

创新的零和博弈：在不确定领域徘徊的软件供应商

过去三年我作为主要的咨询顾问，担纲了两家世界500强企业的敏捷转型项目，主导软件工程改进和交付能效提升。在转型过程中，作为利益无关方的专业顾问，为客户解决一系列最为复杂和重大的交付争议之后，我来谈谈一个急需被重视的普遍性问题，即在当前企业服务领域存在的认知差异，导致IT投入和服务成本产生了巨大的浪费，最后形成一场甲乙双方的零和游戏。

这个认知的差异，主要在于对“**不确定性**”的价值与成本的把控。在下面的几个咨询案例中，我们尝试在双方表述中找到理解对方立场和彼此呼应的部分，从而建立起全局的共识。

不确定性产生的浪费

案例一：某五百强集团互联网金融事业部

接手这个咨询项目时，面临的最大问题就是虽然开发部门做到了每周双发布，需求还是严重积压，业务部门满意度低下。我们尝试通过建立持续交付体系来提升产品交付效率与质量。但在咨询过程中发现，开发团队即使有大量的功能测试完全通过，也有完善的灰度部署机制，仍然无法向生产环境发布，必须采取复杂的特性分支来进行代码管理。该事业部的反馈是，业务部门经常让他们开发大量的功能，最终又决定不上线，让开发测试疲于奔命，开发组长也浪费大量时间在发布代码的挑拣上。深入调研后发现，这也是业务部门的一大痛点：每年开发了约40%没有上线的废弃功能；近一半的产品开发预算，由于对最终产品功能设计的不确定，实际上被浪费掉了；而业务部门真正的目标，比如对标行业标杆，提升产品在行业中的排名，也受困于当前开发浪费的拖累。

案例二：某五百强企业IT某产品线

此前已针对该产品线进行了半年的敏捷咨询，明确了需求管理的标准：特性分解、故事卡与验收标准。回访时发现，产品的某项新特性，为就合理的工作量达成一致，竟然投入了80人天来评估；由于工作量标准的不确定，针对需求的估算成本几乎已经超过了需求的开发成本。

案例三：某科技初创公司创新型产品

从创意产生已历时两年，延期半年，还未有过交付，成为一个典型的失败项目。供应商的成本已经远远超出了当初的合同金额，并且坚持为每个变更走完商务流程再开发，继续拖延了版本整体交付时间的同时，也在抱怨变更引发的商务流程导致开发人员空载，甚至一度想中止合同关系。这给客户造成了数百万级的直接损失，而市场机会的损失更难以估算。

面对这些显而易见的成本浪费，客户希望寻求专业的解决措施。

不确定的价值

在新功能的开发过程中，客户坦言自己的无力：“其实我们也并不知道用户想要什么。”另一位来自传统工程领域的客户观点则一针见血：

我们需要的是供应商能够快速响应我们的变更需求。

在不确定领域，特定功能的价值在真正被市场验证之前，其实是不确定的。而一旦接收到可靠的用户反馈（对新产品而言，更可能的是针对某些功能的抱怨），却需要及早响应。很多供应商只定位于完成开发编码，使用承载“**功能操作描述**”的需求说明，预设的业务价值并未明确，导致开发测试团队的理解偏差，交付出错误的功能；而更大的困境在于，不理解创新产品的试验和试错机制，无法通过识别与创造价值来管理此类项目的成本和客户满意度。

我曾经在一次会议上听完供应商汇报迭代进度后问：“你们是否知道在这个迭代完成A功能，可以给客户节省多少人力管理成本？”供应商面面相觑，客户很尴尬地说：“这个问题只有BA能回答，他们都是DEV，不知道很正常。”而在另一起会议中，供应商高层听完客户讲收集用户行为对他很重要时，非常吃惊地问：“不好意思，你刚才说你拿到用户访问信息后功能还要修改，这个功能是修改成什么样？怎样才算把功能改到满意？”比起理解客户的真正目标，抑或是新产品投放市场阶段的合作模式，供应商更关注的是功能修改的成本是否由己方负担。

寻找确定性

除了上面提到的一个鲜明的客户观点——**快速响应变化**——之外，还有一个双方共同的诉求：**风险可控地创新试错**。

而很多客户都误解了在不确定情况下“**控制风险**”与“**控制成本**”的区别。

去年我在一家正在进行“**数字化转型**”，追求高速增长的大型集团子公司担任转型项目负责人。令我吃惊的是他们对“**精益**”概念的误解。鉴于中国企业的现状，精益理念中追求尽善尽美只能作为宣传标语，但“**消除浪费**”这样的成本优化手段则在很多企业中成为切

实可行的工具。而该企业理解的精益并非是在项目过程中寻找和减少浪费，而是在早期尽可能减少投入成本。贯彻这一理解的项目，极有可能因为投入不足而无法保障基本产出而失败，浪费的时间与人力资源还会影响整个年度多个其它机会项目的最终有效促成与落地。

客户教训：金融社交产品案例

风险可控地创新试错是从组织流程、技术平台和交付质量保障的整体体系，而快速响应变化的能力则是基础。在给某五百强集团互联网金融业务部门高层讲解基于实验式发布测试用户兴趣和新功能市场价值时，客户响应强烈，意识到过去的一年在金融社交产品的推出上，走了严重的弯路。

在该集团成功运营旗下互联网金融产品到1200万用户时，业务部门希望针对如此大规模的用户推出新的社交功能。而前文提到，该金融产品开发部门已经严重负荷，根本无法保障其它侧翼功能的快速推出，业务部门新组建了金融社交产品团队，而原金融产品的平台扩展支撑并不好，在种种内外部压力之下，社交产品团队从零开始，终于完成了所有功能，而业务部门将该产品投入市场却发现，没有用户，又开始组建了一支社交产品运营团队，而绩效表现并不出色；随后，该部门又转为寻求技术方案，将原金融产品用户持续同步到社交平台……前前后后，花费一年的人财物力，都没能开发出一款成功的产品。

而风险可控的模式在于，基于现有平台推出新功能并且测试功能在试验用户中的表现。随着用户正面反馈的增加，逐步完善功能并且将使用面扩大到整个用户群体。以这个社交平台为例，在早期可以先尝试在金融产品中推出一两个社交功能，并向部分用户开放，试验目的可能是：

- 测试用户是否对社交功能感兴趣。
- 寻找潜在的金融社交用户细分市场。

一旦缺乏足够的新功能使用者，业务部门可以选择及时停止。这样就避免了投入方向错误的风险。

但这一切，都是基于既有团队与技术架构对需求的高响应力。

探索确定性下的新模式

当下很多希望提升市场响应力的企业，都在进行组织级别的敏捷转型。对于规模大、层级多的组织，无疑是一场旷久反复的变革。

有趣的是，以供应商合作为主的企业，寄望于将业务分包出去降低管理成本；而以供应商分包为主的企业，在成本完全失控的情况下提出改为驻场合作模式。

运用有限的资源达成业务目标，这其实是管理者的职能与职责；而众多项目的失控，则是当前采用的管理方法的失效。包括针对咨询与转型项目客户所作出的巨大投入与收益的不成正比，都值得我们反思咨询管理方法本身。

功能开发 vs 服务提供

按照传统对软件项目的理解，主要成本分布在功能开发与测试等几大过程耗费的工时。而不少软件项目最终的失败，均是因为忽略了一个经济学概念：“**外部性**”。外部性是指，在直接的买家与卖家之间，还有第三方受到交易的影响。在软件项目甲乙双方被忽略的外部性，即是最终用户。当前企业应用领域的一大转变，即是由内部系统转向外部，甚至面向最终用户进行创新，以保持市场份额或获取新领域的用户。借鉴经济学对外部性的消除手段之一，“**降低交易成本**”，通过敏捷方法，将项目所需的各个角色组建为面向一个产品领域的全功能团队，减少协同、沟通与等待成本，可以有效地降低整体的交付成本。

新的思考是将最终用户满意度敏感的开发项目视为“**服务**”而非“**功能开发**”。在这类项目中，我们需要按照服务响应能力的要求寻找合适的团队成员，并直接针对“**需求响应周期**”、“**产品积压量**”和“**交付满意度**”对团队进行结果衡量，评估最早发布时间而非计费工时。在这种模式下，关于工时的争议将被避免，团队为客户或最终用户的满意度负责，将最大限度减少过程中的内耗，当然，一定程度上也意味着信息透明与向下授权：企业的顾客将成为某个团队的直接顾客。

合同期限 vs 尽早发布

至今仍然有许多供应商甚至客户并不知道早期MVP（最小可用版本）尽早发布的好处，即使未找到早期用户，也能够通过尽早部署发现产品技术上的致命缺陷。以上面提到的创新产品项目为例，供应商依然坚持按合同条款约定在项目验收完成才能交付所有源代码，而我已经快十年没有听说过超过三个月未有发布的项目了！当然，并不清楚国内其它分包类项目是否存在同样情况。

关于实验式发布，是许多遗留系统的技术改造难点；初期试验用户的选择，也是许多缺乏用户数据的企业做到数据运营的痛点；但最大的阻碍，还是在于转变思维。

短期成果 vs 长期成本

我将软件项目保持快速交付的秘诀归纳于：

- 采用成熟技术。
- 适合业务的架构设计。
- 模块化开发分工。
- 控制技术债务。

- 团队可持续成长。
- 阶段性引入新技术。

而这几项除了成熟技术之外，都需要时间和管理成本的投入，新技术的引入并非一味追求新技术，而是懂得技术管理：即对一项技术的可行性、学习成本、替换成本及潜在收益进行评估并制定出技术路线图。许多项目在早期追求快速出成果，而随着时间的推移花费的成本却越来越高，就是因为技术项目缺乏质量治理产生的大量债务，甚至是技术选型阶段的仓促导致后期全面推翻重来。

长期成本和短期交付的均衡，对产品架构和技术负责人的挑战非常高，这里的应对措施是，借助和整合外部优质资源，通过技术顾问或第三方专业公司来进行产品的高阶设计再转入开发过程。这意味着，在服务市场的进一步专业化，以及高端化，一些小型的精品公司将有机会建立起独特的品牌，成为细分市场的领先者——目前我创立的公司**场量科技**，即在一家传统企业数字化转型中扮演了这个角色，成为新兴市场的吃螃蟹者。

新模式的探索总结：

1. 采用管理与考核服务组织的模式，代替过去的研发交付模式。
2. 基于实验和数据驱动尽早发布早期版本，及时发现问题。
3. 关于产品的长期成本，通过模块化架构快速更新局部功能。

鼓励学习与创新

在敏捷转型的开始有客户反馈说，实施敏捷后供应商很不满意，以前只管写好代码就行了，现在还要一天琢磨理解需求。后来一位顾问同事给我说，有天经过我负责的那个团队，听到开发人员在抱怨，以前一天自己写完代码就完事，现在真费事，来了一个顾问，一天就看我写的代码，看了不满意还让我改来改去，烦死了。

这是一个正在快速变化的时代，尤其在技术领域。最近的一位客户说，我本来希望供应商能够跟着我们一起学习，一起提升，最后能开发出好的产品，但最后发现，他们对我们的要求就是，不要让他们学习，不要改变。而我之前负责的一位客户的供应商，在经过一段艰难时间后给我说的话很值得参考：

以前我们招聘的人考察的是这个人是否诚实可靠，但现在发现必须要有很好的业务理解能力。现在对我们来说，确实需要多想想甲方想要什么，提前想出来、做出来，主动向甲方展示我们的能力，没有创新能力，我们的团队会很被动，很累。