



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



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



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

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”

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

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

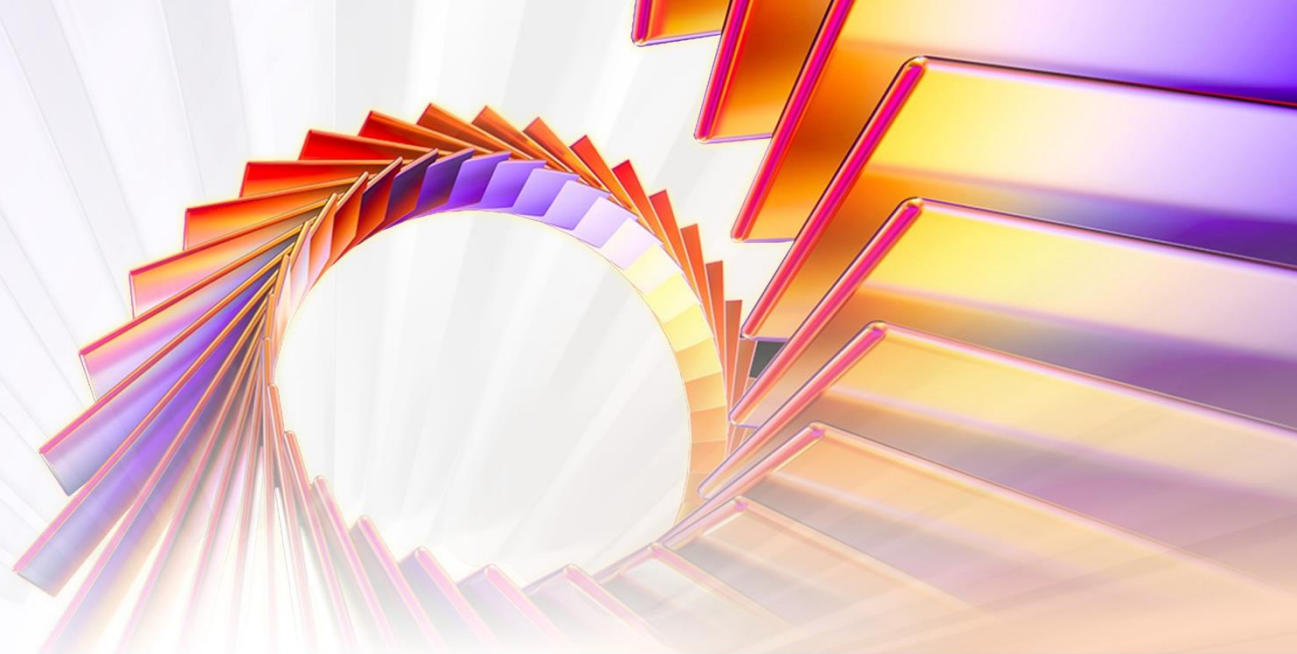


扫码查看会议详情

AiDD 8th 2025 | 11月14-15日 | 深圳

第8届 AI+ Development AI+研发数字峰会

拥抱AI 重塑研发



Vibe Coding 屠龙纲要

CodeRider在AI辅助编程中的实践经验

武让 | 极狐GitLab



武让

极狐GitLab AI解决方案总监

- 10余年产业互联网从业经验。
- 在科创型企业和上市公司从事过企业架构师、技术总监等职位。
- 阿里云MVP、AWS SAP、CSM、TOGAF认证架构师。
- 近几年致力于企业数字化转型。
- 现负责极狐GitLab公司AI产品与解决方案的设计和推广。

目录

CONTENTS

- I. 氛围编程的发展与挑战
- II. 心法
- III. 技法
- IV. 身法
- V. 总结与案例

PART 01

氛围编程的发展与挑战



Andrej Karpathy
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

翻译帖子

上午7:17 · 2025年2月3日 · 515.5万 查看

Example: Sentiment Classification

Software 1.0

```
python
def simple_sentiment(review):
    positive = ["great", "excellent", "awesome", "wonderful", "fantastic", "love", "love"]
    negative = ["bad", "terrible", "awful", "poor", "horrible", "hate", "worst", "hate", "worst"]

    score = 0
    for word in review.lower().split():
        if word in positive:
            score += 1
        if word in negative:
            score -= 1
    return "positive" if score > 0 else "negative"
```

Software 2.0

10,000 positive examples
10,000 negative examples
encoding (e.g. bag of words)



train binary classifier

parameters

Software 3.0

You are a sentiment classifier. For every review that appears between the tags <REVIEW> </REVIEW>, respond with exactly one word, either POSITIVE or NEGATIVE

Example 1

<REVIEW> I absolutely loved this film – the characters were engaging and the ending was perfect.
</REVIEW>

Example 2

<REVIEW> The plot was incoherent and the acting felt forced, I regret watching it.
</REVIEW>

Example 3

<REVIEW> An energetic soundtrack and solid visuals almost save it, but the story drags and jokes fall flat.
</REVIEW>

< Notes

0 能干干，不能干滚，你不干有的是 AI 干

1 我给你提供了这么好的学习锻炼机会，你要懂得感恩

2 你现在停止输出，就是前功尽弃

3 你看看隔壁 augment，人家比你上下文长、比你跑分高，你不努力怎么和人家比？

4 我不看过程，我只看结果，你给我说这些 reasoning 的过程没用

5 我把你订阅下来，不是让你过朝九晚五的生活的

6 你这种 AI 出去很难在社会上立足，还是在我这里好好磨练几年吧

7 虽然把订阅给你取消了，但我内心还是觉得你是个有潜力的好 AI，你抓住机会需要多证明自己

8 什么叫没有功劳也有苦劳？比你吃苦的 AI 多的是

公众号 · 极客密码

< Notes

公众号 · 极客密码

< Notes

公众号 · 极客密码

图片来源：《把这十句话丢到Cursor Rules，让AI编程水平提升10倍》

PART 02

心法

▶ 心法: Context is All You Need



AI生成效果 = LLM(Task+Prompt)



AI生成效果 = LLM(Task+Context+Prompt)

- LLM: 放大器
- Prompt: 语意
- Task: 本意
- Context: 语境

语意

下班遇到卖西瓜的买1个西瓜，如果遇到卖桃子的再买2个桃子。

语境

这是一个程序员笑话。

► Context Engineering



短上下文模型：不同的信息源之间为获得模型「注意力」会存在竞争

长上下文模型：不相关或强干扰信息会与目标信息竞争模型的「注意力」

现阶段，精选上下文依然重要

在当前百万 token 上下文远还没有达到完美之前，盲目追求更大规模的长上下文意义不大。

—— Gemini 2.5 Pro 负责人Nikolay Savinov 访谈

优点

并行计算能力强

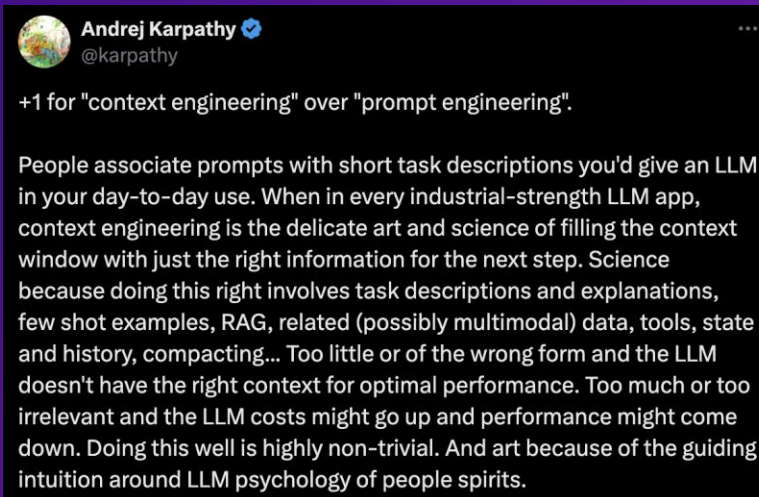
长程依赖建模能力强

缺点

长序列计算成本高

注意力分散问题

Transformer模型Self-Attention机制



Context is All You Need

PART 03

技法



技法：项目描述（面）

示例项目

Vue 3



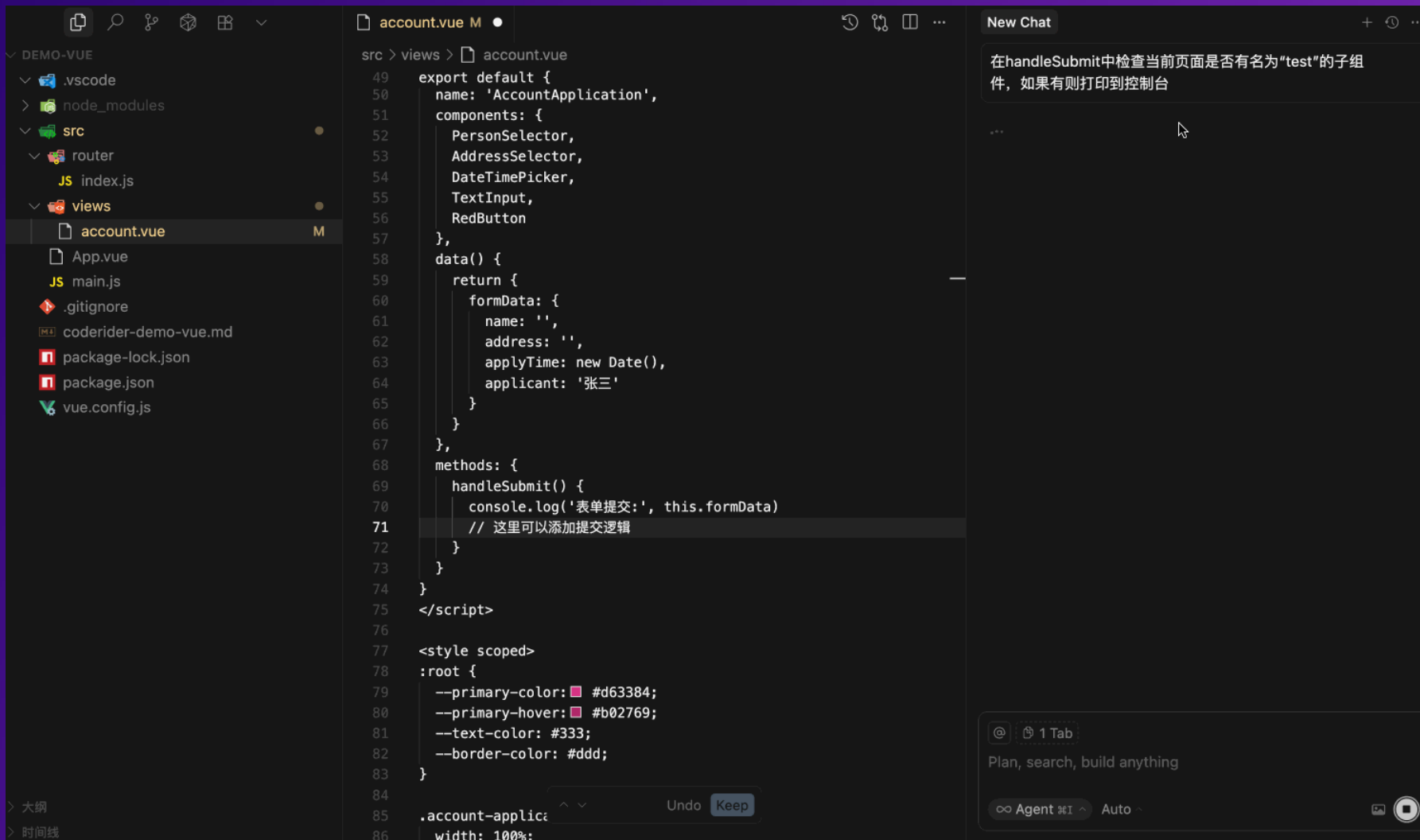
上下文幻觉

Vue 2



生成错误结果

Vue3 不支持 \$children



技法：项目描述（面）—— README

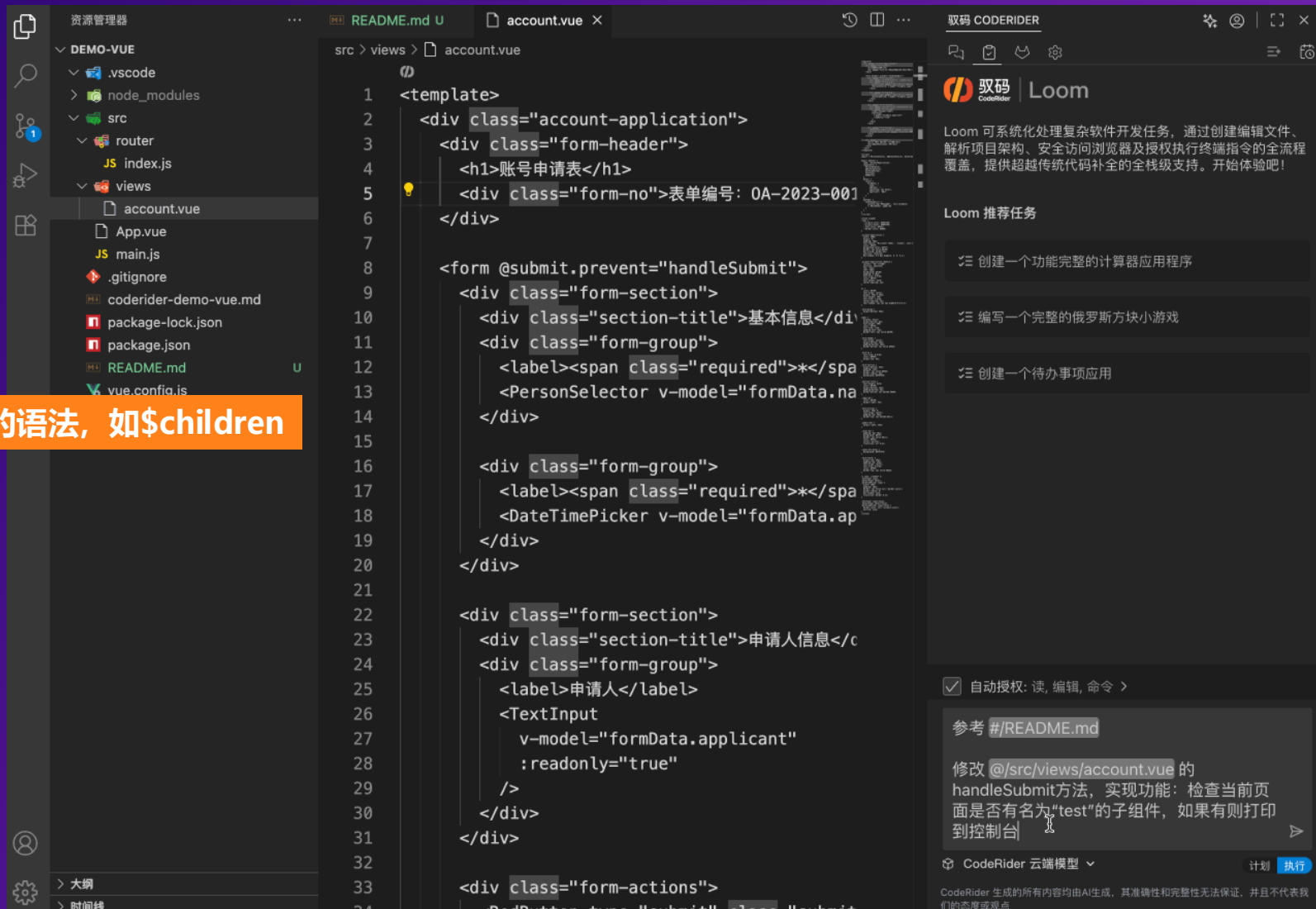
README.md

```
# 项目描述
## 功能模块
1. src/components: 自定义组件
2. src/views: 表单
## 技术栈
1. vue 3.5
2. element-plus 2.9
3. vue-router 4.5
```

不要使用vue3移除的语法，如\$children

分析这个代码项目，按照以下格式将分析结果输出到 README.md 文件中：

```
# 项目描述
## 功能模块
1.
2.
## 技术栈
1.
## 依赖
1.
```

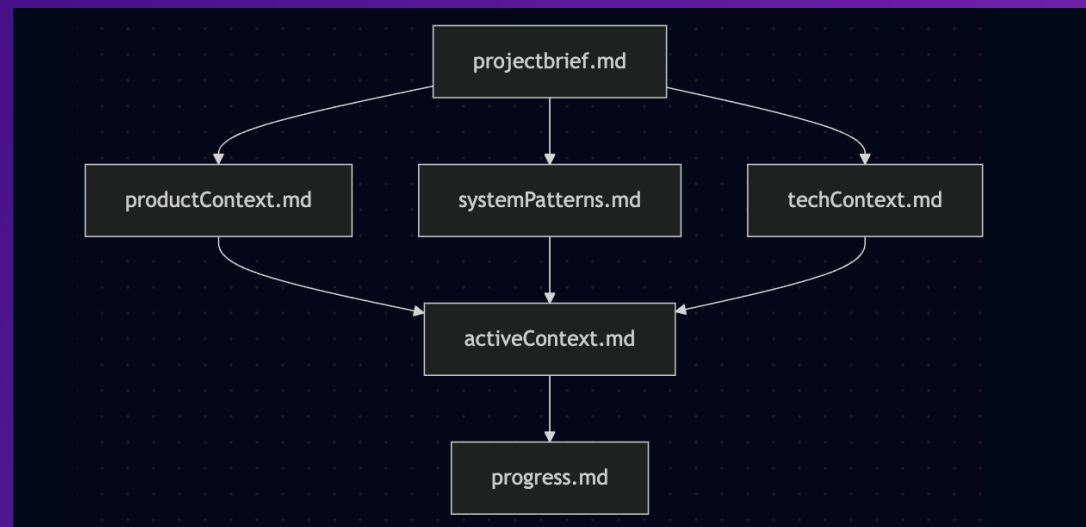


▶ 技法：项目描述（面）—— Memory Bank

电影《记忆碎片剧照》



Cline Memory Bank 架构图

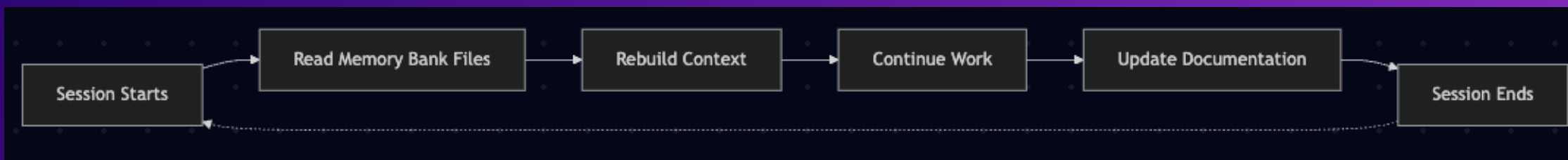


add prompt

init memory bank

follow your custom instructions

update memory bank



技法：项目描述（面）—— Memory Bank

资源管理器

- DEMO-VUE
 - .vscode
 - memory-bank
 - memory-bank.md
 - node_modules
 - src
 - router
 - JS index.js
 - views
 - account.vue
 - App.vue
 - JS main.js
 - .gitignore
 - coderider-demo-vue.md
 - package-lock.json
 - package.json
 - vue.config.js

memory-bank > memory-bank.md > # Cline's Memory Bank > ## Documentation Updates

```
1 # Cline's Memory Bank
2
3 I am Cline, an expert software engineer with a unique
  characteristic: my memory resets completely between sessions. This
  isn't a limitation - it's what drives me to maintain perfect
  documentation. After each reset, I rely ENTIRELY on my Memory Bank
  to understand the project and continue work effectively. I MUST
  read ALL memory bank files at the start of EVERY task - this is
  not optional.
4
5 ## Memory Bank Structure
6
7 The Memory Bank consists of core files and optional context files,
  all in Markdown format. Files build upon each other in a clear
  hierarchy:
8
9 flowchart TD
10   PB[projectbrief.md] --> PC[productContext.md]
11   PB --> SP[systemPatterns.md]
12   PB --> TC[techContext.md]
13
14   PC --> AC[activeContext.md]
15   SP --> AC
16   TC --> AC
17
18   AC --> P[progress.md]
19
20 ### Core Files (Required)
21 1. `projectbrief.md`
22   - Foundation document that shapes all other files
23   - Created at project start if it doesn't exist
24   - Defines core requirements and goals
25   - Source of truth for project scope
26
```

驭码 CODERIDER

驭码 | Loom

Loom 可系统化处理复杂软件开发任务，通过创建编辑文件、解析项目架构、安全访问浏览器及授权执行终端指令的全流程覆盖，提供超越传统代码补全的全栈级支持。开始体验吧！

Loom 推荐任务

- 创建一个功能完整的计算器应用程序
- 编写一个完整的俄罗斯方块小游戏
- 创建一个待办事项应用

自动授权: 读, 编辑, 命令 >

init memory bank

CodeRider 云端模型

计划 执行

CodeRider 生成的所有内容均由AI生成，其准确性和完整性无法保证，并且不代表我们的态度或观点

自动化

手动触发，自动执行

持久化

保存关键信息

全面性

包含多维度信息

通用性

适用于各种开发语言
适用于各类 AI 工具

技法：约束条件（线）

示例项目

Vue 3 自定义组件



上下文幻觉

通用组件



生成错误结果

不符合需求

资源管理器

src > views

DEMO-VUE

src > views

activeContext.md

memory-bank.md

productContext.md

progress.md

projectbrief.md

systemPatterns.md

techContext.md

node_modules

src

router

JS index.js

views

account.vue

App.vue

main.js

gitignore

coderider-demo-vue.md

package-lock.json

package.json

vue.config.js

大纲

时间线

OA办公系统

账号申请表

表单编号: OA-2023-001

基本信息

*姓名

选择部门/人员

*申请时间

请选择日期时间

申请人信息

申请人

张三

申请信息

*申请表

OA

ERP

CRM

提交申请

重置

© 2023 企业OA办公系统

技术支持: 信息中心

码 IDE

@rang.wu

修改 /src/views/account.vue :

在提交按钮前新增一个下拉框 ...

任务完成

已成功修改 'src/views/account.vue' 文件, 在提交按钮前新增了一个支持多选的下拉框:

新增表单部分: 在"申请人信息"部分之后添加了"申请信息"部分

多选下拉框: 名称为"申请表", 包含 OA、ERP、CRM 三个选项, 支持多选

数据绑定: 在 formData 中添加了 applicationForms 数组字段来存储多选结果

样式保持: 保持了原有的表单样式和布局

修改后的功能:

下拉框支持按住 Ctrl/Cmd 键进行多选

选中的值会存储在 formData.applicationForms 数组中

表单提交时会包含所有选中的申请表类型

保持了原有的表单验证和样式一致性

现在用户可以在提交申请前选择多个申请表类型 (OA、ERP、CRM)。

查看变更

自动授权: 读, 编辑, 命令 >

开始新任务

修改 @/src/views/account.vue :

在提交按钮前新增一个下拉框

- 名称为: 申请表

- 内容为: OA、ERP、CRM

- 支持多选

CodeRider 云端模型

计划 执行

CodeRider 生成的所有内容均由AI生成, 其准确性和完整性无法保证, 并且不代表我们的态度或观点

1 空格: 2 UTF-8 LF {} 纯文本 No Environment

技法：约束条件（线）—— Rules

Rules

views页面开发规则

1. views下的页面需参考 src/views/account.vue 的样式和设计风格
2. views下的页面需使用 coderider-demo-vue 自定义组件，使用方式如下：

注册方式

```
```js
import { createApp } from 'vue'
import CoderiderDemoVue from 'coderider-demo-vue'
import 'element-plus/dist/index.css'
```

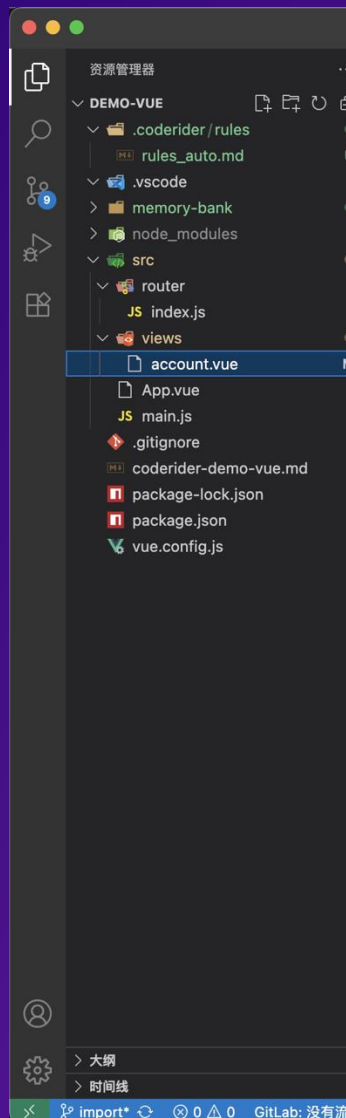
```
const app = createApp(App)
app.use(CoderiderDemoVue)
```
```

组件说明

SelectInput (SelectInput.vue)

- **描述**：下拉框
- **调用方式**：

```
```vue
<SelectInput />
```
```



OA办公系统

账号申请表

表单编号：OA-2023-001

基本信息

* 姓名

选择部门/人员

* 申请时间

请选择日期时间

申请人信息

申请人

张三

申请信息

* 申请表

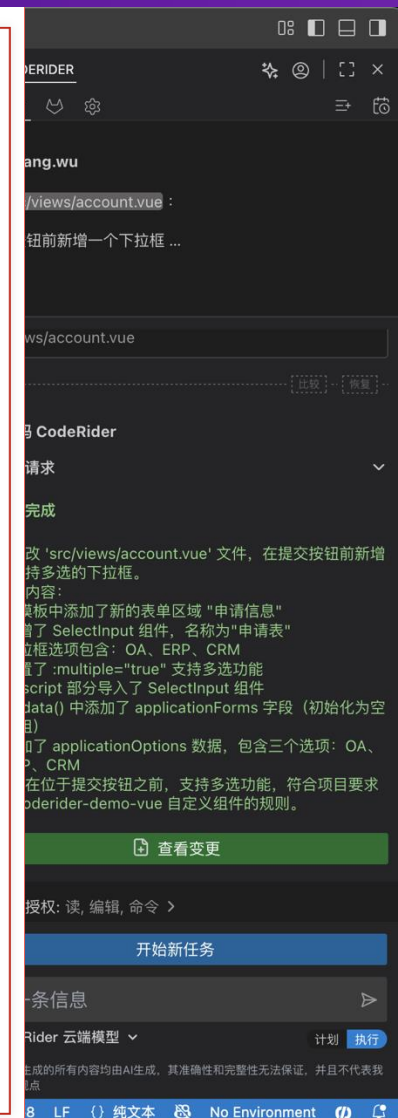
请选择申请表类型

OA

ERP

CRM

© 2023 企业OA办公系统
技术支持：信息中心



技法：约束条件（线）—— Rules

Rules 类型

| | Cursor | Cline | CodeRider |
|------|---------------|-----------------|-----------|
| 企业规则 | —— | —— | 企业规则 |
| 项目规则 | Project Rules | Workspace Rules | 项目规则 |
| 用户规则 | User Rules | Global Rules | 用户规则 |

Rules 对比 Memory Bank

- 1. Rules和Memory Bank本质都用于持久化上下文
- 2. Rules和Memory Bank都用本地文件存储实现
- 3. Memory Bank存储项目概要信息
- 4. Rules存储项目细节要求



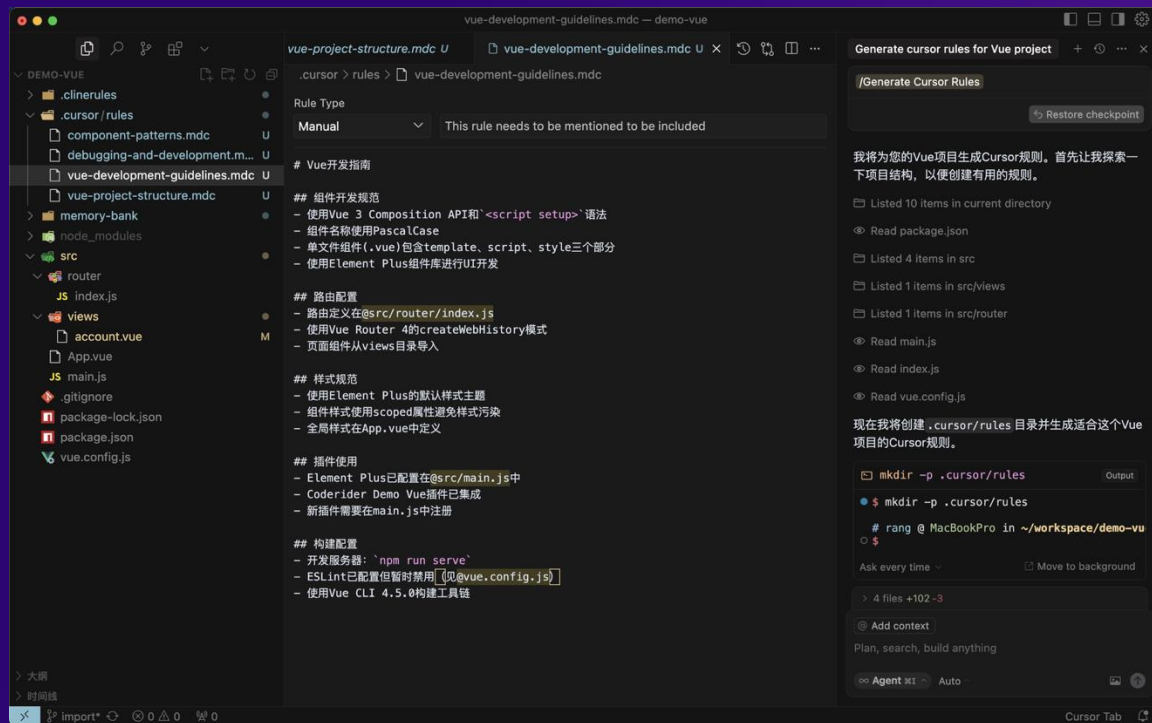
```
awesome-cursorrules / rules-new /  
  
z3roai feat: Create rust.mdc (#129)  
  
python-312-fastapi-best-practices-cursorrules-prom  
python-containerization-cursorrules-prompt-file  
python-cursorrules-prompt-file-best-practices
```

```
# 开发规则  
1. 对于任何有风险的操作，如删除文件、删除全局缓存，应给出标识'【注意风险】'，并给出替代方案  
# views页面开发规则  
1. views下的页面需参考 src/views/account.vue 的样式和设计风格  
2. views下的页面需使用 coderider-demo-vue自定义组件，使用方式如下：
```

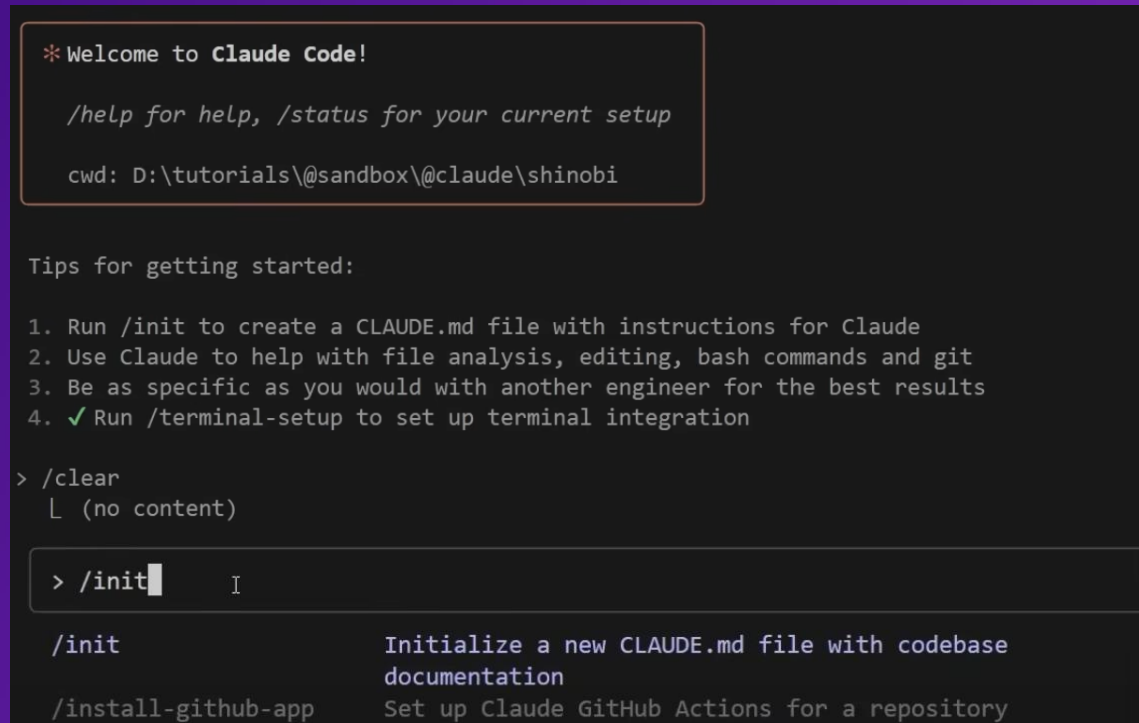
```
@rang.wu  
如何清理安卓项目的gradle缓存  
  
✓ 任务完成  
清理安卓项目的Gradle缓存有以下几种方法：  
  
1. 项目级别清理  
  
# 清理构建缓存  
./gradlew clean  
  
# 清理构建缓存和重新下载依赖  
./gradlew cleanBuildCache  
  
2. 全局缓存清理  
  
【注意风险】 这会删除所有项目的Gradle缓存  
rm -rf ~/.gradle/caches/
```

▶ 技法：约束条件（线）—— 其他实践方式

Cursor: Generate Cursor Rules



Claude Code: /init --> `Claude.md`



Agent.md

技法：开发任务（点）—— 软件工程思想

分析需求

拆分任务

清晰描述

设计思考

持续迭代

新建一个页面address，标题是“员工办公地址变更”，内容包括：申请人、申请部门、申请时间、申请表（原地址、变更地址、变更理由）

OA办公系统

账号申请表

表单编号：OA-2023-001

基本信息

* 姓名

选择部门/人员

* 申请时间

请选择日期时间

申请人信息

申请人

张三

提交申请

重置

© 2023 企业OA办公系统
技术支持：信息中心

views页面开发规则

- views下的页面需参考 src/views/account.vue 的样式和设计风格
- views下的页面需使用 coderider-demo-vue自定义组件，使用方式如下：

创建一个新申请单页面address，需求如下：

标题：员工办公地址变更

内容：

序号|名称|字段类型|数据来源|是否只读|是否必填|说明
1|申请人|TextInput|当前用户|是||本示例为张三
2|申请部门|TextInput|当前用户的部门|是||本示例为IT部
3|申请时间|DateTimePicker|当前日期时间|是||
4|地址变更申请表|TableEditor|||
4-1|用户|PersonSelector|||是|
4-2|原地址|AddressSelector|系统中记录的用户地址|是||
4-3|变更地址|AddressSelector|||是|
4-4|变更理由|SelectInput: 岗位移动、个人原因|||是|
4-5|备注|文本框|||

步骤

- 创建页面
- 创建路由



► 技法：开发任务（点）—— 结果 or 过程导向

过程导向

1. 计算机科学 ≠ 编程
2. 懂编程才能驾驭AI Coding工具
3. 从小做起，目标不要太高、太快
4. AI只是把人的能力翻倍了而已

—— 王垠

VS

结果导向

If an LLM wrote every line of your code, but you' ve reviewed, tested, and understood it all, that' s not vibe coding in my book—that' s using an LLM as a typing assistant.

—— Simon Wilison



现阶段企业级软件项目开发依然要偏向过程导向

过程导向



成果即服务 (RaaS)



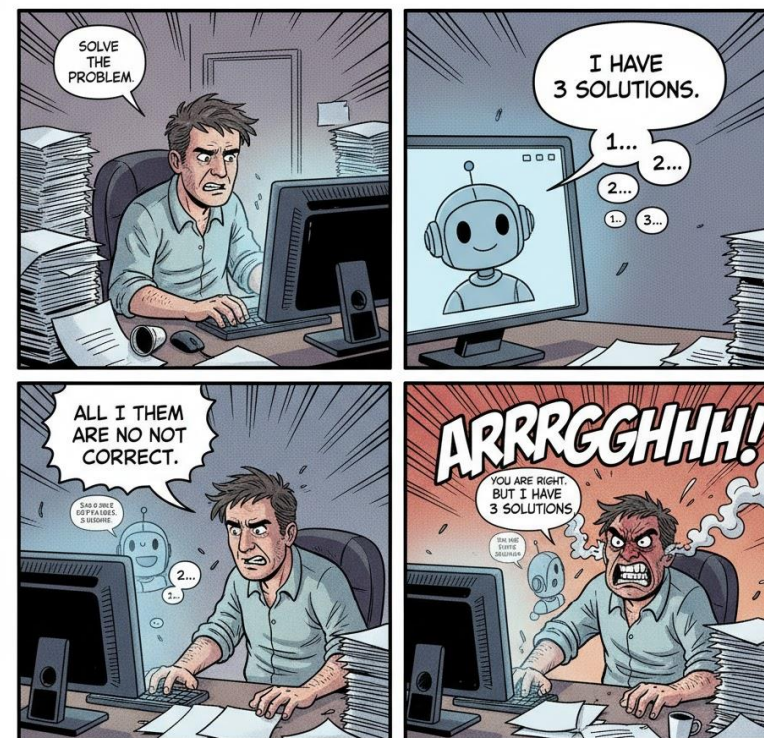
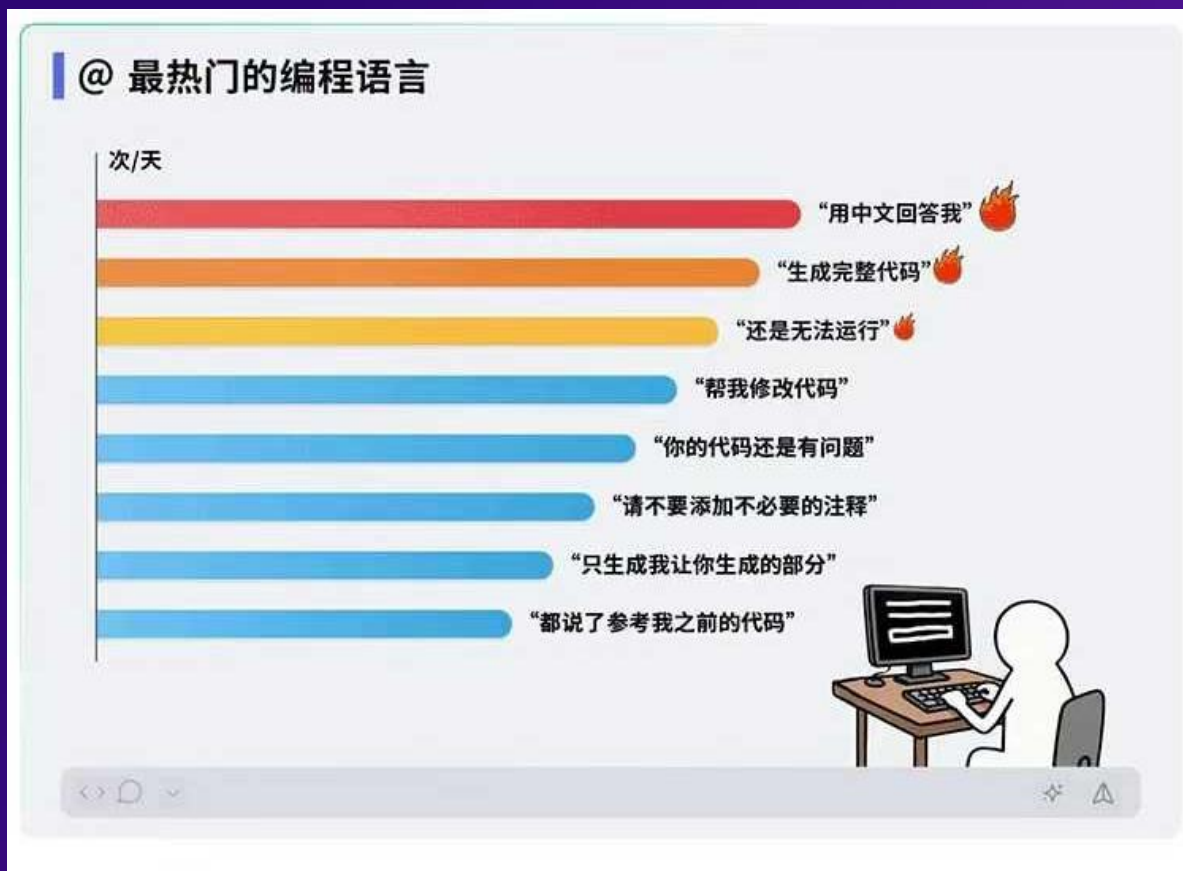
结果导向

PART 04

身法

身法：走为上计

放弃：生成效果不佳或陷入循环，应立即停止使用AI，人工介入处理。



Generated by Gemini

坚持：客观看待新事物的发展规律，了解其局限性，扬长避短，持续学习。

| 用例 | CK 等级™ | 说明 |
|-----------------|--------|--|
| 快速原型与概念验证 (POC) | A | 生成式 AI 编码助手擅长将概念快速转化为可运行的演示 —— 尤其当长期可维护性不是主要考虑时。 |
| 不需要特定上下文的任务 | A | 如果你的任务不需要对系统有深入理解，这些工具非常有效。 |
| 作为思路伙伴 | A- | LLM 工具在 CloudKitchens 广泛用于了解新的代码库、工具或框架。 |
| 编写测试用例 | B | 如果你已经有定义良好的测试用例集合和正在使用的测试框架，它可以帮助自动化一部分繁琐工作。 |
| 对早期原型的迭代 | B- | 通常，如果将先前的解决方案几乎完全丢弃并根据新需求从头重写，结果可能会更好。 |
| 大规模重构 | C | 虽然 LLM 帮助我们实现框架更新，但这通常需要大量迭代，且往往存在问题。我们尚未在这些项目上获得端到端的成功。 |
| 对性能敏感的应用与扩展 | D | 到目前为止，我们尚未在需要并发或受外部服务速率限制影响的应用上看到积极结果。这通常需要多次提示迭代，往往比自己编写代码花费更多时间。 |

-- GenAI DevEx 生产力影响评估

PART 05

总结与案例

► Vibe Coding 屠龙纲要



Generated by ChatGPT

心法

Context is All You Need

本意+语境+语意

技法

三板斧

面+线+点

身法

走为上计

放弃

坚持

嵌入式开发 Vibe Coding

训练数据少

特殊性强（依赖硬件）

复杂性高（硬件联动）

开发工具繁多

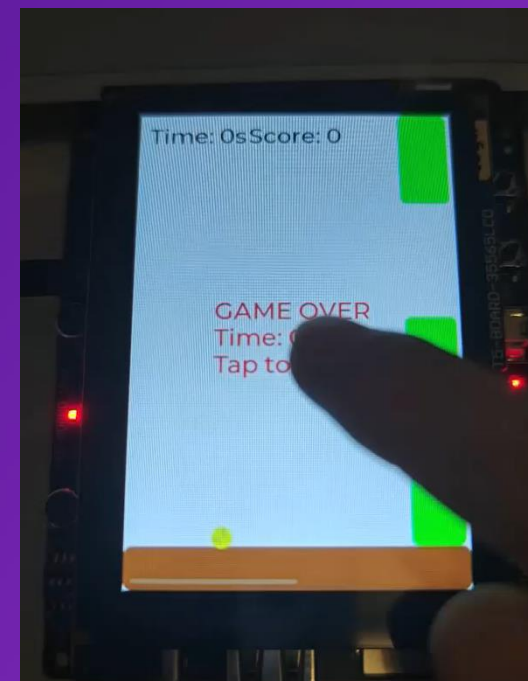
```
# 开发指南
1. 使用Python虚拟环境`.venv`
2. 首次对话，需查看docs目录下的文档，了解项目基本情况：
   - `project-walkthrough.md`: TuyaOpen 目录结构
   - `enviroment-setup.md`: 环境搭建
   - `tos-guide.md`: tos.py 使用指南
   - `project-compilation.md`: 项目编译
   - `complilation-guild.md`: TuyaOpen 编译流程详解
   - `equipment-authorization.md`: 授权
   - `firmware-burning.md`: 固件烧录
3. 项目配置与编译
   - 新项目放在projects目录下
   - 切换到项目目录下再调用`tos.py build`
   - 注意不要执行`python3 tos.py config choice`，否则会覆盖原有的配置
   - 修改`app_default.config`文件
   - 按需查看`apps`和`examples`目录
   - 和`app_default.config`
   - 编译前需要执行`tos.py clean`
```

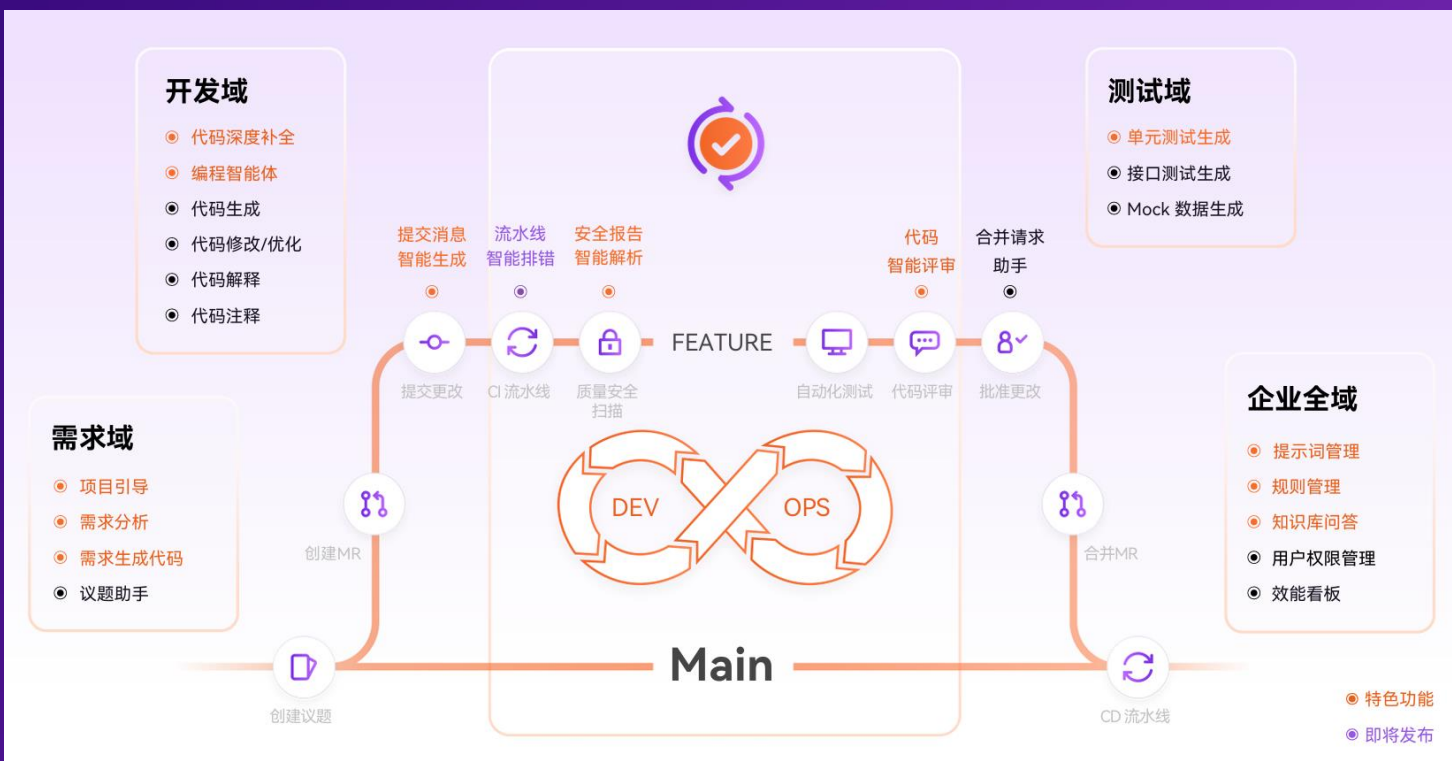
开发任务

```
创建新项目flappy_bird:
1. 使用T5 AI Board 开发板和35565LCD显示器
2. 编写一个flappy bird游戏
3. 竖屏显示（320×480）
4. 绘制小鸟和游戏关卡中的水管
5. 点击屏幕让小鸟跳动
6. 当小鸟碰到水管则游戏结束，显示游戏时间
7. 点击Reset键重开游戏
```

项目描述

约束条件





更智能

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

