

# 2023小程序河狸家模式系统开发

产品名称	2023小程序河狸家模式系统开发
公司名称	广州茂林网络科技有限公司
价格	.00/件
规格参数	
公司地址	广州市黄埔区城门大街2号之三5楼5103室
联系电话	13128642589 13128642589

## 产品详情

河狸家运维发展历程，是从无到有的过程。河狸家起初聘请了一家外包公司帮助开发早期的产品，但使用中出现了一些问题。后来逐渐发现原来软件部署上去之后，不只要进行增删改查功能，还要有专门的运维工程师去线上运维做相关的工作。

运维团队的职责也经历了从不清晰到明确的过程。之前有的工程师同时兼做很多工作，日常工作40%、50%的时间都是做帮领导导报表，久而久之他变得非常郁闷，本职工作也很难做得好。

河狸家的运维，其实也是从黑盒到白盒的过程。作为一个程序员，个人会有很强的控制欲：如果说不知道要上线的东西源码是怎么写的，心里会很是慌兮兮的。运维希望各个数据都可以看得到，又全部都可以串起来。之前可能只是用简单地监控读取一些数据，但是其中整个请求串起来之后，代码中出了什么问题，比如说活动的场景出现问题，可能就束手无策了。

此外，这还是从混乱到规范的过程。什么是混乱？大家很多都是从创业公司出来的，遇到过很多创业公司发展中的问题。在创业公司，包括团队、人员、项目，都需要不断地进行自我学习、\*\*和修炼，是一个不停成长的过程，因此不可能一下子都很完美。所以说在这个过程中，公司整个项目，包括项目开发流程等等都是缺失的。

举个例子，团队做一个简单的版本发布，怎么做呢？发布之前，产品和运维的同学给了你一些不同的功能提议，结果确定下来之后发布日期马上就到了，这时候只能加班加点去开发。这种情况下去发布，运维半夜三更被叫起来去做，结果发布上去出现一堆问题，顿时傻眼了。这样下来，运维的黑锅越背越多，就被别人说成“不靠谱”。

这个时候需要反思。当团队把来龙去脉想通之后，会认识到项目管理的过程都是一环扣一环，而往往后一道环节是产品、技术、运维可以看见的这一部分。因此一旦出了问题，就得背黑锅，前面的环节干了“坏事”的人都逃之夭夭了。把这个东西梳理清楚之后，规范就可以推行起来。项目管理到底要怎么管？提下我们的想法和思路。

## 项目管理的想法和思路

首先，是从业余到标准运维的思路。什么叫从业余到标准？之前线上的服务端用java的代码比较多。以部署为例，在之前的部署当中，发包的过程是把手工的包丢过来，而后是手工脚本。有些同学改了代码，发现线上跑的代码跟你的对不起来。所以就要制定标准，这个标准是什么？包括软件怎么打包、怎么部署、怎么成为一个标准的软件上线，都需要去梳理；在框架方面，比如java工程能一键生成一个可执行、可交互的软件。

此外，关于上线流程也研发了统一的系统，用这个流程去对接，包括SVN开发标准梳理等。前期推行时，开发同学可能不太适应，因为不管怎么说，从作坊式的走向规范化都不是件容易事。另外还建立了紧急发布环节：可以让你通过审核，但是要找领导去批。当然大家一开始不适应找领导签字。过了几个月之后，统计紧急发布了多少，正常发了多少，为什么每天都是紧急发布？真的业务到了每天都需要发的情况吗？这些问题都有据可查了。所以这个东西不仅仅是一个标准，也可以作为记录，充当是否合理的有力评判依据。

要从被动往主动去发展。我们一直希望运维能反过来推进一些事情：包括提前预测一些业务上的情况，以及未来业务的发展规划等。当然这个实现起来可能很困难。

BAT有些系统很牛逼，做得很完美。但公司通过某个渠道把它拿过来，纳为己用是不是就高枕无忧了呢？不是这样的，技术没有牛逼不牛逼、高大上不高大上，只有合适不合适，能不能在自己的场景里面适用。

运维在这个系统上线之后，不停地发现线上的问题，包括技术的、业务的，因而不停地去优化，让公司软件业务不断地往好的方向发展。因为问题永远不可能消除，学过矛盾各位的都知道，矛盾只可以转移，不可以消灭，一个矛盾接着一个矛盾，我们只能兵来将挡、水来土掩。后台有报警系统、用户投诉、客户投诉、后台分析等，现在会统一提交到运维做综合性的分析，再去判断问题的可能性。比如是自身的问题就去做修复，如果不是我们这里的问题，就要跟研发等相应的团队同学去沟通、配合。

运维在整个公司项目管理过程当中，其实是后一个环节。所以运维把工作做好的前提是：把整个流程全部都串起来。只有把这个工作做好了，运维的工作才能做好。而如果这个工作不做好，运维永远没有话

语权去反推其他部门。因此，流程可能不一定按照这种方式去走，但是一定要找到一条适合自己公司、自己场景的道路，去把这个流程规范好。

为什么要做监控系统？

在做这个系统之前，团队曾面临一些问题。，运维系统之前只是针对服务器，用zabbix等做了一些监控，接入了短信的报警。但后来发现监控一旦报警之后，首先没人处理；其次报警没有分类，具体报了什么无从知晓，没什么意义；再次研发人员不关注运维，只是需求功能的叠加。之前研发团队只关注开发功能，不关注上线之后代码到底产生什么作用，运维好不容易把工作做好了，结果隔了一段时间，这些问题又来了。因为开发不关注这些东西，也不去审查这些问题，运维根本就无法做到。

第二，研发团队自身的能力也很难得到成长，因为他们永远只是在开发功能，但是不知道功能上去之后对业务有什么影响，也根本不关注。久而久之，团队也很难更好地成长，并且人员归属感、能力\*\*等等都会受到一系列的制约。后，公司的系统成了一个黑盒，如果真的出了问题，除了底层zabbix可以报警之外，其他一无所知。只能随便乱搞，不知道到底是哪里的问题，这时候真的是抓狂的。

提出这些问题之后，这个平台开始建立，设想有这么几个初衷：

系统，基础软件，服务状态等都是可以可视化的。第二，整个流程是可以跟踪到的。第三，能对线上的\*，包括PV、UV进行统计和分析。第四，可以定期跟踪到线上的代码性能情况，比如说代码执行站终性能卡在哪里，哪个函数调用比较慢等。第五，希望对数据库的慢查做一些分析。第六，业务稳定性监测，比如说商品订单支付可以做一些稳定性的监测。第七，对报警做分级，做到可分支，多个通道都可以接入，可以短信、邮件包括微信报警。第八，从运维的角度便于业务分析数据。

考虑自己的问题。不管是业务还是技术，可能运维只是考虑到系统服务器怎么运维，但产品也需要运维，如果没有一些可视化的东西，你都不知道自己的数据是怎么回事。产品上线之后，久了也不知道问题出在哪里，这从广义的角度也是一个运维的过程。