



链滴

为什么开发的时间永远不够

作者: [zjhch123](#)

原文链接: <https://ld246.com/article/1545487353108>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

为什么开发的时间永远不够

正式参加工作已经6个月了。在这6个月中，我陆陆续续的直接或间接的负责过一些大可小的项目。而，我在项目制作过程中发现咱们程序员们的时间似乎永远也不够用。我尝试着从多个方向进行了思考，并归纳总结成这篇文章。

在文章之前，先介绍一下各个关键词：

1. 开发：就是我们，程序员。当然，我这里特指负责运营支持的前端，其他责任岗位的前端可能可部分指代；
2. 运营：就是策划，负责整理需求并提给开发；
3. 视觉：负责将运营的想法输出成具体图像，其载体为Photoshop或者Sketch；
4. 测试：开发完成后负责测试开发页面的同学，活动的上线流程也由测试进行。

1. 开发眼中的需求与运营眼中的需求

对于开发来说，项目起始于“需求评估”。能快速仔细的逐一核对需求点是项目初始阶段最关键的一步。在需求会上策划应该准确的说出自己的想法以及具体需要实现的功能，开发也应该仔细分析其中利弊，给出相应的解决方案，评估准确的开发时间。

然而，经常发生的情况就是运营和开发之间的沟通没有到位，比如：

1. 运营说要有“推送”功能，那么作为开发自然想到的“推送”是APP的推送，可是实际上运营想要是短信 + 站内信推送。
2. 运营说要给新用户展示一张专用的优惠券，开发认为新用户就是纯新三无用户，而实际上运营认为新用户是没有下过单的用户。
3. 运营说，这块地方展示商品，开发认为展示商品就是纯展示逻辑，运营觉得商品都展示了就应该能击商品进入商品详情页。

还有很多很多这样的例子。其实我们在核对需求的时候一要仔细听，二要分析其中的逻辑，有什么分点一定要理顺，有什么问题一定要在需求会上直接提出，千万不能往后拖延，因为拖的时间实际上都自己的开发时间。

运营说完需求之后，就需要开发给出具体开发时间。此时开发千万不能过度乐观的评估自己的开发时间，如果实际需要2天工作量的话请一定要说3天，以免因为各种不可控的因素而导致自身背锅。总结一点就是，在排期阶段开发实际上应该在准确评估工作量的情况下再多给自己预留半天到一天的时间于周转，这点很重要，可以放到后文去说。

2. 天马行空的视觉

在需求会结束之后开发往往会和视觉碰头，确认一下整个流程中在视觉方面需要注意的点。开发也会视觉说好，哪块哪块地方做的规范整齐一些，这样不仅节省视觉时间，也能节省大量开发时间。

但是开发永远不会知道，笑着说“好好好，一定做简单些”的视觉，心里想的到底是什么。

视觉实际做出来的视觉稿可能会和之前约定好的“规范、整齐”大相径庭，可能会出现需要展示8个品的地方视觉用了5种不同的样式展示前5个商品，又用了1种固定样式去展示剩下的3个商品。也有可本来约定是“豆腐块”布局，视觉做出来的效果图变成了“瀑布流”布局...这样实际上相当于变相的开发增加了开发需求，在另一个方面压缩了开发的工作时间。【上文所说的周转时间，在这里就用到】

当然我的意思并不是在说视觉做的效果不好，而是我觉得开发和视觉应该在样式布局层面要有一个简的统一。而且视觉设计不能是一言堂，不能所有的样式都由视觉一方来决定，开发也应该参与到最的定稿流程中，也应该有一定的“否决的权利”。

这里其实还有一个存疑点，就我而言对于紧急需求，我认为应该可以在大致实现视觉样式的情况下做快速迭代上线，而不应该在紧急情况下还拿着放大镜找页面和视觉稿之间的区别。但是有的运营可能是会强制要求达到100%还原度...事实上，对于“快速上线”和“精工细作”之间的或许还需要一定周转吧。

3. 可以转运营岗的测试

开发终于赶在排期内完成页面了，接下来就进入了测试环节。

测试环节是测试的主场，很多时候一个老牌有经验的测试是完全可以这个阶段取代运营。因为测试道，这个活动用户最关心的利益点在哪里，最能吸引用户的地方在什么位置，可能会出现的问题有等等...老牌测试人员甚至可以针对不同的情况给策划提出各种建设性意见。比如：

1. 有一个商品轮播模块，运营说“1个商品轮播”，测试说“之前xxx活动用1个商品效果不好，推荐成3个”，运营说“好好好”，于是开发改。
2. 有一个发红包功能，运营说“给所有用户发2元红包”，测试说“有的用户是被风控了的，不能发2的”，运营说“对对对”，测试又说“有的用户是会员，规则里说会员需要多发5毛”，运营说“对对”，于是开发改。
3. 有一个查看用户信息页面，运营说“直接展示用户信息”，测试说“有时候可能突然取不到登录状”，运营说“好好好，加一个未登录状态展示”，测试又说“有时候可能接口出问题取不到用户信息”，运营说“好好好，加一个接口出问题的容灾页面”，于是开发改。
4. 要展示商品信息，运营说“价格显示券后价”，测试说“有的商品没有券后价”，运营说“没有券价的商品展示原价”，测试说“有的商品参与了活动不可用券”，运营说“不可用券的商品展示活动”，于是开发改。

以上场景可能是假的，也可能是发生过的。有的问题完全不应该出现在测试阶段，而应该在更早之前发现并解决。究其根本原因是因为在需求说明和需求评估阶段运营和开发就没有做好针对用户体验、控、容灾、异常情况等的沟通，导致在测试阶段的时候由测试来补充这些实际上非常关键的需求点。

这些奇怪的需求点有的很容易发现，有的却是被隐藏在整体流程中很深的节点中，需要仔细梳理才会发现。对此我认为，在开需求会时一定要仔细！仔细！不能给自己留坑，在开完需求会之后我们要据需求草稿拟流程图、容灾图用于梳理流程，这样就能在开发之前尽可能发现一些埋在流程中的坑。

4. “不怎么难”的改动

项目终于上线了！但是可怕的事可能才刚刚开始。

运营突然说 - “哎咱们的页面有没有xxxx功能啊？”，如果这个功能页面上有，那就万事大吉；如果个功能页面上没有，那运营的潜台词就是“这个功能必须要有”。

模拟场景

“那咱们能不能加一下这个功能啊？” - 运营问。

“不能” - 开发回复。

“为什么不能” - 运营问。

“因为xxxxx, xxxxx以及xxxxx” - 开发答。

“哦，那我去问问xxx(你的领导)，看看能不能做” - 运营回复。

5分钟后，你的领导向你笑着走来——

还有一种情况令人特别生气，有时候运营要改某个逻辑，他会说“就改这里这里，把XXX改成XXX，又不难。”或者说“上次那个谁谁谁都改了，你为什么不给我改？”

“改这又不难”和“xxx都给改我为什么不给改”这两句话堪称大杀器，对开发而言基本上是无语吐的同时又有一种欲哭无泪的感觉。（咱们开发都是文明人，又不能动手。

在面对繁修改需求的时候，现在的我已经不会很生气或者很焦虑，因为我知道，不管用什么情绪面对些，该修改的需求还是要改，该做的工作也逃不掉。因此我往往会说：

“好的，这个需要排一下期哦，你先把需要修改的地方列成List同步一下给视觉和测试~”

5. 不满意的领导

在第四点中我提到了由运营主导的需求新增/更改，虽然可以用排期的接口稍微缓一缓自己的情绪，是实际上这真的很让人头大。

但是如果出现领导对这个功能点不满意，或者领导一排脑袋，想要给我们的活动增加一些功能，那就直是要用吐血来形容。

由领导主导的修改是不会有排期的，一旦领导说“要xxx功能”或者“xxx要改”，那这个事情就不经过审核不经过排期完全不走流程得直接落实到个人，哪怕手上有再多的事都要立即停下去处理领的需求。再重复一遍，领导的需求是没有排期的，是要求越快越好的，这就意味着现在只要还没过夜0点，视觉、开发、测试就得一直在公司待命，视觉负责处理需求的样式，开发负责实现，测试负回归。

还有，哪怕过了午夜0点，还是有可能被抓回来加班的。

哪怕今天是周末。

说个题外话，快过年了，我们现在特别怕在年二十七、二十八这种快要放假的日子里，领导觉得我们要加某某传播活动...希望这个flag不要应验。

6. 永远背锅的前端

负责和用户做交互的永远是页面，负责做页面的是前端，因此，一但页面上有任何问题，所有人找的一定是负责这个页面的前端。

午休的时候运营找到前端，“啊呀这个页面上的商品怎么不显示了啊” - 前端起床，查看页面，发现运营在后台配置的商品出错了，于是反馈给运营，并再次手把手教他配置了商品。

半夜2点领导打电话给前端，“啊呀这个页面上怎么有一个地方显示得不对啊。” - 前端起床，开机看页面，发现是接口异常，于是反馈给领导。

周末下午客服打电话给前端，“啊呀这个页面怎么404了啊？” - 前端查看情况，发现可能是运维加某个转发，于是反馈给客服和领导。

不管出什么问题，第一个找的肯定是前端。因为前端是面向用户的，必须对这个页面的展示情况负责到底。【没办法，我们就是这么负责~

我觉得这个也无法解决。我们实际上已经讨论过好多次想解决这个问题，想建立完善的异常排查机制但是对于不懂技术的客服、运营还有广大客户来说，页面上有问题，那一定就是做页面的人负责。

这种惯性思维无法改变，也就奠定了前端永远背锅的原则。

总结

没有什么实质性的总结。只是发现踩得坑多了，就越来越有一种“预知”的能力，面对一个需求能很发现哪个功能点需要注意，哪个模块可能会有坑。这些在学校可能没办法经历到，基本上是只有在实际工作中才能学到的经验和教训。

大人往往说，有压力才有动力。当面对的压力多了可能能见到自己正在快速成长；也有可能，会见不明天的太阳。

共勉。