PRODUCT FUNCTION ANALYSIS REPORT

主流聊天 AI 等待态 / 思考态 / 生成态

产品功能分析报告 v1.0

等待反馈 · 思考展示 · 工具调用 · 流式生成 · PRD草案 · 风险治理

产品类型： AI 聊天产品 / AI Agent / AI 工作台功能模块

文档用途： 产品分析报告 / 竞品分析 / PRD 前置分析 / 作品集项目资料

版本日期： 2026年6月

报告信息

项目	内容
报告标题	主流聊天 AI 等待态 / 思考态 / 生成态产品功能分析报告
研究对象	ChatGPT、Gemini、Claude、DeepSeek、通义千问 / Qwen、豆包
研究范围	等待态、思考态、推理展示、流式生成、工具调用、任务进度、异常状态、最终答案态
不研究范围	模型能力完整排名、训练数据、公司整体战略、价格体系全量对比
报告定位	产品功能专项分析 + 竞品分析 + 交互设计拆解 + 商业价值分析 + PRD 草案
适用读者	AI 产品经理、交互设计师、Agent 产品团队、模型产品经理、AI 应用创业者、作品集读者
方法	产品体验观察、官方资料研究、用户心理分析、功能拆解、指标设计、风险评估、PRD 化
核心结论	AI 等待态不是 loading，而是 AI 产品的认知交互层和任务过程可见层

重要说明

本报告从 AI 产品经理与交互体验分析视角出发，研究聊天 AI 在用户发送消息后如何展示等待、思考、推理、工具调用、流式生成与最终答案。

本报告不是模型能力排行榜，也不以“哪个模型最聪明”为核心判断标准。本报告关注的是：AI 产品如何把不可见的推理、检索、工具调用与任务执行过程，转化为用户可理解、可控制、可信任的产品体验。

目录

第 0 章：报告封面与基本信息

0.1 报告标题

《主流聊天 AI 等待态 / 思考态 / 生成态产品功能分析报告》

副标题：**从 ChatGPT、Gemini、Claude、DeepSeek、通义千问 / Qwen、豆包六个主流聊天 AI 出发，分析用户发送消息后，AI 产品如何展示等待、思考、生成、推理、工具调用与最终答案。**

0.2 研究对象

本报告研究六个主流聊天 AI 产品及其相关模型能力：ChatGPT / OpenAI、Gemini / Google、Claude / Anthropic、DeepSeek、通义千问 / Qwen / 阿里云百炼、豆包 / Doubao / 字节跳动 Seed / 火山引擎。

本报告关注的不是这些产品“谁更聪明”，也不是对模型综合能力做横评，而是聚焦一个非常具体的产品功能：**当用户点击发送后，在 AI 尚未完成回答之前，产品如何向用户展示状态、过程、进度、推理、工具调用与控制权。**

0.3 研究时间口径

本报告以 2026 年 6 月前后公开可见的官方资料、产品文档、产品体验和模型能力说明为主要依据。需要强调的是，聊天 AI 产品的等待态、思考态和生成态经常随模型版本、会员等级、地区、Web / App 端、实验开关、深度思考模式、工具调用能力变化而调整。因此本报告不会把某个界面状态视为永久固定事实，而是重点分析其背后的产品功能逻辑与行业趋势。

0.4 报告适用对象

本报告适合以下角色阅读：AI 产品经理、交互设计师、UX 设计师、模型产品经理、Agent 产品团队、AI 应用创业团队、AI 办公与企业级产品团队，以及希望把 AI 产品功能分析放入作品集的人。

0.5 核心研究问题

本报告试图回答一个核心问题：**聊天 AI 在回复前的等待过程，到底是不是一个值得独立研究和设计的产品功能？**

答案是：是，而且越来越重要。传统软件里的等待态，通常只是 loading、spinner、进度条或 skeleton screen。它的作用是告诉用户“系统正在加载”。但聊天 AI 的等待态更复杂，因为用户等待的不是一个静态页面，而是一个看起来会思考、会判断、会调用工具、会生成内容的智能系统。

因此，AI 产品的等待态已经不只是“加载中”，而是一个完整的认知交互层：它负责告诉用户 AI 是否收到问题、是否真的在处理、正在思考还是搜索、为什么需要这么久、是否可以中断、是否可以切换模式、是否可以查看过程、最终答案是否值得相信。

0.6 核心术语

术语	含义
等待态 Waiting State	用户发送消息后，AI 尚未开始输出正式内容前的状态反馈
思考态 Thinking State	模型进行推理、计划、拆解、验证时，产品向用户展示的状态
推理展示 Reasoning Display	产品是否展示模型推理过程、思考摘要、CoT 或 reasoning 内容
生成态 Generation State	答案以逐字、逐句、逐段方式流式输出的过程
工具调用态 Tool-Calling State	AI 调用搜索、文件、代码、图片、网页、日历、邮件等工具时的展示状态
最终答案态 Final Answer State	AI 完成生成后，用户看到的最终答案、引用、操作按钮和追问入口
Thinking Budget	模型或产品分配给推理过程的 token、时间或算力预算
Reasoning Content	部分模型 API 中与最终答案分离的推理内容字段
Thought Summary	对模型原始思考过程的摘要式展示，而非完整原始思维链

第 1 章：执行摘要

1.1 核心结论

聊天 AI 的等待态，正在从一个简单的“加载中”状态，演变成 AI 产品体验中非常关键的功能模块。传统互联网产品中的等待态主要解决系统反馈问题：用户点击后，系统告诉用户“我正在处理”。但在聊天 AI 产品中，等待态还承担了更多任务：降低等待焦虑、解释响应延迟、建立用户信任、展示模型能力、提供用户控制权、区分产品品牌心智、承接深度思考、联网搜索、文件分析、工具调用与 Agent 执行。

换句话说，聊天 AI 的等待态已经不只是 UI 细节，而是 AI 产品的“第二答案”。最终答案告诉用户结果是什么；等待态告诉用户 AI 是如何进入结果的。

1.2 从 Loading 到 Thinking UX

过去的软件等待态大多是转圈、进度条、骨架屏、“正在加载”“请稍候”。这些交互的核心目标是减少用户不确定感。AI 的等待态不只是内容没加载出来，而是 AI 是否理解了问题、是否在推理、是否在搜索、是否在读取文件、是否在调用工具、是否在检查答案、是否需要更长时间给出更好结果。

因此，AI 等待态从传统的 Loading UX 变成新的 Thinking UX。这个变化是本报告最重要的理论起点。

1.3 六个产品的总体路线

当前六个主流产品大致形成四条路线：

路线	代表产品	核心理念	用户感受
答案优先型	ChatGPT、Gemini 普通模式、豆包普通模式	用户要的是最终结果，不应被过程打扰	干净、快、少干扰
摘要透明型	Claude、Gemini thought summaries、部分 ChatGPT reasoning summary	可以让用户看过程，但要经过摘要和折叠	专业、克制、可信
过程展示型	DeepSeek	把思考过程直接变成产品体验的一部分	透明、强推理感、过程可观看
模式切换型	Qwen / 通义千问、Claude、ChatGPT Thinking	简单任务快答，复杂任务深想	可控、灵活、工程化

1.4 为什么这个功能变重要

这个功能变重要，不是因为 UI 设计师突然重视 loading，而是因为大模型产品本身发生了变化。推理模型让等待时间变长，AI 从聊天走向任务执行，用户对 AI 的信任要求提高，深度思考开始成为付费能力，Agent 任务需要更强的过程展示。

1.5 本报告核心判断

AI 等待态的本质，是用户与模型之间的信任缓冲层。它连接模型能力、产品表达、用户心理和商业价值。如果等待态设计得好，用户会觉得“它确实在认真处理我的问题”；如果等待态设计得不好，用户会觉得“它是不是卡了”“它为什么这么慢”“它是不是在胡说”。

核心判断：聊天 AI 的等待态 / 思考态 / 生成态，是 AI 产品从“会回答”走向“可信任地完成任务”的关键功能。

第 2 章：研究范围、样本与分析方法

2.1 研究范围

本报告研究用户发送消息后，AI 回复完成前后的完整状态链路，包括发送反馈、等待反馈、思考状态、推理过程展示、工具调用状态、流式生成、中断控制、异常处理、最终答案展示、答案后的操作入口。

换言之，本报告关注的是聊天 AI 从“用户发送”到“答案完成”的完整体验，而不是单独看最后输出内容。

2.2 不研究范围

本报告不重点研究六个 AI 产品的整体模型能力排名、订阅价格体系、训练数据来源、参数规模、公司整体战略、多模态能力完整评测、App 内所有功能模块、API 性价比全量对比。这些内容都重要，但不属于本报告的核心。

2.3 样本选择理由

产品	代表意义
ChatGPT	全球 AI 聊天产品标杆，代表答案优先、工具增强、工作台型体验
Gemini	Google 生态型 AI 助手，代表多模态与搜索生态结合
Claude	专业用户和长文本场景强势，代表折叠式透明与 extended thinking
DeepSeek	强思考展示心智突出，代表过程展示型 reasoning 产品体验
Qwen / 通义千问	中国大模型重要代表，工程化 thinking / non-thinking 切换能力突出
豆包	大众化 AI 助手代表，普通端轻量，模型侧具备深度思考能力

2.4 研究方法

本报告使用四类方法：产品体验观察、官方文档研究、用户心理与 UX 理论分析、产品策略推断。

产品体验观察关注六个产品在用户发送消息后的可见状态：是否立即反馈、是否显示正在思考、是否流式输出、是否展示推理过程、是否支持展开 / 折叠、是否显示工具调用、是否允许停止生成、是否区分最终答案与过程内容。

官方文档研究用于确认模型是否具备 reasoning / thinking 能力、是否有 reasoning tokens / reasoning_content / thought summaries、是否支持 thinking budget / thinking effort、是否将推理内容与最终答案分离、是否支持开发者控制深度思考模式。

用户心理与 UX 理论用于分析等待、进度反馈、感知等待时间、系统可见性、控制感和信任感。产品策略推断用于分析 ChatGPT 为什么不默认展示完整思考、DeepSeek 为什么强化思考过程、Claude 为什么选择折叠透明、Qwen 为什么强调 thinking / non-thinking 切换、豆包为什么普通端更轻量。

2.5 信息可靠性分级

等级	类型	说明
A 级	官方事实	来自官方文档、模型说明、开发者 API、公司技术报告
B 级	产品体验观察	来自用户端可见体验，但可能随地区、版本、会员等级变化
C 级	合理推断	基于产品体验和官方能力做出的功能逻辑推断
D 级	战略判断	面向未来趋势和产品路线的分析判断

第 3 章：功能定义

3.1 什么是 AI 等待态

AI 等待态指用户发送消息后，在 AI 正式输出可阅读答案之前，产品向用户展示的所有状态反馈。传统产品中的等待态可能只是 loading，但 AI 产品中的等待态至少包括：消息发送成功、AI 已接收请求、AI 正在理解问题、正在思考、正在搜索、正在读取文件、正在调用工具、正在组织答案、即将开始输出。

AI 等待态的核心价值是消除用户对系统是否工作的不确定感。如果用户点击发送后什么都没有发生，哪怕只过了 2 秒，用户也可能怀疑系统卡住。等待态的第一使命，就是让用户知道请求已经被接收，AI 正在处理。

3.2 什么是 AI 思考态

AI 思考态指模型在正式回答之前，进行内部推理、任务拆解、方案比较、假设验证、工具规划时，产品向用户展示的状态。它可以分成三种形式：完全隐藏、摘要展示、过程展示。

完全隐藏意味着用户看不到任何思考过程，只看到最终答案，代表 ChatGPT 普通体验、Gemini 普通模式、豆包普通模式的许多场景。摘要展示意味着用户看到的是整理过的思考摘要，而不是完整原始思维链，代表 Gemini thought summaries、Claude summarized thinking。过程展示意味着用户能看到较长推理过程，甚至接近 chain-of-thought 的内容，代表 DeepSeek thinking mode、Qwen deep thinking、部分豆包深度推理模型能力侧。

3.3 什么是推理展示

推理展示不是简单地把 AI 的所有思考都贴给用户，而是一种产品化后的解释层。它可以分为五个层级：不展示、状态提示、思考摘要、详细过程、原始推理 / CoT。

层级	展示形式	用户价值	风险
L0	不展示	干净、高效	复杂任务不透明
L1	状态提示	告诉用户正在做什么	信息量有限
L2	思考摘要	解释推理方向	可能过度简化
L3	详细过程	可审查、可学习	过长、可能误导
L4	原始推理 / CoT	最大透明感	忠实性、安全、隐私、认知负担问题

成熟产品不应该简单追求“展示越多越好”。更成熟的方式是展示有用、经过组织、可审查的推理摘要，而不是把所有过程无差别暴露。

3.4 什么是生成态

生成态指 AI 开始输出内容后的可见生成过程。主流聊天 AI 几乎都采用流式输出：逐字、逐句、逐段出现，先给部分内容，再继续补全，用户可以在生成中阅读、判断和停止。流式输出的价值不是单纯加快模型速度，而是降低用户的感知等待时间。

3.5 什么是工具调用态

工具调用态包括正在搜索网页、正在打开网页、正在读取 PDF、正在分析表格、正在运行代码、正在查看图片、正在生成图片、正在查询日历、正在整理引用、正在重试失败工具等。

随着 AI 从聊天问答走向 Agent 和工作流，工具调用态变得越来越重要。它把用户看不见的 AI 后台行为，转化为用户可以理解的任务进度。

3.6 什么是最终答案态

最终答案态是 AI 完成回答后的稳定状态，通常包括最终答案正文、引用来源、思考过程折叠区、复制按钮、重新生成按钮、点赞 / 点踩、继续追问建议、导出 / 分享、编辑 / 改写、继续执行下一步。

成熟 AI 产品中，最终答案应该和思考过程分离。用户真正要使用的是最终答案，而不是所有推理草稿。

3.7 为什么这是认知交互层

传统软件界面主要展示数据、按钮、页面和操作结果。AI 产品界面还需要展示一种新的东西：模型的认知状态。用户会关心 AI 是否理解我、是否正在推理、是否知道自己不确定、是否在查资料、是否在验证、是否能解释为什么这样回答、是否能让我控制它思考多久。

这就是本报告把等待态 / 思考态 / 生成态定义为 AI 产品认知交互层的原因。它不是简单 UI 组件，而是用户理解 AI、控制 AI、信任 AI 的关键入口。

第 4 章：用户场景分析

4.1 普通问答场景

普通问答包括“今天适合穿什么”“这个词是什么意思”“解释一个概念”“推荐几个电影”“这句话怎么翻译”等。用户最关心快、简洁、直接、不要废话、不要长时间思考。

因此，普通问答不适合默认展示大量思考过程。ChatGPT、Gemini、豆包普通模式的轻量等待态，更适合这类场景。简单问题如果先输出 500 字思考，体验反而会变差。

4.2 复杂推理场景

复杂推理包括数学题、逻辑题、代码调试、商业判断、产品策略、竞品分析、多变量决策、长链路规划。这类任务中，用户不只是要答案，还需要知道 AI 为什么这样判断、有没有遗漏关键因素、有没有误解问题、推理链条是否成立、结论是否可复查。

复杂推理场景更适合展示思考摘要、关键假设、分析框架、步骤，并支持展开详细过程和重新深度思考。Claude、DeepSeek、Qwen 在这类场景中更容易形成“它确实在认真分析”的用户感知。

4.3 搜索研究场景

搜索研究场景包括最新新闻、行业资料、公司信息、产品资料、学术论文、政策法规、价格规格、版本发布时间等动态信息。用户不仅关心答案，还关心 AI 是否真的搜索了、搜索了哪些来源、来源是否可靠、有没有只看单一来源、有没有引用过期内容、有没有把推断当事实。

搜索研究类任务的等待态必须展示正在搜索、正在打开来源、正在比较资料、正在整理引用、正在核对时间、正在形成结论。

4.4 文件分析场景

文件分析包括 PDF 阅读、简历修改、合同分析、表格分析、报告总结、论文精读、产品文档整理、多文件对比。用户最常见的不安是：它真的读了我的文件吗？

文件分析等待态应该尽量具体：已接收文件、正在解析文件、正在读取第几部分、正在提取关键结构、正在整理摘要、正在对比多个文件、文件有无读取失败、哪些内容可能无法识别。

4.5 代码与开发场景

代码任务包括写代码、修 bug、解释报错、重构项目、阅读仓库、运行测试、部署排查、生成技术文档。这个场景非常依赖过程透明。用户需要知道 AI 是否看到了报错、是否定位到文件、是否理解依赖关系、是否运行测试、是否修改了哪些文件、是否引入新问题。

代码类 AI 产品的等待态应更接近执行日志：正在扫描项目结构、正在定位错误文件、正在分析调用链、正在修改代码、正在运行测试、测试失败正在重试、修改完成生成 diff。

4.6 写作与创作场景

写作场景包括文案生成、文章改写、简历优化、报告撰写、视频脚本、产品 PRD、演讲稿、社交媒体内容。等待态不一定要展示复杂推理，但可以展示创作方向：正在整理结构、正在生成初稿、正在优化语气、正在调整风格、正在压缩字数、正在补充案例、正在统一格式。

写作场景的关键不是展示逻辑推理，而是给用户一种内容正在成形的方向感。

4.7 Agent 多步骤任务场景

Agent 不只是回答问题，而是执行任务，例如整理这周邮件、做行业研究、分析竞品并生成报告、修改项目代码并部署、规划旅行并预订、生成视频脚本和分镜、把文件整理成作品集页面。

在 Agent 场景中，等待态必须升级为任务执行面板，展示任务计划、当前步骤、已完成步骤、失败步骤、需要用户确认的节点、最终交付物、中间产物、可回滚操作、可暂停 / 继续 / 终止。

4.8 不同用户群体差异

普通用户更关心快、简单、直接；学生用户既需要答案，也需要过程；专业用户需要假设、依据、推理链、资料来源、不确定性和风险提示；开发者需要工具调用日志、文件路径、错误信息、测试结果和 diff；企业用户关注可审计、可回溯、可控、权限、合规和数据安全。

第 5 章：用户心理分析

5.1 用户等待时的核心心理问题

用户等待 AI 回复时，内心并不是空白的。他会不断判断：系统是否卡住、AI 是否理解问题、是否值得继续等、输出会不会有用、能不能中断、它是不是在乱编、为什么不给我看过程。

等待态真正管理的是用户的不确定感、失控感和不信任感。

5.2 “它卡了吗”：系统存活感

这是最基础的问题。用户发送消息后，如果 1-2 秒内没有任何反馈，就会怀疑网络断了、消息没发出去、模型崩了或产品卡住了。AI 产品必须做到发送后立即反馈、明确进入生成中状态、按钮状态变化、显示停止生成、必要时显示“正在处理”。

5.3 “它真的在想吗”：能力感知

聊天 AI 的特殊之处在于，用户不是在等一个页面，而是在等一个智能体。因此用户会把等待过程理解为：它正在想、正在查、正在组织语言、正在分析、正在判断。

DeepSeek 的思考展示很容易形成强心智：用户看到大段推理，会更容易产生“它认真想过”的感受。但这里有风险：看起来在思考，不等于真的更可靠。外显 reasoning 文本只能作为辅助判断，不应该被当作绝对真实的内部机制。

5.4 “它为什么这么慢”：延迟解释

用户可以接受慢，但需要知道为什么慢。如果产品只是沉默，用户会觉得慢；如果产品告诉用户正在搜索资料、分析文件、运行代码，用户会觉得这是合理等待。

等待态的本质不是消灭等待，而是解释等待。

5.5 “我能不能控制它”：控制感

用户最怕失控：AI 一直生成停不下来、思考过程太长、答案偏离方向、想中断找不到按钮、想直接看结论却被迫看过程、想要更深入却只能得到短答案。

因此等待态必须包含控制权：停止生成、重新生成、继续回答、直接给结论、展开思考、隐藏思考、切换深度思考、切换快速模式、缩短回答、只输出最终答案。

5.6 “我要不要看过程”：认知负担

展示思考有价值，但也有成本。对于普通用户来说，大段思考过程可能看不懂、看不完、不知道重点在哪里。更成熟的设计应该是默认简洁、复杂任务给摘要、专业用户可展开、简单任务不展示、最终答案独立清晰。

5.7 等待态如何影响用户对模型能力的判断

用户对 AI 能力的判断，不完全来自最终答案，也来自生成过程。同样答案，一个模型沉默 20 秒后突然输出，另一个模型先显示正在搜索资料，再逐步生成结构化结论，用户更容易相信后者。

等待态设计	用户感知
没反馈	卡顿、不可靠
简单转圈	正在处理，但不知道做什么
流式输出	有进度，能继续等
状态文案	知道当前任务
思考摘要	感觉 AI 有逻辑
工具调用展示	感觉 AI 真正在执行
最终答案 + 来源	感觉答案可检查
可中断控制	感觉自己掌握主动权

5.8 透明感与误导感的双刃剑

展示思考过程有好处：提升信任、帮助学习、方便检查、增强专业感、解释等待、形成差异化。但也有风险：错误推理被用户误信、过程太长造成疲劳、模型自相矛盾、用户误以为这是“真实内心”、暴露不必要中间内容、简单问题复杂化、增加成本与延迟。

本报告的基本立场是：AI 产品不应该追求绝对透明，而应该追求有效透明。有效透明是让用户看到足够帮助判断和控制的信息，但不把未经组织的全部过程强行灌给用户。

第 6 章：行业背景

6.1 大模型产品正在从快答进入慢想

早期聊天 AI 的主要体验是用户问一句，模型马上开始回答。这种体验像一个会说话的搜索框或自动写作助手。但从推理模型兴起之后，行业明显进入“先思考、再回答”的阶段。

OpenAI reasoning models、Claude extended / adaptive thinking、Gemini thinking、DeepSeek-R1、Qwen deep thinking、Doubao Seed-Thinking 等模型路线，都在强化 reasoning / thinking 能力。这说明行业从快速生成答案进入根据任务分配推理时间的阶段。

6.2 Test-time Compute 成为产品体验问题

过去模型能力主要来自训练阶段：更大模型、更好数据、更强训练。现在，越来越多推理能力来自回答阶段的额外计算，也就是 test-time compute。模型在回答前花更多 token、时间和算力去推理、反思、验证。

这会带来更好的复杂任务表现，但也带来产品问题：等待更久、成本更高、输出更长、用户更焦虑、需要更强状态解释、需要模式切换、需要 thinking budget 控制。

6.3 思考展示成为产品差异化

当模型能力越来越接近，前端体验会成为重要差异。DeepSeek 的强体验点是思考过程可见；Claude 的差异化是透明但克制；Qwen 的差异化是模式可切换；ChatGPT 的差异化是成熟工作台；豆包的差异化是大众轻量。

6.4 从聊天窗口到 Agent 执行面板

今天多数用户仍把 AI 看成聊天窗口，但产品演化方向正在变成 Agent 执行系统。聊天窗口适合问答、写作、总结、翻译、轻量建议；Agent 面板适合多步骤任务、文件处理、项目修改、数据分析、资料研究、自动化工作流、代码开发、商业分析、内容生产。

当 AI 开始做多步骤任务，传统聊天气泡就不够了。用户需要看到任务目标、任务计划、当前步骤、工具调用、中间结果、是否失败、是否需要确认、最终交付物。

6.5 从模型黑箱到过程可审查

企业级 AI 场景尤其需要过程可审查。企业用户不仅关心结果，还关心数据是否被正确使用、来源是否可靠、过程是否可追溯、工具调用是否合规、是否误操作、是否泄露信息、是否能复盘、是否能审计。

未来企业 AI 产品的等待态，本质上会和审计链路结合。

6.6 从看见思考到控制思考

行业正在经历三个阶段：隐藏思考、展示思考、控制思考。未来用户或系统可以控制是否思考、思考多久、思考多深、是否展示、展示摘要还是完整过程、什么时候调用工具、什么时候直接回答、什么时候需要人工确认。

6.7 中国 AI 产品与海外 AI 产品的等待态差异

海外主流产品中，ChatGPT 更偏成熟工作台，Claude 更偏专业透明，Gemini 更偏生态助手。中国产品中，DeepSeek 更强调思考过程可见，Qwen 更强调 thinking / non-thinking 的工程化切换，豆包更偏大众助手。

这不是绝对划分，而是当前产品表达上的倾向差异。

6.8 行业正在形成的新共识

AI 不是所有问题都应该用同一种方式回答。简单问题需要快，复杂问题需要深，专业问题需要依据，搜索问题需要来源，文件问题需要读取过程，代码问题需要执行日志，Agent 任务需要步骤面板，高风险问题需要不确定性提示。

第一阶段核心结论：AI 等待态正在从通用 loading 走向任务感知型状态系统。

第 7 章：六个产品总体对比

7.1 总体结论

六个主流聊天 AI 在等待态上的差异，本质上不是 UI 风格差异，而是产品哲学差异。

路线	代表产品	核心理念	用户感受
答案优先型	ChatGPT、Gemini 普通模式、豆包普通模式	用户要的是最终结果，不应被过程打扰	干净、快、少干扰
摘要透明型	Claude、Gemini Thinking、部分 ChatGPT reasoning summary	可以让用户看过程，但要经过摘要和折叠	专业、克制、可信
过程展示型	DeepSeek	把思考过程直接变成产品体验的一部分	透明、强推理感、过程可观看
模式切换型	Qwen / 通义千问、Claude、ChatGPT Thinking	简单任务快答，复杂任务深想	可控、灵活、工程化

7.2 六个产品在等待态上的总体定位

产品	等待态定位	思考展示强度	过程控制能力	适合用户
ChatGPT	成熟工作台型	中低	高	普通用户、专业办公、开发者、研究者
Gemini	生态助手型	中	中高	Google 生态用户、多模态用户、普通用户
Claude	专业透明型	中高	高	专业用户、写作、代码、研究、长文档
DeepSeek	过程展示型	很高	中	推理用户、学习用户、技术用户
Qwen / 通义千问	工程可切换型	高	很高	开发者、企业、Agent 产品、复杂任务用户
豆包	大众轻量型	消费端中低，能力侧较高	中	普通大众、内容创作、轻办公、移动端用户

7.3 关键差异不是有没有思考，而是如何产品化思考

现在几乎所有主流 AI 厂商都在强化 reasoning / thinking 能力。真正差异不是底层有没有推理，而是是否让用户看见思考，看见的是原始过程、摘要还是状态提示，用户能不能控制是否思考，能不能控制思考强度，思考过程和最终答案是否分离，工具调用过程是否可见，思考过程是否会增加用户负担。

ChatGPT 把复杂能力藏在干净工作台后面。Gemini 保持助手轻量感，同时在底层支持 thinking 能力。Claude 让专业用户可控地查看思考摘要。DeepSeek 把思考过程本身做成可见卖点。Qwen 把 thinking / non-thinking 做成工程化开关。豆包大众端轻量，深度能力后置到特定模型和开发者平台。

7.4 六个产品在回复过程链路上的强调差异

产品	等待反馈	思考展示	工具调用展示	流式生成	最终答案整理
ChatGPT	强	克制	强	强	强
Gemini	中强	中	中强	强	中强
Claude	强	强，偏折叠摘要	中强	强	强
DeepSeek	中	很强	中	强	中
Qwen	中	强，可切换	强，偏开发者能力	强	中强
豆包	强，轻量	普通端弱，模型侧强	中	强	中强

7.5 核心判断

六个产品中，长期更成熟的方向不是全部隐藏思考，也不是全部展示思考，而是默认轻量反馈、复杂任务自动增强、思考摘要可见、详细过程可展开、用户可控制思考强度。

第 8 章：ChatGPT 功能分析

8.1 产品定位

ChatGPT 是六个产品中最典型的成熟 AI 工作台。它的等待态设计不是为了让用户一直看 AI 怎么想，而是为了让用户尽快进入可用答案。

ChatGPT 的核心产品哲学是：把复杂模型能力、工具能力和推理能力隐藏在相对干净、稳定、低干扰的聊天界面背后。

8.2 用户端等待链路

ChatGPT 的典型链路是：用户发送消息，输入框进入生成中状态，出现等待 / 思考反馈，如需工具则显示搜索、分析、读取、生成等状态，答案开始流式输出，用户可停止生成，输出完成后出现复制、重试、追问、引用等操作。

ChatGPT 的强项不在于展示大段思考过程，而在于整体链路很顺：发送反馈及时、首 token 体验稳定、流式输出成熟、生成过程中可中断、工具调用状态相对清晰、最终答案结构通常干净。

8.3 思考展示策略

ChatGPT 的关键特点是：不默认展示完整原始思维链。模型可以深度推理，但用户不一定看到完整推理；产品更强调最终答案质量；必要时提供摘要式推理或工具状态；不把原始思维链作为默认用户界面内容。

这种设计对普通用户非常友好。用户不会被大段过程淹没，也不会在简单问题里看到过度复杂化的推理。

8.4 Thinking 模式与用户控制

ChatGPT 已经把“思考深度”产品化为用户可感知的控制项。不同 thinking time 或 reasoning effort 代表不同推理资源投入：更快、更平衡、更深、更适合高难任务。

这不是单纯选模型，而是在选推理资源投入级别。

8.5 工具调用等待态

ChatGPT 的另一个优势，是将等待态与工具能力结合得比较自然。在文件、数据、搜索、图片、生成图像等任务里，ChatGPT 能够把后台工具行为以较自然的状态反馈呈现出来。

当 AI 开始调用工具时，用户真正等待的不是语言生成，而是任务执行。ChatGPT 的产品路线正在从聊天机器人向 AI 工作台扩展，等待态也随之从简单 loading 变成任务状态展示。

8.6 优势

ChatGPT 的优势包括低认知负担、答案主体清晰、工具状态成熟、控制能力逐渐增强。普通用户不需要理解 reasoning tokens、thinking budget、CoT 等概念。发送问题后，直接等答案即可。

8.7 不足

ChatGPT 的不足在于复杂推理透明度不足、部分复杂任务等待过程仍偏黑箱、思考摘要能力还可以更前端化。专业用户有时希望看到“它到底为什么这样判断”。

8.8 产品判断

ChatGPT 的等待态是六个产品中最像成熟生产力工具的。它的路线不是让用户欣赏 AI 的推理表演，而是尽量减少过程打扰，把复杂能力沉淀到稳定、可控、可交付的工作流里。

第 9 章：Gemini 功能分析

9.1 产品定位

Gemini 的等待态更接近 Google 产品一贯的体验风格：轻量、自然、助手化。它不像 DeepSeek 那样强行把思考过程作为前台展示重点，也不像 Claude 那样把 extended thinking 作为专业用户心智核心。Gemini 的方向更像普通用户端保持轻量，复杂任务中通过 thinking、thought summaries、多模态和搜索生态增强能力。

9.2 用户端等待链路

Gemini 普通聊天中的等待链路通常比较轻：用户发送，出现简洁等待反馈，答案开始生成，如涉及搜索、多模态、工具，则显示相应处理状态，输出最终答案。

这种设计适合大众用户，尤其是搜索、图片理解、邮件办公、移动端问答等场景。用户通常不想看大段推理，只希望 Gemini 快速整合信息、给出结果。

9.3 Thinking 与 Thought Summaries

Gemini 并不是没有思考展示能力。Gemini API 支持 thought summaries，并将其定义为 raw thoughts 的 summarized versions，可以提供对模型内部推理过程的 insight。这说明 Gemini 的思考展示更接近摘要透明型，而不是 DeepSeek 那种强过程展示型。

9.4 等待态核心价值

Gemini 的等待态价值主要体现在多模态处理中的状态解释、搜索生态中的可信度、复杂任务中的可解释推理。尤其在图像、视频、文本、搜索和 Google 生态联动任务中，等待态需要告诉用户正在识别、理解、整合上下文和生成结论。

9.5 优势

Gemini 的优势是轻量感好、多模态一致性强、思考能力产品化空间大。Thought summaries、thinking levels、thinking budgets 给 Gemini 留出了很强前端设计空间。

9.6 不足

Gemini 的不足是等待态品牌记忆点不够强，专业任务中的过程感还可以更强，消费端与开发者能力之间存在表达落差。

9.7 产品判断

Gemini 的等待态路线是普通场景轻量化、复杂场景 thinking 化、多模态场景状态化。如果 Gemini 想强化专业用户心智，需要把 thought summaries 和复杂任务计划展示得更显性。

第 10 章：Claude 功能分析

10.1 产品定位

Claude 是六个产品中最典型的专业透明型等待态代表。它不像 ChatGPT 那样默认强克制，也不像 DeepSeek 那样大量展示推理文本，而是走了一条中间路线：用户可以让 Claude 深度思考，也可以展开查看思考摘要，但产品不会强迫所有用户一直看完整过程。

10.2 等待链路

Claude 的典型链路是：用户发送，Claude 进入等待 / 处理状态，如开启 extended thinking，出现 Thinking 指示和计时，思考区域可展开，输出最终答案，用户可查看思考摘要、修改模型 / effort / thinking 设置。

10.3 Extended Thinking

Claude 的关键能力是 Extended Thinking。当 extended thinking 开启时，用户可以看到 Thinking indicator 和计时器，并在回复上方看到可展开的 Thinking 区域。用户点击后可以查看 Claude 的 thought process summary 和 problem-solving approach。

这种交互思路非常成熟：不把思考混进正文，思考区独立于最终答案，用户需要时再展开，思考内容是 summary 和 approach，而不是无差别长草稿。

10.4 summarized thinking

Claude 的产品判断不是“所有推理都展示给用户才透明”，而是把模型思考过程转化成可阅读、可审查、可控的摘要，才是更成熟的透明。这比 DeepSeek 式大段输出更适合专业工作流，也比完全隐藏过程更容易建立信任。

10.5 用户控制能力

Claude 把控制权设计得比较明确：用户可以换模型、调 effort level、开启或关闭 thinking，thinking 与 effort 是两个独立设置。简单任务可以关闭 thinking，复杂任务可以开启 thinking 或提高 effort。

10.6 优势

Claude 的优势是透明度和简洁度平衡最好、专业用户友好、控制项清晰、符合复杂任务工作节奏。

10.7 不足

Claude 的不足是普通用户理解成本略高，过程展示仍以文本为主，工具调用状态还可以更任务面板化。

10.8 产品判断

Claude 是六个产品中最值得学习的“折叠透明型等待态”样板。它证明透明可以设计得很优雅，而不是只能粗暴展示。

第 11 章：DeepSeek 功能分析

11.1 产品定位

DeepSeek 是六个产品中最典型的过程展示型等待态代表。它的用户心智非常明确：AI 会先思考，而且用户能看到它思考。

这让 DeepSeek 在消费者认知中形成强烈差异化。很多用户第一次使用 DeepSeek-R1 或 deep thinking 模式时，印象最深的不是最后答案，而是前面那一大段可见推理过程。

11.2 等待链路

DeepSeek 的典型链路是：用户发送，进入思考状态，输出 reasoning / thinking 内容，思考结束，输出最终答案。支持流式输出的场景中，用户可以看到模型先输出 reasoning_content，再输出 content。

11.3 reasoning_content 设计

DeepSeek 的 reasoning_content 与 content 分离，是非常重要的产品结构基础。它使前端可以把思考区和答案区分开，让用户选择是否展示思考，并对思考内容做折叠、摘要、复制、隐藏或教学化处理。

11.4 核心体验优势

DeepSeek 的优势包括等待过程被内容化、信任感强、学习价值高、品牌记忆点极强。用户不再只是等答案，而是在看 AI 逐步拆解问题。

11.5 思考过载问题

DeepSeek 最大的问题也是它的优势反面：思考过程太重。简单问题可能显得复杂，阅读负担高，错误推理会暴露，用户可能误以为可见思考等于真实内心。

11.6 适用场景

DeepSeek 的等待态最适合数学推理、逻辑题、代码调试、学习辅导、商业分析、产品分析、多条件决策、需要过程复查的任务。不太适合默认用于简单翻译、普通闲聊、简单事实问答、快速改写、移动端轻问答、只想要一句结论的场景。

11.7 改进方向

DeepSeek 未来最应该增强的是思考分层：默认折叠长思考，先给一句摘要，自动分段，支持只看结论，支持展开详细思考，简单问题自动关闭深度思考，对思考过程做摘要，区分假设、推理、验证、结论。

11.8 产品判断

DeepSeek 证明等待态本身可以成为 AI 产品的核心卖点。但从长期产品成熟度看，DeepSeek 需要从“展示思考”升级到“管理思考”。

第 12 章：通义千问 / Qwen 功能分析

12.1 产品定位

通义千问 / Qwen 的等待态和 DeepSeek 不同。DeepSeek 更像把过程展示给用户看，Qwen 更像把 thinking 模式做成可控制的工程能力。

Qwen 的核心产品哲学是：复杂任务先思考，简单任务直接答；是否思考、如何思考，可以被参数和模式控制。

12.2 deep thinking 能力

Qwen deep thinking models 会在生成答案前进行推理，以提高逻辑推理和数值计算等复杂任务准确性。其 hybrid thinking mode 可通过 enable_thinking 参数启用或禁用 thinking；true 时模型先思考再回答，false 时模型直接回答。

12.3 reasoning_content 与 content 分离

Qwen API 会在 reasoning_content 字段返回 reasoning content，在 content 字段返回正式 response。深度思考会增加 latency，多数模型支持 streaming output。对前端而言，这天然支持“思考区 + 答案区”的信息架构。

12.4 Hybrid Thinking 产品意义

Qwen3 的 Hybrid Thinking 使 Thinking Mode 适合复杂问题，会一步步推理后给最终答案；Non-Thinking Mode 适合简单问题，可以快速近即时回复。这本质上是在做任务价值与推理资源的匹配。

12.5 用户端等待态特点

Qwen 消费端不一定总是强展示长过程，但在深度思考模式下会有更明显的思考展示。理想流程是系统判断任务复杂度，简单任务直接生成，复杂任务进入 deep thinking，展示思考状态或思考过程，再输出最终答案。

12.6 优势

Qwen 的优势是工程化控制清晰、适合企业级和开发者场景、最终答案和思考内容分离、适合多模型多场景路由。

12.7 不足

Qwen 的不足是消费者端表达还可以更产品化，思考展示需要更强的信息设计，需要避免模式选择负担。

12.8 产品判断

Qwen 是六个产品中最适合做“AI 推理模式系统”的代表之一。它的核心启发是未来 AI 产品不是只有一个回答按钮，而应该有任务感知的 reasoning mode。

第 13 章：豆包功能分析

13.1 产品定位

豆包首先是一个大众化 AI 助手，而不是纯专业工作台。它的普通用户端体验更强调轻量、快速、易懂、移动端友好、内容创作友好、日常陪伴和工具化，不让用户被复杂模型机制打扰。

13.2 普通用户端等待链路

豆包普通聊天的典型链路是用户发送、轻量等待反馈、直接生成答案、如涉及搜索、创作、图片或工具则显示相应处理状态、输出最终内容。

这种设计适合大众用户。大多数普通用户问豆包的问题可能是写一句文案、改一段话、查一个知识点、生成短视频脚本、生活建议、学习问题、聊天陪伴、图片或语音相关任务。

13.3 模型侧深度思考能力

豆包普通端轻量，并不代表字节体系没有深度思考能力。ByteDance Seed-Thinking、Doubao 多模态深度思考等资料均说明模型侧具备 reasoning / thinking 能力，并可在回答前拆解问题、执行逻辑推理、生成 reasoning_content。

13.4 为什么不适合默认强展示思考

豆包作为大众化产品，如果默认展示大段思考，会有明显风险：大众用户不一定需要过程，移动端空间有限，内容创作场景更重结果，过度思考会破坏轻量心智。

13.5 优势

豆包的优势是大众友好、轻交互适合高频场景、能力侧有深度思考储备、适合做场景化深度入口。

13.6 不足

豆包的不足是专业任务的可信感不足，深度思考入口需要更明显，模型能力与消费端心智之间存在距离，等待态还可以更任务化。

13.7 改进方向

豆包适合采用双层等待态：大众默认轻量，适合普通问答、生活建议、文案改写、简单创作；专业深度模式，适合复杂任务、学习、分析、研究、代码、多模态理解。

13.8 产品判断

豆包的等待态路线是普通端轻量化，专业能力后置化。这对大众市场合理，但如果未来要进入专业生产力工具或企业 Agent，需要强化深度思考入口、任务进度展示、思考摘要与最终答案分离。

第 14 章：功能模块拆解

14.1 功能系统总览

聊天 AI 的等待态系统，不能只理解为一个“正在生成”的动画。它是一套完整状态系统，贯穿用户发送消息后的整个过程。

完整链路是：用户输入、点击发送、系统接收反馈、任务识别、等待 / 思考 / 检索 / 工具调用、流式生成、中断 / 继续 / 切换模式、最终答案、引用 / 操作 / 追问 / 导出。

完整功能模块至少包括：基础反馈、首 token 反馈、流式生成、思考展示、模式切换、推理强度控制、工具调用状态、任务进度、中断控制、最终答案、异常状态、安全与不确定性提示。

14.2 基础反馈模块

基础反馈模块是用户点击发送后产品立即给出的确认反馈。它回答用户最基本的问题：我的消息发出去了吗？AI 收到了吗？系统还活着吗？

功能组成包括消息发送成功反馈、输入框状态变化、发送按钮变化、AI 头像 / 气泡激活、等待动画、状态文案、可中断入口。基础反馈必须即时、明确、可控。

14.3 首 token 反馈模块

首 token 反馈指 AI 开始输出第一段可见内容的时机。它不一定是完整答案，也可以是“我会从三个方面分析”“正在整理结果”“先给你结论”“我需要先检查资料”。

首 token 的意义在于降低用户的感知等待时间。更好的首 token 不是“好的，我来帮你分析一下”，而是有信息量的结构预告。

14.4 流式生成模块

流式生成是 AI 答案逐字、逐句、逐段出现的过程。它的价值是降低等待感、让用户提前判断方向、允许用户中途停止、形成“AI 正在工作”的持续反馈。

不同内容适合不同输出粒度：闲聊逐句输出，代码按代码块输出，表格整块输出，长报告按章节输出，搜索结果先摘要后展开，Agent 任务按步骤输出。

14.5 思考展示模块

思考展示模块决定用户能否看到 AI 的任务理解、问题拆解、推理路径、假设判断、工具选择、答案验证、不确定性判断。

思考展示应采用五层模型：完全隐藏、状态提示、思考摘要、结构化过程、详细推理链。成熟产品不应该默认详细层，而应该根据任务和用户自动选择层级。

最佳实践是：默认摘要，允许展开；简单任务隐藏，复杂任务增强；最终答案永远独立。

14.6 模式切换模块

模式切换模块允许用户或系统选择 AI 当前回答策略。典型模式包括快速回答、深度思考、自动判断、只看结论、展开过程、工具增强。推荐四个主模式：自动、快速、深度、专家。

模式切换不能让用户负担过重。更好的做法是默认自动，输入框旁提供深度思考按钮，生成中允许直接给结论，生成后支持用深度模式重答。

14.7 推理强度控制模块

推理强度控制模块决定 AI 在回答前投入多少推理资源。它可以对应 thinking budget、reasoning effort、最大推理 token、最大思考时间、是否允许多轮自检、是否允许工具调用、是否允许反思与验证。

用户侧不应暴露 technical 参数，而应转译成快速、平衡、深度、专家等表达。

14.8 工具调用状态模块

工具调用状态模块用于展示 AI 在回答过程中调用的外部能力，包括搜索网页、阅读文件、分析图片、运行代码、查询数据库、调用日历、读取邮件、生成图片、创建文档、部署网站、调用第三方 API。

工具状态卡应包含工具名称、当前状态、输入对象、输出摘要、风险提示、可操作项。工具调用状态必须真实，不能假装。

14.9 任务进度模块

任务进度模块适用于复杂多步骤任务，尤其是 Agent 场景。它回答 AI 当前做到哪一步、还剩哪些步骤、有没有失败、是否需要用户确认。

推荐结构是任务目标、任务计划、当前步骤、已完成步骤、进行中步骤、失败 / 风险步骤、用户确认节点、最终交付物。

14.10 中断控制模块

中断控制模块让用户在等待、思考、生成、工具调用过程中保持主动权。基础能力包括停止生成、取消任务、暂停执行、继续执行、重新生成、缩短回答、直接给结论、切换深度模式、隐藏思考、展开思考。

14.11 最终答案模块

最终答案模块是 AI 完成回复后的稳定内容区域，应包括结论、正文、表格、引用、思考区、操作按钮、追问建议、导出入口。最终答案与思考过程必须分离。

14.12 异常状态模块

异常状态模块处理网络失败、模型超时、工具失败、文件读取失败、搜索失败、内容过长、上下文超限、生成被中断、安全策略阻止、第三方 API 失败等。

异常文案必须说明发生了什么、哪一步失败、是否影响最终答案、用户可以怎么做。

14.13 安全与不确定性提示模块

安全与不确定性提示模块用于高风险或信息不足时提醒用户，适用于医疗、法律、金融、投资、政策、新闻、未验证资料、动态事实、文件读取不完整等场景。

第 15 章：信息架构分析

15.1 核心问题

AI 等待态的信息架构解决哪些内容应该展示、展示在哪里、默认展开还是折叠、过程和答案如何分离、用户如何继续操作的问题。信息架构不好，会导致思考过程挤占答案、工具状态和正文混乱、用户找不到结论、来源不清楚、按钮太多、移动端阅读困难、专业用户无法审查、普通用户被吓退。

15.2 核心信息层级

建议把 AI 回复过程信息分成五层：系统状态、任务过程、思考过程、最终答案、后续操作。

层级	内容	目标
系统状态	正在生成、正在搜索、已失败	告诉用户系统在工作
任务过程	当前步骤、已完成步骤、下一步	告诉用户任务进度
思考过程	摘要、假设、推理、验证	帮助用户理解判断路径
最终答案	结论、正文、表格、建议	交付可用内容
后续操作	复制、重试、展开、导出	支持用户继续使用

15.3 推荐页面结构

普通聊天结构适合简单问答：用户问题、AI 简短等待状态、AI 流式答案、操作按钮。

深度分析结构适合产品分析、商业研究、策略判断：用户问题、AI 任务理解摘要、思考 / 分析框架折叠区、最终答案正文、来源 / 假设 / 风险、操作按钮。

工具任务结构适合搜索、文件、代码、Agent：用户任务、任务计划、工具状态卡、中间结果、最终交付物、操作按钮。

15.4 思考内容应该放在哪里

思考内容可以放在答案前、答案上方折叠区、答案后方。答案前适合学习、数学、逻辑、代码讲解，但会延后结论；折叠区适合专业分析、研究、写作、决策；答案后适合高频办公和快速建议。

15.5 推荐方案：结论优先 + 思考可展开

本报告建议采用：一句话结论、完整答案、思考摘要可展开、详细过程可选。原因是大多数用户优先要答案，专业用户可以查看过程，思考不会挤压主体内容，移动端更友好，便于多层信息架构。

15.6 思考区的信息结构

思考区不应是一整段长文本，而应结构化：我如何理解这个问题、采用的分析维度、排除了哪些方向、依据哪些信息、最大风险是什么、为什么得出这个结论。

15.7 工具状态的信息结构

工具状态应独立于正文，推荐卡片结构：工具名称、状态、处理对象、输出摘要、风险提示、操作。对于文件分析，必须显示读取范围与未解析内容。

15.8 最终答案的信息结构

最终答案必须高可用：标题、一句话结论、核心判断、详细分析、表格 / 框架、风险与限制、行动建议、引用 / 来源、操作按钮。

15.9 移动端与桌面端差异

移动端屏幕小，适合短状态、折叠卡片、底部固定操作、默认隐藏详细过程、用按钮切换看答案 / 看过程。桌面端空间大，适合左侧任务计划、中间答案、右侧来源 / 思考 / 工具状态、多栏布局、Agent 执行面板。

第 16 章：交互设计分析

16.1 AI 等待态的状态机

一个完整的 AI 等待态系统本质上是状态机。推荐状态机包括 Idle 空闲、User Typing 用户输入、Message Sent 已发送、Request Accepted 请求已接收、Classifying Task 识别任务类型、Thinking / Retrieving / Tool Calling 思考 / 检索 / 调用工具、Generating 生成中、Completed 完成。

异常分支包括 Tool Failed 工具失败、Timeout 超时、User Stopped 用户停止、Safety Blocked 安全拦截、Partial Completed 部分完成、Regenerating 重新生成。

16.2 发送后即时反馈

用户点击发送后，应立即发生：用户消息进入聊天流、输入框清空或进入可继续输入状态、发送按钮变为停止按钮、AI 回答区域出现占位、状态文案出现。

推荐状态文案：已收到，正在处理；正在理解问题；正在准备回答；正在分析你的请求。不推荐“请耐心等待”“稍等一下哦”“马上就好”，因为它们不具体。

16.3 等待动画设计

等待动画可以是点状跳动、呼吸气泡、骨架屏、进度条、步骤列表、工具卡片、思考折叠区。进度条适合确定性任务，不适合不确定任务。AI 生成更适合步骤进度而不是百分比进度。

16.4 思考区域展开与折叠

思考区域建议默认折叠，除非用户明确开启深度过程模式。折叠状态下仍保留一行摘要，如“已完成问题拆解、竞品比较和风险判断”。这样用户即使不展开，也知道 AI 做了什么。

16.5 直接给结论按钮

在深度思考或长生成过程中，应提供“直接给结论”。适用于用户赶时间、AI 思考太久、思考过程太长、用户只需要最终建议等场景。

16.6 展开过程按钮

专业用户需要查看过程，按钮文案可以是展开思考、查看分析过程、查看依据、查看推理摘要、查看工具记录。普通用户更容易理解“查看依据”而不是“查看 reasoning”。

16.7 模式切换交互

推荐在输入框附近设置模式入口：自动、快速、深度、专家。每个模式需要一句短说明：自动根据问题复杂度自动选择；快速更快，适合简单问题；深度更慢，适合复杂分析；专家展示依据、过程与风险。

16.8 工具调用卡片交互

工具调用卡片需要支持未开始、进行中、已完成、失败四种状态。失败后应提供重试或跳过搜索继续回答等操作。

16.9 进度条是否适合 AI 生成任务

进度条适合文件上传、视频生成、图片生成、批量处理、已知步骤的 Agent 工作流、长文档解析。不适合普通对话生成、创意写作、不确定长度分析、多轮推理。

16.10 桌面端交互设计

桌面端可采用多面板：左侧任务计划 / 历史对话，中间最终答案，右侧思考过程 / 来源 / 工具状态，底部输入框与模式控制。

16.11 移动端交互设计

移动端更适合单列 + 折叠：顶部当前模型 / 模式，中间对话内容，卡片显示思考摘要 / 工具状态且可折叠，底部输入框 + 停止按钮，悬浮直接给结论 / 展开过程。

16.12 交互设计总结

AI 等待态交互设计核心原则是即时反馈、状态具体、过程分层、答案独立、用户可控、工具透明、异常诚实、移动端克制。

第 17 章：文案系统分析

17.1 为什么文案系统重要

AI 等待态文案不是装饰，而是用户理解系统状态的主要方式。一句好的等待文案可以让用户知道 AI 是否收到请求、当前在做什么、为什么需要等待、是否使用工具、是否存在风险、用户能做什么。

17.2 等待文案五个标准

好的 AI 等待态文案应该真实、具体、短、可解释、可操作。不能说没做的事；不要只写“处理中”，要写清楚处理什么；等待文案不是正文，应尽量短；要解释等待原因；必要时给用户选择。

17.3 文案分层体系

等待文案可分为基础状态文案、思考状态文案、工具调用文案、生成过程文案、异常状态文案、安全与不确定性文案。

17.4 基础状态文案

状态	推荐文案
已收到	已收到，正在处理
准备回答	正在整理回答
简单问答	正在生成
稍长任务	正在分析你的问题
长任务	正在拆解任务结构

17.5 思考状态文案

场景	推荐文案
产品分析	正在拆解产品功能模块
商业分析	正在比较市场、用户与商业价值
代码任务	正在定位问题和可能原因
数学推理	正在逐步验证计算过程
决策建议	正在比较不同方案的利弊
研究报告	正在整理分析框架和资料口径

17.6 工具调用文案

搜索类：正在搜索最新资料、正在比较多个来源、正在核对发布时间和来源可靠性、正在整理可引用来源。

文件类：已收到文件，正在解析；正在读取文档结构；正在提取关键章节；正在根据文件内容生成分析；部分页面无法读取，已标注影响范围。

代码类：正在扫描项目结构；正在定位报错来源；正在生成修改方案；正在运行测试验证；测试未通过，正在分析原因。

Agent 类：正在制定任务计划；正在执行第 2 步：收集资料；需要你确认后继续；已完成任务并生成交付物。

17.7 生成过程文案

长内容生成中应展示具体进展：正在生成执行摘要、正在扩写用户场景、正在整理竞品对比表、正在补充风险分析、正在输出最终结论。

17.8 异常状态文案

异常文案必须诚实、具体、可操作。比如：网络请求失败，当前无法继续搜索。你可以重试，或让我基于已有信息继续分析。文件第 12-15 页未能解析，可能影响图表部分分析。可以重新上传，或先基于已读取内容继续。

17.9 安全与不确定性文案

动态信息：这类信息可能变化，我会先核对最新来源。医疗健康：这不能替代医生建议，我会尽量基于可靠资料解释。法律政策：具体适用可能因地区不同，需要结合当地法规核验。投资金融：这不是投资建议，我会重点说明风险和不确定性。

17.10 六个产品文案风格启发

产品	文案风格启发
ChatGPT	简洁、工作台化、偏专业
Gemini	轻量、助手化、生态自然
Claude	克制、解释性强、适合专业任务
DeepSeek	思考感强，但需要减少冗余
Qwen	适合用快速 / 深度 / 自动翻译技术模式
豆包	适合大众化、短句化、低门槛表达

17.11 推荐文案规范

不要说假状态，不要过度拟人化，不使用空泛词，不把技术词直接丢给普通用户，不承诺无法保证的结果，复杂任务解释等待原因，异常任务给出下一步操作。

17.12 文案模板库

场景	文案模板
普通问答	正在生成回答。
复杂分析	正在拆解问题结构，并整理分析维度。
产品分析	正在从用户场景、功能机制和竞品差异三个角度分析。
搜索研究	正在搜索并核对多个可靠来源。
文件分析	正在读取文件结构，并提取关键内容。
代码任务	正在定位报错来源，并检查可能的修复路径。
深度思考	正在深度分析，这会比普通回答更慢，但会给出更完整的判断。
直接结论	已跳过详细过程，直接输出核心结论。

第 18 章：产品价值分析

18.1 功能价值总览

聊天 AI 的等待态 / 思考态 / 生成态，表面上看是交互体验，实际上影响完整产品价值链。它对用户的价值是降低等待焦虑、提升信任、增强控制感、帮助理解 AI 的工作过程。它对产品的价值是提升留存、降低中断、增强专业感、形成差异化、支撑深度思考付费、承接 Agent 任务。它对商业的价值是把不可见的模型算力、推理能力、工具能力，转化成用户可感知、可理解、愿意付费的产品能力。

18.2 降低等待焦虑

用户等待 AI 回复时，最直接的问题不是慢，而是不知道发生了什么。等待态的第一层价值，就是把未知等待变成可解释等待。用户不一定不能接受慢，但不能接受不知道为什么慢。

18.3 提升对答案的信任

AI 答案的特殊之处在于，用户看不到它的形成过程。等待态和思考态可以在这个缺口中起到信任桥梁作用：展示正在搜索说明答案可能基于外部信息；展示正在读取文件说明它处理了用户上传内容；展示正在推理说明它不是直接随机生成；展示正在验证说明它对结果做了检查；展示引用来源说明答案可以被复核。

18.4 增强控制感

AI 产品的问题之一是一旦用户点击发送，后续过程常常由模型接管。等待态系统必须给用户控制权，包括停止生成、继续生成、直接给结论、展开思考、隐藏思考、切换快速模式、切换深度模式、重新深度思考、跳过失败工具、重试工具调用、修改任务计划。

18.5 提升复杂任务可完成感

普通聊天 AI 的核心是回答问题，Agent 化 AI 的核心是完成任务。复杂任务需要任务进度系统：任务目标、任务计划、当前步骤、工具调用、中间产物、失败提示、人工确认、最终交付。

18.6 提升首轮体验与新用户转化

新用户第一次使用 AI 产品时会快速形成心智判断：这个 AI 快不快、是否靠谱、是否专业、是否好控制、是否值得继续用。等待态强烈影响第一次体验。

18.7 提高留存和使用频次

用户留存不只由答案质量决定，也由使用时是否顺手决定。等待态降低使用摩擦、降低失败感、提高可控感，尤其影响高频使用中的微体验。

18.8 增强专业感和品牌差异化

六个产品的等待态已经形成不同品牌人格：ChatGPT 成熟稳定，Gemini 轻量生态，Claude 克制专业，DeepSeek 推理可见，Qwen 工程化，豆包大众轻量。

18.9 支撑深度思考付费

深度思考往往意味着更多 token、更多算力、更高延迟和更高成本。产品必须让用户感知深度思考的价值，否则用户只会觉得“为什么这么慢”。

18.10 提高工具调用和 Agent 的可信度

AI 产品正在从聊天走向工具调用和 Agent 执行。工具能力越强，用户越需要知道调用了什么工具、是否成功、数据来自哪里、文件是否读全、代码是否运行、哪一步失败、是否需要人工确认。

18.11 降低客服和负反馈成本

等待态差的产品，用户反馈会集中在卡住、太慢、不知道在干什么、文件是不是没读、为什么没有搜索、为什么等这么久还失败。清楚的等待态和异常处理可以提前消化大量负反馈。

18.12 支撑企业级合规与审计

企业级用户不仅关心 AI 能不能回答，还关心数据来源、工具调用、过程可追溯、是否误操作、是否泄露信息、是否能审计。等待态、工具状态和任务进度可以沉淀为审计日志。

第 19 章：数据指标体系

19.1 指标体系总览

评估 AI 等待态功能是否成功，不能只看用户喜不喜欢，也不能只看平均响应时间。建议分为速度指标、过程体验指标、用户控制指标、结果质量指标、商业转化指标、风险与成本指标。

19.2 北极星指标

等待态系统的北极星指标不应是平均响应时间，而应是**有效任务完成率 Effective Task Completion Rate**，即用户认为本次 AI 回复完成了目标任务的比例。可结合复制、继续追问、导出、点赞、未重新生成、未中断、未等待中流失等行为判断。

19.3 速度指标

指标	含义
First Feedback Latency	点击发送到首次可见反馈的时间
First Token Latency	点击发送到第一个可读 token / 短句出现的时间
Time to Meaningful Output	发送到第一段有信息量内容的时间
Time to Complete	完整回复完成时间

速度指标必须按任务分类建基准。同样 30 秒，简单翻译太慢，深度商业分析可能可以接受，Agent 多步骤任务则不适合只看首 token。

19.4 过程体验指标

指标	含义	解释
Waiting Drop-off Rate	等待中流失率	高说明等待太久或状态不清晰
Stop Rate	停止生成比例	需结合任务类型判断原因
Thinking Expand Rate	思考区展开率	判断思考展示是否有价值
Thinking Collapse Rate	展开后快速折叠比例	高说明思考太长或无帮助
Tool Status View Rate	工具详情查看率	判断工具状态对信任的价值

19.5 用户控制指标

用户控制指标包括 Mode Switch Rate、Deep Thinking Adoption Rate、Direct Answer Click Rate、Retry / Regenerate Rate。

Direct Answer Click Rate 对 DeepSeek 类产品尤其重要。高点击率可能说明思考过程太长、用户更关心结果、当前任务不适合展示过程、默认模式可能过深。

19.6 结果质量指标

结果质量指标包括 Copy Rate、Export Rate、Follow-up Depth、Citation Interaction Rate。需要区分复制的是最终答案、思考过程、代码、表格、引用还是任务计划。

19.7 商业转化指标

商业转化指标包括 Deep Mode Conversion Rate、Paywall Trigger Success Rate、Agent Task Completion Monetization。等待态是付费价值展示层，深度模式要让用户看到它确实做了更多步骤、更可靠、更完整。

19.8 成本与效率指标

成本与效率指标包括 Reasoning Token Cost、Cost per Successful Task、Overthinking Waste Rate。深度模式成本高，但如果一次完成任务，可能比快速模式低成本多次重试更划算。

19.9 风险与护栏指标

风险指标包括 Hallucination Complaint Rate、Misleading Reasoning Rate、Tool Failure Silent Rate。工具失败但未明确告知是严重风险。

19.10 A/B 测试设计

可测试隐藏思考 vs 思考摘要 vs 完整思考；自动模式 vs 用户手动模式；工具状态简单展示 vs 详细展示。复杂任务中折叠摘要可能最好，简单任务中隐藏或状态提示可能最好，默认详细过程可能提升透明感但增加认知负担。

19.11 指标总结

等待态指标体系核心不是“快”，而是足够快、足够清楚、足够可控、足够可信、成本可接受、结果可使用。

第 20 章：核心设计矛盾

20.1 透明 vs 简洁

透明意味着展示思考、工具调用、来源、假设、风险、任务进度。简洁意味着不打扰用户、少展示过程、直接给答案、降低视觉负担。DeepSeek 偏透明，ChatGPT 偏简洁，Claude 试图平衡。最优解是默认简洁、复杂任务显示摘要、专业用户可展开、最终答案始终清楚。

20.2 速度 vs 质量

深度思考通常更慢，但可能带来更好复杂任务表现。快速模式更爽，但复杂任务容易浅。产品不能简单追求最快，也不能所有任务都深度思考。

20.3 原始推理 vs 可读解释

很多用户以为展示完整思维链就是最透明，但原始推理可能太长、太乱、有错误、用户看不懂、可能被误解为真实内心。成熟产品更倾向把 raw thinking 转为摘要式、结构化、可读的过程说明。

20.4 专业用户 vs 普通用户

普通用户想要快、简单、直接；专业用户想要依据、过程、来源、风险、可审查、可导出。一个产品不能用同一种等待态服务所有人。

20.5 用户感知能力 vs 模型真实能力

用户会通过等待态判断 AI 是否强，但看起来在思考不等于真的更正确，显示很多推理不等于推理忠实，工具卡片很多不等于任务完成得好。等待态既能增强能力感知，也可能制造虚假能力感。

20.6 展示思考 vs 暴露错误

展示思考过程可以建立信任，也可能暴露错误：引用错误前提、自相矛盾、偏离用户问题、最终答案和思考不一致、高风险问题过度自信。产品要做好结构化和风险提示。

20.7 深度思考价值 vs 算力成本

深度思考真实消耗成本。产品经理必须判断哪些任务值得深度思考、哪些用户值得提供深度模式、是否需要自动降级、是否需要限制长思考、是否需要缓存和复用推理状态。

20.8 自动判断 vs 用户手动控制

自动判断省心、体验流畅、降低选择负担；手动控制给用户掌控感，适合专业用户和企业配置。最佳方案是默认自动、用户可覆盖、专业用户可细调、企业管理员可配置。

20.9 结果优先 vs 过程优先

普通问答、翻译、改写、文案、快速建议适合结果优先；学习解题、数学推导、代码调试、复杂推理、审计任务适合过程优先；大多数专业任务适合一句话结论、分析过程摘要、完整答案、可展开详细过程的混合结构。

第 21 章：产品风险分析

21.1 风险总览

等待态 / 思考态 / 生成态带来巨大价值，也带来风险：误导性透明、CoT 忠实性风险、思考过载、延迟与成本、工具状态造假、过度拟人化、隐私与数据泄露、企业审计风险、高风险场景误用、品牌信任反噬。

21.2 误导性透明

思考展示最危险的地方在于，它会让用户产生“我看到了 AI 的思考，所以我知道它为什么这么回答”的错觉。外显思考过程只是模型生成给用户看的文本或摘要，并不必然等同于模型真实、完整、忠实的内部机制。

应对策略包括不把思考区命名为真实思维，使用分析摘要、推理摘要、处理过程等表达，高风险场景提示过程仅供辅助判断，最终答案仍需来源、证据和验证。

21.3 思考过程过载

长思考可能导致简单问题回答太慢、思考内容占据大量屏幕、用户找不到最终结论、移动端体验变差、用户频繁点击停止、用户觉得 AI 啰嗦。

应对策略是默认折叠长思考、简单任务自动关闭深度思考、提供直接给结论、思考过程摘要化、思考区和答案区分离。

21.4 延迟失控

深度思考、搜索、文件读取、代码执行、Agent 多步骤任务都会增加延迟。延迟失控会导致用户流失、中断率上升、任务完成率下降、服务成本上升。

应对策略是首 token 尽快出现、长任务显示任务进度、超过阈值提示可先给简版结论、自动判断任务复杂度、设置最大 thinking budget、工具调用超时明确提示。

21.5 算力成本失控

深度思考不是免费功能。Reasoning tokens、thinking tokens、tool calls、长上下文、文件处理、联网搜索都会产生成本。产品需要默认自动模式、简单任务不启用深度思考、复杂任务前提示成本或时间、付费用户提供更高 budget、企业用户按任务或 token 计费、监控 Cost per Successful Task。

21.6 工具状态造假或误导

如果产品显示正在搜索但实际没有搜索，显示已读取全文但文件只读一部分，显示测试通过但代码没有运行，用户会迅速失去信任。工具状态必须真实，工具失败必须显示，部分完成必须标注，输出答案要说明依据范围。

21.7 过度拟人化

等待态文案如果过度拟人化，会让用户误解 AI 能力边界。例如“我完全理解你了”“我一定会给你最正确答案”“相信我”。更合适的表达是“我将按你提供的信息分析”“以下结论基于当前信息”。

21.8 隐私与敏感信息暴露

思考过程、工具日志和任务进度可能暴露用户上传文件中的隐私内容、企业内部资料、邮件对象、联系人、文件路径、API 返回内容、工具调用参数。应敏感字段脱敏、企业用户可关闭过程展示、工具日志分权限显示、不在前端展示完整参数。

21.9 高风险领域误用

医疗、法律、金融、投资、心理健康、政策、新闻等高风险领域，等待态和思考态可能放大用户信任。如果 AI 展示了很长推理，用户可能误以为它想得细就一定对。高风险任务应强制显示不确定性、优先展示来源、避免过度自信、给出专业边界。

21.10 品牌信任反噬

等待态本来是为了增强信任，但如果思考过程很长而答案很差、状态显示专业但工具失败、深度模式很慢但没有更好、说正在搜索但来源过期，会造成品牌反噬。用户会从“这个 AI 看起来很认真”变成“它只是装得很认真”。

21.11 产品复杂度上升

等待态系统越完整，产品复杂度越高，会带来 UI 复杂、用户学习成本增加、状态过多、文案维护困难、埋点复杂、多端一致性难等问题。应普通用户默认简洁，专业能力渐进披露，状态机统一管理，文案模板化，工具状态标准化。

21.12 风险治理框架

等待态风险治理应关注真实性、可理解性、可控制性、可审查性、边界性、成本性、安全性。

第 22 章：用户分层策略

22.1 为什么必须做用户分层

聊天 AI 产品不能对所有用户展示同一种等待态。普通用户只想快速得到答案，学生用户希望看到步骤，专业用户需要依据和风险，开发者需要工具日志和错误信息，企业用户需要审计链路，高阶用户需要模式控制和推理强度调节。

如果对所有人展示完整思考，会让普通用户疲劳；如果对所有人隐藏过程，会让专业用户不信任。因此 AI 等待态的核心原则是给不同用户展示他们刚好需要的过程信息。

22.2 用户分层总览

用户类型	核心诉求	等待态策略	思考展示策略
普通大众用户	快、简单、直接	轻量等待	默认隐藏过程
学生 / 学习用户	看懂步骤、掌握方法	步骤化等待	展示解题过程
专业办公用户	可用、可靠、可复制	结构化等待	思考摘要 + 最终答案
研究 / 分析用户	依据、来源、推理链	资料处理状态	展示假设、来源、风险
开发者用户	代码路径、报错、工具结果	工具日志化	展示执行过程
企业用户	可控、可审计、可追溯	任务面板化	审计链路 + 权限控制
高阶 AI 用户	模式控制、效率优化	可配置状态	快速 / 深度 / 专家可切换

22.3 普通大众用户

普通用户使用聊天 AI 的目标通常是问知识点、改句话、生成文案、生活建议、简单翻译、聊天陪伴、快速获取答案。他们不关心模型内部机制，也不想理解 reasoning token、thinking budget、CoT、tool calling。

普通用户等待态应轻、短、快、可跳过。默认应该是已收到、正在生成、直接输出答案，不建议默认展示完整推理链、长思考过程、工具调用细节或专家级日志。

22.4 学生 / 学习用户

学生用户不是只要答案，而是要学会。学习场景适合展示题目理解、解题思路、关键公式、步骤推导、常见错误、最终答案、方法总结和类似题迁移方法。

学习产品更适合教学化思考，而不是原始长推理。应该先说明解题计划，过程分步骤展示，最后总结方法，并允许切换“只要答案 / 讲解过程”。

22.5 专业办公用户

专业办公用户包括产品经理、运营、市场、咨询顾问、内容策划、数据分析师、管理者等。他们需要一个可以直接拿去用、能解释依据、结构清楚的结果。

专业办公用户适合结构化等待：正在拆解任务结构、正在整理分析框架、正在生成正文、正在优化格式、输出最终结果。思考展示以思考摘要 + 框架展示最有价值。

22.6 研究 / 分析用户

研究用户需要来源、假设、推理和限制。他们关心资料来自哪里、是否过期、哪些是事实哪些是推断、结论假设是什么、反例是什么、不确定性是什么。

这类用户适合看到正在搜索资料、筛选来源、比较信息差异、区分事实与推断、形成结论。适合 L3 层级的结构化过程。

22.7 开发者用户

开发者用户需要工具日志、文件路径、错误与验证。代码任务不适合单纯聊天式等待态，更适合执行日志式等待态。理想状态是正在扫描项目结构、定位错误文件、分析调用链、生成修复方案、修改代码、运行测试、输出 diff 和测试结果。

22.8 企业用户

企业用户需要可控、可审计、可追溯。企业场景不一定需要展示完整 CoT，但必须展示输入来源、调用工具、数据权限、输出范围、异常记录、人工确认节点、最终责任边界。

22.9 高阶 AI 用户

高阶 AI 用户需要模式控制和推理资源管理。他们希望控制快速还是深度、是否联网、是否读取文件、是否展示思考、是否只看结论、是否保留中间过程、是否导出结果。

22.10 用户分层方法论总结

用户越普通，等待态越轻；用户越专业，过程越透明；任务越高风险，边界越明确；任务越复杂，进度越可见。

第 23 章：场景分层策略

23.1 为什么要做场景分层

同一个用户在不同场景下，对 AI 等待态的需求完全不同。翻译一句话需要快速输出；分析六个 AI 产品的等待态功能，需要任务拆解、思考摘要、结构化生成、阶段进度和最终报告。

AI 等待态的第二个核心方法论是：不是按产品固定展示，而是按任务场景动态展示。

23.2 场景分层总览

场景类型	用户目标	推荐等待态
普通问答	快速知道答案	轻量等待
事实查询	查准信息	搜索 / 来源状态
写作创作	生成可用文本	草稿生成状态
学习解题	理解过程	步骤化思考
产品 / 商业分析	形成结构化判断	分析框架 + 深度生成
文件分析	读取并总结文件	文件读取进度
代码任务	修改 / 排错 / 验证	工具日志 + 测试状态
多模态任务	看图 / 视频 / 语音	识别状态 + 输出
Agent 任务	多步骤执行	任务面板
高风险建议	谨慎判断	风险提示 + 来源核验

23.3 普通问答场景

普通问答需要轻等待 + 快速答案。不要深度思考、完整推理、长前言、多步骤计划。用户问“MVP 是什么”，理想回答就是正在生成后直接解释 MVP，而不是先分析 MVP 的历史和创业背景。

23.4 事实查询场景

事实查询需要搜索状态 + 来源整理。动态信息必须显示是否检索、是否核对来源。等待态应是正在搜索最新资料、正在核对多个来源、正在整理答案和引用、输出结论。

23.5 写作创作场景

写作创作中，方向感比推理过程更重要。等待态可以显示正在整理结构、生成初稿、优化语气、调整格式。用户关心最终文本是否可用，而不是推理过程是否完整。

23.6 学习解题场景

学习场景中，过程是价值核心。等待态可以显示正在判断题型、拆解步骤、验证答案、输出讲解。输出结构应该是题目理解、解题思路、逐步推导、最终答案、方法总结、类似题提示。

23.7 产品 / 商业分析场景

产品分析、竞品分析、商业模式分析、用户体验分析、行业研究、PRD、复盘报告、策略建议等任务复杂、长、维度多。等待态应显示正在拆解分析范围、建立分析框架、比较关键维度、生成报告正文、整理结论和建议。

23.8 文件分析场景

文件分析必须让读取范围可见。用户上传文件后，最大疑问是 AI 到底读没读文件。等待态必须显示文件名、文件类型、是否读取成功、读取范围、未读取部分、OCR / 图表识别限制、分析是否基于完整文件。

23.9 代码任务场景

代码任务需要工具日志 + 验证状态。等待态应显示扫描项目结构、定位错误来源、分析依赖关系、生成修改方案、运行测试、输出结果。如果没有实际运行测试，就必须说清楚。

23.10 多模态任务场景

多模态任务需要识别状态清楚。用户看不到模型如何处理图片或视频，因此应显示正在识别图片内容、提取关键元素、分析画面关系、生成结论，并说明低清晰度、遮挡或推断限制。

23.11 Agent 多步骤任务场景

Agent 任务不是单轮回答，而是多步骤执行。等待态应该升级为任务面板：任务计划、当前步骤、工具调用、中间结果、用户确认、最终交付。

23.12 高风险建议场景

高风险场景包括医疗、法律、金融、投资、心理健康、政策、安全、个人重大决策。等待态应显示正在核对可靠来源、区分一般信息与具体建议、整理风险和限制、输出谨慎结论。不能用很长思考过程制造专业诊断感。

23.13 场景分层方法论总结

任务越简单，越要快；任务越复杂，越要透明；任务越高风险，越要谨慎；任务越像执行，越要有进度。

第 24 章：六个产品路线总结

24.1 路线总览

产品	路线名称	核心策略	优势	风险
ChatGPT	成熟工作台路线	答案优先，过程克制，工具增强	稳定、低干扰、生产力强	复杂推理透明度不足
Gemini	生态助手路线	普通轻量，多模态与 thinking 补充	自然、生态化、多模态强	等待态记忆点不强
Claude	专业透明路线	折叠式思考，摘要透明	克制、专业、可控	普通用户理解成本略高
DeepSeek	强思考展示路线	把推理过程前台化	透明感强、差异化强	思考过载、可能误导
Qwen	工程化可切换路线	thinking / non-thinking 可配置	适合开发者和企业	消费端表达需简化
豆包	大众轻量路线	普通端轻量，深度能力后置	低门槛、高频友好	专业任务透明感不足

24.2 ChatGPT：成熟工作台路线

ChatGPT 的最大特点是把复杂能力隐藏在稳定、低干扰、可交付的工作台体验后面。它不追求展示完整思考，而是让用户尽快进入最终答案，并在搜索、文件、代码、图像、数据等工具场景中展示必要状态。

24.3 Gemini：生态助手路线

Gemini 的路线是普通用户端轻量自然，复杂能力通过 thinking、多模态、搜索和 Google 生态补充。它的启发是 AI 等待态不一定要强展示推理，而可以围绕生态任务展示正在查、正在看、正在整理。

24.4 Claude：专业透明路线

Claude 的路线是默认克制，需要时透明，复杂任务可深度思考。它证明透明不是把所有过程都展示，而是把有用过程摘要化、折叠化、可控化。

24.5 DeepSeek：强思考展示路线

DeepSeek 把等待变成可观看的思考过程。它证明等待态本身可以成为产品差异化，但也提醒我们可见思考需要管理，否则容易从透明变成负担。

24.6 Qwen / 通义千问：工程化可切换路线

Qwen 把是否思考做成工程化开关，把思考深度做成可配置能力。它证明未来 AI 产品不是只有一个回答按钮，而应该有任务感知的 reasoning mode。

24.7 豆包：大众轻量路线

豆包普通用户端尽量轻量，深度推理能力在特定模式和模型平台中体现。它证明大众产品不应默认复杂化，但必须给高级任务保留深度入口。

24.8 长期融合趋势

未来主流 AI 产品很可能融合六种路线：ChatGPT 的工作台稳定性、Gemini 的生态任务感、Claude 的折叠式透明、DeepSeek 的强推理感知、Qwen 的模式可控、豆包的大众轻量入口。

24.9 谁更适合什么用户

用户 / 场景	更适合路线
大众日常问答	豆包 / Gemini / ChatGPT 普通模式
专业办公	ChatGPT / Claude
长文档与复杂写作	Claude / ChatGPT
学习推理	DeepSeek / Qwen / Claude
代码与开发者	ChatGPT / Claude / Qwen
企业 Agent	Qwen / ChatGPT / Claude
多模态生态任务	Gemini / 豆包 / ChatGPT
强过程感体验	DeepSeek
工程化模式控制	Qwen
低门槛移动端	豆包

第 25 章：最佳实践提炼

25.1 总原则

本报告提炼出的 AI 等待态 / 思考态 / 生成态核心方法论是：**默认轻量，复杂增强；过程分层，答案独立；工具真实，风险明确；用户可控，任务可见。**

25.2 默认不要打扰用户

普通问题不要默认展示长思考。用户问一个简单问题时，最好的体验是快速回答，不是展示“我正在从多个角度思考”。

25.3 复杂任务必须解释等待原因

如果 AI 需要很久，必须告诉用户为什么：正在搜索资料、读取文件、拆解问题、运行代码、生成报告、整理引用。复杂任务中的等待态要从“生成状态”升级为“任务状态”。

25.4 首 token 要有信息量

为了显得快而输出“好的，我来帮你分析”没有意义。更好的首 token 是“我会从用户场景、功能机制、竞品差异和商业价值四个角度分析”。

25.5 思考过程要分层展示

思考展示应分隐藏、状态提示、思考摘要、结构化过程、详细推理。默认不要直接展示最详细层。

25.6 最终答案必须独立于思考过程

不要让用户在长思考中找答案。推荐结构是一句话结论、完整答案、思考摘要可折叠、详细过程可选、来源与操作独立。

25.7 工具调用必须真实可见

如果 AI 调用了工具，就应该展示。如果没有调用工具，就不能假装调用。没搜索，不显示搜索；没读全文，不说读全文；没运行测试，不说测试通过；工具失败，必须说明失败。

25.8 失败状态必须诚实

失败不可怕，假装成功才可怕。异常文案应说明失败步骤、影响范围和下一步操作。

25.9 用户必须能中断和改方向

等待态必须有控制权，包括停止生成、继续、直接给结论、缩短、展开过程、隐藏过程、改用深度模式、改用快速模式、重新生成、基于当前继续。

25.10 模式数量不要太多

推荐四种以内：自动、快速、深度、专家。不要把 reasoning_effort、max tokens、CoT、thinking budget、enable_thinking 等技术词直接扔给普通用户。

25.11 任务越像 Agent，越需要进度面板

Agent 等待态应该变成任务进度面板：任务目标、任务计划、当前步骤、工具调用、中间产物、用户确认、最终交付。

25.12 高风险场景必须加强边界

医疗、法律、金融、投资、心理健康等场景中，等待态和思考态不能制造过度专业感。高风险场景中透明不是展示更多推理，而是展示更清楚的边界。

25.13 普通用户和专业用户必须用渐进披露

采用默认简洁、点击查看摘要、继续展开过程、专家模式查看工具和来源。这样普通用户不会被吓退，专业用户也不会觉得产品太浅。

25.14 等待文案必须具体

不要写“请稍候”，要写“正在读取文件结构”“正在搜索最新资料”“正在比较六个产品”“正在生成报告目录”。

25.15 不要制造表演式思考

思考展示的目的不是让 AI 看起来聪明，而是帮助用户理解、判断和控制。不要展示看起来很聪明的过程，要展示对用户完成任务有帮助的过程。

25.16 把等待态设计成数据驱动系统

上线后必须监控首反馈延迟、首有效内容时间、等待流失率、停止生成率、思考展开率、直接结论点击率、深度模式使用率、工具失败率、答案复制率、导出率、深度模式转化率、成本 / 成功任务比。

25.17 最终目标是完成任务

AI 等待态不是为了展示 AI 很忙，不是为了让用户看模型表演，不是为了堆动画、堆状态、堆推理。最终目标只有一个：让用户更放心、更清楚、更可控地完成任务。

第 26 章：未来趋势判断

26.1 总体判断

AI 等待态会从“生成反馈”升级为“智能任务操作系统”。未来等待态不再只是回答前的一段状态，而会成为 AI 产品的任务中枢：用户发送任务，AI 判断任务复杂度，自动选择快速 / 深度 / 工具 / Agent 模式，展示任务计划、思考摘要、工具调用、阶段性结果，允许用户中断 / 修改 / 确认，输出最终交付物，并保留过程记录和可复盘链路。

26.2 从 Loading UX 到 Thinking UX

传统软件等待态主要是 Loading UX：加载中、请稍候、进度条、转圈、骨架屏。AI 产品等待态正在变成 Thinking UX：正在理解问题、拆解任务、搜索资料、读取文件、调用工具、验证答案、生成最终结论。

26.3 从展示思考走向控制思考

早期 AI 产品的核心是让 AI 回答；推理模型出现后，用户开始看到 AI 可以先思考再回答；下一阶段核心会变成用户和系统可以控制 AI 如何思考，包括是否开启深度思考、思考多久、思考强度、是否展示思考、展示摘要还是详细过程、是否使用工具、是否优先速度、是否优先准确性。

26.4 从原始思维链转向摘要式透明

长期来看，主流成熟产品不会无限制展示原始推理过程，而会更倾向摘要式透明。原始推理可能过长、混乱、包含错误或无效路径，不一定忠实反映模型真实决策机制。未来更合理结构是模型内部原始推理、系统生成思考摘要、专业过程结构化、最终答案清晰独立。

26.5 从单轮回复转向多步骤 Agent 执行面板

今天多数聊天 AI 还是气泡流：用户问、AI 答、用户追问、AI 再答。Agent 任务更像目标、计划、执行、工具调用、中间结果、人工确认、失败重试、最终交付。当 AI 开始做真实任务时，等待态必须升级成执行面板。

26.6 工具调用状态成为信任基础设施

未来 AI 产品会越来越多调用外部工具。工具越多，信任问题越大。用户必须知道调用了什么工具、是否成功、输入是什么、输出是什么、是否失败、是否重试、是否需要确认、是否影响最终答案。

26.7 从模型选择到任务策略选择

今天很多产品让用户选择模型，但未来普通用户不应该被迫理解模型型号。更重要的是选择任务策略：快速回答、深度分析、联网核验、文件精读、代码执行、专家报告。

26.8 等待态成为付费价值展示层

深度思考、长上下文、工具调用、多模态、Agent 执行都会带来成本。AI 产品必须让用户理解为什么高级模式更慢、更贵、更有限。等待态不只是体验层，也是付费价值展示层。

26.9 从人等 AI 到人与 AI 协作

当前等待态常常是用户被动等待。未来会变成协作式等待：AI 先给计划，用户确认，AI 执行第一步，用户中途调整，AI 继续执行，用户选择输出格式，AI 完成交付。

26.10 移动端轻量化，桌面端工作台化

移动端屏幕小、使用碎片化，适合简短状态、底部悬浮按钮、折叠思考、只看结论、语音 / 图片输入、卡片式任务状态。桌面端更适合工作台化：左侧任务计划 / 历史，中间答案 / 文档，右侧思考摘要 / 来源 / 工具日志，底部输入与控制。

26.11 进入企业审计与合规系统

企业 AI 的核心不是会聊天，而是可控地完成任务。等待态会和审计系统融合，记录谁发起任务、使用哪些数据、调用哪些工具、是否访问外部网络、是否出现失败、是否经过人工确认、最终输出给谁、是否涉及敏感信息。

26.12 成为产品差异化重要前线

当模型能力逐渐接近，用户会更明显地感受到产品体验差异。未来用户评价 AI 产品，不只会说模型聪不聪明，也会说它知不知道自己在做什么、是不是让我等得明白、有没有给控制权、会不会假装读了文件、能不能像真正助理一样推进任务。

26.13 趋势总判断

未来 2-5 年，聊天 AI 等待态会从 Loading 到 Thinking，从回答生成到任务执行，从展示思考到控制思考，从完整 CoT 到摘要透明，从聊天气泡到 Agent 面板，从模型选择到任务策略选择，从被动等待到协作执行，从消费体验到企业审计，从体验细节到商业付费层。

第 27 章：如果从零设计一个聊天 AI 等待态功能

27.1 产品目标

从 0 设计聊天 AI 等待态 / 思考态 / 生成态功能，目标不能写成“做一个好看的 loading”。正确目标是：在用户发送消息后，通过即时反馈、任务识别、思考展示、工具状态、流式生成、中断控制和最终答案结构，让用户清楚知道 AI 正在做什么、为什么需要等待、是否可以控制、最终结果是否可信。

一句话：把 AI 的工作过程变得可见、可控、可信。

27.2 产品定位

该功能可命名为 AI Response Process System，即 AI 回复过程系统。它不是单点功能，而是覆盖发送反馈、等待反馈、任务识别、模式选择、思考展示、工具调用、任务进度、流式生成、中断控制、异常处理、最终答案、后续操作的系统。

27.3 用户问题

该功能要解决 10 个用户问题：我发出去了吗？AI 收到了吗？AI 是不是卡了？AI 在做什么？为什么这么慢？AI 是不是理解错了？它有没有真的搜索 / 读文件 / 跑代码？我能不能停止？我能不能直接看结论？最终答案能不能信?

27.4 核心设计原则

从 0 设计时建议遵守八条原则：即时反馈、状态具体、过程分层、答案独立、用户可控、工具真实、异常诚实、高风险有边界。

27.5 功能架构

一级模块包括状态识别层、任务分类层、模式选择层、思考展示层、工具调用层、流式生成层、用户控制层、异常处理层、最终答案层。

27.6 状态识别层

状态识别层负责判断 AI 当前状态，推荐状态机：Idle 空闲、User Typing 用户输入、Message Sent 已发送、Request Accepted 请求已接收、Task Classified 任务已识别、Thinking 思考中、Retrieving 检索中、Tool Calling 工具调用中、Generating 生成中、Completed 完成。异常状态包括 Stopped、Timeout、Tool Failed、Partial Completed、Safety Blocked、Regenerating。

27.7 任务分类层

任务分类层判断用户问题属于普通问答、写作改写、事实查询、深度分析、文件分析、代码任务、多模态任务、Agent 任务、高风险任务，并决定默认等待态。

27.8 模式选择层

推荐四种主模式：自动、快速、深度、专家。技术层可能是 reasoning_effort、thinking_budget、enable_thinking、max_tokens、tool_choice，但用户层应该是快速、深度、专家、是否联网、是否看过程、是否只看结论。

27.9 思考展示层

思考展示层采用五级模型：L0 不展示、L1 状态提示、L2 思考摘要、L3 结构化过程、L4 详细推理。默认不把详细推理全文铺开。

27.10 工具调用层

工具调用层展示网页搜索、文件读取、图片理解、代码运行、数据分析、日历、邮件、数据库、第三方 API、文档生成、图片生成等外部工具状态。必须遵守没搜索不显示搜索、没读全文不说读全文、没运行测试不说测试通过、工具失败必须说明失败。

27.11 流式生成层

流式生成层负责内容输出节奏。普通回答逐句输出，长报告先目录再分章，表格整块输出，代码按代码块输出，搜索总结先结论再来源，Agent 任务按步骤输出。

27.12 用户控制层

用户控制层必须覆盖生成前、生成中、生成后三个阶段。生成前选择模式、是否联网、是否深度、是否只要结论、是否需要引用；生成中停止、直接给结论、展开思考、隐藏思考、缩短、继续、跳过工具、重试工具；生成后重新生成、更详细、更简短、转表格、导出、继续下一步、查看来源、查看过程、反馈问题。

27.13 异常处理层

异常处理覆盖网络失败、搜索失败、文件读取失败、工具调用失败、超时、安全阻断、上下文过长。异常文案必须包含哪一步失败、为什么失败、对结果有什么影响、用户可以怎么做。

27.14 最终答案层

最终答案结构应是一句话结论、核心分析、详细内容、表格 / 清单 / 框架、风险与限制、行动建议、来源 / 文件 / 工具记录、后续操作。

27.15 信息架构设计

三种信息架构：轻量聊天型适合大众产品；折叠透明型适合专业聊天产品；任务面板型适合 Agent 和企业产品。

27.16 交互流程设计

完整流程：用户输入，系统识别任务类型，显示建议模式，用户发送，立即反馈，显示任务状态，如需工具显示工具卡，如需深度思考显示摘要，流式生成答案，用户可中断 / 修改 / 继续，输出最终答案，提供后续操作。

27.17 版本规划

MVP 解决基础等待焦虑，包含即时反馈、正在生成、停止生成、流式输出、简单错误提示、最终答案复制。V1 支持不同任务状态，包括搜索状态、文件读取状态、代码 / 工具状态、思考摘要、快速 / 深度模式、直接给结论、查看过程、部分失败提示。V2 支持复杂任务和 Agent，包括任务计划、步骤进度、工具卡片、中间结果、用户确认节点、失败重试、导出交付物、过程日志。Pro / Enterprise 支持专家模式、推理强度控制、工具权限、审计日志、数据来源记录、敏感信息脱敏、管理员配置、企业合规策略、成本控制面板。

27.18 埋点与数据设计

核心埋点包括 First Feedback Latency、Time to Meaningful Output、Waiting Drop-off Rate、Stop Rate、Expand / Collapse Rate、Mode Switch Rate、Deep Thinking Adoption、Tool Success / Failure、Copy / Export Rate、Deep Mode Conversion、Cost per Successful Task、Silent Tool Failure / Misleading Reasoning。

27.19 验收标准

用户发送后 300ms 内必须出现反馈；普通任务必须快速进入答案；复杂任务必须显示具体等待原因；工具调用必须真实显示；工具失败必须明确提示；思考摘要不能遮挡最终答案；用户必须能停止生成；用户必须能直接看结论；文件分析必须显示读取范围；搜索任务必须显示来源或说明未搜索；高风险任务必须显示边界提示；最终答案必须可复制、可继续、可导出。

27.20 风险控制设计

从 0 设计时必须预留表演式思考、假工具状态、成本失控、隐私泄露等风险机制。默认摘要、不展示无意义长推理、工具状态与后端调用绑定、失败必须显示、自动模式和思考预算上限、敏感字段脱敏、工具日志权限控制。

第 28 章：PRD 草案

28.1 文档信息

项目	内容
产品模块名称	AI 回复过程系统
英文名称	AI Response Process System
功能范围	等待态、思考态、生成态、工具调用态、任务进度、异常状态、最终答案态
适用产品	聊天 AI、AI Agent、AI 办公助手、AI 编程助手、AI 研究助手、AI 工作台
目标用户	普通用户、专业用户、开发者、企业用户、高阶 AI 用户
核心目标	让 AI 回复过程可见、可控、可信
PRD 版本	V1.0 草案

28.2 背景与问题

聊天 AI 的用户体验过去主要围绕最终答案质量展开。但随着推理模型、深度思考、工具调用、联网搜索、文件分析、多模态理解和 Agent 任务的发展，用户等待 AI 回复的过程变得越来越复杂。用户在等待 AI 回复时关心 AI 收到请求了吗、是不是卡住了、到底在做什么、为什么需要这么久、有没有真的搜索 / 读文件 / 跑代码、能不能停止、能不能直接看结论、最终答案是否可信。

因此，本功能的目标不是做 loading 动画，而是建立一套完整的 AI 回复过程系统。

28.3 产品目标

用户目标：确定感、进度感、控制感、信任感。产品目标：将模型推理能力转化为思考摘要，将工具调用能力转化为工具状态卡，将 Agent 执行能力转化为任务进度面板，将深度思考能力转化为快速 / 深度 / 专家模式，将最终生成能力转化为结构化答案与可复用交付物。商业目标：提升首轮体验和新用户留存、降低等待中流失、降低负反馈、支撑深度思考付费、支撑 Agent 多步骤任务、支撑企业审计与权限控制。

28.4 用户画像

用户类型	典型需求	等待态需求
普通用户	快速问答、生活建议、文案改写	简洁、快速、少打扰
学生用户	解题、学习、语言、论文理解	步骤化、解释性、可学习
专业办公用户	报告、分析、方案、总结	结构化、可复制、可导出

用户类型	典型需求	等待态需求
研究用户	资料分析、来源核验、复杂判断	来源、假设、风险、过程摘要
开发者	代码、报错、测试、项目修改	工具日志、文件路径、测试状态
企业用户	内部流程、知识库、审批、审计	权限、日志、人工确认、合规
高阶 AI 用户	工作流、Prompt、Agent、复杂任务	模式控制、工具控制、过程可展开

28.5 使用场景

普通问答应快速进入答案，不展示复杂思考。复杂产品分析应显示任务拆解、分析框架和阶段进度。文件分析必须显示文件读取状态和读取范围。代码任务应展示项目扫描、错误定位、修改和测试状态。Agent 多步骤任务应显示任务计划、执行步骤、中间产物和最终交付。

28.6 功能范围

本期包含基础等待反馈、流式生成、思考摘要、模式切换、工具状态、中断控制、异常提示、最终答案操作。本期不包含完整企业审计后台、全量原始 CoT 展示、多 Agent 协同面板、跨应用自动执行、高风险行业专用合规模块。

28.7 功能结构图

AI 回复过程系统包括状态识别、任务分类、模式系统、展示系统、操作系统五大部分。状态识别包含空闲、已发送、已接收、思考中、工具调用中、生成中、已完成、异常 / 中断。任务分类包含普通问答、写作创作、搜索查询、文件分析、代码任务、多模态任务、深度分析、Agent 任务、高风险任务。模式系统包含自动、快速、深度、专家。展示系统包含状态短句、思考摘要、工具状态卡、任务进度、最终答案。操作系统包含停止、继续、直接给结论、展开过程、重试、导出、反馈。

28.8 状态流转设计

状态流转为 Idle、User Typing、Message Sent、Request Accepted、Task Classified、Mode Selected、Thinking / Retrieving / Tool Calling、Generating、Completed。异常分支包括 Tool Failed、Timeout、User Stopped、Partial Completed、Safety Blocked、Regenerating。每个状态都必须有对应 UI、文案和操作项。

28.9 核心功能需求

编号	功能	优先级	验收标准
FR-01	发送后即时反馈	P0	发送后 300ms 内出现明确反馈
FR-02	任务类型识别	P0	复杂任务不只显示“正在生成”

编号	功能	优先级	验收标准
FR-03	模式切换	P0	自动 / 快速 / 深度 / 专家可用
FR-04	思考摘要展示	P1	思考区不遮挡最终答案
FR-05	工具调用状态卡	P0	工具失败必须明确提示
FR-06	流式生成	P0	长内容先输出结构再扩写
FR-07	中断控制	P0	停止后有后续操作
FR-08	异常状态处理	P0	说明失败步骤、影响范围和可操作方案
FR-09	最终答案操作	P1	答案独立清晰，可复制、重试、导出
FR-10	高风险场景提示	P0	不用长思考制造过度确定感

28.10 非功能需求

类型	要求
性能	普通任务首反馈 ≤ 300ms；首有效内容尽量提前
稳定性	工具状态与真实调用结果一致
可用性	用户可随时停止生成
可访问性	状态文案清晰，不能只依赖动画
安全性	敏感工具日志可脱敏
可扩展性	支持后续接入更多工具和 Agent 流程
可配置性	企业版支持管理员配置默认模式
可审计性	企业任务可保留过程日志

28.11 埋点指标

埋点包括 First Feedback Latency、Time to Meaningful Output、Waiting Drop-off Rate、Stop Rate、Thinking Expand Rate、Direct Answer Click Rate、Tool Success Rate、Silent Tool Failure Rate、Copy Rate、Export Rate、Deep Mode Conversion、Cost per Successful Task、Hallucination Complaint Rate。

28.12 A/B 测试方案

测试思考展示层级：不展示思考、状态短句、折叠思考摘要、默认详细过程。测试模式入口：仅自动、自动 + 深度按钮、快速 / 深度 / 专家三模式、每次发送前强制选择模式。测试工具状态展示：仅显示正在处理、显示工具名称、显示工具状态卡、显示完整工具日志。

28.13 风险与应对

主要风险包括表演式思考、误导性透明、工具状态造假、成本失控、等待过长、隐私暴露、高风险误用、移动端过载。应对策略是默认摘要、不默认长推理、工具状态与后端绑定、自动模式 + budget 上限、首有效内容 + 任务进度、脱敏 + 权限控制、边界提示 + 来源核验、默认折叠 + 只看结论。

28.14 上线策略

MVP 上线发送后即时反馈、正在生成、流式输出、停止生成、简单异常提示、复制 / 重试。V1 增加自动 / 快速 / 深度模式、思考摘要、工具状态卡、直接给结论、文件读取范围、搜索状态、部分失败提示。V2 增加任务计划、Agent 步骤进度、工具日志、人工确认节点、中间产物、导出交付物、任务恢复。
Enterprise 增加权限控制、审计日志、敏感信息脱敏、管理员默认策略、成本面板、合规提示、工具白名单、高风险操作审批。

28.15 验收标准

用户发送后 300ms 内有反馈；普通任务不进入冗长思考；复杂任务说明等待原因；工具调用真实展示；工具失败明确提示；文件分析显示读取范围；搜索任务显示来源或说明未搜索；用户能停止生成；用户能直接看结论；思考摘要不遮挡最终答案；高风险场景显示边界；最终答案可复制、可重试、可继续、可导出。

第 29 章：最终结论

29.1 等待态不是 loading，而是认知交互层

传统软件的等待态主要解决系统是否在加载的问题。聊天 AI 的等待态则要解决更复杂的问题：AI 是否理解任务、是否正在推理、是否调用了工具、是否搜索了资料、是否读取了文件、是否正在生成答案、用户是否可以中断、答案是否可信。

因此，AI 等待态不是一个转圈动画，而是用户理解 AI 工作过程的主要入口。

29.2 六个主流产品代表六种路线

产品	路线	总结
ChatGPT	成熟工作台路线	低干扰、重结果、工具能力强
Gemini	生态助手路线	普通轻量，多模态与 thought summaries 补充
Claude	专业透明路线	折叠式思考、adaptive thinking、专业控制
DeepSeek	强思考展示路线	reasoning_content 前台化，推理感强
Qwen	工程化可切换路线	enable_thinking 控制 thinking / non-thinking
豆包	大众轻量路线	消费端轻量，模型侧支持深度推理

29.3 最优方向是分层透明

未来最成熟的方案不是全部隐藏，也不是全部展示，而是默认简洁、复杂任务展示摘要、专业用户可展开过程、工具调用真实可见、最终答案独立清晰。

29.4 等待态会影响商业化

等待态会影响新用户首轮体验、用户是否愿意继续等、用户是否信任答案、用户是否愿意复制 / 导出、用户是否愿意开启深度模式、用户是否愿意为 Agent 任务付费、企业是否愿意接入工作流。

29.5 最大风险是表演式思考

展示思考可以增强信任，但也可能制造虚假信任。如果产品展示大量推理过程，但过程不忠实、不准确、不必要，用户会误以为 AI 更可靠，长期会伤害品牌。

29.6 Agent 时代，等待态会升级为任务执行面板

当 AI 从回答问题变成执行任务，等待态也会从正在生成升级为任务计划、当前步骤、工具调用、中间结果、人工确认、失败重试、最终交付。

29.7 最终判断

聊天 AI 的等待态 / 思考态 / 生成态，本质上是 AI 产品从会回答走向可信任地完成任务的关键功能。

未来真正优秀的 AI 产品，不是只在最终答案里表现聪明，而是在等待、思考、执行、生成、失败、修正、交付的全过程中，都让用户感到清楚、可控、可信。

附录 A：六个产品竞品对比表

维度	ChatGPT	Gemini	Claude	DeepSeek	Qwen / 通义千问	豆包
产品路线	成熟工作台	生态助手	专业透明	强思考展示	工程化可切换	大众轻量
普通等待态	简洁	轻量	克制	中等	中等	轻量
思考展示	克制 / 摘要	thought summaries	extended / adaptive thinking	reasoning_content 强展示	enable_thinking + reasoning_content	消费端轻，模型侧支持
工具状态	强	中强	中强	中	强，偏开发者	中
用户控制	高	中高	高	中	高	中
专业适配	高	中高	高	中高	高	中
大众友好	高	高	中高	中	中	高
主要优势	稳定、结果好、工具链成熟	多模态与生态	克制透明、专业	推理感强	模式控制清晰	低门槛、高频
主要风险	过程透明度不足	记忆点不强	概念略专业	思考过载	消费端表达需简化	专业感不足
可借鉴点	工作台化	生态任务状态	折叠透明	等待内容化	thinking 模式工程化	大众默认轻量

附录 B：AI 等待态功能清单

B.1 基础功能

功能	优先级
发送后即时反馈	P0
正在生成状态	P0
流式输出	P0
停止生成	P0
重新生成	P0
复制答案	P0

B.2 进阶功能

功能	优先级
思考摘要	P1
展开 / 折叠思考	P1
直接给结论	P1
快速 / 深度模式	P1
工具状态卡	P1
文件读取范围	P1
搜索来源状态	P1
部分失败提示	P1

B.3 专业功能

功能	优先级
专家模式	P2
任务计划	P2
Agent 步骤进度	P2
工具日志	P2
中间产物展示	P2
人工确认节点	P2
导出文档	P2
引用管理	P2

B.4 企业功能

功能	优先级
权限控制	P3
审计日志	P3
敏感信息脱敏	P3
工具白名单	P3
管理员配置	P3
成本控制	P3
合规提示	P3
高风险操作审批	P3

附录 C：术语表

术语	解释
等待态 Waiting State	用户发送消息后、AI 正式输出前的状态反馈
思考态 Thinking State	AI 在回答前进行推理、拆解、规划、验证时的状态
生成态 Generation State	AI 正在输出答案的过程
流式输出 Streaming	答案逐字、逐句、逐段出现
思考摘要 Thought Summary	对模型原始思考过程的摘要式展示
Chain-of-Thought / CoT	模型生成的逐步推理内容
reasoning_content	部分模型 API 中用于返回推理内容的字段
content	模型最终正式回答内容
reasoning effort	控制推理模型投入多少思考资源的参数
thinking budget	控制模型思考 token 或推理预算的参数
enable_thinking	Qwen deep thinking 中用于控制是否开启思考的参数
工具调用 Tool Calling	AI 调用搜索、文件、代码、图片等外部工具
Agent	能执行多步骤任务的 AI 系统
任务面板 Task Panel	展示任务计划、进度、工具和结果的界面
分层透明 Layered Transparency	默认简洁、复杂任务显示摘要、专业用户可展开过程的设计策略
表演式思考	展示大量看似推理但对用户任务帮助有限或可能误导的过程

附录 D：资料来源与可靠性说明

D.1 资料来源分级

等级	类型	用途
A 级	官方文档 / 官方技术报告	确认产品与模型能力
B 级	产品体验观察	分析用户端可见体验
C 级	学术论文 / 研究资料	分析 CoT、reasoning、thinking 风险与趋势
D 级	产品分析推断	基于事实进行策略判断

D.2 主要资料来源

- OpenAI 官方 reasoning 文档：用于确认 reasoning models、reasoning effort、reasoning tokens 与 reasoning summaries 相关能力。
- Google Gemini API Thinking 文档：用于确认 thought summaries、thinking levels / budgets 相关能力。
- Anthropic Claude Extended Thinking / Adaptive Thinking 文档：用于确认 Claude 的 extended thinking、adaptive thinking、thinking budget 和 summarized thinking 机制。
- DeepSeek 官方 thinking mode / reasoner 文档：用于确认 reasoning_content 与 content 分离、先思考再回答的能力。
- 阿里云 Model Studio / Qwen deep thinking 文档：用于确认 enable_thinking、hybrid thinking mode、reasoning_content / content 分离。
- ByteDance Seed / BytePlus / 火山引擎相关资料：用于确认豆包 / Seed 模型体系的 thinking before responding、多模态推理和 reasoning_content 能力。
- Nielsen Norman Group 等 UX 资料：用于确认进度反馈、等待动画、骨架屏与用户感知等待相关理论。
- Chain-of-Thought 忠实性相关研究：用于说明外显思考不等同于模型真实内部机制，避免误导性透明。

D.3 可靠性说明

本报告中的事实性能力描述优先基于官方文档；用户端体验观察可能因地区、会员、模型版本、Web / App 端、实验开关不同而变化；对产品策略的解释属于分析判断，不代表厂商官方立场；对未来趋势的判断属于基于当前行业能力演进的产品推断。

报告收束

这份报告最终定位为一份围绕“聊天 AI 等待态 / 思考态 / 生成态”的专业产品功能研究报告，兼具竞品分析、交互分析、商业分析、方法论提炼和 PRD 设计价值。

它的最终结论不是“哪个 AI 产品最好”，而是：**AI 产品进入推理模型和 Agent 时代后，等待态已经成为用户理解、控制并信任 AI 的核心产品层。**

如果未来要从零创造一个新的聊天 AI 产品，等待态不应再被当作界面尾部的小细节，而应该从立项初期就被设计为完整系统：任务识别、模式选择、状态反馈、思考摘要、工具展示、流式生成、用户控制、异常处理、最终答案、数据指标、风险治理。

真正优秀的 AI 产品，不会让用户只是“等 AI 回答”，而是让用户参与、理解并控制 AI 完成任务的全过程。