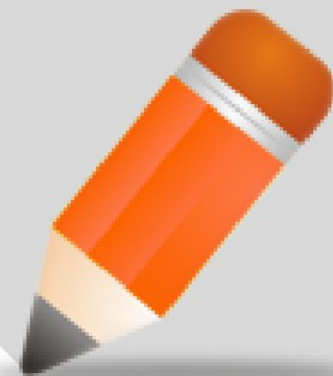




从助手到架构师

迈向 AI 驱动决策模型的高风险一跃

*自主编写决策逻辑在哪里创造价值——又在哪里悄悄侵蚀
让你的逻辑值得拥有的那份优势。*

**DECISION
CAMP 2026**

Aug 25 – Aug 28, 2026

Carole-Ann Berlioz

cberlioz@sparklinglogic.com
联合创始人兼首席产品官

Sparkling Logic

原作：Carole-Ann Berlioz，Sparkling Logic | SparklingLogic 上海信数 译 | 原始

slides： <https://decisioncamp2026.wordpress.com/wp-content/uploads/2026/08/decisioncamp2026.caroleann.pdf>

© Sparkling Logic 版权所有



本次内容

01 典型的 AI 应用历程

自主权如何在没有任何治理决策的情况下悄悄增加

02 两种模式，同一系统

定义 AI 助手与 AI 决策架构师

03 需求梳理

为什么需求梳理应由人主导

04 自主化的风险

当没人验证整个模型（而不只是局部）时，会出什么问题

05 竞争优势

为什么 AI 编写的逻辑会向行业平均水平漂移

06 架构缺口

问题不只是工具缺口：使用方式也必须同步演进

07 护栏与治理

必须先满足的技术与组织条件

典型的 AI 应用历程

1

第 1 个月

" 帮我起草这条准入规则。 "

起支持作用的助手。一条规则，一次人工审核。

2

第 4 个月

" 为这个规则集生成测试用例。 "

仍限于单个组件，仍逐行审核。

3

第 9 个月

" 直接把整个进件（ origination ）决策生成出来。 "

端到端完整编写。审核变成抽查。

核心问题

**" 导航员 " 在哪个时间点变成了 " 司机 "——
而这时又改变了什么？**

两种模式，同一系统

A AI 助手

风险较低 · 作用于单个组件

- 按需解释现有规则与逻辑
- 回答针对决策模型的查询
- 起草一条孤立的规则或条件
- 每项产出都由人审核并整合
- 错误被限定在一个组件内

D AI 决策架构师

后果严重 · 作用于整个系统

- 端到端编写完整决策模型
- 排列规则顺序并消解规则间冲突
- 产出可部署、可执行的决策流水线
- 人工审核缩减为抽查或签字确认
- 错误是系统性的——直接固化进生产逻辑

需求梳理

1 隐性知识没有写下来

做了 20 年的核保人员往往无法按要求说清自己的经验法则。把它挖出来需要灵活的、基于场景的访谈——这是需要技能的人类活动，而不是检索任务。

2 需求本身带有 " 政治性 "

真正的需求收集要在合规、法务、销售和精算等干系人之间斡旋，他们对同一条规则想要的东西各不相同。这需要读懂信任与权力关系，而不是汇总各方表态。

3 发现 " 未知的未知 "

优秀的需求分析师会追问自己亲身踩过坑的场景。这靠的是与实际运营中留下的教训做模式匹配——而不是训练数据里的统计概率。

4 那些从不说出口的东西

未明说的风险偏好、竞争定位，以及 " 大家心照不宣 " 的潜在法律风险，都会影响需求，但只看得到输入文字的模型看不见它们。

这个角色不会缩小——反而更加关键

当需求梳理不完整时，AI 决策架构师不会留下看得见的空白——它会用 " 统计上最典型 " 的假设把空白填上。薄弱的需求梳理在第一条规则起草之前，就已在上游一层助长了 " 平均化 " 问题。

自主化的风险

1 失去整体验证

每条规则单独看都可能是正确的，组合起来却仍可能得出整体错误的决策——条件冲突、永远走不到的分支、悄无声息的优先级错误。AI 助手从不需要跨规则推理；AI 决策架构师必须做到，而现有工具大多无法检查它是否做到了。

2 可解释性 "赤字"

人工编写的模型，每条规则都有可追溯的理由。模型编写的流水线可能在优化一个无法逐条规则拆解的目标函数——你得到的是一个答案，却不一定是站得住脚的理由。

3 测试范围爆炸

验证一条起草的条款是可控的。验证数百条相互作用的 AI 生成规则所产生的涌现行为，则是一个组合爆炸问题，大多数 QA 流程从未为此设计。

4 问责问题

当全自主流水线做出错误决定——错误拒绝、保单定价错误——谁对结果负责？事后他们真的能解释它是如何发生的吗？

竞争优势



风险不只是出错。

不受约束的 AI 决策架构师不只是有把决策做错的风险——它还可能悄悄磨平那些不显眼的规则和边缘情况，而正是它们让你的逻辑成为专有。最终你可能得到一个技术上正确、却让你看起来和竞争对手一模一样的模型。

这种回归并非始于生成环节——它可能始于上游，即不完整的需求（见第 03 节：需求梳理）。

现有架构的不足之处

1 缺乏整体验证标准

多数验证工具只检查语法或单条规则逻辑，而不检查生成时出现的跨规则涌现交互。即便能检查，业务方也很少把 KPI 预期写成文档。

2 出处缺口

追溯一条模型生成的规则为何存在——是哪种训练模式或提示上下文驱动了它——的能力有限，导致审计和监管应对困难。而当追溯工具和可解释模型替人完成这项工作时，风险又变成业务方“偷懒”，不再质疑 AI 生成的逻辑。

3 漂移监控缺口

决策模型会在多轮 AI 辅助编辑中退化或被悄悄覆盖。目前几乎没有针对“决策语义”的版本对比（diff），只有针对资产文件本身的。就算有，业务方会关注吗？

4 激励错位

当前生成式 AI 工具优化的是看似合理、流畅的输出——而不是保留某个组织特有的、不显眼的竞争逻辑。

护栏与治理问题

技术护栏

- ? 任何部署前强制进行整体模拟运行（Simulation） / 测试
- ? 对生成的逻辑提出逐条规则的可解释性要求
- ? 人工检查点按决策后果分级设置，而不是一刀切

治理问题

- ? AI 编写的生产流水线由谁负责？
- ? 一年后你能还原任何一个决策的原因吗？
- ? 需要什么样的审计轨迹，由谁负责维护？

组织准备度

- ? 是否定义了“助手 → 架构师”升级的门槛？
- ? 是否有具名负责人真正审批跨越该门槛？
- ? 差异化约束是被明确编码，还是听天由命？

引擎室还没准备好 " 自动驾驶 " ——至少现在还没有

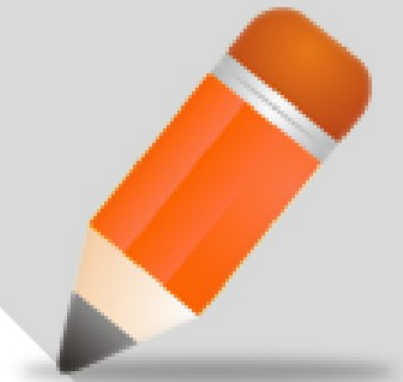
从助手到架构师的跃迁是风险类别的变化，而不是功能升级。
要在规划和治理中这样对待它——而不是把它当作顺理成章的下一个版本。

向决策管理系统（DMS）社区发出的呼吁

这需要共同的标准——整体验证、出处追溯、差异化保留——
而不只是各厂商各自的解决方案。

留给大家的一个问题：

" 你的决策逻辑里，哪一条规则是你绝不希望被 AI ' 平均掉 ' 的？ "



谢谢!

Carole-Ann Berlioz

cberlioz@sparklinglogic.com

联合创始人兼首席产品官

Sparkling Logic

原作: Carole-Ann Berlioz, Sparkling Logic

SparklingLogic 上海信数 译

原始 slides: <https://decisioncamp2026.wordpress.com/wp-content/uploads/2026/08/decisioncamp2026.caroleann.pdf>

© Sparkling Logic 版权所有

