

<<软件之道>>

图书基本信息

书名：<<软件之道>>

13位ISBN编号：9787115270443

10位ISBN编号：7115270449

出版时间：2012-3

出版时间：人民邮电出版社

作者：Andy Oram, Greg Wilson

页数：438

译者：张玳, 鲍央舟, 沈欢星

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<软件之道>>

内容概要

《软件之道：软件开发争议问题剖析》集合了几十位软件工程领域顶尖研究人员的实证研究，通过呈现他们长达几年甚至几十年的研究成果，揭示了软件开发社区普遍存在的一些确凿事实和虚构之事。

书中探讨了更有效的编程语言，对比了软件开发人员之间的效率差异，验证了康威定理，并反思了软件行业的最新模式。

《软件之道：软件开发争议问题剖析》将帮助读者拓宽视野，更好地选择适合的工具和技术，并最终成为一名更加优秀的软件行业从业人员。

《软件之道：软件开发争议问题剖析》适合所有软件开发人员和研究人员阅读。

作者简介

作者：(美国)欧莱姆(Andy Oram) (美国)威尔逊(Greg Wilson) 译者：鲍央舟 张玳 沈欢星

<<软件之道>>

书籍目录

第一部分 搜寻和使用证据的一般原则

第1章 探寻有力的证据

第2章 可信度，为什么我坚决要求确信的证据

第3章 我们能从系统性评审中学到什么

第4章 用定性研究方法来理解软件工程学

第5章 在实践中学习成长：软件工程实验室中的质量改进范式

第6章 性格、智力和专业技能对软件开发的影响

第7章 为什么学编程这么难

第8章 超越代码行：我们还需要其他的复杂度指标吗

第二部分 软件工程的特有话题

第9章 自动故障预报系统实例一则

第10章 架构设计的程度和时机

第11章 康威推论

第12章 测试驱动开发的效果如何

第13章 为何计算机科学领域的女性不多

第14章 两个关于编程语言的比较

第15章 质量之战：开源软件对战专有软件

第16章 码语者

第17章 结对编程

第18章 现代化代码审查

第19章 公共办公室还是私人办公室

第20章 识别及管理全球性软件开发中的依赖关系

第21章 模块化的效果如何

第22章 设计模式的证据

第23章 循证故障预测

第24章 采集缺陷报告的艺术

第25章 软件的缺陷都从哪儿来

第26章 新手专家：软件行业的应届毕业生们

第27章 挖掘你自己的证据

第28章 正当使用“复制-粘贴”大法

第29章 你的API有多好用

第30章 “10倍”意味着什么？

编程生产力的差距测量

撰稿人

章节摘录

版权页：插图：1.2 当今证据的状态回顾以往，我们现在才知道当时对有力证据的定义是多么幼稚。精确、统计性强、重复证据，这些都最终被证明比我们想的要难找得多。

另外，它们并不能满足与研究更相关的目标。

1.2.1 精确性研究的挑战 我们发现精确的研究可能对那些接受过足够科研培训的人来说很有说服力，但对那些普通从业者却很难解释。

因为这样的研究通常会有所限制和简化，从而使得研究背景不是以代表真实的开发环境。

比如，Basili和Selby的研究仅针对“迷你”问题运用了研究中的技巧，该问题只在虚拟环境中不超过400行的代码。

这项研究经常被引用，虽然它是被经常复制的对象，但似乎没有一项复制使用了更大规模的或者更具代表性的应用。

虽然这项精确的研究对我们理解铲除代码缺陷的不同方法的优劣有重大的贡献，但如果我们在这一主题上的大部分思维来自相对小的代码段，这并不理想。

1.2.2 统计强度的挑战 什么构成了对现实世界问题的“强有力”统计数字？

对于这个问题的共识出奇得少。

首先，有一个关于外部有效性的问题：经过充分测试的度量方法是否能反映我们所关注的真实世界的现象。

比如，Foss等人证明了常用估计工作量的评估方法根本就是有问题。

<<软件之道>>

媒体关注与评论

“虽然我们自称是‘工程师’，然而编程过程并非机械地由数据驱动，而是更多地取决于编程人员的感受。

以软件开发的大量经验性数据为基础，编程过程完全可以达到个性化与系统化的统一。

” ——Jason Cohen，Smart Bear和WPEngine公司创始人

<<软件之道>>

编辑推荐

《软件之道:软件开发争议问题剖析》编辑推荐：相信大家常常听说某些工具、技术和实践方法可以改进软件开发，但其中哪些说法是可被证实的，哪些仅仅是人们一厢情愿的想法？

《软件之道:软件开发争议问题剖析》收录了Steve McConnell、Barry Boehm和Barbara Kitchenham等几十位软件工程领域顶尖研究人员的文章，深入讨论了软件开发社区中常见的一些观点，一些是确凿事实，一些则是荒诞说法。

他们的深刻见解定会让你大开眼界。

某些编程人员的工作成效果真是他人十倍之多？

测试驱动的开发果真能帮助更快、更好地开发代码？

软件的bug数量果真可以利用代码度量进行预测？

设计模式果真有助于构建更好的应用程序？

人员个性会对结对编程产生何种影响？

地理位置的距离和公司职位的差距，究竟何者影响更大。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>