

19 | 不再掉队，研发流程、工程方法趋势解读和展望

2019-10-07 葛俊

研发效率破局之道

[进入课程 >](#)



讲述：葛俊

时长 15:52 大小 14.54M



你好，我是葛俊。今天，我们就来聊一聊研发流程和工程方法的一些趋势吧。

软件行业从诞生之日起，就一直是充满了发展和变化，各种工程方法、研发模式不断涌现，而且涌现的速度越来越快。对于开发团队和个人来说，这既是挑战也是机会：说是挑战，是因为我们需要持续学习才能跟得上它的发展；说是机会，是因为如果能够快速学习并应用这些实践，我们就可以在竞争中取得优势。

这些挑战和机会，并不强依赖于资源、背景，其实是为我们提供了一个相对公平的竞争环境。这，也是软件行业这些年来涌现了许多白手起家的成功公司和个人的重要原因。

在今天这篇文章中，我会针对当前比较流行的研发流程、工程方法的趋势，尤其是与国内研发比较相关的部分，做一些解读和展望，和你说说我的理解、预测，希望作为你以及你的团

队，在技术选型以及工程方法选择上的一些参考。

接下来，我将会从协作方式、云计算平台、应用开发和 AI 这 4 个方面与你展开讨论。

协作方式的相关趋势

在我看来，协作方式的相关趋势，主要表现在以下两个方面：

首先，团队远程办公、灵活工时办公，会越来越普遍；

其次，聊天工具和其他工具的集成，会越来越普遍。

接下来，我与你说说我为什么会有这样的预测吧。

团队远程办公、灵活工时办公，会越来越普遍

远程办公之前用得不是特别多，主要是因为沟通效率比较低，也不利于掌控团队氛围。但，视频会议以及团队协作工具的快速发展，使得这些问题不再那么明显了。

这时，远程办公的巨大好处，也就是可以**克服人才的地域局限性**，就凸显出来了。比如，很多人在考虑要不要应聘工作时，都会考虑通勤距离。所以，一旦能够打破这个地域限制，招聘的空间就会开阔很多。

远程办公的另一个好处是，可以**大量减少通勤时间，进而提高研发产出**。近些年来，交通拥挤、通勤时间过长的情况，越来越严重。美国有科学研究表明，如果每天上班单程超过 30 分钟，就会对人的情绪产生比较大的负面影响。而国内的一线城市，从我接触到的样本看，上班单程低于 30 分钟的情况大概只占 20%。

所以最近这些年，越来越多的公司开始或多或少地采用远程办公的方式，以此来缓解通勤时间过长给员工造成的压力。有些公司做得比较彻底，让绝大部分开发人员绝大部分时间都是远程办公，我知道的公司包括 GitHub 和 Atlassian。而更多的公司，则采用的是每周允许员工部分时间在家办公，比如 Facebook 的开发人员每周三可以在家办公。

至于灵活工时，本质是用任务来驱动。这一点比较符合软件开发的特点，在第 1 篇文章中我已经与你讨论过了，这里不再过多展开了。在硅谷，基本上所有的软件公司，都是这样操作的。在国内，这种情况也越来越普遍。

关于远程办公方式，我还有两个小贴士：

一是，要尽量使用视频会议，而不是电话会议。有研究表明，电话会议不如面对面沟通的效率，主要是因为缺少了由面部表情和肢体语言传递的信息。

二是，做好信息的数字化。比如，建设好任务系统、文档系统等，从而让研发人员在工作中能尽量从这些系统中获取信息，而不用过于依赖面对面或者实时聊天系统，依然能够高效工作。

聊天工具和其他工具的集成会越来越普遍

聊天工具和研发流程中其他工具的紧密集成，最近几年在国外很流行，最典型的就是 Slack。它可以和任务工具、代码审查工具、部署工具、监控系统进行集成，从而实现我前面[第 4 篇文章](#)中提到的工具网状互联，提高研发效能。

所以，我觉得下面这种工作方式会越来越普遍：聊天工具里有各种各样的聊天室和聊天机器人，团队成员在不同的聊天室讨论不同的话题，并通过聊天机器人和其他工具进行集成和交互，来获取信息以及执行一些操作。比如，询问聊天机器人当前线上的服务分别是哪些版本、相关需求有哪些、Commit 有哪些、具体的开发和测试人员是谁等。又比如，通过运维机器人添加两台机器到集群中去。

聊天是人类最自然的沟通方式之一，所以在聊天室和机器人的帮助下，执行这些操作会非常高效，提高研发效能也就不在话下了。

云计算平台的相关趋势

正如我在[第 16 篇文章](#)中提到的，云计算正在改变我们开发软件的方式。利用好云计算平台的趋势，是提高团队研发效能必须要做的事儿。在我看来，**云计算最大的趋势应该是 Docker 和 Kubernetes 带来的各种可能性。**

Kubernetes 自诞生以来，凭借着 Google 公司的强大技术支撑和经验积累，发展得异常迅猛，现在，它已经成为了容器编排的事实标准。在我看来，其中最大的作用是，我们可以用它来建设 PaaS。

使用 PaaS，我们可以快速部署和管理应用程序，把容量预配置、负载均衡、弹性扩容和应用程序运行状况监控的部署细节交给平台来管理，让研发团队聚焦于业务，对高效业务开发极其有用。但是，PaaS 平台容易出现灵活性不足的情况。

比如，我之前在 Stand 公司开发后端服务时，非常希望能够使用 AWS 提供的 PaaS 服务（比如，Elastic Beanstalk 服务），来减轻团队在运维方面的工作压力，但试用之后，发现其无法支持一些定制化的需求，比如平台的技术栈的灵活性不够，而且更新的时候透明度不够。所以我们只能忍痛放弃，最终选择使用更下一层的 IaaS 服务，通过自己管理虚拟机来部署和管理服务。

而解决上述灵活性不足问题的一个方法是，灵活生成新的 PaaS 平台。但，PaaS 平台的建设必须依托于下层的 IaaS 才能实现，所以技术要求很高，工作量和资源要求也很大，只有专门做 PaaS 的公司和云厂商才有能力提供 PaaS。

Kubernetes 出现后，提供了强大的容器管理和编排功能，事实上是实现了一种基于容器的基础设施的抽象，也就是实现了 IaaS 的一个子类。所以通过它，我们终于可以方便地建设定制化的 PaaS 了，一个具体的例子是 FaaS（Function as a Service）。Kubernetes 的出现，极大地降低了建设 FaaS 的工作量，所以很快出现了基于它的实现，比如 [OpenFaaS](#)、[Fission](#)。

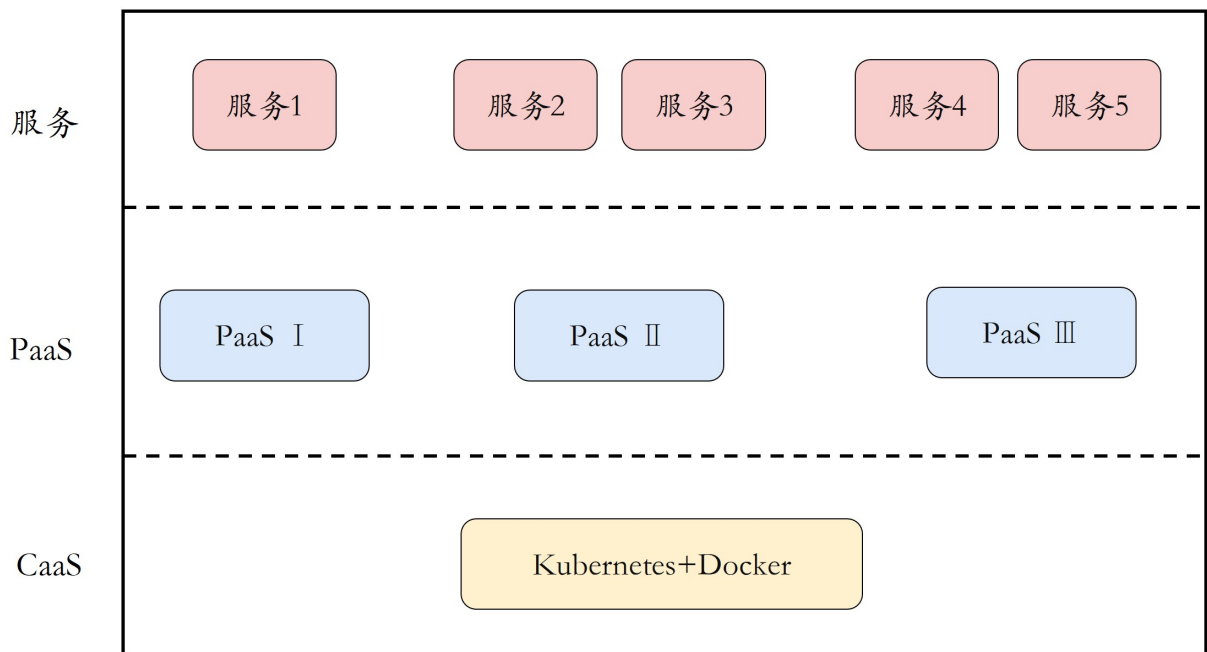
正是基于 Kubernetes 提供的构建 PaaS 的能力，我预期，将来越来越多的产品会构建在基于 Kubernetes 和 Docker 的 PaaS 之上。

今天，很多公司对 Kubernetes 还是直接使用，也就是通过一个对 Kubernetes 比较了解的运维团队，来支持公司的服务运行在 Kubernetes 集群上。但 Kubernetes 的学习成本比较高、学习曲线比较陡峭，整个系统的运行并不是那么顺畅。所以，我觉得将来的趋势将会是这样的：

如果你所在的团队比较小，可能会选择第三方通过 Kubernetes 提供的 PaaS 平台；

如果你所在的团队比较大，可能会基于 Kubernetes 建设适合自己的 PaaS 平台。

另外，我觉得很可能会出现这样的情况：整个公司运行一套 Kubernetes 作为 IaaS，上面运行多个不同的 PaaS 平台，支持各种服务的运行。如下图所示。



备注：CaaS（Containers as a Service），是允许用户通过基于容器的虚拟化来管理和部署容器、应用程序、集群，属于 IaaS 平台的范畴。

应用开发的相关趋势

随着云计算的普及，分布式计算会越来越流行，最典型的例子莫过于微服务的盛行。设计正确的架构，来支持产品的开发和部署，是云时代高效研发的重要因素。

在我看来，应用开发的相关趋势，主要表现在云原生开发方式和服务网格两个方面。

云原生的开发方式

应用程序运行在云端，需要基于云的架构设计，这就意味着我们需要一套全新的理念去承载这种开发模式。这套理念，就是云原生开发。

由 Heroku 创始人 Adam Wiggins 提出并开源、由众多经验丰富的开发者共同完善的[12 原则（12-factor）](#)，是云原生开发理念的理想实践标准。

服务网格

复杂的服务拓扑结构，是云原生应用程序的一个重要难点。而服务网格正是用来处理服务间通信的专用基础设施，它提供了应用间的流量、安全性管理，以及可观察性，比较好地解决了这一问题。

在服务网格架构中，流量管理从 Kubernetes 中解耦，每一个服务对网络拓扑并不知情，通信都是通过代理来进行的。所以，我们就可以通过代理来方便地完成很多工作。

比如，在部署阶段进行的集成测试，我们就可以借助它来实现。假设 A 和 B 是系统中的两个服务，并且 A 会调用 B。我们希望在部署 A 的一个新版本时，在生产环境进行 A 和 B 的集成测试：

通过 A 的 egress 代理，让它对下游服务发出的请求都自动加上一个 x-service-test-b 的 header；

而在 B 的 ingress 代理接收到有 x-service-test-b 的请求时，自动通知 B 这是一个测试请求。测试完毕之后，修改代理去除这个 header 即可。

目前，关于服务网格有两款比较流行的开源软件，分别是 [Linkerd](#) 和 [Istio](#)，都可以直接在 Kubernetes 中集成，也日渐成熟。

AI 方面的相关趋势

这几年，AI 绝对是最火的话题之一。我们讨论研发流程和工程方法，自然也绕不过 AI。不过，AI 在软件研发上的应用还处于起步阶段。

在我看来，AIOps、CD4ML（Continuous Delivery for Machine Learning，机器学习的持续交付）、语音输入是比较适合在软件研发中落地的。

接下来，我们分别看看这 3 个方面。

AIOps

做好 AI 的前提就是，有大量的数据积累。而运维相关的工作，就有大量的线上日志、监控、指标等数据，所以 AIOps 是比较容易落地的一个方向。具体来说，我觉得**AI 可以从以下两个方面来提高运维效率：**

第一个方面是，检测并诊断异常。也就是说，通过对历史故障数据进行分析学习，找到规律，从而在新故障出现或者快要出现时，能够实时探测到，并给出一些可能的诊断。

第二个方面是，智能地对资源进行弹性的扩容、缩容及流量切换。目前，我们通常采用 CPU 利用率、内存利用率等少量维度的指标，去判断是否要扩容或缩容，判断逻辑比较

简单。而利用 AI，我们可以收集更多维度的数据，综合分析后自动进行扩容或缩容，更合理地利用资源。另外，更进一步地，我们还可以利用 AI 做更智能的流量切换，进一步提高资源利用率。

CD4ML

在机器学习中有许多需要人工参与的步骤，我们可以通过提高这些步骤的自动化程度来提高其效率。CD4ML 就是将持续交付实践应用于开发机器学习模型上，以便于随时把模型应用在生产环境中。

语音输入协助软件开发

现在，语音输入效果已经比较好，应该算是 AI 最成熟的领域。从我个人来说，我就经常使用。比如，我会用 Amazon 的 Echo 玩游戏、用小米音箱控制家里的空调、通过 Siri 使用滴滴打车等。

实际上，我在写这些专栏文章的时候，就经常使用语音输入形成第一版文字，然后再手工编辑完善，能够节省不少时间。

在我看来，语音输入将来同样可以应用到软件开发工作中。比如，用语音来输入程序；再比如，通过移动设备上的语音输入完成一部分研发相关工作，包括申请机器、运行流水线，查看系统状态等。

小结

今天，我从协作方式、云计算平台、应用开发和 AI 这 4 个方面，与你分析了如何在软件开发工作中运用这些趋势，去提高研发效能。同时，我也对这些趋势做了大胆预测。当然了，欢迎你在留言中与我分享你的其他看法。

为了方便你学习、理解，我把这些趋势的重点内容整理到了一张表格中，供你参考。

类别	流程、方式
协作方式	远程办公
	灵活工时办公
	聊天工具和其他工具集成，完成查询和执行操作的工作
云计算平台	Docker和Kubernetest灵活提供PaaS
应用开发	云原生的开发方式
	服务网格
AI	AIOps
	CD4ML
	语音输入协助软件开发

除此之外，软件开发行业还有非常多的、创新性的方法和实践，我无法一一展开了。比如，移动端开发中，[GraphQL](#)可以高效、灵活地从后端获取数据，以及使用 JavaScript 之外的语言进行 Web 开发的框架，比如[Vapor](#)。

我一直很庆幸，能够在软件研发这个行业工作。因为它有足够的想象力，给了开发者持续发展的空间。上面提到的这些趋势和方向，都让我非常兴奋。希望能带给你一些帮助，至少引发你的一些思考。当然了，对这些趋势的选择、解读和预测，都只是我个人的观点，欢迎你通过留言与我分享你的看法。

思考题

使用 AI 来提高研发效能，是一个很有趣的话题。除了我今天提到的这些，你觉得还有哪些可能的方式吗？将你的预测与想法，分享给我吧。

感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎你把这篇文章分享给更多的朋友一起阅读。我们下期再见！

研发效率破局之道

Facebook 研发效率工作法

葛俊

前 Facebook 内部工具团队 Tech Lead



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 | 每个开发人员都应该学一些VIM

精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。