

《不是人人都可以做产品经理》

——什么是高质量的需求文档

1 目录

- 一、 前言... 2
- 二、 什么是需求? ... 2
- 三、 撰写需求文档的目的是什么? ... 3
- 四、 什么是需求文档? ... 3
- 五、 什么是高质量的需求文档(PRD)? ... 4
 - 1. 正确... 4
 - 2. 无歧义(无二义性) 4
 - 3. 完备... 5
 - 4. 一致... 5
 - 5. 具有优先级... 6
 - 6. 可验证... 6

7. 可修改... 6

8. 可追踪... 7

六、 参考文献... 7

一、 前言

本文基于 **fanny** 腾讯入职测试实习第三天的当天感想与个人原有感悟。

目的：更好的理解需求文档的目的，了解高质量的需求文档如何撰写。

本文针对互联网尤其 **web** 领域的产品文档、不针对游戏等领域。有错误 请各位及时指出！谢谢

！

文中有任何错误，或者您有任何疑问和建议，欢迎当面一起讨论。不吝赐教！

为了拉高行业准入门槛，提升行业薪资。故叫《不是人人都可以做产品经理》。。。
--

本人 blog: 产品经理新视线 <http://www.gstarwd.com>

二、 什么是需求？

其实我们没有必要去思考什么叫做需求，因为这个议题太大了，无从谈起。针对互联网行业，我们所说的“需求”特指“产品需求”。

那么什么是产品需求？

站在公司的立场上，需求是为了满足领导分配的 **KPI**，为了达到特定的经济效益，由产品经理制定的一些列的产品描述。

那么互联网产品都满足了哪些需求？

如果按照产品来划分，那么腾讯 **1300** 多产品各自满足了不同的需求。

从现有理论角度，根据马洛斯的需求层次理论：

我觉得互联网产品满足了人类的精神性价值需求。传统行业产品是满足了人类的物质性价值需求。精神需求是否被满足是依据用户预期来界定的。这个是动态平衡的过程，也就是说互联网产品的研发的道路是无止境的！

而互联网中的电子商务却很复杂，从几千年前就存在的传统行业演变成了现在线上环节为主导的行业，电子商务倾向于满足人类物质性的需求，这也是电子商务区别于一般互联网的因素之一。本人更倾向于把电子商务从互联网行业里面剥离出来看。。扯远了。。

从更高的抽象层次来讲：

从人类进化角度，弱弱的-，-，我倾向于把人类的需求单纯的分为：

-生存需求

-繁衍需求

。。。。扯远了 下篇文章再展开吧。。

三、 撰写需求文档的目的是什么?

了解了需求文档的目的才能更好的撰写需求文档，才能有张有弛的去控制需求分析的粒度。

我觉得其目的是:准确的描述需求，使得产品最终形态与预期吻合。

要达到这个目标，需求文档对需求的描述必须具备以下特征

:

1. 正确
2. 无歧义
3. 完备
4. 一致
5. 具有优先级
6. 可验证(易于测试)
7. 可修改
8. 可追踪

这样一看，编写高质量的需求文档是一件很难的事情。别着急，本文下文讲详细对每一个特征进行解释。

四、什么是需求文档？

业内对于需求文档有很多种理解形式：**PRD**、**SRS**。。。一般腾讯所指的需求文档都是 **PRD(product requirements document)**，**MRD** 则更偏市场，**SRS** 则更偏技术实现(含有数据流、时序图等)。

有时候也称需求文档为“需求规格说明书”，其实“需求规格说明书”是传统软件行业里面用的叫法，如果需要写得规范一点，那么就需要按照国家标准 **GB, ISO, IEEE** 的规范来撰写“需求规格说明书”，但是那样的文档是十分重型的。在小步快跑，快速迭代的腾讯模式下，轻量的产品需求文档更加符合腾讯。当然这也不是绝对的。一般大型的项目还是需要有详细，规范的，较重型的文档来支撑的。

需求文档里面主要写什么？

我觉得需求文档里面主要的内容是对需求的描述，通过用例去描述。而在腾讯并不是所有人都知道用例是什么，一般来说都用“功能”、或者“特性”来代替描述用例。传统软件行业过来的人一般都知道吧。。。

就是说腾讯内部对于“用例”，“功能”，“特性”并没有形成一个统一的认知。每个人的认知不统一，这是一个比较大的问题，我觉得必须不断的加强这方面的规范。

我觉得用例其实是 **UML** 建模中用例分析环节中的输出物。**UML** 建模是现代软件产业常用的较敏捷的方法。适合产品生命周期较短的项目来使用。其实腾讯是蛮受用的。至于 **UML** 那一套体系是否适合腾讯，还需要实践去论证。

产品需求文档（PRD）重点描述一个被提议的新产品或者现有产品的改进需求。与 MRD 侧重于从市场需要角度看需求的不同，PRD 侧重于从产品本身角度看待需求。通常在特点和功能需求上更深入细节，并也可能包括屏幕截图和用户界面流程。在那些 MRD 不包括具体需求和用例的机构中，PRD 就包含这些具体内容。PRD 通常是由产品经理撰写的。PRD 通常是一份连续的 10-30 页 Word 文档。

在我看来需求文档就是用例的集合。从表现形式来讲需求文档就是：一个用例列表+各个用例的详细描述。

用例的定义是：在不展现一个系统或子系统内部结构的情况下，对系统或子系统的某个连贯的功能单元的定义和描述[2]。

一般用例主要包含以下条目：用例简述、用例优先级、用例图、执行者、前置条件、后置条件、交互稿及其描述、主流程、分支流(异常流)、业务约束。

五、 什么是高质量的需求文档(PRD)?

总的来讲高质量的需求文档就是能最有效的达到需求文档撰写目的的文档(太虚了。。)。

即-能够有效协助产品干系人(视觉、交互、页面、开发、测试)完成与预期吻合的产品。

上文中提到，需求文档对需求的描述必须具备以下特征：

1. 正确
2. 无歧义

3. 完备
4. 一致
5. 具有优先级
6. 可验证(易于测试)
7. 可修改
8. 可追踪

而产品新人需要着重关注其中的：**1、2、5**

1. 正确

当且仅当 **PRD** 中的每一项需求都是产品应满足的需求，**PRD** 才是正确的。

不存在确保 **PRD** 正确性的工具或规程。宜把 **PRD** 与任何适用的上层规格说明(如，立项文档、项目计划书、**MRD**)、其他项目文件和其他适用的标准进行对比，以确保其相互一致。作为一种选择，产品经理可以确定 **PRD** 是否正确地反映了实际需要。可追踪性使相应的规程更加便利并减少缺陷。

2. 无歧义(无二义性)

我觉得需求描述的这个特性是产品经理新人最需要注意的。

需求通常使用自然语言(如，汉语)来编写。但自然语言固有的不明确性，尤其是汉语这样的每个词有很多意思的语言。使用自然语言编制的 **PRD** 宜由独立的一方进行评审，以识别语言的含糊用法并予以纠正。这样可以减少开发减少疑惑，让测试写的测试用例更加正确。

为了消除歧义，其实可以使用需求文档专用的语言的编写，但是这样成本太大了。

那么什么叫做 无歧义？

当且仅当 PRD 中的每一项需求都只有一种解释，PRD 才是无歧义的。这要求最终产品的每个特征至少使用唯一的术语来描述。

当在特定背景中使用的某个术语存在多种含义时，宜将该术语包含在术语表中，以便更加具体说明其含义。

正如在 GB/T 8566 中描述的那样，SRS 是软件生存周期中需求过程的一个重要部分，并被应用于设计、实现、项目监控、验证和确认，以及培训活动中。对于编制人员和使用人员。

其中的 **SRS** 其实可以换成 **PRD**，软件可以换成产品，即：

PRD 是产品生存周期中需求过程的一个重要部分，并被应用于设计、实现、项目监控、验证和确认，以及培训活动中。对于各种产品相关人员。

PRD 宜是无歧义的。但是，产品干系人通常并不具备相同的背景，因而对产品需求的描述不会倾向相同的形式。尤其现在腾讯校园招聘的产品经理几乎很少有相同专业的。每个人可能有各自的描述风格，这样会降低团队效率。所以产品新人必须对需求描述的二义性有深刻的理解和感知。这也是为什么我觉得：需求描述的这个特性是产品经理新人最需要注意的。

3. 完备

当且仅当 **PRD** 包含以下要素，**PRD** 才是完备的：

- a) 所有重要的需求，不论是否与功能、性能、设计约束、属性或者外部接口有关。尤其是由系统条件所施加的任何外部需求都应当得到确认和处理。
- b) 产品响应的定义，以说明产品对所有可实现的输入数据类型的响应。应当注意，对于有效和无效输入数值两种情况，规定产品响应是重要的。
- c) **PRD** 中所有图标的全部标记和索引，以及所有术语和度量单位的定义。

任何含有“待定”词语的 **PRD** 是不完备的。但是有时使用“待定”是不可避免的，若万一使用“待定”时应做如下说明：

- a) 对导致使用“待定”的情形进行描述（为什么答案未知），以便问题能得到解决；
- b) 描述为排除“待定”应采取的措施、由谁负责排除以及何时必须排除。

4. 一致

这里的一致是指内部一致性。如果 **PRD** 与某些更高层的文档（如，概要设计文档）不一致，那么它是不正确的。但是这个不是一致性所涉及的，这个是外部不一致。

当且仅当在 **PRD** 中描述的任何单个需求的子集之间相互不矛盾，**PRD** 才是内部一致的。**PRD** 中可能存在下述三种类型的矛盾

:

- a) 显示世界对象的规定特征可能相互矛盾。如：
 - 1) 报告输出的格式在一个需求中是表格形式，而在另一个需求中是文本形式；
 - 2) 一个需求指出所有的灯是绿色，而另一个需求规定所有的灯是蓝色。
- b) 在两个规定的行为之间可能存在逻辑上的或时间上冲突。如：
 - 1) 一个需求规定程序将对两个输入相加，另一个需求则规定程序将对这两个输入相乘；
 - 2) 一个需求指出“A”必须总是在“B”之后，而同时另一个需求中要求“A和B”同时发生。
- c) 可能两个或更多的需求描述现实世界的相同对象，但使用不同的术语。如，在一个需求中程序对用户输入的请求成为“提示符”，在另一个需求中称为“提示”。使用标准术语和定义可以改善一致性。

5. 具有优先级

如果 PRD 中每条需求(用例)赋有标明其优先级的标示，那么该 PRD 便按照优先级进行分级。

通常，与产品有关的所有需求并不具有相同的重要优先级性。某些需求可能是基本的（即核心需求，最能驱动利益增长的需求，最能达到特定 KPI 的需求），而其他的可能是所期望的需求，可有可无。

PRD 中的每个需求宜予以标识，以使需求在这方面的差异清晰和明确。

另一种需求分级的方式是区别如下基本的、有条件的和可选的需求类别：

- a) 基本的——除非表示同意并满足了这类需求，否则产品将不被接受；
- b) 有条件的——表示这类续期会增强软件产品，但是，如果缺少这类需求，也不会导致软件产品被拒收；
- c) 可选的——表示该类功能需求可有可无，这赋予开发一定开发弹性。

6. 可验证

当且仅当 PRD 中的每个需求是可验证的，PRD 才是可验证的。当且仅当存在某个有限的成本、有效的过程，人或机器依照该过程能够检查软件产品满足某个需求，该需求才是可验证的。一般说来，任何有歧义的需求都是不可验证的。所以一般保证了“无歧义”就可以保证可验证，即关注“无歧义”是最重要的。

不可验证的需求包含诸如“工作良好”、“好的人机界面”和“通常应该发生”之类的陈述。因为不可能定义“良好”、“好的”和“通常”，因此，这些需求不可能验证。陈述“程序应绝对不进入无限循环”是不可验证的，因为理论上该特性是不可测试的。

可验证陈述示例：

程序输出应在时间开始 20s 内达到 60%，在 30s 内达到 100%。

这样的陈述是可验证的，因为它使用了具体的术语和可测量的数值。需求描述一定要定量。

如果不能设计出一种方法，以确定产品是否满足某个具体的需求，那么该需求宜被删除或被修改。

7. 可修改

当且仅当 PRD 的结构和形式能够对任何需求进行容易、全面和一致的修改，同时保持该结构和形式，PRD 才是可修改的。一般地，可修改要求 PRD：

- a) 具有连贯、方便使用的结构，包含目次、索引及清晰的相互引用；
- b) 没有冗余（即，相同的需求在 PRD 不应当出现在多处）；

c) 分别地表述每个需求，而不与其他需求相混淆。

尽管冗余本事不是缺陷，但它容易导致错误。尽管冗余偶尔可以有助于 **PRD** 的可读性，但当对存在冗余的文件更新时，可能会引起问题。例如，可能对出现多处的某个需求仅在一处做了修改，那么使得 **PRD** 内容不一致。当需要冗余是，**PRD** 宜包括一个清晰的交叉索引表，已增加其可修改性。

8. 可追踪

如果 PRD 每个需求的来源是清楚的，并在将来编制或增强文档的过程中便于每个需求的索引，那么该 PRD 是可追踪的。推荐以下两种类型的可追踪性：

a) 逆向可追踪性（即，到以前的开发阶段）。这依赖于每个需求清晰地指向其在早期文件的来源；

b) 正向可追踪性（即，到由 PRD 产生的所有文件）。这依赖于 PRD 中每个需求具有唯一的名称或索引号。

当软件产品进入运行和维护阶段时，**PRD** 的正向可追踪性尤为重要。随着代码好设计文档的修改，最要紧的是能够确定这些修改可能影响的全部的需求集合。

可追踪性好使得需求变更更加容易。

六、 参考文献

【1】 计算机软件需求规格说明规范 **GB-T_9385-2008**

【2】 <http://baike.baidu.com/view/706238.htm>

FAQ:

1、需求文档到底谁会看？开发，设计，前端，测试，都会看吗？

-都会~需求文档是贯穿整个产品生命周期的。

2、需求文档到底包含什么？一个用例列表+各个用例的详细描述。除了你说的这些，还应该有什么？

-用例列表+各个用例的详细描述是必须的，以全面为标准来讲 应该包含：目录、版本修订记录、背景、前言（引用文档列表、术语列表(定义、简写

和缩略语)）、产品综述或概述、用户界面(腾讯是交互稿)、功能需求(一般由用例形式逐个展现)、非功能

需求。

而项目进度等和项目有关的东西，按照规范的来讲是应该在放在项目计划书里面。但是也可以放在

PRD 里面。