其实上一篇我有讲过一些项目初期我做过的一些工作，还有公司运用敏捷的方法。只不过到这里我才有写一个系列文章的想法。这里我还是啰嗦几句，上篇我们谈到了，我们公司真正做敏捷其实是在称之为Iteration 0（或者叫Iteration Prepare）之后，也就是我们已经有了很充足的准备，可以开足马力实施项目的时候了。其实这样做也是为了降低风险，一来我们在前期不用投入很多人力，只是投入几个金兵强将，做好构架以及在这之上完成一个Demo。然后后面的Iteration就有了参照，就可以理解大家都在做Iteration 0的复制即可。有点想开连锁店的感觉。这种方法很适合项目外包型的公司，前提投入牛人，后期在随时调入/调出不用很强也不用很固定的人力。外包公司的人力调配可是一门艺术，这里我就不多说了。

好，开始说项目吧，目前我的项目正在做第一个Iteration，也就是Iteration 1，前几天开过Iteration 1 Planning Meeting，个人认为非常的失败，在这几天的运行来看也确实非常糟糕，问题主要表现为，我们在Planning Meeting上面没有完全按照要求把Story分解为2-16小时的Task。或者说我们根本就没有分解。而是把完整的Stroy估算成为时间。基本上都是5天3天这样的时间。可能大家都认为一开始我们拿到的Story都非常的简单，轻视了。所以在后面的几天力，我可以说完全无法跟踪项目进展，Iteration都快要结束了，可是一个Story都还未完成，也不知道Story完成了百分之几。所以这是我在下一个Iteration首要要解决的问题。

还有一个很大的问题就是每日晨会的时候大家都只是说了昨天做了什么，今天要做什么，可是从来都没有人说问题，我也不知道这是好事还是坏事，我总感觉项目初期应该大家都有很多问题才对。我不知该如何引导，希望后期会有所改进。

目前项目还出现了一个风险，前期投入的一个人，现在临时变更了，突然替换了一人，我没有料到会这么快换人，我还未很好的考虑换人如何应对，如何能让他快速的融入到我们的团队中来。我在想是否应该整理一个项目快速入门手册，方面后面如果再有新人加入时，能快速掌握目前开发进度。

还有一点，也是作为项目经理的我最失败的一点。关于User Story的编写问题。因为初次尝试，写User Story的时候我只知道按照需求的方式把User Story照搬了一遍，但是这样给开发带来了不小的麻烦。首先一下子开发无法转变关联，一大堆大大小小的Story丢给开发，他们都不知道如何下手了。毕竟开发习惯了功能性的需求。不习惯完全业务的Story描述。后来我查阅一些文章才发现其实User Story也可以和MVC结合，按照大小一致的粒度去编写。分为数据（史诗级User Story）和操作（一般User Story）这样更便于与开发的沟通和理解。

还有一个问题，就是项目初期大家对敏捷的认识也不够。都是初试敏捷，我对这方面也没有系统的培训，而只是在项目进行中做了一些宣导，我觉得还很不够，所以我准备过两个Iteration专门对敏捷召开一次座谈会，大家一起讨论运行几个Iteration之后的心得，与大家一起分享，一起学习，如果效果不错的话，我准备每隔一个月就召开一次。当然如果是下一个项目我一定会项目开始之前就做这方面的引导培训。

从项目运行到现在我们已经开过两次Iteration Review Meeting会议，因为我们的Iteration周期比较短（一周），所以我们采取Iteration Review和Iteration Planning两个Meeting连续开。

通过两次Iteration Review会议，我们吸取第一次Iteration Review的问题，在第二次Iteration Review做了很多的改进，首先在准备上我们更加的从容，我们在Iteration Review之前我们就把我们做的软件打包发布可在本机运行的版本。那么在会议上就可以马上进入Demo环节，避免浪费大家的时间。第二个改进，我们在回顾上一个Iteration我们那些做的好，那些做的不好，那些需要改进的环节，我们不再由我一个一个的点人来回答。而是用事先准备好的小字条，让大家把自己的想法写下来，每人至少要写一条好的方面，一条不好的方面。写完之后每个人依次上台来向大家分享，并且要求就每一条分享举出在上一个Iteration中的实际事例出来，这样的分享更有价值。我觉得效果非常的好。最后我会就好的方面做一个总结，让大家在下一个Iteration对好的方面继续发扬。不好的方面，我选出了其中最重要的2-3条做了一下分析，大家一起讨论提出了一些可行的解决方案，并且我要求他们在下一个Iteration进行改进。整个Iteration Review持续了2个小时，我觉得这次的效果非常的好，达到了我们预期想要的结果，给下一个Iteration建立的良好的信心和鼓舞。

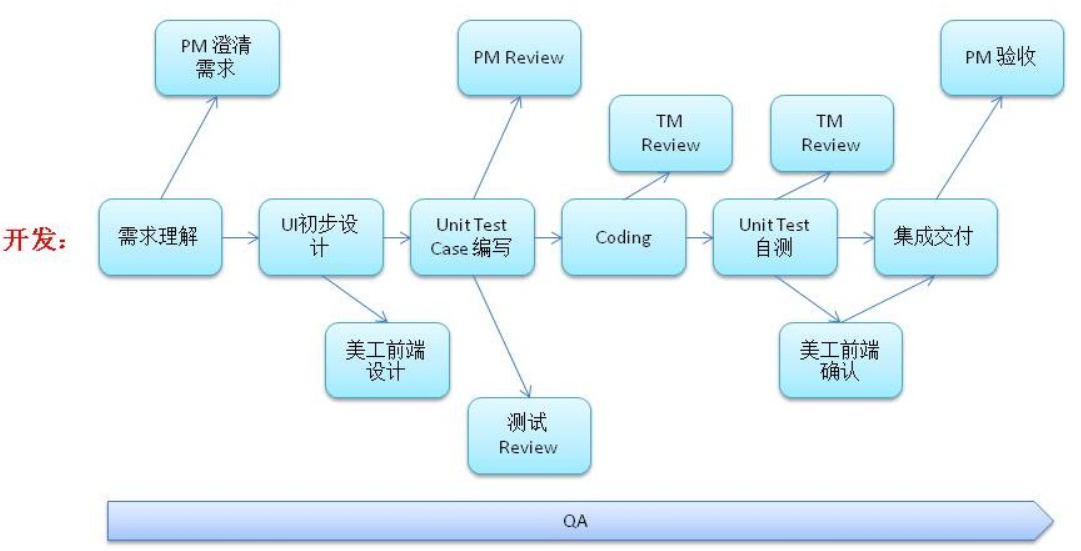
我自己的一点点感受，我觉得不要把会议流于形式，而要真正发挥其作用，那么才有意义。不让就是浪费所有参会人员的时间，那是一件非常可怕的事情。

我尽量保持每周都会跟大家分享一点我在上一个Iteration周期中碰到的问题以及收获，所有不一定有很多东西可以写，但是我尽量能把我认为有意义的东西留下来。

每次到了周五都会有一种罪恶感，到底我上一个Iteration完成了没有，对于敏捷来说，我们每一个Iteration或者是Sprint也好，都应该是阶段可交付。也就是说我们应该有一些对用户有价值的Story完成了。Iteration的产物不在多，而在于是否有可交付的Story，而对于可交付这个词我不知道大家的理解是不是一样，至少我认为就是得到客户的认可，对客户是可以用的场景或功能。而到目前位置我们已经结束第三个Iteration了。应该来说应该有东西可交付了。但是作为PM或者PO的我来说，我觉得还没有一个Story是可交付的。因为我认为可交付的意思是说，这个Story的价值都已完成，可以Demo演示给用户看，并且达到验收准则。那么关键词就是价值、Demo、验收准则。什么是价值？简单来说就是对用户来说是有意义的，最实惠的。比如说用户要一个查询系统特定用户权限的功能，那么我们有两种做法，一种是做一个权限查询的功能模块，可以查询所有用户的所有权限的通用模块。还有一种做法是，在特定场景下，按照特定用户查询权限。第一种很通用，满足客户要求了，甚至超出了，但是这样就是好的吗？第二种看起来很呆板，但是是不是用户更想要的呢？而且开放花的功夫比第一种更小，带来的价值更大。然后是Demo，为什么要强调Demo呢，我看来有这么几个优点，1.不是开发环境，可以更接近正式环境。2.更直观，不管是演示还是用户体验也好都很好。3.如果要做Demo开发人员会更用心，不会认为只是随便跑跑看。最后是要收准则，我的验收准则很简单就是在我这里0缺陷。有的人可能会说是不是太苛刻，我觉得并不苛刻，我是一个非专业测试人员，我更关注的是业务、价值，在我常规业务操作下还有缺陷的话，我相信真实用户也是不能接受的。

最后我想给大家分享一下项目运行到目前为止碰到的沟通问题，我是如何应对的。

首先要费点口舌介绍一下我目前的团队成员大致情况，PM项目经理1人，TM技术经理1人，开发6人，美工前端2人，测试2人，QA1人。 因为我们不是完全按照敏捷Scrum角色来定义，所以角色会比较多，公司规定嘛，没有办法。所以这么看来沟通就非常的重要了。我画了一个简单示意图，该图以开发一个完整Story为前提，以开发这个角色为主线，在开发过程中我们的团队协调沟通示意。



系列谈上一回我们主要谈到了沟通问题，相信大家都知道Scrum的5个价值观：沟通、公开、专注、尊重、勇气。所以我一直认为沟通是第一位的。只有拥有良好的沟通敏捷才可以成功。当然其他几个都同等重要。

到目前为止我们项目敏捷已经实施了将近3个多月了。按道理说我们应该有很多收获了，但是我个人总觉得收获不大，或者说是因为我们的进程太慢。总感觉使不上劲，最近看一些论坛说过一句话，敏捷实施的好不好或者彻不彻底要看你Scrum Master（SM）这个职位或者职责实施的彻不彻底。现在我们团队没有定义SM这个职位，所以我们敏捷其实是非常的不彻底的。只能靠我个人的努力不断的去接近。心力交瘁啊。

上干货，今天主要谈谈几个会议吧，其实Scrum中的所有会议都是非常重要的，1、Sprint计划会。2、每日例会。3、回顾会。4、反思会。有的地方还会加一个Release发布计划会。

Sprint计划会议：（这个会议我们做的最不好，所以我说的最多）

Product Owner（PO）讲解Story，特别是业务背景，验收准则等。然后Team团队一起评估Story，并分解成2~16小时的Task。我们一直都没有做好这一部分的工作，主要问题有这么几个方面：

1、我们没有真正意义上的PO，基本上是客户提出需求，然后主要有Project Manager（PM）来分析理解需求，然后Technical Manager（TM）协助PM编写Product Backlog，当然写出来的Story就更偏向于功能级的Feature List。所以首先在需求讲解这一部分就比真正的PO要弱了很多，业务背景也差了许多。这是我们的不足之一。

2、开发人员没有很强的主动意识，或者还不愿意去改变，或者说公司并没有很直接的去推动敏捷。开发人员总是依赖于PM或者TM去分配任务，在计划会议的时候不喜欢或者说不善于去思考问题。导致Story分解Task的时候无法进行下去。或者非常困难的进行下去又都是形式上的分解。何为形式化分解了，就是每个任务都是分解成同样的5步：（1）需求理解。（2）界面设计。（3）测试用例和概要设计文档编写。（4）编码。（5）自测。当然分成这几步没错，也总比不分解要好的多，但是过于形式化，大家并不喜欢真正的去思考，最后还是跑到了老套路去了，一个功能模块一个人去完成，不论分解成5步也好还是10步也好。完全没有体现合作、协作、团队。不过追根到底可能还是跟我们一开始分解的Story太偏向功能模块有关。

3、评估时间与单位问题，常见的评估工作量有两种度量单位，1、Story Point。2、理想时间。对于第一种Story Point来说比较抽象对于第一次实施敏捷的团队来说可能不好掌握，所以我们采用了理想时间。可是问题来了，虽然我解释了无数遍理想时间的意思是你完成一个task所需要的连续的理想的时间，强调是无打扰不间断很理想的。可是开发人员还是喜欢用8小时，16小时这样的时间段来评估任务。总是无法抛开1天8个小时这样的概念。在我强调我们最大利用率最多时有70%，也就是一天只有5.6小时理想时间时，他们依然很茫然。意识的改变真的是一件非常困难的事情。每天我在收集时间所用task时间的时候他们也总是把一天填满8个小时，每次我问实际时间是多少，他们总是说是8小时，似乎不填满8小时就是犯罪一般。

分析以上几方面的问题，我给出的一些建议是，1、增加与客户的联系，没有PO是硬伤，这个我也没有具体的办法，只能做好一切可能的沟通理解用户的业务背景与真实需求。2、Story编写应该跨功能，趋近客户价值。不应该是功能模块，这样不是敏捷的思想，没有价值可言。而且在分任务时必定会趋近瀑布模型一个人编写一个功能模块。增强开发主人翁意识，强迫每个人来分析Story应该如何划分task，要给出初步的界面设计和概要设计，要说出来，如果你来开发这个Story准备分几步来完成。并且如何完成。（简要说明）3、收集时间时，一天不然填满8小时甚至是不然填7小时，最多可填6小时或更少。估算时间时，就可以参考以前收集的时间来分析估算现在任务所需时间。

每日例会，也叫晨会：

晨会其实说起来简单，但是做好他也不是一件非常容易的事情。那么我觉得应该注意的几点是：

1、不要超过15分钟。（7人左右团队）

2、回答三个问题的时候，要注意，1）我昨天做了什么？但是要说明做的情况，是否已经做完？做完了是否解决了什么特别的技术问题？没做完预计什么时候可以做完？2）今天将要做什么？但是要说明今天的任务预计什么时候做完？3）遇到了什么困难？包括昨天我们遇到的困难？超过2小时候的困难一定要记录到Block看板里。还有今天我处理的事情中我可能会遇到的困难，我需要帮助。那么其他人员尽量主动的思考是否可以帮助到他的。然后约定时间会后一起沟通解决。

3、会上不要解决任何技术问题，如果会上有很多技术问题的话，那么一定是会下大家都很少沟通的原因。会上应该找好可以帮助到自己的人，然后会下去解决。

回顾会：暂时我们没有这一部分。这里也不做分析。

反思会：

反思会个人认为对于敏捷的持续改进来说非常有作用。在反思会当中，团队会自我发现我们在上一个sprint中大家最关心的碰到的问题。或者我们做的好的地方，大家都很认同的。那么反思会应该是在一个好的沟通环境当中，大家都非常放松畅所欲言。好的东西，我们要总结出来为下一个sprint发扬使用。不好的方面我们要选出2~3项做更深入的分析，找出深层次的原因，拿出解决方案。然后制定负责人，在下一个sprint中取改进。很重要的一点就是，如果有多个sprint都提到了同样的不好的地方，我们就要引起注意，把他特别的拿出来做分析讨论。拿出可执行的解决方案。派专人跟踪执行情况。作为一个团队，我们每一个人都有责任和义务一起去思考拿出解决方案，并一同协作解决。等会我会加上我们几次反思会的一些成果分享给大家。我也非常想看到大家的分享。呵呵

最后还是强调Scrum的价值观：沟通、公开、专注、尊重、勇气。

项目在跌跌撞撞中终于尘埃落定，这个项目不大，计划总人力34.2人月，实际人力投入大概28左右。由于是内部项目，并且也可以算是实施敏捷方法的试验田，所以估算比较充足，不过不管是CPI还是SPI，从各方面的数据上来看这个项目还算是比较成功的。缺陷率比较低；项目运行顺利人力节省；客户满意；基本上没有加班；总之我对此项目还是非常满意的，当然还要多谢项目中的每一位成员的努力，特别是TM（Technical Manager）的合作与支持。

最后跟大家一起再回顾一下我的整个项目：

整个项目过程有点像网上常说的Fall-Scrum-Fall的过程，先瀑布->然后敏捷->最后瀑布

1. 项目初期，出资人或者项目提出人预指派（还未授权）PM跟进客户提出的一些想法，与客户沟通，结合公司的现有能力或者技术，提出产品的一个愿景，然后产生Feature List，以及系统的大致业务模型和构架。

2. 立案（PM真正授权），与客户达成意愿，把上面所说文档提交公司申请立案，这个时候需要做人力预估和时程的预估。当然这个是很粗的一个预估只是用来立案之用。

3. 市场分析与进一步计划，此时TM会参与进来，与客户一起讨论更细一点的Feature，具体到模块与功能，PM和TM一起做出细部的人力预估，产生人力与时程的基线。

4. 预研、架构与实作，对于关键技术点我们首先需要去做风险评估，并且为了降低风险而做一些预研性的工作。设计系统骨架，并实作一两个例子。因为我们运用的是敏捷，所以这里不会做很细的设计，我们尽量先做出来后面不断重构完善，但切记一定要不断重构，不要积累设计问题，不然必成大祸。

5. 开始真正的敏捷，这个时候已经进入真正的实质性的开发工作，整个开发阶段我们运用的是敏捷Scrum框架。

5.1. 我们的Product Backlog也就是我们上面讲到的细化过后的Feature List。首先我们会做Release Plan，定义迭代周期。

5.2. 然后选取PB中的一些Feature到Sprint Backlog中，然后开计划会议Sprint Planning Meeting。Scrum的几个会议我在前面的文章中已经有讲，这里不再细说。

5.3. 然后产生任务墙看板。在迭代的运行期，会通过每日晨会，任务墙以及燃尽图去跟踪项目进度，控制偏差和风险。

5.4. Sprint结束后会有一个Demo Meeting，大家一起看一下这个迭代我们的成果如何，是否我们是Story可以交付。然后收集反馈问题记录下来，作为下一个Sprint可能将要完成的任务。

5.5. 紧接着我们会开反思会，回顾一下上一个Sprint我们哪些做得好的，哪些做的不好，哪些本可以做的更好的？选取其中大家认为最重要的2-3条，拿出措施Action Plan，再下一个Sprint中去改进。（我觉得这一点我们做的非常好，个人认为这个会议也非常的重要，可以使我们一直都在持续改进）

然后周而复始一个接着一个的Sprint，直到开发结束。

6. 进入系统测试阶段，专门的测试人员会对整个产品做测试与验收工作。开发和测试一起合作完成。

7. 测试验收完毕后，提交最终版本给用户，用户做UAT验收工作。

8. 验收通过，最后做结项工作。总结项目，收集整理文档等。

整个项目完结。

会议议题

一、如何开好Iteration Planning Meeting迭代计划会议

1. 会议的目的是什么

2. 会议的流程是什么

3. 如何提高会议中的估算效率

二、本项目中的敏捷实践方法

1. 沟通机制的建立

2. 开发流程的定义

3. 敏捷宣讲Training

4. Team所有成员主动性（RD直接参与设计）

5. 测试的早期参与

一、Iteration Planning Meeting

1. 会议目的

ü 全体成员了解Story，PO讲解本次迭代的所有Story

ü 估算本次迭代的规模

ü 划定范围，下一个Iteration我们将要完成什么

ü 制定本次迭代计划，产出修正的Sprint Backlog

ü 产出任务看板或者与之相类似的东西（我们用的是Excel看板）

2. 会议流程

2.1 会前准备：

2.1.1 根据Product Backlog挑选出本次迭代会议可能完成的Sprint Backlog列表（可能包括上个Iteration演示后的客户反馈以及技术改进重构等，列表必须已经按优先级排序）。

2.1.2 提前把此Sprint Backlog列表发给团队所有成员，大家事情可以预习，并尽可能对列表中的Story提出问题记录疑点。

2.1.3 在会议前，还要根据本次计划会议要讨论的Story的个数以及难易度，对每个Story的讲解以及评估计划好时间。以便在会议中主持人可以控制好时间，如果超时可考虑跳过。

2.2 计算本次迭代可以利用的资源时间，根据上次Iteration经验计算本次Iteration可能完成的SP（Story Point）。

2.3 请PO或者SA按照初步的Sprint Backlog条目顺序讲解需求，讲解完后团队所有成员与PO或者SA进行Q&A。之后团队对Story进行评估，估算SP。

2.3.1 Q&A的技巧，我们总结了一下做了一个Q&A checklist如下（一致认为最后三点最为重要）：

ü 要做什么功能？有什么特殊要求？如性能等

ü 有无UI界面？有特殊操作上的要求吗？

ü 与系统中其他功能的关系是什么？

ü 检出点是什么，最重要的几个Test Case

ü 验收标准是什么或者说是如何Demo

2.3.2 估算技巧（文章后面单独列出）

2.4 估算完一条后再进行下一条，直到团队本次Iteration时间饱和为止。

2.5 大家一起最后再确认一次确定好的Sprint Backlog，并维护任务看板或与之相类似的东西（我们用的是Excel看板）。

2.6 维护看板时，如果有充足的时间最好每个Story都要被分解完Task，团队一起对Task过一遍看有无异议。

2.7 确认完毕后退出会议。

3. 如何提高会议中的估算效率

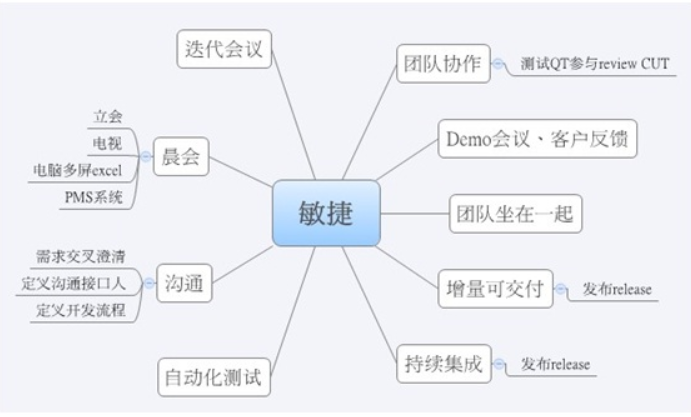
PO讲解需求，直到可评估即可，细化内容下去和单独沟通；

只对Story进行SP评估，不评估Task；

PO澄清需求并做Q&A后，利用评估纸牌法做SP评估；

三个六，6分钟PO讲解需求，6分钟Q&A，30秒内出牌。如果有异议6分钟PK（并无绝对，总之要控制时间）；

二、敏捷实践



版本发布顺利原因总结，四步骤走：

1. Lock code；

2. 确定范围；

3. To do list 验证；

4. 可交付范围。

具体动作：

1. 发布前一天：

2. 清楚要发布的范围；

3. 自测通过，锁定代码；

4. 准备工作充分，List检查项，一一检查，包括安装手册、数据库脚本、程序包、干净环境（虚拟机）安装；

5. 发布当天：

6. Sanity test执行到位；

7. 发布范围核实与确认；