# SaaS 软件工程

## 云计算时代的敏捷开发

Armando Fox　David Patterson 著 / 徐葳　曹锐创 译

清华大学出版社

# SaaS 软件工程

## 云计算时代的敏捷开发

Armando Fox　David Patterson 著 / 徐葳　曹锐创 译

清华大学出版社
北 京

## 内 容 简 介

本书介绍了21世纪以来软件工程的两个新的发展方向——软件即服务和敏捷开发。本书第一部分介绍了软件即服务及其相关的编程环境，包括Ruby、Rails和JavaScript等内容；第二部分介绍了现代软件开发并强调了敏捷开发相关技术，包括行为驱动和测试驱动的设计与开发，遗留代码处理，项目团队管理，设计模式，软件架构等内容，并给出了许多有用的建议。

本书的作者Armando Fox和David Patterson不仅在计算机科学领域建树颇丰，而且在软件工程方面也有着丰富的教学和研究实践经验。他们将软件即服务和敏捷开发的内容与具体的实例相结合，完成了这本内容丰富且易于学习的书。我们希望本书能够帮助你通过实践学到软件开发的知识与技能，并能开发出漂亮而持久的软件。

**本书封面贴有清华大学出版社防伪标签，无标签者不得销售**
**版权所有，侵权必究. 侵权举报电话：010-62782989　13701121933**

**图书在版编目(CIP)数据**

SaaS软件工程: 云计算时代的敏捷开发/(美)福克斯(Fox, A.), (美)帕特森(Patterson, D.)著; 徐葳, 曹锐创译. --北京：清华大学出版社，2015

书名原文: Engineering long-lasting software: an agile approach using SaaS and cloud computing

ISBN 978-7-302-39967-4

Ⅰ. ①S…　Ⅱ. ①福…②帕…③徐…④曹…　Ⅲ. ①计算机网络–程序设计　Ⅳ. ①TP393.09

中国版本图书馆CIP数据核字(2015)第090887号

**责任编辑：**付弘宇　李　晔
**封面设计：**常雪影
**责任校对：**白　蕾
**责任印制：**李红英
**出版发行：**清华大学出版社
**网　　址：**http://www.tup.com.cn, http://www.wqbook.com
**地　　址：**北京清华大学学研大厦A座　　**邮　　编：**100084
**社 总 机：**010-62770175　　**邮　　购：**010-62786544
**投稿与读者服务：**010-62776969, c-service@tup.tsinghua.edu.cn
**质量反馈：**010-62772015, zhiliang@tup.tsinghua.edu.cn
**印 装 者：**清华大学印刷厂
**经　　销：**全国新华书店
**开　　本：**203mm×260mm　　**印　张：**30.25　　**字　数：**600千字
**版　　次：**2015年6月第1版　　**印　次：**2015年6月第1次印刷
**印　　数：**1～2000
**定　　价：**69.80元

产品编号：053799-01

# 译者序

在上一次清华园里绿树成荫的时候，我们拿到了这本书的原稿，开始着手将这部优秀的作品介绍给广大读者。这段时间里，园子里的银杏叶黄了又落了，在不知不觉间枝头上已擎起饱满的嫩芽，只待东风唤得春回大地的时候重新将绚烂的生命力绽放。而我们的书稿也终于得以付梓，也算是对这一学年以来在文字间的辛苦耕作有了一个交待。

本书封面图中的塞戈维亚大渡槽历经千年风雨，到如今依然屹立不倒，甚至还发挥着它原本的作用。而在更早的战国时期由李冰在成都设计建造的都江堰，至今仍为沃野千里的天府之国带来福祉。软件开发在经历了简陋而效率低下的发展阶段后，进入了软件工程阶段，同样也需要像这些建筑工程一样的漂亮而持久的设计。

尤其是进入21世纪以来，软件即服务和敏捷开发成为了软件工程的两个新的发展方向。然而，对软件工程的从业者来说，想要找到合适的学习资料却似乎并不容易。软件即服务方面的书籍太过繁多，读者要从中分辨并找到适合自己的读物殊为不易。而对软件工程以及敏捷开发来说，情况可能更糟糕：我们很难找到令大家满意的书籍，有的太过冗长，有的局限于定性分析，有的偏于理论而没有告诉读者该如何做。如果将范围缩小到中文书籍，情况只能是更加艰难。

本书的原作者结合自己的研究教学和实践经历，将软件即服务和敏捷开发的内容结合到具体的示例中去，最终形成了这本易于学习而又内容丰富的书。书中包含了大量的真实可运行代码和阐述，还有视频帮助大家学习。初次接触到这本书的时候，书中的内容让人耳目一新。结合本人从事软件工程相关工作的经验，本书讲述的内容会给有志于开发SaaS软件的人们带来巨大帮助。我很高兴这么快就有了机会亲自来组织翻译这本书。

我要首先感谢本书的原作者，也是我的博士导师，Armando Fox教授和David Patterson教授。两位作者不仅在计算机科学研究领域建树颇丰，而且都致力于提升计算机科学的教学水平。本书的英文原版不仅内容丰富深刻，而且语言生动优美。这也使我们在翻译的过程中不得不反复忖度，生怕失之偏颇。好在最后的结果也还差强人意。

本书的翻译是集体劳动的成果。本书中文版翻译过程中，得到了许多人的支持与付出，我们感谢所有为本书的编写和翻译付出辛苦努力的人。

我的博士生曹锐创负责了整个翻译团队的组织工作，并对本书的翻译结果进行了全面校对。感谢交叉信息研究院的李浒同学、李古月同学、陈天佳同学，他们利用课余时间共同完成了本书的大部分翻译工作。

感谢清华大学交叉信息研究院的殷和政同学、宫澄光同学、王康宁同学，以及清华大学软件学院的马洪天同学、张桂菊同学、江梦琪同学、陈欣洁同学、蔡历明同学、卢悦婷同学。他们利用自己的课余时间，协助完成了本书的主要校对工作。

感谢清华大学交叉信息研究院、教务处、在线教育办公室，支持我用本书和本书配套的在线课程进行“云计算与软件工程”课程混合式教学的试点。感谢本门课程的助教贾永正、魏达、刘佳倩。同时也要感谢选修此门课程的同学，他们在课程中使用本书，并协助指出了翻译中的不少错误。祝他们通过这门课程和这本书学到更多有用的东西。

祝愿国内所有的读者都能从本书中发现价值，获得提高，同时祝本书也能成为一个漂亮而持久的设计示例。由于译者的专业技术及语言水平有限，翻译时间紧，加上翻译过程中发生的疏漏，书中难免会有我们没有察觉到的不足甚至是错误之处，在此希望广大读者能够谅解。同时，我们热切希望读者在阅读的过程中，指出书中的谬误，对本书提出宝贵的意见。

徐　葳

2015年于清华园

## 大规模开放在线课程（MOOC）

学堂在线是清华大学发起的大规模开放在线教育平台，提供了大量免费的国内外优秀课程。

xuetangX.com 学堂在线

本书的作者还同步制作了两门大规模开放在线课程（MOOC）。目前这两门课程在全球范围内已经有数十万人选课，数以万计的学生获得了课程合格证书。课程的每一讲都与本书的一个章节对应。

我们在2013年春季学期尝试了在北京几所高效推广这两门在线课程，秋季学期在清华大学开设了基于这两门在线课程的混合式教学试点，都取得了很好的效果。2014年暑期开始，我们通过学堂在线平台（http://www.xuetangX.com[1]）引进了本书配套课程的MOOC版本，采用原书作者的授课录像，并配以中文字幕，免费向全国学生提供，目前已有数千人选修。希望届时通过作者原汁原味的精彩讲解，以及在线提交并且自动批改的编程作业和小测验，能够真实还原美国顶尖大学的课程体验。学堂在线平台提供了本地中文助教和讨论区的支持，加以本书中文版的配合，相信会使读者获得更好的学习效果。



试读结束，需要全本PDF请购买 www.ertongbook.com

## 对“软件工程即服务”课程和本教材的褒奖

很荣幸看到学生发信息强调产生真正有用的软件的重要性。我也赞赏强调在过程的早期获得结果的看法，没有什么比这更能鼓舞学生的士气并激励学生开展活动的了。

——Frederick P. Brooks, Jr.，图灵奖获得者及《人月神话》作者

和其他任何我见过的毕业生相比，我可能会更倾向于选择这个项目的毕业生。

——Brad Green，Google Inc.工程部经理

C3 Energy的很多软件工程师不断报告称，这本书和它配套的网络课程让他们能够迅速地在SaaS开发上达到熟练的程度。我向那些希望开发或改善其SaaS编程技巧的人推荐这本独特的书和这门独特的课程。

——Thomas M. Siebel，C3 Energy创始人及首席执行官
Siebel Systems（顶尖的客户关系管理软件公司）前首席执行官

广泛而深入地覆盖了你在SaaS入门中所需要的所有内容。

——Vicente Cuellar，Wave Crafters Inc.首席执行官

这本书填补了我在云计算方面的知识缺口，它的课程也很容易学习。也许最令人兴奋的部分就是编写一个云应用程序，并上传部署到Heroku。

——Peter Englmaier，瑞士苏黎世大学

一本优秀的学习Ruby、Rails和测试驱动方法的入门书籍。此书完美介绍了现代Web开发，深入涉猎探索了其基本原理，是现代Web开发最好的介绍，它应该是一名新晋工程师的必备书籍。

——Stuart Corbishley，南非Clue Technologies/CloudSeed

一本令人振奋的书，让你在短短数天内就可以逐步构建运行SaaS应用。本书的视频和代码实例部分是无价的。这是一门非常实用的敏捷软件开发方法。你不需要知道它，你甚至可以在不知道正在做什么的过程中学到软件工程的技术。

——Rakhi Saxena，印度德里大学助理教授

作者完成了对任何现代高级软件工程课程的开端来说，都非常受欢迎的一个理论与实践的融合。一方面，它囊括了关键的软件工程基础，包括开发流程、需求工程、软件测试、软件体系结构、配置管理、实现以及部署。另一方面，它们传达了以Ruby/Rails为中心建立一个“真实世界”的方法，包括其丰富的敏捷生态系统、测试驱动和行为驱动开发工具与技术，并指明了在云端部署运行优质软件的康庄大道。我已经在本科生高级软件工程课程里成功地使用了这本书的Beta版本，它几乎完美地充实了我的课程和项目。

——Ingolf Krueger，加州大学圣地亚哥分校教授

一本真正好的介绍实用敏捷开发的书。所有你需要的东西都汇聚在这本书里，其中还有很多实用的例子。

——Dmitrij Savicev，瑞士Sungard Front Arena

## 译者简介

**徐葳**博士现任清华大学交叉信息研究院助理教授，博士生导师，中组部“青年千人计划”特聘专家。博士毕业于美国加州大学伯克利分校，是本书的两位作者David Patterson和Armando Fox联合指导的博士生。他主要从事超大规模分布式系统以及大数据处理方向的跨领域研究工作，相关研究工作发表在SOSP，ICML，ICDM等顶级国际会议上。教学中他主要负责计算机科学实验班的“软件工程”（使用本书作为教材）和“操作系统”两门全英文授课的课程。徐葳博士具有丰富的大型分布式系统工程经验和项目管理经验。加入清华大学前曾在Google总部工作，负责系统日志和监控系统的研发工作。他在清华大学和美国宾夕法尼亚大学完成了本科学业，并在加州大学伯克利分校获得硕士和博士学位。

**曹锐创**现为清华大学交叉信息研究院博士研究生，徐葳教授的学生。目前主要从事云计算与大数据方面的研究。2012年毕业于清华大学计算机科学实验班（“姚班”）并免试获得博士生资格。在校期间多次获得学业和综合优秀奖励，两次赴香港中文大学访问交流，并曾在惠普中国研究院从事智能打印方面的研究实习。

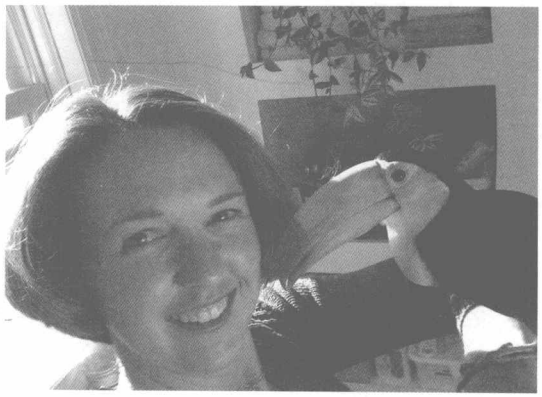
## 关于作者

**Armando Fox**是加州大学伯克利分校计算机科学教授，同时也是该校MOOC实验室的学术顾问。他在斯坦福大学期间获得了由斯坦福大学学生会、女性工程师协会以及国家工程荣誉协会（Tau Beta Pi）颁发的教学与指导奖。他获选2003年的“科学美国人50强”（Scientific American 50），同时还是NSF CAREER奖和美国国家工程院Gilbreth Lectureship奖的获得者。在此之前他参与设计了英特尔奔腾微处理器并创建了一个公司，成功地将他在伯克利进行的移动计算方面的论文研究商业化，其中就包括世界上第一个运行在移动设备上的图形Web浏览器（Palm Pilot上的Top Gun Wingman）。他在麻省理工学院和伊利诺伊大学分别获得了电气工程和计算机科学学位，并获得ACM杰出科学家称号。他也是一位经过古典音乐训练的音乐家和自由音乐总监，以及一个移民到旧金山的同时掌握两种语言和文化（古巴裔美国人）的纽约人。

**David Patterson**是加州大学伯克利分校计算机科学方向的Pardee讲席教授。过去他曾担任伯克利计算机部门的主席、CRA的主席以及美国计算机协会（ACM）主席。他最著名的研究项目是精简指令集计算机（RISC）、磁盘冗余阵列（RAID）以及网络工作站（NOW）。这些研究产生了多篇论文、6本书以及超过35项荣誉，包括入选美国工程院和美国科学院院士，以及硅谷工程名人堂，还有计算机历史博物馆、ACM、IEEE、美国艺术与科学学会（AAAS）、美国科学促进会（AAAS）的会士。他的教学奖项包括卓越教学奖（加州大学伯克利分校）、Karlstrom杰出教育家奖（ACM）、穆里根教育奖章（IEEE）以及本科教学奖（IEEE）。他在加州大学洛杉矶分校获得了所有的学位，学校授予了他杰出工程学术校友奖。他在加州长大，在业余生活中，他和两个成年的儿子共同参加体育赛事，包括每周的足球游戏、年度慈善自行车赛以及短跑三项全能运动，还偶尔参加举重比赛。

## 致谢

Armando Fox将这本书献给他的妻子，也是他最好的朋友Tonia。在他写作的时候，她的支持让这本书与众不同。还有Pogo，这本书的大部分是在它的细心监督下写出来的，而它充满生气的精神也将永远留在我们的家里，留在我们的心中。



David Patterson把这本书献给他的父母和他们所有的后代：

——给我的父亲David，从他身上我继承了创造性、崇尚运动还有为正义而战的勇气；

——给我的母亲Lucie，从她身上我继承了才智、乐观和性情；

——给我们的儿子David和Michael，他们是我的朋友、运动伙伴，并鼓励我成为一个好男人；

——给我们的儿媳妇Heather和Zackary，她们是我们的孙子孙女们聪明、有趣且有爱心的母亲；

——给我们的孙子孙女Andrew、Grace和Owyn，他们给了我们延续生命的机会（并且帮助营销这本书）；

——给我的弟弟妹妹Linda、Don和Sue，他们给了我第一次教课的机会；

——给他们的后代，他们让Patterson家族变得更大，并友好快乐地相处着；

——给我美丽且善解人意的妻子Linda，她是我最好的朋友，是我生命的挚爱。

# 前言

如果你想造一条船，你不能只是叫人来捡木头，分派工作，然后不停地下命令。相反的，你应该教他们去向往那广阔无尽的海洋。

——Antoine de Saint-Exupéry，*Citadelle*，1948

## 欢迎！

2000年以来的两个重大技术进步组成了本书的两个部分。第一部分讲解了**软件即服务（Software as a Service，SaaS）**，SaaS正在给软件工业带来一场革命。SaaS在云端运行单一的软件副本，并满足百万级用户的不同的需求。与之对应，传统的软件要求客户在自己的计算机上安装这一软件的百万个副本之一。

SaaS的用户和开发人员的激情带来了新的、高效的SaaS软件开发框架。我们在本书中使用**Ruby on Rails（RoR）**这一架构。RoR这一架构被广泛认为是开发SaaS的最好工具。当然也有许多使用其他语言的SaaS框架，例如Python/Django、JavaScript/Jasm ine以及Java/Enterprise Java Beans。

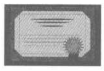
现在的问题在于，什么样的软件开发方法最适合SaaS。因为SaaS软件只有一个副本，并且运行在一个严格受控的环境中，我们可以很容易地以增量的方式增加功能，因此SaaS软件进化的速度要远快于传统的软件。因此，我们需要一个能够将代码改动作为常态，而不是非常态的软件开发过程。

工业界经常抱怨软件工程教育的薄弱。我们同很多顶尖软件公司，例如亚马逊、eBay、Facebook、谷歌和微软的代表沟通过，我们非常震惊地发现，这些公司一致认为，第一要务是让学生了解如何改善缺乏文档的遗留代码。这些公司的其他要求，按照重要性排列，分别为：让测试成为软件中的一等公民，与非技术背景的客户一起工作，还有团队合作。与非技术背景的客户合作，以及团队合作等社交技能显然对软件开发者的职业发展是大有帮助的，但是问题是如何在本书中介绍它们。因此，我们需要一个软件工程方法，这个方法能够帮助处理遗留代码，强调测试，能够整合非技术用户，并且能更好地支持团队合作而非单干。

碰巧的是，与SaaS出现几乎相同的时间，一些开发者在2001年提出了**敏捷宣言（The Agile Manifesto）**。敏捷宣言是传统软件工程方法根本上的变革。敏捷的一

个基本信条是："响应改变重于遵循计划"，因此它与"计划文档"方法，例如瀑布模型、螺旋模型和RUP模型相比，更加适合快速演化的SaaS软件开发。敏捷的另一个基本信条是："与客户合作重于合约的谈判"。这条的直接结果是每周与非技术背景的客户交流。敏捷开发的两个基础是**行为驱动开发（behavior-driven development）**和**测试驱动开发（test-driven development）**。这两条意味着，测试是在代码之前完成的，因此在敏捷开发中测试真正成为了软件中的一等公民。敏捷开发中的**结对编程（pair programming）**和**Scrum**等理念，强调了与他人合作的重要性。正如我们将要看到的，敏捷开发的技术对于演进遗留代码也有重要的作用。

因此，本书后半部分通过开发并部署一个使用RoR的SaaS应用来讲述敏捷开发。另外，每章也介绍了计划文档方法的不同视角，例如需求、测试、管理和维护。这种对比让读者可以自己决定哪种方法更适合他们的SaaS或者是非SaaS的应用。



从授课教师的角度看，这种对比的方式让这本书适宜作为软件工程课程的教材。我们确保了这些材料能够满足2013年ACM/IEEE软件工程课程大纲的标准。事实上，章节后的习题中大约有45个直接来自大纲标准中的学习目标（我们制作了一个特殊的页边标志）。用另一种指标来看，大约40%的大纲要求可以直接从习题中获得，另外40%直接对应了章节中的内容。两者合计远超过了大纲中要求的，包括45%的知识点的要求。在配套的教师指南中，我们更深入地探讨了这一问题。

很多教师（包括我们自己）的经验表明，学生很喜欢在项目中学习和使用敏捷开发。敏捷开发是基于迭代的，计划周期很短，因此对于日程繁忙的本科生和一个快节奏的课程来说，是绝好的搭配。繁忙的学生会很自然地拖延项目的进展，而只是在期末快要交项目的时候连续熬几夜把项目对付掉。敏捷开发不仅能够防止学生这种对付课程项目的策略（因为我们在每次迭代后都会对学生进行评估），而且我们的经验表明学生会采取一种更负责任的态度，能够获得实在的进步。

学生能够遵照敏捷开发方法的一个重要原因是，本书中介绍的RoR工具使得敏捷开发变得更为容易。应用这一方法能够真正让学生更为容易地做好他们的项目，而学生学到的敏捷开发方法也能在需要时应用在非敏捷的项目中。

我们甚至展示了如何将敏捷方法用在非敏捷方法开发的遗留代码中，并且说明了敏捷方法不一定只是对写新代码才有效。

## 为什么我们要写一本新书？

作者如果对一些已有的书很满意，就不会去写一本新书。我们对一些已有书籍的不满意之处，各有不同。

对于第一部分来说，问题并不是关于SaaS的好书太少，而是因为太多了！图1和图2列出了我们参考的50本书中的24本。仅仅这24本书就有超过10000页！这么多的书可能会吓走初学者。因此，我们写作这本新书的目的之一就是将SaaS的相关内容，用一个相对较短的篇幅和相对便宜的价格，给出一个一致明了的介绍与综述。本书测试版的一个评论者说，这本书的第一部分没有新意。当然，如果您愿意花时间和金钱来阅读那么多其他的书，我们认为这个评论是合理的。



试读结束，需要全本PDF请购买 www.ertongbook.com

图 1　图中12本书包括了超过5000页的内容，本书的作者在写作本书的过程中阅读了超过50本图书，这些书中的大部分都在适当章节中“深入学习”部分列出的

图 2　另外12本包括了超过5000页内容的图书，这些书中的大部分都在适当章节中的“深入学习”部分列出

对于第二部分，目前只有少量软件工程的教材的选择，并且没有既简短又便宜的选择！我们看过的SaaS的书的书评都是很好的——在亚马逊上的书评一般都是4星以上——但是软件工程教材的书评可就不是这样了。两本最常用的软件工程教材的书评都在2星和3星之间，并且有许多很差的评语。

原因之一在于这些软件工程教材大都是对那些文献的冗长而定性的综述——在每个内容上仅仅列出了基于研究论文的一些方法的选择——而没有告诉读者任何具体如何在其中进行挑选的方法。另一个原因可能是这些教材的初版创作于SaaS和敏捷开发出现之前很久，导致现在很难将这些截然不同的新理念融入到过去的材料中。

这种情况似曾相识。本书作者之一在25年前对计算机体系结构的教材有着同样的感觉：这些书都是很厚，都是对于当时研究论文的定性的调研和综述，并且没有告诉读者任何关于如何选择这些实现方法的信息。更糟糕的是，那时的计算机体系结构领域已经有了巨大的、并且有争议的变革，这些变革并没有被包含在那些教材中。这种对当时的教材的不满，导致了作者之一和一个朋友共同写了一本全新的计算机体系结构的教材。

重复历史的成功经验，本书的第二部分与传统的软件工程教材截然不同。我们把敏捷过程当做一等公民来介绍，并且包含了具体的、可以动手实践的代码和工具的实例。这些实例让读者能够遵循敏捷过程，更容易地实现满足客户需求的产品。如上面提到的，第二部分的每章中也讨论了计划文档过程的视角，以帮助读者更好地认识敏捷开发的优劣，并认识到什么时候应该选择哪种过程。

我们在每部分的目标是，将一组广泛的内容用**单一的陈述**表达出来，并且给出具体的例子来帮助读者理解。我们可以让已经熟悉敏捷开发的读者略过第二部分而只读第一部分来学习SaaS，或者反之。如果你刚开始接触这些内容——或者你学习的是SaaS和敏捷开发之前的内容——那么你可以一箭双雕地学到SaaS和敏捷开发这两个新的、令人振奋的理念。这一极有针对性的策略使本书能够用传统图书的一半章节和页码，以及四分之一的价格，讲解SaaS和敏捷开发两个内容。

## 大规模开放在线课程（MOOC）

我们决定开始写书之后，把应邀在加州大学伯克利分校讲授的课程的第一部分做成了一个大规模开放式在线课程（MOOC）。这一课程目前可以在saas-class.org[2]看到。MOOC的开发影响了这本书的写作，反之亦然。MOOC中的视频片段几乎与本书的章节一一对应。事实上，正像MOOC倡导的碎片化教学那样，本书中的每一节后边都包括了两道很短的“自测”题，并包含了答案。

在2012年2月开始的MOOC中，多达25 000名学生观看了视频，其中10 000多名学生尝试提交了编程作业，3500名心理承受能力极强的学生最终完成了整个课程。由于这一MOOC很受欢迎，我们在EdX平台上再次开设了这门课，又有几千名学生参加（CS169.1和CS169.2两门课分别对应伯克利5星期的课程）。2012年有超过10 000名学生取得了课程证书。因此，本书获得了比我们预期多很多的测试读者。

因此，你应该考虑注册这一免费的在线课程。你可以在saas-class.org[3]了解更多关于这门课程的内容。

MOOC中的作业和考试都是和本书对应的。如果你只是想测试一下是否学懂了本书的内容，你可以在任何时间注册MOOC并提交作业和考试。这些作业和考试将会像正式参加MOOC课程的学生一样获得自动评分。如果你希望获得一个证书，你必须在特定的开课时间内注册并按时完成课程要求，且考试和作业成绩合格。然而，如果你不需要证书，你可以在任何时候观看视频或者完成作业和考试。

## 为什么用了那么多引述?

我们认为引述可以让本书读起来更有趣，而且引述是老人将他们的智慧传给下一代的方式，并且能够为软件工程设置好的文化上的标准。我们也希望读者能了解一些这一领域的历史。这就是为什么我们在每章开头以及章节中间，引用了很多图灵奖获得者的妙语。

## 勘误和辅助内容

从作者的角度看，电子书的好处是，我们可以在读者指出书中错误后，随时更新。我们会收集读者的勘误表，并且每年更新几次电子版图书。

如果你阅读的是纸质版图书，请按照传统的方式，在saasbook.info[4]上查找勘误表*。

## 教师参考材料

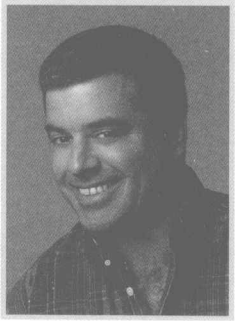
网站saas-class.org[5]包含了本书的许多相关资源，包括MOOC的小规模专用版本、可以自动批改的作业题、课堂幻灯片、习题集、教师手册以及更多内容。

*译者注：此处勘误表针对英文版。本章节部分其他链接也针对英文版书籍及课程，不再赘述。

## 本书的历史

本书是加州大学伯克利分校的一个研究项目[6]的副产品。在那个研究项目中，我们开发了一个能让下一代互联网服务开发变得更容易的工具。我们认为，年轻人更有可能开发出有意思的互联网服务。因此我们在2007年开设了一门关于SaaS的本科生课程，并在课上使用了敏捷方法。这门课的范围和流行程度每年都在增长，并且我们使用了很多Rails提供新的工具。在2007—2012年间，这门课的注册人数以摩尔定律的速度增加：35、50、75、115，直到165人。

Armando Fox

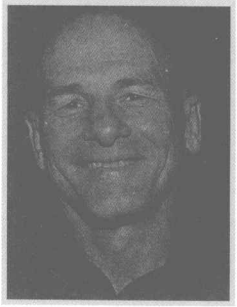


一个同事建议说，这些内容可以用于改进伯克利已经讲了无数年的软件工程课程，因此我们中的一个（Fox）就这样做了。我们获得了非常好的效果，因此另一个作者（Patterson）建议写一本教材来使其他大学也能从这一课程中受益。

我们坚信两点：电子书将会蓬勃发展，避免传统出版商的高额费用和漫长的出版时间将有可能实现。所以，在2011年3月，我们开始合作写这本书，使这些内容更为广泛地被人们所接触。重新思考电子书应该是什么样子一样令我们感到非常激动，因为电子书最先是纸质书的PDF文件。

我们和很多人讨论了本书的内容。我们参加了SIGCSE（计算机教育年会）、软件工程教育和培训会议以及计算机联合研究会议等，并在会议上向同行们进行了调研。根据教育家和工业界人士的观点，我们写出了能够阐明这些问题的提纲，并与当年6月开始合作。由于Fox在这些问题上有非常深入的了解，我们最初计划由Fox写大约三分之二的章节，Patterson写其他章节。我们在全书的组织结构上紧密合作，并且互相检查了对方的章节。最终我们都写了对方章节中的一部分，因此最终成书的章节的作者比预想的更为混杂。Fox撰写了以下章节：2、3、4、5、6、8、9、11、12、附录A，以及10.4~10.6节。Patterson撰写了以下几章：1、7、10、前言、后记、教师手册，以及以下章节中的计划文档视角部分：7.10、8.10、9.7、10.7、11.8、12.10。Fox编写了LaTeX的模板，使我们能够生成多种不同格式的电子和纸版图书。这一模板目前已经开放在github.com/armandofox/latex2ebook[7]，供有需要的作者获取使用。

David Patterson



我们在2012年春季学期为115名伯克利学生和上千名MOOC学生提供了本书的Alpha版本。写作被教学任务延误了一些。在2012年6月获得了学生的反馈后，我们继续加紧写作。Beta版图书在2012年秋季学期和2013年春季学期被其他大学采用。基于Beta版读者的反馈和对2013年版ACM/IEEE计算机协会的软件工程课程大纲的学习，我们在2013年2月推出了Beta版第二版。我们将在2013年秋季学期在伯克利和MOOC学生中再进行一次Beta版测试，最终推出经过严格测试过的版本。

请到如下网址查看勘误表[8]，并给出您对本书的建议。

我们提前为您在这一版本书中可能发现的错误致歉，并且希望听到您对如何能进一步改进本书提出的意见和建议。

Armando Fox和David Patterson
2013年5月于加州大学伯克利分校

## 免责声明

在本书中，我们使用了网站、工具、产品或者商标名称来作为真实的案例。除非特别说明，作者和这些网站、工具、产品并没有正式的关系，这些例子只是用于介绍，而非商业上的认可。任何商标都属于其所有者，这里的使用只是为了介绍。在任何可能的情况下，我们都专注于免费的和开源的软件和服务，使得学生可以动手实践这些实例，而无须花费更多的钱。

作者的观点是他们自己的，而不是他们的雇主的。

## 鸣谢

我们感谢我们在工业界的同事们对于本书及我们的课程提出的建议和反馈，特别是以下做出了杰出贡献的个人（以公司的字母顺序排列）Peter Vosshall，Amazon Web Services；Tony Ng，eBay；Tracy Bialik，Brad Green和Russ Rufer，Google Inc；Peter Van Hardenberg，Heroku；Jim Larus，Microsoft Research；Brian Cunnie，Edward Hieatt，Matthew Kocher，Jacob Maine，Ken Mayer和Rob Mee，Pivotal Labs；Jason Huggins，SauceLabs；还有Raffi Krikorian，Twitter。

我们感谢我们学术界的同仁，感谢他们对本书的方法和思想的反馈意见，特别是：Fred Brooks，北卡罗来纳大学教堂山分校；Marti Hearst和Paul Hilfinger，加州大学伯克利分校；Timothy Lethbridge，渥太华大学；John Ousterhout，斯坦福大学；还有Mary Shaw，卡内基梅隆大学。

我们由衷感谢以下审阅了特定章节的专家们：Danny Burkes，Pivotal Labs；Timothy Chou，斯坦福大学；Daniel Jackson，麻省理工学院；Jacob Maine，Pivotal Labs；John Ousterhout，斯坦福大学；还有Ellen Spertus，密尔斯学院。

感谢悉尼大学的Alan Fekete及时指出了我们应该参考2013年ACM/IEEE计算机学会软件工程教学大纲，使我们能在本书中包含这些内容。

我们特别感谢在他们的课堂中试用本书的测试者们：Daniel Jackson，麻省理工学院；Richard Ilson，北卡罗来纳大学教堂山分校；Samuel Joseph，夏威夷太平洋大学；Almudena Konrad，密尔斯学院；Ingolf Krueger，加州大学圣迭戈分校；Kristen Walcott-Justice，科罗拉多大学科罗拉多斯普林斯校区；还有Rose Williams，宾汉姆顿大学。

本书配套软件bookware的一部分是一组优秀的支持SaaS的第三方网站。他们帮助我们找到了合适的产品和服务，并且将这些产品和服务免费提供给学生在课堂上使用，他们还提供了如何在教学中使用这些工具的建议，我们感谢：John Dunham，SauceLabs；Maggie Johnson和Arjun Satyapal，Google Inc；Dana Le，Salesforce；James Lindenbaum，Heroku；Kami Lott和Chris Wanstrath，GitHub；Rob Mee，Pivotal Labs；Ann Merrihew，Kurt Messersmith，Marvin Theimer，Jinesh Varia和Matt Wood，Amazon Web Services；还有Juan Vargas和Jennifer Perret，Microsoft。

我们感谢研究生助教Kristal Curtis和Shoaib Kamil帮助我们在校内教学改革中做出的努力，最终促成了本书的创作。我们感谢研究生助教Michael Driscoll和Richard



试读结束，需要全本PDF请购买 www.ertongbook.com