

个体软件过程

PSP引论

[美] Watts S. Humphrey 著

吴超英 车向东 译

周伯生 校

Introduction to the **Personal Software Process**SM



软件工程和过程工程丛

个体软件过程 PSP 引论

[美] Watts S. Humphrey 著

吴超英 车向东 译

周伯生 校

(样本书 非卖品 待出版)

北京航空航天大学出版社

内 容 简 介

《个体软件过程 PSP 引论》是 Watts S.Humphrey 于 1997 年特为美国大学一年级学生编写的教科书。书中描述了很多资深的软件工程师解决软件工程问题的方法，特别是有关软件项目计划和软件质量控制方面的先进方法，并提供了很多练习来帮助读者掌握这些方法。在 Embry-Riddle 航空大学计算机科学系，以本书初稿为教材，为该系一年级学生讲授了这门课程，经验证明，这对造就学生成为合格的软件专业人员很有帮助。在每章之后还附有习题，以帮助读者复习与掌握该章的主要内容。本书可作为各类专业本科学生学习个体软件过程 PSP 的教材，也可作为工程技术人员自学个体软件过程 PSP 的教材，是进行软件过程改善和能力成熟度模型 SW-CMM 评估的重要参考资料。

图书在版编目 (CIP) 数据

个体软件过程 PSP 引论/ (美) 汉弗莱(Humphrey,W. S.) 著; 吴超英等译. —北京: 北京航空航天大学出版社, 2001.4

ISBN 7 81077 071 3

I.个... II.①汉...②吴... III.软件工程
IV.TP311.5

中国版本图书馆 CIP 数据核字 (2001) 第 23990 号

个体软件过程 PSP 引论

Watts S.Humphrey 著

吴超英、车向东 译

周伯生 校

责任编辑 曾昭奇

*

北京航空航天大学出版社出版发行

北京市学院路 37 号 (100083) 发行部电话: 82317024 传真: 82328026

<http://www.buaapress.cn.net>

E-mail: pressell@publica.bj.cninfo.net

北京市宏文印刷厂印装 各地书店经销

*

开本: 787×1092 1/16 印张: 13 字数: 330 千字

2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷 印数: 5000 册

ISBN 7-81077-071-3/TP·034 定价: 28.00 元

软件工程和过程工程丛书

编委会

主 编 周伯生

编 委 （按汉语拼音音节排序）

董士海 顾毓清 金茂忠

施伯乐 尤晋元 郑人杰

丛书序

社会生产力的基本结构表现为人类利用生产工具来加工劳动对象。生产力的进步取决于生产工具的变革与创新，而生产工具的进步又取决于人类利用资源的能力，呈现出一个由资源、工具、生产力、时代四个元素组成的连锁反应过程。物质、能量和信息是制造生产工具的三项基本资源。一个不以人的意志为转移的历史规律是，人类对资源的认识和利用必然从物质资源开始，进而到能量资源，再深入到信息资源。目前我们所处的时代是信息化的时代，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现代化的知识型经济。

知识经济不是以资金、设备、厂房等有形资产为主的经济，而是以知识、信息、技术等无形资产为基础，以知识增长率、知识利用率和知识回报率为核心，以信息数字化为代表的经济。知识经济是当代发展的主旋律，也是未来几十年发展的基本潮流。这种趋势是对我们的严峻挑战，也给我们带来千载难逢的良机。

我们所从事的软件工程和过程工程领域，是信息技术领域的重要组成部分。国际上，这一领域在前几年内取得了前所未有的进展，其成果超过过去 10 至 15 年来的成就总和。其中以英特网作为全球通信工具最为引人注目。把各种信息和知识有机地联系在一起，正在改变人们的生产方式、工作方式、学习方式、社交方式、生活方式以至思维方式。

计算机软件是计算机应用的灵魂。没有先进的软件产业，不可能有先进的信息产业。关于软件产业，在美国学术界与工业界一致认为将经历三个不同的发展阶段。第一个阶段是软件结构化生产阶段（70 年代中期至 90 年代中期），以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征。从 80 年代中期开始，软件生产开始进入以过程为中心的第二个阶段，以提出过程成熟度模型 CMM、个体软件过程 PSP 和群组软件过程 TSP 为标志，这个阶段在美国预计于 2005 年前后结束。从 1995 年开始，以软件过程、面向对象和构件重用等三项技术为基础，正在逐步进入以软件工业化生产为特征的第三个时代。

我们在十多年来的研究和开发工作实践中，一直在努力寻找适合我国特点的发展软件产业的技术途径，积累了一些经验，也有不少教训。其基本观点是：为了适应 21 世纪对信息技术的要求，软件产业必须走软件工业化生产的道路。一方面需要营造软件工程文化，培养大量既懂信息技术又懂企业管理的高级人才，建立必要的信息产业的基础设施；另一方面还需要建立过程工程、系统工程、面向对象技术、软件过程以及软件质量工程等五个支持环境，以全面支持和促进软件产业的建立和发展。下面简要论述我们对这五类环境的基本观点。

在知识经济时代，企业正面临着科技、信息、资金、人才、市场等方面的严峻考验，企业要在激烈的竞争环境下生存和发展，不能仅仅依靠缩短周期、降低成本和提高质量等传统手段，还必须根据企业内外条件的变化，对变化迅速的竞争环境，动态地、准确地、快速地作出反应，并通过决策、规划、管理、调度和运作等各个层次付诸实施，实现过程系统技术和管理技术的综合集成和整体优化。我们建议建造集成化过程工程环境来为这个目的服务，使其在柔性信息系统的研制阶段起确定需求和优化结构的先导作用，在柔性信息系统的运作阶段起监视运行和动态优化的支持作用。

集成化系统工程环境宜采用快速原型与程序变换相结合的技术,由系统需求规格说明直接生成该系统的各个功能部件及其可执行系统,然后通过系统模拟进行系统功能与性能的优化,并将该系统中的各个部件分别用硬件、软件或非计算机部件来实现,使整个系统能够合理地满足原先提出的系统需求。我们建议将集成化系统工程环境用于各类计算机应用系统的建模、模拟、分析、评估和优化,以大幅度提高软件的生产效率,同时为各类非计算机专业人员设计他所需要的软件创造条件。

面向对象技术是软件工程和过程工程领域中的重要技术,出现于 70 年代末期。它不仅是一种程序设计方法,更重要的是一种对真实世界的抽象思维方式。由于它比较自然地模拟了人类认识客观世界的方式,一经诞生就具有强大的生命力。标准建模语言 UML 不仅支持面向对象的分析与设计,还支持从需求分析开始的软件开发的全过程,是近年来最重要的、具有划时代意义的成果之一。我们认为,面向对象技术是软件产业的主要支柱技术之一,是每一个合格的软件开发人员必须掌握的基本技术;采用标准建模语言 UML,至少是近 10 年内值得提倡的途径。

50 多年来计算事业的发展使人们认识到要高效率、高质量和低成本地开发软件,必须改善软件生产的过程。软件生产转向以改善软件过程为中心,是世界各国软件产业或迟或早都要走的道路。当代世界上关于软件过程有三个流派:美国的 CMM/PSP/TSP、ISO/9000 系列以及 ISO/SPICE 标准。目前,以美国国防部支持的 CMM/PSP/TSP 流派研究得最为深入,试用得也最为广泛。这里过程成熟度模型 CMM 侧重于开发组织内外的各项管理,个体软件过程 PSP 侧重于训练开发人员的技能,群组软件过程 TSP 侧重于开发项目的组织。我们主张建造面向 CMM/PSP/TSP 的软件过程支持环境。

软件系统的质量往往是决定复杂巨系统成败的关键。软件系统质量难于保证的主要原因首先是由于软件系统固有的复杂性,次之是由于软件系统及其管理工作固有的不可见性。我们认为,采用定量软件工程有利于提高软件产品的质量,改善管理工作的可见性,但是还不可能做到完全透明,因此建造软件质量工程环境势在必行。建造软件质量工程环境,除了必须有软件过程支持环境作为基础设施之外,这个环境还应该包含相互联系但可分别独立使用直接针对软件系统质量的四个子环境:软件测试系统、软件质量度量系统、软件产品评测系统以及产品质量监控系统。

我受北京航空航天大学出版社的委托,邀请了数位软件界的资深学者组成了编委会,组织编写和翻译有关过程工程、系统工程、面向对象技术、软件过程以及软件质量工程等五个支持环境方面的丛书,重点介绍有关这方面的理论基础和工程技术,以便将国内外学术界和工业界已经积累的知识供大家学习、借鉴和使用。我希望这套正在编写和翻译的丛书能对营造学术气氛、提高学术水平和技术水平起推波助澜的作用,以适应知识经济时代的要求,促进我国建立以过程工程为中心的第二代软件产业,并尽早向以软件工业化生产为特征的第三个时代过渡。但我深深知道,这一领域的知识浩如烟海,本丛书从选题至内容难免存在不足之处,诚请专家、读者对我们提出建议、批评和指正。

北京航空航天大学软件工程研究所

周伯生

1999 年 6 月 6 日

译校者前言

本书是 Watts S.Humphrey 特地为大学一年级学生编写的教科书。该书描述了很多资深的软件工程师解决软件工程问题的方法，并提供了很多练习来帮助学生掌握这些方法。根据美国 Carnegie-Mellon 大学软件工程研究所 Humphrey 等人的实践数据，在应用了 PSP 之后，总的缺陷缺少了 58.0%，在测试阶段发现的缺陷减少了 71.9%，生产效率提高 20.8%。在美国，Embry-Riddle 航空大学计算机科学系首先以本书初稿为教材，为该系一年级学生讲授了这门课程，取得了很好的效果，接着在几十所大学中开设了这门课程。经验证明，这对造就学生成为合格的软件专业人员很有帮助。

为了在企业中推广 PSP，需要高层经理的大力支持，需要有合格的教员和合适的教材，而且还需要所有参与人员的积极参与。然而，遵循个体软件过程 PSP 开发软件并不是一件轻而易举的事，需要人们改变自己的工作方式。但是，人们通常很难改变自己的工作习惯，他们虽然愿意作些细微的改变，但往往离不开原有的习惯。只有在用他们自己的事实证明新的方法确实有效之后，才愿意真正进行改变。因此，推广个体软件过程需要以认真的实践为基础，并努力总结亲身经历的经验和教训。

个体软件过程可以作为两学期的软件工程课程的一部分，也可作为一门独立的课程。但这是一门实践性很强的学科，需要通过编写程序才能真正掌握 PSP 的精髓。为了精简讲授内容，减少重复实践，可以将 PSP 课程的程序设计练习与其它课程的作业结合起来。在最近与 Humphrey 的一次通信中，他告诉我们如果在讲授个体软件过程 PSP 之前，先讲一些统计过程控制原理，对理解 PSP 的方法和理念非常有利。

诚然，PSP 方法的成功与否，在很大程度上取决于教师激发学生学习和实践这些概念的能力。这不仅需要教师本人对 PSP 持积极态度，而且需要努力实践，认真积累数据，用自己的经验向学员展示 PSP 方法的威力。可以认为，教员是否按 PSP 原理进行认真的实践，是讲授好 PSP 课程的重要前提。

在 Embry-Riddle 航空大学计算机科学系的教员和学员为本书撰写的前言以及 Humphrey 本人在本书正文和他为本书撰写的前言中，都特别强调采集数据的重要性，并特别提出千万不要对数据进行粗制滥造，否则所收集的数据就没有什么用处。因此，我们诚恳地期望学员在学习 PSP 课程时，按正确的方法去做，努力理解 PSP 的整体框架和所蕴涵的概念，积极采集真实的数据，在自己的工作中坚持贯彻 PSP 的原则。因此，开展 PSP 培训，不仅需要学员的认真参与，而且需要在后续课程中进一步提高、巩固和扩展 PSP 方法。可以认为，这是掌握 PSP 方法的基本保证。

自从 1994 年以来，Humphrey 一直大力倡导这种方法，他在美国很多著名公司推行 PSP 方法，获得了很好的结果。而且还从 1996 年开始，大力倡导群组软件过程 TSP 方法。软件能力成熟度模型 SW-CMM、个体软件过程 PSP 和群组软件过程 TSP 三者相互配合、各有侧重，形成了一个不可分割的整体，犹如一张具有三条腿的凳子，缺一不可。在国际上，印度

的软件产业界颇为重视 SW-CMM、PSP 和 TSP 的培训和推广，因而在软件产品质量、软件项目的进度和费用等方面都得到了大幅度的改善。

此外，我们注意在美国 Carnegie-Mellon 大学软件工程研究所对外开设的课程中，对本软件过程 PSP 给予了特别的重视。其中《个体软件过程引论》是一门历时两天的课程，讨论了 PSP 的关键概念以及相关的术语和理念，并为软件工程人员参加群组软件过程作好准备。其中为软件工程师开设的 PSP 课程有两门，第一门讨论软件项目规划，第二门讨论软件产品质量，每门课都是五天。在第一门课中，除了讨论个体过程引论，还讨论规模测量、规模估计、如何基于历史数据来估计规模和资源以及如何进行过程测量等问题。第二门课主要讨论缺陷管理、设计过程、设计验证、如何将 PSP 应用于大型项目以及如何进行过程开发等问题。这两门课程都是以 Humphrey 所著的另一本书《软件工程规范》(*The Discipline for Software Engineering*) 为教本。

北京航空航天大学软件工程研究所从 1977 年开始在硕士研究生中开设《个体软件过程引论》课程，最近两年改在本科生的《软件工程》课程中讲授 PSP 的内容，今年还对一些公司的软件工程师进行了 PSP 培训，这些都取得了一定的效果。但是由于实践时间不长，数据采集不全，需要进一步坚持改善与总结提高。

从去年下半年以来，在我国软件企业界正在掀起能力成熟度模型 CMM 评估的热潮。需要着重指出的是，在软件能力成熟度模型 SW-CMM 的 18 个关键过程域中，有 12 个与个体软件过程 PSP 紧密相关，有 16 个与群组软件过程 TSP 紧密相关。因此，如果能熟悉个体软件过程 PSP 和群组软件过程 TSP，不仅有助于工程师改善工作效率，而且也非常有利于组织的过程改善。可以认为，在大学中教授 PSP 和 TSP 课程，对培养合格的软件专业人员和加速我国软件产业的发展将起积极的推动作用。

我们把《个体软件过程 PSP 引论》奉献给读者，希望能对我国软件产业的发展起到推波助澜的作用。但由于我们水平的限制，缺点和错误在所难免，敬请读者、特别是北京航空航天大学计算机科学与工程系参与个体软件过程授课的学生，对本书翻译中的缺点和错误提出批评、指正，并希望对授课方式和练习内容提出积极建议。

吴超英 车向东 周伯生

2001 年 4 月 22 日于北京

教师前言

我们在 Embry-Riddle 航空大学计算机科学系，以本书初稿为教材，为该系一年级学生讲授过程原理课程。本书包含个体软件过程（PSP）的元素和活动的主要子集，一年级学生可以像传统的程序设计课程那样来接受这些内容。本书还为引导学生进行规范化的个体实践提供一种结构。我们已经领略了以本书作为教材进行教学的好处，它对造就我们的学生成为合格的软件专业人员很有帮助。

多年来，我们一直在尝试如何对学生提供实际的软件工程经验。我们曾经在低年级课程中引入软件工程理论和实践，在高年级课程中增设小组项目，并取得了一定的成功。幸运的是，我们发现学生在做这些项目时，他们并不懂得如何进行时间管理、进度安排和质量管理的。正如软件工业化生产已经证实的，工程小组高效开发高质量软件产品的能力，在很大程度上取决于个体工程师的能力。对学生在时间管理和质量管理上的问题我们并不感到惊奇，因为我们并未给学生讲授如何计划和如何管理方面的课程。因此，我们尝试着大学低年级课程中开始介绍过程的概念。

我们感到，大学一年级新生最容易接受时间管理的实践并从中获益。所以我们在计算机科学系的第一门课程（CS1）中介绍本书第 1~10 章的内容。虽然所有选修 CS1 课程的学生都有一定的程序设计经验，但他们离正式定义的软件开发过程的要求还相距甚远。因此，在他们真正深刻理解软件工程师的角色和实践之前，需要了解现代软件开发工作的各种问题。

在完了 CS1 课程和大学第一学期其它课程的学习之后，学生已经能用更加规范化的方法来开发程序，我们采用本书第 11~20 章的内容，在 CS2 课程中介绍了 PSP 过程。这时，学生已经可以为他们的生一个程序设计项目制订计划。他们遵循所定义的 PSP 实践，使用自己的历史数据来估计项目规模、工作量和质量（缺陷分布情况），并在项目总结报告表中收订和记录每个项目的实际数据。

在进行了一年的实践之后，我们发现给计算机科学第一年级新生介绍过程活动的方法相当有效。这里说有效是指学生学会了如何使用本书所描述的过程。他们最终确实明白了记录工作量、程序规模和质量数据的价值，并且能利用这些数据计划项目和分析他们个人工作的有效性。他们在工作中收集数据，就为估计提供了定量基础。他们定期进行结构化复查，并学地在工作中遵循已定义的开发阶段（例如：计划、设计、编码、编译和后置处理）进行工作。我们也感到，如果把介绍 PSP 的课程推迟到另一个学期（或另一年），只会使不正确、无规范的程序设计实践在学生心目中更加根深蒂固，从而对改变将更加抵触。

PSP 已经帮助学生认识到规范化方法对开发软件的重要性，也为将来进一步介绍高级个体过程和群组过程奠定了更严谨的基础。学生的大部分数据是准确的，但必须仔细分析并抛弃有怀疑的数据。我们感到遗憾的是，学生的计划工作能力并无提高，很多学生仍然把作业拖到指定的交付日期才交。这是一个新程序员常见的问题。

我们毫不惊奇地发现，PSP 方法的成功与否，在很大程度上取决于教师激发学生学习和

实践这些概念的能力。我们使用本书的概念和论点鼓励学生对过程方法持积极的态度；对课堂数据提供定期的反馈和分析，以激发学生进一步考虑个人数据的兴趣；并邀请工业界专业人员在课堂上讨论他们的过程经验。这些对学生学好这门课程都很有帮助。

在开始讲授这些新课时存在一些困难。例如，我们开始没有将 PSP 与 CS1 和 CS2 课程的其它内容紧密地结合起来。因此，学生很难把时间管理活动与他们的程序高度工作相结合。此外，我们也没有对课堂的整体数据提供足够的反馈。

PSP 的一个有趣且有益的副产物是给教师提供了大量的数据。在 CS1 课程中，我们获得了学生花费在 CS1 课程活动上的时间的周报告。在 CS2 课程中，我们获得了有关每个程序设计项目规模、工作量和缺陷数据的 PSP 总结报告。这些数据经常引起课堂上对所讲授的方法以及它们如何影响程序员的生产率和程序质量进行热烈的讨论。PSP 为这些问题的详细讨论和分析提供了定理基础。

在我们的计划中，将在大学一年级继续安排讲授 PSP 课程。我们也要求修完 CS1 和 CS2 课程的学生在数据结构和算法课程中继续使用 PSP 方法。我们相信，这将为他们在三、四年级遇到复杂的群组项目时作好准备。我们还计划在他们的后续课程中指导学生进一步提高和扩展 PSP 方法。

我们已经看到，本书对学生学习专业人员的软件规范很有帮助。我们希望其他使用本书的学生和教师也能得到同样的收益。

Thomas B.Hilburn, Aboalfazl Salimi, Massood Towhidnejad
Embry-Riddle 航空大学

学生前言

当我们完成了一年级的 PSP 课程之后，Embry-Riddle 航空大学的一些教师问我们是否愿意联名为这本书写一篇前言。我们同意了。但由于我们不能肯定如何才能写好一篇前言，他们建议我们仅仅回答一些问题。下面教师向我们提出的问题以及我们对这些问题的答案：

1. 你们在 PSP 课程中做了哪些类型的工作？

我们记录了在程序设计作业和项目上花费的所有时间。为了跟踪时间数据，需要做大量的文书工作。我们还记录了程序的规模和缺陷，并用所收集的数据估计今后项目的时间、规模和缺陷数目。

2. 本书的内容合适吗？这个教材与一年级其它课程的教材是怎样相配合的？

它很好地满足了课程的要求。而且有了一个估计之后，可以帮助你正在进行的工作产生自信。

刚开始时，PSP 课程似乎对其它课程有干扰。但在课程结束时，我们认识到，这些活动对完成其它课程也确有帮助。在学习这门课程的整个过程中，我们经常问自己：“为什么我要帮这些工作？”但是后来认识到，对自己要完成的程序工作有了估计，实际上对自己很有帮助。

非常重要的一点是，千万不要对数据（时间）进行粗制滥造，否则所收集的数据就没有什么用处。

3. 你学到了什么？

除了上面已经提到的那些之外，还学习如何才能更有效地利用时间以及在上机之前要做那些准备工作。在上机之前应该完成许多纸面工作。

通过示例和讨论，还将了解自己和其他人所犯的错误。由于在上机前做了许多纸面工作，因此在进行程序设计时井井有条。我们还认为，PSP 不仅可以用于软件开发；只要对表格稍作修改，也可用于其它各种活动。

4. 对将来要使用 PSP 的学生有什么建议？

按正确的方法去做，对数据不要粗制滥造，要遵循指南。要努力理解 PSP 的整体框架和所蕴涵的概念。不要对文书工作产生抵触，它最终是有回报的。

Ben Bishop, Andrew Henderson, Michael Patrick
Embry-Riddle 航空大学

序

如果你正在学习成为一个软件工程师，这本书就是为你写的。书中描述了很多资深的工程师解决软件工程问题的方法，并提供了很多练习来帮助你掌握这些方法。每章描述了一个不同的主题，当你做完作业规定的练习之后，你将掌握这个技能。完成每个练习中的例题，对检查自己的学习情况很有帮助。

为什么我要写本书？

开发软件产品不仅是将程序设计批令汇集在一起，并使其能在一台计算机上运行，而且要求在商定的费用和进度下满足用户的需求。为此，软件工程师需要始终如一地在计划的费用下、按进度开发出高质量的程序。本书向你展示如何进行这些工作，介绍个体软件过程（PSP），PSP 是用规范化的个人实践从事高级软件工程的指南。

PSP 将向你展示如何制订计划并跟踪你的工作，并指出怎样始终如一地生产高质量的软件产品。使用 PSP 也将向你提供工作有效性的数据并识别出自己的优势和弱点。PSP 犹如运动场上的秒表和测距器，要对参赛做出明智的判定，需要通过测量来了解自己的擅长以及那些地方需要改善。就像一个田径队，软件工程也有许多专业，软件工程师需要各种技能和才干。为了事业的成功，需要了解和改善自己的技能和才智，并在工作中充分利用自己独特的才能。PSP 对帮助你做到这一点很有帮助。

使用 PSP

通过使用 PSP，你将对专业软件工程师经过多年的试验和失败所开发（研究、探索）出来的各种技能和方法进行实践。在前人经验的基础上前进，将会学得更快，而且可以避免重复前人所犯的错误。作为一个专业人员，关键是要了解别人已经做过什么，并注意吸取他们的经验和教训。

学生将如何受益

虽然现在通常在研究生的软件工程课程中介绍 PSP，但其基本原理也可以大学本科新生中讲授和实践。本书旨在学生学习其它课程的同时，循序渐进地了解 PSP 方法。当阅读每一章之后，要完成课后的练习。这些内容将向你展示如何管理时间，如何制订计划并跟踪你的工作，以及怎样始终如一地生产高质量的程序。

要掌握有效的技能和习惯需要花费时间，因此在每一个软件作业中都应该用 PSP 方法进行实践。如果你这样做了，在从事软件工程师工作需要这些技能之前，你就已经学习、实践和完善了这些技能。

已经工作的工程师将如何利用本书

已经工作的软件工程师也可从本书学习 PSP 的基础。我建议你从头到尾做一遍书中的

练习，并将它们作为改善你的日常工作方法的指南。实践每个练习直到熟悉这种方法，然后再阅读下一章，以增添新的方法。然后采取同样的方法，在进行下一步学习之前，既要实践新方法，也要实践以前学过的方法。关键是一定要花时间来掌握某一个方法，然后才学习下一个方法。

只要坚持并遵循一定的规范，自学本书不应该感到困难。但若在一个班级中或几个同事组织起来学习，可以互相交流经验并分享见解，这样做效果会更好。无论采取哪一种方式，大约每周要花费 1~2 个小时学习 PSP 课本，记录和分析有关数据，并把 PSP 方法应用于你的工作。虽然学习 PSP 所需要花费的时间与你现在的习惯和实践有关，但若一旦完成了本书的学习，对你今后专业生涯的持续开发就奠定了坚实的基础。然而应该注意，学习 PSP 的关键是要对你的工作数据进行考察和思考，看看这些数据对你的个人性能说明些什么。

对教员的一些建议

本书是作为传统上两学期的计算机科学或软件工程课程的一部分而设计的，只要求大学一年级新生的程度。本书一步一步地介绍 PSP，学生可以在他们的日常课程中使用。前 10 章的练习都是一般性的，既可以面向程序设计工作，也可以面向其它工作。后 10 章的练习是为 6~8 小时或更多一点时间的小型程序设计练习而设计的。

虽然有些学生在进入大学后才第一次学习程序设计，但现在有很多学生在高中就学习程序设计的基础知识。因此将本书设计成既可在第一门程序设计课程中使用，也可在更高级的程序设计课程中使用。无论学生是否已经学过程序设计，他们都应该很容易理解本书的内容，并会立刻发现本书对他们非常有用。

本书是 PSP 引论，而不是 PSP 的全部内容。例如，本书没有包含准确估计或数据分析所需要的统计技术；也没有包含适用地大项目的 PSP 方法，即没有包含将 PSP 用于编写大型程序时所需要的过程定义和过程改善技术。因此，全面介绍 PSP 技术应在学生教育计划的后期进行。¹

当你指导学生学习本书并完成企业时，在第一学期，学生将学习跟踪和监控工作、管理时间以及制订计划的方法。在第二学期，学生将学习有关程序质量、进行复查以及利用各种质量测量和质量管理的办法。他们也将学习有关缺陷、引起缺陷的原因以及工程师对他们生的产品质量应该承诺的个人责任。在第二学期课程结束时，学生们可以掌握 PSP 的基本内容。为了在这个基础上继续深造，并帮助他们各累使用这种方法的经验，在后续课程中应该要求学生继续使用 PSP。

教学方案

由于本书是两个学期的计算机科学或软件工稇导论课程中使用，所以把本书的内容分成两部分，在第一学期中讲授时间管理（前 10 章），在第二学期中讲授质量问题。在两个学期中讲授本书共需要 6 个课内学时。由于学生在他们现在进行的课程中使用 PSP 方法，讲授本书并没有明显加重学生的课业负担。学生在学习这种方法上所花费的时间，都将在其它

¹ 在我的另一本教科书《软件工程规范》（*A Discipline for Software Engineering*, Reading MA: Addison-Wesley, 1995）中，对 PSP 和 PSP 课程有更为详细的描述。在该书的辅助材料中，有教员指南以及包含课堂投影胶片和作业材料的光盘。——作者注。

工作中提高效率而得到补偿。

当讲授这本书时，应在每章的课后安排作业。经验证明，最好在第一学期有前几周讲完前 10 章的内容。然后在该学期的后几周对所介绍的方法进行实践。第二学期应该遵循同样的策略，在前几周进行 PSP 的课堂讲授，在后几周对所介绍的方法进行实践。

特别重要的是，应把本书所讲授的内容作为整个计算机科学或软件工领导课程的有机组成部分。要对学生讲清，本书所介绍的各种方法是软件工程方法的精髓，学生必须学习和实践。当你安排每一个作业时，还要向学生说明，他们的分数不仅依赖于他们工作的质量，也依赖于他们运用 PSP 方法的水平。他们必须完成每个 PSP 练习，并且在介绍该 PSP 方法后继续使用它。课程方案、建议讲授的内容以及作业所需要的工具等，在本书最后一页的教师指南和支持材料中介绍。

教员准备

在讲授这门课程的过程中，你将会发现自己使用这种方法也有好处。例如，你可以使用计划和时间管理的方法来准备课堂讲座或批改作业。在你亲身使用 PSP 方法之后，将会更加赞同需要个体规范。有了这种背景，对你向学生解释并指导他们使用 PSP 方法都有好处。当学生发现你已经使用过 PSP，他们也就更有可能使用它。

感谢

在撰写本书时，我要特别感谢 Embry-Riddle 航空大学计算机科学的全体教职员。他们鼓励我编写本书并友好地评阅了我的手稿。由于他们在给几个新生班级教课，并且自己也在使用 PSP 方法，因此他们给我提出了很多有益的建议。在此我要特别感谢 Tom Hilburn、Iraj Hirmanpour、Aboalfazl Salimi、Davie Srachet 和 Massood Towhidnejad 教授对我的支持和鼓励。我也要感谢他们的三位学生 Ben Bishop、Guillermo Jose Hernandez 和 Richard Rickert 为我提供了他们的课程数据和课堂经验。此外，我还要感谢 Ben Bishop、Andrew Henderson 和 Michael Patrick 在本书的学生前言中非常诚恳地描述了他们使用本书的经验。

在 SEI 和其他单位的一些朋友和同事也友好地评阅了我的手稿，并提出了许多有益的建议和评注。我很感谢 Steve Burke、Howie Dow、John Eikenberry、Andy Huber、Julia Mullaney、Glenn Rosander、Marie Silverthorn 和 Bob Stoddard。我的秘书 Marlene MacDonald 在阅读和评注部分手稿以及在分发手稿进行评阅的过程中，也给了我很大的帮助。我还要感谢 Peter Gordon 和 Helen Goldstein 以及 Addison-Wesley 的专业人员在本书的出版过程中的有益支持。

最后，我为有一个很好的妻子感到骄傲。在我编写一本又一本书的过程中，我的妻子 Barbara 给我极大的帮助。就像我在出版其它书时一样，在本书交付出版之前，总是由好阅读本书的最后一版手稿。虽然我得到这么多的帮助和支持，但我深信，无论是 Barbara 或给我帮助的很多同事，都不可能发现我所有的疏忽和错误。所有留在书中的错误完全由我自己负责。

Watts S.Humphrey
Sarasota,Florida

目 录

第1章 软件工程师的任务

1.1 什么是软件工程	(1)
1.2 为什么工程质量很重要	(1)
1.3 个体软件过程	(2)
1.4 高质量工作的规范	(2)
1.5 高质量工作的重要性	(3)
1.6 如何提高工作质量	(3)
1.7 过程改进的步骤	(4)
1.8 本书的策略	(4)
1.9 总 结	(5)
1.10 练习1	(5)

第2章 时间管理

2.1 时间管理的逻辑原理	(6)
2.2 了解时间的使用情况	(7)
2.3 工程记事本	(7)
2.4 工程记事本的设计	(8)
2.5 工程记事本示例	(9)
2.6 总 结	(10)
2.7 练习2	(10)

第3章 时间跟踪

3.1 为什么要跟踪时间	(11)
3.2 记录时间数据	(11)
3.3 跟踪时间	(12)
3.4 使用标准的时间记录日志	(12)
3.5 处理中断	(15)
3.6 跟踪已完成的任务	(15)
3.7 在工程记事本中登记时间日志	(16)
3.8 时间记录的提示	(17)
3.9 总 结	(17)
3.10 练习3	(17)

第4章 阶段计划与产品计划

4.1 阶段计划和产品计划	(18)
---------------------	------

4.2 周活动总结表	(19)
4.3 总结每周的时间分配	(20)
4.4 计算阶段时间和工作效率	(22)
4.5 使用周活动总结表	(25)
4.6 总 结	(26)
4.7 练习 4	(26)

第 5 章 产品计划

5.1 产品计划的必要性	(27)
5.2 产品计划的用途	(27)
5.3 什么是产品计划	(28)
5.4 产品计划	(28)
5.5 制订小型任务的计划	(28)
5.6 术语定义	(29)
5.7 作业编号日志	(29)
5.8 关于使用作业编号日志的几点建议	(33)
5.9 使用产品的时间和效率数据	(33)
5.10 总 结	(34)
5.11 练习 5	(34)

第 6 章 产品规模

6.1 产品计划过程	(35)
6.2 规模测量	(35)
6.3 使用规模测量的注意事项	(36)
6.4 程序规模	(36)
6.5 其它的规模测量方法	(38)
6.6 程序规模估计	(39)
6.7 较大规模的估计	(39)
6.8 在作业编号日志中使用规模测量的方法	(41)
6.9 总 结	(44)
6.10 练习 6	(44)

第 7 章 管理好时间

7.1 时间管理的要素	(45)
7.2 活动分类	(45)
7.3 收集活动的时间数据	(45)
7.4 时间分配的评价	(46)
7.5 作出时间安排	(46)
7.6 找出更多的时间	(48)

7.7	制订基本规则	(48)
7.8	设定时间分配的优先级	(50)
7.9	管理好时间安排	(52)
7.10	关于管理可变动的时间的几点建议	(52)
7.11	时间管理的目标	(53)
7.12	总 结	(53)
7.13	练习 7	(54)
第 8 章 契约的管理		
8.1	什么是契约	(55)
8.2	认真制订契约	(56)
8.3	契约的一个示例	(57)
8.4	工业中的示例	(58)
8.5	处理没有完成的契约	(58)
8.6	管理契约的重要性	(59)
8.7	不对契约进行管理的后果	(59)
8.8	管理契约的方法	(60)
8.9	总 结	(61)
8.10	练习 8	(61)
第 9 章 进度管理		
9.1	进度管理的必要性	(62)
9.2	Gantt 图	(63)
9.3	制订项目进度表	(63)
9.4	检查点	(64)
9.5	跟踪项目计划	(65)
9.6	跟踪积分	(67)
9.7	总 结	(70)
9.8	练习 9	(70)
第 10 章 项目计划		
10.1	项目计划的必要性	(71)
10.2	项目计划总结表	(71)
10.3	项目总结	(73)
10.4	程序规模	(75)
10.5	开发阶段的时间	(76)
10.6	估计的准确性	(78)
10.7	总 结	(78)
10.8	练习 10	(78)