

软件项目管理世界经典教材丛书·影印版

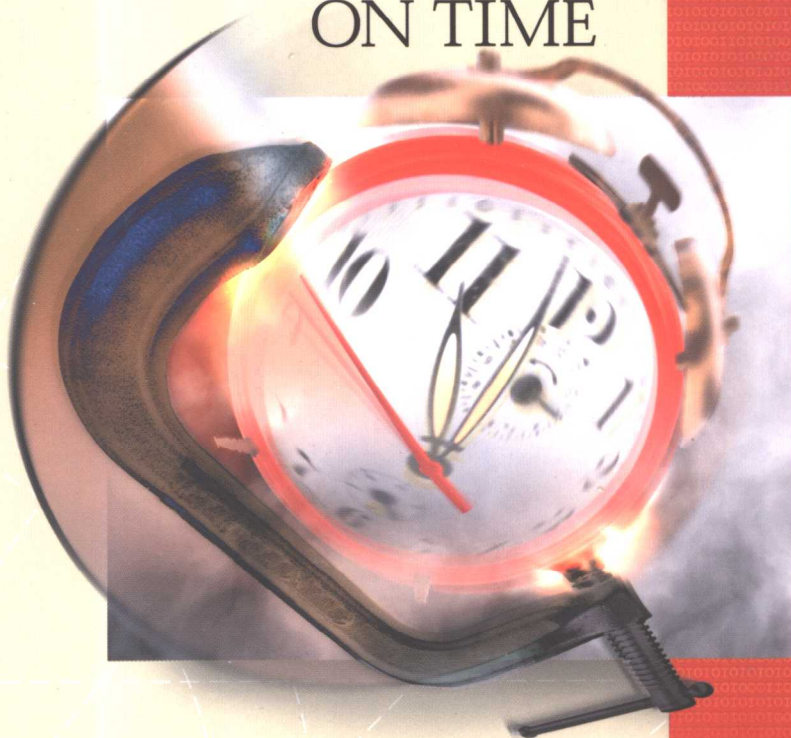
Microsoft Press

压力与时限

软件项目管理

UNDER
PRESSURE
AND
ON TIME

建立优秀团队与
开发一流软件
的实用策略
与成熟模式



Ed Sullivan
John Robbins 作序

✓ Best
Practices



清华大学出版社

软件项目管理世界经典教材丛书

压力与时限

—— 软件项目管理

(影印版)

[美] Ed Sullivan 著
John Robbins 作序

清华大学出版社
北 京

内 容 简 介

本书介绍在不断发展壮大的环境中开发优秀商业软件时应该遵循的原则、过程和技术。本书将介绍对发布高质量的软件来说最为关键与基本的原则。本书比其他项目管理类书籍更加深入的阐述了经过实践检验的经验与细节,它们都来自于实际工作,这些实践经验会有助于组织优秀的开发团队并带领他们开发出一流的软件。

本书面向在软件工程项目中作为主管的读者,本书也适用于欲从事软件项目管理的读者。

Under Pressure and On Time.

Ed Sullivan

Copyright © 2002 by Microsoft Corporation

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation

All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the Publisher. For sale in the People's Republic of China only.

本书影印版由 Microsoft Press 授权清华大学出版社在中国境内(香港、澳门特别行政区和台湾地区除外)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。

北京市版权局著作权合同登记号:图 01-2003-0831 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

压力与时限=Under Pressure and On Time: 软件项目管理/(美)沙利文著.一影印本.

—北京:清华大学出版社,2003

(软件项目管理世界经典教材丛书)

ISBN 7-302-06433-4

I.压... II.沙... III.软件开发—项目管理—教材—英文 IV.TP311.52

中图版本图书馆 CIP 数据核字(2003)第 018197 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

<http://www.tup.tsinghua.edu.cn>

责任编辑: 赵飞虎

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张:18

版 次: 2003 年 4 月第 1 版 2003 年 5 月第 2 次印刷

书 号: ISBN 7-302-06433-4/TP·4848

印 数: 2001~1000

定 价: 33.00 元

序

首先必须说明此书与其他书不同。现在有很多关于项目管理的书，然而《压力与时限》这本书实际上针对处于发展初期的公司来讨论软件交付的问题。交付对这类公司来说是最重要的事情，这也是本书的中心内容。这本书的特别之处在于，本书作者并不只是幸运的交付了一个产品。Ed Sullivan 交付了一些优良的产品，然而更重要的是，他揭示了项目管理成功的秘密，这些管理成功的项目交付产品的数量比大多数软件开发组织所期望的要多。在这个行业里，不断扩展的项目很难获得成功，然而 Ed 成功了很多次，本书将介绍使他获得成功的技术。

在软件行业中，项目管理变得比以往更加重要，这是因为在此领域工作的人要遵循 Internet 时代的开发进度表，这样就促使开发团队在自己的项目管理中寻找捷径，而这种捷径最终导致比以往更加背离进度表以及更多的产品错误。这种问题特别容易出现在比较小的软件公司，这些公司急于从较小的市场份额中获得利润。这种问题在大公司也比较普遍，他们正在从事科研性质的项目，期望以此来解决这个问题。

不要认为出现这种情况的原因是项目管理没有跟上进度，其实问题比预想的要严重。世界各地的计算机科学课程中没有实用的项目管理技巧，许多地方根本不教授这种课程。因此，很多开发人员自学或从已经定型的经理中学习经验。编写代码的工作应该只占项目的 20%。

但是很多公司达不到这个要求。所幸本书能为编写代码以外的工作提供帮助，这种帮助只有本书能提供。本书并不介绍项目管理的理论，它只是介绍在一个获得巨大成功的创业公司中所做的工作。

本书特别强调团队。很多公司的组成结构为工程部门、质量保证部门、可能还有文档部门。这种是由个人组成的集体共同开发产品(最好不要将他们称为产品开发团队)，而各个部门有各自的工作方式。因此，许多公司由于这种糟糕组织结构从一开始就注定要失败。Ed 则介绍了在 NuMega 中实际的团队组织情况，在那里编码人员、质量保证人员、人事与客户培训工作人员都只对一个项目经理负责。即使当 NuMega 逐步壮大而应该按照传统方式进行组织时，Ed 仍旧力争使产品开发团队保持原来的状态。由于每个员工必须只对一个产品负责，因此 NuMega 能够避免很多人为的问题，公司通常都会因为自身原因出现这类问题。这样做的另一个好处是能够让产品开发团队中的每个人看到其他人的工作对成功交付产品来说是如何的重要。这与常见的其他公司实行的开发形式有显著区别，这种公司通常将员工分为开发人员与质量保证人员。

当然，如此组织团队肯定很辛苦费力。现在关于软件业的新闻报道几乎都会提及公司总是短缺员工。如果不能找到合适的员工，那么要建立优秀团队的所有希望就会成为泡影。NuMega 用非传统方式把开发人员组织成公司的做法比较特别。Ed 说如何找到并留住优秀的人员是 NuMega 的机密，它使公司能够高效运行并将优秀的产品及时推向市场。Ed 作为一个主管在软件企业所做的其他特别的

工作是，一旦找到优秀的员工，他就会采取各种方式让他们各尽所能的工作。有时，项目经理会离开 Ed 4、5 天。Ed 让项目经理们通过所犯的错误进行学习，然而更重要的是，他也为他们所学到的东西付酬劳。

一旦具备了合适的组织结构与优秀的人员，就能够很容易地开展项目管理工作。正如 Ed 在本书所述，重要的是要保持平衡，即管理要适当，不能太多也不能太少。他在项目规划和项目执行部分中介绍如何成功的指导项目经理。要知道，成功的及时交付产品并不容易。这一部分内容是从很多错误、令人恼火并令人惊愕的事情中总结出来的经验教训。

由于我也参与了一些令人惊愕的事情，因此从那些经历中知道本书内容的价值远比它的定价高。我熟悉书中提到的每件事情。当 Ed 在作项目经理时，我在 BoundsChecker 3.0 开发小组当工程师。当 Ed 成为 NuMega 的工程主管时，我也最终成为 TrueTime 与 TrueCoverage 产品的项目经理。如果没有 Ed 在本书中介绍的技术，那么 TrueTime 和 TrueCoverage 开发小组就不可能成功的交付产品。

我很嫉妒读者，因为他们不用在让人喘不过气的进度压力下学习所有主要的技巧。然而，我更加幸运，因为我能从主管那里学习本领。Ed 使我从工程新手成为成熟的工程师与项目经理(没想到我能够成为)，他现在正在为公司作全面的调试与项目管理工作。他是我所遇到的最好的老板，我还未曾对他表示感谢。Ed 总是开玩笑说他知道怎样能使我们这些开发人员诚实。我从 Ed 那里学到了成功工程师的经验。

当 Ed 最初让我为此书作序时，我很吃惊。这是很高的荣誉。他说是因为我所犯过的错误才使他想起要写这本书的。我感到十分的高兴与惭愧。在 NuMega 工作时，我的最大疑问是，“你这家伙在 NuMega 是如何交付如此优秀的软件？”现在我可以从 Ed 的书中找到我需要的答案。现在令我激动的是，每个人都有机会看到作为业界最优秀的项目经理之一，Ed 是如何一次次地为公司交付一些最重要的产品。

John Robbins

Hollis, New Hampshire

2001 年 1 月

前 言

本书作者在 1994 年夏天加入 NuMega Technologies 有限公司，他是 BoundsChecker 的开发经理与产品推广经理，这是一个用于检测 Windows 软件错误的软件。当时 NuMega 仅有 14 人，除了两位创立者外，公司还有 3 个工程师、1 个首席运营官、4 个销售人员、1 个办公室管理人员、1 个发货职员、3 个支持工程师和 1 个市场联络经理。这是一个小公司，成立时用的是另外一个名字。每个人要干许多工作并要对很多事情负责。即使该公司比较小，可是他们有宏伟的发展计划并且对未来充满希望。他们开发出了非常优秀的软件，并且组织了一支优秀的研发队伍，致力于开发世界上最好的开发工具。

4 年后，在创立者和一流管理团队的良好管理下，公司增长到了 150 人。本书作者伴随公司而成长，最终成为工程主管。在头 4 年，公司交付了 6 个主要产品，其中有两个新产品 BoundsChecker 和 SoftCE，自己研发了 4 个新产品。几乎所有产品都按时交付了。收入迅速增长，利润很高。实际上这些产品获得很多好评，它们荣获下面的工业奖：

2000

- **DevPartner® for Visual Basic** 获得 Vbxtra 的“Best component or utility for ensuring software quality”奖励。
- **BoundsChecker® VC++ Edition** 获得 Programmer's Paradise 中“Best Selling Testing and Debugging Tool”类别的“Riding the Crest Award”奖励。
- **CodeReview™ 6.1** 获得 VBPI “Reader's Choice”奖励。
- **FailSafe™ 5.21** 获得 VBPI “Reader's Choice”优质奖。

1999

- **DevPartner Studio** 在 Software Development Magazine 获得 Jolt Cola 的“Award for Product Excellence”奖励。
- **DevPartner Studio** 获得 Software Development 5 Star 级。
- **DevPartner for Java™** 获得 Java Developer's Journal 的“JDJ World Class Award”奖励。
- **SmartCheck® 6.01** 获得 VBPI 的“Reader's Choice”奖。
- **CodeReview 6.1** 获得 VBPI 的“Reader's Choice”奖。
- **DevPartner 6.1 for Visual Basic** 获得 VBPI “Reader's Choice”优质奖。
- **TrueTime®** 获得 VBPI 的“Reader's Choice”优质奖。

1998

- **DevPartner for Visual Basic** 获得 Vbxtras 的“Thunderbolt”奖励。

- **SmartCheck** 和 **TrueTime** 获 Visual Basic Programmer's Journal 的“Editor's Choice”奖。
- **SmartCheck** 和 **TrueTime** 获 Visual Basic Programmer's Journal 的“Reader's Choice”奖。
- **DevPartner for Visual Basic** 在 Software Development 中获得 Jolt Cola 的“Productivity Award”奖。

1997

- **BoundsChecker®** 在 Software Development 中获得“Jolt Hall of Fame”奖，这是由于它已经获得很多奖项。
- **TrueTime® Visual Basic Edition** 获得 BYTE 的 Comdex “Best of Show.” 奖励。
- **SmartCheck®** 在 PC Week 获得“Analyst Choice”奖励。
- **BoundsChecker** 1997 年在 Visual Basic Programmer's Journal 连续两年获得“Reader's Choice”奖励。

1996

- **CodeReview™** 作为 1996 年最优秀的软件开发工具之一获得 Windows Tech Journal 的“Star Tech”奖。
- **BoundsChecker** 在很多竞争对手中获得 PC Magazine 的“Editors Choice”奖励。
- **BoundsChecker for Windows NT** 获得 PC Magazine 1996 年的“Technical Excellence Award for Best Development Tool”奖励。
- **BoundsChecker** 在 Software Development West '96 Conference 获得 Jolt Cola 的“Productivity Award for Product Excellence”奖励。
- **BoundsChecker** Visual Basic Programmers Journal 获得 1996 年的“Reader's Choice”奖励。

1995

- **BoundsChecker** 作为 1995 年最优秀的软件开发工具之一获得 Windows Tech Journal 的“Star Tech”奖励。
- **BoundsChecker for Windows NT** 在 Software Development West '95 Conference 上获得 Jolt Cola 的“Award for Product Excellence”奖励。

1994

- **BoundsChecker for Windows** 在 Software Development '94 Conference 上获得 Jolt Cola 的“Award for Product Excellence”奖励。

上面这些荣誉令人感到非常愉快，不是吗？然而，公司面临与其他软件开发团队同样的问题：冲突的目标、进度压力、无效资源、枯竭和沟通等问题。公司必须及时发布高质量的软件，否则就会被淘汰出局。

公司的工程开发团队所面临的障碍也是现今软件工业存在最广泛的问题，即开发团队与项目

的无效管理。特别是在建立公司时每天都会遇到此问题，一定要解决它否则就会走向破产。对 NuMega Technologies 公司来说，解决此问题的能力是成功的最重要因素。

本书的目标是与读者分享在持续繁荣向上的环境中开发优秀商业软件时所获得的原则、过程和技术。本书将介绍对发布高质量的软件来说最为关键与本质的原则。本书不讲解理论或讨论很多可能会被尝试的事情，而是讲述在刚起步的商业环境中的成功经验。作者将介绍他在 MuMega 的经验，即他们如何在一个刚起步的公司中壮大工程团队以及开发产品。

作者认为刚起步的开发环境与从最小到中等的开发项目(30 人或更少)差别不大。不管是在小公司还是大公司、独立软件开发商还是 IT 商店，并且不论是汇编语言还是 Web 开发人员，这些都需要及时的交付优秀软件。人们想要在压力下处理相同的问题并想取得同样的成功，那么究竟什么时候开发团队才不用面对冲突的目标、进度问题、无效的资源以及所有这些在软件开发生命周期中出现的典型问题呢？

请注意本书并不详细讨论任何课题。其他的书深入详细地探讨了特定的课题，例如招聘人员、发布工程、质量保证、人为因素等。

如何使用本书

尽管本书内容不是适用于每个软件开发小组或软件公司，但是本书中的许多信息对很多组织结构与项目都非常有用。应该自己从项目中尽可能的总结经验。本书内容并不是开发软件的惟一方法，但却是经过验证的成功方法。

读者对象

本书面向在软件工程项目中处于(或希望成为)主管位置的读者，读者对象如下所示：

- 高级工程经理(副总经理，主管，项目小组经理)
- 开发经理
- 项目主管
- 开发主管
- 系统设计师
- 程序开发经理
- 技术写作经理
- 技术写作主管
- 质量经理

- 质量主管
- 可用性经理
- 可用性主管
- 发布经理
- 发布工程主管

作为开发团队的成员也应该阅读本书。本书介绍了整个团队在软件开发中所要考虑的问题，而不仅仅是经理或主管。整个团队要作为一个整体进行工作这一点非常重要，团队应具备一致的概念、态度、信心与文化。

本书结构

本书分为 3 个部分。每个部分介绍软件工程管理的一个重要方面。

第 I 部分：人员、组织与方法

在开始规划项目或编写软件之前，必须确保做好了基本的准备工作。这就需要合适的人员、恰当的组织结构与正确的程序才能有效的进行。如果这些条件不具备，当项目进行速度加快并且压力增加时就不可能按照计划工作，最终会导致项目崩溃。在第 I 部分中介绍了任何快速开发项目的所需的基本条件，其中包括：

- **人员** 需要谁，怎样找到并留住他们。
- **组织** 开发团队中每个成员的基本角色与责任。
- **工具** 什么是关键的开发工具，如何使用它们。
- **质量** 怎样在开发过程中同时执行质量保证。
- **发布工作** 怎样在整个开发周期中保持项目的完整性与可用性。

第 II 部分：项目定义与规划

如果想要及时地交付软件，就应该在创建它时知道想要生成什么以及怎样生成。即使最聪明的人也需要了解他们要做什么、要使用什么技术以及产品完成后的样子。因此需要下面的定义：

- 项目的基本要求。
- 项目的技术。
- 项目的使用模式。

具备这些条件后就可以制定进度表了，此进度表会使员工与领导人员明确项目的目标。

这样就能确定，在某种程度上来说是肯定，现在拥有一个项目的可实现进度表。

这 4 个主题，即要求、技术、可用性以及进度表是紧密相关的，因此如果想要成功的管理项目就必须同时对它们进行处理。如果没有它们就会猜测、假设或忽略项目中的关键因素，这样就会带来不可预知的风险，这些风险通常会给进度表带来问题并且会使进度严重背离进度表。要记住，许多重大错误都是在项目的头几个星期产生的，即在规划阶段。

第 III 部分：项目执行

一旦完成了规划就要准备制作产品了。由于有优秀的员工、可靠的开发过程以及良好的规划，可以很容易按计划完成任务。但现在还需要确保在完成计划时不迷失方向。

第 III 部分主要介绍完成日常产品开发工作的执行模式，其中包括以下主题：

- **执行** 怎样跟踪项目以便尽早发现和修正问题。
- **测试版测试** 怎样从测试板程序中得到真实的反馈信息并同时为测试工作提供参考。
- **发布准备** 怎样管理项目的最终阶段并确保其为交付做好准备。
- **项目完工** 项目完工是什么，为什么需要它，以及怎样做。

附 加 内 容

在每章的末尾列出了使用本书时通常会遇到的疑问和问题。几乎所有的疑问和问题都来自于实践经验，因此希望这些解决方案能对解决实际问题有所帮助。

本书中突出表示的 **Back at work** 栏目介绍在 NuMega 怎样运用某个特定原则或概念。这些栏目讲述了使软件开发变得愉快的趣味故事、评论和轶闻。

反 馈 意 见

非常高兴能够得到关于本书内容的意见和建议。作者很想知道读者总结的经验教训，并且想知道读者是如何及时交付自己软件的。请通过地址 eds_books@hotmail.com 和作者联系。

目 录

前言	v
第 I 部分 人员、组织与方法	
1. 如何找到优秀的员工	3
2. 简历、面试与雇佣	21
3. 项目组织	41
4. 等级与文化	61
5. 软件工具	75
6. 质量保证基础	99
7. 发布工作基础	123
第 II 部分 项目定义与规划	
8. 需求	139
9. 研究、评估与原型	157
10. 用户界面	171
11. 进度表	185
第 III 部分 项目执行	
12. 监控过程	205
13. 测试版的测试	221
14. 准备发布	237
15. 项目完工	247
索引	257

*PEOPLE,
ORGANIZATION,
AND METHODS*

1

*Great People
and How to Find
Them*

Great people make great software. They define the requirements, create the technology, and stick to the schedule. They test, document, and support the product. The ideas they have, the decisions they make, and the effort, professionalism, and excitement they put into their work make or break the development effort. Because the people who work on the project have the biggest impact on its success or failure, it is essential to hire the most qualified people possible.

Although it's true that everyone wants to hire great people, project teams often accept less qualified candidates than they should. The inability to find candidates or to discern talent can combine with the pressure to ship the product and overwhelm even the best of intentions. If you are unable to solve these problems, you'll have a mediocre team at best and an under-performing one at worst. You can't assume that talented people will just come to you—not regularly, anyway. Instead, you need a firm, organization-wide commitment to getting and keeping the most qualified people possible. This commitment must encompass three key activities: recruiting, interviewing, and retention.

In this chapter and the following chapter, I'll discuss some of the best techniques for finding, selecting, and keeping talented people. I'll also cover how these activities, like software development itself, require planning, discipline, and execution.

DEFINING “GREAT”

Before you set out to find great candidates, you must be sure about what you are looking for. If you can't describe what you want, how are you ever going to know if you've found it? What are the attributes that separate great developers, awesome technical writers, and superstar quality engineers from the not so great? There are obviously many different attributes and many different opinions, but I've boiled it down to six basic attributes.

EXPERTISE

Each potential candidate must have an area of expertise. If you're looking for a developer, he or she needs to have a specific area of technical expertise. If you need a technical writer, the candidate should have expertise in the creation of educational material. If you're hiring a quality engineer, that person's experience should be in automation and testing techniques.

I'm not talking competence here—I'm really talking about expertise. The candidate must have mastery or near mastery of a subject that is pertinent to the project's needs. He or she should be able to give an impromptu talk about his or her subject at any time and show a deep level of understanding about what was done and how it was accomplished. A developer, for example, could talk about any technical knowledge he or she possesses that matches the requirements of the position you have open. Some possible topics include:

- ◆ C++ and object oriented design
- ◆ Creating COM controls
- ◆ MFC and user interface development
- ◆ Assembler and Windows internals
- ◆ Device driver development
- ◆ Networking protocol development
- ◆ Performance tuning

Why is an area of expertise so important? First, if this person has been able to master at least one subject, he or she likely has the skills to master other subjects as they emerge. Technology changes quickly, and the ability to educate oneself and to comprehend significant complexity will be a critical skill.

Second, in high-powered development teams, it's important that there be mutual respect between the team members, and respect is often based on knowledge and ability. As you create a highly talented team, it's very important that each person has one or more areas of expertise in which he or she can contribute as much or more than anyone else.

Does this mean that every developer must have a PhD in computer science and 20 years of experience in software development? No, that's not the idea. But if you've decided to hire a junior developer for support and maintenance, you want to make sure he or she will have the expertise to excel in the position—not just get by.

When you interview the candidate, you should work hard to find and verify that person's area of expertise and make sure it complements your immediate or expected needs. (I'll talk more about interviewing later.)

COMMITMENT

Commitment is another important factor when looking for the most qualified people. In virtually any development project, there will be times when the project isn't going well. You never know what the problems will be ahead of time—your competitor announces their product first, a key developer has health problems, the performance of the product is pitifully slow—but it will be your people's commitment to the project and their belief in what they are doing that will carry the project through. They will show dedication to their tasks and won't stop working and helping until the project is complete. The most committed people see a project through to the end, and they are willing to make the sacrifices necessary to bring about success.

ATTITUDE

Great people often have a great attitude and great energy. They are positive and not cynical; they dwell on the good and not on the bad. They are upbeat

Back at Work

NuMega's release of BoundsChecker 4.0 is an excellent example of what a committed and motivated team can do. It was December 1995, the Internet revolution was just beginning, and Bill Gates had just announced Microsoft's new plans for the Internet. On December 8, the day after the announcement, we got a call from Microsoft asking if we could support their Sweeper SDK and new Internet initiatives. If so, we could do joint press releases and demos at the upcoming Software Development West show in early March. The offer was great for a startup company, but we had less than three months to get this work done, we were in the middle of a release, we had a very small team, and we were just entering the holiday season.

What was the team's response? It was awesome. Everyone recognized the opportunity and knew we needed to take the risk. Starting that afternoon, we put together an updated development plan based on the new facts that we had before us. Of course it wasn't perfect, and there were a lot of unknowns, but we had the germ of a plan, and we worked together. During the next three months, the team was committed to doing whatever was needed to get the job done. And yes, we made it, and it was a great show!