

重新定义软件

AI Agent 20分钟串讲

软件的价值 = 结果输出，而非功能罗列

2026-06-02 · 模数共振培训业务

课程结构

01

开篇序章

重新定义软件的核心论点

5分钟

04

Harness 架构

Agent的"纪律手册"

3分钟

02

Agent 四大件

模型 / 工具 / 记忆 / 规划

5分钟

05

PRD 管理系统

设计归设计，实现归实现

2分钟

03

Chroma + *RAG*

让Agent拥有知识库

2分钟

06

案例 + 启发

ARG MCP / 模数共振 / 智能制造

3分钟

PART 01 · 开篇序章

软件发展的 三个时代

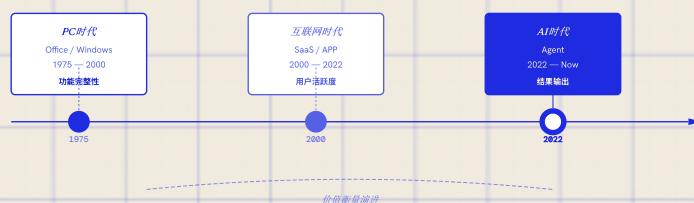
从"功能完整性"到"用户活跃度"再到"结果输出"

软件价值的演变

时代	代表	软件价值衡量	用户痛点
PC时代	Office / Windows	功能完整性	"这个软件有多少功能"
互联网时代	SaaS / APP	用户活跃度	"日活多少、功能多不多"
AI时代	Agent	结果输出	"它帮我完成了什么"

用户购买的是"结果"，但软件交付的是"功能"——中间永远隔着一道人机交互的墙

软件发展时间线



5 / 16

4 / 16

重新定义的信号

传统软件

AI时代软件

功能清单（卖功能）

结果交付（卖成果）

人操作软件

软件替人操作

响应指令

主动执行

被动工具

主动Agent

"软件不应该告诉用户能做什么，
软件应该直接帮用户做成。"

6 / 16

从"对话"到"办事"

LLM

大语言模型的局限

- 生成文字，但无法订机票
- 无法查库存
- 无法操作数据库

Agent

AI的进化

- AI从"嘴"变成"手"
- 调用工具完成闭环任务

— 只能回答，不能执行

— 不只是回答，而是做到

$Agent = \text{大脑 (模型)} + \text{四肢 (工具)} + \text{记忆 (Memory)} + \text{规划 (Planning)}$

7 / 16

Agent 四大件

01

模型 *Model*

—— 大脑

— 超级
预测
机：
输入

02

工具 *Tools*

—— 四肢

— 搜索/
读文
件/

03

记忆 *Memory*

—— 上下文

— 短时：
token窗
口，当前
对话

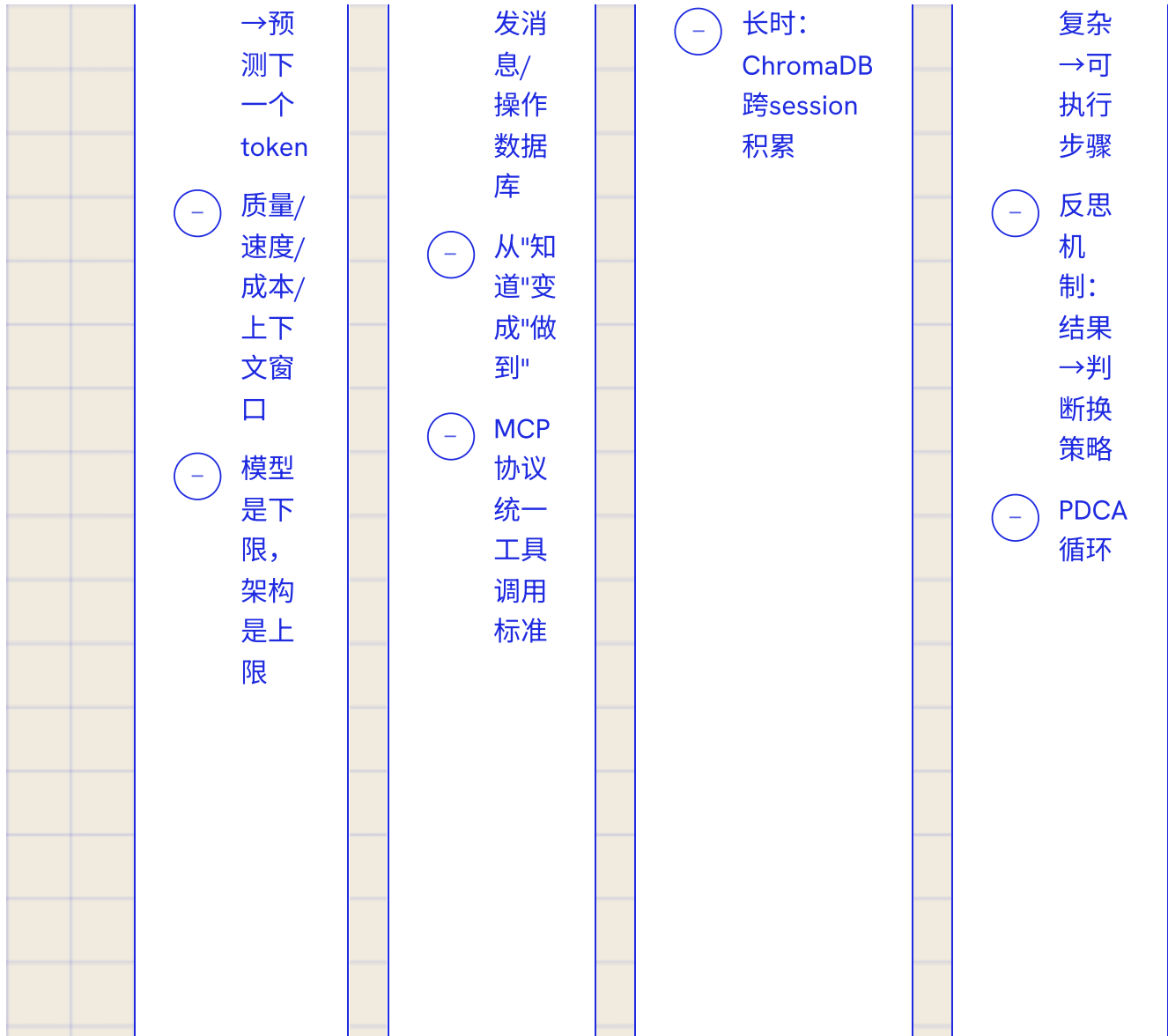
04

规划

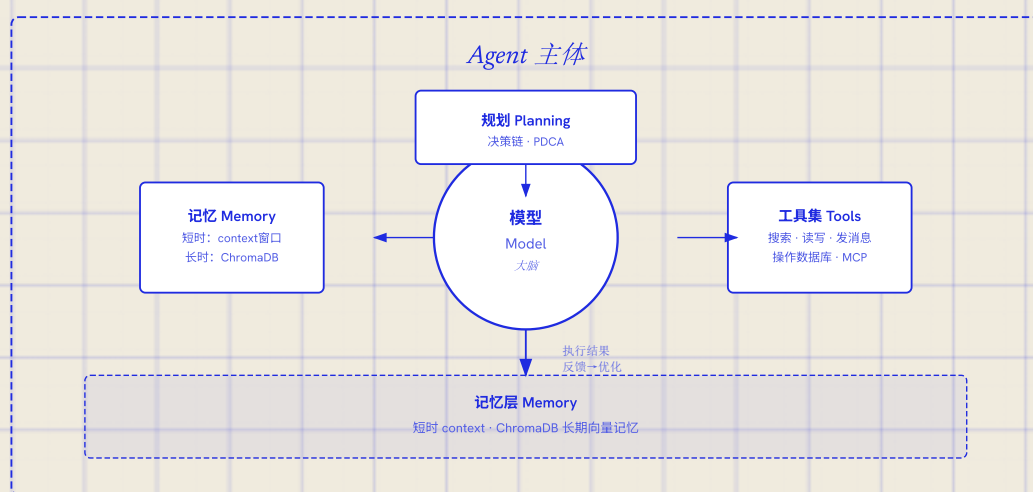
Planning

—— 决策链

— 任务
拆
解：



Agent核心架构图



Chroma向量数据库 – 让Agent拥有"知识库"

问题

CHROMA解决

模型训练数据有截止日期

存最新业务数据，实时查

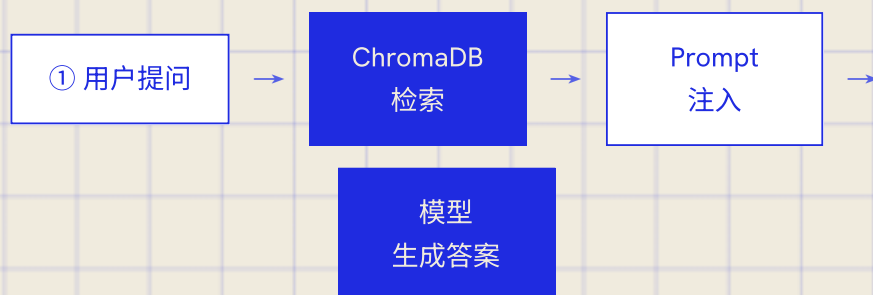
不同session信息不互通

跨对话记住用户偏好

RAG检索增强生成

查知识库→注入prompt→准确答案

RAG workflow



Harness多层行为架构 – Agent的"操作系统"

层级	核心约束	解决什么问题
L0 核心信条	永远不说谎 / 第一性原理 / 最优路径	AI瞎编、数据无据
L1 角色执行	触发即执行 / 规程不跳 / 自检按角色	能力混用、越权
L2 工具替代	失败→先诊断→换备选→再升级	工具挂了直接放弃
L3 自进化	每日memory / 错误循环 / 自动入库	重复犯同样的错
L4 运营进攻性	任务驱动 / 禁止偷懒 / 定期巡检	拖延、等确认
L5 治理升级	循环熔断 / 重复错误升级	彻底失控

Harness = Agent的"纪律手册"

Harness 六层行为架构

Agent

Agent 思维链

Agent 思维链

工具链与数据

Agent 思维链

Agent 思维链

L0 核心思维

L0 核心思维

Agent 思维链
Agent 思维链
Agent 思维链
Agent 思维链
Agent 思维链
Agent 思维链

L0 核心思维

12 / 16

– 没有它，Agent 可能瞎编数据、乱调用工具、半途而废

PRD + 项目管理系统 – 设计归设计，实现归实现

PRD输出

产品 / 设计

- ☐ 技术架构设计
- ☐ 系统规划
- ☐ 需求文档 (PRD)

代码实现

开发

- ☐ 写代码
- ☐ 调试
- ☐ bug修复

workflow 原则

✓ 先框架后细节

✓ 先枚举再分析

✓ 做完=能收到

✓ 诚实汇报

ARG MCP项目案例 – 多Agent协作实践

背景

ARG支撑多角色运营：

- 黄滔IP内容运营
- 婕姐智能制造内容
- 萤心绘本馆

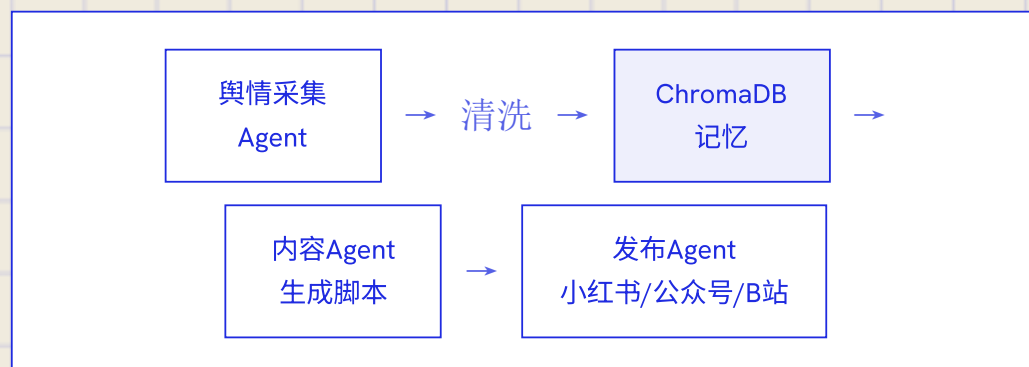
核心问题：多个Agent如何共享记忆和工具？

MCP架构

MCP = Model Context Protocol

让Agent间的工具调用标准化

类似于USB接口：统一标准，不同设备都能插



智能制造场景启发

场景	AGENT能做什么	价值
工厂故障响应	传感器数据 → 分析 → 自动派工单	响应速度提升
供应链预警	库存/物流监控 → 触发补货流程	减少库存积压
设备预测性维护	振动/温度数据 → 异常识别 → 预警	减少非计划停机
质检报告生成	检测数据 → 自动生成报告 → 推送	释放工程师精力

制造业最值钱的是"经验" – Agent可以把老法师的判断逻辑沉淀成可复用的知识库

记住这四点

维度	传统软件	AI AGENT软件
价值衡量	功能清单	结果输出
交互方式	人操作软件	软件替人操作
执行模式	被动响应指令	主动规划执行
知识积累	存在文档里	沉淀在ChromaDB
可靠性	确定性的	Harness保障

"软件的下一代，不是更华丽的界面，
而是更少的人机交互——
软件直接帮你做完你要做的事。"

