

一、阅读需求

拿到一个功能，第一件要做的事情就是完整地阅读需求。在阅读需求时，尽可能地做到以下几点：

- 不要站在开发者的角度去理解需求，而要站在用户的角度。第一遍阅读需求时，一定不要想应该怎么做实现，而是立足需求本身，整理需求的逻辑，如果发现了不合理的地方，及时与产品沟通；
- 对于复杂的业务，如果理不清流程，可以画一个流程图来辅助理解。（善于使用工具来解决复杂问题）

tips：

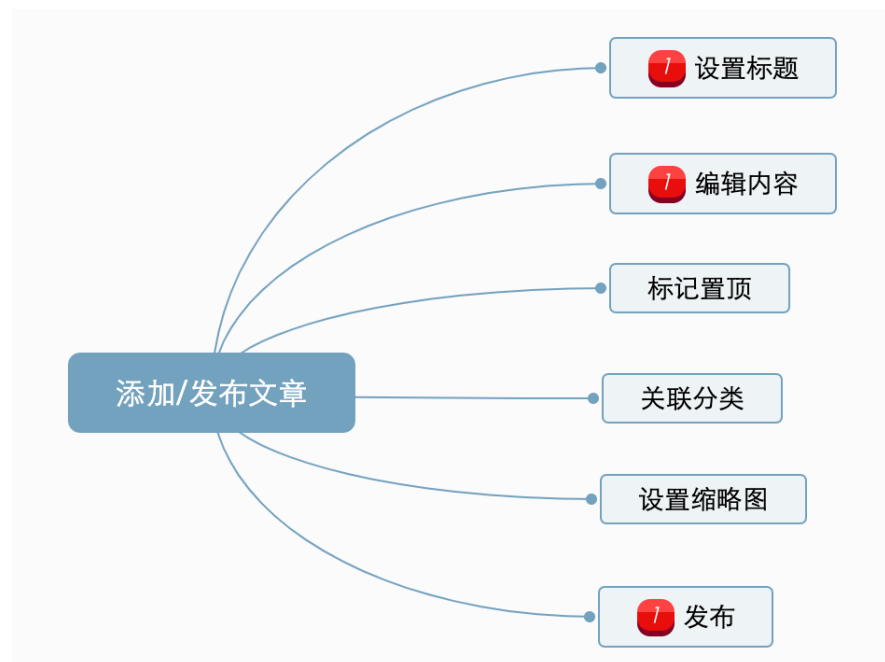
1. 为了验证你是否真的做到了以上的这些点，前期可以给自己一个硬性规定：每一个功能点的需求，阅读完成后，都需要整理至少5个问题与产品进行讨论。
2. 在这个阶段，你要勤于与产品沟通，以防出现因为前期需求理解不到位导致的返工问题。除了理解产品在需求设计的一些用心之处外，通过集思广益，你还可以通过表述自己的观点，帮助产品从更多的角度思考问题，改进产品。

二、拆分需求

现在，假设你已经完整地梳理了需求的流程，并对需求有了较深的理解。接下来，可以想想实现的那些事了，但是，不要着急具体的实现，而是将需求进行拆分，将实现的过程梳理出来。在这个阶段，你需要思考这些问题：

- 哪些功能点是重要的，哪些是次要的？一般情况下，在流程线上的功能点往往是重要的，因为缺少这些功能点会导致流程走不通，所以需要优先实现。其他不在流程线上的功能点，我们称它们为可拆卸的，可以在需要的时候把它们装上，也可以随时将它们卸掉；
- 哪些功能点是容易的，哪些是困难的？除了将功能点按照重要程度维度来拆分，还需要从难度维度继续拆分，然后按照先易后难的原则来进行实现。

举例来说，我们要实现一个博客的添加/发布文章功能，对于读者来说，要看的无非就是标题和内容，所以对于文章是否置顶、属于什么分类、有没有缩略图，在第一个版本里，都是不重要的，而且在后续的版本这些功能点可以被随时装上。



拆分需求后，我们可以将各个功能点细分出来，然后按照下表的顺序一一实现：

功能点	重要程度	难度
设置标题/编辑内容	重要	容易
发布文章	重要	困难
标记置顶	不重要	容易
关联分类	不重要	困难

三、理解数据结构

表面上我们在使用软件、操作功能，其实最终操作的是数据，亦或说信息，理解数据结构尤为重要，有一个叫Eric Raymond的人，写了一本书《大教堂与集市》，总结了Linux的成功之道，书中就这么说到：

精巧的数据结构即使搭配笨拙的程序代码，也比精巧代码加笨拙结构的组合要强得多。

在这个阶段，你需要做这些工作：

- 梳理出功能的数据结构，并为它们做好关联关系。如果关联关系太复杂，可以借助关系图来辅助理解；
- 站在设计者的角度思考数据结构的合理性，而不是站在实现者的角度。想一想，如果让你来设计，你会将数据结构如何组织？

tips：

1. 对于后端而言，数据结构是表结构；对于前端而言，数据结构是前后端接口约定的返回值；
2. 在这个阶段，你需要与数据结构的设计者多多沟通。除了感受种种结构设计的精妙之处外，你还可以提出你的想法与方案，帮助设计人员改进设计。

四、功能设计

接下来，我们要对整个功能进行全局的设计，在这个阶段，你需要考虑这些问题：

- 哪些功能点是公共的（或可复用的）？找到公共的部分，优先实现它们，为后面的具体功能做准备；
- 界面的风格、交互效果是怎样的？如果不知道，提前与产品、美工沟通清楚；
- 确定每个功能点的实现顺序。一般情况下，尽量按照流程顺序来设计功能点的实现顺序，例如：先实现添加文章功能，再实现文章列表功能，因为后者依赖了前者。

tips：公共的部分可能是样式、工具函数、公共组件、甚至是公共页面。

五、功能的实现要点

每个人实现功能有不同的方式与技巧，纵使你可以尽情发挥，但不要忘记功能在这些方面的规范：

- 加载态。信息在呈现给用户之前是需要漫长的等待过程的，在这期间，一定不要给用户呈现空白内容，设计一个好的加载效果，明确告知用户：请放心，我在努力加载；
- 提交态。用户提交信息到远程服务器是需要漫长的等待过程的，在这期间，要注意明确告知用户正在提交，还要注意将提交入口关闭，防止用户在这期间多次提交，使得服务器产生多余的数据；
- 空状态。服务器不一定总是有信息呈现给用户，如果没有，要注意明确告知用户：抱歉，我还没有数据可供呈现；
- 错误态。服务器不一定总能成功地给用户的操作给予反馈，如果失败了，要注意明确告知用户失败了，最好能同时告知失败的原因和恢复手段；
- 交互态。功能的活泼之处在于交互，每一个可交互的元素，都需要在用户操作时，给予适当的反馈（可以是颜色变化、大小变化或动效等等）。

tips：想要将用户体验做好，需要开发者想得更多，而事实是，不是所有开发者都知道那条条框框的用户体验理论。别担心，并非没有解决之道，你需要真正地、设身处地地将自己作为一个真实的用户来思考问题。总之，做用户想要的，而不是容易做的。

总结

1. 在需求的整理阶段要与产品沟通，对不理解的问题进行提问，或提出建设性意见；
2. 在数据结构的整理阶段要与后端沟通，对不理解的设计进行提问，或提出建设性意见；
3. 在功能设计阶段要与美工沟通，对UI、交互进行确认，或提出建设性意见；
4. 在功能的实现过程中，碰到任何问题，与他人进行沟通，提出问题并总结解决办法，形成文档；
5. 每个阶段，你需要把自己当成一个用户，这个用户有多“傻”，你的功能实现得就会有多完备。