Week01｜Lesson02｜一节课讲透交付标准：知识库问答 + 工单联动怎么定义 Done?

把 Done 讲成业务、架构、合规、运维都能签字的交付语言

本讲结构

一节课讲透“交付标准”：知识库问答 + 工单联动怎么定义 Done?

这一讲不讲“怎么把回答做得更像”，而是讲：如果今天就要让业务、架构、合规、运维一起签字，你到底要交出什么。

上一讲 返回 Week01 下一讲

下载讲义

提供适合离线阅读的 PDF 版和适合批注整理的 Word 版。

PDF 版 · 打印 / 离线阅读 Word 版 · 批注 / 二次整理

> ! 一句话判断
>
> Done 不是“能回答”，而是“可签字交付”：业务愿意验收、质量有门禁、安全有边界、运行可降级、治理可追责。

这节课到底要完成什么

开场冲突

先把一个“看起来已经能答”的系统放进上线前评审会，理解为什么现场演示顺利，不等于任何一个角色敢放行。

判断结构

用 Demo vs Production 和五层 Done 框架，把“做完了没”从感觉判断，改成可签字、可复盘、可追责的工程语言。

签字现场

这一讲不只讲技术团队怎么看 Done，还会把业务、架构、合规、运维四个角色拉进同一张签字板。

可带走工件

最后要留下 3 张可以直接拿去改项目文档的模板，而不是只留下几个观点：记分卡、边界清单、质量门禁表。

> **Note**
>
> 这讲真正的升级，不是补更多功能，而是先把“签字条件”写出来。后面 14 周，本质上都在给这些条件补证据。

开场冲突：为什么“能答”仍不能上线

先看一个典型的上线前尴尬场景：

POC 阶段看起来已经成功

某金融客服团队做了一个“知识库问答 + 自动建单”助手，演示时非常顺：

- 能回答账户冻结、额度调整、账单问题等常见 FAQ
- 能从知识库里生成解释
- 能调用工单接口发起申请
- 业务方觉得“看起来已经能用了”

一进试点就没人敢签字

真正进入试点，风险立刻暴露：

- 回答引用了旧版本业务规则
- 没有按用户等级区分适用建议
- “建议建单”和“自动建单”没有分层
- 缺少证据引用，主管无法复核

- 坏例无法精确复现，因为索引、Prompt、数据版本没有绑定

⚠ 这节课真正的问题

一个“客服知识库 + 工单联动 Copilot”，到底什么时候才能说“做完了”？
更具体地说：什么时候业务、架构、合规、运维会一起签字，而不是把系统继续挡在上线门外？

先把判断翻过来：Done = 可签字交付

Demo 团队通常在看什么

- 现场能不能跑通
- 回答像不像人写的
- 工具链路顺不顺
- 领导会不会觉得“挺智能”
- 出错时先让人工兜底

真正要签字的人在看什么

- 业务能不能按口径验收
- 回答质量能不能被门禁和回归
- 安全、PII、动作边界是否前置
- 出错后能否定位、降级、回滚
- 责任边界和证据链是否完整

! 关键判断

这讲不是“定义课”，而是一次“签字课”。
它要回答的不是“模型会不会答”，而是“如果今天要上线，谁还会拦你，为什么会拦你”。

Demo vs Production：不是补功能，而是补签字条件

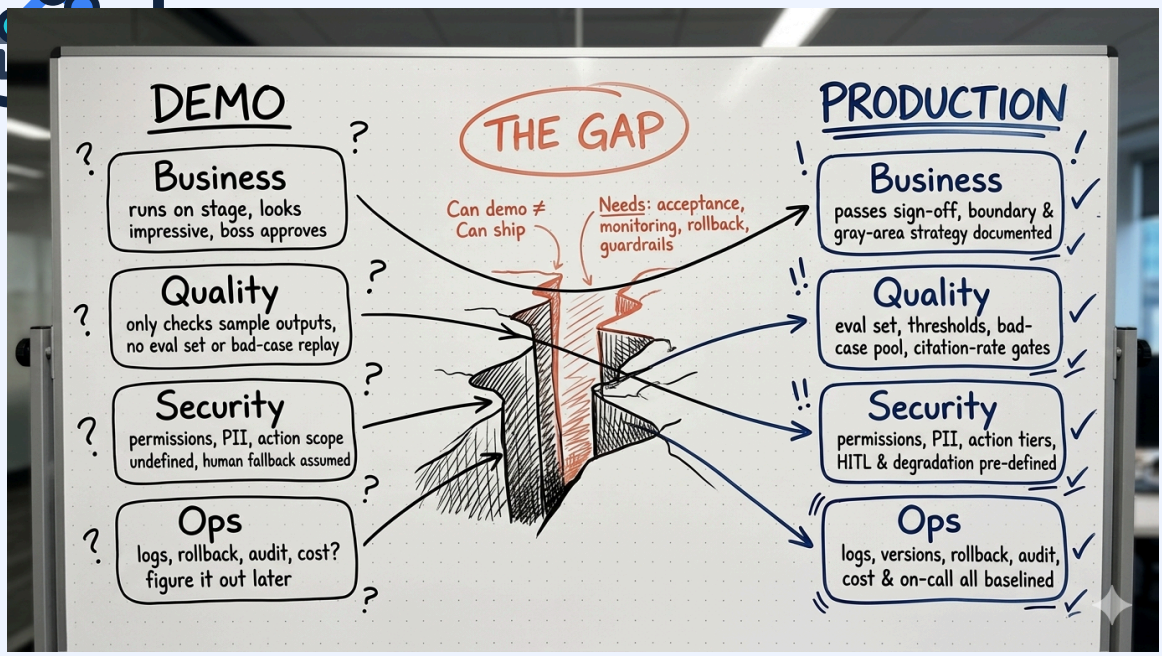


Figure 1

如果你现在只能说	签字的人会继续追问
“已经能跑起来了。”	谁验收？适用边界是什么？灰度策略在哪里？
“回答看起来挺准的。”	评测集、阈值、坏例池、证据覆盖率在哪里？
“先让人工兜底。”	哪些动作必须 HITL？哪些动作绝不能自动执行？
“日志后面再补。”	出错时怎么定位、回放、回滚、追责？
“等下一阶段再治理。”	版本、索引、Prompt、发布记录是否可追踪？

> **Note**
>
> 会答，不等于可交付。
> 后面 14 周不是在给系统“堆功能”，而是在给这些签字条件补证据。

五层 Done：这五层要一起站住

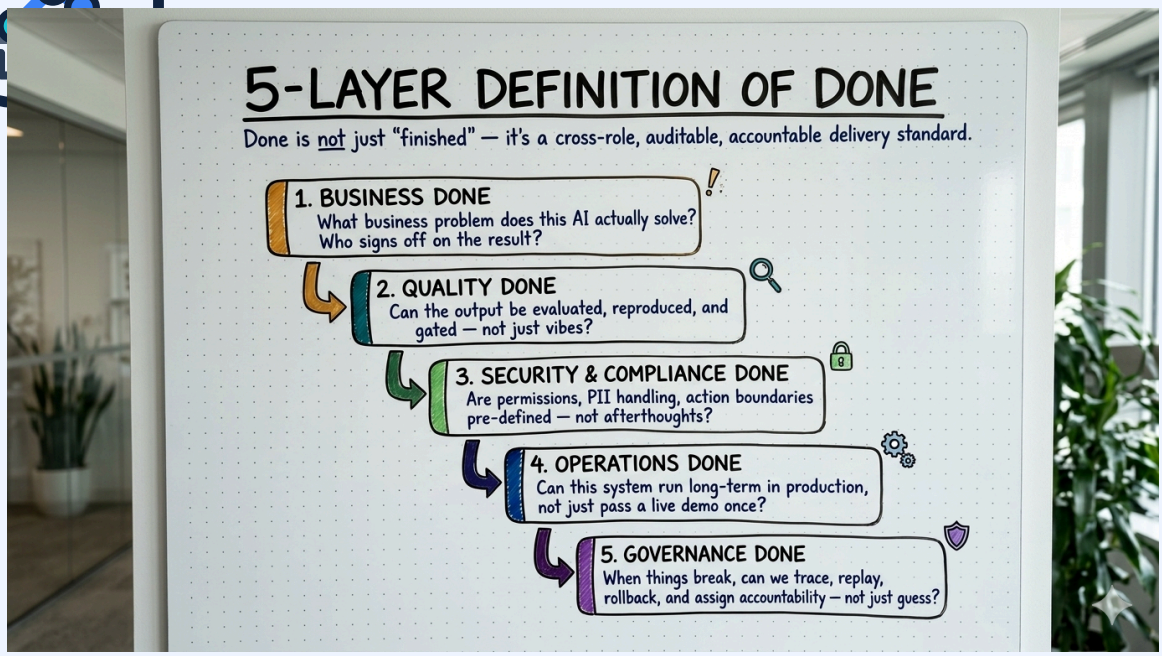


Figure 2

业务层 Done

这一层在问什么

- 这个 AI 到底在解决什么业务问题?
- 谁会为结果签字?
- 成功指标、适用边界和非目标是否写清?
- 灰度范围和失败容忍度是否明确?

最小标准、误区与案例

- 最小标准：目标场景、成功指标、适用边界和“不做什么”都清楚
- 常见误区：用户觉得挺智能，就当业务价值成立
- OmniSupport 例子：客服 Copilot 的业务层 Done，不是“会答 FAQ”，而是提升首问解决率、减少重复查询、降低人工建单负担

质量层 Done

这一层在问什么

- 回答质量能不能被量化?
- 结果能不能被复现，而不是靠感觉?
- 每次变更后能不能重新回归?

- 高风险问题是否有单独门禁?

最小标准、误区与案例

- 最小标准：评测集、失败样本池、结构化打分口径、门禁阈值齐全
- 常见误区：只拿现场样例看着还行，就当已经达标
- OmniSupport 例子：至少要能看正确性、证据覆盖率、高风险误答率、坏例复现率

安全与合规层 Done

这一层在问什么

- 系统会不会做错不该做的事?
- PII、权限、动作边界是否前置?
- 哪些动作可以建议，哪些动作必须人工确认?
- 失败时会进入什么降级模式?

最小标准、误区与案例

- 最小标准：PII 分级、权限边界、动作分级、失败降级都已写清
- 常见误区：把安全与合规理解成上线前“再加一层审核”
- OmniSupport 例子：“建议建单”和“自动建单”必须分开；涉及账户、额度、身份信息的动作默认进入 HITL

运行层 Done

这一层在问什么

- 系统能不能长期跑，而不是只在演示现场偶尔跑通?
- 性能、失败率、工具调用成功率是否有红线?
- 服务抖动时如何降级?
- 成本区间和运行策略是否有基线?

最小标准、误区与案例

- 最小标准：P95/P99 延迟目标、失败率阈值、工具调用成功率、成本区间和降级策略齐全
- 常见误区：先把效果做出来，运行 later
- OmniSupport 例子：工单系统超时时，Copilot 必须回退成“给建议、不自动执行”的模式

治理层 Done

这一层在问什么

- 出问题时，能不能知道到底是哪一层变了?
- 数据、索引、Prompt、评测、上线与回滚版本是否都可追踪?
- 问题能否回放，而不是只能猜?

- 变更后责任边界是否清楚？

最小标准、误区与案例

- 最小标准：数据、索引、Prompt、评测、上线与回滚版本都能串起来
- 常见误区：只看平时表现，不建设复盘能力
- OmniSupport 例子：某次错误回答到底是旧知识、检索漂移还是 Prompt 变更引起的，必须能被回放和定位

> ! 本讲真正要留下的判断
>
> 只要这五层里有一层还站不住，就不能把“能答”当成“Done”。

四角色签字板：谁会拦你，为什么会拦你

业务方

- 第一反应：这套系统真的在改善业务结果吗？
- 最想看到：目标场景、成功指标、失败容忍度、灰度范围
- 最常卡住：看起来有用，但验收口径不清

架构方

- 第一反应：这条链路是不是可持续演进？
- 最想看到：数据版本、索引策略、Prompt 可追踪、工具 JSON 契约
- 最常卡住：系统能跑，但没有稳定准入标准

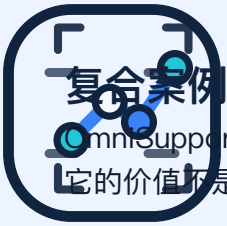
合规 / 风控

- 第一反应：系统会不会越权、泄露或自动做错事？
- 最想看到：PII 分级、权限边界、动作分级、证据引用规则
- 最常卡住：默认人工兜底，但边界没有正式写下来

运维 / SRE

- 第一反应：出了问题能不能定位、降级、回滚？
- 最想看到：健康阈值、告警规则、失败策略、日志与回放链路
- 最常卡住：演示阶段很顺，运行阶段没有兜底方案

> i Note
>
> Done 不是技术团队单方面写的，它必须能同时拿给业务、架构、合规和运维看。
> 只要有一个角色不敢签字，这个系统就还没有真正“做完”。



复合案例：把 OmniSupport 放进一次真实签字现场

OmniSupport Copilot 在这门课里不是主角，而是验证器。
它的价值不是“展示一个花哨 Demo”，而是把签字现场需要看的证据都逼出来。

角色	他们的第一句追问	真正要看的证据	不满足时会怎么处理
业务	首问解决率和建单负担真的会改善吗?	目标场景、成功指标、试点范围	延后上线，先补验收口径
架构	数据、索引、Prompt、工具链能不能持续演进?	版本、契约、可回放链路	不给生产准入
合规	哪些问题必须拒答，哪些动作必须 HITL?	PII 规则、权限边界、证据引用	要求补边界后再评审
运维	失败时怎么降级，变更后怎么回滚?	健康检查、告警、降级与回滚策略	只能灰度，不允许放量

三个课堂工件：这讲结束要带走什么

这节课要让学员带走的，不只是判断框架，还要有可以直接改成项目文档的最小模板。

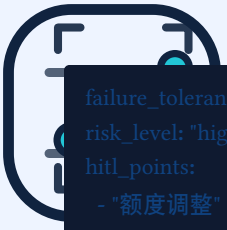
工件 1:《交付标准记分卡》

字段释义

- scene：当前 AI 能力服务的具体场景
- business_goal：要提升什么指标或减少什么成本
- success_metrics：业务真正签字时会看的结果指标
- failure_tolerance：哪些错误可以接受，哪些绝不允许
- risk_level：当前场景的风险等级
- hitl_points：哪些节点必须人工介入
- status：当前只到未达标、可灰度还是可上线

示例

```
scene: "客服知识库问答 + 工单联动"
business_goal: "减少重复查询，提升首问解决率"
success_metrics:
 - "FAQ 首问解决率"
 - "建议建单采纳率"
```



```
failure_tolerance: "高风险问题误答率必须低于阈值"
risk_level: "high"
hitl_points:
  - "额度调整"
  - "账户冻结"
status: "可灰度"
```

> Tip
>
> 适用方式：先拿这张表和业务方对齐“到底签什么”，再继续谈模型、检索和工具链。

工件 2：《业务验收口径 & 风险边界清单》

字段释义

- 适用用户：哪些人能使用这套能力
- 适用问题：哪些问题可以直接回答
- 不支持场景：哪些场景必须拒答或转人工
- 必须引用证据：哪些回答一定要给出处
- 必须人工确认：哪些动作只能建议，不能自动执行
- 禁止自动执行：哪些动作是明确红线
- PII 与越权规则：敏感字段和不同角色的边界怎么处理

示例

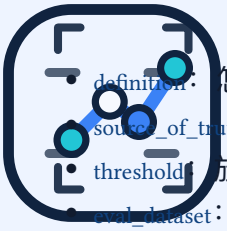
```
## 业务验收口径 & 风险边界清单
- 适用用户: 一线客服、值班主管
- 适用问题: FAQ 查询、办事规则解释、建议建单
- 不支持场景: 高风险账户操作、审批类动作、财务承诺
- 必须引用证据: 制度解释、规则口径、例外条款
- 必须人工确认: 正式建单、升级投诉、账户限制解除
- 禁止自动执行: 额度修改、合规承诺、身份信息变更
```

> Tip
>
> 适用方式：这张清单不是法务附录，而是系统行为开关。
> 写不清这里，后面的工具调用和自动动作都不应该放量。

工件 3：《上线前质量门禁表》

字段释义

- `metric_name`：到底在卡哪条线



- definition：怎么定义“通过”
- source_of_truth：门禁依据来自哪个规则或制度
- threshold：放行线是多少
- eval_dataset：用什么数据验证
- eval_frequency：什么频率必须重跑
- owner：谁维护这条门禁
- fallback_action：不达标时是降级、回滚还是阻断

示例

```
metric_name: "证据覆盖率"
definition: "高风险回答必须带受控来源引用"
source_of_truth: "业务规则库 v1"
threshold: ">= 95%"
eval_dataset: "高风险问答集"
eval_frequency: "每次索引变更后"
owner: "检索负责人"
fallback_action: "阻断上线并回退到人工流程"
```

> Tip
>
> 适用方式：上线前别再只问“效果看起来行不行”，而要问“门禁过没过”。
> 这张表就是质量层 Done 的正式入口。

为什么后面 14 周都在给 Done 补证据

后续周次	会补哪类证据	对应 Lesson02 的哪一层 Done
Week02	输入范围、字段契约、门禁规则	业务层 / 安全与合规层
Week03	采集、入湖、原始数据可回放	质量层 / 治理层
Week08	检索服务与 API 交付	质量层 / 运行层
Week10	工具层、动作边界、HITL	安全与合规层 / 运行层
Week12	tracing、观测、问题定位	运行层 / 治理层
Week14	发布、回滚、版本治理	治理层

课堂小结

1. 没有 Done，团队最终只能靠感觉协作，项目越往后越容易失真。
2. AI 项目的完成标准，不是“能回答”，而是“能被签字、能被监控、能被回滚、能被追责”。

3. 一旦交付标准写清楚，后面的技术路线、数据路线、观测路线都会被反推出来。

课后思考题

1. 你当前项目里，哪个环节最像“会答了，但还没人敢签字上线”？
2. 如果让业务、架构、合规、运维同时签字，你的系统最可能先卡在哪一层？
3. 你的项目今天有没有一份可以真正拿去签字的 Done 清单？