<<团队软件过程>>

图书基本信息

书名：<<团队软件过程>>

13位ISBN编号：9787115251893

10位ISBN编号：7115251894

出版时间：2011-6

出版时间：人民邮电出版社

作者：Watts S.Humphrey

页数：382

<<团队软件过程>>

前言

前言　　这本书是为已经学习并使用过个体软件过程（PSP）的学生和工程师准备的。
你可能在研究生课程、高级培训课程或者初级导论课程中学过PSP。
或者，你可能是工程师，正在探索如何在实际的团队工作环境中应用PSP。
不论是哪种情况，只要你学过PSP，你就具备了使用本书中的方法和实践的基础。
学习过PSP之后，你可能需要指导，以便弄清如何将它应用于软件过程的诸多任务上。
这就是团队软件过程（TSP）的主旨：为在软件开发领域中应用成熟的工程学方法提供一个框架。
关于团队协作有很多东西可讲，本书只涵盖了最基本的内容。
TSPi（团队软件过程导论）介绍了团队的概念，组建团队的基本步骤，以及在团队中工作的方法。
但是要注意，本书是为导论课程设计的，并没有涵盖在大规模工业项目中使用TSP的全部资料。
TSPi是如何帮助工程师的　　本书向工程师介绍有关软件开发团队协作的内容。
TSPi提供了一系列结构化的步骤，告诉工程师每一步该做什么，并且详细阐述了如何将这些步骤连接起来以开发完整的产品。
TSPi还提供了两个饶有趣味且相对具有挑战性的项目练习，每个练习都具有适当的规模。
既不过大，以保证在几周之内可以完成：也不过小，以保证可以模拟典型的小规模项目。
如果有能力的工程师遵循本书的指导，那么他们一定能开发出完整的工作产品。
在本书建议的TSPi策略中，团队在两到三个周期内开发一个产品。
在第一个周期内，团队构建工作产品内核。
在后续的每个周期内，逐渐向内核加入新功能。
这种策略体现了使用历史项目数据来制定新项目计划的好处。
另外，由于在每个周期中都有新的角色，工程师们可以在一个项目中就经历两到三种不同类型的工作。
经过几个开发周期后，工程师们就会对团队协作方法有一个广泛的了解，这样一来，他们就更有可能在自己将来的工作中继续使用TSPi方法。
为何要学习TSPi课程　　经实践证明，在培养软件工程专业的学生方面，项目课程是行之有效的，因此，越来越多的大学开始开设相关课程。
这些课程很受学生欢迎，修习人数一般都会超过预期。
学生想学习可应用于将来工作的知识，而团队训练课程正好满足了他们的需要。
毕业后的学生及其雇主普遍反映，软件项目课程为实际工作打下了良好的基础。
现在，在团队项目课程方面，已经有很多有益的实践。
尽管大多数此类课程都是成功的，但普遍有三个问题：第一，学生经常去尝试过大的项目；第二，他们大多只强调产品而忽略了过程；第三，总有一些成员会破坏团队气氛。
虽然TSPi不能完全避免这些问题，但它提供了有关避免这些问题或减少这些问题影响的有益指导。
为了高效利用课程时间，团队软件课程应该精心组织，并且要基于经实践证明的项目经验。
如果没有明确定义的过程或结构化的团队框架，工程师就必须自己决定该如何完成项目。
如果没有过程和框架，工作组就必须自己学习团队建设和团队协作的基本要素，而这一过程通常都会伴随着痛苦的尝试和失败。
显然，这样做代价高昂并且没有必要，因为团队协作的原则通常众所周知、简单明了。
TSPi以行之有效的团队协作方法训练工程师，首先帮助他们熟悉团队建设的过程，然后指导他们使用经过明确定义和度量的框架来开发产品。
在经过PSP训练的前提下，工程师一般都能遵循TSPi脚本步骤，并使用TSPi支持工具来计划和管理自己的工作。
遵循TSPi的指导，项目工作会变得更有效率，工程师也可以更加专注于学习软件工程，而不是在团队建设和团队管理方面花费大量的时间。
TSPi明确定义了团队角色，每个团队成员都要以某种角色进行工作。
每个角色都有详细说明，指出这个角色应该做什么，何时以及如何去完成任务。

在每个团队成员都知道他们自己和其他人应该做什么的情况下，他们就能更好地作为一个团队高效工作。
如果一个团队成员没有完成工作，其他团队成员就会知道相关情况，从而及时地采取措施处理相关问题。
如果团队不能独立解决人际关系问题，他们可以求助于教师或管理人员。
本书的教师手册给出了处理很多常见的团队协作问题的有效方法。
　　如果学生团队成员担任明确的角色和职责，并使大家都了解，教师就能给出更加公平和详细的成绩。
除了给整个团队打分，还可以给每个人的表现打分。
这不仅激励学生表现得更好，同时也是一种更公平的给团队训练课程打分的方式。
　　本书组织结构　　本书是为引导团队学习TSPi过程而设计的。
前两章（第一部分）是简介，第二部分阐述了完整的团队开发周期。
书中详细解释了过程脚本，并且给出了TSPi表格的完整示例。
　　第三部分给出了TSPi团队成员角色的详细说明：团队领导、开发经理、计划经理、质量和过程经理，以及支持经理。
当阅读有关每个角色的章节后，你可以将TSPi角色脚本作为工作参考。
　　TSPi课程伊始，每个学生都要填写一个INFO表格（见附录F），这个表格记录了有关学生兴趣和背景的信息。
教师根据这些信息将整个班级划分为5人小组，再给每个小组成员分配初始角色。
如果个别小组有4或6个人，教师就必须对角色进行适当调整。
每个角色都要有人担任，每个工程师必须担任至少一个角色。
对于一个4人的小组，支持经理角色的任务要在所有小组成员间分摊。
对于一个6人小组，质量和过程经理角色要分割为两个角色：质量经理和过程经理。
　　选择团队成员并进行角色分配之后，各个团队就启动各自的项目，并定期汇报项目进展。
每个开发周期结束时，工程师要评估团队的整体表现和每个角色的个人表现。
基于这些信息，教师就能评估每个团队和每个人的工作，并且在后续的开发周期中更好地分配角色。
如果有必要，教师可以对团队的人员组成进行调整，但是，除非出现了严重问题，否则团队的人员组成应该在整个课程中保持稳定。
　　使用标准及预定义的问题　　尽管TSPi可以应用于各类项目，但本书只提供了两个标准和预定义的问题，它们是为满足课程多样性的要求而专门设计的。
虽然使用真实的客户需求也有好处，但是，因为以下三个原因，我们不推荐这样做。
第一，课程有严格的进度要求。
虽然多数客户一开始都同意按照固定的时间表推进项目，但是很少有客户真正知道开发软件需要多长时间。
另外，因为入门工程师一般不知道如何按照严格的进度来管理项目，所以项目失败的几率是很高的。
这个问题的根源在于，实际的客户需求大多既不够清晰也不稳定，导致频繁变更和大量延期。
　　第二，团队协作课程应该为特定主题而设计。
虽然开发工作产品是项目的目标之一，但是本课程的主要目的是展示使用成熟的软件工程方法的好处。
对于真实的客户需求，满足客户要求永远都是最高优先级。
一旦需求变更或客户插进来回答问题，工作就会延期，进度就会压缩，导致团队经常把精力集中在完成产品上，而忽略了过程。
从结果来看，事与愿违，我们从这类课程中得到的主要经验就是如何避免开发软件。
　　第三，使用标准及预定义的问题，有助于比较各团队的表现。
同样的需求有不同的实现方式，所有的团队都可以加入课程评价。
每个团队都可以介绍自己的方案，解释有关设计、实现和测试的问题。
这个过程能够充分体现出各种开发方案的实际效果，同时也为评价将来的团队提供了参考数据。

虽然使用预定义的标准练习具有很多好处，但是这也让学生没有机会接触某些重要问题。
例如，如果没有实际经验，就很难察觉用户需求描述的混乱和模糊。
应付模糊和多变的需求是一种重要的经历，但是这可以在专门讲授需求过程的课程中进行详细学习。
本书采取这种方案，首先是要讲授高效的团队协作和过程方法，然后，在后续课程中集中研究大规模开发项目中的复杂问题。

给教师的建议　　本书可以多种方式使用。
最主要的用法就是作为一学期或两学期的团队训练课程的教材。
这种情况下，TSPi可用来开发一个单独的产品，例如附录A中介绍的两个产品之一。
一学期的课程大概有2-3个周期，而两学期的课程大概有3个或者更多的周期，以开发更大规模的产品或者附录A中产品的完整版本。
过程步骤可根据项目工作的规模进行适当增加和减少。
表P1、P2、P3给出了3种课程方案。

在表P1所示的每个开发周期内，团队计划并跟踪每一步工作，最终完成一个完整的小型项目，包括需求、设计、编码和测试。
在每个开发周期结束时，团队评估整个团队和每个角色的表现，之后，教师重新分配团队角色。
在一个包含3个开发周期的项目中，工程师们实质上获得了3个完整项目和3种不同团队角色的经验。
同时，他们也得到了每个开发周期的数据，在每个开发周期中，他们都可以学习如何使用从前面的开发周期中获得的经验。

本书还可在其他课程中作为团队协作练习来使用。
小型项目可以在3～7周的单个开发周期中完成，例如，简单的需求开发周期大概需要3～4周，设计开发周期大概需要四周或五周，而最短的完整开发项目可能需要6周或7周。
表P2展示了一个时间跨度为几周的，开发一系列需求的团队项目。
类似地，表P3展示了一个设计项目。
本书还可用于半个学期的课程教学，这样一来，完整的3个周期的课程需要15周，两个周期的课程需要11周，单个周期的开发项目只需要7周。

不管采用哪种课程方案，标准TSPi脚本都将指导学生成立团队、计划并实施项目。
除非团队已经有过TSPi课程经历，否则任何团队都不太可能在少于3周或4周的时间内完成任何一个项目周期。
原因在于，新的团队成员需要时间去学习团队过程，还要学会如何作为一个团队一起工作。
这也是第一个TSPi周期需要7周时间，而后续周期只需要4周时间的原因所在。

学习本课程的基础　　本课程的主要先修课是PSP课程，无论在研究生课程中或在PSP导论课程学习过都可以。
如果学生是在几个学期以前学习的PSP，则要求他们在最近的课程中使用过PSP，否则，他们就需要一到两节课来回顾一下PSP计划、数据收集和质量管理的相关知识。
如果学生使用PSP的经验不足或者根本没有用过PSP，那么就需要在整个课程期间对他们进行仔细的指导和帮助。

在尝试团队项目以前，学生应该具有软件设计和软件需求的知识背景，配置管理、项目管理和软件测试的相关知识和经验也会很有帮助。
另外，学生还必须熟练掌握一门编程语言，熟练使用编程工具。

Watts S.Humphrey　　Sarasota，Florida

<<团队软件过程>>

内容概要

《团队软件过程(第2版)》(简称“tspi”)，是美国embry-riddle aeronautical大学为计算机科学系研究生和高年级本科生开设的一门软件工程课的教科书。
这本书系统地论述了如何以开发团队的形式来进行软件的开发，并对开发过程作出了具体而详尽的指导，包括团队成员之间的协调、进度的管理、质量的控制等令读者最感兴趣的方面。

《团队软件过程(第2版)》内容包括四个部分：第一部分——绪论，包括前两章，是对理论的简单介绍，介绍了什么是tspi、tspi的组织结构等内容。
第二部分——tspi过程，包括第3章到第10章，则是整个小组研究周期的详细内容，详细解释了小组软件开发的步骤，并且给出了
tspi完整形式的例子。
第三部分——小组角色，包括第11章到第15章，提供了小组成员角色的细致描述：小组领导者、开发经理、计划经理、质量/进度监督经理，以及技术支持经理。
第四部分——使用tspi，包括第16章到第18章，讲述了在使用本书的过程中需要注意的一些原则。

《团队软件过程(第2版)》实用性与可读性较强，适用于软件开发项目经理、程序员和一般编程爱好者在开发软件时参考，也可作为高等学校计算机软件工程课程的参考书使用。

<<团队软件过程>>

作者简介

Watts S.Humphrey 是位知名作者，在软件开发过程和软件过程改进方面著有多本影响深远的图书：Managing the Software Process(1989)、A Discipline for Software Engineering(1995)、Managing Technical People(1997)。

Humphrey曾长期在IBM公司担任高级软件开发经理，获得了大量的软件开发过程方面的经验，是卡内基梅隆大学软件工程研究所的研究员，就软件质量和软件过程方面的主题著书立说、提供咨询，并在世界各地发表这方面的演讲。

<<团队软件过程>>

书籍目录

<<团队软件过程>>

版权说明