真正的世界不在你的书或地图中，而是在外面。 ——《霍比特人1：意外之旅》  
  
从来没有人告诉我产品经理应该在实际操作中怎么做。有的时候你可能完成了需求文档，就每天呆坐在位置上等待开发完成需求；有时候你可能并不知道自己要做哪些事情，而只能看着产品经理的第X本书，奉为宝典，期待书里给出一个答案。我第一次介入项目，并开始做分配的第一个需求时，意识到自己显然不能再依靠书上的知识混日子了，而是应该从零开始了解整个过程。  
  
不管是参与到一个项目中，或是创业，或是自己组织一个业余项目，都面临一个问题：开始了之后，我应该怎么继续下去？  
  
不管是哪一种，都必须意识到做一件事情必须是有流程的，做产品并非是凭借持久的热情和激情推动，并非是通过责自我内心的驱动力和成就感去推动，大部分时候团队会进入倦怠期，因此流程化是一个比较好的选择。当然，流程这东西有时候好，有时候也容易成为阻碍，这是后话。  
  
所以，一个产品经理刚开始介入到一个项目的时候，一定要注意对于流程的理解和掌握，任何项目都依赖制度和流程去推动。流程就好像是地图上的线路，从北京到深圳要如何走呢？高速路上可能会出现许多标识，提示我们：离深圳还有100公里。当我们顺利到达深圳，则流程就结束了，任务也完成了。  
  
流程的重要性并非是在于指明一个方向，更是在于确定一个共同的目标，并在整个项目时间点进行分割，安排各项事务，并设立项目的里程碑。比如我刚入职的时候，不明确自己的职责是什么，还自以为是地画了一个甘特图表达自己对于手里需求的重视。实际上，这些事情应该是项目领导去制定，而我们参与其中，大家一起按照流程同步前进即可。  
  
**什么是项目？**  
  
项目可以是一个产品，可以是一个功能，可以是一个活动，项目可以是任何一个东西——但项目不能缺乏目标。项目目标确立意味着整个团队拥有了前进的方向和目标。曹操望梅止渴的故事，就是一个小项目，可以戏谑地称之为：“行军缺乏水需要尽快撤退保证军心稳定之寻找看不见的梅子项目”，目标就是“稳定军心，尽快回到大本营”。

｜项目目标很重要，并需要每个人都明确地了解这个目标。

**我在项目中的定位是什么？**  
  
不同的项目会有不同的人员角色配置，所以产品经理一定要明确自己的职责和定位。如果有项目经理存在，一般产品经理就不需要负责项目的大小事务及推动项目进展，只需要专心搞好用户体验即可；如果并没有配置项目经理，则需要产品经理承担这部分的责任。  
  
不仅如此，产品经理还要了解自己负责的需求是什么，这个需求是否和其他的产品经理有交集，需要和哪些开发人员及设计师去沟通，需要哪些资源和支持，这些都需要在项目之处了解清楚，不然有可能并没提前评估工作，陷入困境。  
  
**除了项目之前的准备，还需要明确产品经理在整个项目过程中有哪些职责，一般是这样：**

* 制定Feature List，并排好优先级
* 与项目其他成员宣讲Feature List，并重新根据开发计划调整优先级
* 敲定了Feature List之后，开发人员、产品经理等相关成员需要输出各自的计划和方案
* 输入方案之后，产品经理撰写需求文档并进行评审
* 评审通过之后与开发、测试等需求相关的人员进行评审及宣讲需求文档的具体内容和功能
* 在开发过程中，保持与相关人员的沟通，并随时体验产品，保持体验和需求文档一致
* 在测试环节，产品经理需要和测试工程师一起保证产品质量和体验过关

发布前，产品经理则需要确保功能体验良好，并输出后续的计划。  
  
**从0到1 —— 先接触用户  
  
产品经理离用户有多远？**  
  
日常查看用户反馈和建议时，我常常遇到这样一个问题，如何去有效地判断哪些反馈和意见时需要我这边进行处理的，如何安排优先级？  
  
面对一大堆ID，我着实犯难了。对于他们，我没有现成的用户个人资料，不了解他们的场景和使用问题，不了解该如何重现对应的Bug，不了解他们急切反馈的需求是否真的已经影响了整个产品的使用，不了解还有一些喜欢发送一些负面字眼的用户，究竟时遇到了什么问题，让太慢如此恼羞成怒。每当看到反馈意见，我都不得不反复思量，逐条验证和思考。有时候，不得不感叹，产品经理满嘴离不开“用户”这个词，但实际上，我对用户了解有多深入透彻呢？  
  
产品经理总是喜欢使用新奇的玩意。当我使用iPhone 5的时候，绝对想不到还有许多使用iPhone 4的用户，他们反馈的闪退和卡顿，是我无法体会的感受；当我习惯使用某个功能时，绝对想不到许多用户根本不理解那个按钮的意思，大部分人很担心这个功能会造成不可逆的破坏；当我身处于各种WiFi网络的环境下，想不到普通用户对于流量的珍惜。有一天，我在公交车上观察到有个人每次登录QQ都会再退出一次，连续好几次，就是担心iOS后台推送的消息会消耗流量。虽然退出帐号的方式并不能减少流量，但每个人用户对于产品功能的认知深深震撼了我。  
  
用户群体非常复杂，如果需要给每个人列一个标签，估计会列出一大串来。对于用户群体的复杂性，很多产品经理无法很好地理解。举个例子，许多人都抱怨人多，但却不知道自己所在城市有多少人，甚至不知道全国有15亿人时一个什么概念。很多人生活在一个500万的人的城市，每天接触的人不超过500个，他就很难想象一万个人大概是多少，他们会说，大概就比现在稍微拥挤一点吧。但是到了春运的时候，看到整个火车站排了那么多人，才会惊讶整个城市原来可以容纳这么多人。  
  
产品经理其实对于用户群体人数不是很敏感，尤其是缺少用户调研经验的产品经理，用户的概念只是一串数字而已：增长了1.3%、下降了5.4%、新增了XXX万用户。我常常和一些朋友开玩笑：国内的互联网公司简简单单地推出一款产品，动不动就上百万的用户群，对于一些国外团队来说，真的是望尘莫及，羡慕死了。在大公司做产品经理更容易对数字麻木，因为极少有产品需要艰难地成长，想尽办法新增用户，大部分时候都是依靠各种资源推广，轻松地就达到了目标。同时，用户数越庞大的产品，用户反馈越多越杂，产品经理需要大量精力去处理这些问题。所以，大部分时候，产品经理离真实的用户场景远着呢！  
  
彼得·德鲁克(Peter F. Drucker)在《创新与企业家精神》谈到：但是，真正缺乏的，是有人愿意倾听，愿意将每个人都挂在嘴边的信念——“产品或服务的目的是满足顾客的需求”真正当做一回事。  
  
产品经理满口提到的都是用户，大部分时候是从自己的生活经验出发，或者是通过猜想和观察去获得用户的经验。这样的方式不仅没有合理性，也不具备普遍性。观察方法是用户调研中最简单的方法，往往大部分时候用户的行为和表现不能直接用于论据，因为这些行为很有可能不具备普遍意义，而且行为很难推断动机。  
  
史玉柱在他的自述新书中提到一个案例：去年9月份《巨人》里一个失败的修改案例，就是调整拉车的颜色。为什么会做这个决策呢？当时有玩家提意见说为什么我总是拉白车，别人都紫车。于是《巨人》的产品策划就听取了玩家的意见，就取消了随机性。  
  
这个决策看起来很合理，用户提出别人都是紫车，我是白车，很不公平。如果我是《巨人》的产品经理，我可能也会做类似的事情。对于产品经理来说，不就是一个颜色嘛，从技术角度来看，很容易实现。通过调整这个颜色满足了用户的需求，很合理。但实际上，这就是一个只看到表面的决策——用户觉得不爽，不一定是产品的问题，很有可能是其他因素影响。从技术视角来看，颜色是否随机就是修改一个简单的函数，但从产品视角就需要确认这个产品调整颜色带来的后果。  
  
史玉柱后来说道，本来玩家之前还有企盼，取消了车之间的差距后，再一看，没人来拉车了。后来又改了回来，改回来后，拉车的人气又起来了。  
  
用户的心思很难琢磨，今天是一个颜色修改的问题，明天说不定还有其他的问题。每一个产品经理都会遇到类似的“陷阱”，看起来是一个很容易满足的产品需求，但大部分时候用户没有准确地说出内心真正想要的东西，往往产品经理没有深入思考，很容易就被用户误导，做出了错误的决策。  
  
那么，应该如何有效地保证满足用户的需求，又避免被误导做出错误的决策呢？  
  
**到用户中去：找到你的核心用户**  
  
**史蒂夫穆德（Steve Mulder）在《赢在用户》中提到：**

* 用户的目标和我的目标不一样。
* 用户的关注点和我的关注点不一样。
* 用户和用户也不完全一样。

正是因为有这三点问题，所以产品经理在日常思考的过程中难免陷入到为自己设计的惯性思维。当然，并非产品经理不能成为用户，而是因为每个产品都有特定的用户群体，而往往这些用户不是产品经理一个人所能代表的。因为日常缺少接触真实用户场景的机会，所以大部分时候产品经理只能依靠自己的生活经验去判断用户场景，这种决策环境往往不会带来真正打动用户的改变。  
  
不仅如此，一个产品的用户群有共同性，也有差异性。如果只是单纯满足共同性，产品决策很容易，只需要满足一种情况，但往往这种很容易满足的功能，不会带来产品经理希望看到的那种结果。很有可能用户对此的评价是：

* 哦，又多了一个新功能
* 还行吧
* 感觉有点多余了

万一这个功能做得不够好，有一些小问题，用户的反馈就不那么客气了：

* 这个新功能好烂啊
* 还不修复之前的Bug，现在又出来一堆问题
* 产品经理脑子进水了吗？

在处理共性与差异性的问题上，产品的改版是最容易受到用户反馈的。对于一个新产品来说，产品经理发现很多功能并没有真正有效地满足用户需求，于是希望通过改版去引导用户使用新功能，获得更加好的体验（当然，也包括产品自身获得更多商业价值），但用户却部这样认为。有些用户会抗拒这种改变，找产品经理哭诉要求改回原来的版本；有些用户欢迎改变，但往往没几天也开始加入到吐槽队列中；有些用户则没有表达态度，默默地使用着。豆瓣是一个神奇的网站，每次改版都被用户吐槽。继2012年12月豆瓣针对首页进行了改版之后，2013年9月又进行了一次改版。与之前改版不一样，这次并没有提供一个“返回旧版”的按钮。毫无疑问，用户开始了一轮吐槽。  
  
不要让一个人的特质影响你的判断能力，这是最重要的。



图1豆瓣最新改版页面

老版本的首页是一个推荐信息页面，而“友邻广播”则是单独一个选项。新版本则把两者结合成“首页”，并加入了一些主题内容，允许用户去订阅。从产品本意来说，通过定制化的主题，增强内容的可读性和UGC质量，是一个非常好的出发点。但是这就有一个问题，是否所有的用户都非常关注首页的订阅内容？  
  
这就是共同性和差异性在这里的体现。削弱了友邻广播，对于喜欢通过豆瓣交朋友的人来说，大量的订阅内容很可能会成为一种困扰；增强了内容可读性，对于通过豆瓣获取信息的人来说，这是一个非常好的改变。对于习惯于旧版的人来说，这次改版没有提供“回到旧版”的按钮，则会表达地比较激进。  
  
不同的用户之间有着许多的差异性，产品经理在其中就需要不断调整方向，尽量保持平衡，满足主流用户的体验。正如钟摆的两端，一端是用户需求，一端是产品方向，每一次改版都是在寻找最佳的中间点。但这个中间点并非是一个明明确确的点，恰如赤道往北23.5度一定是北回归线那般精确，而是在每一个产品决策中进行取舍——这就引发了一个问题：产品经理每次做决策的时候，是否会出现摇摆的问题？  
  
什么时候应该考虑用户利益，从用户角度去改善；什么时候应该牺牲部分体验，而增强产品的商业价值，的确不是特别可以把握。即使是乔布斯，也需要有一个参考系去判断，哪些时候应该选择保证最佳的体验，哪些时候应该适当弱化功能。  
  
张小龙曾经说iPhone的设计非常符合人性，“足够自然”。不仅是因为触摸符合人的天性，home键做成硬件按钮，足够简单明了。但正是这样的设计，引发了其他问题：

* 触摸设计，忽略了用户喜欢按键的反馈体验
* Home硬件按钮容易损坏

即便是站在人文与科技路口的乔布斯，也在决策过程中不得不做出选择。正如iPhone开始流行的时候，大部分用户不会使用iTunes和iPhone进行同步，人们很难想象，为什么不应该是U盘式的体验，直接拖动文件到手机即可？但iTunes的设计理念是鼓励用户通过商店直接购买音乐传输到iPhone，而不是先下载音乐再放到iPhone中。此外，“同步”的概念很难被用户接受，为什么每次连上iPhone，都会打开iTunes进行同步操作？  
  
正是这样的情境下，国内衍生出了许多类iTunes的服务，例如91助手，通过更符合国内用户的使用习惯，帮助用户管理iPhone。正是看准了iTunes在用户体验上的缺失，91助手等产品快速成长，通过广告等商业模式积累起了产品的价值。2013年7月，91助手被百度用19亿美金收购，让人咂舌。  
  
没有最完美的产品，只有变幻的用户群和市场环境，因此对于用户的研究是每个互联网公司都必须十分关注的问题。lina是一名资深用户研究员，她曾和我谈起在一些互联网公司中，虽然安排了用户研究这样的岗位，但是大部分时候用户研究成为一个寻找论据的工具，而非是一个长期积累用户数据、分析用户习惯的过程。  
  
正如上文所说，在用户价值和商业价值之间，产品经理常常需要做出决策。如何做出最佳决策，尤其是关乎用户体验的决策。一千个人眼里有一千个哈姆雷特，每个人都有自己的生活体验，因此在决策过程中，每个人都发表自己的观点，少见有一致的决策方向（即便是一致的方向，其实并不代表是最正确的决策）。这个时候，用户研究人员就出场了，他们负责根据需求进行可用性分析，寻找用户进行快速测试，判断哪个决策才是最佳选择。  
  
“这很容易失去用户研究的意义！”我在日常的产品开发过程中，也深刻地感受到一个问题：如果用户研究仅仅是为了做出决策，那么用户研究的价值究竟有多大？作为一个有点偏执的产品经理，我曾经对于用户研究抱有深深的怀疑。  
  
2012年时，我曾经和负责用户研究的同事一起参加用户访谈。在访谈过程中，我发现大部分时候用户研究人员为了尽快输出结果，常常需要用户进入到一个特定的情境中，然后让用户进行选择。如果失去了场景，选择是否有意义呢？  
  
有一个有趣的故事可以分享。当年Sony引入Boom Box的概念时，他们着急了一些潜在的消费者，组成焦点小组来讨论这个新产品应该是什么颜色的。当时有两种选择，黑色或者是黄色。经过一番深入地讨论之后，用户一致认为消费者更喜欢黄色的Box。  
  
先打断这个故事，如果各位是产品经理，这个时候你是否就可以把这个用户调研结果拿来成为一个说服老板的理由了呢？这就是把用户研究当成寻找论据的问题，大部分时候，用户表达和他们真正想要的不一定是同样的东西。  
  
那个故事的结尾很有趣，主办方为了感谢这群用户，允许他们带走一台Boom Box。让人震惊的是，这群用户都不约而同地选择了黑色，而不是他们之前口口声称的黄色。  
  
**因此，我特别赞同《赢在用户》一书中对于用户研究的态度。Steve Mulder和Ziv Yaar在书中描述了这样一个定位：**

1. 项目伊始，通过用户研究建立人物角色，并赋予这几个人物角色一些特定的资料和任务特征。
2. 项目过程中，通过这几个角色人物进行有效的考虑。假如这个功能这样做，这几个角色会有哪些反应，会导致什么后果？
3. 产品灰度过程过程中，针对用户群进行验证，人物角色是否有效？
4. 根据用户反馈重新调整人物角色，为整个项目发展及后续新产品的提供更多思路，例如商业价值等。

在这样一个过程中，用户研究就不再是一个沦为决策工具的应急手段，而是成为项目及产品的支柱。通过对于用户的深入研究，不仅对现有的产品有参考意义，对于后续团队发展和新产品规划也有对应的积极意义。  
  
**用户调研的方法**  
  
明确了用户研究在整个项目中的作用，产品经理就应该配合用户研究同事一起关注用户研究和人物角色的创立。根据用户研究方法来看，可以分为两种：

* 定性研究
* 定量研究

定性研究，即从小规模的样本中发现问题或者真实的用户想法。这种研究方法不会证明任何事情，但是能反应这样一种事实：即用户在使用产品的过程中有可能遇到这样那样的问题和一些困惑。通过建立在用户角度的阐述，产品经理可以了解到之前没有想过的一些使用场景和体验问题。值得注意的是，这些有可能只是一个表象，而非是真实问题的根源。因此，产品经理在获得了这些信息之后，需要去伪存真，深入了解问题的本质。正是因为样本较小，所以通过定性研究获得的信息，需要通过另一种方式去论证——定量研究。  
  
与定性研究不同，定量研究注重针对特定问题的深入了解，而前者主要是获得一些不了解的信息。定量研究是通过大量的样本来测试和证明某些事情的方法。对于定量研究来说，样本的数量和准确性，往往决定了结论的合理性。通过对于“数据库”的统计分析，产品经理可以通过统计学的原理获得一些有效且相对可靠的结论。  
  
换句话说，定性研究是为了了解用户使用的问题，注重于WHY（为什么会这样？）；而定量研究则是为了验证猜想和论证问题，注重于WHAT（事实上这个产品功能在使用过程中有哪些问题）。简而言之，定性研究和定量研究并非是独立互斥的方法，往往可以一起使用。前者发掘出用户说了什么，后者证明用户实际上做了什么。正如Sony Boom Box的案例，定性研究只能判断用户喜欢黄色甚于黑色。如果通过大量的数据分析，可能会得到一个相反的结论，用户可能更喜欢黑色。  
  
当用户说的和用户实际做的事情出现了矛盾之后，其实就引入了一个新的用户研究方法问题：这两种研究方法究竟应该如何有效地运用到产品日常用户研究中？当用户声称喜欢黄色，事实却选择了黑色之后，产品经理应该如何决策呢？  
  
结合史玉柱谈及的《巨人》拉车颜色的问题，如果当时做一个定量研究，用户一定会选择要拉紫色的车子，不要白色。因为前者比较稀缺，而后者比较普遍；如果此时不做定量，而是做定性研究，估计用户才会说自己为什么想要紫色的车——两种研究方法各有弊端，就需要用户研究人员和产品经理善于设计用户调研的方案，综合考虑得出最合理的相对客观的结论。  
  
**产品经理该如何利用用户调研方法？**

｜李明远,曾经在博客中如此描述产品经理对于用户研究的态度：

很多人是带着自己知道的结果去证明自己想的应该是对的，这不是调研，调研是对已知事实部分有未知结果和发展的（是帮助形成思路、启发而不是具体决定的），是用于防止决策错误和片面思维，而不是单纯以此形成决策的（所以为什么我们过去总是让新做产品pm的同事去做大量的调研分享，这不是抓苦力呀）；  
  
我们做产品调研，和做统计调研的区别就是，我们是主要调研失败、尚未成功而不是主要调研已经成功的（看得失而已，不要为“所得”动），那不是产品部的工作，是商业分析部之类的工作，他们是统计分析，我们是摸排探索。  
  
这段话让人印象深刻。产品经理并非需要了解成功的经验，并站在经验主义的角度去看事物——毕竟用户和场景都会会变化的。往往失败需要花时间精力进行调研。彼得·德鲁克在《创新与企业家精神》中亦谈到“意外的失败”对于创新的意义：  
  
分析虽然严谨——它需要进行测试、试验和评估——但必须建立在对变化、机遇和新情况的认知上，还要敏锐地察觉到大多数人仍然十分确信的所谓现实与实际现实之间不一致的地方。这需要人们愿意承认：“我所知道的仍然不足以进行分析，但我会去发掘。我将走出办公室，四处观察，提出问题，并用心聆听。” 意外事件能使我们跳出先入为主的观念、假设以及原先确定之事，所以它是创新取之不尽的源泉。  
  
IBM公司的历史同样表明，只要对意外的成功加以重视，就能产生效果。IBM之所以有今天的辉煌，在很大程度上是两度（而非一次）利用意外成功的结果。20世纪30年代初期，IBM几乎要完蛋了。它倾其所有资金设计了第一台银行专用的电动机械记账机，但是在30年代初的大萧条时期，美国银行并不想添置任何新设备。即使在那时，IBM也不曾实施减员政策，于是它继续制造这种机器，并将成品囤积在仓库里。  
  
就在IBM处于低谷时——故事就这么展开了。一天，IBM的创始人老托马斯·沃森（ThomasWatson）参加一个晚宴，碰巧坐在一位女士旁边。当她得知他的名字时，说道：“你就是IBM的沃森先生吗？你的销售经理为何拒绝向我展示你们的机器呢？”一位女士要记账机做什么？老沃森有点丈二和尚摸不着头脑。当她告诉老沃森自己是纽约公共图书馆馆长时，他仍旧迷惑不解，因为他从未去过公共图书馆。但是第二天早上，图书馆一开门，他就出现在那里。  
  
当时，图书馆拥有数目相当可观的政府拨款。两个小时后，当沃森走出图书馆时，手中拿着一份足够支付下个月工资的订单。后来，他只要谈起这个故事，就会笑着补充一句：“我当场创建了一项新政策：先交款，后送货。”  
  
因此，用户研究并非只是简单地套用方法那么简单，大部分时候产品经理和用户研究人员一定要明确每一次用户调研的方案和目标。平时我使用产品时，常常会有一些困惑，询问一些朋友，会用最简单的“人之常情”来告诉我一个答案。例如，年轻的用户比年老的用户更喜欢消费不等于年轻的用户消费金额比年老用户高，但通过“理所当然”的分析，我从身边获得答案往往充满了经验主义的回答。比如用户时常反馈想要主题，于是产品经理通过大量调研，分析了喜欢主题的用户群体，于是投其所好设计了好几款主题——一段时间后，发现默认主题依然时使用率最高的，而众望所归的主题并没有特别引人关注。  
  
那么，我们可以断定丰富主题的功能时错误的吗？我认为这里饱含许多因素：入口的可见度（有可能大部分用户没找到对应入口呢！）；主题的排序（大部分时候用户偷懒就懒得去选择）；主题的风格（用户想要的并非此类风格？）……还有另一种猜想：用户喜欢主题并非是自己喜欢，而是想给其他人看，而该功能没有满足。正是因为有如此多的因素，所以产品经理需要确认最关键的问题和每一次用研的目标，而不能理所当然地否决自己最初的判断：主题功能是有用的，要么就是没用的。  
  
彼得·德鲁克给予了我另一种视角：产品功能没有发挥预想的效果，有可能并非是功能没有存在的意义，而有可能是新的机会。在创新中有三种情况值得考虑：

* 意外事件
* 不协调事件
* 认知变化

上面对于主题功能的困惑就是一个意外事件，如果产品经理可以做调研，及时调整主题功能，有可能获得成功。与“理所当然”的思维不同，传统思维告诉我们，失败、意外应该被归结于运气、时机等因素，但彼得德鲁克则认为，这些事情的本质是需求没有得到满足，而且尚未有产品可以满足对应的需求，如果找到真实的用户场景和需求，有可能找到一个利基市场。  
  
从失败中探寻到机遇，与巴菲特的投资理念“在别人贪婪的时候恐惧，在别人恐惧的时候贪婪”有着异曲同工之妙。  
  
因此产品用研对于产品经理来说，其实一个手电筒。当你需要探索未知领域，从黑暗的失败深渊找到产品成功的曙光，从经验主义的认知中跳出来找到创新之源时，用户调研可以给予你很多灵感并指明方向。正如我前文所说，我对于一些互联网公司的用户研究方式表达怀疑，但用户研究依然是产品经理值得关注的方法；如果真正想要做一个反馈用户心声，为用户设计为中心的产品，那就在团队建设之初就开始投入用户研究的资源吧！  
  
**建立人物角色，确立核心用户**  
  
建立人物角色并非是一朝一夕之事，也并非是临时抱佛脚的措施，人物角色的创立有实际的意义。当产品经理通过大量分析确认了几个核心用户角色之后，相当于为产品经理做决策时提供了一种有效的参考。这几个核心角色，可以代表核心用户群的一些特征。角色的创建无疑为整个用户数据提供了基本的维度，用户数据不再是杂乱无章的，而是可以归纳分类的。虽然每个用户都是独立的个人，有着差异性——但从社会学研究方法来看，人的群体化属性依然奏效，因此通过核心人物角色去代表群体特征，简化了用户研究分析流程，但得到的结论依然可靠。  
  
不仅如此，角色创立并非是应急之用，而应该成为一个模型，通过这个模型可以规避一些“不符合用户场景”的设计，并成为决策的一个参考物，贯穿整个项目始终。  
  
对于创业团队来说，核心用户角色的创立更有深远的意义。大部分时候创业团队需要在市场中发掘用户未被满足的需求和机遇，但苦于没有一种有效的方法去接触用户（毕竟用户研究是一个很花费资源的事情）。如果建立了基本的角色模型，创业团队根据一些行为数据可以比较有效地了解市场，及时调整决策，不断投石问路找到最佳的切入点。  
  
创建人物角色可以没有场景，独立存在。日常使用人物角色的目标是产品团队的每一次决策都可以通过用户连接起来。考虑到不同的产品和功能，不同的阶段和产品目标，都有各自的使用场景，因此，人物角色要注重独立，避免场景化太明显，导致人物角色的使用带有偏差。

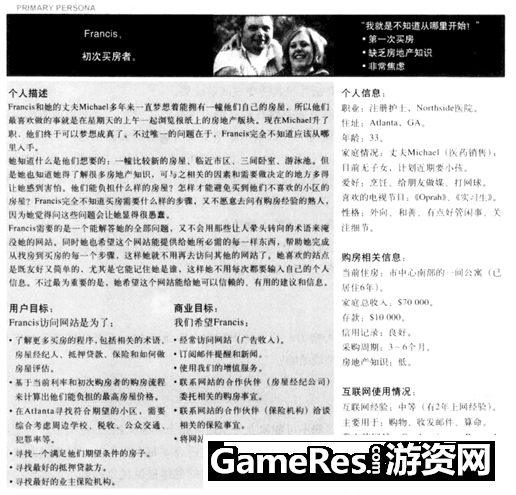


图2《赢在用户》核心人物角色设计模版

按照《赢在用户》一书中建立人物角色文档的方法，人物角色的展示可以有很多的形式。Steve Mulder和Ziv Yaar提供了一个人物角色文档的模版：

* 人物角色的姓名、照片、语录和关键差异列表，特别明显地标识出来，作为人物角色的一幅快照来使用。
* 描述性的简介、完整的关键差异（通常是用户目标的列表）和对于这个人物角色的商业目的，是第二个版块，占用这个页面最大的一块版面。
* 使人物角色充实起来的细节，比如个人信息、行业信息和计算机和互联网使用情况，安排在剩下的区域内。
* 确定人物角色的信息优先级，去掉不必要的内容。

通过建立人物角色文档可以让一些场景更加具体化，让人容易接受。  
  
既然我们已经知道如何运用人物角色的方法，那么对于产品经理来说，面临一个新的大问题：哪些人是我们的核心用户群？  
  
柯南道尔（Sir Arthur Ignatius ConanDoyle）在《血字研究》中描绘了福尔摩斯超强的洞察力：  
  
“正是如此。因为我有那么一种利用直觉分析事物的能力。间或也会遇到一件稍微复杂的案件，那么，我就得奔波一番，亲自出马侦查。你知道，我有许多特殊的知识，把这些知识应用到案件上去，就能使问题迎刃而解。那篇文章里所提到的几点推断法则虽曾惹起你的讪笑，但在实际工作中，对我却有着无比的价值。观察能力是我的第二天性。咱们初次会面时，我就对你说过，你是从阿富汗来的，你当时好象还很惊讶哩。”

｜“没问题，一定有人告诉过你。”

“没有那回事。我当时一看就知道你是从阿富汗来的。由于长久以来的习惯，一系列的思索飞也似地掠过我的脑际，因此在我得出结论时，竟未觉察得出结论所经的步骤。但是，这中间是有着一定的步骤的。在你这件事上，我的推理过程是这样的：‘这一位先生，具有医务工作者的风度，但却是一副军人气概。那么，显见他是个军医。他是刚从热带回来，因为他脸色黝黑，但是，从他手腕的皮肤黑白分明看来，这并不是他原来的肤色。他面容憔悴，这就清楚地说明他是久病初愈而又历尽了艰苦。他左臂受过伤，现在动作品来还有些僵硬不便。试问，一个英国的军医在热带地方历尽艰苦，并且臂部负过伤，这能在什么地方呢？自然只有在阿富汗了。’这一连串的思想，历时不到一秒钟，因此我便脱口说出你是从阿富汗来的，你当时还感到惊讶哩。”  
  
如果产品经理可以像福尔摩斯一样拥有如此强大的洞察力和推理能力，那还需要创建什么人物角色呢？不过福尔摩斯的洞察力并非是针对群体，而是个体，对于大部分产品经理来说，即便学习到了福尔摩斯的能力，也不一定可以快速地说出产品的核心用户群体是哪些人。  
  
用户群是一个很有趣的研究方向——目前业界内有许多关于用户群体的研究书籍，巴拉巴西在《爆发》提到，93%的用户行为都是可以被预测的；詹姆斯·索罗维基则在《群体的智慧》描述了群体决策的质量在很多时候超过群体中的单个个体的决策质量……群体的研究意义对于产品来说，非常重要，因为这可以提供一个参考，即群体的共性是什么，群体的反应会是怎么样一个过程。正如《引爆点》一书所说，流行并非是个体行为，而是个体引发的群体行为。如果一款互联网产品可以准确地掌握群体的行为逻辑，那么成为一款流行的产品只是时日问题。但遗憾的是，大部分时候用户群体的行为很难判断。因为群体的经验并非是固定不变的，会受到很多因素影响，不同的情境、不同的中间分子会产生不一样的效果。  
  
除了群体的行为，个体的差异性值得产品经理关注。在产品运营初期，或者是在做日常的用户调研时，常常会遇到用户反馈一些非常细节的问题，有时候产品经理会觉得这些问题很细小，并不在意；有时候则时在产品组内部做小小的调研，发现没有类似的问题，则忽略了用户反馈的一些小细节。但很有可能，这个个体的差异背后代表着一个群体都受到了类似的影响。在腾讯内部，一个产品如果有一个用户反馈Bug，那就会乘以对应的数量级，预测上万或者是百万数量级的用户受到影响。通过这种方式，让产品经理注重用户反馈的问题，起到良好的预警效果。  
  
正是因为群体行为无法预测，个体反馈很难判断多少用户受到影响，所以产品经理在寻负责产品的核心用户群时，往往需要借助数据分析及一些专业咨询公司的帮助。  
  
**用户调研：区分真需求和伪需求**  
  
蒂姆·罗斯（Tim Roth）在美剧《别对我说谎》（lie to me）中饰演了一位可以通过观察“微表情”来判断对方是否说谎的专家。剧中描绘了多种判断对方是否说谎的方法，例如：当对方在说话的时候，眼睛不自觉地往右上看。假如蒂姆·罗斯饰演的莱特漫博士做产品经理参加用户调研，是否可以准确地把握用户需求呢？  
  
在前几章我多次提及“伪需求”这个概念，不仅仅是因为伪需求很容易造成产品经理误解用户真实需求，更是因为大部分时候伪需求有这样一种影响力：当产品经理从用户反馈中获得一个需求，例如“用户希望加上一个快速查找的功能”，于是产品经理就会通过定量的方法去询问用户：你们是否想要一个快速查找的功能？大部分情况下，用户当然喜闻乐见新增加一个功能，而且理由也足够充分，这个功能可以方便用户快速找到对应的目标，没有人会拒绝，他们会表示“多多益善”。但实际情况呢？当时一致拍手称赞，想要新增查找功能的用户，从来都不用这个功能，而且会抱怨因为这个功能的入口存在，影响了原来的操作体验。  
  
伪需求是这样一种需求，它缺乏真实使用场景，一般都属于拍脑袋的行为。假如一个情侣应用找单身男做用户调研，把功能说的天花乱坠，单身男肯定会表示，他会使用这个应用。对于产品经理来说，这个调研很满意，全国有20%的单身男，说明这个市场潜力非常巨大——但是，单身男使用情侣应用的前提是，他真的有一个女朋友。缺乏场景下的需求，都很容易迷惑了所有人的判断。  
  
除此之外，需求需要区分表象和本质。因为用户在表达的时候，往往是最真实的反应，连莱特曼博士也无法判断用户是否说谎。大部分时候，产品经理听到用户反馈的需求，往往是用户在自己的经验上做出的判断和描述。  
  
在日常用户调研的过程中，因为时间、流程及调研方式的限制，往往很多时候无法获得最准确的结论，而是获得许多素材。我参加一次用户调研访谈，询问用户是否会使用iPhone通讯录右侧的字母检索条。当时用户都反馈，他们了解这个功能，知道该怎么用。于是我又询问，假如你的好友都是按照这种形式进行排列，你是否会觉得可以很方便地找到对应的好友？用户再次表示，他们可以很方便地通过这个功能查找好友。于是，用户调研的结论是：大部分用户通过iPhone通讯录的教育，会使用右边的字母检索条，并且欢迎把这个功能应用到现在产品中。  
  
随时用户调研人员又安排了一次可用性测试，发现用户在“指导”下知道如何使用这个功能，并且表示可以很快找到对应的好友。于是，这个功能就排期进行开发了。  
  
事实上，经过数据统计发现一个很凄惨的事实：人们并非像当时说得那样如此“热爱”这个功能，使用该功能的比例大大低于产品经理的猜想。相信许多人已经发现了这个案例的问题所在：想用和愿意用并非是对等的关系。  
  
**在得知了实际使用数据之后，我从中发现几个有趣的点：**

* 当你阐述功能的好处时，用户往往不会拒绝。
* 用户表达自己会用，不代表真实场景下，他们真的会使用。
* 产品设计的操作路径，不一定是用户最喜欢的；有时候，用户有自己的关键路径，而这条路径往往不是最短的那条。
* 习惯的力量是可怕的。

当你阐述功能的好处时，用户往往不会拒绝。在用户调研中，常常会遇到这样的情况。如果用户不用该功能，调研人员则会反问：为什么不用这个功能呢？它可以做很多事情。用户则会放弃坚持回答：我会使用的；如果用户使用该功能，调研人员则不在深入了解，默认这个态度为用户的赞同。不管出于何种原因，此类引导谬误常常会影响用户的判断，制造一种假象，即用户真的会使用该功能。  
  
《晋书·惠帝纪》中描述了晋惠帝对于天下饥荒的态度：何不食肉糜？用户调研过程中，常常会遇到“何不食肉糜”的问题。许多情况下，产品经理会好奇用户为什么不用这些“棒极了”的功能；出于掩饰自己无知的心理，大部分时候用户往往会遮盖自己真实的情况，赞同产品经理的引导。  
  
这个问题主要是因为产品经理在用户调研中过于积极，缺乏观察，有先入为主的情况存在，影响了调研的真实性。  
  
用户表达自己会用，不代表真实场景下，他们真的会使用。正如前文所说，需求需要结合场景。如果剥离场景，大部分时候用户会出现言行不一矛盾。  
  
早在19世纪30年代，美国心理学家理查德·拉皮尔（Richard LaPiere）就意识到一个人的态度与行为并不总是很牢固地联系在一起。LaPiere 得出这样的结论：如果你想预测一个人在面对某一真实的特定情景或特定人物时将如何表现的话，对假设性情景的口头回答（即用户的态度）是远远不够的。  
  
《启示录》的作者Marty Cagan提供了一个简单的评判方法：可以向用户提出一个问题，“你是否愿意把产品推荐给同事（或其他你在意的任何人）”，之后我们再让用户以推荐值自己打分，0代表极不愿推荐，10代表相当愿意推荐。答案很简单，只有用户打9-10分才是真心大爱，表示用户真的会“掏钱”使用我们的产品；用户打5-7分只是出于礼貌的敷衍你，表示对我们的产品根本不感冒。  
  
通过这种方式可以比较好地判断用户是否言行一致，符合产品经理的预期。  
  
产品设计的操作路径，不一定是用户最喜欢的路径在日常讨论中，常常发现一些很有趣的现象，即用户都有自己的“关键路径”。例如前文曾经提到“通过Google打开百度进行搜索”的事例。Jim Kalbach在《Designing Screens UsingCores and Paths》提到”计划外的路径”，让人深思：  
  
想象一下，当你隔着一片草坪想要到达对面的巴士站时，你是绕着草坪四周的人行道走过去，还是从中间穿过去呢？ 假如草地是干的，也没有被禁止进入，那么你很可能会选择最近的路线——穿过草坪直接走到巴士站。如果之前有不少人也这么干过，就会出现一条“走出来的路”。 这种计划外的道路连接了两点之间的最短距离，我们周围到处是这样的例子。 在城市规划中，它们被称为“交通需求线”(desire pathsdesire lines)，意思是人的自然行为和人为规划路线间的差异。  
  
在产品日常使用中，常常会出现这样的情况。产品经理及交互设计师在产品设计之初就定义了一条“道路”，希望用户可以老老实实地根据这条道路一步一步达到对应的功能，但很有可能这条道路不起作用。皮克斯的动画电影《赛车总动员》描绘了这样一个剧情：美国66号公路（US Route 66）曾经是贯穿中西部的主干道，但因为洲际公路的兴起，这条道路逐渐失去价值，人们不再经过这条曾经辉煌的公路。  
  
实际上，如果一个产品一开始没有明确整个路径的设计，很有可能整个规划都会如66号公路一样，成为被遗弃的路径。

｜“不要一上来就从主页和整体导航方案来开始，而要从核心内容入手，由内而外来设计。“

Jim Kalbach在文中引用信息架构师Are Halland的观点，并把该原则拓展到产品设计交互原则中。但我要提出一个问题，为什么大部分时候这个原则不一定起作用？用户为什么会选择产品设计之外的路径和最短的路径？  
  
影响用户选择路径的因素有很多，有可能是用户一开始接触的路径并非是常用路径，有可能是因为页面层级过深，这些在接下来的产品设计章节继续深入。  
  
总之，用户调研过程常常会见到用户使用非正常路径的情况，产品经理一定要关注这样的情况，很有可能这个操作的背后暗示了这样一句话：你们产品的交互设计糟透了。  
  
习惯的力量是可怕的观察用户使用产品的时候，常常会发现许多用户的使用习惯不符合产品经理的预期。例如许多IM产品提供了一个“最近表情”的面板，希望可以帮助用户尽快地找到表情，但大部分时候用户依然尊重原有的习惯，一页一页地滚动表情面板查找想要的表情。从操作方式来看，选择最近表情只需要三步就可以发送表情，而滚动面板查找则需要五步，但用户下意识地都选择了最常用的方式：直接查找，而且他们觉得很快捷。  
  
这个事情深深影响我对于用户行为的理解，有的时候，用户的行为已经被深深写入到神经链路，下意识地通过习惯影响行为。在功能机时代，常常会发现一个很有意思的事情，我身边的人都学会了盲打，即他们打字的时候不需要看手机键盘，也可以准确地找到想要打出的字，这一行为被广泛用于考试作弊（注意，请勿模仿）。但是到了智能输入法时代，盲打失效了，大部分人不得不查看屏幕上显示的究竟是哪个词语。虽然智能输入法提高了人们打字的效率，但受到词库的影响，词语的顺序发生改变，让人很难通过习惯的行为去快速选字。  
  
要么是最短的操作，要么是最熟练的操作。人天生就会适应于各种环境，正是这一种适应力造成了许多情况下的意外之行为。产品经理如果缺少生活体验，接触用户较少，很难理解用户的习惯行为如何形成并准确地优化产品体验。大部分时候，产品经理只能通过推断，选择产品设计需要合理的操作，而忽略合情的操作。因此，遭遇习惯力量的抗拒时，常常会有一些争议：我们究竟需不需要调整为用户习惯的方式？  
  
用户调研就是如此有趣的过程，产品经理可以观察到许多有趣的事情。正是因为有如此多的问题和不合理的情况，所以产品经理在关注用户需求时，一定要倾听用户背后的故事，减少主观的理解。伪需求和真需求的判断，最关键的辨别方法就是先观察，多倾听，把产品使用融入情境，深入了解用户使用场景，而非武断地判断。  
  
在《怪诞行为学》中，艾瑞里描绘了多种人们“非理性”的状态。在产品使用过程中，非理性的状态同样存在。产品经理在日常调研过程中，对于这些“非理性”的用户行为应该给予重视，而非主观地忽略。正如前文所说，用户反馈的需求千奇百怪，但无外乎三种情况：

1. 伪需求，缺乏场景的描述和建议
2. 真实存在的需求，但并不是本质问题
3. 根本需求，触达本质

其中第二种需求是最让产品经理头疼的问题，通过上面我列举的一些经验来看，想要透过表象找到本质并非易事，也无法通过用户调研方法有效地找到最根本的问题。我们能做的，无非是结合用研即猜想，通过灰度方式去试错，像福尔摩斯一样发现线索。正如福尔摩斯在《四签名》中所言：  
  
不要让一个人的特质影响你的判断能力，这是最重要的。

来源： <<http://www.gameres.com/msg_300451.html>>

 本章主要讲解了需求是如何诞生这个话题，欢迎分享到朋友圈并积极讨论。  
  
在经过如此漫长的阅读旅途之后，欢迎各位有志成为产品经理的朋友来到这一小节。别担心我们即将上路会带来的紧张，事实上，当你阅读完前面铺垫的章节，已经走在了成为产品经理的路上。而接下来，抛却理论性的分析和描述，我将从项目需求写起。请记住：需求不是终点，而是起点。  
  
如果把产品看成是产品经理的孩子，难免会和许多父母一样担心“输在起跑线”上。就好像脑子中冒出了许多好点子的人，时刻担心自己的点子被别人抢先实现了。实际上，这些战略层面的事情还不是最应该考虑的问题，对于一个项目来说，真正需要关注的意义是开始，当项目正式启动了之后，是否可以按照Roadmap 进行，各个环节和相关人员是否可以保证质量？而项目最开始就要求产品经理提供本次迭代的Feature List，并及时提供需求文档给开发团队、设计团队及测试团队。因此，项目开始的时候，产品经理的压力会非常大。  
  
Feature List 的来源是需求，而需求的来源有很多：用户、产品经理、老板……与其说人人都是产品经理，不如说人人都有需求。因此，一开始产品经理需要做的是搜集各方的需求。通过搜集对应的需求，产品经理可以有效地进行需求管理，准确评估需求的优先级，然后根据所选择的需求思考对应的功能方案。  
  
一个需求的诞生就是如此简单。  
  
**一个需求的诞生**  
  
上文所述是刚入门产品经理时的我对于需求的理解。随着经历的项目越来越多，发布的产品版本也越来越多，我深刻地感受到产品需求并非只是简单地积累、筛选、排优先级这么简单。正如蓝图之于摩天大楼一般，需求的规划决定了这个产品的发展方向——是成为平庸的产品，还是成为优秀的产品；是迎合主流群体，还是迎合特定人群；是专注一个方向，还是四处出击。对于产品经理来说，需求与孩子不同，我们可以等他出生后再思考如何培养这个孩子，而需求则从一开始就要考虑后续的发展。  
  
一般来说，产品经理可以将整个项目分为两个阶段。

* 探索产品：产品经理需要清楚产品的规划和发展。
* 执行计划：推动开发进度，让功能尽快实现。

探索产品就是管理需求的过程。产品经理不仅需要到处搜集产品需求，还要明确自己及整个产品团队对于产品的阶段性目标和长期目标，明确产品功能的定位，从全局视角出发思考产品方向。除了关注眼前，还要考虑市场的发展。市场并非是一成不变的，而是在发生着日新月异的变化，今天流行的产品可能就是明日黄花。用户是流动的群体，他们追求好用的产品，而不是固守成规，成为过时产品的拥趸。虽然产品经理不用考虑太多技术方面的问题，但产品的拓展性及方案的实现需要被有效管理。  
  
**明确产品定位及目标**  
  
虽然人人都在说需求，每个人都可以说出几个痛点来，但产品并非一开始就为了不断修补体验。产品应该有自己的性格，而这种“人格化”的内涵，是产品经理赋予的。乔布斯崇善“NO BUTTON”，才有了iPhone 的设计；贝索斯耐得住寂寞，追求长远目标，才会掏钱设置一个“万年钟”（Long Now Foundation）。  
  
我在写作这本书之前，就一直在思考本书的目标和定位是什么。产品经理这一行其实存在这样一种现象，产品经理与产品经理之间很难直观地进行对比，尤其是优秀的产品经理更难分出高低。如果像金庸小说中描述的那样，举办一场产品经理的“华山论剑”，估计三天三夜也比不出个孰高孰低。我认为，对于产品经理来说，业务能力熟练程度的差距真的不是特别明显，但对于产品的领悟力则很容易出现很大的差距——幸好，大部分大公司的产品经理不需要太多领悟。有老板在做决策，有极强的执行力也是优秀的一种体现。  
  
而我内心怀有一种情结，正如彼得•德鲁克所说：企业家精神。什么是企业家精神？创新是企业家特有的工具。凭借创新，他们将变化看作是开创另一个企业或服务的机遇。企业家必须有目的地寻找创新的来源，寻找预示成功创新机会的变化和征兆。产品经理应该是具备企业家精神的人，勇于寻找创新的来源，通过不断地探索，找到更好的产品解决方案。他们不仅可以大幅度地提高资源产出比，而且可以努力开创新市场和新用户群，尽力满足他们的需求。  
  
因此，我把我的想法聚合成这样一个产品，灌入企业家精神及创新的源泉，希望通过另一种视角去讲述产品经理成长的故事。我曾经想写关于我这两年的所见所闻、所思所得，写一本名为“产品经理日记”的书，但大部分时候，这种缺乏条理的描述对于想要成为产品经理的入门者来说，难免有点看小说的意思；而接下来几年，对于初入门的产品经理来说，业务能力和对于整个产品研发流程的熟练掌握才是重点。因此，我把此书定义为：产品经理的第一本书。不需要太多生涩的道理，而应该像三字经一样，方便刚入门和未入门的人学习理解。  
  
正是基于这样的考虑，我决定多加入一些人文视角，弱化技术视角，即通过日常的事物去描述做产品的“道”与“术”，而非像一本严肃的教科书絮絮叨叨地说着各种技术名词和概念。做产品应该是快乐的。我在写作此书的过程中虽然极度痛苦，不仅要考虑日常工作的压力，每天把下班时间都花费在这个产品上，还要时常面临思路不畅、缪斯女神离我而去的情况，但当我描绘做产品的分分秒秒时，我发现自己其实在不断地学习和成长，把自己所学所知整理出来，不仅能提升写作能力，对于产品的感悟也会越来越深刻。  
  
所以，产品一定带有产品经理的性格，而不是不断累加需求生成的乐高积木。产品经理在一开始就应该想清楚，我需要做一款什么样的产品，要有哪些功能，满足哪些需求，服务于哪些用户，最关键的是在探索产品的过程中，我是否有一个基本的产品原则？有了一个比较明确的产品原则，需求的优先级才有筛选和比较的意义。  
  
那么如何定义一个产品原则呢？虽然网上时常有一些文章分享产品设计的原则，但我宁愿称之为方法，而不是原则。如果把产品拟人化，产品原则应该体现产品的三观。

* 价值观：哪些能做，哪些不能做？
* 世界观：我们要满足用户的哪些核心需求？
* 产品观（人生观）：产品需要什么样的生态环境？

哪些能做，哪些不能做？条条大路通罗马，产品想要成功自然也少不了一些捷径。恰如虚竹凭借逍遥子百年功力，从小沙弥成长为逍遥派掌门；萧峰从小刻苦练习，锻炼出阳刚内力；而段誉误打误撞学习了“北冥神功”，专门吸取别人的内力。做产品一样会有类似的问题，所以Google 的名言是“不作恶”（Don’t beevil）。产品经理除了要有道德操守之外，还要明确一点：尊重用户。  
  
我们要满足用户的哪些核心需求？其实这句话有问题，我们不需要满足用户的“哪些”核心需求，只需要满足“一个”。每一个产品都有一个核心场景，而核心场景下会有一个最需要被满足的需求。如果有其他需求和这个核心需求不相关，甚至影响这个需求的满足，那么这些需求都应该先被搁置，专心把资源投入到一开始就定义好的核心需求上。一般来说，建议产品经理在探索产品的时候专注于满足一个需求即可，然后再考虑从核心需求出发进行拓展。这样做，不仅容易锁定核心用户群，而且容易形成有效的产品原则，也便于在后续的产品开发过程中使所有人员都保持一致的目标。  
  
产品需要什么样的生态环境？市场竞争激烈，产品层出不穷，恰如白垩纪时代，巨型恐龙横行但却遭遇天灾，无法继续生存；小型哺乳动物则因为身体具备恒温特性，而且身材小，热量流失较少，帮助它们在黑夜和寒冷下存活。产品如何在某个生态环境下生存发展，是每个产品经理都需要考虑的问题。正如很多学生在高考结束之后都不清楚自己要去哪所学校、哪个专业一样，眼瞅着计算机专业热门，结果几年后大量计算机专业毕业生狭路相逢；考量着金融贸易是今后的重头戏，结果一毕业就遭遇了金融危机，只能感慨就业形势严峻。产品经理在一开始给产品进行定位并设立产品原则时，就必须考虑到后续的发展。  
  
当然，产品经理需要根据当时环境做出最佳决策。最佳的决策可能不是最好的决策。例如，一个网站一开始就要求完美拓展的架构，可以同时承载一百万人的后台服务，最佳的容灾策略……这是典型的脱离实际。产品经理需要看到几年后的发展，也需要关注眼下的情况。没有最完美，只有最适合。  
  
定义了基本的产品原则之后，我相信大部分产品经理都会从中受益。在接下来的产品宣讲和开发过程中，产品经理不需要多次重复产品方向，也不需要犹豫是否做成决策。因为产品原则就好像一个标尺，可以在产品经理做决策的过程中起到度量的作用。  
  
除了产品原则之外，产品经理最好定义一下每个原则的优先级。有的时候，虽然大家的方向一致，但是在实际的操作过程中，常常出现具体细节上的分歧。  
  
大部分时候虽然大家都尊重基本的产品原则，但每个人都会有自己的观点，他们认为自己的想法是正确的。不仅如此，每个人都从自己的视角出发，就会出现观念上的差异，尤其痛苦的是每个人都觉得自己是用户，而其他伙伴的方案都没有满足自己的需求。这时就体现了产品原则优先级的重要意义，如果一开始就已经把对应的优先级定义好，大家按照这个优先级来解决问题，或许会顺利很多。不过，在项目过程中会出现一些沟通上的问题，为了方案而争论是个好事情，可以把一些场景讨论得更加清楚——但请记住，无论如何，产品经理一定要在会议和讨论上进行有效地管理，按照《罗伯特议事规则》来提高会议的价值。产品经理要倾听，要快速找到争论的关键点并进行解答，及时把天马行空的伙伴拉回到现实，总结出一个最佳的方案。  
  
**关注市场发展**  
  
无论是什么样的产品，如果不推向市场，那就不能称之为真正意义上的产品。在日常的市场分析和用户调研中，常常会出现很有趣的一幕：产品经理经过一系列的调研，发现市场巨大，用户都十分欢迎自己的产品，而且好像没有什么竞争对手。这真让人兴奋！但实际上，永远不要相信没有竞争对手这回事，很多时候往往是因为你在搜集信息的时候忽略了一些关键信息，或者是调研的内容不够全面，没有及时发现竞争对手的踪迹。如果没有比较好的工具，产品经理对市场的预测和用户需求的调研时常会陷入自己设立的陷阱——夸大了市场。  
虽然大部分时候，产品经理不需要专职做市场分析的事情，但产品经理可以花时间去了解市场动向和用户，这是一件多么有价值的事情。QQ 空间的专家工程师Stone 是一个非常关注用户的人，他有一个有趣的习惯：时常会去百度等搜索引擎输入“为什么”三个字。因为大部分搜索引擎都有自动关键词智能提示，所以就可以看到以“为什么”开头的一些热门问题。通过搜索引擎的用户行为去分析市场和用户需求，真是一个让人称赞的做法！  
  
接下来介绍一下我了解的市场调研方法，可能会比较粗糙，但希望能抛砖引玉，供大家一阅。  
用户调研问卷：问卷虽然看起来是一种朴实无华甚至漏洞百出的方法，但因成本低、效果也还可以使其成为众多公司最喜爱的调研方法。使用调研问卷的关键是题目和题型的设计，要保证被调查者大部分时候都能按照事实反应，而不是随意填写，否则容易出现废卷。  
  
数据搜索：市场上有如此多的第三方咨询公司，他们的报告都是非常有价值的参考资料。如果有更多的成本投入，可以考虑长期使用付费的服务，随时方便地查询智库文档，可以从中找到许多优秀的报告。  
  
产品测试：Dropbox曾经做过一个有意思的试验：他们在产品开发之前先建立了一个网站，并通过一个概念包装吸引了种子用户，通过种子用户的数量，他们准确判断了云储存市场的前景。  
  
在Dropbox 早期的种子用户获取阶段，经历过两次用户小规模爆发阶段。一次是配合产品的演示视频加内测邀请的活动，在网站首页上面嵌入了一个演示的视频，一句鼓励人们留下Email 的文案，以及一个可以注册邮箱的输入框（LandingPage在这里）。配合着这个登录页面，以及之后在Hacker News 论坛上面发布的帖子，Dropbox 获得了第一波用户。  
  
Dropbox 的第二波用户来自 Digg。2008 年3 月11 日，在登上了Digg 之后，Dropbox 因为瞬间大流量进入，造成服务器宕机。而这一次的报道，为Dropbox带来的效果是超出预期的，登记排队的用户数在一天之内从5000 上升到了75000。  
  
如果从Dropbox 两年（已达到400 万用户数）的发展历程来看，这两次的效果仅是其用户数增长曲线上面一段小小的曲线。但是，从这两个网站（HN 和Digg）上面过来的用户，因为具备极客精神、勇于尝鲜的态度，以及可以接受不完美科技产品的宽容性格，成为了 Dropbox 在接下来的迭代和测试过程中的风向标。而更重要的是，在这个过程中，Dropbox 的团队左手抓用户数量的发展，右手抓产品功能的专注，通过有“节制”的定位，将产品的气质一直维持在一个微妙的平衡点上。——摘自ifanr

Dropbox 的试验非常值得学习。这种测试方法不仅有效地了解了用户对于产品价值的认可，并且可以很快速地评估市场机会和规模，cool！

**竞争对手分析**  
别小看竞争对手产品的作用，别人的失败或者成功经验，都是产品经理值得参考的信息。为什么他们会失败？是因为对用户需求的理解有问题，还是因为市场饱和了？是产品策略有问题，还是核心用户找错了？如果产品经理可以分析这些问题，就可以快速地找到竞争对手做得不好的地方，而这有可能就是市场的机会。  
  
还有许多有效的方法可以帮助产品经理分析市场、了解用户，但一定要了解产品的目标，善于发现机会。产品目标是多次谈及的话题，如果产品经理失去了目标，那绝对是糟糕的事情。发现机会则是在考验产品经理的嗅觉。善用信息的产品经理可以从信息中挖掘出许多有价值的机会。  
  
无论如何，了解你的用户才是最根本的，难道不是吗？  
 **对技术方案的关注**  
  
关注技术方案并非是要和工程师探讨方案的实现，也不是要越过边界对工程师指手画脚。产品经理关心技术方案是为了确认需求的实现，虽然已经进行了文档沟通和当面沟通，但有一些需求细节仍可能未沟通清楚，容易造成误解。如果找工程师了解技术方案，就能比较容易地达成一致的理解，确保后续工作有条不紊地进行。  
  
了解技术方案不仅能方便双方理解需求，更能方便产品经理及时预估风险。根据需求复杂度的不同，项目的开发资源常常得不到最好的支持，产品经理一开始就应该把所有相关人员都召集到一起，制定好基本的方案，确认每个环节都有人认领，大家都达成一致的理解。如果可以提前安排好排期和联调时间，那真是再好不过的事情。  
  
不过在实际方案的制定过程中，还会出现很多意外情况，最常见的一种就是如何对多个技术方案取舍，并确保每个工程师都认可。因为客户端和后台经常会有一些技术方案上的灰色地带，有些工程师为了减轻自己的压力，常常会推荐最佳方案之外的另一套方案。这时，如果产品经理了解技术方案就可以做出恰当的决策。  
  
如果产品经理对技术比较了解，和工程师交流时会更加方便，尤其是当自己希望加入一些锦上添花的元素时，可以和工程师一起讨论需求，以及如何将它做得更加完美——对于工程师来说，将产品做得足够有趣和有用也是他们所追求的目标。  
  
**PK！需求难产怎么办**  
  
制定了产品原则，了解了用户画像，沟通了技术方案，这时需求就可以直接变成文档交付给开发工程师了吗？在一个常见的产品研发流程中有一个阶段叫作产品需求评审。这种评审是有多方介入的，但最关键的是产品经理团队内部如何明确需求的方案和优先级。毕竟开发资源比较有限，产品版本规划需要有对应的目标。  
  
正如Marty Cagan在《启示录》一书中曾提到过的一个概念：产品评审团，其目标是更好更快地做出可靠的产品决策。腾讯亦有这种流程。产品负责人每周都会聚在一起，集中讨论接下来的发展计划和产品规划。在这个漫长的会议上，会诞生产品的版本规划和阶段性目标，同时明确Feature List 中每个功能的优先级。明确了优先级之后，产品经理就需要把需求转化成方案。大部分时候，需求变成功能方案要经历这样一个过程。  
  
1、产品经理和交互设计师先进行沟通，输出基本的方案。  
  
2、产品经理用交互稿和需求文档向上级汇报具体方案。  
  
3、召集开发工程师和测试工程师，正式宣讲需求，交互稿则交给视觉设计师进行设计排期。  
  
这三个阶段是产品经理遭遇PK 最多的时候，需要和交互设计师PK 哪个交互方案更佳，需要和上级讨论方案可行性，需要和工程师沟通需求详情和实现。写到这儿，我想起了电影《勇敢的心》中，梅尔•吉布森饰演的华莱士带领苏格兰人大吼：“Are you ready for a war？”  
  
  
当然这样说有点夸张，但是产品经理在方案细化的过程中的确需要不断地和别人PK。毕竟，产品经理最懂用户，最了解用户的需求点；产品经理应该知道用户喜欢什么样的设计，需要在设计感和实用性之间找到一个平衡点；产品经理可以和工程师探讨技术方案的实现，找到实现需求功能的最佳方案。  
  
因此，产品经理需要不断地和不同角色的人沟通功能的目标和效果。人与人的沟通是很有趣的事情，尤其是与设计师、工程师这种专业领域的伙伴，他们怀有专业的知识体系和理念，有各自的目标和审美，而产品经理如果缺乏决策力，很容易做出一个不令人满意的方案。如果产品经理缺乏影响力，则合作的伙伴们就很难接受方案。在专业度和用户体验方面进行权衡，是产品经理需求PK 的关键。  
  
对于产品经理来说，这三个评审阶段是非常有价值的。因此产品经理可以通过了解其他人的反馈来调整方案，并找到更好的解决方案。对产品经理来说，有效地结合其他产品经理、设计师、工程师的建议，并对原有方案进行优化、评审和PK，是非常有价值的。  
  
但产品经理一定要在评审之前就明确自己的观点、方案的可行性和坚持的理由。如果一个产品经理足够自信，在评审过程中就很容易获得其他人的认可，产生影响力。  
  
反之，如果产品经理在评审阶段依然没有足够的自信和对方案的足够熟悉，那么其他人会觉得你没有想清楚这个方案，或者是缺乏信心去说服别人采用自己的方案，这常常会引起工程师和设计师对于产品经理能力的不认可，减弱产品经理自身的影响力。  
  
随着产品评审流程的进行，产品经理常常会收获许多意见，有的来自老板，有的来自资深设计师，有点来自开发Leader，有的来自测试。参与的人越多，意见就越不容易统一，产品方案最后往往就变成了一个大家的产物，但谁对这个被修改多次的方案负责呢？  
  
产品经理不得不负责这个方案（满怀着大家的期待，多次争论的产物），然后去说服设计师，说服工程师……方案常常会变成一个大家都喜欢但是奇怪的方案。适度妥协方案是常见的现象，但如果产品经理对于别人的评审意见照收不误，最后出来的方案会变成四不像，但因为项目的要求开发工程师会尽快开发，谁知道后续还会出现什么“黑天鹅”事件，很有可能最后开发出来的效果不是产品经理想要的，也不是用户想要的。  
  
对于这个评审过程，我称之为需求难产。如果每次需求都遭遇这样那样的问题，如果每次需求最后都变成了众人评审妥协的方案，那么对于产品经理来说就是个灾难。我们可以做些什么来确保方案始终是为用户设计的方案呢？不如试一试下面的一些方法。  
  
●尽快确认产品原则，在评审之前就告知所有相关人员，确保产品经理和其他人原则一致。  
  
●产品经理要思考清楚自己的方案，以便随时应对别人的挑战。  
  
●往往在只有一个方案时（尽管是最佳的方案）最容易受到挑战。所以产品经理一开始就可以多考虑出几套方案。当评审人员了解了多个方案之后，自然会做出最佳的选择。  
  
●有效地引导评审人员认可产品经理最想实现的方案。对产品经理来说，最难过的一关就是老板的挑战。老板往往会有很多的想法和需求想要在这个方案中被满足，产品经理要学会控制老板的预期和需求，引导老板认可自己的方案。最简单的莫过于同时推荐几套方案，让老板自己选择。通过对比，老板会认可产品经理的方案。  
  
●提前和相关的设计师、开发者定好方案，避免他们对方案的调整一无所知。  
  
●最好可以提前和用户研究员一起沟通方案，了解在方案实现过程中有可能需要关注的数据。产品经理善用数据是专业度的体现。

来源： <<http://www.gameres.com/msg_301431.html>>

**说服力：从场景化出发的用户价值**  
  
“按钮应该放在右边，这符合规范！”“但是用户习惯在这边寻找按钮，放那边不够自然。”“这个可以做啊，但是用户会觉得很好用。”“有没有更好的方案，为什么一开始就要做那么复杂的操作呢？”  
  
这样的讨论总是在产品经理评审需求的时候出现。大家目标都很一致：为了用户和产品的发展。但每个人的视角或有不同，这时，产品经理就应该具备上帝视角。什么是“上帝视角”？就是指产品经理不仅可以看到主流用户的需求，还可以看到其他伙伴各自的出发点，可以和他们进行良好沟通，推动产品实现。前文曾谈到为什么需要产品经理这个职位，我认为，产品经理在整个项目中不仅需要对需求负责，更要成为伙伴之间的“润滑剂”——可以和产品经理PK 需求，也能和开发人员探讨技术方案的实现，还可以和设计师聊一聊最新的设计风格，最关键的是产品经理需要了解用户。  
  
但事实上，谁都觉得自己更了解用户，觉得自己本身作为一个用户没有被满足需求。这种情况时常出现在每次的评审中，这就要求产品经理可以做适当的沟通。前文出现的激烈争论并非是坏事，对于需求来说，可以越辩越明，但对于决策来说，却不符合呆伯特会议规则，无法说服其他伙伴同意产品经理的方案。还有一种可能的情况，即产品经理强势地要求开发工程师或者设计师按照自己的理解去进行，通过“项目时间”和“老板需求”来强迫他们“心甘情愿”地实现需求。这种“狐假虎威”的做法有时候有效，但对于产品开发来说，并不是一件好事。  
  
在产品开发过程中，如果相关人员都没有被产品经理说服并同意产品经理的方案，就容易产生情绪——而情绪化的结果则是潜伏在产品开发过程中的各种风险都会不断出现。例如，开发工程师没有准确地理解需求，马马虎虎地完成功能，忽略了相关逻辑和细节实现，或者将这位产品经理的需求优先级放低，不断拖延……这种情况很常见。我私底下和一些工程师关系不错，有时候就会问他们，为什么有的需求完成度很低，有的时候需求bug 会特别多？他们会回答：“不想做或者有其他优先级的事情。如果充分理解了需求，我们也是很乐意去做的。但是平时产品经理都不和工程师一起玩，没有太多的革命友谊，所以大家都是公事公办，公事公办的后果就是先做其他事情。”  
  
产品经理们，去施展你们的影响力吧！准确地阐述用户需求和价值，让伙伴们认可你的方案，这对于方案的实现有着重大的意义。用户需求时常被挑战的重要原因之一就是缺乏场景，大部分时候把自己当作用户并不能很好地反映客观情况，需求往往是伴随着场景而变化的。正如《探索需求》中所言：“除此之外，人们并不经常购买他们所需要的东西，却常常追求他们并不特别需要的东西，即使这种期望是短暂的。”  
  
场景化需求才会产生“期望”，而不考虑场景提出的需求，虽然客观存在，却不一定是最想要和最需要被满足的。因此，产品经理在面临日常的方案挑战时，一定要关注将需求和场景结合起来，如果可以考虑用户的使用习惯和行为数据，则会更加具备说服力。  
  
**第一次做功能——从失败到成功**  
  
第一次做功能，对于许多产品经理来说，意义非凡。但对于一个入门者来说，第一次做功能往往需要人进行辅导。在腾讯内部，导师一般会安排刚入门的产品经理承担一些简单的需求，并会进行对应的指导。即便如此，看过本书前几章的产品经理可能依然会有点茫然：“虽然我都知道该怎么做了，但是如何开始第一步呢？”  
  
许多人在产品分析时表现得头头是道，但一面临实践操作往往就发现，理论再靠谱，也难以应用到现实中来。记得我第一次跟进的需求是做一个阅读文章的功能。当时我画了思维导图，又用PPT 做了几个线框图做说明，还写了Word 版本的需求文档。但事实上，这几个东西都没有起到太大作用——因为每个产品经理做需求的第一步，不是动手去做，而是思考。  
  
**第一次做功能：产品设计阶段**  
  
三思而后行，这是我的个人经验。如果一个产品经理在接到任务之后，不假思索地就开始要各种资源，拉设计师和工程师讨论需求，肯定会受到各种挑战和不信任。任何一个需求都应该被拿来好好思考，清楚了整个流程，考虑了各种情况都会有哪些效果，产品经理的心里才会有底，才能有效地传达需求给其他人，才可以更快地推进需求实现。  
  
当时，我接手了阅读文章的功能，于是马上找到设计师传达需求，就遇到了类似的问题。  
  
产品经理：我这边有一个阅读文章的需求，你能帮忙看一下吗？  
  
设计师：要阅读什么文章，是一个怎么样的功能？  
  
产品经理：就类似Google Reader。  
  
设计师：Google Reader 有哪几个页面，大概流程你清楚吗？这个功能的目标是什么？  
  
产品经理：目标就是为了看文章，和Google Reader 一致就可以了。有很多的订阅文章，然后把文章排个版，加入微博分享、收藏这些功能。  
  
设计师：是需要和Google Reader 一致吗？但是我依然不清楚阅读文章是要干什么，是为了收藏？  
  
产品经理：……  
  
很显然，产品经理对于需求的理解很模糊，没有具体的目标。虽然从入门开始就一直被灌输用户第一的理念，但是一接到需求就什么都忘记了。对于一个需求任务，如果产品经理未能很好地进行分析和界定，一开始就找对应的人员进行沟通，或者简单地把方案类比为其他产品，是一种很不专业的做法。  
  
在进行了详尽地分析之后，我重新调整了需求功能点，通过思维导图展示了基本的需求说明（参见图3.4）：

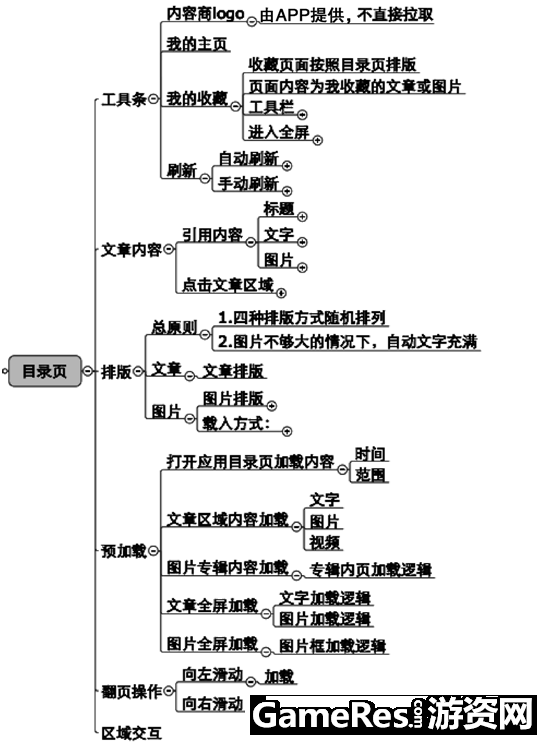


图3.4 阅读产品的需求列表展示

粗略看起来，好像这个功能点覆盖得很全面，但让人很难理解，这个功能究竟是什么样的？看起来好像有一个目录页和一个全屏页，但页面内的效果该如何展示呢？目录页中的几个子选项——工具条、文章内容、预加载、翻页操作和区域操作——究竟是什么关系呢？  
  
当时的我还没有意识到一个问题：思维导图并不适合说明需求，而更适合整理需求点——而整理需求点，是产品经理自己的作业，拿出来给工程师和设计师看，谁都会很困惑吧。于是，费尽了功夫的我获得了两个教训：  
  
选择工具很重要。  
罗列功能并不是需求文档。  
  
经历了现实的打击之后，我重新调整了需求文档，将改用Word 模版的需求文档交付给对应的设计师。  
  
**第一次做功能：开发阶段**  
  
没过多久，交互设计师就输出了基本的交互稿，和我的想法是基本一致的，于是我开心地找工程师评审需求了。然后又遇到了第二个问题：那些轻描淡写的“自动刷新逻辑”、“页面排版逻辑”究竟是什么玩意？  
  
工程师对此表示很遗憾，他们找遍了需求文档也不知道要怎么做，于是我给他们又进行了一次方案宣讲，并把对应的内容都写入了需求文档中。对于工程师来说，他们的目标就是准确地实现需求文档的要求。因此越清晰的要求，对于他们来说越省力。所以入门的产品经理需要关注这个案例：尽量用详尽的文字去描述每一个细节。  
  
**自动刷新逻辑**  
  
触发条件：当用户进入该页面，触发自动拉取最新数据的操作。  
  
刷新展示：如果刷新成功，则页面展示最新内容（渐显效果）；如果刷新失败，则弹出“刷新失败，请稍后再试”的提示。  
  
图片展示：未拉到图片时，需要展示一个占位图，在一分钟内进行多次图片拉取；如果图片拉取失败，则出现一个裂图；超过一分钟，允许用户手动点击占位图进行拉取图片的操作响应。  
  
每篇文章的摘要显示及排版：××××××××  
  
如果由于网络原因数据拉取失败，需要提示网络错误。  
  
如果由于网络比较慢，则在一分钟内多次尝试拉数据；超过一分钟则等待用  
  
户手动触发刷新操作。  
……  
  
以上只是展示了正常的逻辑，还有许多异常情况没有考虑，因此产品经理在描述需求时，切记把每一个细节都考虑到，尤其是异常情况。对于工程师来说，没有歧义且包含多种场景的需求描述，才是好的需求文档。“自动刷新”虽然是简简单单的四个字，却包含了有可能四百字都描述不完的细节。  
  
**第一次做功能：产品体验及改bug 阶段**  
  
只有真正参与过项目，产品经理才会了解人与人之间的沟通是多么重要。如果仅仅依靠需求文档，产品经理可能会发现开发工程师实现的功能和当初构想的功能会有许多差异。先不要担心，这些差异的存在是必然的，毕竟开发工程师不会读心术，而文字描述本身就容易出现歧义，这些差异都可以在体验阶段修改。  
  
但这个事情从侧面反映了一个问题：需求文档还是不够细致。  
灾难到这里就结束了吗？产品经理体验了产品之后交给测试工程师测试，这才是产品经理发现自己无知的时候呢！测试工程师会根据测试用例提出许多异常情况，而产品经理会发现，怎么这些场景我都没有考虑到！怎么会有如此多的bug！  
  
先别着急，这些问题几乎在以后的产品开发过程中都会遇到。但这些“突如其来”的灾难的确容易让刚入门的产品经理手忙脚乱。有条理地去解决这些问题吧！产品经理之路还远着呢！  
  
虽然第一次做产品需求就像个灾难，但后来回首这样的经历，可以发现一个有趣的事实：尽早失败，尽快改正，是产品经理成长的必经之路。而在我往后的产品经历中，往往都会想起这样一句话：把功能想透彻。对于产品经理来说，需要想清楚以下几个问题。  
  
●Who：用户是谁，他们有哪些特征？  
●What：这个功能具体是怎么样的，确认需要做哪几个功能？  
●When：什么时候会使用？  
●Where：使用场景是什么，功能页面之间是什么样的关系？  
●How：用户如何正常使用对应的功能？  
  
什么叫作“想透彻”？不如请各位产品经理先想一想如何用一句话准确地表达手头的需求，如果不需思考就能脱口而出，那说明你已经在思考，反之则需要扪心自问，自己是否已经了解该需求。如果想要检测自己是否想透彻了，还可以通过以下的几个问题进行自测：  
  
●用户的需求是什么？  
●这个需求可以分解成多少个小点？  
●这些小点有哪些可以满足，哪些不需要满足？  
●每个需求点可以通过什么样的功能去满足？  
●这几个功能之间的关系是什么？  
●整个方案有几个页面，和整个产品的关系是什么？  
●功能的入口放在哪儿？  
●用户发现了这个功能之后，他们会怎样使用？  
●如果用户中断了操作，会出现什么提示？  
●是否针对功能操作设置了保护机制？  
  
这些只是最简单的一些问题，接下来我会结合自己做产品的经验及一些朋友的经历举出一些常见的问题，希望可以给刚入门的产品经理提供更多启示。  
  
**常见的问题案例**  
  
**问题一：没有分解需求。**  
  
每个人对于需求的理解都有所不同。产品经理在描述需求的时候，需要尽可能详细和无歧义。例如：  
  
●我想在坐电梯的时候可以获得资讯。  
●我想在坐电梯的时候可以获得关于股票的消息。  
●我想在坐电梯的时候可以了解到公司股票的实时走势图和实时价格。  
  
各位觉得哪个描述更容易被理解，而且没有太多含糊的信息呢？不仅仅是描述容易出现问题，大部分时候需求描述有问题就是因为需求没有得到有效地分解。产品经理还需要注意对需求进行分解，像严谨的解构主义者，不放过任何一个概念。越是分解细致，越容易降低含糊信息出现的几率。  
  
**问题二：缺乏对用户的了解。**  
  
虽然在前面一直提到以用户为中心的产品理念，但因为某些原因，大部分时候产品经理不得不尽快明确需求，而等不及对用户进行了解。这种闭门造车的需求很容易产生一个可怕的后果：用户不买账。历经千辛万苦做完一个功能，产品经理没有功劳也有苦劳，但用户不买账。这就是缺乏对用户了解的后果，即使产品经理耗费大量资源和时间，但是做出了用户不喜欢的产品，那么一切都是在做无用功——即使产品经理自己觉得在做创新，用户不买账是因为他们视野不够，但不被用户接受就失去了产品的使用价值，不符合市场的供求机制。  
  
**问题三：急于推动方案，而非获得支持。**  
  
因为刚入门的产品经理往往缺乏自信和魄力，所以很难得到工程师和设计师的信任。但由于产品经理希望尽快推动功能进行开发，所以会不断地催促设计师和工程师动手做需求。  
  
做需求本身是一个合作的过程，如果忽略沟通，不管需求的背景和目标，而仅仅是传达任务，那么对于设计师和工程师来说，这是一种缺乏合作精神的行为。虽然产品经理是无授权的领导，但在日常需求合作的过程中，产品经理更重要的角色是“润滑剂”，即沟通各方人员，确认他们都了解需求的背景和目标，保持信息一致。  
  
得道者多助，失道者寡助。获得了设计师和工程师们的支持，对于产品经理来说是好事，因为这意味着你们在同一条船上，可以一起朝目标前进。而失去支持的产品经理，时常会遭遇到不合作，或者不认真的合作，这种情况下,需求在实现过程中常常会出现恶性循环，导致需求总是返工，每个人都不好受。  
  
**问题四：容易被忽悠。**  
  
不懂技术的产品经理时常会被忽悠。这种情况并非开发工程师不合作，而是他们在暗示对产品经理缺乏信任。  
  
●你自己根本不清楚需求要怎么做。  
●你根本不了解我的工作很多，总是打扰我。  
●和你沟通很费力，忽悠你赶快走！  
  
刚入门的产品经理缺乏经验，总是容易被各种术语忽悠——想要改变这种情况，那就要让工程师们信任你。我常用的方法可以分享给各位试一试。  
  
●复杂的功能如果已经在其他应用上实现了，工程师们二话不说，拼了自尊也会去实现的。  
  
●如果他们冒出大量术语，你可以试一试请教他们这些“牛逼哄哄”的术语都是什么意思，他们愿意介绍这些“牛逼哄哄”的术语。  
  
●不管如何，确认他们对需求理解到位，必要的时候可以让他们重复表述一遍。  
  
其实以上几个问题都非常容易出现，但大部分时候都可以归结为一个概念：含混性。什么是含混性？产品经理对用户的需求理解出现偏差，工程师对产品需求理解出现偏差，设计师对产品需求理解出现偏差，这些都是含混性的表现。

来源： <<http://www.gameres.com/msg_302170.html>>

**如何避免含混性出现**  
  
　　含混性，指的是不清晰的程度。在项目开发过程中，含混性会一直贯穿整个过程。从需求出现到把需求变成产品，含混性始终影响着整个项目。因为含混性的存在，产品经理必须注重沟通和理解，尽可能准确地探索需求和传达需求，而工程师们则需要尽可能理解需求，在产品诞生之后，产品经理还需要关注用户需求是否发生了变化。简单地说，含混性越大，项目风险也会越大。著名思想家温伯格提到一个计算含混性的公式：  
  
　　含混性 = 实现需求的最大花费/实现需求的最小花费  
  
　　这个公式在项目评估的时候非常有效，尤其是技术评审阶段。不同的工程师会有不同的工作量评估。如果含混性越大，则这个工作量评估的差距就越大，说明产品经理传达需求的过程越有问题。反之，则说明大家基本保持了理解一致。  
  
　　那么，我们为什么要避免含混性的出现？因为含混性意味着产品有可能被用户遗弃而造成失败，意味着在开发过程中会出现大量成本，尤其是沟通成本和开发成本。从信息论角度来看，这些含混性就好像是大量的噪声，会影响人接收到正常的信息。把“信噪比”的概念引用到产品设计中，其实是一个比较有趣的方式。  
  
　　所谓的讯息就是为了沟通而产生的，而使用者界面就是承载着资讯的载体。  
  
　　使用者透过界面和各式各样的系统进行五花八门的资讯交换。资讯可能由使用者产生，例如某个人在 Twitter 上发了一则讯息，通过界面上传到网络系统，然后通过网络传递到所有人的荧幕前，然后再由其他阅读到这则讯息的使用者接收。  
  
　　因此在资讯的生命周期中，包括了产生、传递、接收这三个重要的阶段，而每个阶段都有可能造成资讯的损耗。而传递中的资讯又可以分为“真正有用的资讯”和“造成干扰的杂讯”。举例来说，一张讨论各国人口数量的图表，如果采用了过多且花哨的装饰或特效，这些装饰和特效就会成为资讯的“杂讯”。  
  
　　如果要将讯息完整地传递给使用者，设计师可以选择强化原有的讯息，或是降低多余的杂讯，来提高“信噪比”（Signal-to-Noise Ratio）以增加讯息成功被判读的几率，也让使用者能更轻松地阅读资讯。——摘自陈威帆的《台湾设计师谈资讯与视觉设计的绝妙平衡》  
  
　　想要有效地避免含混性的出现，应该尽可能地通过一些方法保证信息的完整性，避免出现过程中的损耗。  
  
　　比如我在技术需求评审阶段总是会让技术人员重新复述一遍需求的细节和目标，确保我们理解一致。这种做法很有效果，尤其是在降低含混性上是非常有效的。正如许多人很喜欢看书，一年可以看好多书，但总觉得自己没有长进。实际上，当你要他复述这些书的内容时，他都不知道作者的主要观点和推导的过程，只记得自己看书的时候觉得这本书很好看，而不知道为什么好看，为什么引人入胜。反之，通过复述书中的内容，其实就是对通过阅读吸收的知识进行重新理解和思考的过程，记忆会更加深刻，而且对于信息的损耗也会降低。  
  
　　另一方面，在需求评审过程中，不一致的情况时有发生，这其实也是含混性的外在表现。有的开发人员说这个功能10 天就可以实现，而有的开发人员则说5天就可以实现，这里含混性的比值就是2：1，说明存在许多不一致的情况，需要重新理解并达成一致。所以，需求评审的意义其实在于控制风险，降低含混性的出现。  
  
**如何定义需求和功能**  
  
　　我们往往发现看了再多书也抑制不住跃跃欲试的心情，于是遭遇了第一次失败的经历。那么，我们真的可以开始聊聊“需求”这个伴随产品经理整个职业生涯的名词吗？我相信大部分产品经理以后的口头禅会有这样几种：

* 用户想要……
* 这体验太差了！
* 需求不合理！

　　关于需求，我们前面已经聊了许多，但往往第一次做需求的时候，会发现自己手足无措。如何对需求进行管理？如何排需求的优先级？需求之中还有许多需求，如何分解？当我第一次接到需求任务时，我和各位一样突然发现自己一无所知。阅读是需求，那么什么样的人是我们的用户，需要什么样的阅读内容，需要怎么样的阅读方式？我当时的脑子里充满了各种问号。  
  
　　阅读产品可以复杂如Google Reader，也可以简单如Instapaper；内容可以像Flipboard支持订阅，也可以像Zite一样支持推荐。我究竟要把这个功能做成什么样呢？是面面俱到，还是尽量满足基础需求即可？是囊括各种最新潮的交互方式，还是只包含人们最常用的交互方式？我思索了一天，直到路过星巴克才发现我现在在面临什么样的问题：切入点是目前最关键的问题。找到切入点之后，我才知道这个功能应该做成什么样，与其他产品的差异在哪里。正如星巴克有如此多种类的咖啡，因为调制方法不同而产生了许多品类（参见图3.5），而这些品类都有各自的拥趸，这就是切入点和竞争力。



                   图3.5 不同比例咖啡搭配的配方

　　即便如此，我们喝完了拿铁和卡布奇诺，依然不知道为什么这两种看起来配方类似、只是比例不同的咖啡会成为咖啡调制历史上重要的两款产品——最有趣的是，我喜欢拿铁，而我的朋友喜欢卡布奇诺、摩卡等品种。那么，同样的阅读功能是否也有这样一个“比例”可以让产品经理找到真正的切入点？  
  
　　约翰•冯•诺依曼曾说：“如果根本不知道自己在讨论什么，那么对其强求精确则毫无意义。”对于产品经理来说，如果不知道自己要做什么样的需求，那么强求一个方案也是毫无意义的。经过和其他产品经理讨论，以及对用户进行一些了解之后，我发现自己好像陷入了一个坑：阅读这个需求可大可小，如何控制这个功能的广度和深度才是这个需求的关键。如何控制自己的欲望，如何避免为了做功能而加上一些花哨的功能，是产品经理时常会纠结的问题。  
  
　　无论如何，让我们回到“切入点”的问题。通过与用户进行沟通，我知道了他们希望有一个可以快速获得对应资讯内容的功能。  
  
●场景：日常无聊的时间可以看对应的资讯消遣休息一下。  
●需求：了解发生了什么事情，有什么好玩的，可以适当有些深度。  
●竞品：Zaker和Flipboard。  
  
　　我特意把竞品列出来，是因为竞品的出现让我看到了找出“切入点”的一丝希望。在用户调研过程中，大部分用户会通过自己的使用经验去描述需求——对某个产品进行举例就是他们常用的方法。对于用户来说，他们感性地用经验去解释自己的需求，是非常合理的；对产品经理来说，用户通过某个产品去描述，是把需求具象化的有效手段。尽管这个具象化有可能出现偏差，但是对产品经理来说至少找到了一扇门，也许默默说声“芝麻开门”就能获得灵感。  
  
　　于是，结合用户的需求和对竞品的分析，我获得了一手资料，而需求也开始明晰化，产品原则在心里慢慢浮现。

1. ●原则一：快速地阅读，丰富内容和类别。
2. ●原则二：减少自定义的操作，仅仅支持订阅来源。
3. ●原则三：文字越少越好，内容趣味性越强越好。

　　但这只是基本的产品原则，与实际开发过程中参考的需求文档还是少了很多细节的，尤其是面临一个问题：产品原则仅仅是缩小了需求的范围，更像是一个边界，而具体的产品方案还没有基本的雏形，我们还不知道用户的需求具体是什么，该如何满足。就好像用户告诉大众汽车公司，需要一个更小更便捷的经济用车，于是大众推出了经典的“甲壳虫”系列。那么，是否可以把阅读的需求也做一个类比？假设把阅读的需求比喻成开车，用户需要什么样的车子呢？按照产品原则，我们可以继续深入。

1. ●原则一：允许个性化定制。
2. ●原则二：操作便捷。
3. ●原则三：驾驶充满乐趣。

　　针对这三点，大家心里应该都有几个基本款：MiniCooper、Smart、Beatles。这几辆车的共同点是针对年轻人，但每辆车各有特点。于是大家心里都明白用户想要的大概是什么样的东西了。  
  
　　现在把比喻去掉，是否有一些对应的阅读产品可供对比和分析呢？在接下来的头脑风暴会议上，大家各抒己见，针对对应的产品原则提供参考的产品。最后发现Flipboard、Zite、Feedly、Digg 和微博（最有意思的是，微博出现在了这次的分析中）出现在参考列表中。看来通过比喻找到具象化的产品，的确可以详细地理解需求。  
  
　　于是在对比了几个产品之后，我们也逐渐清晰了想要的阅读功能切入点是什么：快速阅读和丰富的个性化内容。  
  
**功能与属性**  
  
　　既然已经找到了切入点，就好比找到了一根精灵绳索，可以把佛罗多拉到山顶，远眺前方正是旅途的终点——摩多。但我们依然没有拿出用户真正想要的东西。或许我们可以根据该切入点制作一些纸上原型进行用户调研，但正如之前的章节所说的，用户对于这些原型很有可能言行不一，起不到真正的效果，但我们可以通过用户对原型的描述来发现一些有趣的需求描述。

1. ●我平时看资讯一般都喜欢有图片的。
2. ●我平时玩微博，但是感觉其中的内容不是很好玩。
3. ●功能能否强大一些。
4. ●最好可以把这个资讯分享到微博等地方。
5. ●我平时用很多软件，但是没有一个可以换肤，能不能出一个类似的功能，更保护眼睛。
6. ●图片最好是高清的，打开可以连续看。
7. ●……

　　最后发现我被用户的一个小问题难住了：这个功能挺好的，但是我可以用这个来做什么呢？最后我解释道：你可以认为这是一本杂志，但是图很多，文字很少，而且杂志的内容都是你自己想要看的。  
  
　　这个时候，我才反应过来——这个产品本质上应该是做什么的，而产品经理应该打造一个什么样的产品：个性化的图片杂志。  
  
　　产品经理在进行需求分析的时候，常常会忽视一个小问题：如何向用户解释这个功能或者产品——用户可以拿来做什么？大部分时候，产品经理都会说：“我们这个产品有很多很多功能啊，可以换肤，可以提醒……”但实际上，用户只需要一句话解释就足够了：通过这个产品我可以做什么。  
  
　　正如上面用户在调研过程中提出的一些问题，其实都暗示了一个潜在的逻辑：我觉得它是一个什么样的功能，所以我可以做什么，想要什么。按照《探索需求》一书中的说明，温伯格认为对这些用户的描述都可以进行归类，而归类之后就可以更好地了解真实的需求是什么样的。  
  
　　●明显的功能。  
　　●隐藏的功能。  
　　●装饰性的功能。  
  
　　通过温伯格的方法对上面用户的反馈结果进行分类，则可以看到该方法的魅力。  
  
　　产品A 方案：  
  
　　●看资讯消息（明显）。  
　　●想要分享到微博（装饰性）。  
　　●图片最好是高清的（隐藏）。  
  
　　产品B 方案：  
  
　　●看资讯消息（明显）。  
　　●想要分享到微博（隐藏）。  
　　●图片最好是高清的（装饰性）。  
  
　　产品C 方案：  
  
　　●看资讯消息（隐藏）。  
　　●想要分享到微博（装饰性）。  
　　●图片最好是高清的（明显）。  
  
　　产品D 方案：  
  
　　●看资讯消息（隐藏）。  
　　●想要分享到微博（明显）。  
　　●图片最好是高清的（装饰性）。  
  
　　产品E 方案：  
  
　　●看资讯消息（装饰性）。  
　　●想要分享到微博（隐藏）。  
　　●图片最好是高清的（明显）。  
  
　　产品F 方案：  
  
　　●看资讯消息（装饰性）。  
　　●想要分享到微博（明显）。  
　　●图片最好是高清的（隐藏）。  
  
　　通过这个方法似乎很简单地就看到了对应的产品特征和方向。  
  
　　●A方案：注重看图文内容，尤其是文字资讯。  
　　●B方案：注重分享资讯。  
　　●C方案：注重看图及部分资讯。  
　　●D方案：注重分享资讯。  
　　●E方案：注重看图并分享。  
　　F方案：注重分享图片。  
  
　　经过和其他产品经理讨论，最后这个阅读功能被定义为C 方案，所以在内容上，我偏重选择以图片为主体的资讯，满足用户看图的需求。于是这个功能就变成了阅读图片资讯的功能。

来源： <<http://www.gameres.com/forum.php?mod=viewthread&tid=306910>>

**编者注:**第一部分章节发布之后，许多同学都在后台提出了许多好问题，我们会从中选取一部分，邀请胡澈进行回答，请各位继续关注我们的连载。  
  
本章主要讲解了需求是如何诞生这个话题，欢迎分享到朋友圈并积极讨论。

衡量体验与符合期望

既然已经定了产品方向，那就可以写文档进行开发工作了。虽然是敏捷迭代，但依然要付出两个星期时间去开发产品——假设产品研发完成之后，不符合用户的预期怎么办？

在日常的项目阶段，开发结束之后都会进入一个名为“产品体验”的阶段。在这个时期内，产品经理会不断地针对产品进行使用并优化方案。**在这个时期内，常常会遇到几个问题：**

●开发完成的功能和需求文档有所出入。

●有一些边界数值问题没有进行限制。

●流程体验不是闭环。

●有些真实操作的体验还有点不爽。

产品经理注定是解决问题的人。产品体验的阶段是针对产品进行最后调整的时期，产品经理不仅需要确保产品的正常交付，还需要确认用户需求是否发生改变，市场是否出现相应的变化。很有可能在开发阶段已经有其他相似的产品提供对应的功能，或者是用户有了新的途径去满足对应的需求。不仅如此，产品是值得打磨的，尤其是用户体验——什么叫用户体验？用户体验是用户在使用产品时候产生的感受。

张小龙曾说，产品要满足用户的“贪嗔痴”。这个描述很准确，大部分时候人对于产品都有一个期望，而能否满足他们的期望，就是产品体验好坏的衡量标尺。如果要用一个公式去表达体验好坏，可以这样看：

用户体验 = 实际操作 - 用户期望

实际操作带来的感受大大超出期望，那用户体验则是好，反之为不好。如果要让产品实现最佳的体验，产品经理不得不做两件事情：

●提升实际操作的“快感”。

●把控用户的期望。

提升实际操作的“快感”非常考验产品经理的功力。我认为，提升实际操作的体验取决于这样几个因素：

●便捷性

●反馈性

●感官性

便捷性是指用户在使用过程中总会有一个使用的目标，我们可以让他最快最方便地达成目标。对大部分用户来说，他们缺乏对于流程化概念的理解，不会对烦琐的流程和操作有太多的耐心，因此少一步操作，就可以留住更多用户。众所周知，汽车一般分为手动挡和自动挡。在自动挡没有出现之前，许多人认为自己学不会驾驶汽车的能力，什么时候要踩离合器，什么时候要加挡，什么时候要空挡……如此多的判断就好像一个一个复杂的流程，让人望而却步。自动挡汽车出现之后，人们的心理门槛就降低了，之前觉得自己学不会的人也开始试图学车，因为自动挡很简单。只要你把挡位调整为D 挡，然后就只需要踩油门和踩刹车即可。这一功能大大简化了操作，带来了便捷性，提升了驾驶体验。产品经理应该注重化繁为简的能力，让操作更便捷，这一点必不可少！

反馈性是指用户在使用中，每一个操作都要有对应的反馈。比如，按下按钮需要有一个与平时状态不同的点击状态，操作失败时要有错误提示，游戏产品中要有虚拟道具奖励用户，这些都是反馈的作用。我曾经上过内部的一个测试课程，发现测试工程师对于反馈和体验有着丰富的经验。例如，从Windows 的启动界面、传文件进度条提示和Photoshop 的启动界面，都可以看到反馈的重要性。对于用户来说，Windows XP 系统的启动界面（参见图3.6）绝对是再熟悉不过的——但大家看到这张图片后能否告诉我还有几秒钟，Windows 才能正常启动进入桌面呢？乍看之下没有太多方法，对于用户来说，这个图基本就是没有反馈的，没有提示可以告诉用户Windows 要启动好了。如果机器性能比较差劲，有可能会在这个界面一直卡住。

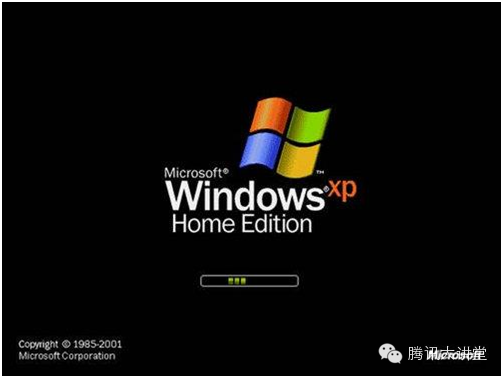


图3.6 Windows XP 系统开机启动中出现的加载画面

但是，真正熟悉Windows XP 系统的人应该都明白一个“潜规则”，这个进度条走满5 次就可以进入下一个启动界面了。所以，了解的人可能会通过这种方式去判断开机速度，而大部分人可能都不知道自己的电脑什么时候可以准确启动。

如果遇到一些不明白这个“潜规则”但是急需要使用电脑的用户，或许早就恼羞成怒了。所以，明显的反馈是非常重要的。

相比之下，Windows 7 系统的传文件界面（参见图3.7）就做得非常好，它准确地描述了传文件的情况：速度、剩余时间等。如果性急的用户看到这个传文件的过程总共要花大约2 分钟时间，那他可以准备点爆米花、上个厕所，轻轻松松地度过传文件的等待时间，而不是焦虑地等着——那会让人很不爽。

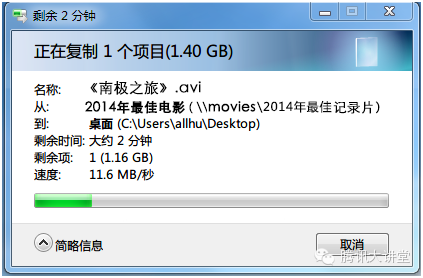


图3.7 Windows 7系统中移动文件出现的传文件窗口

感官性是指通过界面变化来示意用户操作，或带来视觉上的体验。例如，日本的一个名为Yoritsuki的APP（参见图3.8）绝对称得上是艺术品。因为它让人觉得打开这个APP 就会带来美的享受，让人愉悦。这就是感官性对于产品的意义。



图3.8 日本知名的iPhone应用，通过设计庭院展示禅意

把控用户的期望是产品经理常常面临的问题。为了推广产品，产品经理大部分时候必须通过各种方式去宣传产品的易用性、强体验等，但如果宣传过度则有可能会引起用户期望太高，而等到新产品推出之际，发现产品使用低于预期，这时常会造成一些负面的效果。因此，产品经理不能把用户的期望提得太高，而应该考虑如何避免产品和用户的期望脱节而产生明显的心理落差。

iOS 7 的发布就非常有意思：发布前大家都对iOS 7 非常期待，库克在发布会上发布了之后，许多用户开始吐槽对应的界面设计，下载beta 版的用户都开始抱怨各种bug。正式发布了之后，之前强烈反对的一些用户又觉得这个系统在手机上看起来还挺顺眼的。

用户的期望是一个很难琢磨的东西，大部分时候产品经理都很希望有一个测量仪，可以准确地把用户的期望量化——显然，这个很难做到。或许我们可以使用用户调研的方法去判断用户的期望：用户的行为和态度表达了对应需求的期望值。产品经理可以考虑进行灰度测试，看看用户的反馈和需求是怎么样的——通过这样的方法，搜集用户的反馈内容并提炼出关键词进行优先级排序，这样可以看出一些问题。

当然，这个时候不能忘记了重要的决策工具——人物角色。在之前通过人物角色进行用户调研的时候，我曾提到人物角色应该贯穿整个项目决策，并成为有效的依据。通过核心人物角色进行描述，并结合真实的用户访谈结果，可以有效地了解用户的预期。

在写作本书期间，我突然迷上了看电视剧。之前我从来没有关注过太多电视剧，觉得它们剧情又臭又长，需要花费大量时间，而且人物刻画很糟糕——但是在我迷上看电视剧的这段时间里，我发现一个很有趣的现象：人物角色的原理可以运用到电视剧中，而理解电视剧（尤其是情景剧）人物刻画和剧情推动的方式可以反过来更好地理解人物角色在互联网产品中的作用。

通过版本迭代把控用户预期

**说到看电视剧，我时常听见身边的人在吐槽：**

●《纸牌屋》第二季怎么还没有出来呀？

●《生活大爆炸》还演得下去吗？

●最近都没有什么剧可以看，如果重新看一部电视剧，又要追剧，好累……

**换个思路，产品经理是不是也经常看到用户反馈：**

●夜间模式怎么还不出来啊？

●上次提到的增加管理功能都等了好几个版本了！

●说好的高清图片呢？我才不想要为了省流量的小图。

因为开发成本及版本迭代计划等原因，大部分时候产品经理不得不背着“骂名”推动每个功能的排期。比如，夜间模式和管理内容这两个功能，如何判断优先级呢？在日常产品需求排期会议上，时常会遇到下面这样的对话。

**产品经理：**好像夜间模式反馈的用户比较少，不如我们先做管理功能吧？

**项目经理：**本次需求排期要优先考虑几个大功能，开发人力不足以支持再去做这两个功能。

每个项目计划都会遭遇到很多变故。随着市场变化和用户需求的变化，产品计划也需要有所变更，可能彼时用户还在反馈某个功能不好用，此时就再也没有用户吐槽这个功能了。假如把产品迭代看成美剧，就可以发现这些编剧对于节目的编排值得借鉴，以《生活大爆炸》为例。

●场景：场景布局基本在室内，且不做变化。

●演员：每个人物角色都非常饱满——社交能力很弱的技术宅、性感的金发女主角、有点萌的女博士、努力融入美国文化的印度人和犹太人等。

●剧情：每个人的个性都很突出——冷笑话及冲突大部分发生在谢尔顿身上；关于种族的剧情总是出现在霍华德和拉吉身上；爱情故事主要由佩妮触发；而莱纳德是最圆润的人，大部分时候都是在扮演和事佬的角色。

●发布日期：每一季之间大概间隔半年到一年时间；每一季内的片集则是每周三准时播出。

如果把每个剧情都看成是一个版本迭代，产品经理就会发现美剧虽然迭代时间比较慢，但是用户的抱怨比较少，而且每次新的一季内容一出现，大波人马上开始观赏并津津乐道。对于用户来说，提前知道了产品迭代的周期就容易消除期望的起伏。例如，小米公司的MIUI 形成了一个有趣的风格：每周五都会更新一个版本。

MIUI 为什么要每周五发布？这个项目开始的时间是2010 年4 月底，项目负责人是黎万强，工程师是我和刘新宇，然后就没有其他人了。新宇负责电话相关的改动，我负责系统升级，要保证可以在线升级（OTA）到下一个版本。两个人非常兴奋，也非常努力，畅游在Android 代码的海洋里面，很顺利地在劳动节放假前发布了第一个内部版本。

那一天，2010 年4 月30 日，正好是周五。在那之后，公司内部的版本总是在周五发出来，当时是每周工作六天，周六还有时间修复周五版本的问题，有时候会发一个紧急更新，确保小米公司的同事们都能用上放心的系统。

2010 年8 月16 日MIUI 正式发布之后，我们还是坚持每周五发布，其实没有什么特别的，因为之前已经是那样了，每周五发布和OTA，就是MIUI的基因。做到每周五按时发布，其实并没有那么轻松，不同的时期，不同的团队规模，都有不同的挑战。随着MIUI 的团队规模越来越大，累积的功能越来越多，用户预期也越来越高，开发流程也跟着变化。从开始到现在，MIUI 大概分为狂野、克制、精准三个阶段。新功能都在体验版开发，开发版的用户其实不会受到影响，因为这些用户喜欢体验完整的新功能，而不是带有问题的新功能。

体验版上的新功能完整之后，提交到开发版分支上，发出去之后对于开发版用户来说还是新鲜的。现在MIUI团队有300 人，同时开发的项目有十几个，每周达到完整的新功能很多，这样可以保证开发版的新功能足够吸引人。

其实开发版还有内测，每周二下午开始，每天都有一个版本，提供给那些稍微喜欢折腾的开发版用户，同时也能帮我们发现问题。写了这么多，不知道有没有回答问题，总结起来就是MIUI 一直都在尝试更好的开发模式，同时满足不同用户的需求，又能够保证作为基因的每周发布。很多事情做不到，不是没有能力去做，而是没有想到可以那么做。

当然，如果能够有MIUI 这么一个既有创新能力、又有执行能力，既充满激情、又能够克制的团队的话，做到每周发布就太容易了。——摘自小米工程师孙鹏在知乎问题“为什么小米的系统MIUI 能够一周更新一次，而魅族的固件更新周期很长？”的回答，http://zhi.hu/UJgUMIUI 的开发风格是不是有点美剧的风格？定期更新、绝不延误、尽量满足用户。

但和美剧一样，这样的节目编排计划就对编剧能力要求特别高，不仅需要保证内容合理而超出观众预期，还要定时定点输出对应的内容。正如孙鹏所言，MIUI在尝试更好的开发模式，同时满足不同用户的需求，又能够保证作为基因的每周发布。因为MIUI 的这种开发风格和及时响应用户的习惯，MIUI 在近两年来获得了许多用户，让人称赞。

**控制迭代节奏，把控用户期望不仅仅是产品经理所需要关注的事情，还是整个项目都需要关注的问题。毕竟，真正做到定期更新版本、交付符合用户预期的功能和产品，非常考验整个团队的凝聚力和战斗力。**

来源： <<http://djt.qq.com/article/view/1296>>

**通过人物角色了解用户期望**  
  
　　无独有偶，不仅仅是《生活大爆炸》中人物角色个性鲜明，宁财神的《龙门镖局》中的人物亦有类似的特征。

* 陆三金：有点面但是爱说道理。
* 盛秋月：冲动喜欢，武力解决一切的女性。
* 白敬祺：爱面子的刚性男。
* 吕青橙：武功好但是比较单纯的女性。
* 温良恭：有很多感情故事的男性。
* 邱缨络：爱财花痴的女医生。
* 蔡八斗：想赚钱的男厨子。

　　大部分时候剧情的推动往往是依据每个人的个性安排矛盾和冲突的。比如，一旦出了事情，蔡八斗永远是第一个说“出事了！”的人；温良恭则永远与前女友纠结不清。这些都是剧本的创作手法，但我认为，也是互联网产品非常值得学习的地方。  
  
　　在日常互联网产品的使用过程中，产品经理与用户之间的距离不能说很近，虽然两者依靠产品进行连接，但大部分时候用户不会表达许多想法。因为对于产品经理来说，抱怨功能缺失的用户A 和吐槽界面设计的用户B 可能都是模模糊糊的两个人，似乎属于用户的范畴，但实际上两个人可能在现实生活之中非常不一致，情境也各有各的特点，但这些细节都是产品经理很难去获得的信息。  
  
　　因此，在一开始产品经理通过用户调研方法建立了核心人物角色之后，可以按照电视剧的模式描述每个人物的特点，然后根据特点进行头脑风暴，讨论这几个核心用户在使用产品的过程中会遇到哪些问题——甚至是会吐槽什么类型的话。我时常看用户反馈，会发现一些很有意思的句型。

* 无理由抱怨型：垃圾，真是垃圾！！
* 有理由的反馈和建议：如果有个xx功能就好了。
* 恨铁不成钢：唉，上次反馈的那个问题，这个版本还是会出现……
* 语言侮辱和人身攻击：傻X的xxx和xxx公司！

　　在我刚入职做产品经理的时候，会和用户在QQ 群里讨论产品问题，甚至有些热心的用户时常会加好友和我聊产品问题。我发现用户与用户之间的确有着不同的背景和期望，有许多类型的用户。在观察了各大安卓市场、电商网站的用户反馈评论之后，更坚定了这一观点。  
  
　　创立核心人物角色的方法在前面已经有所阐述，相信各位产品经理都希望去试一试这个方法。不过，我建议产品经理一定要多和用户进行沟通。和用户沟通不一定必须是关于产品体验的，可以适当发散一些，聊一聊生活等问题，可以通过这些信息更全面地了解用户，让整个人物角色更加饱满。正如谢尔顿和莱纳德都是技术宅，但后者更加了解日常社交的方法；邱缨络和蔡八斗都爱钱，但一个想要通过傍大款获得，一个希望自力更生艰苦奋斗获得。  
  
　　把控用户体验是每个产品经理都头疼的问题，但这也是锻炼产品经理能力的阶段。尤其是刚入门的产品经理在今后遇到的项目都有可能会出现各种用户体验问题。别担心，方法可以慢慢摸索，关键是始终记得那句话：以用户为中心。这句话也是产品经理在项目前期和探索需求阶段需要特别关注的，别为了做功能而做功能。  
  
**目标至上——细节和ROI 的故事**  
  
　　敏捷迭代和人月神话  
  
　　说完了用户体验，让我们回到项目的真实情况。等产品经理写完文档，以为一切都正常进行的时候，发现自己突然陷入一种忙乱的状态——相关的人一波又一波地过来找你，各种措手不及的事情让你原本心情很好的早晨变成了一团糟的开端。别太担心这种情况，每个项目启动后都很容易遇到这样的情况。这不仅仅是含混性的问题，更因为大部分时候项目规划都是在理想情况下进行的，而现实是这些理想情况都有可能背道而驰，出现一些意外。  
  
　　我有时候会和其他产品经理讨论这种现象：究竟是什么原因使得每个项目在过程中会出现这样的情形？我的一个朋友说，首先你的预期就不应该是整个项目都按照理想状态进行，这显然是不现实的。对于一个项目来说，没开始做之前永远不知道会遇到什么问题。所以在项目进行中，应该带着问题不断前进，推动问题解决和项目进度，这才是产品经理需要关注的事情。另一个产品经理则提到，很多时候产品经理无法控制所有的细节，而这些细节会随着项目而产生。  
  
　　那么，如何将这样的意外控制在有限的范围呢？互联网公司现在流行讲敏捷迭代，实际上有多少公司真正做到了敏捷？假设百米跑是项目经理常说的敏捷迭代，那么跨栏跑则是真实状态下的项目：既要敏捷，又要解决难题。这才是大部分项目的真相，也是产品经理不得不面对的真相。  
  
　　如果把写作本书看成是一个个人项目，我深刻地理解了敏捷迭代是多么重要。因为还需要进行日常的工作，所以我每天都是利用下班之后的时间进行写作，尤其是周末时间。按照理想状态，我应该每天写2000 字左右。但事实上，我大部分的写作时间都来自周末。不如来看看我的“项目计划”：

* 工作日每天输出2000字，每个月争取输出40000字以上，周末用来修改文字和查找案例。
* 每个月输出一章内容让朋友提一提意见。看起来似乎挺可行的，每天2000 字对于我来说不算太难——毕竟我日常也有写博客的习惯；文字质量我对自己有信心，因为平时写的文章都有一些业内的朋友关注，他们觉得挺不错的。那么，我就把这种理想状态的计划看成是我后续写作的原则。但事实呢？
* 工作日基本每天的输出都达不到2000字，甚至大部分时候都没有输出；而周末因为在星巴克待两个下午，可以勉强完成10000字；几乎没有修改时间。
* 每个月输出的章节基本都欠缺一些关键内容，时常会出现“待补充”的字样，不忍心让朋友们阅读。这就是写作项目中的大风险——我很有可能无法保证在约定时间内写出对应的字数和有观点有价值的内容。按照计划，我现在应该每天轻轻松松地坐在电脑前，敲出2000 字，然后起身投入到生活中。但自从写作之后，我发现自己大量的时间花费在了写作之外：
* 加班。最直接的因素，常常会影响到下班后的状态。
* 社交活动。人天生就是社交动物。
* 呆坐在电脑前。我时常打开文本编辑器，然后一脸茫然地看着发光的显示屏，从亮到暗到关闭，然后一敲回车键，屏幕从关闭到亮。我在电脑前想：为什么缪斯女神总喜欢在我坐下的时候就离我而去呢？
* 排解没有灵感的痛苦，进行“劳逸结合”的活动。大部分呆坐着的时间非常痛苦。因此，我常常选择起来运动一下，或者是看看微博、刷刷豆瓣……然后就陷入了现代人常有的拖延症状态，将时间花费在刷各种资讯上。
* 日常的运动健身。为了保持高强度的大脑活动，我从写作开始就保持长跑，希望向村上春树的《当我跑步时，我谈些什么》一书致敬，并希望像他一样在跑步中感悟到一些思路，通过释放压力更好地获取灵感。还有一些日常的零零碎碎的事情，于是我的写作计划基本破产。按照阿瑟•布洛赫在《墨菲定律》一书中的阐述，我显然被墨菲定律缠身：事情如果有变坏的可能，不管这种可能性有多小，它总会发生（Anything that can go wrong will gowrong）。于是在一个月之后，我推倒了这个计划，重新写了一份我能接受的计划：
* 周末写作10000字，工作日写1000字。完成一章之后再进行编辑工作。
* 保证内容都可以连贯阅读，然后就给朋友们看最新章节。

　　或许有人会说这是朝三暮四的故事——不，我想称之为项目管理。我曾经戏谑地和我的朋友说，我这个才叫敏捷开发：先做再调整计划，小步快跑，随时根据自己的现状进行调整，但不影响整个项目周期，甚至还能取得更好的效果。  
  
　　事实上，敏捷迭代并非只是简单的一套方法论，更像是一种理念，贯穿整个项目的始终。  
  
　　Scrum 是一种迭代式增量软件开发过程，通常用于敏捷软件开发。腾讯目前在使用的项目迭代应该就是基于这种迭代方法的。最具标志性的产品就是内部使用的TAPD 系统。通过这个系统，项目相关人员可以很方便地看到各自想要的信息：需求单、开发进度、bug 单等。对于项目经理来说，通过TAPD 可以有效地统计整个项目迭代周期中出现的情况，并及时发现问题。  
  
　　除此之外，腾讯内部流行在开发中后期举行定期的例会，比如每天上班之后的晨会，会上需要每个人阐述三个问题：

* 昨天做了什么？
* 今天打算做什么？
* 完成昨天的任务时遇到了什么样的困难，需要什么样的支持？

　　通过简单的信息同步之后，负责人就可以尽快联系对应的人员，沟通对应的难题。一般来说，这些问题大部分都是信息不同步造成的，产品经理通过参加这样的简短会议，可以快速找到问题要害，及时召集对应的人员给予支持。敏捷还体现在整个项目迭代中，会有许多参与角色，比如产品经理、开发工程师、测试工程师等。每个人有每个人的迭代阶段，比如：

* 需求收集阶段。
* 需求设计阶段/需求输出阶段。
* 开发技术预研及框架构建。
* 功能开发阶段。
* 功能测试阶段。
* 系统测试阶段。

　　上述粗略的阶段划分主要是想说明，每个阶段其实都不是堆栈式的，而是当有了部分输出之后，下一个阶段就会马上同步进行，通过多次迭代实现整个版本需求。  
  
　　通过持续集成等环境，版本质量在不断提高。与此同时，测试人员在进行测试，开发人员在进行开发，产品经理则可以通过持续集成的版本，进行体验优化。当整个项目接近尾声时，则可以通过灰度测试的方式论证用户对于新产品功能的认可度。  
  
　　这就是腾讯比较有特色的敏捷迭代方式，对于一些小团队也非常受用。在我写作的过程中，虽然只有我一个角色，但依然需要给自己不断规划迭代周期，并按照任务数进行排期实现。虽然说敏捷迭代相比传统的瀑布式开发会更加没有节奏感，但实际上，敏捷也需要良好的规划时间和任务，不然一环连一环可能引起更多的问题。  
  
　　对于产品经理来说，了解基本的敏捷迭代方法非常有必要。大部分时候，产品经理的执行力其实就在于如何规划功能并引导相关成员把功能实现。敏捷迭代方法是一种成本低、速度快的方式，也是提高执行力的方法之一。通过细分任务，每天定时汇报，产品经理及时体验，可以在短时间内把功能做得比较完善。  
  
　　在敏捷方法中，对于产品经理最关键的启示就是：目标导向。在日常需求讨论的过程中，时常会遇到这样一种情况：  
  
　　这个功能符合用户场景，而且做得很炫。  
  
　　产品经理时常会遇上这样一种情况：用户提出了一个反馈，产品经理需要考虑如何去满足；或者是有一个功能可以做得很炫，但对用户不一定有太大的价值。  
  
　　按照常规，这些需求理所应当地被剔除，或者是延缓实现。但有一种情况就很尴尬：有价值但成本较高的功能。在《用户故事与敏捷方法》一书中展示了一个排列优先级的方法——莫斯科（MoSCoW）规则：

* 必须有（Must have）
* 应该有（Should have）
* 可以有（Could have）
* 这次不会有（Won’t have this time）

　　大部分尴尬的情况都源自于“可以有”的需求。这种需求优先级不高也不低，对于用户来说，是具有使用价值的，但在项目中并非是一个最紧急的需求，时常会被前两个需求挤掉。毕竟项目的周期和成本是关键因素，任何时候资源都需要用在“刀刃上”。因此会出现这样一种情况：每一个项目迭代周期都会有许多的高优先级需求（Must have 和Should have），于是每一期的Feature List 上都会有一些“可以有”的需求排不上队，一期又一期都没有进入迭代计划。  
  
　　而随着项目迭代几次之后，越来越多的新功能转移了用户的注意，之前提到的“可以有”的功能只剩下零星的几个用户在反馈了。项目依然在前进，而新功能依然层出不穷，需求变更依然不停地被加塞到项目中。慢慢地，那些“可以有”的功能被边缘化了。直到有一天，产品经理发现用户重新关注起了这个功能。  
  
　　这个时候应该怎么办？做还是不做，这不是一个简简单单的需求问题，而是涉及整个项目的问题。我第一次遇到这种情况时，急急忙忙地找交互人员讨论交互方案，找开发人员确认开发难度及工作量，找测试人员沟通测试工作量，然后还需要和项目经理沟通协调。当得知工作量比较大的时候，这个需求似乎又需要延期了。这类“搭车”需求并非是偶然出现，大部分时候是项目的常客。但作为产品经理，是否应该重新思考一下：这个需求之所以没有实现，是真的因为成本，还是因为其他原因？产品的目标是什么，我们为什么要做这些功能？无独有偶，在每个产品团队内部可能都会有这样的讨论：xxx 功能是否可以加上？往往最后大家的争论都变成了这样的对话：

* 这对用户来说是有场景有需求需要被满足的！
* 但是这对产品来说价值有那么大吗？
* 是否是所有的需求都需要被满足，还是产品经理应该只关注有价值的需求？有的时候，产品经理时常忘记两件事情：
* 用户的需求是源源不断的，而且有优先级。
* 每个版本、每个功能都有对应的目标。

　　在项目进展过程中，时常出现产品经理在讨论细节问题的情况，但很多细节的讨论纯属没有目的的扯皮——例如按钮放左边还是放右边比较符合规范和用户习惯。这个事情可以直接做一个内部调研或者运用灰度测试的机会测试用户的选择，而没有必要争论各自的观点，变成了两个人为了维护自己的立场进行的争辩——况且他们拥有的是同一个目标。目标很重要。  
  
　　除了日常时常出现的争论，项目时常出现“可以有”的需求往往还有另一个原因：优先级问题。优先级意味着有一个目标，而所有的需求都应该向这个目标看齐，然后一个一个进行排队。优先级是如何安排的呢？按照A～Z 的顺序排列吗？或者是根据产品经理喜好进行排列吗？这个问题并没有准确的答案，如果需要有一个答案，那就是：权衡用户的需求和产品的价值。  
  
　　事实上我们会发现，用户每天都大声嚷嚷的需求在经历了一个迭代之后，用户并没有欣喜：提供这个“缺失”的功能是应该的。如果这个功能还有bug，那么第二天迎接你的又将是一大波反馈。反而，产品推出了一个用户未曾想到的功能，即使问题多多，用户依然会喜欢并忍受一些缺陷。这真的是一个很有趣的现象。  
  
　　既然如此，产品经理应该如何抉择呢？把主要精力投入到新功能开发，还是修复bug？相信每个人都有自己的答案，但我的答案是选择做新功能。产品需要前进，需要有前进的动力。  
  
　　正是因为用户预期与产品功能有着微妙的关系，所以“可以有”的需求大部分时候都隐隐于市，被新功能遮盖了。项目的目标一开始就确立，主打新功能的开发，减少成本在修修补补的工作上，除非是特别紧急的工作。在上一章曾谈到把控用户期望，这也是一种方式，即通过新功能的方式推动产品前进。让用户感到“意外之喜”。不过往往这样进行得越久，用户的期望会越高，而老版本的问题依然存在，最后还是会爆发。所以这并非是一个可行之策。  
  
　　正是考虑到种种现状，我们才说产品的目标尤为重要——我们有了产品原则，有了优先级，就应该朝着目标前进。用户的需求不外乎三种：

* 现在就应该实现。
* 可以拖延实现。
* 绝不可能做的。

　　正如上文所说，用户的需求是源源不断的，产品经理必须有自己的目标，在用户体验和产品价值之间找到一个平衡点，像个专业的钢丝杂技演员，徒手踩稳钢丝线而不是摔下来。

来源： <<http://www.gameres.com/307998.html>>

**项目中的墨菲定律——如何防止事情变糟**  
  
　　永远不要相信计划会一帆风顺地进行至结束——要么风险已经来到，要么正在孕育更大的风险。  
  
　　曾看到一句话说：企业家常常看到机会，而投资家则往往关注风险。我认为，产品经理不仅仅需要看到机会，更要看到风险。在项目启动之后，风险往往是导致项目走向失败的不稳定因素。  
  
　　在吕克•贝松的电影《Léon》（国内翻译作《这个杀手不太冷》）中，马蒂尔达（纳塔丽•波特曼饰演）问了这样一个问题：  
  
　　人生原本就是如此艰辛，还是只有童年如此？始终如此。  
  
　　产品经理或许都想问一个类似的问题：所有项目都充满着各种风险，还是仅仅在刚入门时如此？我想借用Léon 的回答：始终如此。风险不可避免，从统计学角度看，项目出现问题的几率不会为零。  
  
　　这句话不是哪位名人说的，是我个人的感触。经历了如此多的项目和版本迭代之后，我重新认识到了项目风险这个变量是多么麻烦。就拿我写作本书来看，把整本书看成是一个产品项目，已经经历了许多风险：

* 写作难产。
* 案例难找。
* 其他事情的冲击。

　　而在写书计划开始之前，我对于写作的评估还是预留了一个月的缓冲期，得出结论需要半年时间。但显然写作这件事情并非是一种累加工作量和时间就可以保证质量产出的项目，未知的因素太多：时间不够、精力不足、生病、没有灵感、缺乏有力的证据和案例、把时间耗费在查找资料上……而我个人作为项目的工程师，同时也是项目经理，可以随时调整自己的时间计划，尽可能地赶上截稿日，但如果是多人参与的项目呢？或者是真正的产品项目呢？  
  
　　在经历了如此多的项目之后，我得出了本节开头的那句话，项目出现问题的几率不会为零，即没有不存在风险的项目。  
  
　　曾经跟进过一个很简单的需求，按照之前的工作量评估只需要3 天就能完成，但由于涉及前后台联调，期间出现了许多意外。开发工程师请假、后台测试环境搭建有问题、联调的时候又出现意外、因为节假日造成拖延，总之这个需求前前后后一共做了10 天。这仅仅是一个小需求，而整个项目肯定不只10 个这样的需求。对于项目来说，这种隐性的未知风险越多，延期的情况会越严重。项目延期可不是一件好事情，不仅有可能错过最佳的发布时机，还有可能引起团队内部的不稳定——毕竟时间越拖越久，每个人都容易产生不满情绪。  
  
　　出现了问题就需要去解决问题，团队内部存在这样一种角色：项目经理。项目经理的职责很简单：团队和项目的所有事情都和他有关。有的团队可能是由产品经理去兼职项目经理的职务，但对于一些大团队来说，项目经理的出现大大减轻了产品经理的工作。比如，产品经理只需要关注产品并保证质量和用户体验即可，不需要考虑太多排期和资源的问题。但如果需要兼职做项目管理的工作，产品经理就无法很用心地把时间花在产品上，而需要不断开会沟通，不断协调资源。  
  
　　产品经理同时兼任项目经理的最不好的情况是，为了避免延期而做出砍功能的决策，但往往砍功能又会陷入另一种困境。为了打造最佳的产品体验，每一个功能都有对应的作用，如何决策又成为了一个难题。  
  
　　那么如何解决问题呢？项目经理常用的手段是建立流程——通过流程来把控项目的进度，这又是一把达摩克利斯之剑。良好的流程的确可以让每个决策、每个风险和难题都拥有处理依据，可以快速定位问题并解决问题。但流程如果不起作用，则是团队中所有人的灾难，产品经理会抱怨流程复杂影响产品迭代；开发人员则抱怨时间太紧张，不允许给出一些缓冲时间；测试人员则会认为整个流程化的结果虽然好，但万一开发阶段就出现了问题，会让测试工作更加难以展开。  
  
　　项目经理可能会据理力争，你们没有完全按照流程来进行——但流程其实也有缺陷，比如流程化影响了快速迭代的方式。产品经理发现了问题应该直接找开发工程师修改，然后由测试工程师尽快验证并得出结果，但流程意味着这些都需要一步一步通过审批的方式来处理——这也是一种风险。  
  
　　所以风险无处不在，我们所能做的并非是拒绝风险，而是迎接风险并快速响应。正如下雨天，我们即便没有撑伞，也不应该继续在街头慢悠悠地走着，而是快速跑到避雨处或者寻求雨伞等可以解决问题的方案。  
  
　　项目中的风险管理提到风险，不得不提到一本书《黑天鹅》。作者塔勒布在书的开头这样描述：在发现澳大利亚黑天鹅之前，所有欧洲人都确信天鹅全部是白色的。这是一个牢不可破的信念，因为它似乎在人们的经验中得到了完全的证实。对一些鸟类学家（以及非常关心鸟类颜色的其他人）来说，看见第一只黑天鹅大概是一种有趣的惊奇体验，但这还不是澳大利亚发现黑天鹅的重要性所在。它说明我们通过观察和经验获得的知识具有严重的局限性和脆弱性。仅仅是一次观察就可以颠覆上千年来对白天鹅的数百万次确定性观察所得出的结论。你所需要的只是看见一次黑天鹅就够了。  
  
**作者得出“黑天鹅”事件的三个特点：**

* 稀有性。
* 巨大的冲击力。
* 事后可预测性。

　　但我从中受益最多的是这样一种视角：经验不一定是可靠的。正如我周六下午一如既往地带着猫咪去宠物店洗澡，但宠物店老板告诉我今天这里都是狗，不能给猫洗澡。按照以往6 次经验，周六不会出现如此紧张忙乱的情况，而且狗的数量也大大超过了从前。这不是关键，关键是我今天有一个重要紧急的事情需要处理，顺便出门把猫带去洗澡——这一意外直接干扰了我的计划，导致我不得不花费更多的时间。原本我可以出门一次就完成两件事情，但因为意外，我不得不在路程上多花费一倍的时间。  
  
　　产品进入项目周期的时候同样会出现这样的问题。有一次开发工程师评估了两个需求，觉得可以一起做，于是就把工作时间缩减了一下。结果在开发过程中发现两个需求的后台机制是不一样的，严重影响开发时间。这就类似“猫咪洗澡”事件。最开始大家都很开心可以在5 个工作日完成两个需求，结果发现这两个需求要花费10 个工作日，这种风险对于项目来说是致命的。  
  
　　著名的项目管理书《人月神话》提出了一个有意思的问题：在项目中增加更多的人手能不能提高效率，从而在更短的时间内完成项目？100 等于25 乘以4，也等于50 乘以2，这是一种线性计算。但在项目中并非如此。100 人•日的工作量并不会因为开发人数增加一倍而变成50 人•日，很有可能还是100 人•日，甚至是超过100 人•日。  
  
　　人月神话（The mythical man-month）：讲述了人力（man）和时间（month）并不呈现线性关系。指出投入大量人员并不能缩短软件的开发进度。一窝蜂式的作业方式无助于软件生产，而且会制造麻烦，产生出更差的软件。向进度落后的项目追加人力，只会使进度更加落后。因为新进的人员需要时间了解整个项目，而增加额外的沟通消耗。若有N 个人必须在这群人之中进行沟通（无阶级关系），当N 增加，其沟通所消耗的M 将抵消其产生的效益，甚至出现负增长（最后几天所完成的进度，远不如刚开始几天所完成的进度，像是发现了许多错误）。  
  
　　不过这种计算方式是在一个宏观层面上去看待项目，落实到实际操作中时，还会源源不断地出现各种问题。举例来说，有一次，产品有一个业务需求是由兄弟部门支持的，之前都定了合理的时间，并保证了质量。最后出现了大家都没想到的情况——这项业务涉及的一些功能无法通过App Store 的审核，不得不打回，重新修改然后上架。对于本团队需求来说，及时调整和修改还算比较敏捷，但对于外部业务来说，沟通的成本和修改的成本就大大提高了——业务需要考虑修改了之后是否会对产品产生影响，功能是否会因此减少，体验是否会有明显的降低。  
  
　　对于测试人员来说，每一次的发布都是工作量最饱和的时候，因为大量的修改都需要验证通过之后才可以发布。何况，这仅仅是一次可以规避的意外，还有许多未知的事情在寻找机会发生——这就是风险所在。  
  
　　对于新人来说，风险更容易出现。我并不否认经验对于工作和降低风险的重要作用，但往往有一些事情无法被预知。恰如混沌理论中的“蝴蝶效应”，牵一发而动全身。初为产品经理时，我曾遭遇过两个与此相关的事件。  
  
**事件一**：风险的出现起源于一份没有说清楚的需求文档。当时我交付了对应的需求文档，但由于缺乏经验，对于细节的把握少有提及，往往是把交互操作和页面跳转的逻辑重述了一遍，忽略了对于字段长度的限制、意外情况展示等方面的描述，导致开发人员对此理解有偏差。但当时我和开发人员并没有建立很紧密的关系，双方都按照各自的理解进行工作，没有及时沟通，甚至在需求评审的过程中也没有对细致的结果进行充分沟通。随着开发时间的逼近，我开始体验功能时才发现了许多小问题没有解决。本以为这些小问题可以很容易地处理，结果被告知整个实现方式并不能修改为我想要的效果，不得不推翻方案重新开发。  
  
　　在这个事件中，沟通是最大的问题。但如果产品经理有经验，一开始就确认一些细节问题，并及时跟进体验，多与开发人员进行交流，可以避免很多问题。  
  
　　**事件二**：无法被预知的风险才是最可怕的事情。当接近开发尾声的时候，发现老板要把方案进行调整。此时就会遇到一个棘手的问题：如果要按照方案调整，就不得不推倒重来；但重来的话，会影响到整个项目进度。于是不得不找到开发人员的负责人和项目经理重新商议此事。这种临门一脚的调整往往有可能带来许多烦恼。  
  
　　所以，我们不得不把风险永远地列入产品经理和项目经理的关键词列表。关于风险的概念不需要我再多做解释，唯一可以承认的事实就是：无法预测。  
  
　　正如时间的流逝，风险的存在并不是一个可控的问题，也不是一个管理问题，而是如何应对的问题。我们无法完全地杜绝风险，而应该迎接风险。正如LinkedIn的创始人霍夫曼在《至关重要的关系》一书中所提到的：风险常在，因此我们所有人都得去冒险。但是，在面对风险的时候，并不是所有人都能聪明地应对。许多人认为，最小化风险就能获得稳定的事业。但是，相当讽刺的是，在一个千变万化的世界里，这会是你做的最具风险的事情之一……  
  
　　风险因人而定，而且是动态变化的。  
  
　　所以，让我们迎接风险，学会与风险并存，及时发现风险，管理风险，这是应对不断变化的现实的最佳策略：趁势而为。  
  
　　不过，就算我们承认需要主动地与风险打交道，还是面临这样的问题：

* 如何有效地规避一些风险？
* 如何制定有效的机制及时发现风险，而不是等到后期才发现？
* 如何管理风险？

来源： <<http://www.gameres.com/309852.html>>