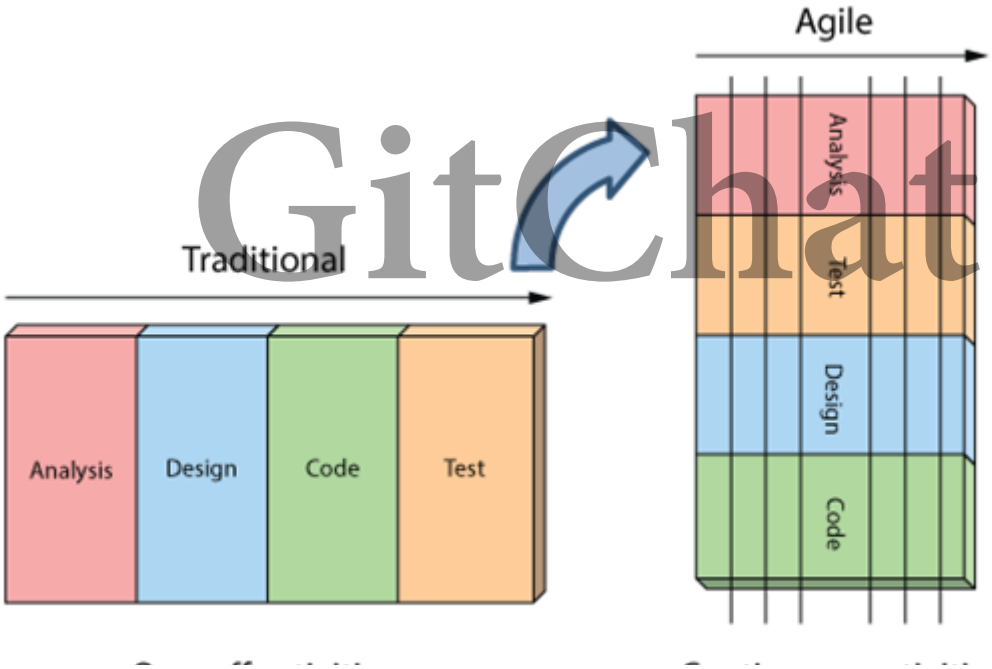
用户故事还可以这样写

用户故事很重要

用户故事很重要，是实施敏捷开发和持续交付的重要开端。

所谓**用户故事就是含有一定业务价值的端到端交付。**瀑布开发是把项目按照不同生命周期阶段**横向**拆分的过程。而敏捷开发是把项目或产品或业务目标**竖向**拆分成若干**可体验**、**可交付**的用户故事的过程。一个用户故事，应该是用一两个月甚至几个星期或更短时间就能做出来的东西，我们可以通过它更早地获取用户反馈，从而确定产品的正确性。这也符合敏捷倡导的“**小步推进，频繁验证**”的原则，是降低项目风险最有效的方法。



既然用户故事是敏捷开发的重中之重，那么如何定义好用户故事呢？诚然，这是敏捷转型的**难点**。业界对用户故事倡导的写法是“作为……（谁），我想要……（做什么），为了……（为什么）”。相对于传统需求仅仅强调做什么，这种写法指出了需求分析中的两个容易被忽略的重点——**谁**和**为什么**。“谁”就是这个用户故事的具体用户，用户不同，使用场景不同，具体需求便不同，这个用户不光是人，也可以是系统（如上下游系统）。“为什么”即业务价值，我们强调只开发有业务价值的用户故事，如果用户说不出这个为什么，即可视该用户故事没有业务价值，应该被摒弃。

但是，对于企业应用项目而言，业务干系人比较多，甚至涉及到不同国家的业务人员，不是每个角色都可以参与面对面沟通，需求分析往往需要业务代表或业务分析员（BA）来完成，他们习惯于书写传统的需求文档，要求他们一下子转换到这样的格式，跨度实在太大。敏捷实施通常从IT起步，但要完成敏捷的整体落地，业务的配合也非常重要，因此从传统到敏捷，我们需要一个平稳过渡的过程。能够把传统的需求文档或大需求拆分成用户故事，已经是非常重要的一步。一次性太多的习惯改变，步子太大了，容易扯着蛋。另外，要把一个业务需求里丰富的信息浓缩到这样短短的句子里，一来困难，二来其涵盖的信息其实比传统的需求写法更抽象，更难以厘清问题。尽管敏捷强调用户故事是说的，而不是写的，但现实中，我们还是要详细记录用户故事的具体需求，以指导开发和将来维护时追溯。

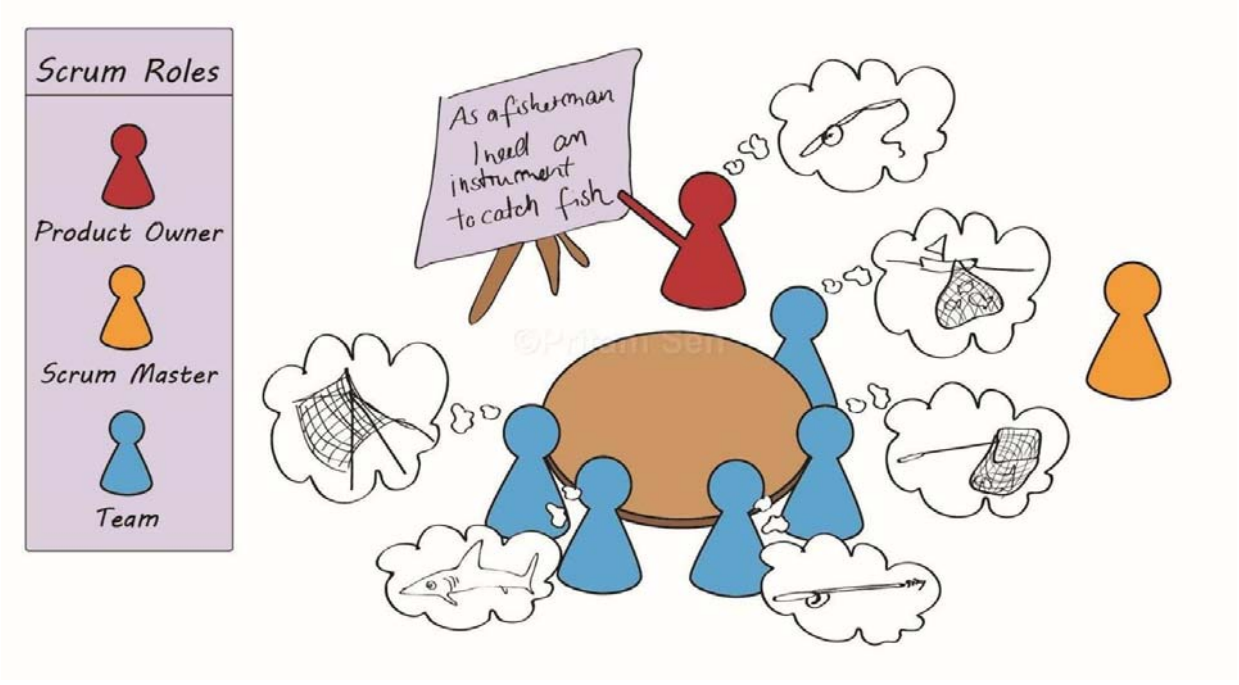
那么有没有一种更容易与传统需求衔接又能兼顾敏捷原则的方法呢？下面我将分享从实战经验中总结出来的用户故事的另一种写法。

用户故事的建议写法

我的方法是为每一个用户故事定义一个统一的模板，涵盖以下内容：

- 需求描述
- 确认理解
- 问为什么
- 验收条件
- 详细设计与实现
- 集成测试结果

复述对需求的理解并要求需求提出方确认我们的理解。由于一般需求的表述都是抽象的，同一件事情不同人的理解可以有天渊之别，复述与确认确保我们的理解是正确的。

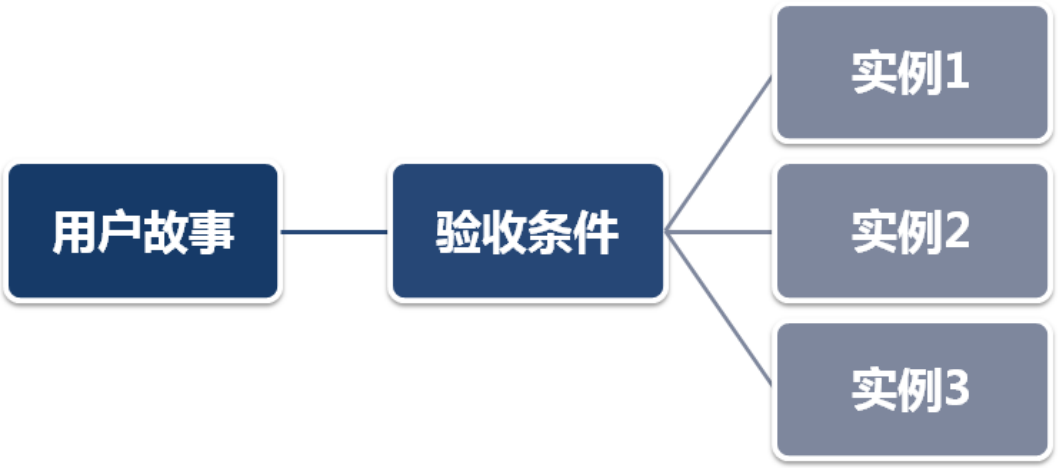


这一原则其实适用于所有的任务分配，当我们把一个任务交代给其他人或接收到一个任务，任务执行人都应该复述自己对任务的理解并得到交代人的确认，以确保对任务的理解，这是正确完成任务至关重要的一步。

问为什么

询问需求提出者为什么需要这个需求和为什么现在就需要。

这里需要技巧，直接问这个问题可能无法得到有效的答案，应该详细询问在什么场景下需要这个需求，包括谁用、什么时候和条件、要解决什么问题、什么频次、怎么发生，以甄别伪需求。有些需求其实是解决方案，需要挖掘其背后的真实用意（这里介绍一篇关于需求分析不错的文章——产品经理：需求做与不做,一念之间，里面有非常好的方法与例子）。



下面我将以一个具体例子来说明什么是含具体例子的验收条件和实例化需求。

假设我要开发一个基金交易系统，其中一个用户故事是“当投资者申购基金后，他/她将在T+2收到交易确认短信，期间如果遇到周末或假期将顺延”（T是交易日的意思，T+2即两个交易日后）。这个用户故事看似简单，但其实隐藏着不同的场景。我们可以通过一些实例来具体说明，构成其验收条件：

> 实例1：成功申购，没有假期顺延情况（快乐路径）
> 假设投资者张三（男）在2017年9月25日申购了沪深300指数基金1000元，当天该基金单位净值为2.5000，申购费率是1.5%；
> 当申购成功后；
> 那么他在2017年9月27日收到短信，内容为“张三先生，您于2017年9月25日申购了沪深300指数基金1000元，单位净值为2.5000，份额为394份，交易成功，份额已登记于您名下。”
>
> 实例2：成功申购，有假期顺延情况（快乐路径）
> 假设投资者张三（男）在2017年9月29日申购了沪深300指数基金1000元，当天该基金单位净值为2.5000，申购费率是1.5%；
> 当申购成功后；
> 那么他在2017年10月10日收到短信，内容为“张三先生，您于2017年9月29日申购了沪深300指数基金1000元，单位净值为2.5000，份额为394份，交易成功，份额已登记于您名下。”

> 实例4：申购失败（意外场景）
> 假设投资者李四（女）在2017年9月11日申购了沪深300指数基金1000元，当天该基金不可申购；
> 当交易确认后；
> 那么她在2017年9月13日收到短信，内容为“李四小姐，您于2017年9月11日申购了沪深300指数基金1000元，由于当天本基金不可申购，交易失败，申购款1000元将在2个交易日后全款退还到您申购的银行卡。”

非功能性需求也可以通过实例来说明，比如抽象的表述是**投资者提交申购申请后，立即收到短信确认**，这个“立即”是什么意思呢？具体的表述可以如下：

> 假设王五提交了申购申请；
> 当系统接收到申请后；
> 那么王五在1分钟内收到申购申请确认短信。

这个过程也是**行为驱动开发**（Behavior Driven Development, BDD）。如果验收条件全部写成“*Given…When…Then*”的Gherkin语言格式，还可以通过像Cucumber这样的自动化测试框架**实现验收测试自动化。**

延伸开来，这一原则适用于任何事情。做一件事情，**以终为始**，在一开始明确要做成什么样子，行成闭环，才能指导行动并确保结果正确。

详细设计与实现

有了以上几点，具体的需求已经厘清。这里记录满足该需求的具体设计以及实现。

集成测试结果

这里包括CI（持续集成）的结果，如果有对应的自动化验收测试，可以把在CI上测试结果的链接放在这里，以建立用户故事与自动化验收测试的连接。

有些用户故事的上线可能需要一些额外的步骤，在做用户故事开发时就应该时刻想着上线时要留意的问题，及时记录作为备忘，以确保上线成功。

这里，**确认理解**、**问为什么**以及**验收条件**是重点，作为“就绪定义”（Definition of Ready, DoR），帮助我们厘清用户故事的具体需求。

从以上的内容，大家也可以看出，其实**每个用户故事的交付就是一个小瀑布**。没错，我们可以这样理解瀑布与敏捷的差别——瀑布是以阶段作为单位，每个阶段完成所有用户故事（当然，严格地说，它们不算用户故事，这里只是通过类比方便理解）的部分任务，如需求分析、设计、编程、测试等；敏捷是以用户故事作为单位，每个用户故事完成所有阶段的任务达到可交付状态，从而实现更早地交付和持续交付。

总结

用户故事很重要，是实施敏捷开发和持续交付的重要开端。如何从传统的需求文档过渡到敏捷的用户故事是敏捷转型的难点。通过确认理解、问为什么以及确定验收条件的方法可以帮助我们更好的厘清每个用户故事的真实需求。