

敏捷开发UserStory

笔记本： ISAMV工作记录

创建时间： 2016/1/12 10:37

更新时间： 2016/1/12 11:16

作者： Barret Ren

URL： <http://wangpengfei360.iteye.com/blog/1478555>

1，基本通用原则

1. Summary: 简单写US要做什么事情
2. Acceptance Criteria: 写明这个US需要通过哪些测试，以一名测试人员的角度去考虑这个US需要满足哪些验收条件
3. Description: 讲一个故事，主角是这个US的用户（可以是真正的用户、也可以是该US服务的软件模块），可以用下面的格式书写：

As a [role/audience], I want [feature] so that [reason]. （作为一个 XX角色，我想要 XX功能，这样的话XXX）

在这个模板中有三个不同的点，分别为用户角色--who，某项操作--what，完成的目标--why（reason）

4. Estimation: 写明估计的点数，点数符合斐波那契数列：1, 1, 2, 3, 5, 8, ...，理论上该点数没有单位。但为了方便统计，在ISAMV feature开发中，我们以PW（人周）为单位。
5. US不能划分的过大，一般我们不建议超过 5点（5 PW）的US，这样不利于频繁的将代码 push进中心库，让CI做 regression test（持续集成）。

2，user story基本概念

在项目中，每个User Story是从稍多于一行的描述逐步精细到一套验收标准的。User Story是把需求分解为有优先级的，可测试的，可评估的小工件的有效机制。

User story是对软件的用户或买主有价值的功能点的描述。User stories 由以下三点组成：

- 用来制定计划和作为提醒的一段书面描述
- 用来充实story的细节的谈话
- 测试用例，用来表达和记录细节并且能够在story实现的时候对进行验证

3，user story准则

Independent：独立性

用户故事之间应该具有独立性（有点类似于UT中的test case），不应该依赖于其

他的用户故事。如果用户故事存在依赖性那么就会导致用户故事之间存在着不同的优先级，只有被依赖的用户故事完成才能继续开发依赖的用户故事。一般可以通过组合用户故事或者分割用户故事来减少用户故事间的相互依赖性。

Negotiable：可协商

用户故事不是签订的商业合同contracts，它是由客户或者PO同开发小组的成员共同协商制定的，如果最开始像商业合同一样设定了太多的条条框框和限制就无法更好的沟通及协商，也就不可能划分出既让客户满意，也能让开发认同的好的用户故事。

Valuable：有价值

用户故事必须对于最终的用户是有价值的，因此应该站在用户的角度去编写，描述的是一个一个的feature，而非一个一个的task。

Estimable：可评估

对于一个用户故事的划分需要足够的领域知识，使得在划分故事之时就能大致了解故事开发的周期，为了减少估算的不确定性，故事本身不能太大。

Small：短小

故事应该尽量的短小，当然也不是说越小越好。短小的故事可以减少划分过程中估算的误差，最好的故事是能够在一个迭代周期之内完成的。如果太大就应该考虑将其拆分为多个粒度更小的用户故事。

Testable：可测试

个人认为这一点在所有的特性中对于用户故事的重要程度最高。首先，如果一个用户故事无法进行测试，那么也就无法判断该故事是否完成。除此之外，对应的验收测试也最好是自动运行的，这样在任何时候都能触发该用户故事的检验。最后，必须在定义了验收测试通过的标准后才能认为故事划分完毕。