

点赞再看，养成习惯，微信搜索【三太子敖丙】关注这个互联网苟且偷生的程序员。

本文 **GitHub** <https://github.com/JavaFamily> 已收录，有一线大厂面试完整考点和系列文章。

## 前言

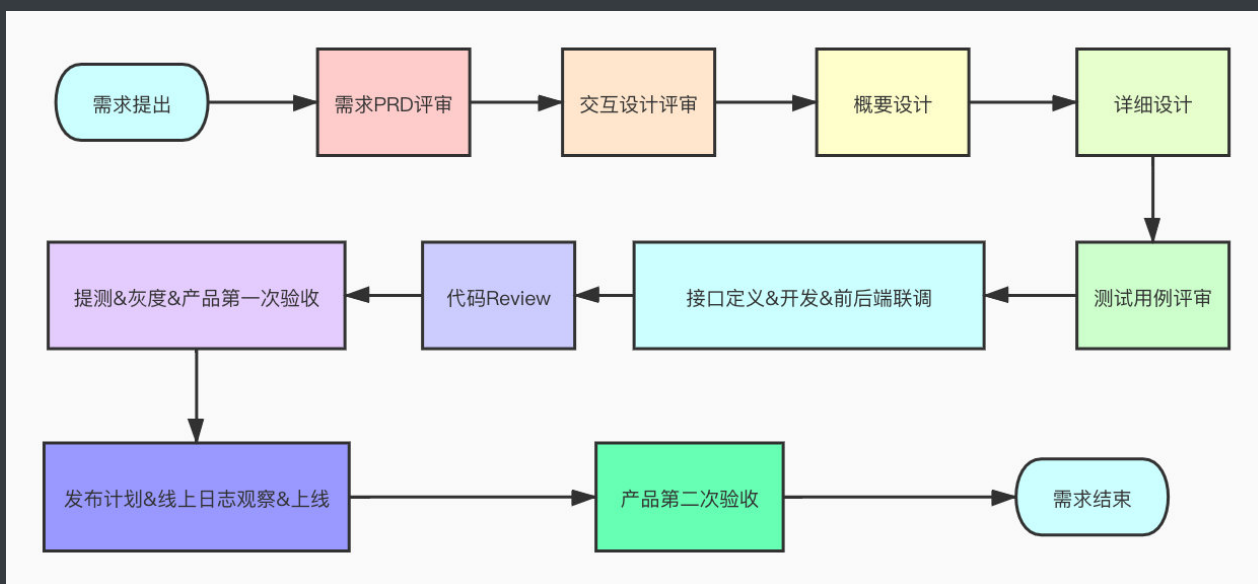
我的读者好像学生居多，然后大家最近问的较多的一个话题就是大厂的研发流程，都比较好奇，整个流程是怎么操作的。

我也不多BB了，那下面就跟随暖男的脚步，走进大厂研发流程吧。

## 正文

我们先看看一个产品有哪些研发流程，帅丙就用自己接触的阿里系的研发流程举例了，这也基本上是互联网大厂的研发流程了，可能细节有出入，但是绝对大同小异。

我问了下字节，多多，腾讯的朋友出入不大，所以还是具有代表性。



看完流程我们就一个个点的去看看每个环节干了些啥，我们开发同学在这个环节需要做啥，以及在每个环节的职能。

### 需求提出：

这个环节主要是产品爸爸给我们提需求，每个需求都是他们从用户，或者自己绞尽脑汁想出来的，但是产品爸爸还拿不准，不能直接敲定，所以需要我们需要大家（产品，UI，前端，后端，客户端和测试）一起讨论一下，看看这个需求是否合理，或者这个需求是否有意义，能否达到预期，技术实现的成本，周期等等。

一旦聊成了，他们就会进入下一个阶段，聊不成他会想方设法让你答应，然后进入下个阶段，知道我为啥叫产品爸爸了吧？



需求PRD提出：

这个阶段，产品爸爸会根据第一版聊下来的结果，大致出一个Demo版本的PRD，会画出初版的原型图，并且配上文字说明，所有涉及到的业务，还有交互细节都会罗列出来。

大致就是下图这样：



这个时候大家又会围绕这一版本去开会讨论，敲定细节，这个环节会久点，因为细节比较认真，逻辑也不能出错，还有UI稿子也得敲定，这里如果不敲定逻辑，UI提前去画原型图，后面假如逻辑推翻，一切重来就会浪费大量时间。

这一环节大家都会**把细节问清楚**，不了解的点也会去了解，测试，开发，UI我们都会在会议上提出自己的观点，自己的意见，然后等产品反馈，最后意见一致之后，产品当天就回改出敲定版本。

UI就会按照产品爸爸的意思去作图，接下来就是交互设计评审了。

### **交互设计评审：**

UI会画出客户端，前端，H5开发所需要的UI图，基本上就是我们看到的产品的样子了，不过还是要敲定细节，比如按钮合理不，或者上面数据是否在这展示，或者这里展示的数据是否合理。

这个环节会比较快，只要UI按照之前敲定的逻辑开发，出入不会很大，一般都是小改。

但是也不乏很多，之前敲定了情况，等UI按照敲定版本出了图，但是却发现出图之后有些不合理的点，比如是否应该在这里展示GMV（销售总额），或者是否这样展示活动规则啥的，会有这种情况，不过是小概率事件，改动也不会特别大。

UI界面：



大家看到的这种操作界面，按钮，图标的各种位置和图案，都是UI在这个阶段设计好的。（我什么都没暗示，不用关注我的B站）

大家敲定后就进入我们开发人员的回合了。

### 概要设计：

**概要设计**，这个是大厂程序员需求下来之后基本上都会做的一步，不过看需求大小，可能很多小需求直接就**详细设计**了，也有啥设计都不用做的小改动，具体需求具体分析嘛。

很多不了解的同学可能会问，需要设计什么呢？为什么要设计呢？

问得好，经常看我文章的都知道，**技术是把双刃剑**，你用了技术之后你是不是需要**列出他的优点缺点**，出问题之后的**解决方案**，还有**可能出现的问题**，**注意点**等等。

这么是让你能有把控力，比如你这个需求接入了新技术Es（Elasticsearch）你什么都不管你就是要接入它，你把他开发好了上线了，但是有啥坑你知道么？上线崩了怎么办？

不主动，不拒绝，不负责，这是渣男的行径，我们需要负起责任。

这个环节你需要考虑这个需求涉及到哪些服务了，需要新增哪些接口，修改哪些接口，表有现场的还是要新建表，字段要新建么？

其实远远不止这些问题，这就是我们做设计的主要原因，也是大家工作里面能成长的途径之一，你以为大佬们的经验是怎么来的？

### 推荐工具：Xmind/ProcessOn

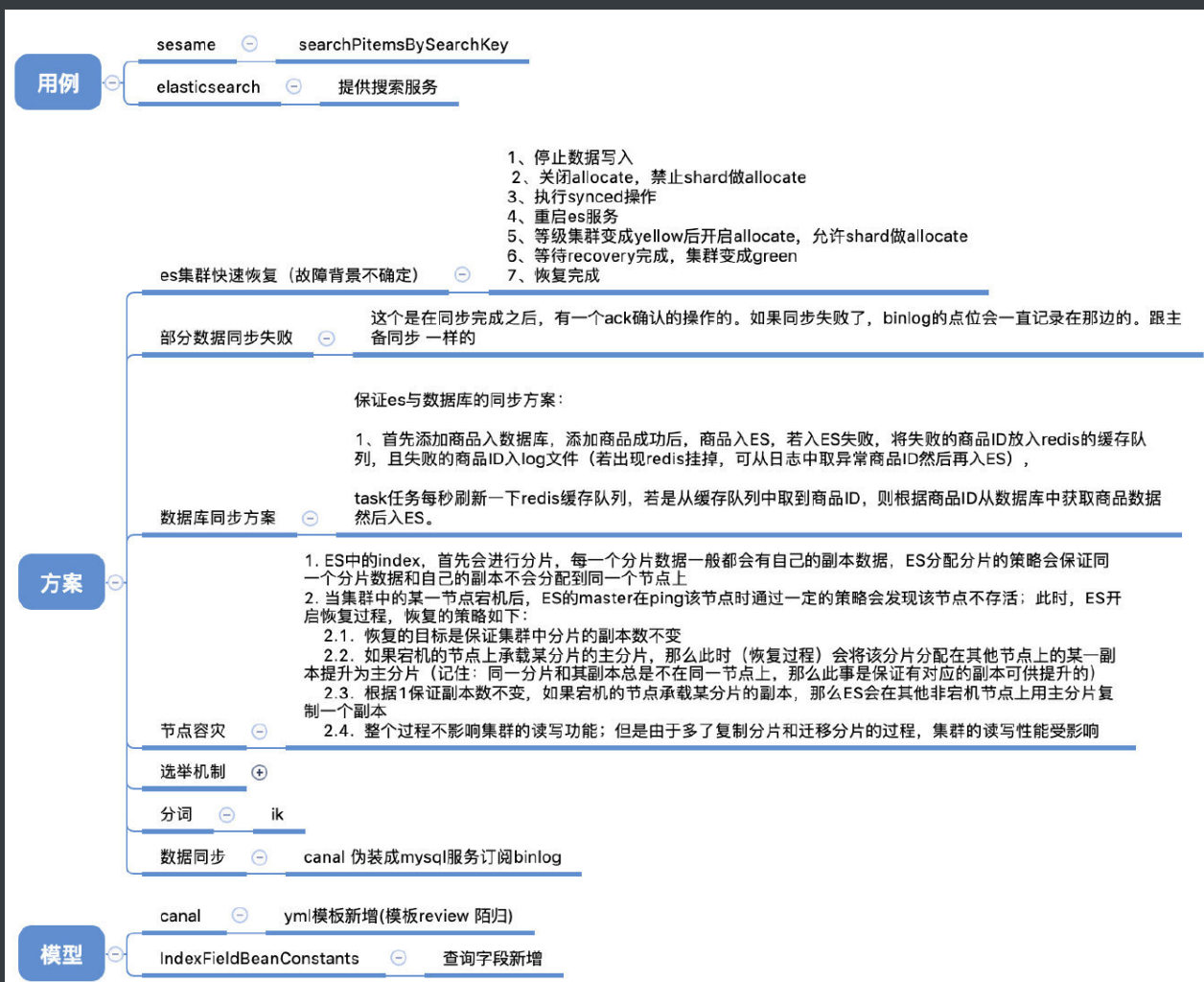
- Xmind官网地址：<https://www.xmind.cn>
- ProcessOn在线作图地址：<https://www.processon.com>

ProcessOn是我使用最频繁的工具了，我身边也有很多小伙伴在用，也推荐大家都使用：



大家在学习，看书等等的时候做个脑图，后面学习和复习的时候思路会很清晰，而且**效率瞬间高很多**，形成知识体系。

概要设计一般就是做个大概，给大家看一下我自己在设计ES相关的需求的时候的概设，比较粗糙看个大概就好了：



这个设计好了, 就需要给Leader看, 看理解程度, 一两次返工是有可能的, 如果你像或者像敖丙一样笨的话, 是有可能被打回N次的, 这里我得提一下, 好好做设计好处大大的有, 自己体会。

然后会进行一轮测试用例评审, 比如你涉及哪些服务, 新增了哪些接口, 改了哪些接口, 都是要同步出来的, 至于为啥?

是因为测试会依据这个数据, 评估影响范围, 方便他写测试用例, 后面会提到。

## 详细设计

小伙伴又要问了啥是详细设计呀帅丙?

傻瓜, 简单呀, 见名知意嘛, 概要设计是大概的设计, 详细设计是详细的设计。

我们研发的时候整个流程往往很复杂, 如果你理解不对直接就写代码, 最后容易造成返工, 延期, 加班, 被骂, 心情差, 回家吵架, 离家出走, 露宿街头, 饥寒交迫, 被迫吃野味, 然后全国。。。。

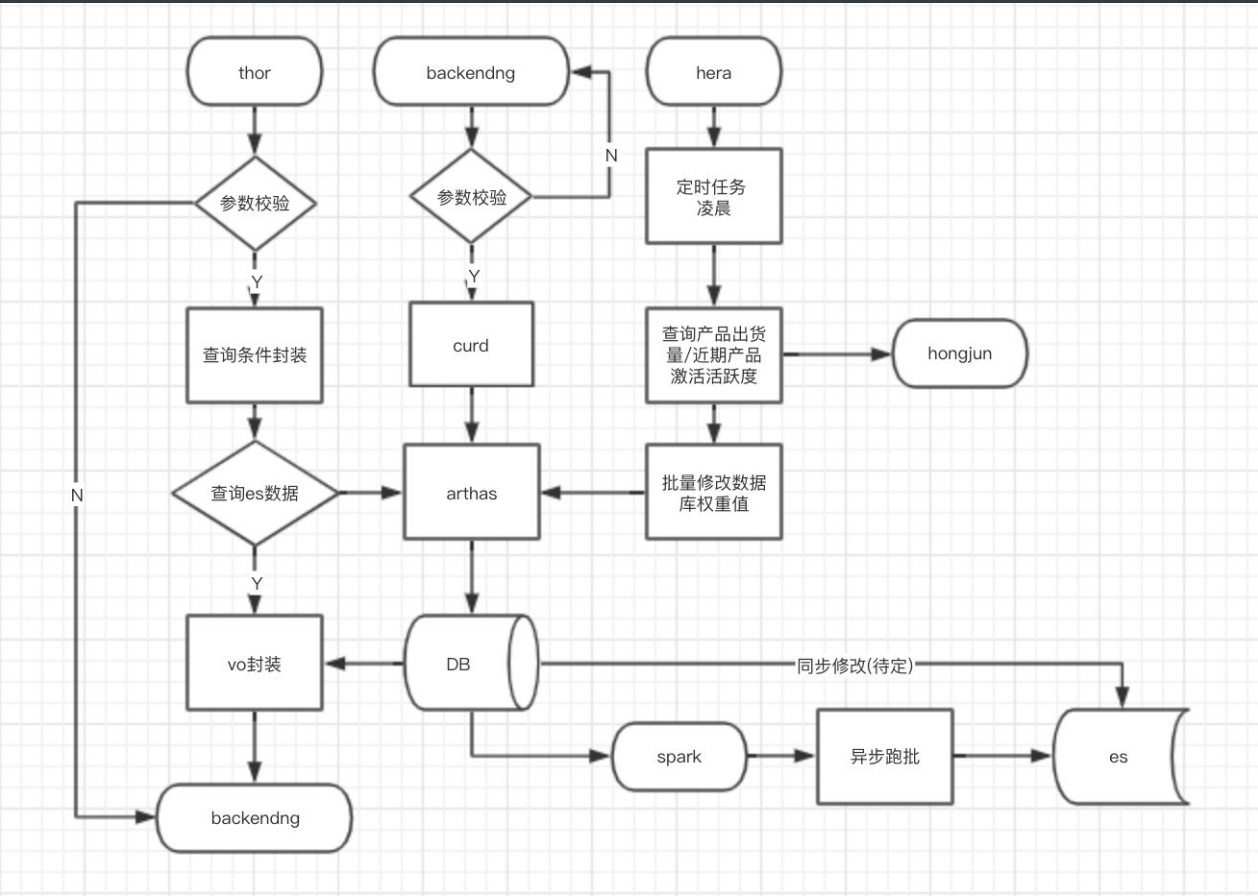
看到不做详细设计的后果了吧, 其实大家花点时间做详细设计很有必要, 你思路完全清晰了, 写代码那就是分分钟的事情, 不是嘛?

那再看看帅丙的一个小设计吧, 之前文章中大量的流程图, 时序图都来自它, 主要是这玩意还是在线的, 都不用下载很方便啊。



详细设计的工具我用的就是之前提到的在线作图神器：ProcessOn <https://www.processon.com>

还是我自己之前设计的一些流程图，大家可以看看：



这个环节一样重要，这个地方如果你能想好很多细节，开发的时候效率会高很多，像我上面的一些点，基本上就是看着图开发了。

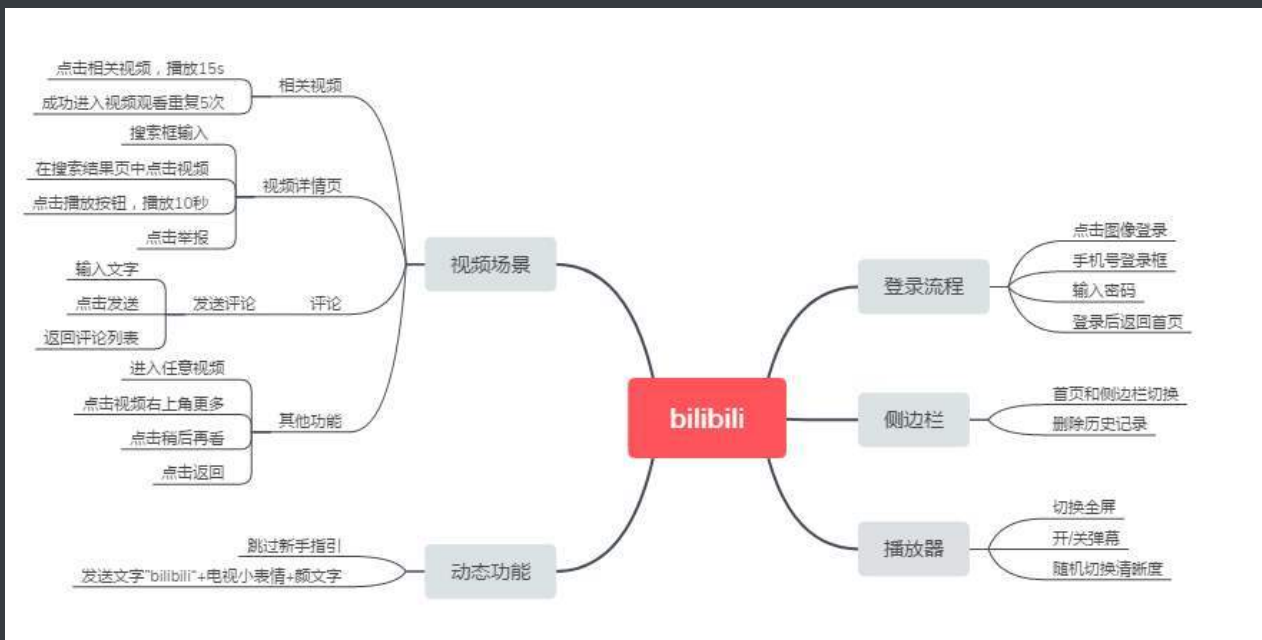
这个环节一般上不需要Leader参与，但是如果你有疑问或者不了解的点还是要提出来的。

测试用例评审：

上面我们说过，测试会根据你的概要设计，评估你的影响范围，你的影响点，新增和改动的接口啥的，去编写自己的测试用例。

测试用例，主要是为了把改动点影响点都考虑到，测全一点，免得上线了影响别的现有业务，也是为了把你开发的功能可能出现的bug给排除了。

我拿个小破站的小用例大家看看，这个比较粗糙但是也有点那味了。



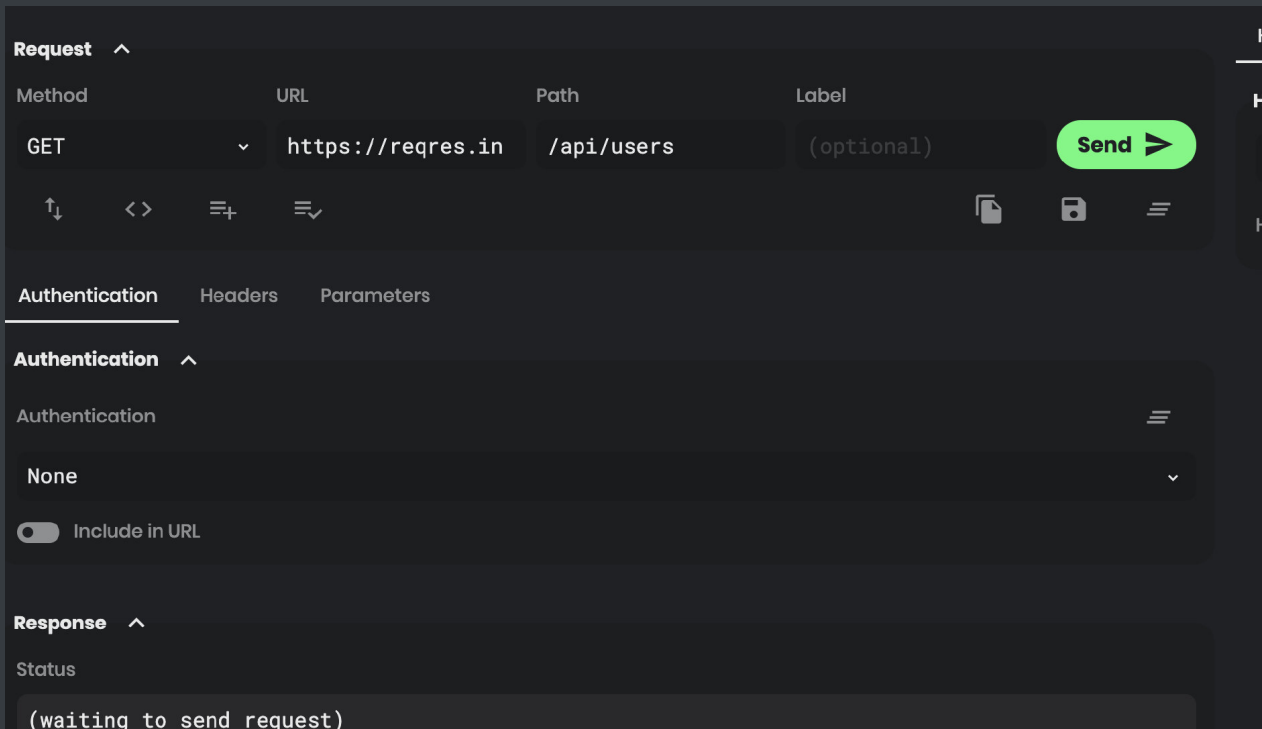
这个环节也会开会讨论，也是细节的确定，比如他写的是否合理，或者有什么点没考虑到，大家有没有补充的。

## 接口定义&开发&前后端联调

这个环节其实比较好理解，啥都敲定了，那就开发呗，开发差不多了，就得前后端联调了。

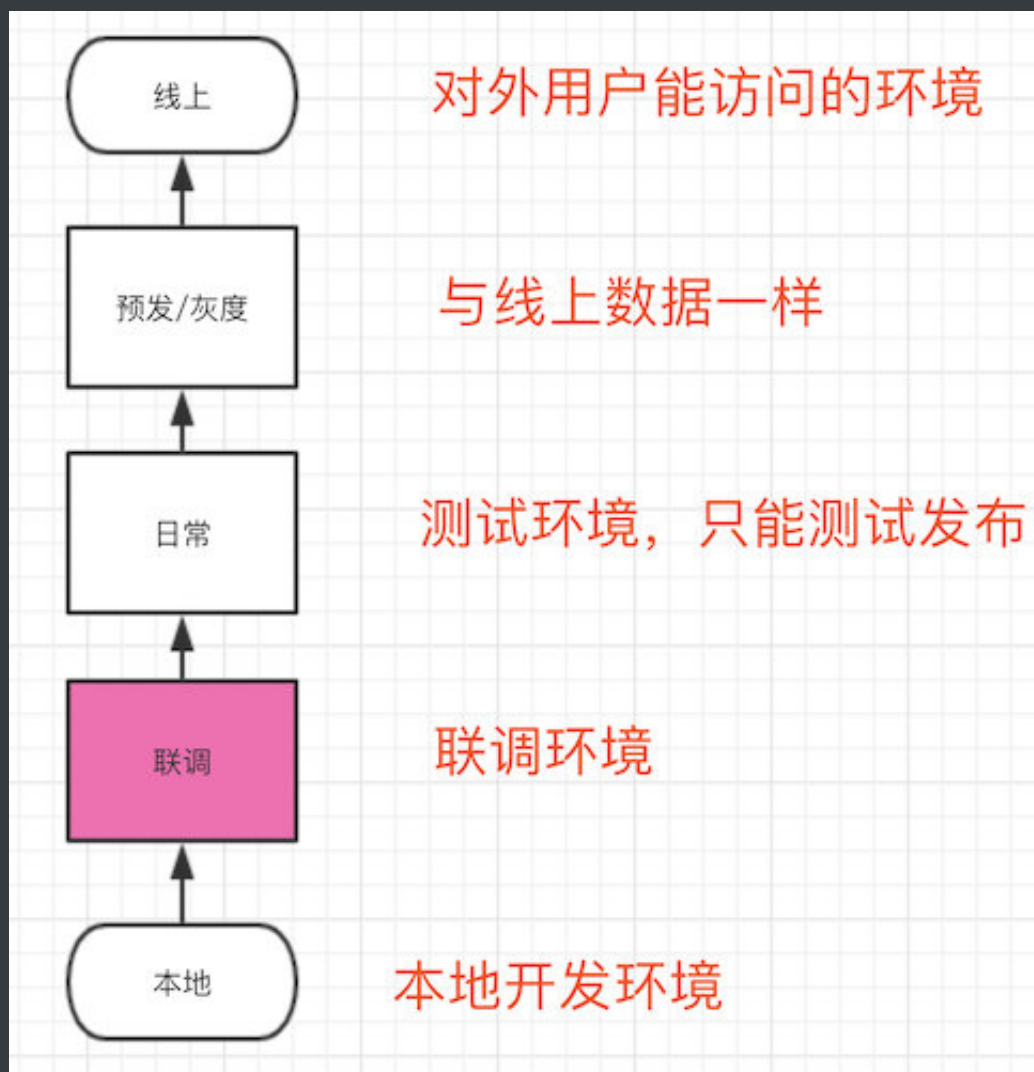
这里有个小细节还是想说一下，一般开发前我们都会提前定义数据类型，接口名称，然后在公司的接口工具上给出链接和参数，方便前端爸爸mock数据。

他总不能等我们后端开发完了，才去开发嘛，这样效率打折扣，所以都是后端先定义好，然后前后端并行开发的。





后端开发好，一般都是会发布到联调环境，我们有哪些环境，联调环境在我们所有的环境中处于哪个地位呢？



大家可以看到我列出了我们开发的所有环境。

Tip：日常环境不能由开发人员发布，是因为测试流程比较久，所以不能中断，如果你一直发布会影响测试的效率，在发布期间他们是没办法干活的，而且很多部门涉及相同的服务，你发布还会影响别人。

测试发布之前，在测试群里问问可以发某个服务么，大家觉得不影响，那么就可以发了，懂了吧。

预发环境，也叫灰度环境，这是跟线上数据一样的一个环境，只是只能内网访问，一般这一步是防止很多是因为日常的数据量不够真实，数据级别达不到线上的量级无法测出的bug。

扯远了，联调完了就是代码Review了。

## 代码Review：

codeReview环节，画一下重点，这可能是整个研发流程中，让你成长最快的一个环节，让组员和Leader Review你的代码，往往他们能给你很多业务上和技术上的建议和意见。

过来人的经验你就说香不香吧，以前老大经常没时间，但是我就是烦着他要Review，后来他说不用review了，但是我还是要组员大佬review，因为我很享受别人对我提建议的时候，这不就是成长，扫盲的好时机嘛。



### 提测&灰度发布&产品第一次验收

这一阶段就是把代码都发到日常环境，然后等测试爸爸测试，这个环节开发同学如果没BUG是比较轻松的，等着就好了，可以看看丙丙的文章啊，看看丙丙的B站视频什么的。

但是如果你BUG多，那我觉得你可能会生不如死，因为有的bug真的找很久很久，调用链路又长，特别是跨服务又涉及消息队列，或者第三方的接口什么的。



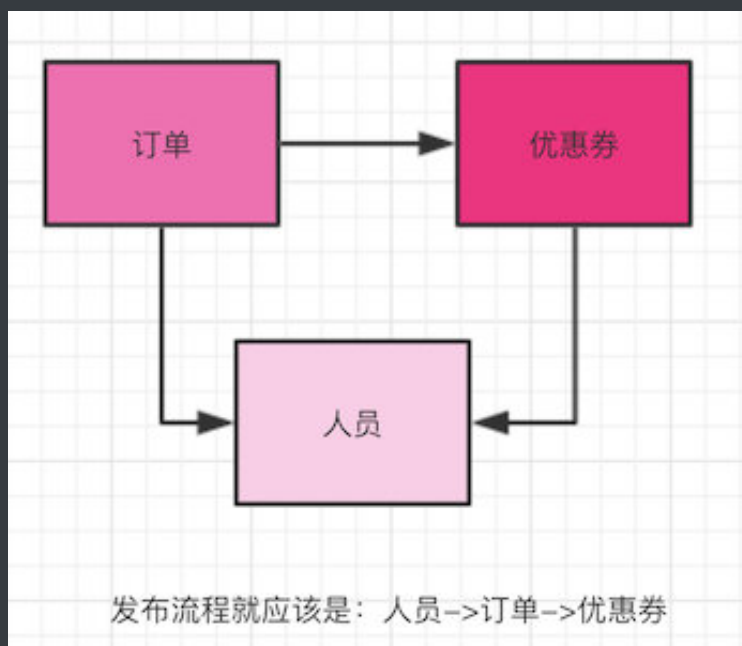
总之你也不知道会出现什么bug，我看身边的大神也只能用经验避免常见的坑，孰能生巧吧。

### 发布计划

敲黑板，这个确实是比较重要的环节，这个环节主要是开发同学和前端同学说好一个发布时间，然后制定一个发布计划，为啥要发布计划呢？

我们开发一个需求，可能涉及到N个服务，这些服务是有依赖关系的，那就需要打包，比如订单系统，依赖人员系统。优惠券系统，也依赖人员系统，然后订单系统还依赖优惠券系统，是不是有点乱了？

我们看图：



打包和发布顺序原则上是一样的，从没完全依赖的服务按照顺序发布到最后一个服务。

### 生成环境上线：

这就是神圣而庄严的上线环节，一般在这个环节丙丙都是要洗手洗澡，然后才点下那个神圣的发布按钮。



一般现在都是自动化发布，界面上点点就好了，记得丙丙大学发布都是进服务器一个个kill进程，替换jar包然后重启。

现在都是分布式的集群，这样发无疑会累死，我之前负责的系统有50多台机器，一般都是4台4台发布。

### 日志观察&产品第二次验收

一般发布第一批之后不会马上发布第二批，而是观察错误日志，看看是否正常，有时候会发现还是会出现异常情况的，那就保留错误日志，然后回滚。

知道解决了再发布，顺利的话就没啥错误，一口气发完了，看了下时间凌晨了，那发完差不多也得回家了。

一次发布可能涉及服务多的话，真的有可能发布这么久，但是没办法，线上出问题就是掉脑袋的事情。

日志观察一般公司都有错误日志搜集系统，或者自己登录跳板机查看就好了。

没问题，发完之后告诉产品大大就好了。

需求结束

至此基本上一个需求可能就结束了，其实还是很不容易的，短的需求几天，长的需求几个月，中间涂涂改改，BUG，技术难点都是你要面对的，不过没啥大问题，我们技术人嘛皮实能顶。

总结

产品研发流程大家是不是觉得有点复杂，或者觉得很多点有点小题大做了，不瞒你说，刚开始我也这么认为的，但是随着时间的推移，你会发现有时候越是这样规范，越是提升了效率，也提升了产品质量。

对自己设计的严苛也会让你的业务能力提升，开发考虑的点也越来越广泛，我想大佬应该都是这样走过去的，那没啥好说的，我们也走。

最后给大家看看我自己搞的一个项目管理模板吧，基本上能适用大部分项目了，要xmind格式的公众号回复【项目】即可。

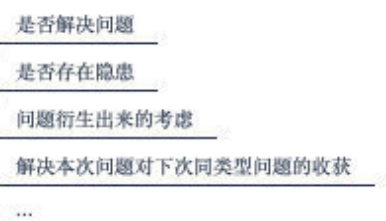


# 项目名称

## ★ 项目方案



## ? 解决问题



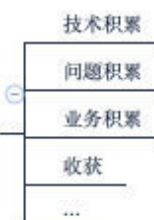
## 📅 项目排期



## 📊 上线准备



## 📁 项目沉淀



我是敖丙，一个在互联网苟且偷生的程序员。

创作不易，不想被白嫖，各位的「三连」就是丙丙创作的最大动力，我们下篇文章见！



文章持续更新，可以微信搜索「三太子敖丙」第一时间阅读，回复【资料】【面试】【简历】有我准备的一线大厂面试资料和简历模板，本文 [GitHub https://github.com/JavaFamily](https://github.com/JavaFamily) 已经收录，有大厂面试完整考点，欢迎Star。

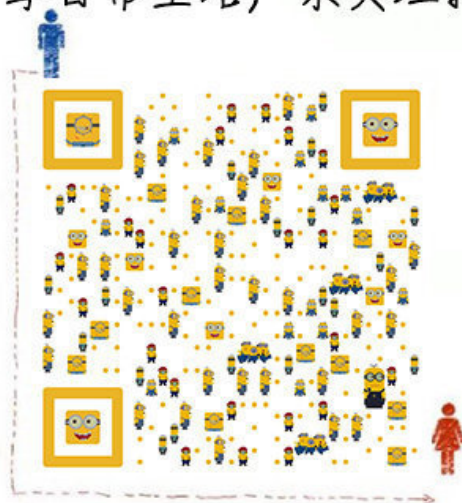
## 资料/面试/联系我/公众号



关注后回复【资料】领整理的互联网大厂技术栈资料  
关注后回复【面试】领整理好的面试线路相关文章  
关注后回复【加群】有我个人联系方式和大佬云集的技术交流群，有问题一起交流。

如果觉得丙丙的文章确实有点东西，求个赞就好啦  
对在博客苟且偷生敖丙来说真的 非常有用！！

文章第一时间我都会发在公众号，微信搜索【三太子敖丙】  
我还会在公众号分享日常生活，来关注我吧。



微信搜一搜



三太子敖丙

别爱我，没结果

你知道的越多，你不知道的越多