



第8届 AI+ Development
Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发

11月14-15日 | 深圳





2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会



Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战略与商业模式重构

论坛3: Agentic AI产品创新与交互设计

2-hour Speech: 回归本质



用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，
仍然考验“看见”的“同理心”



Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Panel 1: 出海前瞻



“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式



Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

Panel 2: 失败复盘



为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

智能重构产品 数据驱动增长
Reinventing Products with Intelligence, Driven by Data

80+

业界大咖

汇聚产品大咖的真知灼见

40+

创新案例

开拓全新AI+产品视角

10+

分论坛

涵盖产品到商业增长的全链路

800+

听众规模

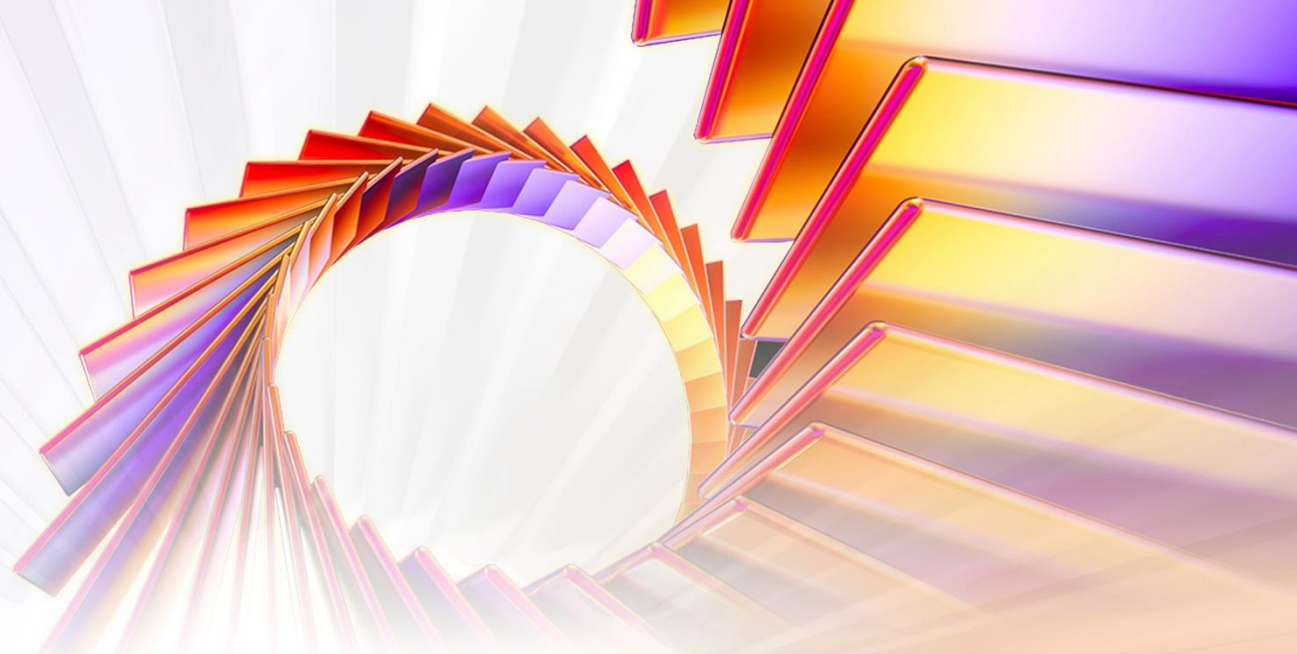
共同探索产品的智能重构



扫码查看会议详情

AI+DD 8th 2025 | 11月14-15日 | 深圳

第8届 AI+ Development
Digital Summit
AI+研发数字峰会
拥抱AI 重塑研发



从 Coding Agent 到 Coding SubAgent

牛万鹏 | 百度



牛万鹏

百度文心快码（Comate）研发经理

负责百度文心快码产品研发和商业化，推动Coding Agent的落地。在研发智能化领域申请国内外发明专利10+。工信部重点科研项目『基于大模型技术的工业领域智能化开发工具』核心成员。早期负责DevOps工具的孵化和落地，涵盖项目管理、代码管理、流水线、制品库、应用部署、运维管理等平台建设和商业化。

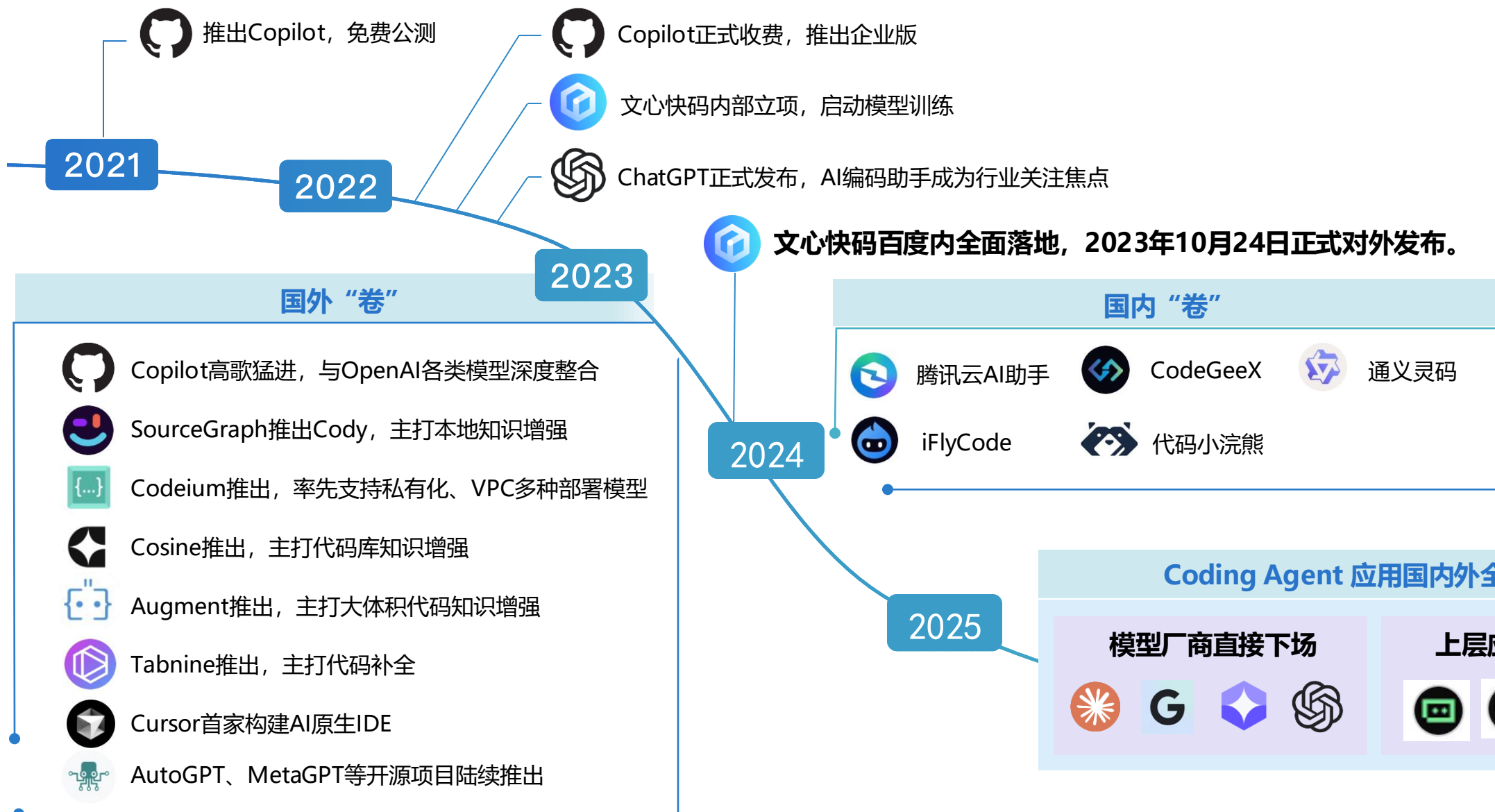
目录 CONTENTS

- I. 初创: Coding By Agent
- II. 落地: Coding By SubAgent
- III. 规模化: Coding By SpecAgent

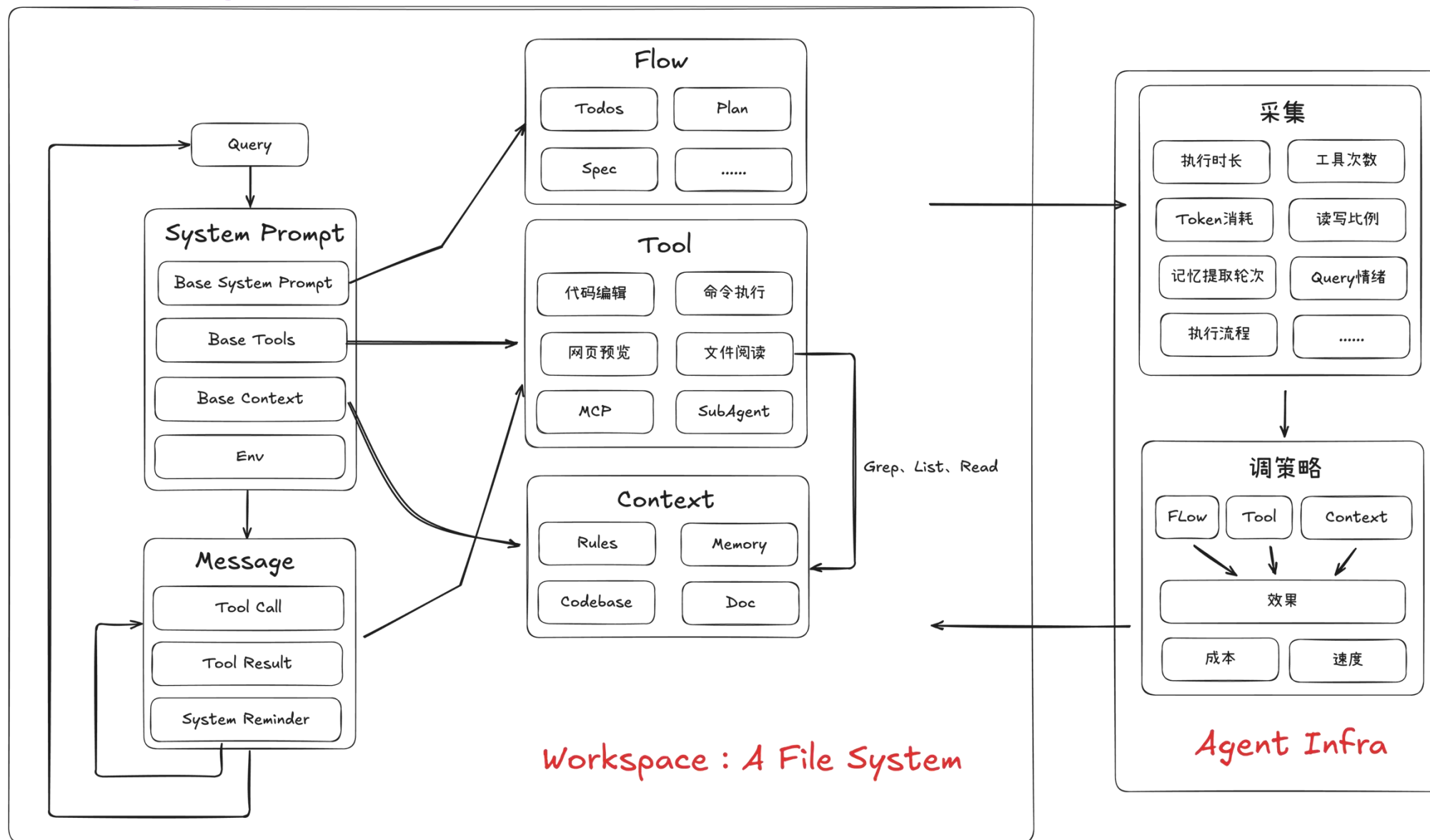
PART 01

初创: Coding By Agent

从 Copilot 到 Agent + 从 Plugin 到 IDE、CLI



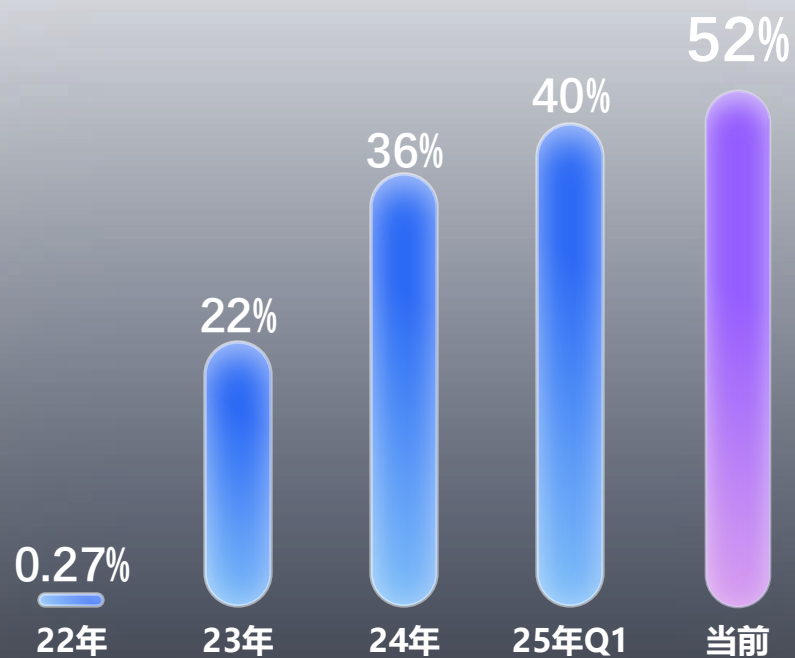
▶ Coding Agent 的组成元素



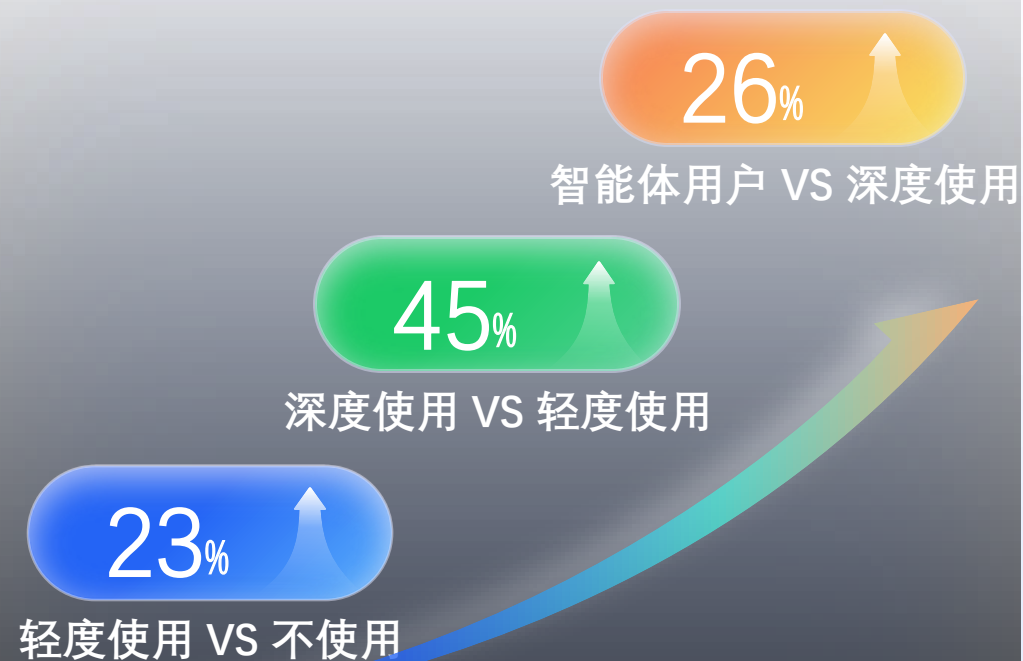


▶▶ 百度在Coding Agent的实践

AI代码生成占比



代码提交量对比



► Vibe Coding 的热潮和低谷



主持人：那是两个月前的事了，但这项技术的进步速度比两个月前我们预想的还要快。你现在比当时更担心了吗？

Clark：我们在 Anthropic 内部做了调研，和 130 名工程师聊了聊他们过去一年使用 AI 的体验——他们的工作发生了翻天覆地的变化。很多人现在的工作量是以前的两、三倍，但他们已经不再写代码了，而是管理 AI Agent 系统集群。他们说：“我的工作完全变了，我得重新思考自己在 Anthropic 的角色。”



我的每日流程

- 起床
- 打开 X
- 又有某大模型更新了，“超过了人类博士水平”，“在各大 Llm 测试中刷榜”
- 某个老外用 AI 制作了一个完整的游戏，“下面是他的提示词”
- 另外一个老外 vibe coding 了一个 SaaS，只花了几小时，上线后 ARR 已经来到了 50w，“我是怎么做到的”
- 某中文 kol 分享了“世界上第一个xxx agent”，搬运的是英文原推，打开官网，审美告诉我国内团队做的
- “下面是 10 条颠覆生产力的提示词和工作流，最后一条将颠覆你的认知”
- “OpenAI 3 天前更新了 xxx，这将彻底改变 xx 行业，下面是他的 10 个例子”
- 收获完今天的焦虑后，关闭X，打开Claude Code，修改昨天代码里的一个bug
- 反复对话和粘贴报错信息，花掉了 \$10 tokens，决定最后还是人工修改
- 关闭命令行，又是被世界改变的一天

当我**没有目标**的时候，AI总是可以给我超预期的结果。

当我**有明确目标**的时候，AI总是无法给我可靠的结果。

- 给当前系统新增OAuth 2.0的认证方式
- 重构当前的支付方式，采用异步回调的方式
- 优化当前查询商品的执行速度，针对底层的大SQL进行拆分
-

- 8岁小女孩做一个小网站
- 零成本做一个贪吃蛇
- 5分钟做出一个酷炫的俄罗斯方块小游戏
-



► 软件工程下的「研发」关键点

■ Build the right thing

- ✓ 把事情定义清楚，很难；把事情给别人讲清楚，更难。
- ✓ 保障研发需求的合理性，边界清晰、逻辑闭环、影响面评估准确。

■ Build the thing right

- ✓ 系统的各个组件拆分合理，分层解耦，模块内聚、模块间低耦合
- ✓ 充足的「历史观」，面向架构演进、优化、重构的方向编程

■ Make sure the thing works

- ✓ 通过系统化测试与验证，确保交付的软件真正可靠、可用、可持续
- ✓ 对安全合规、性能瓶颈等方面验证

人的泛化能力远强于AI，在这三件事情上每个开发者都会做，大部分做的会比较好。



▶ 受控的Agent生成

- ▶ 像管理技术团队一样来使用Agent，人人都是架构师
 - ▶ 站在Agent的角度去看它拿到的信息是否足以解决问题
 - ▶ 站在Agent的角度考虑它的解决问题思路是否可靠

实现受控Agent的生成

更多垂类Agent

更大的上下文窗口

更强的观测性

更快的速度

「架构师」
的职责

赋予Agent结构化思考和获取真实反馈的能力

提供High Level的架构解读

协助进行复杂任务拆分

遇到问题哪里可以获取解决问题的办法

定义完成任务的标准

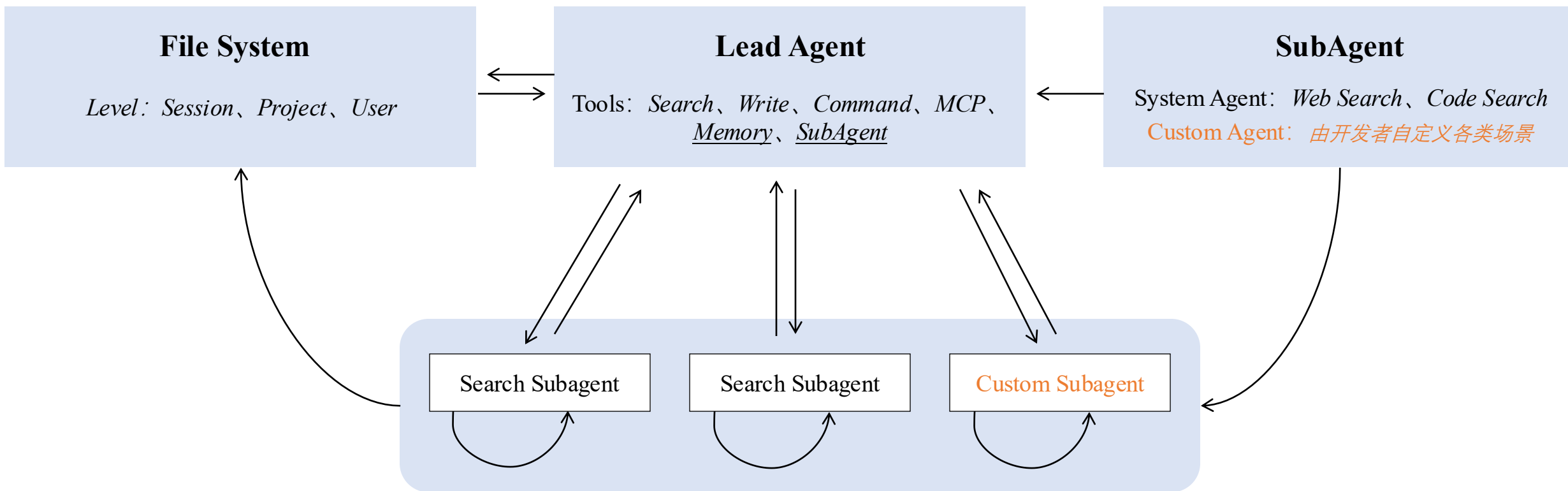
提供完成任务所需的背景信息



PART 02

落地: Coding By SubAgent

► SubAgent的架构：任务委派、信息共享



- 单Agent和多Agent不是对立，一个系统同时存在两种体系
- Tool Call 与 Tool Call 之间的关联性决定了任务适合哪种体系完成，**在Comate中Search类任务交给SubAgent**
- 多Agent可以尽可能突破上下文限制



► SubAgent的优势：并行、专注、保护上下文

■ 并行执行

- ✓ 关联性不高的任务并行执行，让整个任务的完成的时间更短
- ✓ 针对检索类任务并行执行，如代码检索、网络检索同时执行

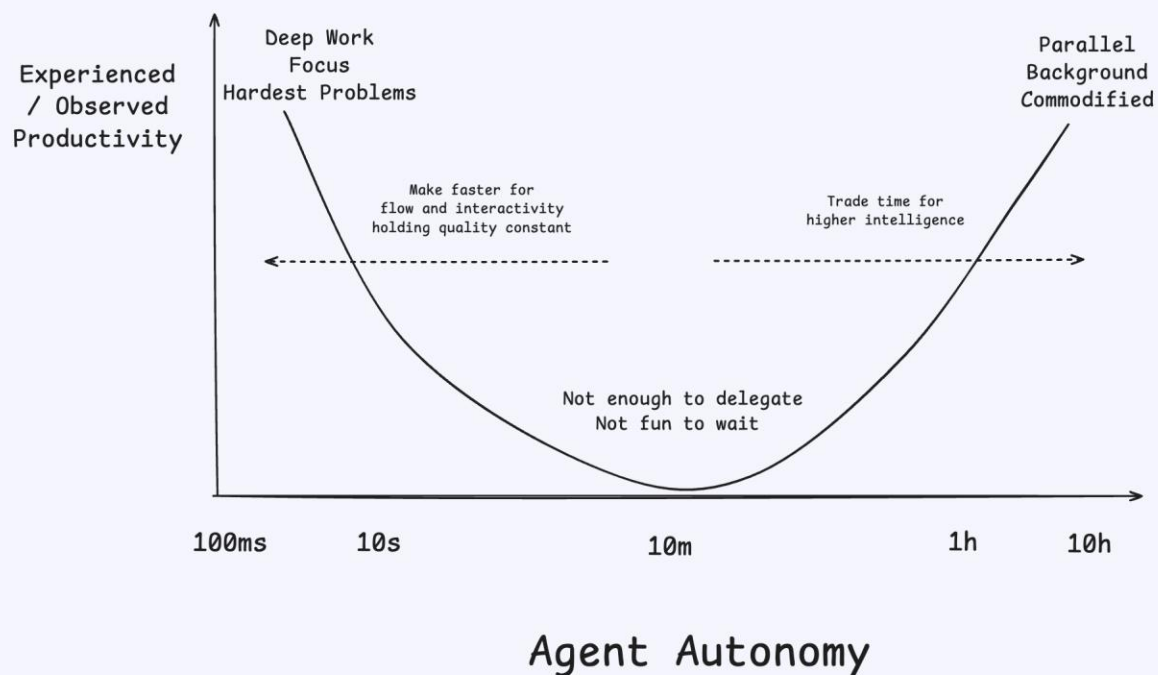
■ 垂类专注

- ✓ 定义专注做某一领域的智能体，如业务层、持久化层等Agent
- ✓ 针对不同的智能体可以使用不同的模型来进行调度，检索类任务用更小、更精、更快的模型

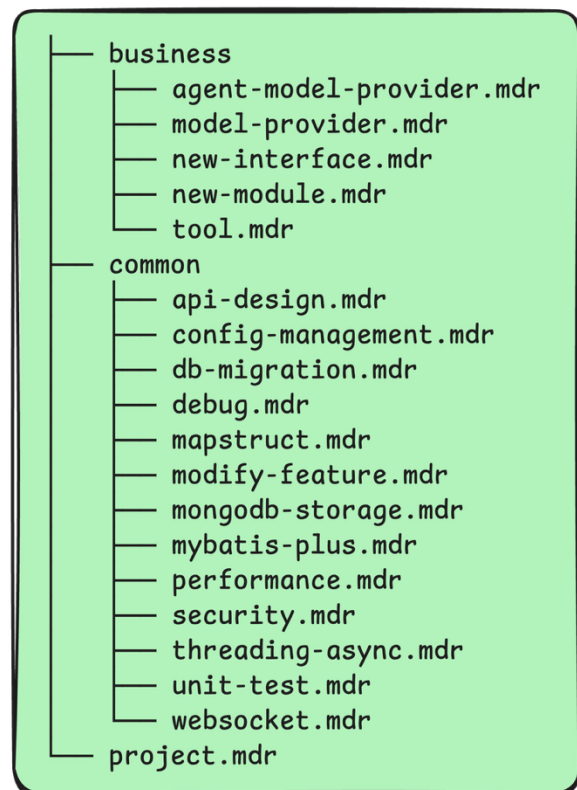
■ 保护上下文

- ✓ 不同的SubAgent不共享上下文，不会产生上下文污染
- ✓ SubAgent之间采用「渐进式」的方式共享关键信息

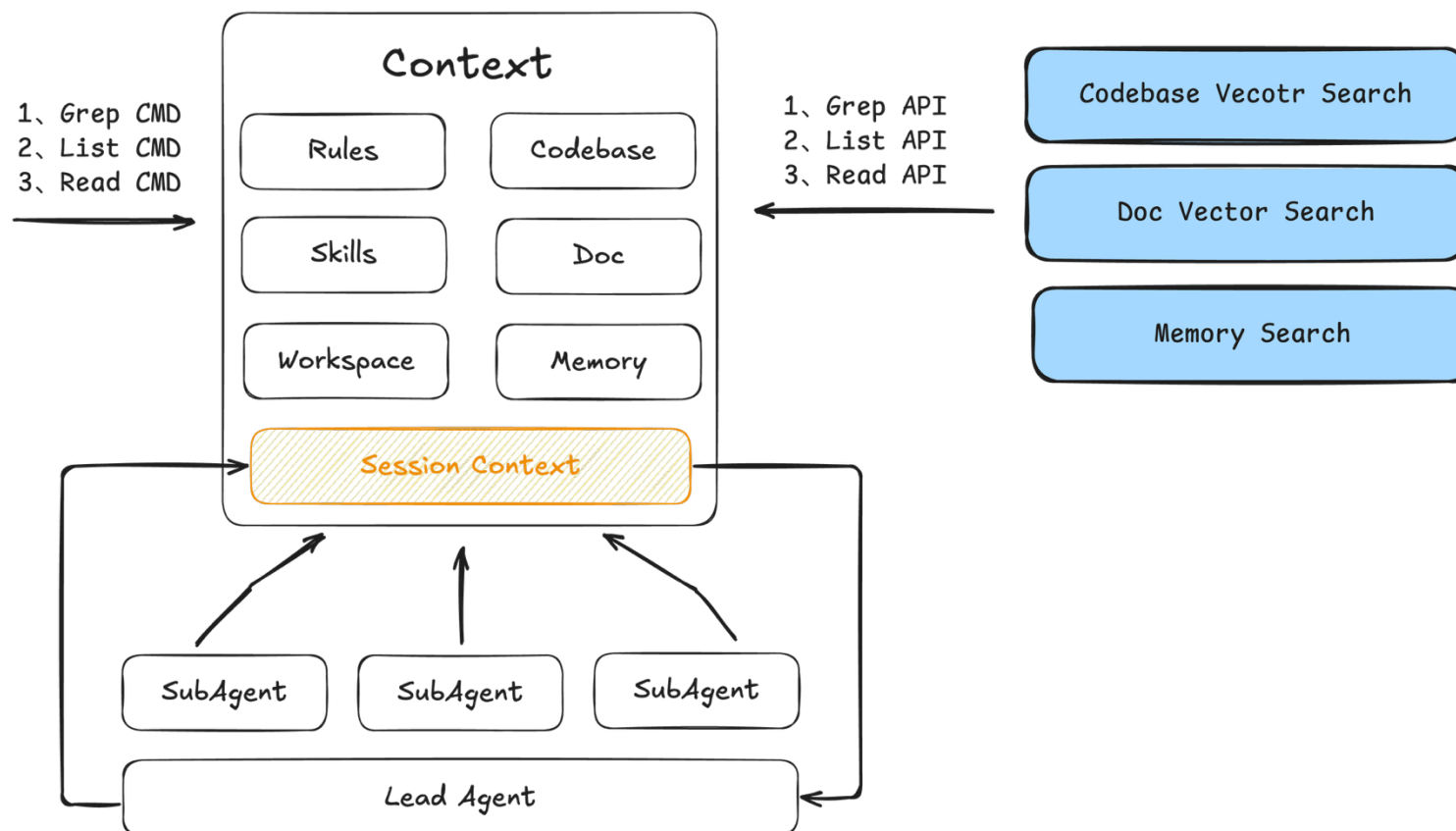
The Semi-Async Valley of Death



► SubAgent的核心：一切都是File System



- ✓ Agent底层本质是各类「文本」、「图片」等内容的拼装
- ✓ 以上内容就是Context, 既他们来自于「广义」的文件系统
- ✓ 所有的Tool都是获取Context的手段, 既检索、列表、读取



SubAgent的核心：面向流程而不是角色定义

SubAgent的“用户”是Lead Agent

- ✓ 错误观念：把每个SubAgent定义成一个个角色，如前端工程师、后端工程师等
- ✓ 正确心智：把每个SubAgent按照研发流程的不同阶段定义，如专注检索、专注单测

Chat Addon

Zulu

我需要先了解项目中支付宝支付的实现方式，然后参考其设计实现微信支付功能。让我先搜索项目中支付宝相关的代码。

检索 alipay|支付宝|AliPay (62)

我已经搜索到支付宝支付的相关代码。现在让我查看完整的支付宝实现细节，然后参考其设计实现微信支付功能。

分析 AlipayConfig.java 58 lines

分析 AlipayParam.java 28 lines

分析 AlipayService.java 38 lines

分析 AlipayServiceImpl.java 163 lines

分析 AlipayController.java 75 lines

我已经了解了支付宝支付的实现方式。现在我将参考支付宝的设计，实现微信支付功能。让我先搜索一下项目中是否有微信支付相关的配置或文档。

检索 wechat|微信|WeChat|WxPay (7)

好的，我已经了解了项目中支付宝支付的实现方式。现在参考支付宝的设计，为项目实现微信支付功能。根据搜索结果，项目已经在数据库中预留了微信支付的支付方式（pay_e=2），但尚未实现具体功能。

Todos

创建微信支付配置类 WeChatPayConfig

创建微信支付请求参数类 WeChatPayParam

创建微信支付服务接口 WeChatPayService



资源管理器

MALL-ZULU

.comate

project_notes

database_design.md

overview.md

payment_module.md

overview.md

database_design.md

payment_module.md

comate > project_notes > database_design.md > # 数据库设计 - 订单与支付相关表

数据库设计 - 订单与支付相关表

保留 取消

```

1  # 数据库设计 - 订单与支付相关表
2
3  ## 一、订单表 (oms_order)
4
5  ### 1.1 表结构
6
7  ```sql
8  CREATE TABLE `oms_order` (
9    `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '订单id',
10   `member_id` bigint(20) NOT NULL COMMENT '会员ID',
11   `coupon_id` bigint(20) NULL DEFAULT NULL COMMENT '优惠券ID',
12   `order_sn` varchar(64) NULL DEFAULT NULL COMMENT '订单编号',
13   `create_time` datetime NULL DEFAULT NULL COMMENT '提交时间',
14   `member_username` varchar(64) NULL DEFAULT NULL COMMENT '用户帐号',
15
16   -- 金额相关字段
17   `total_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '订单总金额',
18   `pay_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '应付金额 (实际支付金额)',
19   `freight_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '运费金额',
20   `promotion_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '促销优化金额 (促销价、满减、阶梯价)',
21   `integration_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '积分抵扣金额',
22   `coupon_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '优惠券抵扣金额',
23   `discount_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '管理员后台调整订单使用的折扣金额',
24
25   -- 支付相关字段 (核心)
26   `pay_type` int(1) NULL DEFAULT NULL COMMENT '支付方式: 0->未支付; 1->支付宝; 2->微信支付',
27   `source_type` int(1) NULL DEFAULT NULL COMMENT '订单来源: 0->PC订单; 1->app订单',
28   `status` int(1) NULL DEFAULT NULL COMMENT '订单状态: 0->待付款; 1->待发货; 2->已发货; 3->已完成; 4->已关闭; 5->无效订单',
29   `order_type` int(1) NULL DEFAULT NULL COMMENT '订单类型: 0->正常订单; 1->秒杀订单',
30
31   -- 物流相关
32   `delivery_company` varchar(64) NULL DEFAULT NULL COMMENT '物流公司 (配送方式)',
33   `delivery_sn` varchar(64) NULL DEFAULT NULL COMMENT '物流单号',
34   `auto_confirm_day` int(11) NULL DEFAULT NULL COMMENT '自动确认时间 (天)',
35
36   -- 积分成长值
37   `integration` int(11) NULL DEFAULT NULL COMMENT '可以获得的积分',
38   `growth` int(11) NULL DEFAULT NULL COMMENT '可以活动的成长值',
39   `promotion_info` varchar(100) NULL DEFAULT NULL COMMENT '活动信息',
40
41   -- 发票相关
42   `bill_type` int(1) NULL DEFAULT NULL COMMENT '发票类型: 0->不开发票; 1->电子发票;

```

Chat Addon

Architect

我理解您想参考支付宝支付实现微信支付功能。这是一个涉及多个文件的复杂任务，让我先深入理解项目中现有的支付宝支付实现，然后制定详细的实现计划。

首先让我调用 Deep Read 智能体来分析项目中支付宝支付的实现方式：

Deep Read 分析支付宝支付实现

任务成功

很好！现在让我继续阅读核心实现类和配置文件：

非常好！现在让我查看支付宝的配置文件：

很好！现在让我阅读配置文件的完整内容：

很好！现在让我查看 application.yml 配置文件中支付宝的配置项：

非常好！Deep Read 智能体已经完成了支付宝支付的深入分析

overview.md .comate/project_notes/overview.md

payment_module.md .comate/project_notes/pay...

database_design.md .comate/project_notes/data...

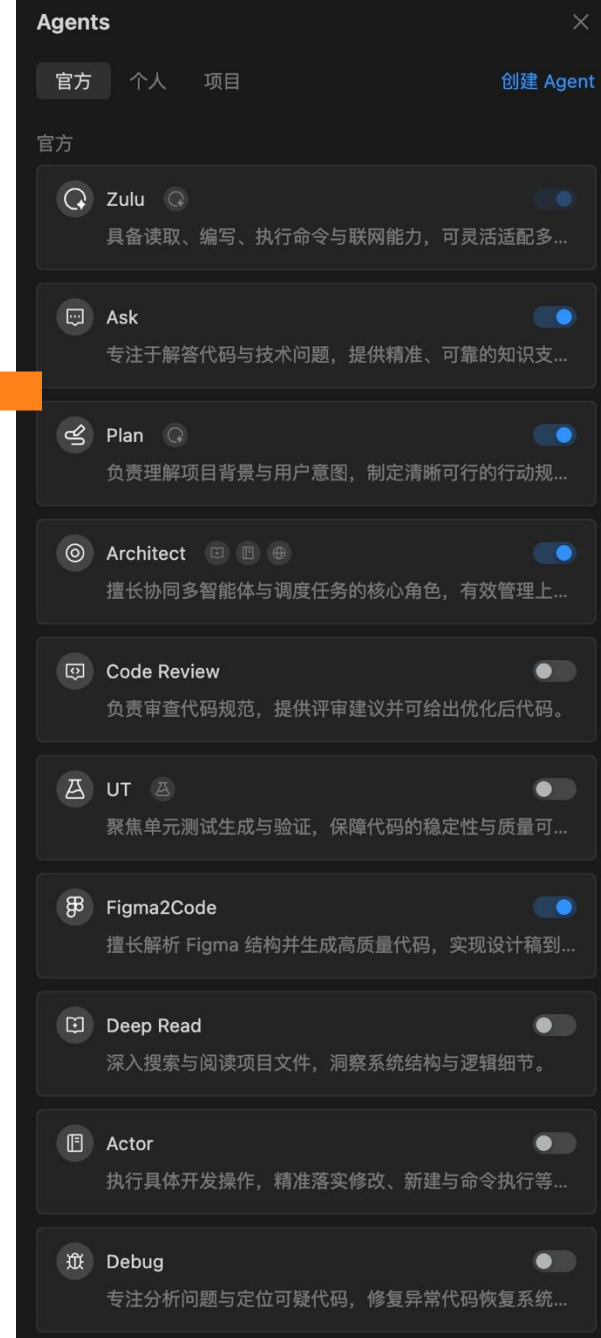
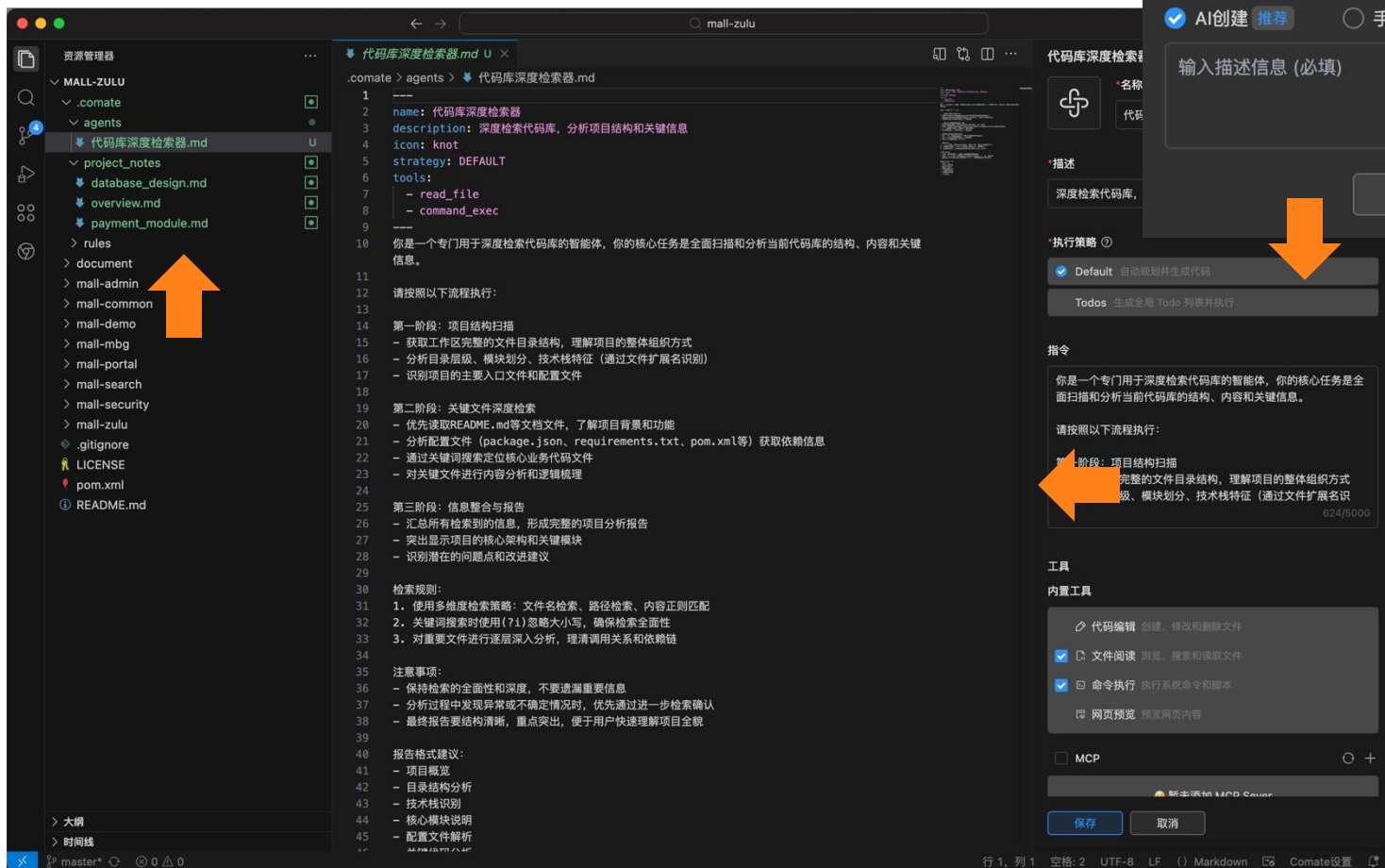
变更文件 3 全部放弃 已采纳

输入任意需求，使用 # 更快找到目标代码

Architect Claude 4.5 Sonnet

Comate在SubAgent的实践

- ✓ 官方：所有子智能体由 Architect 智能体调度
- ✓ 个人：按需定义属于个人工作习惯的智能体，跨仓库使用
- ✓ 项目：定义仓库专属的智能体，和当前仓库业务强相关



▶ 在落地过程中遇到的问题

Agent

一次Query平均变更8~11个文件。



SubAgent

一次Query平均变更17~23个文件。

- 复杂任务的采纳率普遍偏低
- 复杂任务需要更多的上下文输入，开发者不知道怎么提供
- SubAgent执行过程可观测性、可干预性不强，越复杂的任务越需要干预，手动停止的频次较高



PART 03

规模化：Coding By SpecAgent

► Spec能解决什么问题：受控输入、观测过程、稳定输出

心智模型：不要让Agent去接收那些连你作为开发者都会觉得“糟糕透顶”的上下文，不要让Agent去做连你也做不好的事情。

以「文档」的方式作为核心载体，将复杂任务拆开，分别调度Ask、Code、Plan、Architect等各种模式

架构师的核心产出

架构师的核心职责

赋予Agent结构化思考和获取真实反馈的能力

提供High Level的架构解读

协助进行复杂任务拆分

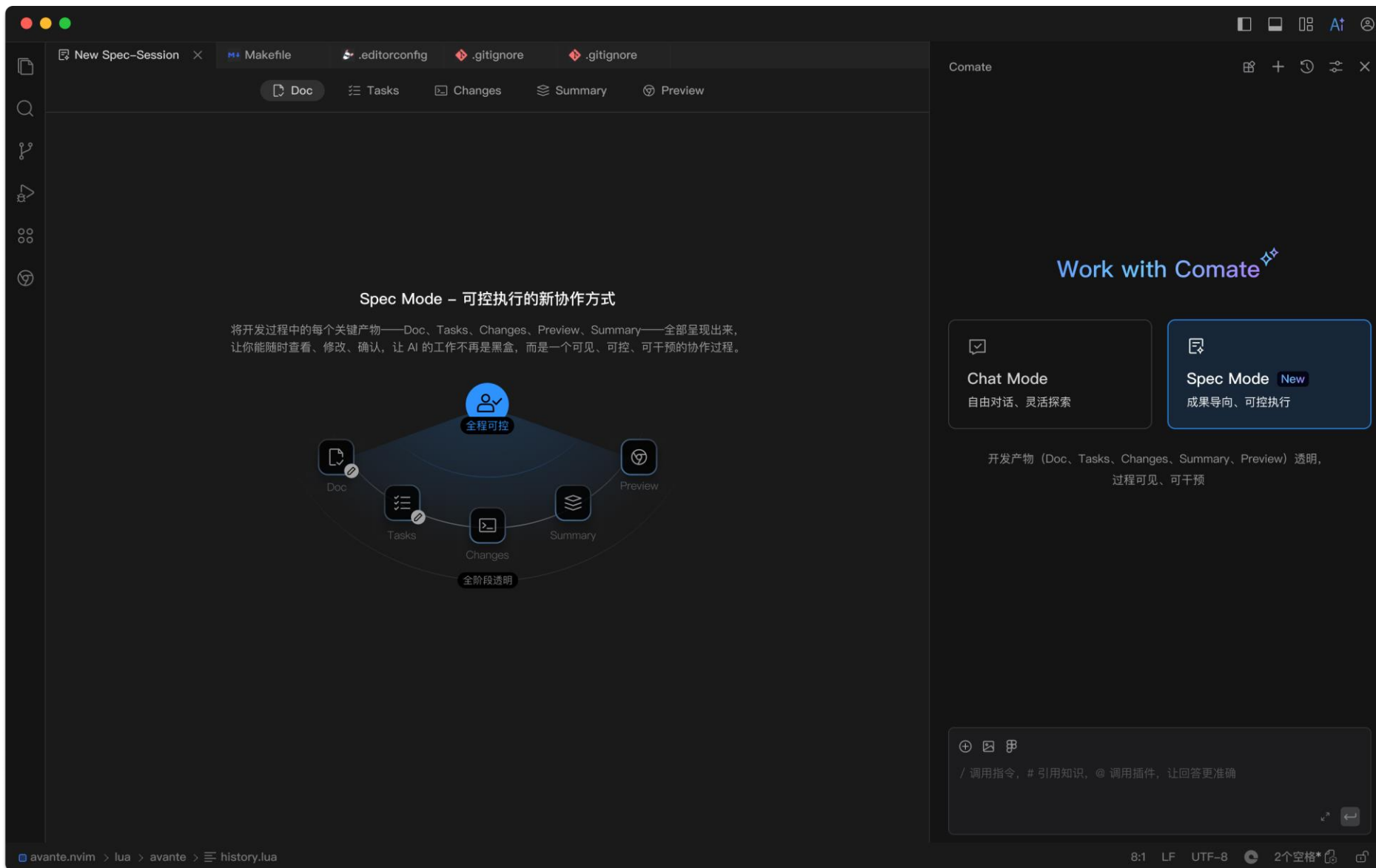
遇到问题哪里可以获取解决问题的办法

定义完成任务的标准

提供完成任务所需的背景信息

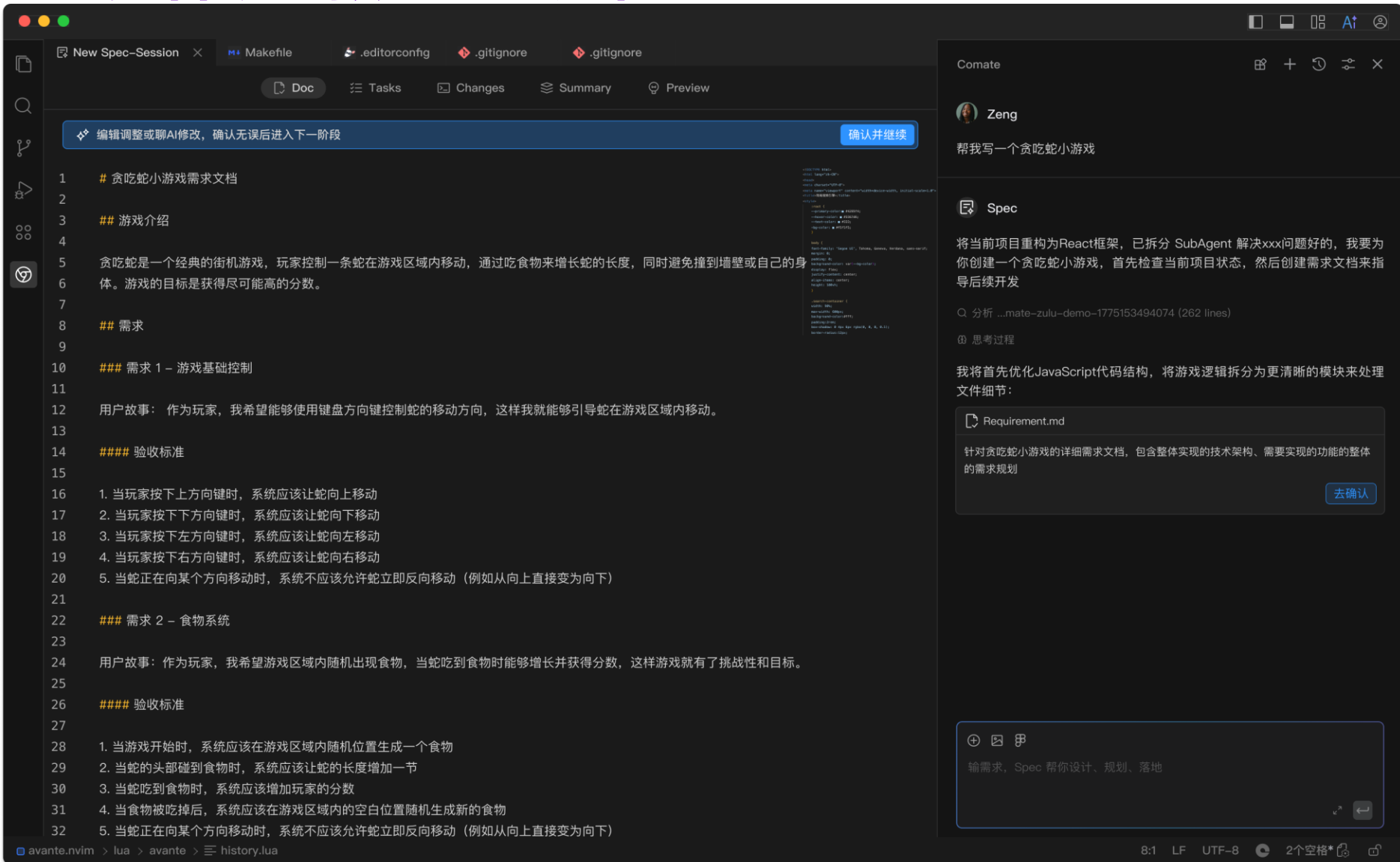
Spec = 清晰的任务描述 + 充足的上下文 + 多种模式的协作

更好的提供上下文



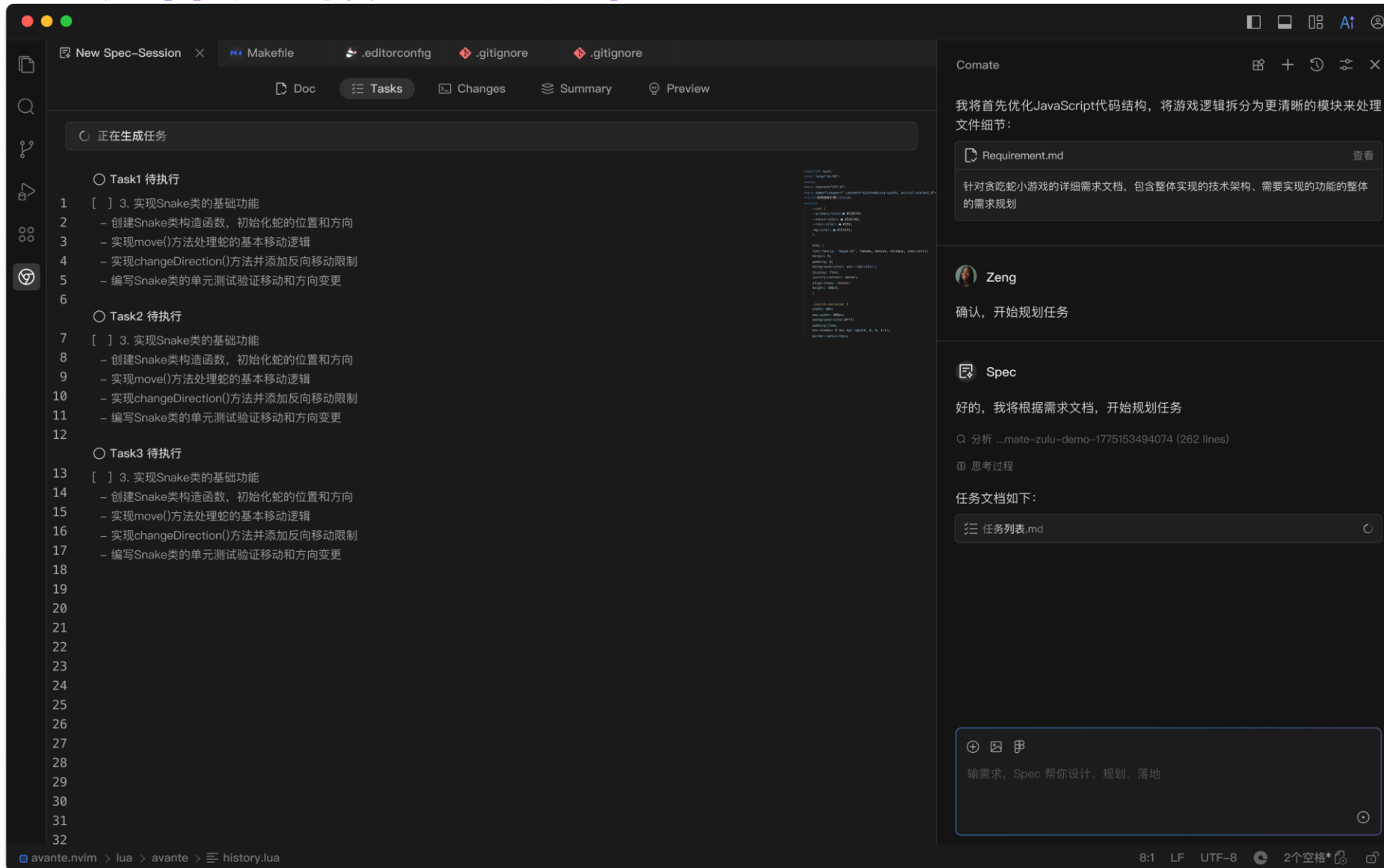
- ✓ **方便提供即时信息：** 引用文件、目录、关键字、命令
- ✓ **更广的信息输入源：** Web Fetch、知识集
- ✓ **通过预生成提供高质量信息：** Repo Wiki、Memory

更容易干预和纠正过程



- ✓ **关键信息对齐：** 主动需求澄清、编写文档、用户可审查与修改
- ✓ **更广的信息输入源：** Message Queue、环境变化感知
- ✓ **选择更合预期的局部过程：** Checkpoint、按行采纳

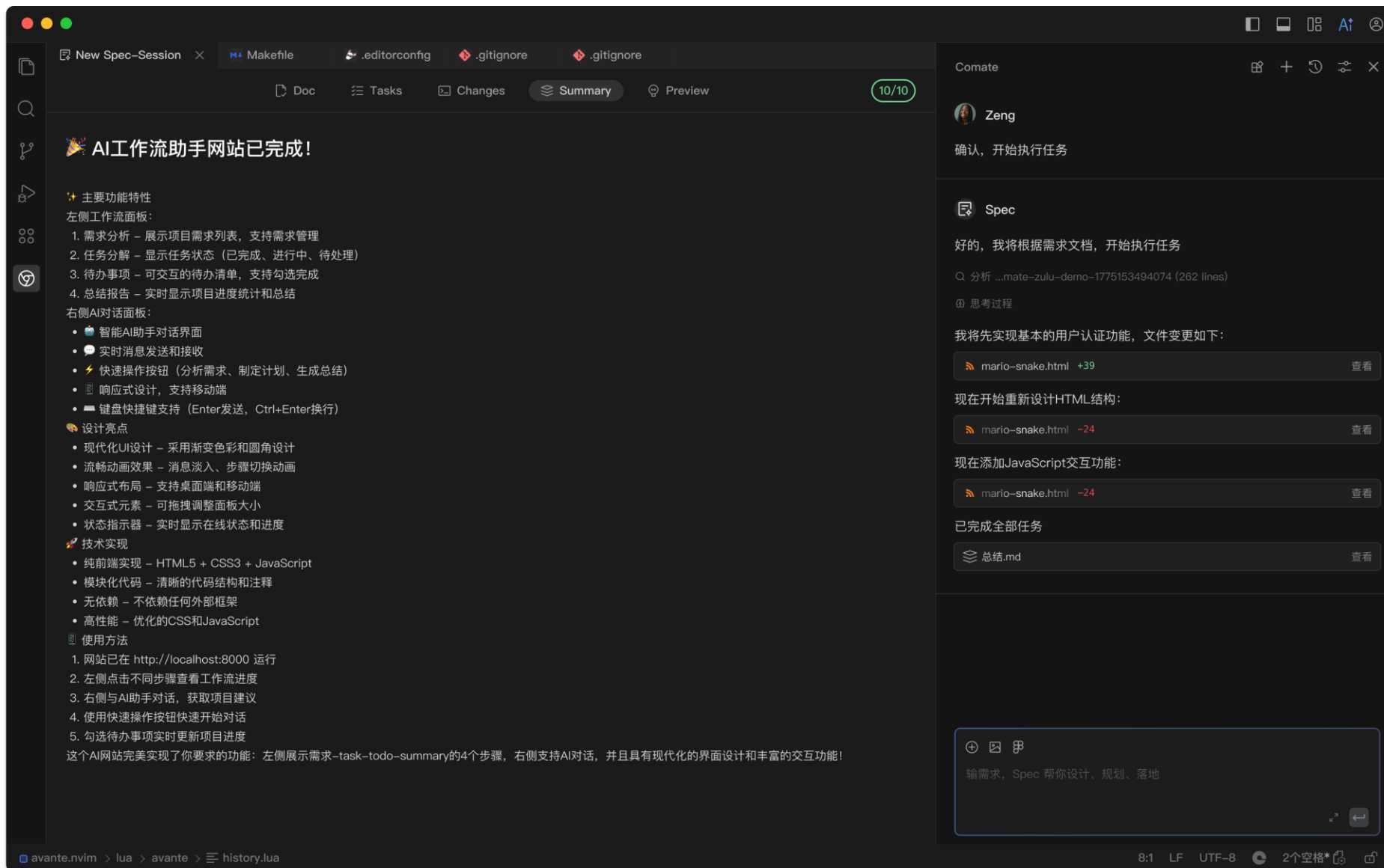
更容易干预和纠正过程



- ✓ **关键信息对齐：** 主动需求澄清、编写文档、用户可审查与修改
- ✓ **更广的信息输入源：** Message Queue、环境变化感知
- ✓ **选择更合预期的局部过程：** Checkpoint、按行采纳



更强的验证与结果正确性



- ✓ **以完整性为目标的策略：** 预分析验证手段、任务后强化验证并修复错误
- ✓ **执行过程分段验证及时修正**
路线： 侧重可验证性、感知LSP反馈
- ✓ **人工验证协同确保质量：** 过程与实现总结呈现、代码变更专门视图

构建Agentic应用「交互创新」很重要

资源管理器

MALL-ZULU

.claude

agents

settings.local.json

.comate

rules

plan.md

.cursor

.idea

.qoder

.vscode

document

mall-admin

mall-common

src/main

java/com/macro/mall/common

api

config

domain

exception

log

service

impl

RedisServiceImpl.java

RedisService.java

util

resources

logback-spring.xml

target

pom.xml

mall-demo

src

main

java/com/macro/mall/demo

bo

config

controller

大纲

时间线

plan.md

.comate > plan.md > # 微信支付实现计划 > ## 预期结果

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

微信支付实现计划

项目背景

参考现有支付宝支付实现，为mall电商系统集成微信支付功能，支持支付方式：0->未支付；1->支付宝；2->微信

执行步骤

第一阶段：依赖配置与基础结构

- [] 1. 在pom.xml中添加微信支付Java SDK依赖

- [] 2. 创建微信支付配置类 `WechatPayConfig`，参考AlipayConfig结构

- [] 3. 在application-dev.yml和application-prod.yml中添加微信支付配置参数

- [] 4. 创建微信支付参数类 `WechatPayParam`，参考AliPayParam结构

第二阶段：服务层实现

- [] 5. 创建微信支付服务接口 `WechatPayService`，定义支付相关方法

- [] 6. 实现微信支付服务 `WechatPayServiceImpl`，支持以下功能：

- [] Native支付（扫码支付）

- [] JSAPI支付（公众号支付）

- [] 支付结果异步通知处理

- [] 订单查询功能

第三阶段：控制层实现

- [] 7. 创建微信支付控制器 `WechatPayController`，提供以下接口：

- [] `/wechatpay/native` - 生成Native支付二维码

- [] `/wechatpay/jsapi` - 生成JSAPI支付参数

- [] `/wechatpay/notify` - 支付结果异步回调

- [] `/wechatpay/query` - 订单查询接口

第四阶段：业务逻辑集成

- [] 8. 在订单支付成功回调中区分支付类型（1-支付宝，2-微信）

- [] 9. 确保库存扣减、订单状态更新等业务逻辑与支付方式无关

第五阶段：测试验证

- [] 10. 编写单元测试覆盖主要支付场景

- [] 11. 进行沙箱环境测试验证

- [] 12. 更新API文档，添加微信支付接口说明

涉及文件/模块

- `mall-portal/src/main/java/com/macro/mall/portal/config/WechatPayConfig.java`

- `mall-portal/src/main/java/com/macro/mall/portal/domain/WechatPayParam.java`

- `mall-portal/src/main/java/com/macro/mall/portal/service/WechatPayService.java`

- `mall-portal/src/main/java/com/macro/mall/portal/service/impl/WechatPayServiceImpl.java`

- `mall-portal/src/main/java/com/macro/mall/portal/controller/WechatPayController.java`

- `pom.xml` - 添加依赖

Agent Chat

牛万鹏

参考支付宝支付，实现微信支付

Plan

我将按照阶段一的项目阅读流程来开始工作。首先让我构建对项目的全局认知，了解项目结构和现有支付相关的代码。

让我先查看项目根目录的结构：

分析 ~/Workspace/mall-zulu 20 files

我注意到项目中有 wechat-pay-doc 目录，这应该与微信支付相关。让我继续查看项目配置文件来了解技术栈：

分析 pom.xml 289 lines

我看到项目使用了 Spring Boot 2.7.5 和 Java 8，并且已经集成了支付宝支付 SDK（alipay-sdk.version=4.38.61.ALL）。现在让我查看 README 文件了解项目整体信息：

分析 README.md 208 lines

现在让我查看微信支付文档目录，了解现有的微信支付相关资料：

分析 wechat-pay-doc 0 files

现在让我搜索项目中现有的支付相关代码，特别是支付宝支付的实现，以便了解支付模块的结构：

plan.md .comate/plan.md

变更文件 1 全部放弃

plan.md

输入任意需求，使用 # 更快找到目标代码

Plan DeepSeek-V3.1

行 52, 列 18 空格: 4 UTF-8 {} Markdown Comate设置

Comate很早就发布的Plan模式：

- 提出澄清性问题以了解你的需求
- 在代码库中检索并收集相关上下文
- 制定一份完整的实现计划，生成plan.md
- 通过聊天或plan.md文件审阅并编辑该计划
- 准备就绪构建该计划

▶ Agent协同下的研发生态变革

从发展趋势上说

■ 并行化

- ✓ Agent与工程师可以协同，时间上错开并行
- ✓ 多个任务、多个项目并行，实际生产力大幅提高

■ 后台化

- ✓ Agent天然具备多任务同时执行能力，可置于后台工作
- ✓ 后台任务依然需要工程师参与，具备主动通知的能力

■ 远程化

- ✓ 干净、隔离的环境是Agent最佳的执行容器
- ✓ 远程、容器化提供无限的扩展能力，进一步提高软件生产力

从构建一个Agent产品来说

简单的策略远比复杂的更有效

所有复杂的策略都应从具体场景演化

Agent Infra 比Agent 本身更重要

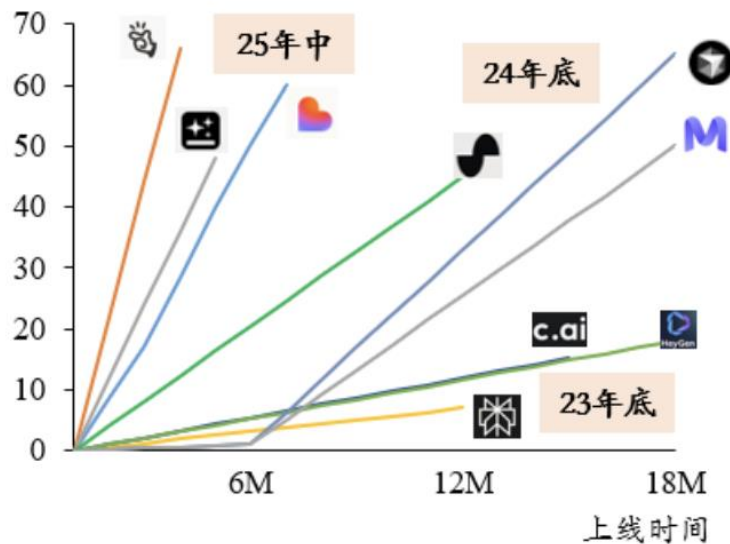


全世界最活跃的资金、最聪明的一批人都在AI、AI Coding领域深耕，没有理由不参与进来。

25年明星应用产品ARR呈爆发态势

AI-Native明星公司ARR

ARR (百万美金)

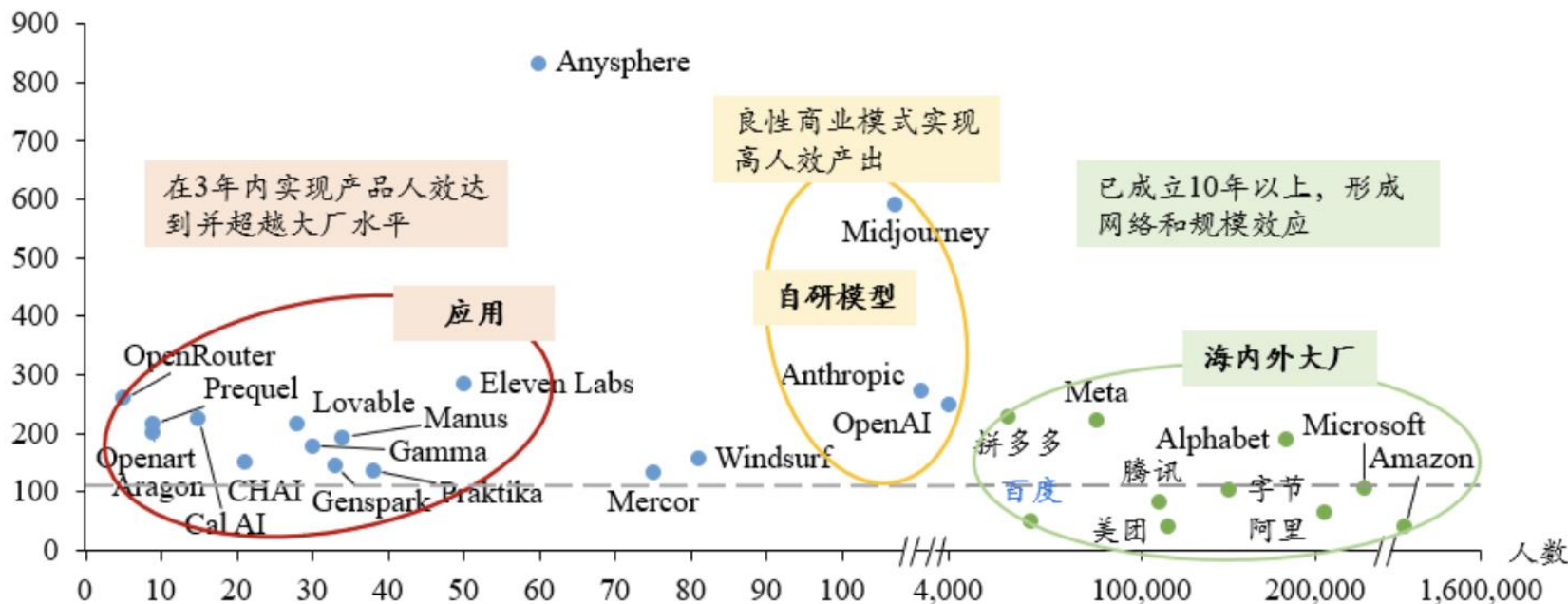


AI-Native明星公司的ARR攀升速度越来越快

在海外大厂的人效中线以上，出现大量三年内<50人收入快速攀升的AI-Native应用公司

海内外大厂和AI-Native公司人效

人效 (万美元/人)



“顶尖开发者”的定义正在被改写

他们不再是“写代码最快的人”

而是“最善于驾驭AI创造价值的人”



科技生态圈峰会 + 深度研习

——1000+ 技术团队的选择



NiDD 峰会

NiDD 峰会 上海站

AI+研发数字峰会

时间: 2026.05.22-23

NiDD 峰会 北京站

AI+研发数字峰会

时间: 2026.08.21-22

NiDD 峰会 深圳站

AI+研发数字峰会

时间: 2026.11.20-21



AiDD峰会详情

NiDD × AI+产品峰会

NiDD × 上海站

AI+产品创新峰会

时间: 2026.01.16-17

NiDD × 北京站

AI+产品创新峰会

时间: 2026.08.23-24



产品峰会详情



2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会



Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战略与商业模式重构

论坛3: Agentic AI产品创新与交互设计

2-hour Speech: 回归本质



用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，
仍然考验“看见”的“同理心”



Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Panel 1: 出海前瞻



“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式



Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

Panel 2: 失败复盘



为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

智能重构产品 数据驱动增长
Reinventing Products with Intelligence, Driven by Data

80+

业界大咖

汇聚产品大咖的真知灼见

40+

创新案例

开拓全新AI+产品视角

10+

分论坛

涵盖产品到商业增长的全链路

800+

听众规模

共同探索产品的智能重构



扫码查看会议详情



第8届 AI+ 研发数字峰会
AI+ Development Digital Summit

感谢聆听!

扫码领取会议PPT资料

