



软 件 工 程 技 术 丛 书

软件过程改善系列

软件过程改进 简明实践

Making Process Improvement Work

**A Concise Action Guide for Software
Managers and Practitioners**

Neil S. Potter 著 尤克滨 译
Mary E. Sakry



机械工业出版社
China Machine Press

软件过程改进 简明实践

Making Process Improvement Work
A Concise Action Guide for Software
Managers and Practitioners

Neil S. Potter 著 尤克滨 译
Mary E. Sakry



机械工业出版社
China Machine Press

软件过程改进活动经常面临着理论和实践的严重脱节, 本书提供了一套简明的系统化方法, 用以规划、实施和监测软件过程改进活动, 本书旨在帮助读者跨越这道鸿沟。

本书作者结合自己丰富的实践经验, 以具体明确的步骤和精选的示例, 将读者引入软件过程改进的核心地带。

对于软件管理人员和专业实践者来说, 这本简明实用的小册子一定会给你带来意外的惊喜。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED and China Machine Press.

Original English language title: *Making Process Improvement Work: A Concise Action Guide for Software Managers and Practitioners* by Neil S. Potter, Mary E. Sakry, Copyright © 2002.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书封面贴有Pearson Education培生教育出版集团激光防伪标签, 无标签者不得销售。
版权所有, 侵权必究。

本书版权登记号: 图字: 01-2002-4812

图书在版编目(CIP)数据

软件过程改进简明实践 / (美) 波特 (Potter, N.S.), (美) 萨克瑞 (Sakry, M.E.) 著; 尤克滨译. - 北京: 机械工业出版社, 2003.4

(软件工程技术丛书 软件过程改善系列)

书名原文: *Making Process Improvement Work: A Concise Action Guide for Software Managers and Practitioners*.

ISBN 7-111-11779-4

I. 软… II. ①波… ②萨… ③尤… III. 软件开发 IV. TP311.52

中国版本图书馆CIP数据核字(2003)第016121号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 姚 蕾

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年4月第1版第1次印刷

787mm×1092mm 1/16·10.25印张

印数: 0 001-4000册

定价: 20.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

序

我从来没见过哪一个软件开发人员能够肯定地说：“我今天还在一如既往地按照以前的方式构建软件”。因而，我相信很多人都会有兴趣阅读Neil Potter和Mary Sakry的这本书。在杂志上可以找到很多关于过程改进的文章，在书店里也能找到几本关于过程模型的书，包括论述软件能力成熟度模型（CMM for Software）、SPICE和ISO9001的图书，但如何实施呢？Neil Potter和Mary Sakry帮助我们大大缩短了与这些文献的距离感：即缩短了模型和实践之间的距离，或者说缩短了概念和应用之间的距离。

尽管很多软件开发机构已经从他们的过程改进活动中受益匪浅，但是还是有一些机构在为取得进展而艰苦奋斗。为了进军下一个能力成熟度级别或是通过审计，众多机构制作出一大堆检查列表。这样做不仅不能解决实际问题，而且会使软件开发团队成员反感于这些纯粹为改进而制定的过程。Neil和Mary提醒我们实事求是地将工作重点集中于到达更高一层的业务目标；当然，具体的改进内容可以来自于既成的过程模型以及各种软件工程实践方法的集合。不过，这些大而全的内容并不能给出一张有针对性的解决开发和质量问题的处方；相反，却给出了大量的指导原则和过程改进策略，这些内容普遍适用于任何机构去应付任何难度的挑战。

本书给出值得信赖的建议和明智的方法，用于改进各类开发机构的工作表现（Performance）：定义你的目标，识别那些阻挡你实现目标的障碍，并实施重点突出的变更从而解除这些障碍。Neil和Mary的著作针对的读者是软件开发管理人员和那些有责任引领改进成功的人们。他们鼓励大家将过程改进行动与机构期望的业务目标紧密联系在一起。他们定义了一个系统化的方法，用于规划、实施和评估改进活动的进展。本书所收录的大量经验和教训源于Process Group公司多年来在过程改进方面积累的第一手资料。

如果客户要求你更好、更快地提供更高质量的软件产品，软件开发和项目管理过程就必须进行改进。本书展示了如何定义实际目标，然后设计出能够达到目标的实用策略。当然，如果你很满意当前开发团队的表现，并且客户也总是很满意你们的产品的话，也就没有必要再去阅读此书了。

Karl Wieggers, Process Impact公司首席顾问^Θ

^Θ Karl Wieggers同时还是《Creating a Software Engineering Culture》(Dorset House, 1996)、《Software Requirements》(Microsoft Press, 1999)和《Peer Reviews in Software: A Practical Guide》(Addison-Wesley, 2001)三本书的作者。其中，《Software Requirements》和《Peer Reviews in Software: A Practical Guide》的中文版也由机械工业出版社出版。——编辑注

前言

本书的读者

你也许了解软件开发机构所遇到的问题。问题列表通常开始于过度的承诺和乐观的完工日期。例如，市场部门承诺在年底交付软件。客户被告知他们能够按期得到软件，高层领导设定了奖项，准备激励完成任务的团队。程序员们长时间地劳作，系统测试组的成员们盼望能够及早地进行全面的测试。技术文档编制员却在300多项下拉菜单中迷失了方向，并且得不到程序员的反馈。与此同时，支持团队的工程师还在修复上一个发布版本中的缺陷，他们对在近期内使情况得到好转并不乐观。

在这些任务之上，机构又接受了一项新任务，使用集团工程部所开发的一套新标准和过程。这听起来像是另一项文档工作，收效甚微或者没有任何积极影响。你已经经历了无数次改进活动，每次都消耗大量的时间，但并不能获得预期的收获。结果往往是在后续的项目中很快忘记过程改进的内容。

这场景是不是很熟悉？如果你在类似的机构中工作一两年，就会对这类慢性问题，即新的改进方案和缺乏实质进展的工作感到倦怠。如果你希望找出一种简单的系统性改进方法，请你继续往下阅读本书。

本书的目标读者是软件管理人员和专业实践者。如果你是部门主管，本书将帮助你理解如何进行系统化的改进以及如何将这些改进和你的业务目标直接挂勾。如果你是负责开发某一产品的项目经理或者活动经理，可以使用本书的内容规划、部署和跟踪团队的改进。如果你是过程改进专员、质量管理工程师或软件开发工程师，可以应用每个章节中给出的技巧指导团队的改进活动。

本书的结构

通过本书，我们将指导你的机构获得更好的工作结果。你将了解在实施有价值的持续变更过程中所需要的关键步骤。本书将分析以下问题：

- 如何改进软件开发机构。
- 改进哪些内容。
- 如何部署和跟踪改进。

本书的内容是基于我们大量的实际工作经验，涉及代表全球100多家公司的3000位相关软件开发专业人员。我们在本书中给出了一些实际的案例，很多专业人员在该公司过程改进工作中采用了我们的想法。我们注意到哪些想法切实有效，哪些想法没有效果。

本书短小精干，你可以很快地吸收和使用相关的信息。本书共分3章，参照了休华特PDSA循环（Shewhart cycle）中计划和管理过程改进的相关概念[Deming86]。

制定计划

在第1章中，你将制定一个过程改进行动计划，其根据是业务目标和机构中存在的问题。该方法将针对很多人都经历过的挫折，即过程改进计划通常与在建项目工作没有直接关联。在本章中，你将获知：

- 为改进计划设定引人注目的目标。
- 全部改进活动都针对达到业务要求和解决机构问题。
- 根据既定目标和问题制定一个行动计划。
- 结合目标和问题使用一种改进模型或标准。
- 针对目标导出度量标准。
- 识别行动计划中的潜在问题（风险），缓解最高级别的风险。

实施计划

第2章介绍了在机构中部署新实践准则的技巧。这些技巧将针对那些顽固的问题、拙劣的方案和缓慢的部署。本章的主题包括：

- 应用推销策略部署新的准则。
- 通过解决最需要的和期望的问题加速部署过程。
- 递增地建立的和部署方案从而降低失败的风险。
- 拖延创建制度文档，直至每项方案都经过实践检验。
- 利用现有资源增加部署的速度。

检查进展

第3章介绍了检查改进活动的技巧，以及如何根据改进过程中所学到的东西采取相应的矫正行动。在实现业务目标和解决问题过程中，检查进展状况是向机构提供反馈的基本活动。所获数据允许我们在早期检测出问题并及时矫正，同时让管理人员能够看到改进工作的进展。在第3章中将探讨以下一些方法：

- 根据既定目标使用跟踪方法。
- 确定将改进活动纳入正轨的矫正行动。
- 阐明经验教训以及在后续改进周期中的增效行动。

在阅读过程中，你可以参考本书的附录，它们给出了每章示例中的更多细节。

附录A和附录B是第1章中参考的示例，在此给出了CMM和CMMI框架中的实践与业务目标和项目问题的映射关系。附录C给出了一个完整的改进行动计划示例。附录D扩展了起始于第1章的风险管理计划示例。附录E概述了本书使用的两个参考模型，即CMM(1.1)和CMMI(1.1)。附录F提供了一个完整的微型评审过程定义，在第3章中已经对此有所介绍，这是一种用于跟踪改进活动进展的技巧。

使用改进模型和标准

现存的一些改进模型和标准能够为你节省大量的时间，例如SEI（软件工程研究所）的能力成熟度模型（CMM）和能力成熟度模型集成（CMMI）[⊖]、SPICE、BOOTSTRAP和ISO9001[⊖]。在每一章中，我们称这些模型和标准为框架。框架的内容包含众多来自全球众多研究与实施过程改进的人们的经验教训。如果你能够明智地采用这些框架，将会大大提升成功的几率。如果你教条化地使用这些框架，将会浪费大量的时间。本书中将展示如何将这类资源集成到你的改进活动中。

本书示例所使用的框架包括SEI的CMM和CMMI。如果你正在使用其他种类的框架，例如SPICE、BOOTSTRAP或ISO9001，可在本书涉及到CMM内容时做相应的替换。如果你没有使用任何一种模型或标准，本书介绍的技巧将帮助你针对机构的问题制定自己的过程改进行动。

⊖ 参见附录E有关SEI CMM(v1.1)和CMMI(v1.1)的概述。

⊖ 参见[Zahran98]有关SPICE、BOOTSTRAP和ISO9001的概述。

目 录

序

前言

第1章 制定计划1

1.1 选定改进范围4

1.2 开发行动计划21

1.3 确定风险和计划缓解35

1.4 总结44

第2章 实施计划47

2.1 基于需求推销解决方案48

2.2 优先处理期望的和必需的工作53

2.3 持续强调目标和问题67

2.4 协调管理人员和实践者的行为69

2.5 总结71

第3章 检查进展73

3.1 是否针对目标取得进展74

3.2 是否针对改进计划取得进展83

3.3 是否针对改进框架取得进展85

3.4 迄今得到了哪些经验教训95

3.5 总结105

结论107

附录A 目标和问题与CMM (v1.1)

映射109

附录B 目标和问题与CMM (v1.1) 及

CMMI-SE/SW/PPD (v1.1)115

附录C 行动计划示例123

附录D 风险管理计划示例129

附录E SEI CMM (v1.1) 和CMMI-SE/

SW/PPD (v1.1) 概述133

附录F 微型评审过程137

参考文献147

第1章

制定计划

“没有计划的过程改进不过是一厢情愿的思考。”

——Watts Humphrey, 《*Managing the Software Process*》[Humphrey89]^①

在过去的14年中，我们看到了很多种过程改进方法，其中有两种最常见。在第一种方法中，小团队记录下开发软件过程中所执行的典型任务。概念是这样的，如果记录下当前的实践，就可以在团队中共享，那些最好的实践方式就成为了最佳经验（Best Practices）。理论上讲，如果每个人都采用最佳经验，质量的问题将被解决。不幸的是，实际的结果往往是这些最佳经验成为一堆被忽略的文档。

在第二种方法中，公司遵循成熟的过程改进框架，例如ISO9001或者SEI的CMM^②。这类改进工作通常是为了响应某种目标，如“12月达到CMM3级”。典型的工作重点是根据框架的建议组织团队，创建一些描述公司如何运作的工序（Procedure）。因为框架要求必须有某些工序，所以将这些内容形成文档看起来是合乎逻辑的第一步工作，直接针对目标。采用过程改进框架的决定很好，如果使用得当，将能带来显著的改进。不幸的是，实际的结果往往

① 该书中译名为“软件过程管理”，作者是瓦茨·汉弗莱（Watts Humphrey），其英文影印版已经由清华大学出版社出版。——编辑注

② 能力成熟度模型（Capability Maturity Model, CMM）是由美国卡内基·梅隆大学软件工程研究所（CMU SEI）研究出的一种用于评价软件承包商能力并帮助改善软件质量的方法。——编辑注

是只有部分的益处、普遍的失望和一大堆文档的混合体。

从这两个示例中可以看出，使用过程改进框架和定义最佳经验本身并没有问题，关键是应用于过程改进的方法。这些过程主导（Process-Centric）的方法能够带来实际的收获。但是，这种做法蕴涵很高的失败风险，因为这种方法鼓励团队将注意力集中在与机构实际工作并不直接相关的内容（例如过程文档）之上。这种方法的成功取决于每个文档作者能清楚地解释所建议过程背后的一些机理以及如何将其应用于当前和后续的项目工作。当这种类型的沟通停止之后，机构中的其他人员会将这次（过程主导方法）的改进结果看成是又一个不实用的质量改进活动。

图1-1形象说明应用过程主导方法的通常结果。工作开始之后，很快就迷失了方向，导致很多人对结果很失望。此时开发人员在仰望着山顶（即他们要交付的软件产品），停止过程改进活动，照旧编写他们的程序。

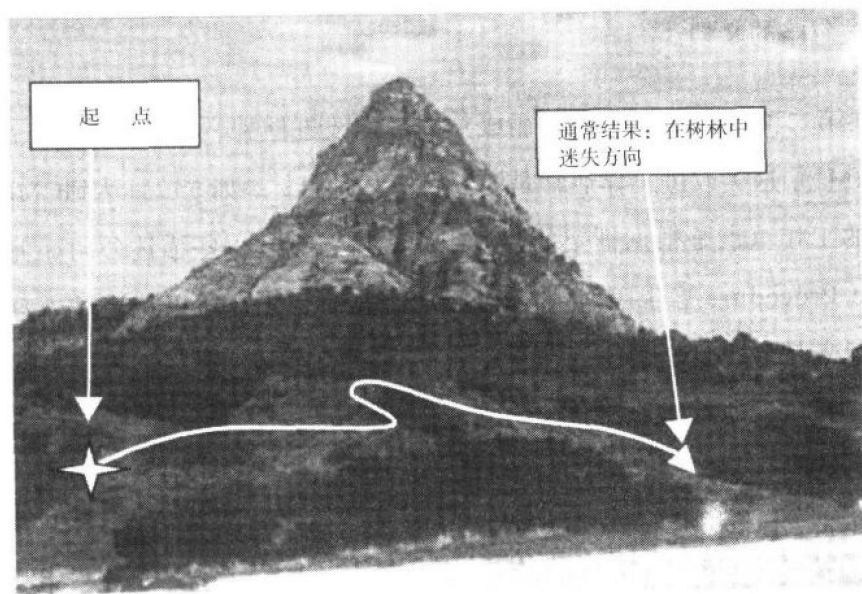


图1-1 过程主导方法用于过程改进

如果过程改进活动的重点是过程框架，那通常不是必须要做的工作，这是一个比较奢侈的要求，只有在业务形势很好的情况下，才可能开展这类活动。即便在业务形势不错的时候，我们也会奔忙于另外的活动和收益，过程改进往往被看成是非必要的任务。

另外一种方法则有所不同，这种方法以机构的目标和问题为工作重点，将过程改进活动和当前项目的工作直接挂勾。使用这种方法，过程改进的重点是机构中的实际问题，每一个变化都被一个明确的需求驱动。过程改进的范围并不是根据过程改进框架而定义，而是根据机构的目标与问题而定义。

应用本书中介绍的“目标－问题”方法，机构将注意力持续集中于引人注目的问题，即那些大家现在就想解决的问题。改进计划围绕机构所面临的挑战，通过一系列持续的小规模行动推动机构不断接近其目标。过程改进活动的进展通过机构被改进的工作结果加以衡量。图1-2形象说明了“目标－问题”方法用于过程改进活动的计划。

选定改进的范围

- 确定计划的责任权
- 陈述主要的目标和问题
- 将问题分组关联到相应的目标
- 确认目标和问题是够明确和引人注目
- 设定目标的优先级
- 导出针对目标的度量标准



开发行动计划

- 使用头脑风暴和过程框架枚举一系列行动

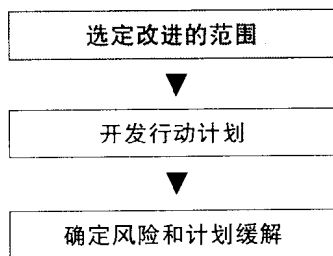
(续)

- 基于目标和问题组织行动计划
- 添加用于检查进展和采取矫正行动的空位

**确定风险和计划缓解**

- 确定风险会议的范围
- 选择团队和仲裁者
- 识别风险
- 分析风险
- 计划缓解
- 计划定期的风险复审

图1-2 “目标-问题”方法用于过程改进活动的计划



1.1 选定改进范围

图1-3突出显示了机构的一个目标。该目标可能是“交付软件产品”、“完成软件安装”或者“升级一个数据库”。目标也可能是“百分之百地符合工期要求”，或者“降低25%的开发工作量”。这些都是业务目标 [Hill79]。“目标-问题”方法起始于业务目标，揭示那些有可能阻碍实现这些目标的问题 [Sakry01]。在此基础上建立改进计划，用于解决实际问题，从而推动机构接近其目标 [Robbins98]。

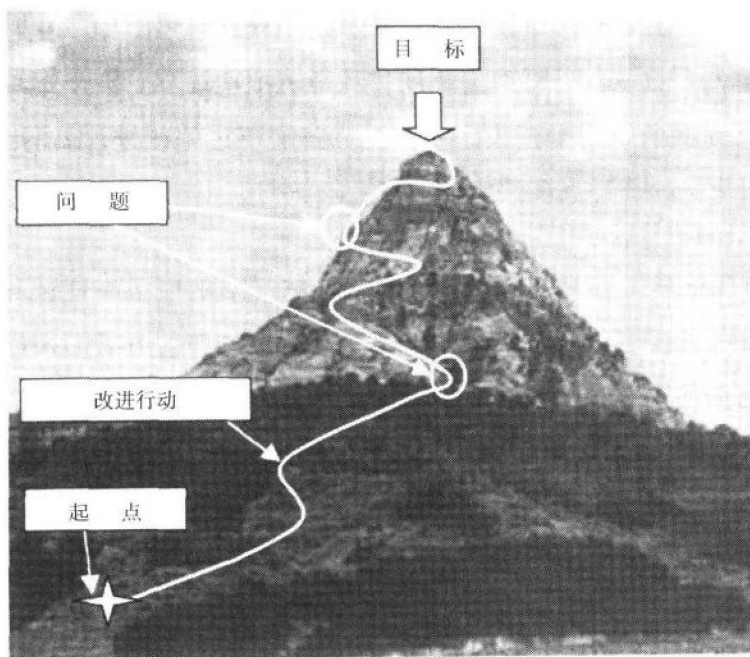


图1-3 “目标-问题”方法用于过程改进

选定改进工作的范围，遵循以下几个步骤：

- 1) 确定计划的责任权。
- 2) 陈述主要的目标和问题。
- 3) 将问题分组关联到相应的目标。
- 4) 确认目标和问题足够明确和引人注目。
- 5) 设定目标的优先级。
- 6) 导出针对目标的度量标准。

第1步：确定计划的责任权

建立的过程改进计划需要一个责任人。找出这个责任人，并确认该计划能满足此人的要求。如果等到计划完成时（再找责任人），面临的危险将是没

人想要这个计划。计划的责任人与计划所针对业务目标的拥有者应该是同一个人。对于大多数业务目标而言，项目经理、活动经理、资深管理人员或部门主管都是候选的计划责任人。在找到真正需要过程改进计划的人选之前，不要投入过多时间去规划你的改进工作。

当遵循本章给出的步骤制定计划时，在确定整个计划的首要责任人基础之上，最好委任不同人负责计划中的各个部分。例如，市场部人员所负责的内容将针对如何向客户做出合理的承诺。类似地，测试经理所负责的内容将针对如何改进产品的测试，开发经理所负责的内容将针对项目规划和产品设计。每个人希望在计划中看到针对其面临问题的内容，这也是他们负责计划中相应部分的动力来源。

改进计划的首要责任人不应该是软件工程过程组（Software Engineering Process Group, SEPG）的人员，不应该是软件质量保证（Software Quality Assurance, SQA）的人员，也不应该是类似支持部门的成员。支持部门可以与拥有业务目标的责任人分担制定计划的责任，但是支持部门不能独立地承担实现这些目标的任务。通常在计划执行过程中，支持部门向软件开发团队提供切实有效的支持并收取相关的费用。

第2步：陈述主要的目标和问题

在一次拜访客户并帮助他们计划过程改进活动的实践中，我们获知过程改进组一开始将改进工作范围选定于CMM第2级的6个关键过程领域（Key Process Area, KPA）^①。建议开发人员和管理人员暂时忘掉第2级，陈述在未

^① CMM第2级的6个关键过程领域分别是：需求管理、软件项目计划、软件项目跟踪和监督、软件配置管理、软件质量保证和软件子合同管理。参见附录E，获知更多关于CMM的参考信息和定义。

来6到18个月内他们希望达到的主要业务目标。接着，要求他们陈述与软件开发相关的一系列问题。经过一小时的讨论，他们创建了图1-4中展示的两个列表。

目 标
1) 创建可预见的日程安排。
2) 成功交付产品X。
3) 减少重复工作。
4) 改进核心软件产品的性能。
5) 保持客户愉快。
6) 保持盈利。
问 题
1) 需要更好的需求。需求的跟踪不到位。需求的变更没有被跟踪；在测试的时候，代码不符合规约的描述。
2) 2.3版产品的管理方针不明确。目标经常变化。
3) 质量部没有接受产品和测试技能方面的培训。
4) 软件变更的状态不明确。
5) 分配用于软件设计的资源和技能匮乏。
6) 针对文档和规约的变更，没有及时地与文档和测试小组沟通。
7) 很难找到从事关键活动（产品开发）的时间。时间花费于应付紧急活动。
8) 测试计划中没有包括与客户相关的内容。
9) 修复缺陷之后破坏产品的基本特性。
10) 一些错误的文件（例如动态链接库）被刻入光盘。不能确定哪些是正确的。
11) 很难修改项目计划。去掉一些旧条目，加入一些新条目，计划就过时了。
12) 我们不能判断自己的承受能力，没有一个需要完成的全部工作列表。
13) 针对那些受变更影响团队，变更跟踪与沟通的规划工作做得很糟糕。
14) 测试人员没有工具管理他们的测试活动。
15) 客户不愉快。存在300多个显著的缺陷还没有被处理。

图1-4 某一机构的目标和问题

对于某些机构而言，他们很难接受暂时忘掉与过程相关目标（例如