

代码之道



[代码之道_下载链接1](#)

著者:Eric Brechner

出版者:机械工业出版社

出版时间:2009-1

装帧:

isbn:9787111251675

本书总共有49个栏目，以一位微软内部人士的视角，揭示了关于软件编码、软件测试和项目管理的残酷现实。作者文笔犀利，见解独到，对软件行业内的很多常见问题提出了解决方案，并提供了最佳实践。通过阅读本书，你将了解到：怎样提高软件的质量和价值；怎样切合实际地管理项目的时间表、风险和规范书；怎样为常见的低效率开发瘦身；怎样应用过程改进方法，避免固执盲从；怎样驱动一个成功的、令你自己满意的职业生涯；以及怎样发展并管理一个欣欣向荣的团队。.

本书提出的观点，是作者对过去在软件行业6个不同的公司、28年的工作经验的一次总结。本书不仅是微软内部员工的必读之书，它也同样适合于软件行业内其他所有的工程师和管理者阅读。

作者介绍:

目录:

[代码之道_下载链接1](#)

标签

软件工程

编程

计算机

软件开发

微软

管理

programming

程序人生

评论

Make it simple, make complex possible

开发人员能遇到的所有事情，另外，我很喜欢他的写作风格

匆匆翻阅，木有《代码大全》那样爽。大概是因为还是学生，没有真正接触项目，所以

无法产生共鸣。

作者见解独到，读后还是很有收获的。“重写的代码实际上将比当前的丑陋代码带来更多的Bug。因为当前的丑陋代码是在第一次编写的基础上，经过了几个月甚至是几年的测试和修复之后才达到的质量水准”，这一段很有体会啊。有时候重写是必要的。重写可以提高代码的性能、扩展性、可靠性、安全性、或者对于新技术的适应能力。在这些情况下，应该把重写当作一个功能来对待；像处理其他功能一样，写一份规范书，然后为它制定时间表。“否则，不要做代码重写这种愚蠢的事情，它只会重新引入一大堆令人讨厌的Bug，而且还对客户价值没有丝毫的贡献。”规范书检查清单。维持一个在线的知识仓库。如何成为一个优秀管理者--我服务于人，避免团队成员之间的比较。不要给他们任何理由去进行相互竞争。

对我而言，Scrum 已经足够了

译名“代码之道”，道倒是很贴切，至于代码。。。似乎欠妥

42

有点意思

本书由一篇篇独立文章构成，虽然书籍组织时已给这些文章归了类，但整体上还是比较散，阅读时虽然能不停的感觉到新启示，但读完后却没几个想法能自然的回忆起。这可能和阅读方法有关。这种文章集不大适合一口气一气呵成的读，分散开，单篇单篇的阅读和整理心得，可能会好些。回头抽几篇单篇细读下试试。

本来这应该是很轻松的书，关于项目，开发，和工程，但是由于翻译不能表现出那种轻松调侃的语气，读来有些不畅快。而且是来自微软，一个庞大而且规范场合，与我们的作坊语境不同。

大学毕业再看一遍，现在看没实感。

很棒的一本书，介绍了微软内部的最佳实践。尤其是成为“优秀管理者”的两个基本要素，很简单，很实用，这么多年来一直鼓舞着我！

pass

[代码之道 下载链接1](#)

书评

近2年来读到的最好的最贴近技术的非技术类书籍。
每一个章节，都找到了自己和身边的团队，项目的影子。直到现在，阅读《人月神话》的时候，都会仰视那些高瞻远瞩的观点，但是不知其所从来，但是这本书可以说是一个桥梁，真正的展开了你身边的问题，你曾经有过但是没有仔细考虑...

非常好。网上可以找到英文电子版，不过有些地方比较口语化，需要多费点力气进行理解。中文版有一部分内容可以在线试读<http://book.51cto.com/art/200810/93998.htm>

写得不错，但是应该出本儿英文的。有些东西翻译成中文很别扭，像break build，code review之类的。

《华章程序员书库:代码之殇(原书第2版)》是《代码大全》的姊妹篇，资深软件开发专家30余年工作经验结晶，被誉为“软件行业的财富”，微软公司软件工程师必读之书。它从软件开发流程、技术、方法、项目管理、团队管理、人际沟通等多角度总结出90余个具有代表性的问题（大多数问...

