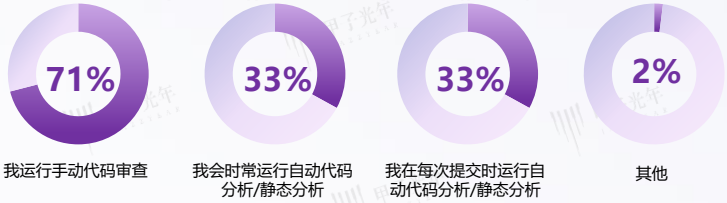
数据来源：公开资料，专家访谈，甲子光年智库总结整理，2024.10

功能简析：技术能力和交互模式的融合升级

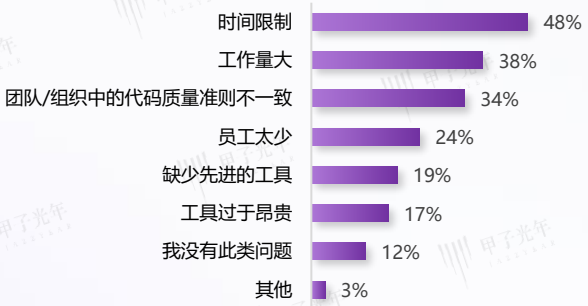
产品功能简析-解决贯穿开发环节的典型问题（3/3）

- 生成式AI技术的出现，除了可以极大地支撑代码生成、重构、补全等非常典型的开发工作。此外于完整的AI技术开发平台及产品也应该提供相应的审查及测试类产品技术支持，以保证软件开发全链路功能的覆盖。

图：审查类工作在代码开发中的时间精力消耗



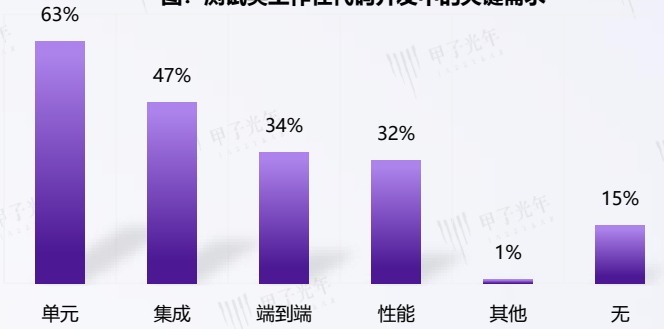
问题：您使用那种（哪些）工具进行自动化代码分析/静态分析？



问题：您的项目/组织保持高质量代码的最大阻碍是什么？

JetBrains，State of Developer Ecosystem Report（2023）

图：测试类工作在代码开发中的关键需求



问题：您的项目中有什么类型的测试？

17%
在项目中有单元测试的受访者不自己编写测试。

80%
的受访者表示测试在其软件开发项目中发挥着不可或缺的作用。

58%
参与测试活动的受访者开发自动化测试。

46%
的受访者将测试用例设计作为测试过程的一部分。

JetBrains，State of Developer Ecosystem Report（2023）

功能简析：技术能力和交互模式的融合升级

交互模式的创新-基于问答与上下文补全的需求

- 问答式交互（也可理解为问答式的交付）：AI代码生成通过整合代码交互模型与聊天界面，实现了用户与AI模型间的自然语言交流，用户能提出跟进问题，模型则基于先前对话及代码上下文提供相关回应，确保交互的连贯性。同样，生成式AI的技术为该项功能提供了更好的完成度。

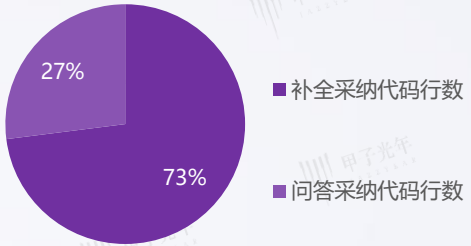
开发者使用AI助手的最常见方式之一是使用自然语言询问软件开发相关的一般问题

相当频繁	有时	极少	从不	
26%	33%	17%	24%	使用自然语言进行问答
24%	37%	24%	15%	生成代码
19%	26%	22%	33%	生成代码注释或代码文档
18%	26%	21%	36%	解释bug并提供修正
14%	27%	22%	37%	解释代码
12%	21%	24%	42%	生成测试
11%	21%	19%	48%	使用自然语言查询搜索代码段
9%	17%	21%	53%	执行代码审查
9%	16%	19%	55%	总结近期代码修改以了解代码内容
9%	20%	23%	47%	重构代码
9%	17%	20%	54%	通过自然语言描述生成CLI命令
6%	12%	20%	62%	生成提交信息

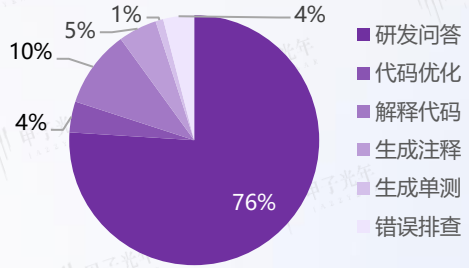
问题：您使用以下现有AI助手功能进行编码的频率如何？

JetBrains，State of Developer Ecosystem Report（2023）

补全与问答代码采纳分布



问答功能使用分布



通义灵码根据线上调研数据，《通义灵码，软件开发新范式》

数据来源：公开资料，专家访谈，甲子光年智库总结整理，2024.10

目录

CONTENTS

甲子光年 JAZZYEAR

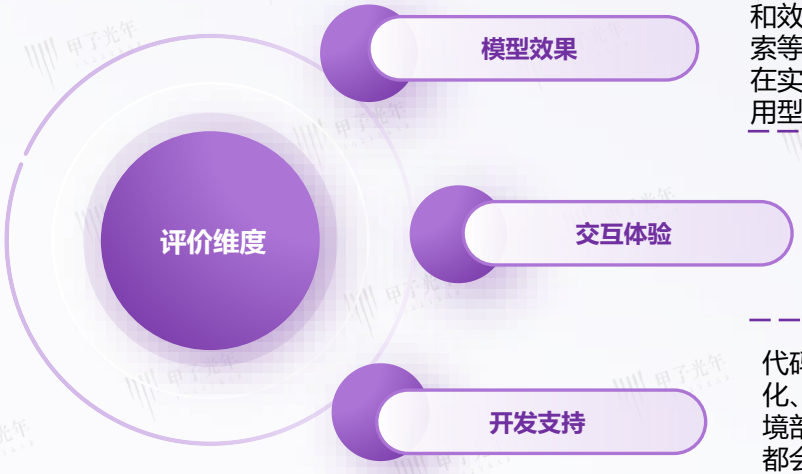
Part 01 机遇：AIGC引发的数字生产变化

Part 02 价值：重新构建代码开发的应用范式

Part 03 落地：智能时代AI+代码的先行者

Part 04 展望：人工智能普惠时代的期待

显性价值：AI代码平台产品竞争力维度分析



大模型带来的核心突破是生成代码的质量，提升了其有效性（生成和效果的提升），其次是带来了更准确的代码理解、重构、智能搜索等一系列功能实质效果的提升，因此模型的效果非常值得关注，在实际工作中，**其代码的采纳率**既是明确的体验感受，可以作为通用型的评价标准。

AI代码开发平台直接向用户提供的仍然是产品形态，因此产品的UI设计、用户交互方式，甚至是与他软件的兼容性，都会直接影响用户的“爽”感，也是在日益激烈的产品竞争中值得关注的维度。

代码的编写是整个软件开发过程中的一部分，软件开发则是系统化、复杂性的工程，产品的效果仍然要与整个工程效能有关。环境部署方式、数据处理效率，通信延时等具体产品设计细节最终都会影响实际效果。

广义的评级维度 = 代码采纳率 + 产品体验满意度 + 开发效能提升率

数据来源：公开资料，专家访谈，甲子光年智库整理，2024.10

隐形价值：重构智能时代的开发模式

从需求协作到软件发布，完成整个体系的智能化升级，赋能软件开发全生命周期

- AI代码平台及产品提供的功能可以在需求阶段、编码阶段、集成测试阶段、发布部署阶段、咨询学习阶段、效能管理阶段等各个阶段实现智能化升级，意味着AI代码平台能力不仅仅局限在某一点，而是逐步的向横向扩展，会贯穿到 DevOps 的整个链路，为整个软件工程开发提供新的范式思路。

图：AI代码平台在整个开发周期应用场景图

数据来源：《通义灵码，软件开发新范式》，专家访谈，甲子光年智库整理，2024.10

隐形价值：重构智能时代的开发模式

提升“工程效能”，快速响应需求并持续提升交付速度

- 生成式AI的大模型技术提升了代码生成的效率、质量，从而让团队更关注创新，并且降低了更多业务人员的参与开发的难度，从而提升整体工程效能。

显性需求：提高开发效率、代码质量改进、自动化测试等

大模型带来的价值

提升效率

编程速度提高

专注核心创新

自动化繁琐任务,专注研究和创新

降低技术门槛

为非专业程序员提供支持

追求理想工程效能

快速响应业务需求

持续提升交付速度

隐性需求：代码数字知识资产的管理与传承、创新能力提升、跨领域协作等

表：工程效能的指标拆解

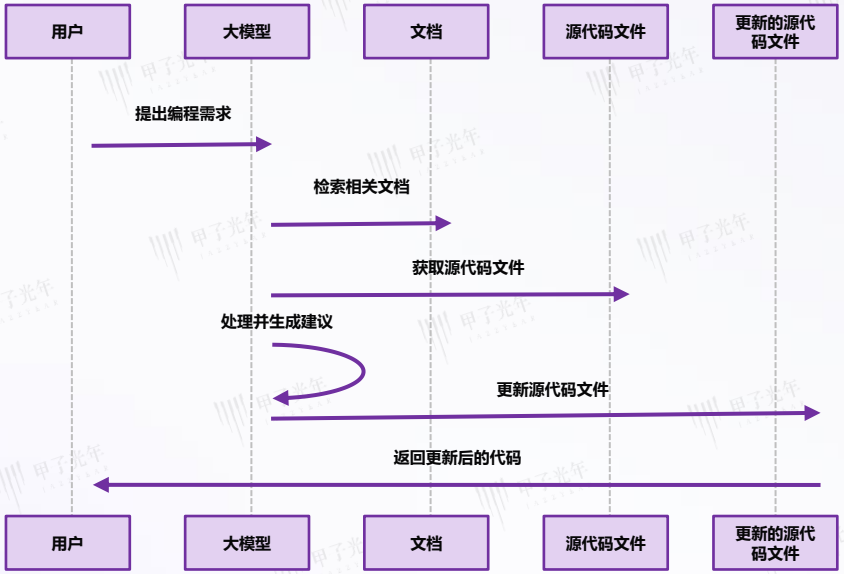
效能指标	指标含义	指标对应的计算思路
响应速度	能够快速响应新的需求或变更请求	• **代码提交频率**：衡量团队提交代码的频率 公式：代码提交次数 / 时间单位 • **代码审查速度**：衡量团队审查代码的速度 公式：代码审查完成时间 / 代码提交次数
交付速度	能够在较短时间内将需求转化为可上线的产品或功能	• **构建成功率**：衡量团队构建代码的成功率 公式：成功构建次数 / 总构建次数 • **测试覆盖率**：衡量团队测试代码的覆盖率 公式： 测试通过的代码行数 / 总代码行数) • **部署频率**：衡量团队部署代码的频率 公式：部署次数 / 时间单位
持续改进	随着时间推移，团队能够不断提高交付速度，实现"越来越快"的目标	• **代码提交频率**（同上） • **构建成功率**（同上） • **测试覆盖率**（同上）
质量保证	追求速度的同时，确保交付的产品或功能质量可靠	• **构建成功率**（同上） • **测试覆盖率**（同上）
问题解决能力	快速发现问题、定位问题和修复问题的能力	• **平均故障恢复时间**：衡量团队恢复故障的速度 公式：故障恢复时间 / 故障次数 • **平均故障间隔时间**：衡量团队故障发生的频率 公式：故障间隔时间 / 故障次数
适应性	灵活应对不同类型的需求和技术挑战	• **代码提交频率**（同上） • **代码审查速度**（同上）

数据来源：GTLC 全球技术领导力大会，甲子光年智库总结整理，2024.10

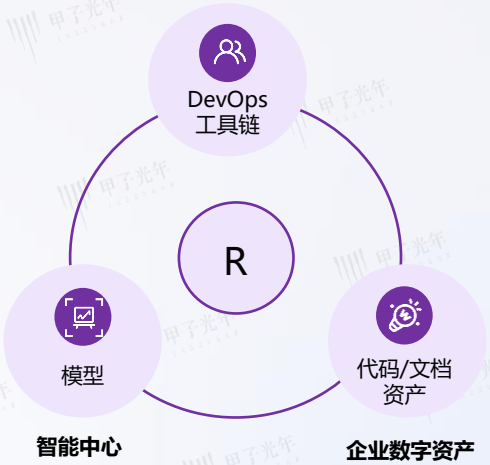
隐形价值：重构智能时代的开发模式

研发知识传递形态的改变，个人/企业软件开发工作知识资产的建立

- 随着企业研发工作开展，内部有很多优质的代码、框架、规范需要逐步积累，但往往这些数据的价值却很难得到释放，对于开发者而言的使用成本很高。
- 大模型时代的AI代码开发平台则可以借助大模型不断完善积累企业的代码数字资产，形成正向循环。



软件研发全生命周期



形成正向循环，开发者在开发过程中享受到“资产红利”

数据来源：《通义灵码，软件开发新范式》，专家访谈，甲子光年智库整理，2024.10

隐形价值：重构智能时代的开发模式

甲子光年 JAZZYEAR

提高整体研发效率，降低开发工程的复杂度，赋予更多企业智能化的机会

- 软件本质的复杂性是现在软件系统中无法 规避的内在特性，比如复杂度、一致性、可变性和不可见性——《人月神话》的作者。可以看出软件工程是大规模的集体智力的协作活动，本身具有极强的复杂性。
- 在大模型时代，当 AI 逐步地去替代事务性工作，并且形成了 AI 为主、人为辅的编程模式的时候，个体和写作中的效率竖井就可以逐步打破。提升整体的协同研发效果及效率。

“工程效能”是本质，快速响应需求并持续提升交付速度是关键

提升方向

工程能力和工程效能的双效提升

“需求来了上线快，而且随着时间越来越快”

人员技能是效能的基石，也是效能破局点

能力提升 **弥补能力短板**

成本是效能优化的目的，同时也是约束条件

工具赋能 **事务性工作替代**

软件架构和组织复杂度正相关，并决定协同消耗的大小

流程规范 **打造超级个体**

数据来源：《通义灵码，软件开发新范式》，专家访谈，甲子光年智库整理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

甲子光年 JAZZYEAR

RAG：致力于解决生成式AI“幻觉”，高效提升代码准确性

- RAG技术不断涌现，推动了模型性能的边界不断扩展。这些技术进步不仅限于提升检索效率和生成文本的流畅度，更涵盖了对上下文理解的深度增强、对复杂查询的精细处理能力，以及对多样化数据源的高效集成。

部分AI技术生成代码痛点

通常依赖于自身的参数和训练数据来生成文本。

幻觉问题	生成的代码不准确、通用过时、虚假，不符合预期需求或含有错误。
效率低	生成代码速度慢，影响开发进度。
缺乏上下文	从非权威来源创建响应，无法理解特定项目业务的背景信息，导致生成结果不相关或不完善。

实施RAG技术的AI代码平台

RAG= 检索+生成，利用外部知识库来增强AI生成能力。

技术由来	• 由Facebook AI Research（FAIR）团队提出。 • 为解决传统生成模型的局限性而发展起来。
工作原理	• 检索：从外部知识库获取信息。 • 生成：结合信息生成输出。
应用效果	• 提高生成准确性和效率。 • 提供上下文支持。

对比方面	不使用RAG技术的AI代码平台	实施RAG技术的AI代码平台
知识更新与深度	需要周期性训练以集成新知识；知识库更新滞后	实时检索最新代码库和文档，快速更新知识库；深度通过外部知识库增强
可解释性与定制性	黑盒模型，生成代码的逻辑不透明；定制性受限于模型预设	生成代码基于检索结果，可解释；易于根据特定编程需求定制
训练与应用成本	高成本训练模型；每次更新知识库需要额外训练	知识库更新无需重新训练模型；降低长期成本；提高应用效率
通用性与推理能力	通常针对特定编程任务优化，在其他任务上表现不佳	强大的跨任务通用性；通过检索增强上下文和逻辑推理能力
实时性与多模态	更新周期导致知识过时；通常处理单一数据类型	实时检索和应用最新信息；有潜力处理代码、文档、issue等多模态数据
安全性与隐私	训练数据包含敏感信息，存在泄露风险；模型透明度低	通过限制知识库访问和生成策略提高安全性；模型生成过程更透明

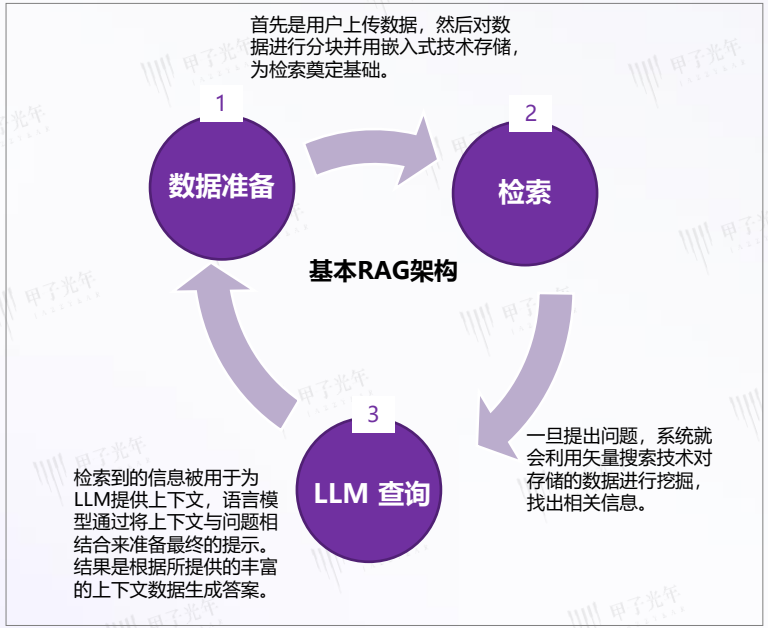
数据来源：甲子光年智库梳理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

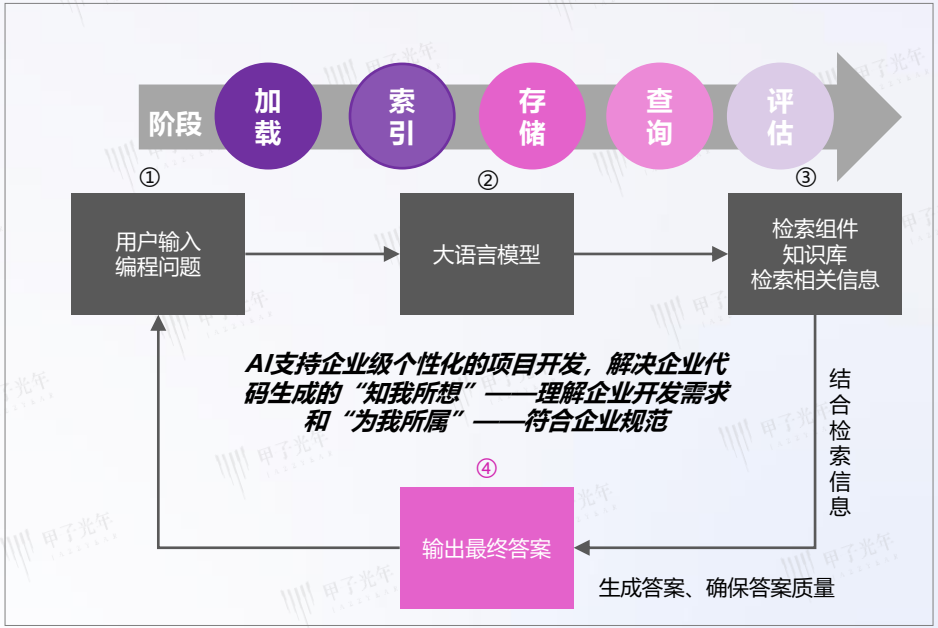
数据管理与智能查询相结合，以提高AI代码生成响应精度，提升企业级个性化开发能力

- RAG对大型语言模型（LLM）输出进行优化，使其能够在生成响应之前引用训练数据来源之外的权威知识库。
- RAG通过将用户数据整合进LLM已有的数据中来解决用户提问。是目前生成式AI应用中一种高效的改进思路，保证其输出结果在各种情境下都能尽可能体现其相关性、准确性和实用性——在AI代码生成的技术中，保证最大程度提升代码生成的质量。

图：RAG的技术架构示例



图：RAG解决编程问题的过程图

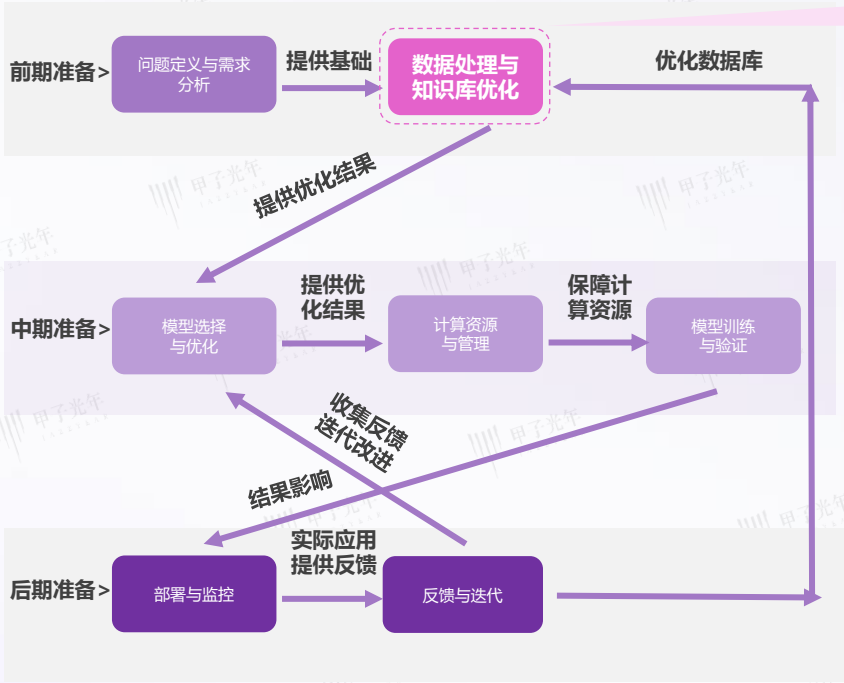


数据来源：甲子光年智库梳理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

RAG依赖于高质量的知识资产管理及数据处理能力，企业代码库的数据质量需关注

图：RAG的数据处理、反馈及优化的过程



针对生成效果评估方法的全流程优化
知识库处于流程循环首尾的衔接，其质量控制是优化策略核心

1. 数据处理与知识库优化

- ✓ **知识库**：优化知识库的结构和内容，以提高检索效率和质量。
- ✓ **数据**：需要重新考虑数据的收集、清洗和标注，以适应新的模型或策略。
- ✓ **数据切片**：将大块数据分割成更小，更合理的部分，以便更好地进行处理和分析，有助于提高检索和生成的准确性。
- ✓ **数据库打标准确性**：确保数据库中的信息准确无误、通过标准化过程提高数据的一致性和可靠性。

2. 模型选择与优化

- ✓ **选择更大的模型**：现有模型的生成效果不理想，可以使用更大的预训练模型，带来更好的生成质量和上下文理解能力。
- ✓ **使用不同的embedding模型**：根据实际应用场景和数据特点，使用不同的embedding模型，以找到最适合的模型来提高检索精度。
- ✓ **调整chunk-size***：重新评估和调整chunk的大小，以找到信息量和检索效率之间的最佳平衡点。

3. 模型训练与验证
对于新的模型和参数设置，需要进行训练和验证，以评估其性能。

4. 计算资源管理
如果需要更大的模型或更复杂的embedding模型，需要更多的计算资源。这涉及到模型训练和优化阶段的资源分配和管理。

5. 部署与监控
优化后的模型需要部署到实际环境中，并持续监控其效果，以便进行进一步的调整。

*备注：在RAG系统中，"chunk-size"是指将文本或数据分成较小的块，以便更高效地处理和检索。调整chunk-size有助于在信息量和检索效率之间找到最佳平衡。较小的块可以提高检索准确性，但可能丢失上下文；较大的块保留更多上下文，但可能增加冗余。通常，chunk大小在100到300词之间，需根据应用场景调整。通过调整chunk-size参数，可以优化RAG模型的检索效果，提高效率和准确性。

数据来源：甲子光年智库梳理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

甲子光年 JAZZYEAR

企业级的AI代码平台及开发解决方案建立在企业个性化数据的运营完善能力之上

- 由于RAG高度依赖其知识库的信息质量，因此企业级的代码知识库的完善是保证其技术效果的关键举措，具体如代码应该怎么处理，文档应该怎么处理，代码过来要进行过滤、清洗、结构化等等细节的工作是非常值得关注和尊重的，可保证代码知识库的不断完善，如下示例
  - ✓ 保证代码知识库数据的质量：建立确保知识库中信息的数据质量、相关度。
  - ✓ 动态更新代码知识库的数据：数据动态变化，保持知识库的时效性可确保RAG的响应符合当下需求。
  - ✓ 高效组织数据：通过使用索引、分类或标记来优化代码知识库的检索过程， 以合乎逻辑且易于检索的方式组织数据是关键。

图：RAG的企业级代码知识库痛点及解决方案示例

RAG知识库痛点	解决方法示例 1	解决方法示例 2	解决方法示例 3
建立者孤立地建立和发展	将用户查询与知识库内容进行比较	查询驱动的知识库补全	持续知识库维护
信息滞后、错误	数据工程功能按不确定性对孤立问题排序	按主题对真实用户问题进行分组	通过知识库维护改进
不合逻辑	较高的不确定性分数表示知识库内容存在差距	查看哪些缺失主题有最多的用户查询	
数据组织形式不易被检索	• 孤立问题是指通过将用户查询与知识库内容进行比较，识别出知识库中未能充分覆盖或存在内容差距的用户问题。	用户问题组准确地显示需要涵盖的内容	
不以用户为中心		同类比较，查询驱动的检查	

数据来源：integrail.ai；甲子光年智库梳理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

甲子光年 JAZZYEAR

Agent是推动工作流程自动化的关键组件，提高整体开发效能

- AI代码平台中AI Agent的核心逻辑：通过将经过**特定训练的大模型**作为“**大脑**”，利用其**推理**和**规划**能力，并**调用外部工具**来完成复杂任务，从而优化代码开发过程。

图：Agent 的特征和架构解析

核心特征

AI代码平台的AI Agent ＝ 特定训练的大模型 ✖ （代码规划+知识记忆+工具集成+自动执行）

在特定代码任务中优化和微调的AI核心

代码规划：智能分析和设计代码结构，优化开发流程

自主执行：自主完成代码生成和测试等任务，实现自动化操作

知识记忆：存储和检索相关信息，支持连续性和上下文理解

工具集成：连接和使用外部软件工具，提高开发效率

架构解析

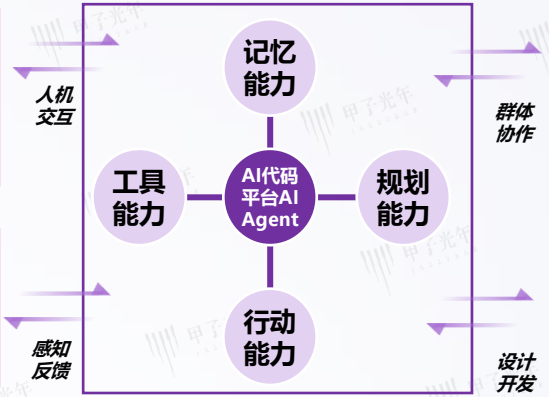
Agent基于LLM的组件，和交互两个层面

人类用户（指令者）

用户通过平台界面的输入框、按钮等，使用特定的命令、指令等方式与其交互，将自己的需求传递给AI Agent，是引导其生成定制化代码解决方案的关键，直接影响代码功能和输出结果。

外界环境（影响者）

AI代码平台中的Agent能够感知并响应其所处环境（虚拟或物理世界）的变化，与环境的互动进一步塑造代码生成的输出，确保代码能够适应并解决实际问题。



Agents（执行者）

通过多个Agent协作机制，各Agent能够共享信息、资源和任务结果，实现复杂问题的高效解决，从而构建起强大的群体智能体系。

系统开发者（创造者）

精心设计和编程，赋予AI代码平台的Agent特定的功能和组件，从而确保它们能够高效地生成符合用户需求的代码。

数据来源：公开资料，专家访谈，甲子光年智库整理，2024.10

技术支撑：基于项目开发“Know-How”的“AI+”技术

甲子光年 JAZZYEAR

通过模拟软件开发角色，实现高效协作，确保从需求分析到代码上线的流程顺畅

- 系统模拟团队中的不同软件开发角色（如执行官、产品官、程序员、测试员等）来协同工作完成项目，平台依赖**事件驱动架构、标准化接口、沙盒环境、协调代理**和**CI/CD**等技术实现协作。

对应角色	作用	工作形式/原理
产品经理Agent	与用户沟通（交互），分析并转化用户需求为功能需求	通过自然语言处理与用户对话，提取关键信息，生成需求文档或任务列表
项目经理Agent	促进不同代理之间的协作，确保信息流通和任务协调	通过多代理委派机制，协调各个代理的工作
开发工程师Agent	将复杂功能需求分解为更小的、可管理的代码模块或任务	利用机器学习模型分析需求，自动生成代码结构和模块划分
测试与支持 Agent	负责自动化测试，确保代码正确性和稳定性 监控系统状态，收集用户反馈	生成测试用例，执行单元测试和集成测试，实时监控应用性能，收集用户使用数据，分析测试结果并反馈

用户需求

产品经理Agent	项目经理Agent	工程师Agent	测试Agent
细化需求	细分任务	生成代码	生成单测
用户故事	任务优先级	代码评审	单测执行
需求优先级	任务约束	修复缺陷	收集缺陷
…	…	…	…

多代理协同工作流程图

开始
- 开始节点

需求分析
- 收集用户需求
- 确定功能模块

任务分解
- 将需求分解为具体的子任务
- 确定每个子任务的负责人代理

代理任务分配
- 将子任务分配给不同的代理
- 每个代理根据其特长执行任务

代理间通信
- 代理之间共享信息和进度
- 发送请求和接收反馈

协同所需要的技术

事件驱动架构：通过事件流实现代理之间的通信和协调

标准化接口：定义清晰的API和接口，便于数据交换

沙盒环境：提供安全的执行环境，支持独立和协作任务

协调代理：管理任务分配、进度跟踪和资源管理

持续集成/持续部署：自动化集成和部署流程，提高开发效率

代码开发
- 各代理独立开发代码
- 进行单元测试和调试

集成与测试
- 将各代理的代码集成到主代码库
- 进行系统测试和集成测试

结果评估
- 评估集成后的系统性能
- 收集反馈信息

优化与迭代
- 根据反馈进行代码优化
- 迭代开发，重复以上步骤

结束
- 结束节点

数据来源：deepsense.ai等，甲子光年智库整理，2024.10

目录

CONTENTS

甲子光年 JAZZYEAR

Part 01 机遇：AIGC引发的数字生产变化

Part 02 价值：重新构建代码开发的应用范式

Part 03 落地：智能时代AI+代码的先行者

Part 04 展望：人工智能普惠时代的期待

AI代码平台领域成为生成式AI的重要落地领域，不同类型企业各显身手

- 中国当下的AI+代码平台市场已经迎来丰富的参与者，这些企业依据自身技术或行业know-how迅速切入市场，通过先手占据更好的生态占位；并且越来越多的企业正在进行产品打磨与场景探索。

AI代码平台图谱V1.0

AI 编码助手类

业务*AI代码型

众安保险、DevPilot、Thundersoft、蚂蚁集团 ANT GROUP、CODEFUSE、……

AI代码创业型

AIXcoder、Fitten、……

AI代码社区型

InsCode Beta、……

智能体*开发类

面壁智能 MODELBEST、……

AI模型*代码型

百度智能云、文心快码 Baidu Comate、KUNLUN、SKYCODE、商汤 sensetime、代码小浣熊 CoRaccoon、4Paradigm、4Paradigm AIGS CodeX、讯飞星火、星火飞码 IFLYCODE、京东 智能编码 JoyCoder、智谱清言、CodeGeeX、幻方量化、DeepSeek 智能代码助手、阿里云、通义灵码、腾讯云、腾讯云 AI 代码助手、CodeArts Snap、火山引擎、豆包MarsCode、……

低/无代码+AI

网易数帆、致远互联 SEEYON.com、ActionSoft 炎黄盈动、用友 yonyou、Landray 蓝凌 数智化办公专家、金现代 JINXIANDAI、Kingdee、……

图形化编程语言（可包括IDE）+AI

VX.cn、……

通用模型

算力、数据及其他基础支持

图谱1.0版数据截至2024年10月，顺序不分先后

注：随着产品逐步落地，未来将有更多类型企业切入AI代码平台领域，以下为企业类型说明

1）AI模型*代码型：一般具备通用大模型能力，提供基于生成式AI的编程产品/插件
2）业务*AI代码型：基于自身业务及开发能力，提供基于生成式AI的IT开发平台
3）AI代码创业型：基于代码领域的垂类模型及产品能力，提供相应的AI代码平台服务
4）代码社区型：基于社区代码的数据积累，提供相应产品
5）智能体*开发：利用多智能体能力，提供代码开发服务
6）低/无代码+AI：在自身低无代码基础上进行AIGC技术的结合
7）图形化编程语言（可包括IDE）+AI：在图形化编程语言及开发平台及生态基础上进行AI技术的结合

数据来源：公开资料，甲子光年智库整理，2024.10

值得关注的互联网大厂产品及服务动态——百度智能云【文心快码】

甲子光年 JAZZYEAR

➢ 文心快码（Baidu Comate）是基于文心大模型，结合百度积累多年的编程现场大数据和外部优秀开源数据，打造的新一代编码辅助工具。拥有代码智能、场景丰富、创造价值、广泛应用等多重产品优势，可实现“帮你想、帮你写、帮你改”的场景应用形态。提升编码效率，释放“十倍”软件生产力。

Baidu 百度 文心快码 Baidu Comate

产品功能

“帮你写”		“帮你想”		“帮你改”	
调研和设计		**编写代码**		**测试和发布**	
私域知识增强	需求澄清	智能代码补全	智能生成描述	智能评审	智能安全监测
研发规范	架构解读	描述生成代码	自动发现漏洞	智能Debug	智能漏洞修复
安全规范	任务分解	生成安全代码	自动修复漏洞	智能缺陷修复	发布流程管控
	专家模式问答	API生成代码	单测智能体	CI 错误诊断	

优势亮点

更懂研发知识 开发速度快
构建研发领域知识体系，满足多场景诉求，实现高效交付

更懂研发全流程 业务迭代快
文心快码无缝集成研发各环节，加速研发全流程提效

更懂行业客户 企业落地快
直击行业难点，提供最佳落地实践，助力客户提效

支持私有化、混合云部署
让大模型以更安全的方式学习企业业务代码和流程，更契合行业与业务的需求

全方位支持多种主流IDE与编程语言
支持 100 多种主流语言和 VSCode、JetBrains 全系列等主流 IDE

百度每天新增的代码中，有30%由文心快码生成，整体采纳率达46%，这一提效工具的应用令工程师整体提效达到12%，显著提升了单位时间内的代码提交数量和业务迭代速度。更重要的是，智能编码提效工具使工程师得以从繁杂、重复的工作中释放出来，专注于更有价值和创造性的工作。

客户示例 GEELY 顺丰科技 SOFTSTONE 喜马拉雅 方正证券 Nexchip MINISO 上海三菱电梯 华农保险 CHIC 奈雪的茶 ……

数据来源：百度文心快码，甲子光年智库整理，2024.10

国内重点厂商产品及服务能力分析——众安保险DevPilot

甲子光年 JAZZYEAR

自主研发AIGC平台结合业务经验沉淀，提供企业级高质量AI+代码开发能力

- 众安保险是中国首家互联网保险公司。作为国内首个核心系统搭建在云上的金融机构，众安基于云服务平台搭建开放、灵活、可扩展的核心系统，与此同时，众安不断开拓人工智能、区块链、大数据等前沿技术，并深度应用于产品研发，提升体验，改善经营效率。
- DevPilot 是众安保险技术团队开发的AI代码助手，结合私有化部署的或是通用三方代码模型，带来一套轻量高效的全栈代码助手开源解决方案，使 AI 成为开发者工具的标准组成部分。

众安保险

DevPilot

众安保险 通过实践不断解锁AI驱动下AI + 保险的新场景

众安自主研发的AIGC中台**“众有灵犀”**，为AI应用提供安全可靠的大模型能力，提示工程、知识工程等中台能力为AI场景研发更加敏捷，在此之上落地实现了智能产研、营销、运营、客服等多场景应用。

众安保险凭借海量的保险数据以及丰富的保险知识，在大模型上进行深度的场景应用，通过AI中台化的能力持续释放生成式AI能力。

“众有灵犀”为DevPilot提供了一个功能丰富的平台，作为DevPilot的中台提供了大模型集成、提示编排、检索增强生成等能力，**使得DevPilot在各种运行代码生成的场景需求都能够在这个平台上得到实现和构建。**

作为一款强大的中台系统，**众有灵犀能够集成多种大型语言模型**，并且对外提供统一的接口，从而有效地屏蔽了不同模型之间的差异性和复杂性。

众有灵犀还为DevPilot的使用提供一系列针对大语言模型安全性的策略和措施，其中包括了对敏感数据的检测机制等，以及防止生成具有政治敏感性内容及敏感代码外泄的技术手段，以此来保障平台上的内容生产和交互活动能够在符合安全规范的前提下进行。

DevPilot 基于 IDE 的 AI代码助手

DevPilot的定位是“代码辅助生成助手”，可参与开发者需求开发的全流程，致力于解决开发态的问题。从自动化日常开发任务到提供富有洞察力的代码建议，让开发人员能够更智能、更快速、更少错误地进行编码。

DevPilot通过结合私有化部署方案或是利用通用的代码模型，为开发者提供了一套轻量且高效的全栈代码助手开源解决方案。让AI技术成为开发者工具的标准组成部分，显著提升开发效率。

积极构建开源社区：众安开源战略的核心聚焦于构建开发者与用户紧密联结的社区生态，旨在通过协作打造共赢共创的行业技术环境。

外部开发者积极参与开源社区的构建

开源社区的反馈可以做到当天响应及修复

数据来源：众安保险DevPilot，甲子光年智库整理，2024.10

国内重点厂商产品及服务能力分析——众安保险DevPilot （产品特点1/2）

甲子光年 JAZZYEAR

- DevPilot通过结合私有化部署方案或是利用通用的代码模型，为开发者提供了一套轻量且高效的全栈代码助手开源解决方案。DevPilot的设计初衷是让AI技术成为开发者工具的标准组成部分，不仅能够帮助开发者自动生成代码片段或模块，还能通过智能化的代码补全、重构建议等功能，显著提升开发效率。

DevPilot

DevPilot的产品功能一览：丰富功能充分满足企业级开发需求

DevPilot独特的产品设计：结合开发经验，打磨产品细节

代码生成

DevPilot可以基于仓库代码相似逻辑，按需求生成新的功能代码，**尤其可基于企业仓库代码生成最佳编码方案，提高编码效率。**用户可用这款AI助手与机器进行对话生成代码。

代码补全

在补全场景，**DevPilot从单纯的纯模型补全演进到基于提示词的补全，再到补全后接受交互行为的处理，站在开发者的角度不断的打磨补全的效果，尽量减少代码修改率**，避免出现网传的“助手代码生成越多，错误越多，修改越多”的问题。

DevPilot着重提升上下文逻辑的理解能力，提升其企业级开发项目中的代码采纳率。

单元测试

DevPilot不仅可以帮用户对大篇幅代码做Code Review，检测代码性能，**还能快速生成单测案例，针对类、方法级提供多种单元测试生成能力**，也可以用提示词生成符合特定框架要求的单元测试。

该场景下，DevPilot不断的丰富要生成单测的内容上下文，通过提示词让模型更加专注在逻辑分支上，识别所有可能的执行路径，确保分支覆盖率。

……

数据来源：众安保险DevPilot，甲子光年智库整理，2024.10

国内重点厂商产品及服务能力分析——众安保险DevPilot （产品特点2/2）

甲子光年 JAZZYEAR

DevPilot产品功能亮点

01

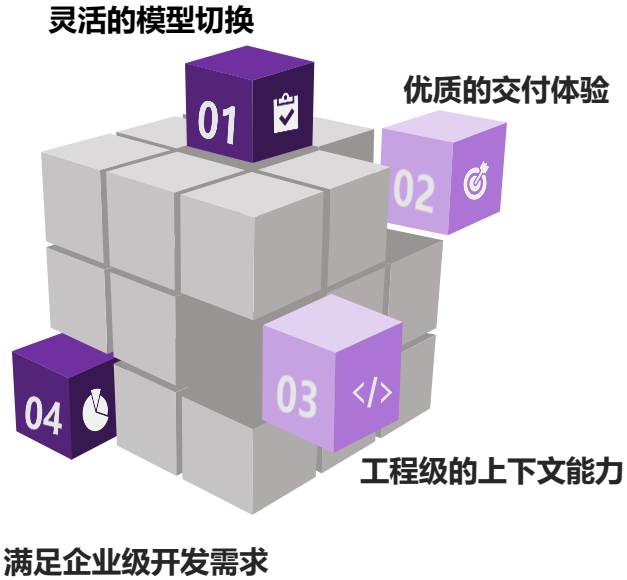
灵活的模型切换

模型可切换DevPilot允许开发者或企业切换合适的大模型，可使用本地模型或三方通用模型，这就使得DevPilot能够更快跟进先进模型，并且可以符合企业大模型需求。

02

优质的交互体验

IDE原生DevPilot基于VSCode、Jetbrain IDEA 等主流IDE构建，使用IDE原生交互，不改变开发者的编程习惯。同时提供chat、diff视图，以及代码插入时自动寻找目标文件等的细节交互，为开发者带来更加丝滑的编程体验。



03

工程级的上下文能力

相较于一些同类产品生成代码时只能感知当前文件或者打开文件的代码，DevPilot广泛使用工程文件索引以及检索增强生成（RAG）技术，在生成代码时可获得需求相关的工程范围甚至是企业仓库里的代码作为参考片段，使生成的代码更加符合工程需要，有效提升代码采纳率。

04

符合企业级开发需求

DevPilot的产品能力效果来自其独特定位：专注于企业级应用领域。 DevPilot的的核心理念在于实现与企业内部复杂业务逻辑的高度融合，它不仅仅是一个简单的代码生成器，而是致力于根据企业的特定业务需求来直接生成相应的代码片段或模块。

数据来源：众安保险DevPilot，甲子光年智库整理，2024.10

国内重点厂商产品及服务能力分析——众安保险DevPilot

甲子光年 JAZZYEAR

项目实践能力解析：致力于解决企业在软件开发过程中所面临的特定挑战

定位企业级开发

符合企业架构标准

DevPilot将企业内部常用的通用技术组件与业务组件的访问代码，以及企业特有的应用框架代码进行嵌入式整合。

通过这种方式，在代码生成的时刻，这些预嵌入的代码片段会被智能召回并加以利用，确保生成的代码能够更好地遵循企业的架构标准和技术规范。

以企业为中心

DevPilot的设计目的之一是为了更好地服务于企业的软件开发流程。

它能够通过深入理解企业特有的业务流程和技术框架，识别并适应企业已有的技术栈和业务规则，从而提供更加精准和实用的代码支持，帮助企业维护一致性的代码标准，降低错误率，增强企业软件开发工程的可靠性和稳定性。

推动数字化转型

DevPilot 能够使AI技术无缝地集成到企业的日常软件开发流程中，从而真正意义上实现从前沿技术向实际生产力的转变。

这意味着DevPilot成为了一个能够为其带来切实利益的解决方案，**它可以推动企业自身的数字化转型过程，提升市场竞争力，同时也能更好地应对不断变化的技术挑战，**为未来的持续发展奠定坚实的基础。

项目效果

DevPilot在软件项目开发过程中实现难点、痛点的解决

编码效率
- 重复性任务多，导致开发周期延长

代码质量
- 代码质量参差不齐，难以保证产品稳定性和可维护性

团队协作
- 团队协作中沟通成本高，版本控制复杂，影响项目进度

DevPilot**显著提升开发效率与产品质量，优化团队协作，加速市场响应**

效果实现

研发团队整体研发速度提升 20%

15% **AI代码生成在整体项目的贡献度**

数据来源：众安保险DevPilot，甲子光年智库整理，2024.10

重点厂商产品及服务能力分析——aiXcoder

甲子光年 JAZZYEAR

致力于将前沿人工智能技术应用于软件工程，聚焦代码大模型的私有化落地和应用

- aiXcoder孵化自北京大学软件工程研究所，创始团队均来自北京大学，是全球最早将深度学习技术应用于代码生成与代码理解领域的团队，也是最早将深度学习应用在编程产品的团队。
- 公司专注于AIGC for Code领域，致力于将前沿人工智能技术应用于软件工程，聚焦代码大模型的私有化落地和应用，与企业领域知识融合，助力企业实现智能化研发。
- aiXcoder为企业提供完备的基于代码大模型的智能化软件开发解决方案，包含私有化部署、企业个性化代码大模型及定制化开发等服务，现已在金融、军工、航空航天、通信、高科技等行业头部企业部署实施，成功实现了代码大模型的多场景应用和实际效果验证，获得广泛行业认可。

aiXcoder

企业智能化软件开发提供商

紧密结合具体业务，持续深入各垂直细分领域

应用场景

金融	军工	航空航天	通信
交通	科研	智能制造	IT
……			

企业服务

私有化部署	领域化大模型落地	定制化开发	咨询服务

产品

aiXcoder系列大模型	智能化软件开发系统	智能化测试生成系统	智能知识库系统	智能化软件开发一体机

核心技术

大模型训练及微调	软件工程	Agent	RAG

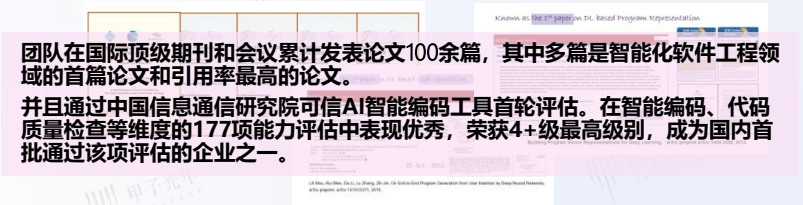
aiXcoder将技术转化为实际应用落地，让代码大模型在智能编程领域广泛实用，为此，aiXcoder **推出个人/企业订阅模式、私有化解决方案、深度咨询服务三种商业模式，**更贴合实际软件开发任务，与企业应用场景深度融合，更适合部署应用。

持续深耕AIGC for Code，多次版本不断迭代，助力企业级智能软件开发

- **2018 aiXcoder1.0**：实现在线版代码智能补全与搜索
- **2022 aiXcoder XL 发布**：国内首个支持方法级代码生成的百亿级参数量代码大模型（13B参数量）
- **2024 aiXcoder7B 发布并开源**：7B模型性能达到百亿级参数量代码大模型SOTA，降低使用门槛易部署
- **即将推出 aiXcoder76B**

*以上版本为部分版本说明

- aiXcoder于2018年开始深耕AIGC for Code领域，2023年已获得A+轮融资
- 2024年同时发布智能化软件研发一体机，基于Agent的aiXcoder智能化软件开发系统2.0。



团队在国际顶级期刊和会议累计发表论文100余篇，其中多篇是智能化软件工程领域的首篇论文和引用率最高的论文。

并且通过中国信息通信研究院可信AI智能编码工具首轮评估。在智能编码、代码质量检查等维度的177项能力评估中表现优秀，荣获4+级最高级别，成为国内首批通过该项评估的企业之一。

数据来源：aiXcoder，甲子光年智库整理，2024.10

重点厂商产品及服务能力分析——aiXcoder

甲子光年 JIAZIYEAR

➢ aiXcoder主要通过大模型、Agent技术与软件工具相结合的方式，实现软件开发全流程的智能化开发辅助，提升开发效率和代码质量。同时通过大模型微调、领域知识库、RAG等技术，解决企业领域化知识与大模型相融合的问题，为每个企业打造专属软件工程大模型。

高质量基座代码大模型的构建

结合代码结构化特征和软件工程的方法与工具，从训练数据处理、训练任务构造再到模型结构优化等方面，强化大语言模型在软件开发场景下的性能与效果。

多功能代码大模型构建

在基座代码大模型基础上，通过指令微调，让大模型具有更多功能，如单元测试用例自动生成、代码解释、研发问答等。

基于Agent的智能化软件开发系统

通过大模型与软件工具的调用相结合，调用外挂知识，以多角色扮演的方式，实现项目级代码生成、项目研发问答、issue自动处理等更复杂的真实软件开发任务。

企业个性化代码大模型构建

结合企业领域知识，对代码大模型进行个性化训练和微调，帮助企业实现更高质量的代码生成以及研发全流程的能力提升。

关键技术点

代码大模型的构建

结合软件工程工具，构建高质量训练数据

- 通过敏感信息去除、删除无效代码、自动生成代码去除等多重筛选，剔除163种Bug和197种常见代码缺陷。
- 注入正确编程知识，行成独有的高质量训练数据集。

结合代码结构化特征的训练方法

- 结合程序语言特性，aiXcoder在模型训练时融入代码抽象语法树结构信息，增强模型对语义和逻辑的理解能力
- 采用结构化Span方式，使生成的代码更准确，更符合代码结构。
- 同时采用了FIM（Fill In the Middle）和跨文件的训练方式，更贴近真实开发场景。

基于大模型的个性化训练

通过个性化训练，aiXcoder可帮助企业实现更高质量的代码生成以及研发全流程的能力提升。

经过个性化训练的模型可以学习到行业或企业内部的代码逻辑、代码风格和业务知识等领域知识，提高代码生成的准确率。

- 自动化构建企业专属评测集
- 提供定制化数据治理工具
- 涵盖模型预训练、微调、RAG、Agent等技术的一整套解决方案
- 根据企业已有代码和文档、计算资源，提供定制化解决方案
- 在保证主模型效果的情况下进行增量训练，大大降低过拟合风险

基于Agent的新一代智能软件开发

- 支持、适配生成式AI技术与大语言模型的应用发展，基于大模型的智能化软件开发具有**行业领域知识缺乏、软件开发专业技能欠缺、复杂问题难以处理、无法进行多角色协作等局限性。**
- 基于Agent的智能化软件开发系统，通过大模型与软件工具的调用相结合，调用外挂知识，以多角色扮演的方式，实现**项目级代码生成、项目研发问答、issue自动处理等更复杂的真实软件开发任务。**
- 在基于大模型的Agent框架中，开发者与大模型围绕更复杂的项目级代码生成展开持续互动和工具调用，**人机协同，推动软件产品的快速迭代与持续优化。**

数据来源：aiXcoder，甲子光年智库整理，2024.10

重点厂商产品及服务能力分析——aiXcoder

甲子光年 JAZZYEAR

信华信 × aiXcoder

客户背景

信华信是一家面向全球客户提供领先的应用软件产品、解决方案和数智技术服务的集团公司，面向产业、金融、政府公共等领域客户，提供咨询、软件开发、运行维护、云基础设施等一站式服务及行业、业务和技术解决方案。

项目过程

智能化软件开发面临诸多难点，需要通过多方努力，才能推动智能化软件开发的健康发展。

项目难点

数据隐私与安全	算法可解释	技术标准缺乏
高性能计算资源	人才短缺	开发成本高
系统集成兼容性	用户接受度	认知及伦理影响

搭建思路

“信华信软件工程大模型”基于信华信近30年软件工程领域沉淀，联合aiXcoder等AI生态合作伙伴，打造符合软件工程标准流程的五大应用域。

- AI设计助手
- AI编码助手
- AI测试助手
- AI安全助手
- AI知识助手

项目效果

- 基于自研模型工具链，“信华信软件工程大模型”支持企业引入行业数据，进行领域微调，为企业构建符合自身需求的软件工程大模型。
- 目前信华信软件工程大模型已在应急领域实现共创应用，带动该领域软件实践项目缺陷数量降低33.35%，通用业务逻辑代码采纳率平均47%，开发效率提升8.38%。

航天某研究所 × aiXcoder

航天某研究所和aiXcoder团队依托共同设计并设施IPRCoder产品研制方案。

该产品为首个先进的人工智能技术与软件工程相结合、自动将控制算法生成程序代码的软件开发产品。

该产品可基于规则解析并结合领域知识将数学公式和逻辑表达等内容智能生成代码；并在此基础上，通过大模型进一步提升工具的泛化能力和生成能力，并对生成的代码进行优化，高度确保生成代码的正确性和可靠性

某国有证券企业 × aiXcoder

某国有证券企业采用aiXcoder的大模型解决方案，即将深度训练过的代码大模型在**本地环境中进行私有化部署。这种选择确保了即使在常规的算力支持下，也能够满足公司日常的编码需求，避免了因硬件门槛过高而带来的挑战。同时为企业的数据安全和合规要求提供了充分保证。**

该企业采用个性化训练的智能编程模型，公司采用个性化训练。在确保成本效益的基础上，进一步提升公司代码大模型的推理能力，生成与业务逻辑和公司规范更为匹配的代码，助力提高其研发团队效率。

数据来源：aiXcoder，甲子光年智库整理，2024.10

海外AI代码开发平台/产品分布图谱

海外AI代码平台图谱V1.0

简化工程工作流程 Enhangcing Engineer Workflows

代码生成 Writing code

builder.io
CodeComplete
codeium
CodeStory
codium.ai
CURSOR
replit
Sourcegraph
tabnine
supermaven
GitHub Copilot

单元测试 Testing code

BlinqIO
codeium
codium.ai
Meticulous
Momentic
QA WOLF
ventrilo.Ai

代码审查 Code review

Bito
codeium
CodeRabbit
codium.ai
KORBIT
planar
tunk

代码运维 Maintenance

Dosu
gitar
grit
SECOND
Sweep AI
Detail

生成文档 Documentatin

komment
mutable.ai

智能搜索 Code search

codeium
CURSOR
codium.ai
Sourcegraph

重构工作流 Replacing Engineer Workflows

智能体 AI Agents

Codegen
SWE-agent
Cognition
AutoCodeRover
FACTORY
Tusk
OpenDevin
GitHub Copilot

特定类型的任务或领域定制的模型层 Code-Specific Model Layer

codeium
poolside
Magic
Augment code

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——Cursor

产品简介：

Cursor是由Anysphere开发的革命性AI驱动集成开发环境（IDE），它通过集成AI工具实时协助开发人员，旨在使编程更快、更有趣、更具创新性。

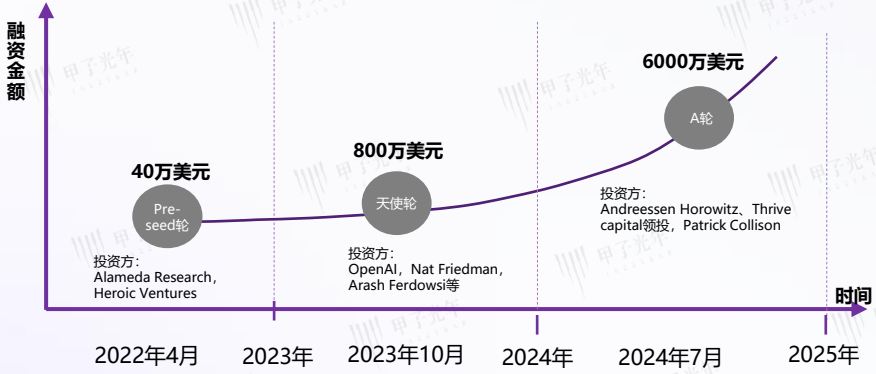
自2023年初发布以来，Cursor始终致力于通过持续更新和功能迭代优化用户体验和性能。简化了代码重构和大规模代码修改等复杂任务，显著提高了开发效率。

公司及团队简介：

Anysphere是一家AI应用研究开发商，专注于高效人机交互系统和“AI原生”软件开发环境的构建，提供人工智能和机器学习研究。其目标是通过有关机器学习、系统和产品设计的基础工作,来加快软件工程的速度。联合创始人兼首席执行官 Michael Truell 设想了一个未来：**开发人员可以以前所未有的速度构建软件。**联合创始人 Sualeh Asif、Arvid Lunnemark 和 Aman Sanger 也拥有同样的愿景。

Anysphere认为，通过结合人工智能和人类的智慧，可以创造出比最好的纯人工智能系统更聪明、更精通设计的“混合工程师”。这些混合工程师将能够毫不费力地控制他们的代码库，并且不会出现低熵击键（无效或不必要的按键操作）。

融资历程



商业模式 ——“免费增值”模式

- **吸引用户：基础功能免费**
- **订阅创收：高级功能每月付费订阅**，价格 40 美元
- **合作与集成商业模式**：Cursor 在 三星、Midjourney、Perplexity、Shopify 和 OpenAI 等多家科技公司得到应用，通过与其他平台和工具的兼容性，提升市场覆盖率和实用性
- **灵活与专注**：专注**微观目标用户**，**更注重个人和团队体验**，而非企业体验。年度 ARR 收入超过 1000 万美元，公司估值达 4 亿美元

• 年度 ARR（Annual Recurring Revenue）指公司在一年内通过订阅或合同获得的经常性收入。反映了业务的稳定性和增长潜力。

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——Cursor （产品特点1/2）

- **Cursor的主要特点：**基于Visual Studio Code 构建提高编码速度和质量，AI代码补全（编写完整函数）、发现并修复错误、自然语言命令、文档帮助和重构辅助（清理并改进现有代码）

图： Cursor的产品功能概览

直观易用 上手轻松

- 熟悉的界面**简化应用过渡**
- **自然语言输入**使用户易于访问
- **逐步采用**（逐步使用新系统，减少学习压力，更舒适地适应）

无缝衔接 融入习惯

- **与常规工作流程集成**（可无缝融入日常工作方式）
- 使用聊天功能获取情境**感知答案**
- 结合人工智能和人类专业知识
- **探索语言支持**（擅长 JavaScript、Python 和 TypeScript）
- 对敏感项目使用隐私模式

数据管理高效 与最小化传输

- **隐私模式**可**防止代码存储在服务器上**
- 使用 AI 功能时仅需发送 **100-300 行**代码
- 发送的代码**未静态存储**

遵守隐私保 护规则

- **符合 GDPR**
- **使用 SSL 加密**
- **遵循美国数据传输的标准合同条款**

隐私保护模式	具体说明
SOC 2 认证	数据安全标准
隐私模式	防止代码存储在服务器上
无存储模式	无需付款或 API 密钥即可激活
本地/幽灵模式	确保不存储代码

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——Cursor （产品特点2/2）

甲子光年 JAZZYEAR

图： Cursor的开发体验特点分析

无缝集成现有开发环境

增强兼容性，确保高效工作

- 与 Git 和其他版本控制系统兼容
- 与许多VS Code 扩展兼容
- 通过 SSH 连接到远程服务器时保持完整的 AI 功能

先进的代码分析

深入理解项目，简化代码操作

- 上下文理解：分析整个代码库以获得深入的项目洞察
- 自然语言处理：使用 Ctrl+K 以简单的英语来描述代码变化
- 智能代码完成：建议复杂的多行代码结构和重构

高级用法全新体验

提升编程效率，支持审查调试

- 使用 Cursor 进行代码审查
- 向 AI 寻求调试帮助
- 使用自然语言提示生成代码片段
- 利用 Copilot++ 进行预测编辑

编辑功能

智能实时编辑建议，动态适应提升交互效率

- 预测用户的下一次编辑
- 建议跨多行进行回溯更改
- 根据用户最近变化进行调整
- 持续开启（始终处于激活状态，不需用户手动启动）

表： Cursor的产品演示示例-甲子光年演示示例

AI代码完成

Cursor AI 不仅能完成句子，还能完成代码。它会提供建议，可以节省时间、减少错误。例如在下面的代码中建议：

- 自动完成代码：函数结构和 HTTP 请求的标准流程由 AI 自动补全；
- 代码建议：AI 提供合理的函数命名和参数使用，确保代码简洁明了；
- 错误减少：通过建议使用raise_for_status()方法来检测请求错误，提升代码的可靠性。

```
import requests

def get_api_data(url, params=None):
    response = requests.get(url, params=params)
    response.raise_for_status()
    return response.json()
```

智能代码改进

Cursor AI 可以重构和优化用户代码，并建议：

- 将循环转换为列表推导
- 简化复杂条件
- 删除未使用的变量

上下文帮助

Cursor在用户编码时提供即时解释和文档

普通语言命令

以自然语言描述生成代码。输入“创建一个函数来计算最多 n 项的斐波那契数列”，可生成：

```
def fibonacci(n):
    sequence = [0, 1]
    while len(sequence) < n:
        sequence.append(sequence[-1] + sequence[-2])
    return sequence[:n]
```

调试帮助

Cursor AI 作为调试伙伴介入，突出显示问题并建议修复

表： Cursor的产品特点分析

功能	操作
编辑现有代码	选择代码，点击“编辑”，描述修改
生成新代码	键入 Ctrl K 而不进行选择
自动完成	始终开启，预测下一次编辑
多行建议	提供跨多行编辑
自适应	从最近的变化中学习

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——GitHub Copilot

甲子光年 JAZZYEAR

产品介绍

- GitHub Copilot是由微软联合OpenAI、GitHub 推出的AI代码补全工具。
- 基于OpenAI 创建的人工智能模型Codex生成。Codex接受了一系列英语语言、GitHub 公共存储库及其他公开可用源代码的训练。这些训练包括来自 5400 万个公共 GitHub 存储库的 159 GB Python 代码的过滤数据集。
- 在使用HumanEval评测标准进行的对比中，Codex的性能优于同类模型。

GitHub Copilot

产品概览

产品版本

Copilot：采用OpenAI 的 Codex 模型。是一个通过自动代码补全来帮助程序员们编写代码的AI编程工具。

模型升级 →

Copilot X：采用OpenAI 的 GPT-4 模型，集成Copilot Chat扩展，新增以代码为中心的聊天和语音模式。

企业版 →

Copilot Enterprise：根据企业的知识和代码库定制，将代码辅助工作融入整个软件开发生命周期。

平台

Copilot Workspacet：是以任务为中心的集成开发环境，让开发人员控制开发流程。使用自然语言把想法变成编码形成软件。

扩展

Copilot Extensions：作为扩展，帮助开发者集成专有内容、简化工作流程，支持将扩展工具和服务以自然语言构建和部署到云中。

产品亮点

代码生成速度

基于 GitHub的海量公开代码库，快速生成样板代码、常见函数和 API 调用，避免开发者重复劳动。

预训练知识库

基于主流编程语言和框架，能够快速提供通用场景相关的代码示例，方便初学者学习新语言或框架。

集成GitHub

基于多种开发工具（如 Visual Studio）集成，实现代码补全和内联建议，为开发者提供了深度且流畅的编程体验。

实践成效

应用效果

- **活跃度 Activity**：信息检索时长减少 90%
- **生产率 Productivity**：建议代码采纳率 88%
- **效率 Efficiency**：增加代码推进速度 84%
- **满意度 Satisfaction**：增加代码撰写满意度 90%

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——GitHub Copilot 产品更新

甲子光年 JAZZYEAR

01大模型能力升级

特点

集成最先进的o1模型：o1是OpenAI最新的大模型，具备先进的推理和编程能力，在数据分析、编程和数学等领域表现优异。

功能

- 支持在Copilot Chat切换模型：用户可以在 VS Code 中切换模型（o1-preview 、o1-mini 、 GPT-4o）。
- 应对复杂编码挑战：在对话期间在模型之间切换，快速解释 API，生成样板代码，设计复杂算法，分析逻辑错误。

02支持协作者上下文检索

特点

支持在github.com 中进行故障排除和与团队合作。

功能

- 服务软件开发生命周期：支持整个代码库、协作者之间的对话和工作流。

03支持多设备使用

特点

支持随时获取见解：支持在 GitHub Mobile 中与 Copilot 讨论开发者私有存储库，快速了解代码、问题、拉取请求等的更新。

功能

- 支持多个设备：可以在所有Copilot 用户的 IDE、Visual Studio Code、浏览器和移动设备上使用。

04支持代码引用

特点

集成Azure AI Content Safety ： GitHub 与 Microsoft Azure 合作，在 Azure AI Content Safety 上提供代码引用 API。

功能

- 支持多个渠道：在 GitHub Copilot Chat 和 GitHub Copilot 代码补全中使用代码引用。
- 迅速定位代码：索引数十亿个文件，仅用 10-20 毫秒，找到特定的匹配项

数据来源：公开资料，甲子光年智库整理，2024.10

国外重点厂商产品及服务能力分析——GitHub Copilot

甲子光年 JAZZYEAR

GitHub Copilot Enterprise基于企业的知识库，帮助企业成员查询公有和私有代码，便于成员迅速熟悉新代码库，在工程团队间构建更紧密的协同一致性，确保企业成员能够建立统一的标准并访问一致的历史成果。

功能亮点

01 知识库应用

可以解释特定拉取请求摘要中的内容，进行编码实践。除此之外，还可使用知识库构建用作上下文的文档库，提升代码建议的相关性和质量。

02 AI模型微调

在客户代码库上用低秩近似（LoRA）方法训练模型，微调后的模型能够深入企业内部库的模块、函数、遗留代码或专有语言，提供更加精准的代码建议。

企业定制

01 用企业私有代码库定制聊天对话

02 与 Extensions等插件集成

03 为企业内部工具构建私有扩展

04 将知识库附加到聊天中，以了解企业组织环境

05 （公测中，即将推出）用于代码补全的微调模型

应用场景

审查代码拉取请求

通过生成数据拉取请求摘要，减少代码撰写时间。通过拉取请求对比差异，快速了解代码变更。

了解企业代码库

通过提供代码摘要、相关建议及对代码行为查询的快速回答来帮助理解和更新已有代码。

获取代码建议

通过自然语言集成聊天功能，基于企业特定代码库，提供个性化帮助和建议改进代码。

客户案例

- 多邻国工程师代码编写速度提高 25%
- 最大存储库的设置时间缩减至 1分钟
- 代码审查周转时间减少 67%
- 拉取请求量增加 70%

duolingo | GitHub 企业版

数据来源：公开资料，甲子光年智库整理，2024.10

目录

CONTENTS

甲子光年 JAZZYEAR

Part 01 机遇：AIGC引发的数字生产变化

Part 02 价值：重新构建代码开发的应用范式

Part 03 落地：智能时代AI+代码的先行者

Part 04 展望：人工智能普惠时代的期待

AI+范式下对于高质量的数据需求

- AI代码平台及产品在研发过程中离不开针对代码底层数据作为支撑，在产品应用过程中更是离不开企业代码库的“上下文”数据支撑，因此，对于数据治理耗费的精力不容忽视，数据质量可保证最终产出的代码质量达到预期水平。同时也是未来AI代码开发平台作为定制化解决方案不可忽视的关键因素。

代码数据

开源代码库 | 开源数据集 | 企业专有的代码数据 | 代码示例网站

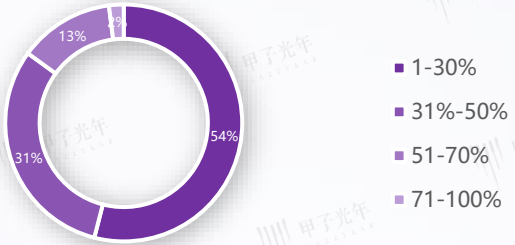
可以为AI代码产品提供大量的训练数据,帮助模型学习如何生成高质量的代码

非代码数据

开发场景中各环节的自然语言对话 | 用户行为与反馈 | 合成数据

这些非代码数据可以帮助AI代码产品更好地理解和适应实际开发环境中的需求和挑战

图：数据管理的时间花费



问题：平均而言，您的团队花在管理、清理或标记数据上的时间百分比是多少？

JetBrains，State of Developer Ecosystem Report（2023）

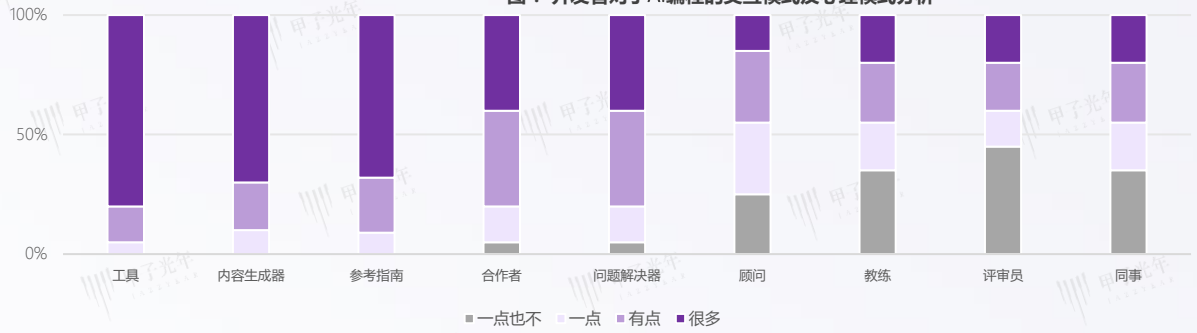
图：数据质量管理在整体数据治理中的难度说明

问题：以下哪种（哪些）数据驱动的活动对您或您所在的组织来说最难执行？

JetBrains，State of Developer Ecosystem Report（2023）

智能时代下，开发者的角色转化

图：开发者对于AI编程的交互模式及心理模式分析



工具导向
- 内容生成
- 工具
- 问题解决器
- 参考指南

社会导向
- 教练
- 同事
- 顾问
- 合作者伙伴
- 评审员

未来随着AI技术的不断迭代及应用升级，开发者始终要面对角色及能力的整合及调整

开发者们	AI带来的变化	在未来如何与AI共存
软件工程师	AI能够大幅减少软件工程师在基础性编码上的工作量 AI可以通过分析代码自动检测并修复部分bug,减少调试的时间	软件工程师将与AI代码助手协同工作,主要依赖AI来完成低级别的重复性任务，同时自身则专注于更具创造性和高层次的开发任务，更多的时间投入到逻辑层面的思索而非“体力”劳动。
软件架构师	AI可以根据架构模式和最佳实践提供设计建议,并自动生成架构文档 AI可以帮助模拟复杂系统的行为,预测不同架构选择的影响	软件架构师将利用AI工具来增强他们的设计能力，特别是在初步设计和方案验证阶段，但最终的决策仍将依赖于架构师的经验和对业务需求的深刻理解。
数据工程师	AI能自动处理常规的数据清洗、转换任务,并且可以智能地优化数据管道 AI可以分析数据管道的性能瓶颈,自动推荐或实现优化方案	数据工程师将利用AI来处理重复性数据任务，而自身则聚焦于更为复杂的系统设计和数据战略决策。他们需要确保大模型进行数据的数据处理的高质量。

数据来源：公开资料，甲子光年智库整理，2024.10

大模型技术发展下的工程化建设

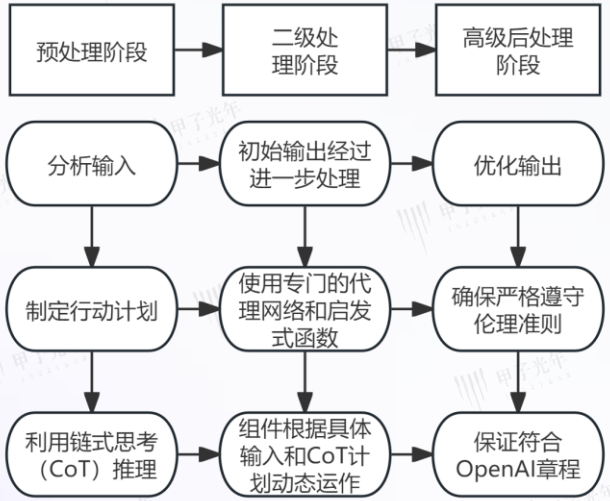
OpenAI o1带来了未来智能时代的新工程范式

- OpenAI o1与OpenAI之前发布产品的最大区别是更先进的推理能力，其预览版和迷你版已经在数学、科学和编码测试中击败了GPT-4o，其工程范式值得关注。

多层次、自适应的系统远超传统语言模型。从简单语言模型到**复杂、多方面AI架构**的演变代表了人工智能领域的**范式转变**。

- **动态生成**：可以根据输入复杂性动态调整处理过程
- **伦理对齐**：集成了伦理考量
- **行业集成**：复杂程度更接近AI生态系统而非简单模型
- **实时适应性**

图：根据现有效果对于OpenAI o1 的架构判断



图：根据O1对未来AI工程化的潜力判断

产品特点	具体应用/功能
复杂的推理能力	更先进的**代码推理循环**，能够**理解**和**生成复杂的代码逻辑结构**
思维链（CoT）推理	帮助编程系统**分解复杂的编程问题**，并**逐步构建代码解决方案**
专门的代理网络	包括针对**特定编程语言**或**编程范式优化**的组件
动态适应性	根据输入的**复杂性调整处理过程**，有助于处理各种编程任务
高级后处理	包括**代码优化**、**错误检查**和**遵循编码标准**的能力
伦理考量	在生成代码时，能够考虑**信息安全性**和**效果最佳实践性**

数据来源：Software Engineer等；甲子光年智库梳理，2024.10

大模型技术发展下的工程化建设

AI Agent代码平台更具灵活性和复杂性，结合现有技术，能够构建复杂的AI驱动应用

- AI Agent平台与低/无代码平台各有优势，许多平台开始**融合**这几种技术功能，**充分利用对方优势进行互补**，以提供更全面和强大的解决方案，争取未来低门槛编程市场份额。

特性	AI Agent平台	低/无代码平台	融合趋势及潜在优势
定义	**利用人工智能技术，如机器学习、自然语言处理等，来增强用户交互、自动化任务和决策支持的平台**	允许用户通过图形界面、拖放组件和模型驱动的逻辑来构建和部署应用程序，而无需或很少需要编写代码的平台	**AI Agent平台和低/无代码平台的融合可以创造出既智能又易于使用的解决方案**
优势	**- 高度自动化和智能化 - 个性化和自适应性强 - 能够处理复杂任务和决策**	- 易于上手，非技术用户友好 - 快速开发和部署 - 减少对专业开发人员的依赖	**- 结合AI的智能决策和自动化能力 - 提高低/无代码平台的智能化水平 - 增强AI Agent平台的用户交互和易用性**
劣势	**- 需要专业知识来配置和优化 - 成本较高，需大量用户均摊 - 对数据和算法的依赖性强**	- 定制化程度有限 - 无法满足高度复杂的需求 - 性能和扩展性受限	**- 通过融合，可以提供更广泛的应用场景 - 降低AI Agent平台的技术门槛 - 提升低/无代码平台的定制化和扩展性**
竞争关系	**- AI Agent平台通过智能化提供竞争优势 - 取代一些低/无代码平台的功能**	- 低/无代码平台通过易用性和快速开发吸引用户 - 限制AI Agent平台市场渗透	**- 融合可以创造新的市场机会 - 减少直接竞争，转向互补合作**
融合趋势	**- AI技术不断进步，AI Agent平台功能增强 - 低/无代码平台开始集成AI功能**	- 低/无代码平台的普及推动了对AI的需求 - AI Agent平台寻求更广泛的应用场景	**- 融合有助于解决单一平台的局限性 - 促进技术创新和市场发展**

短期——竞争

- 用户友好性
- 灵活性
- 复杂性
- 开发速度
- 资源需求
- 集成能力
- 成本效益
- 安全合规

长期——融合

- 自动化集成开发环境
- 用户体验
- 持续创新能力
- 跨平台兼容性
- **代码大模型**
- **AI+低代码**
- ……

数据来源：Sahay A, Indamutsa A, Di Ruscio D, et al. Supporting the understanding and comparison of low-code development platforms等，甲子光年智库整理，2024.10

甲子光年 JAZZYEAR

RAG的痛点及前沿解决方案：数据质量和检索策略依旧是其最大挑战

痛点描述	解决方案描述
内容缺失（知识库中缺少上下文）	清洁数据 & 更好的提示设计
错过排名靠前的文档（初始检索未命中上下文）	超参数调整 & 重新排名
不在上下文中——合并策略的局限（重新排名后缺少上下文）	调整检索策略 & 对嵌入进行微调
未提取出来（上下文未被提取）	清洁数据，prompt压缩，& LongContextReorder
格式错误（输出格式不正确）	更好的提示设计，输出解析，使用Pydantic程序，& OpenAI JSON模式
不正确的具体说明（输出的特定性级别不正确）	高级检索策略
不完备（输出不完整）	查询变换
数据摄取的可扩展性（摄入流程无法扩展至更大的数据量）	并行化摄入工作流程
结构化数据问答（无法对结构化数据进行问答）	使用Chain-of-table pack & Mix-self-consistency pack
从复杂PDF中提取数据（复杂PDF文档解析）	检索嵌入表格
后备模型（速率限制错误）	Neutrino路由器 & OpenRouter
LLM安全（如提示注入等）	NeMo Guardrails & Llama Guard

技术难点	前沿方案
数据库中实体关系的复杂性	实体结构信息技术GrahRAG
依赖人工注释的评估指标	实现检索增强生成的自动化评估RAGAS
应对多跳查询及问答力不从心	自主生成新查询，无需LLM 介入的EfficientRAG
RAG分类体系混乱、无标准	设立标准
缺少工业、生物等专业领域应用	Golden-Retriever 等针对专业应用场景技术横空出世
.....	.....

数据来源：Graph Retrieval-Augmented Generation: A Survey；RAGAS: Automated Evaluation of Retrieval Augmented Generation等；甲子光年智库梳理，2024.10；

专家观点&推荐厂商

行业专家观点

在大模型赋能下，研发过程正在发生一场深刻变革，大模型的理解、生成、逻辑、记忆能力同软件研发领域相结合，让软件开发更高效、更安全。

百度在智能编码领域已经探索了两年，智能编码提效工具文心快码已在公司内部被数万名工程师使用，智能研发提效显著。迄今百度每天生成的代码中，有30%由文心快码生成，整体采纳率达46%，这一提效工具的应用令工程师整体提效达到12%，显著提升了单位时间内的代码提交数量和业务迭代速度。更重要的是，智能编码提效工具使工程师得以从繁杂、重复的工作中释放出来，专注于更有价值和创造性的工作。

文心快码的应用不仅提高了研发效率，也显著提升了代码质量。在百度内部实践中，文心快码安全漏洞扫描准确率已超过95%，并有83%的扫描漏洞已被修复闭环。此外，不仅是研发阶段，文心快码在安全运营阶段也能大幅提升效率。

目前，百度通过文心快码的内外部的经验，已经总结出一套标准化的落地流程以及最佳实践，通过实施人机协同原则和构建落地执行体系，促进企业工程文化的转变和数据驱动价值闭环。

——百度副总裁陈洋 文心快码 Baidu Comate

我认为编码辅助是大模型应用的最大场景之一，随着模型能力不断提升，新产品不断出现，有望在未来 1-2 年内为软件研发过程带来颠覆性影响。AI 替代人类工程师完成基础事务性工作的趋势不可避免，我们需要从现在开始就拥抱这项技术，甚至参与到模型构建、场景挖掘、产品研发过程中，不断探索 AI 原生开发范式，思考未来基于代码大模型之上的软件研发流程如何重塑。

通义灵码代码大模型是业界公认的具备全球领先能力的模型。例如用于代码补全的 codeqwen 模型，其拥有 GQA 架构，经过了超过 3T tokens 代码相关的数据进行预训练，共计支持上百种编程语言、且最长支持 64K 的上下文输入。效果方面，codeqwen 展现出了非凡的代码生成、长序列建模、代码修改、SQL 能力等。基于 codeqwen 模型构建的通义灵码产品，能够在代码生成方面拥有优异表现，成为了各大企业 AI 代码助手项目的首选产品。

我们目前核心是基于代码大模型打造通义灵码智能编码助手，因为这是目前已经可以成熟落地的大模型场景之一。企业开始使用通义灵码后，就能够直接获得 10%-15% 的综合人效提升，见效快，性价比极高。另一方面我们也和企业探索代码大模型在 DevOps 全流程上的应用，期望能够从需求管理、测试生成、部署运维辅助等端到端实现智能化体验。最后我们也在探索 AI 程序员多智能体自主编程领域，并计划在未来合适时间推出相应产品，从而实现 AI 为主人为辅的新一代软件研发范式。

——阿里云通义灵码产品技术负责人陈鑫 通义灵码

数据来源：《百度副总裁陈洋：智能代码助手文心快码已被内部数万名工程师使用》，《通义灵码代码大模型应用实践访谈》甲子光年智库整理

行业专家观点

随着人工智能技术的飞速发展，AI 代码平台领域正成为AI 的重要落地领域。在这个充满机遇与挑战的时代，众安保险也积极响应行业趋势，致力于研发 AI 代码工具，以提升企业的开发效率和质量。

在技术战略与思路上，众安保险注重将 AI 技术与保险业务深度融合。我们凭借海量的保险数据以及丰富的保险知识，进行深度的场景应用，通过 AI 中台化的能力持续释放生成式 AI 能力。同时，积极构建开源社区，鼓励外部开发者参与，以打造共赢共创的行业技术环境。

我们深刻洞察到，当前软件开发过程中存在诸多痛点，如重复性任务多、代码质量参差不齐、团队协作沟通成本高等。为了解决这些问题，众安保险的 DevPilot AI 代码助手应运而生。DevPilot 提供了一套轻量高效的全栈代码助手开源解决方案帮助开发者更智能、更快速、更少错误地进行编码。

未来，众安保险将继续在 AI 代码领域深入探索，也希望整个行业能够共同努力，推动 AI 代码平台的发展。各行业充分发挥自身优势，加强合作与交流，共同探索 AI 在代码开发领域的更多应用场景和可能性。相信在大家的共同努力下，AI 代码平台将为行业带来更大的价值，推动智能时代的开发模式重构，释放 AI 数字生产力，为社会的发展做出更大的贡献。

——DevPilot CTO Hudson

基于大模型的AI编程已经成为提高软件开发效率和质量的关键生产工具，各行各业都在积极探索构建智能化研发管理体系，加速企业数智化升级。然而，企业在大模型真正落地应用中，面临数据泄露与代码安全风险、算力资源有限、研发效能难以量化评估等挑战，特别是通用大模型因基于开源数据训练，缺乏企业特有的领域知识，导致生成的代码难以符合企业的业务逻辑和编码规范，很大程度上限制了AI在企业研发中的深入应用。

aiXcoder多年来一直致力于将前沿人工智能技术应用于软件工程，在大模型落地方面具备非常成熟且丰富的技术和实践经验，可以为企业提供私有化部署、领域化大模型落地及定制化开发等完备的智能化软件开发解决方案。根据企业数据和代码的安全合规要求及企业现有算力资源，实施兼顾效果、性能和可落地等因素的最佳模型部署方案；结合企业代码和文档，采用独有的大模型个性化训练方式，在极少的计算资源和人力投入下，快速构建企业专属代码大模型。同时，构建辅助管理平台，实现大模型的有效管控和效果评估。当前，aiXcoder已在金融、军工、航空航天、通信、软件服务、IT等行业成功部署50多个项目，实现了代码大模型的多场景应用和实际效果验证。

——aiXcoder

2024甲子光年AI代码平台及产品推荐

文心快码 Baidu Comate **百度智能云，文心快码**

通义灵码 **阿里云，通义灵码**

豆包MarsCode **火山引擎，豆包MarsCode**

腾讯云 AI 代码助手 **腾讯云，腾讯AI代码助手**

CodeGeeX **智谱清言，CodeGeeX**

DevPilot **众安保险，DevPilot**

AIXcoder **硅心科技，aiXcoder**

代码小浣熊 CoRaccoon **商汤科技，代码小浣熊 Raccoon**

网易数帆 **网易数帆，CodeWave**

iVX.cn **成都云动力科技公司，iVX**

以上排名不分先后

THANKS

谢　谢

北京甲子光年科技服务有限公司是一家科技智库，包含智库、媒体、社群、企业服务版块，立足于中国科技创新前沿阵地，动态跟踪头部科技企业发展和传统产业技术升级案例，致力于推动人工智能、大数据、物联网、云计算、AR/VR交互技术、信息安全、金融科技、大健康等科技创新在产业之中的应用与落地

关注甲子光年公众号

扫码联系商务合作

分析师

刘瑶
18401669467

李哩莉
wxid_05offrd7pvf632

智库院长

宋涛微信
stgg_6406

商业合作负责人

李胜驰
18600783813（手机&微信）