

AI 驱动的产品创设与实现流程方法论

The Council 团队 | v2.0 (Methodology Optimized)

2025-12-16

AI 驱动的产品创设与实现流程方法论

作者: The Council 团队 (1名碳基包头 & 他的硅基施工队)
版本: v2.0 (Methodology Optimized)
日期: 2025-12-16
案例: The Council (AI Native Workspace)

引言

本文档记录了一种创新的 AI 原生产品开发方法论——将人类的创造力与 AI 的分析能力深度融合，通过结构化的对抗式审查、智能工作流和严格的质量门禁，实现从 **模糊 Idea** 到 **成熟产品** 的高效转化。

这是一种“人机共创 (Human-AI Co-Creation)”的全新范式：

- 人类: 负责愿景输入、价值判断和最终决策。
- AI (Dialecta): 负责多角度辩论、漏洞挖掘和方案优化。
- AI (PMO): 负责规格审计、一致性检查和进度追踪。

全流程可视化



核心工具链: Dialecta

仓库: github.com/hrygo/dialecta

Dialecta 是我们自主研发的本流程核心引擎，一个多角色 AI 辩论系统。它克服了单一 LLM 的“盲目顺从”和“幻觉”问题，通过对抗产生真知。

核心机制

- 多角色对抗 (Multi-Persona Debate)
 - Pro (正方): 全力维护现有方案，寻找亮点。
 - Con (反方): 魔鬼代言人，无情挖掘漏洞和风险。
 - Judge (裁决): 上帝视角，评估双方论点，给出公正评分和建议。

2. 智能循环 (Auto-Optimization Loop)

- 系统可配置 max_loops，在无人值守的情况下自动进行“辩论 → 修改 → 再辩论”的迭代，直到评分达标。

3. 判例法记忆 (Common Law Memory)

- Dialecta 会读取历史裁决 (history_summary.md)，防止在新一轮修改中重犯错误或发生“Flip-Flop”(反复横跳)。

Phase 1: 创意孵化 (Idea)

1.1 方法：启发式交互

与 AI 进行苏格拉底式对话，利用 5W1H 和 反向推演 技巧，将“我想做...”这种模糊的原始诉求，扩展为包含核心功能、用户画像和痛点的 PRD 草案。

1.2 产出物

- PRD Draft: 包含基础功能列表和核心逻辑的结构化草稿。

Phase 2: PRD 对抗式审查 (Dialecta Process)

2.1 目标

通过高强度的 AI 辩论，在写一行代码前，发现逻辑漏洞和产品缺陷。

2.2 流程

- AI 立论: 正方阐述方案价值。
- AI 驳论: 反方攻击方案弱点 (如: 成本失控、隐私泄露、用户疲劳)。
- AI 裁决: Judge 给出 Verdict (通过/条件通过/驳回) 和修改建议。
- 智能循环: 系统自动应用修改建议，并进行下一轮辩论，直至评分达标。

2.3 产出物

- Final PRD: 经过多轮洗礼，逻辑严密的文档。
- Debate Reports: 详细的辩论记录，作为决策依据。

Phase 3: 技术设计 (Technical Design)

3.1 双轨并行设计

- TDD (Technical Design Doc): 架构、接口、数据结构。
- Design Draft: UI/UX 布局、交互流程、视觉规范。

3.2 设计审查

设计方案同样需要经过 Dialecta 审查。AI 会针对设计蓝图中的交互逻辑、安全隐患进行对抗性推演，确保设计方案不仅美观，而且健壮。

3.3 产出物

- TDD & Design Draft (Production Ready)

Phase 4: 研发规划 (Planning) & AI PMO 审计

4.1 规格生成 (Spec Generation)

AI 架构师基于技术蓝图 (TDD & Design Draft)，将其细化并衍生为海量且高精度的初始 Spec 文档 (Content Expansion)。* 每份 Spec 都包含明确的接口定义、测试用例要求和验收标准，为开发提供原子化的执行指令。

4.2 AI PMO 审计 (The AI Auditor)

这是一个完全由 AI 扮演 PMO 执行的闭环验证过程，其深度远超传统的人工 Review。

4.3 审计逻辑闭环

- 全局读取 (Full Context Audit): AI 同时读取 PRD, TDD, Design Draft 以及刚生成的所有 Specs。
- 差距分析 (Gap Analysis):
 - 构建“需求-实现”矩阵，逐条核对 PRD 功能点 (Feature ID) 是否在 Spec 中有对应的验收标准。
- 自动增补 (Auto-Remediation):
 - 一旦发现 Gap，AI PMO 自动触发 Create Spec 动作，补齐缺失的规格。
- 追溯性锁定 (Traceability Lock):
 - 确保 Spec 没有违背 TDD 定义的架构原则 (如: 前端状态必须遵循 Zustand Store 定义)。

4.4 产出物

- Development Plan
- Specs 集合 (完整覆盖)
- Audit Report (包含 Traceability Matrix)

Phase 5: 冲刺执行 (Execution)

5.1 TDD 开发模式

开发者依据 Spec 和 Test Case 进行编码: 红 (Fail) → 绿 (Pass) → 蓝 (Refactor)。

5.2 验收

每个 Sprint 结束，对照 Spec 中的 验收标准 (Acceptance Criteria) 进行逐项确认。

附录：实战案例 - The Council (48小时从概念到落地)

The Council 项目的诞生是本方法论的极致验证。通过 Git Log 追溯，我们复原了这惊人的 48 小时 (Dec 15 - Dec 16) 突击战。

1. 概览 (Overview)

- 时间窗口: 2024-12-15 15:00 (v0.1.0) → 2024-12-16 20:00 (v0.6.1)
- 总代码提交: 20+ Commits (涵盖 Core Engine, Memory, API)
- 核心产出: 完整的 PRD, Design, TDD, 32 份 Specs, 以及可运行的 MVP。

2. 演进战报 (The Battle Report)

Dec 15, 12:00 PM - PRD Debate

在项目初期，AI 成功拦截了一场灾难性的技术选型。

2.1 目标

通过高强度的 AI 辩论，在写一行代码前，发现逻辑漏洞和产品缺陷。

2.2 流程

- AI 立论: 正方阐述方案价值。
- AI 驳论: 反方攻击方案弱点 (如: 成本失控、隐私泄露、用户疲劳)。
- AI 裁决: Judge 给出 Verdict (通过/条件通过/驳回) 和修改建议。
- 智能循环: 系统自动应用修改建议，并进行下一轮辩论，直至评分达标。

2.3 产出物

- Final PRD: 经过多轮洗礼，逻辑严密的文档。
- Debate Reports: 详细的辩论记录，作为决策依据。

Phase 3: 技术设计 (Technical Design)

3.1 双轨并行设计

- TDD (Technical Design Doc): 架构、接口、数据结构。
- Design Draft: UI/UX 布局、交互流程、视觉规范。

3.2 设计审查

设计方案同样需要经过 Dialecta 审查。AI 会针对设计蓝图中的交互逻辑、安全隐患进行对抗性推演，确保设计方案不仅美观，而且健壮。

3.3 产出物

- TDD & Design Draft (Production Ready)

Phase 4: 研发规划 (Planning) & AI PMO 审计

4.1 规格生成 (Spec Generation)

AI 架构师基于技术蓝图 (TDD & Design Draft)，将其细化并衍生为海量且高精度的初始 Spec 文档 (Content Expansion)。* 每份 Spec 都包含明确的接口定义、测试用例要求和验收标准，为开发提供原子化的执行指令。

4.2 AI PMO 审计 (The AI Auditor)

这是一个完全由 AI 扮演 PMO 执行的闭环验证过程，其深度远超传统的人工 Review。

4.3 审计逻辑闭环

- 全局读取 (Full Context Audit): AI 同时读取 PRD, TDD, Design Draft 以及刚生成的所有 Specs。
- 差距分析 (Gap Analysis):
 - 构建“需求-实现”矩阵，逐条核对 PRD 功能点 (Feature ID) 是否在 Spec 中有对应的验收标准。
- 自动增补 (Auto-Remediation):
 - 一旦发现 Gap，AI PMO 自动触发 Create Spec 动作，补齐缺失的规格。
- 追溯性锁定 (Traceability Lock):
 - 确保 Spec 没有违背 TDD 定义的架构原则 (如: 前端状态必须遵循 Zustand Store 定义)。

4.4 产出物

- Development Plan
- Specs 集合 (完整覆盖)
- Audit Report (包含 Traceability Matrix)

Phase 5: 冲刺执行 (Execution)

5.1 TDD 开发模式

开发者依据 Spec 和 Test Case 进行编码: 红 (Fail) → 绿 (Pass) → 蓝 (Refactor)。

5.2 验收

每个 Sprint 结束，对照 Spec 中的 验收标准 (Acceptance Criteria) 进行逐项确认。

附录：实战案例 - The Council (48小时从概念到落地)

The Council 项目的诞生是本方法论的极致验证。通过 Git Log 追溯，我们复原了这惊人的 48 小时 (Dec 15 - Dec 16) 突击战。

1. 概览 (Overview)

- 时间窗口: 2024-12-15 15:00 (v0.1.0) → 2024-12-16 20:00 (v0.6.1)
- 总代码提交: 20+ Commits (涵盖 Core Engine, Memory, API)
- 核心产出: 完整的 PRD, Design, TDD, 32 份 Specs, 以及可运行的 MVP。

2. 演进战报 (The Battle Report)

Dec 15, 12:00 PM - PRD Debate

在项目初期，AI 成功拦截了一场灾难性的技术选型。

2.1 目标

通过高强度的 AI 辩论，在写一行代码前，发现逻辑漏洞和产品缺陷。

2.2 流程

- AI 立论: 正方阐述方案价值。
- AI 驳论: 反方攻击方案弱点 (如: 成本失控、隐私泄露、用户疲劳)。
- AI 裁决: Judge 给出 Verdict (通过/条件通过/驳回) 和修改建议。
- 智能循环: 系统自动应用修改建议，并进行下一轮辩论，直至评分达标。

2.3 产出物

- Final PRD: 经过多轮洗礼，逻辑严密的文档。
- Debate Reports: 详细的辩论记录，作为决策依据。

Phase 3: 技术设计 (Technical Design)

3.1 双轨并行设计

- TDD (Technical Design Doc): 架构、接口、数据结构。
- Design Draft: UI/UX 布局、交互流程、视觉规范。

3.2 设计审查

设计方案同样需要经过 Dialecta 审查。AI 会针对设计蓝图中的交互逻辑、安全隐患进行对抗性推演，确保设计方案不仅美观，而且健壮。

3.3 产出物

- TDD & Design Draft (Production Ready)

Phase 4: 研发规划 (Planning) & AI PMO 审计

4.1 规格生成 (Spec Generation)

AI 架构师基于技术蓝图 (TDD & Design Draft)，将其细化并衍生为海量且高精度的初始 Spec 文档 (Content Expansion)。* 每份 Spec 都包含明确的接口定义、测试用例要求和验收标准，为开发提供原子化的执行指令。

4.2 AI PMO 审计 (The AI Auditor)

这是一个完全由 AI 扮演 PMO 执行的闭环验证过程，其深度远超传统的人工 Review。