

DevOps

转载

猎户星座。 于 2021-03-26 17:38:07 发布 2361 收藏 71

分类专栏: [J2SE基础](#)

原文链接: <https://juejin.cn/post/6922974630131171341>

版权



[J2SE基础](#) 专栏收录该内容

5 篇文章 0 订阅

订阅专栏

提到DevOps这个词，我相信很多人一定不会陌生。

作为一个热门的概念，DevOps近年来频频出现在各大技术社区和媒体的文章中，备受行业大咖的追捧，也吸引了很多吃瓜群众的围观。

那么，DevOps是什么呢？

有人说它是一种方法，也有人说它是一种工具，还有人说它是一种思想。更有甚者，说它是一种哲学。



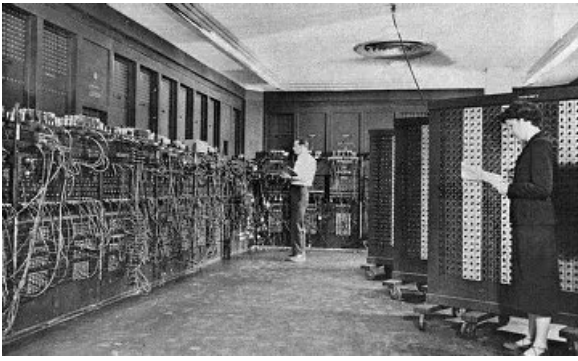
南无DevOps

越说越玄乎，感觉都要封神啦！DevOps这玩意真的有那么夸张吗？它到底是干嘛用的？为什么行业里都会对它趋之如鹜呢？

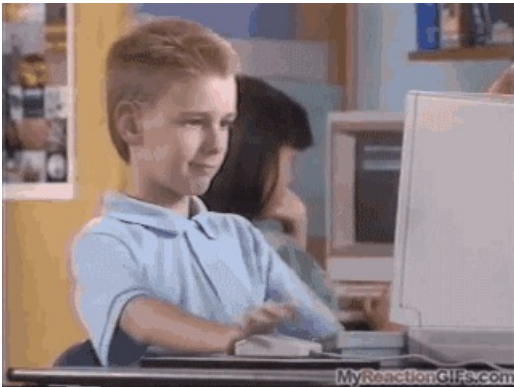
今天这篇文章，就和大家好好聊一聊这个DevOps。

DevOps的起源

上个世纪40年代，世界上第一台计算机诞生。从诞生之日起，它就离不开程序（Program）的驱动。而负责编写程序的人，就被称为“程序员”（Programmer）。



随着人类科技的不断发展，PC和Internet陆续问世，我们进入了全民拥抱信息化的时代。越来越多的企业开始将计算机作为办公用的工具，用以提升生产力。而普通个人用户也开始将计算机作为娱乐工具，用以改善生活品质。



于是，计算机的程序，开始变成了一门生意。程序，逐步演进为“软件（software）”，变成了最赚钱的产品之一。



我们知道，一个软件从零开始到最终交付，大概包括以下几个阶段：规划、编码、构建、测试、发布、部署和维护。

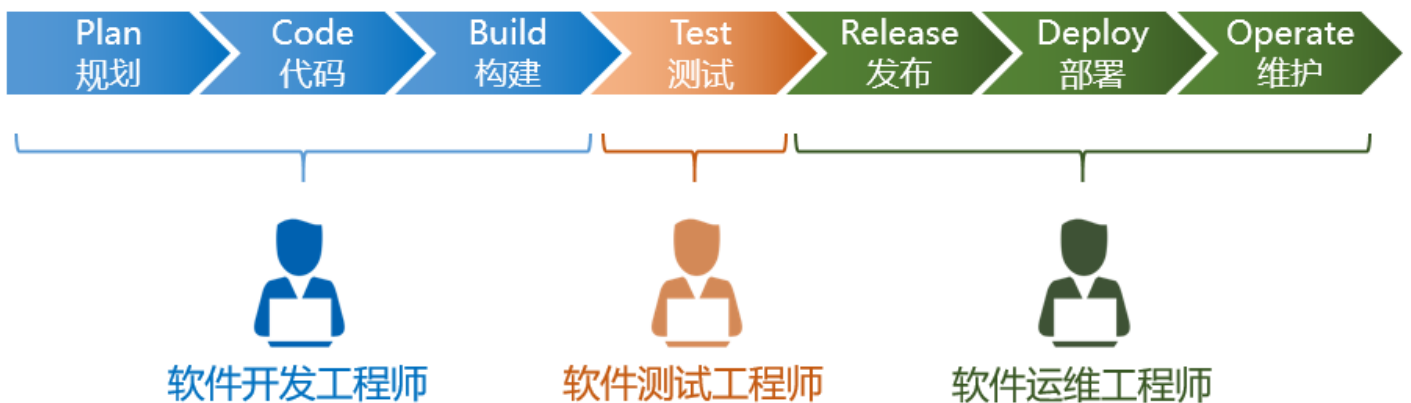


最初，程序比较简单，工作量不大，程序员一个人可以完成所有阶段的工作。



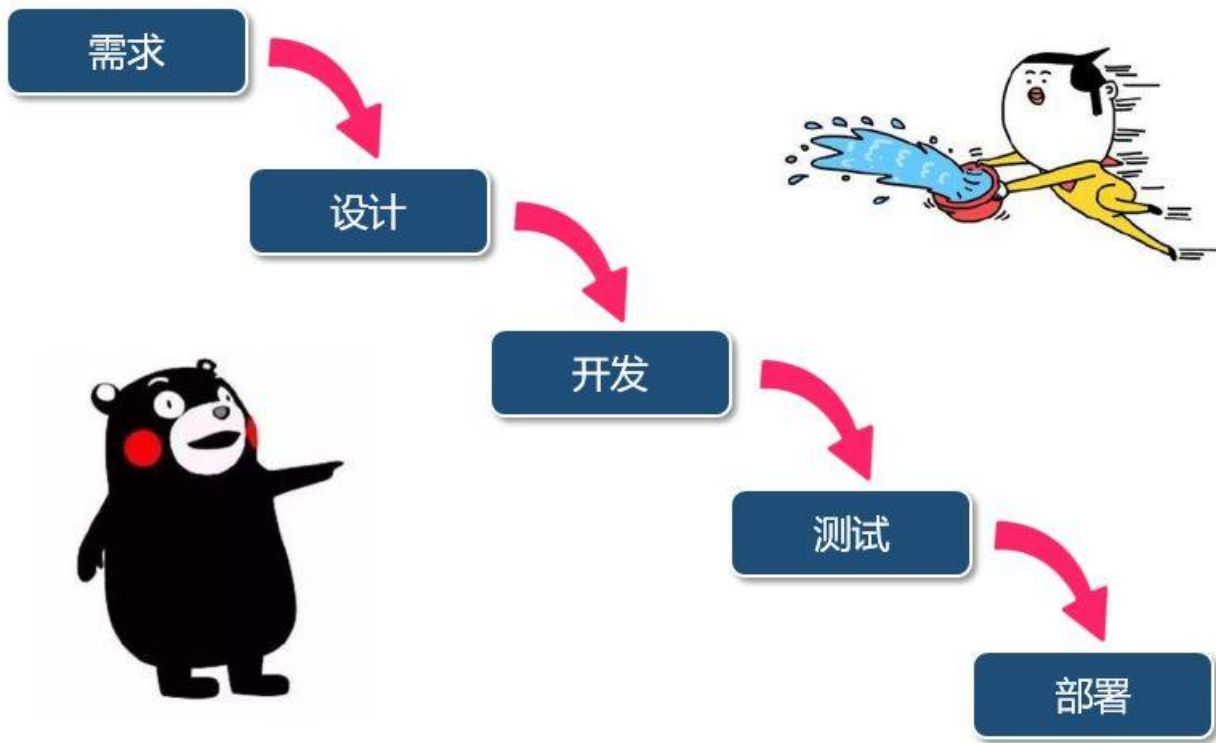
随着软件产业的日益发展壮大，软件的规模也在逐渐变得庞大。软件的复杂度不断攀升。一个人已经hold不住了，就开始出现了精细化分工。

码农的队伍扩大，工种增加。除了软件开发工程师之外，又有了软件测试工程师，软件运维工程师。



软件开发人员花费数周和数月编写代码，然后将代码交给QA（质量保障）团队进行测试，然后将最终的发布版交给运维团队去布署。所有的这三个阶段，即开发，测试，布署。

早期所采用的软件交付模型，称之为**“瀑布（Waterfall）模型”**。



瀑布模型，简而言之，就是等一个阶段所有工作完成之后，再进入下一个阶段。



随着时间推移，用户对系统的需求不断增加，与此同时，用户给的时间周期却越来越少。在这个情况下，大家发现，笨重迟缓的瀑布式开发已经不合时宜了。

敏捷开发在2000年左右开始被世人所关注，是一种能应对快速变化需求的软件开发能力。其实简单来说，就是把大项目变成小项目，把大时间点变成小时间点，然后这样：



有两个词经常会伴随着DevOps出现，那就是CI和CD。CI是Continuous Integration（持续集成），而CD对应多个英文，Continuous Delivery（持续交付）或Continuous Deployment（持续部署）。

美其名曰：“持续（Continuous）”，其实就是“加速——反复——加速——反复……”，这样子。

画个图大家可能更明白一点：

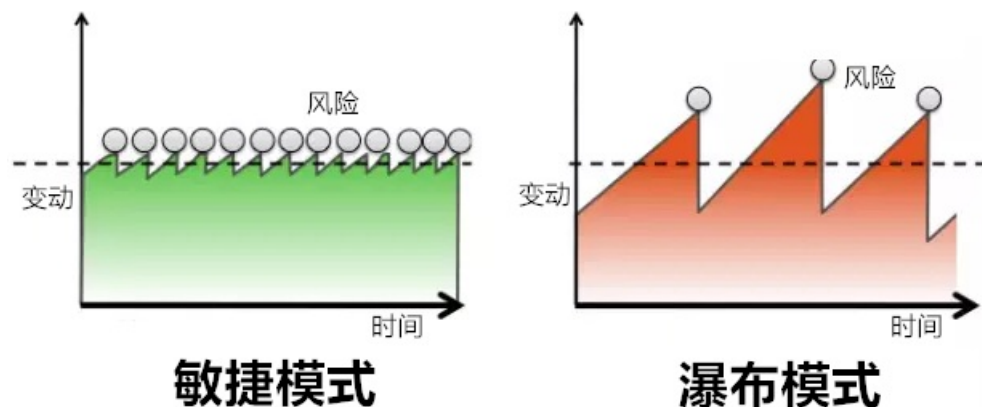


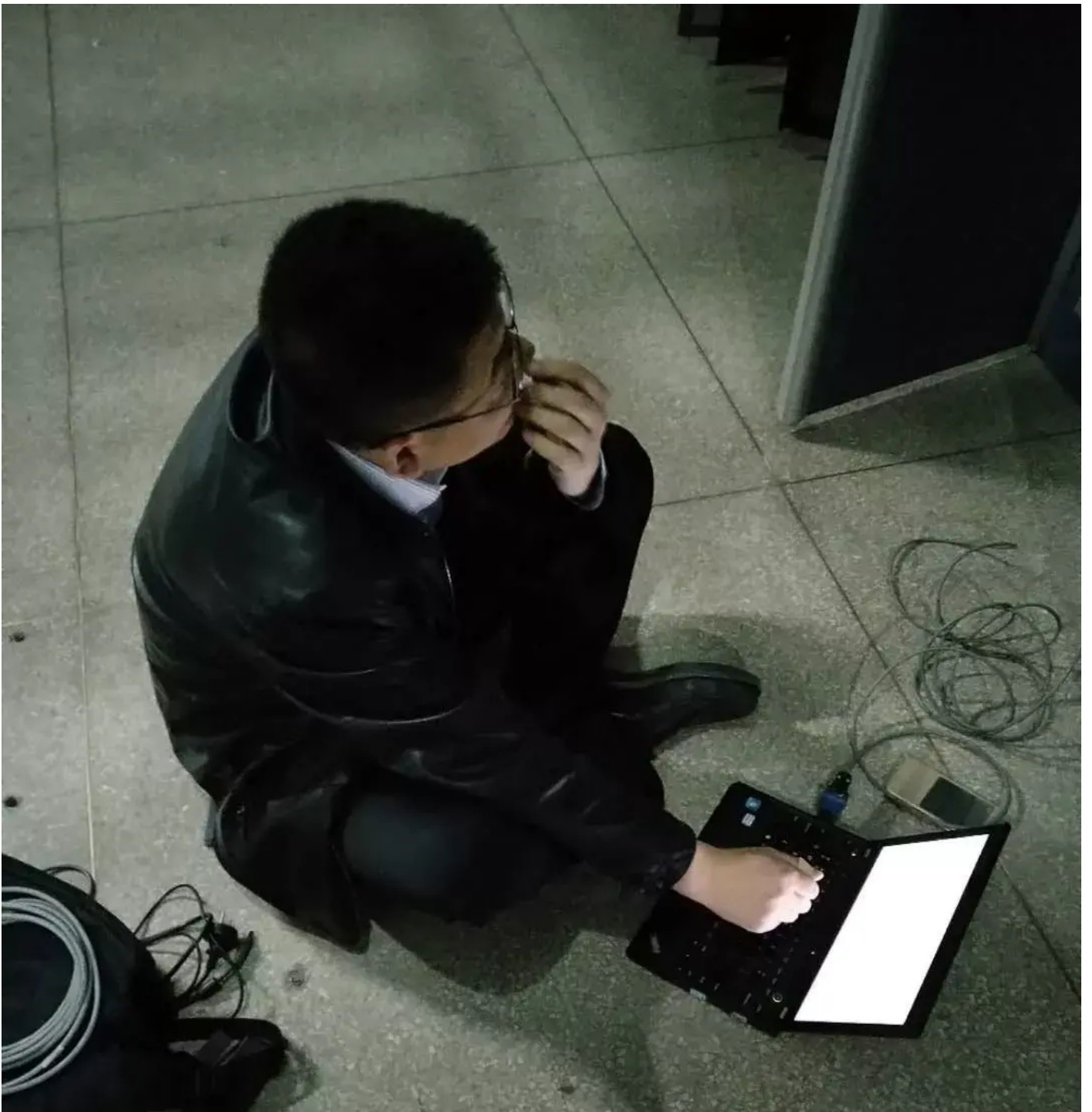
敏捷开发大幅提高了开发团队的工作效率，让版本的更新速度变得更快。

很多人可能会觉得，“更新版本的速度快了，风险不是更大了吗？”

其实，事实并非如此。

敏捷开发可以帮助更快地发现问题，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。而且，DevOps小步快跑的形式带来的版本变化是比较小的，风险会更小（如下图所示）。即使出现问题，修复起来也会相对容易一些。





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。

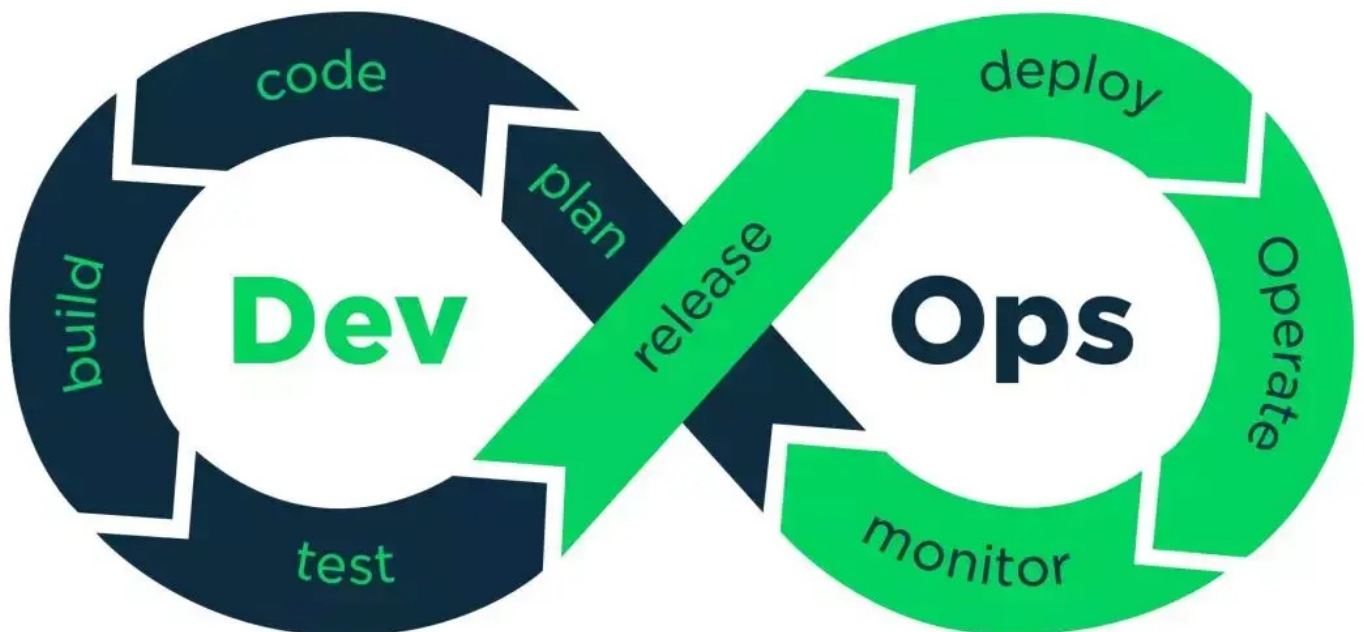
什么情况下最容易出问题？发生改变的时候最容易出问题。所以说，运维非常排斥“改变”。



这个时候，我们的DevOps，隆重登场了。

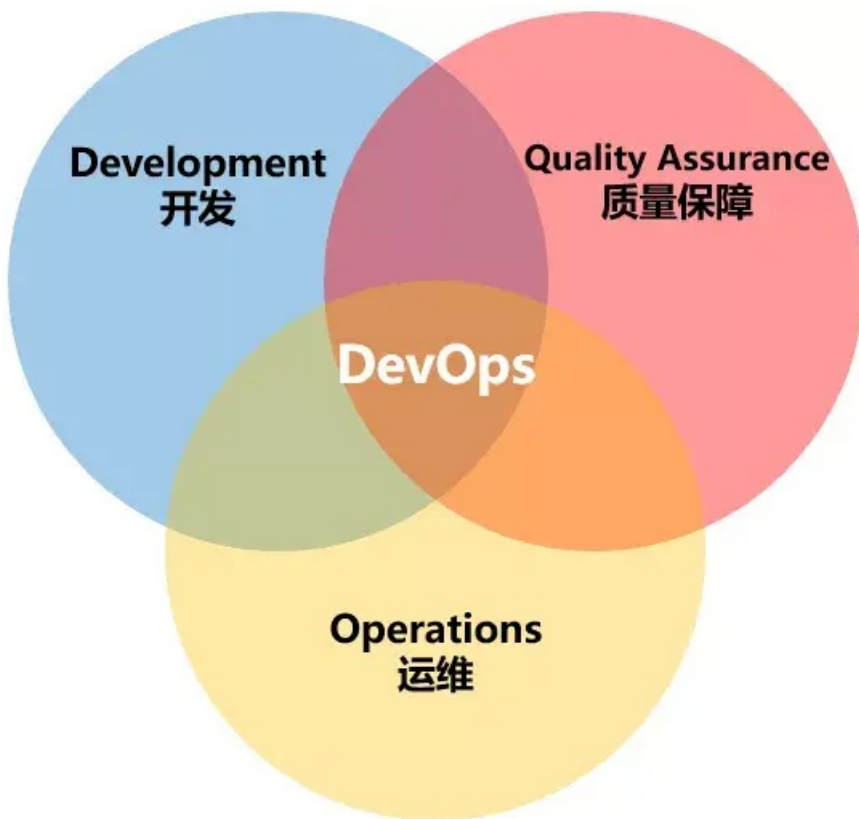
DevOps到底是什么

DevOps这个词，其实就是Development和Operations两个词的组合。它的英文发音是 /de'vɒps/，类似于“迪沃普斯”。

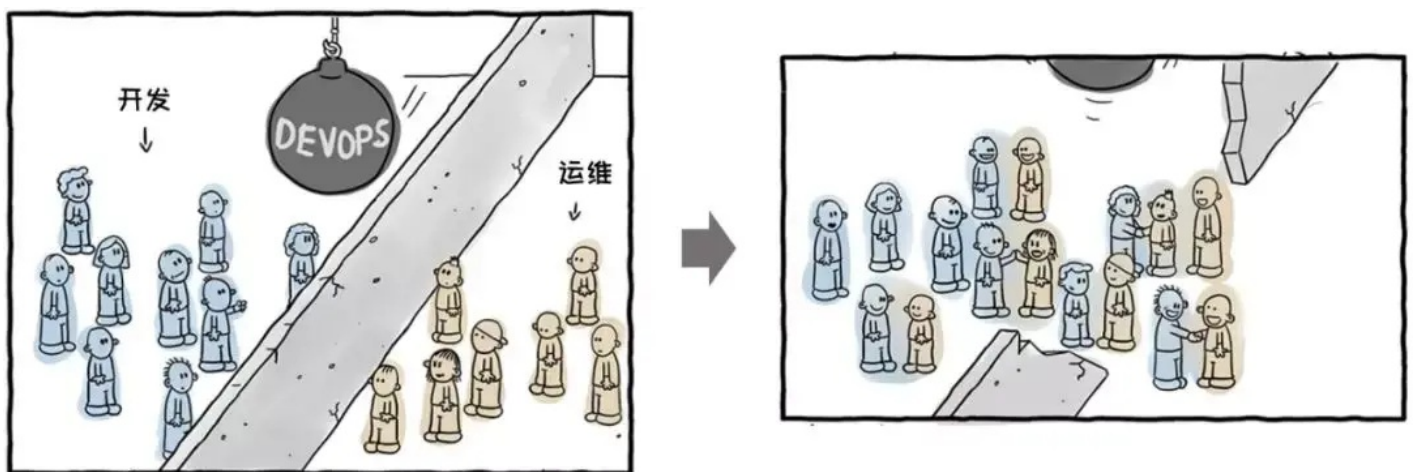


DevOps的维基百科定义是这样的：

DevOps是一组过程、方法与系统的统称，用于促进开发、技术运营和**质量保障（QA）**部门之间的沟通、协作与整合。



这个定位稍微有点抽象，但是并不难理解。反正它不是某一个特定软件、工具或平台的名字。



DANIEL STORĪ (TURNOFF.US)

破墙工具

很多人可能觉得，所谓DevOps，不就是Dev+Ops嘛，把两个团队合并，或者将运维划归开发，不就完事了嘛，简单粗暴。

注意，这个观点是不对的。这也是DevOps这些年一直难以落地的主要原因。

想要将DevOps真正落地，首先第一点，是思维转变，也就是“洗脑”。不仅是运维的要洗，开发的也要洗。员工要洗，领导更要洗。

DevOps并不仅仅是组织架构变革，更是企业文化和思想观念的变革。如果不能改变观念，即使将员工放在一起，也不会产生火花。

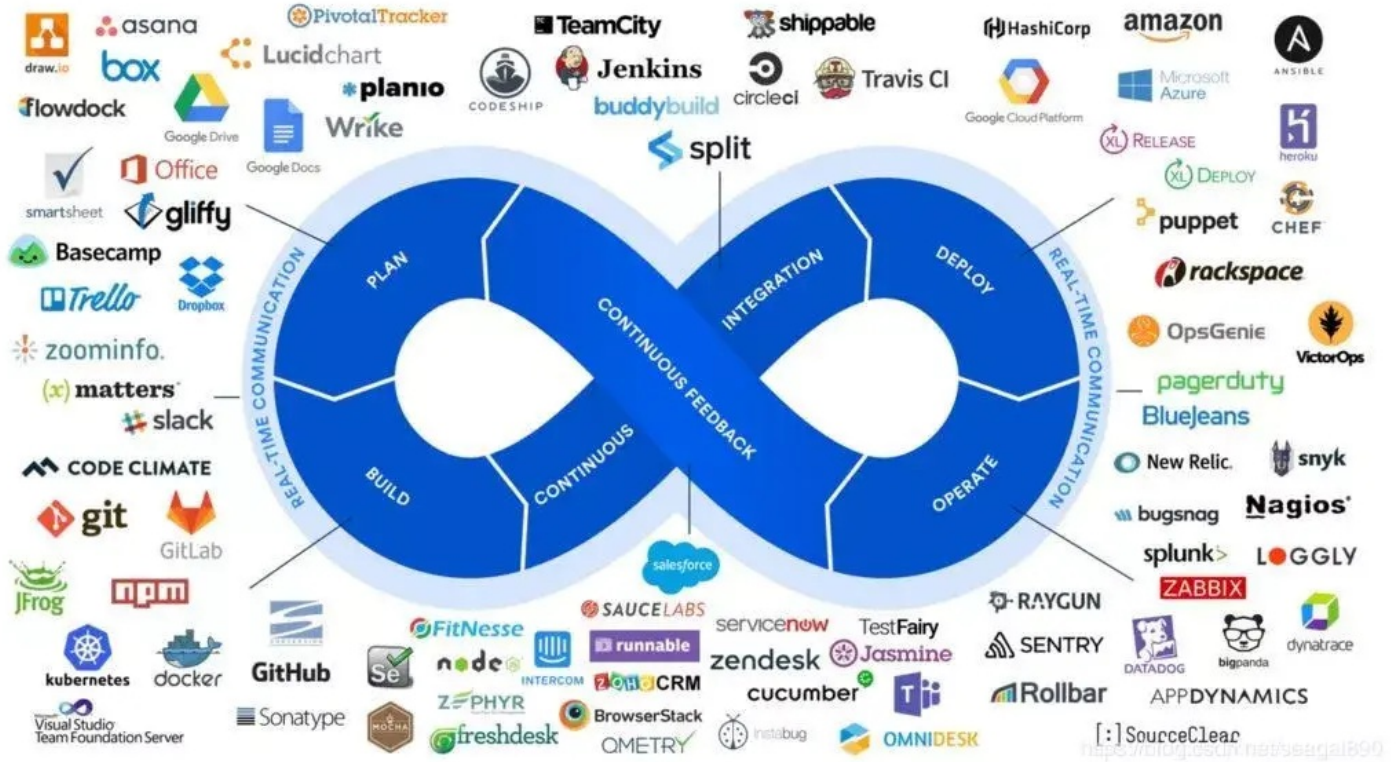
除了洗脑之外，就是根据DevOps思想重新梳理全流程的规范和标准。

在DevOps的流程下，运维人员会在项目开发期间就介入到开发过程中，了解开发人员使用的系统架构和技术路线，从而制定适当的运维方案。而开发人员也会在运维的初期参与到系统部署中，并提供系统部署的优化建议。

DevOps的实施，促进开发和运维人员的沟通，增进彼此的理（gan）解（qing）。

在思维和流程改变的同时，想要充分落地DevOps，当然离不开软件和平台的支持。

目前支持DevOps的软件实在是太多了。限于篇幅，就不一一介绍了。话说回来，现在DevOps之所以被吹得天花乱坠，也有这些软件和平台的功劳，可以趁机卖钱啊。



上述这些关键要素里面，技术（工具和平台）是最容易实现的，流程次之，思维转变反而最困难。

换言之，DevOps考验的不仅是一家企业的技术，更是管理水平和企业文化。

瀑布式开发:



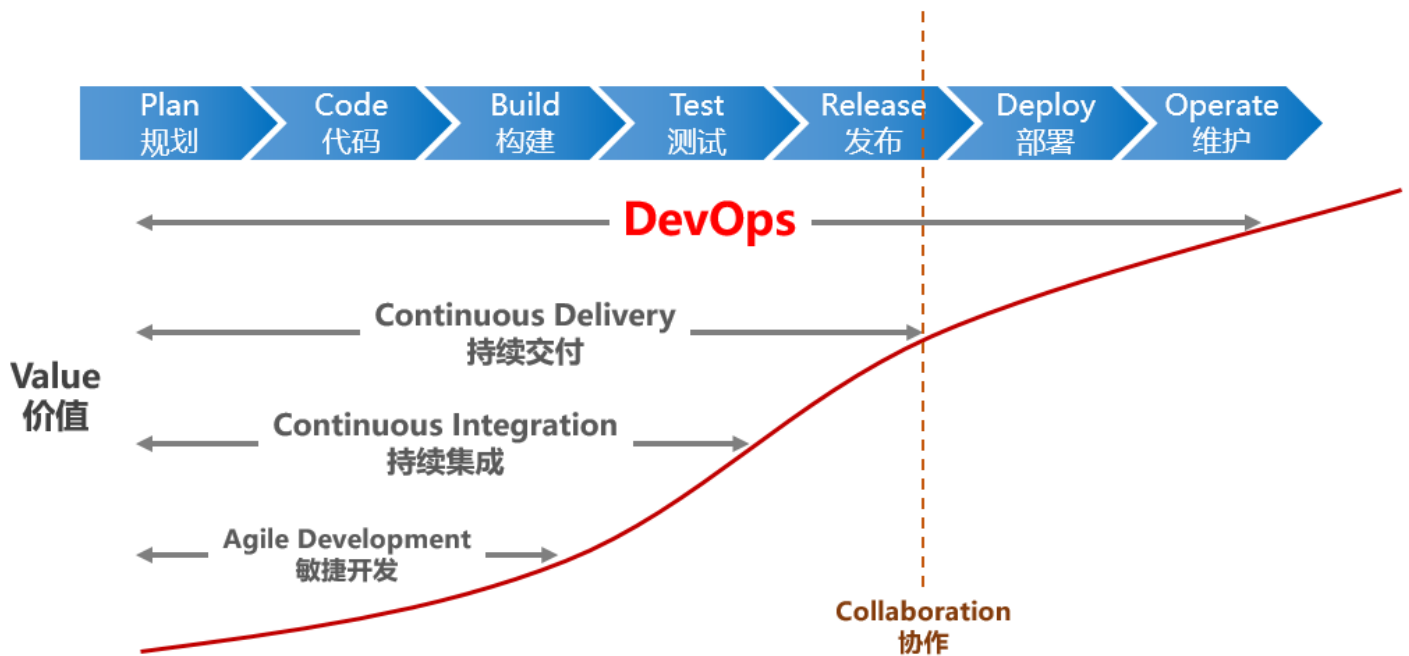
敏捷开发:



DevOps:



下面这张图，更明显地说明了DevOps所处的位置，还有它的价值：



DevOps的发展现状

DevOps这个词来源于2009年在比利时根特市举办的首届DevOpsDays大会，为了在Twitter上更方便的传播，由DevOpsDays缩写为DevOps。

目前，DevOps处于高速增长的阶段。尤其是在大企业中，DevOps受到了广泛的欢迎。

根据2018年的调查发现，74%的受访者已经接受了DevOps，而前一年这一比例为66%。

越大的企业，越喜欢DevOps。包括Adobe、Amazon、Apple、Airbnb、Ebay、Etsy、Facebook、LinkedIn、Netflix、NASA、Starbucks、Walmart、Sony等公司，都在采用DevOps。

如今，**DevOps**几乎已经成为了软件工程的代名词。

DevOps迅猛发展，相关专业人才的薪资待遇也跟着水涨船高。

根据调研，DevOps工程师在美国的平均年薪为130000美金，在中国平均年薪也在40万-50万区间，能力强者年薪百万也是比比皆是。

北京devops工程师·工资收入水平

北京devops工程师平均工资：¥ 23710/月，取自 401 份样本

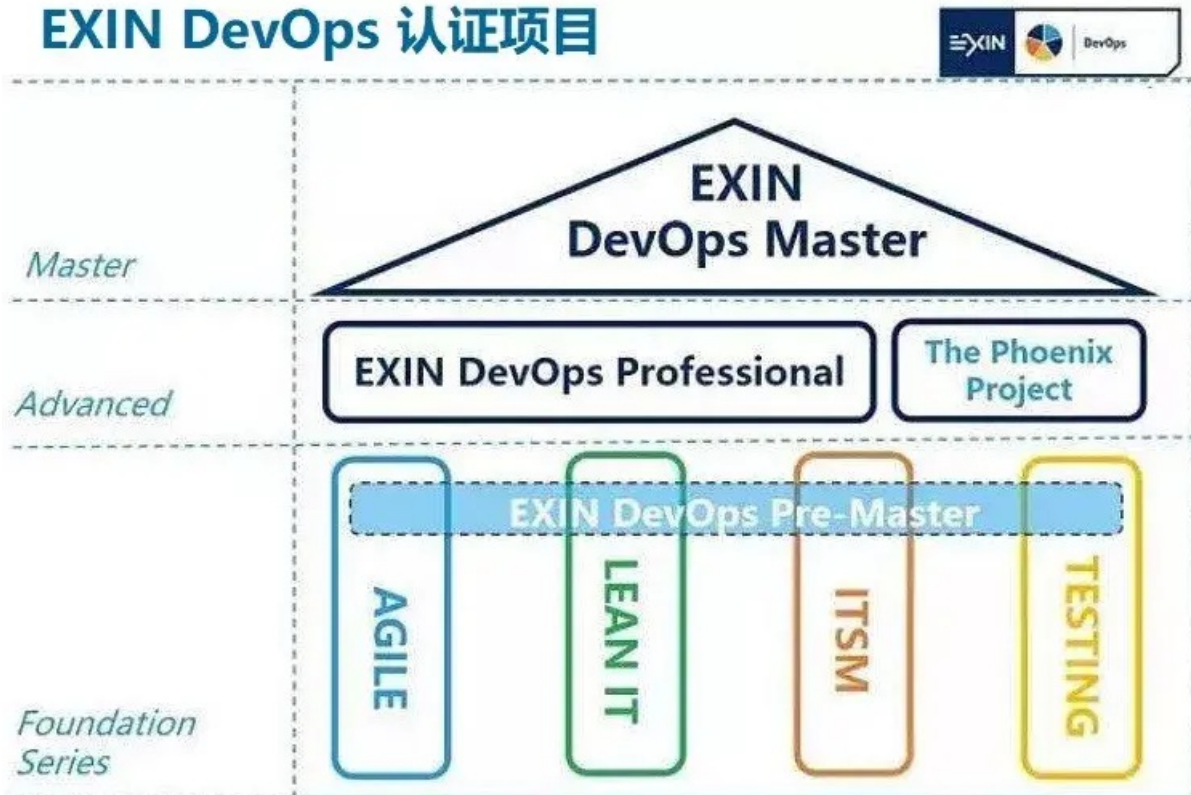
最新招聘 ⁹⁹⁺ 工资收入 就业形势



薪资的猛涨，又带动了IT工程师们学习和认证的热潮。

DevOps的认证目前最受欢迎的就是EXIN DevOps Master和EXIN DevOps Professional。这些认证的培训费用不低，但是仍然吸引了很多人踊跃报名。

EXIN DevOps 认证项目



EXIN DevOps认证体系

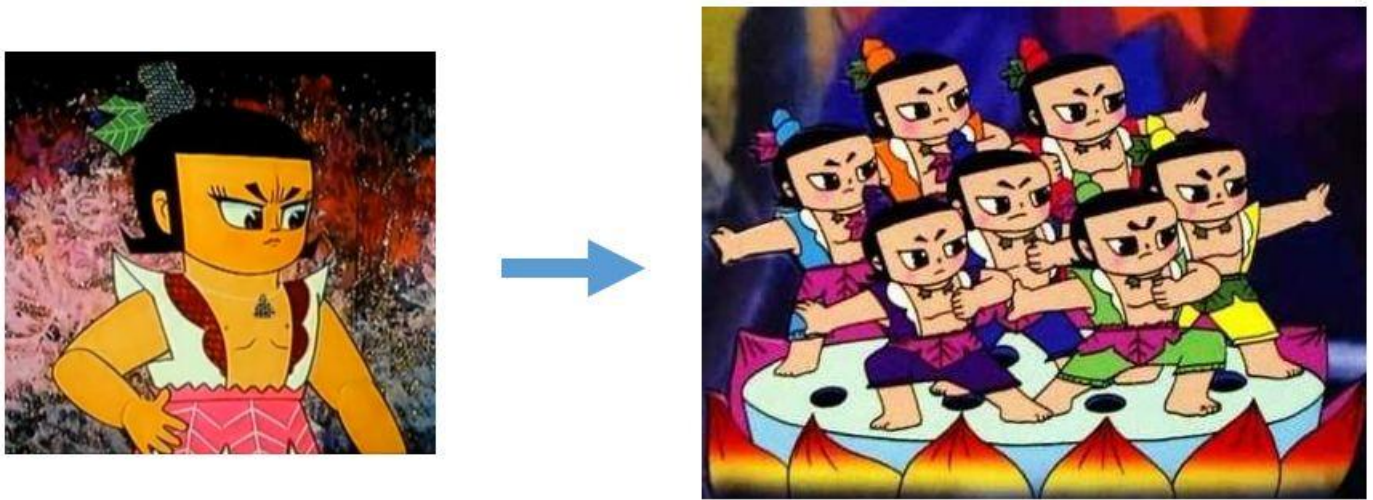
DevOps与虚拟化、容器、微服务

它们之间有什么联系呢？

其实很简单。

显然是拆分之后会更加方便。

所谓“微服务”，就是将原来黑盒化的一个整体产品进行拆分（解耦），从一个提供多种服务的整体，拆成各自提供不同服务的多个个体。如下图所示：

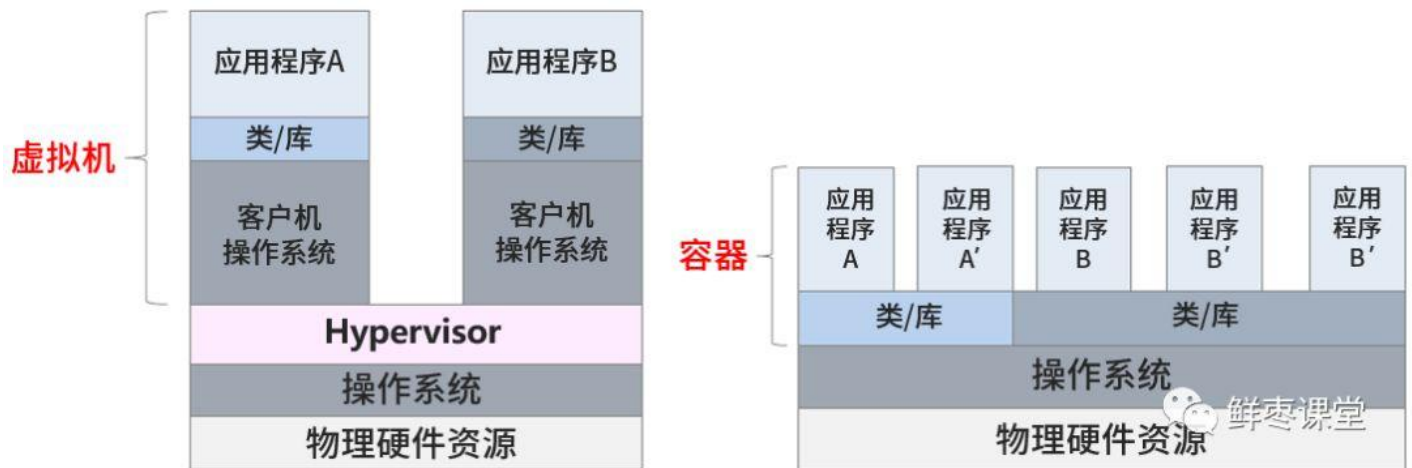


单体式架构（Monolithic）→ 微服务架构（Microservices）

微服务架构下，不同的工程师可以对各自负责的模块进行处理，例如开发、测试、部署、迭代。

而虚拟化，其实就是一种敏捷的云计算服务。它从硬件上，将一个系统“划分”为多个系统，系统之间相互隔离，为微服务提供便利。

容器就更彻底了，不是划分为不同的操作系统，而是在操作系统上划分为不同的“运行环境”（Container），占用资源更少，部署速度更快。



明白了吧？虚拟化和容器，其实为DevOps提供了很好的前提条件。开发环境和部署环境都可以更好地隔离了，减小了相互之间的影响。

这也是DevOps为什么2009年时不火，现在越来越火的一个主要原因之一。

DevOps和通信

作为一名通信工程师，小枣君再说说DevOps和通信的关系。

最开始接触DevOps的时候，我和很多人一样，都以为这是一个纯IT的概念，和我们通信没有什么关系。

后来，随着对DevOps的深入了解，我才发现，这个理念和我们通信有密切的关系。甚至说，早在十多年我刚入行的时候，其实就已经遇到了DevOps所面对的问题。

那时候（2005年左右）的电信业，产品的稳定性和可靠性是压倒一切的（其实现在也是）。所以，电信业的软件版本，更新速度非常慢。对朗讯、爱立信这样的传统巨头来说，通常大半年才出一个正式版本。这个版本经过重重把关、精雕细琢，所以非常稳定。

随着3G的兴起，全球运营商开始对网络进行更新换代。华为和中兴开始趁机切入国际运营商市场，试图从国际巨头那边分一杯羹。



除了价格之外，华为中兴最大的杀手锏是什么？就是**响应速度**。

那个时候，运营商客户对电信设备软硬件的需求非常多、非常频繁。像印度这样的地方，客户尤其难缠，每天都会提出新的需求。

当时几家海外设备商的响应速度是非常慢的，从不轻易同意接受需求。即使接受，也会答复半年甚至一年后实现。客户听了直接就崩溃了。

而华为和中兴则不同，两家公司的售前市场人员对于客户需求非常“大方”，基本上有求必应。（当时售后同事都会骂售前同事，可是仔细想来，不答应的话，根本没有进入市场的机会。）

当时华为和中兴的版本发布频率，快到什么程度呢？最快的时候，三天一个版本。甚至，长期都有大批研发人员驻扎在客户办公室，现场改版本，提交“热补丁”。

那时候是2006年，DevOps这个概念的影子都还没有。研发那边，好像也就是刚刚提出**敏捷开发**。在没有理论框架和工具平台的支持下，纯靠人力，实现了版本的飞速迭代。当然，这其中的代价和风险也是很高的。

不仅是开发人员很累很辛苦，项目里的工服（工程服务）工程师，也就是技术支持工程师，本文里面的运维工程师，更是苦不堪言。你想啊，以前几个月升一次级，现在几天就要升一次级，能不辛苦么？

但就是这样的辛苦付出，才硬生生从传统巨头嘴里抢下来市场份额，最终一步一步做大做强。

后来，才慢慢有了敏捷开发的概念，现在更是有了DevOps，各种工具啊平台啊都有了，给版本快速迭代提供了很好的条件。

对通信行业的运维来说，**DevOps**是机遇更是挑战。

就像前面说的容器、虚拟化。5G核心网采用的NFV虚拟化技术，让网元功能隔离，就大大降低了核心网工程师的操作风险和难度。这是一个积极的变化。但是，DevOps对运维工程师的能力要求，是大大提高了。。。

通信软件是IT软件的一个重要分支，和DevOps有很紧密的关系。建议通信工程师好好了解一下DevOps，升级一下自己的知识库，做好技能储备。

最后的话

时代发展到现在，客户的需求瞬息万变，市场的风向也难以预测。作为企业，想要生存下去，只有让自己变得更快。作为员工，必须让自己眼光更加长远，内心更加包容。