

## 特别放送：学习DevOps不得不了解的经典资料

2019-11-28 石雪峰

DevOps实战笔记

[进入课程 >](#)



讲述：石雪峰

时长 16:56 大小 15.52M



你好，我是石雪峰。

今天又到了特别放送的环节，在学习交流 DevOps 的过程中，经常有人会问这样的问题：

我想学习 DevOps，可以推荐一些好的书和资源吗？

DevOps 相关的最新行业案例，我可以在哪里获取呢？

你是怎么知道这么多有趣的故事的呢？

这些问题的“出镜率”特别高，所以，我今天专门来跟你聊聊有关 DevOps 学习资料的事情。

你应该也有感觉，在这个信息爆炸的时代，如果想要了解一个新的事物，相关的信息不是太少，而是太多了。像 DevOps 这种热门话题，相关的资料网上一搜就一大把。各种新书也像“采用了 DevOps 实践”一样，发布频率越来越快。信息一多，我们就很容易焦虑，这么多资料，什么时候才能看完啊？

更何况，如果单单只是臻选有用的资料，就要花费大量的时间，按照精益的理论来说，这也是不增值的活动呀。在这个时间稀缺的时代，想要花大段的时间投入到一件事情上，找到一个靠谱和有价值的信息，就成了很多人开始学习的第一步，

所以，为了让你在专栏之余可以更加有效地持续学习，我特意整理了一份我认为 DevOps 从业人员需要了解和关注的资料，你可以参考一下。

需要强调的是，有针对性地精读一本好书的一部分内容，要比泛泛地读好几本书要更有收获一些，也就是“**贵精不贵多**”，先定下一个小目标，然后沉下心来反复地学习实践，这个道理在大多数领域都是适用的。

## 一份报告

如果说，DevOps 领域有行业公认的权威资料的话，DevOps 状态报告自然是不二之选。

从 2014 年开始，这份报告每年发布一次，主要编写方也经历了好几次变迁，从最开始的 Puppet 实验室、IT Revolution 到 DORA (DevOps Research & Assessment) 的加入，再到去年，DORA 和 Puppet 分家，两边各自推出了自己的 DevOps 状态报告。

但从影响力来说，我更推荐 DORA 的这份报告，从去年开始，这份报告正式改名为：加速度，DevOps 状态报告。

提到 DORA，你可能不太熟悉，但是如果说到 DORA 的两位核心创始人 Nicole 博士和 Jez Humble，相信你一定有所耳闻，他们也是我今天推荐的一些书的作者。

有意思的是，去年 DORA 宣布加入谷歌，其主要成员也被谷歌云收编，比如 Jez Humble，目前就是谷歌云的技术布道师。

回到报告本身，我在 2017 年就开始进行报告的本地化工作。从近两年来看，报告的体量在持续扩大，比如，今年的报告洋洋洒洒有 80 页内容，而且是全英文的。

那么，关于这份报告，重点是要看什么呢？纵观过去几年的报告模式，我给你画个重点：**核心是看趋势、看模型和看实践。**

**首先看趋势。**

每年的报告都会有一些核心发现，这些发现代表了 DevOps 行业的发展趋势。比如，今年的报告就指出，云计算能力的使用依然是高效能组织和中低效能组织的分水岭，所以，如果公司还在纠结是否要上云，不妨从 DevOps 的角度思考一下，使用云计算能力带给交付能力的提升可以有多明显。

另外，公司内的 DevOps 组织比例也从 2014 年的 14% 提升到了今年的 27%。由此可见，越来越多的公司在拥抱 DevOps，至少从组织层面可以看到，越来越多带有 DevOps 职责，或者是以 DevOps 命名的团队出现。这对于公司内部职责的划分和团队架构演进，具有一定的指导意义。

当然，不得不提的，还有衡量 DevOps 实施效果的 4 个核心指标，也就是**变更前置时间、部署频率、变更失败率和故障修复时长**。

从 2014 年的第一份报告开始，每年的报告都在对比这 4 个核心指标在不同效能团队之间的变化和差异。实际上，就我观察，国内很多公司的 DevOps 度量体系，都深受这些指标的影响，或多或少都有它们的影子。

可以说，这 4 个指标已经成为了衡量 DevOps 效果的事实标准，甚至有人直接把指标拿给老板看，说：“你看，高效能组织比低效能组织的故障恢复时长要快 2000 倍，由此可以证明，DevOps 是势在必行的。”

我个人觉得，没有必要纠结于数字本身，这东西吧，看看就好，更多的还是要透过数据看趋势。

比如，去年的指标数据就显示，在交付能力方面，不同组织间的差距在缩小，相应的质量维度的指标差异却在拉大。这就说明，通过初期的自动化能力建设，团队可以快速地提升交付水平。但是，由于缺少质量能力的配套，很容易产生更多的问题，这就带来一个警示，在快速提升交付能力的同时，质量建设也不能落在后面。

## ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have...



**208**  
TIMES MORE  
frequent code deployments

**106**  
TIMES FASTER  
lead time from  
commit to deploy



**2,604**  
TIMES FASTER  
time to recover from incidents

**7**  
TIMES LOWER  
change failure rate  
(changes are  $\frac{1}{7}$  as likely to fail)



Throughput Stability

关于报告，其次是看模型。

我在 [第 4 讲](#) 中提到过一个观点：**任何技术的走向成熟，都是以模型和框架的稳定为标志的**。因为当技术跨越初期的鸿沟，在面对广大的受众时，如果没有一套模型和框架来帮助大众快速跟上节奏，找准方向，是难以大规模推广和健康发展的。

在软件开发领域是这样，在其他行业也是如此，要不然，为啥会有那么多国标存在呢？所以说，模型和框架的建立是从无序到有序的分水岭。

在今年的状态报告中，研发效能模型进一步细化为软件交付运维模型和生产力模型。今天我不会深入解析模型本身，但我会的专栏后面的内容中结合实际案例进行详细解释，从而帮助你更好地理解。

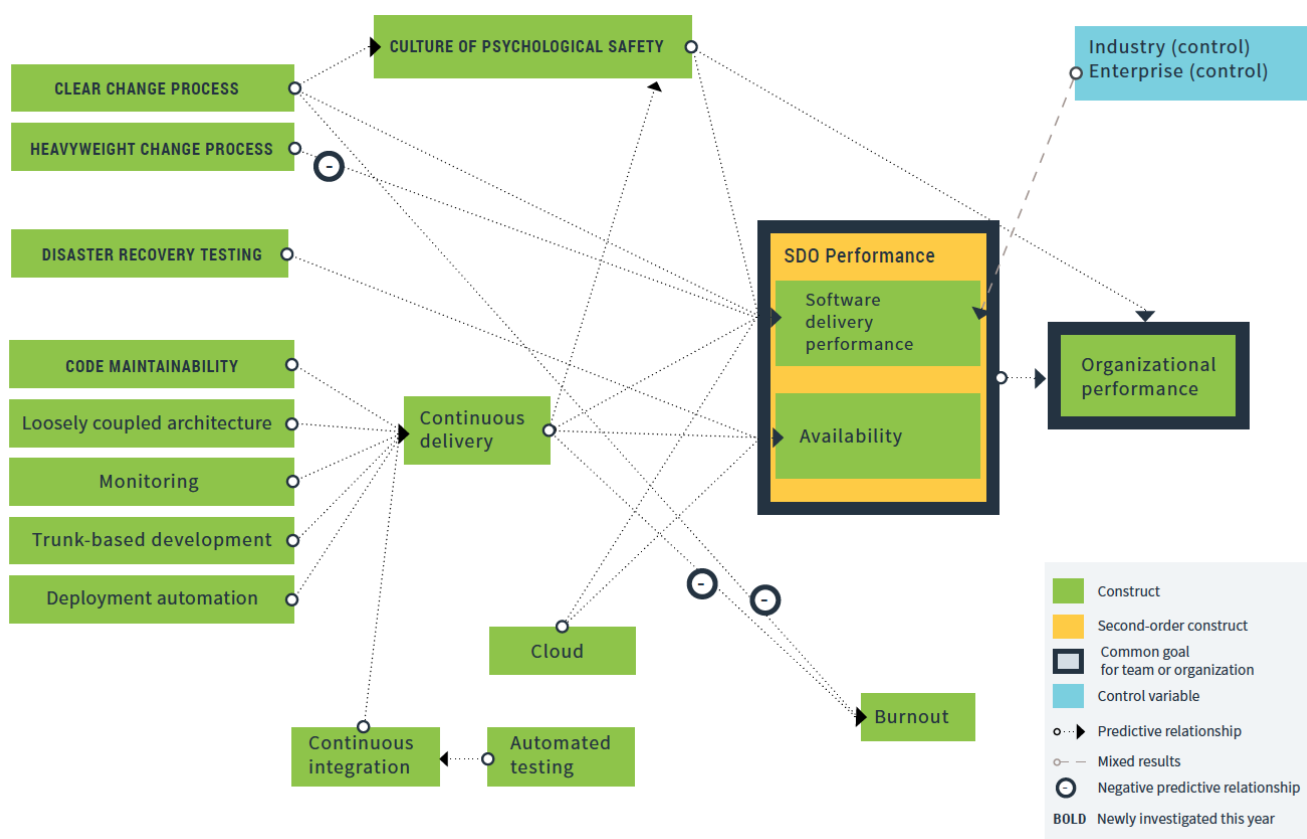
但是，从过往的报告可以看出，每一年关于模型的进化是整个报告的核心内容，报告也在不断覆盖新的领域，试图更加全面地揭示影响软件开发效能的核心要素。在实践 DevOps 的

时候，你可以参考这个能力模型，识别当前的瓶颈点，在遇到拿捏不准的决策时，也可以参考模型中要素的影响关系。

比如，公司内部经常会争论是否需要更加严格的审批流程，希望借助严格的审批流程，促使软件交付更加有序和可靠。很多系统和需求在提出的时候，都是以这种思想为指导的。我一直对这种流程的有效性抱有怀疑，加入更多的领导审批环节，除了出问题的时候大家一起“背锅”之外，并没有带来什么增值活动。

在今年的模型中，这种观点得到了印证。**重流程管控不利于软件交付效能的提升，轻流程管控也不会影响软件交付质量，关键要看公司是否选择一种“更好”的做法来实现管控的目的。**

## SOFTWARE DELIVERY & OPERATIONAL PERFORMANCE



最后，我们要重点关注**实践**。

在实施 DevOps 的时候，经常会有这样的困扰：道理都懂，却仍然做不好 DevOps。所以，DevOps 落地的核心无外乎实践和文化，而实践又是看得见摸得着的，这一点当然值



得关注。在状态报告中，有很大篇幅都在介绍实践部分，这些实践都是在大多数公司实施总结出来的，并且得到了实际的验证，具有很强的参考性。

比如，今年的报告重点介绍了技术债务、灾难恢复测试和变更管理流程这几个方面的实践，这些都是企业实施 DevOps 时的必经之路。

比如灾难恢复测试，很多公司都有非常详尽的文档，但是如果找他们要操作记录，他们却又很难拿出来。

我之前就见过一家国内 Top 的公司，说是在做关键数据的备份，但实际去看才发现，这个备份任务已经很长时间处于失败状态了。

如果有定期的灾难恢复测试，类似的这种问题是一定可以发现的。而**往往在灾难发生的时候，才能体现一家公司的工程能力水平。**

比如，Netflix 正是因为混沌工程，才没有受到 AWS 云服务 down 机的影响，这和日常的演练是密不可分的。

从 2014 年至今的 DevOps 状态报告的中英文版本，我已经收集并整理好了，你可以点击 [🔗网盘](#) 链接获取，提取码是 mgl1。

## 几本好书

讲完了报告，接下来，我再给你推荐几本好书。

### 1. 《[🔗持续交付](#)》&《[🔗持续交付 2.0](#)》

谈到 DevOps 里面的工程实践，持续交付可以说是软件工程实践的终极目标。对于在企业内部推进 DevOps 工程能力建设的人来说，这两本书可以说是案头必备，常看常新。

对我自己来说，因为 2011 年机缘巧合地拿到了第一版第一次印刷的《持续交付》这本书，我的职业生涯彻底改变了。因为我第一次发现，原来软件交付领域有这么多门道。帮助组织提升交付效率这个事情，真是大有可为。

《持续交付》围绕着软件交付的原则，给出了一系列的思想、方法和实践，核心在于：**以一种可持续的方式，安全快速地把你的变更（特性、配置、缺陷、试验），交付到生产环境上，让用户使用。**你可以参考一下软件交付的 8 大原则。

为软件交付创建一个可重复且可靠的过程

将几乎所有事情自动化

将一切纳入版本控制

频繁地做痛苦的事情

内建质量

DONE 意味着已发布

交付过程是每个成员的责任

持续改进

很多人都有《持续交付》这本书，但我敢打赌，真正能沉下心来把这本书看透的人并不多，因为这本书里面通篇都是文字，而且有些难懂，如果没有相关的实践背景，基本上就跟看天书差不多了。

所以，通读《持续交付》并不是一个好的选择，我建议你**尽量带着问题有选择性地去读。**

到了《持续交付 2.0》，乔梁老师创新性地将精益创业的思想 and 《持续交付》结合起来，更加强调整 IT 和业务间的快速闭环，也更加适应当今 DevOps 的发展潮流。

另外，乔梁老师的文笔更加流畅，读起来更加轻松，他会结合案例进行说明，对于实操的指导性也更强。毫无疑问，他是国内软件工程领域的集大成者。

如果你对软件开发流程的工程实践不太了解，你可以读一读这两本书。

当然，对于开发、测试、运维人员这些软件交付过程中必不可少的角色来说，也可以用来拓展知识领域。

**2. 《🔗精益创业》&《🔗Scrum 精髓》&《🔗精益产品开发》&《🔗精益开发与看板方法》**

关于管理实践和精益方面，我给你推荐 4 本书。

《精益创业》提出的 MVP（最小可行产品）思想已经被很多的企业奉为圭臬。它的核心是，只有经过真实市场和用户的验证，想法才是真正有效的，产品需要在不断的验证和反馈过程中持续学习，持续迭代，而不是试图一步到位，耗尽所有资源，从而失去了回旋的余地。

《Scrum 精髓》适合于使用 Scrum 框架的敏捷团队学习和实践，以避免 Scrum 实施过程中形似而不神似的问题。同时，这也是立志成为 Scrum Master 的同学的红宝书。

《精益产品开发》是何勉老师在 2017 年出版的一本基于精益思想和精益看板方法的著作。在精益软件开发领域，这本书和李智桦老师的《精益看板方法》，都是看一本就够了的好书。

这几本书比较适合想要了解敏捷，或者是在实际工作中践行敏捷开发方法的同学阅读。另外，精益思想可以说是 DevOps 的理论源泉，很多的文化导向，以及持续改进类工作都跟精益思想有密切的关系。

### 3. 《🔗 DevOps 实践指南》& 《🔗 Accelerate：加速》

如果你想了解 DevOps 的全貌以及核心理论体系和实践，《DevOps 实践指南》和《Accelerate：加速》就是最好的选择了。这两本书的作者都是 DevOps 行业内的领军人物，作为 Thought Leader，他们引领的 DevOps 的体系在不断向前演进。

其中，《DevOps 实践指南》，也就是俗称的 Handbook，重点介绍了 DevOps 实践的三步工作法，还包含了大量 DevOps 实施过程中的参考案例。而《Accelerate：加速》的作者就是 DevOps 状态报告的作者。他在这本书中揭示了状态报告背后的科学方法，并提出了 DevOps 能力成长模型，以帮助你全面提升软件交付能力。

### 4. 《🔗 凤凰项目》& 《🔗 人月神话》& 《🔗 目标》

最后，我想再推荐三本小说，这也是我读过的非常耐看的几本书了。



其中，《凤凰项目》提出的 DevOps 三步工作法和《DevOps 实践指南》一脉相承；《人月神话》是 IT 行业非常经典的图书，畅销 40 余年；《目标》则是约束理论的提出者高德拉特的经典著作，他所提出的改进五步法构成了现代持续改进的基础。

## 大会，网站和博客

当然，报告和书只是 DevOps 资源中的一小部分，还有很多信息来源于大会、网站和博客，我挑选了一些优质资源，分享给你。

- 🔗 [DEOS](#)：DevOps 国际峰会，以案例总结著称；
- 🔗 [DevOpsDays](#)：大名鼎鼎的 DevOpsDays 社区；
- 🔗 [TheNewStack](#)：综合性网站，盛产高质量的电子书；
- 🔗 [DevOps.com](#)：综合性网站；
- 🔗 [DZone](#)：综合性网站，盛产高质量的电子书；
- 🔗 [Azure DevOps](#)：综合性网站，盛产高质量的电子书；
- 🔗 [Martin Fowler](#)：Martin Fowler 的博客；
- 🔗 [CloudBees Devops](#)：Jenkins 背后的公司的博客。

在这些资源中，有一些值得你重点关注一下。

比如，Gene Kim 发起的 DOES（DevOps 企业峰会）就是获取实践案例的绝佳场地；而 DZone 和 NewStack 经常会推出免费的电子书和报告，也值得订阅；Martin Fowler 的博客，每一篇内容都是精品，对于很多技术细节可以说是起到了正本清源的作用，值得好好品味。

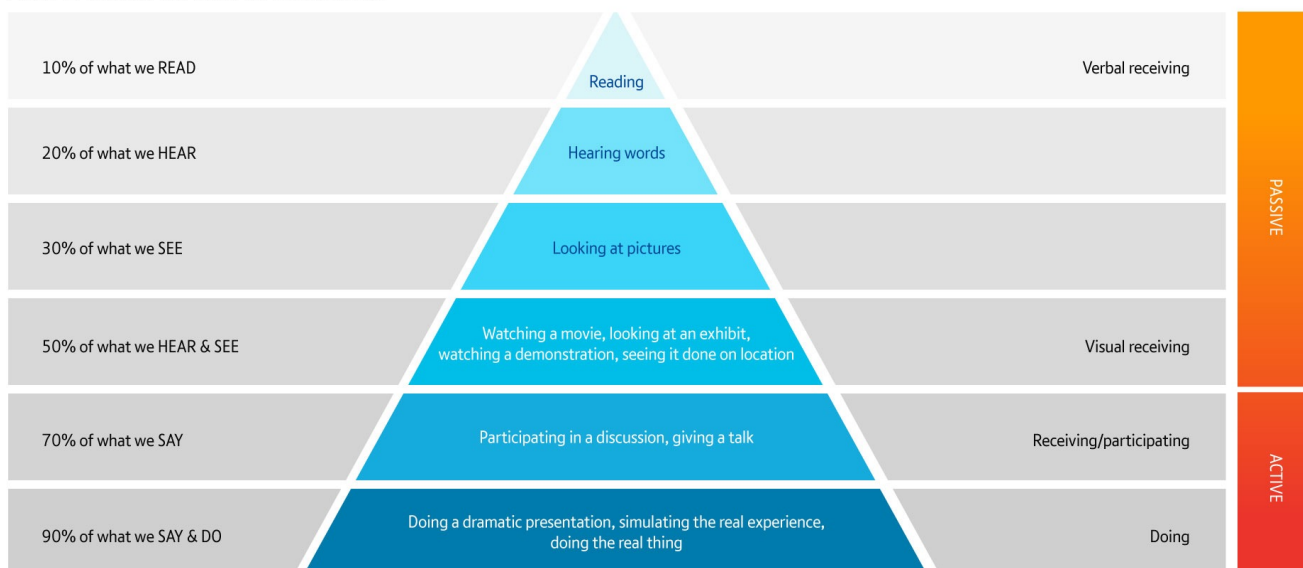
说了这么多，最后我还想再花一点点时间，跟你聊聊学习这个事情。我跟你分享一幅美国学者爱德加·戴尔提出的学习金字塔模型图，这个模型也是目前比较有参考性的模型之一。

# CONE OF LEARNING

(EDGAR DALE)

After 2 weeks, we tend to remember:

Nature of involvement



图片来源: <https://www.businessdirect.bt.com/>

在这个模型中，学习的方式分为两种，一种是主动学习，一种是被动学习。其实，无论是读书，看视频，还是听专栏，都属于是被动式的学习，最终收获的知识可能只有输入信息的一半儿，这还是在记性比较好的情况下。大多数时候，看得越多，忘得越多，这并不是一种特别有效的学习方式。

实际上，对于 DevOps 这种理念实践、技术文化、硬技能、软实力交织在一起的内容来说，主动学习的方式是不可或缺的，比如**案例讨论，线下交流，在实践中学习**等。

所以，希望你能多思考，多总结，结合工作中的实际问题，摸索着给出答案，并积极分享，跟大家讨论。只有主动思考，才能消化吸收，最终总结沉淀出一套自己的 DevOps 体系认知。

## 总结讨论

好了，今天我跟你聊了 DevOps 的学习资料，包括状态报告、书籍和大会、网站、博客。不过，对于 DevOps 来说，这些也仅仅是点到为止。

我想请你来聊一聊，你自己在学习和实践 DevOps 的过程中，有没有私藏的干货和渠道呢？如果有的话，希望你可以分享出来，我们共建一个 DevOps 相关的资源库，并在

GitHub 上进行开源维护，从而帮助更多人了解和学习 DevOps。

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，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：成为DevOps工程师的必备技能（下）

下一篇 期中总结：3个典型问题答疑及如何高效学习

## 精选留言 (4)

写留言



leslie

2019-11-28

不错：适当补充一下要买书了。目前觉得结合项目管理去思考，其实很多事情就通了。



工画师

2019-11-28

老师推荐的书中，有几本已经阅读过了。不过我有一个实际的感受不知道对不对？个人觉得除了DevOps和敏捷相关知识体系，还需要把领域驱动设计一起结合进来研究和思考，这三者在组织结构和流程设计中相辅相成且相互牵扯。

展开 ▾



**雷霹雳的爸爸**

2019-11-28

我又一次发现我属于屯书不看型...

展开 ▾



**Robert小七**

2019-11-28

老师是否推荐考devops master

展开 ▾

