



敏捷开发

2011年9月

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

1 基本框架

1.1 敏捷宣言

敏捷开发模式通过长期经验积累，总结出敏捷宣言，也是敏捷开发的价值观：

- **人和交互** 重于 过程和工具。
- **可以工作的软件** 重于 求全责备的文档。
- **客户协作** 重于 合同谈判。
- **随时应对变化** 重于 循规蹈矩。

虽然右边也有价值，但是我们认为在软件开发过程中，左边价值更大。

1.2 敏捷原则

- 1、我们最优先要做的是通过尽早的、持续的**交付有价值的**软件来使**客户**满意。
- 2、即使到了开发的后期，也**欢迎改变需求**。敏捷过程利用变化来为客户创造竞争优势。
- 3、经常性的交付**可以工作的**软件，交付的间隔可以从几周到几个月，交付的**时间间隔越短越好**。
- 4、在整个项目开发期间，业务人员和开发人员必须天天都在一起**工作**。
- 5、**鼓舞起每个人的积极性**，以个人为中心构建项目，提供所需的环境、支持与信任。
- 6、在团队内部，最有效果并且富有效率地传递信息的方法，就是**面对面的交谈**。
- 7、**可工作的软件**是首要**进度**度量标准。
- 8、敏捷过程提**可持续的开发速度**。责任人、开发者和用户应该能够保持一个长期的、恒定的开发速度。
- 9、不断地关注**优秀的技能和好的设计**会增强敏捷能力。
- 10、**简单**---使未完成的工作最大化的艺术---是根本的。
- 11、最好的构架、需求和设计出自与**自组织的团队**。
- 12、**每隔一定时间**，团队会在如何才能更有效地工作方面进行**反省**，然后相应地对自己的行为进行**调整**。

1.3 敏捷优势

普通开发模式在多数情况下总会失败，为什么我们需要敏捷开发模式？这个问题在日新月异，飞速发展的今天似乎很容易解释。尽管普通模式能够在**一个迭代周期内**表现优异，但是，在如何管理需求变化面前，瀑布模式却显得无能为力。而事实上，大多数的软件项目都具有以下一些特点：

- **需求不清**：在初始阶段，最终用户通常不能准确得知道他们需要什么样的软件。即便知道，也很少有人能准确清楚的表达出来。
- **细节不明**：对于某些项目，在一开始我们可以很好的定义其所有的功能，但是可能有很多细节只能随着项目的不断深入才能被挖掘出来。即便是我们了解了所有的细节，大多数人还是不能很好的处理这些细节，特别是在项目开发初期。
- **业务变化**：外部环境如客户的业务模式，技术进步，甚至是系统的终端用户都有可能在开发过程中不断改变。而预想或试图阻止这些改变通常都是徒劳的。
- **预期变化**：在互联网时代，许多 Web 应用程序的开发都是基于对远景客户的预期，而非当前用户的实际需求。在这种情况下，变化从开始就有，而且在系统开始应用后几乎每天都会发生。

敏捷方法处理需求和技术变化主要通过迭代过程来管理。在每一次迭代周期结束时，都应交付用户一个可用的，可部署的系统。使用并体验该系统所获得的有价值的反馈意见将按顺序，在随后的迭代周期中和其它需求变化一起在产品中实现和集成。每次迭代周期应尽可能短，以便能及时频繁地处理需求变化和用户反馈。

采用敏捷开发方式将会给企业和用户带来诸多好处：

- **精确**：它将带给用户真正需要的软件系统。普通模式通常会在产品起点与最终结果之间计划出一条直线，然后沿着直线不断往前走。然而当项目到达终点时，用户通常会发现那已经不是他们想去的地方。而敏捷方法则采用小步的方式向前走，每走完一步，都需要及时调整并为下一步确定当前的方向，直到真正的终点。
- **质量**：敏捷方法对每一次迭代周期的质量都有严格要求。一些敏捷方法如 XP 等，甚至使用测试驱动开发 (TDD: test-driven development)，即在正式开发功能代码之前，先开发该功能的测试代码。这些都对敏捷项目的整个开发周期提供了可靠的质量保证。
- **速度**：敏捷开发提倡避免较大的前期规划，认为那是一种很大的浪费。因为很多预先计划的东西都会发生改变，大规模的前期规划通常是徒劳

的。敏捷团队 只专注于开发项目中当前最需要的，最具价值的部分。这样能很快地投入开发。另外，较短的迭代周期使团队成员能迅速进入开发状态。

- 丰厚的投资回报率（ROI）。在敏捷开发过程中，最具价值的功能总是被优先开发，这样能给客户带来最大的投资回报率。

- 高效的自我管理团队。这既是采用敏捷开发的必然结果，也是推动敏捷开发不断前进的动力。敏捷开发要求团队成员必须积极主动，自我管理。在这样的团队中工作，每个团队成员的技术能力，交流，社交，表达和领导能力也都能得以提高。

2 涉及角色

2.1 产品经理（Product Owner）

产品经理（或其他类似职责）是该角色负责产品的远景规划，平衡所有利益相关者（stakeholder）的利益，确定不同的产品需求积压的优先级等。它是开发团队和客户或最终用户之间的联络点。

负责与市场、客户、高层沟通，整理和分析用户需求，并完成需求管理工作。产品经理负责的不只是把需求说出来，还应该为需求背后的商业目标负责。

完整地解析产品经理的职责，将包括：

1. 描述需求
2. 为需求排序
3. 为客户价值负责
4. 为客户群价值负责，即长期、广泛的客户价值
5. 为因满足客户价值而带来的乙方企业商业利益负责

2.2 项目经理(scrum master)

该角色与产品之间有直接或间接的利益关系，通常是客户或最终用户代表。他们负责收集编写产品需求，审查项目成果等。

2.3 开发人员

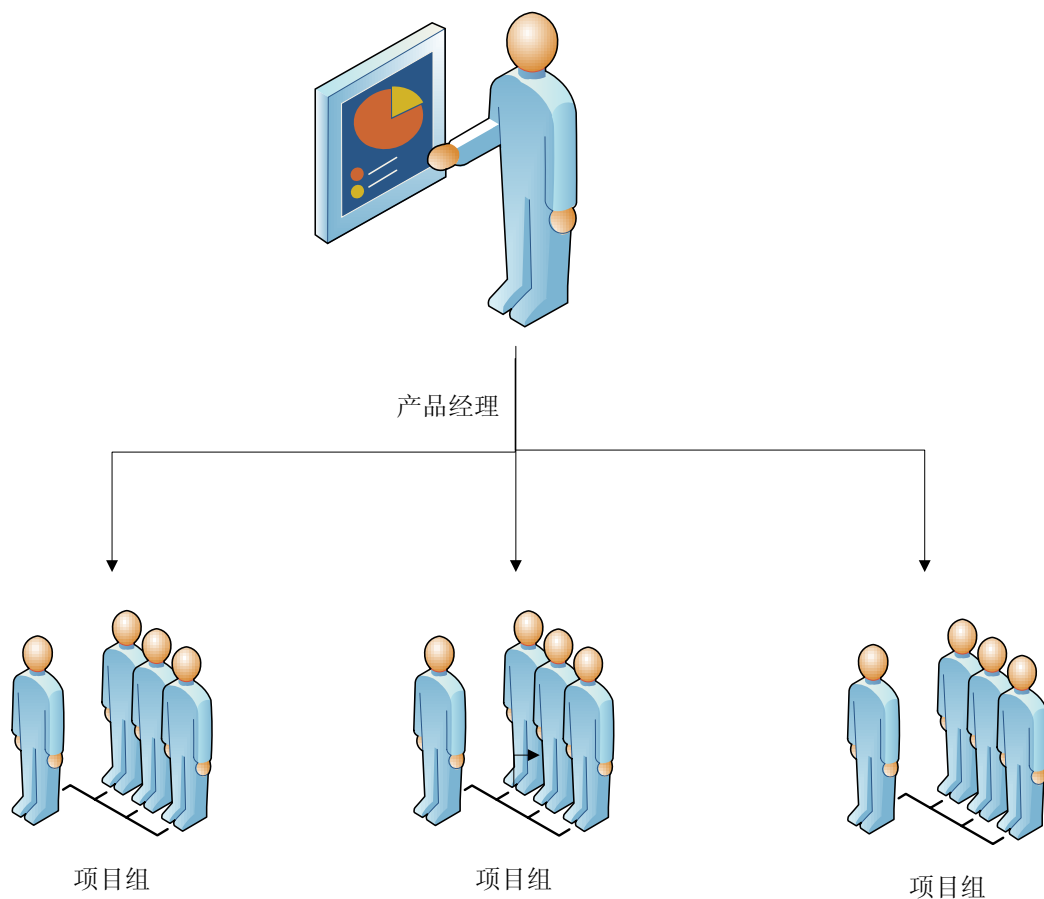
参与系统设计、代码编写、代码走查、产品测试等工作。

2.4 测试人员

参与系统设计、代码编写、代码走查、产品测试等工作。

2.5 人员配备

对于一个产品或者项目，只是要有 1 名产品经理。1 名产品经理可以对多个项目组（一般不要超过 3 个），每个项目组要有 1 名项目经理，开发人员和测试人员在 7 人左右，偏差不要超过 2 人。



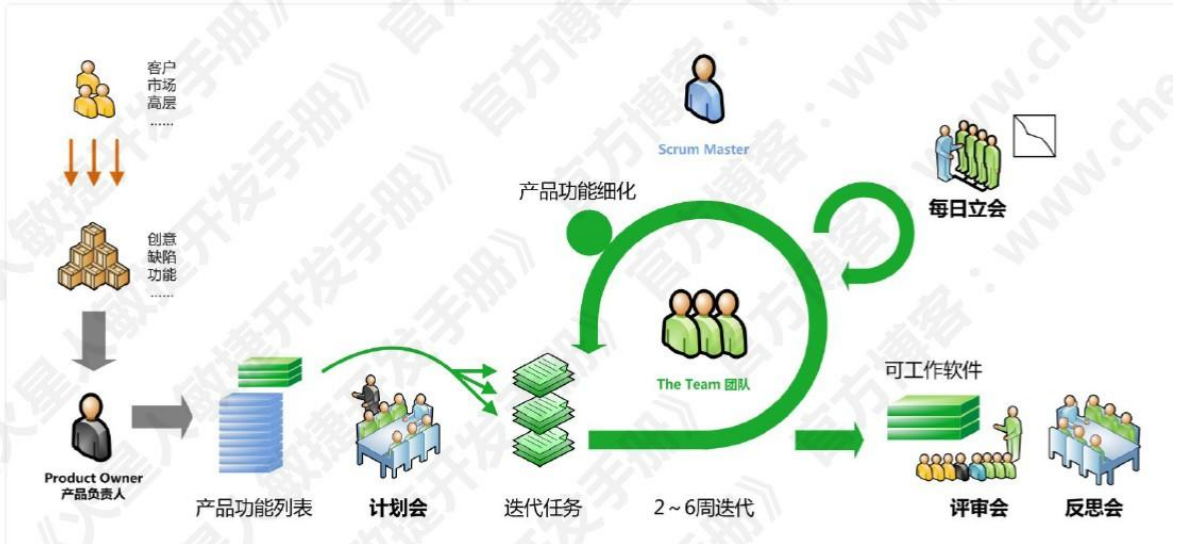
3 开发流程

产品负责人建立条目的产品待开发项，并进行优先级排序。

在迭代计划会上，产品负责人讲解本迭代要开发的条目，团队进行估算并放入下一个迭代。

团队在迭代内完成所列需求，每天都开“每日”立“会”以沟通进度和问题。

在迭代终点的迭代评审会上，团队向产品负责人等展示开发成果。



3.1 编写产品 backlog

产品 backlog 又称为用户故事，是对用户或客户有价值的功能的简短描述。

要以客户或者用户角度来用商业性或者业务性语言来描述故事，而不是用技术术语来描述。

故事卡示例：

作为一个基层治安民警，可以在保安员数据录入时，通过二代身份证直接获取保安员的基本信息，以便数据快速、准确录入。

注：保安员信息要符合公安部发布的标准和规范

3.1.1 三要素

用户故事三要素：角色，功能，价值。

按“作为一个（角色），可以（功能），以便（价值）”样式和思路写成的用户需求，就是

用户故事，即包含角色、功能、价值这三个要素。

角色切记不要总是写“作为一个用户”，而是要把用户区别对待。这样才能更好地理解他们使用什么功能，如何使用，为何使用。要把角色特化出来后，更容易理解功能的价值和风险。如果用户的“价值观”差别很大，就要分别为他们写故事，才能吸引他们使用功能，达成价值观。

功能即用户能亲自执行的操作。功能不等于价值，要理解用户操作功能的业务目的，不要随意抛出万能的功能，尤其不要涉及到用户界面。

价值是完成操作后，客户所得到的价值。“客户价值”是要从客户的角度来理解的，否则极可能跑偏。

3.1.2 编写原则

好故事具有 6 大特征：

1. 独立性：

要避免故事间的相互依赖。如果出现依赖关系，有 3 种方式进行处理：

- 用不同方式去分割故事
- 将相互依赖的故事合并成一个独立的大故事
- 在故事上加限制条件

2. 可讨论的

故事是功能简短描述，其内容不是具体的需求本身，细节在客户团队和开发团队的讨论中产生。讨论出的细节可以添加到故事卡中或另外记录中。

3. 有价值的

故事要对用户或客户有价值，二者价值要求点不同。

4. 可估算的

开发人员要能够依据故事，估算出大概的编写可用代码时间量。可能有 3 个方面造成不可估算：

- 开发人员缺少业务领域知识
- 开发人员缺少相关技术知识
- 故事太大

5. 大小合适的

故事大小是关键，太大或者太小都无助于制定计划。合适的故事大小最终取决于团队、他的容量及所用技术。

- 分割故事

史诗故事分为复合故事和复杂故事。

复合故事是有多个小故事组成的。可以通过按照数据处理方式不同进行拆分，比方数据的添加、修改、删除；也可以按照数据范围不同进行拆分，比方说员工的基本信息、工作经历、奖惩信息等。

复杂故事是具有不确定性技术因素在其中的故事。可以通过分拆成技术预研故事和业务故事，以进行分解。

- 合并故事

将多个太小的故事合并成一个故事。

6. 可测试的

只有通过测试才可以证明开发人员正确实现了此故事。

3.1.3 编写步骤

- 第 1 步：列出尽可能多的用户
- 第 2 步：识别关键用户
- 第 3 步：合并次要用户
- 第 4 步：面向关键用户，描述故事

3.1.4 故事卡的内容元素

故事卡元素包括以下内容：

序号	故事描述	重要性	估算 (故事点)	操作演示	申请人

3.1.5 确定优先级

确定优先级的标准：

- 必须有的：系统的基本功能
- 应该有的：很重要，但是短期内有替代功能
- 可以有的：如果没有时间，在此版本发布时可以不予考虑
- 这次不会有的：客户期望拥有，但同时承认需要在后期发布的版本中实现

3.1.6 用户故事不良症状的其他表现

- 镀金
- 细节太多
- 过早考虑用户界面
- 想的太远
- 很难排优先级

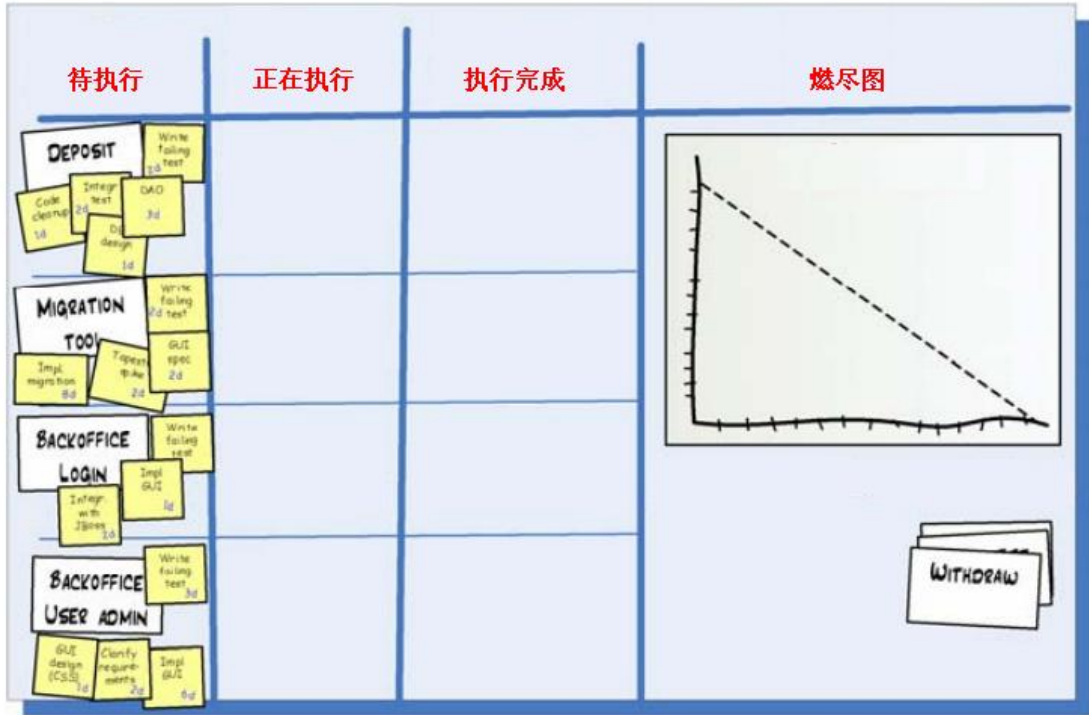
3.2 sprint backlog

Sprint backlog 意为冲刺待开发项，是从开发技术角度理解的迭代开发任务。将大的任务分解成小的任务，形成可单独交付产品，开发周期正常情况下，定为 2-6 周（具体周期根据实施情况来确定）。在无特殊原因情况下，本次迭代过程中，产品经理不能够再增加、改变、删除待开发项，应协助开发人员对产品功能细化。

3.2.1 任务看板

任务看板包括 4 块区域，分别为待执行，正在执行，执行完成和燃尽图；展现任务状态，跟踪任务开展情况。计划会议后，将本次迭代的所有任务卡片粘贴于任务看板上对应区域。

任务看板内容如下图所示



- 待执行区域放置待执行任务卡片，
- 正在执行区域放置正在执行任务卡片，
- 执行完成区域放置已经完成的任务卡片，
- 燃尽图区域是绘制燃尽图的，展示项目剩余任务所需要的时间；

3.2.2 功能分解（任务分解）

将用户故事卡片根据业务规则和开发流程分解成多个任务，任务以任务卡片的形式呈展，粘贴于任务看板上。

任务卡片内容必须包括编号、内容、价值、剩余时间、认领人。如

正面

编号：3.1
 内容：保安公司许可申请资料
 附件上传代码实现
 价值：2
 剩余时间：4
 认领人：songwj

背面

注意点：
 1、单附件上传
 2、不大于 2M

编号：3.2
内容：保安公司许可申请资料附件上传代码走查
价值：3
剩余时间：2
认领人：gaoy

编号：3.3
内容：保安公司许可申请资料附件上传测试
价值：4
剩余时间：2
认领人：gaohl

编号规则:X.Y 如：3.1 3.2 3.3

X 为用户故事价值

Y 为故事中的任务的先后关系，数字越小，优先级越高，从 1 开始；

3.3 计划会议

会议参加人员：产品经理、项目经理和团队其他成员

会议开展时机：前一迭代结束，下一迭代开始

3.3.1 会议内容

- 1、产品经理介绍本次迭代的总体目标，概述产品 backlog，确定演示时间和地点；
- 2、团队估算时间，在必要的情况下拆分 backlog 条目，产品经理在必要时修改价值评分，理清每个条目的含义，重要条目要确定如何演示；
- 3、定义任务优先级；
- 4、团队选择本次迭代的故事，确定本次迭代的待执行任务。

3.3.2 需求讲解

产品经理讲解需求，使开发人员充分理解需求，便于估算和系统研发实施。

实施过程如下：

- 1、产品经理在会前需要整理产品需求，并定义需求的优先级；
- 2、负责讲解高优先级需求；
- 3、团队成员随时发问，产品经理要予以解答；
- 4、产品经理不清楚的需求，则需要推迟该任务执行；
- 5、产品经理明确本轮迭代需要演示的功能和交付物，以及功能完成的标准。

3.3.3 定义优先级

定义方法：

- 13、 业务依赖：在业务实现在，有业务的先后顺序，必须先实现的任务，优先

级最高;

- 14、 技术依赖：在业务优先级的基础上提升任务优先级;
- 15、 项目风险：在业务优先级的基础上提升任务优先级。

3.3.4 价值估算

任务估算方法：采用扑克牌估算

估算对象：任务卡片

估算值规定为：2, 3, 4, 5, 8, 13, 20, 40

从实现难度、优先级方面给出一个分数。

具体实施步骤如下：

- 1、 从本次迭代中选取最简单的任务作为样板任务，系数定义为 2；
- 2、 以样板任务为参考，团队成员确定估算任务的价值；
- 3、 团队成员独立出暗牌，听口令一起开牌；
- 4、 数值最大者与最小者说明估算的理由并让大家一起讨论；
- 5、 讨论结束后重新出牌和开牌；
- 6、 重复上述过程，直到结果比较接近。

3.3.5 任务估算

队所有成员参与，集体完成，产品经理在估算过程要对需求进一步讲解。

任务估算注重团队能力，非个人能力，是以集体的智慧和知识对“做什么，怎么做”达成共识。

任务估算方法：采用扑克牌估算

估算对象：任务卡片

估算值规定为：1/2, 1, 2, 3, 4, 5, 8, 13, 20, 40

估算单位：小时

具体实施步骤如下：

- 7、 每人各自估算后独立出暗牌，听口令一起开牌；
- 8、 数值最大者与最小者说明估算的理由并让大家一起讨论；
- 9、 讨论结束后重新出牌和开牌；
- 10、 重复上述过程，直到结果比较接近。

3.4 任务实施

3.4.1 任务执行

任务包括但不限于以下内容：

1. 技术准备
2. 代码编写
3. 文档编写
4. 测试用例
5. 系统设计
6. 环境搭建
7. 代码走查
可以通过自动化代码检查工具
8. 技术评审
9. 测试执行
可以通过自动化测试工具进行测试。

3.4.2 配置管理

要考虑到自动集成解决方案。

- 上传
代码和文档要求每天都要上传到 SVN 服务器上，不作为版本处理。
- 提交
当代码和文档需要测试时，才做提交处理。提交时一定要标注版本号码和相关说明。

3.5 站立会议

团队所有成员必须参加，让每位团队每位成员了解各项任务的进度，便于团队成员间工作进度的沟通和协调，做好每日规划。

3.5.1 站立会议 3 问题

- 1、我从上一会议到现在做了什么？
- 2、我到下一次会议要做什么？还有多少工作？
- 3、我遇到了什么困难

3.5.2 指导原则

- 1、主题明确，不能引入无关话题；
- 2、只允许“猪”说话，“鸡”不能讲话；
- 3、所有人站立围成圈，不能围坐在桌子周围；
- 4、确保整个团队都要参加每日站立会议；
- 5、会议是团队交流，不是报告会议；
- 6、站立会议应该控制在 15 分钟之内；
- 7、不要把站立会议作为一天的开始；
- 8、要在同一时间同一地点举行。

3.5.3 实施步骤:

每一步: 回答站立 3 问题

第二步: 在任务看板中, 移动任务和重新估算任务剩余时间

第三步: 更新燃尽图

3.5.4 任务认领

任务认领是团队成员主动去领取任务卡片, 团队成员在任务认领时, 需要移动故事墙上的任务卡片, 更新作任务状态, 以便其他任务的顺序开展。

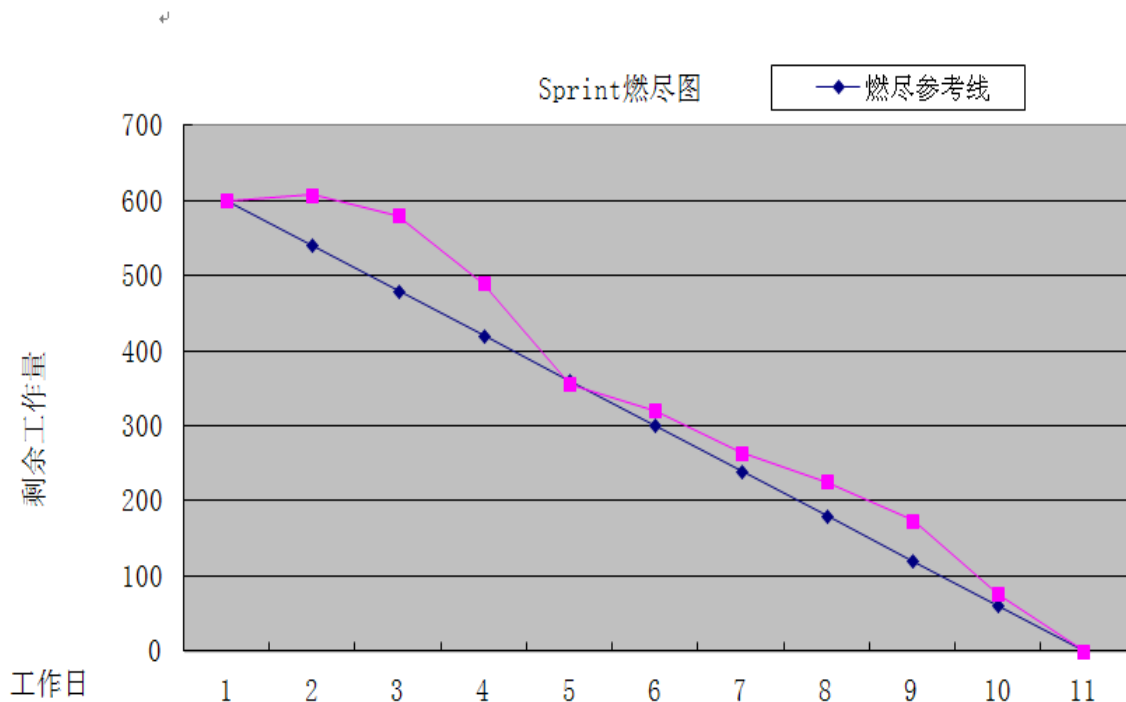
任务认领流程:

- 1、移动已完成的任务卡片到执行完成区域
- 2、领取将要开展的任务卡片
- 3、填写任务卡中认领人
- 4、更新任务卡片中的完成任务所需要的时间
- 5、将任务卡片移到正在执行区

3.5.5 燃尽图更新

燃尽图是以时间为横坐标, 本次迭代所有开发项的剩余时间为纵坐标, 反映项目完成所需要的时间, 若时间达到终点而剩余时间也趋于零, 则表明, 本次迭代顺利完成。

绘制时间: 每日站立会议后, 团队成果需要更新正在开发项的剩余时间, 更新燃尽图。
燃尽图示例如下:



3.6 演示会议

演示会议的参与对象为产品经理、团队及项目相关人员，其意义是团队向产品经理展示本轮换代的工作成果，产品经理给出评价和反馈，以功能是否能成功交付来评价任务完成情况。

3.6.1 检查 backlog

实施过程：

- 1、产品经理准备功能检查列表；
- 2、项目经理介绍本次迭代原始计划；
- 3、团队成员展示工作成果；
- 4、产品经理根据任务完成标准，实施检查；
- 5、如有任务卡片完成未达到要求，则认为未完成；
- 6、登记没有完成的任务；
- 7、确定用户故事能否交付。

检查原则

- 1、确保会议所有成员清楚迭代的目标，如有不清楚的，需要描述；
- 2、不演示没有完成的用户故事；
- 3、不花太多时间准备演示，集中精力演示可实际工作的代码；
- 4、保持演示的快节奏；
- 5、可以让观众演示系统产品；
- 6、关注重点故事。

3.6.2 收集反馈

在演示会议中，项目经理或测试人员主动的给产品经理等实际操作人员直接演示。

- 1、产品经理要对演示要给出反馈；
- 2、产品经理登记本次迭代中的问题；
- 3、产品经理将未完成的用户故事更新到产品列表中；
- 4、产品经理定义更新问题的优先级。

3.7 回顾会议

回顾会议目的：

- 1、总结经验，寻求改进；
- 2、改进团队沟通，加强团队建设；
- 3、肯定成绩，增强团队信心。

会议时间：在一次迭代结束、下一代尚未开始的时候进行。

通过如下四个问题来做好总结与改进：

- 1、上一迭代发生了什么？
- 2、我们哪些方面做得比较好？
- 3、哪些方面还可以改过？
- 4、优先级最高的问题是什么？

会议合理分析问题出现的原因，划定问题归属，指定处理人和处理时间，给出具体的改进建议。

4 迭代发布

4.1 持续集成

持续集成帮助开发人员快速发现并解决问题(代码不规范、编译失败，测试失败等)。

工具：采用 Hudson 持续集成工具，
采用 checkstyle 做代码规范检查，
采用 Junit 实施单元测试。

管理人员：配置管理员；

实施步骤：

- 1、配置管理员搭建持续集成环境，设置环境参数；
- 2、团队成员每天必须提交可执行的代码，有问题或编译不通过的代码不允许上传；
- 3、集成工具 Hudson 每 4 小时自动集成项目，集成中需要做代码检查，单元测试，编译；
- 4、集成工具 Hudson 每次集成操作后，不管集成成功与否，集成工具 Hudson 自动以邮件形式发送状态报告给项目相关人员；
- 5、项目相关人员及时处理状态报告中的问题，以便集成工具 Hudson 再次集成。

4.2 小版本发布

每个迭代或多个迭代结束后，根据演示会议的内容，发布项目可交付的产品。

具体实施过程：

- 1、项目经理整理发布内容清单，拟定发布申请，提交发布申请；
- 2、测试部根据发布内容清单，说明发布内容中遗留缺陷情况，并给出建议；
- 3、研发中心主任审批发布申请；
- 4、配置管理员以审批通过后的发布申请单为凭证，发布可交付产品。

5 实施方案

5.1 培训

5.2 准备

5.2.1 设备

5.2.2 工具

5.3 实施

5.4 改进