

# 从ChatGPT到Claude Code：一个股票研究员如何让AI真正干活

原创 奔波儿r 奔波儿r 2026年1月25日 18:31

## 每天第一件事

早上到办公室，把刚取的瑞幸放在桌上，打开电脑的第一件事就是启动Claude Code。输入一句话：“帮我生成昨天的监控日报。”

10分钟后，近100只股票的异动监控、新闻分析、假设追踪就全部完成了。

\* 读取核心配置文件... (`ctrl+c to interrupt · ctrl+t to hide todos`)  
└  读取核心配置文件 (`watchlist`、最新 `today`、`todos`、模板)  
 运行股票数据脚本获取异动  
 对重要异动搜索新闻  
 假设追踪 - 搜索最新进展  
 生成并保存 `Today` 报告

公众号 · 奔波儿r

这不是ChatGPT能做到的事。

**Claude Code是ChatBot的下一代产品形态——一个真正可以个性化、本地部署、和你协作做事的agent。**

如果发生异动，他会帮我搜索相关的新闻，分析原因，附上公开的分析师观点。同时把这些异动和我最关键的几个投资假设链接在一起，帮助我更新核心的投资主题。

## # Today - 2026-01-22

### 今日焦点

#### ● 重要异动 (4只)

##### 1. INTC Intel ↑ 11.72%

- 价格: \$48.56 → \$54.25
- 市值: \$2,704亿 (大盘股)
- 原因:
  - NEW 2026-01-22 分析师关注 Agentic AI 趋势下的 Intel 机会
  - CES 新芯片发布
- 关联假设:  假设#5 Agent爆发 - CPU需求增长
- 逻辑链: Agentic AI 用量增长 → CPU 需求刚性 → Intel IDM 2.0 转型 + 18A 工艺 → 受益
- 行动建议: 验证涨幅是否由 Agent 驱动，关注后续财报验证

公众号 · 奔波儿r

## 假设 #5：2026年 AI Agent 爆发带来的硬件需求变化

确定性：待标注

最新进展（2026-01-21）：

- AI Agent 采用率持续增长：GitHub Copilot、Claude Code 等工具在企业中快速普及
- Intel 上涨 +3.41%：验证 CPU 需求上升 (Agent 驱动)
- 逻辑链：Agent 用量 ↑ → Token 消耗 ↑ → CPU/HBM 需求 ↑ → Intel/SK Hynix 受益

关键指标：

指标	上次	本次	趋势
----- ----- ----- -----			
Intel 股价   -   \$48.56 (+3.41%)   ↑			
AI Agent 采用率   -   持续增长   ↑			

来源：

- [AI Coding Assistants 2026: Enterprise Guide](#)
- [GitHub Copilot Statistics 2026](#)

公众号 · 奔波儿儿

同时在异动和主题监控以外，也会和我的todo系统链接，提醒我今天应该做什么。todo是从前一天我写的日报中提取的。

### ● P0 - 今日必须

无

### ● P1 - 本周重点

任务	截止	状态

### ● P2 - 有空再做

--

公众号 · 奔波儿儿

读完这篇监控日报后，我快速了解了昨晚夜盘发生的事情。对于一些影响核心假设判断的事件，我再用GPT和Claude Code同时搜索一遍，让他们更大范围地搜索相关新闻去了解来龙去脉。

同时我还会再开一个Claude Code窗口，要求他和我继续升级我们俩的协作系统，每次升级完成以后，我还得让他反思一下，我们这次交流有什么收获，**AI也是需要反思和复盘才能进步的！**

你这次和我协作有什么反思和收获吗？

- 几点反思：

发现计划与实际数据不符时的处理

  - 我直接按实际内容创建，事后才说明差异
  - 应该做的：发现不符时先暂停，问 Yibo 是按计划命名还是按实际数据命名

做得好的部分

  - 并行读取两个文件、并行创建7个假设文件，效率高
  - 用 `TodoWrite` 追踪进度，清晰

关于系统本身的思考

- 当前设计依赖 Claude 在 `/today` 时理解并执行 Step 7 同步逻辑
- 没有自动化脚本校验同步是否成功
- 如果未来假设数量增加或命名变化，需要手动维护映射表

公众号 · 奔波儿r

## 人如何帮助AI完成工作

Claude code完全改变了我的工作模式。

此前我需要打开Wind看看自选股里面的一百多只股票的涨跌幅，如果有明显变化的，自己去搜索相关新闻。Chatbot形式的AI只是提高了搜索这个阶段的效率，但无法更加个性化地满足我的其他工作流和需求。

相比于Chatbot，Claude code最大的优势就是**本地部署**——分层的记忆能力、本地文件的读写权限，这使得他可以在本地完全基于你的要求、记忆和工作流，部署一整套属于你自己的系统。而且作为最强大的Coding模型，Claude code 还能够帮你手搓各类工具，例如把报销发票一键输出到Excel中，PDF转Markdown等各种工作。

以下就是Claude code帮我设计的协作文件架构

## ▣ 架构设计决策

### 文件结构设计

```
D:\BaiduSyncdisk\Claude\  
├─ CLAUDE.md # 全局协作元规则  
│  ├─ 三层穿梭哲学  
│  ├─ Linus 代码哲学  
│  ├─ Token 优化原则  
│  └─ 项目索引  
      └─ workspace/ → 投资工作系统  
  
└─ workspace/ # 投资工作系统（集成式）  
    ├─ workspace-config.md # 投资场景交互原则配置  
    ├─ today/ # 日常运营  
    ├─ research/ # 研究协作  
    ├─ decisions/ # 投资决策  
    ├─ knowledge/ # 知识管理  
    ├─ weekly/ # 周报归档  
    ├─ monitoring/ # 监控系统  
    ├─ principles/ # 投资原则  
    ├─ memory/ # 上下文存档  
    └─ meta/ # 系统元数据  
  
└─ Claude 使用指南.md
```

### 关键设计原则

#### 1. 分层配置

- **CLAUDE.md**: 全局协作元规则 (通用、静态)
- **workspace-config.md**: 投资场景专用配置 (轻量、交互原则)
- **专用文件**: 动态内容 (假设、进展、笔记等)

#### 2. 自动加载机制

触发主题：投资、股票、研究、假设、决策、周报、行业、公司、异动、监控等

**人+AI>AI>>人**

我觉得未来所有白领的工作流，都会被AI深度地改变。我用了一个月左右，与其说如何更好地让AI帮助我们工作，不如说是人如何更好地帮助AI去理解和运行我们的工作。

四十二章经中有一期播客，请了在硅谷创业的Palona AI联合创始人来讲AI时代初创公司的组织能力，这期强烈建议大家去听。他的核心结论是：

**未来追求的是 人+AI > AI，而不是 人+AI > 人**

完整公式：**人+AI > AI >> 人**

用不好的话：**AI > AI+人 >> 人**

未来的执行主体是AI。

那么为了帮助AI更好地完成我们的工作，到底需要怎么做？AI所需要的我目前看来主要是三点：数据、工作流、负责的人。

## 一、干净的数据

数据很好理解，在做股价监控的时候我满大街找免费API，当时想着如果有一个Wind的数据库API那该多好。利用API接入数据，就可以用自然语言提出对公司各类的财务指标的筛选要求，同时根据结果随时进行修改。

以后就终于不用打开落后的excel，一个一个单元格输入落后的公式，还要忍耐公式太多的excel卡机。

其次是历史积累的数据，我会把我的过去几年的报告都喂给他，让他总结一个我的投资画像，我的风格，我关注的要点，然后让他以后研究的时候就按照我的原先的思路来，让他真正意义上成为了解你的agent。

## 二、清晰的工作流拆分

这个很重要，如果你问Claude code，“请你帮我找出下一个一年十倍的股票”，大概率他是无法帮你完成的。但是如果你把你的工作流拆分成几个环节，然后让他从总体以及各个环节都用Plan Mode规划一遍，他就能够至少建立一个能用的自动化系统。

对于研究员来说，工作的流程我会把它拆解成为5个环节：

**投资想法的生成 → 相关股票的筛选 → 研究获得结论 → 给投资决策反馈 → 监控和修正**

目前看可以最大程度实现自动化的是股票筛选（需要全面数据）以及监控和修正，Idea生成和研究AI仍然只是一个辅助的角色，给领导交流反馈这种人和人的交互可能会是研究员最坚固的壁垒所在。

## 三、为结果负责的人

目前AI的通用能力可能已经达到了90分，但是在具体的场景里，可能需要99.9999分才能够真正的创造价值，例如工业场景，肯定不能靠通用模型去解决，必须针对各个场景去做优化，把欠缺的9.99分补上，这就需要特定场景的RL。

对结果负责的人包含了两层含义：

**第一层：做端到端交付的人。** AI会给每个人做任何事情可能90分的基础知识，所以我们需要努力去做一个能端到端交付的人，例如前面提到的Palona AI，他们的工程师-产品经理-销售的角色就已经三合一了，任何一个结果里面需要不同的人去沟通协作的模式，都会损失大量的效率。人和AI的沟通简单直接，不包含任何情绪，无论是你让AI改，还是AI让你改，你和AI都不会不爽。但如果是人和人之间不断的反馈和修改，次数多了以后，大家会是什么情绪，大家懂得都懂。你需要对AI的结果负责，快速学习各个环节的基础知识，帮助ai把整个端到端的交付结果做好。

**第二层：AI会出错，需要有责任人。** AI目前只有90分，所以他还是会出错的，一旦出错，就需要有个责任人，而不是责任AI...所以请你肩负起作为人的责任吧！

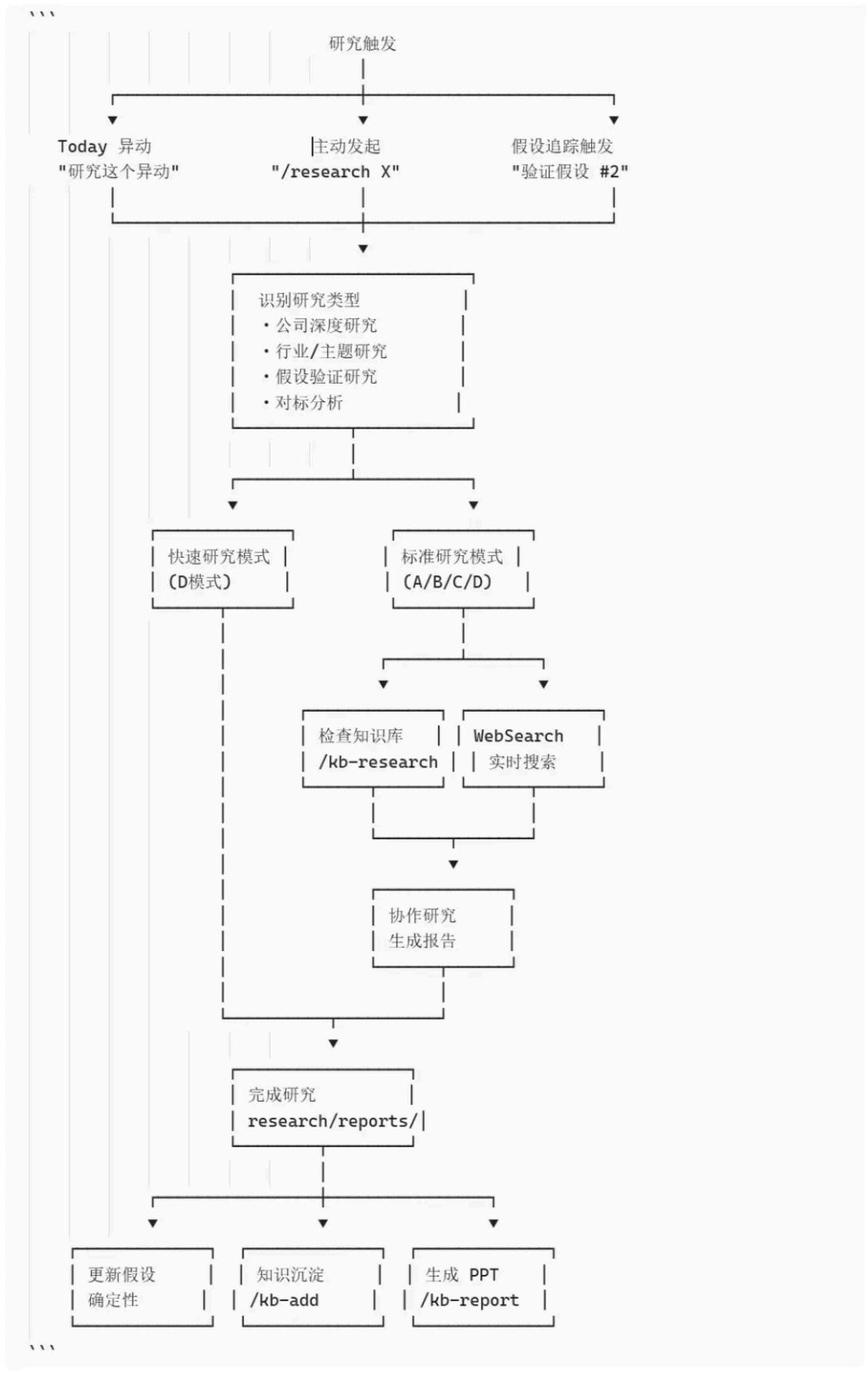
---

## 案例一：研究协作系统

研究本身是循序渐进的，没有人可以从一开始就得到最终的结论，都需要经过：核心假设逻辑  
→ 资料收集 → 思考 → 确认大方向没问题 → 深入研究 → 进一步收集数据 → 确认细节同样支撑 → 提高结论的conviction → 最后是股票的选择和价格时机的选择。

以下是claude code帮我设计的研究核心流程

## 核心流程：研究到知识沉淀



公众号 · 奔波儿r

### 以前用ChatBot的痛点

如果你用ChatBot的deepsearch，例如我在研究AI硬件架构的演变的时候，可能我让他写了十几篇的深度报告，最后仍然需要我去全部阅读，并且把要点抽离出来，然后整合成我的报

告。

## Claude Code的区别：迭代式研究

但是claude code，他的研究报告是迭代式的，可以先生成一篇快速研究，然后结合最新收集到的资料，包括你喂给他的一些fact，在同一篇报告文档内，修正之前的结论或者提高原来结论的conviction。这个文档完全对应的是你对这个方向的研究不断迭代的过程，而不只是给你提供一篇AI写的深度报告作为参考文献。

## 本地部署的优势

其次是因为研究协作系统是本地部署，他可以随时读取我过去几十篇的报告，引用里面的结论和观点，同时自动总结我的写作风格和关注要点，保证重点不漂移。

同时我还要求他在最后作为反方，来挑战我的观点，帮助我一定程度上克服主观偏差，因为有时候越研究自己的观点可能越狭隘，反而会忽视了一些反方提出的有道理的观点。

可以在本地的文件里面反复地去讨论，而且不用每次给他上传新的知识库。

---

## 案例二：其他日常场景—例如报销

前几天我在弄我的出差报销，需要用excel初步总结每个发票的金额，类型，备注（上海-北京机票）等等。我觉得这个事情应该完全可以自动化，我就让claude code读取我过去报销的文件夹，里面包含了高铁票、酒店、飞机、打车的各个发票，和原先写好的excel文档。这次让他按照过去的格式帮我生成这次报销的excel，结果他不仅分类和金额都写对了，还把备注都填好了，例如亚朵酒店写成了星朵，因为开票的主体叫做星朵集团。

我想这个功能我可以直接打包然后让公司内的人都用上，大家以后不用自己填，只要生成了以后核对一下就够了。

于是我让他把这个功能封装成一个exe，直接在放了发票的文件夹内双击后就自动生成本次报销的excel。我没有任何的编程基础，曾经尝试两次学习python，但就停留在了print("Hello world!!!")的阶段。但这次我十多分钟就完成了一个发票提取助手的exe程序。

**你在工作和生活中碰到的任何你觉得“应该”有一个程序去自动化实现的，你都可以去尝试一下，用它写出来的速度可能比你搜索、下载再应用的速度更快。而且这个程序还是根据你的要求，完全定制化实现的。我到现在就写了好几个，包括pdf文件转Markdown， claude code的API一键切换，发票的提取等等。**

---

## 使用原则

前面已经提到了AI需要的三个能力的拆解，这里我再补充几点。

**最核心的原则就是：未来需要完成任何一件事情的时候，先问问AI，想想AI怎么能够帮你。多用+快速迭代，只有这样才能够提高你在AI时代的竞争力。**

**记忆是当前的瓶颈**，目前Claude code的一个瓶颈在于Context的长度，一次对话只有20万的tokens，一旦接近这个上限，他就会忘记你设置的原则，以及之前的聊天内容。所以尽可能地少说废话，也让你的AI少说废话。

---

## 如何开始

那么最后，如何能够使用claude code呢。本质是需要基于你自己的工作习惯和流程，构建一整套协作系统。

### 基础步骤

**第一步：安装。** 参考这篇教程：<http://mp.weixin.qq.com/s/pEPweAhpzZb1ef5oMvaDDA>

**第二步：部署API。** 请参见某鱼，有各种API任君挑选，强烈建议用最先进的模型opus4.5，体验差别很大。

**第三步：授权。** 启动的时候可以用claude --dangerously-skip-permissions，让claude做任何操作的时候都自己进行，不需要你每次来确认。只要把Claude code的权限设置在你们的协作文件夹就不会太大风险，毕竟咱不是Coding，也不怕他删你代码。

**第四步：规划系统。** 打开Claude code的界面以后，切换到Plan mode (Shift+TAB按两次) 请详细地把自己的工作内容和工作流程告诉claude，然后非常谦虚地问他，你能够基于我的工作内容和工作流帮我设计一些和你的协作系统吗？

**第五步：交互磨合。** 开始和Claude 进行交互和磨合，有事没事都问他，然后结束每次对话的时候，都要他反思一下这次交流的收获，把它记录到项目文件的更新日志里面。

### 理解记忆分层

注意Claude的记忆是分层的：

层级	文件	内容	长度建议
全局	CLAUDE.md	交互原则、元规则	短，每次都读
项目	xxx-config.md	场景配置、工作流	可以详细
动态	假设、笔记等	具体内容	不限

简单来说：CLAUDE.md 是最高记忆，每次打开都读，所以不要太长；xxx-config.md 是项目配置，触发关键词时读取，可以详细；具体的假设、笔记可以更长。

### 进阶用法

- **Skill**: Claude code还有很多高级的用法，大家可以自己搜索。我自己常用的一个是Skill，可以把你要他做的一些流程创建一个Skill，简单来说这个是封装好的工作流程，可以节省token以及context的空间，在skills.sh网站也可以去套用别人封装好的skill。
- **底层记忆和画像设置**: 设置本地知识库，把原来的报告都从PDF转化成Markdown模式，然后喂给他，让他总结你的投资画像和特征，录入到他的底层记忆里面。
- 手机使用claude code: 远程链接到电脑的claude code，但不建议，会让你在哪都想掏出用一下，严重影响生活。大家可以参考[手机上也能用Claude Code和Codex! 很方便，一键搞定](#)

## 学习资源

可以在B站上搜索"完整教程：50分钟用Claude Code打造你的个人操作系统 | Teresa Torres"，可以直接学到作为一个产品经理+创作者是如何使用的，这个和研究员的工作有异曲同工之处，他们交付的产品，我们交付的是可能可以赚钱的股票。

另外推荐几个很好的学习使用claude code的公众号：

- **刘小排** (Claude token消耗第一人)
- **花叔** (不会编程的产品经理，开发的APP登顶过app store，分享实例)

未来如果有需要，我也会继续分享一些更加具体的AI用法，也欢迎大家在评论区分享自己在用的AI工具和工作流。或者在后台私信我



奔波儿儿

股票研究员一枚，喜欢到处乱看、瞎琢磨。分享关于研究和生活的感想。最近高强度学... >

1篇原创内容

公众号

## 感想

用上Claude code的这一个月里面，虽然工作效率显著提升，但是工作时长却也显著提升了。每天早上6-7点，睁开眼睛脑子里就想着“我应该怎么样去更好地用Claude code”，非常兴奋。

这有点像在网游里面，你突然爆了一把极品装备，原来要打60分钟的boss，现在只需要10分钟就能打完。有两个选择摆在你面前，10分钟打完boss然后关掉游戏，或者花100分钟暴打十个相似甚至更强的boss。玩过游戏的人都会选择后者，暴打曾经虐你千百遍的boss，这感觉不要太爽。

但游戏终归是游戏，还是不能够太沉迷了。工作其实说到底也就是另一种游戏而已，也切忌沉迷或者被迫“沉迷”。

希望大家早日多用AI提升效率，最后实现马斯克所说的大部分人都不需要工作的世界。更好地去用自己的身体和心去体验这个世界，毕竟亲身体验的快乐是AI无法替代的。

---

写于 2026 年 1 月，和 Claude 协作完成



奔波儿r

喜欢作者