PRD之道：4个撰写PRD的关键思路

2017/03/09 http://www.woshipm.com/pmd/588636.html

作者分享了4个撰写PRD的关键思路，希望这些思路，可以让你在撰写PRD时有所受用。

我看了一下互联网上面的文章，浏览量高的文章，基本上在事无巨细地讲PRD的每个环节该怎样写，甚至直接提供了PRD模版，可能的确对于产品小白来讲是比较受用的。

那么我这篇更偏“道”一些，想讲一讲做产品两年多以来，对PRD撰写的一些思考：

**一、撰写PRD应该是一个的动态获取信息的过程**

心理学上，有一个效应叫做：锚定效应。

大概是讲，人会倾向于依赖容易获得的信息，快速得出结论。

比如，我们常常说的“第一印象”；或者《思考，快与慢》中作者提到的大脑中无意识的“系统1”依赖情感、记忆和经验迅速作出判断；或者《乔布斯传》中提到乔帮主常用的判断工具“直觉”。

这是原始进化动物生存的本能——站在剑齿虎面前思考的猿猴早被灭绝了。这种决策路径效率高，省心省力。

但是在未尽量获取足够信息的情况下（或经验欠缺的情况下），快速确定一个产品需求或产品方案（下决策）是很危险的，特别是面对复杂需求、复杂项目。

做完决策后，还有会有一种沉没成本效应的心理作祟。决策者即使后来认识到错误，往往也难以说服自己改变船头的方向（脑补一下跟开发哥哥说要改需求的痛苦情景）。



（摘自《神秘的程序员》by西乔）

因此在开始撰写产品PRD之前，需尽量获取更多的信息，使用用户调研、数据分析等手段。最土又最有效的办法就是多问几位前辈的意见，兼听则明。 但是，如果在执行过程中，如果真的发现方向偏了，也要有勇气踩刹车，勇于承担错误。

**二、面向对象设计产品**

其实我大学学的是物流管理，半毛钱编程都不懂，但是曾经读过Java的介绍留下印象：Java是一门面向对象的编程语言。什么是面向对象？结合两年多的产品经验，以我粗浅的理解，主要有几个特点：抽象、封装、可复用。

**世间的万事万物都是可以被抽象的。**

比方说，你要做一顿饭。锅碗瓢盆，可以算作炊具。油、糖、盐、醋，可算作调料。青菜、肉，可算作食材。那么炊具，调料，食材，就是你做饭时需要调用的对象。

**封装是个啥意思呢？**

简单说，就是老死不相往来。炊具、调料、食材三者互不关心各自有什么内容。你把生菜换成芥蓝，我还是可以把它做成一道菜。

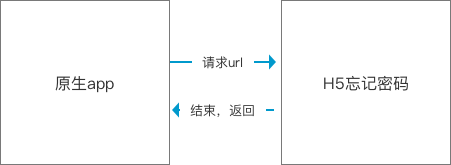
**那么可复用呢？**

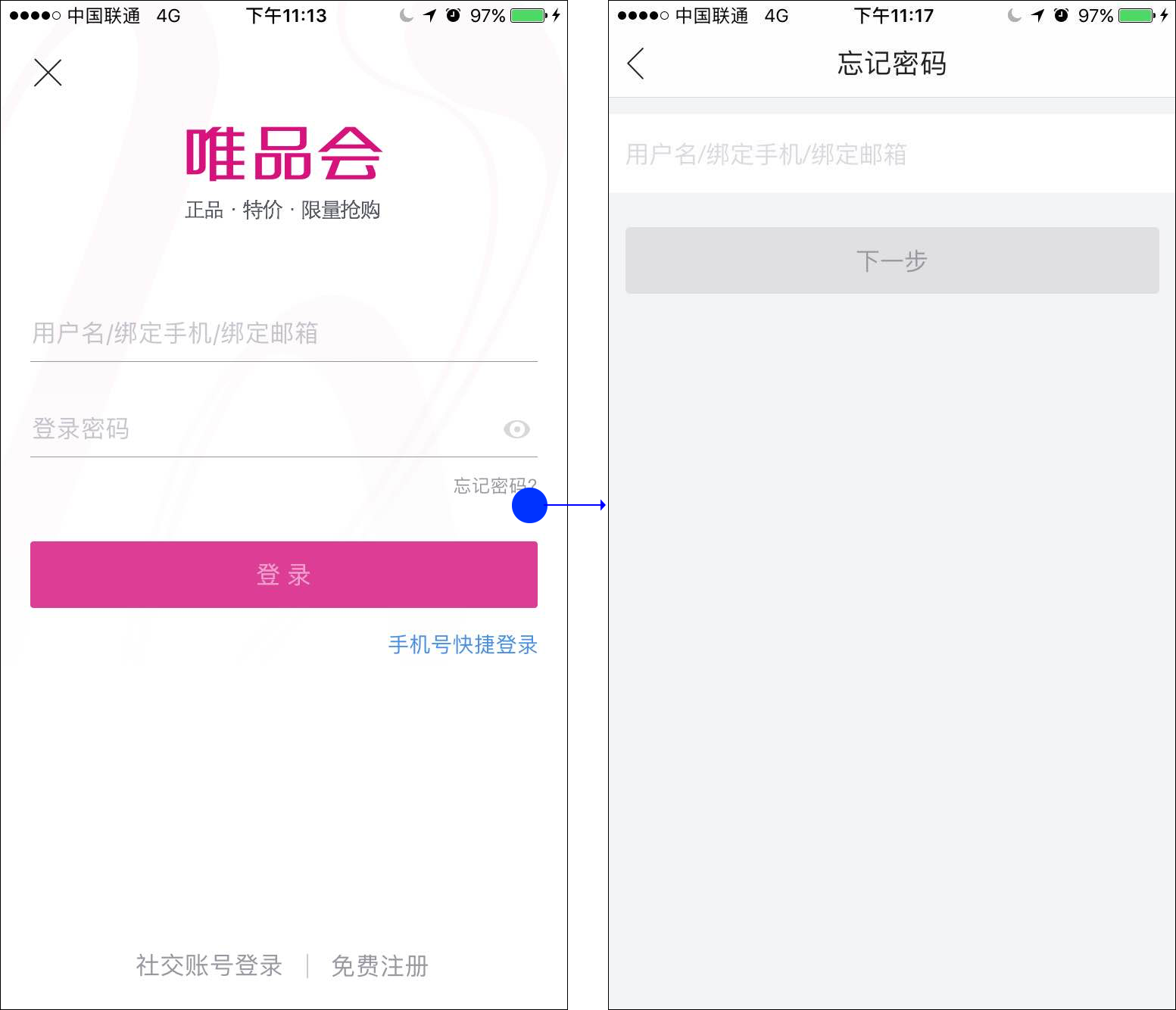
我每天都可以拿这三个对象做饭。老王来了，他也可以拿这三个对象做饭，不用自己带一套过来。

（如果理解错了，求程序员勿喷。）

我认为设计产品时，也需要面向对象设计。

举个简单的例子。以往唯品会app的忘记密码功能，iPhone、Android、iPad、WAP都需要开发一套原生界面，维护成本高，后来改成了统一的H5流程，那么原生app再也不用关心H5忘记密码的流程到底是怎样的，忘记密码的迭代也不依赖app的发版。以后如果多了一个安卓pad，也可以直接拿H5页面用。





第二个例子。以往唯品会个人中心菜单数据在中间层（服务层）配置维护，且不支持分流。但是公司内部有一个开关分流系统，可以创建开关，并灵活配置各种复杂规则。



那么，按面向对象的思维设计：

1. 对于前端。前端去请求菜单数据时，需要获取的仅仅是：哪些菜单有数据？中间层返回了数据，前端就展示出来。不返回数据的，不展示；
2. 对于中间层。如果某个菜单需要分流，那么可以在这个菜单上配置一个开关编码，按这个开关编码，带上前端请求时带的一些参数（用户信息），去请求开关分流系统。那么中间层也不必理会开关的配置复杂规则，仅仅需要知道一个结果：开或者关。开，就向前端返回某菜单的数据；关，则不返回。
3. 对于开关分流系统。仍然是干自己的事情，管理开关的创建及规则配置。

三者之间的边界清清楚楚，而且对于运营来讲，极其灵活。



再扯远一些，很多交互设计定义的组件（报错组件、弹窗组件等），其实也是面向对象思想的一种应用。

面向对象设计产品的好处也是显而易见的，最直接的便是节约开发、维护成本。另外，也能帮助产品经理提高抽象思考能力，关注问题的本质。

**三、面向PRD用户角色进行设计**

PRD其实本身也是一件产品。它的用户是：开发、测试、设计师、项目经理。好的PRD也应当符合交互基本原则：don’t make me think。

在撰写PRD的时候，需要仔细的考虑面向用户角色的需求：

1. 开发。涉及多开发团队时，不同的团队希望清楚的知道自己的开发范围边界是什么？当然，最重要的，需要怎样开发？和以往相比，改动了哪些东西？
2. 测试需要知道，用户场景有哪些，便于撰写丰富的用例，模拟真实的操作场景。
3. 设计师，需要了解哪些界面需要重新设计。
4. 项目经理需要知道项目的基本人员组成、预期上线时间、资源、风险等状况。

在撰写文档时，最好能针对不同角色，划分不同的模块进行阐述，向不同的人展示不同的重点内容。同样也是面对对象的思维。

**四、结构化、图化**

PRD应尽量使用目录、表格、流程图、交互图等结构化的表现方式。程序猿哥哥都喜欢。

曾经有一些产品经理写了一堆的文字山，然后就没有然后了。

**总结一下**

1. **撰写PRD应该是一个的动态获取信息的过程**。最好在需求早期尽量获取大量的信息，同时要保持开放的心态，接受各方的信息。
2. **面向对象设计产品**。可以显著提高产品的可扩展性、降低维护成本。也可以锻炼自己的抽象能力，把握问题核心的能力。
3. **面向PRD用户角色进行设计**。理解你的用户，善待你的用户。
4. **结构化、图化。**子曰：程序员爱看图。

此外，撰写PRD是一件需要不断实践，不断总结的事情。但愿上面的这些思路，可以让你的PRD变得更牛逼一些。共勉。