

一个项目团队的敏捷之旅



撰文 / 莫映

题记

近日，欣闻首届“敏捷中国日”开发者大会将于六月初召开，这无疑是中国敏捷开发者的一件大事。敏捷方法历经数年传布，以其注重实效的特质，已渐入人心，并开始焕发出勃勃生机。不少一线团队更是早已将其付诸实际，并从中获益良多。本文讲述了一个团队改进的故事，取自笔者近期接触的一个真实项目。希望能对那些关注敏捷方法如何应用于实际项目的读者有所帮助。文中采用的敏捷实践主要来自极限编程（**Extreme Programming**），读者可以从近期出版的《解析极限编程（第二版）中文版》中找到有关极限编程的详细介绍。

背景

不久前我接触到一个项目团队，并且作为开发过程的跟踪者介入了其中。在我进入项目时，需求调研已于先期完成，部分模块的开发工作也已由程序员自发的开始了。但是，因为向客户承诺的项目启动时间在一个月多前，而部门人手一直捉襟见肘，因此真正的大规模开发直到最近才展开，客户对此却毫不知情。

先说说系统：整个系统由三个模块构成，其中一个基于第三方系统开发，并且有一部分工作被外包了出去；另一个则迫于进度压力，沿用了来自其他项目的一个遗留系统，虽然系统设计混乱，代码维护困难，但是以往的运行记录良好；最后一个模块则要重头做起。

再说说团队：由于项目中要用到 **Struts**、**Spring** 和 **iBatis** 框架，而成员们对这些框架的熟悉程度还不够。好在先期已为他们做了简短的培训，通过一个迷你教学项目，大家在短期内已经掌握了如何使用“**SSI**”的基本技能。不过，从我对他们代码的复查情况来看，成员们的编码习惯尚有欠缺。代码中随处可见的“**Bad Smell**”让人不禁心生忧虑。

看上去这是个风险颇大的项目：

- “ 我们真正能够控制的部分，其实只占整个系统的三分之一。团队中每个成员对能否如期完成预定的功能都没有把握，当项目经理问及系统能否如期交付时，我很坦白的回答说：目前还不知道。
- “ 团队成员有待历练，团队过程有待改进。改进需要时间，在有限时间内还要将部分精力投入到团队改进上，这无疑增加了难度系数。好在通过与包括管理人员在内的所有成员的沟通中，我了解到，大家对团队改进的必要性和紧迫性都有所认同。这在一定程度上为团队改进工作的开展创造了条件。

第一个迭代周期

在和团队成员进行短暂磨合后，我开始着手规划第一个迭代周期。

- “ 为了让这个复杂而棘手的问题尽量简单，同时也为了使我们能在力所能及的范围内做得更好，我将大部分精力集中于指导团队实现系统的三分之一功能，就是重头做起的那个模块。同时，这也为我们赢得了更多团队改进的时间。
- “ 为了尽量打消包括项目经理在内的团队成员们对项目进度的疑虑，我将“**迭代计划**”作为团队改进的第一个“故事”（极限编程中的概念）。从目前现状来看，相信这是团队改进最迫切的，同时也是实施难度和阻力相对较低，最有望获得成功的。考虑到成员对敏捷方法还较为陌生，我并不急于将诸如持续集成、测试先行一类，听起来很“酷”的敏捷实践放到团队的改进计划中。
- “ 作为一项辅助措施，我让程序员在开发期间每天都向代码库提交当天完成的工作，然后在次日复查所有昨天的代码。通过一个叫 **Jupiter** 的代码复查工具，我把藏在代码中的“**Bad Smell**”逐一找出，并告知相关责任人。这么做的目的是希望通过一段时间的持续复查，在一定程度上矫正开发者的编码习惯。在目前还无法要求程序员彼此互查的情况下，复查工作毫无疑问落到了我的肩上。不过，常人在一天内完成的工作不会太多，因此以天为单位来复查代码还算轻松。更重要的是，“**Bad Smell**”一旦出现，会在至多一天的时间内被发现，这样的反馈速度应该是足够敏捷了。在实际实施过程中，我发现程序员的响应速度很快，当通知他们复查结果后，一般都能在半天到一天的时间内完成所有的代码修改。当然，起初“复查——修改——再复查——再修改”的反馈十分频繁。有时，我甚至因为不厌其烦的打搅他们而觉得很不好意思。还好大家都理解，在迭代周期快结束时，成员们对代码复查的效果都还算满意。

插曲一：加班

在迭代开始后的第一个周五，团队召开会议。为加快进度，决定周末加班（当时迭代计划还没做好）。听到这个消息，成员们的情绪明显有些糟糕，阴郁的气氛笼罩着团队。

也许很多团队都已习惯加班这样的极端做法了，听起来这似乎应该是极限编程的实践，“极端”的英文不就是 **extreme** 吗**J**。做出加班的决定，很大程度上归因于人们对未知事物的恐惧心理和缺乏明确有效的反馈机制。大家对团队的开发现状和项目进度都心里没底，因此，在前途未卜的情况下，尽量往前赶以争取主动的心情是可以理解的。

不过，在决定加班前，也许我们可以让事情变得明朗化一点：项目经理或业务分析师根据客户的商业价值将复杂的问题域分割成一系列“用户故事”，然后由程序员负责将故事细分成任务并估算时间，估算的过程也是业务观点和技术观点彼此调和的过程。经过这道工序之后，含糊不清的问题在充分沟通中被逐渐澄清，人们的恐惧心理在很大程度上也得到了缓解。

为了赶进度，人们经常会打两张牌：时间和人员。在有限的时间内增加工作时间，就是加班；而增加人手在《人月神话》中已经被证明是不可靠的。有趣的是，我就真的经历过一个“堆人式”的项目。伴随项目的延期，不断有人填补进来，团队规模最多时有近二十人。可惜那时我尚不知敏捷为何物**J**

其实，敏捷团队的手里还有第三张牌：当现有资源不足以完成所有用户故事时，而且团队士气也向我们发出警告，试图通过增加工作时间来追赶进度已经变得十分危险时，不妨在当前的迭代周期里有选择的完成一部分故事。这么做的前提是：要有一个对当前问题域的分解，并形成经过估算的故事。接下来的工作就像是在超市买东西：货价上的商品中，有些是你要买的必需品，有些则不是，有些是你很想要，但又超出预算开支的。当你走到超市出口准备买单时，假如兜里的钱不够支付你挑选的全部商品，那么很简单，你需要重新考虑哪些该买，哪些不该买。在这一点上，我们也许应该向勤俭持家的女士们学习，学会如何精挑细选，在聪明的花掉兜里“钞票”的同时，让“购买物品”的价值最大化。

“加班事件”让我联想到另一件事：某些组织中，时常会对加班加点的员工给予表彰，并将这样的先进事迹广为传播。平心而论，作为加班者，这毫无疑问是他辛勤劳动应得的回报。只是，将这样的事作为典范加以宣传却是有悖于极限编程的价值观的：极限编程更加鼓励精力充沛的工作，因为这是对他人尊重的体现。假如延长工时可以让项目不至失控，我们可以选择加班。但如果不是这样，休息其实是尊重自己和他人的最好方式。对自己身体的透支可能会毁掉未来数天的工作，这对你和团队都是不利的。疲惫不堪的人很难意识到，自己正在增加整个项目的风险。

插曲二：形同虚设的 Wiki 公告

在第一个迭代周期里我犯了一个错误。为了把项目现状及时反馈给团队内外，我用 Wiki 开了一个项目专区，将分解的用户故事、进度预估，还有团队的日常公告贴到了上面。我相信公示是必要的，这体现了极限编程中的沟通和反馈。选择 Wiki 的初衷，则是因为我的个人偏爱。但事实证明，这样的电子化手段并没有起到预期的效果。除了习惯查看 Wiki 最近更新或者订阅 RSS 的员工外，其他人根本不会主动去浏览 Wiki。于是，对那些不了解项目现状的人而言，摆在他们眼前的依然是个黑箱。后来，我吸取教训，改用了常人更容易接受的沟通方式——邮件，还有直接的面对面对谈。这个教训让我再一次明白，在敏捷团队中为何会有一面“故事墙”：将富含信息的内容以一种常人更习惯的形式，摆在显眼之处，体现了很好的人性化。

第二个迭代周期

经过上一个迭代周期，团队对计划游戏已经较为熟悉了，因此在开始新周期时，迭代计划进展得很顺利。在完成故事估算后，我和程序员们一起坐在会议室里，在白板上讨论哪些故事先实现，哪些应该放到后面。然后将讨论结果绘制成图，并通过邮件发送给项目经理和部门主管。团队内外都有一些非常关注项目进展的人，同时他们也拥有决策权，这些决策往往事关团队的成败。为了消除 VIP 们对进度的疑虑和恐慌，将团队内部口头讨论的结果不失时机的以非正式文档的形式反馈出去是非常有必要的。没错，就是文档，这是人们熟悉的一种沟通手段，不过仅是用于沟通之用，刚好够用就行。

进入第二个迭代周期，团队的开发速度渐趋稳定。当我在迭代中期评估项目进展时，欣喜的发现，一切都在有条不紊的按计划进行，团队的工作气氛也变得轻松起来。虽然，个别次要任务因为某些不可控的因素导致了延期——某些开发任务是由团队外的人员承担的，而他们正深陷于另一个项目，自顾不暇——这样的安排看起来有些不合理，不过这并不影响总体进度。我们甚至能安排出一定的时间，将某些延期任务交给团队成员自己来完成。当我们将项目进度再次以图表反馈给项目经理时，他在给大家的回复中表示：非常满意。

项目的稳步进展让我有了更多机会能将精力投入到团队改进中来。本次迭代中，我开始尝试结对编程，方法是：每天投入大半天时间和几乎团队中的每个程序员结对。虽然这种“车轮战”式的结对有些特别，但却是经过慎重考虑的：

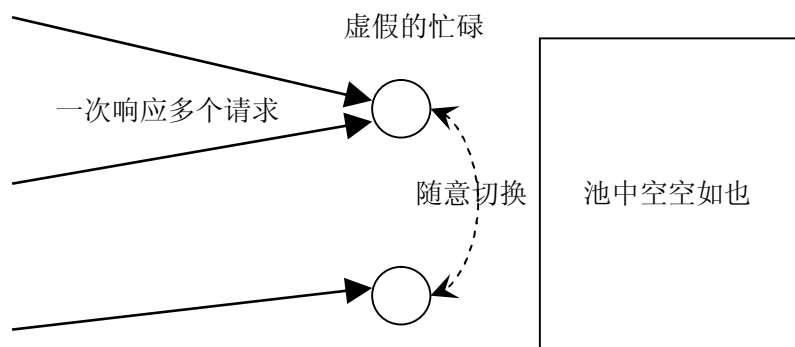
- “ 我希望通过这样的形式，为成员们示范好的开发实践，同时不失时机的灌输敏捷思想。言传身教的效果往往要比纸上谈兵来得好。有一次，一位程序员告诉我，运行在他本机的系统出现了一个诡异的错误，从异常堆栈来看应该是某个 **iBatis DAO** 类的问题，可是经过无数次费时的部署和服务器启停后，依然没有找到症结。此时，我不失时机的为他示范了一个单元测试，原因很快就被找到了——**Spring** 配置文件里有个很隐蔽的错误。对他而言，这是一次“印象深刻”的经历。单元测试还有很多好处，在开始实践测试驱动开发之后，这种好处还会提升到更高的层次。但对于一个原本就对单元测试不“感冒”的人而言，这无疑是一个非常实在的起点。另一次，一位结对者问我：为什么每回我都喜欢改完一小段代码并且确认无误后就往 **CVS** 提交，而他却喜欢做完所有修改后一次性提交。这是灌输敏捷思想的好时机：对一个习惯了敏捷思维的人而言，迭代和小幅前进的原则会很自然的体现在他的日常行事之中，即使这并没有被明确的定义为某项敏捷实践。
- “ 通过结对，可以鼓励充分的沟通，并且更细粒度的把握项目风险。在上一个迭代周期，团队中有时会发生这样的事：当我走到某位程序员身边询问进展时，他告诉我“遇到了一个问题”，而这个问题也许已经困扰了他一天。在习惯开周会的团队中，问题恐怕会被延误得更久：所有问题都将被堆积到周五的会议上讨论，即使问题在周一就已经发生了。按照极限编程的观点：谈话的嗡嗡声是团队健康的标志，沉默只会使风险堆积。一天下午，在和一位程序员热烈讨论过后，我终于说服了他，用户想要的也许不是将当前报表页面导出 **Word** 文档，而是生成一份打印稿，并且该项功能现在的优先级并不高。如果没有这样的沟通，也许会有大段时间被浪费在实现这个从未得到证实的功能上。另一次结对经历则是我和一位程序员一起完成了一个大模块的代码迁移，这次结对的目的在于“**Double Check**”：一个人在操作键盘时，另一个人监督操作的正确性。这样，我们可以及时发现迁移过程中不小心引入的错误。
- “ 充分沟通可以减少孤独感，而结对则形式化的强调了沟通。对于沉默寡言的团队成員，我个人很怀疑他能在这个团队待多久。
- “ 团队成员自身还没有足够的结对经验，为了让结对更有效率，两位结对者中必须有一位能主导结对的过程。

一句题外话，有些人也许对结对存在误解，认为它妨碍了个人的私有空间。其实，结对并不意味着不能独立思考，需要时大家可以坐到一起，不需要时也可以回到各自的私人空间。理想的敏捷团队对工作空间会有一定要求，比如敞开的没有隔板的办公桌，但即便现实环境有限制，也没什么大不了。相反，如果认同结队的价值，人们反而会更加珍惜在“恶劣”环境下的结对机会。

插曲三：人员更迭

项目中期，曾有位出差归来的程序员中途加入团队。可是分配给他相应任务后还不到一周，他又被调到了别的项目。于是，原本安排与他结对的程序员，花了大约两天的工作交接和沟通时间都被白白的浪费了。此外，还有一种常见现象是“单人多任务”，比如有些成员还同时承担着其他项目的工作。敏捷方法的观点认为：工作内容的频繁切换即不利于团队的整体感和连续性，也不利于生产效率。这让我联想到一些组织中时常发生的一类现象，我把它称为“被池化的人”。想必读者对数据库连接池不会陌生：作为一种创建代价高昂的资源，数据库连接一般会被预先实例

化并放入池中，一旦数据库访问请求到来，就会有池中的实例来响应。类似的，某些组织中，人们即被看作是一种可被池化的资源。当有项目“请求”到来时，池中“实例”就会去响应请求。但是很多时候，有的“实例”也许要同时响应多个请求，而响应完毕后又会上马上被安排处理其他的请求，因而池里永远都是空的，员工们得不到充分的修整。更甚者，有的“实例”还未处理完当前请求，就会被“调度逻辑”重新分配，以响应别的请求。在这样的氛围下，人们只是满足于每个个体都在忙于敲打键盘的虚幻效率，而忽略了应该让员工尽量与熟悉和信任的同事在一起工作，也忽略了这种随意切换带来的高昂代价（比如时间和精力的浪费，也许这可以换来两天的休息日，而非加班）。也许将人比作可以随意调配的资源本来就是错误。从这个意义上讲，人力资源（Human Resource）一词，至少在计算机领域恐怕是一种误导。



图：被池化的人

故事还在继续

文章即将结束，不过项目仍在继续，也许这样的敏捷实践案例就是我们不少项目团队每天的真实写照。作为尾声，这里还有些话要对读者说：

教练的作用

我们可以发现，在团队改进中，教练的角色至关重要。就如读者在故事中看到的，我的职责是：发现团队改进的机会，并引导大家尝试解决问题；当团队在压力驱使下做事时，提醒他们做简单的事情；既要有足够的技能来教导团队成员学习不容易自学的知识，示范有效的实践，也要对整个团队负责，保持团队在可承受的节奏下工作，并持续改进。最后，作为教练的一项重要工作还包括：找出知道答案的人，并把信息及时传达给有权力做出决策的人。

更开阔的视野

软件开发不仅是程序员的活动，缺少了对所有参与人的利害平衡，无论对项目还是团队本身，都将是危险的。技术人员对敏捷方法情有独钟往往是从重构、测试驱动这些明显带有浓厚技术色彩的实践开始的。但是要想获得成功，只关注这些还不够。在敏捷方法的受众里，有些人也许不能为开发做出贡献，但他们的观点对团队的成功却很重要，比如：项目经理、测试人员、客户代表。项目进展期间，我们要时刻注意协调技术观点与业务观点的矛盾。此外，要想保证项目的成功，我们还要将视野扩大到团队以外。有时，组织机构中主管人员所做的决策往往会对团队产生很大的影响。如果漠视或回避这些影响，恰恰就是漠视和回避充分有效的沟通。经验告诉我们，在敏

捷方法的实施中，最大的困难在于：你需要用事实来说服别人接受你的方式，尤其是那些拥有决策权力的人。只有所有人都有一致认同的价值观，敏捷方法才能最大限度的发挥价值。