



2024 AI+研发数字峰会

AI+ Development Digital summit

AI驱动研发变革 促进企业降本增效

北京站 08/16-17



与开发者同频 —— 百度构建人机 协同新范式的实践

牛万鹏 百度Comate架构师

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



上海站

K+ 全球软件研发行业创新峰会

时间: 2024.06.21-22



敦煌站

K+ 思考周®研习社

时间: 2024.10.17-19



香港站

K+ 思考周®研习社

时间: 2024.11.10-12



K+峰会详情



上海站

Ai+研发数字峰会

时间: 2024.05.17-18



北京站

Ai+研发数字峰会

时间: 2024.08.16-17



深圳站

Ai+研发数字峰会

时间: 2024.11.08-09



AiDD峰会详情



2024 AI+研发数字峰会

AI+ Development Digital summit

深圳站 11/08-09



AI 驱动研发变革 促进企业降本增效

2024深圳站-议题设置

AI+产品线	LLM驱动产品创新	LLM驱动需求与业务分析	AI驱动设计与用户体验
AI+开发线	AI 原生应用开发框架与技术	AI Agents在研发落地实践	LLM驱动编程与单测
AI+测试线	LLM驱动测试分析与设计	基于LLM生成测试脚本与数据	LLM和AI应用的评测
AI+工程线	AI+DevOps 与工具 (LLM 时代的平台工程)	大模型对齐与安全	端侧大模型与云端协同
AI+领域线	领域大模型 SFT 与优化	知识增强与数据智能	大厂专场

扫描右侧二维码
查看更多会议详情



早鸟票限时抢购中 (截止到9月30日)

¥3680

早鸟票

¥2800

学生票



牛万鹏

百度Comate架构师

百度资深研发工程师，毕业于吉林大学，毕业后入职百度
长期负责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
现负责百度研发智能化，通过构造全新智能编码工具，搭建全新的产品形
态，推动百度万人研发范式的变革。

目录

CONTENTS

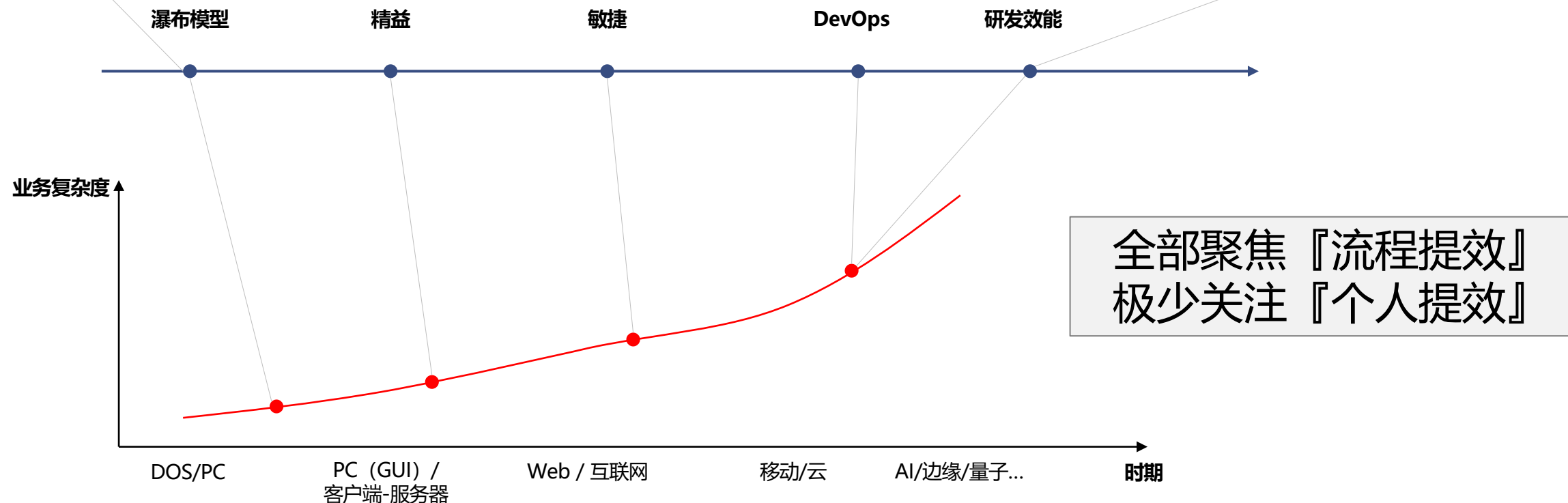
1. 在AI前夜——百度研发现状
2. 在AI时代——智能研发助手
3. 在AI未来——人机协同新范式

PART 01

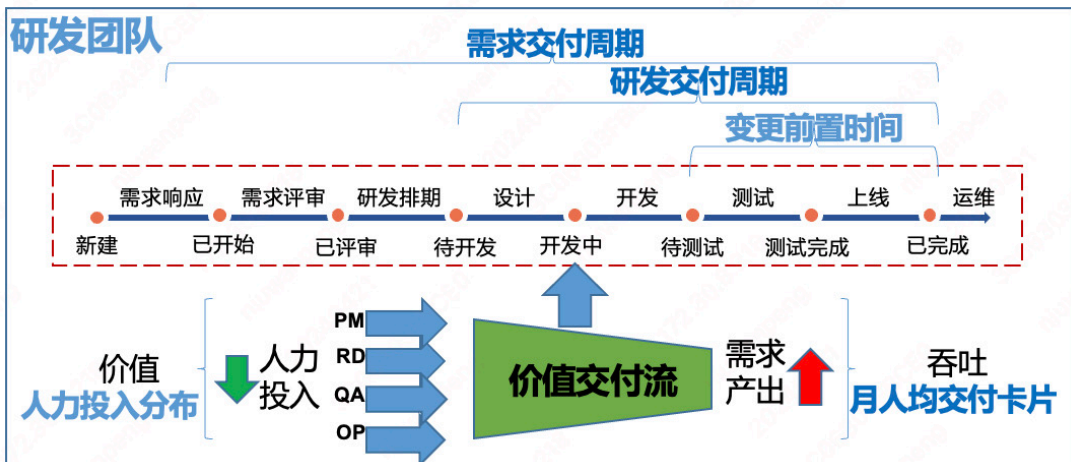
在AI前夜——百度研发现状

业界软件开发模式的发展轨迹

软件开发的基本性质			
沟通密集性	协作密集性	持续迭代性	不可复制性
需要每个角色密切沟通，保证理解一致性，降低随机性	需要从产品经理到研发、测试、运维等一系列的角色参与	代码要被不断组装在一起，不断打补丁，让软件系统能够持续工作	软件规模虽然在不断扩大，但并非重复，工程师每天都写不同的代码
研发协作的密度增高和迭代周期增长，引起效率降低。			



百度研效工具的发展轨迹



流程支撑

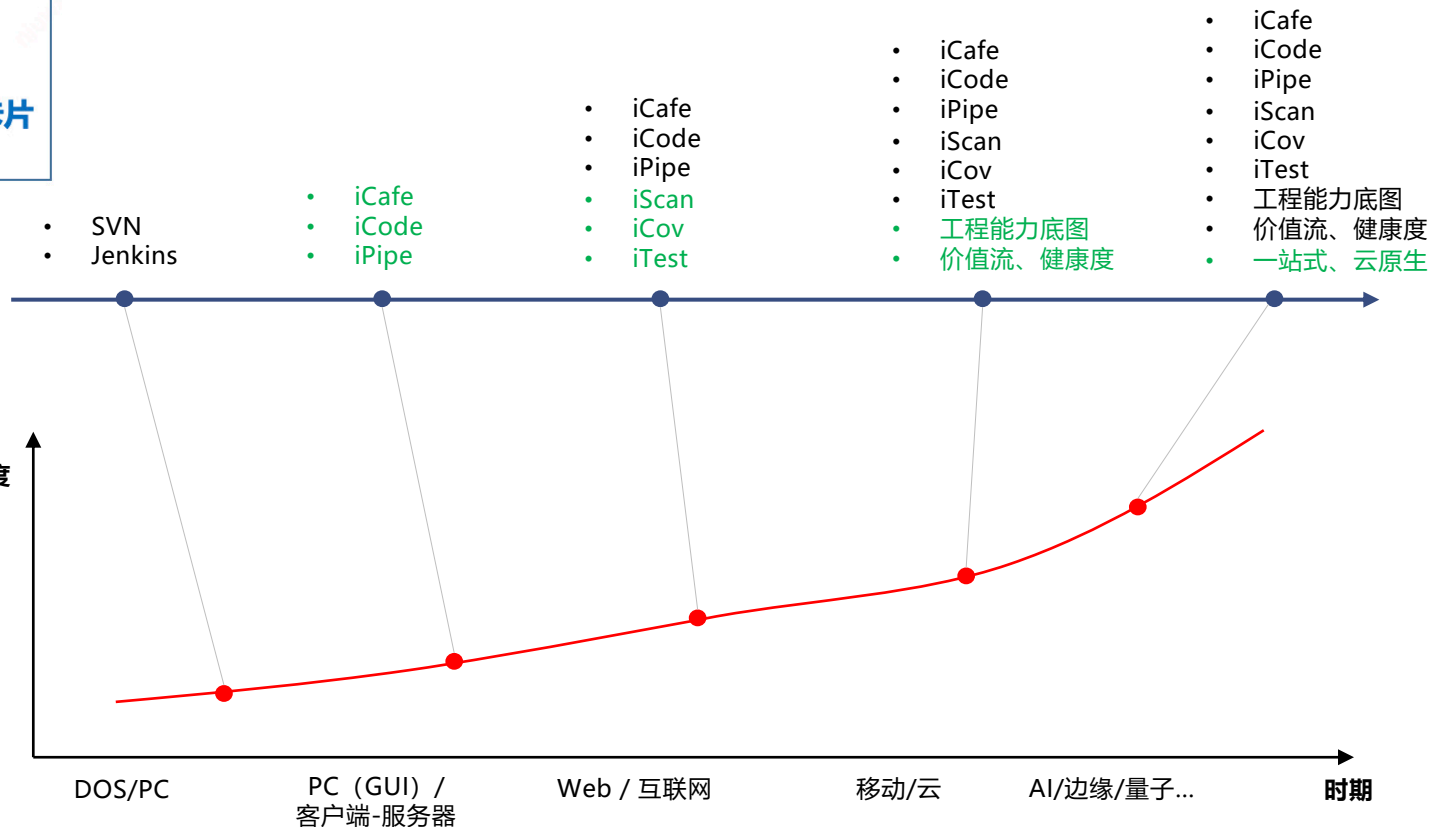


- 1w+工程师, 1k+业务团队, 10w+代码库
- 85%的需求一周交付, 50%的研发资源云化
- 80%的应用从开发到上线全流程在线
- 每天1w次代码提交
- 每月1kw次流水线自动化任务



工具支撑

聚焦『流程』+『协作工具』



▶ 面向流程的研发提效窘境

■ 技术迭代的太快，流程的更新赶不上技术的更新，持续滞后阻碍研发效率提升

- ✓ 流程是实践后的经验、教训等总结，一定是『滞后的』
- ✓ 如，针对Prompt、数据集等在工程团队如何管理，目前没有明确的范式

■ 多数开发者对敏捷、效能、DevOps等不感冒，也不理解其实际意义

- ✓ 面对花样繁多的项目管理流程，多数开发者更想聚焦于研发
- ✓ 过度在团队内推广研发流程，反而引起开发者的抗拒心理，大幅降低开发者的幸福感

核心原因在于整个提效的设计**不是**
站在开发者个人，而是站在组织上



开发者的『iPhone』时刻

Github Copilot + ChatGPT的诞生，
吹响了『开发者个人提效』的号角，
几乎一夜之间所有组织都开始关注

大模型催生了开发者个人提效的『银弹』

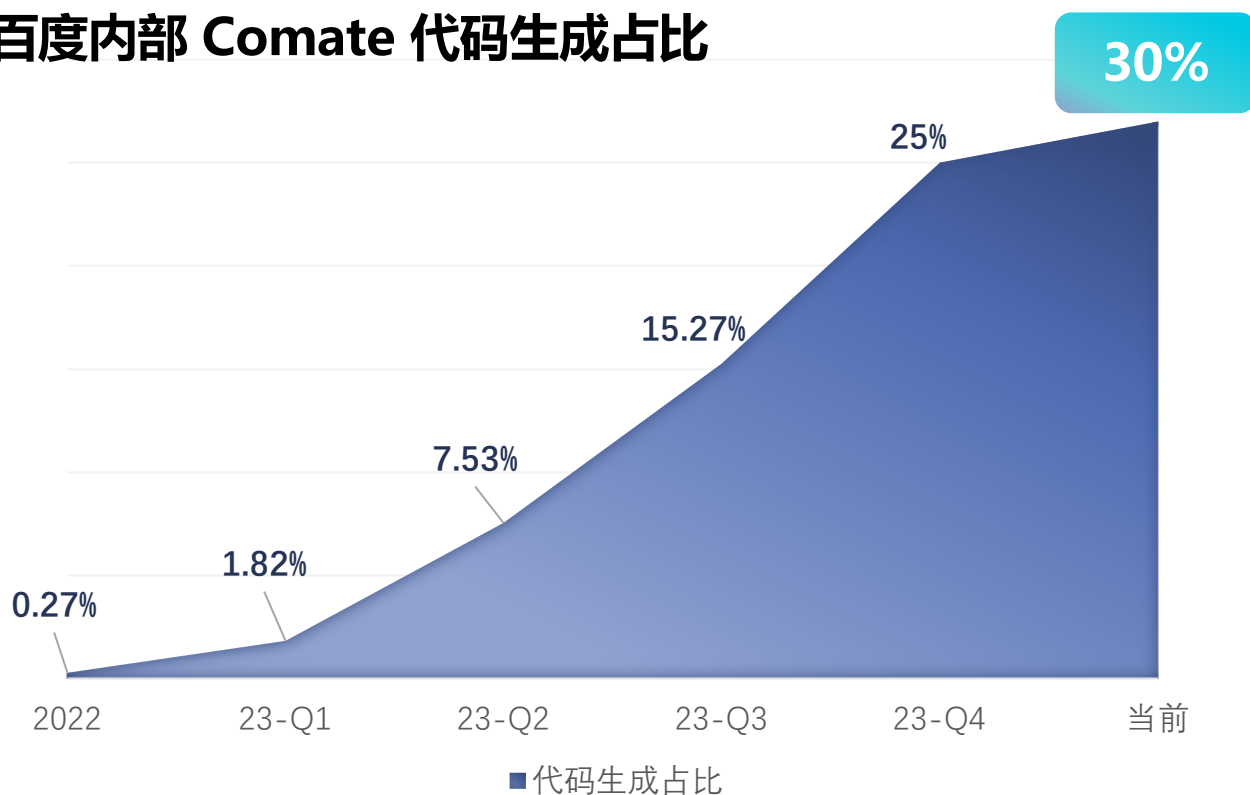
智能研发助手 = 代码自动补全 (Completions) + 理解私域知识 (RAG) + 独立分析需求 (Agent)

PART 02

在AI时代——智能研发助手

▶ Baidu Comate的从无到有

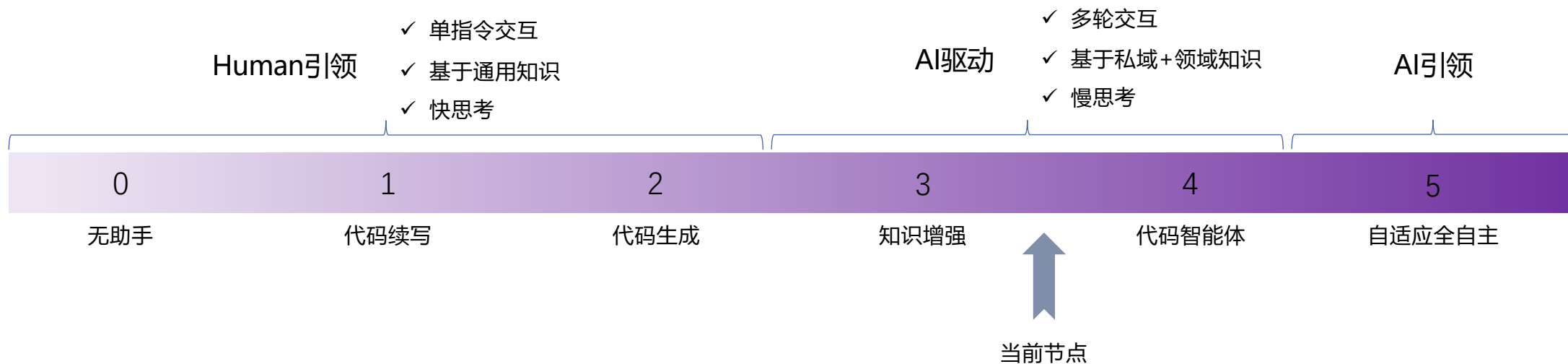
百度内部 Comate 代码生成占比



百度全局提效 **12%**

工程师使用	85%+
用户满意度	90%+
代码生成占比	30%
采纳率	46%

▶ 智能研发助手发展路径



▶ 整体建设思路

能力层

IDE端（目录区、编辑区、Console区等）

知识层

代码类（代码片段、代码依赖等）+ 文本类（技术文档、需求文档等）

框架层

Agent + RAG + P-RAG

模型层

推理调度 + 代码生成 + 代码续写 + Embedding + 意图识别

▶ 能力层

问答区

The screenshot displays the Baidu Comate AI development environment interface, which is divided into three main functional areas:

- 问答区 (Q&A Area):** Located on the left side, it features a chat window with the Baidu Comate logo and a text input field. The chat content includes a greeting: "Hi, 我是文心快码 (Baidu Comate), 您的智能代码助手" and a prompt: "你可以随时让我生成代码、解答本地代码库相关问题或其他编码问题。快来试着问我: 换一换". Below the chat are several AI-generated suggestions, such as "使用知识增强生成效果" with a "#Redis Spring集成Redis" tag, and "总结网页内容" with a link to a Stack Overflow article about Git version control.
- 编辑区 (Editing Area):** The central part of the interface is a code editor showing a Python file named "local_base_builder.py". The code defines a "LocalBaseBuilder" class with a "build" method and a "file_splitter_by_filepath" method. The "build" method includes logic for handling document files and splitting them into chunks. The editor also shows function annotations and a search bar.
- Console区 (Console Area):** Located at the bottom, it displays the output of a terminal session. The output shows a Python execution error: "ModuleNotFoundError: No module named 'knowledge'". The error message includes the file path and the line number where the error occurred.

编辑区

Console区

▶ 能力层 —— 编辑区

```
local_base_builder.py 2, M x
knowledge > builder > local_file > local_base_builder.py > LocalBaseBuilder
23
24
You, 1秒前 | 3 authors (zhuhualiang01 and others)
25 class LocalBaseBuilder(ABC):
26     """
27     本地文件Builder基类
28     """
29
30     activate_executor = concurrent.futures.ThreadPoolExecutor(max_workers=3)
31     def __init__(self, workspace: Workspace):
32         self.workspace = workspace
33
34     def build(self, workspace: Workspace, kg: Knowledge, filepath: str, user: User, **kwargs):
35         """
36         每个子类实现的详细逻辑
37         """
38         try:
39             if kg.file_extension == '.doc':
40                 filepath = self.convert_doc_to_docx(filepath)
41             docs = self.file_splitter_by_filepath(filepath=filepath, **kwargs)
42         except Exception as e:
43             logger.error(f"knowledge {kg.uuid} split failed: {e}")
44             knowledge_repository.update_status(kg.uuid, KnowledgeStatus.FAILED)
45             return
46         active_knowledge(docs=docs, workspace=workspace, knowledge=kg)
```

- 1 代码续写，根据代码上下文自动触发
- 单行推荐
 - 多行推荐
 - 基于注释推荐
 - 基于上下文依赖推荐

- 2 函数头上快捷键，开发者主动触发
- 函数注释
 - 行间注释
 - 生成单测
 - 代码解释
 - 调优建议
 - 函数拆分

▶ 能力层 —— 问答区

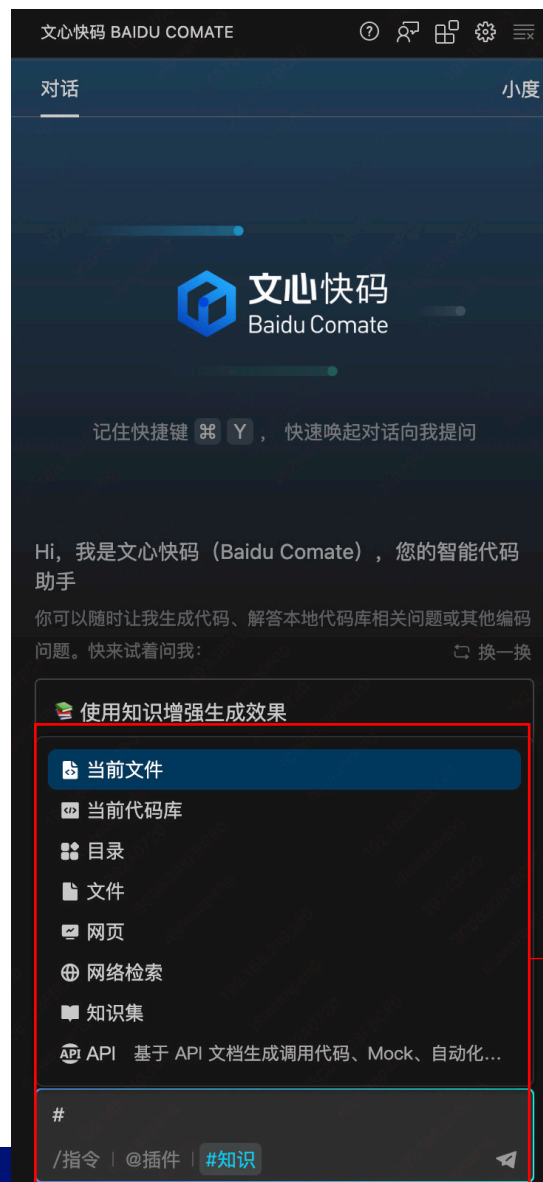
1 点击唤起Comate问答区

2 向Comate提问，咨询任何通用问题、生成通用代码等

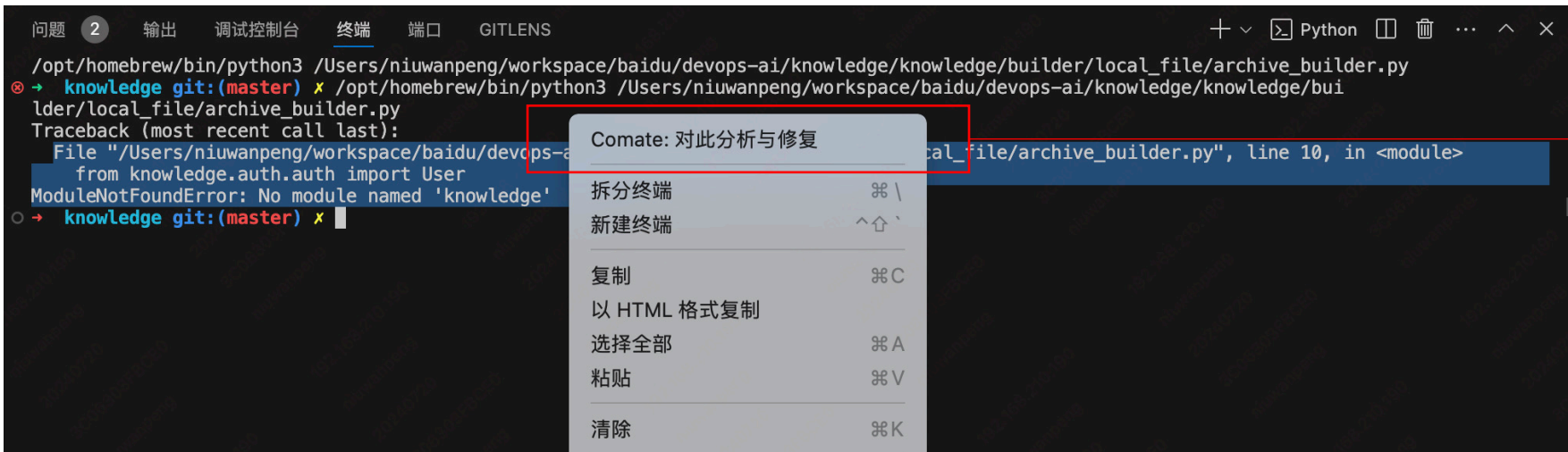


3 通过『#』命令符唤起知识增强选项，如

- 基于网络检索，查询最新的技术方案
- 基于当前代码库，检索代码库内的相关代码
- 基于知识集，查询团队内的技术方案

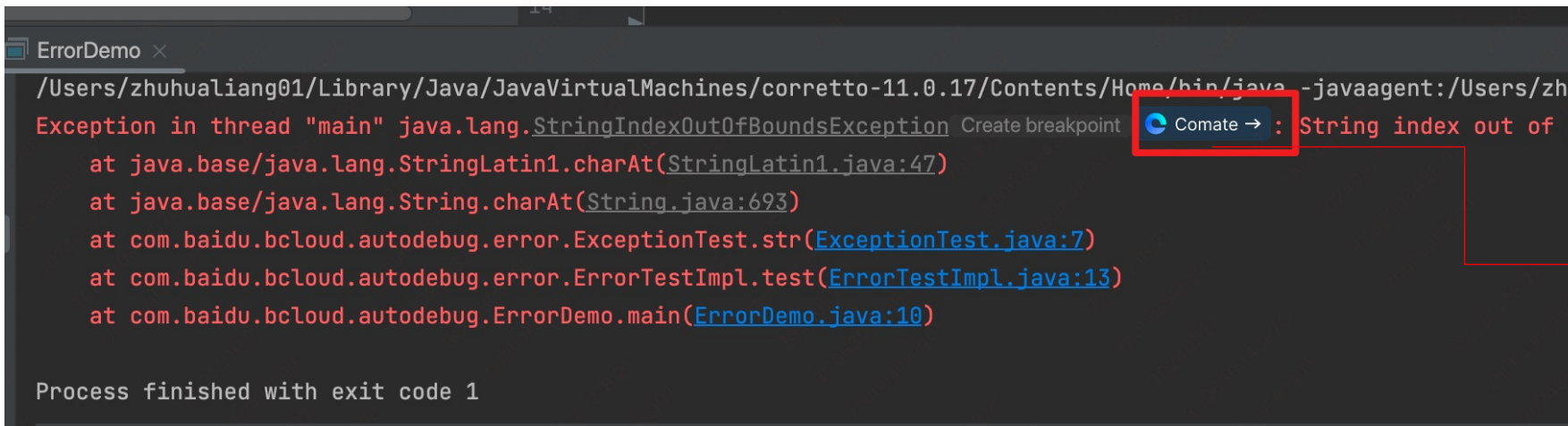


▶ 能力层 —— Console区



The screenshot shows a terminal window in VS Code with a Python error: `ModuleNotFoundError: No module named 'knowledge'`. A context menu is open over the error, with the option `Comate: 对此分析与修复` highlighted. Other menu items include `拆分终端`, `新建终端`, `复制`, `以 HTML 格式复制`, `选择全部`, `粘贴`, and `清除`.

1 在VSCode通过选中错误内容后，右键打开Comate快速修复。



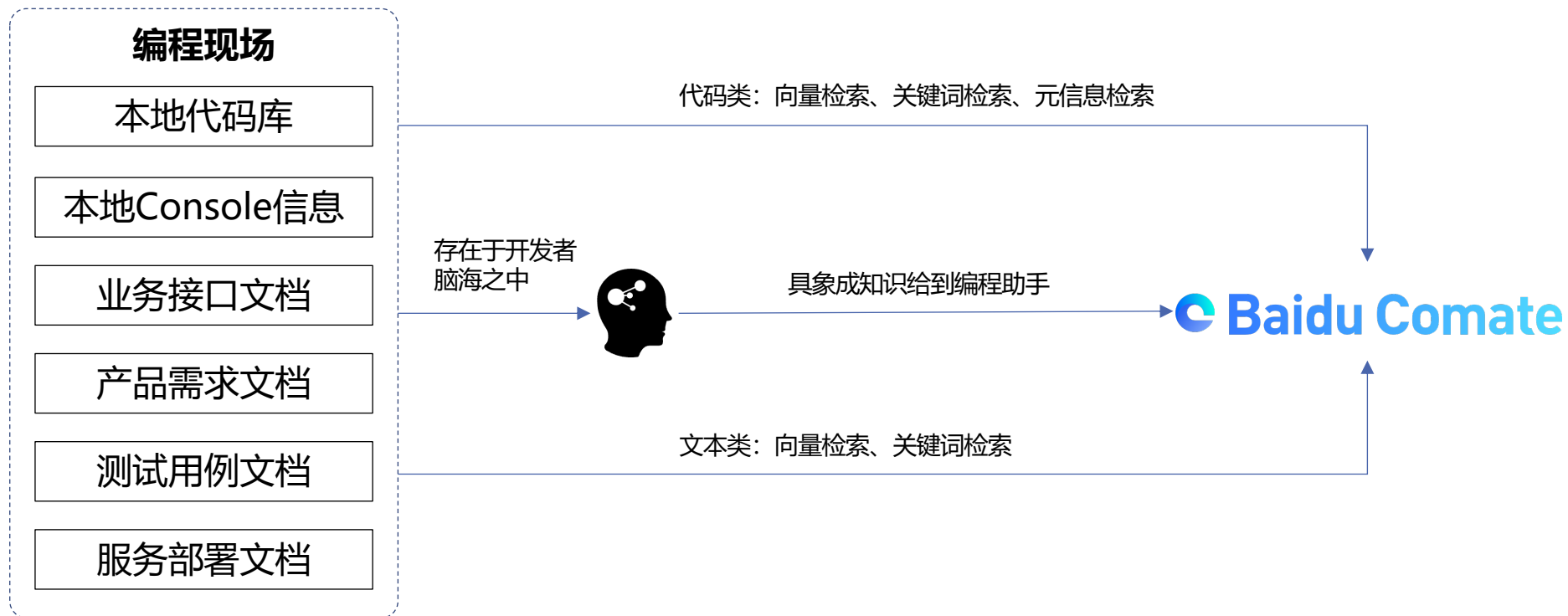
The screenshot shows a Java error in a JetBrains IDE: `Exception in thread "main" java.lang.StringIndexOutOfBoundsException`. A `Comate →` button is visible next to the error message, indicating an automatic quick-fix suggestion.

2 在JetBrains自动捕捉错误，开发者可一键点击修复。

▶ 知识层 —— 为什么需要知识增强

- **通用模型存在瓶颈**：GitHub采纳率停留在46%，长期没有显著增长。主要在于依靠模型内部压缩的知识（以及少量的Neighborsource）没有办法再给出更符合『当前代码库』、更符合『当前业务逻辑』的代码，达到模型瓶颈。
- **人类程序员在开发时也需要掌握额外的知识**：编程现场会存在大量『私域知识』，如本地代码库、业务接口文档等。这些私域知识组成了每个『业务/项目/服务』的全链路开发指南，当我们对这些知识掌握的越完整、越熟练，在开发新的代码时速度越快，编写的代码质量越高。
- **知识增强是Comate进化成智能体的必要前提**：对『编程现场』的理解是所有编码助手的下一步方向，是能够突破现有瓶颈，大幅提高开发者效率，构造真正的人机协同的必经之路。

▶ 知识层 —— 与开发者同频



依靠对开发者编程现场的理解，保持和开发者同频，帮助开发者解决繁琐、重复的问题。对『编程现场』的理解：

- 是所有编码助手的下一步方向
- 是从『简单续写』到『复杂生成』的必经之路
- 是能够大幅提高开发者效率，构造真正的人机协同的必经之路

知识层 —— 如何完全理解文本

文本

业务接口文档

产品需求文档

测试用例文档

服务部署文档

```
1 # Linux上GPU预测部署示例
2
3 ## 1 C++预测部署示例
4
5 C++示例代码在[链接](https://github.com/PaddlePaddle/Paddle-Inference-Demo/tree/master/c%2B%2B/cuda_linux_demo), 下面从`流程解析`和`编译运行示例`两方面介绍。
6
7 ### 1.1 流程解析
8
9 #### 1.1.1 准备预测库
10
11 请参考[推理库下载文档](https://www.paddlepaddle.org.cn/documentation/docs/zh/develop/guides/05_inference_deployment/inference/build_and_install_lib_cn.html)下载Paddle C++预测库, 名称中带有`cuda`的为用于GPU的预测库。
12
13 ... 省略部分内容
14
15 #### 1.1.4 设置Config
16
17 根据预测部署的实际情况, 设置Config, 用于后续创建Predictor。
18
19 Config默认是使用CPU预测, 若要使用GPU预测, 需要手动开启, 设置运行的GPU卡号和分配的初始显存。可以设置开启TensorRT加速、开启IR优化、开启内存优化。使用Paddle-TensorRT相关说明和示例可以参考[文档](https://www.paddlepaddle.org.cn/inference/mas
20
21 ```cpp
22 paddle_infer::Config config;
23 if (FLAGS_model_dir == "") {
24     config.SetModel(FLAGS_model_file, FLAGS_params_file); // Load combined model
```

PaddlePaddle某篇技术说明文档

1 明确的主题和子标题, 分层清晰

2 段落描述清晰, 有总述、有样例、有更详细的参考文档

文字描述丰富

文档中尽可能增加详实的文字描述而不仅仅是图片, 将会显著提高学习质量。

文档逻辑清晰

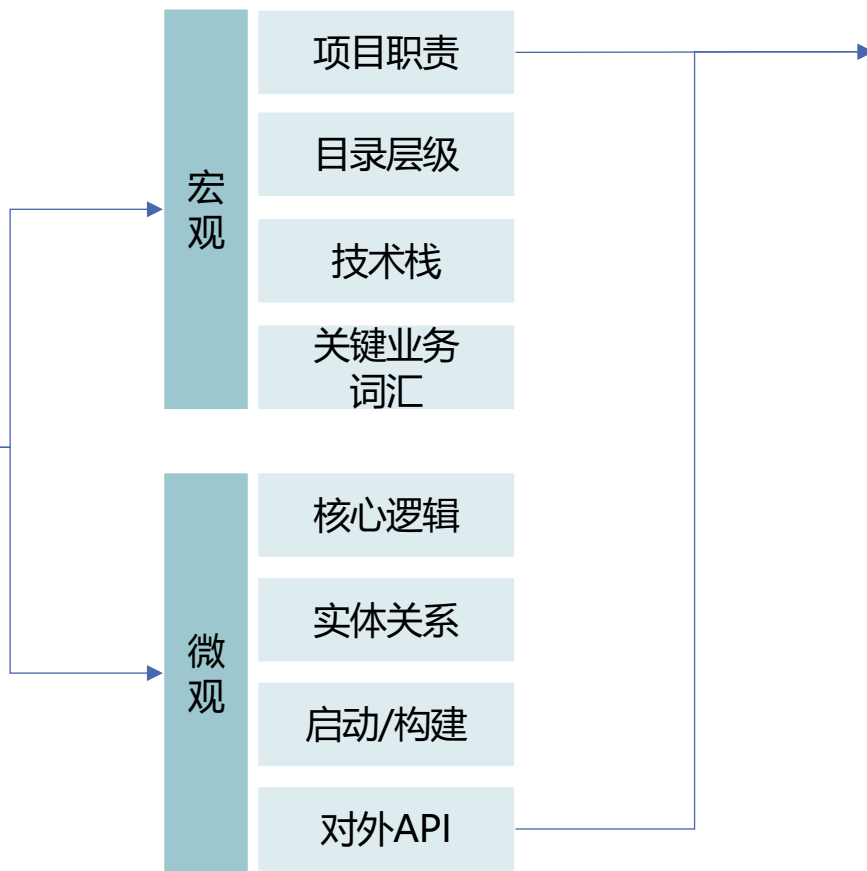
文档内容有层次、语言描述言简意赅无歧义、每个段落有标题等结构, 会显著提高检索质量。

目录结构清晰

如果有大量的文档分散在不同的目录中, 建议每个层级目录的命名言简意赅, 能够充分表达这一层级目录的文档类别。

AI阅读文档的逻辑和人一样, 文档结构越清晰, AI学习的越好。

知识层 —— 人类如何完全理解代码



脑海中『直接』沉淀出如下总结

显性代码知识

这是一个用来xxxxx的项目，它的使用了xxxx、xxxx等框架，分为xxxx、xxxx等模块。主要逻辑包括xxxx、xxxx、xxxx。对外暴露了xxxx、xxxx等关键API。它使用xxxx方式进行部署，启动入口在xxxx。

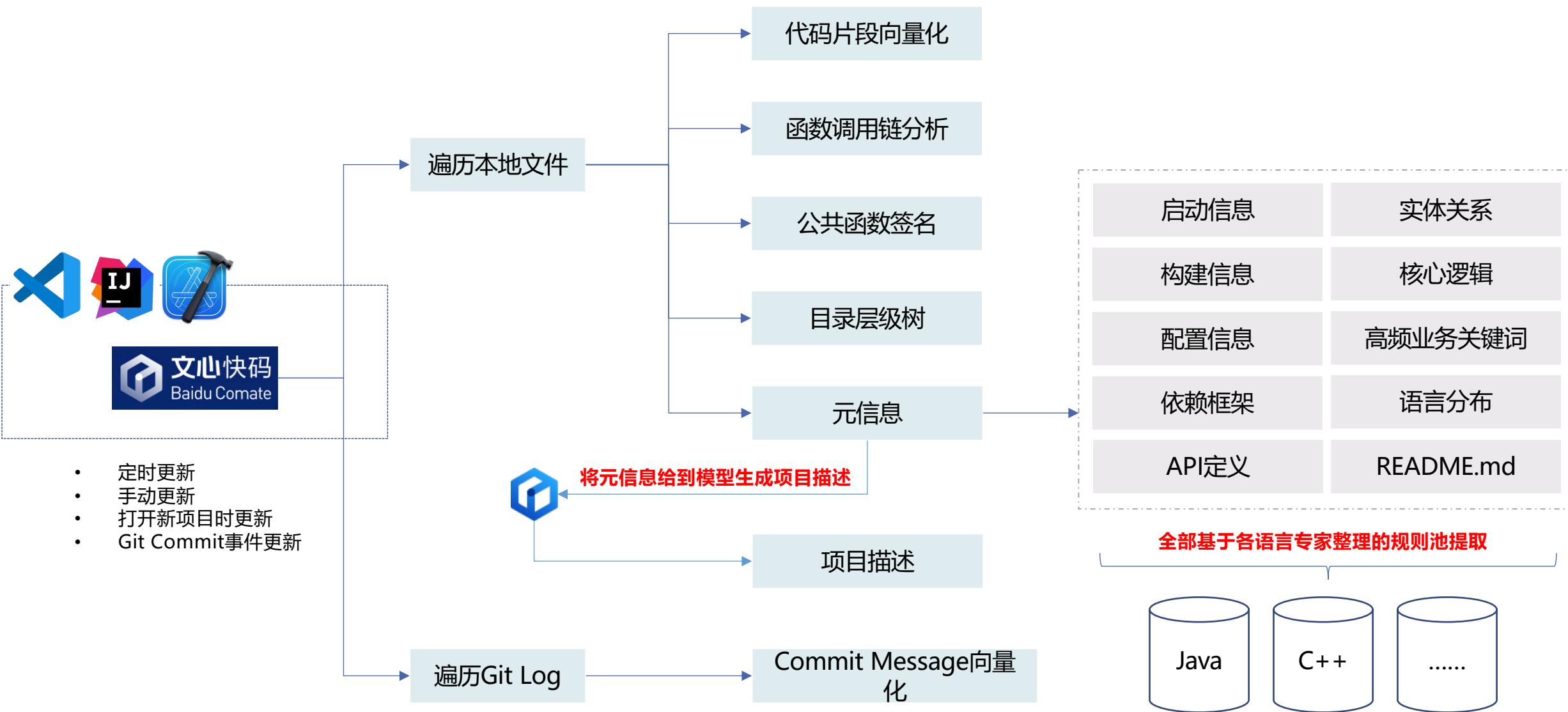
脑海中『间接』沉淀出如下总结

隐性代码知识

这个项目中，新增xxx相关的开发任务需要在yyy、zzz等目录下，命名格式是xxxx，如果新增一个xxxx，需要配套生成一个xxxx。如果要调用外部API，可以使用已经封装好的xxxx工具。

知识层 —— AI如何完全理解代码

显性代码知识



知识层 —— AI如何完全理解代码

隐性代码知识



代码开发规则

什么是代码开发规则

每个代码库独有的开发规则，和开发语言、应用框架强相关，和代码库建库之初定义的规则强相关。



代码开发规则有什么用

每位开发者接手代码库时首先尝试掌握的就是代码开发范式，这决定了新增一个目录、文件、函数等需要放在哪里、如何命名、结构如何定义。

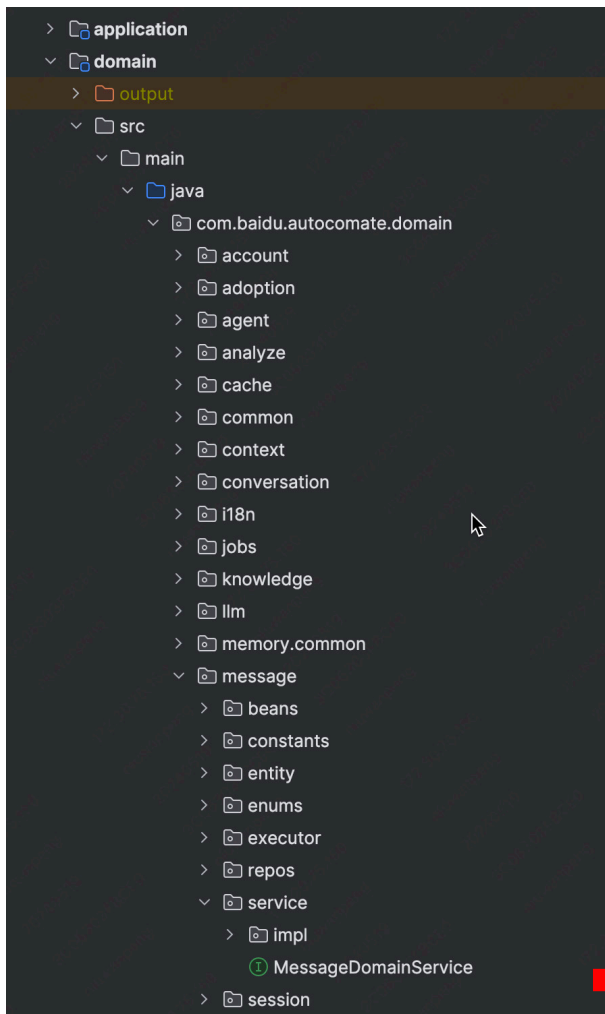


Comate如何利用

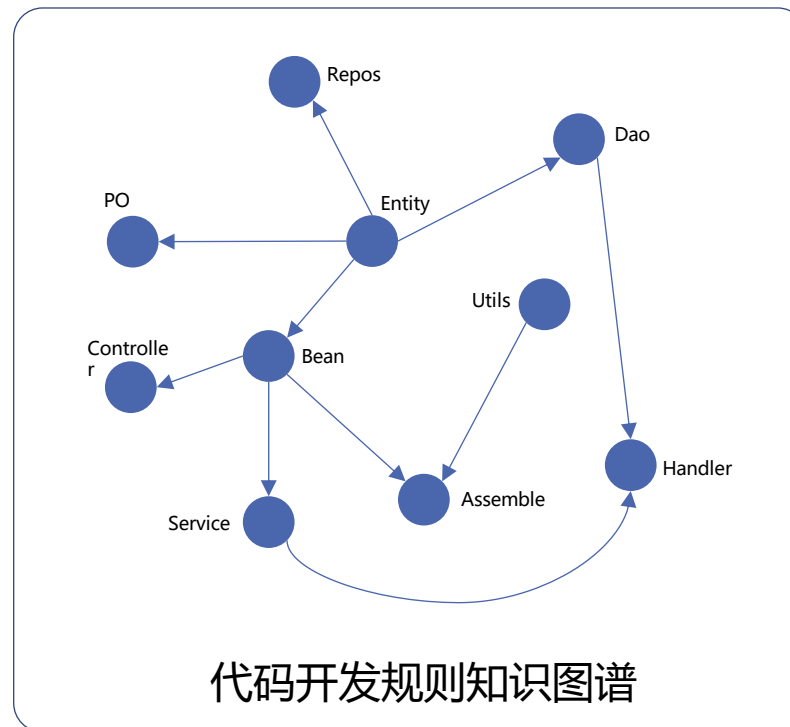
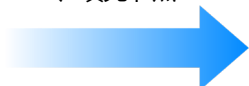
当开发范式可提取、可量化，编码助手在用户新建文件、新增函数时可以预测更大范围的代码，如新建一个Entity文件，同步将Dao、Service、Controller创建。

知识层 —— AI如何完全理解代码

隐性代码知识



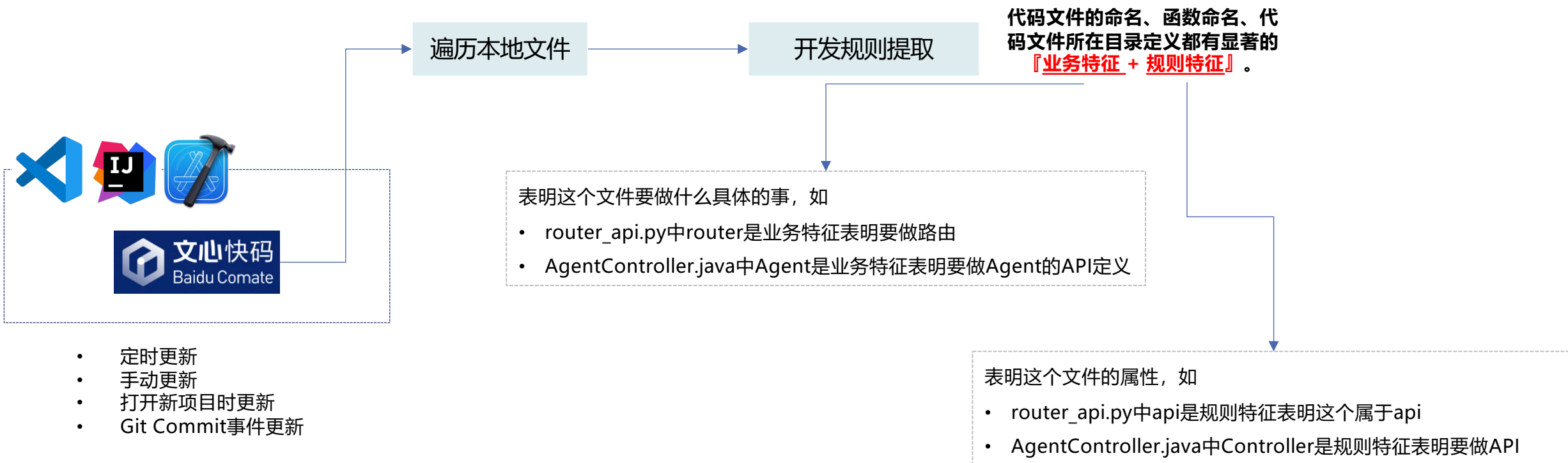
- 1、遍历文件
- 2、依赖解析
- 3、权重计算
- 4、填充节点



一个典型的Spring应用，有bean、entity、po、dao、service等层级关系。

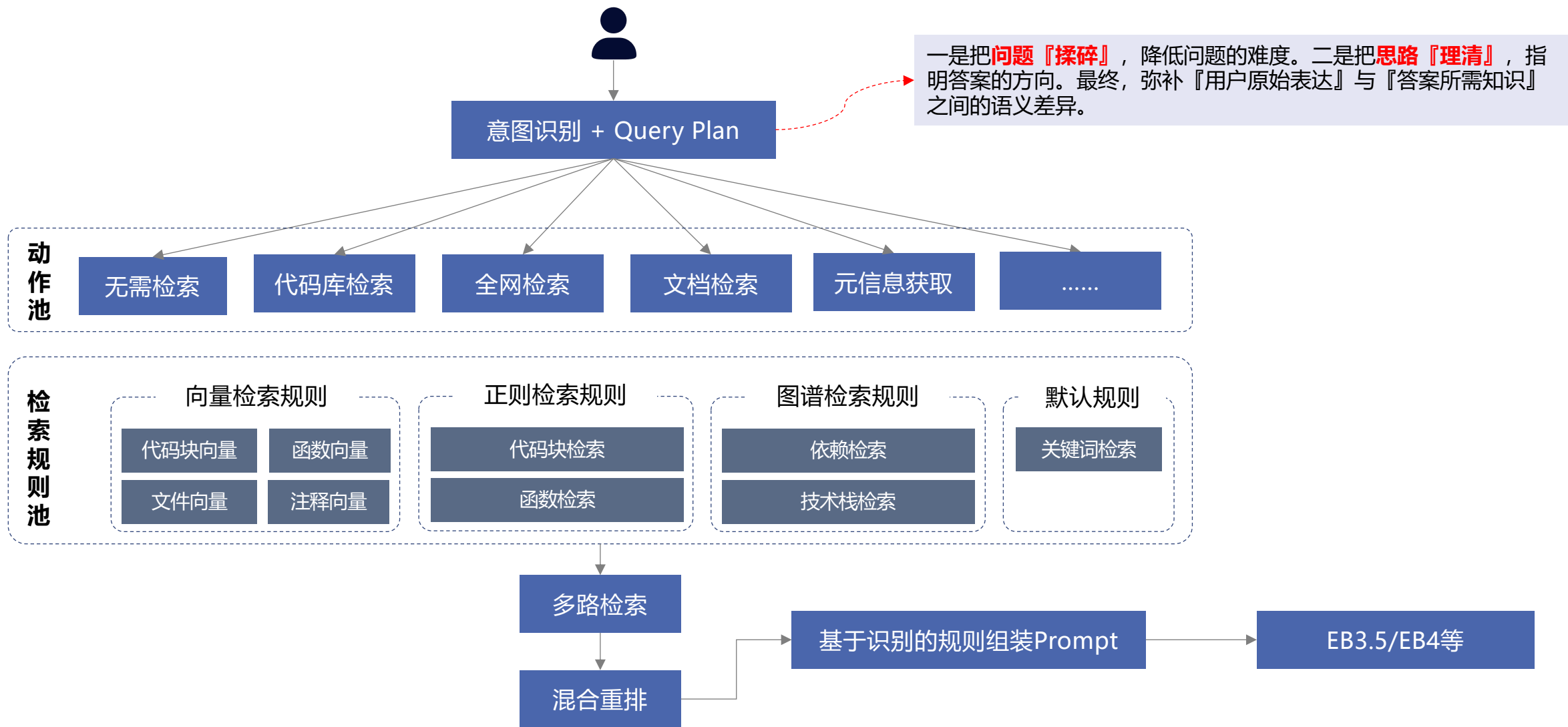
知识层 —— AI如何完全理解代码

隐性代码知识



▶ 框架层—— RAG + P-RAG

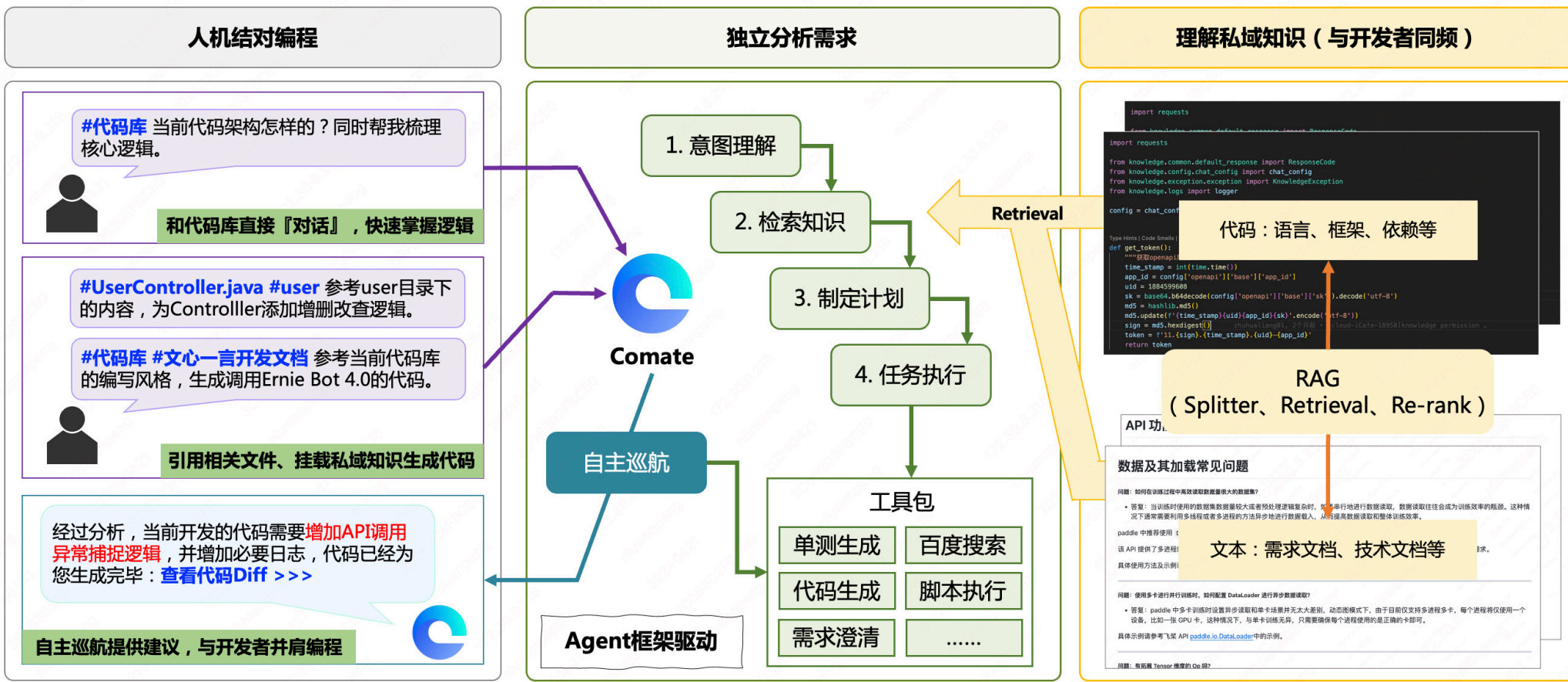
意图识别+RAG = 更能激发检索质量



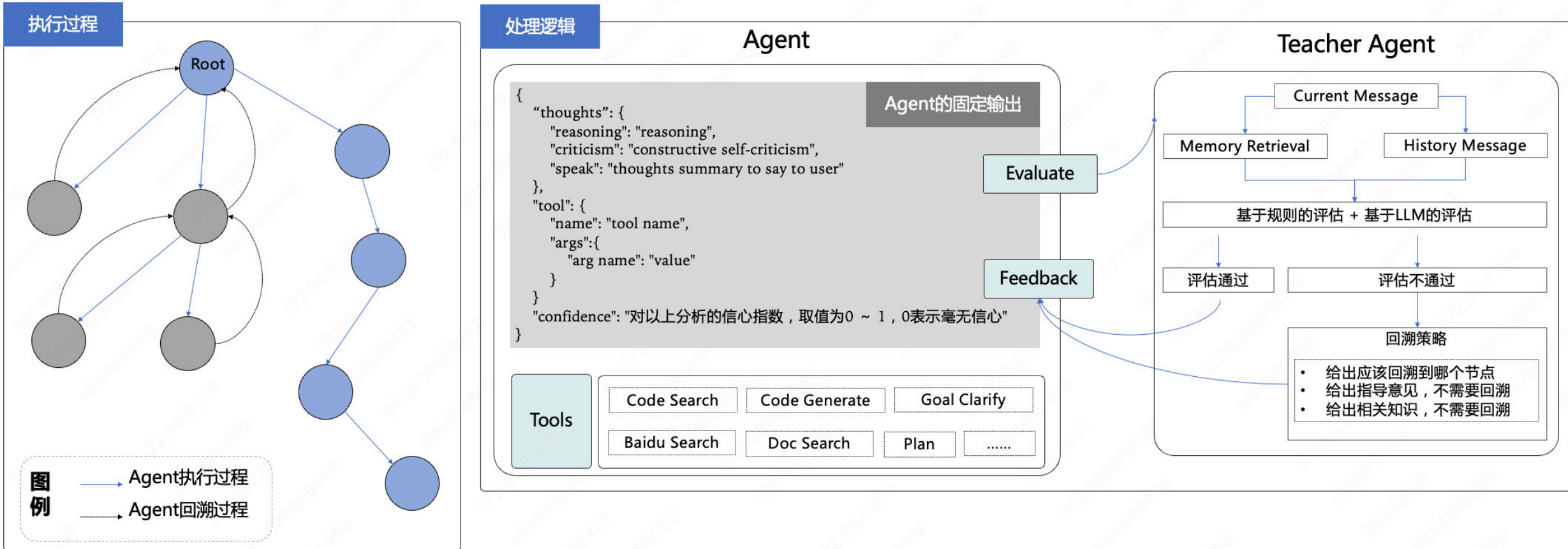
▶ 框架层 —— Agent

人机协同新模式：『人』专注创造性的、复杂性高的逻辑设计并进行决策，『机』负责具体的实现，帮助『人』解决繁琐、重复的问题。

Comate = 理解私域知识 (RAG) + 独立分析需求 (Agent) + 人机结对编程 (Pair)



▶ 框架层 —— Agent



■ 基于Tree of Thought思想，模拟『人』的自我反思行为，提高Agent输出质量

- ✓ Agent的思考过程中是一个连续的、多步骤的过程，每一个步骤执行完毕之后，都会等待Teacher Agent对执行结果做出评估，评估策略的核心是判断该执行结果对全局执行过程是『正向』还是『负向』的影响，并给出『整改策略』
- ✓ Agent根据『整改策略』判断是继续执行，还是回溯到上游某个步骤重新执行，重复这个过程，直到Teacher Agent认为输出结果达到标准

▶ 模型层



PART 03

在AI未来——人机协同新范式

▶ 如何实现和开发者同频 —— 典型场景

代码续写

当我在某文件编写一段代码时，Comate怎么才能基于对代码库的理解，预测出当前文件不存在且需要导入的文件、类、函数？

代码解释

当我要求Comate分别解释代码块、函数、文件、目录、代码库时，它怎么才能从业务角度给出解释？如基于调用链分析给出流程图、ER图、出架构图等？

代码检索

当我新接手一个代码库，需要了解代码架构、具体的业务逻辑，如何才能给出像Mentor一样的讲解？

代码生成

当我希望将整个代码库的Hibernate组件替换成Mybatis组件，期间涉及到实体定义、SQL语句、事务等方方面面的分析，Comate怎么做才能把这件事做好

▶ 如何实现和开发者同频 —— 代码续写

当我在某文件编写一段代码时，Comate怎么才能基于对代码库的理解，预测出当前文件不存在且需要导入的文件、类、函数？

以 `List<WebSearchResult>` 为key，检索哪些函数返回了它，得到这些函数的签名

```
private static String generateWebChunk(CodeSearchResult codeSearchResult) {  
    if (CollectionUtils.isEmpty(codeSearchResult.getWeb())) {  
        return StringUtils.EMPTY;  
    }  
    List<WebSearchResult> webSearchResults = 1
```

```
private static String generateWebChunk(CodeSearchResult codeSearchResult,  
    if (CollectionUtils.isEmpty(codeSearchResult.getWeb())) {  
        return StringUtils.EMPTY;  
    }  
    List<WebSearchResult> webSearchResults = codeSearchResult.getWeb();  
2
```

以 `List<WebSearchResult>` 为key，检索哪些函数把它作为入参，得到这些函数的签名

增强续写效果的知识

公共函数签名

开发规则提取

正在编辑: `WeChatPayController.java`

提取规则和业务特征: `.*Controller.java`, `WeChatPay`

寻找具有相似规则特征的文件: `ALiPayController.java`

提取业务特征: `ALiPay`

寻找具有相似业务特征的文件: `ALiPayService.java`

提取规则特征: `.*Service.java`

组合业务和规则特征: `WeChatPayService.java`

找到以上文件并提取函数签名

▶ 如何实现和开发者同频 —— 代码解释

当我要求Comate分别解释代码块、函数、文件、目录、代码库时，它怎么才能从业务角度给出解释？如基于调用链分析给出流程图、ER图、架构图等？

- API类函数解释：
 - 基于调用链找到向下找到此函数的调用关系，以及对应的Auth、Interceptor、Filter等
- 持久化类函数解释：
 - 基于调用链向上找到此函数的调用关系，以及对应的SQL语句、实体定义等
- 通用函数解释：
 - 同时向上、向下获取N层调用关系

直接获取已经初始化好的项目描述，结构如下：

这是一个用来xxxxx的项目，它的使用了xxxx、xxxx等框架，分为xxxx、xxxx等模块。主要逻辑包括xxxx、xxxx、xxxx。对外暴露了xxxx、xxxx等关键API。该项目有如下重点业务关键词：

- xxxxxx
- xxxxxx

增强解释效果的知识

函数调用链

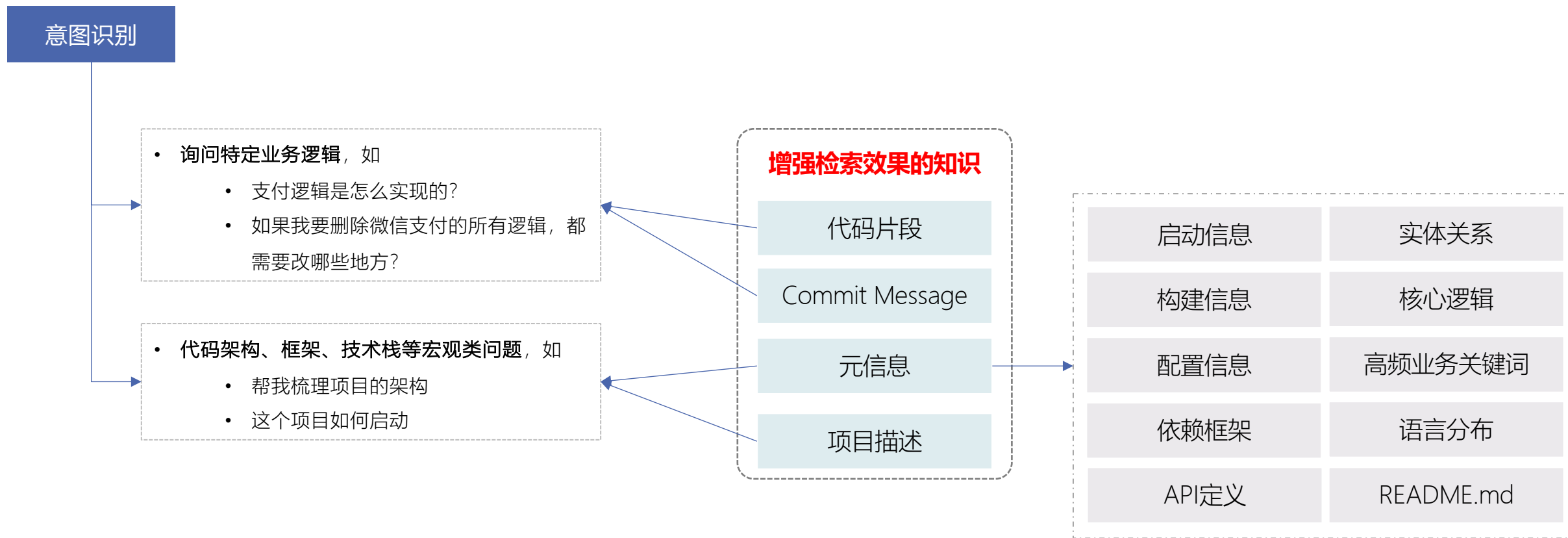
项目描述

```
函数注释 行间注释 代码解释 函数拆分 调优建议
2 usages  ↗ minxiangfer 1 *
private static String generateWebChunk(CodeSearchResult codeSearchResu
    if (CollectionUtils.isEmpty(codeSearchResult.getWeb())) {
        return StringUtils.EMPTY;
    }
    List<WebSearchResult> webSearchResults = codeSearchResult.getWeb()

    List<WebSearchResult> leftWeb;
    if (CollectionUtils.isEmpty(codeSearchResult.getData()) || ignore0
        leftWeb = WebSearchResult.trimWebSearchResult(leftLength, webS
    } else {
        leftWeb = WebSearchResult.trimWebSearchResult( totalCharactersLimit
```

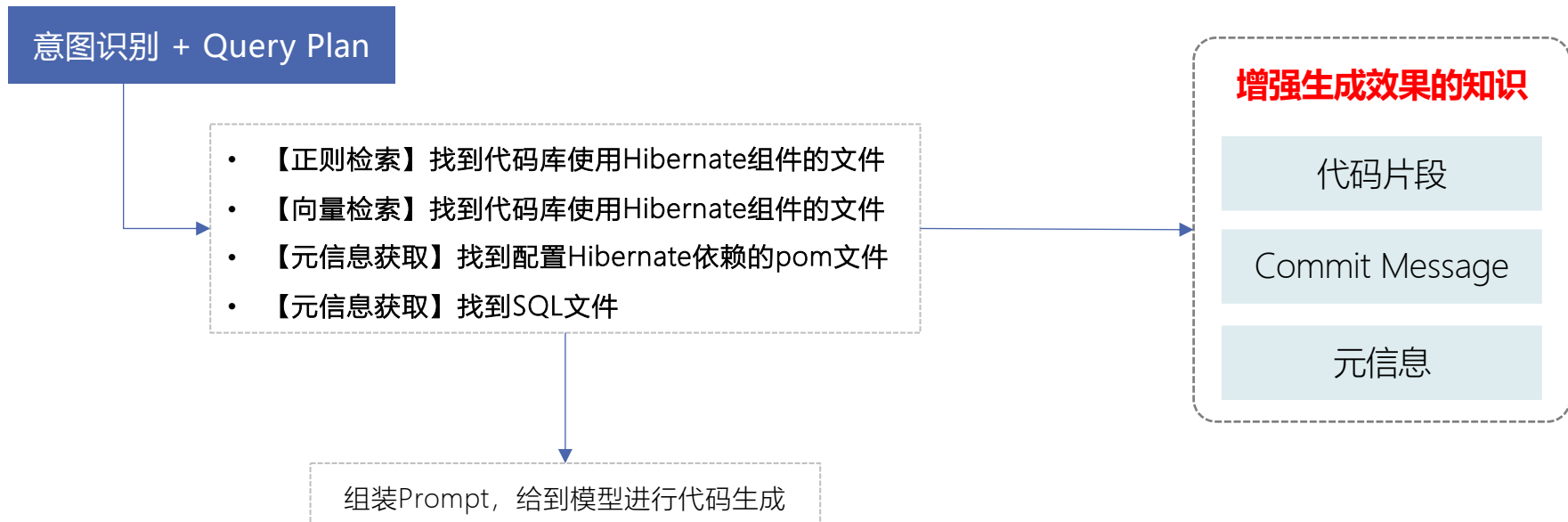
▶ 如何实现和开发者同频 —— 代码检索

当我新接手一个代码库，需要了解代码架构、具体的业务逻辑，如何才能给出像Mentor一样的讲解？



▶ 如何实现和开发者同频 —— 代码生成

当我希望将整个代码库的Hibernate组件替换成Mybatis组件，期间涉及到实体定义、SQL语句、事务等方方面面的分析，Comate怎么做才能把这件事做好？



但这远远不够，只靠一次知识增强没有办法获取全部知识，对于复杂问题的解决没有办法做到立竿见影。需要获取哪些知识大部分是在在解决问题的过程逐步中分析、明确出来的——Agent。

▶ 人机协同新范式——效能提升银弹

AI 对效能的提升是替代Task而不是Job! AI 对效能的提升绝不仅限于编码领域! AI 可以赋能整个软件工程!



▶ 人机协同新范式——人机协同宣言

用好AI将是工程师的基本功

掌握并善用AI将成为工程师的必备能力，率先用好的组织和个人将建立竞争优势。

You are in control

AI能极大的提升研发效率，但需要人确保方向的正确性，最终为业务结果负责。

人机协同，共同进化

繁琐、重复的事情AI来做，创造性的事情人来主导，多使用、勤反馈，实现人机共同成长。

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



上海站

K+全球软件研发行业创新峰会

时间: 2024.06.21-22



敦煌站

K+思考周®研习社

时间: 2024.10.17-19



香港站

K+思考周®研习社

时间: 2024.11.10-12



K+峰会详情



上海站

Ai+研发数字峰会

时间: 2024.05.17-18



北京站

Ai+研发数字峰会

时间: 2024.08.16-17



深圳站

Ai+研发数字峰会

时间: 2024.11.08-09



AiDD峰会详情



2024 AI+研发数字峰会

AI+ Development Digital summit

深圳站 11/08-09



AI 驱动研发变革 促进企业降本增效

2024深圳站-议题设置

AI+产品线	LLM驱动产品创新	LLM驱动需求与业务分析	AI驱动设计与用户体验
AI+开发线	AI 原生应用开发框架与技术	AI Agents在研发落地实践	LLM驱动编程与单测
AI+测试线	LLM驱动测试分析与设计	基于LLM生成测试脚本与数据	LLM和AI应用的评测
AI+工程线	AI+DevOps 与工具 (LLM 时代的平台工程)	大模型对齐与安全	端侧大模型与云端协同
AI+领域线	领域大模型 SFT 与优化	知识增强与数据智能	大厂专场

扫描右侧二维码
查看更多会议详情



早鸟票限时抢购中 (截止到9月30日)

¥ 3680

早鸟票

¥ 2800

学生票

THANKS

