**第1章写给-1到3岁的产品经理/11**

* 1. **为什么要做产品经理/13**

用好产品改变世界

* 1. **我们到底是不是产品经理/20**

产品是什么：产品就是用来解决某个问题的东西

全世界第一位产品经理：麦吉利负责宝洁的Camay香皂

产品经理的出现是为了适应公司发展的需要。随着企业越来越大，产品越来越多，原来按职能划分部门的组织结构已经无法适应，所以出现了产品管理的矩阵型组织

在如今互联网、软件行业，产品经理的概念已经所有变化了。侧重产品从无到有、从有到优，而不是在有产品之后去管理、推广、营销产品。



管理的能力，其实就是“在资源不足的情况下把事情做成”的能力：

**第一，信息不足以决策。**时间有限，能力有限，每次决策前不可能掌握所有信息，

做决定时总是很头疼，我估计是“拍脑袋”拍得太多的原因。

**第二，时间不足以安排周密的计划。**总是接到 3 个月、1 个月，甚至 1 个礼拜完成某项目的命令，每次都让我们张大嘴巴说不出话来，应承下来后如何计划？不过一次又一次的实践表明，办法总比困难多。

**第三，人员不足以支持工作强度和难度。**不但时间不足，人员也不足，就算数量足，能力够不够？能力够了，团队士气高不高？哪个公司不加班，又有多少公司有加班工资？但还得完成任务，难不难？难！

**第四，资金不足以自由调配。**俗话说钱要花在刀刃上，买机器要钱，招人要钱，产品推广要钱，而花这些钱的前提是公司还得赚钱，每一分钱都恨不得掰成两半用。

* 1. **我真的想做，怎么入行/29**

**确定自己真的想做产品。**这一点非常关键！**做产品的大前提是要喜欢做产品**，不然将来你痛苦，团队痛苦，用户也痛苦。网络上那么多好玩的应用，是很有意思，通常自己想做产品的同学都会去大量尝试、注册各种各样的产品，去用，去玩，去想……

需要进一步明确的是：你说自己喜欢产品，**到底是喜欢做用户，还是喜欢做产品经理**？

对于应届生来说，最看重的是有没有激情，是否够机灵、好学，逻辑思维是否清晰，沟通表达是否顺畅等。其他的都会次要一些，比如对行业的熟悉。

第2章一个需求的奋斗史/41

**2.1 从用户中来到用户中去/44**

2.1.1 用户是需求之源/44

“用户是需求之源”，我们要拥有“以用户为中心的思想”，不断“体会真正的用户。常用的需求采集方法，有“数据分析”、“调查问卷”、“用户访谈”等。

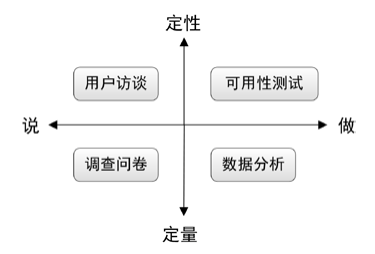
2.1.2你真的了解用户么?/49

用户说了很多需求，产品经理要“听用户的但不要照着做”，必须“明确我们存在的价值”是“把用户需求转化为产品需求”，这一过程即需求分析过程。产品经理要通过“给需求做一次 DNA 检测”，来“确定需求的基本属性”、“分析需求的商业价值”、“初评需求的实现难度”，从而计算出需求的“性价比”。

资源总是有限的，所以我们只能做那些性价比高的事情。有意识的“尽可能多地放弃”。

**2.2 需求采集的大生产运动/55**

用户研究，或者说需求采集的过程，都会有如下几步：明确目标、选择采集方法、制定采集计划、执行采集、资料整理，然后进入下一步的需求分析阶段



2.2.1 定性地说:用户访谈/56

用户访谈经常出现如下问题：

**第一，“说”和“做”不一致的问题。**

索尼问一些用户喜欢黄色的还是黑色的游戏机，结果发现说喜欢黄色的用户比较多。之后送游戏机，而同样一批用户选择黑色的游戏机带回家的更多。 1、之前没想过，编的看似合理的理由；2、故意说访谈者希望听到的答案

**第二，样本少，以偏概全的问题。要尽量随机**

比如为了成本考虑，我们上门访谈的时候只找了本市的用户，这样很可能得出一些与地域有关的错误推论；又如电话访谈时，为了提高联系成功率，我们优先拨打留了手机的用户，而留手机很可能代表这批用户忠诚度已经比较高；再如邀约用户来公司访谈，“愿意来的用户”，就已经和全体用户有差异了……

**第三，用户过于强势，把我们往沟里带。**

**第四，我们过于强势，把用户往沟里带。**

还有就是用户大会

2.2.2定量地说:调查问卷/60

用户访谈的提纲通常是开放式问题；而调查问卷通常封闭式问题比较多，适合大用户量的信息收集，但不够深入。

作答时间最好不要超过 10 分钟，否则很多人看一眼就被吓跑了。**开篇一般放一些简单的不需要思考的问题**；**很想知道的内容，需要思考的，较敏感的问题一般放在中间**；而有关**被访者个人信息的题目一般放在问卷的最后**，以免应答者在回答这些问题时有所顾忌，进而影响其他答案。

《长尾理论》里说到“沉默的大多数”，那么站出来的总是很少数，而且往往是非典型的用户，不能保证其代表了目标用户的想法；而“骑墙的大多数”说的是，大多数人是没有明确观点的，尤其在网络这样一个不用负责任的环境下，所以常见的情况就是开始表态的那几个人的观点引导了群体的观点，随机的初始值决定了结果，这个时候你只有单独和跟风者交流，才会发现他根本不是那么想的。

调查问卷的客观性、多份问卷之间的独立性，可以有效避免上述问题，但其容易出现的问题在于：

第一，样本的偏差，即样本与想了解的目标用户群体出现偏差。

调查问卷的样本选择，就有几个注意点：尽可能覆盖目标群体中各种类型的用户，比如性别、年龄段、行业、收入等，要保证各种类型用户的样本比例接近全体的比例，**比如目标用户中男女比例为 7：3，那么我们的样本也应该保持这个比例**。

所以，在类似情况下得出结论的时候，大家最好把这些潜在的筛选条件标明，让报告的读者**知道数据获得的方法与来源**，同时如果我们是报告的读者，也要一直**带着问号去阅读里面的数据**。

还有一个小技巧，就是可以**把目标群体的特征也定义成一系列问题**，放入问卷中，待我们回收问卷以后，就可以通过这些问题评估作答者是否能代表目标群体了。**如果发现了偏差，我们也可以从回收的问卷中再筛选出一个接近目标群体的子集**来分析。

第二，样本过少的问题。

样本量过少时，采用百分比来分析是没有意义的。这是很多新人会犯的错误，比如只问了 5 个人，3 个人选 A，就在报告中说有 60%的用户选 A，这是很不严谨的。因为如果换 5 个人再做一次，很可能就是 40%了，而这样的数字百分比要具备稳定性才有价值。所以，此时只能说“问了 5 个用户，有 3 个用户选 A”。抛开严谨的统计理论不谈，**要给出百分比答案的话，至少得有大约 100 份的答案**。

第三，问卷内容的细节问题。

首先，问题表述应**无引导性**，这点和用户访谈类似。比如，不要问“你喜欢某个产品吗”，这时用户可能会考虑到提问者的情感而回答“是”，正确的问法是“**你是否喜欢某个产品？**”

**答案的顺序，可能产生“顺序偏差”或“位置偏差”**，即被调查者选择的答案可能与该答案的排列位置有关。有研究表明，对陈述性选项被调查者趋向于选第一个或最后一个答案，特别是第一个答案；而对一组数字，如价格和打分，则趋向于取中间位置。为了减少顺序偏差，可以准备几种形式的问卷，每种形式的问卷选项排列的顺序都不同。

2.2.3定性地做:可用性测试/64

常见问题与对策

第一，如果可用性测试做得太晚（往往在产品将要上线的时候），这时发现问题也于事无补了。

其实，可用性测试在产品的各个阶段都可以做。在尚无任何成型的产品时，可以拿竞争对手的产品给用户做；在产品只有纸面原型的时候，可以拿着手绘的产品，加上纸笔给用户做；在产品只有页面 Demo 的时候，可以拿 Demo 给用户做；更多的时候，在产品已经可以运行以后，可以拿真实的产品给用户做。不同阶段不同做法，从中都能发现相应的问题。

第二，总觉得可用性测试很专业，所以干脆不做。

可用性测试，听着很专业，但收益又无法量化，所以对很多老板来说，不太愿意在这个上面投入资源，经常因为项目时间过紧被略过。我们知道，可用性测试通常来说做 5 个左右的用户才可以发现大部分的共性问题，前前后后的准备也耗时不少，但**只做一个用户，并且简化步骤，也比不做要好**。

**第三，明确是测试产品，而不是测试用户。**

可用性测试要邀请用户来做测试人员，我们在开始之前，应当明确地告诉用户，这个测试的目的是发现软件产品中的问题，**而不是要测试用户是否有能力来很好地使用软件**。所以，不要让用户听到“可用性测试”的术语，而是说“**来试用一下我们的新产品，提点意见**”。清楚地说明这一点将有助于减轻用户的压力，使得他们能像在真实环境一样来使用软件。

**第四，测试过程中，组织者该做的和不该做的。**

刚开始的时候，可以告知用户大概持续的时间，要做哪些事情，让用户心中有数，轻松愉快地完成任务。

可用性测试中，我们可以要求用户在使用产品的过程中采用一种名为“发声思维”的方法，即在使用产品的同时说出自己的思考过程，比如为了完成某个任务，用户想先做什么，后做什么，为什么要做某个动作，等等。

做测试的过程中千万不要有任何的引导与暗示，而只是观察和记录，因为任何引导都可能使得原本可以发现的问题无法暴露。用户行为和预想的不一样时，可以提问，实在进行不下去的时候，给予提示。记住，一切的错都是产品和我们的错，用户绝对没有错。如果真觉得用户错了，那也是你找错人了，不是这个人错了。

软件升级改版的方法：

比如先**从部分次级页面改起**，像“我的淘宝”历时多年的改版；

再如**新旧版本并存一段时间**，并允许用户自由选择，比如 2007 年的雅虎邮箱和新

浪邮箱改版；

三如**小面积试验**，选择一小批测试用户放出新版本，监测效果，做用户调研，比

如 Gmail 在发布某些新功能的时候；

四如**傍上一个用户已经习惯的风格**，比如网店版的前身“高级店铺”升级到网店

版 1.0 的时候，讨论了很多方案，最终还是决定模仿“我的淘宝”的页面风格。

总之，对于改版，对于升级，我们要把“暴力革命”变成温柔和谐的“和平演变”。

2.2.4定量地做:数据分析/68

虽然绝大多数情况下的经验证明，只要在用户的选择上没犯什么低级错误，他们是“具有代表性的”，或者说接受这种假设是一种性价比很高的廉价解决方案。不过，我们还有数据分析，一种定量的研究方法，数据来说话，看看用户到底是怎么做的，不论是考察目标用户全体、还是采样，都完全可控，所谓“According to the data”是最难被驳倒的。

最关键的就是对结果的解读，通常**数据分析只能发现一些现象和问题，并不能了解原因**，所以分析完成后通常会伴随着一些用户访谈，听听用户怎么解释。

用户“怎么说”和“怎么做”不同，甚至经常有矛盾，有**时候用户的行为比语言更能反映出他的真实需求**。比如用户说在搜索的时候应该加一个“按交易额搜索”的条件，也许只是他某次特殊的需要使然，但如果我们听他的做了这个功能，之后通过用户行为的数据分析发现，只有 1/10000 的人用过，那就表明我们被用户的说法骗了，但数据永远不会骗我们。

数据分析的问题：第一，过于学术，沉迷于“科学研究”。科学研究通常只注重“性价比”的性，只要结果好，往往不在乎投入，因为相对而言科研的结果不是为了马上应用，而是为了证明实力。但实际生产环境就更注重综合的性价比了，甚至给出“A>B”的结论时也用不着做“显著性检验”，一切的一切需要的只是一种感觉，一种对数据的敏感，对商业的敏感。

第二，虽然数据不会主动骗人，但我们经常无意或有意地误读数据。

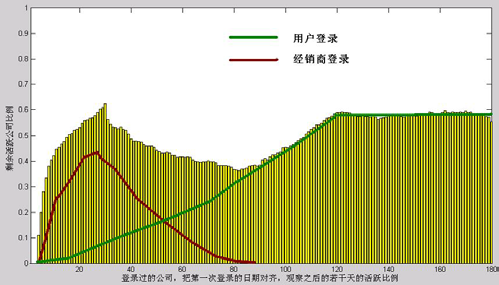
1学习统计学的知识，努力提高自己的水平。

2主动地误读数据，是比较有趣的现象。在提取数据之前，我们心中通常已经有一些结论了，无非是想验证它，而抱着这点思想，就总能找到数据来证明自己已有的想法。

第三，平时不烧香，临时抱佛脚。

我们应该在产品设计的时候就把数据分析的需求加进去，比如记录每个按钮的点击次数、统计每个用户的登录频率等，这也算一种典型的非功能需求，这样做对产品的可持续发展非常必要。

一个例子：数据分析是如何转化为商业价值的。整体的思路是：在对产品足够熟悉的基础上，先做出方向性的假设，再提取相应的数据并分析，得到一些现象，最好是之前没发现的现象，然后尝试解释，接下来做用户调研修正解释，最终指导产品发展方向。



2.2.5需求采集人人有责/73

单项需求卡片，理念就是：产品的需求工作不只是需求分析人员的事，而是涉及

产品的每个干系人的义务，至少得参与“采集”的过程。



几个有特点的需求采集方法：

现场调查；

AB 测试：基于大用户量比较合适，比如有一个按钮不知道是放页面的左边好，还是右边好，而我们有 10 万用户，那就先随机挑选少量的用户发布这个按钮，1000 人放左边，另外 1000 人放右边，然后过一段时间分析结果，再决定剩下的 98%用户该怎么办。很明显，这也是让用户直接参与了设计，这样低成本的方法让很多传统行业的同学羡慕不已；

日记研究；

卡片分类法；

自己提需求；

需求采集的各种新方法层出不穷。和学习任何领域的知识一样，建议大家在了解知识框架后，坚持“需求驱动学习”

**2.3 听用户的但不要照着做/78**

有的用户很“危险”，在提意见的同时还说你们应该做成什么样子，这时候产品经理一定要头脑清醒了，用户提的解决方案往往是站在自己的立场上的考虑的。比如对“快递单打印”的功能，用户提出要添加一个他经常用的小快递公司的快递单模板，而我们会发现，这家快递公司可能只是一个区域性的快递，最终的解决方案是做了一个“自定义快递单”的功能。

有时候，用户给出的做法存在明显的逻辑矛盾，就算他给出的解决方案合理，也要再深挖用户内心根本的需求，比如用户描述“新建非支付宝交易订单的时候必须要选择用户不合理，希望能自己填写客户”。这里更深层的需求就可能是他需要把线下客户也管理起来，所以我们或许更应该做一个新增线下客户的功能，而不是在新建非支付宝交易的时候让用户自己填写客户姓名。

我们是产品经理、产品设计师，最终怎么做应该由我们决定。

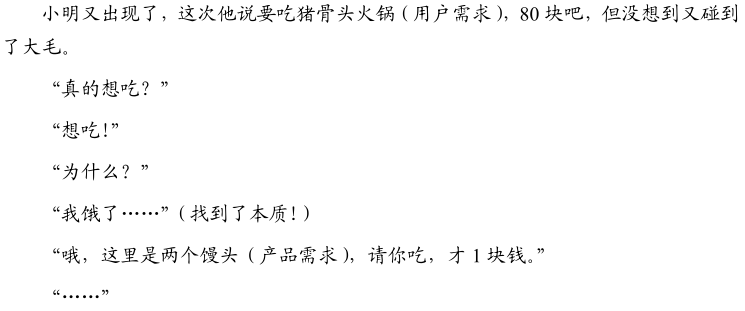
用户跟福特要一匹更快的马，福特却给了用户一辆车。

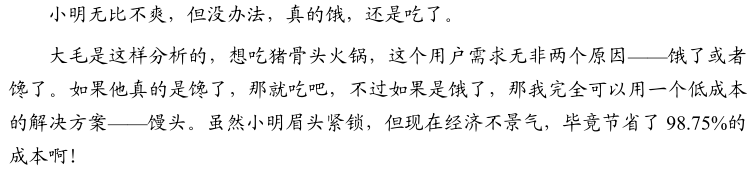
2.3.1明确我们存在的价值/78

用户需求：用户自以为的需求，并且经常表达为用户的解决方案。

产品需求：经过我们的分析，找到的真实需求，并且表达为产品的解决方案。

需求分析：从用户提出的需求出发，找到用户内心真正的渴望，再转化为产品需求的过程。

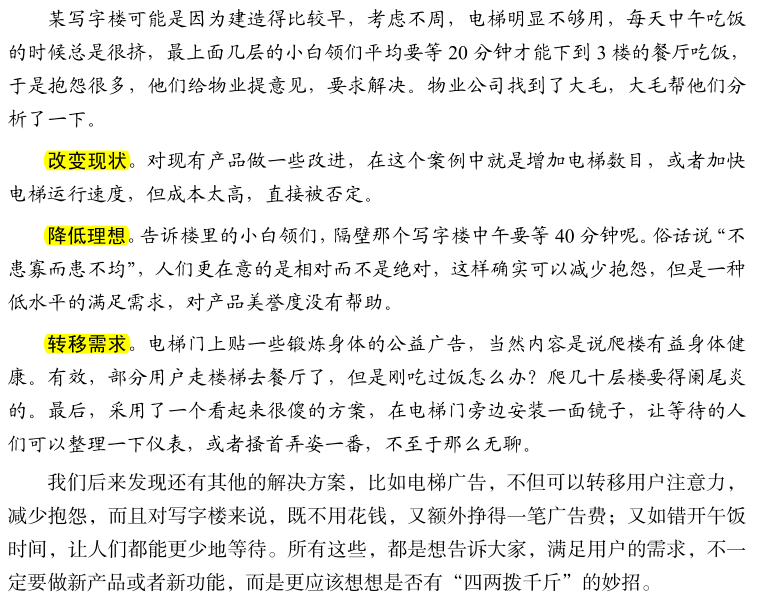




伟大的需求分析师，可以无视用户想要的东西，去探究他内心真正的渴望，再给出更好的解决方案，或者说是用户真正需要的东西，这就是本节标题的意思——我们存在的价值。

这是短期利益和长期利益的权衡，如果是一锤子买卖，卖出以后又不用售后，那么采用实用主义，不妨用户要什么就给他什么，这样他掏钱最爽快，这种情况下就要追求短期利益。但是，我们的产品通常都是希望用户长期使用的，并且后续的服务也是我们来做，所以为了长期利益，我们就有必要找到用户的真实需求，然后给他真正合适的产品了，哪怕这个过程不那么讨好。

需求来源于理想与现实的差距，那么减小这个差距就有三种方式：



上面都是在谈满足用户提出的需求。然后他就想，能不能不劳用户的大驾，直接达到产品设计的最高境界——创造需求。

有过不少案例，最终获得了用户的认可，但更多的被证明是过于天马行空。

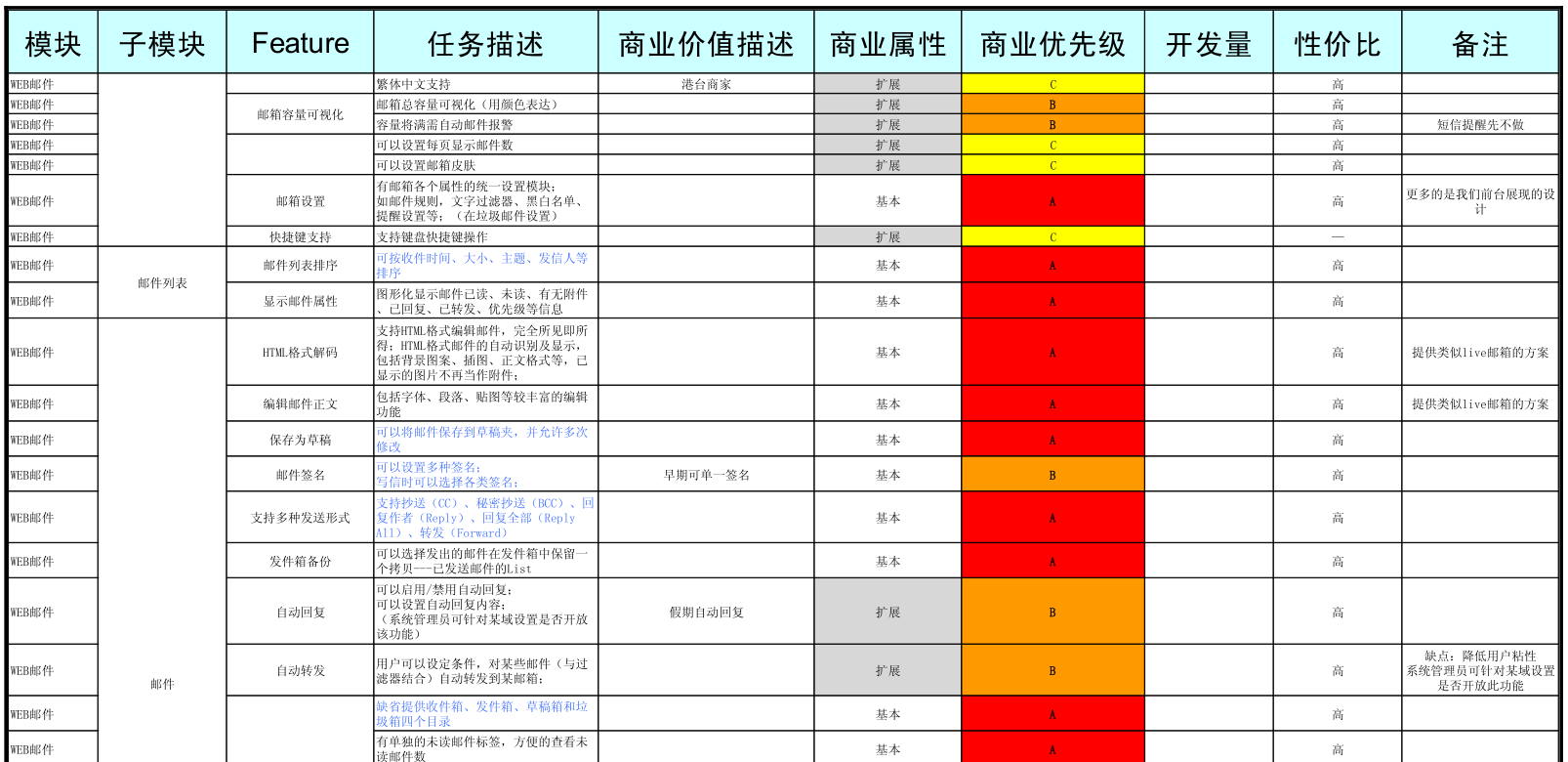
更实际的，我认为需求分析的过程其实也有创造需求的成分，当一个新人真的能力不足的时候，不妨先做用户提出的需求，而不要自己去胡乱分析用户需求，而对于一个团队来说，要尽量避免“只有能力不足的需求分析人员”这种情况出现。

2.3.2给需求做一次DNA检测/82

先把用户需求转化为产品需求，然后一步步确定每个产品需求的基本属性、商业价值、实现难度、性价比等

1需求转化：在一个团队里，还是建议大家统一一种记录用户需求的形式。

然后举行头脑风暴，产生想法。



表格中每一行是一个产品需求，而每一列描述了产品需求的一种属性

需要做一轮筛选，把明显不靠谱的用户需求直接过滤掉，不计入上述列表，当然，是否“明显不靠谱”就要由你来把握了，不要把“没资源做”、“短期内有技术难点”的用户需求给错杀了。



需求的分类：可以分为“新增功能、功能改进、体验提升、Bug 修复、内部需求”等。其实产品需求远非我们直接可以想到的功能需求，还包括了很多非功能需求，比如：性能、可培训、可维护、可扩展……有很多需求不是为终端用户做的，而是为销售、服务、测试团队的同学做的

需求的层次：把需求分成“基础、扩展（期望需求）、增值（兴奋需求）”三层。

雪中送炭是基本功能，对用户很有用，产品缺了这个功能根本跑不起来；锦上添花的功能是指非必须的，用户有时用得到，有的话会给用户的使用带来方便，而当一个锦上添花的功能被用户普遍接受以后，几乎所有的产品也都拥有了，也就渐渐发提升为雪中送炭的功能了。



商业价值，或者叫商业优先级，是对上述几种商业价值指标的综合评判。这一条是整个需求列表中最核心的部分。

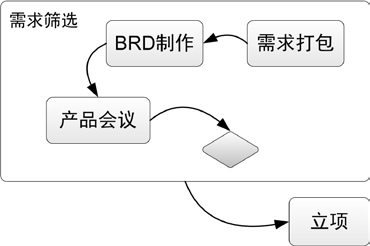
绝对不能因为某个需求的商业价值很大就马上去做，也不能因为另一个需求的商业价值不大就不做。还需要另一个关键指标，那就是——实现难度。开发量是非评估不可的，我把它叫做“初评”，允许误差，并且会要经验丰富的人来评估，通常是技术经理，或者系统分析师、架构师。

我们已经做了需求采集，把用户需求转化为产品需求 ，知道了某个需求的基本属性、种类、商业价值、开发量，现在似乎应该开始写文档、干活了，但经验告诉我们不是这样的：绝对不能因为某个需求的实现难度很小就马上去做，也不能因为另一个需求的实现难度大就不做。一个需求实现难度不大，但一直在功能列表里放着没动，一切皆看性价比。

性价比 = 商业价值÷实现难度（简化为开发量）

**2.4 活下来的永远是少数/92**

需求筛选



2.4.1 永远忘不掉的那场战争/93

更早的时候，公司是按照产品线划分部门的，对于某个产品来说，有自己的产品设计师、开发与测试等，下一段时间要做哪些需求，完全可以在产品经理的层面上决定。

公司组织结构调整，变成了按职能线划分团队，有了统一的产品中心，每个产品还是由原来的产品人员做，但是开发与测试资源在一定程度上就有了流动的可能。每个产品想做的需求都很多，所以都想尽可能多地抢到开发与测试的资源，然而人力资源总是严重不足的，所以最终把资源投给哪个产品，就必须上升到几个中心的大老板层面来决定了。

1准备出发：把需求打个包

做项目，终极目标就是：多快好省，即范围大、时间短、品质高、资源省。

2战场：产品会议

3武器：商业需求文档

BRD 怎么写，都包含哪些内容。

项目背景：我们在哪里？为什么要做这个项目，解决什么问题.

商业价值：我们去哪里？最关键的重点！

功能需求描述：我们怎么去？通过做哪些事情来达到目标

非功能需求描述：提一下重要的非功能需求，如果有的话。

资源评估：第二个重点！。

风险和对策。

其实本质上也是在追求那个词——性价比。大家都希望花费最少的资源获得最大的商业价值。

2.4.2别灰心,**少做就是多做**/100

有 100 个需求，资源只够做 10 个，是的，当时就是这样。一直都是这样。情愿把一半的功能做到尽可能完美也不要把全部功能都做成半吊子。越来越觉得当发现一个功能可有可无的时候，甚至只要是没有强烈的理由要做的时候，要明确的选择：不做！

尽可能多地放弃

第3章项目的坎坷一生/111

**3.1 从产品到项目/114**

项目是只会进行一次，包含多项互相关联的任务，并且有绩效、时间、成本和范围限制的一项工作。

产品是一个解决问题的东西，而项目是一个过程，根本是两个“做项目”,其所做的事情确实有几分相不同层面上的概念，可人们经常说“做产品”似，所以我们有必要把它们放在一起比较一番。

第一，从生命周期的角度来看。

做产品的生命周期相对较长，关注的是整个产品从规划到制造，再到最终维护和消亡的整个过程。而项目有特定的目标，所以生命周期较短，通常在项目开始以前就有明确的起始和结束时间，通过验收则表示项目生命周期就算完成了。

第二，从具体要做的事情来看。

做产品的过程，会有更多的探索，随着各种内外部信息的变化，产品负责人需要不断地修正自己的判断,给出适宜的创新;而项目在开始时就已经有明确的目标,更注重计划与控制，项目的过程很像是执行一个任务。

第三，从产出物的角度来看。

产品是可以批量生产，或者提供给大量用户的，所以相对通用，通常考虑用有限的资源去满足更多的、能有更多回报的需求;而项目只进行一次，意味着每次都是定制的、个性化的，通常为了满足特定的需求，产出物也比较个性化。这就意味着，项目要满足那个特定的需求;当然我们会看到有的项目产出物经过改造，成为更通用的产品，或者有的产品也可经历“个性化定制”(即做项目)。

“做产品”和“做项目”也是分不开的，“做产品”的过程，正是通过做一个又-个项目实现的，但并不是项目的简单累加。在产品渐渐满足目标用户群体的通用需求后，继续做下去的话，会使产品成为项目的集合体，肿不堪，因此我们就需要细分市场，这时候产品可能升级为“产品线”，按不同的细分市场，推出不同的产品，表现形式上叫版本、模块、增值服务，什么都行，本质上是通过合理地安排项目来实现产品的规划。

产品经理--靠想。产品经理是做正确的事，其所领导的产品是否符合市场的需求，是否能给公司带来利润。

项目经理--靠做。项目经理是把事情做正确，把事情做得完美，在时间、成本和资源约束的条件下完成目标。

用我自己的话总结，就是:一个内部驱动，一个外部驱动。

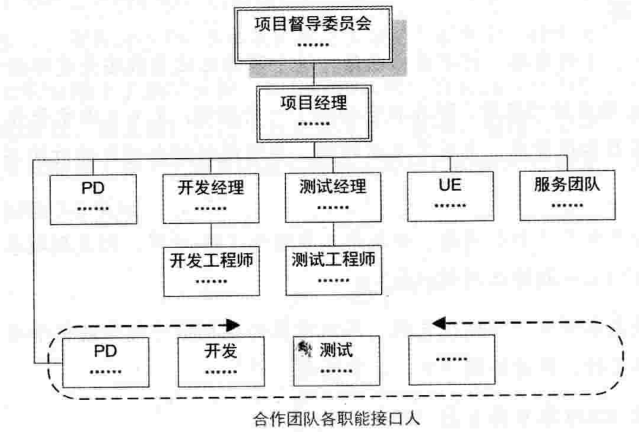
对产品经理来说，最重要的是判断力与创造力，产品经理决定做不做、做什么、做多少，保证方向正确。他是产生一个想法，“我要把它实现!”

对项目经理来说，最重要的是执行力与控制力，项目经理决定怎么做、谁来做何时做，保证方法得当。他是接到一个任务，“我要把它完成!”

很多团队里，都让产品经理管项目。事务具有两面性，2009年的时候，有的团队又调整为让开发经理担任项目经理了。主要是发现 PD会不断地给项目增加新的需求，导致项目总是无法按期完成，继而大家怨声载道，影响了团队的士气。

**3.2 一切从Kick Off开始/118**

因为产品经理并没有行政上的管理权，也就意味着每次项目都需要去跟不同团队的主管、经理要人，所以其中的艰辛也只有经历过才能体会。



典型的项目组织结构。这个组织结构肯定会随着项目的进行不断微调，我们也无须在项目一开始的时候就把人凑齐，一般来说，最关键的几个人到位了，就可以继续做下面的事情。他们通常是项目经理，需要统筹管理整个项目;PD，开始写需求文档:开发经理，制定项目的开发计划。

工作量估计常用的方法是“三点估算法”即评估出最乐观的工作量、最悲观的工作量、最可能的工作量，然后按下面的公式估算出工作量:

“工作量=(最乐观+最悲观+最可能)/3”

或“工作量=(最乐观+最悲观+最可能x4)/6”

“1 人天”通常等价于 5~6“人小时”，而并不是按照一天工作8小时计算，因为每个人都很难保证持续高效，并且不被其他事情打断。一个没人打搅的日子，能有5个小时高效的工作，就已经很不错了。

之后，工作量做汇总，推算工期，需要注意的是，工作量只是说某人完成这件事需要多少时间，而工期是转化到日历上，说明这件事从何时开始到何时可以结束。这时候要考虑到各项任务的依赖关系，以及项目成员在这段时间内的其他工作，并适当留出机动时间。

比如某个开发任务只需要工程师甲做“2人天”，但它需要等工程师乙“5 人天”的任务完成后才能开始，那么这两个任务就没法并行，工期也就是“7工作日”而不是“5工作日”;又如开发团队每周五下午有周例会，项目中的成员必须要参加各种评审会议，还需要响应随时可能发生的线上故障，排工期的时候都需要考虑到这些。

项目管理的新手总是认为可以通过加大资源投入来缩短项目时间，一般来说，如果各个任务相对独立，则可以更多地并行，通过投入资源来缩短时间，比如给一大片果园摘苹果就可以采用此方式;但如果任务互相依赖严重，就只能更多地串行，这时候加人也往往无能为力，就像我之前提过的那个“十月怀胎”的例子，更坏的情况，是在软件项目中，经常因为“延期、加人、老人带新人耽误时间、更加延期”，而进入恶性循环。

项目沟通:

周期:以“日”或“周”为单位，主要取决于项目时间的长短及变化的频率。

渠道:会议、邮件等，需要在成本和效率之间取得平衡。

发起者:一般由项目经理、开发经理、测试经理主导相应的沟通。

参与者:发起者确定参与者，不要遗漏项目边缘的同事。

不可或缺的誓师大会

上面说的团队组建、时间计划、沟通方法准备好以后，我们就可以举行誓师大会了，也就是项目Kick Off会议。项目KickOff会议，我们简称KO，也许有的人会觉得很形式化，但我认为很重要。我们自己全程参与项目，所以对各种信息都已经了解但是还有很多项目成员在KO之前很可能完全不知道这个项目是要做什么，而且KO的很大作用是在心理上的，就好比周瑜在战前发表“演讲”、敬天地鬼神，鼓舞士气。自古以来，就有这样的 KO。

KO需要传达：

项目背景：我们在哪里，为什么要做。

项目意义、目的与目标：我们去哪里，什么算成功。

需求、功能点概述：我们怎么去。

项目组织架构：互相认识，有什么事该找谁。

项目计划。

沟通计划：很多事情不顺是沟通的问题，用规则来逼大家沟通。

最后提一点，凡是会议，就总会有人缺席，所以别忘了把会议记录发给所有干系人。

**3.3 关键的青春期，又见需求/127**

3.3.1 真的要写很多文档/127

**BRD:Business Requirements Document，商业需求文档**。这是产品生命周期中最早的文档，其内容涉及市场分析、销售策略、盈利预测等，通常是给大老板们演示的PPT，也就比较短小精炼，没有产品细节，有点像创业者给投资人看的商业计划，主要为了获得认可，争取资源。

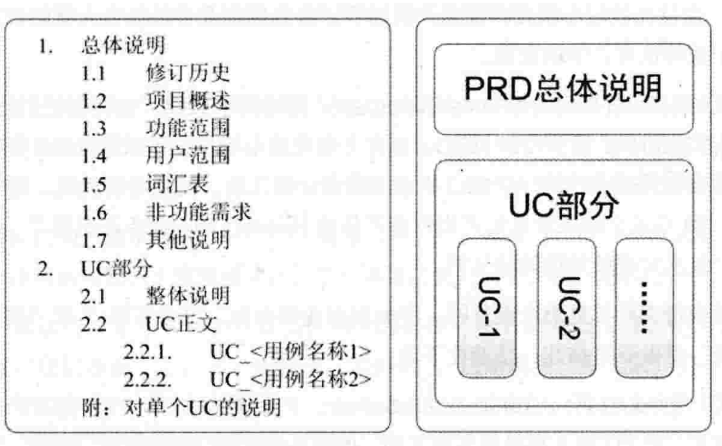
**MRD:MarketRequirements Document，市场需求文档**。获得老板们的支持后产品进入实施阶段，需要写出MRD，要有更细致的市场与竞争对手分析，包括可通过哪些功能来实现商业目的，功能、非功能需求分哪几块，功能的优先级，等等。实际工作中，PD在这个阶段常见的产出物有产品的FeatureList、业务逻辑图等，这是从商业目标到技术实现的关键转化文档。

**上述两份文档都是给老板看的**。

**PRD:Product Requirements Document，产品需求文档**。PRD是对产品功能的进一步细化，是 PD新人写得最多的文档，也就是我说的“需求开发”过程。文档主要包含整体说明、用例文档、产品Demo等，会对产品功能做具体描述，更多内容在下一节详细讲述。

**FSD:Functional Specifications Document，功能详细说明**。比较像我们常写的用例文档，经常包含在PRD中，从这步开始会出现很多技术的内容，产品界面、业务逻辑的细节都要确定，比如网页上的某个表格中的数字，应该左、中、右对齐?保留几位小数?等等。有一点像“概要设计”。与此同时，硬件系统的设计、数据库设计表结构设计等工作，也开始由架构师、系统分析师们编写了。

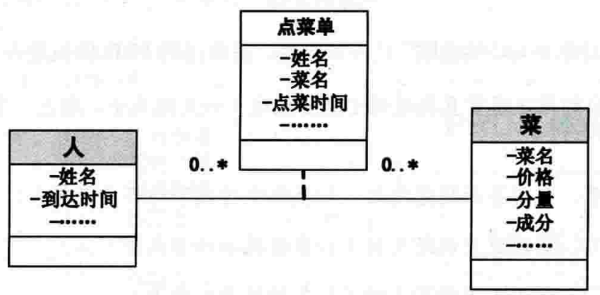
两种文档主要是**写给技术人员看的**，再往后，就到“详细设计”和编码了。

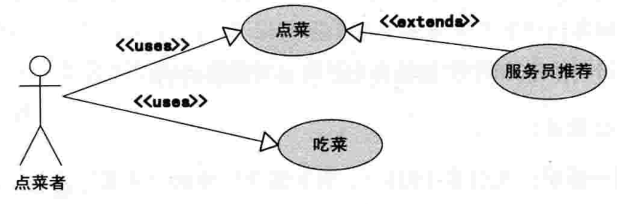


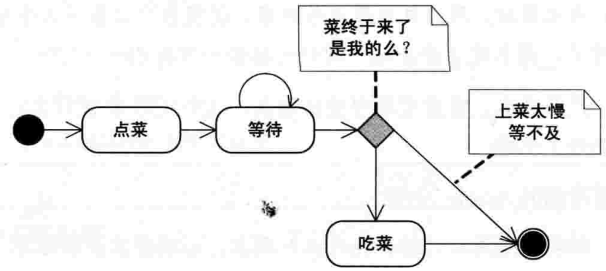
PRD模板与结构

公司一般会经历“人治一法治一德治”的三个阶段:人治是“由外而内”的被治，靠的是领袖魅力，更适合小公司;法治靠的是“硬法律+软伦理+执行者的以身作则”;而德治是“由内而外”的自治，靠的是企业文化，更适合大公司。很多小公司停留在第一阶段，很多大公司停留在第二阶段，只有少数公司才可能进入第三阶段。**第三阶段和第一阶段表面上很像，但区别在于第一阶段根本无法，而第三阶段的法在每个人的心中而非纸上**。

这也是产品和团队发展的必经阶段。一开始我们非常推崇个人英雄主义，或者在老板的带领下，或者自己带领几个菜鸟，没有章法地怎么快怎么做。慢慢地，产品越来越复杂，团队的新人也越来越多，于是我们开始在文档上做起了规范化的事情，从第一阶段走向第二阶段。

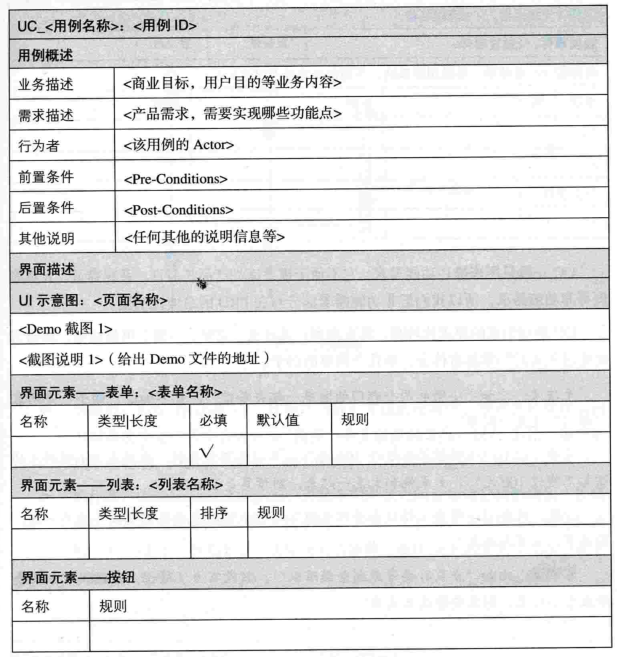
类图：

例图：

状态图：

用例文档，UC

模板





3.3.2需求活在项目中/140