《完美代码》

图书基本信息

书名：《完美代码》

13位ISBN编号：9787111292401

10位ISBN编号：7111292405

出版时间：2010年1月

出版社：机械工业出版社

作者：Donis Marshall,John Bruno

页数：243

译者：徐旭铭

《完美代码》

前言

软件工程和传统意义上的工程完全不同。作为一名软件程序员，我非常自豪可以自称是一名工程师。工程师能够通过细致的计划思考并制造出一次就能工作的东西。所以，在我的职业里包含“ 工程师 ”这个词实在是一件很酷的事情。我们先来看看要是把普通的软件工程方法应用到航空工程里会发生什么事情。一驾飞机正停在登机口等待乘客登机，这时一名航空工程师（不管是一时兴起还是被老板强迫）打算要换掉尾翼部分。毕竟那只是一个尾翼而已，直接把它弄下来换一个上去就好了。绝对没问题，绝对能行！要是航空工程采用和软件工程一样的流程的话，我估计乘客都会瞬间逃离飞机的。但是，这类改变在软件项目的世界里基本上每天都要发生。过去有种自相矛盾的老笑话说是“ 军事情报 ”，我觉得“ 软件工程 ”也属于这类范畴（这种笑话就是把两个矛盾或者互斥的词放在一起，通常都是为了幽默搞笑——译者注）。而更让人头疼的是，虽然软件真的是在“ 统治 ”这个世界，但是几乎所有人在创建它时所采用的方法无一可以称得上是“ 工程 ”。为什么我敢说我现在正在使用的计算机可以正常工作，但是我正在使用的程序（MicrosoftWord）却可能会搞乱我列表的自动编号呢？我这么说可能会得罪我那些电子工程的朋友，但是我还是要说，因为硬件比较简单。电子工程师要对付的只是很有限的输入，不像软件工程师要面对几乎无限可能的输入。管理学也认为电子工程是“ 真正的工程 ”，所以管理的时候可以给予适当的时间和资源。而软件业，作为一个独特的领域，还算不上成熟，它真正存在的时间不是很长。其实说起来，我比软件这个行业年轻不了多少，所以我“ 不成熟 ”的看法才能透露出这些问题。要是我和电子工程一样老的话，那么现在我就该躺在坟墓里写书了。软件开发的另一个难题有时候是软件工程师自己。老实说，成为一名软件工程师的门槛还是很低的。我就是一个现成的例子：在获得计算机科学学士学位之前，我就已经是一名全职的软件工程师了。当初获得这份写程序的工作只不过是因为我在面试的时候“ 特别能吹 ”而已。我的老板根本就不在乎我没有受过正规的教育，毕竟我要求的工资比较低。在所有真正的工程领域里，你都必须先获得认证资格后才能在名字里加上专业工程师（Pro.fessional Engineer，PE）的头衔。在软件行业里就没这种规定。其中一部分原因是这个行业太年轻了，没有人知道在从业之前应该必须掌握什么知识才能成为一个软件工程师。而在其他领域里，专业工程师通常意味着大量管理上的责任。如果一名有资格的工程师觉得一个设计是行不通的，她就不会在计划书上签名，项目也就不可以进行下去。

《完美代码》

内容概要

本书简单明了地介绍了软件开发中的最佳实践，展示了工程流程在编写优质代码上的重要性以及测试的重要性，总结了很多资深工程师的经验教训，并提供了很多真实案例。书中介绍的经验可以应用到产品开发周期的每个环节，从设计到开发以及最后的发布和维护。本书的中心思想就是要在设计和实现的过程中改进代码质量，包括类建模、性能、安全性、内存使用以及调试，帮助读者构建完美的项目。本书适合专业及业余程序员阅读。
采用一流的工程实践来帮助你编写更健壮、无错的代码。两位微软的.NET开发专家与你分享优化软件开发生命周期的真实案例和经过实战考验的解决方案——从避免代价昂贵的编程陷阱，到提高开发团队整体效率的方法等。无论你是来自哪个层次的托管代码程序员，都能在这里找到设计、原型开发、实现、调试以及测试的技巧，进一步提升代码的品质。
本书涉及开发流程中每一个阶段的优化（从设计到测试），以及如何开发出更优质的应用程序软件：
通过元编程来降低代码的复杂度，同时还能增加灵活性和可维护性。
把性能当做一项功能，并且在开发周期中对它进行管理。
为应用程序的伸缩性采取各种最佳实践。
通过预防性的安全措施来抵御各种恶意攻击。
在运行之前通过防御性编程来捕捉错误。
在每日工程流程里植入自动化构建、代码分析和测试等工作。
实现更好的源码控制管理和提交流程。
建立一套质量驱动、基于里程碑的项目节奏，并最终产生更好的结果。

作者简介

Donis Marshall 是Debuglive.com的CEO，他管理的专家软件工程师团队开发出第一个基于Web的Windows应用程序调试器。凭借20年的开发经验以及深厚的微软.NET背景，他编写了好几本书，其中包括《Programming Microsoft Visual C# 2008: The Language and .NET Security Programming》。Donis还是一名培训师和咨询师，专门讲授并主持关于.NET编程、调试、安全性以及设计和架构的研讨会。John Bruno 是微软的资深程序经理，有着超过10年的应用开发经验，他擅长使用微软.NET技术来设计并构建可扩展的Web应用和服务。加入微软以来，John对 Windows Live的发布起到重要作用，同时他还负责Windows Live Spaces的服务架构和程序员平台，Windows Live Spaces目前在全世界的用户数超过了1亿。现在他的主要精力都放在了开发Windows Mobile下一代Web服务上。

《完美代码》

书籍目录

《完美代码》

章节摘录

插图：作为软件工程师，我们都希望自己的产品是出色的。我们希望能让用户享受到无错的使用体验以及最佳的程序质量。而在同行之间，我们希望能用优雅和稳定的代码展示出自己超凡的技艺。每天我们都在向着这些目标努力工作。我们不断地重复努力开发出高质量的软件，加上我们不断的对改进方法和实践的渴望，凝聚成程序员的习惯。正如亚里士多德所说，卓越是通过重复不断的练习而达到的。然而，构建高质量的软件是一件非常困难的工作。即便是最基本的程序里也有会bug。每个人（包括最好的工程师）都会写出有bug的代码来。而人类本身就是不完美的，我们经常会犯错。因此当我们将人类语言翻译成按照我们的指令来工作的软件应用程序时，它一定会出错，一定会有意外发生，所以出现质量问题也就没什么好奇怪的了。但是，仅仅把软件质量的责任归咎于程序员是不公平的。软件工程是一个有很多来自不同领域的人参与的过程。例如在微软，工程团队通常会分成三块：程序管理、开发以及测试。程序管理保证了产品的设计规范是准确的，经过充分考量的，并且他们会明确最终产品所要达到的质量目标。程序员负责创建最有效以及最灵活的设计，保证算法的准确性，以及在代码实现里应用各种最佳实践。测试人员考虑代码和应用程序行为的每一种可能的组合，并且完整检查软件的每一条代码路径。质量属于工程生命周期里的每一个环节，因此每一位参与者都对它负有责任。软件开发组织对此有着深刻的理解，并且花费了大量的时间和精力来实现各种过程和程序以确保每一位团队成员都能专注于质量的保证上。软件开发过程和方法论自20世纪90年代末就在业界存在了。随着计算能力的增加，软件程序变得越来越复杂。而在软件开发里增加的复杂度，再加上业界相对的不成熟，导致了软件项目里的诸多问题，这包括了成本超支，缺乏形式化的质量保证过程而导致糟糕的软件质量，以及代码的低可维护性等。因此，诞生了形式化的软件开发过程。它们的主要目标是把一系列的任务放到形式化的结构里，最终能让开发工作产生更好的结果。简单来说，采用形式化的工程流程的目的就是为了产生更优质的产品，让市场或者部署更加容易对其进行预测。使用过程来保证产品的质量是所有提供产品或者服务的产业进行工作的基础。

《完美代码》

媒体关注与评论

《完美代码》在管理书籍和技术书籍之间做到了出色的平衡。从如何进行软件建模，到安全性设计，再到防御性编程，本书展示了可以改进开发工作的各种最佳实践。　　——Wintellect联合创始人，John Robbins《完美代码》不仅仅是一本关于代码的书，它阐述了如何开发一个健壮的项目。这本书简单明了地介绍了软件开发中的最佳实践，并提供了很多实际产品中的案例和经验教训，帮助读者构建完美的项目——从设计到开发，以及最后的发布和维护。　　——微软.软件开发工程师，Jason Blankman作为一名有20年经验的软件工程师，能让我每过几年就重读一遍的书并不多见，而《完美代码》就是其中之一，每次温故都能知新。　　——微软,软件开发工程师，Don Reamey对任何专业软件工程师来说，《完美代码》的价值都是无法衡量的，书中到处都是可以立刻投入使用的实践经验。《完美代码》绝对是一本让你爱不释手的必读书籍。　　——ALL Software执行股东，微软区域总裁，John Alexander《完美代码》对任何rr从业人员来说都是必读书籍，特别是如果你打算使用托管代码的话。它不仅涵盖了工程上的最佳实践，还通过已经过实践检验的案例来展示它们。　　——微软,发布经理，Andres Juarez这本书提供了在高效软件开发过程中的最佳实践，因此可以避免很多典型的程序员错误。作者提供了可实践的检测错误的方案，并解释了微软是如何进行软件开发和测试的。——微软，测试经理，Venkat B.Iyer无论你是新手还是专家，任何级别的程序员都可以阅读本书。它为优秀的开发实践提供了坚实的基础，不管开发团队的规模是大还是小，甚至只有一名程序员，也应该采用书中的经验。　　——独立软件工程师，John Macknight

《完美代码》

编辑推荐

《完美代码》：采用一流的工程实践来帮助你编写更健壮、无错的代码。两位微软的.NET开发专家与你分享优化软件开发生命周期的真实案例和经过实战考验的解决方案——从避免代价昂贵的编程陷阱，到提高开发团队整体效率的方法等。无论你是来自哪个层次的托管代码程序员，都能在这里找到设计、原型开发、实现、调试以及测试的技巧，进一步提升代码的品质。

《完美代码》

精彩短评

1、这本书我本来想给他5星，从软件工程讲到设计，讲到Coding，讲到工具，讲到编码态度。
之所以扣了一星是因为书名为“ Solid Code-Optimizing the software development life cycle ”， 命题很大，作者却只从.net开发工程师的角度去讲述（当然，他本来就是Live开发工程师）。
建议.net的Coder看看
2、无可看性

《完美代码》

精彩书评

1、花了一个晚上的时间，熬到晚上一点多钟将此书泛读了一遍。我挺喜欢这本书的，将纷繁芜杂的软件开发以毫不过时的理念做了全面的介绍，最终的目的就是solid code。通过此书，比较容易找到了自己在软件开发上的短板和未来需要提升的方向，良师益友一类的书。

《完美代码》

版权说明