

## 第十章 Make It So

生产力高的团队根据他们的实践结果来完成回顾活动。

企业号飞船上的 Jean-Luc Picard 船长总是说 “Make it so”，如果我们每次面对变化时，也能像他这样，那该多好啊！但是仅说 “Make it so” 是不够的。具体的行动计划才是产生结果的必要条件。把要尝试的实践安排到迭代的工作计划中，要保证团队成员都注意到这些安排。即使这样，有时还是不够的。

如果你试着改变自己的某个个人习惯（比如咬指甲之类），你会知道那基本上是不可能的，除非你能找到其他的事情来替换旧习惯。添加新行为比消除旧行为要简单。对人来说是这样，对团队和组织也是如此。

Lynn 的团队在进行回顾时，商定要停止不做任何计划就直接着手编码这样的做法。可是，在随后的下一个迭代的规划会议上，两个团队成员在一开始就打开笔记本，将他们在周末完成的代码分享给团队成员。他们俩相信：这样做能够给团队一个全新的开始。

Lynn 提醒大家注意他们此前达成的协议，并向大家分享了他在一个敏捷讨论组中看到的有关迭代计划的一些好想法。团队也同意要坚持执行此前做出的决议，并开始尝试 Lynn 提出的关于计划的建议。当团队开始彻底讨论他们要做的工作之后，他们发现：那些花费了整个周末时间所完成的代码，对当前迭代的团队目标没有任何贡献，纯粹是做了无用功。

没有替代物（比如本例中关于迭代规划的想法）的话，团队会很容易陷入到他们的旧有行为当中。

人们在采用任何新行为时，一开始总是感觉很差；随着实践的深入，会逐渐觉得轻松起来。不管是学习一种新的网球发球方式，还是使用全新的开发语言进行编码，都是这样。要允许人们在试图学习新技能时犯错误，并且要在这时为他们提供支持。

### 10.1 提供支持

引领变化的工作并不会随着回顾活动结束而完成。即使是最小的变化也需要细心关注并给予支持。可以采取多种不同的形式表示支持：强化作用、移情作用、学习机会、实践机会，以及提醒方式等。某些类型的支持来自团队，比如移情作用和提醒等。其他一些支持需要资源和预算。对于需要占用资金支出的支持方式，团队的领导、教练和管理层都有责任来协助团队获取。

**强化作用：**改变是困难的。要向大家通告项目进程，以表示你对团队（和你自己）的支持。要对实际运作效果好的行为或状况给予鼓励：“我们新使用的单元测试可以让系统的构建版本保持一种干净的状态，干的不错啊！再接再厉！”当你向团队进行了鼓励之后，你接受了挑战并且激发了大家的士气。

向大家发布当前运作效果好的行为或状况，能够使得团队认识到他们正在不断进步。要保证你给出的反馈中包括对行为及其效果的描述，“我注意到咱们昨天的每日立会开的不错。咱们决定要把精力放在四个问题上，而且咱们也做到了。这样确实帮助我看清了团队所面临的障碍”。

**移情作用:** 要知道人们产生失意和挫折感是很正常的。在下面的例子中，作为团队领导的 Fred，当有团队成员来与他讨论某个变化的时候，他却错误的处理了这种状况。由于团队决定转移到一个开放的空间中工作，Katie 不得不放弃自己的私人隔间，她向 Fred 抱怨她并不喜欢这种感觉。Fred 听完之后，回应道：“我已经想过这个问题了，你不应该这样想。”这不是移情作用。要从别人的视角去看问题，并体会他们的感觉（并不一定要同意解决相关的状况）。简单的说“我听到了你说的话”就足够了。

**学习机会:** 要展现出来对学习和钻研活动的支持。根据团队行动计划，他们可能要学习新的技能，以完成他们在计划中所要做的一些实验性活动。组织一些午餐会和讨论活动，让团队成员可以互相学习。为团队成员提供 web 资源和文章列表来激发他们的新想法。在团队内外寻找非正式的导师对大家进行指导。鼓励大家通过结对编程来学习新的开发语言和技术。几乎不用花一分钱，你就可以做到上面这些。

要愿意为支持改变花费金钱。只从网站上或是文章中，不可能学习到所有的技能。通过付费培训来打造学习新技能的坚实基础。还可以建立图书馆来让你的团队能够随时访问书籍资源。

**实践机会:** 熟能生巧；人们需要实践来取得进步。让团队暂时放下手中的产品，去尝试新的东西，这是一种方式。另一种方式是利用非常小的项目，或者实践区域，甚至是一个 Hello World 程序，来制造一个比较正规的实践空间。

发起持续时间为一至两天甚至更短的项目，来让大家钻研新的解决方案或是尝试新的方法。如果团队没有空闲的时间，那就让小项目既对新技术展开研究，又能实际推进当前的项目进度，这样一举两得。小项目的时间限制是对实践效果的显式检查点，团队通过它可以评估他们的学习效果，并为实验活动做出结论。

实践区域，是一个虚拟的空间概念。团队可以在其中尝试新的技术，而不影响实际产品。实践区域可以是一个特别的测试或开发区域，并且没有被当前的产品开发所使用。

多鼓励团队尝试类似 Hello World 这样的小程序。Hello World 程序很简单，基本上就是打印或显示“Hello World”的字样。但是可以用他们来测试开发环境和配置，并且快速定位问题（或确认一些基本的概念没有出问题）。

**提醒:** 可见的大图表或是登记表，可用来帮助团队将注意力集中在变化上。例如：Terry 的团队决定他们要更加频繁的进行重构。他们做了一个很大的图表，团队成员每完成一个重构任务，就在上面贴一个绿色的大圆点。在每天工作结束之时，他们会检查这个图表并讨论结果。这个图表让团队的重构活动效果显著。

登记表可以让团队报告他们做了那些特别的改变。要保证问题和回答都简明扼要，“用一到

两个词汇来描述我们的估算做的如何?”。将回答作为评价新实践效果好坏的标杆。

## 10.2 共享责任，制造变化

在团队中，如果总是由一个人来承担具体行动的责任，会出现下面三个问题：

- 团队会寄希望于某个成员成为英雄拯救者，而且这个成员可能很享受做英雄的感觉。但此类做法实际上却伤害了团队。团队成员之间的协作关系，以及团队对代码的共享所有权都被破坏了。
- 当一个正式或非正式领导者总是在承担责任（团队之外的系统问题除外）时，该领导者实际上在纵容团队成员成为无助的受害者。互相协作，共同改进，这样可以增强团队的能力。如果把共同责任从团队身上去掉，反而会削弱了他们的能力。
- 如果团队总是把解决问题的责任分配给团队中的某几个人，这就会造成一种感觉，让大家觉得这几个人是所有问题的根源。寻找替罪羊会将团队打散。应当共享责任，并且时常转换变化的主导者。

## 10.3 支持更大的变革

对迭代的回顾通常会引发一些影响范围比较小的变化；团队在下一个迭代或者几个迭代中便可以完成这些改变。更大范围的回顾则产生更广阔的变革，需要更长的时间来实施。广阔的变革需要更多支持和关注，以帮助人们应对变化。

当人们放弃过去，准备迎接新事物时，就算这个改变是他们选择的，就算他们有了应对计划，他们将要所经历的转变过程也是可以预期的。对于小变化，人们通常不需要外部支持就可以适应。如果是大型变革，转变过程会持续更长的时间，不同的人对过程发生速度的感觉也不尽相同。理解变化的四个转变阶段，有助于帮助你更好的支持团队。

### 变化的四个转变阶段

**迷失 (Loss)：**掌握新事物总是要从放弃旧事物开始。人们会感到迷失——失去了某种能力、失去领地、失去关系、失去确定性等等。对新事物的兴奋感会让他们很快度过这个阶段，不然他们会用更多时间去调整。不管怎样，如果不能放弃过去，他们将不能也不会走上前进的路途。

**混乱 (Chaos)：**放弃过去并不是说我们就已经完全理解新事物了。在变化过程中，人们会感到迷惑，而且尽力要重新调整自己的方向。他们会探索变化如何发生，以及新的方式对他们来说意味着什么。虽然有困惑之感，混乱状态却有可能会激发全新的创意。人们也许会发明新的方式来解决，因为这时游戏规则还没有最后确定。

**转变想法 (Transforming Idea)：**最终，人们会看到或者体验到新的方式是如何起作用的。实验和探索会引导他们产生全新的理解。外部影响也会为他们带来新的观察角度。团队成员开始尝试新行为和新想法。

**实践与集成 (Practice and Integration):** 只有想法是不够的。人们需要实践来学习新技能或是适应一个新结构。刚开始可能成效并不明显,但是随着实践的深入,效果会有明显提升。

当人们在改变的不同阶段之间进行转换时,可以通过对下述**三个方面**的关注来帮助他们。

**什么东西人们认为最有价值 (What People Value):** 识别团队成员在旧有方式中认为最有价值的东西。放弃已经失效的事物,同时要想办法把人们珍视的价值带入到新过程中。通过认知过去方式中价值的事物,你会发现人们当时并没有错。过去人们认为那是个好主意,在当时确实如此。发生变化并不意味着他们过去的所作所为是错误的,当人们相信这一点的时候,他们可以前进的更加容易。

例如,在一个团队进行产品发布之后的回顾时, Lakshmi 和她的团队认识到他们需要将团队人员增加 50%,以跟上对产品开发的需求。他们对于产品的成功感到非常兴奋,与此同时,他们又担心会丢掉现在有很强凝聚力的小规模团队。加入新成员之后,团队领导者会向他们澄清团队的价值观,并让团队继续保持他们需要的实践。当团队规模变大之后,团队的原有成员会对他们之前的实践积累排定优先级,以决定哪些是需要继续坚持并发扬广大的。

**临时结构 (Temporary Structures):** 临时结构帮助团队渡过在旧方式与新方式转换之间的混乱阶段。临时结构可以是计划、角色、会议、方法——只要是有助于完成团队从现有状态到目标状态的过渡,任何机制都可以作为临时结构。