

# 产品经理整理 PRD 时，需要注意哪些点

本文作者将结合自身经验与你分享，在整理 PRD 时，产品经理需要注意的几方面。enjoy~

PRD，产品需求文档。世界上没有两片相同的树叶，也没有不做修改的需求文档，需求文档不是在去修改的路上就是正在修改 ing，开发看到我这言论估计会干死我，其实大家没有必要总是想着一版搞定需求文档，那基本是不可能的，开发也会写出 bug，测试也会遗漏用例。

为什么产品的需求文档就不能有错误呢？

之所以需求文档的修改会引起开发和测试的极大反感，主要是产品经理的需求文档处在一个产品开发的源头，一开始源头就出现了问题，后面的流程能顺利实现目标吗？可想而知源头出现了错误，会对后面加班工作的开发兄弟造成多大的打击。所以，作为产品经理的我们必须尽可能地将 PRD 整理的详细易懂，不要遗漏，不要误解，不用意淫。

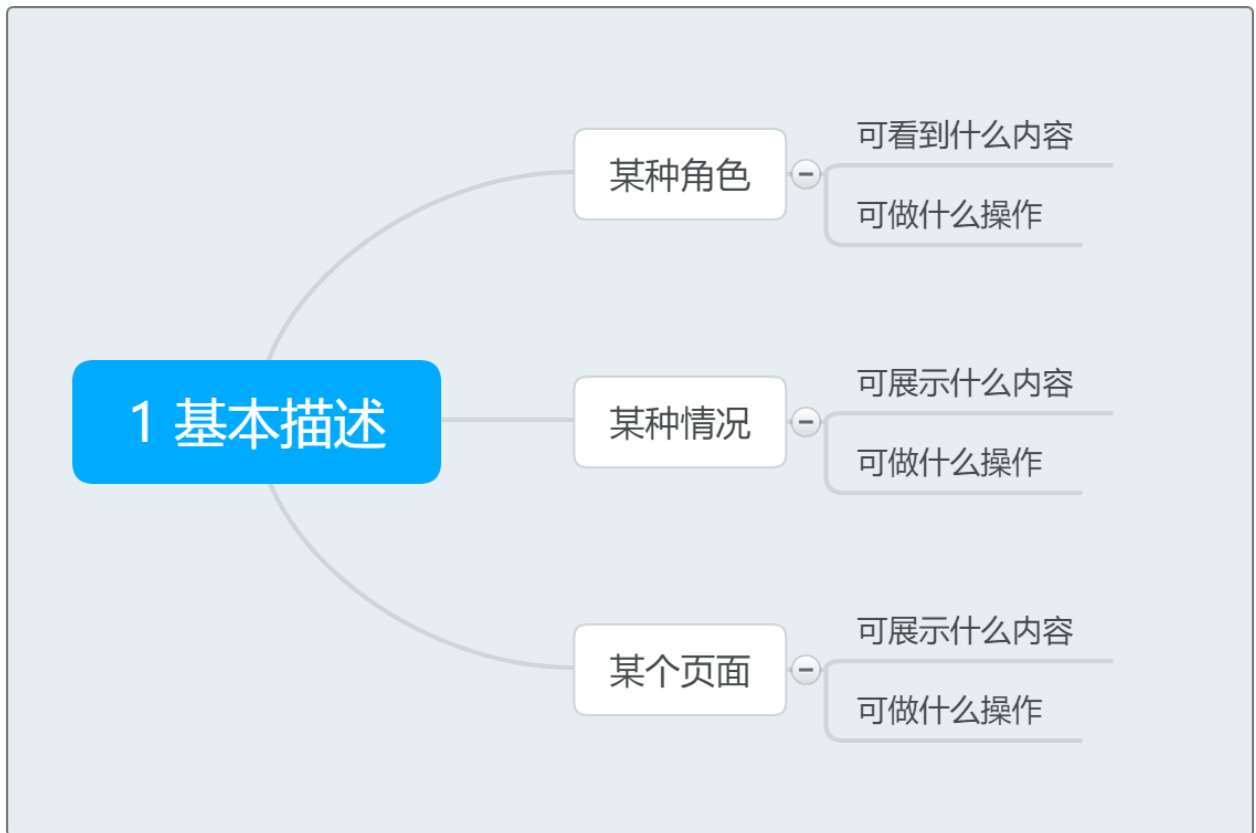
前段时间同样是在人人都是产品经理社区上看到一篇关于如何撰写 PRD 的文章，写的很好但是个人觉得还不是很全面，以下 PRD 写作要点都是个人躺过的坑，被开发被测试屌过的路，希望可以给大家一个参考，帮助大家更好的完成产品需求文档，不免有遗漏，大家可以留言补充。

## 一、需求点的基本描述维度

### 角色

往往一个按钮一个界面，不同的角色所能进行的操作，所能看到的内容都不尽相同。

例如一个组团参加活动的需求，其中角色可分为队长、成员、非团队成员、组织者，队长可以做什么？队长在界面上可以看到什么？成员又可以做什么？成员又可以看到什么……这些都需要界限清楚，这是一个需求点的“定位”，“定位”偏了，也许后面写了乌压压的一片，到头来全白搭。



## 二、需求中常见但又容易遗漏、描述不全的 9 大类型需求点

### 1、退出机制

退出分为 4 类：退出登录、退至后台运行、杀死程序、关机；特别是数据型的产品，当发生这 4 类退出时分别对应的数据同步、数据中断都该如何进行，你可以简单处理一视同仁，但是必须了解有这几类，不认测试问起就懵逼了。

### 2、显示机制

显示其实可以按照正常/异常来分类，也可以按照静态/动态来分类；我这里按照静态和动态进行分类，一个界面或一个按钮的展示必须从两方面来描述，静态：展示形态、内容、格式、数量等，动态：初始状态的展示、触发时的状态的展示、触发后的状态的展示、触发成功/失败的展示等。

例如微信底部的四个按钮，分为两种静态显示：选中时填充色为绿色，未选中时无填充色；但是介于选中和未被选中之间是什么颜色呢？这个可以定义为一个过程色，我们可以在微信一级界面左右滑动页面观察底部按钮的变化，当即将被选中时，按钮是从轮廓慢慢变绿并且绿色逐渐加深，然后是慢慢填充绿色；当即将被取消选中时，则按钮颜色变化正好相反。



### 3、排序机制

凡是涉及到列表、记录的均需要考虑排序，不管是按照生成时间倒序还是正序，至少需要确定一个顺序，而不是让开发进行到这个页面时再来问你，你这个时候其实就是失去了主动权，因为是你的遗漏，另外在仓促之下做出的决定很容易导致体验不好或者其他考虑不周的问题。

### 4、刷新机制

如果页面展示的内容是随着时间不断变化的，则必然要考虑到刷新，其中刷新就涉及到一般的刷新前，刷新后，刷新中的展示了，也会有异常刷新失败的考虑，另外也需要考虑是自动刷新还是手动刷新。

## 5、加载机制

其实加载机制和刷新机制经常在一块，有刷新一般会有加载。

## 6、缓存机制

有人觉得像这种偏点技术的需求应该由开发来决定，这里就不做争论了，但是作为一个产品也必须了解，为什么要做缓存？做缓存会带来什么影响？做缓存大多时候是为了让用户体验更流畅，不至于让用户长期停留在一处等待转菊花，如果缓存反而让系统频繁卡机，这时就得重新考虑考虑了。



（右击在新标签页中打开，即可查看大图）

## 7、推送机制

目前推送越来越多，主要是想增加用户的粘性，调动用户使用 APP 的积极性，所有推送不能太多导致用户反感，也不能过少造成用户的遗忘，这些都是产品经理在撰写需求文档之外需要考虑的；需求文档中对推送的描述必须含一下四个方面：触发推送的时机，推送内

容，推送对象，点击推送后的跳转；部分推送还具有时效性，时机一过，也许本该跳转的目的页面已不存在了，这个时候就地具体问题具体分析了。

## 8、中断机制

也许大部分产品只会考虑到网络中断的情况，那是因为在大部分的场景和页面中只考虑网络中断的情况即可，但是不同的业务不同的页面则可能对应不同的中断机制

其中这里总结了 4 中类型的中断：退出登录、来电话、程序进入后台运行、网络中断；列如在 APP 中视频通话时，突然有电话拨入时，是直接中断视频？还是视频不中断但是跳出 APP 进入接听电话界面？…也许考虑的不对或者不合逻辑，但是有这种疑惑时必须提出来可以和开发讨论，否则遗漏了就又懵逼了。

## 9、删除机制

主要是注意下是物理删除还是逻辑删除；物理删除：直接从数据库层面彻底删除，删除的数据无法找回；逻辑删除：仅仅是逻辑和界面展示上删除，数据库中还存在有该数据，必要时可以恢复。

## 三、控件的具体描述

其实在上面的九大机制中或多或少已经涉及到了具体一个控件该如何描述，以下脑图也很简单，就不多赘述了。



## 四、千思万虑

### 1、异常操作

脑图中之所有没有展示，就是因为异常操作是无法预知的，只能尽量汇总全，像测试那种变态的一个按钮非要以掩耳盗铃之势连击两下三下……这种也属于异常操作，进入页面的瞬间点击返回，这也算异常操作，有时候开发也讨厌测试也是有道理的。

### 2、网络情况

估计说到网络情况，很多人想到了有网络的情况和没有网络的情况，但是现在的网络按照网络速度可分为 2G/3G/4G 三种类型啊，另外还分为 wifi 和非 wifi 的情况，考虑到用户体验，不同的界面也许在不同的网络类型下就该展示不同的内容或不同的提示：

像今日头条 APP 中，当你在非 wifi 情况下观看视频时是可以直接观看的，但是当你在非 wif 的情况下查看视频时，系统就会提示你播放该视频需要耗费多少流量，你可以选择继续观看还是返回，或者其他服务。





### 3、账号相关

如果一个需求需要考虑与账号的关联性，则可以问 4 个问题：登录情况下是什么情况？非登录情况下是什么情况？换手机后登录又会有什么情况？登录手机端时是否与 PC 端互斥？

我之所以会将这个点列出来，主要是之前在我做的一个运动健康的需求中有涉及，特别是换手机登录这种情况下运动等时时变化的数据同步的问题很头痛，最后我们是忽略了这种场景，但是作为一个产品经理你必须知道你的文档涉及到哪些内容，哪些场景，同时你还必须知道哪些场景哪些情况是可以不用过多考虑的，综合开发成本、时间周期以及用户的使用频率考虑。



## 4、数据相关

数据的正常和异常展示的考虑就不多说了，数据何时同步？何时刷新？不同阶段的数据存储在那个端：服务端、客户端？这些看起来都是很技术的问题，但是产品也必须了解，其实产品在这些需求方面可以不出任何方案，也没有出方案的权力，因为对这些需求你多半没有开发清楚，但是你可以根据开发给出的方案进行选择，选择“看上去”较好的方案，因为开发有时候也不知道哪个好哪个坏，所以这个时候这个责任就得你来担起了，谁叫你是产品 dog，这个产品的 owner 呢！

## 5、版本相关

主要考虑版本升级后遗留老数据是否会收到影响，该如何做迁移，该如何取舍的问题。

## 6、平台相关

对那些大公司来说可能一个平台 N 个产品经理，但是对初创型公司来讲很多时候一个产品经理负责的需求会涉及到 N 个平台，所以这个时候就的对每个平台都进行考虑啦，做到一套代码可以试用于多个平台。

其他部分可能遇到的比较少，但是基于 PRD 考虑的越全面越好的格调，了解下也是不错的，如果需要完整的脑图或者有补充的可以给我留言。

岁月是把猪饲料，但是我们可以选择有营养的吃，不然怎么去寻找风口，练习飞翔！

作者：安迪(微信公众号：PM 大白)，健康管理产品经理，3 年互联网产品设计经验，曾主导患者 APP，医生 APP 以及医院机构健康管理后台的设计。

本文由 @安迪 原创发布于人人都是产品经理。未经许可，禁止转载。

题图来自 Pexels，基于 CC0 协议