



# **Personal Software Process<sup>SM</sup> for Engineers: Part I**

## **Tutorial: Using PSP0.1**

**This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.**

**Sponsored by the U.S. Department of Defense  
© 2006 by Carnegie Mellon University**



# Tutorial Objectives

---

After this tutorial, you will

- understand the PSP0.1 process
- know how to use the PSP0.1 process scripts and forms
- be prepared to use PSP0.1 for program 2



# **PSP0.1 Objectives**

---

The objectives of PSP0.1 are to help you to

- measure the size of the programs that you produce
- perform size accounting for the programs that you produce
- make accurate and precise size measurements



# New Process Elements

---

There are two new process elements.

- process improvement proposal (PIP) form
- size counting and coding standards

The project plan summary has been expanded.

- a Program Size Summary section has been added
- planned time in phase is calculated based on historical time in phase percentage



# **PSP0.1 Process Script Additions**

---

The additions to the PSP0.1 process scripts include new steps for

- estimating and reporting software size
- distributing development time over planned project phases
- using a size counting and coding standard
- recording process problems and improvement ideas



# Process Improvement Proposal -1

---

To improve your process, you will need to capture process problems and propose improvements for future reference.

You will need to know

- any problems you have encountered in using the process
- any suggestions you have for process improvements
- your observations and findings from doing the assignments



# Process Improvement Proposal -2

You should complete a PIP form for each assignment.

The PIP holds process improvement information.

- date
- problem description
- proposed solution
- notes and comments

The screenshot shows a web-based form titled "PSP Process Improvement Proposal". The form is divided into several sections:

- Header:** Includes the Carnegie Mellon Software Engineering Institute logo and the title "PSP Process Improvement Proposal".
- Form Fields:**
  - Student: James Over
  - Program: Assignment 2
  - Instructor: Watts
  - Start Date: 05-Mar-05
  - End Date: (empty)
  - Language: C
  - Date: 05-Mar-05
- Problem Description:** A section with the heading "Problem Description" and the instruction "Briefly describe the problems you encountered." Below this is a large text area.
- Proposal Description:** A section with the heading "Proposal Description" and the instruction "Briefly describe the process improvements that you propose." Below this is a large text area.
- Other Notes and Comments:** A section with the heading "Other Notes and Comments" and the instruction "Note any other comments or observations that describe your experiences or improvement ideas." Below this is a large text area.
- Navigation:** At the bottom, there are navigation controls including "Record:" and "of 1" (or "of 1 (Filtered)").



# Coding and Counting Standards

---

Coding and size counting standards are needed to write the PSP programs.

These standards are

- tailored to your language and needs
- designed to make size counting easier

The coding standard defines quality-oriented exit criteria for the code phase.





# **PSP Software Size Measures**

---

In the PSP, software size measures are used to

- relate the amount of product produced with effort expended to project total effort
- calculate productivity
- normalize defects
- project the total defects

Software size is measured in LOC.

To accurately relate size to effort, the different types of LOC in your program are counted separately.



# PSP0.1 Project Plan Summary

PSP0.1 adds the Program Size Summary for estimated and actual size data.

The types of size include

- base [B]
- deleted [D]
- modified [M]
- added [A]
- reused [R]
- added and modified [A+M]
- new reusable

**PSP0.1 Project Plan Summary**

Student: jwo  
Program: Assignment 2  
Instructor: wsh  
Start Date: 13-Mar-05  
End Date:   
Language: C

**Program Size Summary** LOC-Lines of code

	Plan Size	Actual Size	To Date
Base (B)	0	0	
Deleted (D)	0	0	
Modified (M)	0	0	
Added (A)	0	0	
Reused (R)	0	0	0
Added & Modified (A&M)	0	0	0
Total (T)	0	0	0
New Reusable (NR)	0	0	0

**Time in Phase**

Phase	Plan	Actual	To-Date	To-Date%
PLAN	0.0	0	0	0.0%
DLD	0.0	0	0	0.0%
CODE	0.0	0	0	0.0%
COMPILE	0.0	0	0	0.0%
UT	0.0	0	0	0.0%
PM	0.0	0	0	0.0%
Total	0	0	0	

**Defects Injected in Phase**

Phase	Plan	Actual	To-Date	To-Date%
PLAN	0	0	0	0.0%
DLD	0	0	0	0.0%
CODE	0	0	0	0.0%
COMPILE	0	0	0	0.0%
UT	0	0	0	0.0%
PM	0	0	0	0.0%
Total	0	0	0	

**Defects Removed in Phase**

Phase	Plan	Actual	To-Date	To-Date%
PLAN	0	0	0	0.0%
DLD	0	0	0	0.0%
CODE	0	0	0	0.0%
COMPILE	0	0	0	0.0%
UT	0	0	0	0.0%
PM	0	0	0	0.0%
Total	0	0	0	

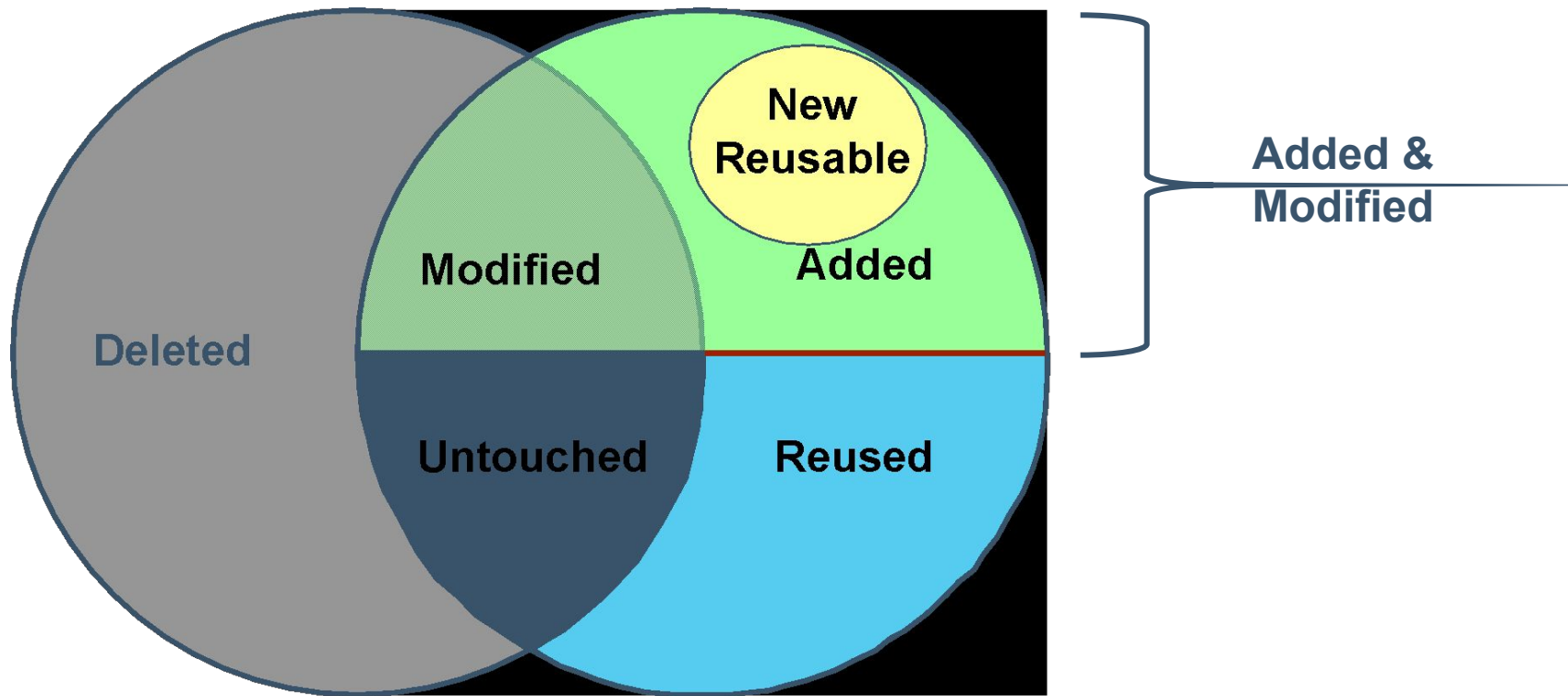
Record: 1 of 1 (Filtered)



# Program Size Type Relationships

Base program

New program





# Estimating Size

During planning

1. If this is an enhancement, measure the size of the base program and enter this in the Base (B) space under Actual.
2. Estimate the added and modified size and enter this in the Total Added and Modified (A+M) space under Plan.

Program Size Summary		LOC-Lines of code	
	Plan Size	Actual Size	To Date
Base (B)		<b>1</b> 0	
Deleted (D)		0	
Modified (M)		0	
Added (A)		0	
Reused (R)		0	0
Added & Modified (A&M)	<b>2</b> 0	0	0
Total (T)		0	0
New Reusable (NR)		0	0



# Estimating Development Time

During planning

1. Enter estimated development time
2. Planned time in phase is then calculated based on the percentage of time in phase for all completed projects

Time in Phase				
Phase	2 Plan	Actual	To-Date	To-Date%
PLAN	0.0	0	0	0.0%
DLD	0.0	0	0	0.0%
CODE	0.0	0	0	0.0%
COMPILE	0.0	0	0	0.0%
UT	0.0	0	0	0.0%
PM	0.0	0	0	0.0%
Total	1 0	0	0	



# Recording Size -1

During postmortem

1. Measure total program size and enter this in the Total Size (T) space under Actual.
2. Count the deleted size and enter this in the Deleted (D) space under Actual.
3. Count the modified size and enter this in the Modified (M) space under Actual.

Program Size Summary		LOC-Lines of code	
	Plan Size	Actual Size	To Date
Base (B)		0	
Deleted (D)		2	
Modified (M)		3	
Added (A)		0	
Reused (R)		0	0
Added & Modified (A&M)	1	0	0
Total (T)		1	0
New Reusable (NR)		0	0



# Recording Size -2

During postmortem

1. Count the reused size and enter this in the Reused (R) space under Actual.
2. Count or estimate the number of new and changed size that will be added to the reuse library and in the New Reusable space under Actual.

Program Size Summary		LOC-Lines of code	
	Plan Size	Actual Size	To Date
Base (B)		0	
Deleted (D)		0	
Modified (M)		0	
Added (A)		0	
Reused (R)		1	0
Added & Modified (A&M)		0	0
Total (T)		0	0
New Reusable (NR)		2	0



# Message to Remember

---

Your principal objective is to measure and estimate the size of the programs that you produce so that you can effectively plan and manage your work.