迭进 AIEvoLoop

PRODUCT HANDBOOK · 产品手册

端到端 AI 受控自进化研发引擎

从生成代码，到交付软件。

— 每一次迭代，都是一次进化。

连接企业知识、研发流程与现有 Code Agent，从 PRD 到通过自动验收、可合并的代码，并把每次迭代沉淀为企业知识。

北京数致未来科技有限公司

2026 年 11 月版

www.aievoloop.com

01 · 为什么需要迭进

AI 研发提效，正在从“个人”走向“组织”

今天：个人提效

- Code Agent 让工程师写代码更快
- 提效只覆盖编码这一个环节
- 需求、设计、验收仍靠人推动
- 经验留在个人，团队没有质变

下一站：组织提效

- 从 PRD 到可合并代码，全流程提效
- 流程自动推进，人在关键节点确认
- 经验沉淀为组织资产
- 研发能力随每次迭代自进化

约 **80%**

企业的 AI 研发诉求，集中在存量业务系统的持续迭代

来自与几十家头部企业的一线交流观察

个人编码提速，不等于组织交付提速

仅提效了编码

Code Agent 只加速写代码，需求、设计、验收仍是瓶颈。

存量系统不敢改

微服务链路长，没有人掌握全局，改一处、牵一片。

改了不知道对不对

缺少可依赖的验收依据，质量只能靠最后一道 QA 兜底。

经验锁在人脑里

项目经验只在个人身上，换一个人就从零开始。

> 根因：关键答案往往不只在代码里。缺少企业本体知识，AI 的输出就难以稳定满足业务与工程约束，最终表现为反复修改和返工。

02 · 迭进是什么

Code Agent 生成代码，迭进交付软件

	生成代码 · 单兵 Code Agent	交付软件 · 迭进 AIEvoLoop
交付物	一段代码	spec · plan · tasks · 代码 · 测试 · 验收报告
需求理解	靠人在对话里反复补充背景	读取企业本体，对齐业务与架构约束
质量	人自己拼验证、自己兜底	按改动类型自动验收，人工确认后合并
经验	留在个人的对话记录里	沉淀为本体知识，下一次迭代直接复用

迭进是端到端 AI 受控自进化研发引擎：从 PRD 到通过自动验收、可合并的代码，并把每次迭代沉淀为企业知识。它由四部分组成：

① 企业本体知识底座

业务 · 架构 · 系统 · 基建四层知识本体，让迭进读懂你的系统。

② SDD 流程编排

七阶段规约驱动开发流水线，内置自动验收，人只做确认。

③ 受控自进化飞轮

人工修正沉淀为经验，确认后回写本体，下次迭代直接复用。

④ 可替换 Code Agent

复用企业已有工具与模型，不被任何一家绑定。

接入，不替换：与企业现有工具的关系

企业已经有的

Code Agent 会写代码，Git 管版本，CI/CD 管构建发布，模型服务提供算力。

迭进补上的

企业本体让 AI 懂业务；七阶段编排；合并前自动验收；经验回流减少重复错误。

协同方式

调用企业已选的 Code Agent；代码仍在企业仓库；合并后照常走 CI/CD；以 MCP / Skill 接入内部系统。

02 · 整体架构

离线构建本体，在线迭代交付，工作台持续治理



离线 · 本体构建

- 知识方法论 + 企业代码仓库
- 四个分层 Skill 生成初版
- 代码全局索引，防上下文塌缩
- 人工确认后启用

在线 · 迭代开发

- Web + 桌面客户端双端工作台
- 每个会话独立运行环境
- 流程控制 Skill 驱动七阶段
- 关键节点人工确认

本体管理工作台

- 本体知识可视化浏览
- 合并冲突人工介入处理
- 经验 → 约束流转人工确认
- 本体随代码同步演进

> 知识与代码留在企业侧，迭进提供知识工程、流程编排、验收治理与自进化能力。

03 · 核心能力 ① 企业本体

四层企业本体：让迭进正确行动

按“变化频率 × 与代码耦合度 × 结构化程度”拆成四层，每层独立选择存储、生成方式与加载时机。

本体层	回答的问题	典型内容	服务阶段	存放位置
业务层	业务是什么、规则是什么	元语（实体）、域、场景、操作、关系、业务规则、历史实践	需求理解	随业务域仓库
架构层	系统之间怎么协作	架构分层、调用链路、架构约束、服务治理、历史实践	方案设计	全局 · 根目录
系统层	服务能做什么、哪里不能碰	服务与接口说明、系统约束、系统红线、代码索引、历史实践	任务设计	随代码仓库
基建层	在本公司怎么用基础设施	中间件使用经验、代码规范、工程规范、历史实践	任务 / 编码	全局 · 根目录

> **系统红线**：哪些代码不能动、必须怎么改，由人基于历史与业务情况给出，是 AI 不可越过的边界。

本体构建：Code Agent 自动生成，人确认后启用

01 建代码索引
代码 + 数据库 Schema

02 提取元语
抽取有稳定身份的业务实体

03 归纳业务域
按归属关系逐层向上建树

04 梳理场景操作
从页面、离线任务、消息入口遍历

05 人工补全确认
约束、经验由人补充后启用

> **双分支同步**：每个需求同时创建代码分支与本体分支，验收后一起合并主干；并行冲突由 Git 发现、在工作台由人处理。代码改到哪，本体就同步到哪。

存量本体化的三个现实约束，迭进这样应对

现实约束	迭进的设计选择
文档必然欠缺	以代码与数据库 Schema 为第一手来源自动构建，文档仅作补充；人只做确认，不必从零编写
隐性知识说不出来	在交付中捕获：AI 出错 → 人纠正 → 确认后回写本体；高风险禁区预先划为系统红线
真值源分裂	按需求逐步厘清：只裁决本次触及的冲突，业务层由业务负责人、架构层由架构负责人裁决，结果版本化

03 · 核心能力 ② SDD 流水线

七阶段流水线：迭进自动生成，人只做确认

产品经理按模板给出 PRD.md（一句话需求 · 场景 · 动作 · 预期结果，写到可验收），之后每一份产物都由迭进自动生成；每个阶段经 MCP 只检索对应层本体，避免上下文膨胀、降低 token 成本。

01	**需求理解**	读业务层本体 · PRD.md → **spec.md**：做什么、怎么验收	人只确认
02	**方案设计**	读架构层本体 · spec.md → **plan.md**：改在哪、怎么协作	人只确认
03	**任务设计**	读系统层 + 基建层 · plan.md → **tasks.md**：可编码的任务	人只确认
04	**编码实现**	Code Agent 按任务编码 · tasks.md → **代码 + 测试报告**	人只确认
05	**测评验收**	按改动类型拼接验证 · 代码变更 + 三份 Spec → **eval.md**	自动验收
06	**经验沉淀**	eval.md + 人工反馈 → **learnings.md**，按来源写回四层本体	确认后回写
07	**合并分支**	代码 + 本体双分支 → **确认后合并主干**，照常走 CI/CD	人只确认

案例走一遍：订单列表增加「删除订单」

阶段	迭进自动完成了什么（示意）
需求理解	查业务层本体：确认是用户侧订单，只能删已完成订单，商家端仍可见
方案设计	查架构层本体：放在订单域处理，只面向单端用户，采用逻辑删除
任务设计	查系统层本体：订单表加删除标记、新增删除接口、列表查询过滤
测评验收	命中“新增接口、变更表结构、新增查询”三类改动，自动拼接验证
经验沉淀	“用户侧已完成订单采用逻辑删除、商家端保留可见”，经确认沉淀为订单域经验

编排
流程控制 Skill 驱动全程，按阶段控权

知识
每阶段经 MCP 检索对应层本体

人
只做确认，不写文档、不写代码

03 · 核心能力 ③ 自动验收 · ④ 受控自进化

合并前先把关，每一次迭代都是一次进化

改动类型	自动拼接的验证范围
新增 / 修改接口	接口契约、业务逻辑、异常分支
新增查询	语句正确性、索引命中
变更表结构	结构兼容性、存量数据正确性
发送 / 消费消息	消息格式、能否正确投递
使用缓存	读写一致性、失效策略

测试飞轮

- 专家设计 + 线上问题 + Bad Case
- 构建并持续扩充测试集
- 自动执行，输出 eval.md
- 失败用例做根因分析（RCA），回到需求理解重做，案例沉淀为验收知识

集成测试与业务验收仍走企业既有流程。

受控自进化飞轮

① **人审核修正**
每处修改都被记录

② **经验沉淀**
生成 learnings.md

③ **回写本体**
按来源写回四层

④ **下一次迭代**
复用经验，减少返工

同一经验在多个场景复用后，系统自检提示升级为约束，经人工确认后生效。**人**负责拆解流程、提供经验、关键确认；**机器**负责节点自动化、吸收经验、持续改进。

进化的治理边界：可审、可控、可回滚

进化什么
本体知识、约束规则、测试集；不训练、不修改模型本身。

谁来审核
经验写回本体、升级为约束，都在工作台由人确认。

如何回滚
本体以 Git 版本化管理，任一变更可追溯、可回退。

如何发布
本体变更走分支，随需求验收一起评审、合并。

如何评估
跟踪验收通过率、返工轮次，看进化是否有效。

边界在哪
系统红线由人设定，AI 不能越过；关键操作逐项审批。

04 · 产品体验与企业级保障

像用 Cursor 一样上手，为企业级落地而设计

会话即需求

每个会话选定 Code Agent，在独立环境运行；Web 与桌面客户端双端可用。

按阶段选模型

按企业模型清单与成本策略配置，各阶段按需选模型、控成本。

产物实时可见

工作台右侧实时查看、修改 spec / plan / tasks 等产物。

人介入少：流程自动推进，人只在关键节点出现

桌面 / 浏览器通知

完成或待确认时主动提醒

跨页面提醒

切到别处也能收到待办

逐项审批

允许一次、始终允许或拒绝

出错提醒

一键重试、继续执行

执行事件先落库、再推送：关掉页面或网络中断，回来可续传完整过程。

企业级保障

不绑定 Code Agent

支持 OpenCode、Cursor、Codex、Claude Code、CodeBuddy、Qoder、Trae 等，跟随企业既有选型。

模型可替换

各阶段按需选模型、控成本；支持接入企业私有化模型。

会话级隔离

每个会话独立容器运行，互不干扰，空闲自动回收。

密钥不落前端

API Key 由网关注入容器，数据库只存引用、不存明文。

过程可追溯

思考、工具调用、产物生成全部事件化落库，可回放。

知识归企业

知识与代码存于企业侧，支持私有化部署与私有模型，数据流向可配置。

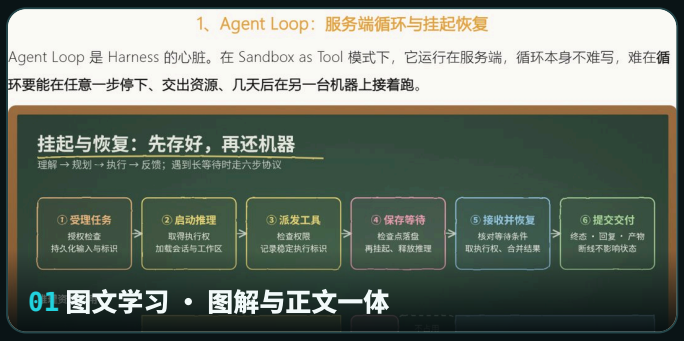
05 · 标杆案例

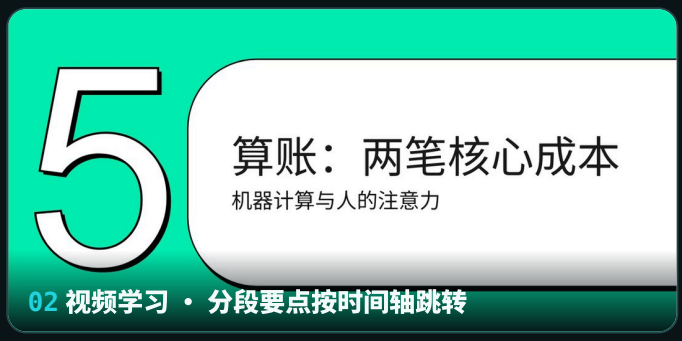
AITutor：由迭进端到端研发交付的 AI 原生产品

AI 原生学习伙伴 www.aiaitutor.com 已稳定运行 10 个月：图文学习、视频学习、播客学习、AI 导师、会员订阅、用户系统六大核心模块，从 PRD 到可合并的代码，均由迭进端到端研发完成，**人只做确认**。

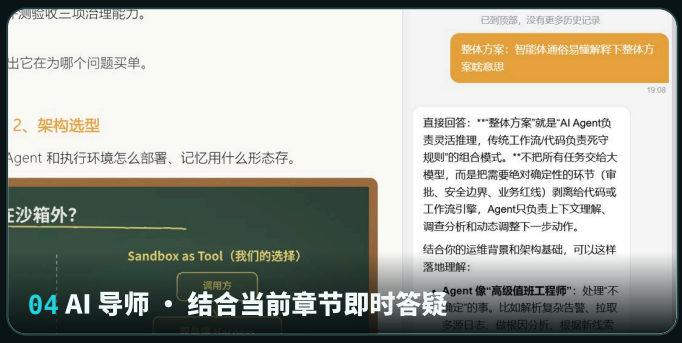
10×+	12	37	86%	99.99%	0
研发效率提升	迭代版本	完成需求	一次通过验收	线上可用性	线上严重故障

数据来源：AITutor 近 10 个月线上运行与研发记录。研发效率口径：同一批需求，传统方式估算 10 人天，使用迭进 1 人天。













覆盖内容生成、多媒体、对话、支付与账户等典型系统形态，交付的是一整套可上线的软件。更多行业客户案例陆续更新中。

06 · 落地方式与合作

三步合作：先免费试用，验收通过再付费

01

免费试用

在线 SaaS 试用 14 天，看懂产品、确认价值；代码不能上云的企业可直接进入本地部署。

02

本地部署

部署到企业内网，部署时陪您交付首批真实需求，FDE 前沿部署工程师全程陪跑。

03

按年订阅

验收通过后开始计费，按研发规模收取年费，含版本升级与标准服务支持。

部署评估会共同明确

部署范围 · 真实需求 · 双方投入 · 实施周期 · 验收指标

验收指标示例：需求研发周期、人工投入、单需求模型成本、返工轮次、一次验收通过率、合并后缺陷。提效，不以牺牲质量为代价。

迭进的承诺

- 每一档都是全能力版：功能不分档，价格只看规模
- 验收通过才开始计费
- 本体和代码都留在您自己的 Git 里

关于我们：懂企业研发，更懂 AI 落地

130+
企业级 AI 大模型应用项目（2023 年以来）

70+
服务的头部企业

200+
场企业 AI 落地培训

18 年
大厂技术架构经验

使命 MISSION

让 AI 交付可信软件，让组织随迭代持续进化

愿景 VISION

让每一家企业，都拥有自我进化的研发组织

创始人兼 CEO 孙玄（玄姐）：18 年大厂技术架构经验，对标阿里 P10；前 58 集团技术委员会主席、前转转集团首席架构师。累计获得超亿元投资，投资方包括初心资本、58 产业基金、梅花创投、中赢创投、泰亚投资等。



预约试点评估

官网：www.aievoloop.com

直接联系玄姐：微信扫码，或邮件 sx@aiaitutor.com