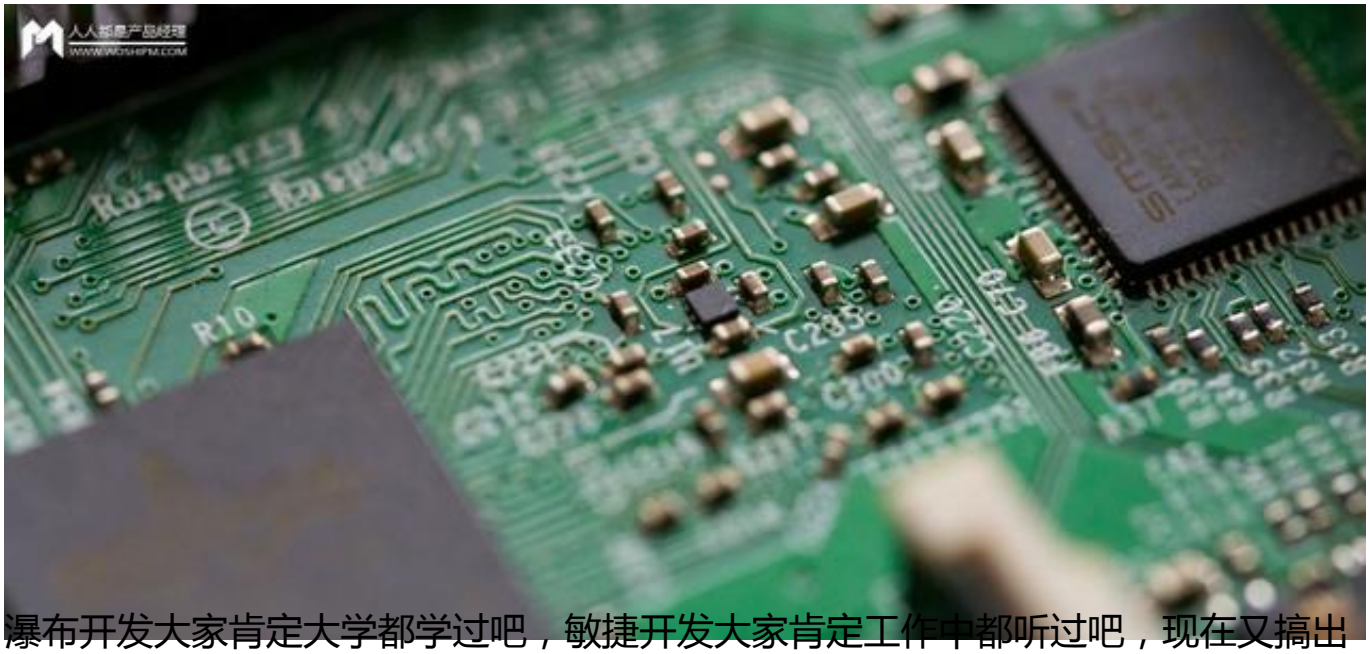
编辑导语：随着互联网的发展，用户对于产品的需求越来越高，更多的用户追
求“好看
”和“好用”。本
文作者分享了一种新的开发模式——DevOps
开发模式的发展和开发模式，感兴趣的一起来看看吧。



瀑布开发大家肯定大学都学过吧，敏捷开发大家肯定工作中都听过吧，现在又搞出来个DevOps开发，这个估计就知道的不多了吧~

来，一起来接受知识的洗礼吧，知识就是力量，就是金钱，奥利给！

先告诉大家两句话，第一句话是趋势，第二句话是概念。

趋势：目前很多大厂如阿里、腾讯、百度、头条、美团
等公司内部都在用DevOps开发模式。

概念：
DevOps=Developers（开发）+Operators（运维），即开发团队和运维团队一体化。

哈，只看概念啥感觉？是不是感觉概念是不可能看懂的，这辈子都不可能看懂的。

没关系，在这里只记住两件事
即可：第一件事就是，
DevOps是把开发和运维绑定到一起了；第二件事就是，
DevOps不是工具什么的，而是一种方法论。

其他的交给我，往下看，保证给你讲解的明明白白的。

一、发展历史

想要了解一个新的事物为什么会出现，就比如DevOps，那么最好的方法，就是去了解他的过去，他的历史。

前面说了，DevOps是一种开发模式，提到“开发模式”这四个字，你能想到什么？是不是瀑布开发和敏捷开发？

好，我们就先来扒拉一下，这两种开发模式的演进历史。

二、瀑布开发

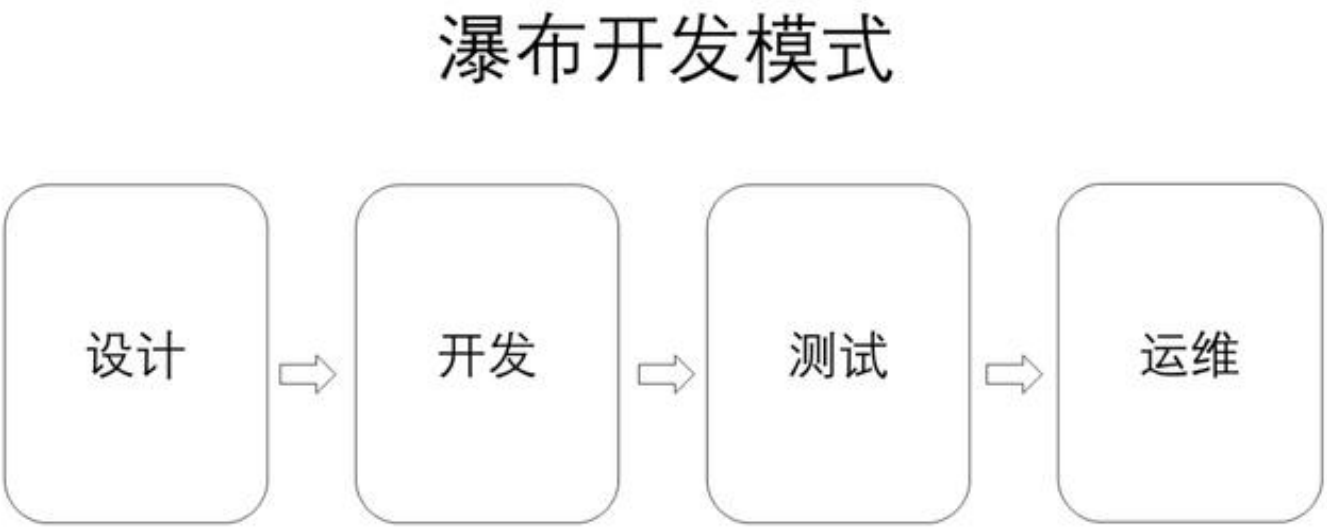
1. 背景

在互联网初期，程序员
还是科学家一样的存在，当时程序员的办公室还被称作实验室。

当时的网民，也还没有那么多，大多数时候，他们都是带着敬畏之心提需求的。

并且开发出来的产品，只要能解决他们的问题，他们都是争先恐后地给程序员烧高香、送锦旗，对研发周期，自然不敢有太多的要求，需求也自然是不变的。

在这种大背景下，就诞生了瀑布开发的模式，如下图所示：



2. 特点

瀑布开发模式，最大的特点有三个：

1. 需求是固定的；
2. 开发周期是固定的，可能为3个月，也有可能是1年；
3. 设计、开发、测试、运维各个环节是独立的，当前一个环节完全搞定，下一个环节才开始介入。

3. 弊端

这种模式的弊端，相信大家都知道，总结一下就是：需求不能快速得到验证！

有可能花了1年开发出来的东西，早已时过境迁，不再适合市场；也有可能你搞了半年之后，发现搞出来的东西跟用户的需求完全驴唇不对马嘴，只能完全推翻，一切重头再来。

这个时候啊，敏捷开发出现了。

三、敏捷开发

1. 背景

随着时代的发展，互联网的网民越来越多了，而且这些网民的口味也越来越刁了。

搞出来的产品，只是“能用”，已经满足不了他们的胃口了，更多的网民越来越追求“好用”以及“好看”。

而且这些人也变得越来越“朝三暮四”了，可能今天觉得这个好，明天就觉得那个好了，软件开发的周期，也被压缩的越来越短。

时代的大背景下，总有那么几个脑子进化的比较快的，于是乎，敏捷开发模式，登上了历史的舞台：

2. 特点

最大的特点就是一个字：更快！

具体来说，敏捷开发可以更快地发现问题，产品被更快地交付到用户手中，团队可以更快地得到用户的反馈，从而进行更快地响应。

另外一个特点，就是将开发与测试从对立关系，变成了同一战线。

瀑布模式的时候，测试的工作，就是为了尽可能地挑开发搞出来东西的毛病，妥妥的你死我活。而敏捷开发，则将二者目标进行了统一，都为了能够更快更好地开发出产品而共同努力。

从敏捷宣言中，我们可以窥见其中的真谛：

十二条敏捷宣言：

1. 我们的首要任务是通过尽早地、持续地交付可评价的软件来使客户满意。
2. 乐于接受需求变更，即使是在开发后期也应如此。敏捷过程能够驾驭变化，从而为客户赢得竞争优势。
3. 频繁交付可使用的软件，交付间隔越短越好，可以从几个星期到几个月。
4. 在整个项目开发期间，业务人员和开发人员必须朝夕工作在一起。
5. 围绕那些有推动力的人们来构建项目。给予他们所需的环境和支持，并且信任他们能够把工作完成好。
6. 与开发团队以及在开发团队内部最快速、有效的传递信息的方法就是，面对面的交谈。
7. 可使用的软件是进度的主要衡量指标。
8. 敏捷过程提倡可持续发展。出资人、开发人员以及使用者应该总是共同维持稳定的开发速度。
9. 为了增强敏捷能力，应持续关注技术上的杰出成果和良好的设计。
10. 简洁——最大化不必要工作量的艺术——是至关重要的。
11. 最好的架构、需求和设计都源自自我组织的团队。
12. 团队应该定期反思如何能变得更有战斗力，然后相应地转变并调整其行为。

敏捷宣言是为银河系带来和平以及维护各自的平衡所迈出的很重要一步。

在很长的时间里，相比此前基于流程和机械化的方式，这是第一次基于文化和“人性”来将不同的关键项目干系人连接在一起的方式。

人们开始互相交流，进行基本的碰头会议，并开始不断的交流意见和看法。

他们开始意识到他们是有着很多比想象中还多的共同点，客户也开始成为他们之中的一员，而不再是像以往一样只是往项目砸钱然后开始求神拜佛祈求一切顺利如愿。

3）弊端

虽然敏捷开发大
幅提升了软件开发的效率和版本
更新的速度，但是它的效果仅限于开发环节
。我们发现，运维那边，非常落后的手动部署上线，就成了新的瓶颈。

运维工程师
，和开发工程师有着完全不同的思维逻
辑。运维团队的座右铭，很简单，就是“稳定压倒一切
”。运维的核心诉求，就是不出问题。

什么情况下最容易出问题？发生改变的时候最容易出问题。所以说，运维非常排斥“改变”。

于是乎，矛盾就在两者之间集中爆发了。

这个时候，神秘的主角DevOps，隆重登场了。

三、DevOps开发

1. 背景

到了当今时代，互联网的网民在海量增加，在这种背景下，诞生了众多的互联网大厂，以及多款现象级产品，比如微信、淘宝、抖音等等，互联网的竞争也越来越激烈。

快速迭代产品，快速占领市场，快速占据用户心智成为了各互联网公司的目标，这个时候，就
对产品开发提出了更高
的要求，需要能够对产品持续开发、持续集成
、持续测试、持续部署、持续监控，需要每天每时每刻都可进行新版本的上线。

开发团队与运维的矛盾，是时候环节一下了，怎么缓解呢？那就把它们也搞到同一战线吧：

2. 特点

这种模式下，将“更快”，又提升了一个层次：用户可以很早地就得到最终产品或服务的一部分进行实际体验，从而可以尽快的把发聩传递回需求管理团队和产品研发团队。

而这的一切，都需要归功于：DevOps将开发、测试、运维都拉到了同一战线！

敏捷有自己的宣言，而DevOps也有着自己的清单：

DevOps清单：

1. 开发团队和运维团队之间没有障碍，两者皆是DevOps统一流程的一部分。
2. 从一个团队流转到另一个团队的工作都能够得到高质量的验证。
3. 工作没有堆积，所有的瓶颈都已经被处理好。
4. 开发团队没有占用运维团队的时间，因为部署和维护都是处于同一个时间盒里面的。
5. 开发团队不会在周五下午5点后把代码交付进行部署，剩下运维团队周末加班加点来给他们擦屁股。
6. 开发环境标准化，运维人员可以很容易將之扩展并进行部署。
7. 开发团队可以找到合适的方式交付新版本，且运维团队可以轻易的进行部署。

8. 每个团队之间的通信线路都很明确。
9. 所有的团队成员都有时间去为改善系统进行试验和实践。
10. 常规性的引入（或者模拟）缺陷到系统中来并得到处理，每次学习到的经验都应该文档化下来并分享给相关人员，事故处理成为日常工作的一部分，且处理方式是已知的。

总体来看，在人力成本如此之高、市场竞争如此之激烈、用户需求变化如此之频繁的情况下，DevOps是大厂必须选用的一条路。

四、结语

好了，以上就是今天为大家总结的内容了，你问我了解这些有啥用？

作为互联网界走在最前沿的产品经理，你说你该不该了解这些个玩意；然后作为颜值与智商兼备，然后日夜期盼世界和平的码农
们，有这样一个机会，摆在你们面前，你们难道不应该珍惜么？

#专栏作家#

晓庄同学；公众号：晓庄同学产品笔记，人人都是产品经理专栏作家。互联网老兵，各大平台专栏作者。

本文原创发布于人人都是产品经理。未经许可，禁止转载。

题图来自Unsplash，基于CC0协议。