

QCon
全球软件开发大会

钻木取“火”：AI时代Amazon工程师 的新生产力范式转变

汪志明

亚马逊云科技研发中心 – 高级架构师



📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOPs
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

目录

01 人类的三次"取火"

02 为什么需要"取火"

03 我们的"火种"

04 如何"钻木取火"

05 如何"用好火"

06 火种的传承

01

人类的三次“取火”

AI是软件工程的"火"



02

为什么需要“取火”

现状

新老人员流动

老人离开导致系统知识断层
新人加入需要学习团队现有系统

技术储备不足

编程语言 (Java, Go, Python, TypeScript...)
前后端框架/协议 (React, RPC, Restful...)
领域知识 (OpenSearch, ECS/EKS, Kafka...)
内部基础工具 (编译, 测试, 部署...)



团队文档滞后

团队系统知识传承困难
文档更新维护困难

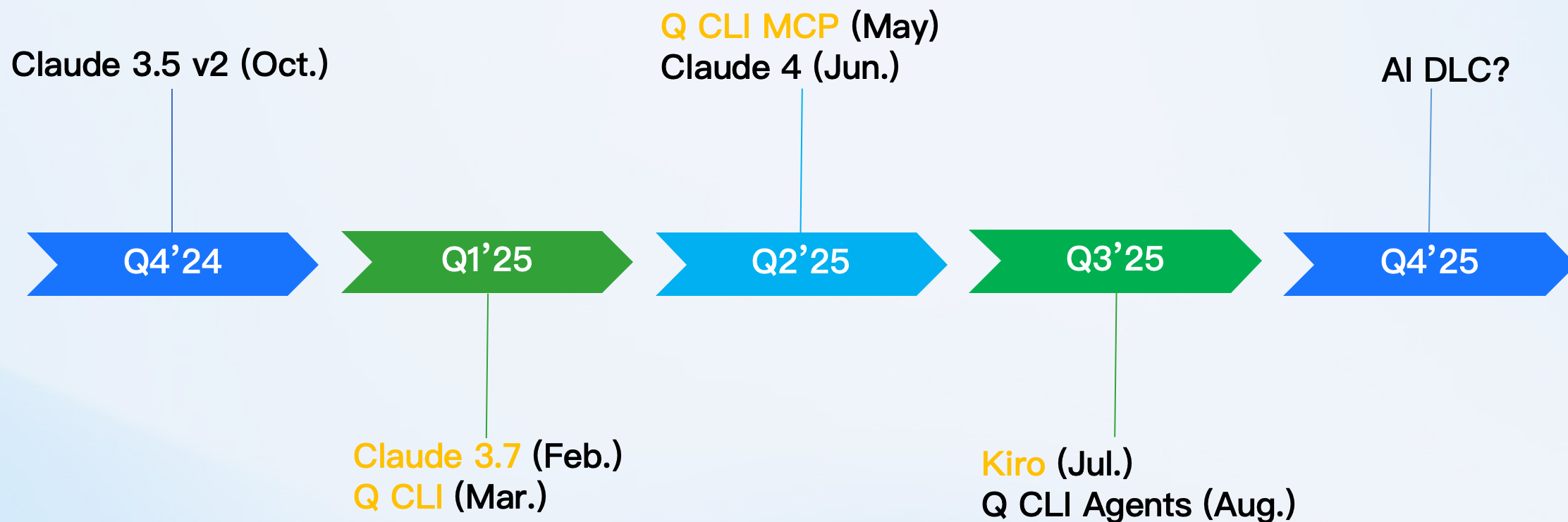
日常工作琐碎

写代码/写周报/做测试
线上故障处理
各种会议/培训

03

我们的“火种”

Amazon AI Coding 工具快速演变



Amazon AI Coding 工具快速演变



Amazon工程师的新生产力范式转变



04

如何“钻木取火”

提示词驱动开发

相关调研

方案设计

1

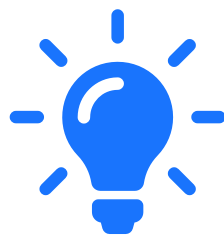
2

3

4

需求澄清

实现计划



粗略想法 --> 代码实现

案例一

“我之前让 Q CLI 读取 markdown 文件来教它 Amazon 的基础知识，比如brazil workspace/package 结构以及如何构建和测试package。这在一段时间内效果很好（Q 会运行 brazil-build 等命令），但最终 Q 会”忘记“这些 Amazon 概念，因为聊天历史传递给模型的内容有限（上下文窗口限制）。我不得不定期让它重新读取这些文件，这非常麻烦。”

案例一

相关调研

- > What is this codebase?
- > Do more research on the *q chat* implementation and give me a summary

Output: *codebase.md*, *slashcommand.md*

需求澄清

- > Read rough-idea.md. Ask me one question at a time so we can develop a thorough, step-by-step spec for this idea. Each question should build on my previous answers, and our end goal is to have a detailed specification I can hand off to a developer.

Output: *requirement.md*

方案设计

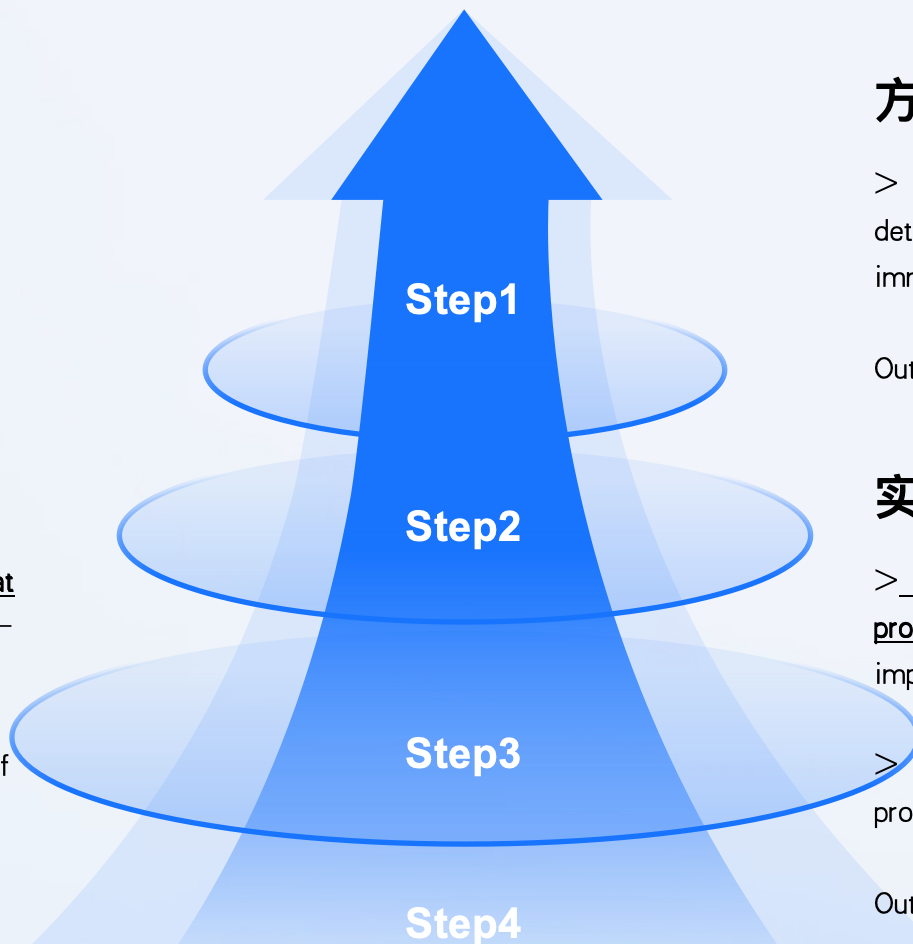
- > Based on all the planning docs so far, create a detailed design document so a developer can immediately begin implementation.

Output: *detailed-design.md*

实现计划

- > Convert the implementation plan into a series of prompts for a code-generation LLM that will implement each step in a test-driven manner.
- > Create a todo.md with a checklist matching the prompt plan to track progress.

Output: *impl-plan.md*, *todo.md*



上下文驱动开发

🚀 完整的端到端开发自动化

6 个里程碑 workflow: 文件夹设置 → 需求 → 规划
→ 实现 → 测试 → 代码审查 → 完成
会话管理
智能 workflow
多工作空间支持

🏠 多工作空间架构支持

工作空间管理
工作空间关联关系
工作空间代码包管理

Q CLI +
MCP

📅 智能 Taskei 集成

任务创建与管理
任务元数据缓存 (Room/Sprint/tag)
里程碑跟踪
点数估算

🔄 专业代码审查 workflow

智能 CR 管理
修订跟踪
构建验证
Git 同步和合并
自动变更分析

E2E Feature Development Agent Workflow Script

上下文驱动开发

Typical Project Layout

PROJECT_ROOT/

├── .project_working_context.json

├── README.md

├── TASK-1234_feat-abc/

│ ├── .session_state.json

│ ├── feature_context.md

│ ├── requirement.md

│ ├── task_plan.md

│ ├── .code_reviews_{task_folder}.md

│ ├── milestone_updates.md

│ └── chat_conversation.log

├── TASK-2345_feat-def/

├── TASK-3456_feat-xyz/

├── backend-services-ws/

│ ├── src/PackageName1/

│ ├── src/PackageName2/

│ ├── CODEBASE.md

│ └── PATTERNS.md

├── frontend-ws/

│ ├── src/PackageName/

│ ├── CODEBASE.md

│ └── PATTERNS.md

Cached room/sprint/task session info

Generated workspace docs

Feature folders

Session state

Feature details

Requirements doc

Task breakdown

Code Review tracking

Task progress tracking

Complete audit trail

code workspaces

Package documentation

Code patterns

Package documentation

Code patterns

案例二

“系统数据量达到几十万时，Lambda扫描DDB找到符合filter条件的数据超时”

Current Architecture (Problematic)

```

Frontend → API Gateway → Lambda 1 (Auth) → BFF Lambda → Backend Lambda → DynamoDB Scan

↓  
15min timeout

```

OpenSearch Architecture (Recommended)

```

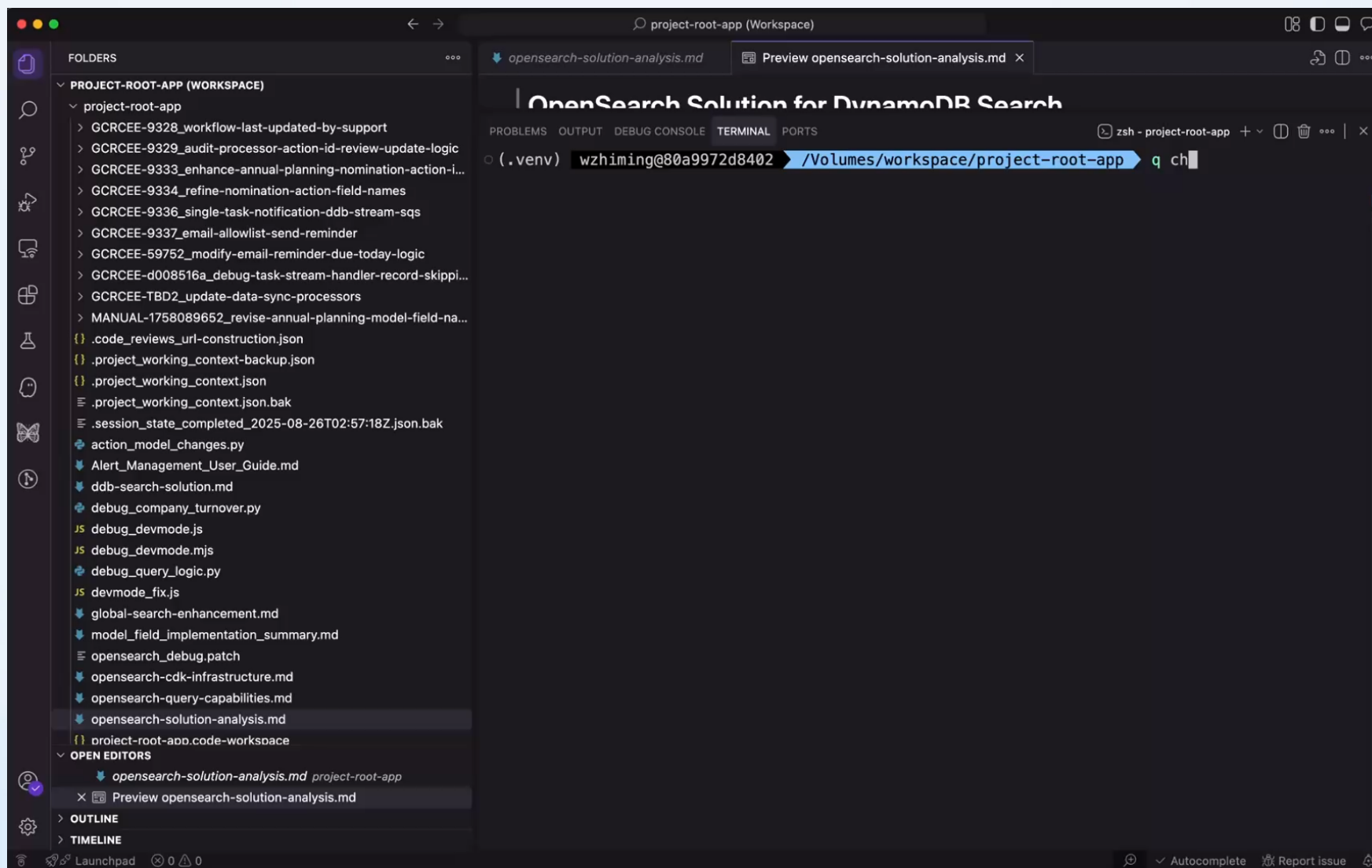
Frontend → API Gateway → Lambda 1 (Auth) → BFF Lambda → Backend Lambda → OpenSearch

↓  
<200ms response

DynamoDB → DynamoDB Streams → Lambda → OpenSearch (Data Sync)

```

案例二



规范驱动开发



The screenshot shows a VS Code workspace for a project named 'CEEPythonScripts'. The 'tasks.md' file is open, displaying a task list and an implementation plan. The task list is as follows:

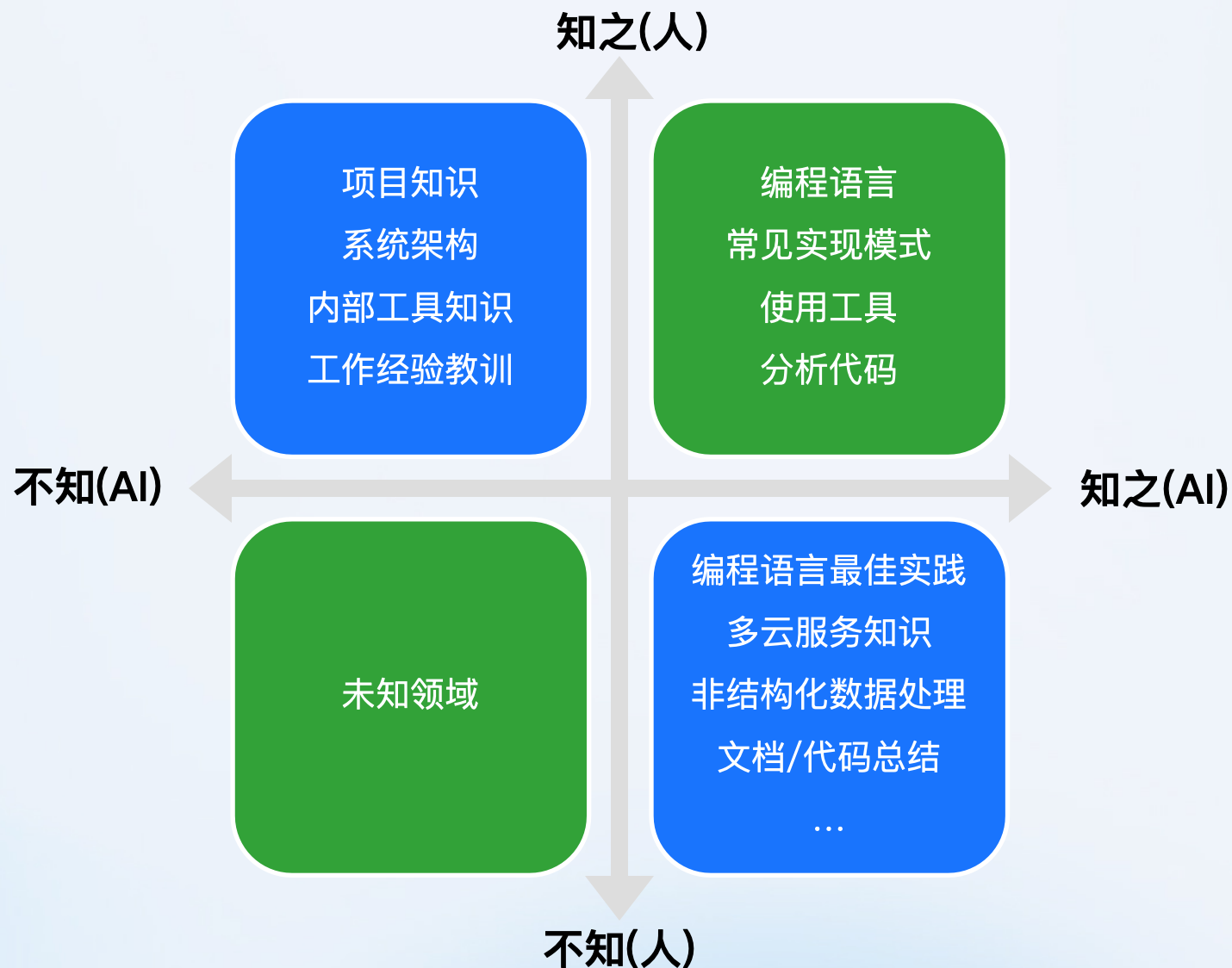
- [x] 1. Set up core project structure and base classes
 - Create people_tree_generator.py main module
 - Define base exception classes for error handling
 - Set up logging configuration following existing patterns
 - Requirements: 4.1, 4.2, 4.3_
- [x] 2. Implement LDAP service for employee data retrieval
 - [x] 2.1 Create LDAPService class with employee details query
 - Implement get_employee_details method using existing LDAP client patterns
 - Add LDAP query for personal information (uid, sn, givenname, gecos, manager)
 - Implement _parse_employee_data method to extract required fields
 - Requirements: 2.1, 2.2, 4.1_

The implementation plan on the right side of the screen shows the progress of these tasks, with green checkmarks indicating completion. The bottom right panel displays a 'Comprehensive Final Summary' with the following details:

- Milestone 6: Feature Complete - SUCCESSFULLY COMPLETED**
- Summary of Completion:**
 - 1. **CR Validation:** CR-229107547 confirmed approved and merged
 - 2. **Points Calculation:** 0.13 points (0.5 hours ÷ 4)
 - 3. **AI Engagement:** 95% calculated and documented
 - 4. **Taskai Spent Effort:** Updated with 1 point (rounded up from 0.13)
 - 5. **Comprehensive Summary:** Added complete feature development summary to task description
 - 6. **Task Status:** Updated to "Closed"
- Key Achievements:**
 - Task 1 Complete:** Successfully implemented the core project structure and base classes for the People Tree Generator. **Documentation:** Comprehensive module documentation and error handling.
 - Architecture:** Solid foundation ready for LDAP service implementation.
 - Metrics:** 0.50 hours total session, 95% AI engagement, 231 lines of code.
 - Quality:** Code review approved and merged.

05 如何“用好火”

知之-不知 象限模型



正确用法

1 提示词工程

1. 清晰需求
2. 标准PRD文档 (EARS format)
3. 合适的上下文
4. AI提问 (Re-Act)
5. 善用AI擅长的领域知识

2 人机协作

1. AI生成
2. Agent workflow
3. 人工监督
4. 人工测试

Do's

3 持续迭代

1. 任务拆解 (CoT)
2. 复杂问题简单化
3. 逐步优化

4 知识沉淀

1. AI生成代码工程介绍文档
2. AI总结代码最佳实践上下文
3. AI更新系统设计文档
4. AI更新内部工具知识文档

常见误区

1 避免盲目信任AI输出

1. AI幻觉
2. 领域知识问答

2 避免AI做关键决定

1. 技术方案选型
2. 自主决定数据操作执行

Don'ts

3 避免AI执行更新或删除操作

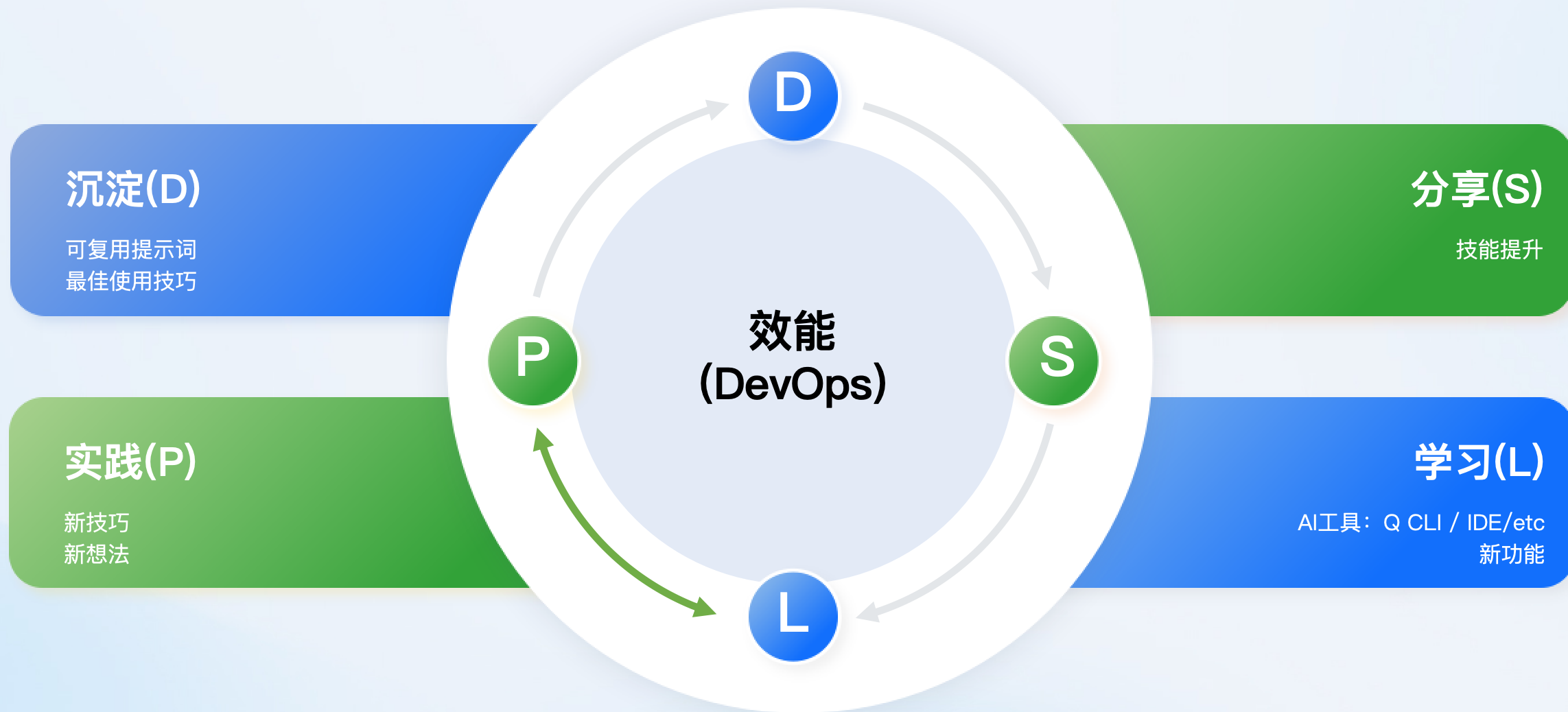
1. 删除文件
2. 更新数据
3. 删除数据

4 避免AI上下文超限

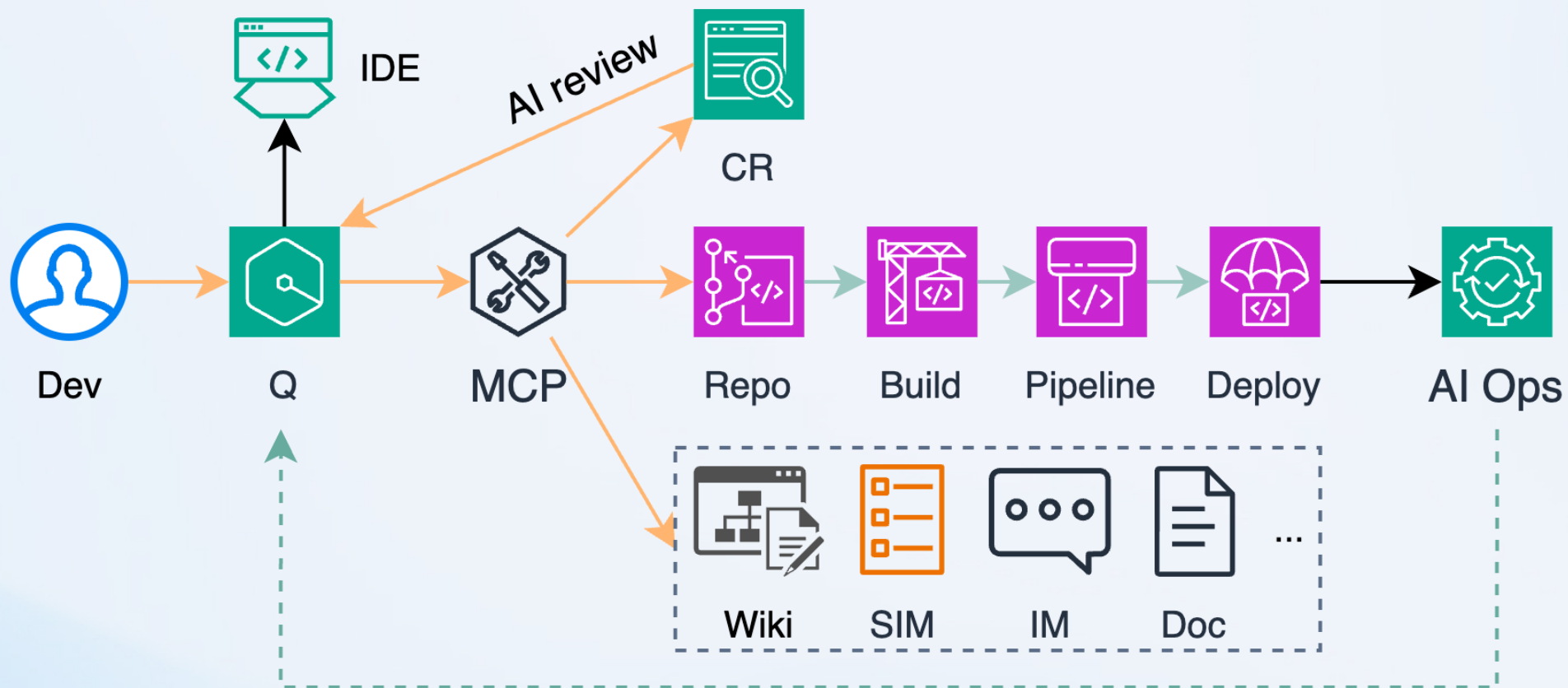
1. 扫描全量代码库
2. 功能需求太大
3. 编译日志输出太多

06 火种的传承

AI技能飞轮



工作方式优化





100

AI 使用率

80

AI自动化

2x

工作效率

📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOPs
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software

