异地团队管理：需求协作和原型、设计的评审

2016/03/15阅读

摘要：做产品的都知道，需求是无处不在的，可能来自于用户的反馈，业务团队的反馈，用户行为的挖掘，团队内部的idea或者上级的要求，人人都是产品经理也就是人人都可以提需求吧…既然有需求就要细化需求文档，制作原型，设计界面，这是几乎每个产品团队都会做的流程，本文不涉及如何更好地写需求文档，只聊一下作为异地的产品团队怎么针对需求文档、原型和设计评审进行协作。

http://www.woshipm.com/pmd/297601.html

我所谓的异地的产品团队，可能会和远程工作有较大区别。远程工作是大部分人员都不在一起工作，就像《Remote》说的那样，异地的产品团队是说一个团队里产品开发人员和市场业务人员不在一个地方工作，这个在一些瞄准海外市场的早期产品团队里是比较常见的，我们就属于这种情况。

在异地协作的创业团队，尤其是存在时差的情况下，会遇到的最大的问题就是对于无论是需求，还是原型和设计，都存在一个异步确认的过程，不能面对面吵一架解决，那就需要合理的协作机制能方便的讨论，反馈和追踪。

**需求文档需要协作的内容**

我这里说的产品需求文档(PRD)就是对于一个产品或者某个单独的功能，PM需要通过与整个团队一起来协作完成的需求描述：

* 与用户、客户、业务人员或产品团队内部的沟通后梳理出需要解决的User Story
* 细化功能需求的前置条件和后置条件，用户流程和规则
* 完成产品原型
* 和设计师沟通输出设计稿供团队评审
* 开发前与开发团队沟通数据模型和任务分配
* 上线前和市场推广团队一起确认功能的Tracking Goal

这个过程不同的团队不尽相同，但都是需要整个团队一起来协作完成产品需求文档(PRD)的过程。

**以前遇到过的坑**

先说一下以前我自己碰到过的坑，也就是这些坑让我们不断思考如何更好的去协作：

**第一，无数个版本的需求文档**

我最早做过的一些项目几乎都是用Word来写PRD，无论是像中国电信这样大型国企，还是在海豚这样的创业团队，在做需求的过程中就会发现一个需求来回讨论确认是很常见，有时候仅仅就是对某个文案的修改，但来回沟通就要发邮件，新建一个版本，然后就会变成下面这样…



文档里面还有更多小版本修改的说明，显得很专业的样子…但是word的评审功能都在本地完成，有时候一个细小的改动也要传文件找修改，实在很out，而且以前的版本在那儿根本没有人去看过，真的想去看讨论中到底做了什么修改的话，做个Diff也是真的很困难。

**第二，确认的需求文档开发人员根本不看**

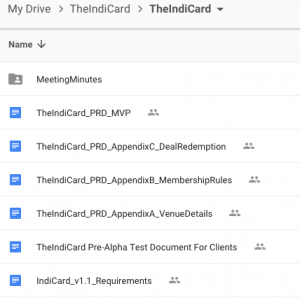
做需求文档的目的就是服务功能的开发，但长篇的需求文档真的对开发人员极不友好，对某个特定功能需求的搜索和反馈都不方便。

**第三，产品迭代中没有人去维护需求文档**

开发过程中的需求变更基本是每个项目都会碰到的，新的需求一般都当面或者邮件的形式沟通好直接就开发了，一段时间后就再也不知道这个需求是谁提出来的，什么时候加的，因为PM很难及时去更新Word的PRD，而且保证每个开发都看最新的PRD开发。

**基于文档协作平台(Google Doc)的需求协作**

碰到“无数个版本的需求文档”问题很自然的想法就是通过在线的文档协作平台来解决，协作平台选择的是Google Doc，协作主要的方式就是由PM把整个PRD放上去，邀请相关核心的业务人员(可能就是Product Owner)，市场人员和开发人员参与讨论，通过Comment的方式来协作完成PRD及相关文档的审阅，用来组织周期性会议(Sprint Meeting)。



我自己觉得这种轻量的协作方式适用的场景是和不太愿意参与产品开发过程的客户协作的时候使用，因为虽然解决了交流协作的问题，但对产品开发团队内部来说依然存在复杂的长文档，小变更难以维护的问题。

**基于项目管理平台(Redmine)的需求协作**

我们从5年前开始接触项目管理平台，开始就是需求文档做好以后通过项目管理工具(Redmine)进行任务分配和进度追踪，大家做开发的都懂，写代码大家都还挺high，但不停汇报进度就很烦躁了，我们通过[Redmine REST API](http://www.redmine.org/projects/redmine/wiki/Rest_api,)开发[githooks][5]让开发可以通过标准格式的commit message提交代码就能更新任务进度来改进流程，得到了开发人员的推崇，这5年来经过不下50个开发人员都使用的很愉快，没用什么学习时间，大部分的开发任务都能拿到较完整的代码提交上下文。

但在做项目的过程中，我们还是不断遇到上面说的需求协作的问题影响到开发效率，初创产品不断改需求经常让大家争吵某个需求到底是什么时候加进来的，在PRD上怎么找不到。我们就思考是不是可以把整个PRD融入到Redmine上，这样的话：

1. 可以用极简的Wiki语法写功能的需求文档，不用纠结格式，还能逐步加入功能相关的流程图，原型和设计稿的链接和相应的开发任务，甚至每行相关代码的提交记录。
2. 可以把特定需求的整个上下文都track下来，包括需求的描述，对应的原型和设计以及围绕这些的讨论，向下能包含所有相关开发任务和任务的完成度，向上又能追溯到关联的需求，这样团队都能尽量focus在没有讨论过的问题和需要开发的任务，而不是重复地产生“这个我们以前确认过…”这样的声音了。

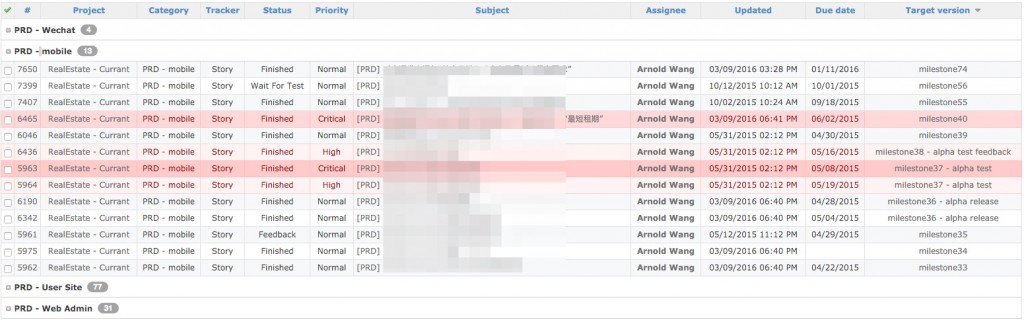
下面具体到需求协作的流程中来实际操作一下：

**一. 记录需要解决的User Story**

摘要里说过，需求是无处不在的，可能来自于用户的反馈，业务团队的反馈，用户行为的挖掘，团队内部的idea或者上级的要求，我们不可能拿到需求就开发，首先需要先记录下来，我们的流程是由PM从各处汇总后在周会上讨论，确认哪些是需要考虑放在哪个milestone来开发，哪些现在不考虑的会放在一个叫future version的虚拟版本里以后再考虑，这两类都新建User Story放到Redmine上进行track。

然后我们对进入Redmine的所有User Story按照不同客户端分为PRD-mobile，PRD-User Site，PRD-Web Admin，PRD-Wechat四类，有的团队可能习惯按照功能模块来划分分类，取决于项目规模，项目规模太大确实有必要。

最后在Redmine上我们会有自定义的Query来查看这些User Story，可以按照状态过滤出目前系统不同客户端已经完成的所有功能，对内部人员来说相当于一个完整的产品需求文档。当然也可以按照不同milestone，是不是已经完成等各种条件来过滤查看。



**二. 细化和讨论确定功能需求**

对于确定要排入milestone的User Story，PM需要按计划完成需求描述，通过协作和各方确认需求：

1. PM描述功能的用户流程，用户规则，说明前置条件和后置条件，包括补充流程图和原型来说明。
2. 为User Story添加相关的业务，设计和开发人员作为Watchers，需求的协作流程和开发任务的类似，就是当User Story的状态为Confirmed以后，大家可以通过评论来反馈来交流有问题的地方。



1. 最后，当讨论确认需求和原型之后就得到下面这样的最终的User Story，整个讨论和修改的history都被记录下来。

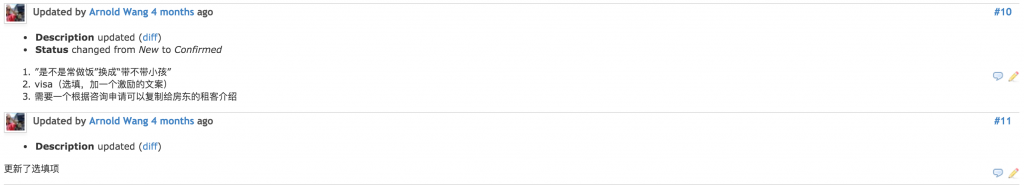


**三. 流程图和产品原型**

制作流程图或者低保真的产品原型都是为了向业务和设计团队展示功能流程，在投入大量时间做设计和开发前得到早期反馈。

**流程图**

流程图绘制工具很多，我一般用Xmind来做，和任务管理系统共同使用，做好流程后作为附件放到Redmine相应User Story的描述里，通过Redmine的评论进行讨论协作:

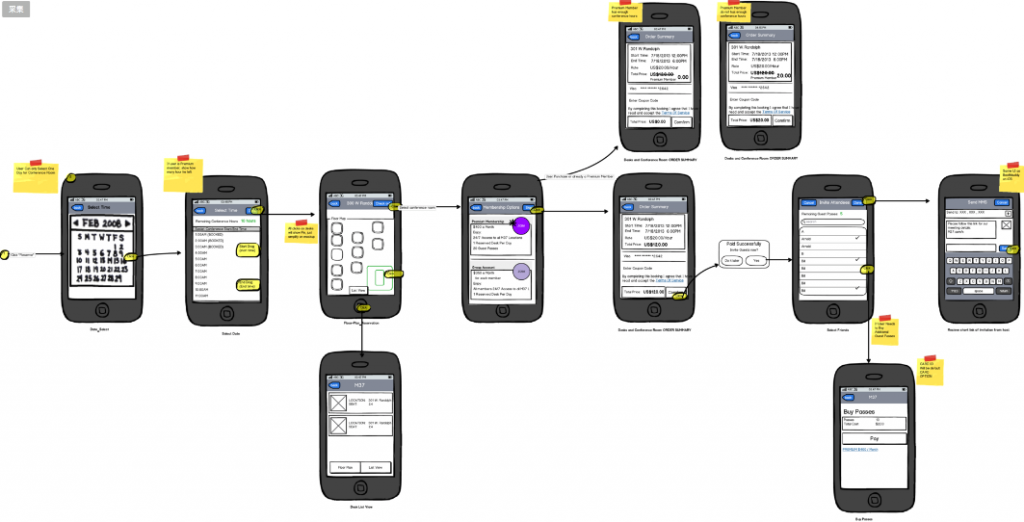


**低保真产品原型**

制作低保真产品原型是的工具有很多，我习惯用Balsamiq和Axure，场景有点差别：

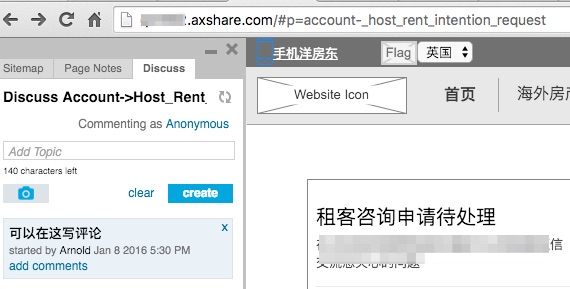
基于Balsamiq Mockup的原型和流程描述

用Balsamiq Mockup制作原型用在Mobile项目上非常合适，因为对于我这种手残的来说可以很轻松制作看起来还过得去的原型，我自己一般的用法是用把原型串成流程图带一些注解，其实以前有个工具designboard.cc可以在线这么做，不过已经不维护了，反正原型都做了其实就是把几个拼成一张流程就可以，做好流程后作为附件放到Redmine相应需求的描述里，协作同样通过Redmine完成：



基于Axure Share的原型评审和协作

Axure还是原型界的老大，Dynamic Panel和全局变量能满足大部分复杂交互描述，Master模板可以加快原型制作速度，加上有着丰富的类库可以选择，更新也很及时，现在推出Axure Share之后可以做页面级别的评审回复，还是我目前主力使用的原型工具：



**四. 基于InVision的设计稿评审**

一般来说需求阶段很大一部分时间都是花在设计师将原型输出为设计稿之后，往往大家对设计图的评审讨论是最多的，出问题概率最大的地方，之前我们也是设计完成之后放在Redmine来讨论的，不过在实践当中还是发现两个问题：

1. 设计稿的讨论经常是针对一两个比较细节的点来讨论，每次讨论都要描述到底在说哪里
2. 如果针对一个流程做一系列设计稿，用文件的方式组织就不如像原型一样来的好理解

本来考虑过也是用Axure里低保真原型替换为高保真的设计稿原型，不过因为Axure Share的评审功能还是基于页面而不是元素，对于细节丰富的设计稿来说还是不太够，后来试用了在线原型工具[inVision](http://www.invisionapp.com/,%22inVision%22)，它的评审功能和Live Share讨论比较惊艳，后来推出的Workflow也很实用，下面重点说一下我们基于inVision的设计评审和协作流程：

1. 相关的人员需要先注册参与评审，一般来说User Story的Owner，设计师，核心开发人员都需要参与进来进行评审。
2. 由设计师在相应Project上传设计图。我们所有的设计素材都在Dropbox上，可以直接连接Dropbox上传，绑定Slack进行评审人员通知，可以完全不用人工挨个通知。
3. inVision的Workflow将设计图划分为固定的几个步骤，默认的Workflow并不是完全合适我们的审核流程，我们给它的意义做了自定义：“待审核(Need Review)”，“开发中(In Progress)”，“已审核上线(Approved)”和“未来版本(On-hold)”，小团队的话可以像我们一样用这些预设的状态，企业级用户可以自定义更多状态。



1. 由业务团队和技术团队对待审核的设计稿进行评审，可以在Web点击任何元素来评论。
2. 对于Mobile版的评审，也有App可以获得更真实的测试环境，进行录屏和录音反馈。
3. 开会讨论时可以直接使用Live Share实时语音讨论，在设计稿上白板演示。这是inVision一个主要卖点，不过可惜网速问题我们用得比较少。
4. 最后把确认好的设计稿都更新在inVision上，把设计稿链接放到Redmine相应User Story的描述内。

就安利到这里，目前缺点就是付费版Workflow还不能自定义，另外国内连接性真是很堪忧，也就能翻墙用…

**五. 确认开发任务分配**

当需求和设计都确认以后就是安排开发人物了，基于项目管理工具来做需求的好处就是可以将所有的任务作为User Story的子任务，这样一个需求的完成度就能很容易的得到，这个更多是项目管理里的部分。



**六. 确认功能推广目标如何记录**

在需求文档里以前经常漏掉的一点，因为推广运营过程中需要对一些数据进行记录和分析，这个在开发之前要考虑好，保证是可以track的，免得上线之后发现一些数据无法获取，浪费推广资源。

**后记**

回过头来总结为什么团队想这些方法去提高需求协作的效率，其实就是为了提高团队的执行力，让开发人员把精力放在开发上。既然不是专业做开发协作的，那肯定有很多疏漏或者不科学的地方，现在的流程只是目前的局部最优方案，欢迎感兴趣的与我交流。

作者：王超（微信号arnold\_wangchao），创业者，5年互联网产品管理经验。本文转自我[个人的blog](http://wangchao.de/)，有其他一些我自己的感悟和实践，欢迎大家吐槽：）

本文由 @王超 原创发布于人人都是产品经理 ，未经许可，禁止转载。