

来自
全球软件设计与定制领袖企业 **ThoughtWorks**
的宝贵创意与实践

软件开发与创新

ThoughtWorks 文集（续集）

ThoughtWorks公司 著
ThoughtWorks中国公司 译

国际知名OO专家、敏捷运动创始人

Martin Fowler

以及诸多世界级程序员
与您一道：

- 应对最严苛的技术挑战
- 经营可持续的业务
- 追求软件卓越
- 推动IT变革



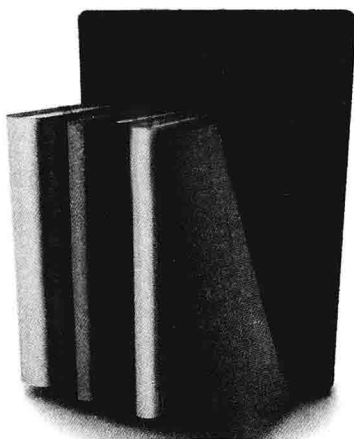
人民邮电出版社
POSTS & TELECOM PRESS

TURIN

软件开发与创新

ThoughtWorks 文集 (续集)

ThoughtWorks公司 著
ThoughtWorks中国公司 译



人民邮电出版社
北 京

图书在版编目 (C I P) 数据

软件开发与创新 : ThoughtWorks文集 : 续集 / 美国ThoughtWorks公司著 ; ThoughtWorks中国公司译. -- 北京 : 人民邮电出版社, 2014. 1

(图灵程序设计丛书)

书名原文: The ThoughtWorks anthology, volume 2:
More essays on software technology and innovation
ISBN 978-7-115-34294-2

I. ①软… II. ①美… ②T… III. ①软件开发—文集
IV. ①TP311.52-53

中国版本图书馆CIP数据核字 (2013) 第315099号

内 容 提 要

本书中涵盖的软件开发主题十分广泛, 从优化敏捷方法论到核心语言都有涉及。其中包括对持续集成、测试和改进软件交付过程提出的独到建议, 以及如何在面向对象语言和现代 Java Web 应用程序中使用函数式编程技术等。

本书条理清晰、思维严谨却又不乏生动活泼之处, 即便是书中专业性最强的文章, 也不会让人觉得难以理解。除了技术人员外, 本书对相关的非技术人员也很有价值。

-
- ◆ 著 ThoughtWorks公司
 - 译 ThoughtWorks中国公司
 - 责任编辑 丁晓昀
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 12
 - 字数: 283千字 2014年1月第1版
 - 印数: 1-4 000册 2014年1月北京第1次印刷
 - 著作权合同登记号 图字: 01-2013-1137号
-

定价: 45.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版 权 声 明

Copyright © 2012 ThoughtWorks. Original English language edition, entitled *The ThoughtWorks Anthology, Volume 2: More Essays on Software Technology and Innovation*.

Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由The Pragmatic Programmers, LLC. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

作者简介

Farooq Ali

作为一位多专业技能的通才以及T型思考者，Farooq乐于从诸多不同的角度思考问题，帮助团队提出创新性的解决方案。作为ThoughtWorks的首席咨询师，他曾经担当过许多不同的角色：开发人员、业务分析师、项目经理、用户体验设计师。Farooq对于可视化思考充满激情，并将其运用于产品构思、代码美学、数据分析等诸多领域。Farooq现任ThoughtWorks美洲社会影响力计划的负责人，为技术、创新与社会影响寻求更好的解决方案。

Ola Bini

Ola Bini来自瑞典，是一名程序设计语言极客，在ThoughtWorks芝加哥分部工作。他是JRuby的核心开发者之一，从2006年开始参与JRuby的开发。曾几何时，Ola厌倦了所有的现有语言，于是他创建了自己的语言Ioke。后来，他又厌倦了，于是又有了Seph。他曾撰写了《JRuby实战》（*Practical JRuby on Rails*）一书，还是*Using JRuby*的合著者。他在无数的会议上发言，为很多开源项目贡献过代码。他还是JSR292的专家组成员之一。

他热衷实现程序设计语言、正则表达式引擎以及寻找更好的YAML解析器的实现方式。

Brian Blignaut

Brian曾在ThoughtWorks担任了3年多的首席咨询师。在此期间，他参与交付了很多知名公司的定制项目，包括大型的面向终端客户的网站和实时流计算平台。他曾做过很多JavaScript测试方面的演讲，现在在伦敦做独立咨询师。

James Bull

James是一个有QA背景的敏捷软件开发者。他在ThoughtWorks做了很多测试自动化方面的工作。他坚信，整个团队都能从中受益的测试套件才是好的测试套件。James没在摆弄电脑的时候，肯定是在摆弄爱车。

Neal Ford

Neal Ford在ThoughtWorks担任软件架构师、Meme Wrangler。他还设计和开发过很多应用程

序，给很多杂志写过文章，做过很多视频或者DVD演讲，还曾出版、编辑、参与过8本涉及多种题材的书。他专注于设计和构建大型的企业应用程序。他还是一名在国际上颇受欢迎的演讲者，曾在300多个开发者会议上做过2000多次演讲。

他的网站是<http://nealford.com>，也可以通过邮件nford@thoughtworks.com联系他，他欢迎各种反馈意见。

Martin Fowler

Martin自称作家、演讲者……其实他就是一个在软件开发这个话题上喋喋不休的博学者。Martin从20世纪80年代中期就开始在软件行业工作了，也是在那时他接触到了面向对象的“新概念”。他在90年代的大多数时间是一名咨询师和培训师，帮助人们开发面向对象系统，专注于企业应用程序。他于2000年加入ThoughtWorks。

他最大的兴趣是，找出一种设计软件的方式，尽可能提高开发团队的生产率。在寻找的过程中，他努力找出好的软件设计模式，以及支持这种模式的开发流程。Martin成了敏捷方法的狂热追随者，还关注演进式软件设计。

Luca Grulla

Luca曾在ThoughtWorks工作过4年。作为首席咨询师，他帮助客户采用敏捷和精益的方法交付高质量的软件，目前在伦敦的Forward做高级程序员。他现在的工作是，试用新的语言和新的技术，每天向产品环境提交几个新特性。他还是全球IT社区的活跃成员，经常在国际性的会议上发言，还是Italian Agile Day、EuroClojure等欧洲会议的委员会成员。

Alistair Jones

Alistair Jones既是开发人员与技术负责人，又是架构师及教练。他组建出来的团队具有良好的决策能力，也能够产出优良的软件。他喜欢向人们展示，与老式的交付方式相比，敏捷方法更需要（且更能形成）严明的纪律。

Aman King

Aman King是一个应用程序开发人员。作为分布式团队的一员，他构建过很多复杂的商业应用。TDD是他的阳光和空气，他做起重构来就好像和代码有仇一样。

Patrick Kua

Patrick Kua在ThoughtWorks是一名活跃的多面手，他不喜欢被人贴上标签。Patrick多数时间都在领导技术团队，培训敏捷和精益方法。有时，他也会拯救团队于水火之中。Patrick热衷研究学习之道和持续改进，也乐于帮助他人，激起他们在这些领域的兴趣。

Marc McNeill

Marc关注把多种不同风格的团队团结起来，精诚一致地打造非凡的用户体验。Marc拥有人因工程专业的博士学位，在ThoughtWorks工作的7年间，他把设计思维和精益创业介绍给了世界各地的客户团队。他做事迅速而且注重成效，通过不断地试错帮助团队把想法转化成成功的产品。他是《当用户体验设计遇上敏捷》(*Agile Experience Design*)一书的作者之一（另一合作者是Lindsay Ratcliffe）。他的Twitter账号是@dancingmango。

Julio Maia

Julio Maia在ThoughtWorks做技术咨询师已经5年了。他在集成、自动化、运营、测试基础和应用开发等领域帮助客户解决问题，构建软件解决方案。

Mark Needham

Mark Needham是ThoughtWorks的一名软件开发人员。在ThoughtWorks工作的6年中，他使用敏捷方法帮客户解决问题，在项目中用过C#、Java、Ruby，还有Scala。

Sam Newman

Sam Newman是ThoughtWorks的技术咨询师，在ThoughtWorks工作8年有余。他在很多公司工作过，始终致力于利用技术扩展IT影响力。

Rebecca Parsons

Rebecca Parsons是ThoughtWorks的CTO，她已经想不起自己接触技术有多长时间了。她对各种技术，尤其是程序设计语言富有热情。她在莱斯大学取得了计算机科学的博士学位，其间，她专攻语言语义学和编译器。她还做过进化计算和计算生物学方面的工作。

Cosmin Stejerean

Cosmin Stejerean是有8年多经验的专业软件开发者。他现在在Simple做运维工程师，生活在得克萨斯州达拉斯市。他之前是ThoughtWorks的首席咨询师和培训师。

中文版推荐序

说起做软件、写代码，不少从业人员的追求止于完成功能、解决问题。毫无疑问，这样的做法也确实能把东西做出来，但为什么渐渐地改点东西这么费劲？为什么上线交付总是磕磕碰碰？为什么用户会有那么多的抱怨？管理人员把原因归咎于流程问题、组织问题、大环境问题，开发人员则信誓旦旦地声称“只要给我足够的时间，给我足够的资源，我肯定……”。这些说法虽然也没错，然而在接触了全球众多不同的团队后，我们一个很简单的观察却是：专业的软件大多出自专业的人员，特别是当软件要解决的问题复杂到一定程度的时候。

所谓的专业人员并不仅仅是要掌握手头的编程语言、平台和工具。公司CTO Rebecca Parsons在一篇文章中诠释了ThoughtWorks对软件卓越的理解，提出了软件开发活动中应该关注的维度：业务价值、客户价值、用户体验、技术卓越、交付效力和运营效力。区别于那些介绍某种语言、工具的食谱类书籍，ThoughtWorks文集的价值在于为想要或已经成为专业从业者的人们提供了一个概要地图，并从问题域出发，着重介绍了在该领域新的实践。

ThoughtWorks文集第一卷《软件开发沉思录：ThoughtWorks文集》出版已经有4年了，在那之后，行业出现一些新的现象和趋势。多语言编程渐渐进入开发人员的视野，并在尝试中展示出一些优势；函数式编程的思想和技巧开始影响人们的编程风格；继“持续集成”之后，“持续交付”将自动化和快速反馈引向软件的全生命周期；越来越多的团队使用特性开关等手段应对快速交付中面临的诸多挑战，例如并行开发、多定制版本等；设计和创新不再是互联网和一些超炫产品公司的专属，品味日高的用户迫使各类软件团队拥抱体验设计和验证的能力。本书则捕捉并展示了这些领域的发展。

这本文集的作者是公司里在各自领域都有相当建树的同事。虽然我在公司也待了一些年头，但平时也不是有那么多的机会能够跟他们直接交流，以学习他们在项目上正在实践着的各项技术和方法。本书可以说是同事们在追求软件卓越的道路上继续前行中的又一次总结，我自己在阅读中获益匪浅。

相对第一卷散文式的组织方式，本卷则通过更加清晰的结构，更加全面地覆盖了软件开发技术和创新几个关键的方面。与此同时，仍然保持了每个章节单独成篇的特点，便于读者选择有兴趣、有需要的章节阅读。

在英文版出版之后，ThoughtWorks中国区的同事们尽快将本书翻译成了中文，方便中国的读者阅读。我衷心希望对软件开发有热情的朋友不妨一读本书，相信会对大家起到不错的借鉴和启发作用。

张松

ThoughtWorks中国区总经理，

《精益软件度量——实践者的观察与思考》作者

译者序

ThoughtWorker们的作品，由ThoughtWorker们翻译成中文，似乎是天经地义的。2009年，因为这个原因，我加入了第一本ThoughtWorks文集的翻译工作中；2013年，又是因为这个原因，我组织翻译了第二本ThoughtWorks文集。于是，我有幸成为唯一一名横跨两本文集的译者。

当初，我带着怎样做好软件的疑问加入了ThoughtWorks。时光荏苒，我在ThoughtWorks工作了差不多7年，当年的疑惑已经有了属于自己的答案。或许，在许多人眼中，这个答案只是“敏捷”，但我看到的是，ThoughtWorker们的精益求精。在一些人眼中已经可以接受的标准，在ThoughtWorker们眼中，却是可以做得更好：当年“日构建”的横空出世，已经让世人为之惊叹，但“持续集成”却把构建这个常见的任务提到了一个新的境界；在人们已经对持续集成甘之若饴时，ThoughtWorker们却说我们应该做“持续交付”，于是，新一轮的软件开发方式变革随之而来。持续交付会是终点吗？我不这么认为，因为我看到ThoughtWorker们仍在不断做出新的尝试。

虽然ThoughtWorker们的所有探索并非都如“持续集成”、“持续交付”般影响深远，但我们却在软件开发的各个领域进行着不断的摸索，尝试找到各种更好的解决方案。《ThoughtWorks文集》正是这样不断探索的阶段性总结。

我虽是译者，同时也是读者。单就个人喜好而言，我更喜欢第二本文集。究其原因，其一是，站在今天的角度看，第一本文集里面的有些内容稍显过时，比如，在如今这个Maven都已经不受待见的时代，谈论重构Ant构建文件显得很不合时宜；其二，纯属我作为程序员的个人偏好：第一本文集留给程序员的空间太小了，对日常开发工作有直接指导意义的内容偏少一些。而第二本文集更接地气，很多文章很程序员化——尽管它们可能不一定与许多人惯常的开发方式一致，当然，如果一致了，便也没有写出来的必要了。这些“不同寻常”的内容如果能够引发思考便是达到了目的了，如果进一步改变了一些人的开发方式，那无异于对作者们的额外嘉奖。

一本由多名ThoughtWorker合作的书，由多名ThoughtWorker合作翻译，也是一件顺其自然的事。朱晓娜完成了本书第1、5、9章的翻译，韩锴负责第2、11、12章的翻译，第3、6、10章由姜鹏翻译，崔鹏飞则翻译了第4、7、8章以及序、关于作者等各种边角余料。而撰写了译者序的我，则在整个过程中打了很多零工，审校了所有的内容，统一了翻译风格。所以，但凡有任何纰漏之处，我难辞其疚。

无论在上下班拥挤的公交地铁上，还是运行构建的间隙，抑或是飘着悠扬音乐的咖啡厅里，但愿随手翻出的这本轻薄小卷，能够带给你一些思考。

祝开发愉快！

郑晔

ThoughtWorks首席咨询师，
2013 Duke选择奖获奖者

前言

虽说商业模式是很多公司的主要立身之本，但ThoughtWorks则是植根于一个社会模型中的。我们以三根支柱衡量公司的成败，并以此作为决策基础。

- 基业永续。
- 止于至善。
- 不为利回，不以义疚。

ThoughtWorks的这种商业模式和社会模型不断地激励着我们挑战组织结构和商业成功的现有定义。ThoughtWorks本是一个社会实验，它会随时间而演进，但我们相信100年后，ThoughtWorks仍会存在，并继续发挥着影响力。如果那时你还健在，想想书架上会有多厚的一摞ThoughtWorks文集吧！

Rebecca Parsons

rjparson@thoughtworks.com

2012年6月

Martin Fowler

fowler@acm.org

2012年6月

目 录

第 1 章 引言	1
----------------	---

第一部分 语言

第 2 章 最有趣的语言	4
--------------------	---

2.1 为什么语言很重要	5
2.2 一些有趣的语言	5
2.2.1 Clojure	5
2.2.2 CoffeeScript	10
2.2.3 Erlang	14
2.2.4 Factor	18
2.2.5 Fantom	21
2.2.6 Haskell	26
2.2.7 Io	30
2.3 总结	33

第 3 章 面向对象程序设计：对象优于类	34
----------------------------	----

3.1 对象优于类	35
3.2 类关注与对象关注	36
3.2.1 角色的角色	36
3.2.2 职责分离	42
3.2.3 测试的角度	45
3.2.4 代码库里的线索	46
3.3 “对象关注”的语言	47
3.3.1 Ruby	47
3.3.2 JavaScript	53
3.3.3 Groovy	56
3.3.4 Scala	58
3.4 要点回顾	58
3.5 总结	59

第 4 章 使用面向对象语言进行函数式编程	60
-----------------------------	----

4.1 集合	60
4.1.1 转换思维	60
4.1.2 拥抱集合	63
4.1.3 勿忘封装	64
4.1.4 惰性求值	65
4.2 “一等公民”和高阶函数	67
4.3 状态最小化	69
4.4 其他理念	70
4.5 总结	73

第二部分 测试

第 5 章 极限性能测试	76
--------------------	----

5.1 问题描述	76
5.1.1 分离性能测试的传统方式	76
5.1.2 极限编程和敏捷软件开发	77
5.1.3 分离性能测试的不足	78
5.2 另辟蹊径	78
5.2.1 独立的多功能团队	79
5.2.2 描述需求	80
5.2.3 设定计划与排定优先级	81
5.2.4 实现性能故事	82
5.2.5 展示与反馈	83
5.3 极限性能测试实践	83
5.3.1 性能负责人	83
5.3.2 自动化部署	84
5.3.3 自动化分析	85
5.3.4 结果仓库	85

5.3.5 结果可视化	86	7.2.2 等待页面元素显示时要小心 (再次强调)	109
5.3.6 自动化测试流程	86	7.2.3 在测试中设置测试依赖的数据	110
5.3.7 健全性测试	87	7.2.4 测试集成点	110
5.3.8 持续性能测试	88	7.3 易于维护的测试	111
5.3.9 规范的性能提升	88	7.3.1 使用页面模型	111
5.4 这对我们有何帮助	89	7.3.2 结构一致的测试集	112
5.4.1 更好的性能	89	7.3.3 测试代码产品代码一视同仁	113
5.4.2 更低的复杂度	89	7.3.4 切勿受限于工具	113
5.4.3 更高的团队效率	90	7.4 付诸实践	114
5.4.4 更合理的优先级排定	90	7.4.1 一地团队	114
5.4.5 开启持续交付	90	7.4.2 维护测试, 人人有责	115
5.5 总结	90	7.4.3 故事启动	115
第 6 章 测试驱动 JavaScript	91	7.4.4 结对测试开发	115
6.1 JavaScript 的复兴	91	7.4.5 故事展示	116
6.2 当前 JavaScript 的处理方式与问题	92	7.5 总结	116
6.3 分离关注点	92		
6.4 测试方式	100	第三部分 软件开发问题	
6.4.1 倾向于交互测试, 而非集成 测试	100	第 8 章 现代 Java Web 应用	118
6.4.2 在具体用例中使用 HTML 夹 具编写集成测试	100	8.1 过去的状况	118
6.4.3 使用验收测试验证所有组件的 集成	101	8.1.1 有状态的服务器	119
6.5 持续集成	101	8.1.2 依赖容器	119
6.6 工具	101	8.1.3 违反 HTTP 规范	120
6.6.1 单元测试	102	8.2 无状态服务器	120
6.6.2 语法检查	102	8.2.1 集群	120
6.6.3 mock 框架	102	8.2.2 cookie 救世	121
6.7 总结	102	8.2.3 区分用户特定的数据	121
第 7 章 构建更好的验收测试	103	8.2.4 安全 cookie	122
7.1 快速测试	103	8.3 容器是可选的	123
7.1.1 基于用户行程的测试	103	8.3.1 容器外测试	123
7.1.2 并行执行测试集	104	8.3.2 我们真的需要容器吗	125
7.1.3 考虑使用多种测试驱动器	105	8.4 按新鲜程度分区	125
7.1.4 将测试分开运行	107	8.4.1 缓存: 可扩展网站的秘密武器	125
7.1.5 等待页面元素显示时要小心	107	8.4.2 选择缓存的内容	126
7.2 有弹性的测试	107	8.4.3 按新鲜程度分区简介	126
7.2.1 单独选择页面元素	108	8.4.4 反向代理和内容发布网络简介	128
		8.5 POST 重定向到 GET	129
		8.6 总结	130

第9章 驾驭集成难题	131
9.1 持续集成方法	132
9.1.1 稳定基准	132
9.1.2 集成 stub	133
9.1.3 构建流水线	134
9.1.4 监控器	134
9.2 定义集成契约	135
9.3 度量和可见性	135
9.4 总结	136
第10章 实践中的特性开关	137
10.1 简单特性开关	138
10.2 可维护的特性开关	138
10.2.1 依赖注入	139
10.2.2 注解	140
10.3 分离静态资源	141
10.4 阻止意外泄露	142
10.5 运行时开关	142
10.6 不兼容依赖	143
10.7 特性开关的测试	143
10.8 删除完成特性的开关	144
10.9 总结	144
第11章 交付创新	145
11.1 价值流向	146
11.2 新方法	147
11.2.1 协作文化	147
11.2.2 敏捷产品调研与发现	149

第1章

引言



Neal Ford撰文

我热爱文集。少年时，我热衷科幻小说，也有幸读到了大量内容丰富的科幻杂志。这些杂志每年都会选出最棒的故事结集成册，那可都是精华之作啊！

这些优选集伴我度过了许多悠闲时光。我之所以热爱这些文集，是因为每个故事由不同的作者写就；每每读到一个新故事，全新的写作风格都会令我耳目一新。每个故事中都有着其独到的世界观、情节设定及背景，所以，我爱文集。

多年之后，我参与编纂了多本（非虚构）文集，包括《软件开发沉思录：ThoughtWorks文集》[Inc08]，第一本ThoughtWorks文集。在变化迅速的软件行业里，博客和杂志内容分散，而专门题材的书籍又内容单一，文集恰如其分地填补了二者之间的空白。本书这样的文集如同一张时代快照，涵盖了流程、技术、哲学等诸多处于当下前沿的思想。

本书是第二本ThoughtWorks文集。编纂第一本时，Rebecca Parsons发出号召征集文章，并收到了很多高质量的投稿，最终呈现给读者的便是一本内容优秀且涉猎广泛的文集。集结第二本文集时，我们同样发出了征文号召。在此之前，大家都已听说了第一本文集，因此，第二轮征文时，大家的兴趣高了许多。我们收到了100多份稿件，其中不乏许多优秀之作。因为大家的热烈反响，我们还让ThoughtWorks技术战略委员会参与其中，这是一个辅助CTO的内部组织，协助筛选和评审这些稿件。委员会成员对提交的稿件进行了精挑细选。新的ThoughtWorks文集可谓优中选优。

正如Rebecca在本版前言中所述，ThoughtWorks是一家注重多样性的公司，包括思想的多样性。我们会在下班之后聚在一起聊聊大家又养成了哪些奇特的嗜好，或者参加午间讨论，讨论一些广泛而深远的话题，远远不只局限于软件，这些都是我们在ThoughtWorks最喜欢做的事情。我想，这些文章会让你感受到这种多样性。尽管所有文章谈的都是软件开发，但每篇文章的观点都独具特色。

正因为内容的多样性，你可以采用多种方式阅读本书。

如果你跟我一样，喜欢在不同的作者所著的全新内容之间不停切换，可以从头至尾通读本书，

当然也可以根据不同的主题进行阅读。

如果你热衷敏捷软件开发流程，请阅读第11章“交付创新”，这一章讨论了将创新力注入交付流水线的技术；你也可以从第9章“驾驭集成难题”开始阅读，其中围绕分散系统集成，讲述了解决这一棘手问题的一些行之有效的办法。

如果你想把范围缩小到敏捷与技术的交叉话题，请阅读第7章“构建更好的验收测试”、第5章“极限性能测试”以及第6章“测试驱动JavaScript”，这些文章覆盖了项目测试的方方面面。

如果你喜爱纯粹的技术话题，可以看第10章“实践中的特性开关”、第4章“使用面向对象语言进行函数式编程”、第8章“现代Java Web应用”，第3章“面向对象程序设计：对象优于类”以及第2章“最有趣的语言”。

最后，如果你相信图表的功用，那第12章“一图胜千言”将为你展示如何创建令人瞩目的可视化技术产品。

当然，阅读顺序无关紧要。这些文章是作者们利用自己的“业余”时间编写的。写作期间，他们放弃了陪伴家人和朋友的机会，也没法参加各种娱乐活动。作者对传达信息的热诚在书中展露无疑，进而也体现出他们为此所作的奉献。希望读者阅读这些文章时，能像作者写作时一样乐享其中。