Anthropic 内部团队如何使用 Claude Code

变革工作流：从开发人员到非技术人员的实战案例

SYS.OPT.v4.1

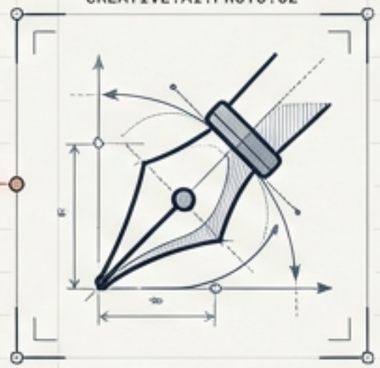
ENGINEERING (工程)

JetBrains Mono

TECHNICAL SPEC25
15C7E05 3YPE: 005.v83

Automated Code
Generation & Refactoring
[FUNC.DEV.01]

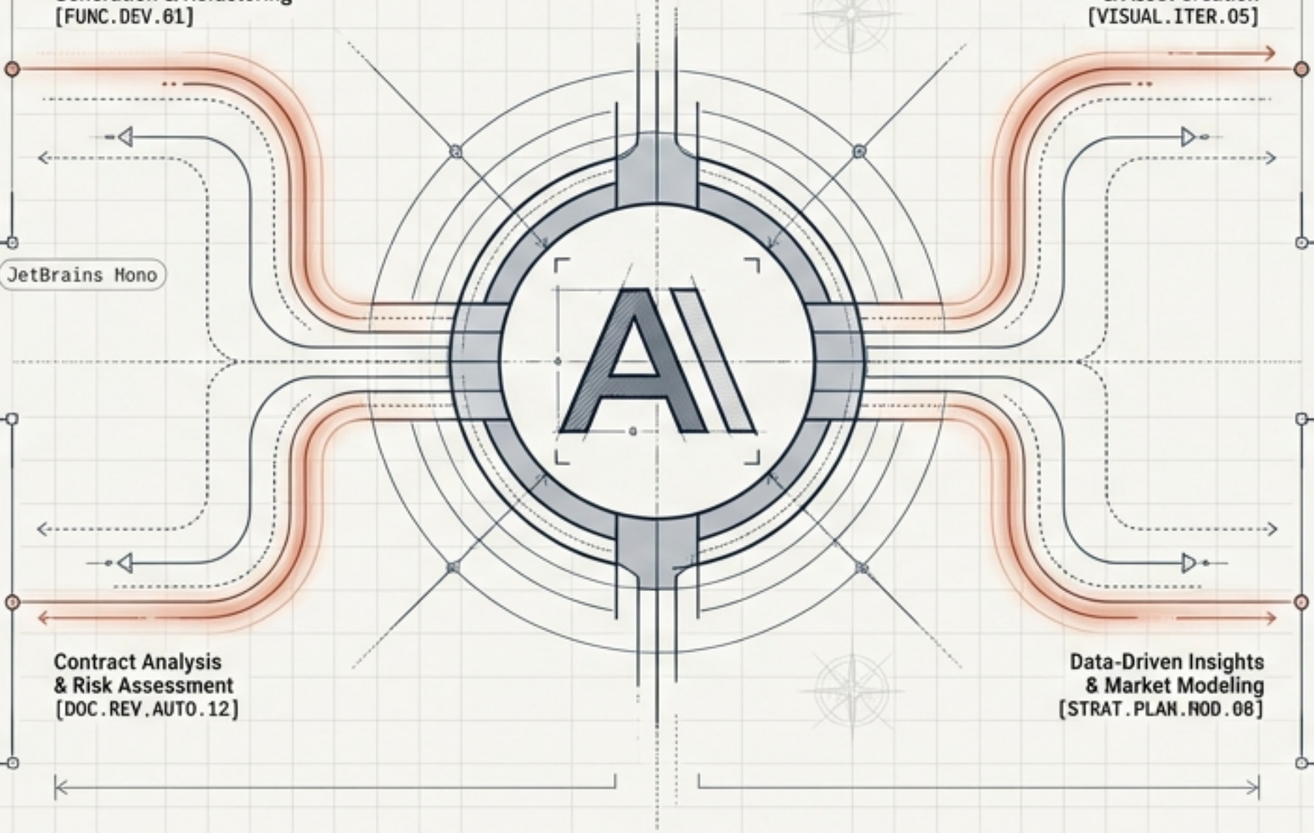
CREATIVE.AI.PROTO.02



DESIGN (设计)

TECHKZCAL SPEC. 32
SPECS 487.58N P5.8.86%

Rapid UI Prototyping
& Asset Creation
[VISUAL.ITER.05]



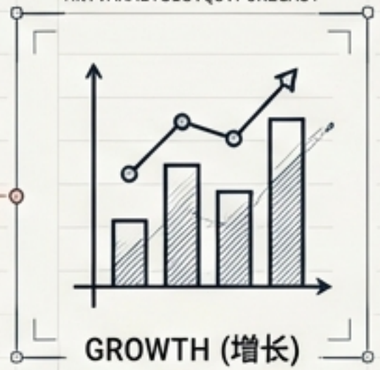
REG.CORPLIANCE.FRAREWORK.8

LEGAL (法务)

DATA COR095£ A8 115A
[0STEN .18B-88-08*]
t6OCK?COBT0: 31.28.15

Contract Analysis
& Risk Assessment
[DOC.REV.AUTO.12]

RKT.AKALYSIS.Q3.FORECAST



GROWTH (增长)

RETRICS AKAL1S1S
NET825 ARG. 2:.S2S
NETRIC NETR6S: 6.82%

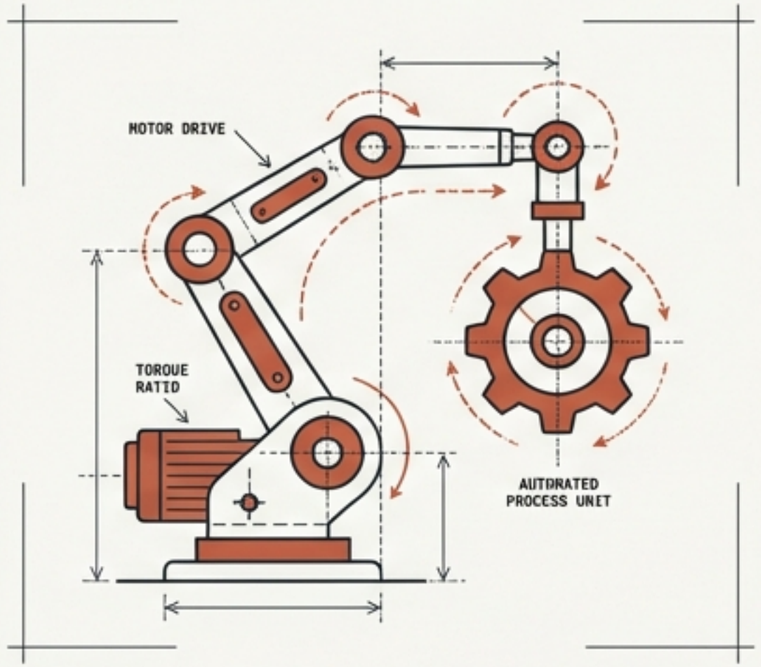
Data-Driven Insights
& Market Modeling
[STRAT.PLAN.R0D.08]

NotebookLM

不仅仅是代码助手：工作方式的全面转型

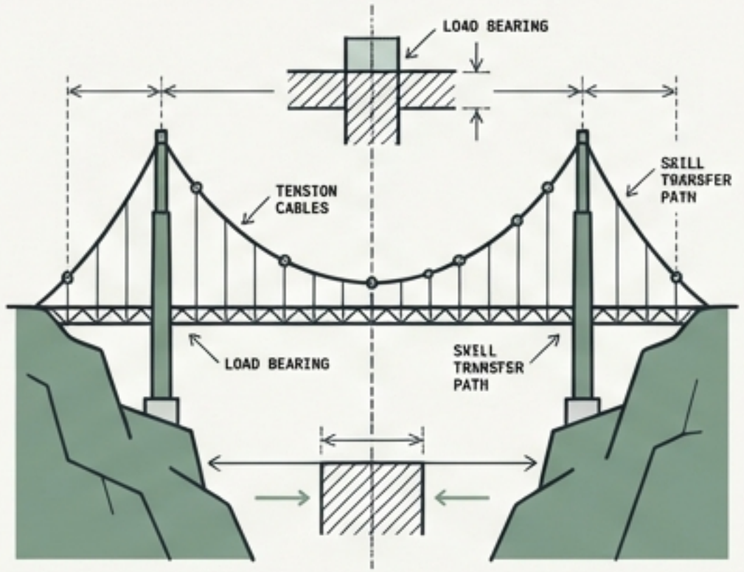
“基于对内部深度用户的访谈整理”

Anthropic 的内部团队正在利用 Claude Code 改造工作流程，使开发人员和非技术人员都能处理复杂项目，并填补此前限制生产力的技能空白。



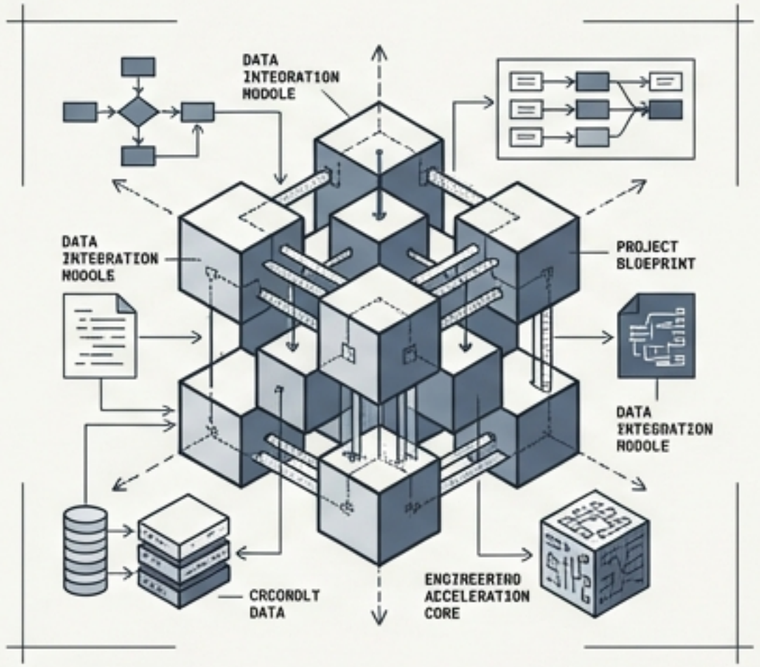
自动化 (Automation)

处理重复性任务



填补技能缺口 (Bridging Skill Gaps)

赋能非技术人员



复杂项目 (Complex Projects)

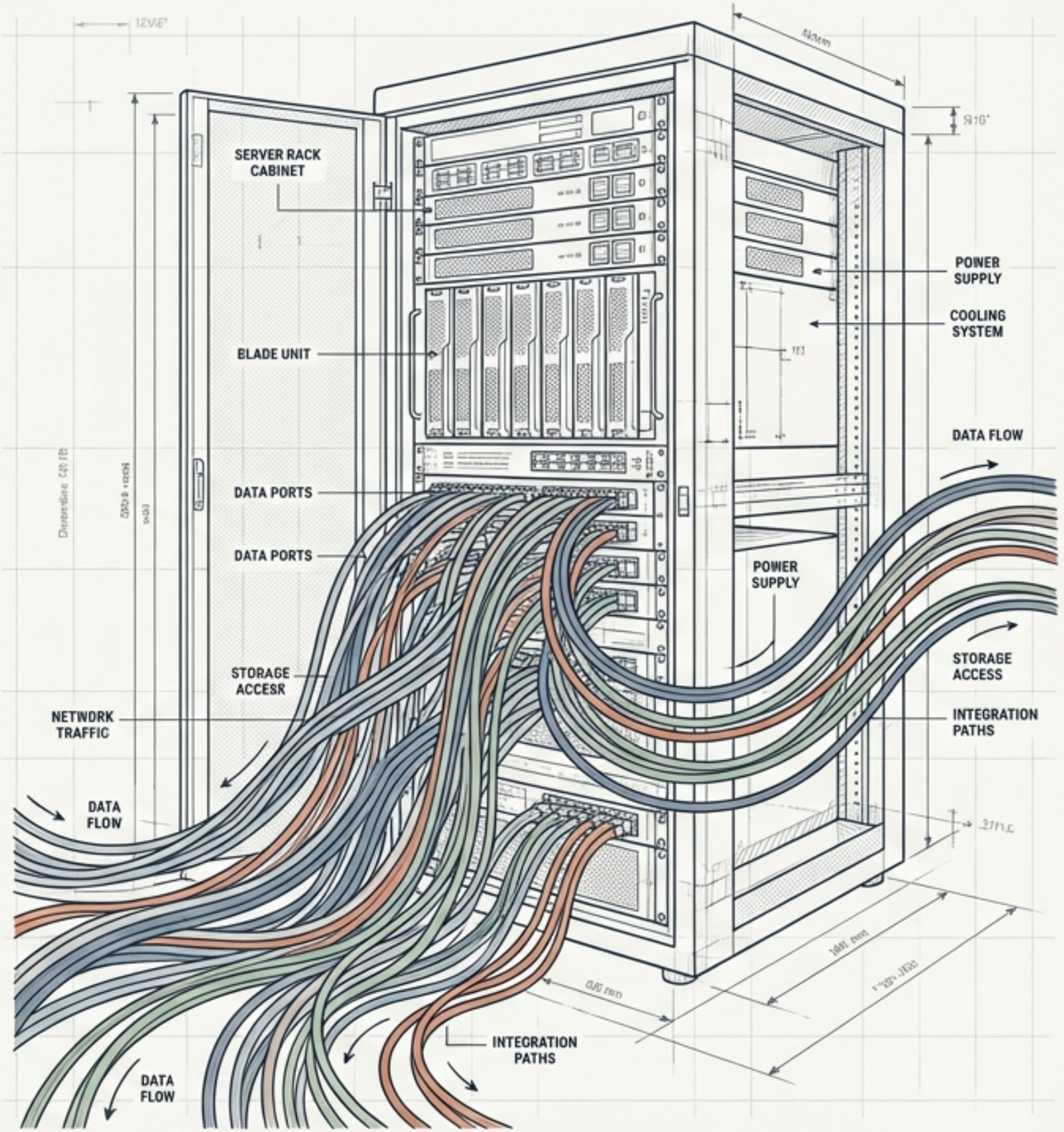
加速工程开发

NotebookLM

PART I

第一部分：数据与基础设施

自动化核心流程与民主化数据访问



NotebookLM

Data Infrastructure: 自动化运维与赋能财务团队

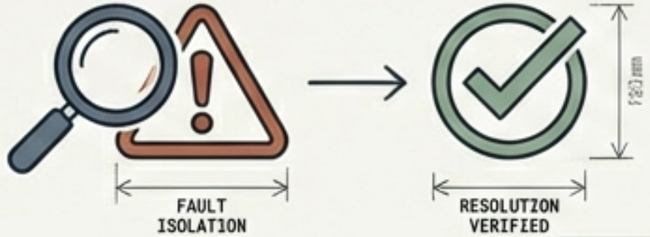
PANEL 1: OPS AUTONATION

Kubernetes Debugging

Challenge (痛点)：集群故障，Pod 无法调度，通常需要网络专家介入。

Action (行动)：将仪表盘截图发送给 Claude Code -> Claude 引导浏览 Google Cloud UI -> 发现 IP 地址耗尽 -> Claude 提供创建新 IP 池的精确命令。

Result (结果)：无需专家介入即可解决。

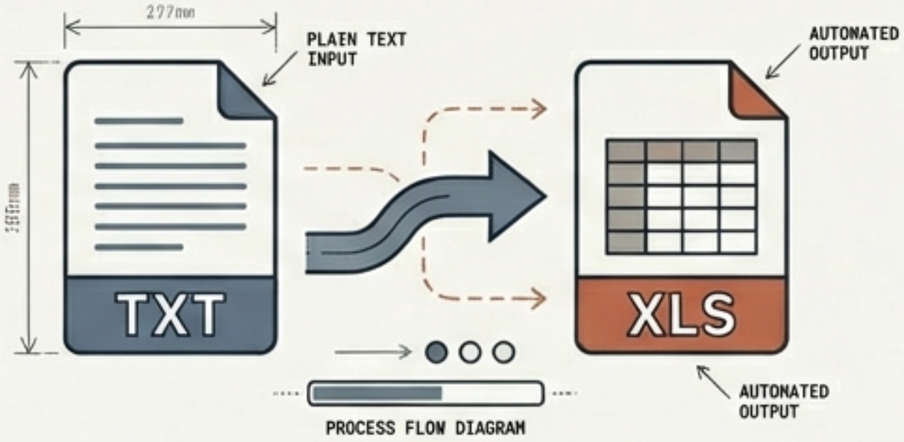


PANEL 2: FINANCE ENABLEMENT

Finance Workflows

Context: 针对无代码经验的财务人员。

Workflow: 财务人员编写纯文本（Plain text）描述需求 -> Claude Code 自动执行“查询此仪表盘，运行这些查询，生成 Excel'）。



NotebookLM

Data Infrastructure: 持续改进与团队影响

Impact

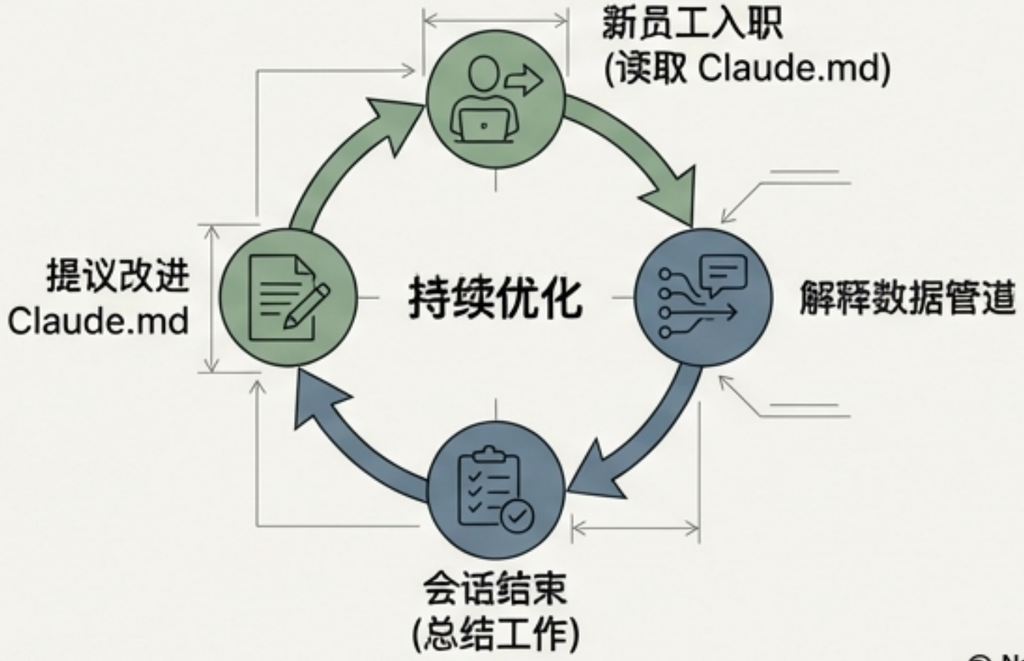
- **Self-Service**: 财务团队无需编码经验即可独立执行复杂工作流。
- **Context Retention**: 在多个代码库实例之间切换时，保持完整的上下文记忆（No context loss）。

Expert Tip

专家级建议 (Expert Tip): Write detailed `Claude.md` files.

建议使用 MCP 服务器而非 CLI 处理敏感数据以确保安全。

The Documentation Loop

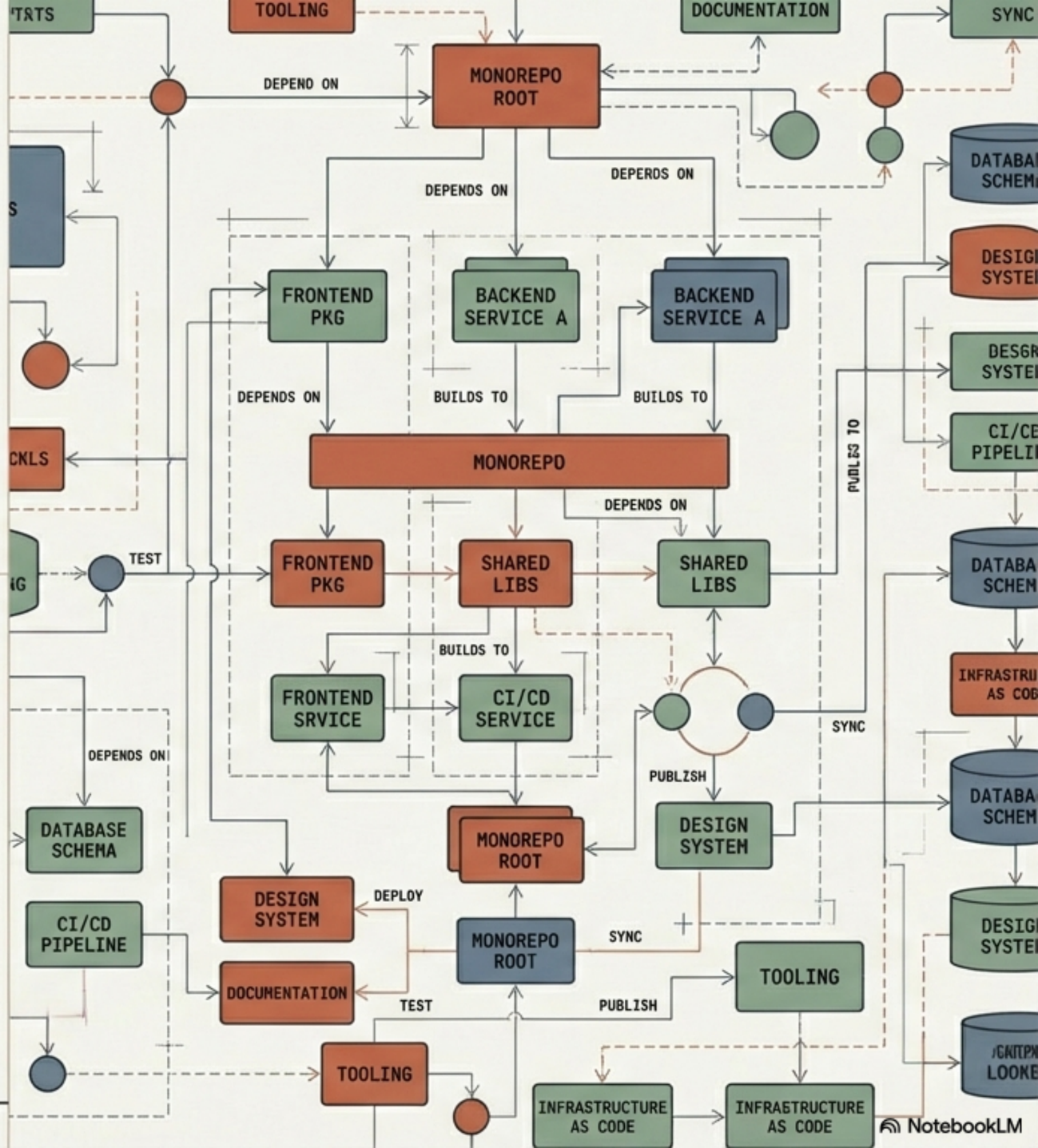


NotebookLM

PART II

第二部分：产品开发与工程

构建自给自足的开发循环



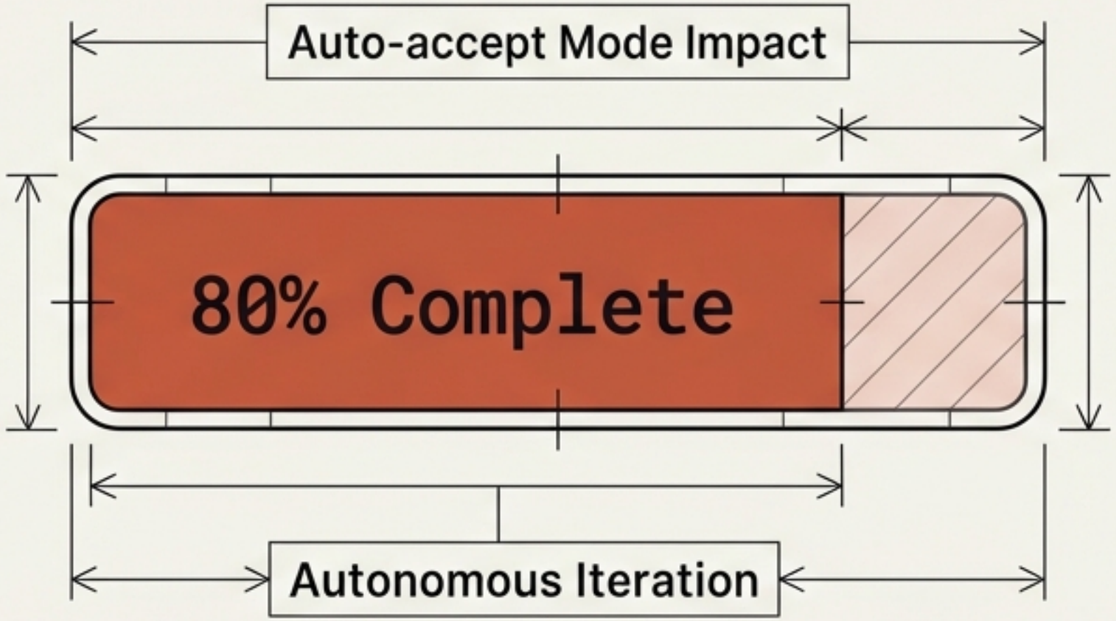
Product Development: 快速原型与核心功能开发

Feature Spotlight: Auto-accept Mode（自动接受模式）

- **Workflow**: 启用 Shift+Tab -> Claude 编写代码、运行测试、持续迭代 -> 工程师审查 **80% 完成** 的解决方案。
- **Condition**: 从干净的 git 状态开始，定期提交 checkpoint。

**Use Case: Synchronous Coding（同步编码）

- 用于核心业务逻辑。
- 工程师提供详细 Prompt -> 实时监控代码质量与风格 -> Claude 处理重复性编码工作。

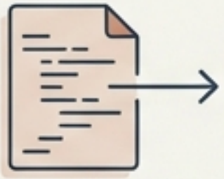


NotebookLM

Product Development: 影响与建议

Case Study: Vim Mode

异步项目，非优先级功能。**70%** 的最终实现代码由 Claude 自主完成。



70%

Impact Areas

- **Test Generation**: 实现功能后自动编写全面测试 –> 利用 GitHub Actions 处理 PR 评论。
- **Codebase Exploration**: 在单体仓库（Monorepo）中快速理解陌生代码，无需等待同事回复 Slack。

Top Tip: Create self-sufficient loops (建立自给自足的循环)

要求 Claude 在写代码前先生成测试，并让它运行构建和 Lint 检查来验证自己的工作。

NotebookLM

Security Engineering: 缩短事件响应时间

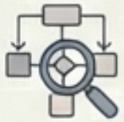
15 min

15 min -> 5 min

手动代码扫描 vs Claude 诊断

5 min

Core Use Cases



- **Infrastructure Debugging:** 输入堆栈跟踪 (Stack traces) -> Claude 追踪控制流 -> 快速定位生产环境问题。
- **Terraform Review:** 将 Terraform 计划复制给 Claude -> 询问 “What’s this going to do? Am I going to regret regret this?’ -> 加速安全审批。

Pro Tip

Let Claude talk first (让 Claude 先说)

告诉 Claude “边做边提交” (commit your work as you go)，而不是只问代码片段的问题。

NotebookLM

Inference: 跨越机器学习的知识鸿沟

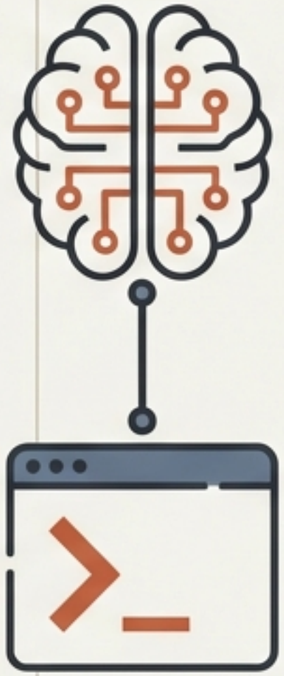
80% Research time reduced（减少的研究时间）

Knowledge Bridging

- **Concept Explanation:** 解释模型特定函数和设置 (1 hour Google search -> 10-20 mins Claude explanation)。
- **Cross-language Translation:** 描述测谳图 -> Claude 用 Rust 编写逻辑-> 无需学习新语言。

Kubernetes Management

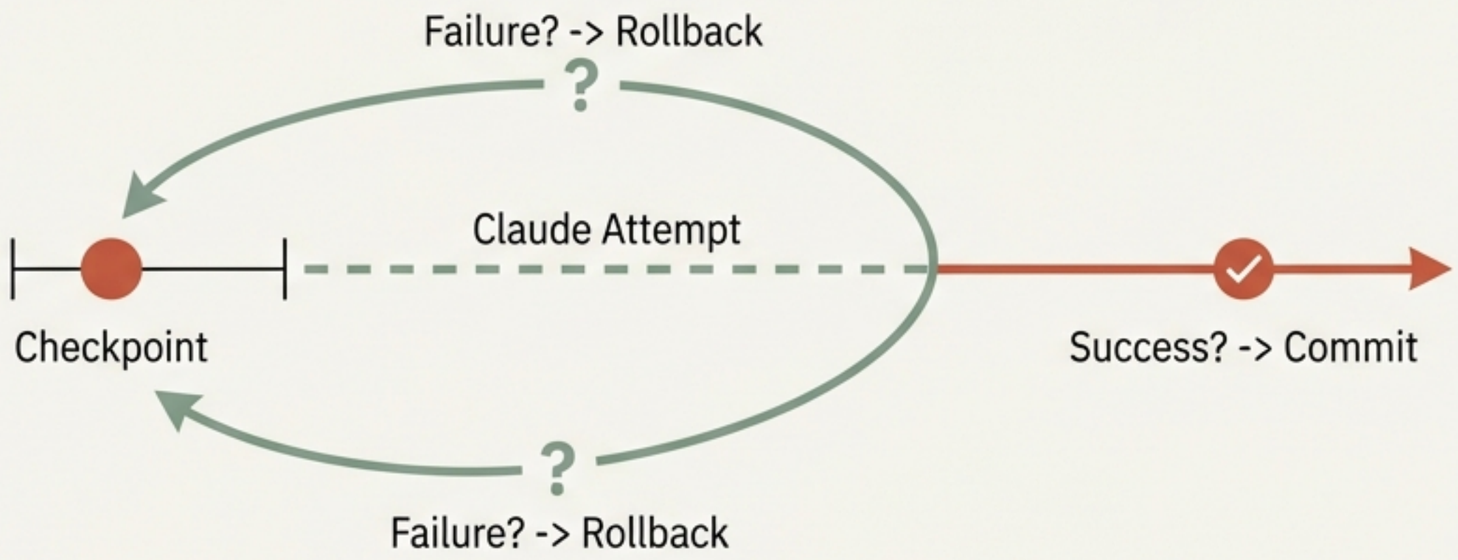
- 询问 “如何获取所有 pod”
- 接收确切命令
- 无需记忆复杂语法



NotebookLM

RL Engineering: 监督下的自主性与实验文化

Workflow Strategy: Try and Rollback



频繁提交 Checkpoints -> 测试 Claude 的自主实现 -> 鼓励实验性开发。

Use Cases

- **Feature Dev**: 编写中小规模功能（如权重传输组件）。
- **Documentation**: 自动添加注释，加速文档编写。

Reality Check

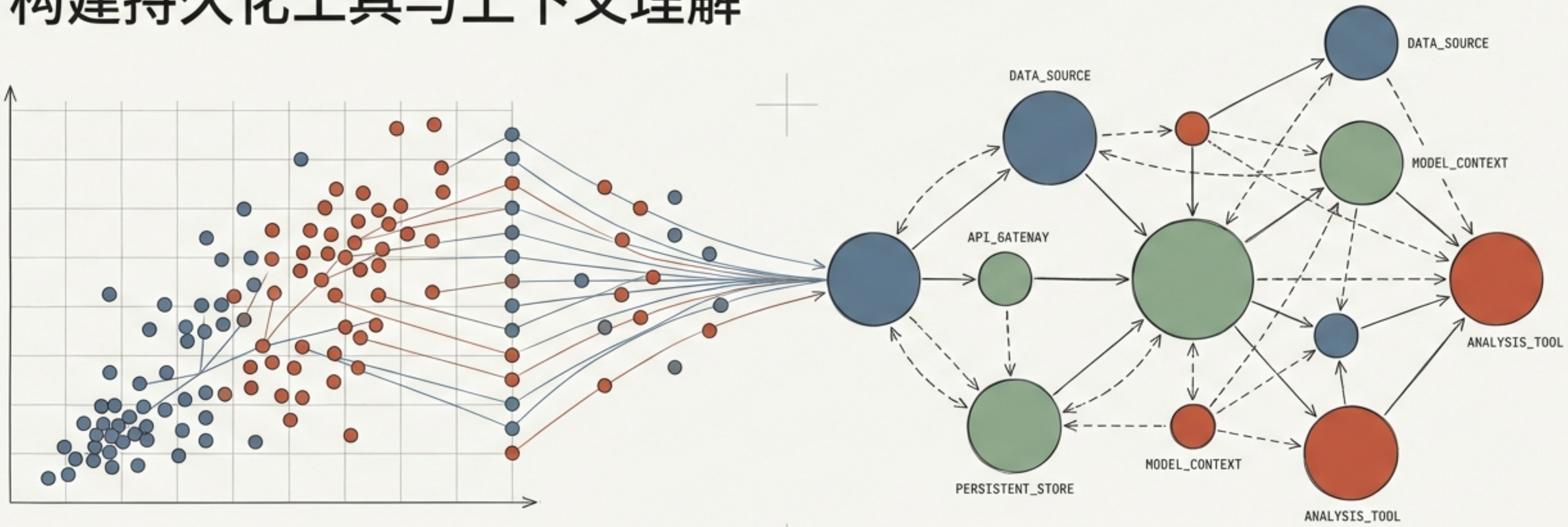
- 首次尝试成功率约为 **1/3**。
- **Tip**: 在 `Claude.md` 中添加指令防止重复工具调用（`'Just use the right path'`）。

NotebookLM

PART III

第三部分：数据科学与 API

构建持久化工具与上下文理解



NotebookLM

Data Science: 从 “一次性脚本” 到 “ 持久化应用”

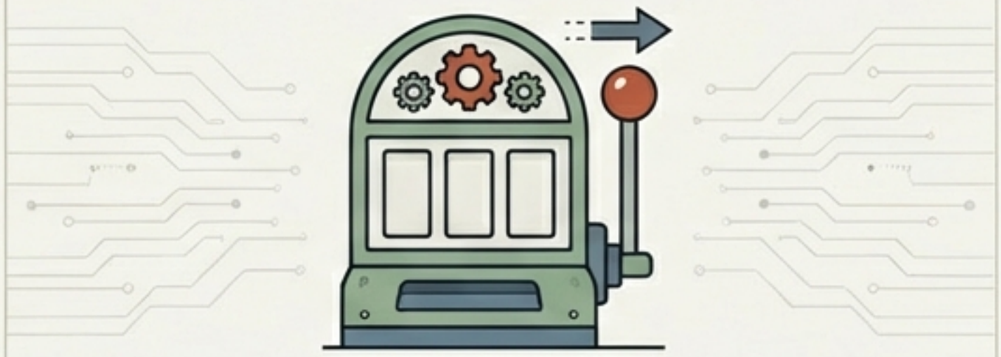
Transformation

Before: 废弃的 Jupyter Notebooks (Disposable).

After: 可复用的 React 仪表盘 (Persistent React Dashboards).

Impact: 即使 “不懂 JavaScript/TypeScript”，也能构建 `5,000行` 代码的应用程序。

The “Slot Machine” Method



保存状态 -> 让 `Claude` 自主工作 30 分钟 -> 接受结果或重新开始（比修复错误更高效）。

深入理解模型训练性能，而不仅仅是看单个数字的变化。

NotebookLM

API Knowledge: 消除上下文切换成本

Increased Confidence (增强信心)

团队成员可以独立调试陌生代码库中的 bug，无需询问他人。

Dogfooding

自动使用最新的研究模型快照（Research model snapshots）-> 在开发中直接体验模型行为变化。

Workflow Shift

Old Way:
复制粘贴代码片段 ->
拖拽 文件到 Claude.ai ->
耗费脑力整理上下文。

Frictionless

New Way:
直接在终端提问 ->
自动获取 Monorepo 上下文 ->
“First stop”（任务的第一站）。

NotebookLM

PART IV

第四部分：增长、设计与计与法律

创意自动化与非技术人员赋能

Creativity

Design

Legal

NotebookLM

Growth Marketing: 自动化创意生成 (10x Output)

10x

Tiempos Headlineg Blueprint

Creative Output (创意产出)

Google Ads Automation

- 处理 CSV -> 识别表现不佳的广告 -> 生成数百个符合字符限制 (30/90 chars) 的新变体。

Figma Plugin

- 编程生成 100+ 广告变体 -> 替换标题和描述 -> 半秒钟完成一小时的工作。

Meta Ads MCP

Editorial Engineering Blueprint

- 直接在 Claude Desktop 中查询广告支出和效果，无需切换平台。
- 战略转变：从手动执行转向构建 Agentic Automation (代理自动化）。

NotebookLM

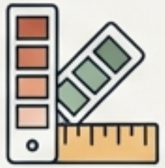
Product Design: 设计与工程的无缝融合

Core Workflow: Image Pasting for Prototyping

粘贴截图 -> Claude 生成功能性原型。工程师直接理解交互意图，取代静态的 Figma 设计图。

Frontend Polish

设计师直接修改字体、颜色、间距（Visual tweaks）-> 无需与工程师反复沟通。



Two Experiences

开发者："增强工作流"。

非技术用户："Holy crap, I'm a developer"（天哪，我成了开发者）。

Tip: "明确告诉 Claude '我是设计师，不懂代码'，要求它逐步解释。"

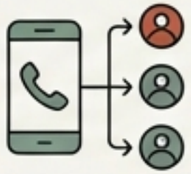
NotebookLM

Legal: 法律部门的无代码原型与自动化

Innovation: Prototype-driven innovation (原型驱动创新)

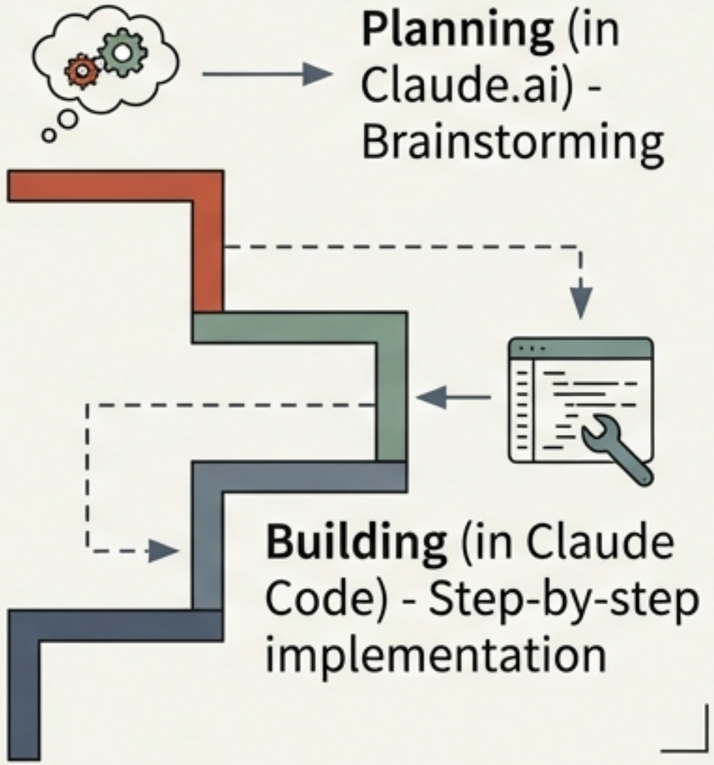
- **Accessibility Tools:** 为有语言障碍的的家庭成员构建语音应用（1小时内完成）。

- **Workflow Automation:** 构建 'Phone tree' 原型 -> 帮助员工联系合适的律师。



- **Compliance:** 快速构建合规工具，平衡创新与风险管理。

Work Style



NotebookLM

跨团队的核心洞察 (Key Cross-Team Insights)

Documentation is Code

编写详细的 `Claude.md` 是让 Claude 变聪明的关键 (Data Infra)。

Iterative Partnership

把它当作合作伙伴而非一次性解决方案；建立自给自足的测试循环 (Product Dev)。

Context First

利用 MCP 服务器和自动上下文获取，减少手动复制粘贴 (Security/API)。

Don’t Fear the Code

非技术人员应利用截图和自然语言大胆尝试原型设计 (Design/Legal)。

从 10x 营销产出到 5分钟的安全调试，Claude Code 正在重新定义 “谁可以构建软件”。

NotebookLM

Anthropic

变革已经开始。

(The transformation has begun.)

Anthropic 内部团队正在定义人机协作的未来。

NotebookLM