**技能GET：高效产出PRD的“三不”原则**

2017/03/20阅读 1.8万 评论 3收藏 126

简单来说，拒绝高保真、拒绝废话、拒绝重复，把自己的时间花在产品的核心驱动力上，减少对无用功的投入，降低单调重复的操作频次，才能保证高效产出有价值的牵引力。

现在业内对于产品需求文档（PRD）普遍认同的交付形式是Axure的原型+文本，当然你也可以说是Sketch或者PPT等形式，不论是哪一种，都是用可视化替代Word（喜欢看文本的程序员可能是假的）。

本文仅陈述个人对于产品需求文档的看法。

**一、PRD的主要构成**

**1.概要**

文档变更记录（版本迭代、版块及功能说明、记录人等，其实就是简化版的功能需求清单）、全局规则说明（升级逻辑、异常逻辑、交互说明、输入限制…）、名词解释（积分、金币、白银会员…）

**2.业务说明**

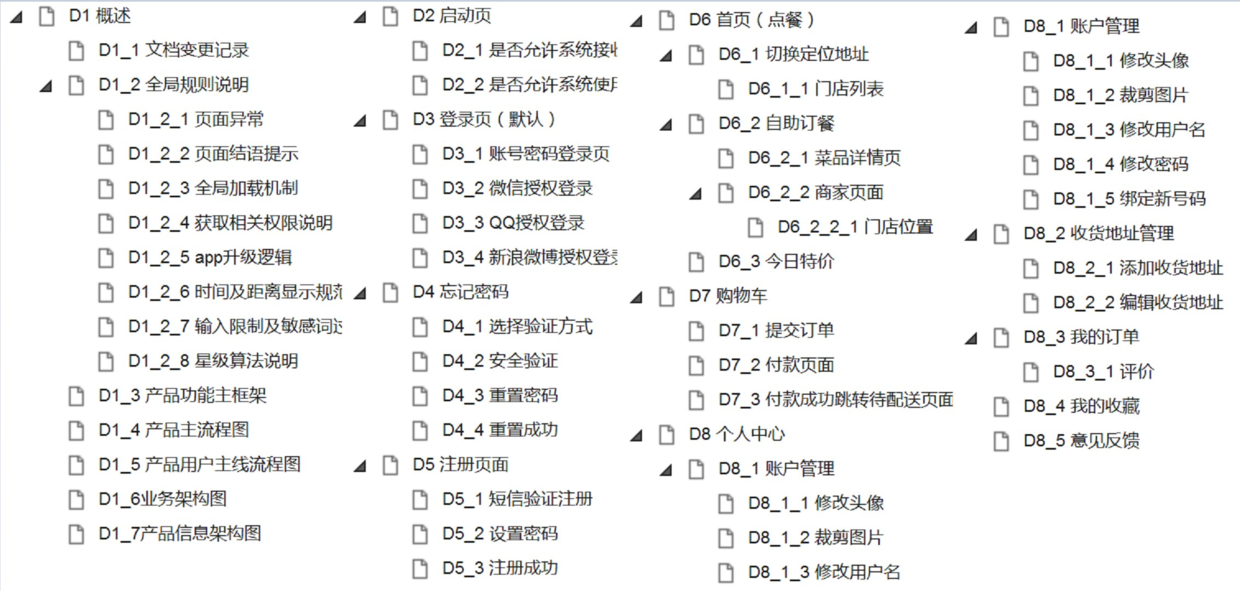
产品功能主框架（功能/信息结构图、业务结构图/流程图、一级页面流转图）、权限说明表（角色权限、数据权限、功能权限）、页面交互/功能模块元素构成、规则说明

**3.原型图**

各页面的线框图、基本的交互形式、文本注释（缺省状态、字段名称、流程说明、等待时长说明…）

**4.非功能性需求**

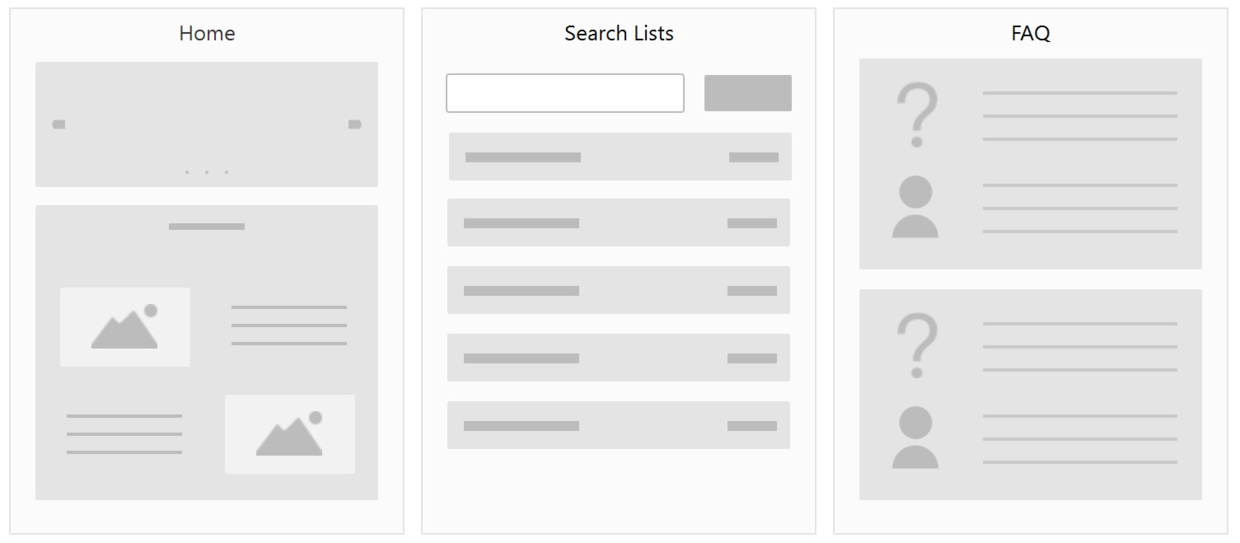
地理位置获取、Push推送机制、敏感词过滤、CDN缓存策略、本地文件存放策略…



如果这是产品的第一个版本，则需要在产品的功能结构、业务流程等逻辑层面上花更多的时间去说明，一方面降低与开发人员、项目经理、UED团队等沟通成本，让整个团队快速理解并取得认同，另一方面作为证据和备份，在后期产品开发出现偏差时能及时调整复位。

**二、抓住本质的“三不”原则**

**1.不做高保真**

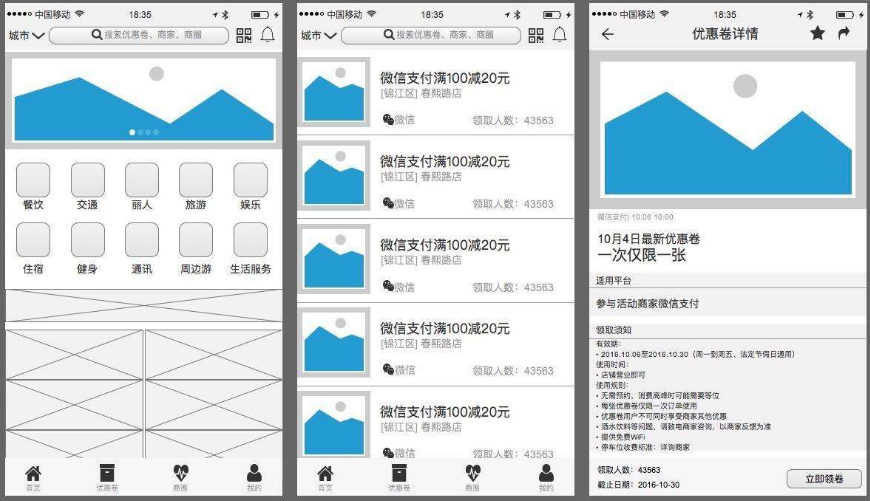


通常喜欢做高保真的产品汪都是在设计方面有一定基础的强迫症，把所有视觉细节和交互特效都堆砌上去，如果是拿出去给外人做展示、给用户做测试，效果自然很好，但通常我们做原型只是为了内部的沟通交流。

在产品规划初期，如果做高保真原型，一旦在评审或后期讨论中遇到需求变更，修改的成本非常之高，严重影响到产品的设计及开发进度。更何况身为产品把原型做成高保真，让团队中的UI情何以堪，是沿用你的风格还是自创一套呢。

所以通常情况下，只需要用线框图将功能架构表示出来，一旦核心功能确定下来，线框图的设计是非常快的，做完一稿立马就能跟大家探讨，制定修改方案，反复几次后就能把产品的设计思路梳理清晰，随后UI和开发人员才能快速跟进，减少后期需求大变更的风险。

换句话说，产品的设计流程应该符合用户体验五要素：战略层→范围层→结构层→框架层→表现层，此时此刻如果产品负责人把精力花在页面的呈现效果上，就会缩减其在战略、范围、结构上的考虑，本末倒置容易把产品带进沟里。



**2.不写废话**

在产品进行评审的时候，开发人员常常会提各种各样技术角度的问题，比如“数据从哪来？”“这里有几种角色？几种权限？”最后问题及目标都搞清楚了，散会。结果到了具体的开发环节，他们又会来问你相同的问题，“这个角色能看到这些数据吗？”所以在需求文档中把关键性的逻辑用文本/图表展现出来是非常重要的。

但如果所有功能比如返回跳转都要说明清楚的话，文本内容会非常多，开发人员依旧是没有耐心看的，因此要精简产品文档中的注释，少写废话。

比如一些你我皆知甚至是用户都了解的规则，就没必要写在原型图上了。比如“密码显示\*\*\*”，“若用户没有输入密码，点击登录时则toast提示用户未输入密码”，这样的文本说明纵然很细致，但确实是白写的，而什么内容是有必要的呢？

比如你需要在输入框右侧设置密码可见的切换按钮，或者说明输错三次之后当日无法登录的规则，又比如对账号密码有特殊的格式限制等等，这些才是开发人员不知道的设计思路，需要你在文档中详细告知。

如果你跟团队成员已经有了足够默契，即你知道对方的开发习惯，对方也了解你的设计套路，那么即使你没有说明“点击删除时弹出模态对话框以确认操作”，开发人员也会很自然地把这个反馈加上，如此一来，团队的内部损耗就降到最低，产品的规划、迭代都会进入高效运转的状态。

相信我，就算你把PRD写得极其精炼，团队成员也不一定会完完整整地看完，更要命的是，即使他们都遍历了一次，也会忘记其中的一部分，只要他们忘了就必定认为是你没写…（其实自己写的，自己也会忘）



**3.不重复创作**

全局都统一的逻辑就在全局中说明规则，没必要在每个页面中嵌套，不要重复发明车轮，这可以帮助我们省很多时间。

比如手势操作、网络异常情况、输入限制等，有些是可以直接复用的，有些可以在原来的基础上根据产品性质再调整，就像设计师的素材库，产品经理也可以积累一套自己专用的控件库。



不重复创作还体现在设计风格、版式上，很多人都执迷于原创，拒绝抄袭现有的类似产品，其实完全没必要，理论上来说，不存在跳脱于现存所有设计的设计套路，所以还不如多看多抄。

还有一个更重要的原因是，如果已经有一款竞品打开了市场，抢占了用户的心智，培养了用户习惯，此刻最该杜绝的就是独树一帜，试图挑战用户的操作习惯，除非找到了用户新的痛点，设计一个全新的功能，比如Snapchat的阅后即焚，不然就应该遵守已逐渐固化的交互形式，尽可能降低用户转移过来的成本。

所以支付宝的聊天界面几乎是像素级复制了微信，所以共享单车的操作流程及页面都近乎一致，从用户体验上来说，这样才能体现出关怀。

微软就是因为培养的用户习惯根深蒂固，所以每次大升级都要付出巨大的成本，遭受用户长时间的质疑和抵制，纵使操作体验确实更好，但在用户看来就是关怀不够，还要花时间重新适应。

**三、写在最后**

简单来说，拒绝高保真、拒绝废话、拒绝重复，把自己的时间花在产品的核心驱动力上，减少对无用功的投入，降低单调重复的操作频次，才能保证高效产出有价值的牵引力。

做产品如是，个人成长亦如是。

END.