

PSP 实施的数据库逻辑模型设计方案初探

杨丽娜 (四川大学人口研究所 610064)



摘要: 将PSP实施于个体工作方式的控制、管理和改进时,我们会发现借助一定的工具会使这一过程更加易行,本文设计的PSP管理数据库,试图为个体软件过程的改进提供所需的工具。

关键词: PSP 数据库 表

1 PSP 概述

PSP (Personal Software Process)——个体软件过程,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,它包括了软件开发表格、指南和规程的结构化框架 [1]。PSP 的实施能有效地帮助软件工程师持续地使用正确的工程方法,尽早排除错误,更准确地做出开发规模和时间上的估计,提高软件循环时间,增加软件产品的生产率。

实施PSP能指导个体如何计划工作,如何对工作进行追踪记录,通过对工作结果的分析改进下一计划中的个体行为过程,它包含了计划、追踪记录和事后检查这一不断循环的过程(如图1所示)。

计划包括对软件规模、时间、缺陷三方面的估计,通过收集以前相似工作所用的规模、时间、缺陷数据,运用基于估计的计划方法(PROBE),预测待开发软件所需的规模、时

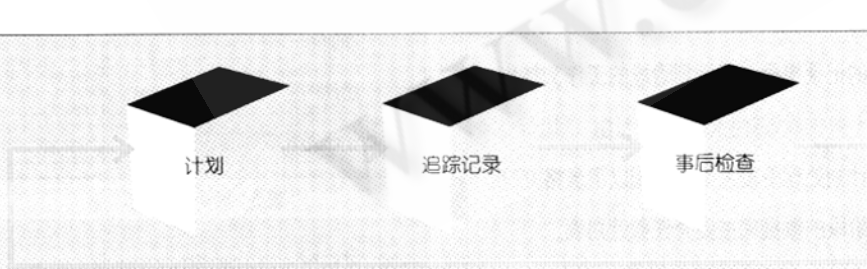


图1 PSP 过程流程的核心框架



图2 PSP 管理数据库结构图

程序员ID		
姓名		
性别		
年龄		
从业时间		
进入企业时间		
担任角色		

表1 程序员个人信息表

项目代码		
项目名称		
项目内容		
起始时间		
结束时间		
负责人		

表2 项目表

任务代码		
项目代码		
任务名称		
任务描述		
任务负责人		
备注		

表3 任务表

子任务代码		
子任务名称		
子任务描述		
基本		
增加		
更改		
删除		
重用		
时间估计		
缺陷估计		

表4 任务计划表

间,可能产生的缺陷,这三项数据有助于跟踪工作的进展状况。

在追踪记录阶段,进行时间度量、规模度量、缺陷度量,通过收集每一过程的数据来控制、管理自己的工作过程,同时也为下一项目的计划工作积累了估计所需的历史数据。

开发工作结束后,必须进行事后检查,把追踪记录下来的规模、时间、缺陷的实际数据与计划时做出的估计数据进行比较和分析,通过事后总结,检查实际与计划的差距,追踪软件过程的改善情况,更新软件一切中积累的历史数据。

PSP实现过程贯穿了对历史数据的收集与总结,合理设计PSP的管理数据库是PSP实施的重要环节。

2 PSP管理数据库概述

在个体软件过程的实施中需要运用一系列的表格,PSP管理数据库为工作过程的控制、管理和改进提供了有利的工具。根据PSP过程流程

(参照图1),数据库的设计为计划阶段、追踪阶段、总结阶段三个逻辑处理单元(见图2)。

3 逻辑处理单元

下面对各逻辑处理单元分别进行描述。

3.1 计划处理单元

3.1.1 概述

该处理单元包括规模估计表、开发时间估计表、缺陷估计表。

规模估计包含对文档的估计(以页为单位)和程序的估计(以代码行LOC为单位)。对程序规模的估计采取把计划要做的工作与以前做过的相似工作相比较,根据至少前三个程序的历史数据对基本的、增加的、更改的、删除的及重用的代码行数做出估计。

对开发时间的估计,是个体根据自己的开发效率(分钟/LOC)、所负责任务的规模及其难度系数做出的。

缺陷包括任何影响到程序完整而有效地满足用户要求的東西[2],需求定义、设计、编码中都可能发生,个人对其所负责任务部分的缺陷最熟悉,因此在计划阶段,应对该部分可能发生的缺陷数做出估计。

3.1.2 目标

为个体提供规模估计、开发时间估计、缺陷估计的记录模板,使计划阶段的工作文档化。

3.1.3 数据库设计

计划处理阶段主要应填制以下表格,应用于这一阶段的数据库主要有任务计划表。

任务计划表填制的目的是对个体所负责任务的程序规划、所花时间、可能产生的缺陷做出估计,它开始的前提条件是已正确、全面地完成需求分析、概要设计,并且个体已具备计划的能力。

对任务计划表各字段值描述如下:

子任务代码:组织将任务分配给个人后,个人根据具体情况将任务细分为若干子任务;

子任务描述:对子任务所属模块或所处阶段、所具有的功能的描述;

基本、增加、更改、重用:在循环开发中,代码行可分为基本的、增加的、更改的、删除的、重用的;

时间估计:输入个体对负责子任务所花时间的估计;

缺陷估计数:输入个体对所负责任务中可能产生的缺陷数的估计。

上述各表中的项目、任务、子任务之间的关系如下:



登记ID		
日期		
子任务代码		
开始时间		
结束时间		
中断时间		
净时间		
活动内容		
子任务完成标志		

表5 时间记录日志

登记ID		
日期		
子任务		
代码		
基本		
增加		
更改		
删除		
重用		

表6 规模记录日志

缺陷代码	
日期	
缺陷类型	
引入阶段	
排除阶段	
更改日期	
更改方案	

表7 缺陷记录日志

子任务代码	
计划完成时间	
实际完成时间	
计划规模	
实际规模	
计划价值	
完成价值	
开发效率	

表8 进度视图

软件开发组织由若干项目组组成，每个项目组负责一个项目，项目组内进行分工，分配给每人若干任务，个人又将其任务划分为若干子任务来完成。任务的分配可以按模块划分，每人负责一至几个模块，个人的子任务则可按模块完成的阶段分为计划、设计、编码、测试等；任务也可按软件开发的阶段进行分配，一部分人负责设计，一部分人负责编码，而测试由另外的人员负责，个人的子任务划分则按个人的任务性质不同而异。

3.2 追踪处理单元

3.2.1 概述

该处理单元包括时间记录日志、规模记录日志、缺陷记录日志。

时间记录日志为个体提供了完整而准确的时间花费记录，通过对每项任务开始时间、中断时间、结束时间的计算和总结，个体能够追踪其花费在具体开发工具上的有效时间，这既有利于个体对时间进行管理，也为下一项目的计划提供了数据依据。

应用规模记录日志，个体对产品规模进行度量和记录，结合时间记录日志中的相关数据可计算个体的开发效率，将个体任务加总可得到整个项目的总工作量。产品规模包括了程序的规模、文档的大小以及其他使用辅助工具生成的程序要素，度量方法有差异。文档大小以页为单位进行度量，利用辅助工具生成的程序要素只统计自己编写的代码行，程序规模使用代码行度量（不统计空行和注释行）。

缺陷记录日志记录了个体在任务过程中发现的每一个缺陷，及其引入阶段、排除阶段、更改方案。有效利用这些缺陷数据可以在将来的开发中减少或避免引入缺陷，高效地解决同类缺陷。

3.2.2 目标

为个体提供时间记录日志、规模记录日志、缺陷记录日志的模板，使开发过程中的工作文档化。

3.2.3 数据库设计

追踪处理阶段主要应填制以下表格，应用于这一阶段的数据库主要有时间记录表、规模记录表、缺陷记录表。

时间记录日志填制的目的是完整而准确地记录个体开发过程中时间上的花费，它开始的前提条件有统一的时间记录日志模板，并且软件个体能够接受时间记录的工作。

对时间记录日志各字段值描述如下：

登记ID：由计算机自动生成；

子任务代码：同任务计划表4；

开始时间：开始某个活动的时间；

结束时间：结束上述活动的时间；

中断时间：执行上述活动时被打断的时间；

净时间：某个活动从开始到结束的时间长度，但要扣除中断时间；

活动内容：对进行的活动进行描述；

子任务完成标志：标注是否已完成某项子任

务，完成时间可注上Y；

规模记录日志填制的目的是完整而准确地记录开发过程中完成的程序规模以及文档大小，它开始的前提条件是有统一的规模记录日志模板，并且软件个体能够接受规模记录的工作。对规模记录日志各字段值描述如下：

日期：程序生成的日期；

登记ID：由计算机自动生成；

子任务代码：同任务计划表4；

基本、增加、更改、删除、重用：在循环开发中，代码行可分为这五种类型，记录下开发中产生代码所属类型；

缺陷记录日志填制的目的是记录开发中所发现的每一个缺陷的详细数据及更改方案，它开始的前提条件是有统一的缺陷记录日志模板，并且软件个体能够接受缺陷记录的工作和具备一定的缺陷查找技术。

对缺陷记录日志各字段值描述如下：

缺陷代码：每个缺陷都有一个计算机自动生成的代码；

日期：缺陷发现的日期；

缺陷类型：按照自己的判断选择缺陷所属的类型(缺陷类型可选择不同标准)；

引入阶段：填入引入缺陷的阶段；

排除阶段：填入改正缺陷的阶段；

更改日期：填入改正缺陷的日期；

子任务代码	评价项目	计划	实际
	程序规模		
	基本的		
	删除的		
	更改的		
	增加的		
	重用的		
	开发时间		
	产生的缺陷		

表9 项目总结视图

更改方案：填入缺陷改正的解决方案；

3.3 总结处理单元

3.3.1 概述

该处理单元主要由进度视图、项目计划总结视图组成。进度视图和项目总结视图均由计划阶段的任务计划表和追踪阶段的规模记录日志、时间记录日志、缺陷记录日志生成。通过计划数据与实际数据的比较和分析，一方面可以追踪计划的完成情况，一方面为今后的计划制订提供准确的数据依据。

3.3.2 目标

为个体提供进度追踪模板、项目总结模板，使开发的总结工作文档化。

3.3.3 数据库设计

总结处理单元由进度视图、项目总结视图构成，视图的数据均来自计划处理单元和追踪处理的数据表。

建立进度视图的目的是将计划数据与实际数据进行比较从而追踪工作进度，它开始的前提条件是有完整的计划数据和实际数据。对进度视图各字段值描述如下：

子任务代码：同任务计划表 4。

计划完成时间：完成某项子任务所计划的时间；可以从任务计划表的时间估计字段得出；

实际完成时间：完成某项子任务的实际时间，可以从时间记录日志中得出；

计划规模：完成某项子任务的计划规模，可以从任务计划表中根据公式（基本 + 增加 + 更改 - 删除 - 重用）得出。

实际规模：完成某项子任务的实际规模，可以从规模记录日志中以同样的公式得出。

计划价值：即计划每项子任务在总任务所占的百分比，其计算方法是：将个人完成某项子任务计划花费的时间除以完成总任务计划花费的时间。

完成价值：即已完成子任务的所有计划价值

之和（一个任务的价值只有在这个任务被完全完成后才能得到）；

开发效率：将实际规模与实际完成时间相比，单位是 LOC/小时；

建立项目总结视图的目的是通过将实际数据与计划数据进行比较，总结经验，提高估计能力，使软件个体能有效地控制、管理并改善自己的工作方式，它开始的前提条件是项目的开发工作已经结束，并有完整的计划数据与实际数据的记录；

子任务代码：同任务计划表 4。

评价项目：包括程序规模、开发时间、产生的缺陷；

计划：从计划处理单元的任务计划表中提取所需评价项目的计划数据；

实际：从追踪处理单元的规模记录日志、时间记录日志、缺陷记录日志中提取所需评价项目的实际数据。 ▀

参考文献

- 1 Watts S.Humphrey.A Discipline for Software Engineering [M].Addison-Wesley,Reading Mass,1995.
- 2 何新贵等，软件能力成熟度模型[M]，清华大学出版社，2000年11月第1版。
- 3 Watts S.Humphrey.The Personal Software Process[R].CMU/WEI-2000-TR-022.
- 4 Karl E.Wiegers.Personal Process Improvement [c].Software Development magazine,May 2000.
- 5 Watts S.Humphrey.The Personal Process in Software Engineering[R].1994.

