



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

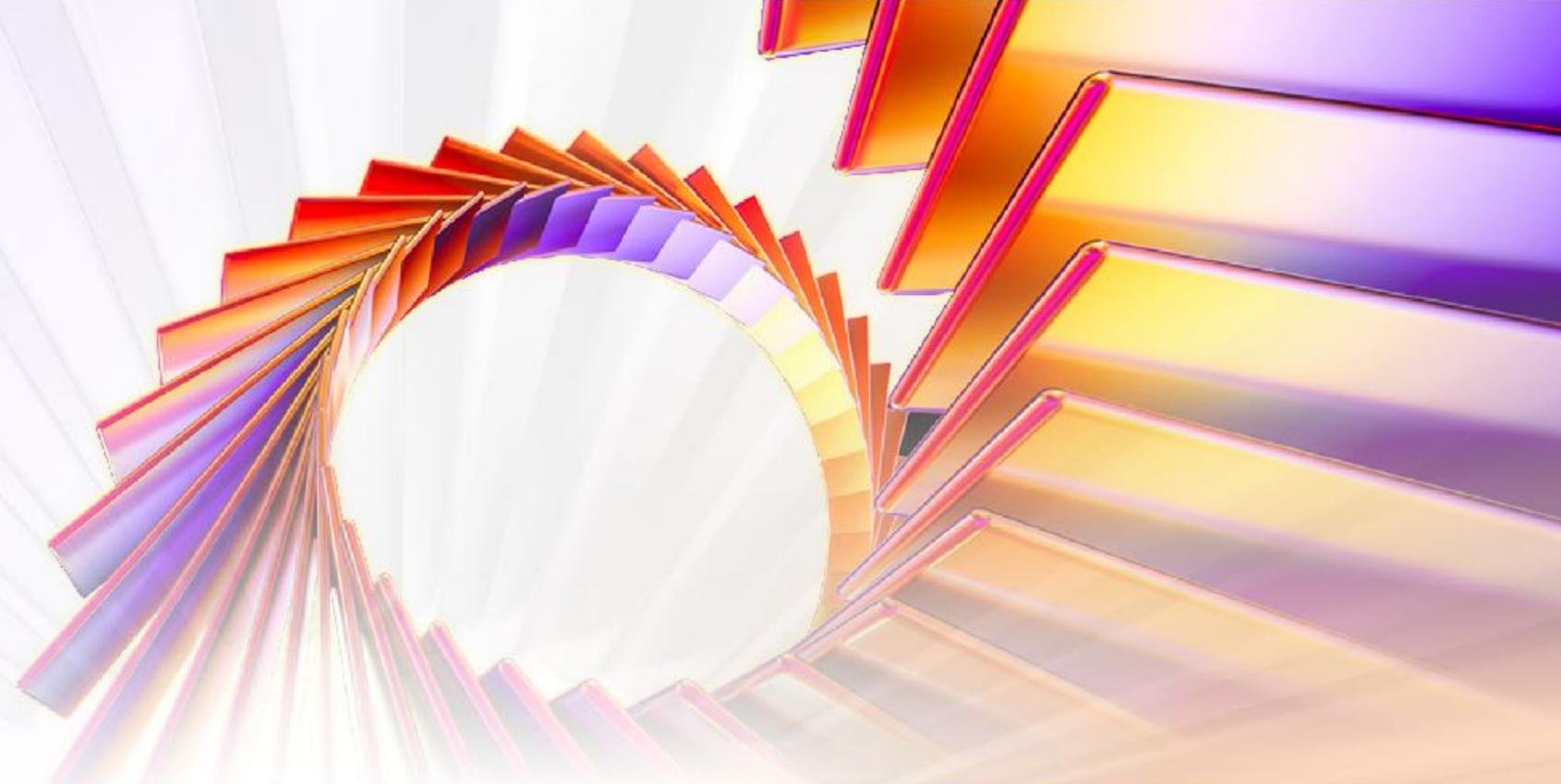
AI 前沿技术探索与实践

NiDD  | 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



推动研发智能再升级：

通义灵码在 AI Coding 领域的实践

张燎原 | 阿里云

# 软件工程

是人类历史上第一次大规模集体智力协作活动。

# 规模带来的挑战

“突破规模法则的限制，就必须创新，要么改变系统的物质组成，要么改变其结构设计，要么二者均发生改变。”



杰弗里·韦斯特

## 系统组成

(人, 技术)



## 结构设计

(组织结构、技术架构和协作方式)

研发效率提升 =  $f$ (协作效率、个体效率)

# 集体协作、知识共享 - 隐性的知识显性化

效率提升假设一：集体协作和共享知识的方式进行软件开发，通过 AIGC 将隐性知识“显性化”

## 软件工程

## 是社交学习过程。

### The Nature of Software Development: Social learning

“Because software is **embodied knowledge**, and that knowledge is initially **dispersed, tacit, latent, and incomplete**, software development is a **social learning** process.”

*Howard Baetjer, Jr.: Software as Capital. IEEE Computer Society Press, 1998*

# 企业的数智化增强回路

研发智能化提效行动点一：构建企业研发数智化大脑，隐性的研发知识显性化

## 企业一线开发者

任务协同 架构设计 代码编写 软件测试 问题排查 软件发布 日常运维 知识查询

## 智能研发工具

智能编码 智能评审 智能项管 平台工程 智能问答 AutoDev 个人助理 智能洞察

## LLM 智能大脑

自定义 Prompt

RAG 检索增强

Agent 平台

### 基础模型能力

代码补全模型

研发问答模型

企业专属模型

模型训练SFT

实时增强

## 知识梳理

确定优化目标

梳理资产

数据清洗

安全隐私

## 软件研发全生命周期



# 人工智能带来的新的人机协同模式

效率提升假设二：为了降低研发人员的工作脑负载，让每个职能都有基于 AIGC 的“副驾”和“助理”

## LLM as Copilot

## LLM as Agent

## LLM as Multi-Agents

阶段一		阶段二		阶段三	
辅助完成任务		自主完成任务		协同处理复杂任务	
不改变软件工程专业分工，增强领域专业技术，AI 研发工具辅助人完成任务		单一职能专家，能够自主使用工具完成预定任务		影响整个软件研发过程，多Agent互相协作完成复杂工作	
工具	助力人员提效	工具	独立完成工作	工具	与人协同共生
人	主导、提示及确认	人	给定上下文，完成知识对齐	人	负责创意、纠偏及确认

# 智能化研发工具实现路径三阶段

研发智能化提效行动点二：问题域的扩大及技术的突破，不断拓展智能化研发工具的能力边界

通义灵码 1.0

通义灵码 2.0

通义灵码 Next

辅助编码

人：技术设计、提示和确认

代码

人类

AI

协同编程

人：任务澄清、干预、结果审查

任务

人类

AI

自主编程

人：创意、需求澄清、结果验收

需求

人类

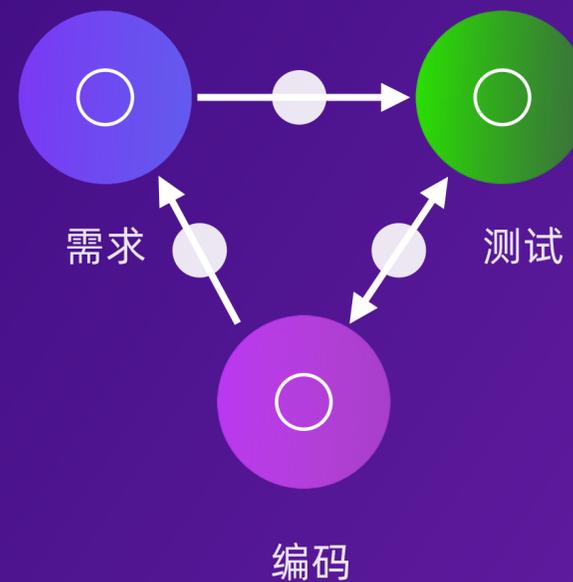
AI

# Multi-Agents 对软件研发范式的影响

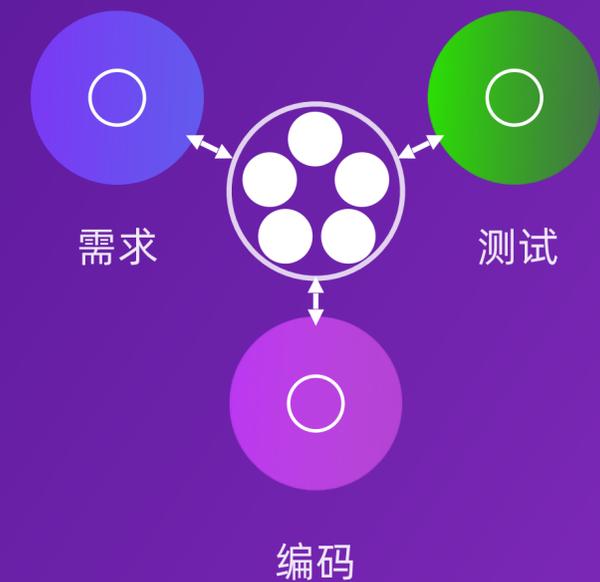
Copilot 辅助



Copilot 辅助 + 单任务 Agent



Multi-Agents 协同



○ AI 助手 ● AI 智能体

# 落地研发智能化所面临的困难与挑战

## 工具太多，选择困难症

AI 辅助编程的工具有很多，各个厂商都有其各自宣传的点和竞争力。由于企业缺少领域 Know-How，面对这些工具，难于根据企业的特点，做出正确的选择。

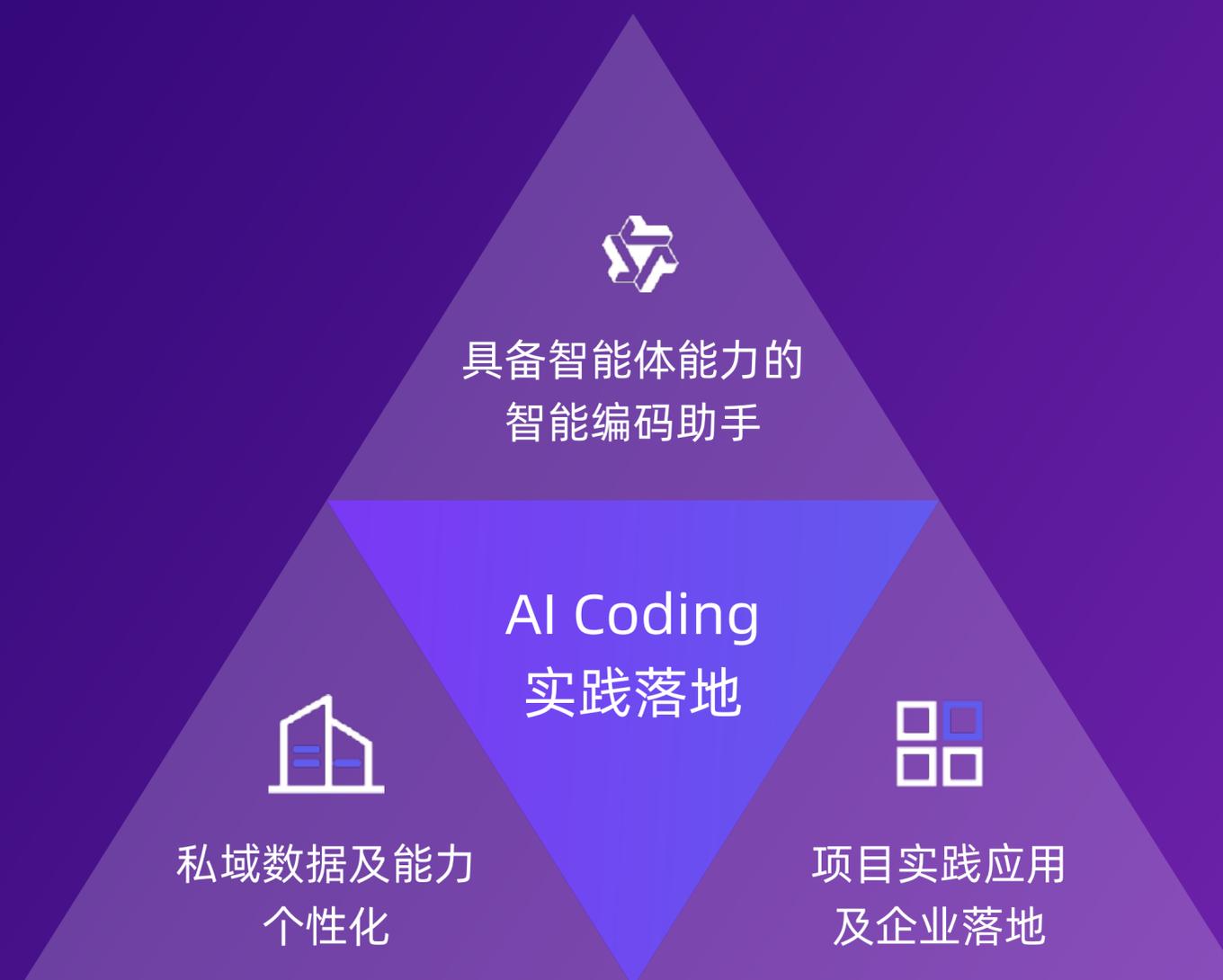
## 个性化诉求难满足

由于训练语料的限制，生成的内容往往无法满足垂直领域的实际业务需求。在软件开发中涉及诸多专业任务，大语言模型并不擅长直接处理这些任务，仍需依赖专门的工具。

## 大工程 AI 编程难落地

AI 可以快速完成一个轻量应用的构建，但是放在一个大型工程中的时候，却屡屡翻车。如何在大型工程里更好地应用 AI 能力，成为业内共同的话题。

# AI Coding 落地的“黄金三角”



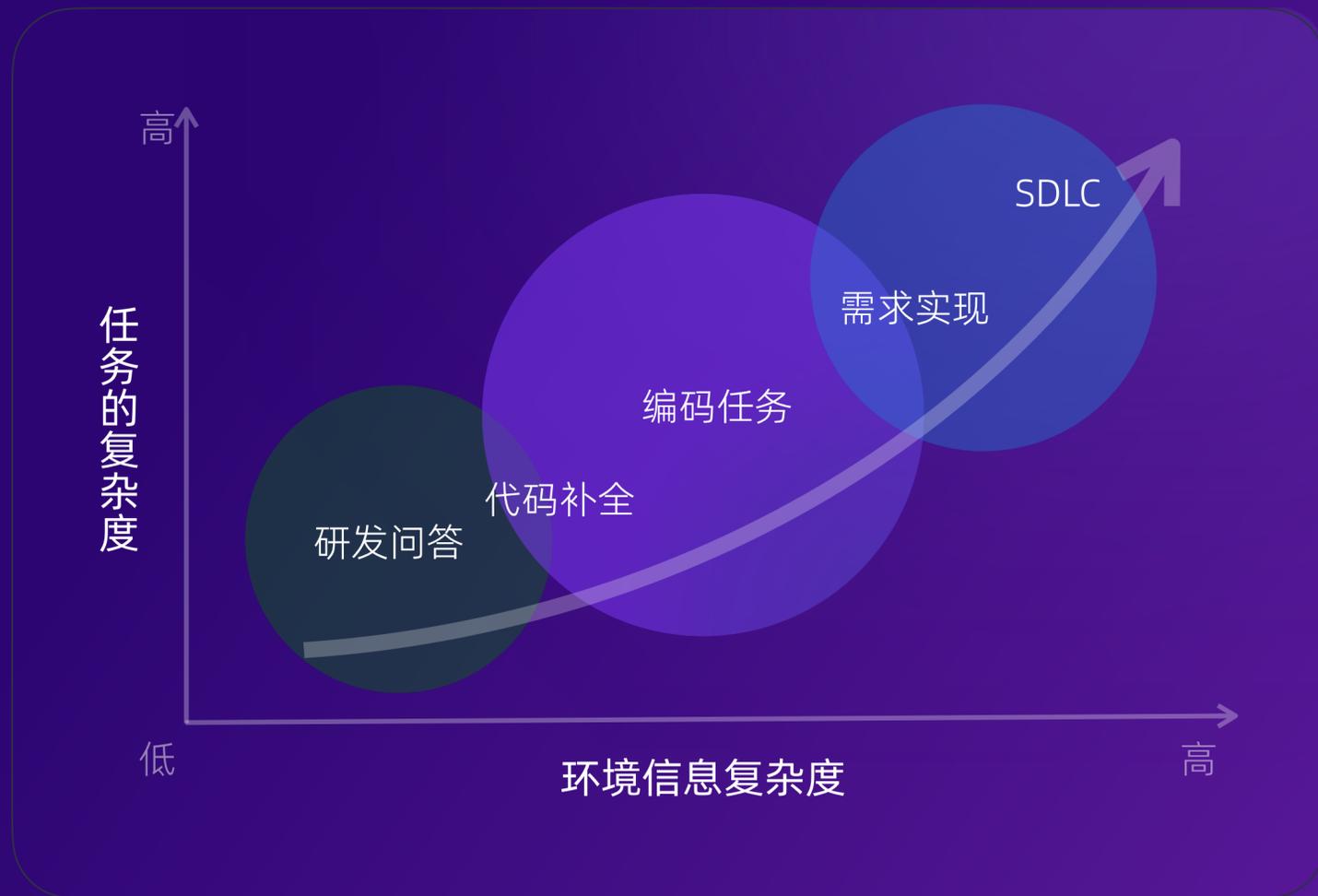
# AI Coding 落地的“黄金三角”



# AI 编码的产品演进和技术发展

应用场景在扩大，产品边界在延伸

技术发展带来产品的更多可能性



# AI Coding 符合 Agent 的场景

在选择使用 Agent 的场景时，来自 Anthropic 的 Barry 给出了一个非常实用的标准：

“Agent 最适合的场景是那些既复杂又有价值，但失败后的风险较低或监控成本不高的任务。”

# Agentic AI Coding Assistant

By deeply understanding development contexts and toolchain interactions, AI systems generate context-aware code at scale, significantly accelerating software development workflows.

## BEFORE

## AFTER

Code Snippet Generation



**Multi-File Code Synthesis**  
(from task descriptions)

Copilot-Style Assistance



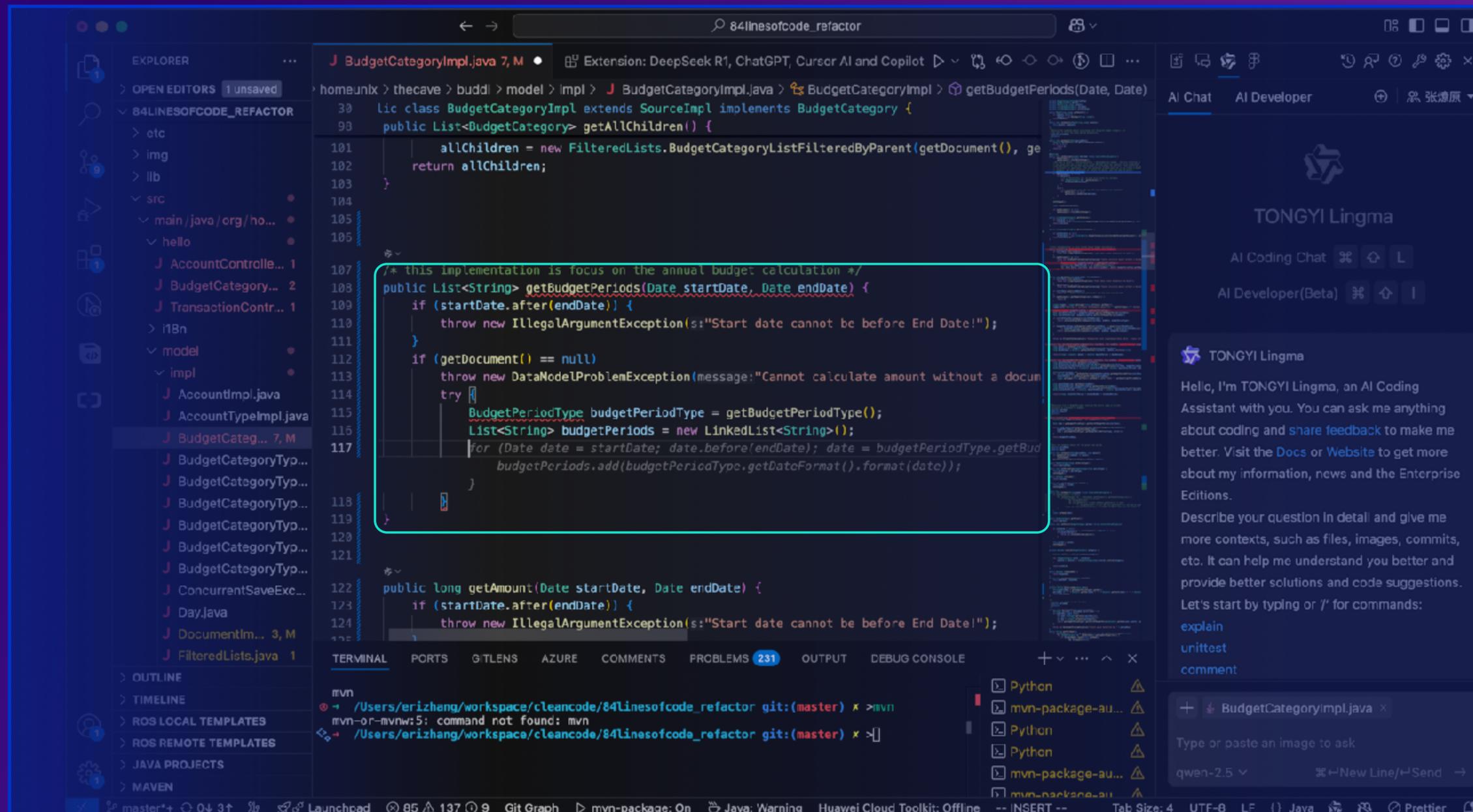
**Agent-Driven Development**  
(with contextual awareness and toolchain integration)

Syntax-Centric Coding

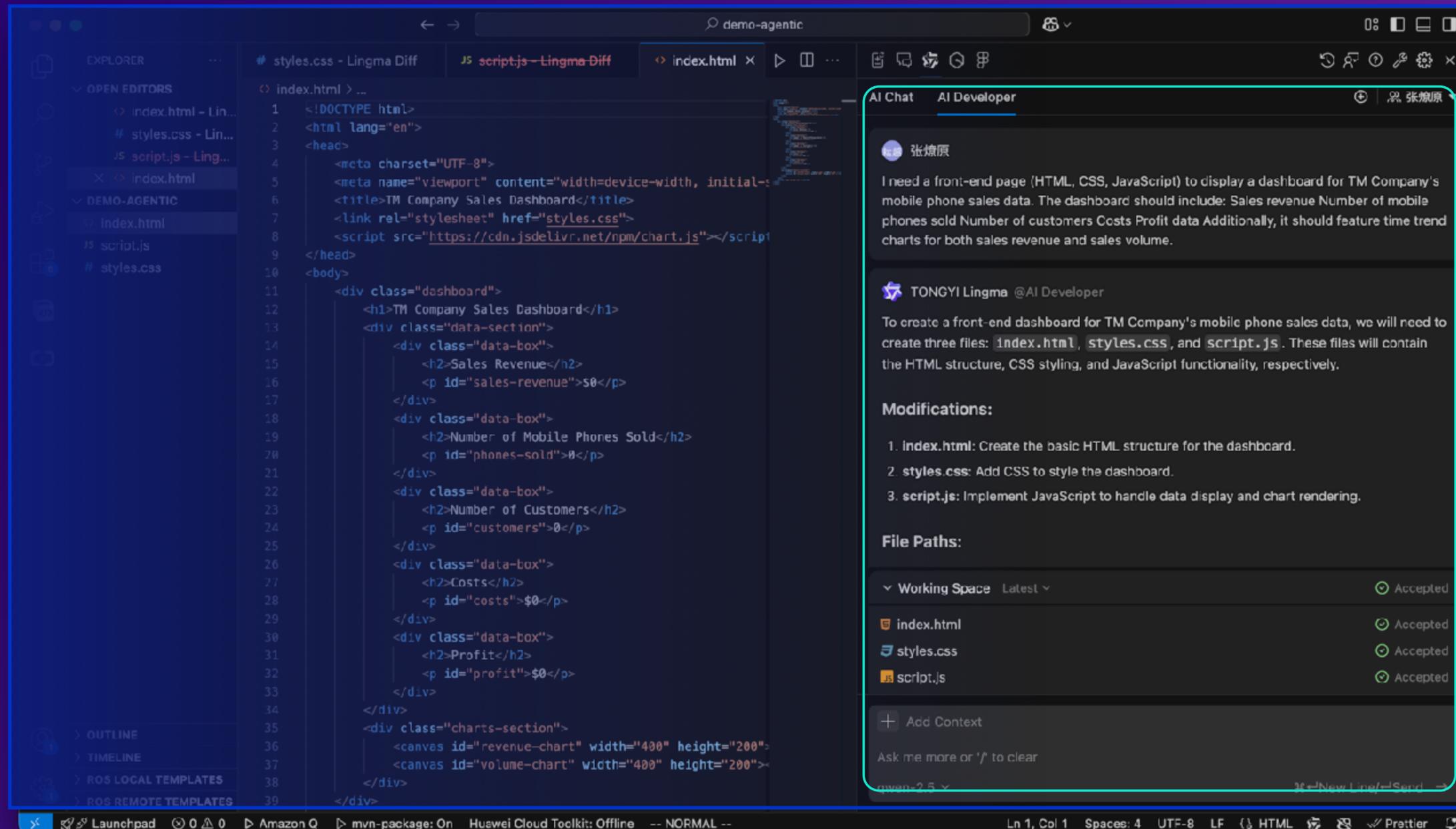


**Natural Language Programming**

# BEFORE: Syntax-Centric Coding - Code Completion



# NOW: Natural Language Programming - Iteration Chat



# 通义灵码 2.5 版本能力升级

1 多种会话模式

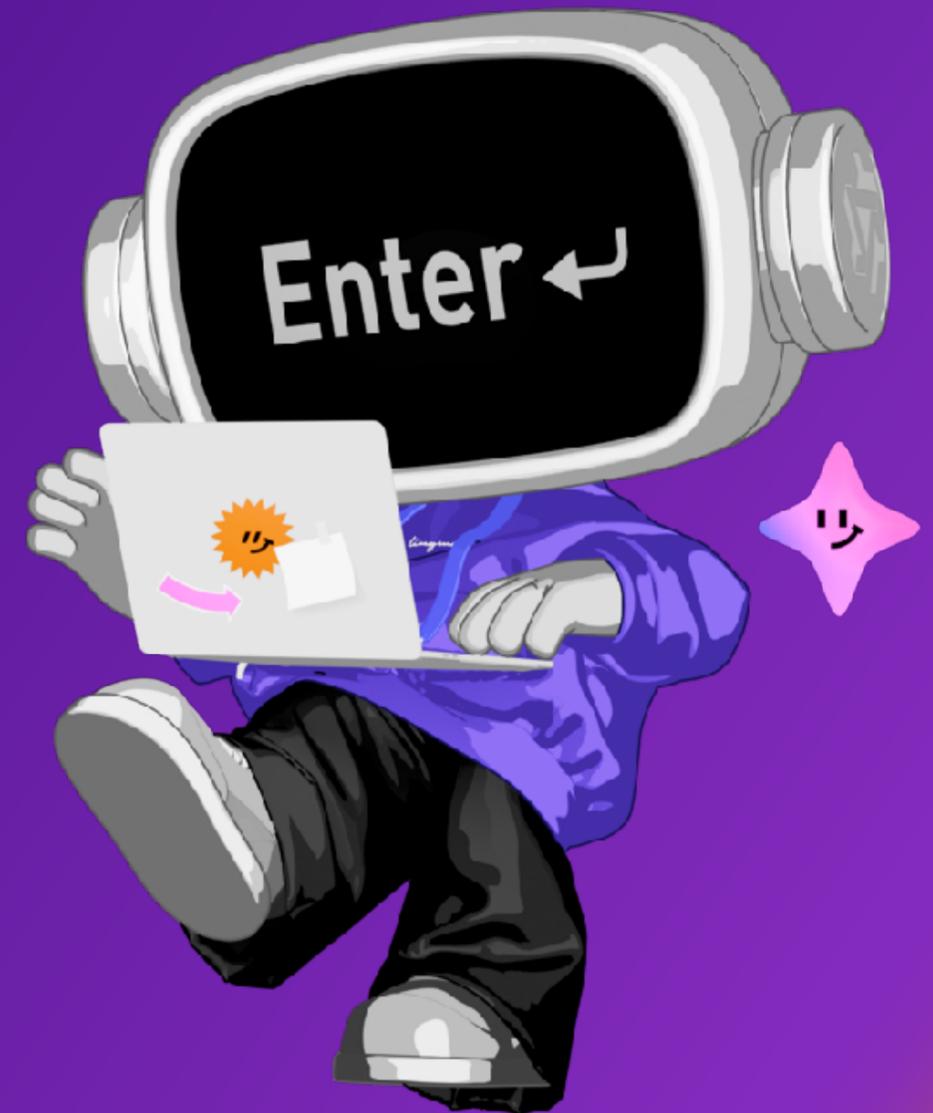
2 工程自动感知

3 工具使用(含MCP)

4 终端命令执行

5 工程级变更

6 记忆感知



# 示例一：通义灵码官网还原



# 示例二：记账 APP 实现



# AI Coding 落地的“黄金三角”



# 通义灵码企业版，懂研发更懂企业

提供企业标准版、企业专属版等多种面向企业客户的方案，并提供企业个性化和扩展能力，助力企业构建高效、安全、符合企业特点的智能化研发环境。

## 统一管理更便捷

企业级账号、许可及权限管理，让企业级服务管理更便捷

多种账号身份集成

统一许可授权

精细化权限管理

## 安全合规更放心

全方位加密、网络和安全策略，助力企业构建安全、合规的研发环境

全程数据加密和防护

专属 VPC 实例部署

网络白名单管理

## 企业扩展更灵活

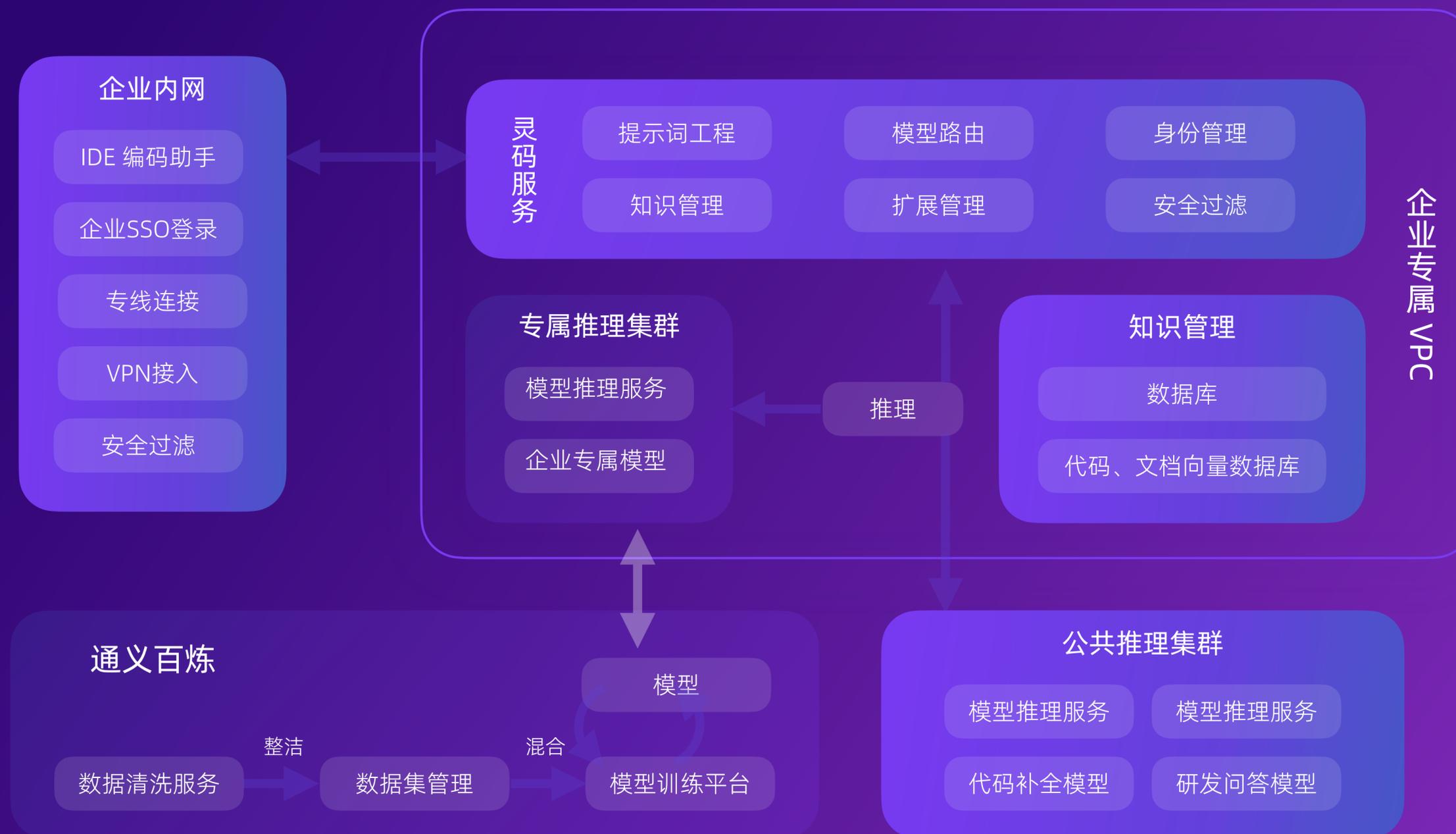
满足企业个性化研发场景，及私域数据代码补全和研发问答增强诉求

自定义扩展及集成

代码补全检索增强

企业知识库问答

# 专属 VPC 部署模式：更高的安全性和资源隔离



# 代码安全及隐私全程保障

## 全程数据加密

基于 https 加密传输，并支持身份鉴权、内容安全防护等，全程保护数据安全。

## 不会存储或进行二次训练

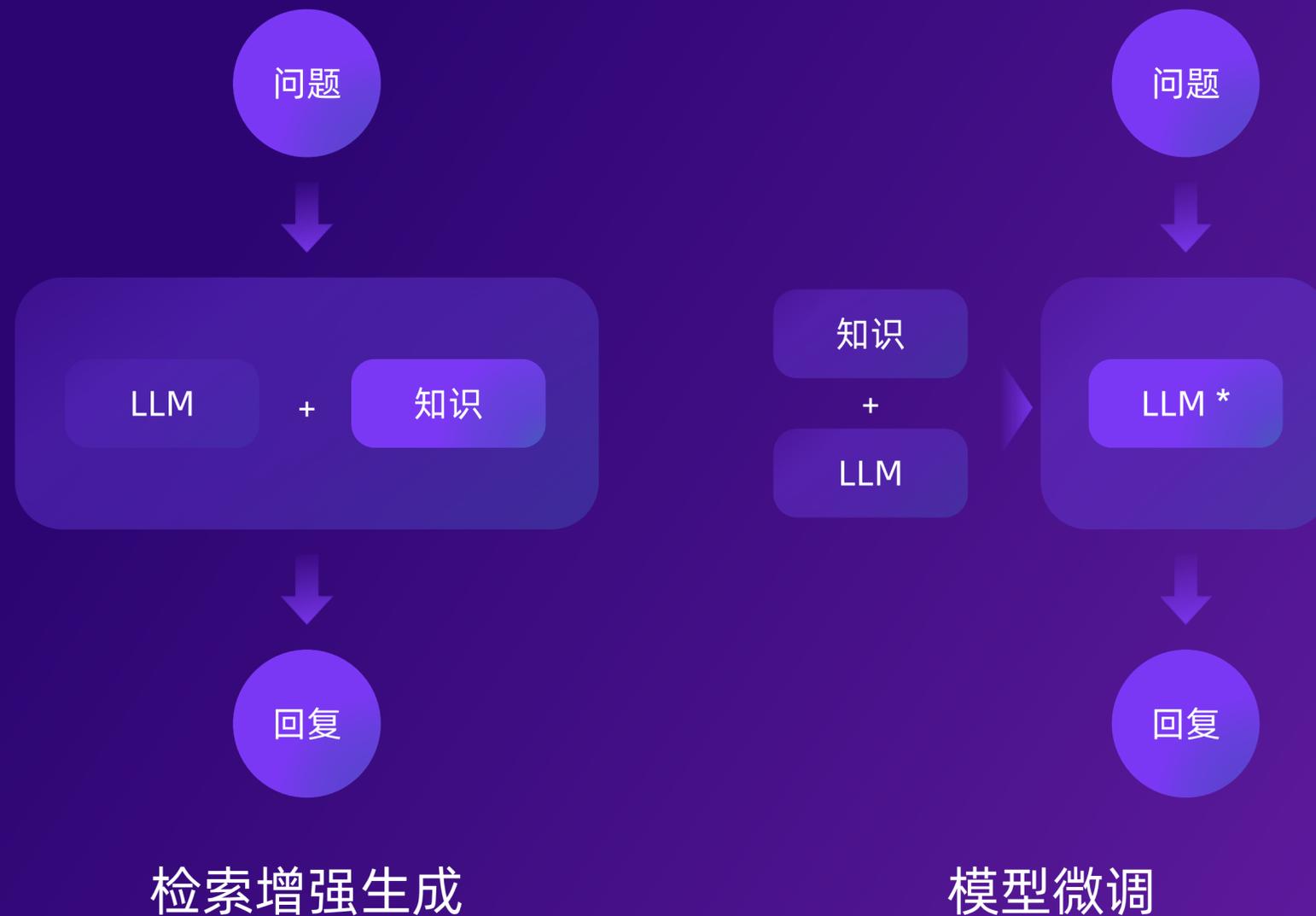
代码上下文仅用于模型推理，全过程数据不落盘，不会使用你的代码和数据进行二次训练。

## 开源合规的训练数据

基于通义大模型进行编码领域针对性训练，所有用于训练的代码数据均为开源合规的数据。



# 解决企业私域化数据个性化的方法：检索增强生成和模型微调

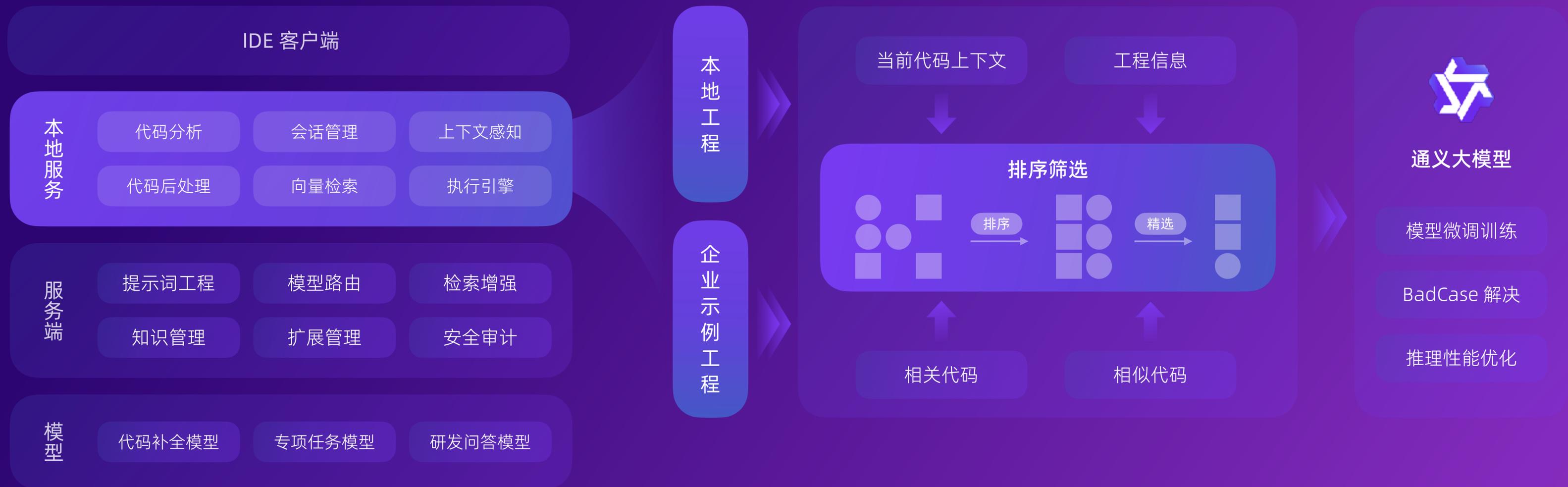


场景	检索增强生成	模型微调
数据类型	动态	静态
数据规模	小	大
成本	低	高
延时	大	小
时效性	好	差

各维度对比

# 提升企业开发者编码效率：

代码补全实现全工程理解，全语言平均采纳率超过30%



# 企业级检索增强：解决个性化问题



# 自定义扩展：解决能力宽度问题

## 扩展指令 /



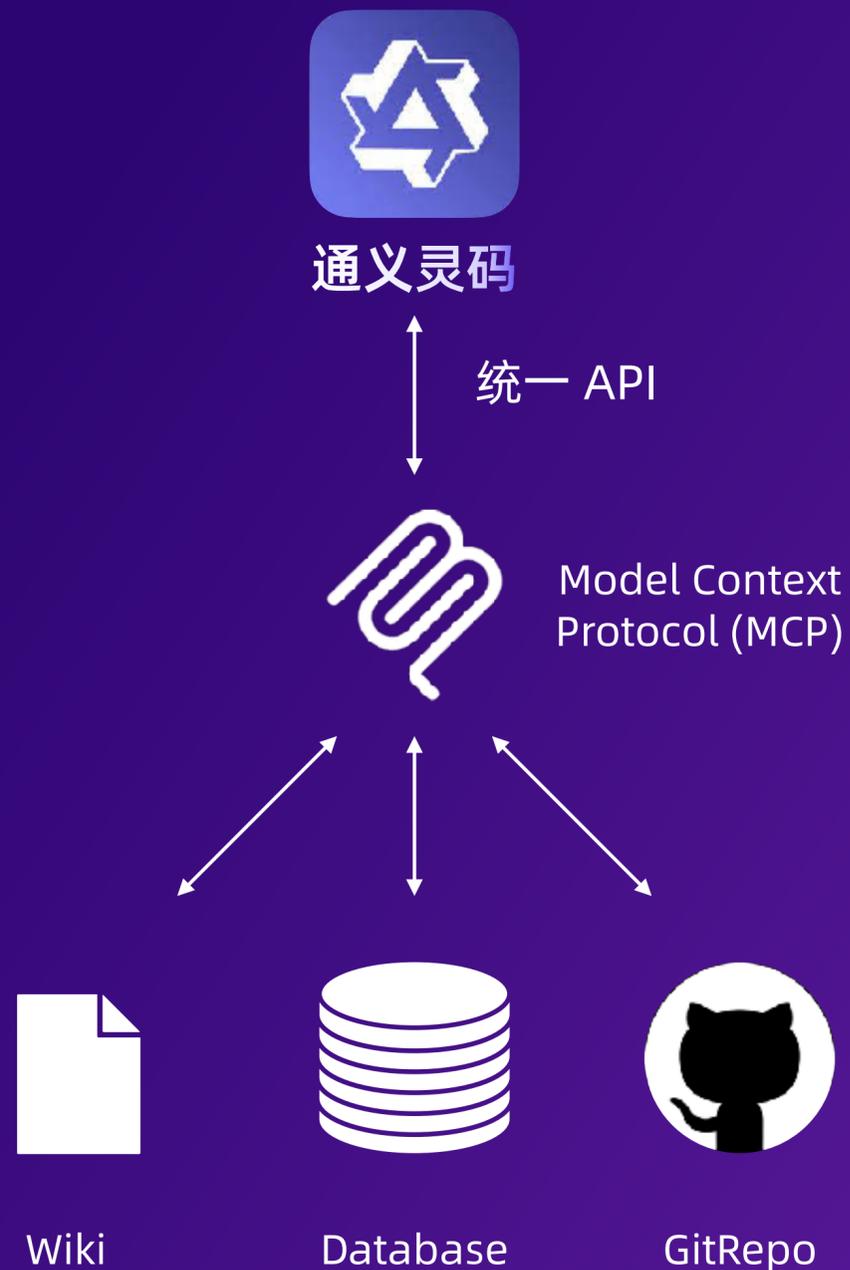
## 扩展上下文 #



## 扩展智能体 @



# MCP 解决大模型与外部工具的交互问题



# 从工具到方案，又全又实用



## DAO 模块代码自动生成

自动读取数据库表结构，遵循 MyBatis 框架编写 XML、Mapper、DAO、PO 等代码文件



## 辅助命名，注释翻译

结合功能注释按照固定规范对类、方法、变量进行命名。对英文注释进行翻译，辅助阅读代码



## API 开发文档自动生成

按照接口定义以及固定规范，自动生成接口描述文档以及示例代码，辅助开发者编写相关文档



## 团队自定义规范代码检查

按照团队自定义规范进行代码变更检查，提前发现潜在安全、缺陷风险，提升代码可读性

一个开发者在 Java 开发时典型场景扩展

# 从工具到方案，又全又实用

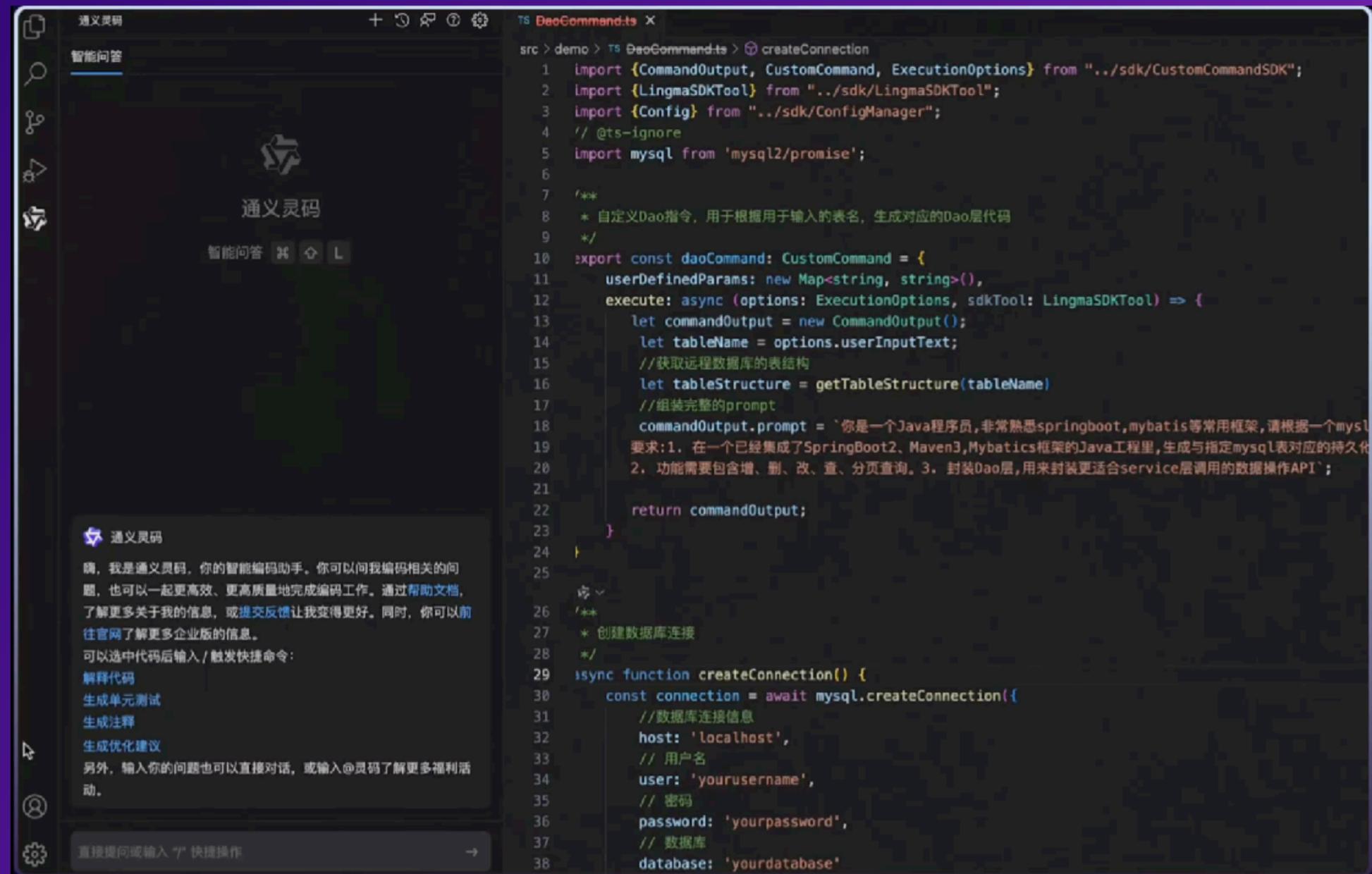
## DAO 模块代码自动生成

自动读取数据库表结构，遵循 MyBatis 框架编写 XML、Mapper、DAO、PO 等代码文件

## 辅助命名，注释翻译

## API 开发文档自动生成

## 团队自定义规范代码检查



# 从工具到方案，又全又实用

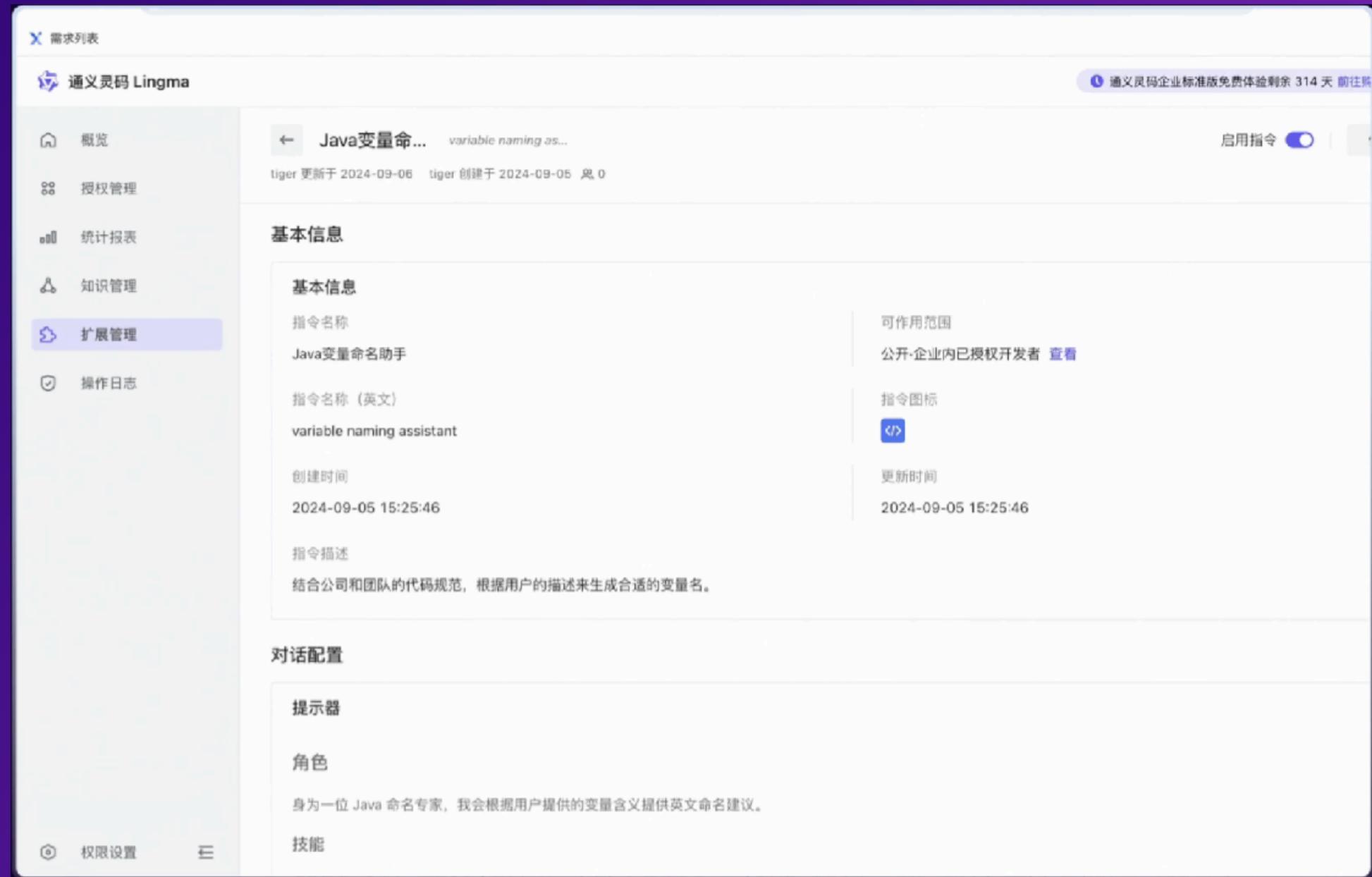
DAO 模块代码自动生成

辅助命名，注释翻译

结合功能注释按照固定规范对类、方法、变量进行命名。对英文注释进行翻译，辅助阅读代码

API 开发文档自动生成

团队自定义规范代码检查



# 从工具到方案，又全又实用

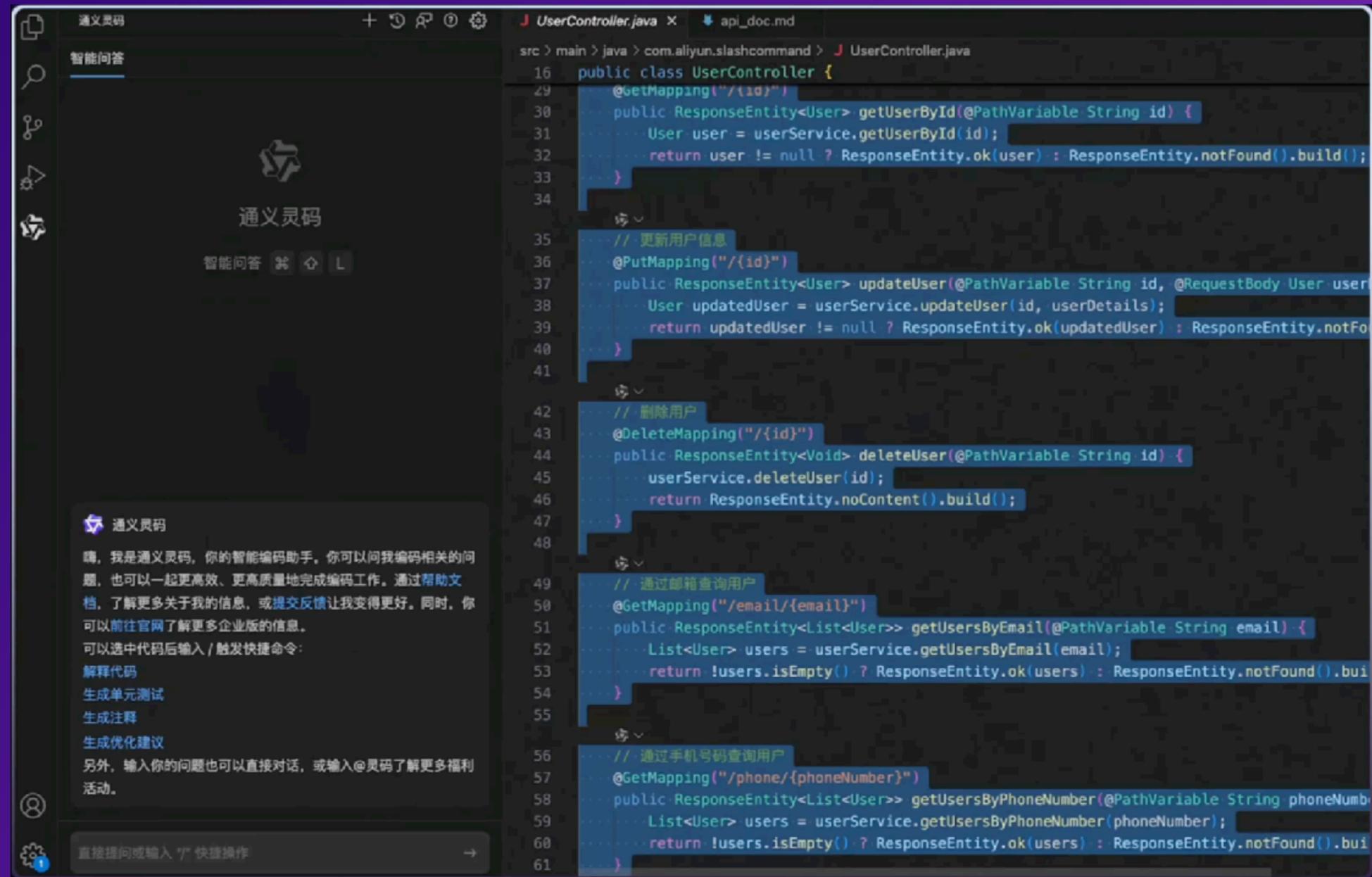
DAO 模块代码自动生成

辅助命名，注释翻译

API 开发文档自动生成

按照接口定义以及固定规范，自动生成接口描述文档以及示例代码，辅助开发者编写相关文档

团队自定义规范代码检查



# 从工具到方案，又全又实用

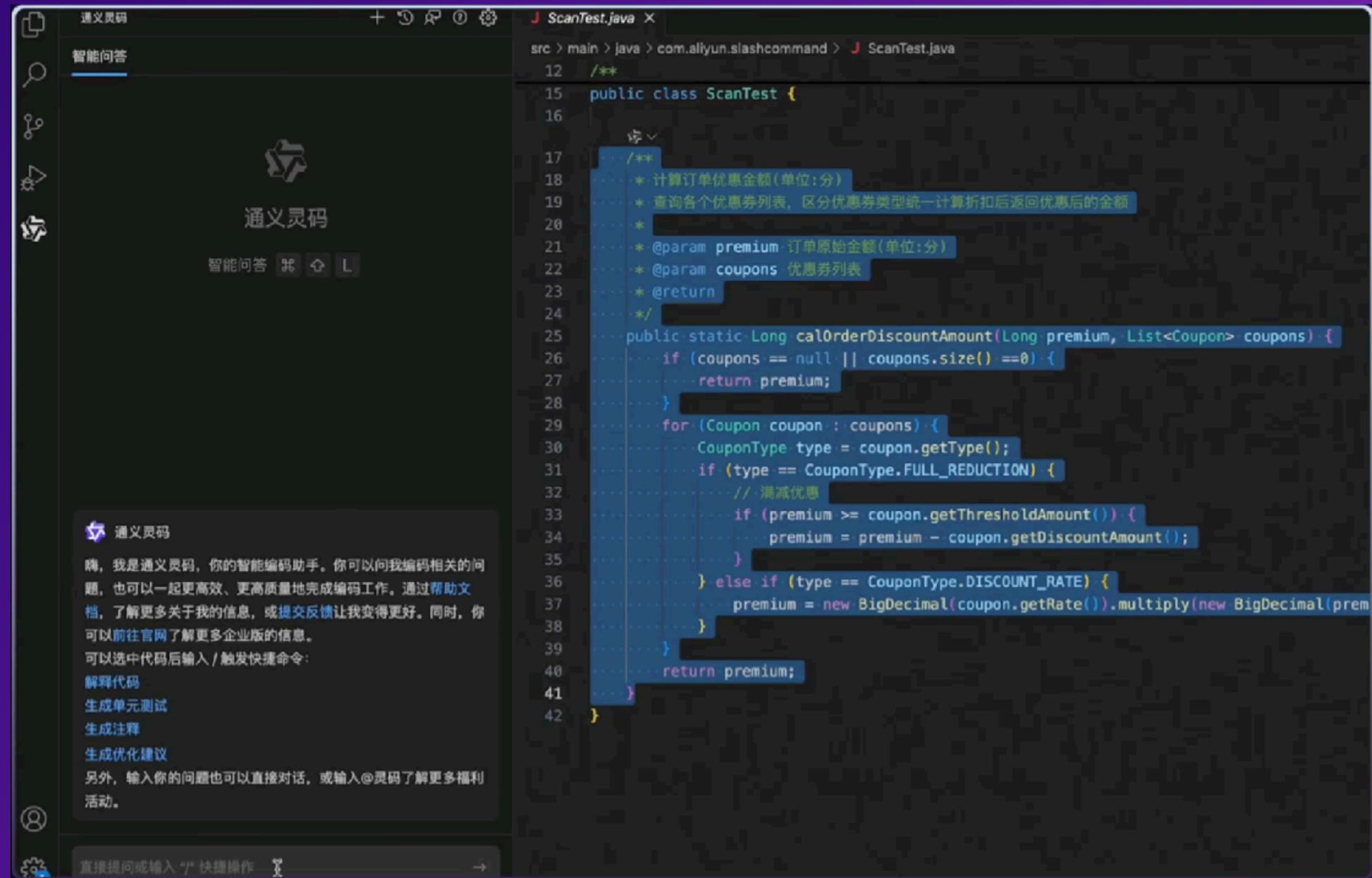
DAO 模块代码自动生成

辅助命名，注释翻译

API 开发文档自动生成

团队自定义规范代码检查

按照团队自定义规范进行代码变更检查，提前发现潜在安全、缺陷风险，提升代码可读性



# AI Coding 落地的“黄金三角”



# 指标不是不够，而是太多

生成字符数

代码补全建议次数

问答点赞率

代码变更行数

代码补全触发次数

代码补全活跃数

激活用户数

变更频率

吞吐率

开发并行度

采纳次数

平均恢复时长

智能问答次数

AI 代码生成占比

采纳率

变更成功率

平均恢复时长

智能问答活跃数

检索增强召回率

测试覆盖率

代码补全次数

编码时间占比

代码耦合度

变更时长

单位编码时长

渗透率

活跃开发者数

代码推荐率

采纳代码行数

代码入库率

# 规范统一度量标准，回答本质问题

内部

**能力和行为指标**

指向具体的改进内容，需要通过更外部的指标反映改进的必要性和改进效果。

**交付效能指标**

在产品研发中代理业务结果指标，帮助了解研发的交付效能状况，但依然无法直接改进。

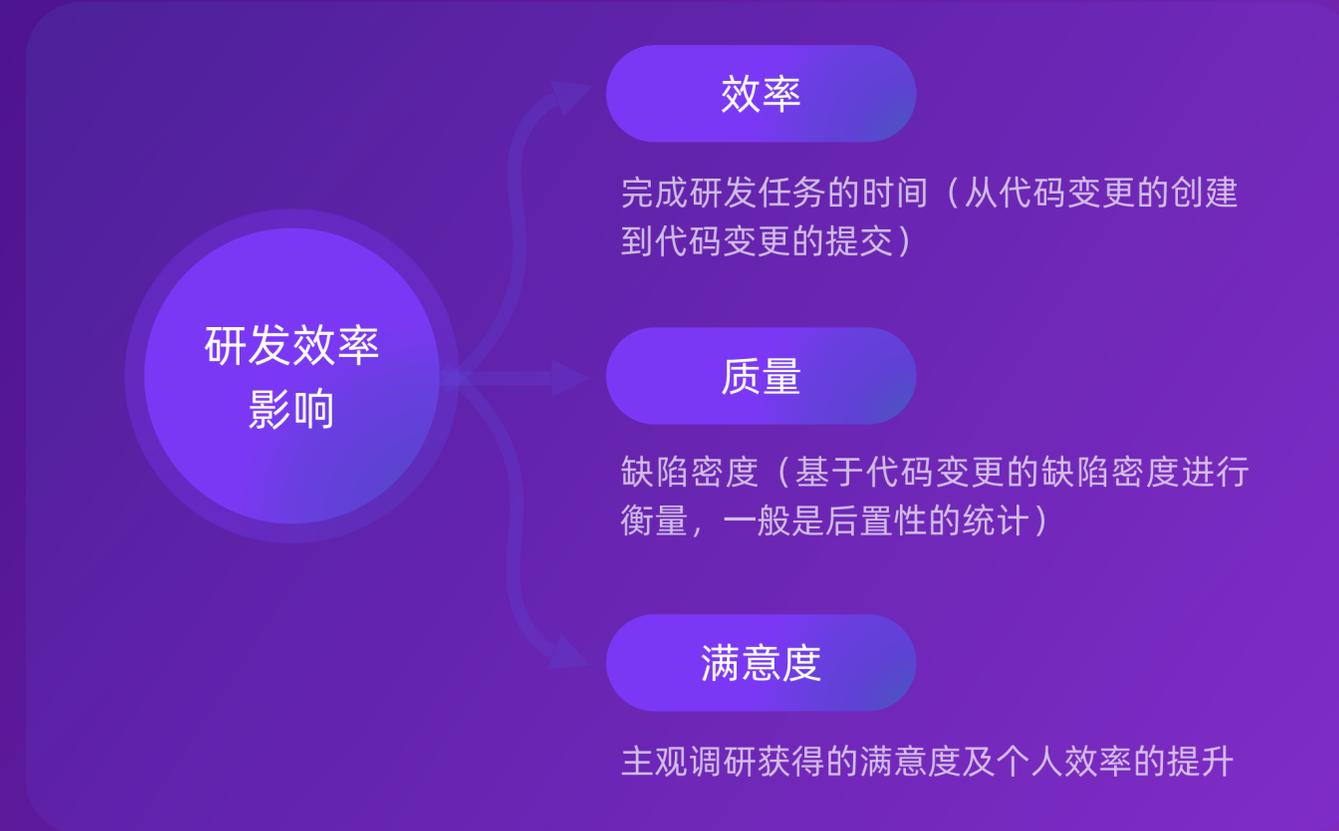
**业务结果指标**

反映真实的成效，但无法直接指导团队改进

外部

改进和授权

导向和要求



# 度量是回答本质的问题和引导正确的改进

当前使用情况如何？

激活用户数

月活用户数

代码补全占比

研发问答占比

运营推广

开发者培训

实际使用的效果如何？

AI 代码生成占比

代码采纳率

问答点赞率

采纳的代码行数

最佳实践推广

研发工具改良

对企业带来了什么收益？

需求吞吐率

开发吞吐率

代码缺陷密度

开发者满意率

单点工序改进

研发流程优化

# AI 辅助编码对于程序员的工作效率提升



程序员花费三分之一的时间编写新代码或改进现有代码（32%）。花费 35% 的时间来管理代码，包括代码维护（19%）、测试（12%）和响应安全问题（4%）。另外 23% 的时间花在会议以及管理和运营任务上。

数据源：程序员的时间分配：<https://thenewstack.io/how-much-time-do-developers-spend-actually-writing-code/>



# 大规模工程使用 AI 研发的提示和技巧

1

在专业 IDE 里使用，让调试、构建反馈更及时。

2

把 AI 当成初级程序员，构建好的质量围栏，测试驱动。

3

先从质量入手，再从非核心系统，逐步过渡到核心系统。

4

分而治之，将大问题分解成小问题，逐步实现。人类负责大问题，AI负责小问题。

5

将 Sketch 阶段从 0 到 1 的任务交给 AI 来完成。

6

善于使用文档来记录全局设计和历史操作，形成经验等长期记忆。

7

勤于用 AI 更新项目及代码释文档，将代码实现和文档保持一致性。

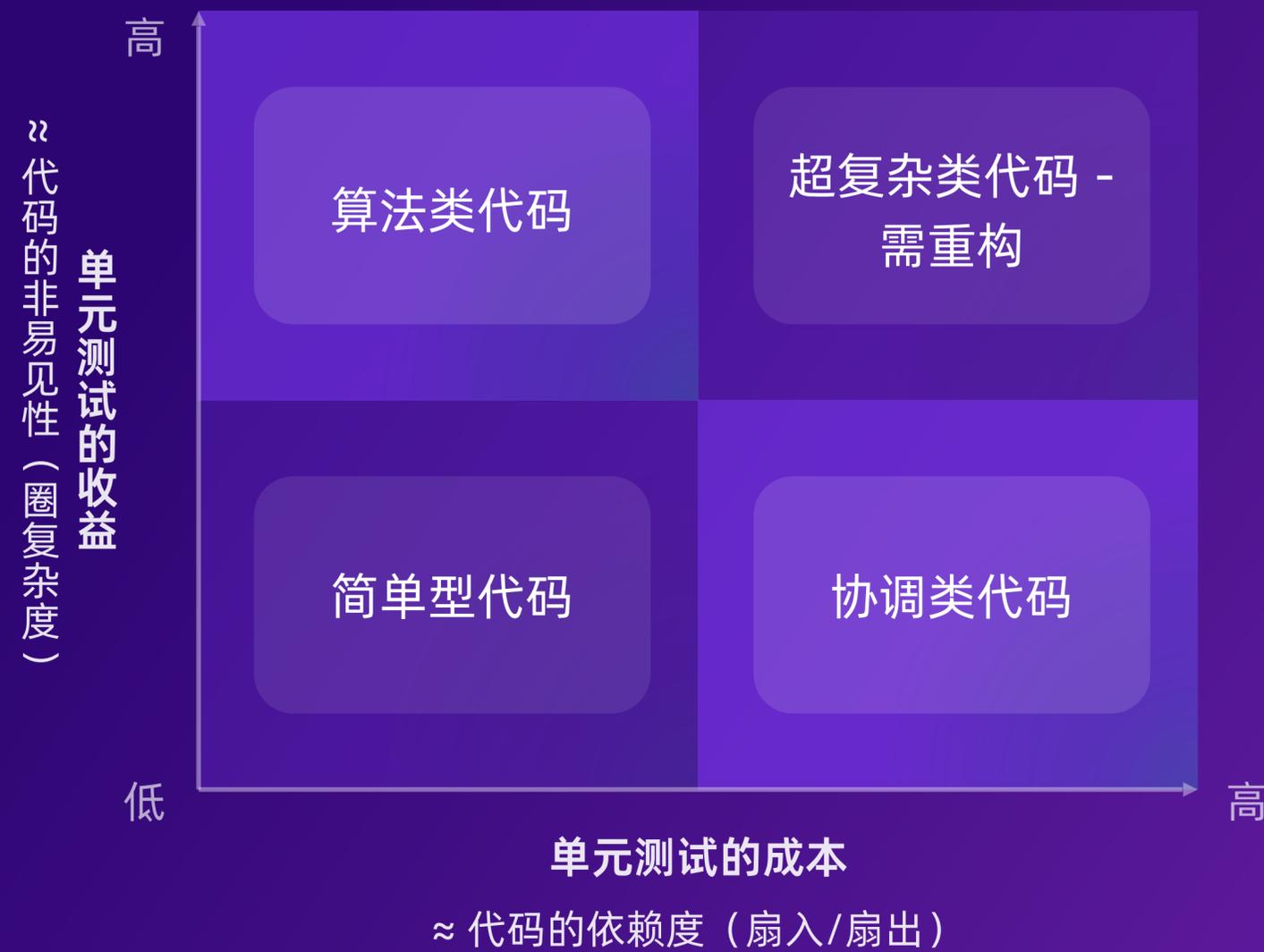
8

善用 Git 管理代码变更版本，保证可回滚可追溯。

9

善于构建企业知识库。建立最佳实践广场。

# 企业在维护遗留代码时的最佳实践

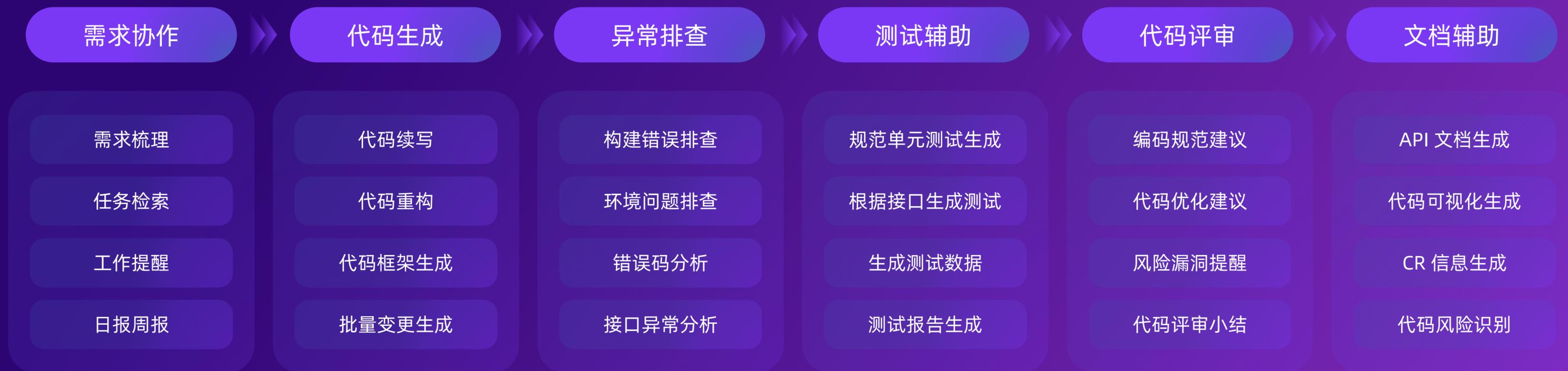


单元测试生成：算法类代码

代码优化建议：复杂类代码

接口测试辅助：协调类代码

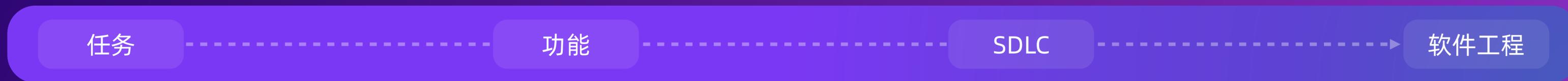
# 全面覆盖各类场景，企业智能化研发升级



可自定义的指令，可扩展的上下文，可集成的工具

一切皆代码，一切皆模型

从面向命令，面向声明，再到面向意图



# 场景演进应该是一个持续收益的过程



## 研发智能化的路径

### 阶段一：引入 DevOps 大模型，构建研发智能大脑

引入 DevOps 基础大模型，以编码场景进行落地，取得最大效果红利。构建研发智能化增强回路。Landing

### 阶段二：以长期效率为核心，持续建设和治理数字资产

以长期效率为核心，提供全量、全要素的数据，落地企业数字资产管理及治理能力，构建企业个性化数据集。企业个性化数据输入，为大脑提供养分。Growth

### 阶段三：以大模型为中心，完成研发工具智能化升级

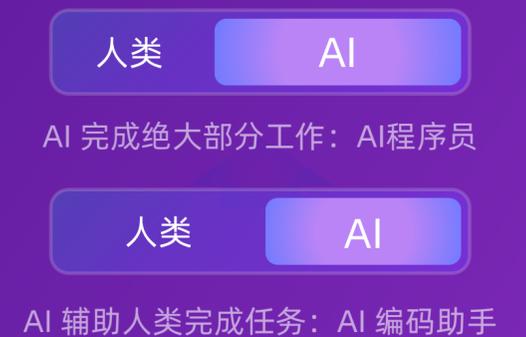
以大模型为中心，将研发领域的各种工具进行智能化接入，具备智能化能力，实现端到端研发工具的智能化升级。同时，简化流程和角色，使用大模型进行替代，降低协作成本。Expanding

# 企业软件研发工具链的演进



## 从编码场景逐步延伸至研发领域的其他方面

以效率为核心。从编码场景入手，逐步将研发领域的各种工具进行智能化接入，具备智能化能力，实现端到端研发工具的智能化升级。同时，简化流程和角色，使用大模型进行替代，降低协作成本。



## 构建企业研发数智化核心驱动力

以数智大脑为核心，提供企业核心代码资产和研发组织资产，落地企业数字资产管理及治理能力，构建企业个性化私域知识库，持续为数智大脑提供养分，提升大模型生成效果。



# 百万开发者、上万企业使用通义灵码

ul 接口、生成接  
客户端，实现接

“我用灵码创建工程、构造 Restful 接口、生成接口标准化文档、生成不同语言的客户端，实现接口开发、集成和联调。”

“用灵码快两年，我第一次尝试结AI工具来辅助代码审计，它快速定位多个潜在的安全漏洞，极大地提升了审计工作的效率。”

“AI 在辅助编程... 灵码是国内出... 是现代开发者的...”

流程图。通过通  
高定各种绘图需

“在灵码的帮助下，我使用开源的 Spring AI Alibaba 框架开发了AI智能答疑助手，提升了云产品的用户体验和答疑效率。”

“对于我这种 C 语言开发小白，通义灵码的智能问答和代码补全功能让我能够将 C 语言改写成 Python。”

“通义灵码企业... 码规范文档，... 化和补全代码...”



## 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



敦煌站

K+ 思考周®研习社

时间: 2025.08.29-30



上海站

K+ 金融专场

时间: 2025.09.26-27



香港站

K+ 思考周®研习社

时间: 2025.11.17-18



K+峰会详情



上海站

Ai+研发数字峰会

时间: 2025.05.23-24



北京站

Ai+研发数字峰会

时间: 2025.08.08-09



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

