

期中总结：3个典型问题答疑及如何高效学习

2019-11-30 石雪峰

DevOps实战笔记

[进入课程 >](#)



讲述：石雪峰

时长 17:43 大小 16.24M



你好，我是石雪峰。不知不觉中，专栏已经上线快两个月了，整体进度过半。我在专栏写作之初就给自己定下了一个小目标：**认真对待每一条留言**。到现在，单单是回复留言，我就已经写了 3 万多字了。

其实，对我来说，每一次看留言和回复留言，都是一个不断反思和学习的过程。实际上，很多时候，很多留言和讨论甚至比文章本身都更精彩，也更接地气。

今天是期中总结，我分为两个部分内容来讲：

第 1 部分：我从众多留言中挑选了 3 个典型问题，进一步展开讲解。

第 2 部分：我想跟你说说心里话。两个月的高强度写作，也让我从一个讲师的角度，重新审视了“如何高效学习”这件事情，我把这些想法分享给你，希望可以帮助你更好地提升自己。

典型问题

首先，我们来看一些典型问题。

问题一

敏捷开发模式没有花费大量时间去研究业务，这会不会出现因为对业务没有分析透，导致方向偏离，甚至系统开发到一半发现总体业务架构不合理，需要返工的情况呢？

相信你也知道，实施 DevOps 有助于产品快速和高质量的交付，那么，我想问的是：快速和高质量的交付，是否就一定意味着业务的成功呢？显然没有这么简单。

实际上，影响业务成功的因素有很多，比如，行业趋势、产品竞争力、用户消费习惯、政策法规等等。在这众多因素之中，需求质量的高低，或者说，需求是否靠谱，也很重要。

毕竟，如果交付了一大堆没用的需求，不仅没法提升业务，反而还会浪费大量的时间和精力，错过真正有价值的机会。

我们身处在一个飞速变化的时代，企业对于用户想要什么其实并不清楚。很多需求都是人为拍脑袋拍出来的。在提出一个新需求的时候，需求价值到底有多少呢？这不仅很难预测，而且还很难衡量。

所以，产品人员就倾向于采用“广撒网”的方式，提出一大堆需求，来提升命中的几率。毕竟，如果一次猜不对，打不准，那就多打几次呗。

这么看来，采用敏捷开发方式，还是瀑布开发方式，与需求是否靠谱并没有直接关系。即便是采用瀑布模式，依然也有“大力出悲剧”的案例，比如摩托罗拉的铱星计划。

既然无法事先预测需求是否靠谱，那么要解决这个问题，就需要业务团队和交付团队的通力协作了。

从业务侧来说，就是要采取精益创业的思想，通过最小可行产品，将需求进行拆解，通过原型产品降低市场的试错成本。这就引出了我在🔗“业务敏捷”这一讲中提到的影响地图、卡诺模型、用户故事等一系列的手段和方法，核心还是采用持续迭代、小步快跑的方式来获取市场反馈。正因为如此，更加灵活拥抱变化的敏捷开发模式才被广泛地接受。

说完了业务侧，再来看看交付侧。

一个想法被提出来以后，需要经过软件开发交付过程，才能最终交付到用户手中。那么，就要用尽一切手段来缩短这条交付链路的时长。

如果开发的时间成本是一定的，那么剩余的部分，就是 DevOps 中的各种工程实践试图要去解决的问题。

比如，通过持续集成来降低软件集成中的解决成本，降低软件缺陷在最后一刻被发现的修复成本；通过自动化测试，降低大量手工回归测试用例执行的成本，降低新功能导致已有功能出现回退的修复成本。软件交付过程中的其他部分也大都如此，这也是每个领域都会有自己的实践集合的原因。

反过来看，功能上线之后，依然需要交付侧提取、汇总和及时反馈业务指标，来证明需求的靠谱程度，从而帮助业务侧更加有序地进行决策。对反映不好的功能及时止损，对反映不错的功能加大投入。

这样一来，**业务侧的需求拆解、需求分析减小了需求颗粒度，提升了需求的靠谱度；交付侧的工程实践大大缩短了上线交付周期，提升了质量。**这就帮助业务在不增加成本的前提下，可以验证更多的需求。这个过程的成本越低，频率越高，企业存活下来的几率和整体竞争力也会越高。这也正是 DevOps 想要解决的核心真问题。

问题二

公司对于配置管理的关注度不是很高，有没有什么好的落地实践方法，来建设完整的配置管理体系呢？

在专栏的🔗第 10 讲，我从 4 个核心原则出发，介绍了配置管理的相关知识，引起了很多同学的共鸣。

的确，作为一个长期被忽视，但是格外重要的实践，配置管理不仅是诸多 DevOps 工程实践的基础，也是工程能力的集大成者。

正因为如此，**配置管理体系的建设，并不只是做好配置管理就够了。实际上，这还依赖于其他工程实践的共同实施。**

关于配置管理怎么落地，我跟你分享一个案例。

这家公司最早也没有专职的配置管理，软件的集成和发布都是由研发团队自行管理的。推动建立配置管理体系的契机，源于公司决定加快版本发布节奏，从三周一个版本变成两周一个版本。看起来，这只是版本发布周期缩短了一周，但是，就像我在专栏 [🔗第 4 讲](#) 中演示的部署引力图一样，想要达成这个目标，需要方方面面的努力，其中就包括配置管理。

于是，公司决定引入配置管理岗位。初期，他们重点就做两件事：

重新定义分支策略，从长分支改为了短分支加特性分支的模式；

管理集成权限，从任何时间都能集成代码，到按照版本周期管控集成。

在这个过程中，配置管理同学梳理了代码仓库的目录结构和存储方式，并基于开发流程建立了在线提测平台，从而实现了研发过程的线上化以及权限管理的自动化。

接下来，配置管理与平台和流程相结合，开发过程开始向前、向后延展。

向前：在需求管理阶段，建立需求和代码的关联规范，严格约束代码提交检查，并且将构建工具和环境配置等纳入统一管控，可追溯历史变更；

向后：在部署运维阶段，定义版本发布和上线规则，建立单一可信的发布渠道，可统一查询所有正式发布版本的信息，包括版本关联的需求信息、代码信息、测试信息等。

团队在走上有序开发的正轨之后，就针对发现的问题，逐步加强了平台和自动化能力的建设。

代码提交失控：做集成线上化，测试验收通过之后，自动合并代码；

环境差异大：通过容器化和服务端配置管理工具，实现统一的初始化；

构建速度慢：通过网络改造和增量编译等，提升构建速度。

这样一来，版本发布这件事情，从原本耗时耗力的操作，最终变成了一键式的操作，团队也达成了预期的双周发版的目标。

在这个案例中，配置管理更多是从流程和平台入手，通过规则制定、权限管控、统一信息源，以及版本控制手段，重塑了整个开发协作的交付过程。

所以，在把握原则的基础上，面对诸多实践，想要确定哪些实践可以解决实际问题，**最好是要从预期结果进行反推。**

如果你不知道该从哪里入手，不妨看看现在的软件交付流程是否是由配置管理来驱动的，是否还有一些数据是失控和混乱的状态，版本的信息是否还无法完整回溯。如果是的话，那么，这些都是大有可为的事情。

总之，**任何一家公司想要落地配置管理，都可以先从标准化到自动化，然后再到数据化和服务化。**这是一条相对通用的路径，也是实施配置管理的总体指南。

问题三

度量指标要如何跟组织和个人关联？这么多指标，到底该如何跟项目关联起来呢？

我在 [第 19 讲](#) 中介绍了正向度量的实践，引发了一个小高潮。文章发出后，有不少同学加我好友，并跟我深入沟通和探讨了度量建设的问题。由此可见，当前，企业的研发度量应该是一个大热门。

但是，度量这个事情吧，你越做就越会发现，这是个无底洞。那么，在最最开始，有没有可以用来指导实践的参考步骤呢？当然是有的。我总结了四个步骤：**找抓手、对大数、看差距、分级别。**

第 1 步：找抓手。

对于度量体系建设来说，很多公司其实都大同小异。最开始的时候，核心都是需要有一个抓手来梳理整个研发过程。这个抓手，往往就是需求。因为，**只有需求是贯穿研发交付过程始终的，没有之一。**

当然，你也可以思考一下，除了需求，是否还有其他选项？那么，**围绕需求的核心指标，首先是需要提取的内容**。如果，连一个需求在交付周期内各个阶段的流转时长都没有，那么，这个度量就是不合格的。

第 2 步：对大数。

对大数，也就是说，当度量系统按照指标定义，提取和运算出来指标数据之后，最重要的就是**验证数据的真实有效性，并且让团队认可这个客观数据**。

很多时候，如果公司里面没有一套权威指标，各个部门、系统就都会有自己的度量口径。如果是在没有共识的前提下讨论这个事情，基本也没什么意义。所以，说白了，一定要让团队认可这些大数的合理性。

第 3 步：找差距。

抓手有了，核心大数也有了，大家也都承认这个度量数据的客观有效性了。但是，在这个阶段，肯定有些地方还是明显不合理。这个时候，就需要对这个领域进一步进行拆分。比如，测试周期在大的阶段里只是一个数字，但实际上，这里面包含了 N 多个过程；比如，功能测试、产品走查、埋点测试等等。

如果没有把表面问题，细分成各个步骤的实际情况，你就很难说清楚，到底是哪个步骤导致的问题。所以，**在达成共识的前提下，识别可改进的内容，这就是一个阶段性的胜利**。

第 4 步：分级别。

实际上，不是所有指标都可以关联到个人的。比如，如果要计算个人的需求前置周期，这是不是感觉有点怪呢？同样，应用的上线崩溃率这种指标，也很难关联到一个具体的部门。

所以，**我们需要根据不同的视角和维度划分指标**。比如，可以划分组织级指标、团队级指标和项目级指标。

划分指标的核心还是由大到小，从指标受众和试图解决的问题出发，进行层层拆解，从而直达问题的根本原因，比如用户操作原因、数据计算原因、自动化平台原因等等。当然，这是一件非常细致的工作。

我们再来回顾下，我们刚刚深入剖析了 3 个 DevOps 的典型问题。

首先，你要非常清楚地知道，DevOps 在面对未知需求时的解题方法和解题套路，那就是业务侧尽量拆解分析靠谱需求，交付侧以最快、最低的成本完成交付。它们之间就是一个命运共同体，一荣俱荣，一损俱损。

配置管理作为 DevOps 的核心基础实践，在实施的过程中，并不只局限在单一领域。实际上，要从研发流程优化的视角出发，驱动标准化、自动化和数据可视化的能力建设。

最后，关于度量指标部分，你要注意的是，**向上，要支撑核心指标；向下，要层层分解，展示真实细节。**

讲解完这 3 个典型问题之后，接下来进入第 2 部分，这也是我极力要求增加的部分。其实，我就是想跟你说说心里话。

如何高效学习？

跟你一样，我也是极客时间的用户，订阅了很多感兴趣的课程。在学习的过程中，我一直在思考，如何在有限的时间内高效学习。直到我自己成为了课程老师，从用户和老师两个角度思考这个问题，有了一些感悟，想要跟你分享一下。

忙，是现在大多数人的真实生活写照。我们每天从早到晚，忙于工作，忙于开会，忙于刷手机.....忙得一塌糊涂。

但是，如果要问，过去的一天，自己都在忙什么，要么是大脑一片空白，要么是碎碎念式的流水账。可见，我们每天忙的很多事情，都没有什么价值。

其实，很多事情，都没有我们想象得那么重要。我们常常把目光聚焦于眼前，眼前的事情就变成了整个世界。但是，如果把时间拉长到一周，甚至一年，你会发现，这些事情，做与不做没有什么分别。

正因为时刻处于忙碌的状态，所以，抽出一整段时间学习，就变成了一件奢侈的事情。但我要祝贺你，因为至少你比大多数人有意识，有危机感，愿意拿出零碎的时间，来充实自己。

既然花了这么难得的时间，你肯定希望能有所收获，无论是在知识上，还是能力上，抑或是见识上，至少不白白浪费这段时间。

那么，我想问的是，你真的有收获吗？

史蒂芬·科维曾经说过，大多数人聆听的目的是为了“怼回去”，而不是为了真正的理解。

Most of people listen with the intent to reply, but not with the intent to understand.

这里面的“怼回去”稍微有点夸张，实际上，我发现，当我在交流的时候，脑海里总是不自觉地想象如何回复对方，而不是专心地听对方讲话，感悟他的意图和情绪。

所以你看，听这种学习方式，总是会受到固有思维模式的影响。也就是说，在很多时候，我们往往会把自己置身于一种评论者的身份。

那么，什么是评论者的身份呢？这就是说，**站在一种置身事外的立场，以一种审视的角度，来看待每一件事情，并试图找到一些问题。当然，这些问题，都是在已有的认知局限中发现的。**

这些反馈，对于知识的生产者而言，其实是一件好事，因为他能够时刻审视自己，反思自己，并从中找到不足之处。

但是，对学习者来说，能不能在学习的过程中，暂时放弃评论者的身份，转而做一个实践者呢？

比如，以极客时间的专栏为例，对于作者提到的内容，你有哪些不同的观点呢？面对同样的问题，你又有哪些更好的手段呢？

其实，每一个作者之所以能成为作者，都有他的独到之处。那么，**能够让他的思想为你所用，让他的知识与你互补，让你自己成为交流的赢家，这才是对得起时间的更好选择。**

最后，以极客时间的专栏为例，我认为：

60 分的体验，就是可以看完所有的文稿，而不是仅仅听完课程音频；

70 分的体验，就是可以仔细学习文稿中的附加资源，比如代码、流程图以及补充的学习信息等。这些都是精选的内容，可以帮助你在 15 分钟之外，扩充自己的知识面；

80 分的体验，就是可以积极参与到专栏的留言和讨论中，甚至可以就自己的问题跟作者深入交流，建立连接；

90 分的体验，就是可以结合工作中的实际场景，给出自己的思考和答案，并积累出自己的一整套知识体系，并且，可以反向输出给其他人；

100 分的体验，就是持续改进。我想，能够具备这种思想，可能要比 100 分本身更重要。

那么，你想做到多少分的体验呢？你可以自己想一想。

好了，接下来，我们即将进入“工具实践篇”，希望你可以继续保持学习的热情，坚持下去，期待美好的事情自然发生。

思考题

对于前面已经更新的内容，你还有什么疑惑点吗？或者说，你在实践的过程中，有什么问题吗？

欢迎在留言区写下你的思考和答案，我们一起讨论，共同学习进步。如果你觉得这篇文章对你有帮助，也欢迎你把文章分享给你的朋友。

DevOps 实战笔记

精要 30 计，让 DevOps 快速落地

石雪峰

京东商城工程效率专家



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 特别放送：学习DevOps不得不了解的经典资料

精选留言 (3)

 写留言



maomaostyle

2019-12-02

学以致用方可收获

展开 ∨



鲍建飞

2019-12-01

终于到了实践篇了

展开 ∨



许童童

2019-11-30

老师关于高效学习这部分说得非常好，被动学习肯定是没有主动学习效果来得好，不过我

们利用碎片的时间学习也比将时间都用于娱乐要好很多，现在的工作、学习竞争压力越来越大，学会如何高效学习也显得非常重要。

展开 ∨

