

51CTO WOT

World Of Tech 2024

# WOT全球技术 创新大会

智启新纪  
慧创万物



# AI 原生软件研发新范式赋能 研发效能提升

彭云鹏

百度智能研发团队高级经理



## 彭云鹏

### 百度智能研发团队 高级经理

- 负责百度代码智能化方向和产品、Comate代码助手、云IDE平台、需求管理平台、代码静态分析平台、代码搜索服务和效率云平台等业务和团队
- 在代码智能化、静态分析、开发工具、软件测试、研发数字化等领域申请国内外发明专利56个，已获授权专利33个
- 国家重点研发计划『基于编程现场大数据的软件智能开发方法和环境』技术骨干
- OpenHarmony技术指导委员会IDE TSG成员

# 目录

---

- 1 AI 原生软件研发范式**
- 2 研发智能化探索与实践**
- 3 研发智能化赋能研发效能提升**

# AI 时代已来，软件工程面临变革

# AI 原生软件研发变革

## AI原生产品

AI四项能力、LGUI、数据飞轮

## AI原生应用架构

规则 -> 模型、MaaS、MoE

## AI原生研发过程

流程重构、研发工具、研发范式

## AI原生应用带来新的软件形态

## 大模型带来生产力的大幅提升

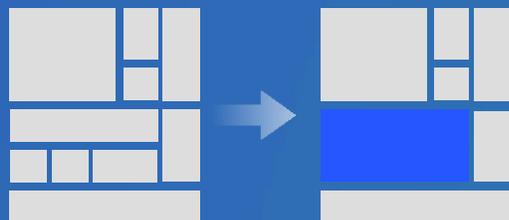
## 软件工程上的变化

软件工程 For AI 原生：新流程、新工具

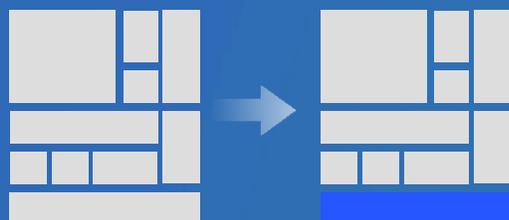
AI 原生 For 软件工程：软件研发智能化

# AI 原生应用架构

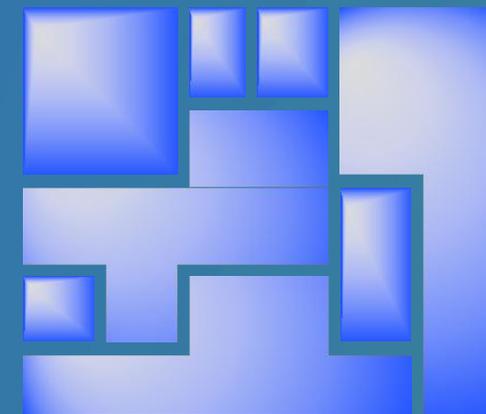
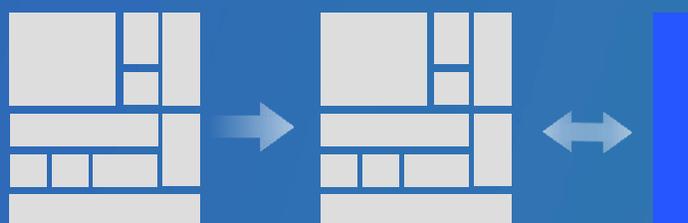
将已有模块替换  
为AI模块



添加一个新的AI  
模块

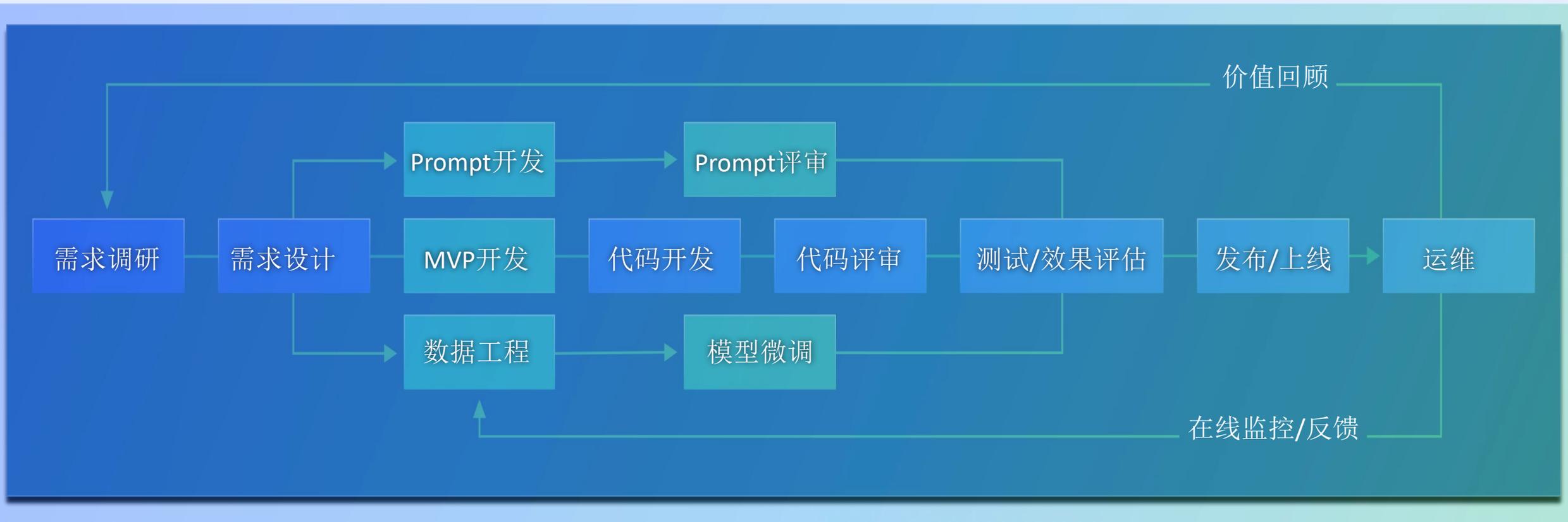


用AI模块驱动  
传统模块



模块中无处  
不在的AI

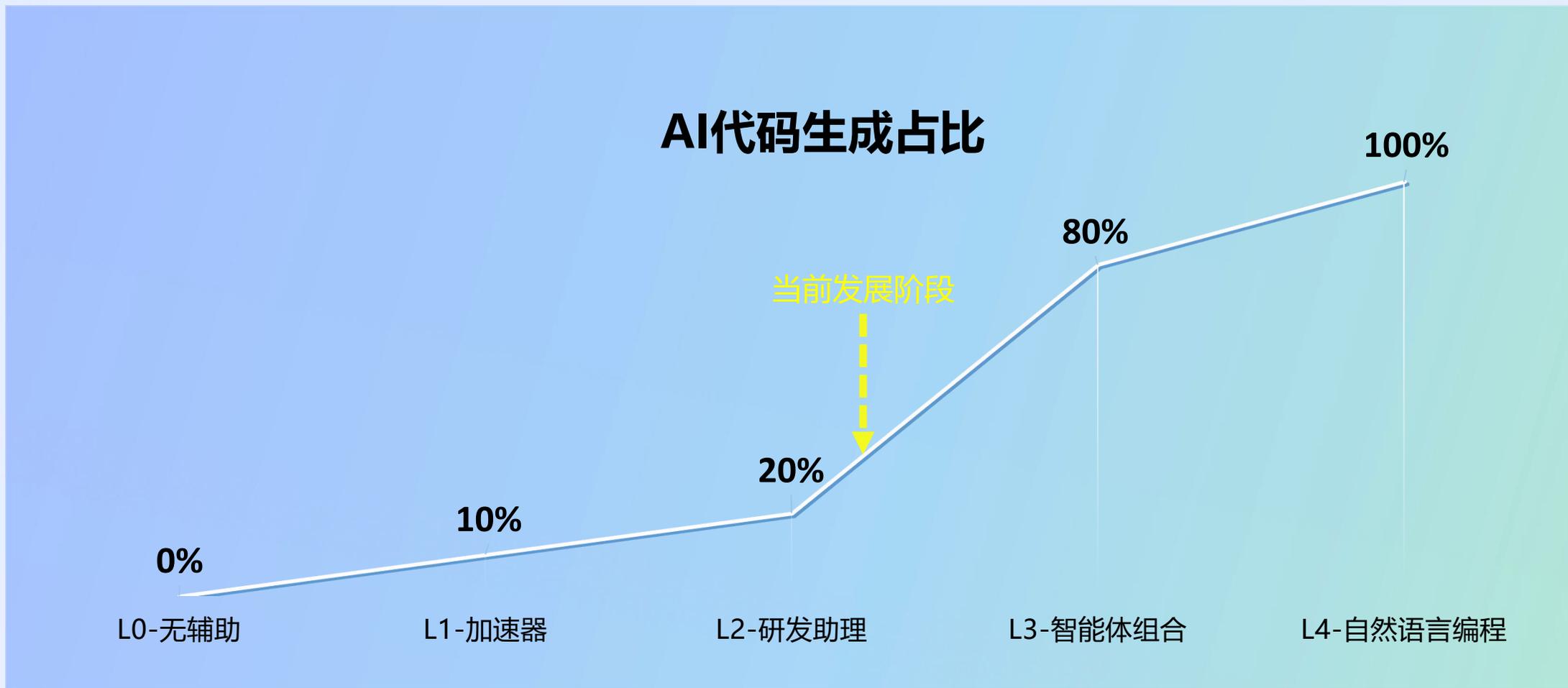
# AI 原生研发过程



# AI 原生研发工具



# 软件研发智能化发展趋势



# 阶段一：加速器

主要以实时续写功能为主，根据代码上下文及依赖文件，生成工程师接下来会写的代码，加速编码过程，此阶段，AI代码生成占比可达到10%以上

```
for owner in response['data']['owners']:
    if 'userName' in owner.keys():
        owners.append(owner['userName'])

GLOBAL_CACHE[repo]['admins'] = ','.join(admins)
GLOBAL_CACHE[repo]['owners'] = ','.join(owners)
return GLOBAL_CACHE[repo]['admins'], GLOBAL_CACHE[repo]['owners']

if __name__ == '__main__':
    main(sys.argv[1:])
```

# 阶段二：研发助理

在工程师研发过程中的各个阶段，对每个研发行为进行单点的赋能，除了加速编码外，对技术设计、阅读调研、调试、评审、测试等各个环节都提供辅助能力，提升速度的同时更提升了研发的质量。此阶段，AI代码生成占比可达到20%以上



# 阶段三：智能体组合

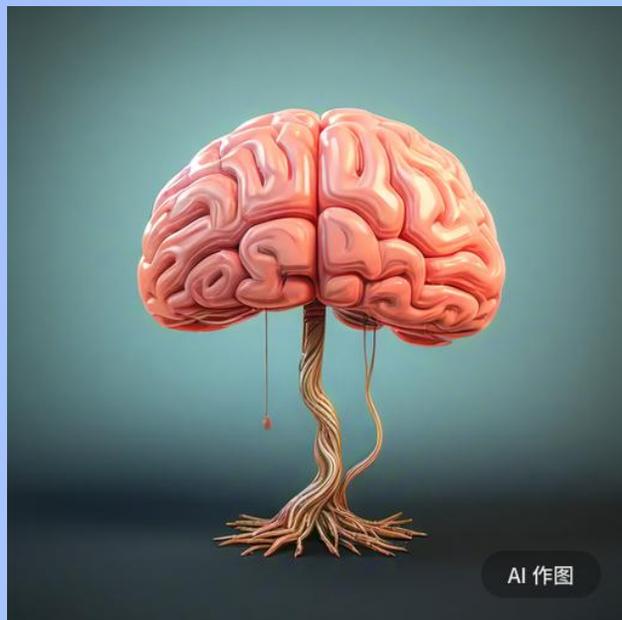
通过私域知识增强、工程架构的理解，从围绕代码的生成提效向左和向右延展，从需求直接生成代码，代码生成测试，打造多个Agent，来实现不同工作，并串联需求-代码-测试-上线，实现从需求到产品的端到端生成，此阶段，AI代码生成占比可达到80%左右，人更多精力投入到需求分析和结果验证上



# 阶段四：自然语言编程

代码作为连接人的想法和机器的桥梁，如果AI能够完全准确的理解人的想法，则将会进入无代码时代，工程师将不再需要编写代码，而是准确描述自己的创意，通过澄清，指导AI构建产品

## 创意



代码



## 产品



# 目录

---

- 1 AI 原生软件研发范式**
- 2 研发智能化探索与实践**
- 3 研发智能化赋能研发效能提升**

# 国家重点研发计划

## 基于编程现场大数据的软件智能开发云平台

2017-2020

# 百度研发智能化探索历程

**2022年9月**

百度内部发布Comate  
代码推荐工具  
支持代码实时续写  
支持内部主流语言和  
IDE

**2023年Q2**

内部全面推广落地，  
80%以上工程师使用  
面向社会发布，支持企  
业试用

**2023年Q4**

发布SAAS版，支持用  
户即申即用  
当前注册企业1万+

**2023年Q1**

Comate升级代码智能  
助手  
上线侧边栏，支持自然  
语言生成代码

**2023年Q3**

模型全面升级  
产品功能丰富度行业  
领先  
准确率对标Copilot

**2024年**

支持RAG，上线私域&公  
域知识增强能力  
上线Comate+开放平台  
发布个人用户免费版

# 代码知识增强升级

全面权威的知识来源，知识更全，生成更准，迈向代码智能新时代。

10000+Github 热门代码库  
全面检索和问答

支持实时网络检索  
获取最新技术知识

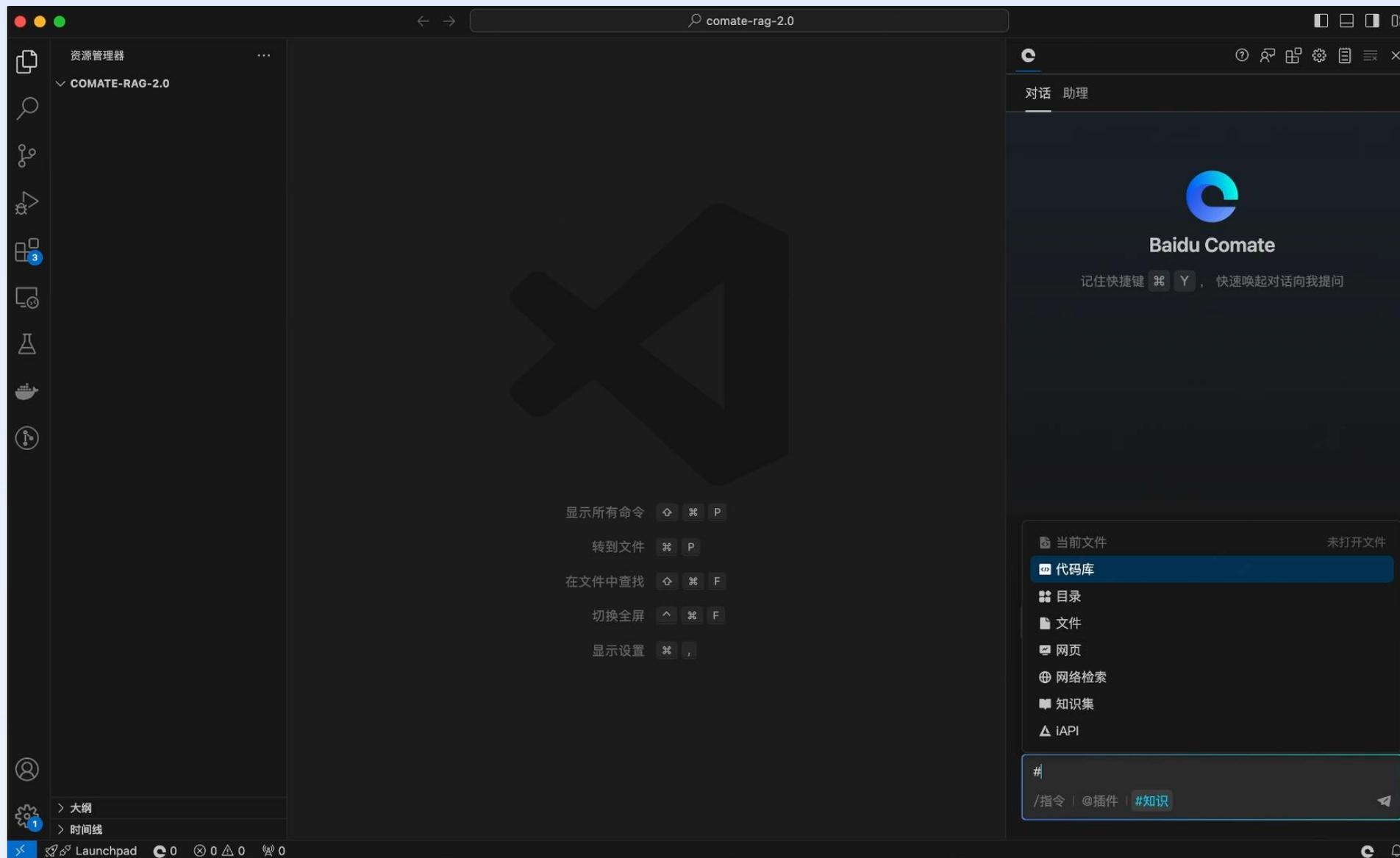
给定API URL  
即可生调用和测试代码

The image shows a code editor window with a Python file named `ernie.py`. The code defines three classes: `Usage`, `FunctionCall`, and `ErnieResponse`. The `Usage` class has an `__init__` method that initializes `prompt_tokens`, `completion_tokens`, and `total_tokens` to 0. The `FunctionCall` class has an `__init__` method that initializes `name`, `thoughts`, and `arguments`. The `ErnieResponse` class has an `__init__` method that takes `response_data` and initializes various attributes like `id`, `object`, `created`, `sentence_id`, `is_end`, `result`, `is_truncated`, and `need_clear_history`.

On the right side of the editor, there is a Comate AI assistant interface. It features the Comate logo and the text "欢迎使用Comate智能编码助手" (Welcome to use Comate AI coding assistant) and "通过自然语言对话，辅助编码过程" (Through natural language conversation, assist the coding process). Below this, there are buttons for "Y" and "⌘" (command key), and a prompt "记住快捷键，快速唤起Comate" (Remember the shortcuts, quickly summon Comate). A prominent button says "输入@，调起Comate 插件能力 去试试" (Enter @, summon Comate plugin capabilities, go try). Below that, it says "可以试着问我" (You can try asking me) and lists two features: "UT 为选中的代码生成单测" (UT generate unit tests for selected code) and "📄 为选中的代码生成注释" (📄 generate comments for selected code). At the bottom, it says "了解更多智能编码能力，请查看Comate官网" (Learn more about intelligent coding capabilities, please visit the Comate website). The interface also includes a search bar with the text "使用 / 调起快捷指令、@ 调起插件或 # 引用知识" (Use / to summon shortcuts, @ to summon plugins or # to cite knowledge).

The image shows a code editor window with a Python file named `ernie.py`. The code defines a `ErnieResponse` class and an `Ernie(LLM)` class. The `ErnieResponse` class has attributes `name`, `thoughts`, and `arguments`. The `Ernie(LLM)` class has a `url` attribute and a `def _llm_type(self) -> str:` method that returns "Ernie". It also has a `def _call(self, prompt: str, stop: Optional[List[str]] = None, history: List = None, **kwargs) -> str:` method that handles the LLM call logic, including message history management and API request construction.

On the right side of the editor, there is a Comate AI assistant interface. It displays the Comate logo and the text "欢迎使用Comate智能编码助手" (Welcome to use Comate AI coding assistant). Below this, it says "通过自然语言对话, 辅助编码过程" (Through natural language conversation, assist the coding process). There are buttons for "对话" (Conversation) and "助理模式" (Assistant mode). A prompt "输入@, 调起 Comate 插件能力 去试试" (Enter @, activate Comate plugin capabilities to try) is shown. Below that, it says "可以试着问我" (You can try asking me) and lists two actions: "UT 为选中的代码生成单元测试" (UT Generate unit tests for selected code) and "为选中的代码生成注释" (Generate comments for selected code). At the bottom, it says "了解更多智能编码能力, 请查看Comate官网" (Learn more about intelligent coding capabilities, please visit the Comate official website).





# 基于多Agent的自评估机制



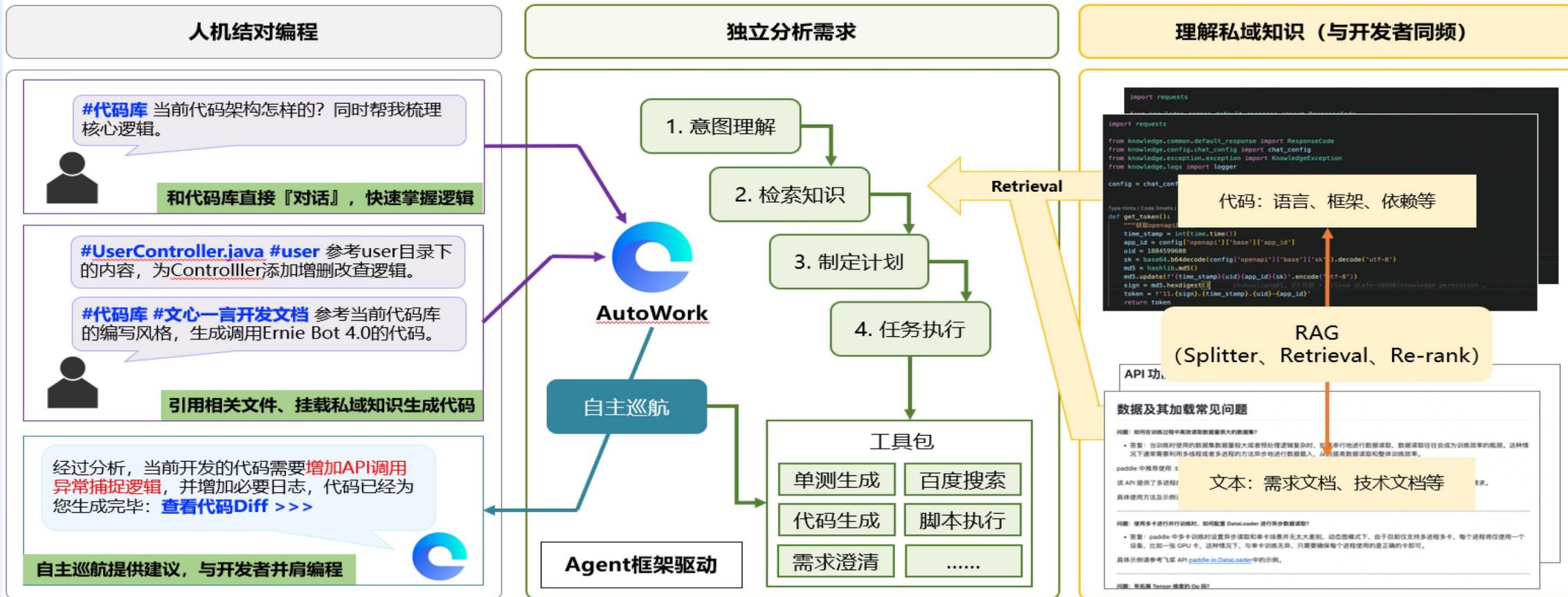
The screenshot displays the Comate IDE interface with several key components and annotations:

- Code Editor:** Shows Go code for `ListallCore()` and `NactestCore()` functions. Red annotations highlight: "2.API设计器生成接口框架代码" (API Designer generates interface framework code) and "3.数据建模生成Dao层代码" (Data modeling generates Dao layer code).
- API Interface Designer:** A panel on the right for defining API endpoints. It includes fields for PATH, Method (e.g., POST), and description. A table lists parameters with names, types (String, Number), and descriptions.
- Database Modeling:** A panel for defining database tables. It includes a table for columns with names, types, and descriptions.
- Downstream RPC Selection:** A panel for selecting RPC services. It shows a list of services (RPC001, RPC002) with their types and descriptions.
- Cloud Control Panel:** A panel for configuring cloud services, including a table for configurations.
- Annotations:** Red text annotations are placed over the code and panels: "4.注册中心选择下游服务生成RPC代码" (Select downstream services in the registry to generate RPC code), "5.设计配置内容生成配置封装加载代码" (Design configuration content to generate configuration packaging and loading code), and "6.边开发边测试" (Develop and test side-by-side).
- Left Sidebar:** Contains a "为下面的函数生成单元测试:" (Generate unit tests for the following functions) section with a "收起" (Collapse) button. Below it is the "Easy脚手架" (Easy Scaffold) section with a table of container items.

# Autowork: 端到端生成探索

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。

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



# Comate AutoWork

复杂任务拆解，全过程智能辅助

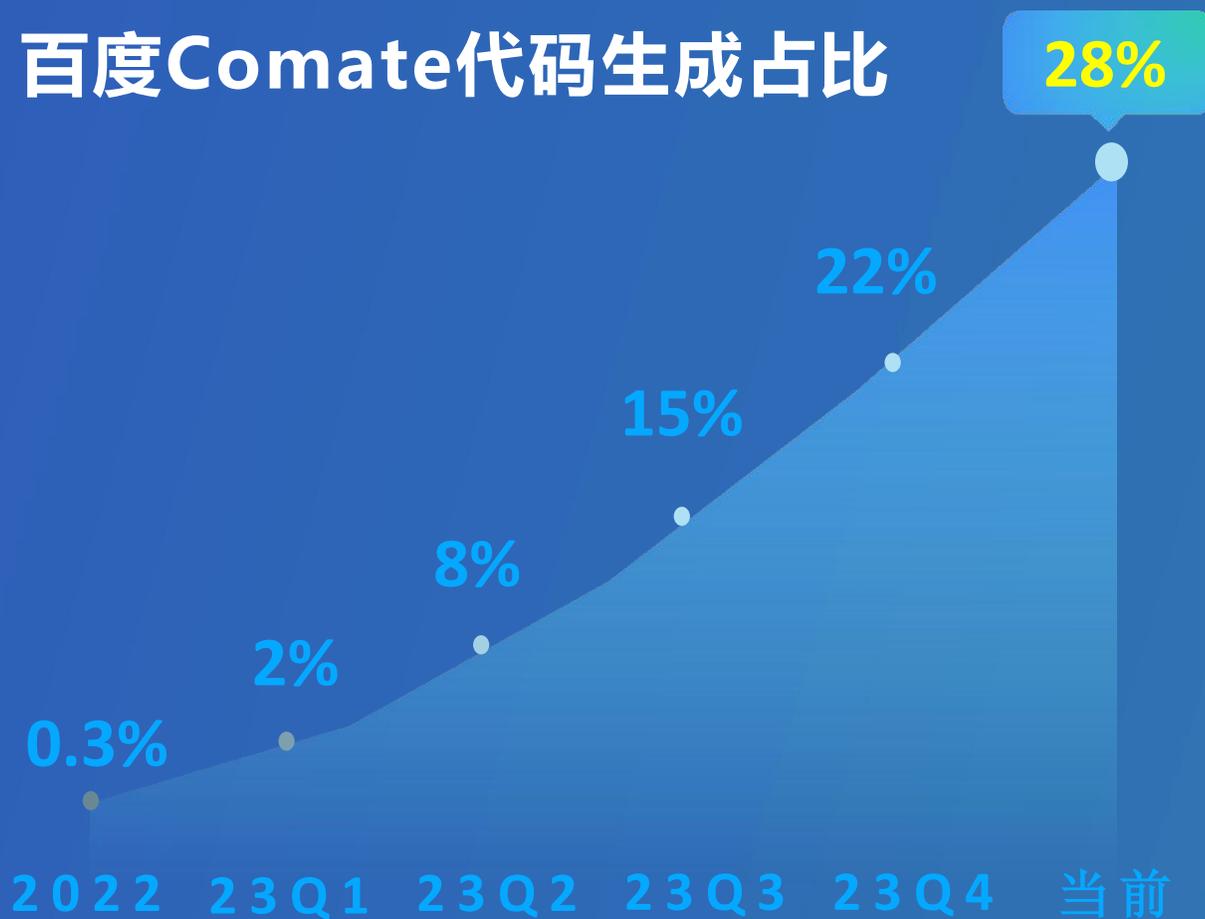
# 目录

---

- 1 AI 原生软件研发范式**
- 2 研发智能化探索与实践**
- 3 研发智能化赋能研发效能提升**

# 百度内部规模落地，赋能公司效能提升

## 百度Comate代码生成占比



百度全局提效 **11%**

工程师使用 **80%+**

用户满意度 **90%+**

编码提效 **34%**

单兵提效 **14%**

# 51CTO WOT 深入论证，AI代码生成与提效的关系

- **收益：当代码生成占比达到28%时，预计为公司研发提效11.2%**
  - 算法一：工程师每天写代码约占一天工作时间的40%，代码生成占比28%
    - **Comate为公司研发提效=40%\*28%=11.2%**
  - 算法二：工程师没采纳时写一行代码耗时16.54s，采纳时耗时1.74s，续写采纳率46%
    - 采纳时耗时主要用来理解推荐代码，不采纳时同样需要先理解代码，故如果不用Comate，则单行耗时为16.54s-1.74s=14.8s
    - 采纳时提效=  $(1-1.74/14.8)*46%=40.59%$
    - 不采纳时降效=  $1.74/14.8*(1-46%)=6.35%$
    - 编码时长占一天工作时间40%
    - **Comate为单工程师研发提效=  $(40.59%-6.35%)*40%=13.69%$**
    - Comate工程师渗透率为80%
    - **Comate为公司研发提效=  $13.69%*80%=11.23%$**
  - 两种算法最终结果比较相近，故Comate当前为公司提效比例约为11%
  - **百度内部需求吞吐提升12%**

## 系统化的安全保障

推理前后双重脱敏，全链路安全合规

## 优秀的效果体验

芯片、框架、大模型、应用，端到端全栈优化

## 极致的推荐速度

生成速度每快100ms将影响5%的实际采纳率

## 场景化开放适配

助力企业打造更适配自己研发环境的代码助手

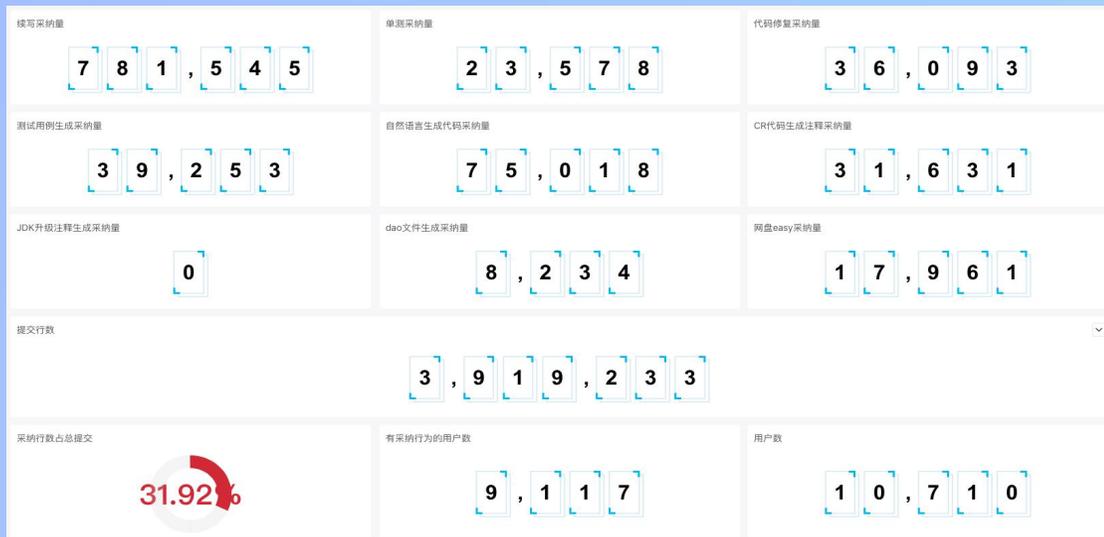
## 数据驱动创新

百度内部万人研发实践，快速反馈迭代

## 低成本灵活部署

公有云、混合云、私有化、一体机

# 数据驱动价值落地和效果改进

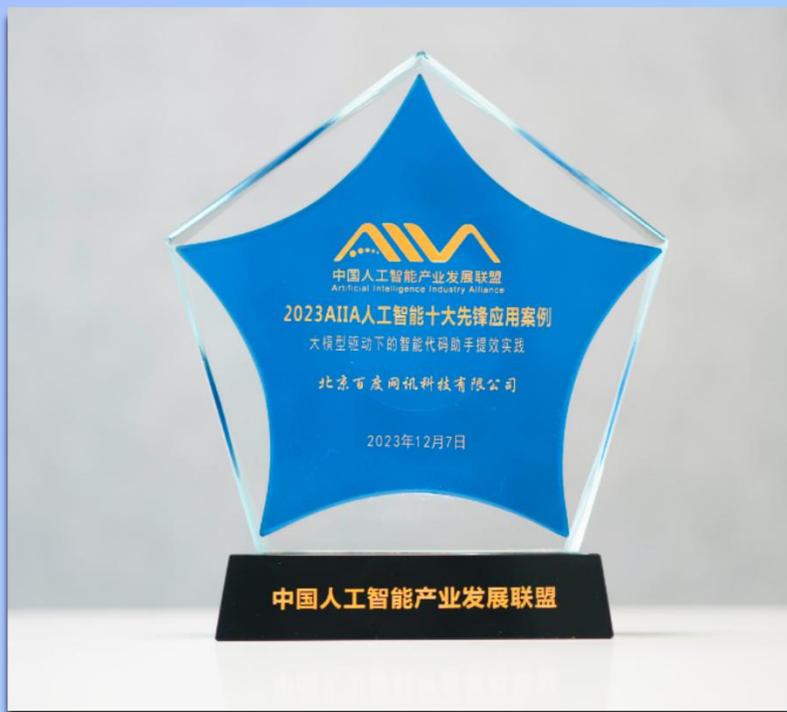


代码生成 (下级组织)

组织	代码生成占比	采纳率	推荐量	采纳量	提交行	渗透率	用户数	潜在用户数
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

# 2023AIIA 人工智能十大先锋应用案例

## 软件工程领域唯一获奖者



1万+

SAAS注册企业数

## 成单客户覆盖各行各业

银行	汽车	通讯	集成电路	软件服务
机械制造	音视频	消费	能源	IoT
云计算	信息技术	大数据	生物科技	旅游

**AI不会替代你  
但善用AI  
会让你变成  
更好的自己!**



视频来源：文心一言根据寄语生成

感谢倾听

THANKS