**基于敏捷的项目管理流程（初稿）**

# 敏捷是什么

Scrum 是一个用于开发和维持复杂产品的框架 ，是一个增量的、迭代的开发过程。在这个框架中，整个开发过程由若干个短的迭代周期组成，一个短的迭代周期称为一个Sprint，每个Sprint的建议长度是2到4周(互联网产品研发可以使用1周的Sprint)。在Scrum中，使用产品Backlog来管理产品的需求，产品backlog是一个按照商业价值排序的需求列表，列表条目的体现形式通常为用户故事。Scrum团队总是先开发对客户具有较高价值的需求。在Sprint中，Scrum团队从产品Backlog中挑选最高优先级的需求进行开发。挑选的需求在Sprint计划会议上经过讨论、分析和估算得到相应的任务列表，我们称它为Sprint backlog。在每个迭代结束时，Scrum团队将递交潜在可交付的产品增量。 Scrum起源于软件开发项目，但它适用于任何复杂的或是创新性的项目。

## 四种核心价值

* 个人和互动高于流程和工具
* 可工作的软件高于详尽的文档
* 客户协作高于合同谈判
* 响应变化高于遵循计划

## 十二条原则

* 我们的最高目标是，通过尽早和持续地交付有价值的软件来满足客户。
* 欢迎对需求提出变更——即使是在项目开发后期。要善于利用需求变更，帮助客户获得竞争优势。
* 要不断交付可用的软件，周期从几周到几个月不等，且越短越好。
* 项目过程中，业务人员与开发人员必须在一起工作。
* 要善于激励项目人员，给他们以所需要的环境和支持，并相信他们能够完成任务。
* 无论是团队内还是团队间，最有效的沟通方法是面对面的交谈。
* 可用的软件是衡量进度的主要指标。
* 敏捷过程提倡可持续的开发。项目方、开发人员和用户应该能够保持恒久稳定的进展速度。
* 对技术的精益求精以及对设计的不断完善将提升敏捷性。
* 要做到简洁，即尽最大可能减少不必要的工作。这是一门艺术。
* 最佳的架构、需求和设计出自于自组织的团队。
* 团队要定期反省如何能够做到更有效，并相应地调整团队的行为。

## 五个价值观

* 承诺 – 愿意对目标做出承诺
* 专注– 把你的心思和能力都用到你承诺的工作上去
* 开放– Scrum 把项目中的一切开放给每个人看
* 尊重– 每个人都有他独特的背景和经验
* 勇气– 有勇气做出承诺，履行承诺，接受别人的尊重

# 为什么要采用敏捷

公司目前开发模式看起来更像[边做边改模型](http://baike.baidu.com/link?url=AFeAg61HAmplMFh_nTO2eoJIVX9KJrcSthhMGs77UXmS0LwwGL2eo3veSwhrBFLEXrfGcBniYZeD8D0mUsEhAKF5sJ5_WRoic3vWJhtgcyUkUWCeZcKqVE-OOO_AY1DW59j4R3qX2SMDDFeJwebAN_)（本人入职时间较短，理解可能不够准确，欢迎补充），无固定的开发流程及过程控制。因“人口健康信息管理平台”建设需要，急需一种开发模式的指导。经讨论并结合现状，选择基于敏捷开发模式的项目管理方式做为该项目的管理尝试。

## 常见项目管理问题

**需求不稳定，变化后改动代价大。**需求为了适应市场会经常调整，但是一旦调整，很多开发工作就会受到相应的影响。这样就会影响到实际功能上线时间，导致上线延时，客户满意度下降，信任感也随之下降。且调整需求后会导致很多工作要重新开始，严重影响了技术同事积极性。

**团队经常加班，但战斗力不强。** 核心团队疲于应对需求、技术开发、老系统维护，没时间指导新同事技术学习，而新同事技能暂时还没有发挥出来干活效率低，任务经常延期，没有成就感。核心团队事情很多，没有时间整理项目文档，新员工没有文档上手慢。大家每天很多事情，需要加班才能完成，比较疲惫。让大家愉悦的开发，比疲惫的上班效率要高很多。

**交付软件质量差，离用户期望差距大。**大家辛辛苦苦做出来的东西，老板不满意、用户不埋单，付出的努力没有人认可。交付的软件没时间自测试，或者自测试不充分，交给测试又是一大堆问题。没有测试，直接出去给用户，相当危险。这样交出去的产品不仅仅影响了用户的使用，还影响了整个公司的口碑。

## 敏捷如何避免这些问题

**拥抱变化**，敏捷开发从一开始就建立在需求变更的前提下，只关注当前sprint时间盒内的需求，关注需求优先级，保证当前时间盒内需求能顺利完成。开发的过程中不需要太多的去关注后续的需求变更。

**自组织团队**，敏捷开发需要一支自组织团队，或者不具备的情况下将团队向自组织团队方向引导，让每一个人都参与到开发的全过程中，充分的调动开发人员的积极性，重点提高工作效率而不是提高工作时长。

**合理规划**，通过定期的规划会议，规划产品backlog和每个sprint计划任务，保证每个sprint都能产生可交付的成果。

**过程可见，**敏捷开发的过程最大限度的对团队和组织公开，每日立会更新任务状态，通过燃尽图使项目进度清晰可见。每个sprint结束都要交付可以使用的产品给客户，增强客户对产品按时交付的信心。

# 敏捷该怎么做

## 敏捷定义

* + 1. **产品负责人PO(product owner)**

主要负责确定产品的功能和达到要求的标准，指定软件的发布日期和交付的内容，同时有权力接受或拒绝开发团队的工作成果，确定产品需求优先级。如：产品经理、客户等。

* + 1. **流程管理员SM(scrum master)**

主要负责整个Scrum流程在项目中的顺利实施和进行，以及清除挡在客户和开发工作之间的沟通障碍，使得客户可以直接驱动开发。

* + 1. **开发团队ST(scrum team)**

主要负责软件产品在Scrum规定流程下进行开发工作，人数控制在5~15人左右，每个成员可能负责不同的技术方面，但要求每个成员必须要有很强的自我管理能力，同时具有一定的表达能力；成员可以采用任何工作方式，只要能达到Sprint的目标。如：程序员、UI设计、测试人员、数据库开发人员等

* + 1. **产品（Product）**

最终要交付给客户的可运行产品。

**对应禅道：产品**

* + 1. **Product Backlog（story）**

产品 backlog 是 Scrum 的核心，也是一切的起源。从根本上说，它就是一个需求、或故事、或特性等组成的列表，按照重要性的级别进行了排序。它里面包含的是客户想要的东西，并用客户的术语加以描述。我们可以叫它 **用户故事 （ story ）** 。每个用户故事应至少包含以下属性：

**故事ID，**需求的唯一标识，便于后续跟踪

**名称，**需求名称。

**描述，**参考的模板：作为一名*<某种类型的用户>*，我希望*<达成某些目的>*，这样可以*<开发的价值>***。**

**重要性（Importance），**PO评出一个数值，指示这个故事有多重要。例如 10 或 150。分数越高越重要。**(禅道中仅提供了四个等级的优先级)**

**预计,** 团队的初步估算，表示与其他故事相比，完成该故事所需的工作量。最小的单位是故事点（story point），一般大致相当于一个“理想的人天（man-day）”**（禅道中此处最小单位是小时）**

**验收标准，**故事应该如何在 sprint 演示上进行示范，是一个简单的测试规范。“先这样做，然后那样做，就应该得到……的结果”。

**对应禅道：产品需求**

* + 1. **Sprint**

Sprint冲刺、短跑，指的是短距离赛跑，这里指的是一次迭代，一次迭代控制在一个较短的时间段内（1-4周），也就是我们要把一次迭代的开发内容以最快的速度完成它，这个过程我们称它为Sprint。

**对应禅道：项目**

* + 1. **完成**

PO和ST需对sprint“完成”的定义达成一致，定义的完成标准是确定该sprint是否完成的唯一检验标准。

* + 1. **Sprint规划会议**

制定sprint周期内要完成的用户故事（story），确定sprint目标，对要做的用户故事进行需求细化并拆分成任务（task），并对每个故事的工作量进行估算。

* + 1. **每日立会**

Sprint周期内，每天特定时间由团队成员站立举行的会议，会议中团队成员轮流发言，每人说三件事情，昨天做了什么、今天要做什么、有什么问题。会议时长应控制在15分钟以内，会上仅表述，不讨论、不解决任何问题。

* + 1. **中期会议**

Sprint中期针对前期积累的问题需要做一些调整，可以发起次会议，会议中PO可以纠正需求偏差，在不影响该sprint时间盒总任务量的情况下可以对该sprint周期的任务进行调整（sprint周期较短时不建议发起中期会议）。

* + 1. **演示评审会**

演示已完成的成果物。

* + 1. **反思会**

Sprint演示结束后，回顾整个sprint中开发任务，在一个约定好的**固定时间段**对遇到的问题进行分析，将问题现象及解决方案录入知识库。

## 敏捷流程



* + 1. **项目立项、启动**

确定产品，确定团队成员及各成员工作职责；对团队成员进行培训使其了解产品，并了解产品开发过程。

* + 1. **Product Backlog编写**

PO将来自客户、市场、高层的一些创意、想法、功能需求和系统缺陷等进行分析整合，获得产品backlog列表（用户故事列表）。

* + 1. **Sprint规划**

召开sprint规划会议视为该sprint周期正式开始，会议上确定sprint目标（什么时间要达到什么样的效果），确定sprint时长后，根据以往生产率计算出该sprint周期内可以完成的用户故事点，依据估算出来的故事点从backlog列表中将优先级最高的用户故事放入该sprint时间盒。团队成员对用户故事进行分析并拆分成任务，对每个任务进行估算；团队成员认领已完成估算的任务。

* + 1. **Sprint执行**

ST成员对认领的开发任务编写测试用例，然后根据测试用例编写必要的单元测试，针对开发任务进行编码和测试工作。每日立会中反馈任务执行情况，更新燃尽图，公开工作进展。

* + 1. **Sprint演示评审会**

Sprint周期约定好演示时间，到了约定好的演示时间，ST成员对当前sprint周期内的用户故事进行演示，按照用户故事已定义好的演示方式进行演示，演示过程中应重点表现出“做了什么”而不是“怎么做”。

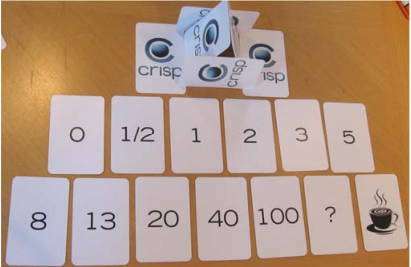
演示可以让其他人了解我们做了什么；演示可以迫使团队真正的去完成一些工作；演示可以增强团队信心；演示可以更直观的获得相关干系人的重要反馈。

* + 1. **Sprint反思回顾**

该环节是sprint周期中仅次于sprint规划会议的重要环节，为了以后的sprint能够更有效顺利的开展，务必确保该环节能顺利进行。

## 敏捷过程中的一些方法

* + 1. **计划纸牌**



每个人都会得到如上图所示的 13 张卡片。在估算故事的时候，每个人都选出一张卡片来表示他的时间估算（以故事点的方式表示），并把它正面朝下扣在桌上。所有人都完成以后，桌上的纸牌会被同时揭开。这样每个人都会被迫进行自我思考，而不是依赖于其他人

估算的结果。

如果在两个估算之间有着巨大差异，团队就会就此进行讨论，并试图让大家对故事内容达成共识。他们也许会进行任务分解，之后再重新估算。这样的循环会往复进行，直到时间估算趋于一致为止，也就是每个人对这个故事的估算都 差不多相同。

重要的是，我们必须提醒团队成员，他们要对这个故事中所包含的全部工作进行估算。而不是“他们自己负责”的部分工作。测试人员不能只估算测试工作。

注意，这里的数字顺序不是线性的。例如在 40 和 100 之间就没有数字。为什么这样？

这是因为，一旦时间的估算值比较大，其精确度就很难把握；这样做就可以避免人们对估算精确度产生错误的印象。如果一个故事的估算值是差不多 20 个故事点，它到底应该是 20 还是 18 还是 21，其实无关紧要。我们知道的就是它是一个很大的故事，很难估算。所以 20 只是一个粗略估计。需要进行更精确的估算？那就把故事分拆，去估算那些更小的故事！另外，你也不能搞那种把 5 和 2 加起来得到 7 的把戏。要么选 5，要么选 8，没有 7。

有些卡片比较特殊：

0 = “这个故事已经完成了”或者“这个故事根本没啥东西，几分钟就能搞定”。

? = “我一点概念都没有。没想法。”

咖啡杯 = “我太累了，先歇会吧。”

* + 1. **结对编程**

结对编程技术是指两位程序员坐在同一工作台前开发软件。与两位程序员各自独立工作相比，结对编程能编写出质量更高的代码。

结对编程可以提高代码质量。

结对编程可以让团队的精力更加集中（比如坐在你后面的那个人会提醒你，“这个东西真的是这个 sprint 必需的吗？”）。

令人惊奇的是，很多强烈抵制结对编程的开发人员根本就没有尝试过，而一旦尝试之后就会迅速喜欢上它。

结对编程令人精疲力竭，不能全天都这样做。

常常更换结对是有好处的。

结对编程可以增进团队间的知识传播。速度快到令人难以想象。

有些人就是不习惯结对编程。不要因为一个优秀的开发人员不习惯结对编程就把他置之不理。

可以把代码审查作为结对编程的替代方案。

“领航员”（不用键盘的家伙）应该自己也有一台机器。不是用来开发，而是在需要的时候稍做一些探索尝试、当“司机”（使用键盘的家伙）遇到难题的时候查看文档，等等。

不要强制大家使用结对编程。鼓励他们，提供合适的工具，让他们按照自己的节奏去尝试。

* + 1. **测试驱动开发（TDD）**

测试驱动开发(Test-Driven Development)，TDD的原理是在开发功能代码之前，先编写单元测试用例代码，测试代码确定需要编写什么产品代码。TDD虽是敏捷方法的核心实践，但不只适用于XP（Extreme Programming），同样可以适用于其他开发方法和过程。

**该方法对测试要求较高，不适合公司现状，有兴趣的同学可以自己查阅相关资料。**

## 所使用的工具

禅道、maven、jenkins、git。

# 执行过程中应注意的要点

Sprint开始后要通知所有相关干系人，声明新的 sprint 已经启动，要包括 sprint 目标和 sprint 信息。

尽最大努力确保sprint可以如期完成。

用户故事的变更必须有产品负责人（PO）执行，此处的变更指的是用户故事的任意一个属性的变更。

确保每日立会可以按时开始和结束。

确保sprint会议必须能产生明确的任务列表，为此必须确保产品负责人能够全程参与sprint规划会议。

与整个团队以及产品负责人一起开 Sprint 反思会议。

更新 sprint 数据文档。加入实际生产率和回顾会议中总结出的关键点。

# 补充说明

该文档是结合书籍进行整理的结果，文档中描述的过程可能表述不当，请见谅，同时也欢迎大家指正。

敏捷开发适合规模较小的团队，团队规模越大，沟通成本越高，管理成本也越高，团队规模最好不要超过15人；另外敏捷模式对人的要求很高，对团队成员的个人能力以及团队成员稳定性要求较高。

关于敏捷开发，如果感兴趣的话，业余时间可以阅读《硝烟中的Scrum和XP》、《敏捷估计与规划》、[scrum中文网](http://www.scrumcn.com/agile/scrum-knowledge-library/scrum.html#tab-id-1)等网络资源。

敏捷并不是万能的，它并不能解决所有问题，这里可能只是一个开端，只有在实践过程中不断的总结才能获得项目管理的最佳实践，希望能够与大家一起努力探索。