



# 应用个人软件程序改善软件开发流程

沈文祥

(侨光科技大学 信息管理系,台湾 台中 40721)

**摘要:** 软件流程一直是软件工程研究中一个重要的课题。对改善企业的软件流程而言,能力成熟度整合模式(CMMI)是现今一个被广泛应用的框架。CMMI 的重要元素之一是在流程定义范畴中包含量测之量化管理基本概念。相对于组织流程改善的 CMMI 方法论,个人软件流程(PSP)是一个训练软件工程师改善个人流程的方法。它被广泛的使用于训练学生和企业的工程师。然而,在现有的文献中缺乏系统化的 PSP 训练成效评估机制。透过系统化文献整理确认 PSP 训练与软件工程连动价值为何,比较评估方法中,提出一种基于 PSP 量测基础之评估 PSP 影响分析模型,之后藉由实证性研究分析所收集的数据,并提出总结性报告。

**关键词:** 能力成熟度整合模式;个人软件流程;软件流程改善

中图分类号: TP311.52

文献标识码: A

文章编号: 1671-931X (2013) 03-0074-06

## 一、前言

信息软件产业从 80 年代早期开始就一直在台湾成长。根据资策会 MIC 研究,2007 年台湾信息软件产业产值可达新台币 2,256 亿元,较 2006 年成长 7.0%。预估 2010 年产值可达 2,907 亿元,2006-2010 年之复合年成长率(CAGR)达 8.4%。从产业结构分析,2007 年嵌入式软件占总产值 3.3%,达 74 亿元,较 2006 年成长 25.9%,预估 2010 年产值突破 118 亿元,2006-2010 年 CAGR 达 19.0%。在计算机软件部分,2007 年产值达 2,182 亿元,较 2006 年成长 6.5%,预估 2006-2010 年 CAGR 达 8.0%。<sup>[1]</sup>台湾地区计算机软件的需求在过去几年来,已经大幅的增加。不久之后信息服务业的收入将超过硬件产业。因为软件应用能降低企业营运的费用,增加企业的竞争优势,并且改进行政管理部门的行政效率,所以增加软件开发成果的质量和提高了生产力,对于台湾

有关方面维持良好的经济是十分重要的。

软件项目的开发经常面临无法符合预定成本,以及准时交付高质量产品的困难。<sup>[2]</sup>研究显示,运用个人软件流程(Personal Software Process,PSP)可以有效地帮助软件工程师了解其软件开发的能力、协助项目的规划与管理、降低软件缺陷的数目和提升个人的生产力,让软件工程师能在预计的时间和成本内开发出高质量的软件,并促进其软件流程的可预测性。

由于软件在许多领域的重要性日益增加,使得程序设计教学被许多科系视为必修的基础课程。目前一般的程序设计课程,教师皆会安排若干课后作业,作为工程师课后练习,其目的不外乎是希望能藉此提高工程师程序撰写的能力,并让教师可以了解工程师学习的情况。一般而言,透过分析缴交的作业程序功能是否达到需求规定,教师可以了解工程师是否具有相当程度的程序撰写能力。然而若能了解工程师在程序练习过程中的一些信息,例如:练习过

收稿日期:2012-11-18

作者简介:沈文祥(1963-),男,侨光科技大学信息管理系助理教授,研究方向:软件工程。

程所出现的程序错误,以及程序撰写所花费的时间等<sup>[3][17][23]</sup>,教师将更能了解工程师的努力程度、练习过程所遭遇的问题、以及工程师程序撰写的能力指针,并可作为教学改善的参考。

要了解与提升个人程序开发的能力,即透过收集个人其程序开发流程的资料,包含程序大小、开发时间、和缺陷相关资料等,了解个人程序开发的能力,并藉由分析这些资料,找出流程可能存在的问题。同时,针对这些问题改变其流程,以改善其程序的质量与生产力(productivity),达到提升个人软件开发的能力。<sup>[15][8][10][12][14][22]</sup>

## 二、个人软件流程评估

个人软件流程是在 1995 年由美国 Carnegie Mellon 大学软件工程研究所 (CMU/SEI)Watts s. Humphrey 领导开发。PSP 是一种可用于控制、管理和改进个人工作方式的自我改善过程,是一个包括软件开发表格、说明的结构化架构。借助PSP提供的度量和分析工具,了解自己的技能水平,控制和管理自己的工作方式,使自己日常工作的估计、规划和预测更加准确更有效率,进而改进个人的工作表现,提高个人的工作质量和产量。

### (一)个人软件流程

PSP 为基于个人和小型团队软件流程的优化提供了具体有效的途径,如制订计划、控制质量、与其他人合作等等。在软件设计时间,PSP 的着眼点在于软件缺失的预防,其具体办法是强化设计的准则,而不是设计方法的选择。

PSP 的研究结果还发现,绝大多数软件缺失是由于对问题的错误理解或简单的错误所造成的,很少是因为技术问题而产生的<sup>[13]</sup>。而且根据多年来的软件工程统计资料发现,如果在设计时间埋下一个缺失,则这个缺失在程序设计阶段会引发 3 到 5 个新的缺失<sup>[21]</sup>,要修复这些缺失所花的费用要比修复这个设计缺失所花的费用增加了相当的成本。因此实施 PSP 为确保软件质量的一个重要途径,尤其是要提高软件设计的质量。

PSP 虽然立意良好,可是在实作上却有很多问题点。其最大的问题点就是需要手动填写很多的表格,因此愿意确实使用 PSP 方法的人并不多。而且,在 PSP 课程中,学员常会使用不同作业平台和不同的开发工具来练习 PSP 作业,但是,在收集 PSP 资料时,常常会遭遇到困难。不同开发工具所收集的 PSP 资料,无法共享利用,导致收集的资料不够完整,使得 PSP 课程的效益大大减低,因为度量的资料不够准确,就无法了解自己的技能水平,也没办法改进个人的工作表现。

### (二)PSP 流程演进

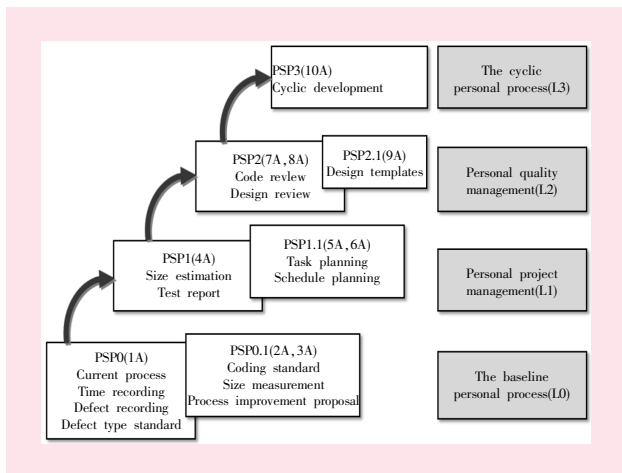


图 1 PSP 生命周期演进阶段

PSP 提供一个渐进式的流程改善机制,开发者可以循着 PSP0 到 PSP3 的步骤,逐步的改善个人的开发质量。图 1 为 PSP 流程演进,以下简述各个流程的主要内容<sup>[23][25]</sup>。

#### 1.PSP0 和 PSP0.1: 个人度量管理(Personal Measurement Management)

提供表格或流程脚本用来收集真实的资料,包含开发的时间及编译时的错误,并且把这些资料当作基线,用于未来活动的比较。

PSP0 包含等计划(Plan)、设计(Design)、程序代码撰写(Code)、编译(Compile)、测试(Test)及验收(Postmortem)等阶段。

PSP0 纪录表格:时间纪录表(Time Recording Log)、程序错误纪录表(Defect Recording Log)、程序错误标准分类表(Defect Type Standard)。

PSP0.1 包含 PSP0 内容,另外追加程序代码撰写标准(PSP 0.1 Coding Standard)、程序大小量测(PSP 0.1 Size Measurement)和程序改进建议书(PSP0.1 Process Improvement Proposal,PIP)。

#### 2.PSP1 和 PSP1.1: 个人开发计划程序(Personal Planning Process)

PSP1 包含利用先前所收集的历史资料,使用 PROBE(PROxy-Based Estimating)的方法来预估程序的大小及所需发展的时间。

PSP1.1 包含 PSP1 内容,另外追加计划工作(PSP 1.1 Task Planning)、计划时程(PSP 1.1 Schedule Planning)及测试报告(Test Report)

#### 3.PSP2 和 PSP2.1: 个人质量管理(Personal Quality Management)

PSP2 包含设计审查(PSP 2 Design Review)及程序审查(PSP 2 Code Review)。

PSP2.1 包含 PSP2 内容,另外追加设计模板(PSP 2.1 Design Template)。

#### 4.PSP3:循环个人程序(Cyclic Personal Process)

扩大应用项目(PSP 3 Scaling Up):将之前的训

表 1 PSP 相关量测参数

参数	定义	表示影响能力	工程价值	参考文献
SEA	Size estimation accuracy	Ability to measure software size	Measurability	[3][17][11][7][18]
TEA	Time estimation accuracy	Ability to predict development time	Predictability	[3][11][17][18]
APR	Actual time to plan time ratio	Ability to meet project deadline	Predictability/ controllability	[20]
DEA	Defect estimation accuracy	Ability to predict defect frequency	Predictability/Traceability	[11][18]
DRR	Defect remove rate in compile and test phase	Ability to decrease defect in compile and test	Controllability	[3][5][11][17][19]
YLD	Yield in percent of defects injected before the first compile that are removed before the first compile	Ability to control defect ere compile	Manageability	[3][5][11][17][19]
DDS	Number of defects removed per thousand lines of code developed	Ability to decrease defect density	Reliability	[3][9][17][18][19]
LOC	Total new and changed LOC developed divided by the total development hours	Ability to increase productivity	Efficiency	[3][7][11][5][19]
AFR	Time spent in design review and code review as a percentage of time spent in compile and test	Ability to manage risk	Reliability/ efficiency	[3][6][5][17][19]

练应用在大的项目上。

循环发展(PSP 3 Cyclic Development):PSP3 被设计去满足渐进式生命周期,藉由将项目分割成数个可被 PSP2.1 完成的组件,如此做循环的发展项目系统。

三、研究内容与结果

从 PSP 演进流程主要的历史研究,Humphrey<sup>[25]</sup>及 Hayes<sup>[3]</sup>等学者以三个基本量测值:程序大小、时间与错误率量测推导 PSP 量测值,汇总各学者的量测参数,本研究提出 9 项量测指标(表 1)并针对五个面向评估 PSP 效能,并针对回馈问卷了解 PSP 可能潜在的问题。

(一)PSP 量测参数

文献中<sup>[3][17][11]</sup>相关的预估参数包含有 SEA、TEA 及 DEA,项目能在化时间内完成的指标 APR,APR 指标亦符合 Boehm 的 COCOMO 模型的 earned value 概念。从错误率的观点相关于预测能力的指标有 DRR、YLD 及 DDS 亦是确认 PSP 训练效能之重要指标,DRR 指的是编译与测试阶段之错误移除率,YLD 是第一次编译程序前错误移除的百分比,而 DDS 表示每千行程序错误移除的个数<sup>[3][17]</sup>。最后阶段的评估指标有 LPH 定义为每小时产生的程序代码行数,亦称为生产力指针;AFR 是前阶(计划、分析及编程阶段)与后阶(编译与测试阶段)所花费时间的比值。正如一个有纪律的软件工程师必须展现错误排除与高质量流程的能力,而 AFR 与 LPH 两项总结指标可以说是有纪律的软件流程开发之全面质量管理(Total Quality Management)重要指标。而综观历年相关 PSP 学者<sup>[20][3]</sup>

表 2 PSP 量测参数趋势准则

Tendency criteria	Effect variables
Approaching zero	SEA,TEA,DEA
Decreasing	DRR,DDS
Increasing	YLD,LOC
Less than 1	APR
Greater than 2	AFR

<sup>[4][7][2]</sup>所讨论的结果亦完全支持以上讨论之结果。

综合各学者研究结果,表 2 呈现所描述 PSP 影响参数的趋势准则,其中 SEA、TEA 及 DEA 预估准确度的计算方式采用绝对值处理,避免 Humphrey<sup>[17]</sup>与 Hayes<sup>[3]</sup>所造成正负总和抵销误差问题,公式(1)表示如下:

$$\text{Estimation accuracy}=\left|\frac{(\text{Planned}-\text{Actual})}{\text{Planned}}\right|\cdots(1)$$

为能正确符合计划 APR 趋近 1,利用 Hayes 所提出的中值改善因子(Median Improvement Factor)比较上升或下降趋势,表示至少 50%以上的工程师呈现上生或下降趋势。根据 Humphrey<sup>[17]</sup>结论 AFR 事前/事后花费时间比应趋近于 2(表 2)。

(二)从五个面向探讨 PSP 效能

在这个研究中从五个面向探讨 PSP 效能,分别为程序大小(size estimation)及时间(effort estimation)估测准确度、产品质量(product quality)、流程质量(process quality)与个人生产力(personal productivity)。详细分析结果描述如下:

1.程序大小预估准确度

PSP 提供 PROXY 为基准的估测方法,帮助工程师利用历史数据分解程序估测程序大小,此一方式

让工程师能更正确的回归估测程序大小,图2显示工程师的程序大小预估效能,图中垂直坐标表示估测的次数,水平坐标表示程序大小预估的精确度,范围从-100%到+250%,从图中明显看出工程师不同阶段有显著提升预估精确度 Level 2>Level 1>Level 0。Level 2>Level 0 约两倍的次数,长尾的现象 Level 2>Level 表示偏离预估次数显著降低。

### 2.时间预估准确度

利用流程所花费时间数据推导出劳力(Effort)的预估情形,图3显示 Level 0~Level 1 高估时间花费,而 Level 2则较为对称,且有较多次的时间预估精确(0%次数最高)。

### 3.缺失密度

一般以产品缺失数目的多寡表示软件产品的质量,PSP 以此一方式评估产品的质量,图4呈现明显于编译与测试所技术的缺失数目明显随阶段的增加而下降。图5更呈现缺失数的变化情形,水平坐标表示于编译与测试阶段每仟行程序移除的缺失数,垂

直轴表示受测工程师观测的次数。

缺失及早移除效率(Defect Yield)是 PSP 效能评估重要指标<sup>[13]</sup>,降低软件开发后阶段(测试阶段或产品递交后)的缺失是软件开发的重要目标,测试阶段缺失愈多表示产品的质量愈差,因为修复后阶段的缺失花费远比及早发现缺失修复成本高出许多,PSP 训练教导工程师有效的缺失移除来保证流程的质量。一般说来希望缺失移除率能达到 100%,也就是说在第一次编译前就完全移除所有的缺失。图6表示 PSP 各阶段缺失移除效率的趋势,图中呈现 Level 1 较 Level 2 平均缺失移除效率由 0.5%提升到 42%有明显成长,因为在 Level 2 导入设计审查(design review)与编码审查(code review)的作为。更详细的缺失效率的比较呈现在图7中,细分每次作业阶段缺失移除效率的比较,图中 Level 1 前在第一次编译前呈现非常差的缺失移除效率,从第7次作业起,超过 80%的工程师在第一次编译前完成移除

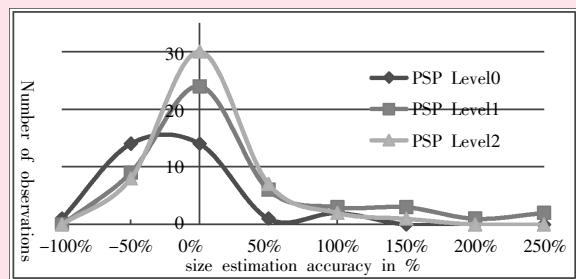


图2 PSP 各阶段程序大小估测精确度

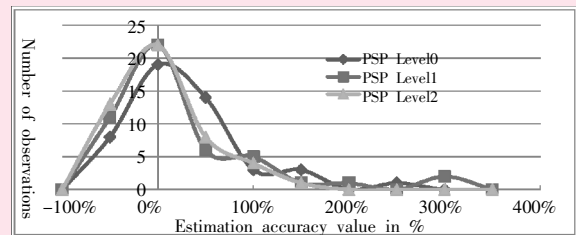


图3 PSP 各阶段时间花费估测精确度

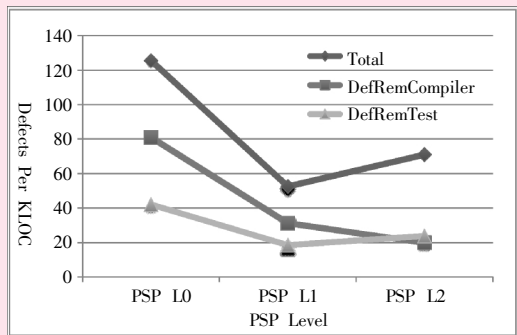


图4 PSP 各阶段时间花费估测精确度

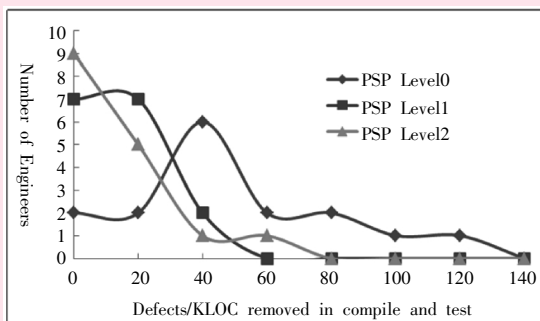


图5 PSP 各阶段平均编译与测试缺失数目

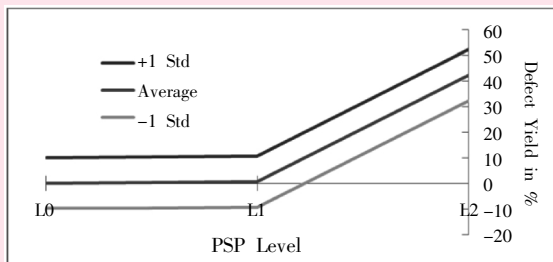


图6 PSP 各阶段平均缺失移除率

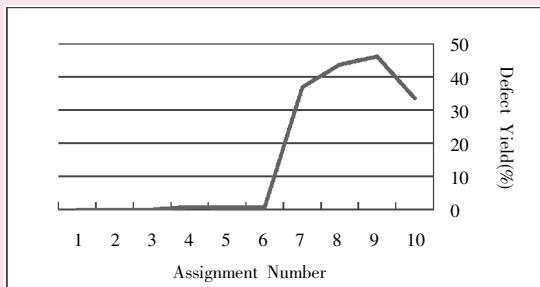


图7 每次作业的缺失移除率



27%缺失。

#### 4. 缺失移除效率

生产力是组织评估生产客户产品的重要参考因素，在PSP训练中藉由计算每小时产出的程序行数(LOC/Hr)数据收集来评估工程师个人的生产力，观察PSP各阶段并没有显著的差异，平均三个阶段的产能接近24.4 LOC/Hr，观察得知从Level 0到Level 1平均每小时增加4行程序代码，从Level 0到Level 2平均每小时增加3行程序代码。

### 四、检讨与建议

通过本次研究与软件工程多年的教学，我们相信所谓软件开发的纪律是需要经由教育训练的，也唯有如此才能满足“网络”时代的各种挑战<sup>[16]</sup>。我们也透过企业软件工程师再回学校训练的趋势获得证明，所谓软件开发纪律的内化，若没有良好的训练工具，是难以透过“银弹”来达成的<sup>[13][24]</sup>。

首先PSP经由PROBE回归估测方法<sup>[21]</sup>协助学习者有效预估时间、程序大小与缺失数目，然而从训练与备训练者的角度来看，此回归预估方法可能随着参考回归的次数增加而失真，且即便是研究生对此方法的学习与理解确实造成阻碍，导致训练者因花费大多的额外时间去协助学习者数据的搜集与数据的验证，从问卷的结果了解这样的预估方式未来将不愿意采用，另一方面因为浪费许多的时间在纪录时间、程序大小与缺失数的数据，似乎如同LEAP与HackyStat的软件工具有必要协助用户的数据搜集<sup>[21]</sup>。另外经由问卷结果了解工程师特别认为程序代码规范标准(code standard)与检讨报告(post-mortem)是有价值的技术，因为他能提供工程师编程的弱点信息，协助他们改弱点(如type of error)。

#### (一) 课程安排建议

PSP训练课程建议安排对象为研究所程度学员或企业工程师，原因是编程的经验影响PSP学习的成效，同时PSP训练所指定的作业内应随训练的对象有所调整，以符合真实的软件项目内容。除专属的PSP课程外，亦可将PSP相关技术导入其他信息专业课程中，相关文献也证明可获得良好的效果。

#### (二) 企业应用建议

当应用PSP流程改善教育于企业中，有一些议题需要考虑。

1. 全期PSP训练时间与相关统计作业内容，在现行企业的训练中式不可行的。所以需要调整(tailor)来符合业界所需。

2. 重新设计作业内容以符合企业项目的挑战所需。

3. 企业在有限的计划排程完成程序的撰写是重要的需求项目，所以在训练时应强调应用外部链接库(external library)，而非现形PSP所提供的数学问题。

4. 调整符合软件公司的训练期程，无论时间长短，所搜集或推演的参数建议涵盖PSP各阶段(Level 0、1、2)与本研究所提出的9项参数指标。

5. PSP另一项缺点，并没有提供接收测试阶段的验证工作，在企业中产品的验证是重要的，在PSP的相关研究中所谓接收测试是经由自己本身完成，但实际上的接收测试应透过客户审查验收，因此企业导入PSP训练时，需聘请外部客户(customer proxy)协助产品缺失审查，检验是否满足当初客户需求。

### 五、结论

观察台湾软件产业在流程改善过程中随着成熟度等级的提升，虽然对质量的提升有显著的影响，但似乎必须增加人员的工作量以提高软件的正确性，以及投入较多的时间(成本)于软件流程改善中以做为质量提升的代价，这也是导致成本构面及时程构面在越高成熟度等级效益反而下降的趋势。故时程面之“达成率”及“准时结案率”都是较低成熟度等级的效益显著优于较高成熟度等级，而质量面中“软件交付后之瑕疵率”则是较高成熟度等级的效益显著优于较低成熟度等级。企业强调成本、时程、质量、客户满意度、员工满意度等五个构面而PSP及涵盖了前三者的训练，有效运用PSP工具训练软件工程师具有成熟的软件开发能力，是目前企业的当务之急。

本研究所提出PSP企业改善意见，期望用于实际的企业训练上，进行个案学习(case study)，以了解所提的参数指针对协助企业软件开发改善的成效，并希望能藉由更多的实务经验，协助改进分析方法，以期能对企业软件流程改善有所助益。

#### 参考文献：

- [1] 资策会. 资策会MIC研究报告[EB/OL]. [http://mic.iii.org.tw/aisp/reports/reportdetail\\_register.asp?docid=2012#](http://mic.iii.org.tw/aisp/reports/reportdetail_register.asp?docid=2012#), 2012-08-11.
- [2] W.S.Humphrey. Software process improvement—a personal view; how it started and where it is going [J]. IEEE Software, 2007, 12(3): 223-227.
- [3] W. Hayes, J. W. Over. The personal software process (PSP): an empirical study of the impact of PSP on individual engineers [Z]. Technical Report CMU/SEI-97-TR-001, Software Engineering Institute/Carnegie Mellon University, Pittsburgh, PA, 1997.
- [4] A. Wesslén. A replicated empirical study of the impact of the methods in the PSP on individual engineers [J]. Empirical Software Engineering, 2000, 5(2): 93-123.
- [5] T.B. Hilburn, M. Towhidnejad. Doing quality work: the role of software process definition in the computer science curriculum [A]. Proceedings of the 28th SIGCSE Symposium [C]. Computer Science Education, 1997.

- [6] P.Abrahamsson,K.Kautz.The personal software process: Experiences from Denmark [Z].Proceedings of 28th Euromicro Conference,2002.
- [7] T.Lee,D.Baik,H.P.In. Cost benefit analysis of personal software process training program [A].IEEE 8th International Conference on Computer and Information Technology Workshops [C].IEEE Computer Society, 2008:631-636.
- [8] M.C.Paulk.An empirical study of process discipline and software quality [D].PhD Dissertation,University of Pittsburgh,2005.
- [9] W.I.Bullers.Personal software process in the database course[Z].Proc.Sixth Australasian Computing Education Conference,2004;25-31.
- [10] M.Morisio.Applying the PSP in industry [J].IEEE Software ,2000,17(6):90-95.
- [11] D. Rombach,J.Münch,A.Ocampo,W.S. Humphrey, D. Burton.Teaching disciplined software development [J]. The Journal of Systems and Software,2008,81 (5): 747-763.
- [12] J.Kamatar,W.Hayes.An experience report on the personal software process [J]. IEEE Software,2000,17(6): 85-89.
- [13] C.Wohlin,A.Wesslén. Understanding software defect detection in the personal software process[Z].Proceedings of the 9th International Symposium. Software Reliability ,1998;49-58.
- [14] J. Iversen,O.Ngwenyama. Problems in measuring effectiveness in software process improvement;A longitudinal study of organizational change at Danske Data[J].International Journal of Information Management,2005,26 (1):30-43.
- [15] W. S. Humphrey,T. R. Snyder,R.R. Willis. Software process improvement at Hughes aircraft [J].IEEE Software 1991,8(4):11-23.
- [16] W.L. Honig. Teaching successful "Real-World" software engineering to the "Net" generation;process and quality win! [Z].21st Conference on Software Engineering Education and Training,2008.
- [17] W. S. Humphrey. The personal software processSM (PSPSM) [Z].Technical Report CMU/SEI-2000-TR-022,2000.
- [18] C. Wohlin.Are individual differences in software development performance possible to capture using a quantitative survey[J].Empirical Software Engineering,2004,9 (3):211-228.
- [19] X.Zhong,N.H.Madhavji,K.El Emam. Critical factors affecting personal software processes [J].IEEE Software, 2000,17(6):76-83.
- [20] W.Boehm.Software engineering economics[M].London: Prentice Hall,1981.
- [21] P.M.Johnson,A.M. Disney. A critical analysis of PSP data quality:results from a case study[J].Empirical Software Engineering,1999,4(4):317-349.
- [22] K.Venkatasubramanian,S.B.T.Roy,M.V.Dasari.Teaching and using PSP in a software engineering course:An experience report[Z].Software Engineering Education and Training Annual Conference,2001.
- [23] W.S. Humphrey.Introduction to the Personal Software ProcessSM[M].New Jersey:Addison-Wesley,1996.
- [24] F.P. Brooks. No silver bullet—essence and accidents of software engineering [J].IEEE Computer,1987,20(4): 10-19.
- [25] W.S. Humphrey. The personal software process;status and trends[J].IEEE Software,2000,17(6):71-75.

[责任编辑：张 磊]

## Applying Personal Software Process Tools to Software Development

SHEN Wen-xiang

(Overseas Chinese University, Taichung 40721, China)

**Abstract:** Software processes have been recognized as an important subject of research since the early days of software engineering. In order to improve software process, it is important to apply Capability Maturity Model Integration (CMMI), a framework for improving an organization's process. One of its important elements is quantitative management where process metrics is embraced in the definition of the process. Aside from the critical role of CMMI played in UML-based profile, Personal Software Process (PSP) can be considered as its equivalent on an individual basis. However, a systematic mechanism to assess and interpret the effect of PSP is invisible in the existing literature. Therefore, an assessment model is also proposed to test against the PSP training effects. In this paper, a set of analyses are conducted to explore the relationships among the PSP metrics and to interpret possible application results.

**Key words:** Capability Maturity Model Integration; personal software process; software process improvement