

08 | 精益看板（上）：精益驱动的敏捷开发方法

2019-10-29 石雪峰

DevOps实战笔记

[进入课程 >](#)



讲述：石雪峰

时长 15:59 大小 14.65M



你好，我是石雪峰。

提到敏捷开发方法，你可能会情不自禁地联想到双周迭代、每日站会、需求拆分等。的确，作为一种快速灵活、拥抱变化的研发模式，敏捷的价值已经得到了行业的普遍认可。但是，即便敏捷宣言已经发表了将近 20 个年头，很多公司依然挣扎在敏捷转型的道路上，各种转型失败的案例比比皆是。

我曾经就见过一家公司，一度在大规模推行敏捷。但是，这家公司很多所谓的敏捷教练都是项目经理兼任的，他们的思维和做事习惯还是项目制的方式。即便每天把团队站会开得有条不紊，看板摆得到处都是，但从产品的交付结果来看，并没有什么显著的变化。没过多久，由于组织架构的调整，轰轰烈烈的敏捷转型项目就不了了之了。

这家公司虽然表面上采用了业界流行的敏捷实践，也引入了敏捷工具，但是团队并没有对敏捷的价值达成共识，团队领导兼任 Scrum Master，好好的站会变成了每日工作汇报会。甚至在敏捷项目复盘会上，领导还宣称：“敏捷就是要干掉变化，我们的目标就是保证团队按照计划进行。”这种“貌合神离”的敏捷，并不能帮助企业达到灵活响应变化、快速交付价值的预期效果。

作为一种最广泛的敏捷框架，Scrum 的很多理念和实践都深入人心，比如很多时候，迭代式开发几乎等同于跟敏捷开发。但是，Scrum 对于角色的定义并不容易理解，在推行 Scrum 的时候，如果涉及到组织变革，就会举步维艰。

实际上，企业的敏捷转型并没有一条通用的路径，所用的方法也没有一定之规。今天，我就跟你聊聊另外一种主流的敏捷开发方法——精益看板。与 Scrum 相比，看板方法的渐进式改变过程更加容易被团队接受。我之前所在的团队通过长期实践看板方法，不仅使产品交付更加顺畅，还提升了团队的整体能力。

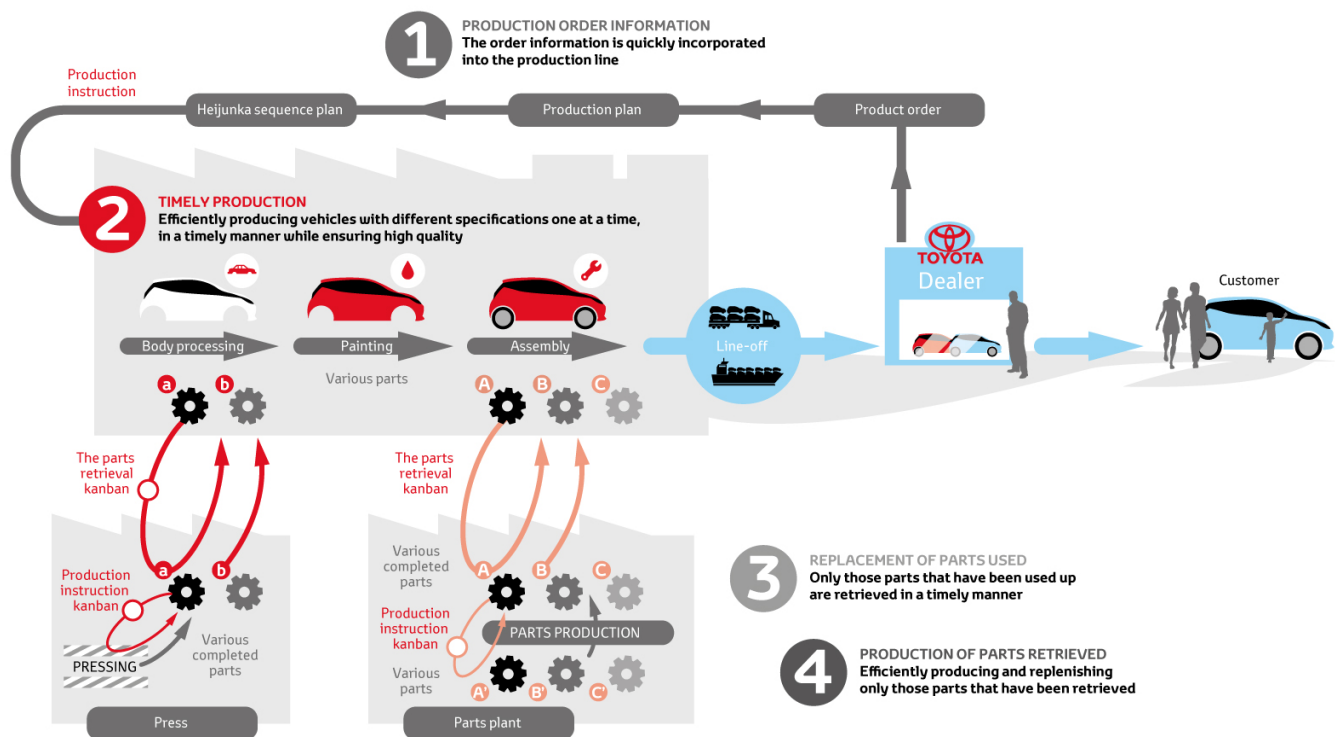
那么，这个神奇的精益看板是怎么回事呢？

如果你之前没听说过精益看板，还是很有必要简单了解下它的背景的。其实，“看板”是一个日语词汇，泛指日常生活中随处可见的广告牌。而在生产制造系统中，看板作为一种信号卡，主要用于传递信息。很多人认为看板是丰田公司首创的，其实并非如此，比如在我之前所在的尼康公司的生产制造车间里，看板同样大量存在。

当然，看板之所以能广为人知，还是离不开丰田生产系统。《改变世界的机器》一书首次提到了著名的丰田准时化生产系统，而看板正是其中的核心工具。

简单来说，看板系统是一种拉动式的生产方式。区别于以往的大规模批量生产，看板采用按需生产的方式。也就是说，下游环节会在需要的时候，通过看板通知上游环节需要生产的工件和数量，然后上游再启动生产工作。

说白了，所谓拉动式生产，就是从后端消费者的需求出发，向前推导，需要什么就生产什么，而不是生产出来一大堆没人要的东西，从而达到减低库存、加速半成品流动和灵活响应变化的目的。我你分享一张有关丰田生产方式的图片，它演示了整个丰田生产方式的运作过程，你可以参考一下。



图片来源： <https://www.toyota-europe.com/world-of-toyota/this-is-toyota/toyota-production-system>

软件开发中的看板方法，借鉴了丰田生产系统的精益思想，同样以限制在制品数量、加快价值流动为核心理念。也就是说，**如果没有在制品限制的拉动系统，只能说是一个可视化系统，而不是看板系统，这一点非常重要。**

比如，很多团队都在使用 Jira，并在 Jira 中建立了覆盖各个开发阶段的看板，围绕它进行协作，这就是一个典型的可视化板，而非看板。那么，为什么对于看板方法而言，约束在制品数量如此重要呢？

就像刚才提到的，**加快价值流动是精益看板的核心。在软件开发中，这个价值可能是一个新功能，也可能是缺陷修复，体验优化。**根据利特尔法则，我们知道：**平均吞吐率 = 在制品数量 / 平均前置时间**。其中，在制品数量就是当前团队并行处理的工作事项的数量。关于前置时间，你应该并不陌生，作为衡量 DevOps 产出效果的核心指标，它代表了从需求交付开发开始到上线发布这段时间的长度。

比如，1 个加油站只有 1 台加油设备，每辆车平均加油时长是 5 分钟，如果有 10 辆车在等待，那么前置时长就是 50 分钟。

但是，这只是在假设队列中的工作都是顺序依次执行的情况下，在实际的软件开发过程中。如果一个开发人员同时处理 10 件事情，那么在每一件事情上真正投入的时间绝不是 1/10。

还拿刚刚的例子来说，如果 1 台加油设备要给 10 辆车加油，这就意味着给每一辆车加油前后的动作都要重复一遍，比如取出加油枪、挪车等。这样一来，任务切换的成本会造成极大的资源消耗，导致最终加满一辆车的时长远远超过 5 分钟。

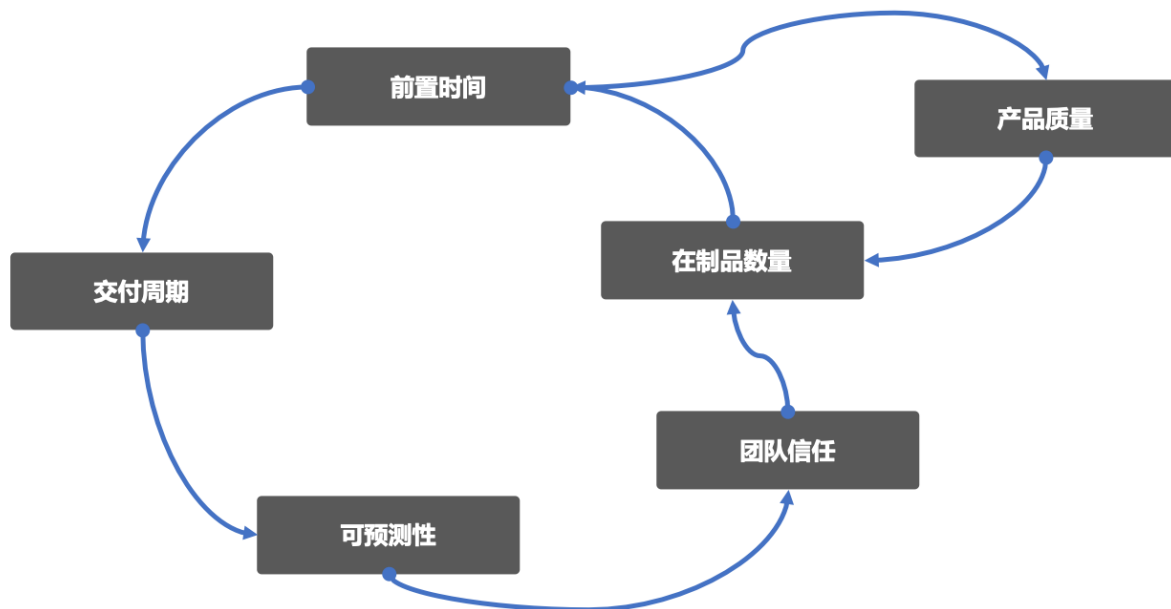
所以，**在制品数量会影响前置时间，并行的任务数量越多，前置时间就会越长**，也就是交付周期变长，这显然不是理想的状态。

不仅如此，**前置时间还会影响交付质量，前置时间增长，则质量下降**。这并不难理解。比如，随着工作数量的增多，复杂性也在增加，多任务切换总会导致失误。另外，人的记性没那么可靠。对于一个需求，刚开始跟产品沟通的时候就是最清晰的，但是过了一段时间就有点想不起来是怎么回事了。这个时候，如果按照自己的想法来做，很有可能因为对需求的理解不到位，最终带来大量的返工。

再进一步展开来看的话，软件开发工作总是伴随着各种变化和意外。如果交付周期比需求变化周期更长，那就意味着紧急任务增多。比如老板发现一个线上缺陷，必须高优先级修复，类似的紧急任务增多，就会导致在制品数量进一步增多。这样一来，团队就陷入了一个向下螺旋，这对团队的士气和交付预期都会造成非常不好的影响，以至于有些团队 90% 的精力都用来修复问题了，根本没时间交付需求和创新。

更加严重的问题是，这个时候，业务部门对 IT 部门的信任度就会直线下降。业务部门往往会想：“既然无法预测需求的交付实践，那好吧，我只能一次性压给你一大堆需求。”这样一来，就进一步导致了在制品数量的上升。

可见，一个小小的在制品数量，牵动的是整个研发团队的信心。我把刚刚提到的连环关系整理了一下，如下图所示：



当然，针对刚才加油站的问题，你可能会说：“多加几台加油设备，不就完了吗？何必依赖同一台机器呢？”的确，当并行任务过多的时候，适当增加人员有助于缓解这个问题，但是前置时间的缩短是有上限的。这就好比 10 个人干一个月的事情，给你 100 个人 3 天做完，这就是软件工程管理的经典图书《人月神话》所讨论的故事了，我就不赘述了。这里你只需要知道，随着人数的增多，人与人之间的沟通成本会呈指数级上升。而且，从短期来看，由于内部培训、适应环境等因素，新人的加入甚至会拖慢原有的交付速度。

了解了精益看板核心理念，以及约束在制品数量的重要性，也就掌握了看板实践的正确方向。那么，在团队中要如何开始一步步地实施精益看板方法呢？在实施的过程中，又有哪些常见的坑，以及应对措施呢？这正是我要重点跟你分享的问题。我把精益看板的实践方法分为了五个步骤。

第一步：可视化流程；

第二步：定义清晰的规则；

第三步：限制在制品数量；

第四步：管理工作流程；

第五步：建立反馈和持续改进。

今天，我先给你介绍精益看板实践方法的第一步：可视化流程。在下一讲中，我会继续跟你聊聊剩余的四步实践。

第一步：可视化流程

在看板方法中，提高价值的流动效率，快速交付用户价值是核心原则，所以第 1 步就是要梳理价值交付流程，通过对现有流程的建模，让流程变得可视化。关于价值流建模的话题，在专栏 [第 5 讲](#) 中我已经介绍过了，如果你不记得了，别忘记回去复习一下。

其实，在组织内部，无论采用什么研发模式，组织结构是怎样的，价值交付的流程一直都是存在的。所以，在最开始，我们只需要忠实客观地把这个现有流程呈现出来就可以了，而无需对现有流程进行优化和调整。也正因为如此，看板方法的引入初期给组织带来的冲击相对较小，不会因为剧烈变革引起组织的强烈不适甚至是反弹。所以，**看板方法是一种相对温和的渐进式改进方法。**

接下来，就可以根据价值流定义看板了。看板的设计没有一个标准样式，因为每个组织的价值流都不相同。对于刚刚上手看板方法的团队来说，看板的主要构成元素可以简单概括成“**一列一行**”。

1. 一列。

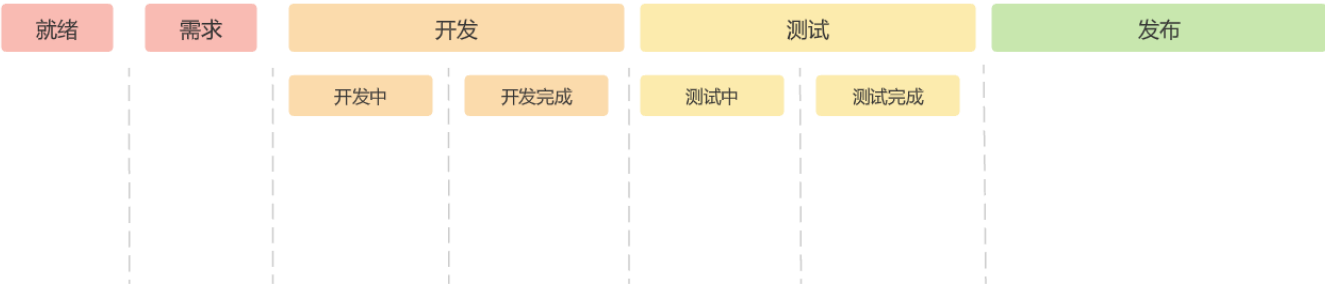
这是指看板的竖向队列，是按照价值流转的各个主要阶段进行划分的，比如常见的需求、开发、测试、发布等。对识别出来的每一列进一步可以划分成“进行中”和“已完成”两种状态，这也是精益看板拉动式生产的一个显著特征。对于列的划分粒度可以很细，比如开发阶段可以进一步细分成设计、编码、自测、评审、提测等环节，或者就作为一个单独的开发环节存在。划分的标准主要有两点：

是否构成一个独立的环节。比如对于前后端分离的开发来说，前端开发和后端开发就是两个独立的环节，一般由不同的角色负责，这种就比较适合独立阶段。

是否存在状态的流转和移交。看板是驱动上下游协同的信号卡，所以，我们需要重点关注存在上下游交付和评审的环节，这也是提示交付吞吐率和前置时长的关键节点。

除此之外，看板的设计需要定义明确的起点和终点。对于精益看板来说，覆盖端到端的完整价值交付环节是比较理想的状态。但实际上，在刚开始推行看板方法的时候，由于组织架构、团队分工等多种因素，只能在力所能及的局部环节建立看板，比如开发测试环节，这并不是什么大问题，可以在局部优化产出效果之后，再尝试向前或向后延伸。

另外，即便看板可以覆盖端到端的完整流程，各个主要阶段的关注点各不相同，所以，也会采用看板分类分级的方式。对于开发看板来说，起点一般是需求准备就绪，也就是说，需求经过分析评审设计并同研发团队沟通一致准备进入开发的状态，终点可以是提测或者发布状态。流程的起点和终点同样要体现在看板设计中，以表示在局部环节的完整工作流程。

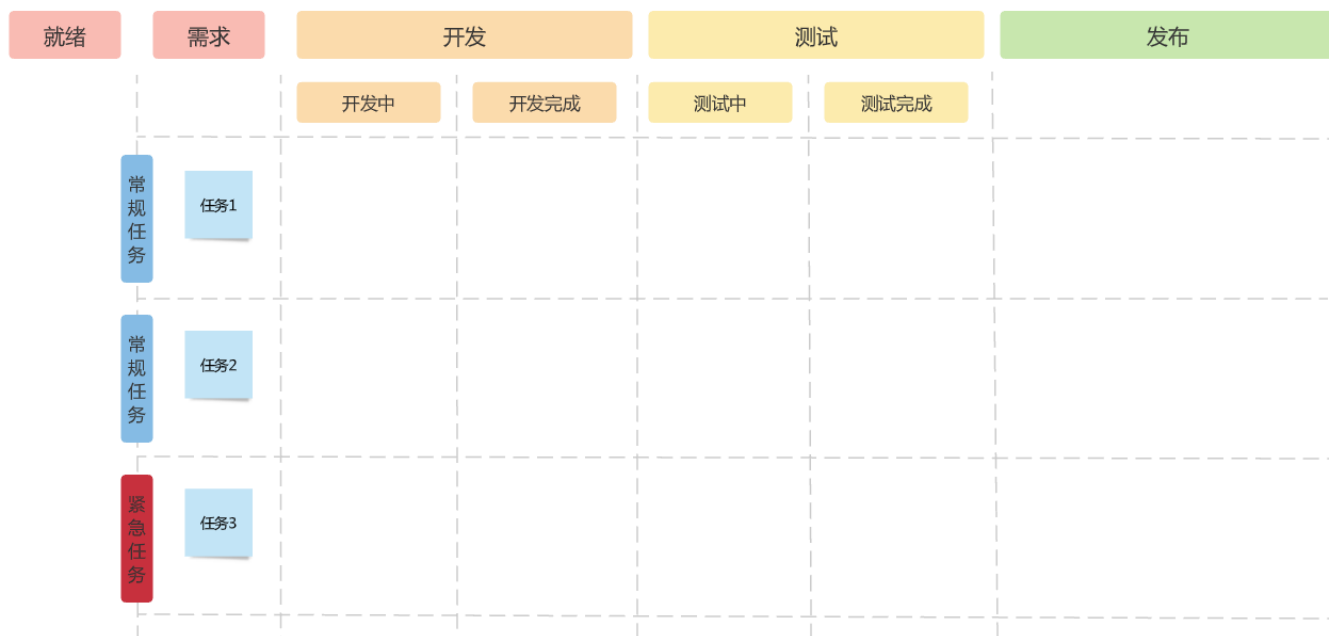


2. 一行。

这是指看板横向的泳道。泳道用于需求与需求之间划清界限，尤其在使用物理看板的时候，经常会因为便利贴贴的位置随意等原因导致混乱，而定义泳道就可以很好地解决这个问题。比如，高速公路上都画有不同的行车道，这样车辆就可以在各自的车道内行驶。

当然，泳道的意义不只如此。泳道还可以按照不同维度划分。比如，有的看板设计中会加入紧急通道，用于满足紧急需求的插入。另外，非业务类的技术改进需求，也可以在独立泳道中进行。对于前后端分离的项目来说，一个需求会拆分成前端任务和后端任务，只有当前后端任务都完成之后才能进行验收。这时，就可以把前后端任务放在同一个泳道中，从而体现需求和任务的关联关系，以及任务与任务之间的依赖关系，快速识别当前阻塞交付的瓶颈点。

当然，看板的设计没有一定之规。在我们团队的看板中，往往还有挂起类需求区域、缺陷区域，以及技术攻关类区域等，用于管理特定的问题类型。比如对于长期挂起的需求，在一定时间之后就可以从看板中移除，毕竟，如果是几个月都没有进入任务队列的需求，可能就不是真正的需求，这些可以根据团队的实际情况灵活安排。如果你在使用 Jira 这样的工具，虽然没有区域的概念，但是可以通过泳道来实现，比如按照史诗任务维度区分泳道，然后新建对应区域的史诗任务就可以啦。



总结

今天，我给你介绍了敏捷常用的两种框架 Scrum 和看板。看板来源于丰田生产系统，以拉动式生产为最典型的特征。关注价值流动，加速价值流动是精益看板的核心，限制在制品数量就是核心实践，因为，在制品数量会直接影响团队的交付周期和产品质量，甚至还会影响团队之间的信任，导致团队进入向下螺旋。

在团队中实践精益看板，可以分为五个步骤，分别是：可视化流程、定义清晰的规则、限制在制品数量、管理工作流程和建立反馈并持续改进。今天我给你介绍了第一个步骤，也就是可视化流程，通过价值流分析将团队的交付路径可视化，建立起看板的主要结构，那么接下来就是开始应用看板了。下一讲，我会跟你聊聊其余的四个步骤，敬请期待。

思考题

最后，给你留一个思考题：你所在的公司是否也在实践敏捷呢？在敏捷转型的过程中，你遇到的最大问题、踩过的最大的坑是什么呢？

欢迎在留言区写下你的思考和答案，我们一起讨论，共同学习进步。如果你觉得这篇文章对你有所帮助，欢迎你把文章分享给你的朋友。

DevOps 实战笔记

精要 30 计，让 DevOps 快速落地

石雪峰

京东商城工程效率专家



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 07 | 业务敏捷：帮助DevOps快速落地的源动力

下一篇 09 | 精益看板（下）：精益驱动的敏捷开发方法

精选留言 (8)

写留言



maomaostyle

2019-10-29

敏捷推行的最大阻力就是改变管理层的价值观和对问题的认知

作者回复: 呵呵，我之前去台湾参加DevOpsDays活动，就给我老板带了一本书做礼物，书的名字是《原来你才是团队绊脚石》，你也可以试试看效果如何哈。



2



sugar

2019-10-29

总有插入队列的“紧急需求”；前期没有明确，开发过程中持续拉扯。看到老师可视化流程这个，想起三年前有个领导针对开发团队提出过看板方法，不过富哦随着那个领导短暂

的停留，也没继续保持下去。

作者回复: 我在团队中也试行过很多实践，看板是自始至终坚持下来的，也的确经历了相当长的过程来养成习惯，但是一旦习惯了这种做法，整个状态的可视化会让你眼前一亮的。另外，我们经常会各种紧急需求打断，说实话一般也很难Say no，但是如果长此以往问题也不会自己好转，所以适当的积累一些数据也是很有必要的。



Robert小七

2019-10-29

老师文章中提了站立会，想请问老师，正确的站立会应该如何做？

作者回复: 你好，正确的站会我觉得应该有以下几个特征：自组织，暴露问题，反馈行动。自组织是说即便没有团队leader组织站会，大家也能按照既定的节奏来自发的组织站会，所以被动到主动的转变。暴露问题，就是团队敢于在大家面前暴露问题和风险，而不是一味的说明成绩，也就是心里紧张到心里安全的转变。反馈行动，就是站会上发现的问题能够直接影响接下来的行动，快速处理问题，也就是边说了白说到立竿见影的转变。



陈斯佳

2019-11-01

这篇文章纠正了我一个观念，就是看板不等于可视化板。看板的核心是加快价值流动，以拉动式生产为典型特征，约束制品数量为实践，灵活响应业务变化，快速交付价值。如果用这个标准来衡量敏捷程度，就不容易跑偏了

展开 ∨



linda.zx

2019-10-30

关于老师提供的看板例子，“就绪”和“需求”不太理解，感觉只需要“就绪”就够了，为什么会考虑加“需求”这一列呢？

我公司一直在高呼搞敏捷，但是偶觉得最大的坑就是留于形式的敏捷，披着敏捷的皮干着瀑布式项目。例如：需求虽然做着细化，但是还是一个人负责一个模块；一个需求开发...

展开 ∨





scorpiozj
2019-10-30

实践中 站会往往开成三四十分钟的汇报会！

展开 ▾

作者回复: 每日汇报会很常见，一开始大家也不习惯自组织的形式，我的建议是组织者后退一步，而不是站在中心的位置，约定好每次从不同人开始，尽量控制时间，当然如果有物理看板建议还是围绕在看板旁边，说话的人向前一步，这时候他就是中心哈。



iiiqueena
2019-10-29

关于最后一张看板图片，有些疑惑：

1、通常需求“就绪”的定义是怎么样的？

我所在的团队对“就绪”的定义是大家对需求达成共识，产品经理的PRD、设计师的交互设计稿准备完成。接下来的工作是开发进行编码/构建，与此同时测试同事进行测试用例/测试脚本编写。...

展开 ▾



leslie
2019-10-29

其实老师的话题中不觉得在抛出另外一个问题么：敏捷是否真的敏捷？敏捷只是且仅仅是安排的紧凑且快捷而已。软件的目的是产生效率或者说简化人工操作/沟通的量：本质上还是节约人力成本。

就像网银和手机支付其实节约的是许多超市大量的收银员雇佣的成本：为了敏捷而敏捷其实就脱离了本质，其实国内不少企业就用了丰田生产系统的做法融入其中。...

展开 ▾

作者回复: 将敏捷注入运维工作正是DevOps之所有诞生的源动力啊，我在公司里面也经常有种感觉，我们在关心流程，关心工具，关心业务，唯独没人关心人的问题，除了要考查人的效率高低之外。敏捷说的再神，说到底还是需要重视人的价值，就像用户故事，如果不让整个团队带入场景之中共创好的解决方案，而只是单纯的宣讲需求分配任务，那么谁还对敏捷感兴趣呢。

