

THE SCIENCE OF LEAN SOFTWARE AND DEVOPS

# 加速

高效能技术组织的构建与规模化



Nicole Forsgren, PhD  
Jez Humble, *and* Gene Kim

*with forewords by Martin Fowler and Courtney Kissler  
and a case study contributed by Steve Bell and Karen Whitley Bell*

# 目录

版权说明	1
推荐序一	2
推荐序二	4
快速指引：驱动改进的能力	5
前言	6
I: 发现	7
1. 加速	8
2. 度量效能	9
3. 乐于度量和拥抱变化的文化	10
4. 技术实践	11
5. 架构	12
6. 将信息安全集成到 DevOps 生命周期环中	13
7. 软件管理实践	14
8. 产品开发	15
9. 可持续工作	16
10. 员工满意度、认同感和共同约定	17
11. 领导和管理者	18
II: 研究	19
12. 本书背后的科学原理	20
13. 心理测验学介绍	21
14. 为什么要度量	22
15. 项目数据	23
III: 转换	24
16. 高效领导力和管理	25
总结	26
附录	27
附录 A: 快速指引：驱动改进的技能	28
附录 B: The Stats	29
附录 C: 研究中用到的统计方法	30
致谢	31
参考文献	32
索引	33



## 版权说明

Copyright © 2018 by Nicole Forsgren, Jez Humble, and Gene Kim.

© 2022 Translated by Dandi Wong.

## 推荐序一

几年前我读到一份报告称“我们确信更高的 IT 效能跟更好的商业表现具有相关性，它们都有助于提高产能、增加营收和扩大市场份额。”一般我读到这类报告时都是直接把它扔到垃圾桶里去，因为这种报告太多挂羊头卖狗肉的了。但当我看到这份名为《2014 年 DevOps 现状报告》的作者中有 Jez Humble 这个名字时犹豫了下，因为这货既是我同事又是我朋友，而他跟我一样很讨厌这种瞎扯。（当然有一说一我没扔进垃圾桶的另外一个原因是当时这报告是在我的 iPad 上读到的 2333。）

于是我给 Jez 发了封邮件，想知道这背后的缘由是什么。几周之后我就跟他和一个名叫 Nicole Forsgren 的人开了个电话会，这个 Nicole Forsgren 耐心地给我解答了诸多困惑。虽然我对他们用的那些方法不是很了解，但她滔滔不绝地阐述让我相信他们确实有在做这方面的分析与研究，而且要比我以往见过的、甚至比学术论文都要来得更靠谱。出于兴趣我就把之后的报告也都读了，但是一读发现问题大了。这些报告只阐述了他们这项工作的成果，却没有包含 Nicole Forsgren 在电话里给过我的那些解答。这极大地影响了它们的可信度，只会让人觉得这就是瞎猫撞上了死耗子，凑巧行得通而已。见识过报告的庐山真面目的我们几个最终说服了 Nicole、Jez 和 Gene，让他们以出书的形式将这些方法论分享出来。那一阵子我可有得等了。现在我很开心将它作为探究 IT 交付效能的一种途径推荐给各位——它可不只是一堆分析师把零散的经验拼凑在一起哦。

他们画的那些图看起来就很引人入胜。它们告诉你那些具备高效 IT 交付能力的组织如何只花费 1 小时的时间就可以把刚提交到主线的代码发布到线上去，而同样的工作在其他更小规模的组织里却往往要花费数月去完成。正因如此，他们可以在一天之内发布数次软件更新而不是每隔几个月发布一次，这为他们在市场探索、事件响应和比竞争对手更快地发布功能等方面都提供了便利。响应度的巨大提升并没有以牺牲稳定性为代价，尽管这些组织发现他们的各种升级会在一小部分低性能的关键节点上运行失败，但通常他们很快就可以修复。这些证据有力地反驳了关于 IT 效能只能在速度和稳定性中二选一的极端分化的观念——恰恰相反，速度是建立在稳定性之上的，良好的 IT 实践完全可以做到两全其美。

因此如你所期，我很高兴他们把这本书发行出来，反正不管三七二十一接下来几年里我都会推荐它（事实上我已经在自己的演讲中用过很多这其中的素材了）。但我还是要提醒几点：在解释他们的度量方法为什么可以为其客观数据奠定一个良好基础这件事上，他们做得很好；但这些度量方法也有主观认知的成分在，那我就会疑虑他们的抽样样本怎么就能代表更普遍的 IT 环境了。或许当有使用其他方法的其他团队作为对照组、并能验证他们的推理时，我就会对他们的结果信心倍增了。当然书中已经有了部分这方面的内容，比如 Google 在团队文化上的做法就有力地佐证了Westrum 组织文化<sup>[1]</sup>对于高效能软件团队的重要性。其他类似的例子也可以让我不至于陷入“他们的结论证实了我的很多观点”的确认偏见<sup>[2]</sup>陷阱（尽管我更多是在别人身上注意到它）。我们也应当谨记，这本书着眼点在于“交付”，即从代码提交到发布上线这一段，而非整个软件开发过程。

当然，以上只是本人的一些谬论，千万不要因此而干扰了您对本书主旨的理解。那些度量指标和在度量基础上的谨慎分析，已然为那些能有效提升 IT 组织交付能力的实践方法提供了一些最佳佐证。任何一个带领 IT 团队的人都该好好地看一看这些工作方式和方法并将其付诸实践。任何与 IT 团队（不管是内部的还是像我司这样第三方的）有协作关系的人，也都应该合理运用这些实践方法并形成持续改进的工作流。Forsgren、Humble 和 Kim 绘制了一张 2017 年高效能 IT 的呈现图，IT 实践者们可以以此为参照，加入到高效能行列中来。

—— Martin Fowler  
ThoughtWorks 首席科学家

[1] 见《DevOps文化：Westrum 组织文化》，<https://cloud.google.com/architecture/devops/devops-culture-westrum-organizational-culture>。

[2] 确认偏见（Confirmation bias）：也称肯定偏误、验证性偏见，指一个人无论合乎事实与否，偏好支持自己的成见、猜想的倾向。通俗讲，我们总是趋向于看见我们想看见的，相信我们愿意相信的。

## 推荐序二

# 快速指引：驱动改进的能力



## 前言

## I: 发现

# 章节 1. 加速

## 章节 2. 度量效能

### 章节 3. 乐于度量和拥抱变化的文化

## 章节 4. 技术实践

# 章节 5. 架构

## 章节 6. 将信息安全集成到 DevOps 生命周期环中



# 章节 7. 软件管理实践

# 章节 8. 产品开发

# 章节 9. 可持续工作

# 章节 10. 员工满意度、认同感和共同约定

# 章节 11. 领导和管理者

## II: 研究

# 章节 12. 本书背后的科学原理

# 章节 13. 心理测验学介绍



# 章节 14. 为什么要度量

# 章节 15. 项目数据

### III: 转换

# 章节 16. 高效领导力和管理

## 总结

## 附录

## 附录 A: 快速指引：驱动改进的技能

# 附录 B: The Stats



# 附录 C: 研究中用到的统计方法

致谢

参考文献

# 索引

## 关于作者