

PSP 的软件测量研究^{*})

侯 红 郝克刚

(西北大学计算机科学系 西安 710069)

摘 要 本文从个体软件过程 PSP 的体系结构和工作改进过程出发,对 PSP 中的软件测量进行了研究,指出了其关键的测量元,最后,为帮助实施 PSP,给出了对实施效果进行评估的评价标准。

关键词 软件测量,个体软件过程,软件过程改进

Software Measurement within the PSP

HOU Hong HAO Ke-Gang

(Department of Computer Science, Northwest University, Xi'an 710069)

Abstract From the architecture and process improvement of Personal Software Process(PSP), software measurement in PSP are researched in this paper. Then the critical measures of PSP are specified. Moreover in order to implement PSP, the evaluation criteria is proposed.

Keywords Software measurement, Personal Software Process(PSP), Software process improvement

1 引言

1995 年, Watts Humphrey 提出了 PSP(Personal software process)个体软件过程的概念。PSP 是 CMM(capability maturity model)能力成熟度模型在个体软件过程水平上的一个应用,它与 TSP(Team software process)小组软件过程和 CMM 一起,为软件产业提供了一个集成化的、三维的软件过程改进框架。

PSP 中设立了 7 个等级^[1]: PSP0、PSP0.1、PSP1、PSP1.1、PSP2、PSP2.1、PSP3,个体过程能力可按照 PSP 的等级不断改进。PSP 能力提高的基础是通过在 PSP 级别中引入新的软件测量来实现的。因此,对 PSP 软件测量的研究,有助于分析个体软件过程改进的关键因素,更加有效地发挥 PSP 在软件开发过程中的作用,提高软件开发质量,缩短软件开发周期。

2 PSP 的体系结构及工作改进过程

PSP 具有明确定义的体系结构,由 8 个向上兼容的个体过程组成的,其进化序列^[1]如图 1 所示。PSP 软件工程规范为软件工程师提供了发展个人技能的结构化框架和必须掌握的方法。按照 PSP 规程,改进软件过程的步骤大致如图 2 所示。

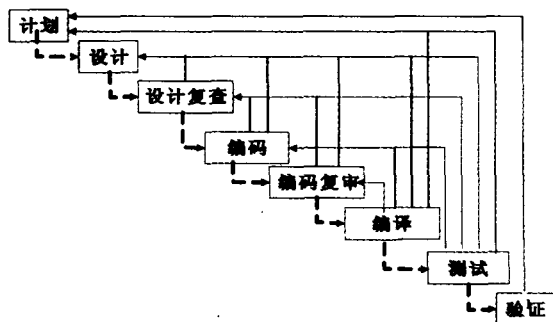


图 1 PSP 模型

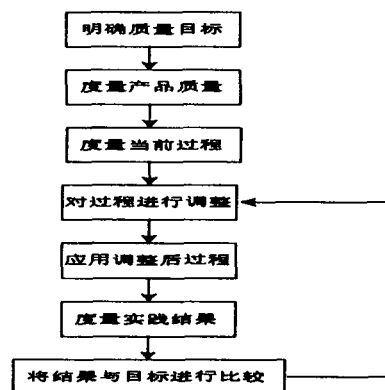


图 2 PSPI 作改进过程

3 过程测量的目的

复杂的过程存在很多元素,因此可以有許多潜在的方式进行改进。然而离开对过程步骤的定量理解,无法知道过程中的哪些部分是有效的,过程测量为客观地理解过程是如何工作的以及哪些地方可以改进提供客观的数据。过程测量的目的是:获得定量的理解;对产品、过程、组织进行评价;对产品或过程进行控制;制定估算或计划。

4 PSP 的软件测量

PSP 采用 GQM(Goal-Question-Metrics)方法对个体软件过程进行度量^[1,2]。在 PSP 中收集数据的目标是:①理解个体软件开发过程是如何工作的;②确定改进产品质量的步骤;③确定过程变更对个体生产率的影响;④建立过程改进的基准;⑤协助制定更为准确的计划。

为了达到上述目标,在个体软件过程中需要考虑下述问题:①个体性能哪些方面是重要的;②如何对这些方面进行测

^{*})本文受西安市科技攻关基金项目(编号:GG200220)资助。侯 红 高工,博士生,主要研究方向为软件过程工程、软件度量。郝克刚 教授,博士生导师,主要研究方向为软件工程与理论。

量;③作为个体软件工程师所能达到的最佳性能是多少;④从这些进展中能够学到什么;⑤还有其他方法对过程改进有帮助吗。

PSP 探索上述问题并定义与其相关的测量。

4.1 PSP 中的基测量

在 PSP 中有三个基测量:开发时间、缺陷和规模。所有其他的测量都是由这三个基测量中派生出来^[4]。

• 开发时间 其测量单位为分钟。为工程师在 PSP 各阶段所花费的时间减去中断时间如打电话、喝咖啡等,通过时间记录日志来记录开发时间。

表 1 时间记录日志

日期	开始	结束	中断时间	实际花费	阶段	注释
5/13	7:58	8:45	3	44	计划	电话
	8:47	10:29	2	100	设计	设计和设计评审
	7:49	8:59		70	编码	主函数和所有功能
	9:15	9:46		31	编译	编译和链接
	9:47	10:10		23	测试	运行测试 A,B,C
	4:33	4:51		18	后置处理	

• 缺陷 定义为设计和编码所做的任何变更以确保程序正确地编译和测试。通过缺陷记录日志来记录。

表 2 缺陷记录日志

日期	编号	类型	引入阶段	排除阶段	修改时间	修复缺陷
	描述:					
	描述:					
	描述:					
	描述:					

• 规模 在 PSP 中进行软件规模测量的主要目的是为估算开发时间提供基础。由于代码行可自动地统计、清晰地定义而且和开发的工作量紧密相关,因此在 PSP 中采用代码行数进行规模测量。

4.2 PSP 中的派生测量

在 7 个 PSP 级别中的每一个级别中都会引入新的测量用以帮助工程师管理和改进他们的工作性能。所有这些测量都是从开发时间、缺陷和规模的三个基测量中派生出来的。

• 中断时间:项目进展过程中被小的中断耽误的时间,如电话。

• 时间差:从开始到结束的时间减去中断时间。

• 各阶段计划时间:项目在某阶段估算的花费时间。

• 各阶段实际时间:项目某一阶段时间差的和。

• 总时间:项目所有阶段计划和实际时间的和。

• 到目前为止各阶段时间:各阶段所完成工作的实际时间和。

• 到目前为止的总时间:所有阶段完成工作的实际时间和。

• 到目前为止各阶段时间的百分比: $100 \times \text{到目前为止各阶段时间} / \text{各阶段的总时间}$

• 编译时间:从第一次编译到编译完成的总时间。

• 测试时间:从第一次测试到测试完成的总时间。

• 修正时间:发现和修改缺陷所用的时间。

• 每小时生产的代码行数(Loc/Hour):新增和变更的代码行数/总的开发时间。

估算的准确度:估算与实际结果匹配的程度。计算其花费和规模。 $\text{准确度} = 100 \times (\text{实际值} - \text{估算值}) / \text{估算值}$

• 测试缺陷密度:在测试阶段,每新增和变更的代码行所排除的缺陷。 $1000 \times (\text{测试阶段排除的缺陷}) / \text{实际的新增和变更的代码行数}$

• 编译缺陷密度:在编译阶段,每新增和变更的代码行所排除的缺陷。 $1000 \times (\text{编译阶段所排除的缺陷}) / \text{实际的新增和变更的代码行数}$ 。

• 总缺陷密度:缺陷密度= $1000 \times \text{实际的缺陷总数} / \text{实际新开发的和修改的代码行数}$

• 过程效益:过程效益= $100 \times (\text{编译前排除的缺陷数}) / \text{编译前引入的缺陷数}$,过程效益的目标为 100%,如果能够做到这一点,在代码复查阶段将发现所有的缺陷,而不会将缺陷留到编译和测试阶段。

• 质检时间:设计评审和代码评审所花费的时间。

• 过失时间:编译和测试阶段所花费的时间。

• 质量成本:质量成本=质检时间+过失时间

• 质检/过失比(A/FR)即质检成本与过失成本之比。在 PSP 中用复查时间除以编译和测试时间。应努力使 A/FR 的值大于 2。A/FR 用来测量在第一次编译前花在查找缺陷上的时间的相对值。

• 评审率:每小时所评审的代码行数。

以上派生测量中,过程效益和质检/过失比为影响 PSP 的关键因素^[5]。

4.3 PSP 的 7 个级别中的测量元

PSP 的 7 个过程级别如图 3 所示。每一个级别都是在软件测量的基础上增加一些过程步骤而设计的。个体能力的级别愈高,个体的过程愈成熟,其测量就愈丰富。换句话说,个体的过程愈成熟,个体过程的测量就愈成熟。

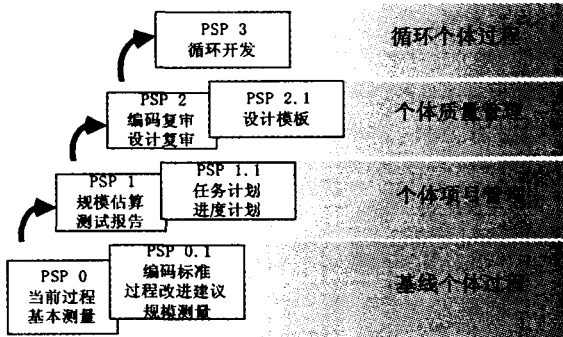


图 3 PSP 过程级别

• 基线个体过程——PSP0 和 PSP0.1。它建立开发时间、缺陷和规模的历史数据的基线。

• 个体项目管理——PSP1 和 PSP1.1。它关注个体项目管理的技术,引入规模和工程量估算、进度计划和进度跟踪方法。PSP 采用挣值的方法进行进度计划和跟踪。

• 个体质量管理——PSP2 和 PSP2.1。增加了质量管理

的方法:个体设计和编码复审、设计表示、设计模版、设计验证技术以及管理过程和产品质量的测量。

在 PSP 中质量管理的目标是发现和排除在第一次编译前的缺陷。与其相关的测量为过程效益。因此为了提高过程效益在 PSP2 中增加了设计评审和代码评审两个新步骤。

从 PSP2 开始,工程师同时使用历史数据制定质量及开发过程中质量控制的计划。目标是在第一次编译前排除引入的所有缺陷。

在 PSP 质量管理中,工程师跟踪自己产生的缺陷,得出缺陷消除量,并计算质量测量的成本。通过 Pareto 守则分析缺陷,导出个人设计和代码评审检查单,工程师根据每个新项目的缺陷数据,更新检查单的内容。

• 循环个体过程——PSP3。强调的是在不牺牲质量或生产率的前提下,在 PSP2 的基础上循环,有效地将 PSP 扩展至更大的项目。

5 PSP 实施的评价标准

PSP 实施过程中,个体软件过程能力的提高可通过对规模和工作量估算的准确度、产品质量、过程质量和个体生产率等 5 个因素的测量来评估^[4]。

• 规模估算的准确度 ASE。ASE=(估算规模-实际规模)/估算规模,如图 4 所示,随着个体能力的提高,规模估算的准确度 ASE 应逐步提高,也就是说,规模的估算值与最终程序的实际规模值越来越接近。随着在 PSP1 中正式引入规模估算技术,工程师规模估算的准确度应显著提高。

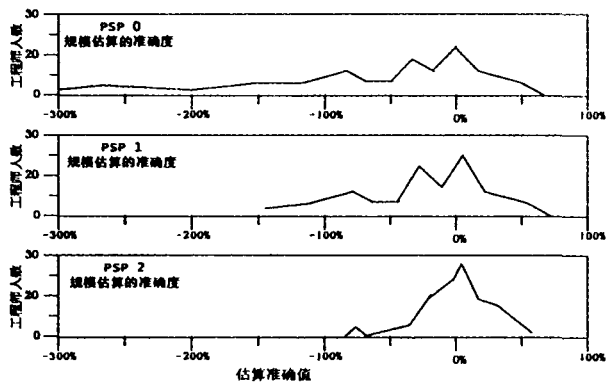


图 4 PSP 级别规模估算准确度分布

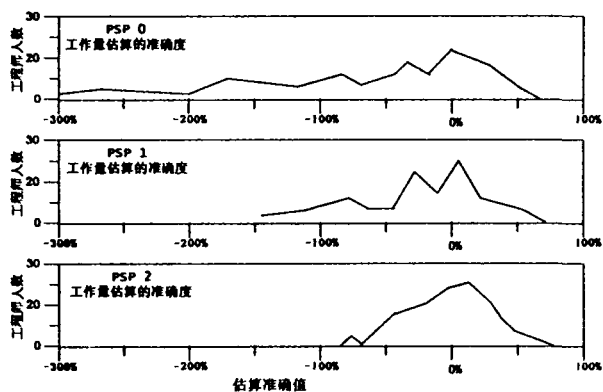


图 5 PSP 级别工作量估算准确度分布

• 工作量估算的准确度 AEE。AEE=(估算的工作量-实际的工作量)/估算的工作量,如图 5 所示,随着个体能力的提高,工作量估算的准确度 AEE 也应逐步提高,也就是

说,工作量的估算值与整个项目生命周期中实际花费的工作量越来越接近。随着在 PSP1 中引入统计质量控制技术,工程师工作量估算的准确度应显著提高。

• 缺陷密度 DD。DD=缺陷数/代码行数,如图 6 所示,随着个体能力的提高,千行代码引入的缺陷数降低,因此千行代码排除的缺陷数亦降低。随着在 PSP2 中引入设计和编码复审技术,进入编译和测试阶段的缺陷密度应显著降低。

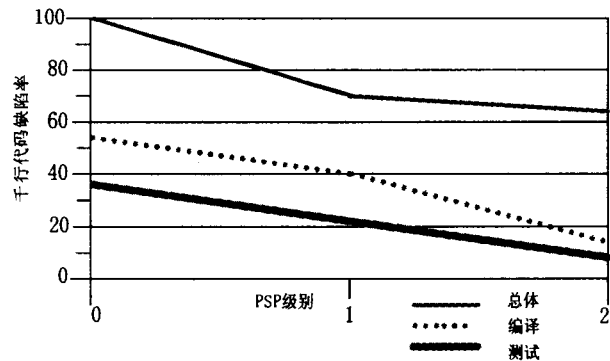


图 6 缺陷密度平均趋势图

• 编译前的缺陷效益。随着个体能力的提高,缺陷效益应提高。特别是在 PSP2 中引入设计和编码复审技术后,工程师的缺陷效益应显著提高。

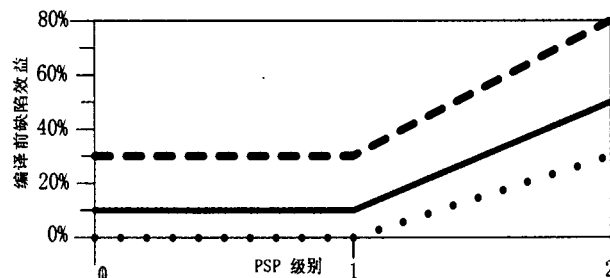


图 7 平均效益

• 生产率。在 PSP 中通过收集工程师每小时所生产的代码行数来进行测量。随着个体能力的提高,工程师的生产率应得到提高,也就是说工程师首次和末次在设计、编写和测试阶段每小时所生产的代码行数应提高。

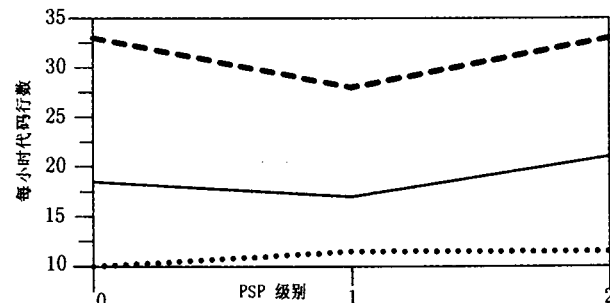


图 8 平均生产率

小结 迄今为止,学术界和产业界公认 CMM 是当前较好的软件过程,截至 2004 年 3 月中国软件企业通过 CMM 认证的已达 152 家,居世界第三位^[7],然而它的成功与否与软件开发单位内部有关人员的积极参与和创造性活动密不可分。而且由于 CMM 中并未提供有关实现子过程域所需要的具体知识和技能,因此进行个体软件过程 PSP 的研究与实践以填补这一空白,且为基于个体和小组软件过程的优化提供

了具体、有效的途径。

但是在 PSP 的测量中未引入软件测量刻度,因此无法对测量的意义本身进行评价^[6]。通过 PSP 的实践表明 PSP 使个体工程师的关键性能得到了提高包括工程师的估算和计划能力,其开发的软件的质量,工作过程的质量,但是到目前为止没有足够的证据能够说明个体的生产率得到了提高。

参考文献

- 1 Humphrey W S. A Discipline for Software Engineering. Reading, Ma.: Addison-Wesley, 1995
- 2 Fenton N E. Software Metrics: A Rigorous & Practical Ap-

proach. 清华大学出版社, 2003. 97~98

- 3 paulk M C. Applying SPC to the Personal Software Process, www.sei.cmu.edu, 2003
- 4 Hayes W, Over J W. The Personal Software Process: An Empirical Study of the Impact of PSP on Individual Engineers; [Technical Report CMU/SEI-97-TR-001]
- 5 郑丽娟, 王保义, 宋雨. 影响 PSP 的关键因素分析. 华北电力大学学报, 2003, 30(1)
- 6 侯红, 郝克刚, 郭小群. 软件测量刻度及选择方法. 计算机科学, 2005, 32(5)
- 7 www.iturils.com

(上接第 88 页)

加密的数据包进行分析, 经过几个小时的时间就可以破译出信息的全部内容。

(2) WEP 在链路层采用 RC4 加密技术来防止非授权用户的监听以及非法用户的访问。可是, 静态 WEP 密钥对于 WLAN 上的所有用户都是通用的。这意味着如果某个无线设备丢失或者被盗, 所有其他设备上的静态 WEP 密钥都必须进行修改, 以保持相同等级的安全性。这将给企业网络带来费时费力的、不切实际的管理任务。

(3) 在 RC4 中, 存在弱密钥。所谓弱密钥, 就是密钥与输出之间存在超出一个好密码所应具有的相关性。在 24 位的 IV 值中, 有 9000 多个弱密钥。攻击者收集到足够的使用弱密钥的包后, 就可以对它们进行分析, 只须尝试很少的密钥就可以接入到网络中。

4 AES 代替 RC4 的解决方案

4.1 AES 设计原理

AES 为分组密码体制, 分组长为 128 比特, 密钥长可以为 128、192 和 256 比特三种。AES 加密算法的数据处理单元是字节, 128 比特的分组信息被分成 16 个字节。AES 算法中引入矩阵的概念, 分组的 16 个字节按顺序被复制到一个 4×4 的矩阵中, 称为 state(状态), AES 的所有变换都是基于状态的变换。AES 变换是由轮函数通过多轮迭代实现的, 根据密钥长度的不同, 轮函数的迭代次数也不一样, 对应上面的三种密钥长度, 迭代次数分别为 10、12、14 轮。轮函数的构成包括非线性, 扩散和密钥调度几种元素。非线性变换的目标就是通过较小较简单的非线性元素得到大的复杂化的非线性构件。在轮函数的每一轮迭代中, 包括四步变换, 分别是字节代换运算(ByteSub())、行变换(ShiftRows())、列混合(MixColumns())以及轮密钥的添加变换(AddRoundKey()), 其作用就是通过重复简单的非线性变换、混合函数变换, 将字节代换运算产生的非线性扩散, 达到充分的混合, 使加密后的分组信息统计特性分布更均匀, 在每轮迭代中引入不同的密钥, 这样便以最简单的运算代价得到最好的加密效果, 实现加密的有效性。

4.2 AES 加密算法性能分析

(1) AES 有较长的密钥, 密钥长度可为 128bits、192bits

和 256bits 三种情况, 明显提高了加密的安全性, 同时, 对不同机密级别的信息, 可采用不同长度的密钥, 执行灵活性较高。

(2) 在 AES 中, 由于密钥扩展函数的特点, 所产生的轮密钥随机性很强, 对初始密钥的选取也没有特别的限制, 可抗击穷举密钥的攻击。

(3) AES 的均衡对称结构既可以提高执行的灵活度, 又可防止差分分析方法的攻击。

(4) AES 算法的实现更简单、有很强的扩散性能, 能把非线性部件产生的非线性很快地扩散开, 形成的密码有很高的随机性。可有效地防止差分分析和线性分析的攻击。

(5) 因为一个一般的积分均衡子集经过一轮迭代后, 就可以使加密结果面目全非, 实际上 Rijndael 经 4 轮迭代后, 便可抗击积分密码分析了。

最后, AES 算法在所有的平台上都表现良好, 其操作比较容易抵御对物理层实现的某些攻击, 能很好地适应无线网络的发展。因此, 无论是从安全还是实现的难易度上考虑, 采用 AES 替代 RC4 算法是一种最明智的选择。

结束语 任何事物都有一个出现、繁荣和灭亡的过程。AES 代替 RC4 的解决方案, 从以上的分析中来看, 它显示出其无可比拟的优越性, 但随着科学技术的发展, 任何一项技术都有被淘汰的可能, 因此, 我们只有不断地跟踪和研究最新技术, 不断发现和解决无线网络所存在的问题, 才能使我国信息安全技术迈上一个新的台阶, 使无线网络更快地得到普及和发展!

参考文献

- 1 Daemen J, Rijmen V. AES Proposal; Rijndael. AES Algorithm Submission. Sept. 1999
- 2 Gladman B. A Specification for Rijndael, the AES Algorithm, Oct. 2000. Available at: <http://fp.gladman.plus.com/cryptography-technology/>
- 3 吴世忠, 祝世雄, 等. 应用密码学. 机械工业出版社, 2000
- 4 卢开澄. 计算机密码学. 清华大学出版社, 1998
- 5 Nechvatal J, et al. Report on the Development of the Advanced Encryption Standard (AES). National Institute of Standards and Technology, Oct. 2000
- 6 林柏钢. 网络与信息安全教程. 机械工业出版社, 2004