

肖正午

AI产品经理 Agent Eval /
Memory / AI-Native Workflow

✉ xzw13537504508@gmail.com

🔗 [Github](#)

☎ 13537504508

👤 [个人网站](#)

教育经历

加州大学圣地亚哥分校 UCSD (USNEWS 2027 全球排名23) 统计学+数据科学 · GPA 3.7/4.0 2023/06 - 2027/03

相关课程：人工智能、数据科学实践、数据分析与推断、概率论、数理统计、随机过程、线性代数、计算机科学基础

荣誉奖项：TRELS Scholar | 凭借 AI 职位薪资分析项目获选，基于1500条招聘数据完成多元回归与假设检验，识别 AI岗位薪资的关键影响因素

专业排名：前15%

核心能力

- AI Agent 产品设计：**理解 Agent Memory、Schedule、Tool Calling、上下文管理及 Skill 工作流等核心机制，能够从用户场景出发，将底层能力转化为产品流程、功能规则、能力边界、异常状态和验收标准。
- Agent Evaluation 与策略优化：**能够从用户最终任务结果出发定义“好的 Agent”，建立固定测试集和产品侧评测指标，对不同版本进行量化比较，并基于 Bad Case 区分模型理解、Prompt、工具调用和产品规则问题，形成持续调优与回归验证闭环。
- 产品设计与项目推进：**具备需求洞察、竞品调研、需求分析、MVP 规划、PRD、信息架构、交互原型、研发沟通、功能验收和上线迭代经验，能够独立推动需求从问题发现到方案落地。
- AI-native 产品交付：**熟练使用 Codex、WorkBuddy 等 Coding Agent，将产品需求快速转化为结构化 PRD 和可运行 HTML / Web 原型；能够设计可复用的 Prompt 与 Skill 工作流，提升需求理解、产品交付和团队协作效率。
- 数据分析与实验验证：**具备统计学与数据科学背景，熟悉 SQL、Python、Pandas 和 Excel，能够围绕产品目标定义指标及统计口径，并通过版本对比、漏斗分析、用户行为分析和 A/B Test 验证产品方案。
- 工具与语言：**Figma、GitHub、GitLab、Jupyter Notebook、Codex、WorkBuddy；具有三年海外留学经历，可适应全英文资料阅读、产品调研和工作沟通。

工作经验

滴普科技 AI产品经理实习生 2026/05 - 至今

参与面向海外用户的 AI Agent 产品建设，负责用户场景分析、需求拆解、产品方案设计、PRD、交互原型、研发协同、功能验收及上线迭代，推动需求从问题发现到产品落地的完整闭环。

- 累计完成 **50+ 个真实产品需求**的方案设计与原型交付，其中 **40+项**进入内测环境，推动需求完成评审、研发、测试及上线，覆盖约 **100 名**邀请制内测用户，包含 Agent Memory、Schedule、Files、Skills、输入交互及第三方渠道等核心模块。
- 将模型、工具调用、长期记忆及自动化任务等 Agent 能力，转化为面向用户的功能流程、产品规则、异常状态和验收标准，并持续跟进研发实现与版本迭代。
- 使用 Codex、WorkBuddy 将需求转化为可运行 Web 原型，提前验证核心流程、业务规则和异常场景，支持产品、设计与研发在评审阶段快速对齐方案。

项目经历

碎片花园 | Vibe Coding 产品 个人项目 2026/06 - 至今

内容：

针对日常灵感、收藏链接、图片资料和未成熟想法分散在备忘录、聊天软件及浏览器收藏夹中，记录后难以重新找到、继续补充和产生实际价值的问题，从 0 到 1 设计并开发“碎片花园”。产品以植物生长作为内容状态的视觉隐喻，将文字、链接和图片整理为可持续补充的个人笔记，并通过“记录—回访—追加—发芽”的核心循环，让历史内容被重新发现、重新组织，并生长为新的完整想法。

职责：

- 需求洞察与产品定位：**从个人高频碎片记录场景出发，分析传统笔记工具依赖提前分类、整理成本较高，以及内容保存后缺少回访动力等问题，将产品定位为“轻量记录与再次发现工具”，而非复杂知识库或任务管理产品。
- 核心产品机制设计：**将每条笔记映射为花园中的植物，由植物视觉表现内容的成长和活跃状态；设计“思维发芽”能力，读取当前主体笔记及全部追加内容，由 AI 重新组织并适度发散为一篇新的完整笔记，经用户预览确认后生成新的卡片与植物。
- Vibe Coding 与工程落地：**使用 Codex 完成产品文档、信息架构、交互方案、技术约束、代码实现和测试回归，基于 Next.js、React、Phaser、IndexedDB 搭建本地优先的可运行 Web MVP，并通过阶段化提交、单元测试和 E2E 测试控制 AI Coding 的交付质量。

结果：

- 连续 **14 天**个人真实使用中累计沉淀 **80+ 条**文字、链接和图片碎片，初步验证“记录—回访—再利用”的核心产品闭环。
- 将链接和图片从手动摘录、整理到形成结构化笔记的平均个人内容处理效率提升约 **65%**。
- 项目开源后获得 **40+** GitHub Stars，并根据使用反馈完成内容详情收敛、链接与图片延迟创建、追加笔记和思维发芽等核心流程迭代。

内容：
针对 Schedule Agent 在信息缺失判断、跨时区时间理解，以及“回复创建成功但任务实际未正确落地”等场景中的不稳定问题，建立产品侧 Evaluation 体系，对不同版本 Agent 的真实任务表现进行量化评估。

职责：

- 定义好的 Agent：** 从用户能否最终获得正确、可执行的定时任务出发，定义端到端任务成功标准，而不是仅判断 Agent 回复是否自然或合理。
- 建立固定评测集：** 设计覆盖信息缺失、跨时区、周期任务、多轮修改和工具异常等典型与边界场景的测试用例，保证不同 Agent 版本能够在同一口径下比较。
- 设计产品侧指标：** 以端到端任务成功率作为核心指标，以虚假成功率作为风险指标，综合判断 Agent 是否正确追问、理解时间与时区、生成完整参数并实际创建任务。
- Bad Case 归因：** 将失败案例归因至模型理解、Prompt 策略、工具调用参数或产品规则，明确问题优先级及对应优化方向。
- 持续调优闭环：** 对不同版本 Agent 进行量化评估，推动相关负责人完成策略调整后，使用同一批用例重新测试，并将历史问题沉淀为回归用例。

结果：

- 在 50 条固定测试用例中，对不同版本 Agent 进行量化评估并推动策略优化，使端到端任务成功率由 68% 提升至 82%。
- 将 Agent 表示创建成功、但任务实际未正确落地的虚假成功率由 14% 降至 6%。

内容：
产品是单窗口的长期陪伴型 Agent，用户无法通过新建对话重置上下文。随着对话累计至数千甚至数万轮，Agent 如何判断哪些信息值得长期保留、实际记住了什么、记忆是否错误，以及错误记忆如何被用户发现和纠正，都会直接影响长期使用体验。
基于这一产品风险，提出长期记忆机制问题，并负责调研 Agent Memory 底层机制、设计 Memory Tab 产品方案，以及推动用户记忆管理能力上线。

职责：

- 拆解记忆机制：** 系统调研长短期记忆、记忆写入、存储、召回、更新、删除和冲突处理机制，理解底层能力如何影响用户侧产品体验。
- 产品能力转化：** 将底层 Memory 机制转化为用户可查看、编辑、删除和确认的 Memory Tab，设计“记忆生成—用户查看—用户纠正—记忆更新—后续召回”的完整产品闭环。
- 规则与边界设计：** 明确用户修改后的记忆更新规则、删除后的召回边界、冲突信息处理及多轮对话下的记忆管理逻辑。
- 产品侧效果评估：** 围绕记忆写入、更新、删除和冲突等场景定义端到端成功标准，验证前台展示、底层记忆更新及后续 Agent 回复是否保持一致。

结果：

- 推动 Memory Tab 从机制调研、产品设计、研发评审到正式上线，首轮实际测试覆盖约 100 名用户。
- 在 30 条长期记忆固定测试用例中，使记忆写入、更新、删除及后续召回的闭环成功率由 67.5% 提升至 85%。
- 将用户修改或删除记忆后，Agent 仍继续使用旧信息的误召回率由 18% 降至 5%。

内容：
针对产品经理使用通用 AI 完成需求分析和原型制作时，需要反复解释业务背景、PRD 与原型容易割裂，以及需求理解偏差导致评审修改和研发返工的问题，基于 Codex 设计 PRD + 可运行原型生成 Skill。

职责：

- workflow 设计：** 搭建“需求输入—理解确认—PRD 生成—原型生成—反馈修改”的连续工作流，将需求分析、产品文档和原型交付统一在同一份结构化上下文中。
- 需求确认机制：** 在生成 PRD 和原型前，由 Skill 输出对产品目标、用户场景、业务规则、功能范围和参考样式的当前理解，由产品经理确认后再进入下一阶段。
- 双模式支持：** 支持从零生成新需求，以及基于现有 HTML 原型补充和修改需求；修改前先确认影响范围，再同步更新 PRD 与对应原型。
- 团队工作流沉淀：** 将个人使用流程沉淀为团队可复用的 Skill，统一需求理解、PRD 输出和原型交付方式。

结果：

- 将单个需求从收到相对完整说明，到交付 PRD 与首版可运行原型的平均周期，由约 5 小时缩短至 2.5 小时。
- 将因需求理解偏差或 PRD—原型不一致导致的重大返工率，由 30% 降至 18%。
- 在相近需求数量与复杂度下，使团队整体需求交付效率提升约 25%。