The Pragmatic Programmers

Cucumber：行为驱动开发指南

The Cucumber Book

Behaviour-Driven Development for Testers and Developers

[英] Matt Wynne
[挪] Aslak Hellesøy 著

许晓斌 王江平 译

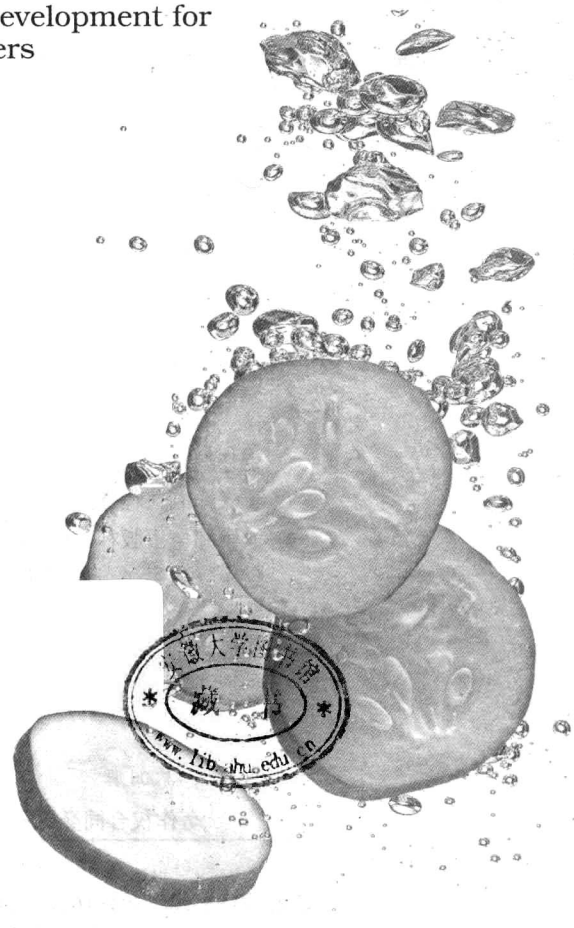
人民邮电出版社
POSTS & TELECOM PRESS

Cucumber：行为驱动开发指南

The Cucumber Book

Behaviour-Driven Development for Testers and Developers

[英] Matt Wynne
[挪] Aslak Hellesøy 著
许晓斌 王江平 译



人民邮电出版社
北京

图书在版编目（CIP）数据

Cucumber : 行为驱动开发指南 / (英) 韦恩 (Wynne,M.) , (挪) 赫勒索 (Hellesøy.A.) 著 ; 许晓斌 , 王江平译. -- 北京 : 人民邮电出版社, 2013.7
书名原文: The cucumber book:Behaviour-driven development for testers and developers
ISBN 978-7-115-31885-5

Ⅰ. ①C… Ⅱ. ①韦… ②赫… ③许… ④王… Ⅲ. ①软件－自动测试设备－指南 Ⅳ. ①TP311.56-62

中国版本图书馆CIP数据核字(2013)第094872号

版权声明

◆ 著　［英］Matt Wynne　［挪］Aslak Hellesøy
译　许晓斌　王江平
责任编辑　杨海玲
责任印制　程彦红　杨林杰

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京天宇星印刷厂印刷

◆ 开本：800×1000　1/16
印张：18
字数：386 千字　2013 年 7 月第 1 版
印数：1－3 000 册　2013 年 7 月北京第 1 次印刷

著作权合同登记号　图字：01-2012-4613 号

定价：59.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

内容提要

本书的两位作者是 Matt Wynne 和 Aslak Hellesøy，前者是 Cucumber 最有经验的用户和贡献者之一，后者是 Cucumber 的创始人，因此本书是一本权威指南，它会提供使用 Cucumber 所需的全部知识，让你和你的团队自信地开启 Cucumber 之旅。尽管 Cucumber 诞生于 Ruby 社区，但你可以用它测试几乎所有系统，从简单的 shell 或 Perl 脚本，到使用 PHP、Java 或任何其他平台编写的 Web 应用。

书中将展示如何用一组清晰、可执行且团队中任何人都能读懂的规格说明来表达用户那些天马行空的想法。你将学会如何将这些示例提供给 Cucumber，并让它指导你的开发过程。本书的第一部分会提供 Cucumber 入门所需的全部知识，引导你从 Cucumber 的核心特性起步，通过 Cucumber 的 Gherkin DSL，使用自然语言来描述客户想要的系统行为，然后带你编写 Ruby 代码来解释这些自然语言描述的规格说明并据此来验证应用的行为。第二部分将通过一个可以工作的例子来巩固学到的知识，同时学习一些更高级的 Cucumber 技术，还将学习如何测试异步系统和使用数据库的系统。第三部分提供了一些解决问题的方法，针对的都是作者曾帮助其他团队解决过的最困难、最常见的问题。基于这些模式和技术，你将学习如何使用 Capybara 和 Selenium 测试大量使用 Ajax 的 Web 应用，测试 REST Web 服务、Ruby on Rails 应用、命令行应用、遗留程序等。

Cucumber 有助于在软件团队中的技术人员和非技术人员之间架起沟通的桥梁。本书的内容既适合开发人员和测试人员阅读，也适合软件团队中的非技术读者阅读。

享受工具

——*The Cucumber Book* 译者序

软件开发领域有一个关于图书的“奥斯卡”大奖，自 1991 年起每年颁发一次，用以表彰每一年对该行业产生重大影响的图书，它就是 Jolt Awards，2012 年这一奖项颁发给了《实例化需求》一书。《实例化需求》是一本讲述如何构建正确产品的书，它提倡借助实例的力量来促进开发人员、测试人员、产品负责人以及客户之间的协作，并总结了一套非常清晰的关键过程模式。当然，要实现这套模式，我们就不得不面对许多非常现实的问题，例如实例该如何编写，怎样驱动实例让它们自动运行产品，再具体一点，如果我的产品是一组 Web 服务，我该如何编写代码运行一个业务场景以达到验证和沟通的目的？这正是工具发挥用武之地的地方，而 Cucumber 可能就是这方面最佳的工具。

一说到“工具”二字，可能你想把手里捧着的书放一边了，“工具而已嘛，没什么干货”。一直以来很多程序员对工具有一种误解，他们认为问题原理是更重要的，工具只是辅助手段，这方面最极端的例子就是，有些程序员喜欢用文本文档写段代码以证明，只要编程能力高了 IDE 那样的工具就并不是必需的。我无意挑战问题原理的重要性，对编程来说，掌握编程范式、语意和语法无疑是至关重要的，但我想说的是工具同样重要！C#的重构工具 ReSharper 能把一次重构的耗时从几十分钟降到几秒钟且更加安全；Unix 下的命令行工具 grep、sed 和 awk 能把数百行的文本分析 Java 代码降到数十行且运行更快；较之于 Subersion，Git 大大简化了分支及合并代码的代价，由此衍生出来的 GitHub 更是极大地促进了开源项目的协作。类似的例子不计其数，借助正确的工具，程序员的生产力能得到极大地提高。不仅如此，工具还能改进我们的思维模式并强化这种改进，例如，当你熟练使用了重构工具后，你脑子里的思维单元是“重命名”、“提取方法”、“内联变量”这样的东西，而不是原始的修改单词、修改/添加方法名、修改/添加方法参数、删除变量等语法级别的思维单元。

这样的现象是有理论依据的，斯坦福大学的生物学和神经学教授 Robert Sapolsky[①]就认为[②]：

---

① http://en.wikipedia.org/wiki/Robert_Sapolsky

② Aspiration Makes Us Human：http://www.scientificamerican.com/article.cfm?id=aspiration-makes-us-human。

人类的一个决定性的、至关重要的特征是：我们总是通过创造各种工具，来使自身变得更迅捷、更聪明、更长寿，努力地打破我们在自然进化中所面临的各种限制。

Cucumber（中译名：黄瓜）是一款卓越的 BDD 工具，它的前辈是早已在 Ruby 社区家喻户晓的 RSpec，而 RSpec 是基于 RBehave 改写的，RBehave 又是 Java 世界 BDD 框架 JBehave 的 Ruby 移植，因此 Cucumber 已经从它的前辈们身上汲取了大量的经验，再加上 Ruby 语言本身出众的灵活性，这让使用 Cucumber 编写实例及代码成为了一种享受！有时候觉得这就像是小时候放学口渴了来一根自家地里的黄瓜，清爽可口！

本书有两位作者，Matt Wynee 是 Cucumber 团队的核心成员，Aslak Hellesøy 更是 Cucumber 这个项目的创始人。全书读来系统性非常强，也有一定的深度。在这样一本实战型的书中，我经常能看到作者们对软件设计、敏捷方法等方面精彩的解释，深入时可以详细地讲述应对遗留代码的策略，浅出时能够耐心地介绍项目的目录结构，不在书名中加入“深入浅出”可真是浪费了。

我希望阅读本书能给你带来享受，当然更重要的是从阅读第 2 章起就打开电脑跟着作者的样例练习，学习一样全新的工具时，没有什么比练习更重要了。另外，从今天起多吃点黄瓜，可减肥、可抗肿瘤[①]，贴脸上还能美容……

在本书翻译到一半多的时候，我遇到了比较严重的视力问题，幸运的是好友王江平（@steedhorse）及时出手相助，完成了余下的部分，在此表示最诚挚的感激！澳大利亚莫纳什大学的陈少青同学也曾协助过本书的翻译，一并感谢。

许晓斌（@juvenxu）

杭州，2013/5/28

① 黄瓜是一种健康食品：http://www.whfoods.com/genpage.php?tname=foodspice&dbid=42。

序

行为驱动开发（behavior-driven development，BDD），自从我在 2003 年首次提出并讨论它之后，已经取得了长足的进步。那个时候，我只是在试着寻找更好的方法来解释带有启发性的 TDD 实践，通常是对一些紧张不安、疑虑重重或至少持保留态度的程序员。为什么要在编码之前先写测试？没道理嘛。而且，为什么是我们要写测试——不是有测试人员吗？

很少有什么东西能代表真正范型（paradigm）上的转换。多数时候“范型”[①]这一术语是市场营销人员用来说服你改换牙膏品牌的。根据免费在线词典（Free Online Dictionary）的解释，范型是“团体共享的一组假设、概念、价值和实践，它们构成了团体看待现实的方式。”没错，范型的转变会打乱一个人的现实感！难怪人们会不舒服。

TDD 就是少有的这种真正的范型转换之一，因此，在你尝试引入它的时候很多人深表怀疑也就不足为奇了。而同样不足为奇的是，我们需要多次以不同的方式，从不同的角度，向不同的受众来明确地表达，然后方能找到真正有效的东西。起初我们从代码深处开始，因为那是程序员关心的地方。后来我们便能采取行动更加接近业务干系人（business stakeholder）并描述多层次的方法，也就是现代 BDD（同时也是传统 TDD，这有点讽刺，Kent Beck 从一开始就将 TDD 描述为工作在多个抽象层次上的）。

Aslak Hellesøy 几乎从最开始就参与到对这一转变的描述中来。作为 BDD 的早期采用者，同时也是 TDD 的热情倡导者，他把我为 RSpec 构建一套场景运行器的痛苦努力重写成了如今我们称之为 Cucumber 的工具。他在工具及其社区上都投入了大量的时间和精力，因此，得知他和 Matt 在写一本用 Cucumber 实施 BDD 的书，我一点也不奇怪。我喜欢他们同时面向开发者和测试者这一做法。如果工具不能将这两个世界更紧密地联结在一起，那它就不是一种好的工具。

得知 Aslak 的“同谋”将是 Matt Wynne，我很开心。作为另一名热情且有经验的 TDD 实践者和 BDD 实践者，Matt 从第一天就跟 Cucumber 结下不解之缘。他是一位风趣又有魅力的演讲者和极好的教师，其作品中知识与智慧源源不断。事实上，我甚至提议为 Matt Wynne

① http://www.thefreedictionary.com/paradigm

定义一个全新的描述成功的形容词——马氏成功（Matt Wynee），其程序介于一般成功（Win）与卓越成功（Epic Win）之间。（噢，太酷了，你注意到了吗？那不仅仅是一般意义上的成功，那是一种“马氏成功”!）[①]

我希望你能同我一样喜爱这本书。记得在我审阅早期书稿的笔记中出现过好多“哦，太可爱了!”这样的语句，具体多少记不清了。那种感觉就像被两位博学而又随和的向导带进一个陌生却又似曾相识的世界。数不完的示例、描述和边框注释（Joe——你很快就能见到这个名字——问了一些我发现自己也想问的东西，不久就成了我的朋友。）在学习的道路上为你提供帮助，同时作者们也努力使情节足够快地发展，从而让你保持专注，对一本技术书籍来说，这永远都是一种挑战。

我说不出再过 8 年 BDD 会在哪里，但有 Matt 和 Aslak 这样的人分享他们的革新和洞见，眼下就是投入敏捷软件开发的一个激动人心的时刻。

——Dan North，DRW Trading 的精益技术专家，BDD 发起者

① 英语中 Matt Wynne、Win 和 Epic Win 这三个词是押韵的，这段话实际上是 Dan 开的一个玩笑。

前言

Cucumber 是一种非常友好的工具。它希望成为团队的一部分，且不介意做一个吹毛求疵的讨厌鬼。每个小组都需要这么一个角色，来记住关于系统能处理什么和不能处理什么的各种细节。

更为出色的是 Cucumber 会无偿做那些重复检查，以确保系统的运行符合预期。它可以把测试人员解放出来去做有意思、有创意的东西，并且给程序员必要时对代码做大手术的勇气。业务干系人对 Cucumber 这种开放的态度十分赞赏。Cucumber 可以用他们能够理解的术语分享开发团队所做的一切。

Cucumber 是一种新兴工具，但人们已对它有了些许误会。那些早期就开始接触 Cucumber 的人已经本能地意识到，Cucumber 不仅是一种测试工具，更是一种协作工具。通过本书，我们希望自己不仅能为你展示怎样使用 Cucumber，还能教会你如何更有效地利用 Cucumber。

本书的目标读者

Cucumber 旨在帮助软件团队在技术成员和非技术成员之间搭建一座桥梁。我们已经考虑到了这两类读者。本书主要写给那些至少掌握了一定编程技能且对自动化感兴趣的技术类读者。然而，本书的一些章节，尤其是前面解释如何编写规格说明本身的那一部分，主要是写给非技术读者的。具体来讲是以下几章。

- 第 1 章　为何使用 Cucumber
- 第 3 章　Gherkin 基础
- 第 5 章　富有表现力的场景
- 第 6 章　Cucumber 常见问题及解决之道
- 第 13 章　为遗留应用添加测试

随着内容的深入，我们将关注更加复杂的测试环境，并且读懂章节内容所需的技术层次也将升高。我们已经努力使知识结构尽可能地循序渐进，以便让刚刚接触测试和自动化的读者能跟上学习的步伐。

阅读本书不需要了解 Ruby，但了解是有帮助的

Ruby 是一种开源编程语言，可以在绝大多数操作系统上安装和运行。Cucumber 最早的版本就是用 Ruby 写的，到今天它也是最流行的版本，本书正是关于这个版本的。

这并不是说被测系统必须用 Ruby 来写。Ruby 的诸多优点之一是与其他语言和平台的完美交互。我们会向你展示如何利用 Ruby 工具来测试可用任何语言编写的基于 Web 的系统。

了解一点儿 Ruby 有助于跟上技术章节中的代码实例。Ruby 语言学起来很容易，同时我们用到的 Ruby 实例也很简单。为了最大限度地吸取本书的精华，我们建议 Ruby 新手同时使用 *Everyday scripting with Ruby*[Mar07]或者 *Programming Ruby:The Pragmatic Programmer's Guide*[TFH08] 。

学习 Cucumber 不必基于测试驱动

从一个失败的 Cucumber 测试开始，然后通过这种失败来驱动应用代码的开发工作，作为一种由外向内的开发方法的一部分，我们已经用 Cucumber 取得了极大的成功。作为开发人员，这种工作方式能让我们实事求是，一步一个脚印，避免我们想当然地开发将来也许有需求但当前没有需求的功能。

Cucumber 可以完善我们的工作方式，但它并不强制。一些团队利用 Cucumber 自动测试开发人员已经完成的工作。这是采用由外向内方法的第一步，因为 Cucumber 可读的测试已经吸引了团队中非技术干系人的视线并逐步使他们参与进来。即使用 Cucumber 编写针对已有代码的测试，你从 Cucumber 中的获得的好处仍然远远超出同类软件，比如 QTP 和 Selenium IDE。我们相信本书会使你收获很多。我们并不是鼓吹这个过程，而是想和你分享关于哪些方法对我们行之有效以及为什么有效的一些感悟。

为何要听从我们

我们已经开发软件二十多年，运用自动测试也有十余年。Aslak 在 2008 年开发了 Cucumber。Matt 从第一天就是最活跃的用户。

我们已经使用 Cucumber 测试过从 Ruby on Rails Web 应用到动漫游戏，再到企业 Java Web 服务在内的所有系统。我们还培训过成百上千的开发人员，教他们如何使用 Cucumber，并在各种会议及世界各大公司中讲授书中的内容。

Cucumber 社区充满了有活力的争论。我们花费了大量时间与用户讨论，让大家挑战并打磨我们的想法。我们希望这本书已经涵盖了我们提炼出的尽可能多的知识和经验。

本书的组织结构

本书分为三个部分。第一部分我们主要带你了解使用 Cucumber 时需要明白的核心概念。初级读者会学到他们继续走下去所需了解的全部知识。已经体验过 Cucumber 的读者也可以了解到更多有用的细节。

第二部分贯穿了一个运用 Cucumber 开发新应用程序的示例。因为我们从设计应用程序起步，你将跟我们一起从零开始构造一个简单的应用，从而有机会体验我们所喜欢的利用 Cucumber 开发软件的方式，并进一步巩固你在第一部分学到的知识。我们还会教你一些 Cucumber 的高级特性，这些特性结合示例讲解更易于学习。

Cucumber 提供了一种定义并执行测试的框架，但需要测试的系统是多种多样的。在第三部分，我们为你提供了将 Cucumber 用于常见情形的广泛指导，比如用于 REST API、Ajax Web 应用和命令行应用的测试。

本书没有的内容

虽然用 Cucumber 测试动画和手机应用是可行的，但具体细节超出了本版的范围。在 JVM、JavaScript 和 C#上运行的 Cucumber 实现允许你使用编写产品代码的语言来编写 Cucumber 代码，但这本书同样不会涵盖这一主题。Cucumber 的连线协议（wire protocol，一种通过 TCP 套接字驱动远程系统的协议）也超出了本书的范围。

关于这一主题，我们在附录 A 中提供了更多信息的链接列表。关于运用 Ruby 自动化并测试不同系统的更多信息，作为对本书的补充，我们推荐你阅读 Ian Dees 写的 *Scripted GUI Testing with Ruby* [Dee08]。

运行代码示例

本书以示例为主，我们鼓励你跟着示例学习本书的绝大部分内容。边读边写能使你学到最多知识。但如果你喜欢，可以在下面的网址下载代码实例：http://pragprog.com/titles/hwcuc/source_code。

Windows 用户

绝大多数代码示例在 Windows 和*nix 操作系统上运行时完全一样。极个别*nix 跟 Windows 系统不一样的情况，你会在附近的注释框中找到 Windows 版本，同时文本中会有说明告诉你去哪个注释框找。

很快你会注意到，我们使用$符号作为命令行提示符。这是大多数 Linux 和 Mac 用户所熟悉的，但 Windows 用户会感到些许陌生。所以看到类似下面的情况时：

```
$ cucumber
```

你可以视同看到的是：

```
C:\> cucumber
```

除此之外，其他都是一样的。

获得帮助

如果你在本书的任何一个环节上卡住了，可以上论坛 http://forums.pragprog.com/forums/166 寻求帮助。

如果你有针对 Cucumber 的普遍性问题，Cucumber 社区欢迎你向邮件列表 https://groups.google.com/forum/#!forum/cukes 发信。Cucumber 是一种开源工具，意味着这个群体中的成员都是无偿奉献自己的时间的。所以，在发信寻求帮助时，务必确保自己已经彻底调查过相关问题。人们只有看到你尝试帮助自己时，才会更愿意帮助你。

目录

第一部分

Cucumber 基础