



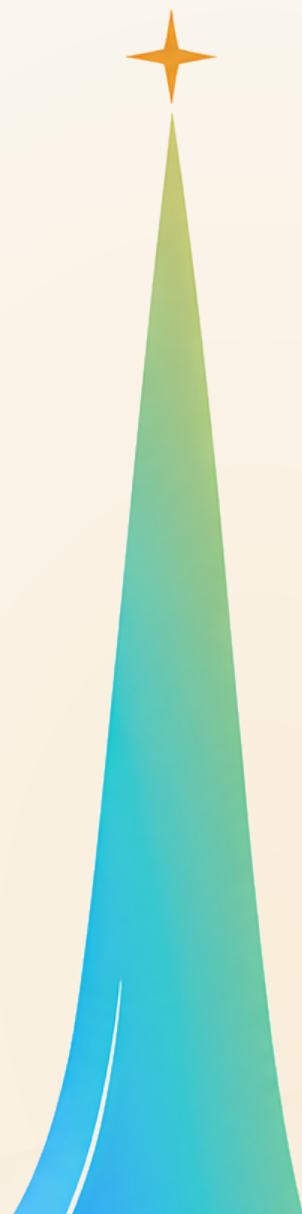
Agent 七强交流局

从工具热潮，回到任务、协作与管理

一场三个小时线上讨论的知识化整理

2026.08.16 AI 神思会线上分享

不替大家选冠军，只帮大家找到适合自己的入口。



这不是一份产品排行榜

原始会议是一场开放、跳跃、带有大量现场追问的线上交流。讲义将它重新编排成一条可以复查的阅读路径：先理解 Agent 的组成，再看不同路线，最后落到真实工作中的选择、文档和治理。

整理原则

- 保留方法论、经验判断和可执行步骤。
- 删去寒暄、重复表达和与主题无关的现场噪音。
- 参与者以职业角色和使用场景呈现，减少个人信息暴露。

阅读提醒

- 产品名称和能力会随版本、地区、权限而变化。
- 本文是社群经验汇编，不是厂商官方说明或安全承诺。
- 涉及删除、发布、外部平台和敏感资料的操作，都应保留人工确认。

“真正的进阶，不是不断追逐更强的工具，而是逐步扩大自己能看见、能验收、能恢复的范围。”

01

THE MAIN CLAIM

先别问谁最强

先问：我想交出什么工作？我愿意给它多大权限？我有没有能力验收和恢复？

01

任务先于产品

同一个 Agent，在写代码、做讲义、管理文件和经营内容时，会呈现完全不同的结果。

02

能力与成本成对出现

能看得更多、记得更久、行动更远，也意味着更高的上下文、权限、费用和维护成本。

03

管理决定上限

需求是否清楚、文档是否持久、检查点是否明确，往往比换一个名字更能改变结果。

本讲义的核心问题：怎样让 Agent 做得起、验得过、管得住？

从聊天到自建：六层升级地图

会议中有人把个人使用路径概括成一条从“会问”到“会管理”的路线。它不是等级考试，而是责任范围逐步扩大。

0	网页版问答 得到一段答案，任务边界和结果检查主要由自己负责。	低风险 · 立即反馈
1	使用 Agent 让它读文件、运行命令、生成文件，开始产生可交付产物。	需要权限意识
2	固定流程 把重复任务做成稳定步骤，加入检查点、模板和验收标准。	开始复用
3	沉淀 Skill 把已经验证过的工作方法写成可复用的能力，而不是每次重写提示词。	工作流资产化
4	改 Harness 与记忆 开始管理模型、工具、上下文、记忆、日志、重试和权限边界。	维护责任上升
5	微调或私有化 只在独有资料、合规隔离或规模化收益足够时考虑。	高成本 · 高控制

模型、Agent、环境：不要混成一件事

会议里最重要的一次澄清是：模型是模型的事，工具是工具的事。你看到的“智能”，往往是三层共同作用的结果。

01 模型 / 大脑

理解、推理、生成和判断。训练数据与模型版本会影响它的基础能力。

02 Agent / 工作身体

系统提示、工具调用、循环方式、计划方式和任务边界，决定它怎样工作。

03 环境 / 工作现场

文件、项目目录、操作系统、数据库、账号权限和外部平台，决定它能做什么。

模型像发动机，Agent 像整车，环境像道路和车库。换发动机不等于换了整车。

Reactive

边想边做，执行一步再决定下一步。

Plan

先拆任务、排步骤，再按计划执行。

Human-in-the-loop

关键节点由人确认，避免自动化越界。

FIELD CHECKLIST

判断 Agent，只问八个大白话问题

不把注意力耗在抽象跑分上，直接看它是否能在你的工作现场完成闭环。

01 听得懂吗？

能否理解你的真实目标，而不是只抓住一句指令。

02 看得全吗？

能否读取需要的文件、资料和上下文。

03 动得手吗？

能否调用工具、修改文件或产生可验收产物。

04 记得住吗？

能否保留项目状态，又不把噪音永久带入下一次任务。

05 干得久吗？

长任务中是否能保持方向、处理失败并继续工作。

06 回得去吗？

出错后能否查看日志、定位变更并恢复。

07 管得住吗？

权限、外部输入和发布动作能否分层控制。

08 养得起吗？

费用、速度、维护和学习成本是否符合你的场景。

手越长、记得越久，越需要权限边界与阶段性验收。

02

THE FIELD MAP

七位选手，四条路线

它们不是七个同类产品，而是四种不同的管理关系。

01

专业程序员

Claude Code
Codex

代码、仓库、测试与交付

02

职场全能同事

WorkBuddy

文档、数据、PPT 与文件

03

长期数字助理

OpenClaw
Hermes

跨软件行动、记忆与复用

04

可改装工作台

Pi Agent
DeepSeek Harness

学习、改造、私有化与深度控制

先按任务选路线，再按自己的验收与维护能力选产品。

Claude Code 与 Codex：搭档，还是委派

01

Claude Code

坐在旁边的资深程序员

更像结对工作：一边理解，一边修改，一边验证。适合陌生仓库、模糊问题和需要持续纠偏的任务。

- 优势

互动感强，能持续讨论任务细节。
- 提醒

长对话会变贵、变乱，关键改动仍需人掌舵。

02

Codex

可以委派的远程开发小队

更像接收工单的执行者：任务清楚时可以阅读、修改、测试并交付，也适合拆分独立工单。

- 优势

适合明确、可验收的任务。
- 提醒

任务不清会高效跑偏，并行越多，验收压力越大。

选择提示 你需要的是“边做边理解”，还是“把结果交回来”？前者先试搭档型，后者先练习委派型。

本页为会议参与者的使用体感整理，不代表统一产品结论；同一工具在不同系统、模型和权限下可能呈现不同结果。

WorkBuddy、OpenClaw、Hermes： 成品、行动与长期记忆

03

WorkBuddy

能交办公成品的全能同事

适合文档、数据、PPT、研究和本地文件。对于刚从网页版走向 Agent 的人，成品导向更容易建立信任。

先从可验收文件开始。

04

OpenClaw

拿着很多钥匙的数字管家

连接聊天、文件和服务，适合跨软件、定时和重复流程。连接越多，外部输入和权限风险越具体。

先只读，再行动；先低风险，再扩大范围。

05

Hermes

会积累笔记和技能的长期伙伴

适合长期任务、技能复用和经验积累。它的优势也是风险：错误记忆会被重复使用。

记忆要复审，技能要瘦身。

长期合作不是“让它永远记住”，而是让它知道什么该记、什么该忘、什么需要重新确认。

Pi Agent 与 DeepSeek Harness：从使用者到改装者

06

Pi Agent

可以拆开的乐高机器人

- 适合学习模型、工具和工作步骤。
- 结构透明，哪里不合适就改哪里。
- 代价是体验、稳定性和配套能力要自己补。

07

DeepSeek Harness

买发动机以后自己造车

- 适合私有化、模型自主和深度定制。
- 可以自己组织工具、环境、记忆和插件。
- 监控、重试、安全、升级和运维也由自己负责。

不要为了造轮子而造轮子 只有控制价值大于长期维护成本，自建才值得。先用成品验证需求，再决定要不要改造底层。

COMPARISON / 一张表

比较的不是强弱，而是管理成本

下面的表格只回答一个问题：什么样的工作起点，更容易和哪种 Agent 建立稳定关系？

路线	像谁	适合的起点	最大提醒
Claude Code	结对程序员	模糊、复杂的代码问题	人仍需持续掌舵
Codex	开发小队	明确、可验收的任务	任务不清会并行跑偏
WorkBuddy	全能同事	文档、数据、PPT 与文件	产物和权限仍需验收
OpenClaw	数字管家	跨软件重复流程	权限与外部输入风险
Hermes	长期伙伴	记忆与技能复用	错误记忆和过时技能
Pi Agent	乐高工作台	学习与改造 Agent	配套能力要自己补
DeepSeek Harness	自建汽车	私有化与深度控制	长期运维自己负责

没有绝对冠军，只有与你的任务、权限和验收能力匹配的路线。

03

REAL CASES

真实案例：Agent 不是玩具，是工作现场

会议最有价值的部分，不是产品名称，而是不同职业的人怎样把 AI 接进自己的业务。

CASE 01

教育工作者

从课后反馈，到排课、内容和家长沟通。

CASE 02

新媒体运营

从零散提示词，到自己的内容工作台。

CASE 03

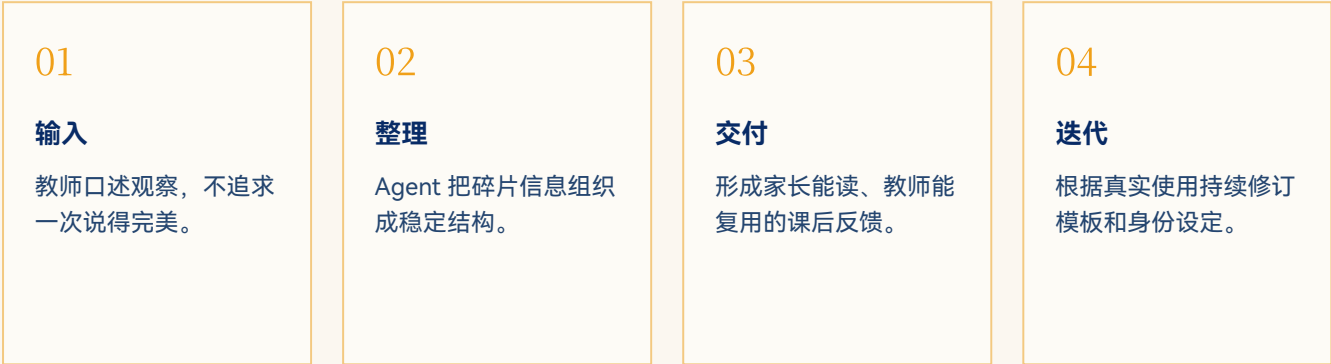
游戏美术

从不会编程，到用 AI 完成独立游戏的大部分流程。

CASE 01 / EDUCATION

把两分钟观察，变成一份可交付反馈

一位数学教师分享：每节课后输入两三分钟语音，Agent 生成包含课堂内容、学生卡点、作业和心理状态的结构化反馈，再直接发给家长。



关键变化 AI 不只是节省打字时间，而是把个人观察沉淀成稳定的服务体验。

可迁移

任何需要“口述观察 → 结构化交付”的工作，都可以尝试。

不能忽略

涉及学生、客户或健康信息时，必须先处理隐私和数据边界。

CASE 02 / CONTENT OPERATION

从零散提示词，到自己的内容工作台

新媒体运营者分享的不是“某个模型最好”，而是一个很典型的组织问题：文案、图文、视频、公众号和小红书，本来都是一件件手工任务。

选题 → 长稿 → 图文 → 平台适配 → 人工发布

有效的做法

- 按业务拆出不同角色，而不是让一个对话框包办一切。
- 把已经跑通的文案、配图和格式要求沉淀下来。
- 让 Agent 交付可复制、可审阅的半成品或成品。

仍然需要人

- 平台规则、账号风险和最终发布不能完全自动化。
- 流量和转化是业务结果，不能只看内容生成速度。
- 每个人的工作台都应从自己的业务出发定制。

工作台不是“多装几个 Agent”，而是把业务中的重复关系组织起来。

CASE 03 / GAME & CREATIVE WORK

当 AI 看见的是整体，而不只是局部

游戏美术从业者分享：使用 Kimi K3 与相关编码工具制作独立游戏，并尝试让模型连接 Blender，根据一张低多边形场景图去复刻整体美术氛围。

“我给它一个场景的效果图，它关注的是整个游戏的氛围，而不是只盯着某一个模型。”

01

全局目标

先说明想做什么样的游戏、场景和体验。

02

局部执行

把模型、机制、素材和代码拆成可操作任务。

03

人类审核

人不必重复控制每个局部，但要判断整体是否偏离。

重要区别 模型、工具和工作环境要分开评价。K3 是模型，Kimi Code 是工具；同一个模型换了工作环境，也可能呈现不同能力。

04

LONG TASKS

长任务怎么跑：上下文 + 持久化

当任务从几分钟变成几小时，真正决定结果的，不是“再补一句提示词”，而是能不能管理上下文和过程。

01

先说清目标

说明要交付什么、给谁用、什么算完成。

02

允许它反问

模糊处在执行中澄清，不必一开始写出巨型 PRD。

03

拆成任务组

方案、实现、测试、回归和验收各自有记录。

04

边做边归档

让下一次只读取需要的资料，而不是重读全部历史。

长流程的两个核心词：上下文，让 Agent 知道现在在做什么；持久化，让它知道过去发生过什么。

PROJECT MEMORY

文档是项目资产，代码只是其中一部分

技术参与者分享了一种轻量的项目文档组织方式：最外层的入口文件负责导航，具体事实放在按主题拆分的目录里。

PROJECT / 项目根目录

AGENTS.md	项目依赖、工作边界、遇到问题先去哪里找
README.md	每个目录和文件的作用，以及当前项目地图
docs/	方案、决策、接口、测试、验收标准
logs/	本次迭代做了什么，出现过什么问题
tasks/	待办、进行中、已完成和下一步

为什么这样做

入口文件短而稳定，细节按需读取。Agent 需要回溯时，可以快速找到对应文档，而不必把整个项目全部塞进上下文。

什么时候要变轻

短期、一次性任务不必建复杂目录。长期项目才值得投入文档结构；场景越多变，越需要按项目拆开。

MEMORY & RAG

记忆不是越多越好，召回才是重点

会议中关于向量库的讨论，最后落到一个很朴素的判断：存进去不等于用得上，能在正确时刻召回正确内容，才是真正的记忆。

全部塞进上下文

简单，但会越来越贵

资料少时容易上手；资料一多，噪音、重复和 token 成本会一起上涨。

按需检索

更省，但依赖召回质量

通过关键词、向量或混合路由取回少量资料；检索错了，后面都会错。

问题 → 检索 → 排序 → 注入上下文 → 生成

工程提醒 大规模文档的向量化需要 embedding 模型和算力；先估算文档量、更新频率、召回延迟和维护成本，再决定是否引入。

权限、平台、数据：别让效率遮住风险

01 文件权限

“移动”不等于“删除”。重要目录先备份，破坏性动作设置确认点。

02 外部平台

自动发帖、私信和跨平台操作可能触发封号或合规问题，默认保留人工发布。

03 私密资料

学生、客户、公司内部数据和未公开资料，不应直接送入不清楚边界的外部模型。

04 长期记忆

记忆需要查重、分层和定期清理；错误经验会在下一次任务里继续放大。

05 费用与上下文

工具、skill 和历史资料越多，初始化上下文和缓存成本可能越高。

06 验收与恢复

每个长任务都要有阶段性产物、日志、测试和回退方式。

先增加观察能力，再增加行动权限；先允许撤销，再允许自动化。

PRODUCT THINKING

新需求来了，先别立刻让它写

会议里一个反复出现的疲惫感是：想到一个功能就让 Agent 做，过几天又想到一个功能，再继续补。最后不是产品越来越完整，而是人越来越累。

灵光一现

我突然想加一个功能。



先当产品经理

它服务谁？最终要到哪里？



形成项目边界

优先级、依赖、验收标准是什么？



再交给执行

让 Agent 在边界内推进，并记录结果。

小任务

目标单一、风险低、一次性完成的事情，可以直接交代并检查。

长项目

需要先建立目标、目录、任务、日志和验收方式，再让 Agent 持续推进。

需求越模糊，越应该先谈方案；任务越危险，越应该先谈边界。

05

A PRACTICAL PATH

给自己的 Agent 选择一条起步路

不要一次学完所有工具。让每一步都对应一个真实任务，并且能在下一次继续复用。

01	02	03	04
从可验收的小事开始	固定一个稳定流程	沉淀成文档或 Skill	再扩大权限与范围
排课、课后反馈、文案初稿、资料整理。	写清输入、输出、检查点和不能做的事。	让下一次少重复交代，而不是增加更多复杂工具。	从只读到可写，从单文件到项目，从单 Agent 到协作。

低门槛入口

WorkBuddy

→

项目协作

Claude Code / Codex

→

长期记忆

Hermes

→

深度改造

Pi Agent / Harness

这是一条会议中的经验路径，不是唯一答案。你的业务、设备、预算和风险偏好，都可能改变顺序。

START CARD / 起步卡

写下下一次可执行的尝试

不追求一步到位，只把下一次真实行动写清楚。

01 我想交出的任务

02 我先试的 Agent / 工具

03 我最担心的失败

04 必须由人确认的动作

最适合你的，不一定最聪明；而是做得起、验得过、管得住。

技术会变，工作方法可以留下

从网页版到 Agent，从 Skill 到 Harness，工具会不断更名、升级和被替代。真正值得沉淀的，是你对任务的理解、对结果的验收，以及对系统边界的管理。

认识

知道模型、工具和环境分别在做什么。

比较

按任务、权限、成本和恢复能力选择路线。

选择

从一个低风险、可验收的真实任务开始。

沉淀

把有效流程写成文档、Skill 和可复用资产。

愿每个人都找到自己的 Agent 入口，也保留对结果负责的能力。

视频号 JKL神思记 · 小红书 JKL神思记

来源与说明

主要来源：2026-08-16 AI 神思会线上分享录音文字稿；并参考《Agent 七强讨论会：PPT 思路与逐字稿（互动串讲版）》的现场结构。产品名、技术术语和明显的语音识别错误已按上下文做有限规范化。

公开发布前，建议再次核对产品名称、版本能力、参与者授权和敏感信息。