

FDE用AI推动企业研发转型

金融行业

AI Coding

研发提效

企业AI转型

Spec协作

吕昭波 MumuLab 创始人

企业AI转型并非多个模块简单叠加，需要FDE让1+1>2



案例简介

AI Coding逐渐进入企业研发现场后，真正难的往往从“有没有人会用AI”变成了“一个团队能不能真正用起来”。在某基金的研发团队里，开发人员此前各自使用不同的AI Coding工具，个人提效已经发生，企业却很难把这种效率变成团队能力。

本案例中，吕昭波从需求拆解、能力摸底、流程梳理开始，通过连续五天半的线下实践，帮助企业统一工具与方法、调整研发角色与协作方式，并沉淀出一套Spec文档（面向AI Coding协作的规范文档）与研发协作方法。最终，这套方法在公司内部完成了从个人使用到团队协同的转型，也在其他企业的实践中得到验证。

适用人群

- 金融行业研发与技术团队
- 企业研发团队与AI Coding转型团队
- FDE、AI产品经理、企业AI应用负责人

Q1: 作为FDE，你应用AI的具体场景是什么？在这个场景下遇到的具体问题长什么样？

这个案例发生在一家基金公司。它本身是偏基金投研类的企业，会围绕股票、汽车等不同领域开展基金管理业务。我们面对的是一支需求变化非常快的内部研发团队，和传统意义上几个月做一个大型软件项目的团队很不一样。

他们的一个特点是，需求非常零散，也非常快。内部产品经理或者业务部门一天可能会提出好几个需求。对他们来说，AI带来的价值，是把研发人员每天能够完成的功能点从三个提高到十个，和“原来一年做完的项目，现在半年做完”的传统想象并不相同。

所以，在我接触这个项目之前，他们其实已经在用AI了，而且有些人用得还挺深。有人用Codex，有人用Claude Code，也有人使用其他AI Coding工具。问题恰恰出在这里：大家都会用，但企业很难让大家一起用。

每个人都觉得自己的工具最好。“我用这个工具效率高，你让我换另一个，我反而不会干活。”当个人习惯越来越成熟以后，企业如果想统一工具，反而会遇到新的阻力。

而从管理层的角度看，企业需要的显然不只是几个员工个人效率提升。老板希望看到的是整个研发团队的转型，希望AI能够成为一种组织能力，而不再只是散落在员工电脑里的个人工具。

所以我当时面对的核心问题，已经超出“怎么教大家使用AI Coding”这一层，具体是三个层次的问题。

第一，是认知没有统一。大家都在使用AI，但对AI Coding到底能做到什么程度、自己的能力处于什么水平，并没有统一标准。

第二，是工具和方法没有统一。不同的人使用不同的工具，有不同的工作习惯，原来的研发协作方式也没有随着AI Coding的出现发生变化。

第三，是组织方式开始出现新的变化。企业管理层看到一些新的研发理念后，希望进一

步调整研发角色和流程，但这些理念到底怎么落到自己的团队里，并没有现成答案。

这也是我理解FDE和普通技术培训最大的区别：面对这种问题，不能只回答“这个工具怎么用”，而要先判断企业真正要解决的是什么。

Q2：在你参与之前，这个问题过去是怎么解决的？

之前就是产品经理按照原来的方式写PRD，开发人员根据文档进行开发，不同角色按照原来的分工推进。后来AI Coding工具出现了，研发人员开始自己尝试各种工具，于是最早出现的是一种非常自然的“个人提效”。

有人发现某个工具好用，就自己用；有人掌握了新的方法，就自己实践。这个阶段其实没有什么问题，因为个人只需要对自己的效率负责。

但当企业希望从“个人提效”进一步走向“团队提效”时，原来的方式就开始出现限制了。

比如，原来大家习惯使用飞书文档写PRD。后来AI Coding的工作方式发生变化，很多内容开始转向Markdown、TXT等文件。原来一个人维护一份文档没有太大问题，但当整个团队开始围绕这些文件协作时，新的问题马上出现了：十几份文档到底应该怎么拆？

产品经理写什么？前端写什么？后端写什么？测试和运维又分别负责什么？如果大家同时修改Markdown，怎么避免冲突？原来的角色边界还能不能继续沿用？

这些都不是单纯学会一个AI工具之后自然就会解决的问题。

所以，我没有把过去的方法定义成“错误”。恰恰相反，它是适应原来研发模式的一套成熟方法。只是当AI Coding改变了研发方式之后，原来的流程、文档和角色分工也需要重新调整。

这也是我后来越来越明显的一个感受：AI真正进入企业之后，很多时候改变的不只是工

具，还有工具背后的工作方式。

Q3：后来你使用AI解决这个问题的整体思路和关键步骤是什么？

这个项目最开始，客户已经确定采用阿里的相关能力，但“买了之后具体怎么做”其实并没有特别清晰的答案。

所以我的第一步，是先重新摸需求，再考虑准备课程。

我先把不同角色的需求拆开来看。老板关注的是企业整体转型到底能不能发生，研发人员关注的是这个工具到底能不能真正提高自己的效率，产品经理关注的是原来的研发流程怎么变化，HR又需要关注培训过程中每一天能够获得什么。

所以虽然最终交付形式是一次五天半的线下培训，我在设计时始终把它当成一次完整的企业AI转型实践来做。

我当时做了几件事情。

第一件事，是先把问题边界摸清楚。

在正式开始之前，我准备了大量素材，因为当时并不确定大家到底处于什么水平。

但这里有一个非常现实的问题：企业自己描述的能力水平，经常不等于现场真实能力水平。

所以我后来越来越重视一件事情：不要完全相信前期调研，一定要尽可能通过真实操作、快速原型和现场交流去验证。

第二件事，是把培训变成一条完整的业务故事线。

五天和半天完全不是一个概念。

如果只是讲工具，可能一天就结束了。但如果希望企业真正发生变化，就需要从前到后

设计完整路径：为什么企业要转型、研发人员为什么要使用、工具应该怎么进入原有流程、角色怎么变化、最终又怎么沉淀成企业自己的方法。

所以这五天半里，我既是在讲，也是在观察；既是在培训，也是在咨询。

现场发现的问题，我会继续往后调整。

第三件事，是针对真实研发流程做定制。

这是整个案例里我认为最有FDE特征的一部分。

比如，原来的研发协作围绕一份PRD展开，现在转向多份文档协作，角色分工就不一定还能直接套用。

于是我们开始重新设计Spec文档。当时行业里其实没有一个特别成熟、统一的标准可以照搬，我也不知道自己设计出来的东西是不是最优解。

所以我没有一开始就把它当成“标准答案”扔给客户，选择不断拆解、逐步抛出、跟客户确认，再根据反馈继续调整。

最终，我从0到1整理出了一套Spec文档。客户一开始甚至以为这是阿里内部已经提炼好的文档，但实际上，这套东西就是我在这个项目过程中结合客户实际情况逐步搭出来的。

第四件事，是在现场不断调整，让方案跟随现场反馈走。

我们给这家基金公司一共做了三期。

第一期最重要的价值之一，是让我真正看到企业现场到底是什么样。第二期开始，就能够根据第一期积累下来的经验继续调整。到了后续阶段，客户的业务人员、产品经理也逐渐进入进来，整个项目从单纯研发培训变成了更完整的组织转型。

这也是我现在比较认可的一种FDE工作方式：先把整个链路跑起来，再根据真实反馈不

断迭代，第一次不必追求100分。

Q4：相比传统方式，使用AI后的实际效果如何？

这个项目的效果，不能简单理解成“用了AI之后公司多赚了多少钱”。

这家基金公司没有向我开放可以对外披露的完整ROI数据。但从组织变化来看，效果是比较明确的。

过去，AI更多是一种个人能力，每个人都有自己的方法和工具。

后来企业开始统一工具，并进一步统一研发协作方式。AI不再只是“某个开发人员会不会用”的问题，而开始进入团队共同的研发流程。

更重要的是，我们把过程中形成的Spec文档和相关方法沉淀了下来。它不再依赖某一个人现场演示，能够在企业内部继续复用。

后续在另一个制造业客户的实践中，也出现了比较明确的短期数据：三个小组使用这套Spec Coding方法和模板后，统计显示写代码的时间节省了约1/3，写文档的时间节省了一半多，测试时间则基本持平。

这个数据是客户在短期几天内做的内部统计，因此我认为它可以作为真实反馈，但不应该被理解成长期、完整的ROI结论。真正有价值的效果评估，还是需要持续两周、一个月甚至更长时间。

所以对我来说，这类项目最重要的结果，是完成了一个变化：从“大家各自使用AI”，走向“企业开始形成一套共同使用AI的方法”。

Q5：在部署AI过程中，是否遇到了新问题或挑战？你是怎么解决的？

第一个挑战，是AI发展太快，而行业标准还没有跟上。

当时大家都在讨论Spec、AI Coding、研发组织转型，但很多方法还没有经过足够长时间的大规模验证。

我自己也会担心：万一我整理出来的这一套方法不适合现场怎么办？会不会刚拿出来就被客户否定？

我的解决办法，是把不确定性暴露出来，然后通过小步验证降低风险，不假装自己已经知道答案。

我会先做一部分，再和客户确认；再结合外部信息和阿里内部能够获得的信息继续调整。发现不对，就修改。

这其实也是我觉得FDE很重要的一种能力：在一个还没有标准答案的领域里，不能因为没有标准就不做，也不能因为自己是技术人员，就假设自己一定是对的。

第二个挑战，是前期调研经常失真。

有一次，我们提前了解到客户团队使用AI Coding的能力比较深，所以准备了很多高阶内容。结果真正到了现场，发现大家连一些基础问题都没有解决。

那怎么办？只能现场降级，从基础开始。

反过来也一样。如果提前判断团队能力一般，结果到了现场发现大家已经用得非常熟练，我就会把原本准备的“压箱底”的内容拿出来。

所以现在回头看，我觉得现场能力本身就是FDE的重要组成部分。

第三个挑战，是企业一开始就想“大投入”。

比如一上来就买模型一体机、做模型微调，或者强行推翻原有全部工作流。我反而觉得这些方式比较危险。

如果业务可以用轻量级方式快速验证，比如API、SaaS或者Agent等方式能够先跑起来，我更倾向于先验证业务价值。除非企业有明确的合规、安全等要求，否则没有必要一开始就把投入做得特别重。

投入应该由轻到重，先把业务价值验证出来，再逐步加深。

Q6：根据你的经验，我们怎样才能更好地把AI部署到实际业务中？

我现在越来越觉得，企业部署AI最容易犯的错误，是把AI当成一个单独的技术项目。实际上，真正落地的时候，技术只是其中一部分。

第一，先理解业务到底想解决什么问题。

很多企业会说：“我要做AI转型。”但这句话本身没有办法指导落地。到底是个人提效、团队提效，还是研发流程重构？不同目标对应完全不同的方案。

第二，不要过度依赖前期调研，快速做原型、跑真实流程。

因为AI的使用水平很难通过问卷准确判断，所以我现在更倾向于快速做原型、快速跑一遍真实流程。哪怕第一次只有七八十分，也比坐在那里把方案讨论到100分再开始更有价值。

因为真正跑起来之后，背景、痛点、方案和限制条件都会变得更清楚。第二遍再优化，就有机会从七八十分做到八九十分。

第三，接受AI落地是一个持续迭代的过程。

尤其是在AI发展特别快的领域，没有人能够保证自己第一次设计出来的就是最佳方案。与其追求一次性解决所有问题，不如先把第一轮跑起来，然后不断修正。

最后，也是我认为FDE最重要的一点：**不要只站在技术人员自己的视角看问题。**FDE需要同时理解技术、业务和客户。

跟老板沟通，老板关心的是组织效率和投入产出；跟研发人员沟通，他们关心的是工具好不好用；跟产品经理沟通，他们关心的是研发流程怎么变；跟HR、采购、运维沟通，又会是完全不同的话语体系。

所以我觉得FDE更像一个“翻译官”：把标准化的AI能力，转化成具体企业、具体团队能够真正使用的东西。

我把FDE的价值归纳成一句话：把AI的能力翻译成企业真正能用、能跑、能沉淀的工作方式，比单纯把一个AI工具带进企业更重要。