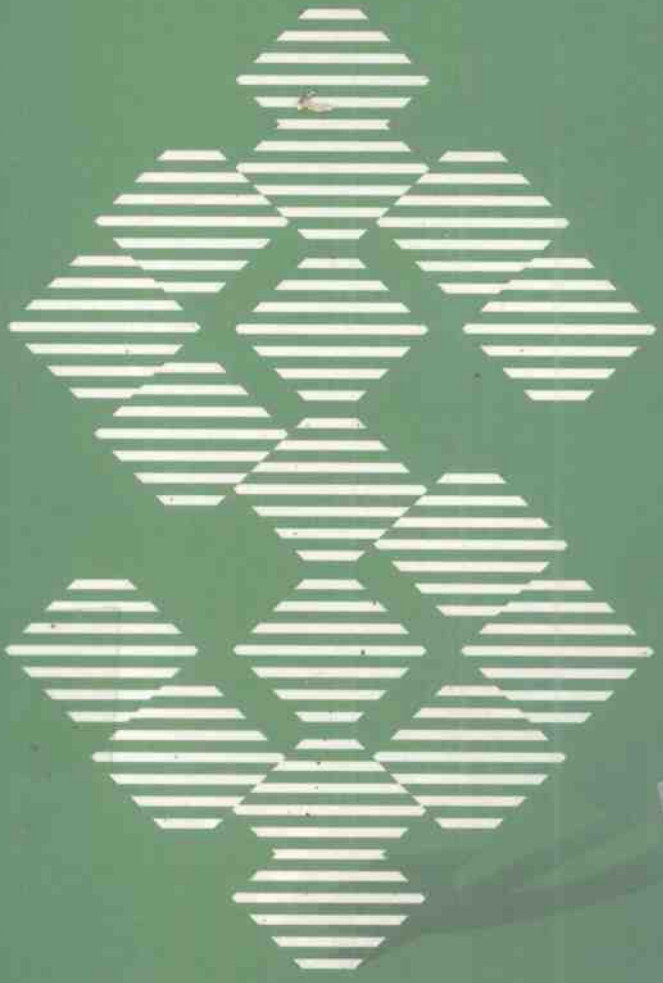
SEI软件工程译丛

用商业组件构建系统

Building Systems from Commercial Components

[美] 库尔特·C·瓦尔诺 [Kurt C. Wallnau]
斯哥特·A·希萨姆 [Scott A. Hissam]
罗伯特·C·塞克德 [Robert C. Seacord] 著

王瑜 译



清华大学出版社

SEI软件工程译丛

用商业组件构建系统

Building Systems from Commercial Components

库尔特·C·瓦尔诺 [Kurt C. Wallnau]

[美] 斯哥特·A·希萨姆 [Scott A. Hissam] 著

罗伯特·C·塞克德 [Robert C. Seacord]

王 瑜 译

清华大学出版社

（京）新登字 158 号

内 容 简 介

商业组件的广泛使用给软件工程学带来了全新的挑战。商业组件的复杂性和商业市场中的不确定因素使得软件人员必须适应从制定组件规范到集成现有商业组件的转变。本书全面、深入地介绍了各种扩展的软件工程思想和方法，并结合实例，详细阐述了各种方法和技术在实际开发中的运用过程。

本书适合所有从事或希望从事软件开发工作的人士阅读。

图书在版编目（CIP）数据

用商业组件构建系统/[美]瓦尔诺，[美]希萨姆，[美]塞克德著；王瑜译.
—北京：清华大学出版社，2002
（SEI 软件工程译丛）
书名原文：Building Systems from Commercial Components
ISBN 7-302-05896-2
Ⅰ.用… Ⅱ.①瓦…②希…③塞…④王… Ⅲ.软件开发 Ⅳ.TP312.52
中国版本图书馆 CIP 数据核字（2002）第 072299 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.tsinghua.edu.cn
责任编辑：尤晓东
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：24.25 插页：2 字数：529 千字
版 次：2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷
书 号：ISBN 7-302-05896-2/TP·3501
印 数：0001～3000
定 价：59.00 元

出版说明

1984年，美国国防部出资在卡内基·梅隆大学设立软件工程研究所(Software Engineering Institute，简称SEI)。SEI于1986年开始研究软件过程能力成熟度模型(Capability Maturity Model，简称CMM)，1991年正式推出了CMM 1.0版，1993年推出CMM 1.1版。此后，SEI还完成了能力成熟度模型集成(Capability Maturity Model Integration，简称CMMI)。目前，CMM 2.0版已经推出。

CMM自问世以来备受关注，在一些发达国家和地区得到了广泛应用，成为衡量软件公司软件开发管理水平的重要参考因素，并成为软件过程改进的事实标准。CMM目前代表着软件发展的一种思路，一种提高软件过程能力的途径。它为软件行业的发展提供了一个良好的框架，是软件过程能力提高的有用工具。

SEI十几年的研究过程和成果，都浓缩在由SEI资深专家亲自撰写的SEI软件工程丛书(SEI Series In Software Engineering)中。为增强我国软件企业的竞争力，提高国产软件的水平，经清华大学出版社和三联四方工作室共同策划，全面引进了这套丛书，分批影印和翻译出版。这套丛书采取开放式出版，不断改进，不断出版，旨在满足国内软件界人士学习原版软件工程高级教程的愿望。

“SEI 软件工程译丛”
编 委 会

SEI
软件工程译丛

总　序

——为清华大学出版社出版“SEI 软件工程译丛”而作

美国卡内基·梅隆大学软件工程研究所(CMU/SEI)是美国联邦政府资助构建的研究单位,由美国国防部主管。他们确认,为了保证软件开发工作的成功,由软件开发人员、软件采办人员和软件用户组成的集成化团队必须具有必要的软件工程知识和技能,以保证能按时向用户交付正确的软件。所谓“正确的”就是指在功能、性能和成本几个方面都能满足用户要求且无缺陷;所谓“无缺陷”就是指在编码后对软件系统进行了彻底的穷举测试修复了所有的缺陷,或保证所编写的代码本身不存在缺陷。

CMU/SEI 为了达到这个目的,提出了创造、应用和推广的战略。这里的“创造”是指与软件工程研究社团一起,共同创造新的实践或改进原有的实践,而不墨守成规。这里的“应用”是指与一线开发人员共同工作,以应用、改进和确认这些新的或改进的实践,强调理论联系实际。这里的“推广”是指与整个社团一起,共同鼓励和支持这些经过验证和确认的、新的或改进的实践在世界范围内的应用,通过实践进行进一步的检验和提高。如此循环,往复无穷。

他们把所获得的成就归纳为两个主要领域。一个是倡导软件工程管理的实践,使软件组织在采办、构建和改进软件系统时,具有预测的能力与控制质量、进度、成本、开发周期和生产效率的能力。另一个是改进软件工程技术的实践,使软件工程师具有分析、预测和控制软件系统属性的能力,其中包括在采办、构建和改进软件系统时,能进行恰当的权衡,作出正确的判断和决策。CMU/SEI 通过出版软件工程丛书,总结他们的研究成果和实践经验,是推广这两个领域经验的重大举措。

SEI 软件工程丛书由 CMU/SEI 和 Addison-Wesley 公司共同组织出版,共分 4 个部分:计算机和网络安全(已出版了 2 本著作),工程实践(已出版了 8 本著作),过程改进和过程管理(已出版了 11 本著作),团队软件过程和个体软件

过程(已出版了 3 本著作)。前两者属于软件工程技术实践,后两者属于软件工程管理实践。目前这 4 个部分共出版了 24 本著作,以向软件工程实践人员和学生方便地提供最新的软件工程信息。这些著作凝聚了全世界软件工程界上百位开拓者和成千上万实践者的创造性劳动,蕴含了大量的宝贵经验和沉痛教训,很值得我们学习。

清华大学出版社邀请我和郑人杰教授共同组织 SEI 软件工程译丛编委会。清华社计划首先影印 6 本著作,翻译出版 15 本著作。据我所知,在 Addison-Wesley 公司出版的 SEI 软件工程丛书中,人民邮电出版社已经翻译出版了《个体软件过程》和《团队软件过程》,还拟影印出版《个体软件过程》和《软件工程规范》;电子工业出版社已经翻译出版了《净室软件工程的技术与过程》、《能力成熟度模型 CMM 1.1 指南》、《能力成熟度模型集成 CMMI》和《软件项目管理》;北京航空航天大学出版社已经翻译出版了《统计过程控制》。这些出版社共计影印 2 本著作,翻译出版 7 本著作。这样,可以预期我国在今年年底共可影印 8 本著作,翻译出版 22 本著作。各个出版社的有远见的辛勤劳动,为我们创造了"引进、消化、吸收、创新"的机遇。我们应该结合各自的实践,认真学习国外的先进经验,以大大提高我国软件工程的理论和实践水平。

在这套丛书中,特别值得一提的是,在过程工程领域被誉为软件过程之父的 Humphrey 先生所撰写的《软件过程管理》、《技术人员管理》、《软件工程规范》、《个体软件过程》、《团队软件过程》和《软件制胜之道》等 6 本著作,将于今年年内全部翻译出版,其中《软件过程管理》、《技术人员管理》、《软件工程规范》、《个体软件过程》和《软件制胜之道》这 5 本著作亦已经或将于今年年内影印出版。

《软件过程管理》是软件过程领域的开创性著作,是为软件公司经理和软件项目经理撰写的。用这本书提出的原理来指导软件开发,可以有效地按照预定进度得到高质量的软件,同时还可了解如何持续进行过程改进。美国 CMU/SEI 按照这本书提出的原理开发了能力成熟度模型,在国际上得到绝大多数国家的认可和广泛采用,是改进软件过程能力的有力武器。在信息技术迅速发展和企业激烈竞争的今天,能否持续改进过程往往决定企业的命运。

作为一个软件经理,在改进组织的能力之前,首先必须明确绝大多数软件问题是由管理不善所引起的。因此,要改进组织的性能,首先需要改进自己的管理模式。同时还要认识到软件开发是一项智力劳动,需要拥有掌握高技能和忘我工作的技术人员。因此,有效的软件管理需要充分注意技术人员的管理。

《技术人员管理》这本著作就是为达到这个目的而撰写的。高质量的技术

工作要求没有差错，这就要求人们高度专心和高度献身。因此要求人们对他所从事的工作不仅具有高度的责任感，而且具有浓厚的兴趣和高度的热忱。在当前知识经济群龙相争的今天，一个能激励人们进行创造性工作的领导群体，是众多竞争因素中最重要的因素。本书提供了大量的实用指南，可用来有效地改进工程人员、经理和组织的性能。

Humphrey 先生还认为这本书特别适合于在我国工作的软件经理。我国是一个人口大国，拥有大量能干的知识分子，而且信息领域的劳动力价格比国际市场的价格要低，因此吸引了许多国家到我国来投资。但若不提高人员的素质，不在产品质量和进度方面也狠下功夫，就不能在这方面持续保持优势。

《软件工程规范》是为编程人员撰写的。它精辟地阐述了个体软件过程(PSP)的基本原理，详尽地描述了人们如何来控制自己的工作，如何与管理方协商各项安排。在软件工程界，这本著作被誉为是软件工程由定性进入定量的标志。目前在世界范围内，有成千上万的软件工程技术人员正在接受有关 PSP 的培训，以便正确地遵循 PSP 的实践、开发和管理工作计划，在他们承诺的进度范围内，交付高质量的产品。

《软件制胜之道》这本著作描述了团队软件过程的基本原理，详尽地阐述了在软件组织中如何应用 PSP 和 TSP 的原理以及它所能带来的效益。此外，虽然 CMM 同样适用于小型组织，但在其他著作中都没有描述如何应用 CMM 于个体或小型团队，这本书填补了这个空白。应该指出，如果一个组织正在按照 CMM 改进过程，则 PSP 和 TSP 是和 CMM 完全相容的。如果一个组织还没有按照 CMM 改进过程，则有关 PSP 和 TSP 的训练，可以为未来的 CMM 实践奠定坚实的基础。

在软件工程技术实践方面目前共出版了 10 本著作，其中《用商业组件构建系统》、《软件构架实践》和《软件构架评估——方法和案例研究》等 3 本著作详尽地阐述了软件构架的构建、实践和评估。鉴于是否有一个稳定的软件构架，对软件的质量和成本影响很大，因此如何获得一个良好的构架就成为当今软件界研究的重点。我相信这几本著作的出版，将对我国软件构架领域的研究与实践有重要的参考价值。此外，众所周知，计算机与网络的安全问题对信息系统的可靠使用关系极大，《CERT 安全指南——系统与网络安全实践》的出版将会对我国在这一领域的研究和实践起积极的促进作用。《风险管理——软件系统开发方法》、《软件采办管理——开放系统和 COTS 产品》、《项目管理原理》、《软件产品线——实践和模式》和《系统工程：基于信息的设计方法》等 5 本著作，分别从风险管理、软件采办、项目管理、软件产品线以及信息系统设计方法等几个

方面阐述了大型、复杂软件系统的开发问题，是有关发展软件产业的重要领域，很值得我国软件产业界借鉴。目前我们所处的时代是信息化时代，是人类进入能够综合利用物质、能量和信息三种资源的时代。千百年来以传统的物质产品的生产、流通、消费为基本特征的物质型经济，将逐步进入以信息产品的生产、流通、利用和消费为基本特征的知识型经济。在这个历史任务中，建造和广泛应用各类计算机应用系统是其公共特征。计算机软件是计算机应用系统的灵魂，没有先进的软件产业，不可能有先进的信息产业，从而也不可能建成现代化的知识型经济。

我们应该看到，在软件领域中我国在总体上离世界先进水平还有相当大的差距。但是，我们不能跟随他国的脚印，走他人的老路。我们应该抓住机遇，直接针对未来的目标，在软件工程技术和软件工程管理两个方面，注意研究 SEI 软件工程丛书中倡导的原理和方法，联系实际，认真实践，并充分利用我国丰富优秀的人力资源和尊重教育的优良传统，大力培养各个层次的高质量的软件工程人员，使其具有开发各类大型、复杂软件系统的能力。我衷心地预祝清华大学出版社影印和翻译出版这套丛书，在把我国建设成为一个真正现代化的软件产业大国的历史任务中起到推波助澜的作用，并请读者在阅读这些译著时，对这套丛书的选题、译文和编排等方面都提出批评和建议。

周伯生

于北京

2002 年 8 月 18 日

前　言

在基于组件的软件设计理论与实践之间确实存在着差异，而且这种差异正在日益扩大。

市面上也有一些关于基于组件设计方面的书籍，但这些书都假设设计任务就是为软件组件开发设计规范。实际上，绝大多数基于组件的设计工作都依赖于现有的组件。这两种观点都有自己的市场。但现在组件带来了新的挑战，同时，使用这些组件也越来越广泛。使用现有组件意味着要遵守现有的组件规范，对于设计者来说这是一种自然的约束。

当前基于组件的设计方法集中在某些特定的设计问题上，这些问题让我们提不起兴趣，也很少出现这些问题。设计方法中更通用、更令人感兴趣的东西是那些设计人员不能够把握的因素。

- 使用现有组件会带来一个完全不同的、由组件规范引起的设计问题。现有的组件使得设计者面临问题选择，而自由定义组件接口的方式又使得设计者面临问题优化。对于软件工程师来说，这两种设计问题的差异是逐渐显现出来的，但是设计方法并没有得到足够的重视。
- 使用现有组件大大削弱了我们对底层设计决策的控制。这些决策包括：系统是怎样分布到组件中的，组件都提供了什么样的功能，组件之间如何协作。在软件工程理论中，这些都属于构架决策方面的内容。这将导致错误的结论，即广泛使用现有组件是违反或至少不适应软件设计方法的。

这里我们只是简要地讨论了基于组件设计方法的现状，还没有得出在这种设计模式的理论和实践中存在着日益扩大的差异的结论。事实上，这种差异的确存在，而且不言自明，只要你知道如何找到这种差异。

基于组件的开发模式已经流行了 15 年，并且已经在商业软件中扎下了根基。许多软件产品，例如关系数据库管理系统、事务监测器、消息代理、事件管理器、加密服务、Web 浏览器和服务器、图形信息系统以及产品数据管理系统等，它们都符合软件组件的标准，或至少为业界所了解。这就是说，这些软件都

实现了某种功能，表现为二进制形式，能独立部署，由一个程序接口所描述，并且支持第三方的集成。

商业市场是软件组件的主要来源。现阶段是这样，在将来无法预期的一段时间内也将保持这种现状。我们认为组件以及软件组件市场之间存在着明显的联系。Szyperski 在他的一本很有影响的著作中也认同这种观点，他注意到组件必须切合市场的环境[Szyperski 98]。Szyperski 关于市场的说法在很大程度上（但并不完全是）有些晦涩。与此相反，我们这里所提及的组件市场是指现在已经存在的状况，其中包括组件供应商、组件基础构架供应商、第三方组件集成商以及最终用户。

忽视市场在软件工程中的作用就如同在机械工程中忽略了摩擦一样。需特别指出的是，商业软件组件具有 3 个特点，这 3 个特点共同作用，导致了由软件组件带来的挑战。

（1）商业软件组件是复杂的。对于组件市场来说，这种复杂性是必然的。很多组件非常复杂，以至于使用它们的专家们也不了解它们的全部特性。对于这些组件的特性和行为总是有不了解的地方。

（2）商业软件组件是与众不同的。商业软件组件的标准虽然有用，但是最吸引顾客的往往是组件的一些新特性。这意味着关于组件的知识是供应商特有的，并且这些新特性（即非标准特性）之间的不匹配增加了集成的难度。

（3）商业软件组件是不稳定的。必须不断引入新的特性不断升级，当这些成功的新特性被竞争对手模仿后，组件必须引入新的特性。组件知识的半衰期很短，并且设计上基于组件特性的假设也是靠不住的。

在部署现实系统时发现了软件组件的这些特点。这些特点使得关于软件设计是一个有序过程的假设也变得让人迷惑起来，而且这个假设是传统软件设计方法的基础。既然所有基于组件的开发方法都会涉及到商业组件市场的问题，所以我们需要从方法学上解释这些新的复杂性。

方法学上的回应

有关方法的一个中心议题是，在基于组件的软件设计中，主要的风险是我们不了解这些组件应如何集成，以及集成后是如何运作的。要降低这种风险，基于组件的设计模式必然涉及探索和发现组件的问题。这种探索活动的主要动机是获取并维持技术（组件）能力。

这种观点看起来像是对软件过程进步所形成的原则的背叛，这种原则强调管理技能而不是技术技能，强调集体行为而不是个体的贡献。其实，一方面，在组件集成细节方面，我们提倡仔细推敲；另一方面，在软件工程方面，关于额外的高深技术能力，我们提倡的却是超越个人英雄主义。这些表明我们对软件过程的理解与基于组件开发的现实是不相符的。事实上一种设计是否可行常常依赖于是否经过仔细推敲。进而，整个设计概念常常依赖这些底层的细节。一个无法回避的事实是，如果需要把握细节的话，精通技术是十分重要的。

下面是我们在方法学上改进的一些核心要素。

(1)作为一种基础的设计抽象，我们引入了组件集成块的概念。集成块提出了组件独立性的要求，并且将我们工作的重点由选择单个的组件转移到选择能协调工作的组件集合上来(这就是集成块)。

(2)我们将黑板理解为一个基础的设计符号。黑板描述了有关集成块已知的知识，并且，更重要的是，描述了将被发现的事物。黑板用于将一个设计项目及已知的设计风险文档化。

(3)为了暴露设计风险及定义集成块的可行性标准，我们引入了一个风险驱动的发现过程，名为 R^3。为产生合适的组件技术以及确立集成块的可行性，我们还引入了一个原型化方法，名为模型问题。

(4)根据集成块的关系及断言，我们引入了设计空间的概念。设计空间的概念用于提供集成之间的独立性，它被用来作为对预期市场行为的响应，例如新组件的发布。同时当集成块的可行性遇到问题时也能处理设计上的一些障碍。

这种方法学上的改进将用以面对由商业组件市场带来的挑战。它一方面防止将设计过程变为一项随意性的工作，另一方面防止某些具有创新意义但不稳定的技术主宰设计过程，导致额外但可以避免的设计风险。我们相信，这里所讨论的方法可以面对这种挑战。

关于本书

本书的目标

我们的目标非常明确。第一个目标就是展现这样一个事实，即软件组件对软件工程提出了新的方法学上的挑战。围绕这个问题，我们希望阐明这些挑战

的本质，并特别地将重点放在由组件市场变化所带来的挑战上。我们的第二个目标是详细阐述用以应付这些挑战的方法和技术。我们认为对于改进任何软件组件的方法学来说，这些方法和技术都是必要的基础。我们的最终目标是：对一个具有现实意义的案例进行研究，这个案例来自于我们在开发一个大型企业系统时所积累的经验。通过这些研究，解释组件设计的复杂性，以及推荐我们的方法和技术所带来的效应。

读者对象

本书既适合那些从事基于组件进行开发的开发人员阅读，又适合软件工程专业的学生。尽管对读者而言本书所有的章节都是有用的，但还是应该根据需要确定重点。

系统构架师 将学习到有关集成块、集成修正和集成块可行性方面的技术，以及和专业技能相关的附加知识。通过对设计空间的学习，系统构架者将掌握概念性的语言，并通过该语言，他们可从多个层面处理紧急情况，并且修正那些具有不同复杂特点的基于组件的系统。

总工程师 系统构架师主要把握设计概念上的完整性，而总工程师则要在实践中把握系统的可行性。如果不加以重视，复杂的组件以及它们之间的相互作用将掩盖系统中潜在的风险。总工程师通过使用 R^3 及模型问题方法能够将这些风险暴露出来。

项目管理者 项目管理者最先考虑到项目的风险，以及如何降低这种风险。来自 R^3 过程（R^3 之一即风险识别）的主动搜索技术将满足这种需要。设计空间为项目管理者精确地提供了对有关技术状态的即时描述，并且提供了一种构架，通过这种构架可以为设计工作提供对于项目目标的定位以及跟踪。

首席技术官（CTO） 现代企业级系统广泛地由商业组件构成，这种规模大、时间长的系统，其成败与设计密不可分，事实上，在任何时候，这种系统都蕴涵着开发周期中的各个阶段。通过阅读本书，首席技术官将发现书中所讲的所有概念和技术都会在更新管理技术时使用到。

本书也适合软件工程师和程序员阅读，在基于组件的系统开发中，一线的开发人员是真正的无名英雄，项目的成功与否与开发者能否保持技术趋势的正确方向密不可分。本书使得开发人员有充足的理由说服管理者不仅要重视开发方法培训，还要重视技术培训。

阅读方法

本书包括 3 部分：

- 第Ⅰ部分讨论了由商业组件带来的挑战。我们解释了用以应对这些挑战的软件工程技术。就如何将这些技术整合到现有的开发过程中去，讨论了工作流技术。
- 第Ⅱ部分以一个项目为案例进行了研究，我们在 1998 年开发了这个项目。这部分的每一章都阐述了由商业组件带来的挑战以及应对这些挑战所要用到的技术。
- 第Ⅲ部分就如何运用本书提出的技术给出了一些建议。就将来基于组件的开发模式，我们也有一些预见。

第 1 章介绍了由基于组件开发模式所带来的问题。第 2 章到第 4 章解释了为什么有必要放弃软件方法中某些呆板的规则。第 5 章讲述了组件集成块和黑板。这些概念都非常重要，并且将贯穿全书。为了进行探索式设计和减少风险，第 6 章定义了过程模型。第 7 章和第 8 章讨论了如何分别管理和使用由这些方法开发而成的设计文档。第Ⅰ部分的其他章节讨论了一些具体的技术（确切地说应该是一些技术族）用以开发基于组件的系统。这些章节可以以任意顺序阅读。您也可以先跳过这些章节，等到学习案例研究时再根据需要回顾头来学习这些章节。

在案例研究中描述了一系列的事件，相关章节都是按照事件的顺序而编排的。假如您没有按照顺序阅读这些章节，可能不会很了解这些章节的写作动机，同时这些章节又是相对独立的。但第 14 章是个例外，它提供了有关公共密钥基础设施（PKI）和安全问题的简要介绍。如果您已经了解有关 PKI 的知识，可以略过该章，否则，您还是需要阅读该章以便了解案例研究中的具体细节。

目录