



PSPSM — Personal Software ProcessSM (Proceso de Software Personal)

SM Personal Software Process y PSP son marcas de Servicio de Carnegie Mellon University

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

1 - 1



Introducción de los participantes



Introducción de los alumnos:

Nombre y organización.

Experiencia.

¿Qué esperan del curso?

Lenguaje de programación que van a utilizar.

Introducción del instructor.

Nombre y organización.

Experiencia.

¿Qué espera del curso?

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

2 - 1



Objetivos del curso



Este curso no es de programación, se espera que los asistentes ya tengan experiencia en el lenguaje de programación que van a utilizar en este curso.

Al igual, se espera que también conozcan el ambiente de programación que van a utilizar.

Los objetivos del curso son:

1. Introducir a los participantes en un enfoque de desarrollo de software basado en procesos.
2. Enseñar a medir este proceso y a analizar los datos recabados.
3. Enseñar a utilizar la información del análisis de los datos personales para mejorar el proceso personal de desarrollo de software.



Herramientas requeridas



Se requiere que los asistentes:

Ya hayan leído los primeros 5 capítulos de "A discipline for Software Engineering" de Watts Humphrey.

Tengan conocimiento de estadística básica.

Cuenten con algún procesador de palabras y hoja de cálculo (de preferencia Microsoft Word y Microsoft Excel).



Estructura del Curso



El curso está dividido en dos partes:

Parte I Planeación.

- Introducción al proceso personal.
- Medición de tamaño.
- Estimación del tamaño.
- Estimación basada en proxies.
- Estimación de recursos.
- Medición del proceso.

Parte II Calidad

- Administración de defectos.
- El proceso de diseño.
- Verificación del diseño.
- Escalando hacia TSP.
- Aplicando PSP y TSP.



Nuestro contexto en la Ingeniería de Software



Contestar la encuesta.



Recomendación



El libro "A Discipline for Software Engineering" cubre todo el material requerido en este curso, al igual, incluye un número de tópicos que son más adecuados a los intereses académicos y no esenciales para la enseñanza de los profesionales del desarrollo de software.

Por lo tanto, en el momento que consideren que se está profundizando demasiado en temas académicos (estadísticos principalmente), háganlo notar para avanzar a temas más concretos y pragmáticos.



Unidad I



Una disciplina en la Ingeniería de Software



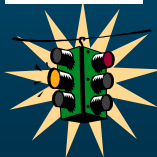
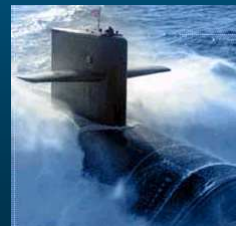
Evolución




- Más grande.
- Más complejo.
- Más demandante.




Presencia





Objetivos





08/09/2008
SM© Carnegie Mellon University
Proceso de Software Personal
Carlos Mojica
11 - I



¿Falta tiempo?!





“Bomberazos” - no hay tiempo para planear.
Más “bomberazos” - no hay tiempo para mejorar.
Más, más, ... , “bomberazos” - no hay tiempo para otra cosa (estimar, registrar datos, analizar, etc.).

Atención en el problema actual.

08/09/2008
SM© Carnegie Mellon University
Proceso de Software Personal
Carlos Mojica
12 - I

Ejemplos de desastres

- Cohete Ariane.
- Sistema de maletas del aeropuerto de Denver.

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

13 - I

Reflexión

¿Me dejaría operar con
software creado por mí?

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

14 - I



El trabajo del Ingeniero de Software



Todo trabajo (producto/servicio) de un Ingeniero de Software se debe ajustar a:

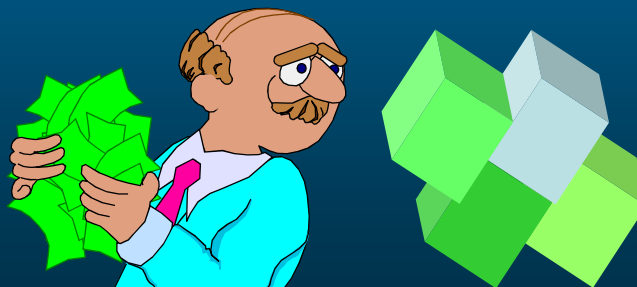
- 1 Producir/Brindar productos/servicios de calidad.
- 2 Hacerlos dentro del costo.
- 3 Completarlos en el tiempo.



¿Qué versus Cómo?



El **cómo** hacer es tan importante como el **qué** se va a hacer.

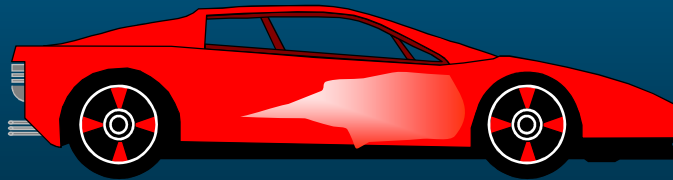




Producto



Si nos centramos en el producto (**el qué**), podemos no darnos cuenta de cómo mejorarlo.

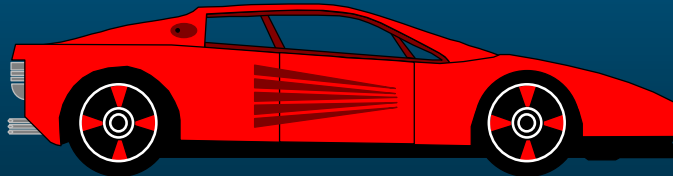


Proceso



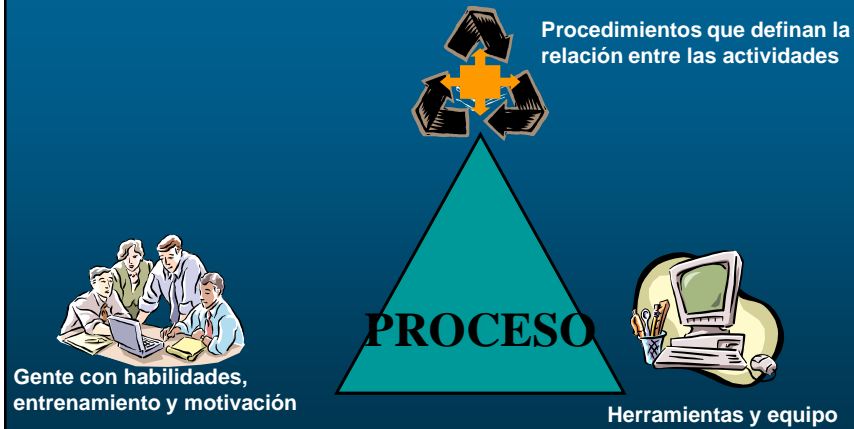
Si nos centramos en el **proceso** (el cómo), podemos:

- *Predecir la repetición de la salida.
- *Conocer las tendencias del proyecto.
- *Controlar las características de calidad en el producto.





Proceso



Creatividad vs. Disciplina



La **creatividad** emerge y se mejora con disciplina.





¿Podemos cambiar?



El desarrollo de software es una actividad muy intensa, *personas en forma individual* siguen escribiéndolo.

Cualquier mejora en la eficiencia o productividad *de estas personas*, resultará en ganancia en los proyectos y en la industria en general.



¿Qué hace falta?



Necesitamos una *disciplina* de Ingeniería de Software que provea a las personas que desarrollan los componentes, un método para planear, dar seguimiento y administrar efectivamente los defectos. Más aún, que les permita aprender de sus éxitos y fracasos.



¿Qué es PSP? [1/5]



PSP es un método basado en procesos, es un método de mejora continua a nivel personal.

Con PSP tenemos la oportunidad de aprender a usar y ver los beneficios de trabajar de una manera *disciplinada* y orientada a procesos.

PSP fue desarrollado en el Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon por Watts Humphrey.



¿Qué es PSP? [2/5]



PSP = Planeación* + Diseño + Revisiones de Diseño* +
Codificación + Revisiones de Código* + Pruebas

Estimar.

Dar seguimiento.

Administrar el esfuerzo del software.

Incrementar la calidad del software.

Reducir el costo del software.

Reducir los ciclos de desarrollo.

Mejorar la productividad.

Eliminar los defectos antes de las pruebas.

Minimizar el costo de las pruebas.



PSP



PSP es un marco de trabajo basado en **procesos**.

Enseña y ayuda a los Ingenieros de Software a:

- Planear y dar seguimiento.

- Cumplir sus compromisos.

- Crear productos de calidad.

- Resistir a presiones irracionales

- Identificar **sus áreas de mejora**.

Lo anterior con el registro y análisis de los datos emanados de **sus propios procesos**.



Si otros pueden, ¿por qué nosotros no?



Otros profesionales, como los químicos, cirujanos, pilotos aviadores, músicos, demuestran competencias básicas antes de realizar aún los procedimientos más simples.

¿Por qué los Ingenieros de Software no?

Con PSP, podemos obtener la competencia necesaria.



27 - 1

28 - I



¿Qué es PSP? [5/5]



Procedimientos

Phase Number	Purpose:	To guide you in developing module-level programs.
	Inputs Required	<ul style="list-style-type: none">• Problem description• PSP2.1 Project Plan Summary form• Size Estimating Template• Historical estimated and actual size and time data• Time and Defect Recording Logs• Defect Type Standard• Stop watch (optional)
1	Planning	<ul style="list-style-type: none">• Produce or obtain a requirements statement.• Use the PROBE method to estimate the total new and changed LOC required and the prediction interval.• Complete the Size Estimating Template.• Use the PROBE method to estimate the required development time and the prediction interval.• Complete a Task Planning Template.• Complete a Schedule Planning Template.• Enter the plan data in the Project Plan Summary form.• Complete the Time Recording Log.
2	Development	<ul style="list-style-type: none">• Design the program.• Review the design and fix and log all defects found.• Implement the design, <i>using design templates where appropriate</i>.• Review the code and fix and log all defects found.• Compile the program and fix and log all defects found.• Test the program and fix and log all defects found.• Complete the Time Recording Log.
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
	Exit Criteria	<ul style="list-style-type: none">• A thoroughly tested program• Completed Project Plan Summary with estimated and actual data• Completed size estimate• <i>Completed design templates</i>• Completed Design Review Checklist and Code Review Checklist



Principios en los que se basa PSP



- 1 La calidad de los sistemas de software depende de la calidad de sus peores componentes.
- 2 La calidad de los componentes de software depende del conocimiento, disciplina y compromiso del individuo que lo desarrolló.
- 3 Los ingenieros de software deben conocer su rendimiento, medir, dar seguimiento y analizar su trabajo y aprender de las variaciones de sus rendimientos e incorporar estas lecciones a sus prácticas personales.





PSP les sirve a los Ingenieros de Software para [1/2]



- 1 Conocer su rendimiento: al medir su trabajo, reconocer que funciona mejor y aprender a como repetirlo y mejorarlo.
- 2 Entender las variaciones: lo que es repetible y lo que pueden aprender de las variaciones.
- 3 Incorporar estas lecciones en una creciente documentación de prácticas personales.



08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

31 - I



PSP les sirve a los Ingenieros de Software para [2/2]



Pasar de ser programadores a ser verdaderos Ingenieros de Software

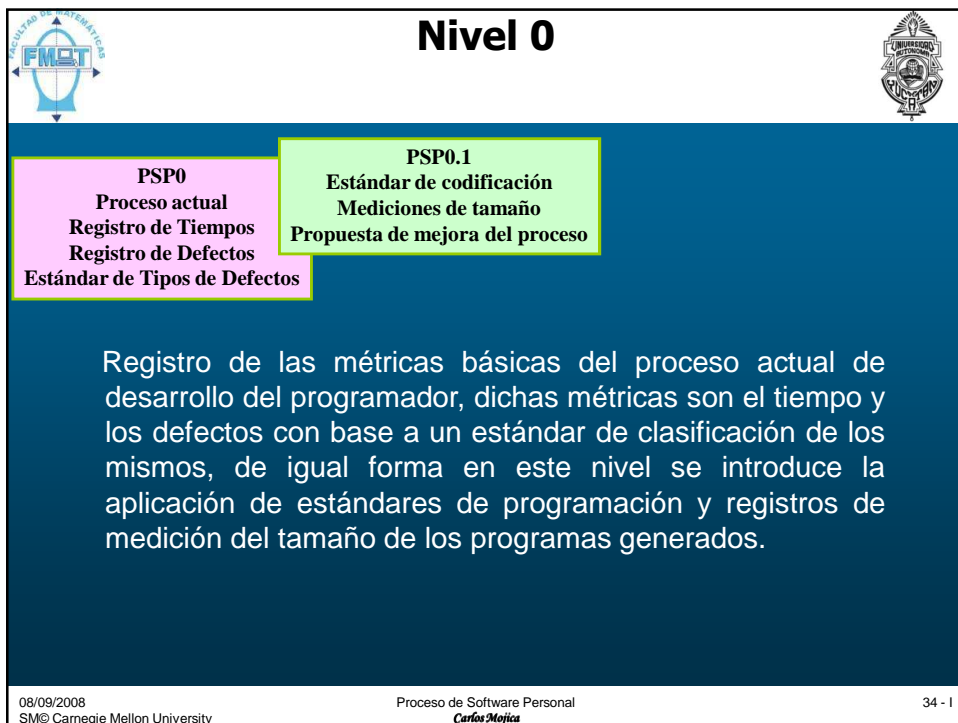
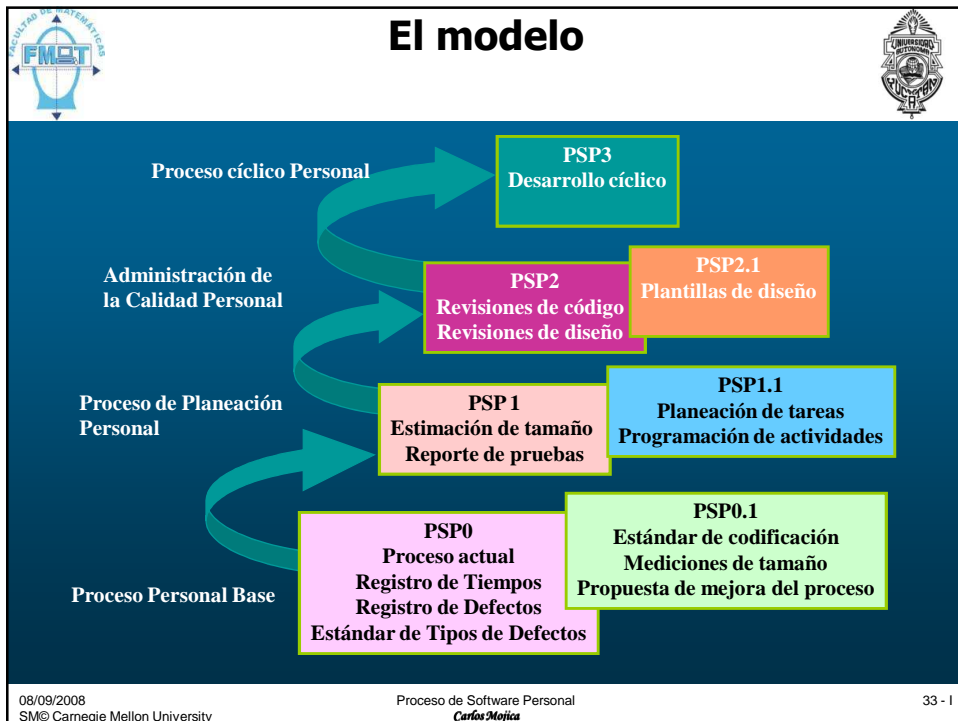


Sin embargo, no es la panacea a todos los problemas de la IS. Todo es posible de ser perfeccionado, pero hay que empezar por algo y PSP es una buena opción.

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

32 - I





Nivel 1



PSP 1
Estimación de tamaño
Reporte de pruebas

PSP1.1
Planeación de tareas
Programación de actividades

Enseña al programador a estimar el tamaño y el tiempo que ocupará en cada una de las fase de implementación de sus programas; esta estimación está basada en métodos estadísticos e información histórica que el mismo programador ha generado a lo largo de la aplicación del nivel 0 en varios programas.



Nivel 2



PSP2
Revisiones de código
Revisiones de diseño

PSP2.1
Plantillas de diseño

Las revisiones permiten verificar el aumento de la calidad de los productos generados por el programador y minimizar el número de defectos inyectados en su diseño y en su programación, esto permite que el número de defectos en las fases de pruebas y entrega al cliente disminuyan significativamente y que por lo tanto el tiempo de detección y corrección de defectos se minimicen, de igual forma en este nivel se incorporan formatos para realizar el diseño de los programas en forma estándar.



Nivel 3



PSP3 Desarrollo cíclico

Se incorporan técnicas de desarrollo cíclico a fin de permitir utilizar los procedimientos definidos por el Personal Software Process a proyectos de mayor tamaño y de mayor duración.



Algunos resultados obtenidos

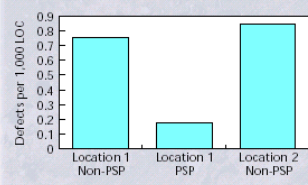
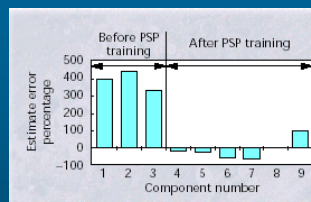
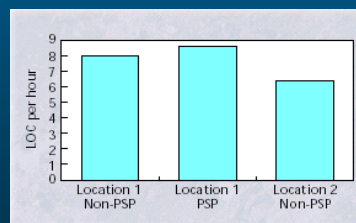


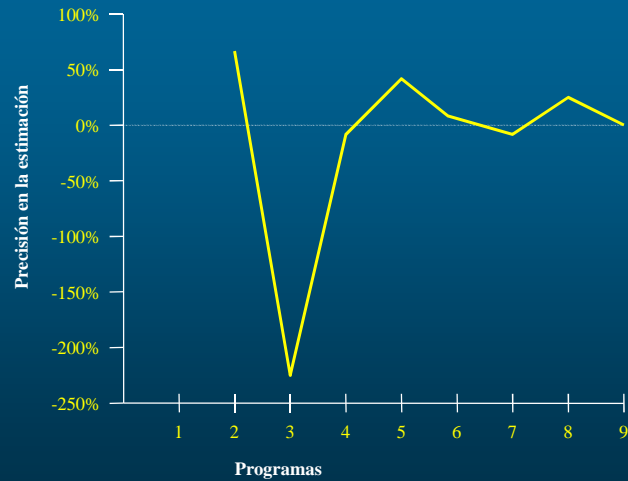
Table 2. AIS system test time improvement.		
Project	Size	System test time
Non-PSP Projects		
A1	15,800 LOC	1.5 months
C	19 requirements	3 test cycles
D	30 requirements	2 months
H	30 requirements	2 months
PSP Projects		
A2	11,700 LOC	1.5 months
B	24 requirements	5 days
E	2,300 LOC	2 days
F	1,400 LOC	4 days
G	6,200 LOC	4 days
I	13,300 LOC	2 days



Results of applying the
Personal Software Process
Pat Ferguson et al.
IEEE Computer 1997



¿Cuál es el efecto en la estimación del tamaño con PSP?



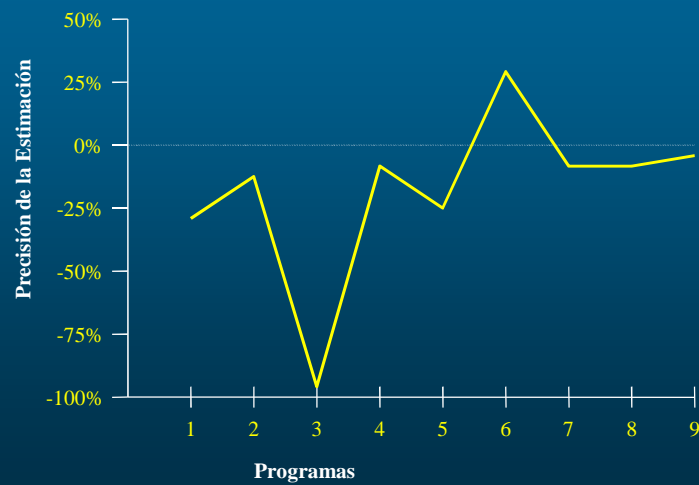
08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

39 - I



¿Cuál es el efecto en la estimación del esfuerzo con PSP?



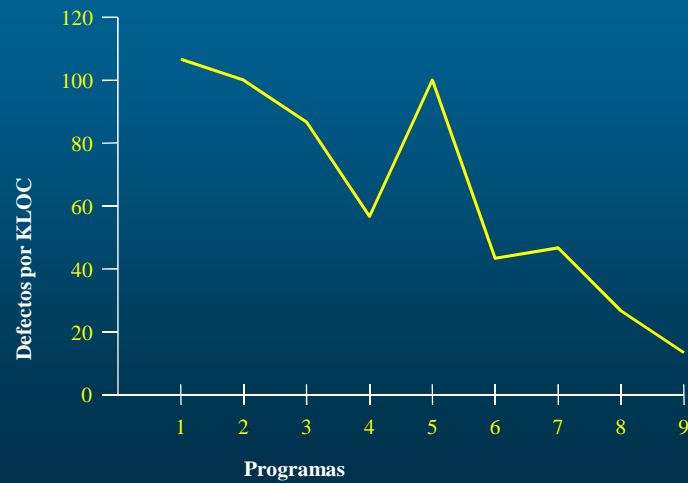
08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

40 - I



¿Cuál es el efecto en la densidad de defectos con PSP?



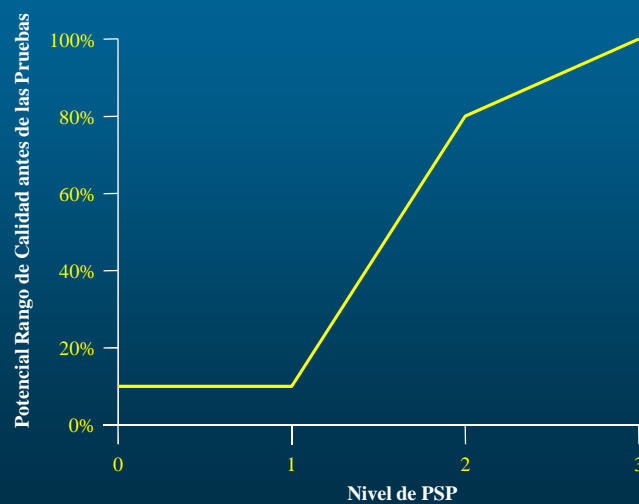
08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

41 - I



¿Cuál es el efecto en la calidad antes de las pruebas con PSP?



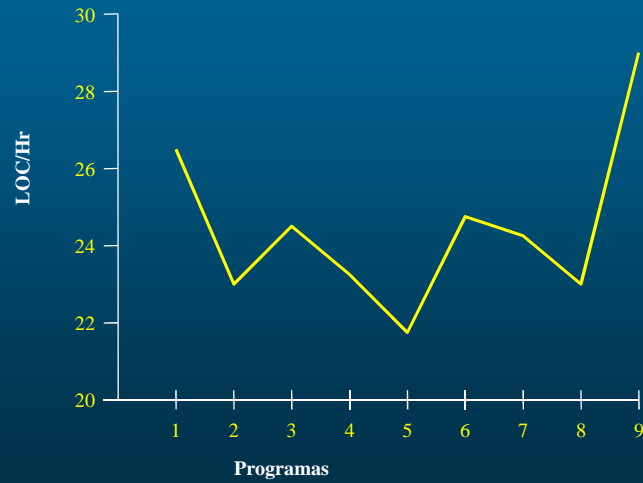
08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

42 - I



¿Cuál es el efecto en la Productividad con PSP?



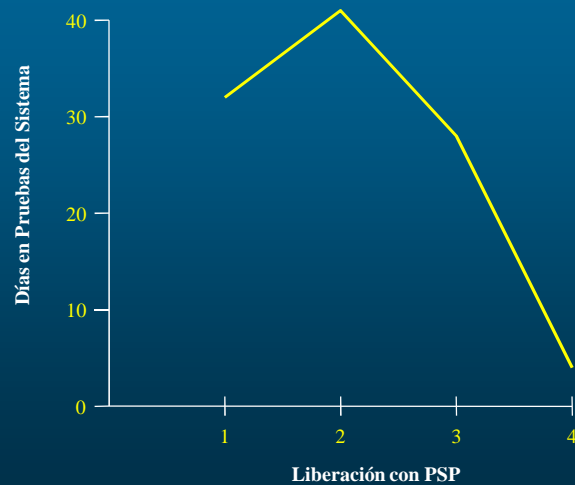
08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

43 - I



¿Cuál es el efecto en las Pruebas con PSP?



08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

44 - I



PSP y CMM

¿ Cual es la diferencia ?



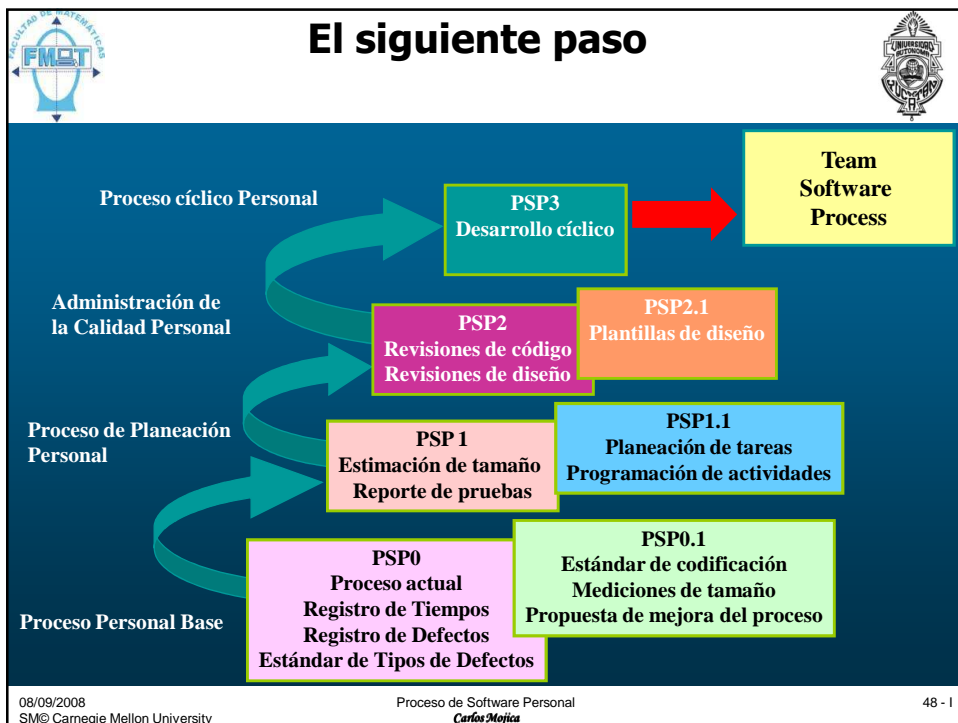
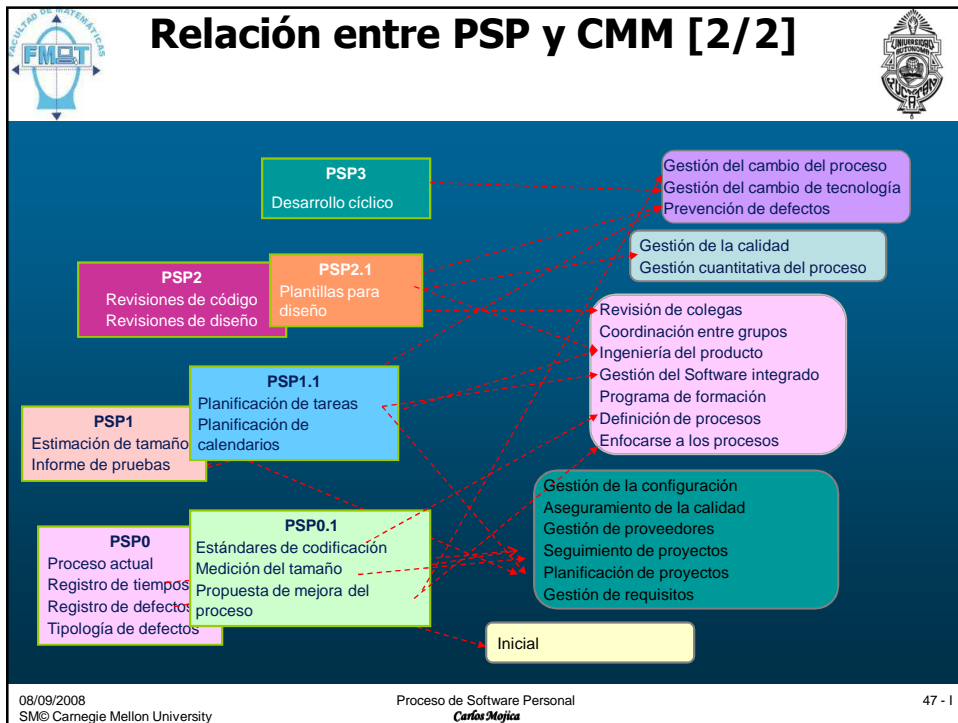
- CMM es un modelo para evaluar los procesos de software de una unidad de desarrollo de software.
- CMM define un conjunto de buenas prácticas de ingeniería de software, pero no muestra como realizarlas.
- PSP es un marco de trabajo que define un conjunto de procesos más que únicamente metas como CMM.
- PSP no puede ser utilizado para evaluar niveles de capacidad de los procesos.



Relación entre PSP y CMM [1/2]

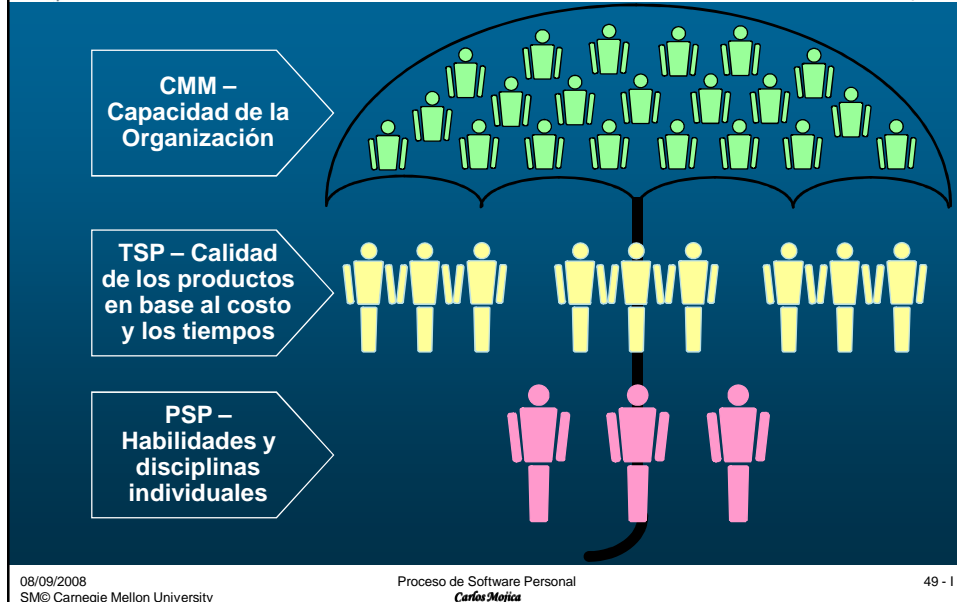


- Las diferentes partes de PSP introducen paulatinamente técnicas para llevar a cabo diferentes prácticas clave asociadas con áreas clave de CMM.
- Las prácticas clave implantadas por PSP son realizadas a nivel individual.
- PSP ayuda a la implantación de prácticas de las áreas clave de CMM desde abajo hacia arriba, es decir desde un nivel individual, hasta la institucionalización de dichas prácticas a lo largo de toda la organización, como es requerido por los niveles de CMM.





PSP, TSP, y CMM



¿PSP, la solución?



Sin embargo, PSP no es la panacea a todos los problemas de la Ingeniería de Software. Todo es posible de ser perfeccionado, pero hay que empezar por algo y PSP es una buena opción.



PSP0

Proceso actual
Registro de Tiempos
Registro de Defectos
Estándar de Tipos de Defectos



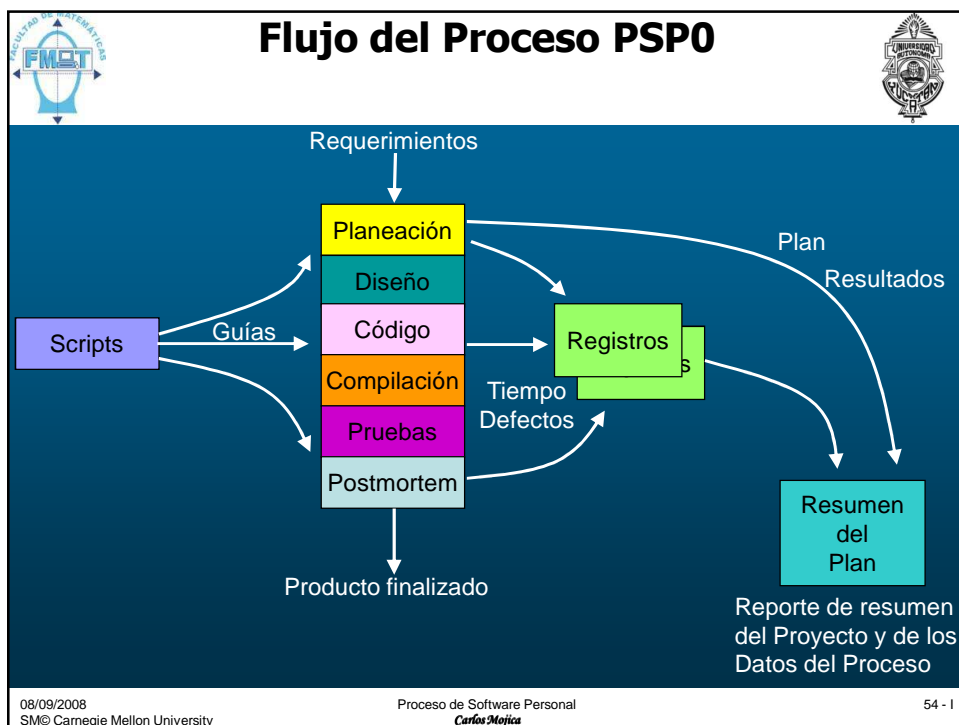
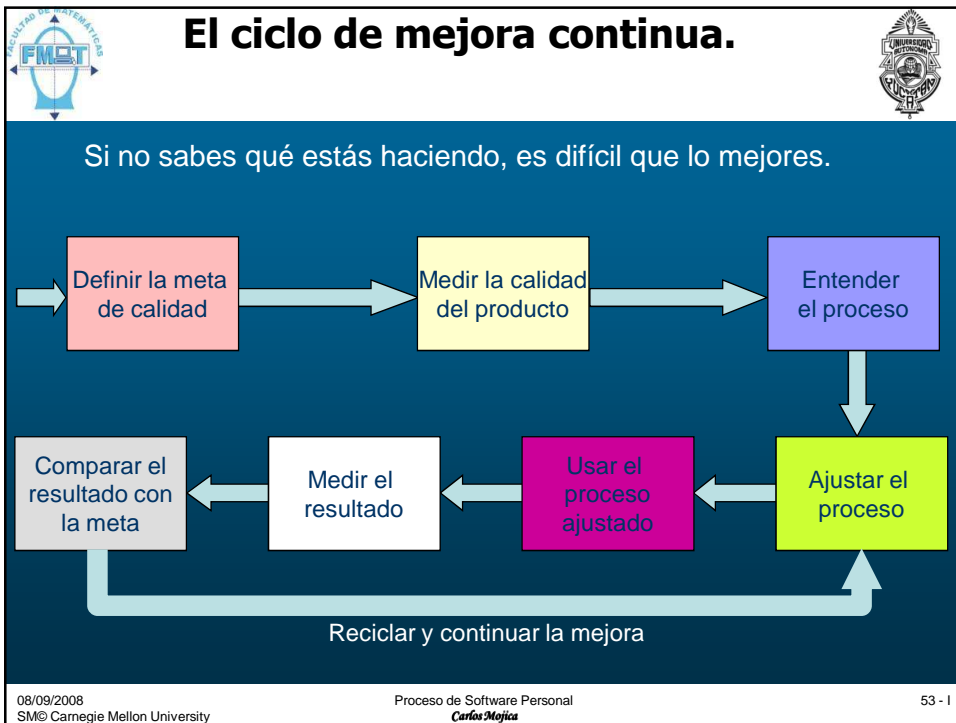
La Base de la mejora personal



PSP0 es la base para la mejora del proceso. Su principal objetivo es proveer el marco de trabajo para recolectar los datos iniciales de **tu proceso. Con la recolección de los datos, obtendrás un entendimiento cuantitativo de tu manera de trabajar.**

Un proceso definido te permite la recolección de datos sobre el tiempo que inviertes en cada tarea de software y darle seguimiento al número de defectos que inyectas y remueves en cada paso del proceso.

Estos datos te ayudan a analizar tu proceso, entender sus fallas y mejorarlo.





PSP0 provee

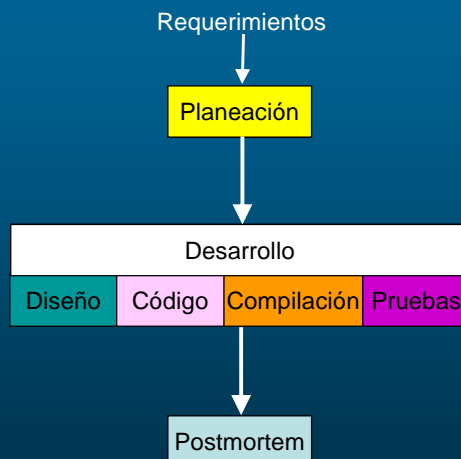


- Una estructura conveniente para realizar tareas a pequeña escala.
- Un marco de trabajo para medir estas tareas.
- Una base para la mejora de procesos.

Planear productos, productos pequeños, para que los podamos crear libres de defectos, sin dejar *algo* para después.



El Proceso PSP0



Aunque para pequeñas tareas no son necesarias la planeación y el análisis postmortem, son esenciales para la disciplina del proceso personal. Sin embargo, muchos programas simples tienen sorpresas que un plan puede anticipar.



Los scripts de PSP



Un script de PSP, es una guía en la aplicación del proceso. Los elementos principales del script son: su propósito, el criterio de entrada, las fases o pasos a ser realizados y el criterio de salida.

El Script del Proceso PSP0.
El Script de la fase de Planeación de PSP0.
El Script de la fase de Postmortem de PSP0.
El Script de la fase de Desarrollo de PSP0.



Las formas de PSP



PSP utiliza muchas formas. Las formas te ayudan a evitar el desperdiciar el tiempo pensando el formato y lo que debe llevar. Pero, las formas buenas son difíciles de construir, por lo que deben ser revisadas periódicamente para ver que cumplan con las necesidades del momento.



Las medidas en PSP0



- El tiempo invertido por fase.
- Los defectos encontrados por fase.



Administración del tiempo [1/3]



La base lógica de la administración del tiempo es la siguiente:

Se empleará la misma cantidad del tiempo y de la misma manera esta semana como se hizo la semana pasada.

Por supuesto que hay excepciones. Por ejemplo, durante el periodo de exámenes, no se atiende al mismo número de clases por estar estudiando. Otro ejemplo, el caso de la presencia de un huracán.



Administración del tiempo [2/3]



Para realizar planes realistas, se le tiene que dar seguimiento a la manera en que se emplea el tiempo.

La gente recuerda algunas cosas y olvida otras. Por ejemplo, la cantidad de tiempo empleado para hacer la tarea; mientras que el tiempo empleado en *relajación* y comidas, podría ser más de lo pensado.



Administración del tiempo [3/3]



Para saber a donde se va el tiempo, se necesita mantener un registro preciso.

Para chequear la precisión de las estimaciones del tiempo usadas en los planes, se debe documentar estas estimaciones y luego compararlas con lo que realmente se hace.



Para saber que hacemos [1/2]



Para conocer como invertimos nuestro tiempo y poder utilizar esta información debemos:

- Categorizar las actividades.
- Registrar el tiempo empleado en cada actividad.
- Registrar el tiempo de manera estandarizada.
- Guardar los datos en un lugar conveniente.



Para saber que hacemos [2/2]



Cuando la gente dice que está trabajando duro, lo que realmente quiere decir es que está trabajando más horas, ¡no que son más productivos!

[illegible]

65 - 1





Tipos de defectos



Cuadro de tipos de defectos.



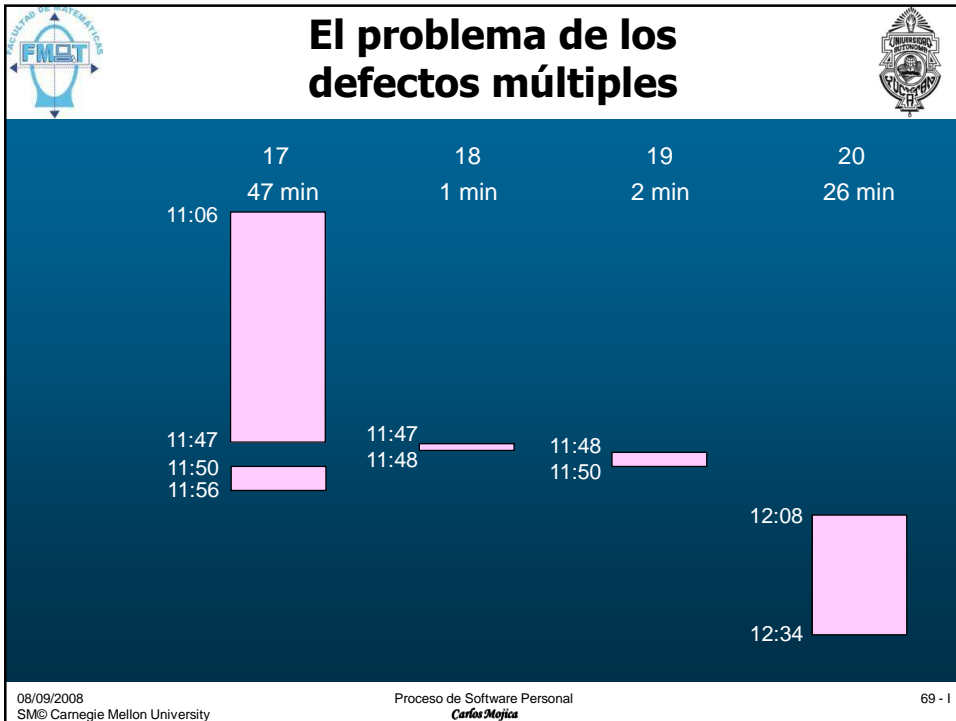
Formato de Registro de Defectos



Hay que aprender de los defectos que encontramos en nuestros programas, pero más importante, hay que aprender de los defectos que otros encuentran en nuestros programas.

Formato de registro de defectos.

La falla más grande es fallar en aprender de lo que hemos fallado.



Forma de Resumen del Plan de Proyecto

Ver forma.

08/09/2008
SM© Carnegie Mellon University

Proceso de Software Personal
Carlos Mojica

70 - I



Ejercicio 1A Usar PSP0



Escribir un programa para calcular la media y la desviación estándar de una serie de n números reales.

La media es el promedio de los números. La fórmula para la desviación estándar es:

$$\text{Std} = \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - x_{\text{avg}})^2}{n - 1}}$$

Usar una lista ligada para guardar los n números para el cálculo, los cuales deben ser leídos desde el teclado o desde un archivo (el usuario seleccionará).



Prueba del ejercicio 1A



Utilizar los siguientes datos para realizar las pruebas.

Número de programa	LOC por Objeto	LOC Nuevas y Cambiadas	Hrs. de desarrollo
1	160	186	15.0
2	591	699	69.9
3	114	132	6.5
4	229	272	22.4
5	230	291	28.4
6	270	331	65.9
7	128	199	19.4
8	1657	1890	198.7
9	624	788	38.8
10	1503	1601	138.2
Desv. estándar	572.03	625.63	62.26



Orden de entrega de las formas



Resumen del Plan del Proyecto nivel PSP0.
Forma de Registro de Tiempos.
Forma de Registro de Defectos.
Código Fuente del Programa.
Pantallas de la interfase gráfica.
Pantallas de los resultados.