

为什么需要 Skill Engineering

Skill Engineering 自学课件

为什么需要 Skill Engineering

让一次成功的任务，沉淀为可复用、可验证、可迭代的团队能力

当前现象

已经开始使用 Qoder
不少同事已把 AI 用到日常工作中

数据处理是主要场景
整理明细、汇总报表、解释结果

主要仍通过 Chat 完成工作
每次重新说明需求，再反复调整

成功经验尚未稳定沉淀
换一个 Chat，往往又要从头解释

➡

下一步目标

感知 Skill 的作用
看见流程和规则复用带来的价值

一次性完成成为可复用能力
把成功方法留在完整的能力包中

个人经验成为团队共同方法
让另一位同事也能按同一口径完成

形成共同的工程语言
知道如何设计、执行、验证和改进

把 Chat 中成功的方法沉淀为 Skill，再通过验证与复用完善它。

出品：AI 经营交付效能专项组

01 / 09

本页讲解

比如，今天花了几轮对话把日报做好，明天换个 Chat，同样的日期、去重和平均耗时又得重新解释。我们可以把今天用过的数据、确认的规则和核对过的结果留下来，再请 AI 把处理方法整理成 Skill。以后拿到同类数据，就有一套现成的方法可以继续用。交给同事时，他也能顺着这些资料看懂这份日报是怎么算出来、怎么验过的。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

01 / 09

从一次数据处理任务说起

Skill Engineering 自学课件

从一次数据处理任务说起

拿到下面这份工单明细，怎样让 AI 把它整理成 10 月 10 日的日报？

输入：7 行虚构工单明细

工单 ID	创建日期	状态	耗时 / 分钟
T001	2026-10-10	已完成	60
T002	2026-10-10	待处理	0
T003	2026-10-10	已完成	45
T004	2026-10-10	已完成	90
T004	2026-10-10	已完成	90
T005	2026-10-10	待处理	0
T006	2026-10-11	已完成	30

T004 重复一行；T006 属于 10 月 11 日。

这份明细要变成一份日报：去重、按创建日期筛选，再统计数量与完成单平均耗时。



交给 AI 的任务

请统计 10 月 10 日创建的工单，按 ID 去重；内容冲突时停止。告诉我总数、完成数、待处理数，以及完成工单的平均耗时。

应该交回的日报

5 单：完成 3 单，待处理 2 单
完成单平均： $(60 + 45 + 90) \div 3 = 65$ 分钟

出品：AI 经营交付效能专项组

02 / 09

本页讲解

这 7 行就是我们交给 AI 的原始明细。T004 出现了两次，同一张工单只能算一次；T006 是 10 月 11 日创建的，也不该进入 10 日的日报。处理后剩下 5 张工单，其中 3 张已完成、2 张待处理。完成的 3 张分别用了 60、45、90 分钟，加起来 195 分钟，除以 3，平均就是 65 分钟。我们希望 AI 把明细整理成这样的日报，并让人能核对它怎么算的。把这些规则保存成 Skill，下次拿到新数据，就有了可以沿用的方法。

继续阅读与实践：点击打开在线深入阅读

Chat、Skill、Skill Engineering

Skill Engineering 自学课件

Chat、Skill、Skill Engineering

把一次成功的任务，转化成能跨对话复用、执行关键操作、具备验证机制的工作能力。

Skill Engineering = Skill Design + Deterministic Execution + Validation + Evaluation + Reuse

技能设计 确定性执行 结果验证 效果评估 复用传播

Chat 使用
单次完成任务
提出问题，获得答案
解决当下的任务

➡

Skill 使用
复用已有能力
调用已沉淀的流程和资源
按约定方法完成任务

➡

Skill Engineering
工程化沉淀与持续优化
分清规则、实现与验证
换对话、换数据、换人复用

更稳定：新 Chat 也能复现。更可靠：关键计算可验证。更高效：团队可复用、可传播。

出品：AI 经营交付效能专项组

03 / 09

本页讲解

刚才那份日报，在一个 Chat 里做对，就完成了当次任务。把规则和处理脚本保存成 Skill，下一次便有了可调用的方法。不过，另一位同事拿到新数据，也能做对吗？还要实际试一试。我们讲的 Skill Engineering，就是在沉淀方法之后，继续检查换对话、换数据、遇到异常时会发生什么。这样才知道，保存的方法离开原来的聊天还能不能工作。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

03 / 09

一个数据处理 Skill 的两个部分

Skill Engineering 自学课件

一个数据处理 Skill 的两个部分

Workflow 安排任务怎样推进，Data Processing 完成数据的解析、转换与计算。

Workflow：任务怎样推进

01
理解任务
确认目标与业务规则

02
检查输入
核对字段、类型、日期

03
执行处理
筛选、关联、去重、汇总

04
验证结果
每次核对，失败停止

05
分析交付
解释结果并交付报表

Workflow（工作流）
决定步骤顺序、如何处理歧义、何时继续、何时停止。

Data Processing（数据处理）
按确认的规则完成运算，并返回结果、异常与校验信息。

Workflow 安排执行与停止；每次运行通过结果校验后，再分析交付。

出品：AI 经营交付效能专项组

04 / 09

本页讲解

算出 65 分钟之前，还有不少判断要做。统计哪一天、字段齐不齐、重复工单能不能合并，都影响最后的结果。假如同一个 ID 的两行耗时不同，就要停下来弄清原因，不能悄悄选一个。图里的 Workflow 管这些推进和停止的决定，Data Processing 按确认的规则去重、筛选和计算。两部分接起来，结果验过了，才把日报交出去。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

04 / 09

AI 推理 vs 确定性执行

Skill Engineering 自学课件

AI 推理 vs 确定性执行

在工作流与数据处理两个层次上，分别确定 AI 和固定代码的职责。

工作层次	AI 推理负责	确定性实现负责
Workflow 工作流	识别任务意图 判断异常原因 选择处理策略	固定步骤顺序 状态检查 必经校验、停止条件
Data Processing 数据处理	理解字段含义 识别非标准格式 解释异常	筛选、关联、去重 汇总、金额计算 格式验证

让 Agent 处理不确定性，让代码处理确定性，让验证机制决定结果是否可信。

出品：AI 经营交付效能专项组

05 / 09

本页讲解

拿刚才的平均耗时来说，195 除以 3，适合交给固定脚本，下一次也按同一规则计算。AI 则可以帮助理解需求：客户说的日期是创建日期，还是完成日期？字段名称换了，含义是否相同？把这些问题说清楚之后，再交给工具执行。脚本也可能写错分母，所以我们还要拿已知结果的样例核对，不能因为用了代码就省掉验证。

继续阅读与实践：点击打开在线深入阅读

一个好的 Skill 长什么样

Skill Engineering 自学课件

一个好的 Skill 长什么样

SKILL.md 保留入口与行为契约，详细 Spec 与 Implementation 按职责分开。

ticket-daily-report/

SKILL.md

执行入口与行为契约

何时使用
执行路径与分支
完成与停止条件

references/

业务与技术 Spec

data-contract.md
business-rules.md
workflow-spec.md

scripts/

确定性实现

输入检查 inspect.py
数据处理 transform.py
结果验证 validate.py

assets/

模板与静态资源

报表模板
实际需要的图片
其他输出资源

按需扩展：tests/ 验证程序，evals/ 验证 Agent 的完整行为。

主文件简洁，细节按需加载（Progressive Disclosure）。简单任务可以只有 SKILL.md。

出品：AI 经营交付效能专项组

06 / 09

本页讲解

把 Skill 想成给接手同事的一份资料包，会容易理解这个目录。打开 SKILL.md，他先知道这项工作怎么开始、什么时候算完成；需要核对口径时，再看 references 里的详细规则；真正处理数据，则调用 scripts 里的实现。我们的练习把检查、处理和校验放在一个 pipeline.py 中，图上的三个脚本名只是职责示例。简单任务用不到的目录，就不用增加。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

06 / 09

Qoder 实操：沉淀一个 Skill

Skill Engineering 自学课件

Qoder 实操：沉淀一个 Skill

以已经做对的工单日报为起点，把任务方法整理成能独立执行的能力包。

01

成功任务

保存输入、口径和核对过的结果

➡

02

提炼 Skill

写清契约与流程
保存关键处理脚本

➡

03

导入完整包

在技能管理中导入 ZIP
确认出现在已安装中

➡

04

新 Chat 运行

用 / 选择 Skill
只给本次必要输入

给 Qoder 的提示词：把刚才成功的数据处理沉淀为 Skill：保存输入、口径和验收结果，写清流程与停止条件；关键处理与校验保存为脚本；新 Chat 不依赖旧历史。

先手动选择验证执行，再用自然语言验证触发。Qoder 操作入口按当前产品版本核对。

出品：AI 经营交付效能专项组

07 / 09

本页讲解

当你和 AI 把一次任务做对了，别只留下最后的报表。这次用的输入、确认的口径和核对过的结果，都是下次验证的参照。请 AI 把处理方法整理成完整 Skill 包，导入后再开一个新对话，给它新任务。我们故意换对话，是想看它能否靠保存下来的规则工作。手动选择能跑通之后，再试自然语言能否找到它；入口按你使用的 Qoder 版本核对。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

07 / 09

如何证明 Skill 可靠



本页讲解

如果只拿一份正常数据试，很多问题不会出现。缺了字段怎么办？同一个 ID 的两行内容冲突怎么办？所以我们给练习准备了正常、可处理变化和应停止异常三类场景，每次都看实际结果和停止行为。自评可以帮你发现该查哪里；想知道是否真的做到了，还得看运行记录。改过 Skill 后把这些用例再跑一遍，才能知道修复有没有影响原来正确的结果。

继续阅读与实践：点击打开在线深入阅读

开始实践与问题反馈

Skill Engineering 自学课件

开始实践与问题反馈

把日常工作中的一个成功方法，变成下一次可以继续使用的 Skill。

今天就可以开始的一次实践

01 选任务并自评

规则清楚，先确认验收依据

02 完成三类验证

正常、变化、异常，核对结果与停止

03 换人复用并迭代

同版本再试，把新问题补进用例

问题反馈

遇到问题，联系张坚、李琦

脱敏输入样例

预期结果与实际结果

出错截图与校验记录

业务同事沉淀方法与验收标准，技术同事协助固定关键处理，团队共同验证和改进。

出品：AI 经营交付效能专项组

09 / 09

本页讲解

你可以从手头一件重复工作开始，不必一次做出一个大系统。选定任务后，用自评看看还缺什么，再准备几份能判断对错的样例。在新对话里跑通了，就请另一位同事拿同版本再试，看看哪些说明还不够。遇到问题，把脱敏输入、预期与实际结果和校验记录一起交给张坚、李琦。这样大家能重现问题，也更容易把修正补进下一版用例。

继续阅读与实践：点击打开在线深入阅读

出品：AI 经营交付效能专项组

09 / 09