CHAPTER 1

The PSP is a self-improvement process that helps you to control, manage, and im­ prove the way you work. It is a structured framework of forms, guidelines, and procedures for developing software. Properly used, the PSP provides the data you need to make and meet commitments, and it makes the routine elements of your job more predictable and efficient.

The PSP’s sole purpose is to help you improve your software engineering skills

A defined process specifies precisely how to do something. If we have not done such work before and we don’t have a defined process readily available, we will not know the proper way to proceed.

We all know what they mean in general terms, but to actually use one of these processes we need to know precisely what to do and the order in which to do it. To do this, we must have what is called an operational process. defined, measured, planned, and quality-controlled personal process.

SUMMARY CH1: A personal process is something that you use to guide your work. With the PSP, you will make accurate plans, consistently meet commitments, and deliver high-quality products.

CHAPTER 2

The software process establishes the technical and management framework for applying methods, tools, and people to the software task. The process defini­ tion identifies roles, specifies tasks, establishes measures, and provides exit and entry criteria for the major steps. When properly used, a defined process also helps to ensure that every work item is properly assigned and that its status is consis­ tently tracked.

In summary, a defined process identifies a job’s principal steps: A defined process also includes measures:  defined process provides a sound basis for project management: A defined process provides a solid foundation for process management and im­ provement:

The principal objective of PSPO, the baseline process, is to provide a framework for writing your first PSP program and for gathering data on your work. The PSPO process provides the following benefits:

* A convenient structure for performing small-scale tasks
* A framework for measuring these tasks
* A foundation for process improvement

Measurement Framework

A defined process establishes the process measures. This enables you to gather data on the time you spend on each task, the sizes of the products you produce, and the numbers of defects you inject and remove in each process step. A defined process also gives measurements explicit meaning. This precision, however, requires that each process phase have explicit entry and exit criteria.

Foundation for Improvement

If you don’t know what you are doing, it is hard to improve the way you do it. Be able to break down tasks into small activities

**Why Forms Are Helpful**

Suppose, instead, that you have a planning form. Now you don’t need to de­ cide what data to provide; the form tells you. All you do is fill in the blanks. The form may even provide guidance on how to do the required calculations. Then, when you are through, you should check to ensure you did the work properly. With a form, this is easy. You just ensure that all of the spaces contain data. Properly de­ signed forms improve your efficiency and help you produce complete and correct results.

**The PSP Process Elements**

The overall PSPO process is shown in Figure 2.1 (page 14). In the planning step, you produce a plan to do the work. Next are the four development steps: design, code, compile, and test. At the end, in the postmortem step, you compare your ac­ tual performance with the plan and produce a summary report. Although these planning and postmortem steps may not seem necessary when you are writing small programs, they are needed to gather the data to manage and improve your personal process.

Also, as you gain experience, you will find that the postmortem is an ideal time to think about your data and to see where and how to improve.

You need to finish the report right after you finish your Project, so your memory is fresh.

In the case of PSPO, the planning, development, and postmortem phases have process definitions, but de­ sign, code, compile, and test are named as unrefined steps. When a process ele­ ment has a definition and a structure, I call it a phase. When it has no defined structure, I call it a step or a task.

* A phase is something that can be broken down into smaller parts.

**The PSPO Process**

The PSP scripts guide you through the process steps.

**2.7 PSPO Measures**

PSPO has two measures:

1. The time spent per phase

The time spent per phase is a simple record of the clock time spent in each part of the PSP process.

2. The defects found per phase

For PSPO, record the specified data for every defect you find during compiling and testing. A defect is counted every time you change a program to fix a problem. The change could be one character, or it could be multiple statements. As long as the changes pertain to the same com­ pile or test problem, they constitute one defect. Note that you determine the defect count by what you change in the program. Some compilers, for example, generate multiple error messages for a single defect. If these were all connected to one problem, then you would count it as one defect.

**2.8 Time Recording**

use a stopwatch for interruptions

**2.9 Defect Recording**

Also, if you make a mistake fixing one defect and later find and fix that new defect, note the number of the de­ fective fix in the Fix Ref column.

**2.11 The Compile Phase**

For those environments that have a compile step, the compile data are ex­ tremely useful and I urge you to gather complete compile time and defect data.