

我所理解的需求文档

需求文档是产品经理日常工作输出最重要的文档，需求文档是在写什么？应当包含什么内容？ 以下是我理解的需求文档，抛砖引玉，供大家探讨。



需求文档最重要的读者首先是产品经理自己，无论是需求实现前用需求文档讲述需求的实现思路，实现时按需求文档进行设计、研发、测试，还是需求上线后回顾需求文档进行复盘总结，都是依照产品经理写的需求文档。其次的读者才是负责需求实现的设计师、研发人员、测试人员以及其他产品经理。

按我的习惯，需求文档会包含以下五个部分的内容。

一、需求变更日志和版本迭代记录

第一部分是需求变更日志和版本迭代记录。需求从提出到最终上线，中间必然经历一些需求变更或方案调整，因而需要有变更日志来记录这些变更。

需求变更日志并不只是写需求新增或减少了哪些功能，而是更仔细些。

例如：需求中的 A 功能点，原来打算用方案一实现，但考虑到资源、现实场景的限制，改为用方案二实现，这个也要写，虽然从结果上看仍然是实现了需求中的 A 功能点，但从过程上看，方案一到方案二的转变，是产品经理思考的升级，也是对资源限制更深的考虑。

此外还应记录清楚需求是因什么原因变更，变更前后是什么样子，可能会产生哪些影响，以便后面查找细节。

而版本迭代记录，除了包括对需求中功能的版本迭代，还应包括产品经理对这个需求思考路径的迭代。

需求中功能的版本迭代，这个大家比较熟悉，不赘述，主要想说明下产品经理对需求思考路径的迭代是什么？为什么要这么做？

之前提过，产品经理很大的工作是在做决策，因此决策质量很重要，而决策质量需要通过不断地优化决策思考路径来提高，产品经理应该记录自己对需求的思考过程，对过程进行不断总结和优化。

另外，将这些思考的过程展示出来及与其他同事讨论，可以跳出自己固有的思维模式，兼容并包。

所以我一直认为产品经理在处理一个需求时，其思考路径、决策依据应该公开透明，能够让所有参与需求的人都能够可查看、可探讨（来自瑞·达里欧，有兴趣的朋友可以去看看他的《原则》）。

那么，我输出需求变更日志和版本迭代记录的过程是怎样的呢？

按我的习惯，在输出一个需求文档的时候，会先按当下我能考虑的情况先写一个 Beta 版，这时候我将它命名为 V0.7 版本，然后隔一天我再重新思考，看自己昨天写的需求文档，这时能发现很多不足的地方，那就从头开始改一遍，标明需求有哪些变更，这时的版本是 V0.8 版本。

下一步找其他产品经理向他讲述一遍这个需求文档或在组内组织一次需求评审，综合意见，再修改一遍，标明需求有哪些变更，这时的版本是 V0.9 版本，最后再找开发测试设计的同学进行需求评审，从开发测试设计的角度对需求的实现做一遍修改，标明有哪些变更，形成最终的版本 V1.0。

这样下来，一份需求文档能够包含产品经理对一个需求实现方案完整的思考过程，其中不仅有自己思考的升级，还有从研发、测试、设计等各个角度对实现方案的调整补充，是针对这个需求，在当前的资源限制、背景约束下最好的实现方案。

有了需求变更日志和需求版本迭代记录，不仅可以做到需求的实现逻辑实现思路可溯源，完整记录整个需求从被提出到上线多个版本，期间的产品思路、实现逻辑有了哪些变化，产品经理可时常回顾拿来

参考，产品团队可针对大的需求做针对性复盘，也可提供给后来接手工作的产品经理了解需求的完整迭代过程。

二、背景&方案&价值

背景、方案和价值，是需求文档的核心，是任何需求在进入到现在阶段前一定要想清楚、一定要反复探讨的部分。

需求是背景下的需求。这里的背景，需要写明白的内容可包括但不限于当前产品所处行业的现状，产品/功能模块所处的状态、目标，开发团队的资源限制、技术限制等。

最开始做产品经理时，体验其他产品的一些功能，不免吐槽，这里怎能这样做？应该那样做啊，要是我的话一定能比他做得更好，诸如此类。

到后来做产品的经验见长，才明白任何一个需求都受限于当时的背景状况、资源限制，抛开这些背景谈实现都是扯淡，产品经理要做的是在当前背景下，找到最佳的实现方案。因此，梳理需求背景是产品经理对当前资源现状的考量，是实现需求的第一步。

方案是背景下需求的实现方案。既然需求会受到资源现状的限制，那么方案也必然有所不同，甚至可能会有折中妥协，会有不完整的方案。有时需求本身就是试验性质的，是为了快速测试效果，那么在方案上选择一些实现简单、开发难度较小的方案也就不足为奇了。

在写方案时，可以按照「用户-场景-问题-方案」这个框架简要写明实现方案，也就是什么样的用户在什么样的场景下遇到了什么问题

题，提供的解决方案是什么——这里要求方案要经过提炼，能够通过一句话说清楚。

价值是指实现这个需求能够带来什么样的价值，包括用户价值和业务价值，用户价值是指实现这个需求能够给用户带来什么样的价值，例如提升用户的使用体验等；业务价值是指实现这个需求能给产品的业务带来什么样的价值，例如提升用户留存或者提升业务收入等。

需求不一定要同时提供用户价值和业务价值，也不一定两个价值都需要为正（例如带来很大的业务价值而牺牲很小的用户价值也是可以的），具体需要依据产品当前的状态来考虑，但不能带来价值的需求一定是有问题的。

此外，在思考需求能够产生什么价值时，同时要思考的是以什么数据指标来评估这个价值，也就是需求上线后效果的好与坏要有量化的指标。不一定所有的需求都能够找到量化的效果指标，但一定要尽量找到这个指标。只有需求的效果能够被衡量，产品才能够往更优的方向迭代。

三、业务逻辑&流程说明&功能需求详述

第三部分主要是需求实现的部分，我把它划分为业务逻辑、流程说明和功能需求详述。

业务逻辑部分描述的是需求中涉及到的数据流向和用户流向，特别是需求涉及到多个系统时，数据和用户在系统之间如何交互的（这部分的内容偏复杂，后面再单独写下我的理解）。

目前针对业务逻辑部分，我主要的输出是多通道的泳道图来描述系统间的交互。这里应该特别注意的是在说明数据流向时要兼顾考虑正常流程和异常流程，以及在说明用户流向时要考虑清楚需求边界。

此外，需求的复杂程度不同，可能还会包含页面流程图、页面结构图等。

功能需求详述就是常说的原型。

我目前的习惯，在需求文档的早期版本不喜欢输出高保真的原型，而是倾向于用低保真原型加文字描述的方式来说明需求中的功能实现。

功能需求详述以需求中的页面为单位，分为原型图、需求说明和交互说明三个部分。

原型图对需求涉及的每个元素进行标注；需求说明针对原型图中的标注进行文字说明，包括字段逻辑、按钮逻辑、页面逻辑等；交互说明则是针对一些非逻辑的交互进行说明，例如某些字段、需要突出显示，页面变化时需要怎样的特殊效果等等。

四、相关文档的集合

日常工作中，时常出现想要找需求的某个相关文档时，四处搜索，浪费很多时间的情况，为此我形成了一个习惯，就是把需求文档作为一个所有相关文档的集合。如埋点文档、设计稿、接口文档、测试用例文档、开发相关的链接、上线后的数据等，都以链接的形式整理在

需求文档中，这样每次需要找需求的相关文档，都可以从需求文档中快速找到。

五、需求上线后的数据

凡是需求，必然要有验证效果的数据，而从每一个失败与成功的需求中不断总结和反思，是产品经理成长的重要途径。如上文所说，产品经理应该保持开放透明，那么就意味着产品经理对于需求输出的实现方案，最终结果无论是好是坏，都应该将效果数据按实际公开，这既能够促使产品经理自己不断改进产品思路，也能够让参与需求的相关同事了解自己的工作成果，增加他们的参与感与成就感。

以上便是我理解的需求文档应该包含的一些内容，可能过于繁杂，具体还是要根据每个人自己的工作习惯做取舍，仅供参考。