

The Exact Claude Setup That Writes My Documents Automatically

Dylan Davis

2026-05-24

目录

1	课程速览	2
2	摘要	2
3	一、核心论点：从”对话”到”文档”是 AI 杠杆的最低垂果实	3
4	二、三步流水线总览	3
5	三、Step 1: 把”好”从模板里抽出来	4
5.1	为什么要先逆向工程	4
5.2	逆向工程 Prompt 的关键词	4
5.3	实操细节	4
6	四、Step 2: 把规格固化成 Claude Skill	5
6.1	Skill 是什么	5
6.2	操作步骤	5
7	五、Step 3: 在 Project 里组装可复用 workflow	5
7.1	WWH 框架：先写 base prompt，再过 Prompt Improver	6
7.2	最终调用：只需一句话	7
8	六、易错点 / 反直觉提醒	8
9	七、本讲核心金句	8
10	八、行动清单	9
11	附录：截图索引	10

1. 课程速览

主讲人：Dylan Davis · Gradient Labs 创始人 栏目：YouTube 个人频道（AI 自动化实战） 时长：约 16:30 类型：教学 / How-to

YouTube: <https://www.youtube.com/watch?v=u2xcbJNhZIY>

本讲主题：把”会议转写 → 客户文档（提案 / 合同 / SOP）”这件每周重复的事，用 Claude Skills + Project 三步固化成一个 30 秒就跑完、风格像素级匹配你既有模板的自动化系统。

建议受众：每周要手写 3 份以上结构化文档的咨询顾问 / SaaS 销售 / 自由职业者；想给团队搭一套”对话转文档”内部工具的 Ops / RevOps；正在学 Claude Skills 的 AI 实操者。



图 1: 开场点题：作者声称自己已 5 个月没从零写过提案。

2. 摘要

- **核心命题**：文档自动化的关键不在”让 AI 写得好”，而在”先逆向工程已有的好文档，把美学固化成 Skill”——风格匹配比内容生成更难。
- **关键方法**：三步流水线——逆向工程模板 → 封装成 Claude Skill → 嵌入 Project（含参考文件 + System Prompt）。最终输入只剩一句”draft a proposal from the below transcript”。
- **核心数据**：单次提案生成 ≈ 30 秒；逆向工程 + 生成 Skill 两个阶段各耗时约 10 ~ 15 分钟（一次性投入）。
- **常见误解**：大多数人以为写好 Prompt 就够了；实际上 Prompt 必须先过 Anthropic Prompt Improver 跑一轮，并且要在 PDF + DOCX 两种格式同时喂给 AI 才能拿到

pixel-perfect 还原。

- **应用边界**：Skills 目前仅 Claude 原生支持；该流程适合”结构稳定、风格统一”的文档，不适合每次结构都变的创意写作。

3. 一、核心论点：从”对话”到”文档”是 AI 杠杆的最低垂果实

作者主张：在他服务过的所有客户里，把一次会议对话转成一份成品文档是 AI 落地中 ROI 最高、最容易跑通的第一个用例。这与目前主流”先做 RAG 知识库 / 先搭 Agent”的路线直接对立——他认为这些更高阶的玩法在企业里几乎都失败，原因是它们跳过了”先用 AI 替代一个明确、重复、单点的人工动作”这一阶段。

他给出三层论据：第一，所有公司本来就有”已经被验证好用”的文档模板（提案、合同、SOP、跟进邮件），意味着**已有 ground truth**，不需要从零定义”好”；第二，文档生成的输入（会议转写）和输出（Word/PDF）都是结构化的，闭环短；第三，Claude Skills 这一新能力让”风格固化”第一次变得可复用——这是过去 ChatGPT + Prompt 做不到的关键变量。

"This use case, being able to take a conversation and turn it into a document or an action, is probably one of the lowest hanging fruits I've seen with any of my clients. And it's the beginning of where real businesses start to see leverage through AI automation."

(约 01:40)

4. 二、三步流水线总览

整个系统由三个阶段串联，每一阶段的产物是下一阶段的输入：

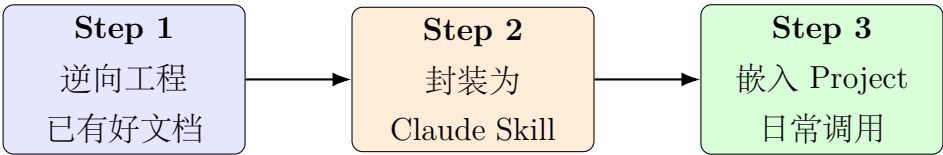


图 2: 三步流水线：从”美学提取”到”日常一句话调用”。

阶段	输入	动作	产物
Step 1	已有模板（PDF + DOCX 双格式）	让 Claude 方法性地逆向工程美学规格	完整 spec（字体、间距、颜色、结构）
Step 2	上一步的 spec	在同一对话里调用 Skill Creator	一个可复用 Skill（如 gradient-proposal）
Step 3	会议转写	在 Project 中以一句话触发	像素级匹配模板的 Word 文档

5. 三、Step 1：把”好”从模板里抽出来

5.1 为什么要先逆向工程

作者强调：你已经有好文档了——这是所有客户都低估的事实。提案、合同、SOP 已经在用，它们就是”好”的 benchmark。你要做的不是创造美学，而是把美学说清楚给 AI 听，让它能在未来无数次复刻。

5.2 逆向工程 Prompt 的关键词

他给出的 Prompt 非常短（几行），但有两个关键术语反复出现，他声称这两个词在实测里显著提升了 AI 的还原质量：

关键词	作用	为什么生效
methodically	强迫 AI 按章节、按维度逐项拆解，而非泛泛总结	降低遗漏率，输出可枚举的 spec
reverse engineer	触发”从成品反推规则”的思维模式	比”describe this document”更贴近设计师视角
pixel perfect	锁定输出标准	让 AI 在歧义处倾向”复刻”而非”改进”

5.3 实操细节

- **双格式喂入**：同一份模板要同时上传 PDF 和 DOCX。PDF 给 AI 看视觉，DOCX 给 AI 看底层样式标签（字体名、段距）。
- **留余地**：Prompt 里要说”extract everything, including but not limited to...”，给 AI 自由补充你没想到的维度。
- **耗时**：作者实测这一步 Claude 跑了约 10~15 分钟，输出一份结构化 spec。

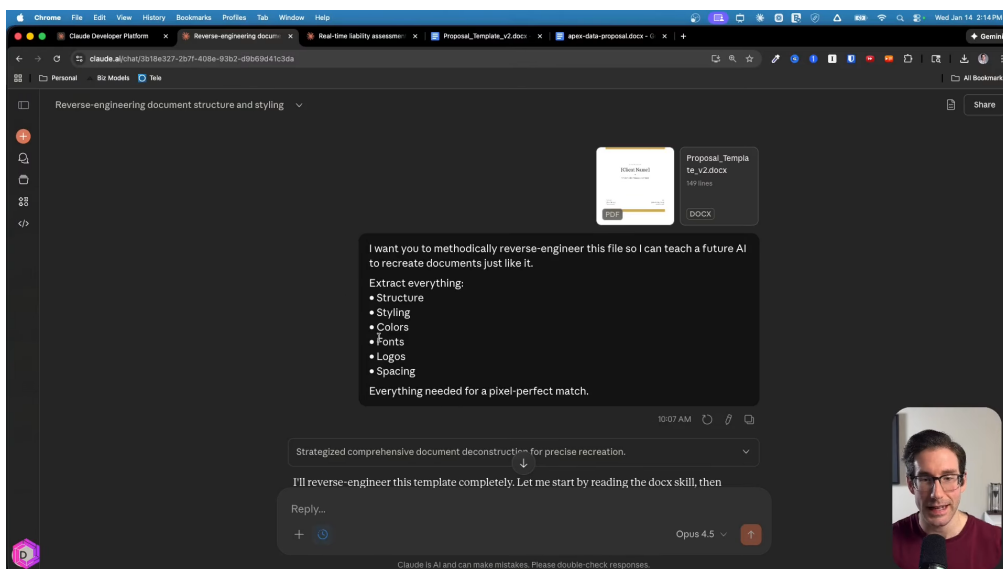


图 3: Step 1 的 Prompt: methodically + reverse engineer + pixel perfect 三个关键词。

6. 四、Step 2: 把规格固化成 Claude Skill

6.1 Skill 是什么

作者一句话定义: **A Claude Skill is just a bunch of prompts inside a folder that the AI can call at any given time.** 它不是 fine-tune, 不是 RAG, 本质是一个被 Claude 自动加载的、可命名调用的提示词包。

6.2 操作步骤

1. 在同一对话里追加 Prompt: "now use the skill creator to create a skill..."——关键是不开新对话, 因为 spec 在上下文里。
2. 进入 Claude Settings → Capabilities → Skills, 确认 *Skills* 和 *Skill Creator* 两个开关都打开。
3. AI 生成 Skill 后, 会出现一个 **Copy to your skills** 按钮, 点一下就永久存到账号里。
4. 想分享给同事? 直接下载 Skill 文件夹, 对方上传即可。

7. 五、Step 3: 在 Project 里组装可复用 workflow

一个 Project 由三块拼装而成, 作者强调 **System Prompt** 是 80% 的工作量集中点:

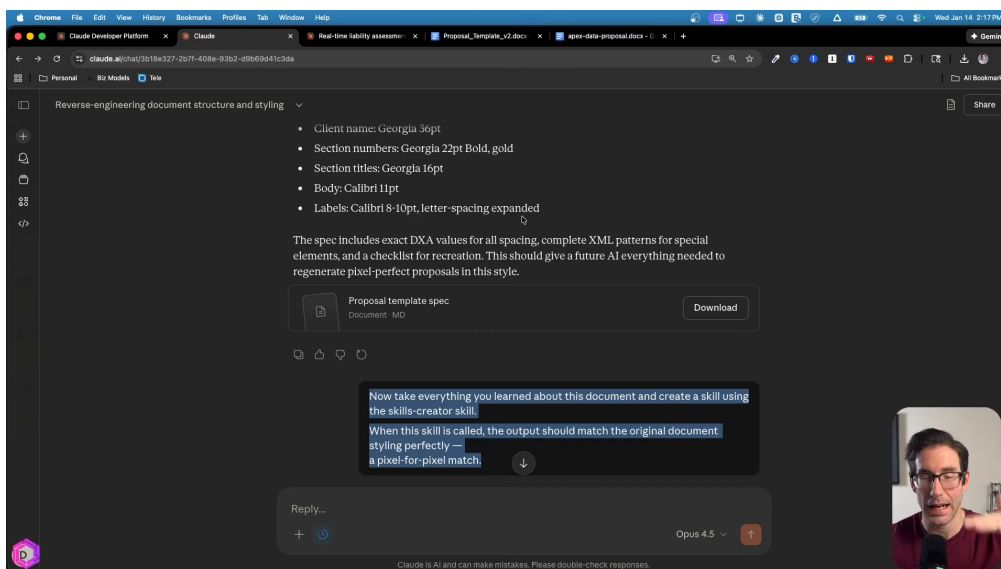


图 4: Skill Creator 输出的 Skill，可一键 Copy to your skills 持久化。

组件	放什么	作用
Reference Files	既往优秀提案、公司 Offering 列表（Markdown）	给 AI 提供内容素材与语气样本
System Prompt	WWH 框架编写的指令，并显式调用 Skill 名	控制行为、结构、输出格式
Skill 调用	在 System Prompt 中写明 Skill 名（如 gradient-proposal）	保证最终输出 pixel-perfect

7.1 WWH 框架：先写 base prompt，再过 Prompt Improver

作者反复强调一个反直觉的顺序：先写朴素的 base prompt，再丢给 Anthropic Prompt Improver 自动改写，而不是一上来就追求完美 Prompt。

维度	你要回答的问题	在提案场景里的具体内容
What	让 AI 做什么	”你是资深提案撰写者，接收会议转写，输出提案”
Why	为什么做这件事	让 AI 推断你没明说的偏好（如基调、长度倾向）
How	有什么硬约束	输出 6 段结构 / DOCX 文件类型 / 调用某 Skill

写完 base prompt 后，去 platform.claude.com 的 Prompt Improver 页面：

- 粘贴 base prompt → 勾选”used by thinking model”（因为我们要用 Opus 4.5 with extended thinking）。

□ 等几秒，Improver 会自动加上：分节标题（让 AI 识别每段功能）、Scratch Pad 段（让 Claude 内部自我批判后再输出）、Format Requirements、Skill 调用块。

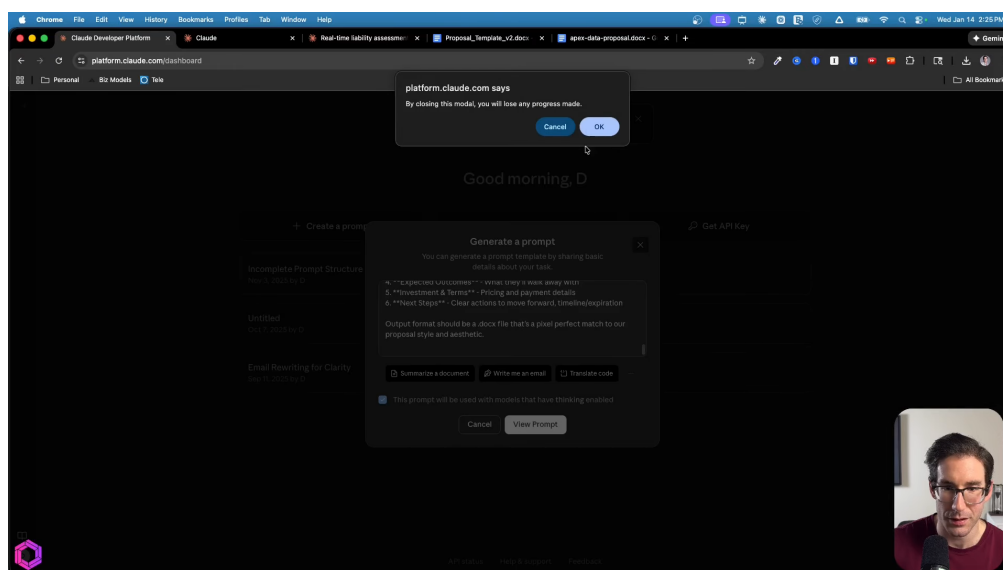


图 5: Prompt Improver 自动注入的 Scratch Pad 段——Claude 在回答前先内部推演。

7.2 最终调用：只需一句话

"Draft a proposal from the below transcript." + [paste transcript] $\Rightarrow$ 30 秒后输出 DOCX，与模板像素级一致。

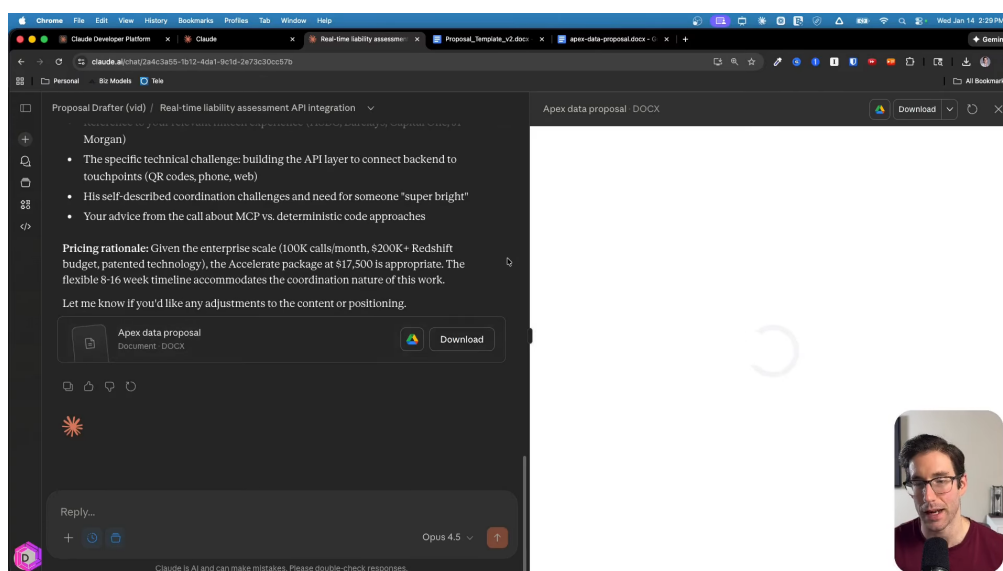


图 6: 最终对照：左为模板，右为 AI 输出，作者声称已逐项核对，pixel-perfect。

8. 六、易错点 / 反直觉提醒

易错点	大多数人的做法	正解	为什么
只喂 PDF 给 AI 让它学样式	觉得 PDF 视觉就够了	PDF + DOCX 双格式同时给	DOCX 含底层样式标签，PDF 只有视觉渲染
开新对话调用 Skill Creator	觉得逻辑分离更清晰	必须在”逆向工程”那条对话里继续	Spec 在上下文里，新开对话会丢
直接手写完美 Prompt	反复打磨措辞	先写 base，再过 Prompt Improver	Improver 会加 Scratch Pad、分节标题等结构性增益，是手写难复现的
用 Prompt Improver 时忘勾 thinking model	用默认选项	必须勾选”used by thinking model”	推理模型和普通模型的最佳 Prompt 结构不同
System Prompt 里不显式写 Skill 名	以为 Claude 会自动找	必须写出 Skill 名（如 gradient-proposal）	Claude 不会自动猜你想用哪个 Skill
期待第一次输出就完美	试一次失败就放弃	持续迭代 Prompt 和参考文件	人眼”好”和 AI 首次输出之间永远有 gap

9. 七、本讲核心金句

“I haven’t written a proposal from scratch in probably five months.” (00:00)

上下文：开场即抛出反直觉结论——一个以撰写提案为核心动作的咨询业者，已 5 个月不动笔。

“A Claude skill, all that is, is a bunch of prompts inside of a folder that the AI can call at any given time.” (约 05:00)

上下文：作者在解释 Skill 概念时，故意去神秘化——避免观众把它当成 *fine-tune* 或复杂工程。

“Our job as AI-native employees, leaders, and business owners is to get better at adjusting the AI over time... eventually these start to converge. And once they converge

is when you can actually outsource that specific task to AI completely.” (约 14:30)

上下文：讲到“*persistence + testing*”这一元技能时——他认为这才是比任何 *Prompt* 模板都更重要的能力。

10. 八、行动清单

读完今天就能做的事：

- ☐ 从你过去 3 个月发出的文档里挑 1 份**结构最稳定、最常重复**的（提案 / 跟进邮件 / 周报），同时导出 PDF + DOCX。
- ☐ 在 Claude 里开新对话，粘贴双格式文件，运行 Prompt：“methodically reverse engineer this document to a pixel-perfect spec, extract everything including but not limited to fonts, spacing, colors, structure”。
- ☐ 同一对话追加：“now use the skill creator to create a skill that outputs a pixel-perfect match”，生成后点 Copy to your skills。
- ☐ 用 WWH 框架写一份不超过 200 字的 base system prompt，去 platform.claude.com 跑 Prompt Improver，**记得勾选 thinking model**。
- ☐ 在 Claude 新建 Project，挂上 2~3 份历史优秀样本 + Offering 说明，把改写后的 System Prompt 粘进去，显式写出 Skill 名。
- ☐ 拿一份真实会议转写测试一次，记录 AI 输出与你期望的 gap，迭代 System Prompt 至少 3 轮再下结论。
- ☐ (进阶) 把这个 Project 复制一份，换成“合同”或“跟进邮件”场景，验证流水线的可迁移性。

11. 附录：截图索引

时间	截图	主题
00:00	frame_0001	开场点题：5 个月不写提案
01:30	frame_0003	三步流水线总览
02:10	frame_0004	Step 1 概念：从已有”好”反推
04:30	frame_0015	逆向工程 Prompt 全文
05:00	frame_0016	Claude 实际操作界面
05:40	frame_0018	双格式（PDF + DOCX）上传
06:40	frame_0024	Skill Creator 生成结果 + Copy 按钮
08:00	frame_0033	Settings → Capabilities → Skills 开关
11:00	frame_0044	Prompt Improver 注入的 Scratch Pad
11:30	frame_0045	Format Requirements 段
13:00	frame_0053	最终输出 vs 模板对照
13:30	frame_0054	Project 配置界面
14:00	frame_0055	Reference Files 上传
14:30	frame_0057	收尾：persistence 与 testing 的元技能