运维自动化实践

**精彩观点抢鲜看**

实现运维自动化闭环，最主要就是**配置管理、状态管理和变更管理能力。**

治大国如烹小虾，我们来类比餐厅老板，看如何实现炒菜的自动化：

* 首先，我要知道我的厨房里到底有些什么东西是可用的，比如备了哪些菜，有那些工具，这些就是配置管理。
* 此外，我要让系统帮我去做菜，是炒、是炖还是煮?是加水、加油还是加火，这些都是变更管理的能力。
* 最后，系统还需要能够知道我炒的菜目前是一个什么样的情况，有几分熟，温度有没有太高，油是不是太少什么的。这些就是状态管理的能力。

不管是什么样的自动化系统，实现本质就是这三个能力的闭环。

**正文**

我结合自己在运维方面的一些工作经验，介绍一下怎么样去设计和建设一套完整的运维系统以便支持分布式架构的系统。

首先简单自我介绍下，本人从事运维相关的工作有很长一段时间了，应该有十几年了吧！

我的第一份工作是做系统集成，期间建过网络、建过机房、爬过天花、搬过服务器，感觉全是各种体育锻炼，锻炼出来的身体正好就是干运维的料子。因为运维首先得有体力搬得起服务器。

印象中我搬过最重的服务器是 IBM的RS6000，应该有个几百斤吧！一个人根本扛不动，四个人搬都非常吃力。我原来身体好的时候能做一百多个俯卧撑，自从不搬服务器了，现在估计30个都做不动了。

2006我加入了腾讯，腾讯企业文化很好，经常会有很多小组活动、部门活动什么的，但是做运维很苦。经常在外面玩得时候，人刚到电话就过来了。

有一段时间我专门负责值班优化，承包了所有的告警处理，那时候每天晚上要起来四五次处理故障，一个故障最少也要搞个半个多小时到一个小时，当时一直觉得这事只熬过来别的事情就应该都是小菜一碟了。

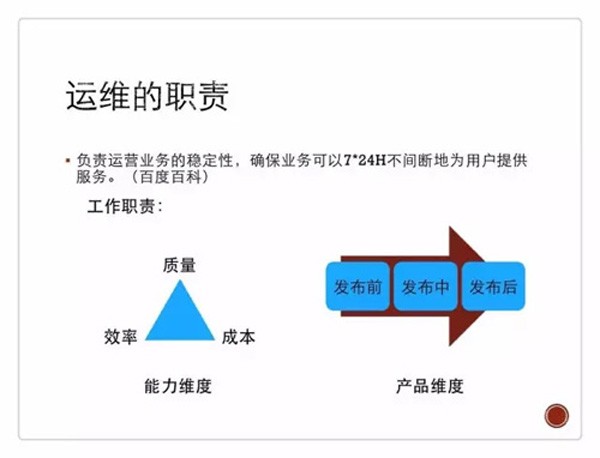
虽然当我有小孩之后，才发现原来还有比干运维更辛苦的事情的。

都说运维苦，但其实只要干好了，也可以是非常快乐和有成就感的。为了让运维都干得比较快乐。

所以，2015年的时候我们几个腾讯的同事一同创业，希望把我们的想法和经验能够传递出来。通过推动和帮助各个企业进行运维平台的建设，来解放运维的压力，帮助运维进行转型，并形成运维技术的企业竞争力。

**1、运维的趋势与挑战**

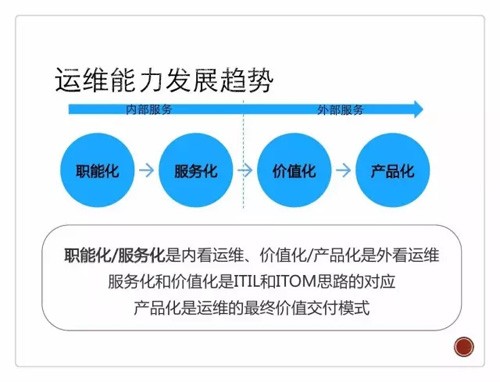
先说说目前的运维的一些变化。

[](http://s1.51cto.com/wyfs02/M00/80/3C/wKioL1c8BarzYy0AAAC6amjbPqI030.jpg-s_66786167.jpg)

首先，从运维的职能来看。只要干好一件事就可以，那就是让我们管的机器，或者业务能够一直正常运行，只要它不故障，基本就没有运维的事了。

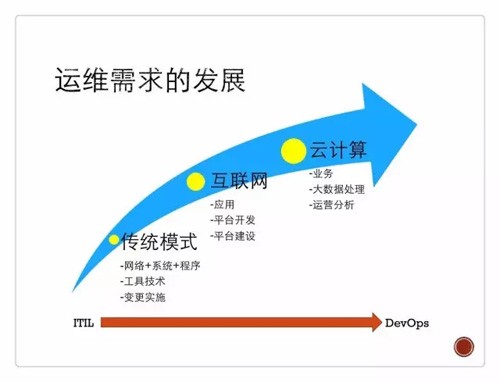
但如果出了异常，不管什么事都会有我们的责任，这就是运维。

为了做好运维，需要关注的事情很多很广。从能力维度来看，我们需要关注运营产品的质量，效率成本。从产品的生命周期过程来看，我们需要关注发布前、发布中和发布后的整个过程。

[](http://s2.51cto.com/wyfs02/M02/80/3C/wKioL1c8BavAcGIBAAD5It5pLa8282.jpg-s_654544315.jpg)

其次，从运维服务的发展趋势来看。很多年前我们经常非常会YY一下，我们在腾讯所做的运维优化和支持是不是可以打包成服务或解决方案去支持商业用户，当年觉得是异想天开。

但随着云计算的出现，大家可以看到，现在上面已经有很多的服务，其实就运维所做的优化和提供的服务。运维的价值不断地从内部向外去传递。运维能力的建设也越来越受到企业的重视。

[](http://s3.51cto.com/wyfs02/M01/80/3E/wKiom1c8BMGCnvxVAACyg7WNzmI242.jpg-s_2556010791.jpg)

最后，来看看运维能力的发展趋势。这里我列出了腾讯互联网运维团队所经历的三个阶段。

最早的时候运维只要关注各种底层的东西，如服务器、网络、交换机等，把安排的事情干完就可以。

但随着你业务规模做大，需要做的事情就没那么简单，不但要把事情做了，还得做得快，做得好，这就需要有能力平台的积累。

通过运维平台，一方面是把我们好的、正确的经验积累下来，二是能够通过平台把我们的工作变得更可靠、更高效。

当平台建设达到一定的水平之后，就进入到了第三个阶段，即数据分析和云计算的阶段，在目前大数据分析能力快速发展的情况下，数据的价值不断地被大家发现和有效利用。

运维作为数据的直接管理人，我们可以在数据的层面上去挖掘很多的价值，尤其是在服务优化和成本优化等方面，运维可以通过把有价值的数据实时采集和分析出来，并反馈给研发、产品团队，来推动产品的不断优化。

从这个角度来看，这里有很多的挑战，比如说云计算带来的一些新技术，对人能力的要求。这些不同的新开源组件，新的技术，新的方法，都会对传统的运维工作带来变革的要求。

甚至今天主题提的分布式存储，分布式架构，各种新的架构方案和技术的流程也对运维工作带来很多冲击，这些都是需要我们去面对，去变革的。

[](http://s4.51cto.com/wyfs02/M01/80/3C/wKioL1c8BauwkAUqAADE7t3dVaY831.jpg-s_3603728576.jpg)

举个例子，我刚到腾讯的时候，腾讯有一个很奇怪的面试官，叫通道委员会。他反复问我什么是ITIL，那个时候完全不懂，大家做运维的应该没有人不熟悉这个东西了。以前流行通过ITIL，通过流程的理念来管理IT系统。

这东西虽然有用，但运维来说非常的烦人，它会设定没多的门槛和流程，其实这里面很多是不科学的。

比如，我们以前要求做故障单管理，故障修复完一定要关闭故障单，我故障早都已经恢复完了，但系统总是记录我忘记结单，处理超时。为了关闭事件单，我就需要浪费额外 的时间去登陆系统，去手工关闭流程。

这种时间上的浪费，当你维护的系统变大的时候，效率的损失就边得很可怕了。所以ITIL的管理理念现在已经不流行了。

现在大家都讲DEVOPS，提研发、测试和运维的协同。以前ITIL讲分工，发布就是运维的责任，现在DEVOPS强调协同，发布就都让研发去做了。

这样很合理，因为程序发布这事你让运维去负责，其实大部分情况下出了问题运维根本识别不出来，你说别人写的代码到底有没有问题我怎么知道，真出了问题，耽误了时间，最后事情还是得交由开发来定位和处理。

而DEVOPS重视的就是高效，整个团队协同去处理这个事情，什么样的模式或什么样的人去做这个事情会变得更高效，谁就是第一责任人，我们就让他去负责，这样团队的流转就更高效和科学了。这是理念上的一些变革。

对应这些变革，运维人员的能力要求也有所变化。以前我们搬搬服务器，写个脚本什么的就可以了。但是随着技术和管理理念的变化，现在不一样了，运维也要开始写代码了，JAVA、PYTHON、C++什么的。

运维在公司的角色定位也不太一样了，以前就只是任务实施，现在慢慢朝平台建设，甚至朝运营分析方向转变。我们不但要有能力写代码还得有能力和研发一起讨论架构，和产品进行运营沟通。

真要想把运维做好，你要学的东西太多了。对于各种新技术的学习、应用和融合，如果说每个公司或每个运维都要去重头开始琢磨，那成本会非常大，对人的要求也会非常高。

**2、平台建设理念**

刚才提了很多挑战和趋势，总的来说，如果我们要去做一个运维平台，去解决运维遇到的这些问题和挑战，我们要怎么做，怎么样才能够把运维的能力通过平台去不断地提升?

我这里给了一些自己的想法，这些也是我们在腾讯这么些年积累下来的经验。

首先想讲的是平台建设的理念。很多时候做事情时，事情背后的理念往往会比做事情的方法会更重要，不知道大家认不认同这一点。

技术人员特别容易陷进一个误区，我要做一个事情，只要关注最新潮的方法和手段，背后的一些背景和因果全部都不管。

就比方说有些技术人员，他们喜欢用Markdown来写文档，但他们就从来不考虑写出来的东西商务人员该怎么使用，结果我们公司的所有商务也得学用Markdown。在公司内部也许这种妥协是容易的，但放到市场环境下这种妥协就不现实了。

所以，我觉得在建设运维平台之前，有必要先沟通一些成功的经验和方法轮。

任何公司想要建设它自己的运维平台都会是一个庞大并复杂的系统工程。这里面牵涉到方方面面。很多人往往在设计的时候会希望一步到位。

比如，希望要一步到位，希望直接就设计一个能力非常全面的平台，这个平台要包含所有需要的功能，要把权限管理好，要把安全控制住，要把稳定性做高、要把用户体验做精。

这种情况下，平台的建设就会很难，建设的周期也会非常的长，很多情况下项目可能还没有建设完成，需求就已经变了，项目也就烂尾了。

其实，我们应该考虑先从0到1，然后再从1做到N。先考虑把核心最迫切的功能功能先快速实现，只有用起来才是好平台。简单的功能可以先实现，再不断地慢慢再完善，不断地丰富它的功能，这样过程中的平台收益才会最大化。

[](http://s5.51cto.com/wyfs02/M02/80/3E/wKiom1c8BMKyPqPYAACYRpEDeL0922.jpg-s_3345426060.jpg)

第二，标准先行。做过运维的人都知道，我要管理的事情非常多，环境会非常复杂。当我们推行标准化的时候，它带来的最大好处是降低平台设计和实现的难度。

标准化能力和系统建设能力是一个翘翘板。我们在业务架构标准化方面做得好一点，那么系统建设的复杂度就低一点。

比如说，如果我们的运维标准化做得较差，我们有10种不同的硬件，每种配置都不一样，上面操作系统也不一样，这种情况下我们的平台就需要做不同系统和环境做兼容性，系统实施成本就非常的高了。

标准化先行，这样系统的建设复杂度和难度都会相应降低。

[](http://s4.51cto.com/wyfs02/M00/80/3E/wKiom1c8BMLCwm-NAACH4-FdaW4040.jpg-s_3038343262.jpg)

第三，快速尝试。复杂系统的设计和建设都是非常难的，而且对于没做过的东西，其实很多方面在一开始的时候跟本想不清楚，也想不明白，这种情况下，应该先简单一点快速实现DEMO原型，而不是停留在反复的讨论和设计。

只要系统在应用环境中跑一阵子，很快你就会发现问题和找到相应对策的。

[](http://s4.51cto.com/wyfs02/M02/80/3C/wKioL1c8BaySXc_2AAB5-kki-b8061.jpg-s_1842719020.jpg)

第四，接受不完美。腾讯现在自动化运维平台对外有两个品牌，织云和蓝鲸，而我刚好在两个团队都待过，也都经历了两个平台的建设和成长。

比如织云平台的建设，最早是从打包规范的推广开始。早期的平台只是一个简单的脚本工具平台，之后才逐渐补充了一此管理功能，如Web管理，组件管理，包发布、配置发布等，最后才逐渐建成面向全业务管理和调度的织云平台。

这里一定是个逐步完善、逐步演进的过程。腾讯也有很多一开始就规划得很大的项目，现在看起来，基本上这些大项目都死光了。所以，这里在设计和建设中，大家都要能够接受或是忍受不完美。

对技术人员来说，这点也许会有点困难吧。记得上次参加GOPS全球运维大会的时候大众点评的一个讲师提到一点很有道理：

我们在面对不完美的平台时，要知道没有任何一个平台是完美的，也没有任何一个技术是完美的。但我们决策用不用它的时候，可以评估，如果我用它，它带来的好处、优点比缺点更多，那这个平台和技术就是有价值的。

[](http://s5.51cto.com/wyfs02/M00/80/3C/wKioL1c8BazCSUEqAACGVU8ofAQ560.jpg-s_346571421.jpg)

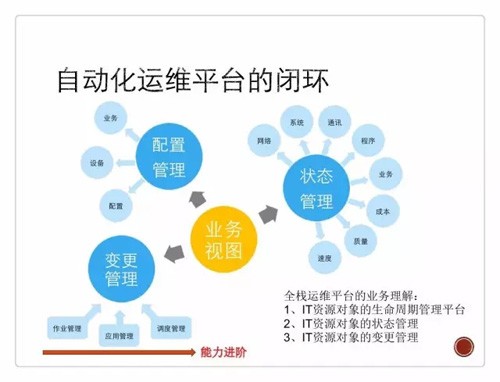
第五，业务导向。建成的平台能否发挥作重就看我们的推广能力，这里实际也是有一些技巧的。任何一个新的东西，任何一个新的技术、在推广的时间其实都会遇到阻力。

就腾讯内部的平台推广经验来看，最有效的手段就是先找一些比较配合的团队、或是重点的业务，这些团队和业务相对的资源也比较丰富。当我们快速完成了这些试点业务的推广，就能够在公司内部建立业务标杆，并形成影响力和口碑。

当建立了业务标杆之后，后面的业务再去推动就会容易很多了。所以，过程中平台快速的上线，尽快输出成效，是非常重要的。

**3、平台建设实践**

讲完方法论，现在回到具体的系统设计和实现。我们应该如何来设计这个运维平吧呢?作为一个完整的运维平台，一定要考虑形成运维管理能力的闭环，并逐步实现自动化。

[](http://s1.51cto.com/wyfs02/M00/80/3E/wKiom1c8BMOiD3nRAADqhDDE3No752.jpg-s_2108210780.jpg)

如何才能实现运维的自动化闭环呢?**最主要就是掌握配置管理、状态管理和变更管理能力。**

运维的自动化大家可能不太好理解，我举个简单的例子。比如我是餐厅老板，我希望实现炒菜的自动化。那要怎么实现呢?其实非常简单：

* 首先，我要知道我的厨房里到底有些什么东西是可用的，比如备了哪些菜，有那些工具，这些就是配置管理。
* 此外，我要让系统帮我去做菜，是炒、是炖还是煮?是加水、加油还是加火，这些都是变更管理的能力。
* 最后，系统还需要能够知道我炒的菜目前是一个什么样的情况，有几分熟，温度有没有太高，油是不是太少什么的。这些就是状态管理的能力。

不管是什么样的自动化系统，实现本质就是这三个能力的闭环。

对于运维平台来说，我们通过配置管理能务来收采和管理所有的系统资源，通过状态管理能力实时的监控资源的运行情况，最后再根据监控的结果来对现多的资源进行变更和调度。

能力闭环实现了，自动化能力也就实现了。

[](http://s5.51cto.com/wyfs02/M00/80/3C/wKioL1c8Ba2RK-f-AAE9p2LDO-Q509.jpg-s_3086982785.jpg)

在运维平台的设计实现上。我里有一张PPT，大家应该经常能够在老王的演讲中看得到。

为了实现一个完整的平台能力，我们需要对整个平台进行分层设计，最底层是各种硬件和资源的管理平台，它能够抽象地去管理各种物理资源和逻辑资源，和这些资源自身有关的管理能力也都会放在这一层。

再往上是通用能力层，这一层实际上是把运维所负责的常规服务都实现成为系统能力。 比如运维日常的配置管理、域名管理、文件管理等。

通过第二层的平台把运维的工作全部服务化，而这里服务化的核心就是把日常手工工作都进行系统封装，变成服务接口。这种服务接口可以供外部的服务或更上层的系统进行调用和扩展。

当运维系统的服务能力构建成熟，我们就可以上更上层构建基础能力平台，比如说用于管理交付的持续交付平台，用于管理服务状态的智能监控平台等。

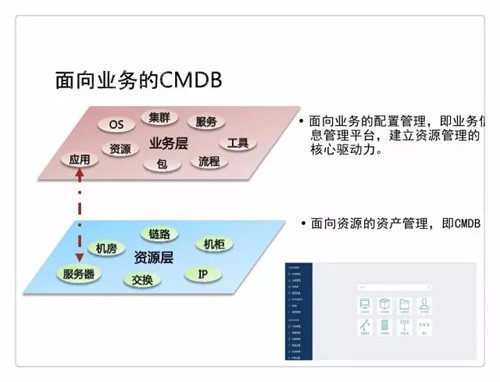
当这些基础能力平台完成后，最后我们可以开始建设各种向向业务场景的精细化管理平台。比如：

* 我们希望能够提升产品服务质量，我们可以建设相应的业务可用性分析平台;
* 我们希望降低产品成本，我们可以建设相应的容量优化平台;
* 我们希望提升变更效率，我们可以建设相应的设备扩容调度平台等等。

从这里可以看到平台的分层设计，一定是从去场景化的基础能力开始向场景化的服务能力延伸，所以我们的系统建设步骤也应该遵循这个规律。

进一步展开运维平台的实现，现在让我们来看一下每一个模块的重要功能和设计挑战。

**4、配置管理平台介绍**

[](http://s5.51cto.com/wyfs02/M01/80/3E/wKiom1c8BMTQGsIPAADPNXXZXu4464.jpg-s_3195632764.jpg)

首先说说CMDB配置管理。其实配置管理这个东西非常简单，不管什么样规模的公司，肯定都会有自己的配置管理的系统或是办法。最简单的配置管理，它其实就是一个excel文档。

如果复杂一点，我们可以搭建一个数据库，它一样也可以实现CMDB的功能。但是如果真正要把这个东西做好，要考虑的问题就比较多了。

比如说，传统的ITIL理念是非常重视CMDB的，但它把所有精力都集中在硬件资源的管理，如机房、机柜、交换机、网络，这些层面的东西。ITIL下的CMDB会把这些东西管理得很好。

但就运维所维护的业务来说，这些信息只有其中的一个很小的一部分，而且它们对业务自动化运维所能够提供的价值其实是非常低的。因为这些硬件信息不管管理得再怎么精细，在具体操作的层面还是需要依赖人来进行操作。

相反，在应用程序的层面，我们可以做的事情就很多。如果能力到位了，我们甚至可以实现故障自愈、自动调度、弹性计算等高级的自动化能力。而这些能力，需要我们的CMDB能够关注和管理面向业务的配置信息。

比如说我业务资源是什么样的，我的程序是什么样的，我的应用是什么样的，我的权限是什么样的，我的流程，我的策略是什么，这些东西是非常有价值的。

只有把这些东西全面管理起来，才能够真正去驱动整个业务的自动化流程。举个例子，比如说常规的一些磁盘空间告警，我要想实现故障治愈，首先我得有明确的处理策略。

比如每一台机器对应的磁盘空间清理策略是什么，而这些是需要在CMDB中管理起来的。当出现任何异常的时候，我们的自动化系统直接到CMDB里查询相应的策略，就可以实时针对不同的业务去实现自动恢复。

除了业务信息的管理，CMDB设计还需要考虑到数据模型的管理能力，即怎么样去支持和实现灵活的数据存储结构。

我们要管理的数据不会都象EXCEL一样简单，相反在真正的业务环境中，一定会是多层的关联数据结构。而且这个数据结果也不可能就是一成不变的，它一定会在业务运营的过程中需要进行动态的调整和修正。

这种情况下，我们的CMDB就需要考虑到存储可灵活性和可扩展性。我们需要实现可配置、可定义，并支持分级数据模型的配置，这些都是建设CMDB时候考虑的事情。

CMDB这里最后要讲讲配置信息的维护，做过运维的人应该都会有同感，配置信息的维护是非常讨厌这的事情。

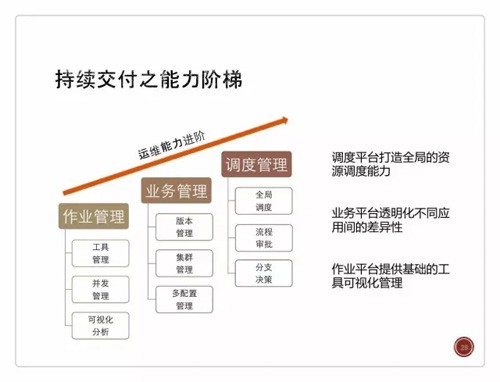
腾讯在早期的时候，配置信息也都是手工维护的，如果我们进行了服务器的上、下线，事后一定要记得登陆系统去手工更新信息，不然其他人再进行操作就会出错。时间久了信息的失去价值了。

所以我们以前的做法公司会考虑配置准确性，也即一段时间运动一次，人工的去修正信息。

更好的方法，应该是要去做信息的自动发现和自动更新。怎么实现呢?

* 一方面这里我们可以做CMDB的探针，它直接去采集设备上的状态和信息，实时的上报回CMDB并更新。
* 另一方面可以提供接口和外围的管理系统对接，当每一次变更结束了以后，都通过接口把变更的信息实时回写和更新CMDB，以止实现信息的自动维护。

**5、变更管理平台介绍**

[](http://s4.51cto.com/wyfs02/M01/80/3C/wKioL1c8Ba6T_NOzAAC-xl6OExM560.jpg-s_651299138.jpg)

关于变更管理平台的实现。这里可以考虑分阶段的实现方案。对于一些基础比较差的团队来说，要想一步到位是非常困难的，而且如果能力没有达到一定的水平，有一些自动化能力也不一定可以用得起来。

**这里需要建设根据团队的技术实力，选择合适的节奏分阶段进行建设。**

建议先从脚本平台开始，先实现最基础的作业管理能力，之后再实现业务管理和流程管理能力。

[](http://s4.51cto.com/wyfs02/M01/80/3E/wKiom1c8BMTwzMWkAAD12R1TnTU640.jpg-s_2309617483.jpg)

作业管理也可以理解为脚本管理。实现自动化最简单的办法，其实就是编写脚本。每个公司的运维团队都一定会有些积累的脚本。这些都是运维人员最宝贵的资产。

所以在变更管理平台建设的过程中，我们应该首先考虑把这些资产的价值发挥出来。

通过实现作业管理平台，来提供统一的可视化脚本管理能力，它一方面能够通过分享和复用来降低脚本开发的成本，另一方面也可以实现集中控制，并保证操作的可靠性。腾讯目前的蓝鲸平台，其实本质上就是一个作业管理平台。

如果团队能力更强一点，就可以开始考虑实现业务管理能力。通过脚本来实现自动化虽然比较简单，但面对业务管理的场景时，我们就会发现即使是一个简单的业务，都会需要大量的脚本才有可能够把业务的关联环境维护好。

这种情况下，我们需要考虑集成的业务管理能力，需要把业务通用的维护手段和管理方法封装成通用的平台功能：

* 比如业务发布后的进程守护、日志清理、启动初始化;
* 再比如业务本身的版本管理、集群管理、实例管理、回滚管理等等。

这样做最大的好处是把业务维护的复杂度封装在平台内部，对外只需要关注到一些通和的服务接口即可。

[](http://s5.51cto.com/wyfs02/M02/80/3E/wKiom1c8BMThTftsAAFJL2nPLtI921.jpg-s_209143283.jpg)

变更管理的第三个阶段是流程和调度管理。

当我们把所需要的各种原子操作和业务管理能力都实现之后，那么我们就可以考虑实现最终的自动化流程和调度管理能力。

比如我们希望实现放牛式的服务器管理，服务器挂了可以直接把服务器杀掉而不影响业务。这时需要流程化的自动管理能力，它不但需要和资源池交互调动新设备，而且还要和业务管理平台交互，把业务实例发布上去。

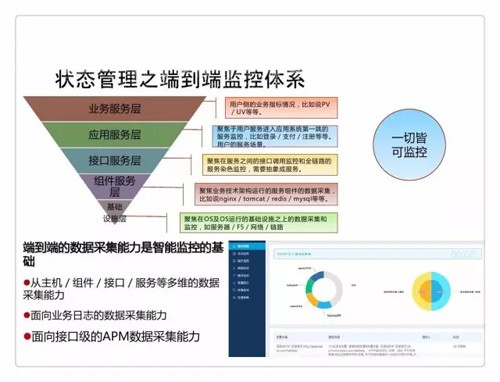
随即还要调动自动化测试能力，检测一下新上的服务器是不是好的，并加载灰度上线的逻辑，把服务器慢慢上线到运营环境中去。

整个复杂的过程不但需要把各种基础工具和平台组织起来，还需要根据接口和各个不同的系统进行交互，并根据交互的结果进行工具和后续流程的决策。

分阶段的变更管理平台建设，可以有效的降低平台建设的启动成本，并且不同的平台能力也可以较好的适配企业不同团队在IT能力发展过程中所处于的不同阶段的需求。

**6、状态管理平台介绍**

对于状态管理平台。说得简单一点就是监控平台。对运维自动化来说，监控平台是一个最主要的活动触发源。如果我不知道业务运行的状态，不知道业务有没有异常，那么自动化也就无法发挥它应有的价值。

[](http://s2.51cto.com/wyfs02/M02/80/3C/wKioL1c8Ba-QdImyAAEUJfOrXCo396.jpg-s_1261676528.jpg)

一个好的监控系统需要能够实现端到端的监控。

目前市面上有很多开源的监控组件，但基本都是单一纬度的监控，比如主机监控或是日志监控的产品。

但真正做过运维的人都应该清楚，现网中出现的很多故障都并没有那么简单，大部分情况下，这种单一唯独的监控都没有办法很好的覆盖业务的监控需求。

比如：我们希望关注业务的运行情况，而采用了CPU的监控，它虽然可以发现到CPU的异常。但对于程序有否有BUG就无能为力了。所以一个全面的监控体系一定是需要考虑端到端的监控能力。

**每一个系统的最终用户都一定是人，所以我们可以从用户的角度，模拟用户的访问路径，从而实现一个全面的端到端监控能力。**

我们可以把用户请求过程中所经过的节点和链路全部监控起来。再通过综合的分析和汇聚，从而有效果的掌握业务的运行状态，并及时发现异常。

具体实现时，我们可以从最底层的基础网络和链路逐步往上是应用服务器、应用组件、组件请求、服务质量、到最上层的业务状态实现全部的监控覆盖。

[](http://s1.51cto.com/wyfs02/M00/80/3E/wKiom1c8BMbixzhbAADLwxDagW0662.jpg-s_715362480.jpg)

在监控的通道上，目前有两种主流的方式。一种是外部的探测，别一种是内部的主要采集和上报。

内部采集方式，我们需要在服务器上部署监控的的探针，它会根据需要定时的去检查业务的运行状态，并收集有价值的日志信息。

由于不同业务所需要的采集逻辑和收集的信息都不一样，所以监控的探针需要设计成一种插件化的模式，以便支持不同业务的灵活扩展。

对于外部探测的监控模式，一般的实现是在业务生产系统的外部寻找合适的探测节点，如公网上的服务器或外网CDN上的缓存节点。**通过这些节点的访问，我们可以模拟用户的访问路径，并还原用户的链路质量对业务的影响。**

对于比较强的团队来说，我们在建设监控系统时，可以同时考虑集成两种不同的监控通道能力，并实现监控能力的互补。

**监控能力对于自动平台来说，最大的价值是能够完成事件的触发。也即实现从数据发现、分析、定位、到问题解决的闭环。**

通过这个闭环，我们可以构建各种故障的自愈能力。通过及时的发现异常，快速的恢复，能够有效的提升业务的可用性和质量。

举个实例的例子：当系统发现机器的CPU有异常的时候，需要对 CPU高负载进行故障干预和恢复，这种情况下我怎么做?

1. 我们可以取到告警的信息，告警里会告诉我这台机器的IP地址和告警的值;
2. 通过IP，可以从CMDB中查一下这个机器属于哪个业务，再根据业务信息可以查询到同业务下还有那些机器;
3. 然后我们通过同业务的IP地址把其它机器的当前CPU值都查询出来，得出的平均值再去和告警的CPU值来对比;
4. 最后判断出是否需要系统干预。如果需要修复，系统会根据告警的IP地址到CMDB中去查询相应的恢复策略，再进行处理。

通过这种灵活和完整的验证处理闭环，我们就可以构建出各种可靠的自动故障恢复策略。

[](http://s2.51cto.com/wyfs02/M00/80/3C/wKioL1c8BbDi4cGeAAByBNUHElo897.jpg-s_1565202446.jpg)

最后，让我们再来总结一下之前提到一些方法论。

先是标准先行、小步快跑、容忍缺陷、业务导向。

对于复杂的运维系统，不要贪大求全，关键是先构建出配置管理、状态管理和变更管理的能力闭环。

我们可以先从标准化开始，通过推动标准化来降低运维系统的设计和实施复杂度，然后才是具体的系统实现。

* 第一步是配置管理，最简单的配置管理可以先从搭建一个MySQL开始。
* 之后的变更管理，可以先梳理运维常用的脚本，形成团队的脚本库或的操作标准。
* 监控能力的建设可以依赖外部的开源组件，也可以通过运维脚本加CRONTAB来实现。

从最简单的平台，逐步积累标准化和服务化能力，等大家形成标准化和服务化的意识和习惯了，之后再逐步完善和丰富运维系统的能力。

对于运维自动化管理平台来说，不管具体的实现手段是什么，只要我们能够覆盖之前所介绍的领域，能够满足业务的需求，那么这个平台就一定会非常的有生命力，非常的有价值。

以上这些就是我对运维自动化平台建设和经验和理解，谢谢大家。

【编辑推荐】

1. [WOT2016吴兆松：Zabbix监控自动化的未来如何发展](http://os.51cto.com/art/201603/506801.htm)
2. [运维必备制度：故障分级和处罚规范](http://os.51cto.com/art/201603/506878.htm)
3. [出色的学习能力，才是运维工程师唯一可持续的竞争优势](http://os.51cto.com/art/201603/507780.htm)
4. [如何以项目的运作方式进行运维管理](http://os.51cto.com/art/201603/507861.htm)
5. [你赞同这五大运维体系划分吗](http://os.51cto.com/art/201603/507969.htm)