

刚入行的产品新人，你其实可以写一份合格产品需求文档

产品需求文档是产品项目由“概念化”阶段进入到“图纸化”阶段的最主要的一个文档，其作用就是“对 MRD 中的内容进行指标化和技术化”，这个文档的质量好坏直接影响到研发部门是否能够明确产品的功能和性能。

一、简述

产品需求文档是产品人员非常核心的基本功！是协调研发、测试、UED、业务非常重要的重要工具。但是，往往很多新入行的 PM 与互联网领域的 PM，产出的文档往往不尽人意，主要体现在：

- 缺乏逻辑，语言啰嗦不精练；
- 通俗的用词过多，整体显的不专业；
- 无法将字段的数据结构、逻辑关系清晰的表达出来；
- 缺乏开发思维；

而出现这种原因在于：

- 新人入职，没有经过严谨的文档撰写、流程设计训练；
- 大多数的 PM 非研发出身，对前后台的业务逻辑、数据结构了解不清晰；
- 分工细，版本迭代迅速，对功能点理解不够透彻；

完善的产品需求文档，不但利于 PM 对所承接的功能进行有效的管控，将业务逻辑梳理清晰，对分解的功能点，进行协调测试、研发、设计、运营同时开展工作；

导航



搜索文档



标题

页面

结果



1 产品概述

1.1 业务背景介绍

1.2 产品功能概述

1.3 产品前景描述

1.4 产品整体流程/逻辑关系

1.5 面对用户

1.5.1 供应商

1.5.2 后台管理员

1.6 名词解释

1.7 参考文档

2 功能需求

2.1 供应商应用

2.1.1 商品上传规则建立

2.1.2 个人违规商品管理

模块化的功能点（倒爷当年做的一个功能）

虽然，不同的公司拥有不同的产品需求文档模板！但，不论模板格式如何，文档的本质在于：“有效的将功能清晰的表达出来，并且能支撑后续的业务交接与版本迭代参考，对上与下进行价值传递。”并且，随着平台业务的发展，归档、仓储起来的业务文档，便是极其有价值的知识库，里面汇总了各个时期里 PM、研发、测试、项目经理、设计…对业务是如何进行思考，对文档研究的本身，侧面也反应了企业是如何将战略进行细节落实。

而，“业务描述、功能描述、其它需求”是组成产品需求文档非常重要的模块；本篇章，将以通用版的角度，对这些模块行介绍；

二、业务描述

任何的业务或需求，都有业务提出方，业务提出是相关的业务部门或产品经理自身。

业务来源的本质，便是希望通过这个业务解决那些实际的问题，达到提升某些转化率或某些目的；业务描述清晰的表达出来即可，不需要多复杂，但常规包括：

业务背景
产品功能概述
产品前景分析
产品功能整体流程
产品逻辑关系
面向对象
应用对象
名词解释
参考文档

上述的这些层面，以通用版的角度，将产品的价值传递给研发方与业务方，实现之间有效的衔接。

为什么，我们需要进行业务、功能、概述这些偏宏观不实际的描述呢？这样不是很麻烦，且浪费时间？

我们要知道，每新增或删除一个功能，狭义来看也没啥大不了。但站在宏观的角度去看，功能研发是需要耗资企业运营成本。如果处理不完善，浪费运营成本同时，甚至影响整个用户体验与开发规则。

身为产品人员在未传达产品的业务价值前提条件下，便强势驱动研发人员进入开发阶段，这是错误的！我要是研发，我也会拍死这位 PM。那么业务描述的本质便很清晰了，便将业务价值，传递给团队成员。

另一方面，非常多的企业，内部的项目流程是不完善的，且并非每一位研发人员都是善类。产品经理往往需要兼备着项目经理的职责，推动着项目实现研发上线。在这种情况下，如果业务价值描述不清晰，功能在开发与上线后出现问题，这个锅注定是要背的。

BTW，这些我就不都说了，自己工作中慢慢积累！

下面，我对组成业务描述的组成元素进行描述：

业务背景描述：

这里，你必须将业务提出方描写出来，并且细致到业务方为什么将这个需求提出来！为什么？一方面，你要告诉研发人员，你为什么设计这个产品或功能，这个需求从来源到设计是有原因的。另一方面，拉上相关业务部门，你至少不是一个人在战斗。

产品功能描述：

对当前功能进行概述，所设计的产品或功能的功能模块，新增、完善、优化那些产品功能；

产品前景描述：

本产品或功能，希望对那些转化率指标或实现那些目的；

产品的整体流程：

Visio、Axure（Axure 画的流程图好丑）。

通过而言，简单的需求将主业务流与逻辑关系流表达出来便可以；但涉及复杂的业务，便将产品或功能涉及的主要流程绘制出来；而流程目的，主要是清晰的将前后台的逻辑关系与数据结构表达出来；一方面方便开发理解业务与数据流，另一方面也方便产品人员梳理自身需求的业务逻辑；利于后续与研发进行沟通。

具体的流程数量，根据业务的复杂程度决定，一般只需要将核心的流程绘制出来便可；

前台：主要是交互、数据流程；

后台：主要是业务逻辑判断、数据流；

前后台的流程凑在一起，能清晰的看到前后台的模块之间，是如何进行耦合的，数据储存、提取、处理、分析。

功能框架：

Mindjet Minmanager、Xmind 画的框架图好丑。

框架图的意义在于，能让查看或了解业务的人，全方位的了解功能之间的功能点的逻辑关系。同时，一份优秀的框架图，能让 PM 站在全局的基本面上，对个人所负责的产品进行全局的规划，对前后台的功能进行把握，达到支撑平台业务。

产品架构：对前后台的各个系统与管理模块的逻辑关系，一般是对业务极其熟悉的业务构架师与资深的产品总监搭建，里面涉及每个接口如何进行对接耦合。

功能架构：所负责的产品或功能的前后台功能的逻辑关系，简单点的就是一个产品或功能的前后台，大一点就是一个系统涉及的功能点之间的耦合。

功能框架：功能点所涉及的逻辑关系。

功能结构：功能点所涉及的逻辑关系。

而“架构、框架、结构”区分在于，所负责的业务究竟有多大。但不论如何，它们的表现的原理是一致的。将分解的功能点，之间是如何联系的功能结构关系清晰、简练的表达即可。

关于架构，包含“功能分解、面向用户”就够用了。若再深入，可将分为：“应用对象、BI 分析（BI 需求也写上去）、系统集成…”。后续可根据 BI 数据，对产品进行版本迭代与优化。

面向对象

表达产品或功能主要是为那类用户服务的。将面向用户是谁，拥有哪些权限清晰的表达出来即可，对个人进行功能设计也有很大的帮助。

应用对象

本功能需要在那些应用端或版本进行上线，清晰的描绘出来，方便后续进行业务交接。

名词解释

将本次文档涉及一些奇葩的明词进行解释，这点很重要！有些 PM 喜欢将一些非常简单的内容包装成非常牛逼，让人看起来很难懂，而事实上也就做哪些一件简单事，但是看的人会很痛苦：看 PRD 时会想：“这玩意，究竟想表达什么。”

参考文档

将所做的本次功能，所参考的那些文档，附属上来；目的在于，方便后续的业务方、研发方进行查看。

三、功能描述

功能描述能否描写清晰，描写清晰，开发找茬都不怕了。如何才能完整的对功能点进行描述呢？围绕三个点“功能是谁？功能来自哪里？功能要到哪里去？

同时，功能需求主要分为核心功能、其它功能。不论是核心功能还是其它功能，都可以由以下元素构成：

功能名称

面向用户

用例图（Axure、mocking（适合移动端进行敏捷性开发））

前置条件

后置条件

功能简述

详情描述

而具体的功能描述内容，则根据业务（功能点）的复杂程度，进行筛选描写。可以全写，也可以不全写。但务必记住：不论何种方式，目的在于将业务价值完整、清晰、有条理的传递给查看文档的参与角色。

功能名称（我是谁）

本功能在系统里的命名。

面向用户

本功能的使用对象。（在前台，功能的参与者是少数的；但后台与企业级应用里，功能的参与者是多个的）

用例图

表达功能在表现层的逻辑图；可以是传统意义上的用例图，或者是简化版的原型图、流程图；

前置条件（我来自哪里）

使用该功能的前提、逻辑关系说明；公司大了后，每个开发都只写个人所负责的业务，所以一定要将每个模块来源都清清楚楚的表达出来，方便开发之间的衔接。

后置条件（我要到那里去）

使用该功能后，对业务、数据功能，产生的影响与结果；

功能简述

描写本功能需要实现的商业价值或目的；

详情描述

将功能”我怎么来，我怎么去“清清楚楚的表达出来。变成计算机逻辑便是，页面布局、操作逻辑进行详细的说明。常规而言：

前台：主要是字段、交互逻辑组成；

后台：主要是判断逻辑、列表表单、查询条件、交互逻辑组成；

四、其它功能

其它功能目的在于，功能描述针对于本次产品功能的核心业务，而其它功能针对于触发或需要其它功能变动的业务。功能描述清晰的让开发了解核心，而其它功能便让开发清晰的了解非核心。

而其它功能，主要由以下内容组成

其他接口

对其它系统产生“字段、业务流程”进行说明；本次产品或业务，对前后台那些非主流程模块产生影响；

系统风险评估

当前设计的功能存在哪些缺陷、注意事项与后期的功能拓展如何解决这些问题；

其它需求

对一些非核心的功能点进行详情描述。如：一些需过滤的关键字、新增某个栏目字段。

五、综述

通过上述内容，能以通用版的形式，清晰的将所负责产品与功能表达出来，而业务描述、功能描述、其它功能。是产品需求文档重要的组成部份，将产品需求较为全面、有效的描述出来。

同时，能训练 PM 逻辑思维，提升文字表达能力、业务理解能力，从整体上让 PM 在需求管理上，明显更加专业，所负责功能的逻辑关系、数据流的来与去都能很好的把控。

六、附语

不论是什么格式，倒爷坚持一个观点，适合团队才是好的模板。当前很多的公司在进行 MVP 迭代的时候会使用 Axure+内容描述的形式。虽然，这种形式，是很难将逻辑关系表达清晰，同时会有非常多的思维漏洞。在进行文档归档时，也很难对根据关键字进行检索。但，确实挺适合进行 MVP 迭代，出现问题修改起来也方便，这种方式比较适合项目流程完善的企业平台使用。

而在敏捷性开发汇总，倒爷习惯流程图+功能框架（功能点）+Axure（原型图绘制），从核心的业务流开始，逐渐迭代至功能完善，这个过程也将文档补齐。

但有些公司会在 EXCEL 里进行需求文档撰写，进行版本管理（这个也不错）。

但，作为新人，需要记住：你能写复杂的东西，简单的东西也能写；但当然一开始只写简单的东西，那你一辈子只能做简单的东西。

大道至简，简难而繁易；经历过复杂的训练与要求，才能简化再简化。

作者：倒爷，微信：ftl_keen。

本文由 @倒爷 原创发布于人人都是产品经理。未经许可，禁止转载。