软件项目管理世界经典教材丛书·影印版

Microsoft Press

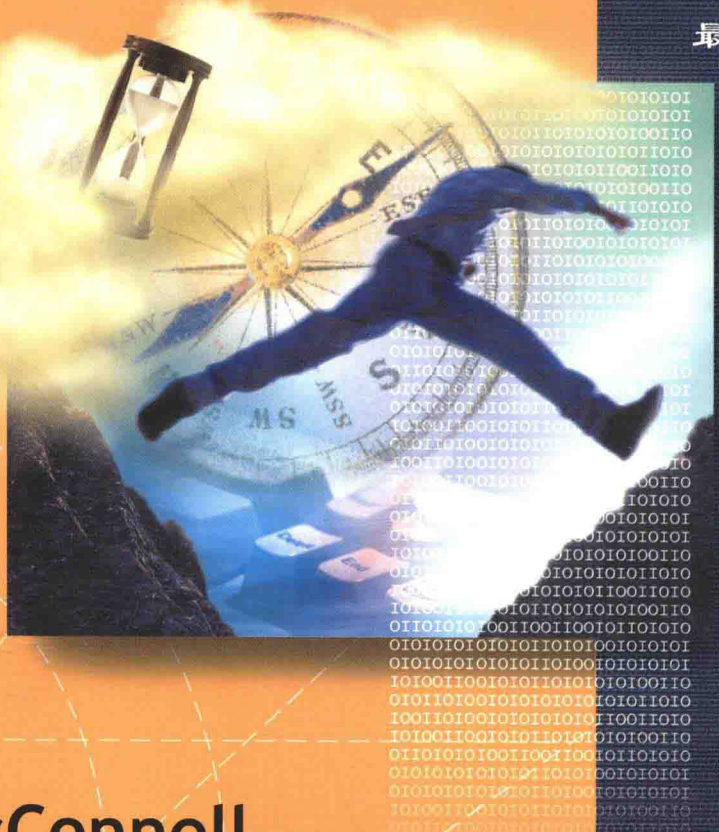
软件项目生存指南

SOFTWARE PROJECT SURVIVAL GUIDE

Amazon读者评价

★★★★½

如何确保
您的第一个
重要项目
不是
最后一个?



Steve McConnell

《代码大全》和《快速开发》的作者

Best Practices

清华大学出版社

"正如Thomas Hobbes在17世纪发现的那样，暴君统治下的生活是孤独、贫穷、野蛮而又短暂的。缺乏管理的软件项目开发工作也是孤独、贫瘠、令人厌恶、野蛮而又极其短暂的。"

——选自《软件项目生存指南》

作者简介

Steve McConnell是业界领先的软件公司（包括微软）的顾问，凭借其著作《代码大全》和《快速开发》而两次获得Jolt奖。他是《IEEE软件》杂志主编、国际著名的软件工程专家、美国Contrux公司总裁，被评为对软件工业最有影响的三位人物之一。

本书是为取得开发项目成果有困难的读者，尤其是那些没有接受过正式软件项目管理培训的软件项目管理人员量身定做的。其中包括高级管理者、总经理、客户、投资者、最终用户代表、项目管理者和技术领导者等。

通过本书读者可以获得广受赞誉的作家Steve McConnell——经典著作《代码大全》和《快速开发》的作者——的指导。他在书中吸收了大量的调查资料和来之不易的职业经验，指明了达到目标的最可靠途径，他称其为"一种特定的对于大多数项目在大多数时间都能很好工作的软件开发方法"。全书共分4个部分19章，涵盖了控制开发过程所需的概念和策略，包括规划、设计、管理、质量保证、测试和存档等。书中吸收了大量的技术积累，创建了一个精简的、可靠的项目管理成功的框架，既适用于初学者，也适用于经验丰富的项目经理。

本书直击帮助项目成功的核心问题，极具参考价值，是每一位业界人士必备的宝典。

读者服务邮箱：service@wenyuan.com.cn

ISBN 7-302-06432-6

9 787302 064329 >

定价：33.00元

责任编辑：冯　涛

封面设计：陈刘源

Best Practices

软件项目生存指南

Steve McConnell

软件项目管理世界经典教材丛书

软件项目生存指南

（影印版）

[美] Steve McConnell 著

清华大学出版社
北京

内 容 简 介

本书是经典著作《代码大全》和《快速开发》的广受赞誉的作者——Steve McConnell 精心编著而成的。4 部分 19 章的内容涵盖了控制项目进程所需的概念和策略，包括规划、设计、管理、质量保证、测试和存档等。对于初学者和有经验的项目管理者来说，本书利用了大量的技术存储创建了一个优美简化的、可靠的项目管理成功的框架。

本书适合任何在软件项目管理上有一定成就的人，尤其是那些没有接受过正式软件项目管理培训的人阅读。包括高级经理、经理、客户、投资者、最终用户代表、项目经理、技术领导以及自学程序员等。也适合任何想学习软件项目管理的人员使用。

北京市版权局著作权合同登记号：图 01-2003-0829 号

图书在版编目(CIP)数据

软件项目生存指南=Software Project Survival Guide/(美)麦克奈尔著.—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 7-302-06432-6

Ⅰ.软… Ⅱ.麦… 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 018509 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：冯 涛
印 刷 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开 本：787×960 1/16 **印张**：18.75
版 次：2003 年 4 月第 1 版 2003 年 5 月第 2 次印刷
书 号：ISBN 7-302-06432-6/TP·4847
印 数：2001～4000
定 价：33.00 元

前　言

目前，在美国大约有 200 万人正在从事约 30 万个软件项目！那些项目中的 1/3 到 2/3 会在发布前超过预定期限和预算。大多数昂贵的软件项目中，大约有一半会因为失控而最终被取消。更多的软件项目以微妙的方式——在萌芽阶段夭折，或发起人简单地宣告成功并在没有新软件显示他们的麻烦的情况下离开了阵地——而被取消。无论您是高级经理、经理、软件客户、用户代表或项目领导，本书都解释了如何使您的项目免遭此厄运。

软件项目失败不外乎两个原因：项目组缺乏成功管理软件项目的知识；或项目组缺乏有效管理项目的决心。本书不能解决缺乏决心的问题，但它一定包含成功管理软件项目所需的大量知识。

导致软件项目成功尤其不是技术因素。软件项目有时看起来很神秘，它根据开发人员是否成功地运用技术而生存或死亡。当被问到为何晚了两周发布一个组件时，开发人员会说如“我们不得不实现一个 32 位形-实转换层以与 OCX 接口对接”之类的话。面对如此的解释，对于不具备深入的专业技术知识的人来说，感觉无力影响软件项目成功并不奇怪。

本书要传达的信息，是软件项目能够生存不只因为考虑了像“形-实转换层”这样的技术细节问题，而是因为更多平凡的理由。软件项目的成功或失败取决于是否对它们进行了认真的规划以及是否慎重地执行了这些规划。大多数软件项目都以一种确定的、可确保成功的方式在动作。如果项目的股东了解决定项目成功的主要问题，他们就能确保项目取得成功。使项目朝着正确的方向前进的人可能成为技术经理或个别软件开发人员——也可为高级经理、客户、投资者、最终用户代表或任何其他有关的团体。

读 者 对 象

本书适用于任何具有一定软件项目成果的人阅读。

高级经理、经理、客户、投资者和最终用户代表

非软件人员通常负责监视软件产品的开发。这些人具有销售、会计、财务、法律、工程或其他领域工作的背景。他们可能没有任何正式的指导项目的授权，但他们仍是监视项目顺利进行的屈指可数的人。至少他们可以在项目开始脱离正轨时发出警告。

如果您处于这样的位置，本书将以简洁、通俗易懂的语言告诉您何谓成功的项目。您可以通过多种途径提前得知项目是朝着成功还是失败的方向发展。本书还将告诉您如何判断：什么情况下没有消

息就是好消息，什么情况下好消息是坏消息，什么情况下好消息确实是好消息。

项目经理

很多软件项目经理是在没有接受过任何管理软件项目的特殊培训的情况下被推到管理位置的。如果您属于这种情况，本书将使您掌握需求管理、软件项目规划、项目跟踪、质量保证和更改控制的关键的技术管理技巧。

技术领导、职业开发者和自学程序员

如果您是技术专家，就可能不必考虑项目领导需要关注的全局问题。在这种情况下，您可能把本书看作带注释的项目规划。通过提供了一个成功的软件项目的概览，本书将帮助您完成从专业技术人员到有效的项目领导的转变。您可以使用在本书中描述的计划作为起点，并根据具体项目的需要制定策略。如果您已经读过了《快速开发》(Rapid Development)，本书的第 I 部分将帮助您回顾其中约一半的内容。您可以跳过第 1 章到第 5 章，仔细阅读第 5 章的结尾，跳过第 6 章，然后再次从第 7 章开始仔细阅读。

本书所涵盖的项目种类

本书中的计划适用于商务系统、简装的大众软件、垂直市场系统、科学系统和类似的程序。本书是为在使用现代开发实践(如面向对象的设计和编程)的桌面客户/服务器项目。本书可轻易地用于使用传统开发实践和大型机计算机的项目。本书的计划是为有 3～25 个成员的项目组、打算用 3～18 个月时间完成的项目而设计的。这些项目被看作是中等规模的项目。如果您的项目规模较小，可根据本书推荐的做法进行调整(书中有必要的说明)。

本书主要面向当前处于规划阶段的项目。如果正处于项目的起始阶段，可使用本书的方法作为项目规划的基础。如果处于项目的中间阶段，在第 2 章中的“生存测试”和在每章结尾的“生存检查”将帮助您决定项目是否成功。

从自身来讲，本书的规划还不够正式或不足以支持生存或安全很关键的系统。本书适用于商业应用程序和业务软件，并显著改进当前已投入数百万美元的大多数项目的规划。

高级技术读者须知

本书描述了一个管理软件项目的有效方法。它并不是惟一有效的运作项目的方法，并且对于特定的项目它可能不是最适宜的。知识最渊博的技术领导通常能够提出比这里描述的一般计划更好、更全

面和更自定义化的开发计划。但是本书所描述的计划会比草率拼凑在一起的计划或根本没有计划要好得多，根本没有计划是屡见不鲜的。

在以下章中描述的计划已详细地阐述了软件项目面临的最常见的缺点。这些计划松散地建立在由软件工程学院(SEI)在 SEI“能力成熟度模型”第 2 级表示的“关键过程领域”。SEI 已经把这些关键过程表示为使公司满足其预定期限、预算、质量和其他目标的关键要素。大约 85%的公司的表现在第 2 级以下，此计划将使这些公司的业绩得到很大的提高。SEI 已经定义了如下所示的第 2 级关键过程领域：

- 项目规划
- 需求管理
- 项目跟踪和监督
- 配置管理
- 质量保证
- 转包合同管理

本书阐述了除转包合同管理之外的所有领域。

参 考 文 献

在编著本书时，除了书中已提及的许多资源以外，我案头必备的是 3 本非常宝贵的参考书。我已尽量浓缩这 3 本参考书的精华，并尽可能以最实用的方式来阐述它们。

第 1 本参考书是 SEI 的《能力成熟度模型关键惯例 1.1 版》(Key Practices of the Capability Maturity Model，Version 1.1)。这本书是在优先实现新开发惯例的过程中来之不易的产业经验的宝藏。大约 500 页的篇幅是有点冗长，而且在这种篇幅下信息仍然很密集。这本书不是一本教程，因此也不是面向初级读者的。但是对于对此书所描述的惯例有基本了解的读者来说，此书提供的总结和结构会是非常有益的。这本书可在 Internet 站点 *http://www.sei.cmu.edu/*免费下载，也可从美国商业部下属的国家技术情报局(NTIS)设在弗吉尼亚州斯柏林菲尔德市的分支机构获得。

第 2 本参考书是 NASA 软件工程实验室(SEL)的《推荐的软件开发方法(修订 3)》(Recommended Approach to Software Development, Revision 3)。SEL 是第一个获得 IEEE 计算机学会的“软件过程成就奖”机构。在此书中可以找到很多通往成功的方法。SEL 的文档描述了一组没有说明如何将这此方法应用于特定项目的惯例，而此书描述了惯例的结构化顺序。这两本书可以互为补充。此书也可在 Internet 站点 *http://fdd.gsfc.nasa.gov/seltext.html* 免费下载。

我准备的最后一本“书”是自己的经验。我不是作为一个要创建完美理论框架的学者来写作的，而是作为要为我的工作和客户创建实用参考的从业者来写的。从这里得到的信息将使我更容易地规划

和管理下一个项目，以及更容易地向客户解释成功的因素。我希望您也能这样做。

目　录

目　录

I

THE SURVIVAL MIND-SET

THE
SURVIVAL
MINDSET

1 Welcome to Software Project Survival Training

Survival prospects for software projects are often poor, but they do not need to be. The first step in surviving a software project is being sure to begin the project in a civilized way. From that starting point, much more than mere survival is possible.

Our standards for software *product* performance are unbelievably exacting compared to our standards for software *project* performance. The software user community expects software products to run for hours without crashing, executing millions of source code instructions with few or no errors. But the software development community expects far less from its projects. Users and clients might complain if a project is delivered one month, three months, or six months late, if it's hard to use, or if it lacks a few critical functions. But if the bulk of the planned software product is delivered at all—at any cost—*ever*—most software consumers will consider that project to be a success. We have suffered so many failed projects that the only outcome we really consider to be a failure is total collapse.

The upper echelons of the software industry have understood for many years what is needed to perform at a much higher level than is currently practiced. A successful project should be one that meets its cost, schedule, and quality goals within engineering tolerances and without padding its schedule or budget. After detailed plans have been made, the current state of the art supports meeting project goals within plus or minus 10 percent or better. This level of performance is currently within the reach of the average software project manager, and in many cases can be substantially influenced by project "outsiders"—upper managers, executives, clients, investors, and end-user representatives.

As chief software engineer at Construx Software Builders, I have been asked to review many failed projects. To the trained eye, the reasons for failure are usually apparent. Failure of medium-size software projects (those with 20,000–250,000 lines of source code) is almost always avoidable. Moreover, I have found that software projects can be optimized to meet any of several goals: shortest schedule, least cost, best quality, or any other goal. Not all of these goals can be achieved simultaneously, and the approach described in this book strikes a careful balance among these objectives so that a high-quality product can be delivered according to an efficient schedule at moderate cost.

Survival Needs

The first step in software project survival is recognizing a software project's essential survival needs. Abraham Maslow observed that human beings respond to a hierarchy of needs that involve a natural progression from

lower motives to higher ones. The lowest level needs are called "survival needs," because they address physical needs that must be satisfied for a human being to exist at all. The lower motives, which are illustrated in Figure 1-1 as the motives below the dashed line, must be satisfied before we can progress to the higher motives. Thus physiological needs for food, air, and water must be satisfied before we can be motivated by the need for "belongingness" and love, self-esteem, or self-actualization.

I have found—along with many software experts—that a similar hierarchy of needs applies to software projects. Software projects have a set of basic survival needs that must be satisfied before it becomes possible for the project team to focus effectively on the project's higher level needs. And the higher levels of the pyramid are where dramatic improvements in quality and productivity take place.

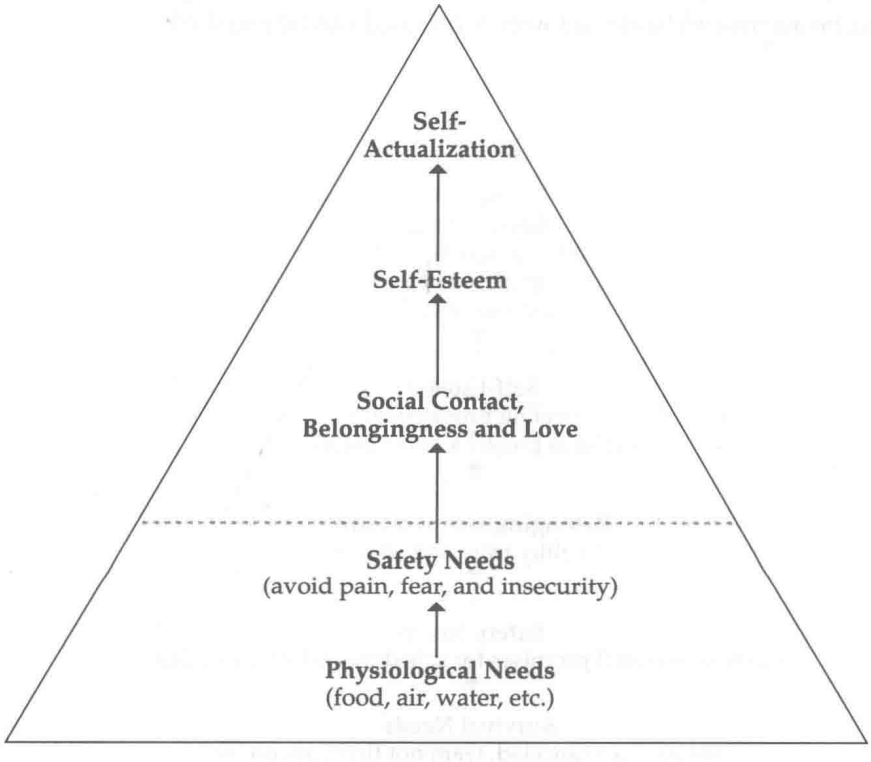


FIGURE 1-1 *Maslow's human need hierarchy. Lower level needs must be satisfied before higher level needs can be addressed.*

A project team must be satisfied that the project can be completed *at all*, for example, before it will begin to worry about whether it will be completed within plus or minus 10 percent of its schedule and budget targets. A project team must be capable of delivering software on time before it will become capable of optimizing a software project to meet an aggressive schedule with a limited budget—and advance the state of the art all at the same time.

As Figure 1-2 illustrates, the needs of the software project hierarchy are not exactly the same as the needs of individual project participants. A developer, for example, will typically prioritize his or her individual self-esteem above the need for healthy team dynamics. But the project typically has a greater need for healthy team dynamics than it has a need for the high self-esteem of individual team members.

This book focuses on the lower levels of the software project need hierarchy, reaching into the higher levels mainly when a higher-level need must be addressed before a lower-level need can be satisfied.

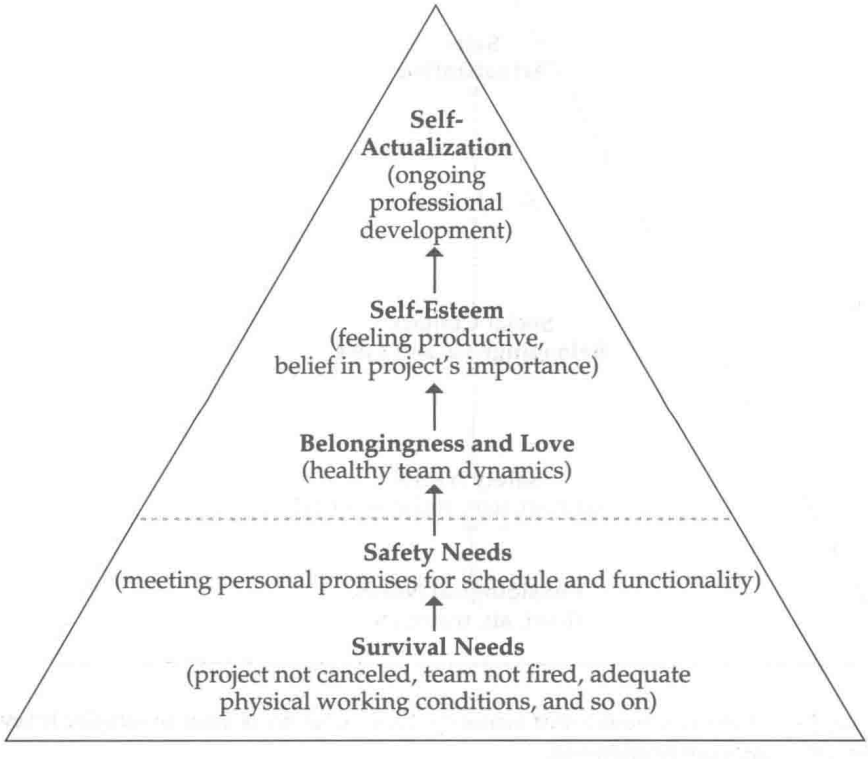


FIGURE 1-2 *Software project need hierarchy. Needs of the project are only approximately the same as the needs of the project's participants.*