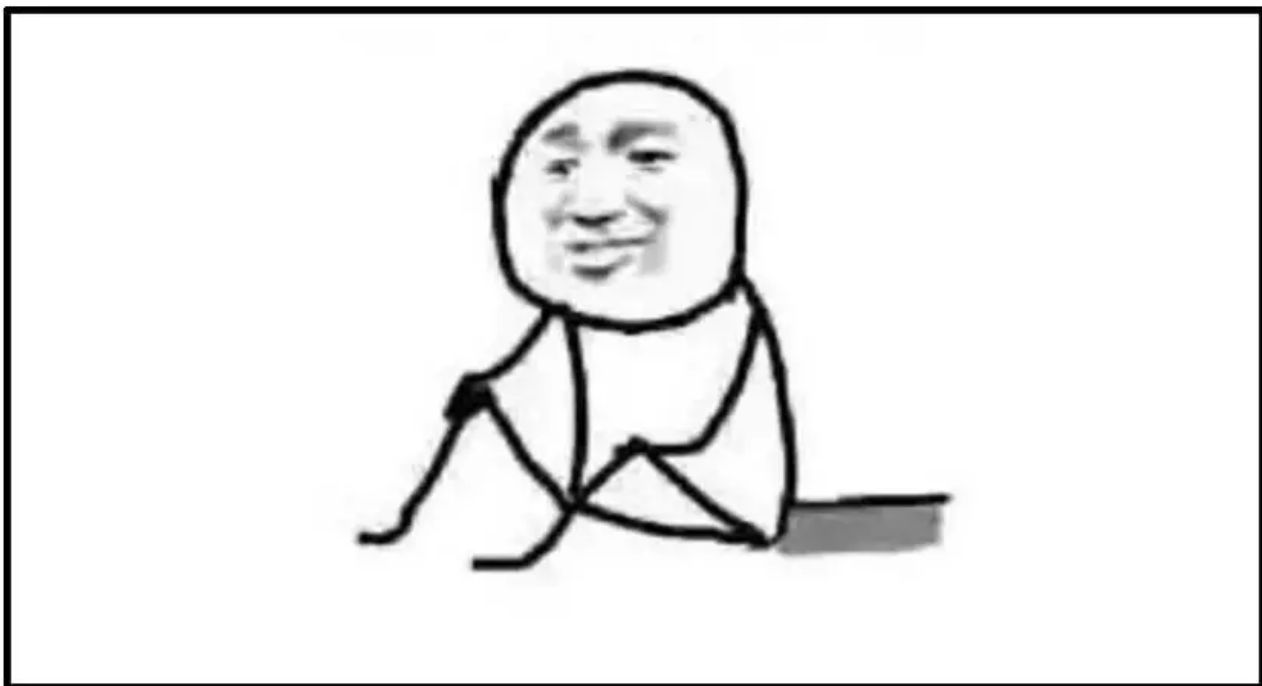


敏捷转型16个月，我总结这期间的得与失

原创 2017-06-29 小蚂蚁 人人都是产品经理



作者：小蚂蚁

全文共 6127 字，阅读需要 12 分钟

—— BEGIN ——

到今天，我们已经在敏捷转型的路上已经磕磕碰碰的走了16个月。从启程时的兴奋到现在的淡然，回想起来还是“图样图森破”，原以为把敏捷那一套运作模式拿过来，随着时间的推移，一切就会好转起来。

然而，并没有那么简单——受限于组织架构，仅仅是把敏捷那一套运作模式照搬过来我们就做不到，更别说照搬过来后能不能适应我们的产品研发了。

虽然，目前敏捷转型和预期比起来进展不够理想，但是在期间我们还是运作起来部分优秀的实践，提高了产品研发效率。

本文将总结在过去的一段时间里，我们在转型过程中踩过的坑，以作前车之鉴。也聊聊在转型过程中，哪些优秀的实践可以尝试，走上渐进变革的道路。

一、初识敏捷：我们的敏捷做对了吗？

在接触敏捷之前，大家对敏捷都是一知半解，更多的停留在字面意思的理解上：“敏捷就是快”。一旦有人觉得不快时，就会发出质疑：我们的敏捷做对了吗？

另外一方面，由于刚接触敏捷，我们开始变得理论化，严格遵循敏捷定义的方式方法，条条框框；带着对敏捷的敬畏，天真的认为敏捷规定的都是好的，敏捷之外的方式方法都是不完美的，有缺陷的。

这就是我们在刚接触敏捷时，最大的两个误区。

1. 目光不要只关注敏捷的快

快，仅仅是敏捷带来的一个结果而已。单纯的只关注这个结果，并不能对我们的转型带来什么帮助。

关于快的定义，我们习惯性的参考别人做产品是什么样的节奏，找到业界的优秀企业作对比，认为做到那样才是快。

然而，我们却很少去关注对应产品的品类和复杂度，忽略他们的组织文化、组织架构、人员素质、开发过程中的取舍等各种细节。

看到别人家的孩子全是优点，自家的孩子一堆问题，却不去了解别人是如何解决那些问题的，付出了什么代价，只想取别人的优点，忽略别人的缺点。

欲思其利，必虑其害，欲思其成，必虑其败。—《便宜十六策·思虑》

我们在做敏捷转型时，引入了外部咨询培训，培训过程中老师举了一个例子：

“我们以前做产品时，一天可以发二十几个版本。”

这样的例子对于我们来说简直是太刺激了——我们的产品要一个多月才能发出去一个版本。

虽然我们知道老师举例的产品是web端产品，与我们的产品完全不一样，我们还是不由自主的认为：我们是不是可以做到一周发一个版本，甚至更快？毕竟别人一天20多个版本啊。

对，我们的目光完全被“快”吸引了。于是什么都围绕怎么能更快去开展改革，想方设法的充分利用资源，各种并行，反而导致混乱。

2. 不忘初心，实事求是

我们是一家销售消费类硬件产品为主的公司，在引入敏捷之前，我们采用的瀑布开发模式，在产品复杂度低，研发人员较少的情况下，运作还比较好。

但是，随着产品复杂度的增加，研发人员增多，人员依赖度的增加，这样的研发模式变得十分笨重；连一个基本的信息传递都出了问题，集成一个版本变得十分困难，发布日期也总是一拖再拖。

从结果上看，就是我们不能按时发布，发布的周期比较长，产品研发变得“不敏捷”。

然而要实际落地解决问题，我们需要在这几方面改善：

- 建立新的沟通机制，保障跨领域协作的有效沟通。
- 需求要进行有效的优先级排序，并控制数量，并承诺对应优先级需求的交付率。
- 避免建立过大的团队组织，如果存在，需要根据业务的独立性，拆解为适当规模的小团队。
- 团队之间人员要解耦，尽量避免一个人在多个项目团队。
- 信息透明，尽量让团队成员能基于已有的信息自己做出决策，而不是询问上级。
- 建立跨领域团队，并对团队进行敏捷开发知识的宣导和指导。
- 引入持续集成。
- 限制在制品，避免一个人同时开展多项工作，原则上不能超过2个未完成任务。

等等.....

我们不再纠结于是否一定执行了敏捷里面规定的活动，又或者说我们现在的的项目过程，也谈不上是完整意义的敏捷开发，我们更注重什么样的招式能有效解决我们当下的问题。

3. 我们的绊脚石

开篇讲到了组织架构导致我们并不能完整的引入敏捷开发的各项机制——做过敏捷转型的人应该都遇到了这样的问题。

以下是我们在敏捷转型中遇到的一些实际问题，当你遇到这些问题时，不要惊讶，对于一个传统的成熟组织来说很难改变，尤其在没有上层领导积极推动的情况下：

3.1 没有产品经理，敏捷团队只关注开发与发布，不管开始，不管结果。

在我们公司的组织架构采用了弱矩阵形式，敏捷团队更多的起到协作开发前端需求的作用，他们不需要对产品直接负责，也不会关注产品在市场上的反馈。

3.2 测试与开发职责界限清晰。

在理想的敏捷中，在开发过程中就开始做测试，而我们还是需要开发完成后再进入测试，走上了“小瀑布模式”之路。

3.3 无法进行短周期迭代，放弃迭代。

由于走上了“小瀑布模式”之路，根本没办法做到较短周期的迭代，对于一个月一个版本来说，超过2周的迭代周期已经失去了实际的意义。

3.4 做好的功能都会发出去，基本没有功能灰度验证，功能越堆越多，维护越来越难。

需求人员都会认为自己想法是完美的，能有效提高产品的价值，他们更关注自己提的需求，而不是这个产品。

3.5 没有项目经理，只有项目负责人，项目负责人不专注，不专业。

由于公司项目是弱矩阵形式，项目更多的是协作作用，项目负责人更多的是起到拉通团队协作的作用，由于项目负责人往往身兼多职，并不能有效的持续投入精力优化项目过程。

等等.....

二、成功的每一小步

固然，我们在转型中问题众多，还有很大的优化空间，但是和以前比起来我们还是取得了不小的进步，以下招式在实际项目开展过程中具有明显的实际意义，并不受传统组织架构排挤：

1. 跨职能团队

对于传统企业来说，一般根据职能属性进行部门划分，各部门互相协作完成项目。

然而，跨部门的协作成本明显过高，当产品需要频繁的跨部门协作时，这样的模式明显无法胜任。

而在这时，公司必然也会暴露出由于这样的缺陷，导致的开发效率低下的问题。

此时，以事业部或核心部门主导，建立虚拟的跨职能团队，也就顺理成章，一气呵成。

跨职能团队能有效解决产品在开发时，频繁协作沟通的问题。

在这样的团队里，沟通由实际的开发、测试、需求人员直接当面沟通，频率更高，信息失真也比较少，也避免了部门之间扯皮的问题。

2. 每日晨会

晨会，是跨职能团队进行沟通的标准活动，每天在固定的地点、固定时间花10分钟左右举行，是一种很有效的团队同步机制。

然而晨会看上去简单，但是要开好，却没有那么容易。敏捷里面的推荐晨会内容主要包括以下三方面。

- 昨天做了什么？
- 今天要做什么？
- 有什么问题？

对于敏捷小白的我们，刚开始照本宣科的组织晨会，但是采用标准的模式，我们始终都开不好这样的晨会。

下面会详细的说说我们都遇到了哪些问题。

- 晨会沟通时，大部分信息对改善项目进展的实际意思很小，口水话太多。
- 当一位同事在描述自己昨天做了什么，今天要做什么时，大部分人并不在意。
- 大部分工程师晨会都是自顾自说，不关心自己表达的内容是否有效传递。
- 推广晨会轮流组织时，发现大部分人并没有这个能力有效的组织晨会，并不太好实践。
- 晨会讨论的内容很离散，看不到整个项目的进展。

怎么开好晨会？

正如敏捷宣言里面提到的“个体交互胜过流程和工具”一样。晨会，需要将当前项目的进展透明到整个团队，并促使团队成员基于各自目前的情况，通过沟通及时主动的解决问题。

那如何才能做到快速有效的沟通呢，我认为需要做到以下几点：

- 团队成员都清晰的知道项目的目标和当前的状态。
- 项目中的各项工作内容都有清晰的优先级，以及明确的责任人和当前的进展信息。
- 晨会有明确的规则，每个人都知道在何时做什么事。
- 参与晨会的每一个人都能随着轻松的了解到他相关人员的任务进展状态。

要做到这些，当然就少不了一面合理的看板墙了。它是整个晨会的核心载体，晨会时，所有的项目成员都基于看板墙的信息进行讨论并更新看板墙的信息。

2.3 看板墙

看板墙，看似只是简单的将项目中要展示的信息帖到一个版本上面去，但是如果贴得不合理，整个晨会就会一团乱麻，思路离散，最终导致晨会的失败。

接下来，我将谈谈我们看板的演进过程：

刚开始，我们采用上图那样的看板墙，纸片上仅仅写了用户故事，每天晨会的时候，大家围成一圈，按顺序每个人讲自己昨天做了什么，今天做什么，有什么问题。当发现某项用户故事改变了状态时，就挪动到对应的列里面去。

渐渐的问题就暴露出来，我们发现看板墙上的卡片会有很多，而且会越来越多——因为我们总是想做更多的用户故事，一旦某项用户故事在开发过程中受阻，我们就会考虑开展一项新的用户故事，我们不能接受一个人“闲下来”。

另外，由于每个人讲的内容，并没有和卡片直接关联起来，导致我们并不能很好的把讲的信息和看板墙结合起来，以至于看板墙成了一个背景，没有起到太多的实际意义。

我们开始质疑，轮流讲问题是不是不合理？后来转变为由一个人主持，根据看板墙上的信息逐个询问探讨。

为了实现轮流主持，我们定期换不同的人进行询问探讨，但是有时候有的人却完全不知道如何有效的主持，导致晨会下来的有效沟通很少。

由于看板卡片上记录的信息很有限，卡片数量又比较多，以至于项目负责人都难以清晰的了解当前项目的进展，更别说其他团队成员了。

于是我们把看板改成了下图的样子：

这样子，看上去比以前的看板清晰了一点，能比较清晰的知道目前视觉、软件、测试的任务情况；可惜还是存在不断的向后面推用户故事的问题，导致看板的在制品过多。

另外，还是存在部分用户故事较大，在某个位置停留过长时间的问题，并且我们并不能清晰的了解到该用户故事目前的进展情况。

这里附带提一下：总会有人提要把需求的粒度拆分得更小，把一个用户故事拆分到1到3天的工作量，可惜我们一直没有实现。

或者我们勉强把一个大的用户故事拆分成了几个小的用户故事，但是在我们这种敏捷并没有完全转变过来的团队，我们还是喜欢一个需求完整的交付——只有这样我们的黑盒测试才能有效的测，不然会让测试重复测，浪费测试资源，前端需求对于一个没有做到发布状态的需求，也觉得不好体验。

为了有效的跟进较大的用户故事的进展，我们开始进行任务拆分。

我们规定：一项任务的粒度最大不能超过三天的工作量。我们趋向于团队成员尽量把任务都拆解到小于一天工作量的粒度，因为这样有利于我们每日晨会进行跟进。

于是看板上的卡片就分为了两类，出现了下面的看板。

我们将看板上的卡片分为用户故事和与用户故事对应的任务，当所有与用户故事相关的任务都完成时，将用户故事挪到待发布中。

用户故事与任务之间使用编号进行关联，这使得我们的晨会过程讨论的内容更加具体，而不是简单的汇报工作，项目进展情况也比以前更加清晰。

然而，问题依旧存在：

当任务卡片较多时，任务与用户故事的对应关系找起来相对麻烦，不能一目了然的了解到用户故事的进展情况。

同样的，这样的看板依旧会存在前端需求大量的向后端涌入，导致后端负荷过重，效率降低的情况。

得益于精益的限制在制品概念，结合我们之前的各种经验，我们最终使用了下面这种看板墙。

在这个看板墙里，只有用户故事卡，用户故事卡上拆分了用户故事的不同任务，我们不再追求多个领域能针对一个用户故事并行开展任务，而是一个领域做完后，再流入到下个领域。

表面上并行开展工作看起来效率更高，但是那样各领域信息并不一致，返工较多，反而导致效率低。由于任务和故事在同一张卡片上，我们可以从看板上一目了然的看到用户故事当前的状态。

另外，我们将卡片分为三种颜色，对应于三个优先级，这样大家在领取任务时，能更加自主。

还有一点十分重要：每个领域我们根据他们的人数设置在制品限制，目前规定一个人同时开展的工作不能超过两项。我们由推的模式转变为拉的模式。

这样能清晰的暴露领域边界的资源匹配情况，同时也能保护相关领域不至于任务过多引起混乱，进而导致效率降低——我们由要做更多的需求转变为“停止启动，聚焦完成”。

一个人的同时工作项不超过两项并不是一个标准答案，我们内部也未完全按照该标准执行。对于基本不会出现阻塞的领域，也许一项也是一个合适的选择。

2.4. 版本规划

由于我们的产品包括iOS应用、Android应用、手表端、服务器、H5这样5个技术领域。

在初期，我们遭遇了各领域互相依赖，因某个领域有bug未能及时修复，导致整体发布延期的问题，而且还经常发生。

因此，我们采用了版本火车的概念：以发布为主，功能为辅——也就是在固定的时间节点必须发布，如果对应功能没有达到发布要求，就推迟到下一个发布版本。

这样做却带来了几个比较揪心的问题：我们版本没有明确的主题，仅仅是一堆已完成的需求而已，导致每个版本的亮点不够。

另外，由于没有规划版本，因此无法给到前端明确的心理预期，他们不知道他们的需求在什么时候能发布，他们也不知道我们的下一个版本将会携带哪些需求。

而对于开发而言，也缺乏明确的目标感，给人一种来了需求我就做，做多少算多少的感觉。

于是，我们最终还是放弃了版本火车，改为版本规划的方式进行整个产品的研发。

版本规划，最头痛的问题就是选哪些需求进入该版本进行开发的问题。要知道，需求永远都是源源不断的来，需求的总量总会远远大于你当前版本能开发完成的总量。

因此就涉及到一个需求收集和筛选的问题：

如果每次版本都要筛选那么多需求，需要耗费很大一拨人大量的时间来进行需求优先级讨论，以及该需求是否在该版本做的问题。

另外，在这么短的时间内也得不到太有效的讨论结果。

在《看板实战》中，有一个“优先级过滤器”的实践方案，如下：

在这个看板上，明确了一个团队的产能，以及接下来要做的需求。

当一个需求被挪走后，就会触发后续需求的进入和讨论，这样每次只需要讨论少量的需求，更加聚焦。

这固然是一个好的实践，可是我们现阶段的团队并不能适应它，因为我们并没有真真的做到精益里面提到的拉动式开发。

于是我们在进行版本规划时，采用了如下方案：

- 在下一个版本开展前两周开始收集需求。
- 一个版本一定要有一个明确的主题。
- 版本有明确的时间节点。
- 需求要明确必须、应该、可以三个优先级，比例要适中。
- 必须的需求未按期完成时，版本发布时间后延。
- 应该的需求要保障至少80%的交付率。
- 可以的需求要保障至少50%的交付率。
- 需求收集完后分为多次进行版本规划讨论会议。

分多次进行版本规划讨论会议很关键，由于需求收集的量比较大，另外参会的人比较多，大家很难一次性达成一致。

当一次会议结束后，需要给到整个团队一些线下时间做更多的思考以及了解更多的附加信息进行进一步评估。

我们一般会开2到3次这样的会议，每次会议时间为1到2个小时，这是一个逐步收敛和达成一致的过程。

三、未来

我们的项目过程，还有很多值得优化的地方。

到目前为止，我们的项目开展过程只能说做到了有序。而在接下来的日子里，我们还需要加强项目的度量，做到“不凭感觉规划版本内容的多少，不凭感觉说项目做得好还是差”。

通过度量数据，更客观和明确的暴露项目中的问题。

四、推荐书单

对于我自身，经历过公司组织的一次敏捷咨询后，就踏上了部门敏捷转型的道路，期间踩坑无数，幸得以下书籍把我从一个一个坑里面拉了出来。

推荐的书单中，部分书籍还未读完，大部分书籍阅读完一两遍后还不得其道，还需要反复阅读和实战，这里就不好意思写自己的感悟和对这些书的评价了，以免误导大家。

但是，这些书都是在我转型的道理上，给予我不少启发和帮助的书籍，我相信读过这些书的人都能有所收获。

- 《敏捷软件开发实战-估算与计划》
- 《敏捷软件测试》
- 《精益创业》
- 《精益创业实战》
- 《看板方法》
- 《看板实战》
- 《网易一千零一夜》
- 《人月神话》
- 《四步创业法》
- 《目标》
- 《卓有成效的管理者》
- 《最后期限》
- 《思考，快与慢》

—— END ——

本文由 @小蚂蚁 原创发布于人人都是产品经理。未经许可，禁止转载

每日一问

■ 思考题：如何理解“好的产品经理，也都是不遵从规则者”？

俞军在近期的一篇文章提到，滴滴的入职性格测试中，产品经理的「遵从规则」的得分特别低，好的产品经理，也都是不遵从规则者。

你是如何理解这个观点的呢？

分享你的观点到留言区，一起碰撞观点！



本期奖品



这次奉上的书籍是《滴滴：分享经济改变中国》，是分享经济的成功践行者滴滴出行CEO程维、滴滴出行总裁柳青首部著作，本书解析“互联网 分享经济”平台构建的九大关键节点，推动传统行业搭上分享经济“顺风车”！

好书不容错过哟~快来拿吧~

■ 符合以下条件的小伙伴们即可获得奖品：

1. 高质量的评论（走心的那种）
2. 评论被赞数名列第一
3. 评论被赞数不少于40个
4. 评论被赞数统计在 周五晚6点前 截止

周六开奖！ 请留意 **周六** 的 头条文章推送哦，不见不散~

加入50万产品经理组织一起学习



扫一扫 · 下载APP

人人都是产品经理

中国最大最活跃的产品经理社区

学产品 | 学运营 | 听讲座

点击 “阅读原文” 下载APP

阅读原文