

软件 工程 系列



Facts and Fallacies of Software Engineering

软件工程的事实 与谬误

[美] Robert L. Glass 著
Alan M. Davis 序
严亚军 龚波 译

Robert L. Glass



中国电力出版社

www.infopower.com.cn

软 件 工 程 系 列

Facts and Fallacies of Software Engineering

软件工程的事实 与谬误

[美] Robert L. Glass 著
Alan M. Davis 序
严亚军 龚波 译



中国电力出版社

www.infopower.com.cn

Facts and Fallacies of Software Engineering (ISBN 0-321-11742-5)

Robert L. Glass

Copyright ©2003 by Pearson Education, Inc.

Original English Language Edition Published by Addison-Wesley

All rights reserved.

Translation edition published by PEARSON EDUCATION ASIA LTD and CHINA ELECTRIC
POWER PRESS, Copyright © 2006.

本书翻译版由 Pearson Education 授权中国电力出版社独家出版、发行。
未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签，无标签者不得销售。

北京市版权局著作权合同登记号 图字：01-2003-2434 号

图书在版编目 (CIP) 数据

软件工程的事实与谬误 / (美) 格拉斯 (Glass, R.L.) 编著; 严亚军, 龚波译.

—北京: 中国电力出版社, 2005

(软件工程系列)

ISBN 7-5083-3142-7

I. 软... II. ①格...②严...③龚... III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2004) 第 143171 号

丛 书 名: 软件工程系列

书 名: 软件工程的事实与谬误

编 著: (美) Robert L. Glass

翻 译: 严亚军 龚波

审 校: 涂艳

责任编辑: 牛贵华

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 88515918 传 真: (010) 88518169

印 刷: 北京丰源印刷厂

开本尺寸: 185×233 1/16 印 张: 10.25 字 数: 178 千字

书 号: ISBN 7-5083-3142-7

版 次: 2006 年 1 月北京第 1 版 2006 年 1 月第 1 次印刷

定 价: 23.00 元

版权所有 翻印必究

谨以本书献给为软件工程带来火花的研究者

和使火继续燃烧的实践者

译者序

本书抽取了软件行业中经典的 55 个基本事实和 10 个谬误进行分析，对于每个观点，作者都提供了观点的来源和证据以及相关的参考文献。除此之外，本书还介绍了业界对这个观点的其他不同认识和理解。

本书的内容和组织方式非常有意思、有特点，市面上绝无类似的书籍，因为一般只有专业论文才采取这样的方式陈述观点。作者的行业经验和性格决定了本书的价值。它可能会让你拍案惊奇，也可能会让你认为有点狂妄，从而进一步促使你思考这些观点，跟同事和同仁进行辩论和分析，而这恰恰就是作者希望达到的目的。

阅读本书总会有一些让你感慨万千的地方，我觉得市面上能够做到这点的书并不多。对于这些行业观点，我们很少收集和总结，这恰恰就是我们的缺陷。你可以在本书基础上进一步扩展和完善你认为有价值的事实和谬误，进而形成自己的经验库。经常翻阅它，提醒自己不要忘记那些最基本的行业规律，规避那些前人已经明确指出来的“陷阱”。

作者列举的事实和谬误，有的令我很惊诧，确实是这样吗？说实话，本书的一些观点确实对我触动很大，有助于我定位一些更新的研究目标。总之，我喜欢这本书，喜欢作者的坦率。

原书的句子比较晦涩，涉及的领域知识比较广泛，我们竭力保证译文的正确、通顺和优雅，但错误之处在所难免，恳请广大读者多加指正。

本书主要由严亚军和龚波翻译，由龚波统稿。李红玲、马丽、李志、邓波等在本书的翻译过程中都提供了很多指导性意见和建议，在此表示感谢！同时还要感谢高军老师的信任，以及编辑老师的辛勤工作！

龚波
2005 年

致 谢

感谢 Paul Becker。

他目前在 Addison-Wesley 工作，几乎编辑过我参与编写的所有公开出版物，感谢他多年来对我的信任。

感谢 Karl Wieggers。

感谢他经常帮助我回忆那些容易被遗忘的基本事实，同时感谢他花费大量精力来审阅和修订本书。

感谢 James Bach、Vic Basili、Dave Card、Al Davis、Tom DeMarco、Yaacov Fenster、Shari Lawrence Pfleeger、Dennis Taylor 和 Scott Woodfield。

感谢他们帮助我确定本书很多事实和谬误的来源和引用之处。

序

初次听到 Bob Glass 准备参照我的《201 Principles of Software Development》来写这本书时，我有一点担心。毕竟，Bob 是该行业中最优秀的作者之一，他的书将对我的书形成激烈竞争。当 Bob 邀请我写本书的序时，我更加担心了，我怎么可以为了一本与自己的书直接竞争的书作序呢？等读完这本书后，我对于有机会为本书作序感到欣喜和荣幸（而不再担心）。

目前的软件行业与 19 世纪晚期的制药行业处于相同的阶段。似乎在我们当中，蛇油销售员和预言者比精明的参与者和讲道理者多。我们每天都会听说有人新发现了治疗某种不治之症的药物。同样，我们常常听到某些快速方法可以解决低效率、低质量、客户不满意、沟通不畅、需求变更、无效测试以及糟糕的管理等问题。不负责任的“博学者”实在是太多了，以至于我们有时候会怀疑那些所谓的万能药。我们该去问谁？在这个行业中我们该信任谁？我们在哪里能找到真理？答案是 Bob Glass。

多年来，Bob 曾经为我们提供了许多有关软件灾难的短篇论文。我一直期待着他讨论这些灾难的共性，以便我们可以从他的经验中受益。在现在这本精彩的书中，Bob Glass 讨论的 55 个事实不仅仅是个人的推想和认识。这些事实正是我所期待的：作者通过详细查阅自己过去发表的文章中的数百个案例，从中精炼出这些精妙的结论。

可能并非所有的读者都喜欢后面的这 55 个事实，其中有些事实与某些所谓的现代方法截然相反。对于忽略这些建议的那些人，我只能祝你旅途愉快，但是我真替你们担忧。你们正走向一条前途险恶的路，其中荆棘满布，许多人会身陷其中，不得不结束自己的职业生涯，我建议你最好去读 Bob 有关软件灾难的早期书籍。对于那些接受这些建议的人，你们正走向另一条前途光明的路，在这条路上充满了成功的实例，这是一条富有知性和知识的道路。应该信任 Bob Glass，因为这是他曾经的经历。他有权在分析他人的数百个成功或失败案例的同时，分析自己的成功和失败。站在他的肩上，你在该行业的成功机率就更大。忽视他的建议，

那你就等着 Bob 在多年之后再打电话询问你的项目——他会将此收入下一版的软件灾难故事之中。

Alan M. Davis

2002 年春

作者附言：

我尽力劝 Al Davis 低调写这个序，毕竟这有点夸张。但是他拒绝了我所有的劝告（我确实努力劝过他）。实际上，在一次交流中，他说：“你值得受尊重，我来帮助你！”我的经验告诉我：捧得越高，摔得越狠。

无论如何，我都想像不出比 Al 所送给我的更宽厚、更精彩的评价。多谢！

Robert L. Glass

2002 年夏

目 录

译者序
致谢
序

第1部分 55个事实

简介	3
第1章 管理	7
1.1 人员	8
1.2 工具和技术	16
1.3 估算	22
1.4 复用	36
1.5 复杂性	48
第2章 生命周期	53
2.1 需求	55
2.2 设计	62
2.3 编码	69
2.4 错误消除	73
2.5 测试	75
2.6 评审和检查	85
2.7 维护	94
第3章 质量	104
3.1 质量	105
3.2 可靠性	109
3.3 效率	114

第 4 章 研究	119
----------------	-----

第 2 部分 5+5 谬误

简介	125
----------	-----

第 5 章 管理	127
----------------	-----

5.1 人员	131
--------------	-----

5.2 工具和技术	132
-----------------	-----

5.3 估算	137
--------------	-----

第 6 章 生命周期	140
------------------	-----

6.1 测试	140
--------------	-----

6.2 评审	142
--------------	-----

6.3 维护	145
--------------	-----

第 7 章 教育	148
----------------	-----

结论	151
----------	-----

关于作者	153
------------	-----

第 1 部分

55 个事实



简介

本书包含很多有关软件工程的事实与谬误。

听起来很无聊，难道不是吗？一个有关软件构建方面事实和谬误的列表听起来可不像足球那样讨人喜欢，不值得花费几个小时。但是，这些事实和谬误有一些特别之处，那就是它们是基础性的，而更重要的是它们经常被人遗忘。实际上，这就是本书的中心主题。关于构建软件，我们理应知道许多东西，但是实际上却因为这样或那样的原因不知道，甚至我们一些想当然的看法实际上是显而易见的错误。

上一段中的我们指的是谁？当然是指构建软件的人。我们似乎需要反复多次接受同样的教训，但如果我们能牢记这些事实，就可以避免那些教训。同时，这里的我们还指那些从事软件研究的人。有些研究人员太局限于理论方面，以至于忽视了一些重要的基本事实，而这些事实甚至可以推倒他们自己创建的理论。

因此，本书读者包括对构建软件感兴趣的所有人，具体指软件专业人士（包括技术人员和管理人员）、学生、教师和研究人員。我认为，不谦虚地说，本书适合所有从事计算机行业的人。

本书最初的书名很冗长，含 13 个单词：**Fifty-Five Frequently Forgotten Fundament Facts (and a Few Fallacies) about Software Engineering** [有关软件工程的 55 个容易忘记的基本事实（和部分谬误）]。这个名字相当繁琐，至少本书的策划者认为繁琐。当今，很多书籍都使用非常“酷”的书名来增加卖点。出版商和我最后商定采用 **Facts and Fallacies of Software Engineering**（软件工程的事实与谬误）。这个名字简洁、清晰，又非常朴素。

我曾尝试过将最初的长书名昵称为 **F-Book**，仅仅是为了与书名中所有以 **F** 开头的单词压韵。但是出版商反对这样做，我想他是正确的。**F** 是字母表中的“下等”字母（有人倡导使用 **H** 和 **D**，但是很少有人推崇使用 **F**），所以就没有叫 **F-Book**（我早期曾经写过一本有关构造编译器的书，仅仅因为自己随意在封面放了一个龙的图片，就称之为 **Dragon Book**。这次，我没有沿袭那种习惯）。

但是我想做如下辩解：这些以 F 开头的单词都是有意义的，对于理解书名的含义很重要。当然，55 只是一个数字，我提供 55 个事实，目的是使书名更押韵（我打赌，在 Alan Davis 那本优秀著作中，也是先确定了数字 201，然后才寻找软件工程中相应的 201 个原则的）。但是其他以 F 开头的单词都是精挑细选的。

frequently forgotten（经常被忘记）？因为其中许多事实确实经常被忘记。读本书时有读者可能会说：“哦，我想起来了。”你应该反思为什么这么多年都忘记了它。

fundamental（基本的）？选择所介绍事实的首要原则是它们在软件领域中至关重要。我们可能忘记了其中许多事实，但这不是说它们不重要。如果你仍拿不准是否继续读本书，那么我给你一个理由，我坚信：在这组事实中，你会发现软件工程领域中最基本的重要知识。

facts（事实）？奇怪，这可能是书名中最具争议的单词。对于我所选的事实，你可能并非完全同意。你甚至会强烈反对其中的一些。我个人看来它们都是事实，但是你也可以不这样看。

a few fallacies（一些谬误）？在软件领域，有许多“神圣”的歪论，我忍不住要揪出来。我必须承认，自己所谓的谬误在别人看来可能是事实。但是在读这本书的过程中，你应该对于我所谓的事实和谬误形成自己的观点。

这些事实和谬误是否已经过时？本书的一位审校者认为有些内容过时了。他的指责让我羞愧。但是，那些经常被忘记的事实和谬误应当持续一段时间。在这个组合中有许多闪闪发光的经典之物。但是，我相信你在其中一定会找到一些惊奇的新观点，因为你并不熟悉它们。这些事实和谬误的关键不在于它们是否陈旧，而是对于你来说是否是新的。

在本书的这一部分，我将简要介绍这些事实。关于谬误将在本书后一部分中介绍。在这一部分，我们将浏览这 55 个经常被忘记的基本事实，看其中有多少与这些以 F 开头的单词有关。客观地说，其中有一些事实并没有被忘记。

- 其中的 12 个事实鲜为人知。许多人从未听说过它们，自然无所谓忘记，但是我敢断言，它们都绝对重要。
- 其中的 11 个事实被广泛接受，但是似乎没有人按其行事。
- 其中的 8 个事实也被接受，但是我们不知道如何（或者是否）确定它们所反映的问题。
- 其中的 6 个事实可能被绝大多数人安全接受，没有争议，也很少被忘记。

- 其中的 5 个事实被许多人旗帜鲜明地反对。
- 其中的 5 个事实被许多人接受，但是有一些人强烈反对，所以相当有争议。

这些加起来不等于 55，因为 (a) 有些事实适合多个种类；(b) 还有一些事实属于其他的类别，例如“只有供应商会反对”。我不具体告诉你哪些事实归于哪一类，我想你会自己做出判断。

你将看到本书中有大量的争议。为了帮助你理解，我在每个事实的讨论之后列举了相关的争议。我希望无论你的观点是否吻合，这样做都能包含你的观点，你可以找出相似之处。

即使有这么多的争议，我还是明智地告诉你：我自己对于选择这些事实和谬误非常自信。（在本书的后面有一个幽默的自传，因此这里会讲快一点。）我作为技术实践者和研究者，已经在软件工程领域奋斗了 45 年。在这一领域，我已经写了 25 本书和超过 75 篇专业论文。我在 3 个最主要的期刊上有定期的专栏文章，分别是：在《Communications of the ACM》的 Practical Programmer、在《IEEE Software》的 Loyal Opposition 和在 ACM SIGMIS 的《DATABASE》的 Through a Glass, Darkly。我知道自己是一个批判者，很荣幸地被称为“最爱发脾气的老家伙”就证明了这一点。你能期待我质疑那些无可挑剔的东西，正如我前面所说的，要揪出“神圣”的歪论。

对于这些事实，我还有一点要说。我已经说过这些事实都经过仔细筛选，以确保它们都是本领域中的基本问题。但是，它们当中到底多少事实被忘记了，从这一点可以反映出我们没有利用的知识量。项目管理者会说其中的许多事实已经被忘记或者从来没听说过。软件开发者因缺乏对这些事实与谬误知识的了解而工作于受限的领域，研究者们认为如果把这些事实与谬误考虑进来，所实现的东西将是荒诞可笑的。我深信那些决定继续读下去的人必须有丰富的学习经验或者极好的记忆力。

现在，在开始讨论这些事实之前，我想提出几点重要的希望。我在阐述这些事实的同时，还提出了该领域的一些问题。我并不想在此给出这些问题的答案。这本书只讨论是什么，而不讨论怎么办。我认为有一点很重要：我只想将这些事实公之于众，这样大家就可以自由讨论并付诸行动。我认为这个目标已经相当重要，没有必要讨论解决方法，否则会使主题模糊。这些事实所带来的问题的解决方法通常见于已出版的专业性书籍和论文中：软件工程教科书、软件工程专题书、

主要软件工程学术期刊和受欢迎的学术杂志（虽然其中许多内容良莠不齐、鱼龙混杂）。

为了便于你阅读，我按照下面的结构来陈述这些事实：

- 首先讨论一个事实。
- 然后提出围绕这一事实的争议。
- 最后提出有关这一事实的信息来源，以及有关背景和前景信息的参考文献。根据软件工程的标准（它们也是经常被忘记的事实），有些文献年代较为久远，有些非常新颖，有些则兼而有之。

我将这 55 个事实分为几类，分别是：

- 管理
- 生命周期
- 质量
- 研究

我将谬误也进行类似的分类，分别是：

- 管理
- 生命周期
- 教育

好了，我们已经做好足够的准备，希望你会喜欢我所提供的这些事实和谬误。更重要的是，我希望它们会对你有用。

Robert L. Glass

2002 年夏

管 理

说实话，我一直认为管理是很烦人的话题。在我读过的有关管理的书籍中，95%都是常识，其余5%是重温陈词滥调。那么在本书中，我为何要首先讨论管理这一话题？

这是因为平心而论，在软件领域中的许多高层次、显而易见的问题与管理有关。例如，我们将很多失败都归因于管理，而许多成功也可以归因于管理。1995年 Al Davis 在有关软件原则的一本精彩著作《201 Principles of Software Development》的第127条原则中说到：“好的管理比好的技术更重要。”虽然我不愿意承认，但是 Al 是对的。

为什么我不愿意承认？在我早期的职业生涯中，面临着不可避免的选择：我可以继续当一个技术人员，继续做我喜欢做的软件开发，我还可以选择另一条道路——成为一个管理者。我认为这个选择非常难。美国的软件业成功之路需要改善产业结构，如果没有优秀的管理方法会很难。但是后来有两个原因使我意识到自己不会把技术抛在脑后。

1. 我愿意实干，而不愿意指挥别人去做。
2. 我喜欢自己做出决定，而不愿意成为夹在中间的管理人员，传达上司的决定。

你也许觉得后一个原因很奇怪。技术人员怎么可能比自己的管理者拥有更大的自由，我的经验告诉我确实如此，但是这对别人很难解释得通。关于这个问题我后来写了一本书，即《The Power of Peonage》（即《雇佣者的力量》，1979）。那本书的主旨——也是让我继续做技术人员的信念——是处于管理最底层且专长于干活的人通常比其他任何人有更大的威力。他们不会降级，通常不会有比他们更低的职位。也许有别的方式可以惩罚一个优秀的技术人员，但是绝不会是降级处理。在我的技术生涯中，我不止一次使用这种威力。