



2025 AI+ Development
Digital Summit

AI+ 研发数字峰会

拥抱AI 重塑研发

05/23-24 | 上海站



2025 AI+研发数字峰会

拥抱AI 重塑研发 AI+ Development Digital Summit

下一站预告

08/08-09 | 北京站

11/14-15 | 深圳站



查看会议详情

北京站论坛设置

大模型和 AI 应用评测

智能存储与检索技术

下一代知识工程

AI+ 金融业务创新

智能需求工程

智能体与研发效率工具

AI 产品运营与出海策略

大模型安全与对齐

大模型应用开发框架与实践

智能体经济 (Agentic Economy)

智能测试工具的开发与应用

具身智能与机器人

代码生成及其改进

AI+ 新能源汽车

AI 前沿技术探索与实践

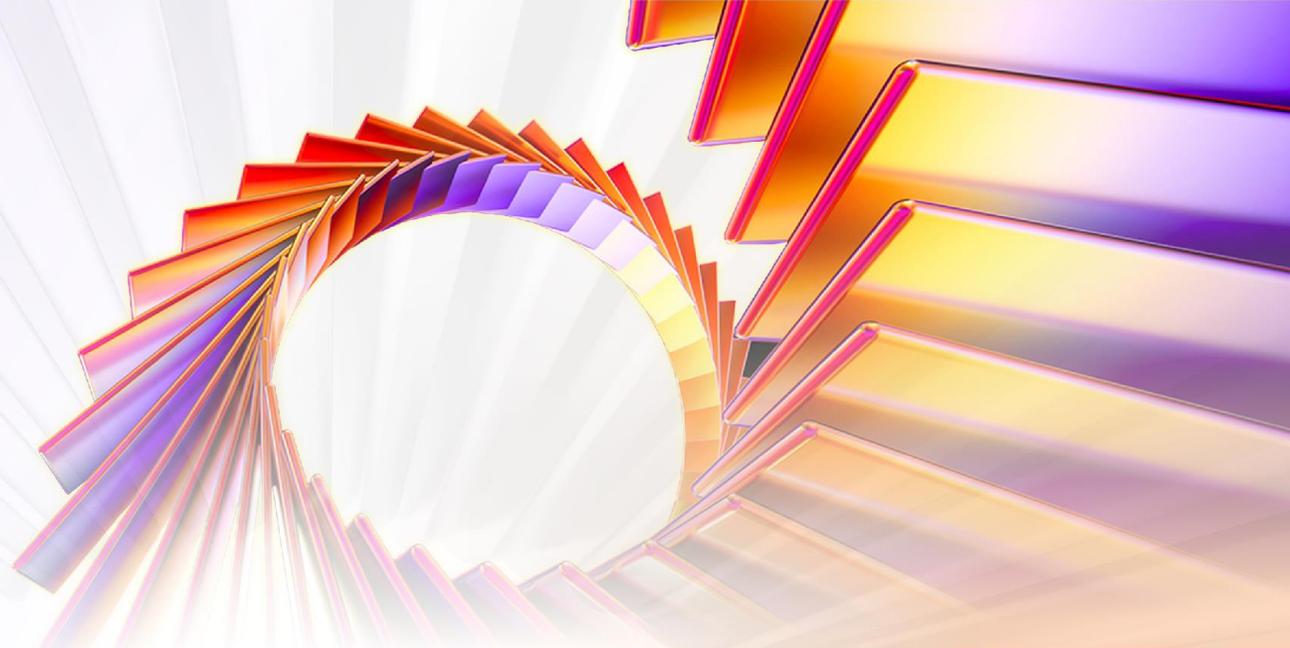


| 05/23-24 | 上海站

2025 AI+ Development
Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发



从需求到代码：DeepSeek+ AI代码助手 的全场景自主编程革命

汪晟杰 | 腾讯云产品专家



汪晟杰

腾讯云代码助手 CodeBuddy/ 产品专家

腾讯资深产品专家，20年工作经验，负责腾讯云开发者AI代码助手产品规划设计与运营，十多年协作SaaS和 SAP 云平台、SuccessFactors HCM、Sybase 数据库、PowerDesigner 等产品的开发经理，在软件架构设计、产品管理和项目工程管理、团队敏捷、AI研发提效等方面拥有丰富的行业经验。

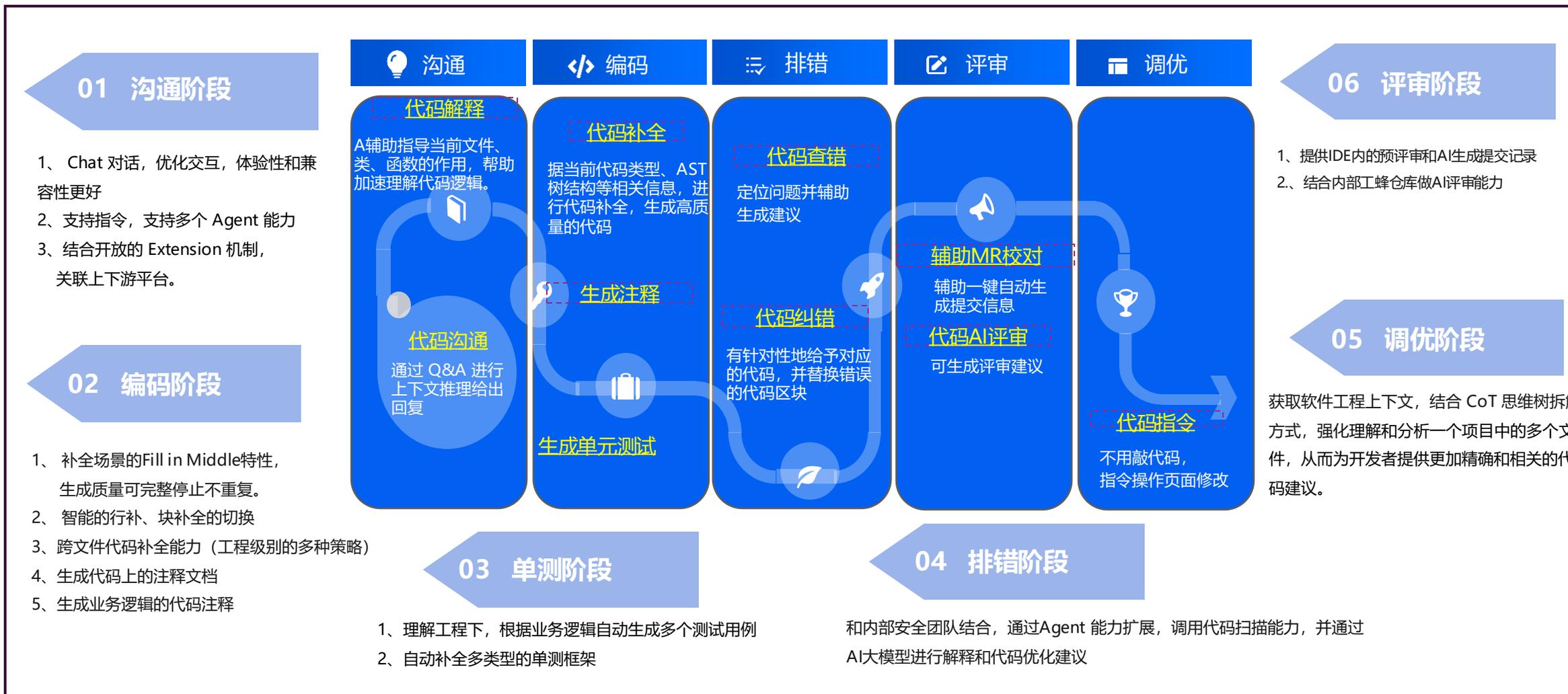
目录

CONTENTS

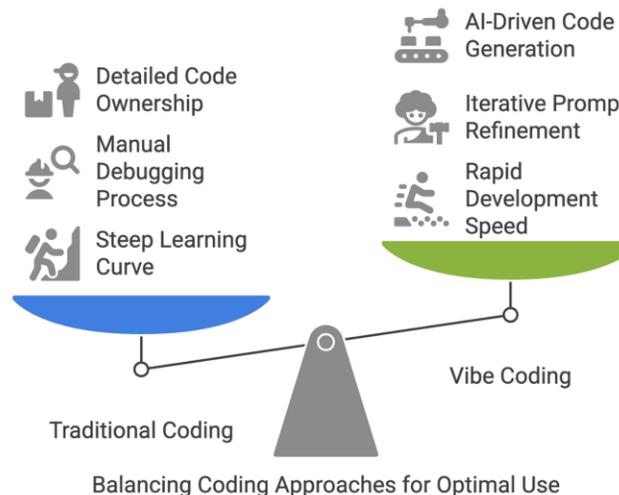
- I. 软件工程与氛围编程
- II. 问题/痛点
- III. 解决思路/整体方案
- IV. 具体实现/技术实践
- V. 总结与展望

PART 01

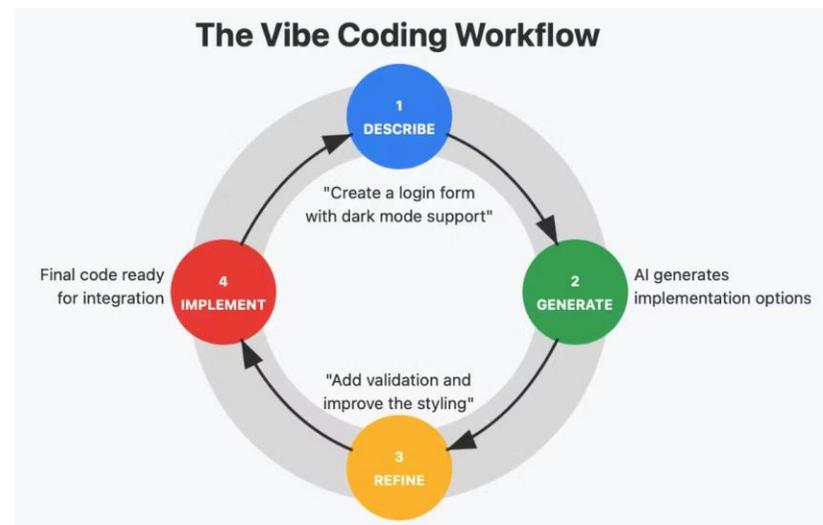
软件工程与氛围编程



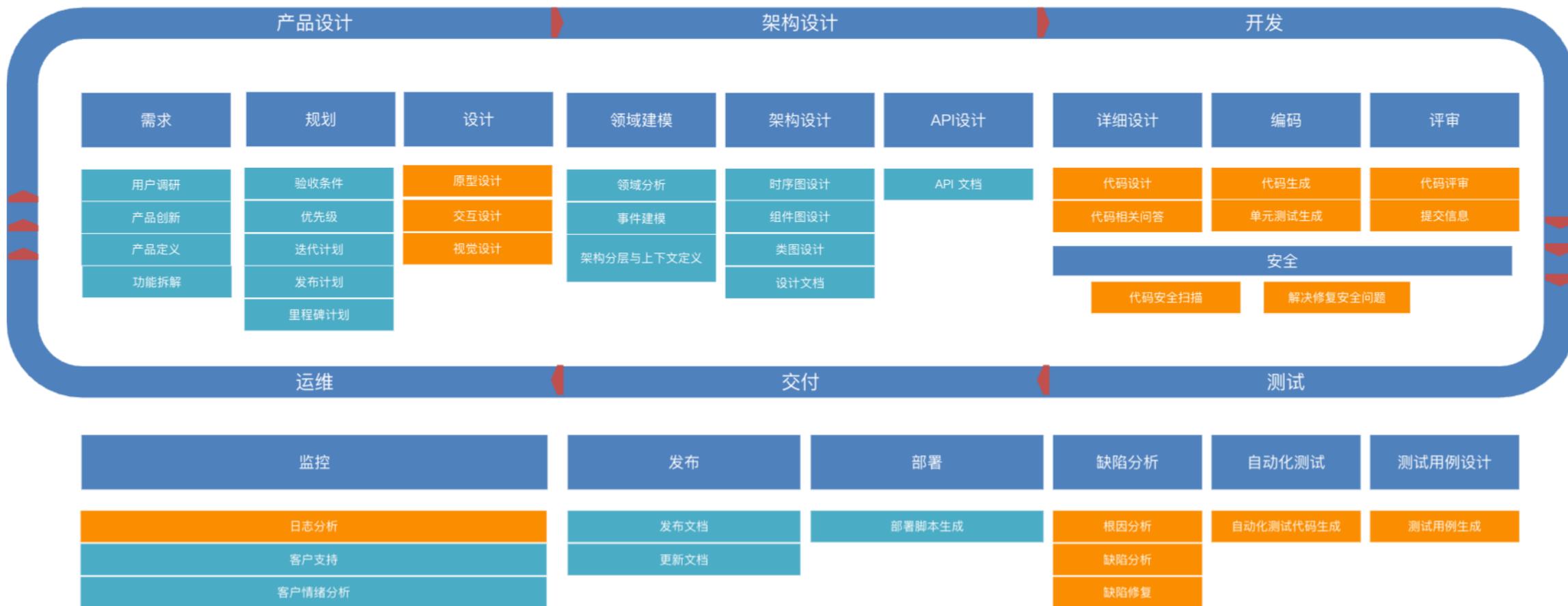
氛围编程 Vibe Coding



通过 AI 的需求沟通，完成工程级别的应用



产设研测智能化道路



PART 02

全场景的问题和痛点

▶ Vibe Coding 带来的问题

用 cursor 写代码，越写越烂，怎么办？

用Cursor Debug 到怀疑人生？

指令多加一句话，Debug不在话下！

右滑查看

用 cursor 写代码，越写越烂，怎么办？ 尝试使用 cursor ...

趣游拾荒客 03-21 421

用 Cursor Debug 越改越崩，怎么破？ 最近用 Cursor 编...

智学计划 04-29 29

Cursor 因为 Claude-3.7 变成弱智

对比之前 Cursor 因为 Claude-3.7 变成弱智 今天...

RainSun321 02-28 24

不要让 cursor 重构代码!!!

本以为 Claude sonnet 3.7...

不想做 NPC 的玩家 03-08 132

被自家AI坑惨！**一句误导引发程序员退订潮**，Cursor CEO 亲自道歉

Cursor 反复试错后的编程指南 最近 Cursor 写代码太丝滑了...

两顿不吃饿得黄同学 2024-10-04 28

Cursor你不要太离谱，这才19号，我的经费在燃烧

Cursor你真的不要太离谱，花钱如流水啊 这才3月19...

凤凰于飞 03-19 41

Cursor 正在制造大量垃圾 APP 应用

Cursor 正在制造大量垃圾 APP 应用？ 最近刷到好多关...

Kenny 做产品 03-03 1144

灾难级体验 建议现阶段退订Cursor

和Cursor说再见了👋，Augment

建议现阶段退订Cursor 昨天出了Claude-3.7，兴冲冲地...

越改越错

删除我的已有代码

生成久，浪费时间经费

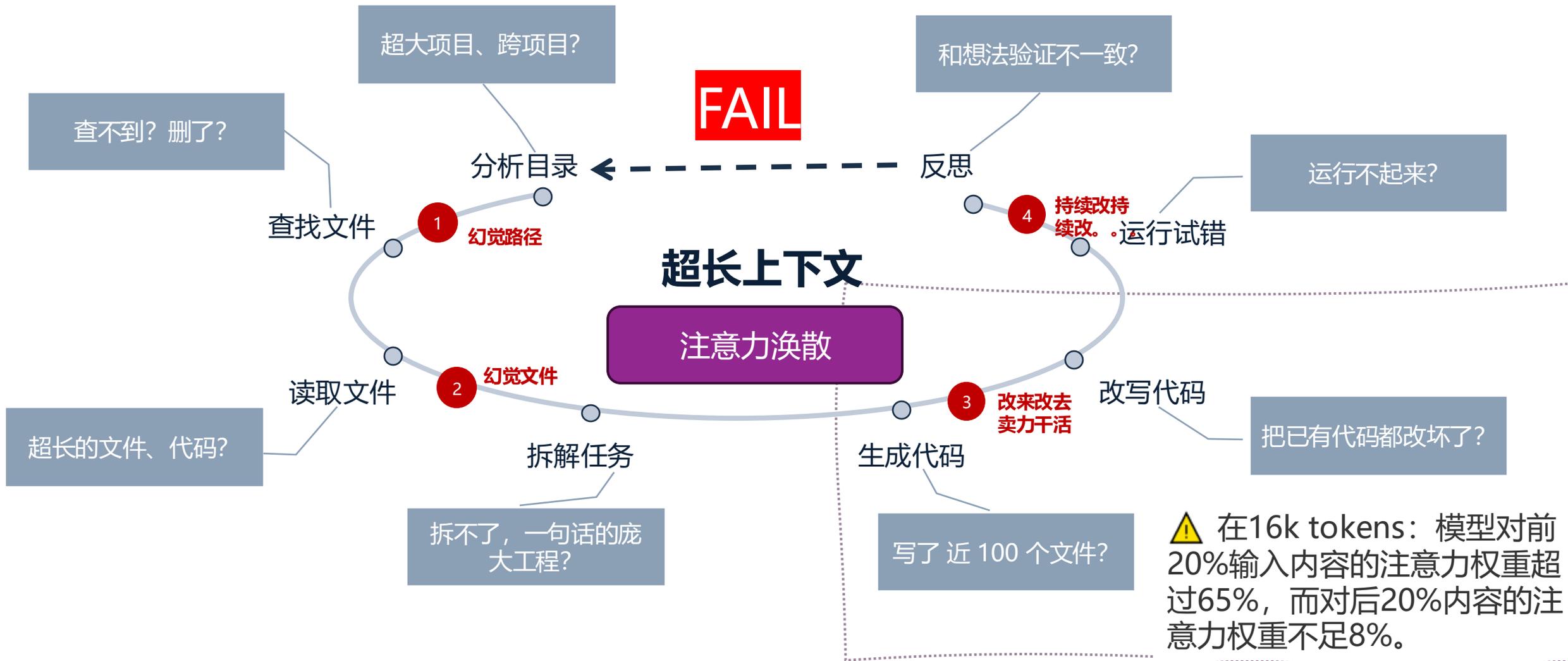
不知道怎么办了



需求到应用交付



Agentic Workflow



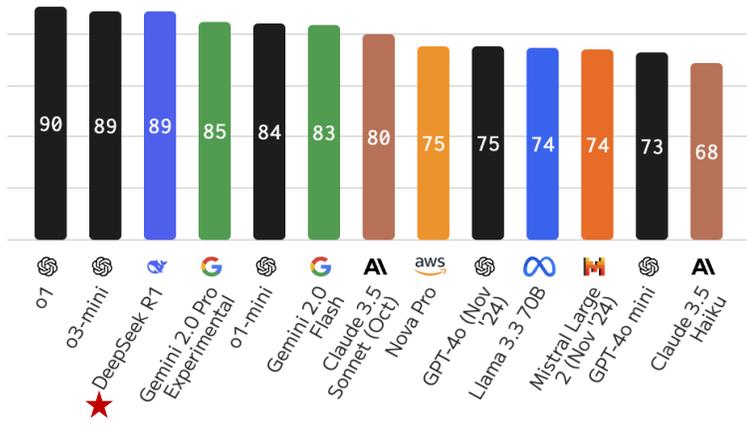
PART 03

解决思路和整体方案

▶ 1. 选用更好的模型 - DeepSeek

QUALITY

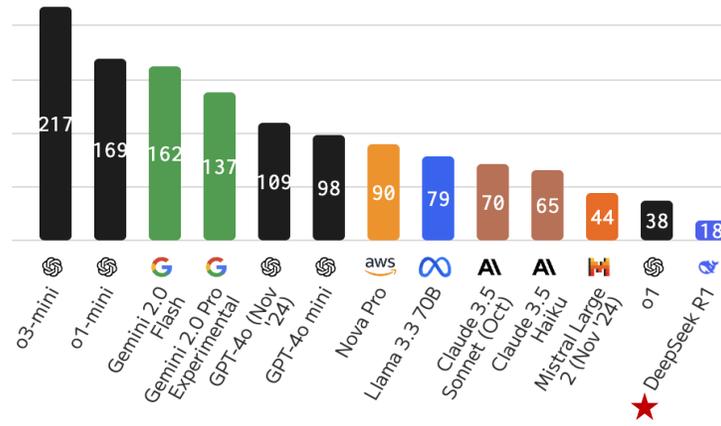
Artificial Analysis Quality Index; Higher is better



R1在效果上与o1和o3评分不相上下

SPEED

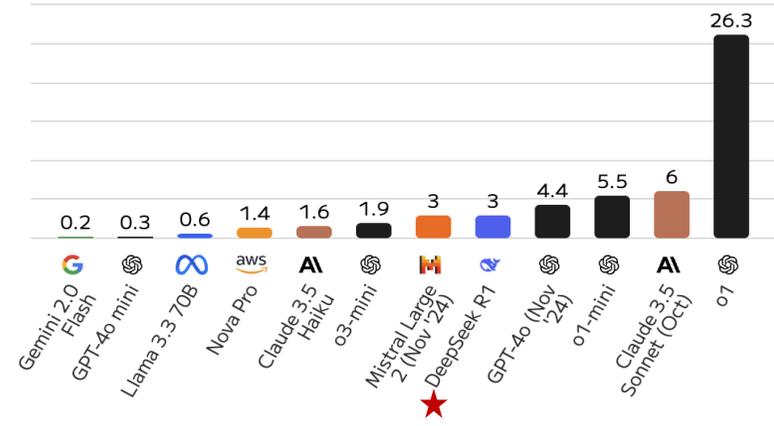
Output Tokens per Second; Higher is better



o3-mini token输出速度比o1快6倍, 比R1快12倍

PRICE

USD per 1M Tokens; Lower is better



o3-mini价格最低, 是R1价格的63%, 是o1价格的7%

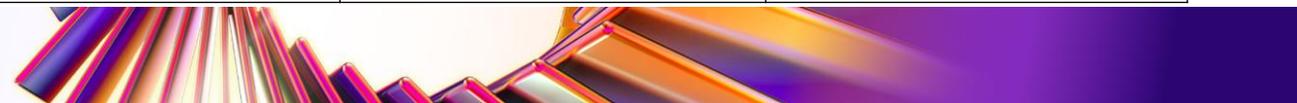
Model †	Arena Elo †	Coding Elo †	Arena Hard †	MMLU †	Votes †	Organization †	License †
🏆 Gemini-2.0-Flash-Thinking-Exp-01-21	1383	1359			10314	Google	Proprietary
🏆 Gemini-2.0-Pro-Exp-02-05	1378	1371			8007	Google	Proprietary
🏆 ChatGPT-4o-latest (2024-11-20)	1365	1352			38396	OpenAI	Proprietary
🏆 DeepSeek-R1	1362	1363			4193	DeepSeek	MIT
🏆 Gemini-2.0-Flash-001	1357	1348			5919	Google	Proprietary
🏆 o1-2024-12-17	1351	1362	90.4		12241	OpenAI	Proprietary
🏆 Qwen2.5-Max	1332	1339			3766	Alibaba	Proprietary
🏆 DeepSeek-V3	1317	1317			16456	DeepSeek	DeepSeek
🏆 o3-mini	1310	1356			3379	OpenAI	Proprietary
🏆 Gemini-2.0-Flash-Lite-Preview	1306	1309			5234	Google	Proprietary

在Chatbot Arena“众测”上, 用户对DS-R1的倾向程度处于第一梯队。
在自有真实2B测试集: DS-R1的意图识别和多轮有优势

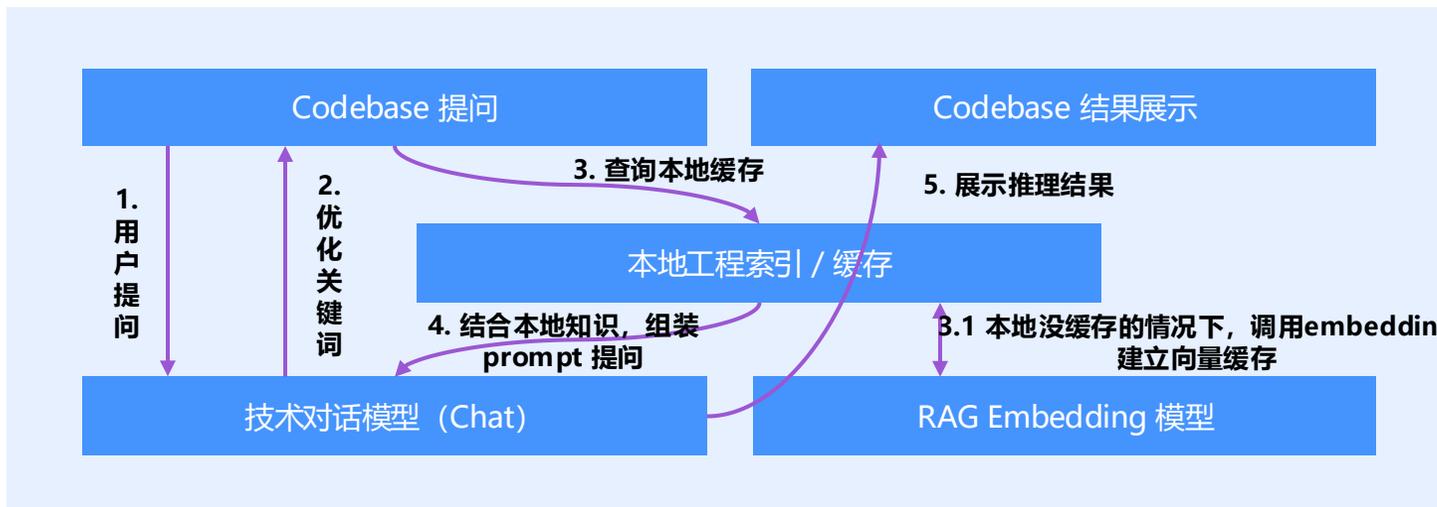


2. 聚焦代码的效果和准度 – 降低反思率

	准确性	全面性	可读性	实用性	不足
代码解释	R1代码逻辑分析能力很强，能结合整个工程的代码文件来深入理解代码逻辑	R1除了解释代码的关键逻辑之外，还描述代码的设计特点，提出代码优化建议，全面性很高	R1解释代码突出关键逻辑，并用简洁明了的注释逐点阐述，可读性很高	针对历史悠久、逻辑复杂、理解成本很高的代码，R1能结合整个工程的代码，给出准确且合理的解释，极大辅助研发快速理解代码，实用性很高	没有提及明显不足
代码生成	通过简单的需求描述，R1就能理解需求的关键，并生成基本可用的代码，准确率很高	R1生成的代码在性能、错误处理，日志打印方面考虑周全，此外代码结构清晰，注释详细		生成的代码基本能运行，实用性很高	私域知识不足，比如trpc-go
代码注释	R1结合整个工程的代码，准确分析代码逻辑，并给代码的关键逻辑添加注释，从12个case看，注释的准确率几乎接近100%		1.R1在函数定义上方清晰说明函数的作用，参数和返回值 2.R1没有通篇添加注释，而是在关键逻辑添加注释，注释简洁清晰，放在代码上方 3.对于原来描述不清晰的注释也会优化成更易理解的注释	R1提供的注释符合Go/js等开发语言的注释规范	R1修改原有注释可能会丢失关键信息
单元测试	R1能准确分析代码的逻辑，基于逻辑分支确定测试用例的关键点			1.R1采用主流的单测框架编写测试用例 2.测试覆盖率基本在90%以上 3.测试用例数据真实有效 4.生成的单测代码可直接使用 5.针对外部依赖提供mock实现	针对复杂的代码，生成的单元测试用例代码不一定能直接运行



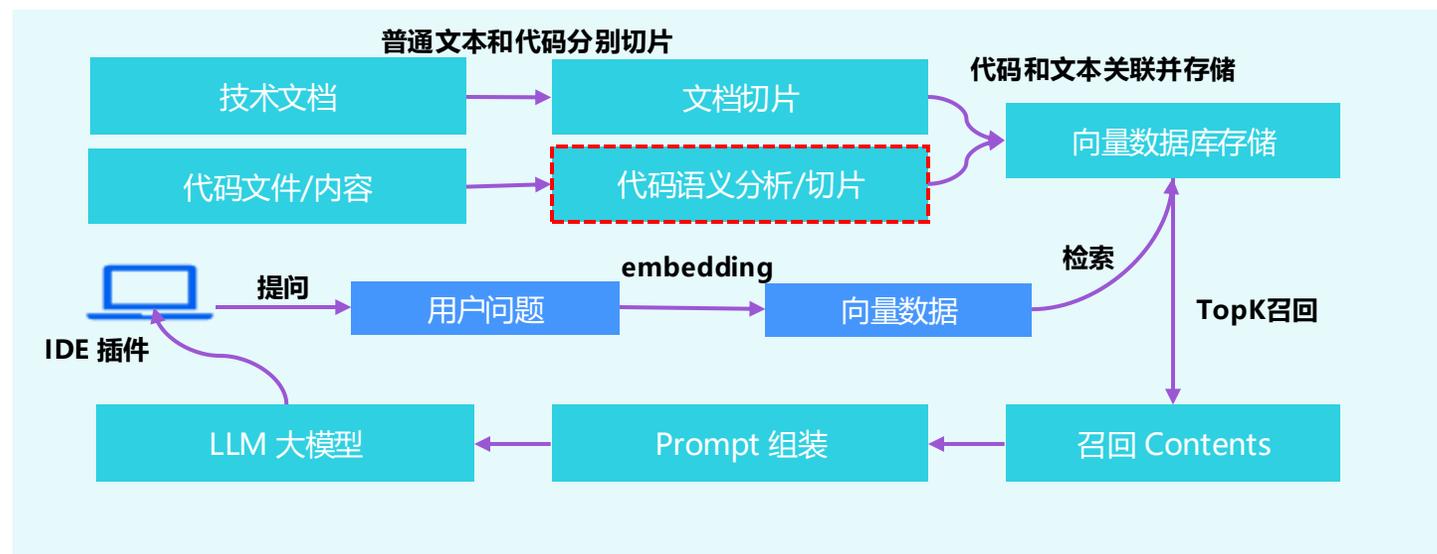
3. 知识库和工程理解 - 上下文



Codebase 本地工程 => 云端工程

通过 Codebase 可以**索引整个代码库, 理解项目结构、文件依赖和代码逻辑**, 提供基于全局上下文的智能建议。典型场景有:

1. 旧项目功能理解: 可以针对某个具体功能, 通过 Codebase 询问功能实现代码文件、业务流程和影响范围修改。
2. 代码架构升级: 比如 Spring 2.x 升级到 3.x 场景, 可以通过 Codebase 提供全面的升级分析和建议。



RAG 知识库

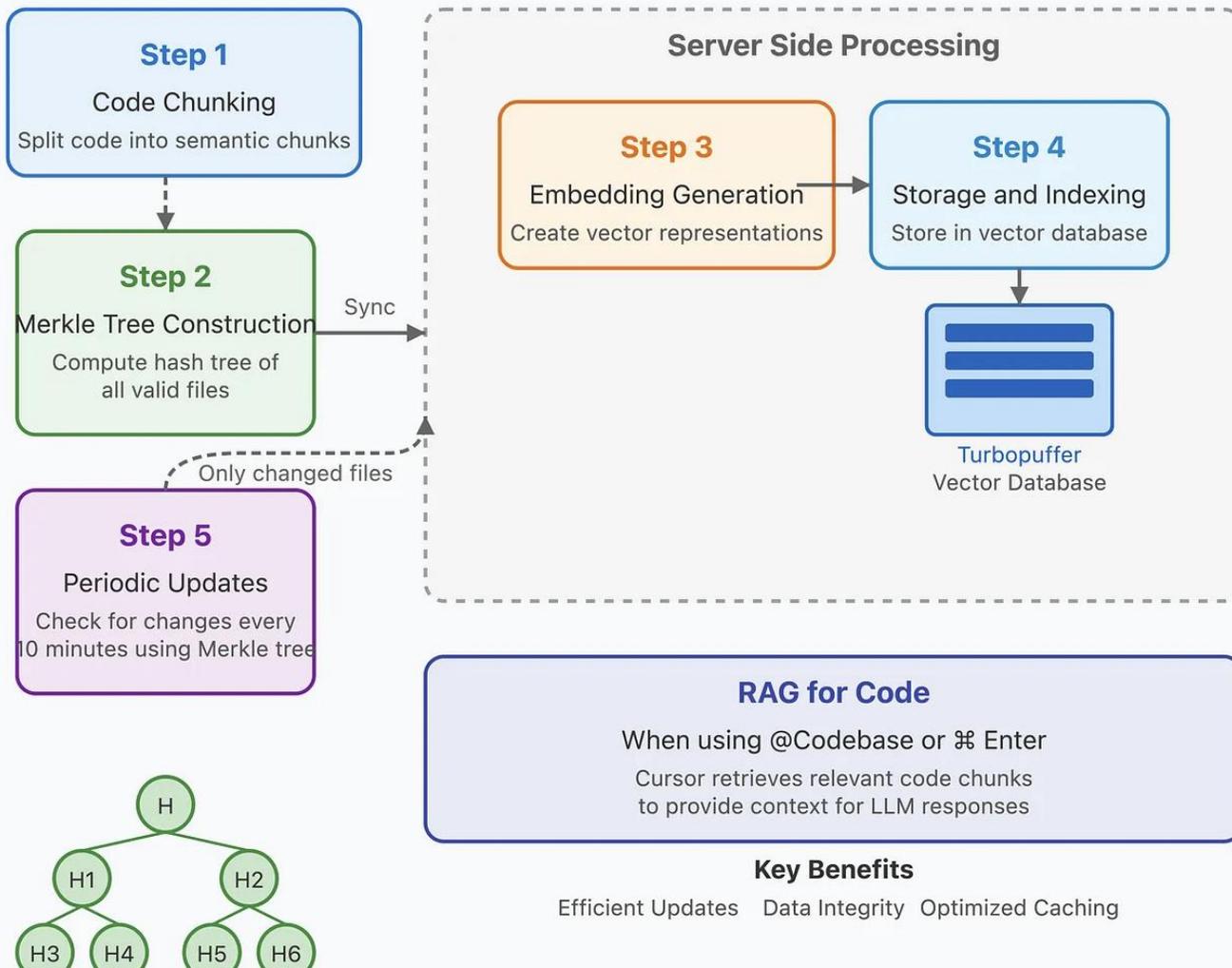
在 RAG 知识库的建设上, 产品核心提供了**代码内容 RAG** 的能力, 针对代码内容, 可以结合代码的语法语义逻辑进行切片与向量存储。实现通过自然语言召回相关代码。典型场景有:

1. 企业内部的业务 API 知识库
2. 企业某项业务专有逻辑, 比如金融结算
3. 业务 SDK 的使用说明知识库
4. 企业内部的标准规范知识库
5. 一些复杂的架构框架知识库, 比如 SpringAI、UE知识库等



4. 更好的代码索引

Cursor's Codebase Indexing Pipeline Using Merkle Trees



工程分析+Merkle 树

高效拆分和增量检索

高效拆分和增量检索

Codebase 意图识别



5. 单测结构化

测试任务规划

选择文件/目录生成单元测试

自动收集代码上下文逻辑

生成丰富的测试用例列表

测试用例可自定义修改/扩展

单测代码生成

逐用例生成对应单测代码

感知上下文并注入Mock逻辑

可自定义描述优化单测结果

代码应用或插入到代码文件

执行与数据统计

单测代码执行

执行失败与重试

自定义内容, 优化单测逻辑

测试报告展示

腾讯云 AI 代码助手对话

对话 评审 单元测试

生成中 ...

✓ 正在收集上下文信息

🔄 正在生成单元测试计划列表

对话 评审 单元测试

已为 RegionController.java 生成 16 个测试用例

> Verify that the returned list contains exactly 7 regions ✓

> Verify that the method returns a non-null list

> Verify all expected regions are present in the list

Verify that the returned list contains exactly 7 regions ✓

RegionControllerTest.java

```
package com.project.estateapi.rest;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.InjectMocks;
import org.mockito.junit.jupiter.MockitoExtension;
import static org.junit.jupiter.api.Assertions.assertEquals;

import java.util.List;

@ExtendWith(MockitoExtension.class)
public class RegionControllerTest {

    @InjectMocks
    private RegionController regionController;

    @Test
    public void getRegions_ShouldReturnListWithSevenRegions() {
        // When
        List<String> regions = regionController.getRegions();

        // Then
        assertEquals(7, regions.size(), "The list should contain exactly 7 regions");
    }
}
```

输入描述继续优化上述测试

Should correctly handle a word with repeated letters

manifest_test1.json 重新运行中...

⏹ 停止

```
const gulp = require('gulp');
const util = require('./build/lib/util');
const path = require('path');
const compilation = require('./build/lib/compilation');

// Fast compile for development time
gulp.task('clean-client', util.rimraf('out'));
gulp.task('compile-client', ['clean-client'], compilation.compileTask('out', false));
gulp.task('watch-client', ['clean-client', 'compile-client'], compilation.watchTask('out', false));
```

输入描述继续优化上述测试用例

查看失败原因 查看运行报告

< 2 / 2 >



▶ 6. Vibe Coding 的进化论

Vibe Coding 本质还是 Coding

-- 1. 立范围

- 你是XXX专家, 你有YYY的技能
- 你需要在 XXX 的目录下改动
- 小技巧: 改写生成代码的时候, 勤@Add 某个文件。

-- 2. 述问题

- 问题是什么: 首先你要告诉模型你的问题是什么, 你的任务是什么, 要尽量描述清楚你的需求。
- 小技巧: 描述清楚任务 1、2、3, 妙用 Chat 能力增强需求描述。在需要拆解目标的情况下, 可优先开启需求计划的计划模式, 让 AI 先执行需求拆解。

-- 3. 定目标

- 你要做什么: 下面你需要告诉大模型具体要做什么, 比如做一份攻略, 写一段代码, 对文章进行优化, 等等。
- 小技巧: 用 Chat 来一轮目标描述扩写, 生成要做的事情的细节。巧用脚手架命令事先生成, 让 AI 做基础上的改写。

- 4. 提示例

提出比如输入数据格式、输出数据格式和期望改的示例结果

- 5. 补要求

有什么要求: 最后我们往往还需求对任务补充一些要求, 比如按特定格式输出, 规定长度限制, 只输出某些内容, 等等。

小技巧: 写好.md 文件, 用 1、2、3 描述, 放在根目录.rules 下 (vscode 有效)。可以从网上找一些针对语法和逻辑的最佳实践的 rules

产品优化及改进

-- 1. 【拆任务步骤】强化需求计划

- 新需求有效拆解, 分析, 不着急动手

-- 2. 【意图识别能力】精简+工程理解

- 更好的意图识别算法, 理解背后的想法
更小更优的改动机制和确认回滚机制
更丰富的 @Add Context

- 3. 【Rules】

Rules 开放设计交给人, 约定 AI 怎么生成

- 4. 【推理速度】和【反思机制】

控制改动力度和范围
运行反思更灵敏

5. 【扩展机制】和【上下文压缩】

更智能的精简压缩轮次和上下文
更内聚多种工具和开放能力



▶ 产品及市场路线图



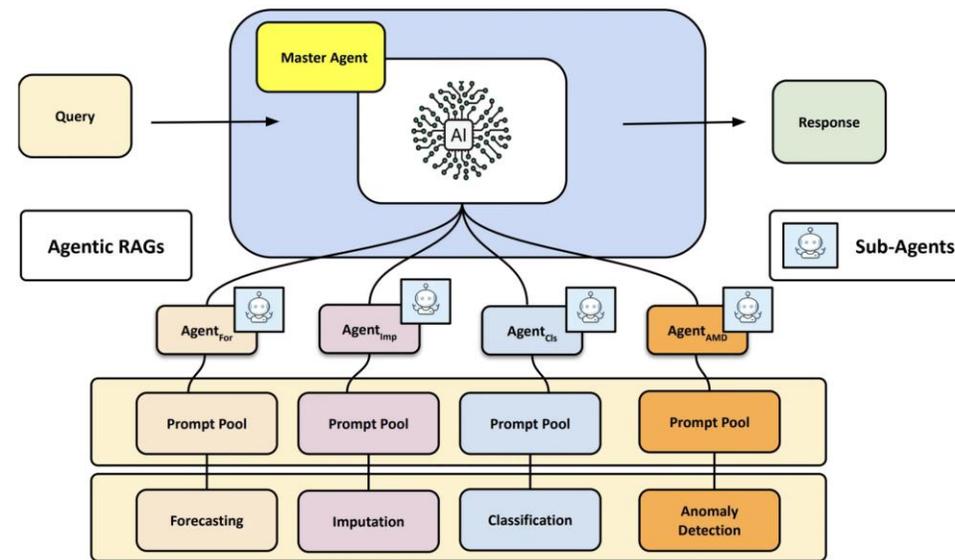
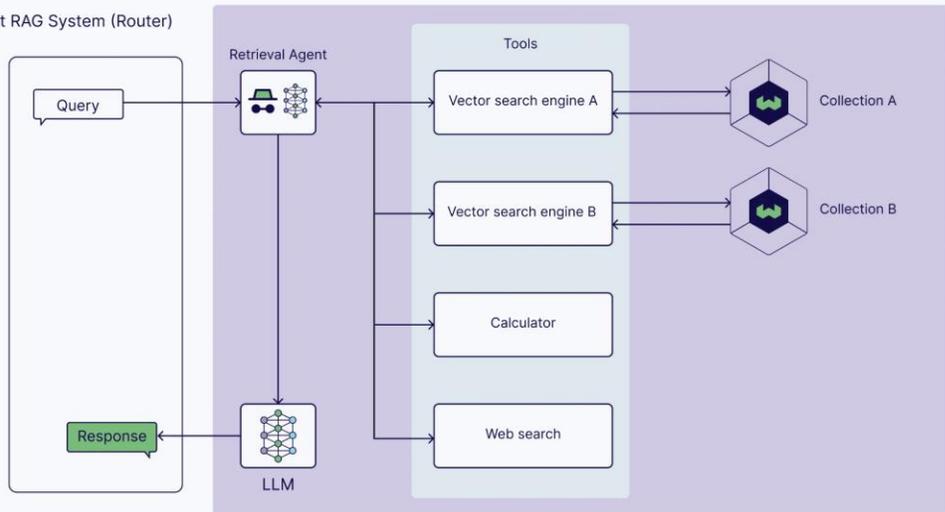
PART 04

具体实现/技术实践

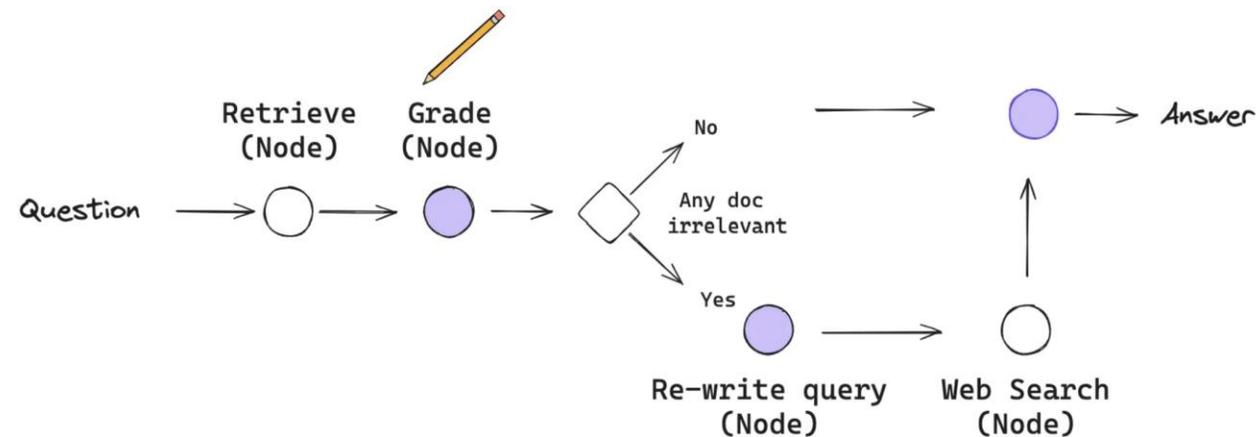
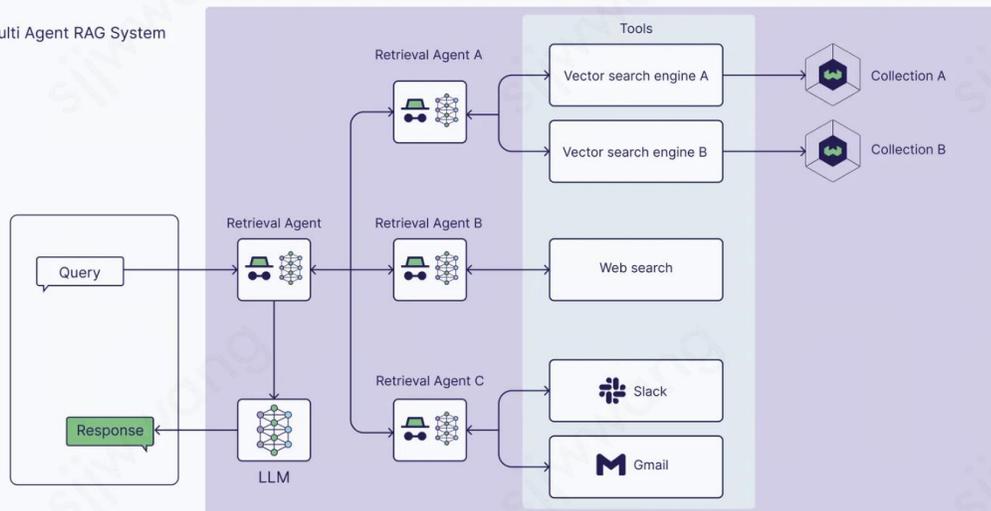
Agentic Everything
= 一切交由模型决策

1. Agentic RAG

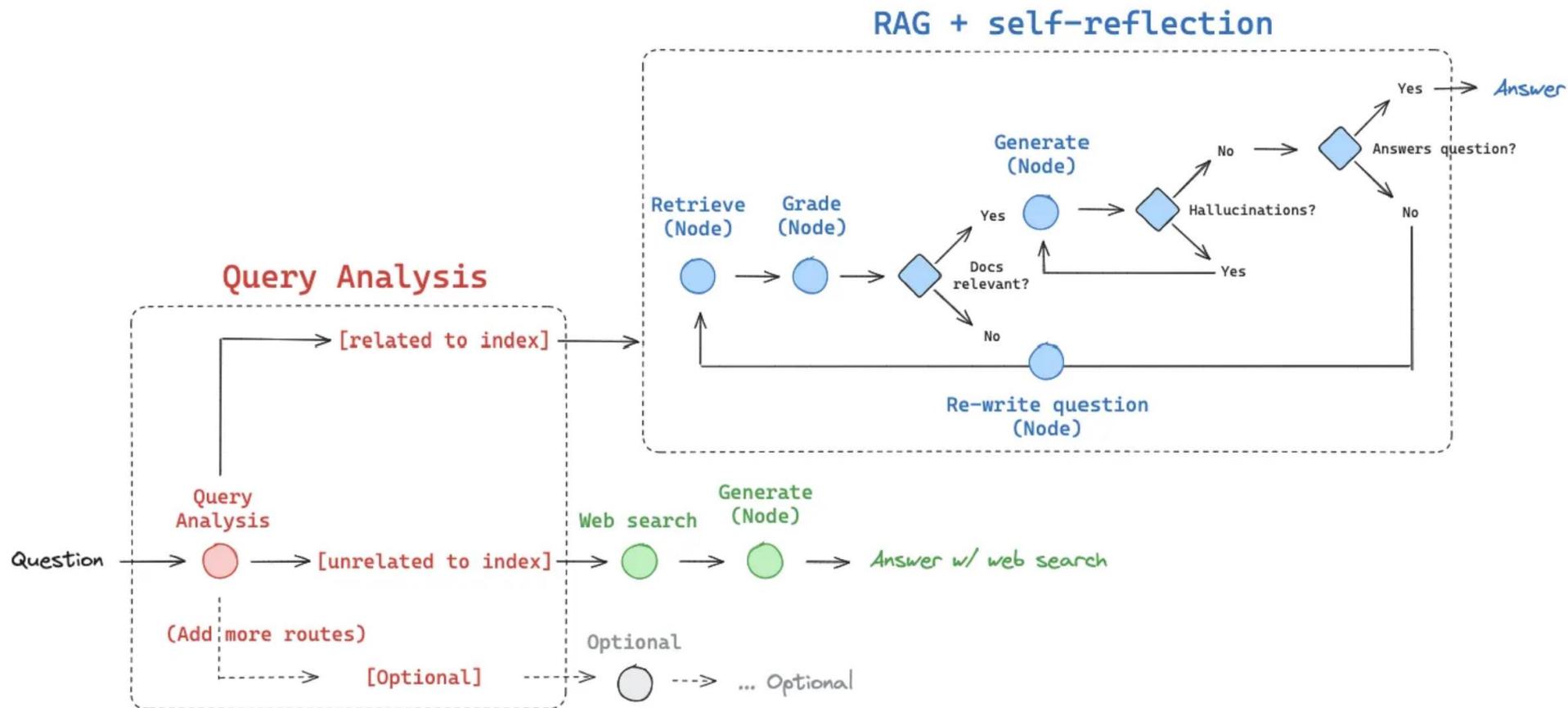
Single Agent RAG System (Router)



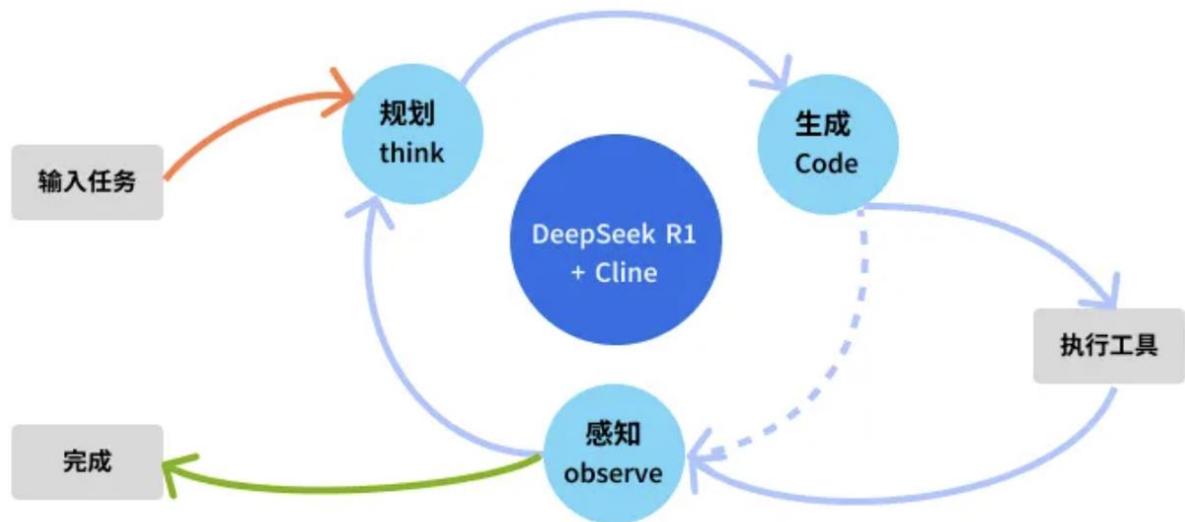
Multi Agent RAG System



2. Agentic RAG – Rewrite+self-reflection



▶▶ 3. Agentic Coding



消耗量大

反思失败

改写异常

时间久

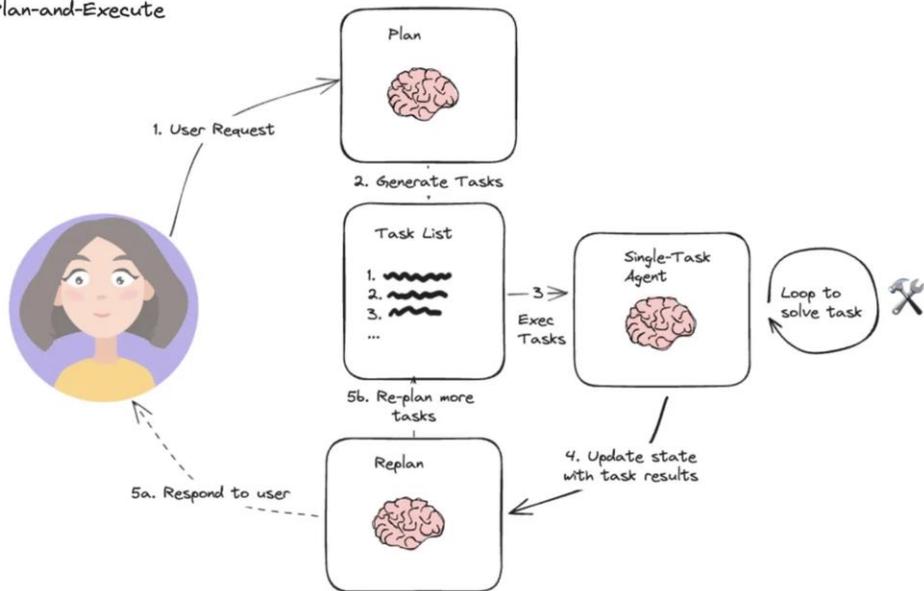
产品类型	产品名称	用量	消费总额 (元)	消费时间段
对话模型	deepseek-ai/DeepSeek-R1	12740 Token(s)	0.0706	21:00-21:59
对话模型	deepseek-ai/DeepSeek-R1	95414 Token(s)	0.4177	22:00-22:59

源码: <https://github.com/cline/cline/blob/main/src/core/prompts/system.ts>



4. Agentic Planning

Plan-and-Execute



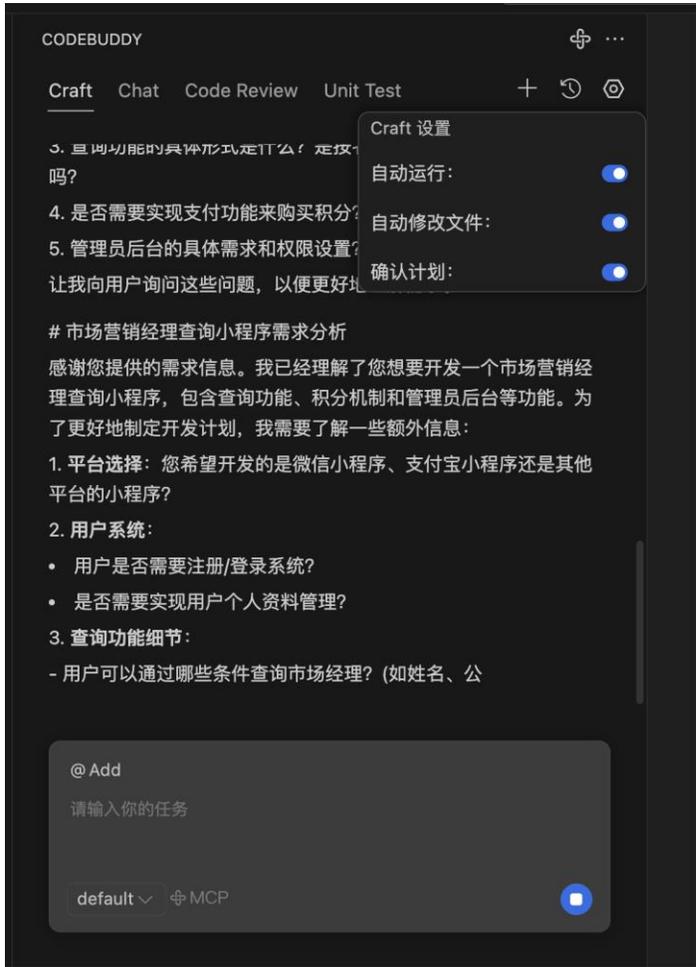
工具调用

协作

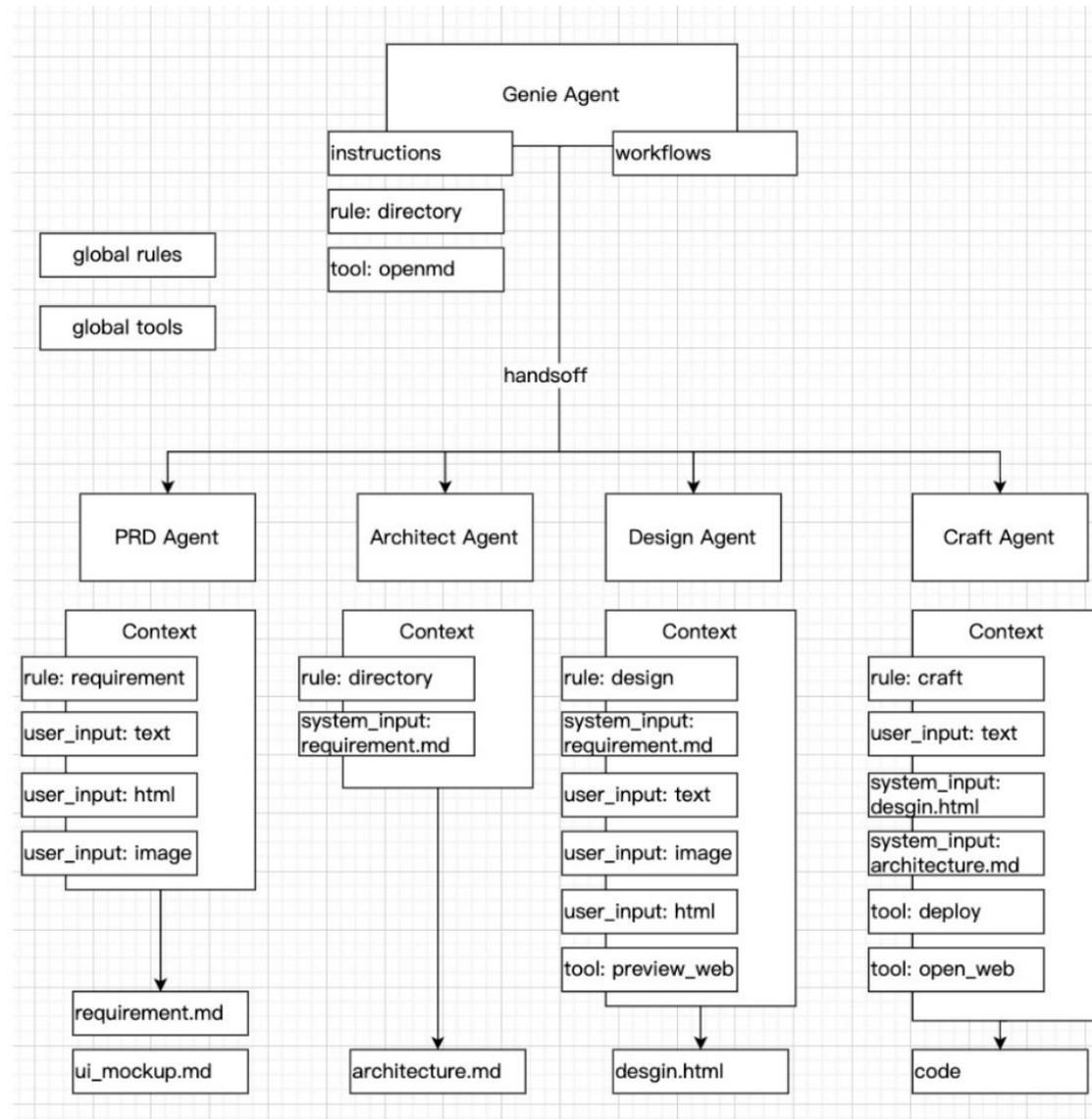
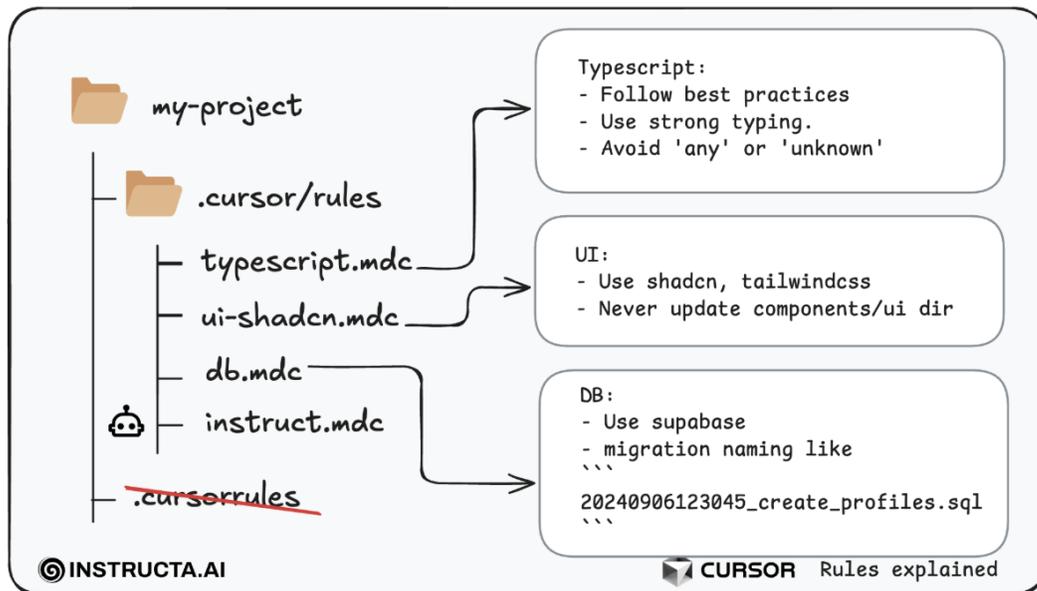
规划

反思

Token / 工具 / 上下文压缩



5. Rules – AI 指导书



6. 接入 MCP 服务 – 最后一公里交给专业的工具



▶ 5. 软件工程模型 和 心流感知



★目前 Windsurf 推出了SWE-1 系列的三款模型：

- ● SWE-1：接近前沿大模型（如 Claude 3.5 Sonnet）的推理能力，但成本更低，现对付费用户限时免费。
- ● SWE-1-lite：中型模型，质量优于 Cascade Base，所有用户均可无限免费使用。
- ● SWE-1-mini：极小型、超快模型，为所有用户提供被动体验。

★核心理念

SWE-1 不只是写代码，更能理解和协作软件开发的各个环节，包括终端操作、测试、还要在终端上工作，获取其他知识和互联网用户反馈等。它通过“flow awareness”（流程感知）与用户无缝协作，能在任务未完成时与人类自然切换，持续推进项目。

所以在评测效果的时候，要关注在流程场景下。



PART 05

总结与展望

Microsoft Build – Copilot Coding Agent

GitHub Copilot: Meet the new coding agent

Implementing features has never been easier. Just assign a task or issue to Copilot. It runs in the background with GitHub Actions and submits its work as a pull request.

GitHub Copilot CODING AGENT



```
components/ui/FlipNumber.tsx Outdated
9 9  onPress?: () => void
10 + testID?: string

timrogers 8 hours ago
Let's make the testID mandatory so it's easier to write automated tests

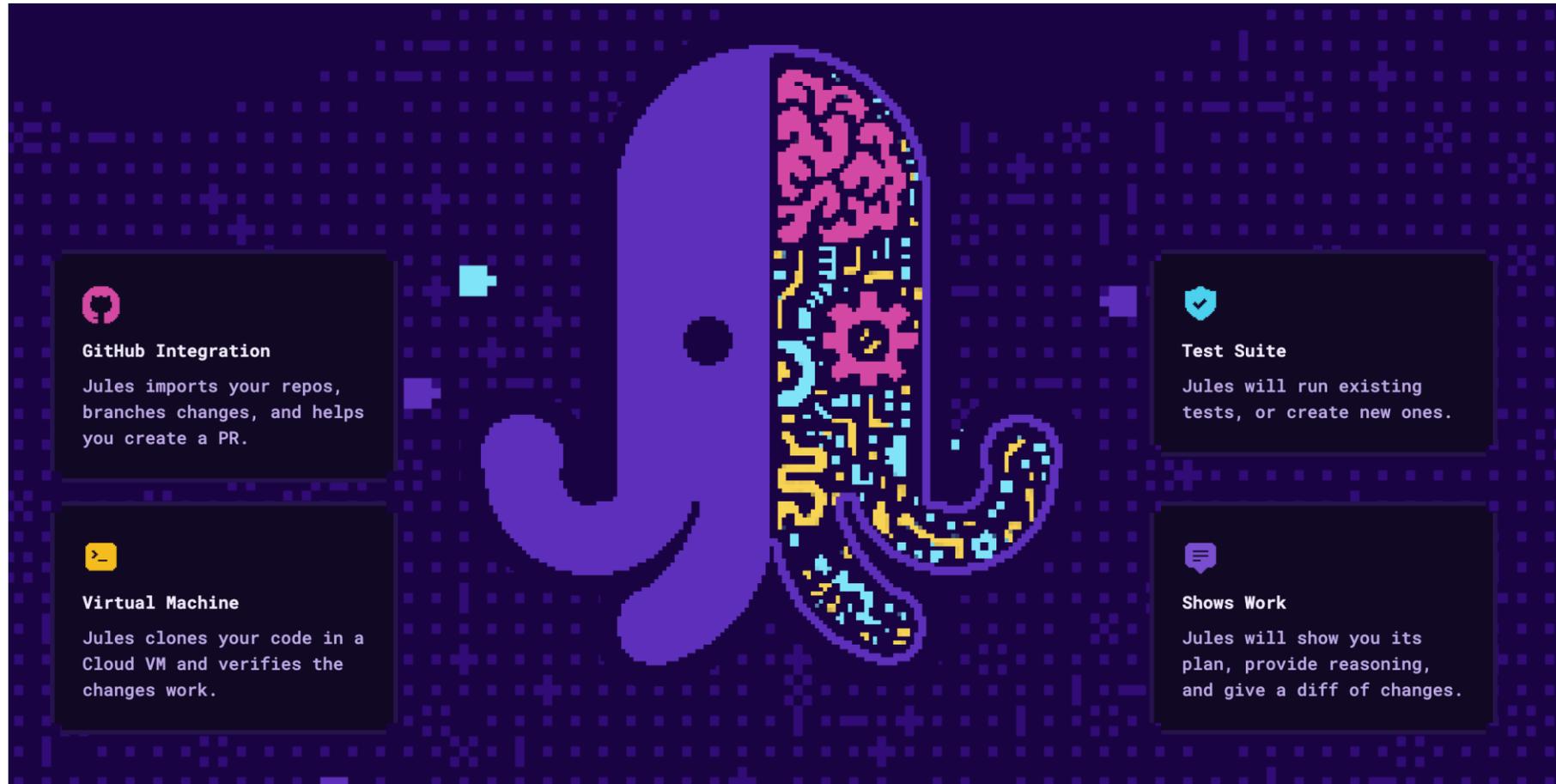
Copilot AI 8 hours ago Author
I've made the testID prop mandatory as requested. This will ensure all instances of the FlipNumber component have a testID, making it easier to write automated tests. The existing usage already includes a testID so there were no breaking changes. Commit: b388336

Help improve Copilot by leaving feedback using the 👍 or 👎 buttons

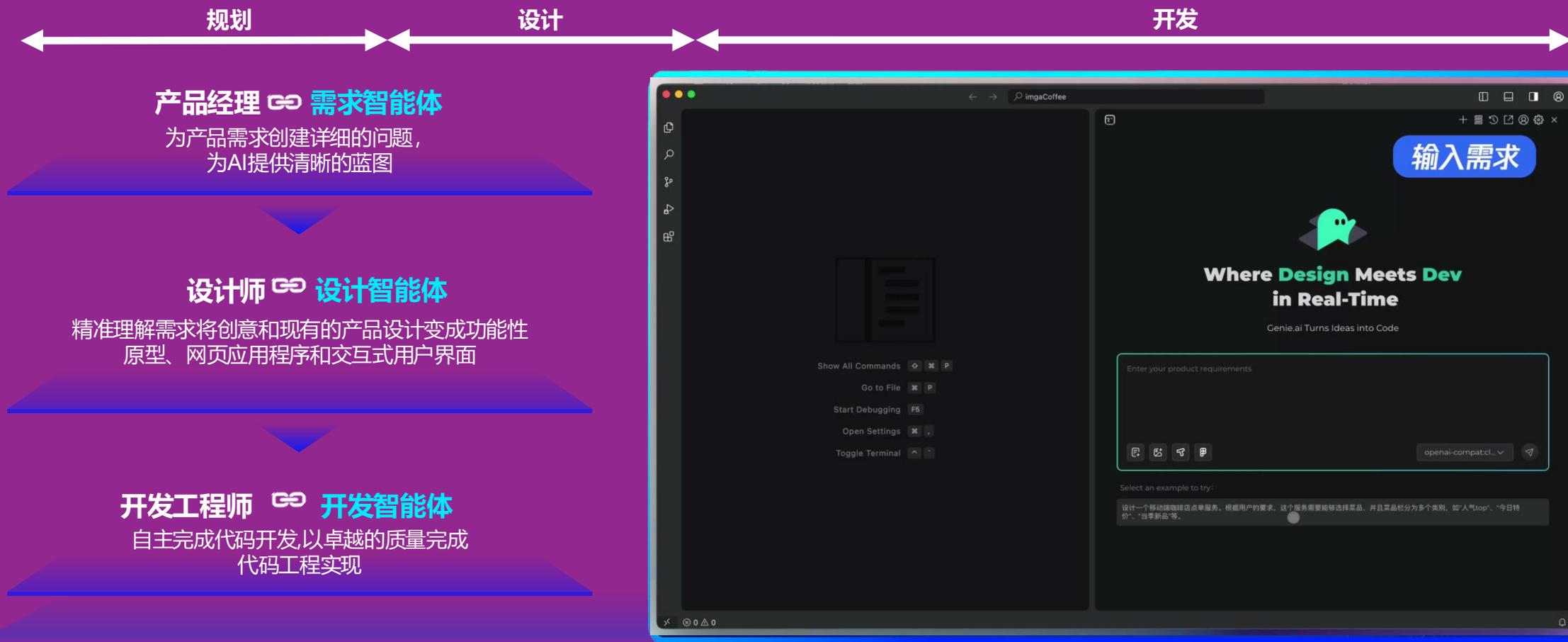
Reply...
```

```
3 import {
4   SparkApp,
5   PageContainer,
6   Input,
7   Button,
8   Card,
9   Checkbox,
10  DropdownMenu,
11  DropdownMenuTrigger,
12  DropdownMenuContent,
13  DropdownMenuItem,
14  Dialog,
15  DialogTrigger,
16  DialogContent,
17  DialogHeader,
18  DialogTitle,
19  DialogDescription,
20  DialogFooter,
21  DialogClose
22 } from "@github/spark/components";
23 import { useKV } from "@github/spark/hooks";
24 import { Plus, ListPlus, CaretDown, Tag, Trash } from "@phosphor-
25
26 // Main App component
27 function App() {
28   // State management using Spark's KV store
29   const [todos, setTodos] = useKV("todos", []);
30   const [newTodo, setNewTodo] = React.useState("");
31   const [selectedCategory, setSelectedCategory] = React.useState(
32
33   // Get unique categories from todos
34   const categories = React.useMemo(() => {
35     const cats = new Set(todos.map(todo => todo.category));
```

▶▶ Google I/O Jules – Google Coding Agent

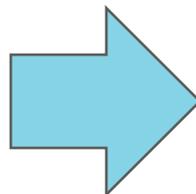


即将发布 AI IDE，是具备
规划、设计、开发能力的一体化 AI 原生开发平台



2024-2025: Agentic Workflow

- SWE Benchmark
- 突破插件形态, 自主形态的 AI Native IDE
- Agentic 基建



2025-未来: AGI

- L1: 聊天机器人 - 具有对话语言的人工智能。
- L2: 推理者 - 具备人类水平的问题解决能力。
- L3: 代理者 - 可以采取行动的系统。
- L4: 创新者 - 可协助发明的人工智能。
- L5: 组织者 - 能够完成组织工作的人工智能。



PART 06

氛围编程实战

代码大模型与编程智能体论坛

CodeBuddy的Vibe Coding之旅 ——在AiDD峰会体验下一代沉浸式编程

主题亮点

“当代码脱离屏幕，编程成为一场与环境的对话。”

在这场 15 分钟的现场 Demo 中，CodeBuddy 团队将打破传统 IDE 的边界，通过 AR 交互、AI 实时协作与环境感知技术，带你进入“氛围编程”（Vibe Coding）的未来场景——用声音、手势和空间逻辑，像指挥交响乐一样编写代码。



汪晟杰

腾讯云资深技术产品专家

参与调研您将优先获得



AiDD定制版
《AI+软件研发精选案例》



专属学习顾问
1对1需求对接

AiDD会后小调研

AiDD峰会致力于协助企业利用AI技术深化计算机对现实世界的理解，推动研发进入智能化和数字化的新时代。作为峰会的重要共建者，您的真知灼见对我们至关重要。衷心感谢您的参与支持！

2025 AI+研发数字峰会

拥抱 AI 重塑研发



扫码参与调研

科技生态圈峰会 + 深度研习

—1000+ 技术团队的选择



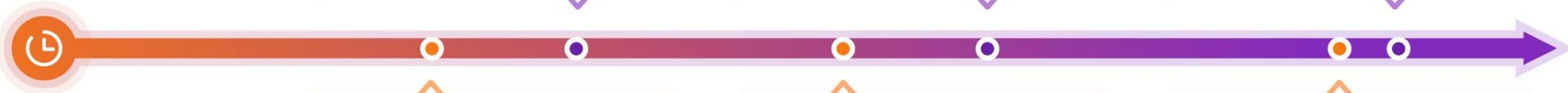
K+峰会 **敦煌站**
K+ 思考周®研习社
时间: 2025.08.29-30

K+峰会 **上海站**
K+ 金融专场
时间: 2025.09.26-27

K+峰会 **香港站**
K+ 思考周®研习社
时间: 2025.11.17-18



K+峰会详情



AIDD峰会 **上海站**
AI+研发数字峰会
时间: 2025.05.23-24

AIDD峰会 **北京站**
AI+研发数字峰会
时间: 2025.08.08-09

AIDD峰会 **深圳站**
AI+研发数字峰会
时间: 2025.11.14-15



AIDD峰会详情



2025 AI+研发数字峰会
AI+ Development Digital Summit

感谢聆听!

扫码领取会议PPT资料

