**软件需求管理总结**

　　**什么是需求**

　　大家都在讲需求分析，但是什么是需求，软件工程中提供了一系列复杂的解释。我所理解的需求就是，用户用的不爽、不舒服、不合适的，我们就要去解决这样的问题。不管在产品的任何阶段，把用户放在首位，是否满足了用户需求，要么解决了痛点，要么带来快感，这才是必要的需求。

　　**需求获取**

　　需求获取有多种途径：用户访谈、调研问卷、其他渠道的反馈

　　用户访谈：通过6-8个（针对一个或者一段时间的产品迭代）用户访谈，定性了解用户的使用情况，最好的访谈形式，是在用户所熟悉的场景中，还原使用产品的整个过程；不过这样的成本确实蛮高，邀请一个用户需要耗费大量的时间和金钱；所以一般都是邀请用户到公司参加**测试**、或者直接电话访谈的形式。

　　O2O的用户除了日常使用者之外，背后还有忽视掉的商户，而商户端的用户情况非常复杂，包含：服务员、前台、迎宾、收银、市场经理、店长、老板、连锁店老板、甚至公关营销和法务，所以针对商户端的用户访谈，电话是不合理的，除了跑到店里体验服务外，就是跟以上角色一对一聊，才会获取到一线的需求，以及产品的评价（这一系列**文章**里面，考虑到专门有一篇，就是写用户访谈，一半重点会放在商户侧）。

　　调研问卷：访谈是定性的了解，那调研问卷就是定量的研究，针对访谈中出现的问题，通过调研问卷的方式，从更大规模的用户来论证，并根据调研问卷结果，将需求做优先级处理。有时候调研问卷不只是为了论证需求的存在，同样可以论证需求是否合理。

　　但是你如果直接问用户，如果我增加一个功能，你需要么？大部分用户回答都是需要（不管是访谈还是问卷），所以需要通过引导式的内容，或者问卷中，开放式的问题，来收集并论证新功能的必要性。如果真的是用户急需，肯定会有用户强烈提出（这部分天使用户需要好好珍惜）。

　　其他渠道的反馈：很多渠道的反馈，吐槽也好，表扬也罢，都说明天使用户对你产品的关心；除了产品自身的反馈渠道之外，还有论坛、围脖、知乎等等第三方网站；另外媒体渠道，现在的36kr等、可以看到竞品的报道，可以尝试分析下报道背后的原因，以及对自己报道后，各界对自己产品的反馈。现在不只是PM会关注围脖神马的，连各个公司大佬们都会积极去关注。

　　**需求分析**

　　通过种种方式，需求收集回来了，整理整理，to do list里面少则十几条，多则上百条；当然了调研问卷会帮你筛选出一批，可是剩下的部分，怎么去论证需求的合理性、必要性、甚至是否为无理需求呢，从实践以及跟前辈交流中，总结了以下三种方式：

　　用户怎么说：用户永远是对的；这句话对，也不对。用户会告诉你想要什么，但是用户不会告诉你他的需求是什么；所以需要从用户那挖掘需求。

　　馒头和海底捞的例子，小明说晚上我们去吃海底捞吧，为什么呢？因为他饿了，其实他的本质需求是饿了，给他两馒头，可能会不爽，但是绝对解决了小明的需求；同样的，老板说我请大家吃饭吧，让你安排，你安排馒头，会被所有人K死吧，相反，这时候海底捞就是个不错的选择

　　所有用户怎么说，只是描述用户的行为，使用习惯，以及所要的预期结果；真正如何去满足用户的这个预期结果，就是PM需要从用户深挖出需求，然后通过产品方式去解决。

　　数据：一切以数据说话，这是产品的准则，虽然数据有时候会骗人，也有很多为了数据好看故意掩盖的行为。但是真实可靠的数据，确实是能为产品增色，给用户带来方便。

　　举个例子：在做App的时候，第一版本是拍脑袋，根据竞品分析和自己判断，显示神马内容；上线一个月之后再进行优化，我就选取了大部分用户使用的几个场景和动作，分别做了匹配；交上去被骂了一顿（略夸张，不过自信心还是被打击到了），肯定有更好的方式来实现。我就静心仔细思考，用户在使用时，每天在不同时间段，他的动作和目标是不一样的，所以我将每天以小时分段，看每个时间段，用户都进行神马操作，发现在每个时间段中，60%甚至更高的用户都是一样的目标和操作；

　　所以就简单了，将24个小时前后有类似操作的时间段做个区分，用户在这个时间段打开App看到的内容，就是大部分人所操作的，可能影响小部分人，但是方便了更多的用户，毕竟产品永远是为大部分人去准备的。

　　竞争对手怎么做：有一句话说得好，我们不需要重复造轮子；圆是上天赐予我们的礼物，前辈们的产品和设计，也是他们赐予我们后辈的财富。在竞品分析的时候，就可以留意别人好的设计；可能有人会觉得不耻，不就是抄袭么？对的，世间那么多产品，对个功能的设计，你能抄袭或者借鉴到，至少说明一你有用心留意并观察别人的产品；二别人的产品至少被用户接受了；三他已经慢慢帮你培养了用户的使用习惯；四直接证明此需求存在。模仿是一种很稳妥的方式，毕竟世间就只有一个乔布斯；此外在模仿中升华，做出更完美的产品，岂不是更好。

　　当然了，不能盲目的去跟风，别人有，我也要有的思路是不对的。这里所讲的，是你通过别人的产品去论证需求，别人成功的产品，去实现你的需求。产品做加法一点都不难，难的是做减法，如果能在别人成功的产品上做减法，那还是需要严密的论证以及大胆的尝试。

**对需求做决策**

　　论证了需求的存在，以及必要性，下面就是对需求的优先级交付开发，有些需求，因为优先级低，可能永远处于被砍掉的部分，或者一直呆在to do list直至天荒地老。

　　**需求优先级：**

　　需求永远是做不完的，而研发资源永远是不够的，怎么办？所以PM需要对所有需求的优先级进行分类，研发按照需求优先级列表，一个个进入开发队列。如何划分优先级：MVP（最小化可用产品），快速迭代，迅速论证需求及产品的合理性。当每个需求出现在列表中时，不停的问，这个需求有必要么？有必要优先级这么高么？不做用户会不会发狂？不做产品是不是能run？不做是否不通过产品线下也有解决方案，成本和线上比怎么样？经过这一系列论证，某些是必须要做，而且立马要做；有些是必须要做，但是并没有那么紧急；有些甚至是必要，但是却不是当前阶段需要的。少即是多，所有功能的累加并不难；难的是只提供用户核心的功能和产品，并让用不离不开他，再在这样的功能上，轻松调整和扩展产品。

　　**那些被砍掉的需求：**

　　从参与工作的第一个月，就整理了一个feature list，都是大家脑暴，或者研究竞品给产品未来做的规划，现在回头来看，里面所描述的功能，绝大部分都没有去做。一方面的原因是产品还很小，没必要大而全；另一方面部分功能，完全拍脑袋决定，根本没有必要在产品中增加。

　　feature list是产品规划方面的需求，具体执行层面，每次需求评审，会故意放进很多需求；老板可以砍，技术可以砍，QA可以砍；需求多研发肯定会叫，象征性地砍掉不需要的需求，适当的把部分需求延期，只要保证你所要的核心需求，在这次迭代完成就好了，毕竟已经砍了一部分需求，不好意思一直砍。具体怎么交付技术需求，跟技术沟通会专门再写一篇。

　　**交互视觉和重构**

　　让专业的人做专业的事情，虽说PM应该是个70%的交互设计师，但是公司既然有了交互设计师，那交互的工作就十分信任地让他们去做；PM做的只是跟交互设计师描述清楚用户使用场景。当然作为新人，我经常犯的错误就是，我这里需要加XX功能，而不是我要解决XX问题。视觉重构同理，PM就是要利用好这些资源，并充分地信任他们。

　　关注用户体验：产品要么给用户带来利益，要么方便用户使用；脱离了这两点的产品都是耍流氓。若一款产品既给用户带来利益又有非凡的体验，才是最成功的。用户体验为啥重要，因为体验会影响用户口碑，口碑影响产品成败，产品成败决定一切。用户体验包含用户所看到的一切元素，以及交互过程，除了显性的特性外，体验上隐性传递给用户的信息，会给造成暗示，如某处金额现实为负时，传递出的隐性情感肯定是偏向负面的。

　　PM要学会讲故事：这里讲故事的意思，是跟UED的童鞋进行沟通，感性的传达肯定比理性的说教要好。某天交互设计师发了这样一条微博：我总是忽略一件事，PM同学提出的究竟是需求，还是ta出于对需求的认知而拟定的一种解决方案。老大回复的是：往往是后者，junior PM因为junior所以会是后者，senior PM因为senior所以还是后者。

　　作为一个刚刚入门，还在摸索阶段的junior PM，反思下平时的工作，面对所有需求时，第一直觉都是想到，如何去解决这个问题；而不是描述用户的使用场景，在这样情况下用户所表现的焦虑和拙计，并将此问题抛给交互，让他以专业的知识来解决。交互设计师不是单纯的画原型图，他们能赋予产品生命和灵感，让用户体验到极致，所以让他们发挥ownership来解决问题，远比执行要好。

　　**需求文档**

　　刚刚开始实习时（不是在点评），写过半年左右的需求文档，当时因为瀑布模型开发，一期需求写一个月，评审后交付开发；然后二期需求文档同时进入编写。当时情形不做评价，对个人的锻炼就是文档算是入门鸟，正式工作后，文档方面也没有任何专门培训，写过几个之后，老大、技术、QA表示还行，半年时间，项目的大部分需求文档都是我产品，当然需求也是我在跟，少说也有几百页，正当我粘粘自喜的时候，发现......

　　发现啥呢，研发基本不会关注的文档，他们都是按照他们的想法和思路进行开发，只有在进行不下去的时候，才会去关注下细节；或者在出现争议的时候，通过需求文档来check；可能文档唯一的读者只剩下QA了，因为他们要写测试用例；看了我们敬业的QA的case，我回过头看我的需求文档，瞬间汗颜。

　　之前犯的错误是，觉得需求文档一定要按照格式来写，当然了这对新人上手有好处；写多了会发现，其实是没必要的工作。文档只需要清晰传递要完成的功能，以及详细的描述就够了，具体啥形式，是无所谓的。

　　以前犯傻，写过的一篇关于如何写需求文档。

　　**那些年犯过的错误**

　　1、替技术思考问题：因为在学校专业是计算机和软件，所以会不由自主地帮技术思考问题，前两个月在需求评审时，会说这个需求工作量不大；甚至会说这个可以这样实现；这是个不好的习惯，还好慢慢改掉了，主要伤害了他们的ownership。

　　2、忽视用户体验：在App端，擅自做决定，界面看似更简洁，但是实际增加了用户的操作成本，这件事情被狠P了一顿，从此长记性了。

　　3、需求没有详细的论证：拍脑袋想一些需求，或者论证的数据不够全面，只看到表面数据，并没有深挖背后实际需求。

　　4、过分强调文档的作用：刚刚入门的PM，对于所有人都忽视你的PRD，那种惆怅是无法言语的，调整好自己心态就好了，一切为了产品。

　　5、木有传递业务价值：不是所有技术都关注业务，但是在传达需求时，需要讲清楚需求的背景、数据、以及原由；如果不做这些，技术就沦为彻底的码农，接受需求，然后开发出来，具体的价值体现，以及自我满足，需要从产品这里得到。