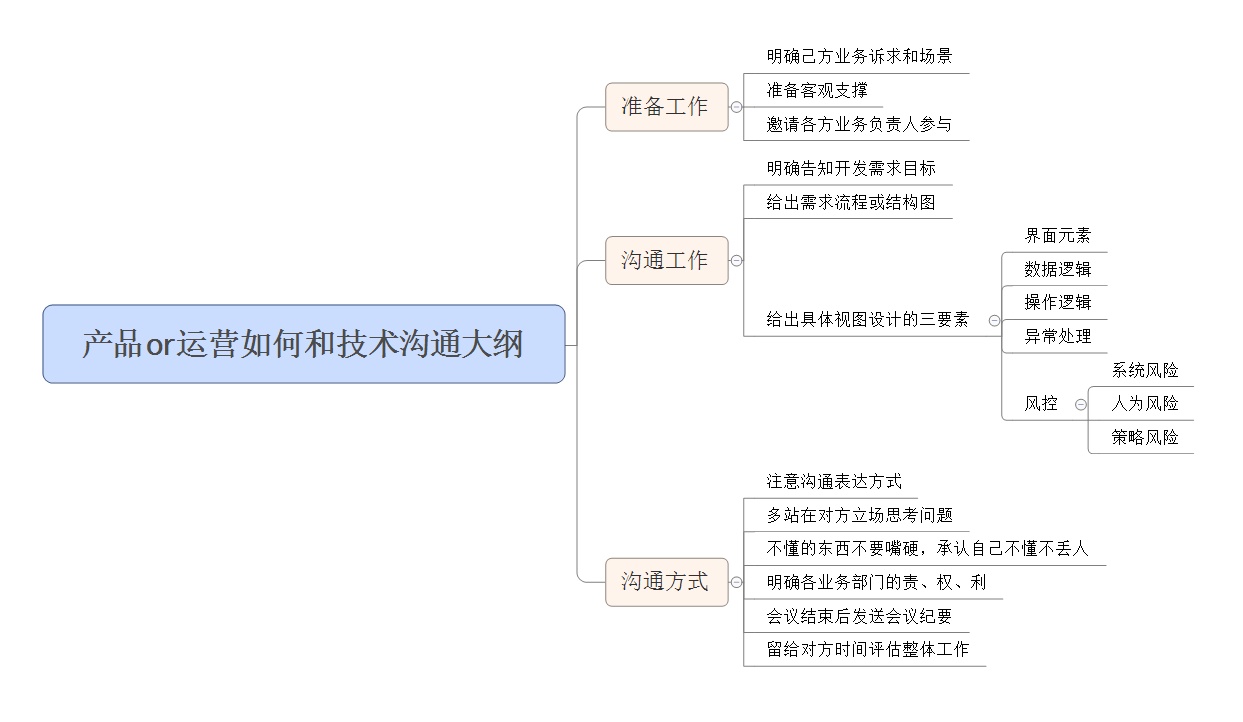
# **产品or运营到底该用什么样姿势和技术沟通？**

作者：[姚伟](http://gad.qq.com/user/index?id=743910)

链接：<http://gad.qq.com/article/detail/38440>

这篇文章算是憋得蛮久的了。在小公司做运营的时候，由于人力问题，提需求都是直接到技术那边，经验不足的情况下，沟通过程自然不是特别顺畅，后来到了现在公司之后，运营的需求一般都要经过产品经理，最后由运营和产品一起过技术。这篇文章算是对这段时间经验的总结吧，希望能帮助到看到这篇文章的各位。



**准备工作：**

一、明确业务诉求和场景

这点非常非常重要！新人最最容易碰到的问题！

很多时候需求不是来自于你，而是来自于同事或者上级，他们的需求很多时候不能算真正的需求，你需要去理清楚业务场景和他们的目的，也许需要花比较久的时间，但是对于业务理解和后续需求沟通有巨大帮助。否则贸然找技术沟通需求，被几个为什么就顶回来了，反复进行沟通，不但会影响同事的工作效率，还可能会给同事留下不靠谱的印象，对于以后的合作非常不利。

关于什么是真正的需求的问题，马和汽车的例子大家都耳熟能详。再举一个典型的杀毒软件的例子，用户为什么需要杀毒软件？本质需求真的是安全需求么？其实真正的需求就一个字：爽！这个爽就包括了打游戏不能卡、不能被盗号、打开网页不要弹乱七八糟的广告、写文档不要卡等等。所以对比一下360和卡巴斯基这些你就知道了，360厉害不仅仅只是因为他免费。

二、提供尽量客观的依据

1、有业务数据尽量提供业务数据作为支撑

2、没有业务数据寻找第三方数据平台作为支撑

3、没有数据提供竞品作为参考

三、拉上双方的负责人

涉及到一些比较大的需求的时候，一定要拉上两边的领导，让他们参与到需求中，因为最后都是要他们拍板的。

最关键是，占用人力资源比较多的时候，可能因为各种各样的原因，你很难去推送这个需求落地，拉上自己的领导可以很好地帮助你。

**沟通工作：**

一、明确告知开发需求目标

1、不要只告诉开发做什么，而不说明为什么要这么做。你以为不需要了解或者他们一看就明白的东西，往往容易产生理解的歧义。同时这样可以避免没有技术背景或者技术背景一般的人（比如我自己）替开发多想，认为这样实现简单而那样实现复杂。开发可以寻找到实现成本最低的方案，而不是盲从产品的需求定义。对于小公司来说，实现成本尤其是时间成本，是一个非常重要的问题。

2、开发参与到需求的讨论中，这样能调动开发的参与意识和积极性，加深对需求目标的理解，从而避免无用功和不符合需求的情况。

二、给出需求的流程或者结构图

在给出具体功能设计之前，如果是比较复杂的需求，一定要给流程图，主要为了加深需求的理解，形成完成的需求概念。

很多时候，运营或者产品会觉得，这么简单的东西我都说得这么清楚了，你TM怎么不明白啊？ 其实主要就是因为在这个环节上运营或者产品对整个需求的背景，结构和目标早已有了代入感，所以觉得每个细节都理所当然是这样的，但是对研发而言，他们并没有得到完整的背景信息，对细节的理解往往就出现偏差和误判。对彼此功能点的关系，相互的联系了解的支离破碎，那么实现起整个系统来难免会出现问题。

三：具体视图设计的三要素

涉及到具体app的功能时，运营或者产品提供以下三点要素：

1、界面元素：比如哪里是文字，哪里是下拉框，哪里是按钮，怎么排版，如何展现，是浮动还是静止？

2、数据逻辑：简单说就是数据从哪里获取的，要按照什么样的逻辑展现。 获取数据的接口一般是之前单独定义的，每次附上需要调用的接口即可。

按照什么样的逻辑展现，比如说游戏中心有个“热门游戏”版块，前三位可以强运营，三位之后可以根据下载和付费等数据进行一个加权算法的排序，如果你懂这些最好，不懂的话要和技术沟通，看看你希望这个地方出现的内容需要体现什么样的特征，然后根据技术的设计，你需要去思考，这样的逻辑是否符合用户的预期。

3、操作逻辑：界面上可以进行操作的有哪些元素，哪个可以点击，可以选择，可以滑动，操作后出现怎样的反馈，比如显示浮层？打开新页面？

操作反馈在游戏设计里面尤其重要。 想起以前刚做游戏策划的时候，那个时候设计系统总是自嗨于规则的设计，忽略反馈的设计。仔细想想，就算规则设计得再好，如果一刀砍到BOSS身上没有声音、没有血迹，对于用户来有多蛋疼。

4.、异常处理：这个非常重要！（我曾经因为这个被我们的产品经理教育过，这可是用户体验的关键呐！），最简单的比如用户注册信息填写完成，点击提交之后，网络突然波动，这个时候怎么办？让用户看着界面转菊花吗？是不是应该弹出一个提示框，告诉用户网络异常，请稍后重试的提示？

5、风控：这里的风险实际上可以分为三类，系统风险、人为风险、策略风险

系统风险：打个不恰当的比方，安卓开放系统下，你的游戏就是有可能被劫持，本地客户端就是有可能被破解，无法避免

人为风险：风控主要是针对的这个部分，比如运营活动规则设计漏洞、负载均衡不合理、未考虑兼容性问题等等

策略风险：同样无法避免，这里不多解释了

**沟通方式：**

1、注意表达时候的措辞和语气，大家都是合作关系，解决问题才是首要目标，对事不对人；

2、多站在对方立场考虑问题，多想一想为什么他们会这么想；

3、不懂的东西不要嘴硬，承认自己不懂没有那么难堪；

4、明确合作部门的责、权、利；

5、需求讨论结束之后写一封会议邮件，周知参会人员，描述清楚需求背景和会议结论，遵循“who do what by when”的原则即可；

6、给开发留有时间评估工作量，不要今天提需求，明天就要实现。