

01 | 灵魂拷问：如何利用敏捷思维更好地解决实际问题？

2020-01-06 宋宁

说透敏捷

[进入课程 >](#)



讲述：宋宁

时长 16:27 大小 15.08M



你好，我是宋宁。

今天是专栏的第一讲，在为你讲解怎么做敏捷之前，我想先用几个案例给你讲讲敏捷的价值。

敏捷在国内推进得如火如荼，我知道很多公司的研发团队都在使用敏捷，并且尝到了其中的甜头。但也有些公司，看别人做得挺好的，就照搬过来，结果却不奏效，因为探索地不是很成功，就觉得敏捷不好用，不适用于他们公司，开始怀疑敏捷的价值。

作为一个过来人，我想谈一下我的看法：我认为敏捷确实是很好的，只是推进它确实不容易。

为什么这么说呢？因为敏捷毕竟是一场变革，它本身涉及很多人的因素，比如人的习惯、团队文化的改变等等，而且它的核心点是持续改进，可以说是一场没有终点的旅行。况且它有一定的章法，但是若想运用好它，你又不能拘泥于它的章法。如果你只是了解敏捷的表面做法，就开始邯郸学步，那注定会失败。

所以，在这里我更想让你了解的是敏捷的思维，多理解一些 why，你自然就能够把敏捷融入到工作中，举一反三。但切记，千万不要为了敏捷而敏捷，所以在这篇文章中我也并不想说服你全盘使用敏捷，如果你觉得敏捷方法多，有些繁杂，不是所有都用得上，那也可以从中借鉴一些去解决你的实际问题。总之，能够更好地解决问题就好。

刚才我说我觉得敏捷很好，这话不具体，接下来我想和你讲一个故事。

记得那是 2007 年~2008 年，我们有一个项目是负责某大型通信公司印度尼西亚工厂的 SAP 实施。在简单了解需求之后，我们制定了项目初步计划，包括预算、人员和进度计划等等；目标是用 10 个月完成项目，包括需求调研、开发、测试、上线以及用户培训。

我们的项目经理很资深，公司也有一套成熟的项目管理模式，于是我们就按照公司规定的项目管理规范来开展工作。

我们花了 1 个月的时间来进行项目的前期筹备，包括组建团队。之后，我们花了 2 个半月的时间做完了需求调研，又花了 3 个月做开发，开发完成后交给测试，测试花了 2 个月完成，然后交付给客户来验收。验收之前，我们仿佛看到了胜利的曙光，就等着庆功宴了。

没想到，噩梦才刚刚开始。当客户看到系统并真正试用时，开始觉得这也不行、那也不可，提了大大小小 50 多条修改意见。记得当时验收会议结束时，客户方相当不满意，就差拍桌子了。

接下来的日子里，我们的工程师开始加班加点地改，项目经理和需求工程师还有技术负责人去跟客户斡旋那些改不掉的需求或问题。等项目真正上线时，工期比预计延期了 50%，预算也超支了不少。所以项目最后虽然上线了，但整个过程客户相当不满意，用客户的话来说：“这是勉强上线……”

好了，到这里，你可以停下来想想，上面的项目到底有什么问题？到底是什么原因导致了这个项目的失败？

这之后我一直在思索，直到我了解了研发模式中瀑布模式和敏捷模式的差别，才真正找到答案。

我上面那个项目所采用的研发模式就是典型的瀑布模式，也就是说，研发的整个工作流程都是按顺序进行的。像上面的项目过程，先做需求调研，再做开发、测试、验收和上线，所有的工作都是串行的，只有达到下一步的准入标准，我们才进行下一步工作。

瀑布模型是 1970 年温斯顿·罗伊斯 (Winston Royce) 提出的，并且一经提出即被各大公司拿来当做标尺使用，20 世纪 80 年代更是狠狠火了一把。因为它简单粗暴容易推广，在没有其他更好的研发管理理论情况下，使用它无疑比没有管理套路导致的野蛮生长要好得多（如果你想了解更多，可以阅读极客时间出品的软件工程专栏）。

然而，瀑布模型被广泛投入使用之后，真正的一线开发人员却感受到各种束缚，研发效率受到阻碍。就连其提出人温斯顿最后都承认说：“在我的经验里，瀑布模型在大的软件开发中从没真正起到作用”。

在项目实践过程中，瀑布模型经常在以下 3 个方面饱受诟病：

1. 研发周期过长，导致研发跟不上业务发展的节奏。在瀑布模型中，所有的工作都是串行的，只有前序环节完成，才能展开后序环节，大量人力与时间成本都在这段时间里被白白消耗，等产品开发出来推向市场，很有可能市场已经没有了，或者业务已经发生了很大变化。

2. 研发不能很好地响应需求变化，导致客户满意度低。要知道，我们很难在设计之初就把所有的需求都想清楚，需求又是不断变化的，因此客户在看到真正的产品出来之前，对产品很可能是无感的，正如上面项目中的客户，他们看到并试用了产品后才感觉到这里应该这样，那里应该那样。瀑布模型是在项目最后才一次性地向客户交付产品，当产品被开发出来之后，客户才发现他们需要的不是这个东西，这是非常可怕的事情。

3. 不能很好地管控风险。因为是研发最后一次性交付，所以项目中的很多风险在前期很难被完全识别出来，最后发现时再想处理就要付出更大的代价。

那么，为什么“瀑布模型”会存在这些问题呢？

如今我们回过头来看会发现，软件研发和传统的生产管理不同，它的产出物具有强烈的不确定性。而瀑布模型，其流程和步骤都是规定好的，且计划与生产分离，说白了更适合确定性高的工作。那么在这种不确定性很高的研发工作面前，我们以处理确定性高的工作的方式和流程来进行管理，毫无疑问是不奏效的。

因此，自 20 世纪 90 年代起，软件业界陆陆续续有很多大师开始探求新的、轻量级的、适合软件研发的管理方法。2001 年，他们聚集在美国犹他州，将他们共同的方法高度提炼了一下，这就是“敏捷宣言”。

回到我上面的那个项目，如果当时能用敏捷方式来进行研发，或许就不需要延期那么久、预算也不需要超支那么多、客户也不会一直阴沉着脸。再反思一下这个项目，如果让我重新做一次，我至少会选择下面这 4 个点来进行改进：

1. 从大的组织结构来讲，我们的团队应该组成一个个小的特性团队，团队是固定的，成员也基本是固定的，这样项目来的时候我不需要花费那么长的时间去组建团队，磨合团队。
2. 从需求梳理的过程来看，我不会一次花 2 个半月去梳理并在最后才给客户看我们梳理的结果，我会边梳理边跟客户交流。
3. 我会把需求按优先级排序形成需求池，迭代地进行研发。
4. 我会让客户积极地参与我们的研发过程，包括前期的需求调研梳理和后面的开发测试上线，在做的过程中迭代有产出时就让客户来验收，让他们提意见和建议，以便我们在后面的迭代随时调整。

有了上面的经验，在我后面的项目中，我就真正地去尝试敏捷，并切身感受到了敏捷带来的好处。下面我就再讲一个我亲身经历的、一家初创公司敏捷实践成功的故事。

那是 2012 年，我到一家初创公司去做项目管理。该公司研发中心有七十多人，我一到任，就听到来自业务部门的各种抱怨，我在拜访他们业务部门副总时，他对我说：“宋经理，你们研发部门交付特别慢，我们一个需求要等好几个月才能做好，做好了，我们的推广时机也过了。做得慢不说，做的东西也真是不怎么好，做好了给我们看时，我发现做的根本不是我想要的东西...”他给我讲了一通问题，并殷切希望我的到来能给公司的研发带来新的改变。

我决定在接下来的一周，调研一下，看看怎么应对。看了一圈，我发现该研发中心在刚开始运作时没有任何流程，也没有任何章法，研发过程随意，出了很多生产 Bug。于是在公司总经理的要求下，大家做了一套流程，本想认真执行，结果执行下来效果也不好，不仅交付速度变慢了，做的需求也不符合业务的要求。怎么回事儿呢？原来他们使用的是瀑布模型，有了前面的经验，我说服了公司领导，带着这个研发中心做了敏捷转型。

那么，我是怎么做的呢？我先给研发中心加业务部门的所有人做了培训，又将组织做了变革，将他们分成了一个特性团队；接下来，我把大需求拆分成小需求，对需求列表进行梳理，排列优先级；最后，我又让业务部门参与进来，迭代地进行研发，每个迭代结束后交给业务人员验收，提反馈。

使用敏捷后的效果还不错，我们的需求流动快了，研发效率提升了，Bug 减少了，业务部门满意了，并博得了公司总经理的赞许。

上面就是我的一些项目经验，相信你已经感受到了做敏捷带来的好处，事实证明也确实如此。我们可以看一下业界的统计数据，根据 VersionOne 的最新统计，97% 的受访者报告他们的组织采纳了敏捷这种开发方式。公司在提升自己的交付能力，提高响应变化的能力，改善协作、减少项目风险等时候，都想到了采纳敏捷，并在这个过程中艰难地探索。

说了这么多，也许你会说，敏捷听起来挺好的，但我还是觉得我们公司用不上。“敏捷到底怎么用”这件事情，我会在后面的文章中为你一一解答。这里我想说的是，**其实有很多公司，他们都在有意或无意中使用了敏捷的一些实践**。Google 就是个典型的例子，虽然它不对外标榜自己在做敏捷，但其实你可以在他们的文化或项目中随处看到敏捷的影子，比如他们的工程师文化跟敏捷所倡导的以赋能和信任个人为中心的文化就是非常契合的。

类似 Google 这样的公司其实不在少数，虽然它们并没有全盘采纳敏捷的所有方法，但是在其做事方式上体现了敏捷的思想，他们在用敏捷的思想来改变自己的一些行为，来达成公司的目标。

比方说，有的公司会考虑加快反馈循环来提升流转效率；有的公司会把自己所有事情按照价值和优先级来排序，定期整理自己的工作列表，就像管理产品待办事项列表一样来管理工作，让员工把精力放在更有价值的工作上；还有的公司引进了很多在线协作管理工具来加强协作等等。

所以，是否打着敏捷的名头，是否冠以敏捷，这本身是无所谓的，只要我们能够用到敏捷的一些方法来帮助到自己的公司或项目就好。

总结

凡事都有两面，敏捷也不例外。前面我花了很大的篇幅带你了解了很多敏捷的益处，但敏捷也不是万能的，不是所有的问题都能用敏捷来解决。比方说产品本身的容错率就很低，不允许试错，用敏捷的话与投入产出不成正比，这就不必非要用敏捷方式来做；或者说公司需要深远地考虑战略问题，那么不去考虑布局，妄想着通过导入敏捷把战略思考省略掉，这也是不现实的。

另外，说到敏捷不利的地方，我觉得主要问题应该是“听起来简单，但实施起来并不容易”，敏捷实施起来具有一定的复杂性，这也是为什么敏捷虽然被提出来接近 20 年，在落地的过程中，直到现在，还是有很多的争吵和讨论，也还是有很多人在不断地摸索。

但不管怎么说，我觉得积极吸纳敏捷带来的好处都远超你的想象。毕竟我们现今的世界，已进入 VUCA 时代，我们正面对着一个易变（volatility）、不确定（uncertainty）、复杂（complexity）和模糊（ambiguity）的世界，而敏捷快速响应变化、允许试错、小步快跑的特点无疑是符合时代潮流的。

从我的角度来说，我希望通过这个专栏，帮你澄清一些对敏捷的误解，让你可以深度了解敏捷背后的意义和原理，了解它的一些做法，并能够让你结合你自身的公司或项目特点，结合你的痛点，借鉴到它的一个或者多个方式，来帮助到你的公司或项目完成预期目标，那么我们专栏的目的就达到了。

思考题

结合我们今天讲的内容，我想请你来思考一下，你在研发过程中碰到过哪些难解决的问题呢？你有过利用敏捷思维或敏捷方法解决实际问题的经历吗？欢迎你在留言区与我分享你的疑问和思考。如果你觉得这篇文章对你有启发，也欢迎你把它分享给你的朋友，我们一起来探讨和学习。

点击参与 

打卡10天，带你 从原理到实战落地敏捷



扫一扫参与小程序话题



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 开篇词 | 重识敏捷，让你的研发管理少走一些弯路

下一篇 02 | 老生常谈：你真的知道敏捷到底是什么吗？

精选留言 (31)

 写留言



丁丁历险记

2020-01-07

一定要早点把客户拉下水。让客户早点确认各种细节。这样就不会后期唧唧歪歪了。

作者回复：赞一个

 2

 13



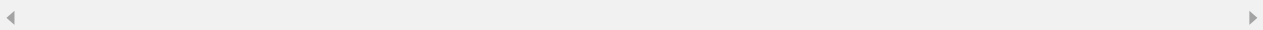
李永智

2020-01-07

敏捷方法使用过，虽然效率有提升，开发理解需求有极大改善，但也做了测试驱动，尽管代价不小，发现有些产品交付质量差，经分析，主要是人员水平造成的，应该是敏捷相比瀑布开发对人员的要求多和高，但现实人员更换受限的情况下，如何提升？或者是我上述的归因有偏差，那问题可能出在哪里，针对这些，有啥应对原则？

展开 ▾

作者回复: 这是个好问题, 关于人员培养的问题是个永恒的话题。这里面涉及2个方面, 一个是态度转变, 一个是能力提升。态度转变的话我们其实有很多方法可以让员工有参与感, 参与到转型过程中, 如后面05会有一些方法, 让员工进到我们的转型过程, 有参与感, 同时也要意识到自己的能力需要提升, 这样很多员工都有提升的动力; 第二个问题, 能力提升的问题方法也非常多, 可以组织拆书会等等快速地了解一个领域的知识, 也可以互相之间分享, 老带新, 反讲, 建立辅导机制, 能力高的辅导能力低的等等, 非常多的方法, 只要人发动起来, 方法总能找得到。最后记住一点是, 人毕生都是在发展的, 减少对人的刻板印象, 鼓励发动他动起来学起来跟上来, 这个很重要。



💬 2

👍 7



qinhua-冰糖橙自产自销...

2020-01-07

特点: 快速响应变化, 允许试错, 小步快跑。

做法: 固定小团队, 优先级的需求池, 客户参与, 实时验收。

展开 ▾

💬

👍 4



王刚

2020-01-07

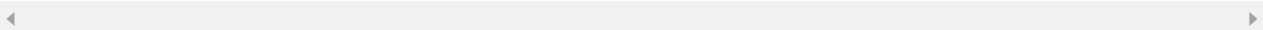
老师, 请教几个问题:

1. 团队拆分为特性小组的标准是什么? 多少人为一组为最佳呢?
2. 使用敏捷模式, 开发前期的接口文档要不要省略呢?
3. 业务参与需求分析, 是不是就是确定原型图的过程?

展开 ▾

作者回复: 第一个问题, 我们在04里有一些关于组织的说明, 特性团队拆分在敏捷里如上面朋友所说, 确实没有固定的死的标准, 但有些原则可供你参考。一个是团队大小, 是5-9个人, 大了要进行拆分; 第二个是全功能团队。

第二个问题, 文档的留与否, 需要根据实际情况来做剪裁, 在剪裁前, 评估文档为产品开发带来的价值大小和实际工作量, 一般而言不建议省略接口文档, 接口文档是开发中的重要文档, 但如果团队找到了更好的工作方法替代这个文档除外。



💬 3

👍 3



阡陌

2020-01-07

一直想尝试，但是总是事与愿违。说到底还是需要公司高层的支持和底层人员的不断尝试才行。



3



桃子-夏勇杰

2020-01-07

第一步组建特性团队就是一个难题，这是一个思维方式的转变，改变了开发团队原有的开发习惯和思维惯性。宋老师是如何说服团队做这样的转变的呢？

展开

作者回复: 思维方式转变确实是比较难的，仅靠口头说服是远远不够的，所以除了宣讲以外，我采取的方式有很多：带他们参观已经转型的团队，有一些感性认识；必要的时候带他们打个胜仗，团队在不同阶段需要的辅助也是不同的，还有我们在05团队试点里的一些方法等等重要的是要让他们觉得是他们需要改变而不是“我”作为咨询师或者敏捷推动者让他们改变。



1



2



吴言乱语

2020-01-07

利用敏捷的方法缩短项目开发时间是我们一直努力的方向，公司在高速成长，产品迭代必须紧跟业务发展，常常遇到的情况是业务部门临时的排版需求上线时间。

在实现过程中，产品经理会在同业务确定需求完成后的第一时间组织立项会议，UI，开发，测试共同参与，进行需求宣导。

接下来，...

展开



2



solaris

2020-01-13

1. 研发过程中遇到的难题

1. 团队有在跑scrum，但是感觉成员态度比较随意，并且容易乐观估计时间点

2. 需求和顶层设计，有时候会面临调整，导致大范围改动，成员抵触情绪比较大，应该如何解决呢？

3. 团队沟通不够充分，随着迭代次数变多，对系统模型、设计、实现的认识不太对齐...

展开



1



1



海棠

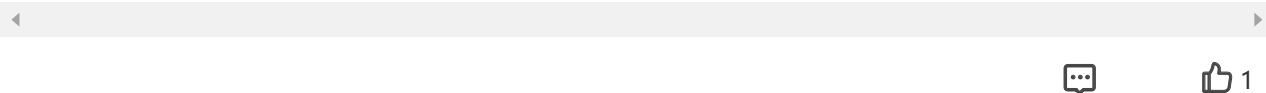
2020-01-07

特性团队对于团队成员的要求其实是非常高的，自己的公司也是FT制，但距离理想的FT状

态还差很远，感觉更像是组件团队与特性团队的中间态。所以，不仅推行敏捷难，转型特性团队也很难，在推行FT化的过程中，有什么心得吗？

展开 ▾

作者回复: 首先是组织设计问题，其次是人员培养问题和带团队的问题，我们后面有涉及，细看一下咱们专栏的实战篇



桃子-夏勇杰

2020-01-07

什么是敏捷思维？要解决什么问题？首先要回答好这个问题，否则，不存在所谓的灵魂拷问。



再不吃胖我们就老了

2020-01-07

不管是瀑布还是敏捷，对于人员的要求都是有的，只不过敏捷因为是快速迭代的，要求会更高。N年前接触敏捷，今天看完这篇发现现在的项目已经是敏捷式的开发了。现在大家都对MVP、迭代这些习以为常了。



escray

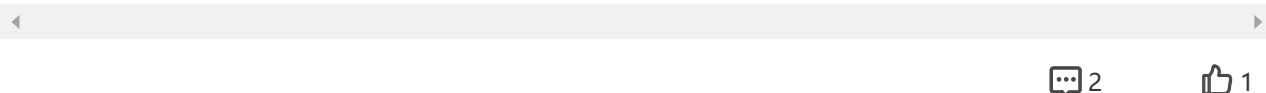
2020-01-07

我对于，“敏捷确实好，但是推进不容易”这个说法深表认同。如果真的落实敏捷，对于研发人员的要求还是比较高的，对于每个开发的工作会有定性和定量的尺度，上班摸鱼就没那么方便了；对于组织架构来说，可能也需要进行一定的调整，比如作者说的划分小团队，但是更主要的可能还是思想上的变革。

...

展开 ▾

作者回复: 现实中没有完美的敏捷模式，虽然这是我们一直追求的目标，但大家都走在路上，推进敏捷的方式有激进式的“貌似一步到位”，但其实后面也有很多工作，另一种渐进式的则是解决一个旧问题出现一个新问题，其实一直在做持续改进，有的企业先做了“形式”上的敏捷，后面发现痛点之后继续改进一步一步地推进了更多的技术实践来助力研发





2020-01-13

我先给研发中心加业务部门的所有人做了培训，又将组织做了变革，将他们分成了一个特性团队。

拆分成特性团队可以详细的举个例子么？

展开 ▾



J

2020-01-13

作为研发人员，宋老师列举的案例是真的感同身受。当时团队没有任何规范的研发流程，并且没有测试。按照瀑布模式，做出来的东西很自然遭到客户投诉。

后面复盘以及处理各个环节如需求挖掘，沟通力度，用人方向等所暴露问题。

而为了整顿就引入了敏捷开发的概念，各个环节的问题都没有妥善解决，最后整个系统虽然还是上线了，但时不时会就遭到客户不同岗位角色对于某一功能缺陷提了好几个月都...

展开 ▾



想想

2020-01-12

您好，我遇到的问题是：

- 1.我们采用原型跟用户迭代确认，也形成需求确认书签字确认。但是在具体测试过程中，用户实际使用后，又有新的想法或者原先就考虑完整的，这时也只能继续进行迭代修改。
- 2.如果存在多个模块处于第1个问题时，因为交付模块一直迭代调整中，导致其他任务无法继续执行。由于成员有限，一个人负责多个模块，就会导致项目工期延期。...

展开 ▾



小老鼠

2020-01-12

1，频繁让客户评审需求，客户会烦吗？2，敏捷与DevOps 区别在哪儿？3，好些嵌入式系统，需求固定，听说也有用敏捷的，你认为这种情形一用敏捷或不用敏捷哪个更好。



飞鹏

2020-01-12

敏捷是一种管理方法，但凡能用的管理方法一定是来源于具体实践的。所以对照手册实施管理方法的，往往都是无疾而终。个人经验可以先找一个很小的问题，用最熟悉的工具稍作改善。虽然很小的改善让人无感，但很小的改进操作一样让人无感。重复以上操作，效率和产出质量的提高最终会显现，大家也能感受到参与其中带来的成就感。而这一定是所有人所期望的。

展开 ∨



Fate

2020-01-11

我现在的公司的项目管理方式很随意。可好处就是 先做基本的功能，做完就演示一遍，看看有没有需要改的，感觉只是参考了敏捷的一些方法，不算是纯敏捷

展开 ∨



不瘦二十斤
不改头像

jeffery

2020-01-10

devops和敏捷的区别是啥、听说devops是敏捷2.0

展开 ∨



曾泽伟

2020-01-09

问个问题，我们在使用敏捷的时候，先获取现有的需求拆分然后排优先级，开始做，但有可能因为需求的不清晰导致拆分有问题，甚至整体的设计方案不合理，后期新需求来的时候，发现要花大量的时间去改之前不合理的设计，甚至数据库表设计都有非常不合理的地方，这个时候重写可能所花费的时间太多，不改，后面需求会越来越麻烦，这时候感觉就很无力，不知道老师有没有碰过这个情况，该怎么处理

展开 ∨

