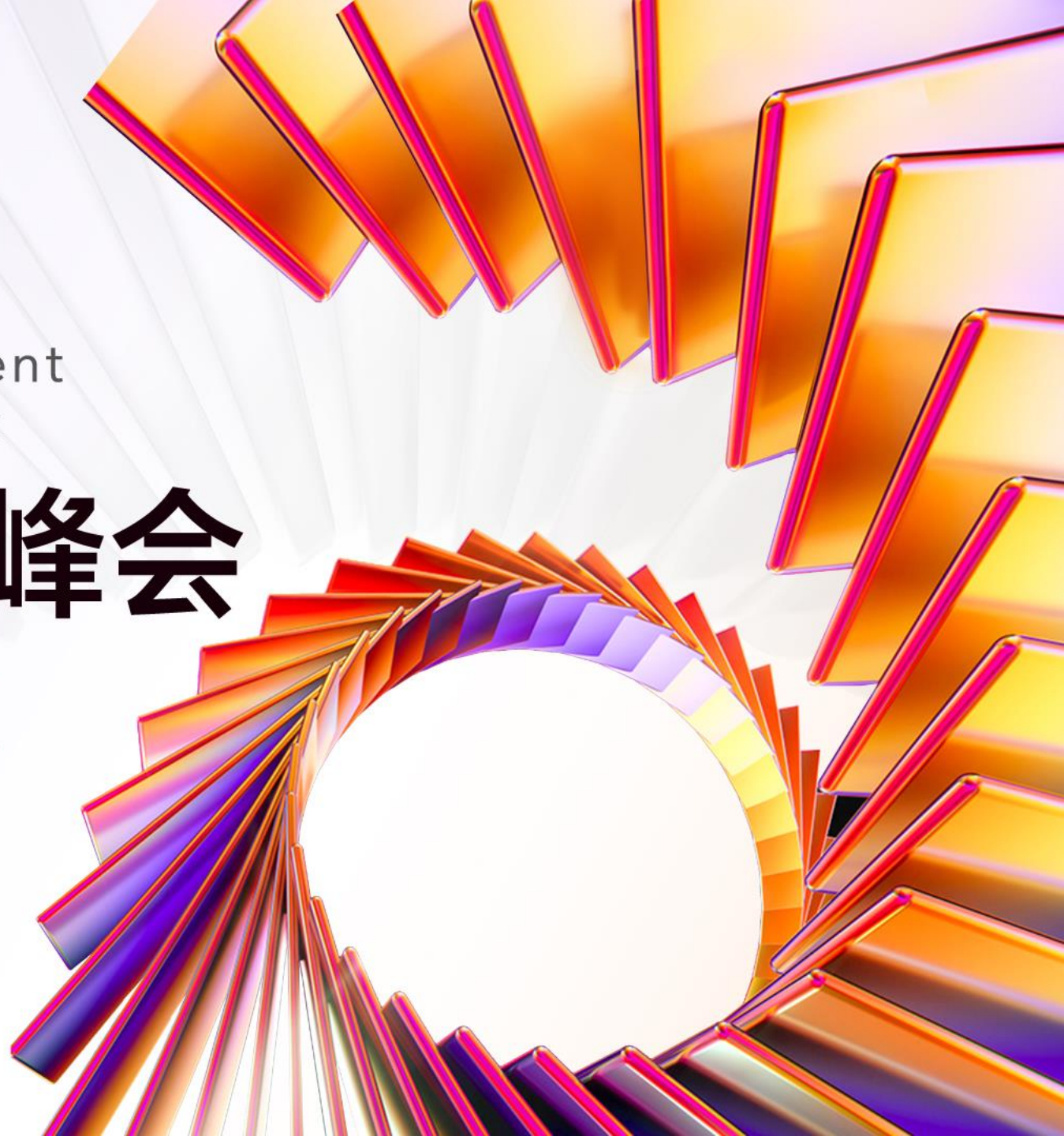
AiDD 8th 2025

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



AiDD × K

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时代的用户洞家与需求发现

论坛2: AI原生产品战路与商业模式重构

论坛3: AgenticAI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

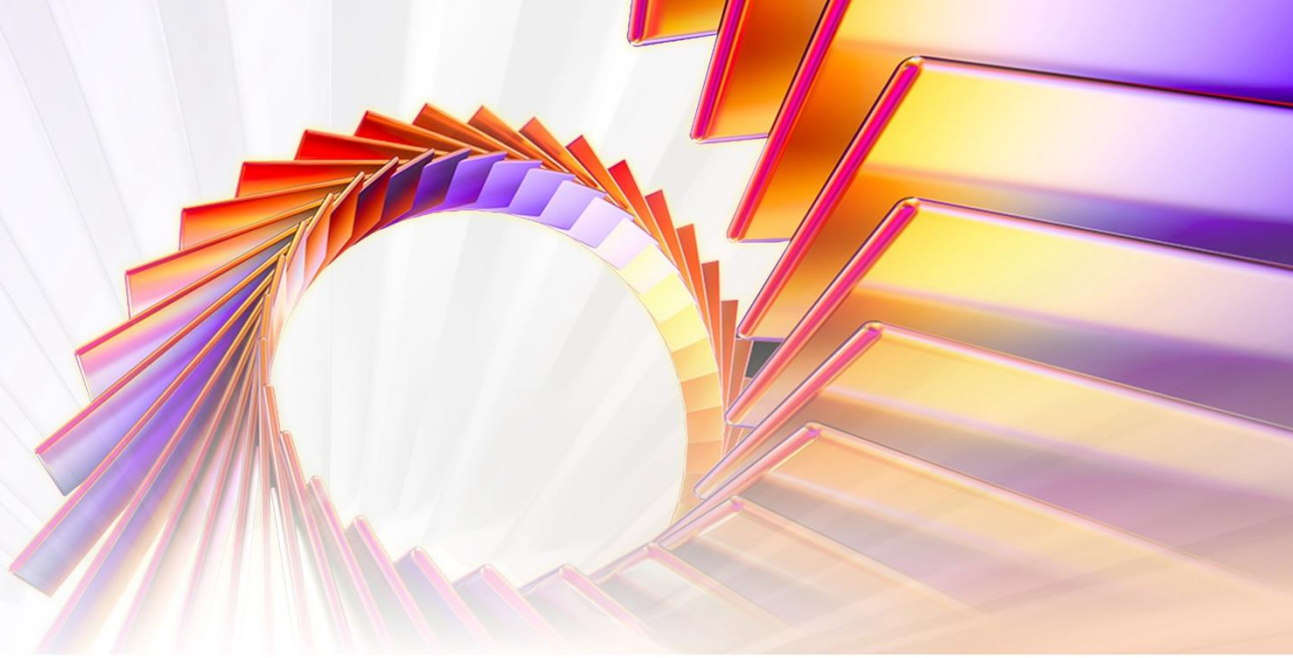
40+ 创新案例 开拓全新AI+产品视角

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 Digital Summit
AI+研发数字峰会
拥抱AI 重塑研发



AI如何重塑开发：京东JoyCode IDE 2.0 的生产级创新

郑少强 | 京东科技

郑少强

京东科技 算法专家

拥有多年互联网算法经验，曾深度参与顶级互联网平台的算法模型优化。目前在京东科技专注于 AI 编程（AI Coding）领域的 探索与落地。长期深耕于 AI 编程核心算法的设计与开发， 涵盖开源代码补全、智能问答以及 Code Agent 等前沿模型，旨在全面 提升研发效能。

目录
CONTENTS

AiDD 8th 2025

PART 01

编程范式的革新-AI如何重塑研发效率

编程范式的革新-AI如何重塑研发效率

传统人产出效率转向“人机协作”流畅度与协同产出提升

速度

传统模式

关注开发进度、代码量和迭代频率，瓶颈通常在人力。

AI驱动的新模式

强调"人机协同"的流程优化，AI大幅压缩需求到交付的流转时间，开发者重点转为引导AI"生成高质量代码"。

质量

传统模式

主要靠人工评审和事后测试，依赖缺陷密度、缺陷逃逸率。

AI驱动的新模式

“预防+过程自愈"，AI辅助规避漏洞、自动生成测试与修复，聚焦缺陷预防率、自动修复率。

成本

传统模式

以优化人力投入为主，关注人均产出和资源消耗。

AI驱动的新模式

从"节约人力"到"智力资源重分配"，AI承担重复和复杂劳动，人力专注创新与价值创造。

编程范式的革新-AI如何重塑研发效率

京东研发模式深入AI，从业务需求到部署上线，力求从端到端解决业务需求，同时细粒度监控每个节点质量

效率飞跃

AI辅助吞吐量大幅提升，需求澄清周期显著缩短，开发效率提升3-5倍

质量保障

“预防+过程自愈"模式，AI辅助规避漏洞，自动生成测试与修复

全流程自适应

从需求到运维，AI深度嵌入每个环节，实现全流程的智能化与自适应优化

AI不仅提升单点效率，更将研发从"人写AI帮"转变为"人机共创"模式

编程范式的革新-AI如何重塑研发效率

需求分析 → 需求生成 → 任务拆解 → 编码开发 → 测试联调 → 部署发布 → 运维监控

产品矩阵

- JoyCode IDE
- JoyCode IDEA 插件
- JoyCode VsCode 插件
- JoyAgent 智能体生态

Agent能力

Coding		行云	CSR		生产级智能体
代码补全	TabTab	PRD生成	关键词	JoyCodeWiki	环境/问题调试
添加注释	代码解释	TRD生成	正则	项目文档	AI 应用生成
代码优化	代码评审	任务拆解	Embedding	PRD文档	智能体工具生成
缺陷修复	报错分析	研发计划	CodeGraph	TRD文档	智能Debug

核心能力

意图识别 | Spec | CSR | JoyCodeWiki | JoyCli | 智能体 | 数据飞轮

底层依赖

京东数据：代码/项目/需求/行云/持续交付

京东云平台：训练/推理/部署/评测等

PART 02

JoyCode IDE 1.0 人机协同AI编码，提升京东研发开发效率

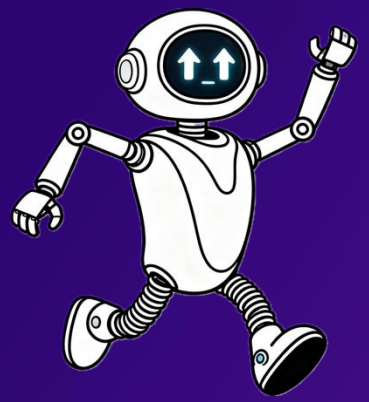
JoyCode：从辅助到自动编程进化

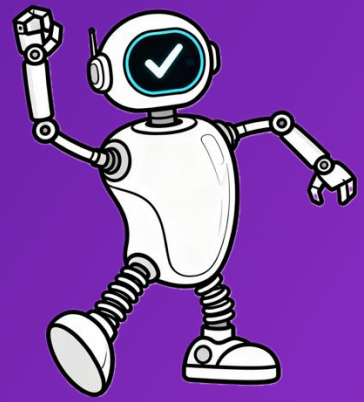
IDE客户端	JoyCode IDE						
JoyCode	基础功能		高级功能		服务端		
	代码补全	代码优化	CSR引擎	PRD助写	工具	身份鉴权	
	tabtab	生成单测	智能体团队	JoyCodeWiki	上下文管理	模型策略	
	生成注释	报错分析	Spec编程	一键上云	MCP	代码知识库	
	解释代码	Text2Code	快速部署上云	一站式开发	Rules	资源管理	
模型	生成模型	对话模型	Agent模型	Embedding	补全模型		
数据	数据采集	数据清洗	数据标注	数据解析	效果评测		
底层算力	JoyBuilder 训练池	JOYAI	JoyBuilder 推理池				

JoyCode：从辅助到自动编程进化

代码补全模型的落地挑战与技术突破

挑战	解决方案
代码生成效果	SFT+GRPO
语法错误及自适应性	Block FIM数据构造
缺乏跨文件全局感知	JoyRepo仓库补全
推理速度慢	Speculative Decoding





JoyCode：从辅助到自动编程进化

问题：通用补全模型难以满足各业务线精细化开发需求

京东多元业务生态

零售中台
商品、订单、用户行为

零售广告
营销、推广、创意

物流
仓储、运输、路径优化

科技
算法、金融、京东云

工业
生产、B端、制造

健康
医疗、药品、病例

通用模型的局限性

知识领域缺失
缺乏特定业务场景领域知识，难以理解业务上下文和专业术语

补全准确率低
跨场景应用导致推荐不准确，难以适应特定场景的技术栈和编程范式

推荐不相关
可能引入与当前场景无关的代码片段，导致错误补全建议

研发效率受限
无法提供针对性的代码补全，开发人员需要更多时间生成代码

JoyCode：从辅助到自动编程进化

核心策略： 分布式构建数据、训练模型

场景专属数据集

针对每个业务场景构建高质量数据集

模型独立训练

利用专属数据SFT、RL

深度理解场景

模型学习特定场景的上下文信息和编码习惯

特定场景模型

针对特定场景提供专有代码补全模型

精准数据划分

针对每个核心业务场景，构建数据集充分考虑特有的业务逻辑、编程范式和技术栈，确保数据能够准确反映场景特性

业务逻辑隔离

避免跨场景数据污染，确保每个场景模型只学习该场景特有的业务逻辑和代码模式，避免引入不相关的代码特征

数据划分隔离优势

- 确保每个场景模型专注于该场景已有的业务逻辑和代码模式
- 有效避免扩场景知识干扰，提高代码补全准确性
- 使模型能够适应个场景独特的编码范式和技术栈
- 为专有场景的代码补全模型提供坚实的数据基础

PART 03

JoyCode IDE 2.0：多工具集成，深度理解企业级代码

深度理解企业级代码-JoyCodeWiki

研发工作中面临的痛点问题：

面对京东众多业务，研发面对上百万行代码，且涉及多人协作，会面临三大挑战：

理解复杂代码困境

- 代码规模巨大、变更频繁、历史包袱重
- 不同工程师风格各异，增加代码整合难度
- 技术债务：过时算法和低效架构增加维护成本

知识沉淀不足形成知识孤岛

- 文档缺失或过时，难以集中管理
- 核心知识依赖"口口相传"，传承效率低
- 新员工上手周期长，影响团队生产力

协作壁垒

- 缺乏统一代码知识视图，信息不对称严重
- 代码变更意图和影响难以理解
- 代码审查负担重，沟通成本高

深度理解企业级代码-JoyCodeWiki

JoyCodeWiki是一个集自动化、智能化、协同化于一体的代码知识库平台。它不仅仅是代码的存储库，更是代码知识的智能管理和共享中心

静态Wiki

承载相对稳定、长期有效的知识内容，为开发者提供宏观的、结构化的Wiki视图。

- 项目背景与设计文档
- 核心模块说明
- 提供稳定的知识基础

实时融合

动态Wiki

自动捕捉代码的"血液"流动，实现知识的实时更新与演进。

- 代码变更日志
- 新功能介绍
- API更新

双层架构优势

通过静态与动态Wiki的结合，JoyCodeWiki确保知识的全面性与时效性，为研发人员提供一个完整、动态更新的代码知识库，有效解决复杂代码理解、知识沉淀和协作壁垒等问题。

JoyCodeWiki： 智能问答功能

JoyCodeWiki通过智能问答功能，实现代码知识的高效检索与应用，显著提升开发效率。

"如何实现这个功能的性能优化？"

"根据代码分析，您可以尝试优化算法复杂度和缓存策略..."

自然语言交互

支持开发者使用自然语言提问，系统自动从代码知识库中检索并提供答案，降低技术文档阅读门槛。

一站式知识闭环

同时给出原理、代码、文档与最佳实践，一键转跳原文，边用边学无需切换页面

即时解答疑问

快速响应开发者的疑问，减少等待时间，提高问题解决效率，加速开发进程。

智能问答带来的价值

通过智能问答功能，JoyCodeWiki实现了知识的高效流通与应用，使研发团队能够更快地解决技术难题， 降低沟通成本， 提升整体研发效率。

JoyCodeWiki：AI 智能评估工作流

1 **输入配置**

用户设定仓库地址、分支、项目类型、输出语言等参数。

2 **AI目录生成**

生成模型进行多次工具调用，分析项目结构并构建目录。

3 **智能质量评估**

评分模型对生成的目录结构、内容详细度进行评估。

4 **输出评估结果**

展示最终的目录结构、评分、总结和优化建议。

配置示例 (Chrome DevTools MCP)

仓库地址:	github.com/ChromeDevTools/chrome-devtools-mcp	生成模型:	Kimi-K2 (推荐)
评分模型:	闭源模型	总用时:	54.12 秒
总Token:	162,011		

JoyCodeWiki：生成与评估结果示例

</> 目录生成结果示例 (JSON)

```
{
  "items": [
    {
      "title": "getting-started",
      "name": "入门指南",
      "prompt": "为Chrome DevTools MCP服务器创建全面的入门指南，涵盖项目概述、技术栈、安装配置和基本使用流程",
      "children": [
        {
          "title": "project-overview",
          "name": "项目概述",
          “prompt”： “分析README和package.json，详细说明Chrome DevTools MCP的核心功能。"
        },
        {
          "title": "technology-stack",
          "name": "技术栈分析",
          "prompt": "基于package.json和源代码分析技术栈：Node.js 22.12.0+运行时、TypeScript语言、@modelcontextprotocol/sdk框架、puppeteer-core浏览器控制、Chrome DevTools前端集成。说明各技术的选择理由和版本要求"
        },
        ...
      ]
    }
```

智能评估

AI 评估总结

主要问题为入门缺少FAQ、Prompt输出限制不统一及工具分类重复，适合已有MCP经验的开发者，但对新手引导与生成一致性仍需优化

优化建议

统一在所有Prompt中明确输出限制与读者定位：1) 在顶层新增“写作规范”说明，规定统一语言（中文）.....

持续优化

JoyCodeWiki Agent Runner 通过实现 Wiki 目录的"生成-评估-优化"闭环，将原本需要数小时手动完成的工作压缩，持续提升文档质量。

JoyCodeWiki：产品页面展示

智能阅读

tiankai11/goshell

分享

快速开始指南

- 项目概述
- 环境设置
- 核心概念
- 基本使用
- 快速参考

深入分析

- 架构分析
- 核心组件
  - 窗口管理
  - 终端抽象
  - 配置系统
- 环境实现
  - 本地终端
  - Docker集成
  - Kubernetes集成
- 数据模型
  - 配置模型
  - 会话模型
  - 连接模型

项目概述

1. 工程背景与技术挑战

1.1 开发者面临的终端管理困境

在现代软件开发和运维工作中，开发者经常需要同时管理多种不同的终端

器、Docker容器、Kubernetes集群等。这种多环境管理带来了几个核心

环境切换的复杂性：开发者需要在不同的终端工具之间频繁切换，每个

式和操作界面。这种切换不仅增加了认知负担，还容易导致操作错误和效

工具碎片化问题：传统的终端管理方案通常需要使用多个独立的工具—

OpenSSH）、Docker CLI、kubectl命令行工具等。这些工具各自为政，

了学习和使用成本[2]。

跨平台兼容性挑战：不同的操作系统（Windows、Linux、macOS）对

Windows环境下，原生的终端体验与Linux/macOS有较大差距，这给跨平

的困难[3]。

配置管理复杂性：每种连接类型都需要维护不同的配置信息——SSH的

的守护进程地址；Kubernetes的API服务器地址和认证令牌等。这些配置

统一管理和维护[4]。

1.2 现有解决方案的局限性

传统的终端管理工具存在明显的局限性：

项目概述

1. 工程背景与技术挑战
   - 1.1 开发者面临的终端管理困境
   - 1.2 现有解决方案的局限性
   - 1.3 GoShell的工程价值定位
2. 系统架构设计
   - 2.1 整体架构模式
   - 2.2 核心组件设计
     - 2.2.1 窗口管理器 (Window)
     - 2.2.2 配置管理系统
     - 2.2.3 终端管理系统
   - 2.3 连接集成架构
3. 核心功能实现
   - 3.1 多环境终端管理
     - 3.1.1 统一终端界面
     - 3.1.2 终端会话生命周期管理
   - 3.2 配置管理系统
     - 3.2.1 配置持久化机制
     - 3.2.2 动态配置表单
   - 3.3 快捷命令系统
     - 3.3.1 命令定义和管理
     - 3.3.2 命令执行机制

来源 项目概述

- cmd.go
- config.go
- docker.go
- go.mod
- k8s.go
- local.go
- main.go
- model.go
- README.md
- ssh.go
- term.go
- window.go

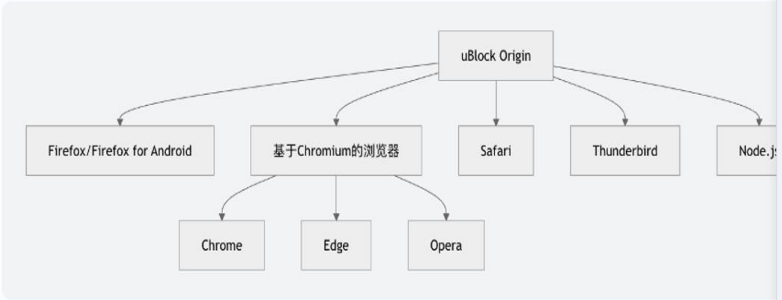
智能问答

这些多样化的过滤能力赋予用户对网络内容的全面控制。

来源：js/

浏览器支持

uBlock Origin支持多种浏览器和平台：



uBlock Origin在**Firefox上表现最佳**，因其拥有更强大的扩展API。由于Manifest V3的限制，Chr

的支持将在Chrome 139之后受限，但团队正在努力适应这些变化。

来源：README.md#L19-L25, platform/

开发理念

uBlock Origin项目遵循以下原则：

1. **用户赋权**：赋予用户对浏览体验的控制

Ask JoyAI

这个项目的核心功能是什么

uBlock Origin是一个高效的广告拦截器和内容过滤扩展，其核心功能是阻止不需要的网络请求和隐藏页面元素。

这个项目的核心功能包括：

1. **网络请求过滤** - 拦截广告、跟踪器、恶意软件等网络请求，在它们到达浏览器之前就将其阻止
2. **外观过滤** - 隐藏或修改网页上的特定元素，比如广告横幅、弹窗等
3. **过滤器列表管理** - 支持多种过滤器列表，用户可以订阅和管理不同的规则集
4. **跨平台支持** - 适配多种浏览器平台，包括Chrome、Firefox、Safari等
5. **性能优化** - 相比其他同类扩展，占用更少的内存和CPU资源

从代码结构来看，src/js/目录包含了所有核心逻辑，platform/目录处理不同浏览器的适配，而assets/目录管理过滤器规则。

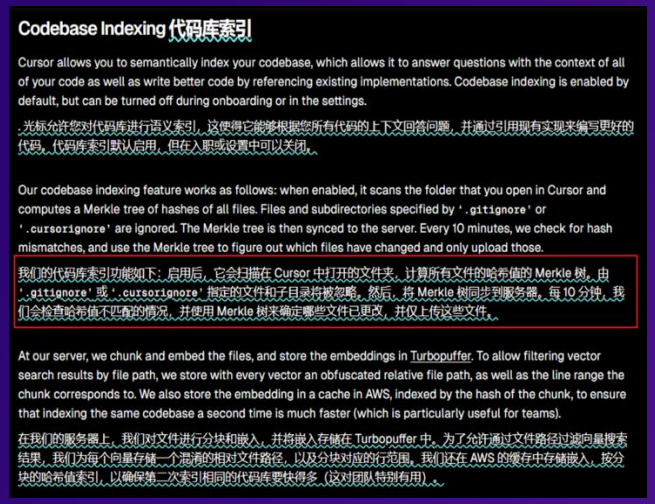
提出后续问题...

回答由AI生成，可能包含错误。

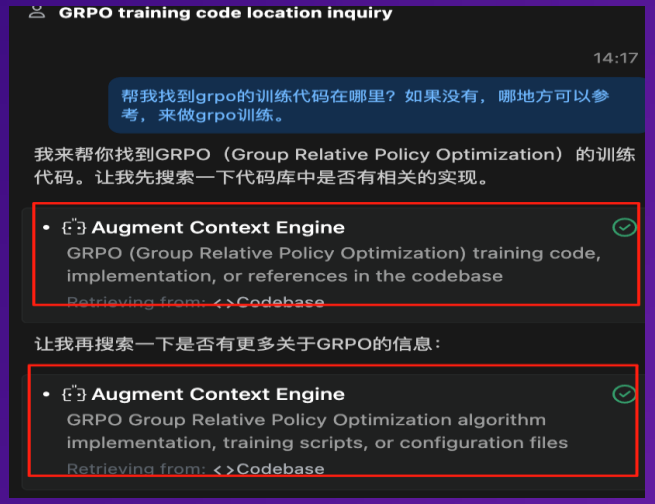
CSR上下文引擎：深度理解企业级仓库代码

业界优秀产品检索工具对比

Cursor



Augment



Claude Code

Codebase navigation and understanding

Teams across the company use Claude Code to help new hires and even long-time employees get up to speed on our codebases.

New data scientists on our Infrastructure team feed Claude Code their entire codebase to get productive quickly. Claude reads the codebase's CLAUDE.md files, identifies relevant ones, explains data pipeline dependencies, and shows which upstream sources feed into dashboards, replacing traditional data catalog tools.

Our Product Engineering team refers to Claude Code as their "first stop" for any programming task. They ask it to identify which files to examine for bug fixes, features, or analysis, eliminating the time-consuming process of manually gathering context before building new features.

	Cursor	Augment	Claude Code
核心技术	Merkle树+Embedding，实现安全高效的代码解析和检索	Context Engine上下文引擎，专有Embedding模型，上下文实时感知	Agentic Search，通过模拟文件系统操作来理解和检索代码
检索方式	云端进行RAG检索	加强版RAG，动态感知上下文状态	依靠模型“智能体”能力
优势	保护用户源代码隐私，快速同步，代码补全能力强	高精度检索，强大的上下文同步机制，能够处理大型业务代码仓库	“任务级”理解，全局视野，始终获取实时信息

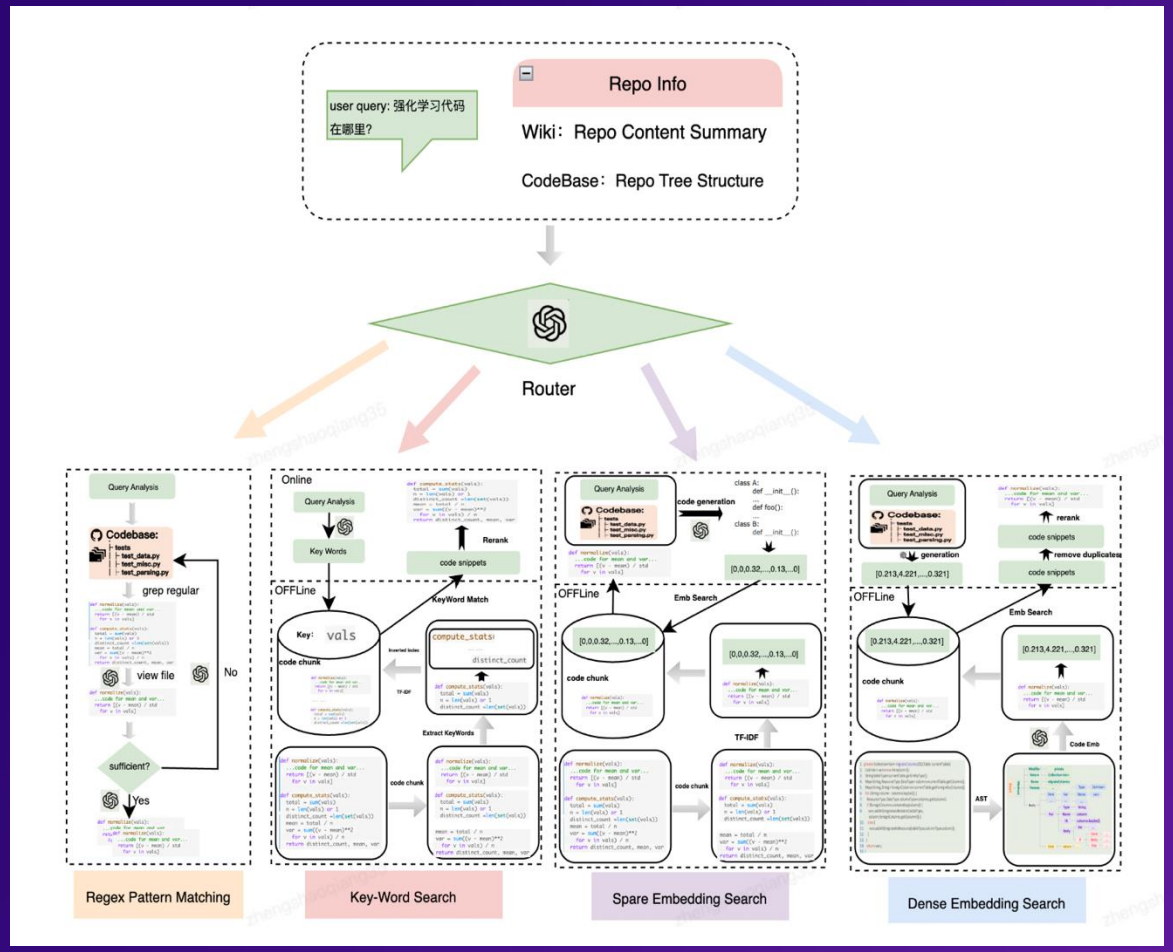
CSR上下文引擎：深度理解企业级仓库代码

京东代码仓库现状：

京东零售	高并发、高可用 超大型代码仓库 模块多、依赖复杂	京东工业	专业术语、软硬协同 新老系统更替 中大型代码仓库
京东广告	实时性、业务复杂 数据和算法代码复杂 大型代码仓库	京东健康	专业术语、数据安全 隐私保护、合规性 中型代码仓库
京东物流	标准化流程 系统集成，专业术语 中大型代码仓库	京东科技	快速迭代、技术前沿 多架构、技术栈 中大型代码仓库

CSR上下文引擎：深度理解企业级仓库代码

动态选择最适合当前场景的检索方式



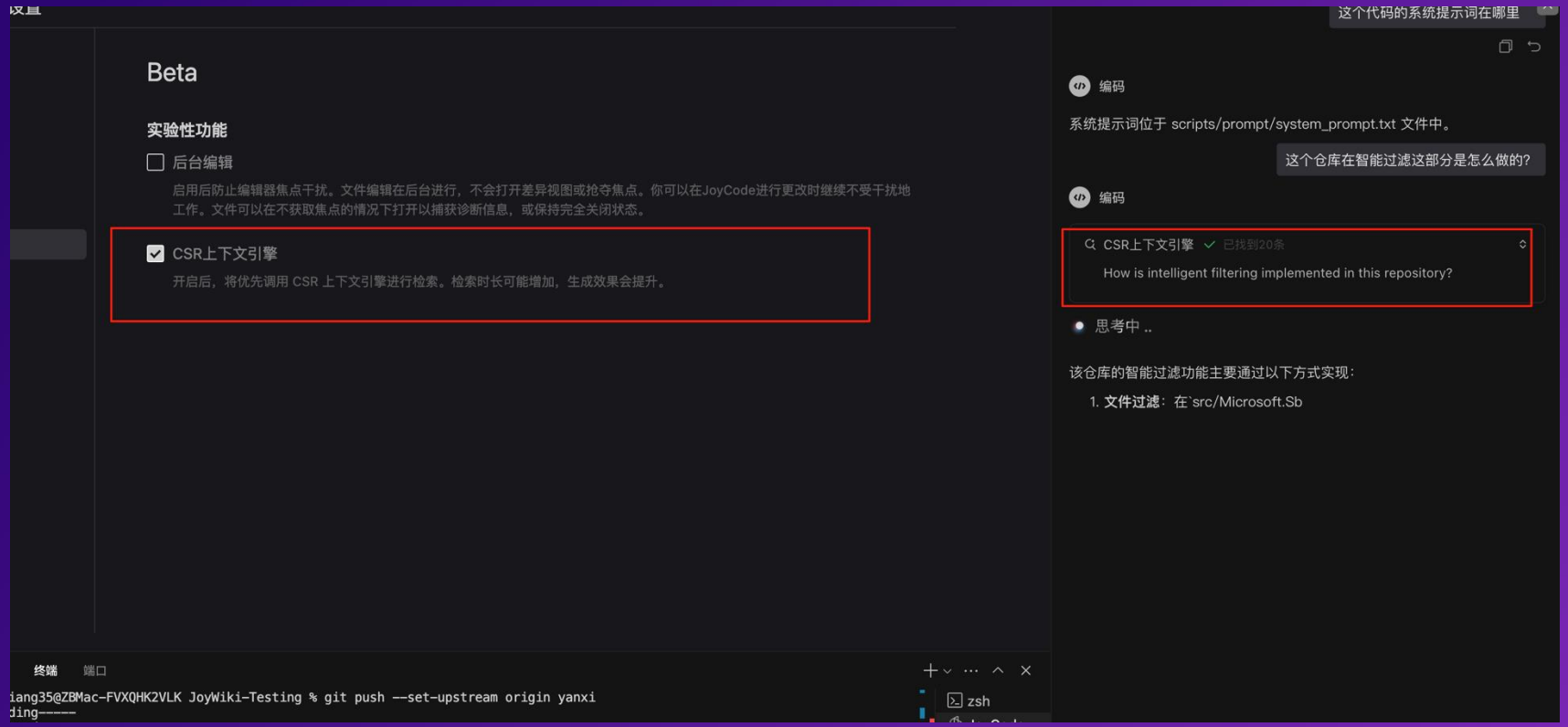
Context Search Router

1. 自适应检索路由模型
2. 性能、精度、成本取得平衡
3. 节省硬件资源，效果达到Emb效果

	Precision@10	Recall@10	nDCG@10
JoyCode-CSR	23.8%	62.8%	61.1%
grep	23.9%	55.6%	56.0%
关键词	20.4%	54.7%	52.3%
Emb检索	23.7%	59.9%	59.0%

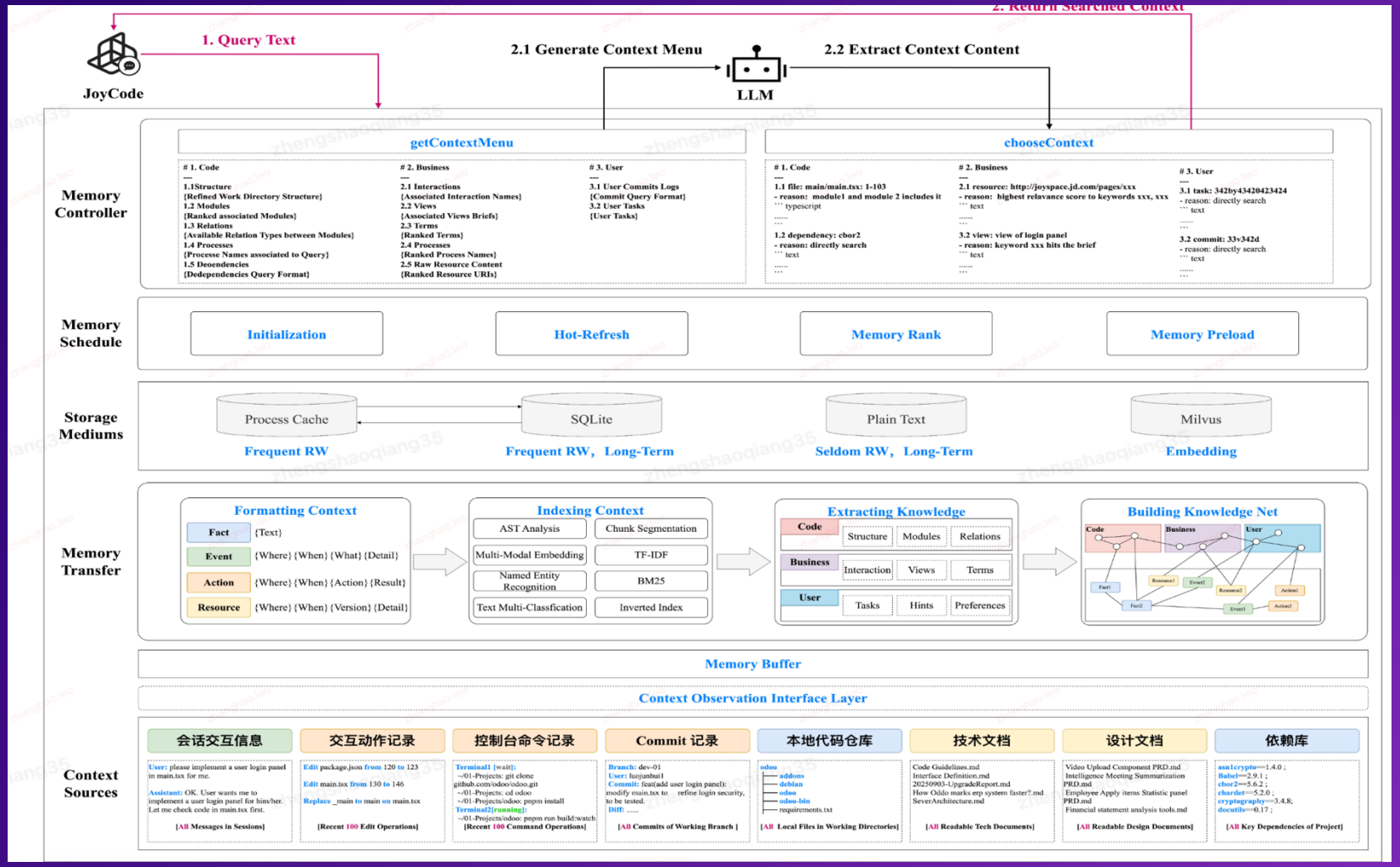
CSR上下文引擎：深度理解企业级仓库代码

CSR在JoyCode上的使用



CSR上下文引擎：深度理解企业级仓库代码

CSR：正在接入更多上下文下信息，支持不同场景的业务检索



PART 04

JoyCode IDE 2.0：生产级Spec编程：如何让AI理解高级指令

行云协作背景：从数字化时代到智能化时代的跨越

传统化时代

流程驱动，手工协同

需求管理：记录非结构化，版本分散不可控，变更评估困难。

研发协作：交付串行，状态不可视，人工协调依赖性高，交付周期长。

质量交付：以人工测试、人工发布为主导，风险与回滚成本高。

数字化时代

行云支撑，数据驱动

需求管理：全流程交付集中在线调度，状态可视，变更可追溯。

研发协作：代码仓库、需求、测试、发布等信息在平台内关联；进度可视，风险可预警。

质量交付：构建、测试、发布、变更全链路可视，质量与变更数据沉淀，问题可追溯。

智能化时代

JoyCode赋能，智能涌现

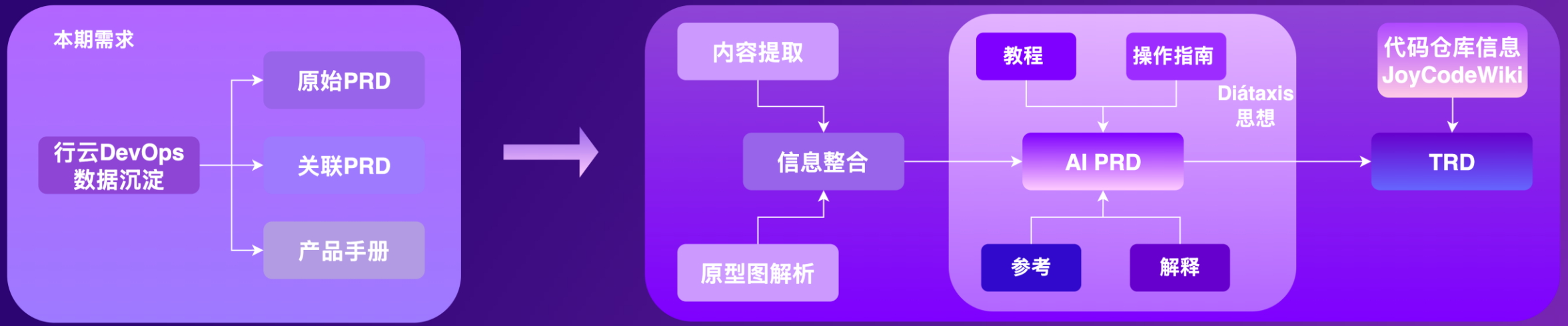
需求管理：AI自动化解析业务需求并生成技术设计，明确变更需求与验收标准。

研发协作：JoyCode赋能，多智能体驱动“意图→代码”的端到端自动化研发流程。

质量交付：AI智能评测，智能部署发布，实时监测与异常自愈，保障高频、低风险交付。

JoyCode + 行云 = 智能化贯通研发全链路

行云协作路径：智能化贯通研发全链路



行云DevOps
集中调度 | 统一协作规范 | 沉淀全链数据

JoyCode + CSR
精准上下文检索 | 高效智能编码 | 代码测试一体化

JoyCodeWiki + 信息整合
汇聚多源数据 | 生成结构化资产

行云协作资产：结构化需求规范AI PRD

Tutorials（场景演示）

将AI PRD的功能演示部分映射为Tutorials的风格，用“从零到一”的叙事风格展示关键场景与典型用户路径，将整个流程结合着原型图解析部分的内容讲解清楚。

Explanation（需求背景）

将AI PRD的背景/动机/权衡等信息映射为 Explanation 的风格，详细分析需求为什么做、价值与范围、做什么、不做什么、策略与权衡、风险与假设、依赖与假设等信息。

Diátaxis
文档设计思想

AI PRD

Explanation（需求背景）

Tutorials（场景演示）

How-to（任务导向）

Reference（客观规范）

……

How-to（任务导向）

将AI PRD的功能实现（怎样实现以达到验收标准）部分映射为 How-to guides 的风格，覆盖主流程与异常分支等的实现步骤。

Reference（客观规范）

将AI PRD的配置详情（术语表、实体与字段约束等）部分映射为 Reference的风格，将整个需求文档的配置规范（如各个功能点的数据字段配置信息等）详细列出。

结构化规范原始需求，零遗漏精准捕捉需求细节，助力AI深刻理解需求

行云协作资产：实际化架构设计规范TRD

关联需求&改动范围

1. 明确关联业务背景与需求目标。
2. 明确改动模块、类型（新增、改造、删除等）等。
3. 分析改动影响，预测改动风险。

架构设计&接口数据定义

1. 设计改造后的目标架构，突出变化点。
2. 数据库变更、数据迁移方案设计、验证。
3. 实现接口改造、接口新增等。

服务实现&项目上线

1. 服务变更设计（模块职责、调用链/状态机、异常处理等）。
2. 监控告警、设计质量评测方案。
3. 设计上线步骤、切量、回滚策略等。

生产级Spec编程：如何让AI理解高级指令

AiDD 8th 2025

Spec编程流程：

Spec编程优势：

消除需求歧义：

采用EARS语法标准化需求描述，确保每个功能点都有明确的验收标准。

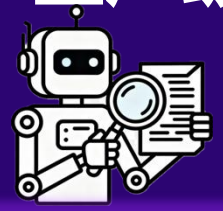
系统架构设计：

从技术视角构建完整的系统，保证各模块间的协同性。

任务原子化管理：

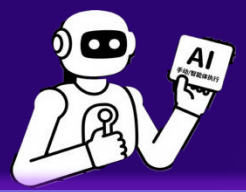
将复杂项目分解为可执行的离散任务，确保开发过程可控可追溯。

生产级Spec编程：如何让AI理解高级指令

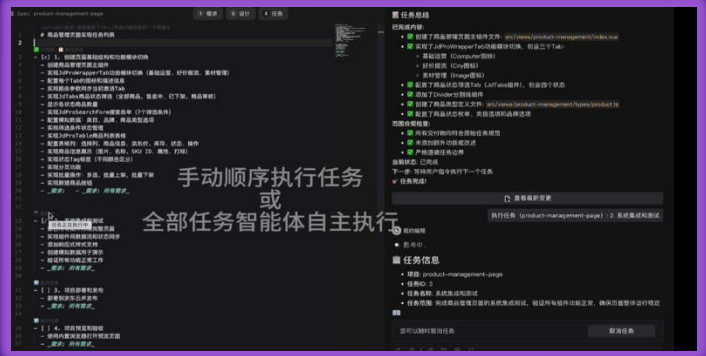


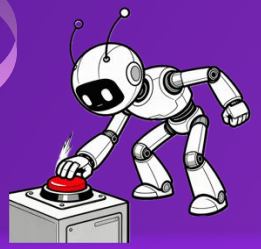
澄清需求

明确任务

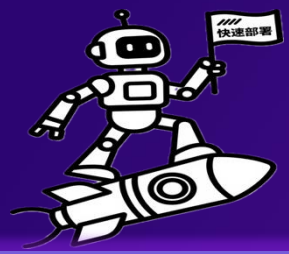


手动顺序执行或
智能体自主执行

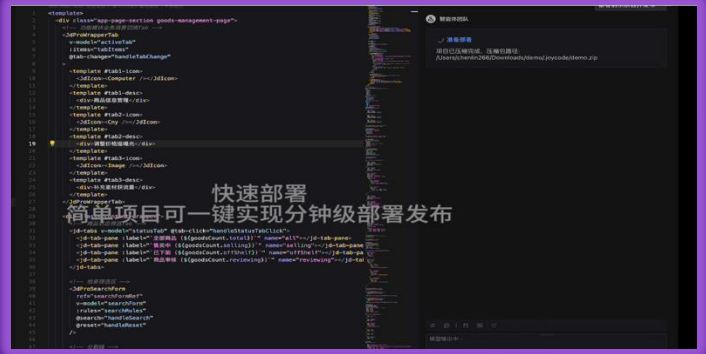


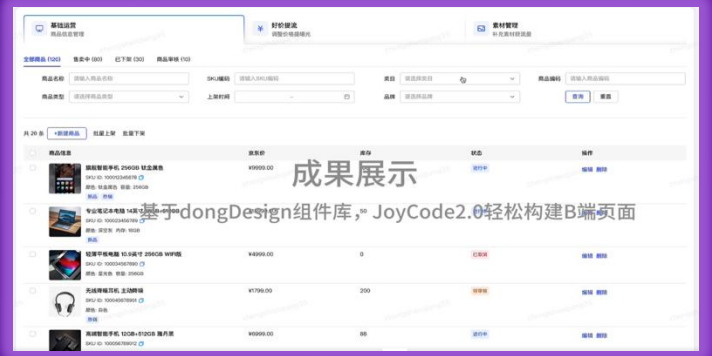


执行任务



快速部署
简单项目可实现
分钟级部署发布





成果展示

JoyCode：从辅助到自动编程进化

JoyCode在京东内部落地成果

40%+
开发周期缩短

50%+
生成代码采纳率

10亿+
累计生产代码量

JoyCode覆盖与活跃率

12,000
集团软开岗员工

90.15%
安装使用率

11,082
月度活跃用户

5,633
日均活跃用户

数亿级
用户产品研发支撑

优秀
内部用户粘性

提升协作效率 • 驱动业务增长

AiDD 8th 2025

PART 05

未来展望：结合多工具场景、定义未来研发

未来展望：结合多工具场景、定义未来研发

AI打破限制，重塑研发模式

AI已从辅助工具进化为深度融合**需求、开发、测试、运维**每一环的“智能体”，实现全流程提效。

AI赋能

AI深度嵌入工作流，改进任务直接集成进开发工具

智能度量

从关注结果到关注过程，度量人机协作与协同产出

持续改进

从数据驱动决策到自动闭环优化

- AI正在重塑软件开发底层逻辑：研发方式和团队协作模式的根本变革
- 新一代软件团队将构建“AI智能体+智能度量+持续改进”的闭环
- 重复性劳动力被AI释放，人类追求更高阶能力，掌握“人机协同闭环”

未来展望：结合多工具场景、定义未来研发

汇聚整合
数据沉淀

多策略构建
全维度知识库

ACE
上下文工程

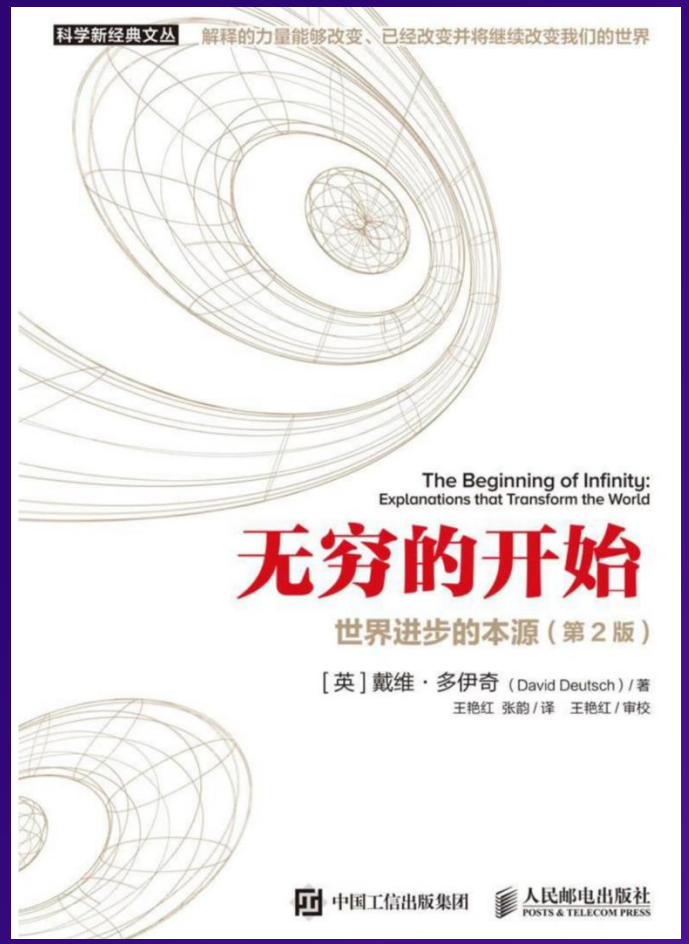
JoyCode + CSR
智能化编码

AI 赋能
自动化评测

端到端提效
研发全流程

全方位构建 “AI智能体+智能度量+持续改进” 闭环

AI真的会让我们“无事可做”吗?



核心观点:

- 新的解决，会带来新的问题
- AI帮我们解决低层次问题，并将我们推向新问题的风口

低层次问题被自动化，我们就会迎来新的需求，随之出现新的工作

科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

KEYLINK ing

AiDD 峰会

AiDD峰会 上海站
AI+研发数字峰会
时间：2026.05.22-23

AiDD峰会 北京站
AI+研发数字峰会
时间：2026.08.21-22

AiDD峰会 深圳站
AI+研发数字峰会
时间：2026.11.20-21

AiDD峰会详情

AiDD × AI+产品峰会

AiDD × 上海站
AI+产品创新峰会
时间：2026.01.16-17

AiDD × 北京站
AI+产品创新峰会
时间：2026.08.23-24

产品峰会详情

AiDD × K

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时代的用户洞家与需求发现

论坛2: AI原生产品战路与商业模式重构

论坛3: AgenticAI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

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