

### **CAPITULO 3. PSP 0 Y PSP 0.1**

A partir de este capítulo se introducen términos, probablemente nuevos para el lector que tienen que ver en su totalidad con PSP. También se dan a conocer los formatos, "*scripts o templates*" y estándares que se utilizan en los distintos niveles de PSP, pero solamente se analizan a fondo los que tienen que ver con PSP 0, PSP 0.1 y PSP 1.

PSP hace uso de un gran número de formatos los cuales son muy útiles para que se haga un análisis a fondo del programa que se desarrollará. Todo programa tiene una serie de pasos definidos para ir cumpliendo con los requerimientos del cliente de manera uniforme y disciplinada. Para ir cumpliendo con estos pasos es necesario tener en cuenta lo siguiente:

- Determinar de que forma se va a resolver el programa.
- Decidir cómo se va a "*atacar*" el problema.
- Resolverlo.
- Estar seguro de que se solucionó de la forma correcta.
- Arreglar los problemas que se han encontrado.
- Entregar el resultado final.

Está claro que éstos pasos toman mucho tiempo, sin embargo cuando está bien definido lo que se va a hacer, sólo resta llenar los formatos hasta obtener el resultado deseado.

Entonces el problema reside en desarrollar buenos planes y por lo tanto, buenos formatos. Está comprobado que procesos automatizados ayudan a resolver el problema de la manera más simple y fácil. [HUMPHREY; 95]

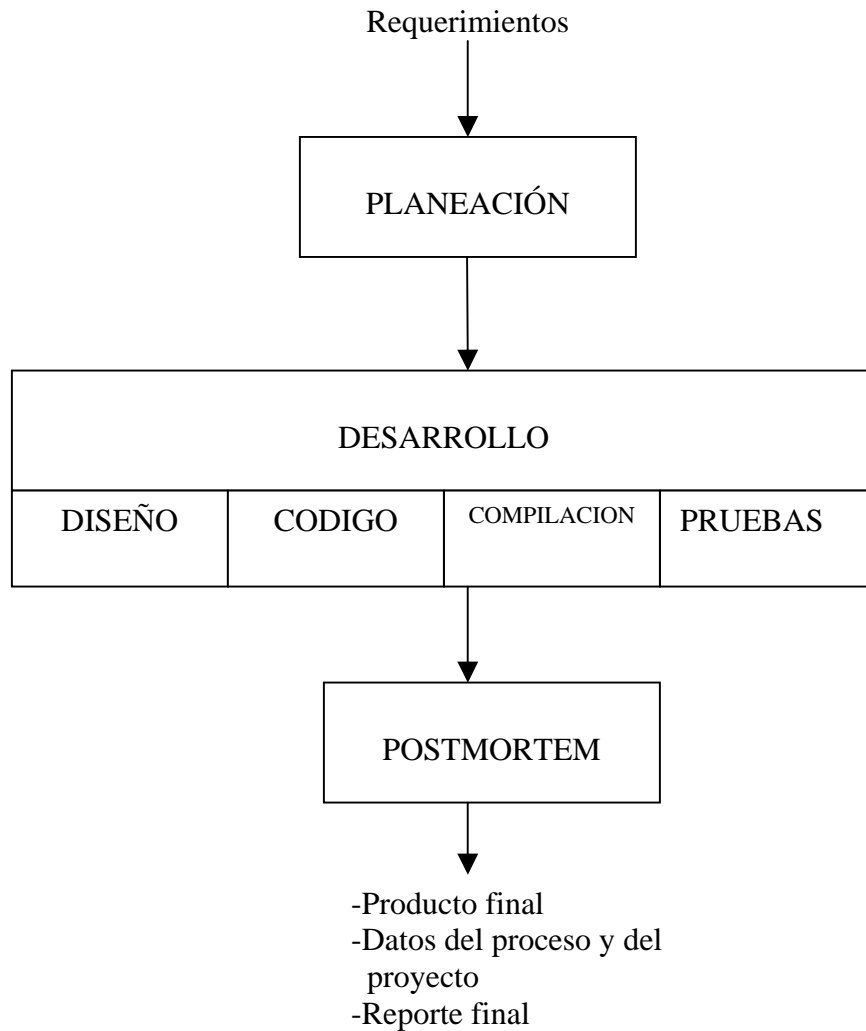
Los procesos definidos ayudan a administrar grandes proyectos, ya sea trabajando en equipo y/o trabajando sólo. Lo primero que se tiene que hacer para definir los procesos que van a intervenir en un proyecto es:

1. Identificar las actividades principales.
2. Separar los elementos complejos que pueden intervenir.
3. Establecer los criterios de entrada y de salida para cada fase del proceso.
4. Medir de manera correcta el proceso, para tener bien entendido el desempeño personal.
5. Estimar correctamente cuando debe finalizar cada tarea.
6. Medir con precisión todos los datos que intervinieron para futuros programas.
7. Identificar las fases del proyecto que más problemas causaron.
8. Mejora continua tomando en cuenta datos anteriores.

Una vez que se han tomado en cuenta todos estos pasos fundamentales para comenzar con el nivel inicial de PSP, las tareas toman una forma mucho más estructurada y racional. Mientras se tenga claridad suficiente respecto al objetivo, las mejoras en las fases del proceso serán notables.

### **3.1 INTRODUCCION A PSP 0 Y PSP 0.1**

La figura 3.1 muestra el proceso inicial (PSP 0) en donde primero se desarrolla un plan para realizar el proyecto. Después se desarrolla el software necesario y finalmente viene la fase de *postmortem* o la fase que se realiza después de finalizado el proyecto.

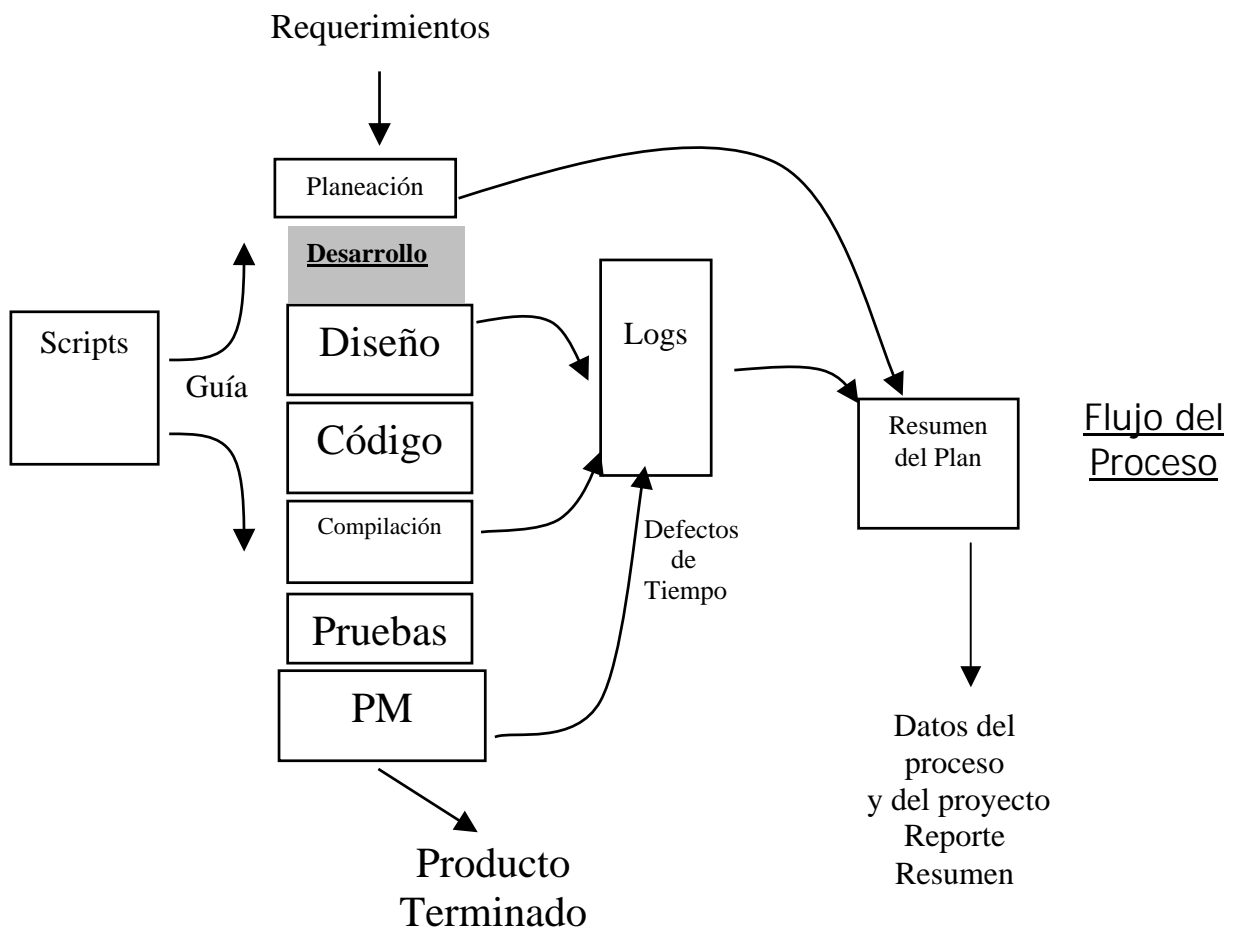


**Figura 3.1** "El proceso PSP 0" [HUMPHREY; 95]

Esta última fase es en la que se compara el desempeño final con el desempeño planeado; también se registran los datos que tienen que ver con el proceso para que finalmente los datos finales se pongan en el reporte final.

Todo estas fases están diseñadas de esta manera porque el objetivo principal del nivel inicial de PSP (PSP 0) es el de proveer un marco bien definido.

Para realizar la recolección de todos los datos que intervienen en el proyecto es muy importante este primer paso ya que proporciona un panorama general de qué métodos utilizar para "atacar" el problema. El flujo que interviene en todo el proceso inicial está detallado en la figura 3.2. Es de gran importancia que se entienda claramente este concepto porque representa los fundamentos de los demás niveles.



**Figura 3.2** "Flujo de PSP 0" [HUMPHREY; 95]

Es necesario notar que éste flujo es muy general y aplicable a cualquier proceso conocido, debido a que sigue un orden lógico.

Dado que PSP tiene un número de métodos que generalmente los ingenieros no practican, estos métodos se introducen en una serie de siete versiones del proceso completo. Estas versiones se etiquetan desde PSP 0 hasta PSP 3, y cada versión tiene un sistema similar de registros, de formatos, scripts, y estándares. Los scripts que intervienen en el proceso definen los pasos para cada parte del proceso.

Los registros y los formatos proporcionan las plantillas para almacenar los datos y los estándares ayudan a dirigir a los ingenieros durante todo el proyecto que desarrollan, desde el principio hasta el final de éste.

Sin embargo también hay que notar que si el deseo del usuario es el de adaptar este nivel a los hábitos de programación personales, entonces éste tiene que modificar los *scripts*, formatos y registros para que se adapten a sus necesidades.

PSP 0 provee una estructura bien definida para tareas pequeñas y la medición de éstas, posteriormente se puede optar por continuar a los niveles siguientes. Como ya se ha visto el primer paso en el proceso de PSP es la planeación. Para este paso existe un *script* de planeación que ayuda a dirigir este trabajo en todos sus niveles y también es necesario llevar un resumen del plan para registrar todos los datos de esta fase de planeación.

De esta forma los ingenieros registran cada dato que interviene en el desarrollo del programa, es decir, registran su tiempo en cada proceso y los datos de los defectos que se encuentran durante todo el proceso de desarrollo e inclusive desde la planeación. Al final del trabajo, durante la fase de postmortem (P.M.), miden el tamaño del programa, e incorporan estos datos en el formato del resumen del proyecto.

Cuando todos estos formatos están hechos, entonces los ingenieros entregan el producto acabado junto con el formulario lleno del resumen del proyecto.

## **3.2 REQUISITOS**

Todas las fases que se emplean en PSP requieren de un *script* para cada etapa del ciclo de vida de cualquier programa que se desarrolla.

Estas etapas, que sólo el Proceso Personal de Software (PSP), utiliza son:

1. Planeación.
2. Desarrollo.
3. Postmortem.

Todas estas etapas forman parte de un proceso integral y a su vez estas etapas tienen sus requerimientos que le exigen al programador. De manera general todo se engloba en el *script del proceso* que ayuda al programador a desarrollar programas modulares y que provee de un panorama general de las etapas de planeación, desarrollo y postmortem.

El *script* de planeación tiene como entrada los requerimientos del programa así como la estimación de los recursos que se emplean en éste. El *script* de desarrollo le exige al programador los requerimientos de cada fase de desarrollo que toma en cuenta PSP (Diseño, Codificación, Compilación y Pruebas) para que al final se cuente con un programa bien probado y a prueba de errores. El *script* de Postmortem pide al programador todos los defectos que se encontraron durante la realización del proyecto y también exige el tiempo final que se utilizó durante la realización del programa, esto para que exista un historial final que ayude a tener un margen de comparación para utilizarlo posteriormente y evitar caer en los mismos errores.

Como en todo proceso de ingeniería de software, los ingenieros comienzan planeando el trabajo con el mayor detalle posible y algunas veces los ingenieros solamente cuentan con una declaración única de los requisitos del proyecto. Si es así, entonces dicha declaración debe ser la base para el plan original.

La exactitud, la estimación y el plan son influenciados en su mayoría por el conocimiento general que tienen los ingenieros respecto al trabajo que se hará. El diseño conceptual comienza cuando los ingenieros primero definen cómo el producto debe ser diseñado para que después éste sea construido, sin embargo, en esta etapa del proceso es muy temprano realizar un diseño del producto completo desde la fase del planeamiento.

Los desarrolladores tienen que estar concientes de que deben realizar un diseño conceptual que se adapte lo más fácilmente posible al proyecto. Ésta conjetura es esencial porque cada vez que los ingenieros tengan dudas respecto al proyecto, bastaría mirar este diseño para tener bien plantadas las bases de lo que se está creando. Más adelante, durante la fase del diseño, los ingenieros examinan las alternativas del diseño y al final todas esas variantes finales producen lo que se conoce como el diseño del producto completo.

La estimación del tamaño y de los recursos del producto debe ser moderada por los equipos o individuos que intervienen en el desarrollo del proyecto. Sin embargo, para los ingenieros que desarrollan software de manera individual, esta correlación tiene resultados generalmente altos.

Por lo tanto, PSP comienza a estimar los tamaños de los productos que los ingenieros desarrollan personalmente. PSP se basa en el tamaño y en los datos de la productividad de cada ingeniero y con estos datos estima el tiempo requerido para hacer el trabajo.

PSP realiza las estimaciones, tanto del tamaño del programa como de los recursos del mismo, con un método que se creó para estos fines y que tiene por nombre *PROBE* (PROxy Based Estimating) por sus siglas en inglés, traducido al español se entiende como Estimación basada en la evaluación. La evaluación se aplica a todos los objetos que se encuentran en el diseño conceptual [HUMPHREY; 95].

La metodología de *PROBE* consiste en que los ingenieros deben determinar primero los objetos que se requieren para construir el producto descrito en el diseño conceptual. Después se determina el tipo probable de los métodos que se emplean en el programa y el número de métodos que cada objeto necesita. También se puede hacer uso de referencias de datos históricos sobre los tamaños de objetos similares que se han desarrollado previamente y que al mismo tiempo utilizan el cálculo de la regresión lineal para determinar el tamaño total del producto acabado.

Puesto que el tamaño del objeto está en función del estilo de programación, el método *PROBE* demuestra a los ingenieros cómo se deben utilizar los datos sobre los programas que ellos han desarrollado personalmente para que después se generen los distintos tipos del tamaño de cada programa y este dato esté presente en todo momento para su uso personal cada vez que se desee. Una vez que se estiman los tamaños de los objetos, se utiliza la regresión lineal para estimar la cantidad total de código o (*LOC*) Lines Of Code por sus siglas en inglés, entendiéndose como líneas de código fuente, que planean desarrollar.

Para utilizar la regresión lineal, los ingenieros deben realizar una comparación de los datos históricos contra el resultado estimado del tamaño del programa actual, esta comparación se debe hacer por lo menos con tres programas anteriores.

El método *PROBE* también utiliza la regresión lineal para estimar los recursos que se emplea en el desarrollo completo.



Una vez más esta estimación se basa en el tamaño estimado del programa contra los datos reales del esfuerzo con por lo menos tres proyectos anteriores. Los datos deben demostrar una correlación razonable entre el tamaño del programa y el tiempo de desarrollo, PSP requiere que el resultado de esta correlación sea de por lo menos 0.5.

Una vez que se ha estimado el tiempo total que se empleará para el trabajo, los ingenieros deben apoyarse en sus datos históricos para estimar el tiempo necesario que cada fase del trabajo tomará. Por medio de los porcentajes que se obtienen en el campo del formato de registro de tiempo, los ingenieros tienen que asignar su tiempo de desarrollo total estimado a las fases de planeamiento, diseño, revisión de diseño, código, revisión de código, compilación, pruebas y finalmente post-mortem.

Cuando estos porcentajes han sido calculados, los ingenieros ahora cuentan con una estimación más real para el tamaño del programa, el tiempo de desarrollo total y el tiempo requerido para cada fase del desarrollo.

Una vez que los ingenieros sepan el tiempo requerido para cada fase del proceso, deben estimar el tiempo que le dedicarán al trabajo cada día o cada semana, según sus preferencias.

Con esa información, los ingenieros distribuyen su tiempo que le dedicarán a cada tarea y al mismo tiempo le asignan las horas de programación que tienen disponibles para cada programa. [ BOEHM; 81 ], [ HUMPHREY; 89 ].

Finalmente en la fase de desarrollo del producto, los ingenieros no deben hacer más que el trabajo de programación.

Mientras que este trabajo normalmente no se considera parte del proceso de planeamiento, los ingenieros deben utilizar los datos que arroje este proceso para hacer los planes futuros y que tienen mayor importancia.

Después de terminar cada trabajo, los ingenieros deben hacer un análisis post-mortem del trabajo. En el post-mortem, los desarrolladores actualizan el resumen del plan del proyecto con datos reales, también calculan cualquier dato que se calcule en el desempeño final y al final repasan qué tan bien realizaron su proyecto contra el plan definido desde el principio del proyecto.

Como paso final del planeamiento, los ingenieros deben actualizar sus bases de datos históricas con los datos del tamaño y de la productividad final alcanzada. Durante esta fase, los ingenieros también examinan cualquier posibilidad de mejora que pueda tener el proceso y de ser necesario le hacen los ajustes necesarios al proceso completo o solamente a una parte de éste.

Los desarrolladores también tienen que analizar los defectos encontrados en la fase de compilación y en la fase de pruebas para que actualicen sus listas de comprobación personales que surgen de las revisiones hechas en cada proceso para que al final esto les facilite encontrar y arreglar defectos similares que se encuentren en proyectos futuros.

Lo anteriormente mencionado depende de la habilidad individual de los ingenieros o grupal en caso de que se trate de un proyecto grande.

La clave para aprender la metodología del proceso personal de software (PSP), está en leer y analizar cada uno de los datos del trabajo que se desarrolla y lo que en realidad estos datos “dicen” acerca del desempeño personal de cada programador.

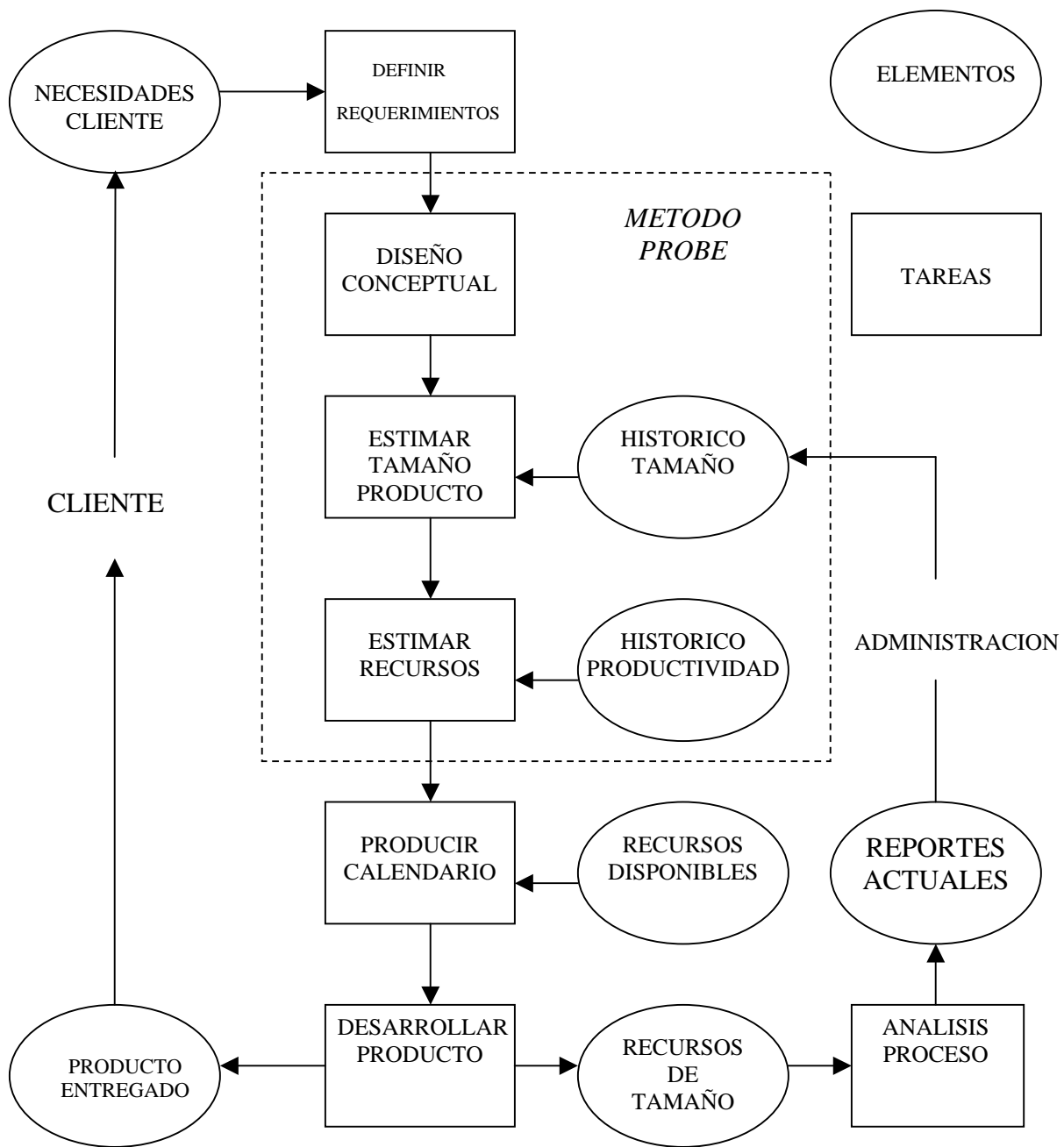
La tabla 3.3 muestra cómo se lleva a cabo la medición personal de las fases 0 y 0.1 del Proceso Personal de Software en donde su fin es el de proveer un marco que sirva para hacer la recolección de todos los datos iniciales del proceso.

<b>Fase</b>	<b>Propósito</b>	<b>¿Qué Hacer?</b>
	Criterio de Entrada	<ul style="list-style-type: none"> <li>● Descripción del Problema</li> <li>● Forma PSP0 de Resumen del Plan del Proyecto</li> <li>● Entradas del Registro de Tiempo y Defectos</li> <li>● Cronómetro (Opcional)</li> </ul>
1	Planeación	<ul style="list-style-type: none"> <li>● Producir y obtener un documento de requerimientos.</li> <li>● Estimar el tiempo requerido de desarrollo</li> <li>● Introducir los datos del plan en la forma de Resumen del Plan del proyecto.</li> <li>● Completar el Log de Registro de Tiempo</li> </ul>
2	Desarrollo	<ul style="list-style-type: none"> <li>● Diseñar el programa</li> <li>● Implementar el diseño</li> <li>● Compilar el programa, corregir y registrar todos los defectos.</li> <li>● Probar el programa, corregir y registrar todos los defectos.</li> <li>● Completar el formato del Registro de Tiempo</li> </ul>
3	Postmortem	<ul style="list-style-type: none"> <li>● Completar la forma del Resumen de Plan del Proyecto con datos actuales de tiempo, defectos y tamaño.</li> </ul>
	Criterio de Salida	<ul style="list-style-type: none"> <li>● Un programa extensamente probado</li> <li>● Forma completa de datos de Resumen del Plan de proyecto con datos actuales y estimados</li> <li>● Formatos completos de registro de Defectos y Tiempo</li> </ul>

**Tabla 3.3** "Medición personal de PSP 0 y PSP 0.1" [HUMPHREY; 95]

Posteriormente, en la figura 3.4, se muestra de manera conjunta a la tabla anterior como sería el proceso de PSP 0 y su interacción con el cliente.

También interviene el método de estimación PROBE y muestra los distintos recursos que exigen cada una de las fases desde el diseño conceptual hasta el producto final.



**Figura 3.4** "Marco de planeación de proyectos" [HUMPHREY; 95]



El contenido de este formato marca el principio del proceso PSP y por lo tanto es parte del nivel inicial de PSP junto con el PSP 0.1. El propósito de éste formato es el de registrar el tiempo empleado en cada fase del proyecto. Al mismo tiempo, estos datos son utilizados para complementar el resumen del plan del proyecto.

Como información general lo único que se necesita es registrar el tiempo total que se emplea en el proyecto; este tiempo debe estar registrado en minutos y por último se debe procurar ser lo más preciso posible.

Los campos que forman parte de PSP 0 y del éste formato son:

- Encabezado.- Los datos que se deben agregar a este campo son el nombre del desarrollador, la fecha actual, el nombre del supervisor o instructor y el número que le corresponde al programa que se está desarrollando.
- Fecha.- Corresponde a la fecha que tiene que ver con el dato que se introduce.
- Inicio.- La hora cuando se comienza a trabajar en el proyecto.
- Término.- La hora cuando se deja de trabajar en el proyecto.
- Tiempo de interrupción.- Aquí se registran todas las interrupciones que se llevaron a cabo durante el proyecto, es decir, el tiempo que no se emplea en trabajar en el proyecto.
- Tiempo delta.- Es el tiempo que se obtiene de la resta del tiempo empleado en el proyecto menos el tiempo de interrupción.
- Fase.- Aquí se introduce el nombre de la fase en la que se trabaja, puede variar, no tiene que ser la misma que en la que se está trabajando desde un principio.
- Comentarios.- Se tiene que procurar hacer todo tipo de comentarios útiles que puedan recordar ciertas circunstancias.

Todos los tiempos llevan el formato de horas y minutos (HH:MM), en caso de que sólo sean minutos los empleados entonces se utiliza el formato (:MM).

Conjuntamente a este formato se debe trabajar con el formato de registro de defectos que se muestra en el formato 3.6 y que como se menciona, es una fuente importante de datos que nos ayudará posteriormente a estimar de mejor manera posible el tiempo del proyecto y a evitar errores que se cometen con frecuencia.

Tipos de Defectos	
10 Documentación	60 Chequeo
20 Sintaxis	70 Datos
30 Construcción, Empacar	80 Función
40 Assignación	90 Sistema
50 Interfaz	100 Ambiente

**Formato del Registro de Defectos**

Estudiante					Fecha		
Instructor					Programa #		
Fecha	Número	Tipo	Encontrado	Removido	Tiempo de compostura	Defecto arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción:

Fecha	Número	Tipo	Encontrado	Removido	Tiempo de compostura	Defecto arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción:

Fecha	Número	Tipo	Encontrado	Removido	Tiempo de compostura	Defecto arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción:

Fecha	Número	Tipo	Encontrado	Removido	Tiempo de compostura	Defecto arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción:

**Formato 3.6** "Formato del registro de defectos" [HUMPHREY; 95]

El propósito general de llevar este registro de defectos reside en promover la mejora continua cada vez que se haga un proyecto. Cada fase de PSP debe de contar con un registro de defectos, ya sean revisiones, compilaciones y/o pruebas.

Los campos que intervienen en este formato son:

- Fecha.- Se introduce la fecha cuando se encuentra el defecto.
- Número.- Se introduce de manera secuencial el número de defecto encontrado en cada programa.
- Tipo.- Se refiere al tipo de defecto que viene en la tabla adjunta al formato, el tipo de defecto va desde 10 hasta 100 en intervalos de 10 en 10.
- Encontrado.- Por medio del mejor juicio se introduce el nombre de la fase cuando se encontró el defecto.
- Removido.- Se refiere al nombre de la fase cuando se removió el defecto encontrado.
- Tiempo de compostura.- Es el tiempo que tomó reparar el defecto encontrado.
- Defecto arreglado.- Este campo quiere decir si se encontró algún defecto extra mientras se reparaba el defecto detectado primero, en caso de no haber ninguno, se introduce una X.

Cada vez que se encuentra un defecto, se debe mantener un registro de estos errores ya que para proyectos futuros es importante evitar repetir o caer en los mismos errores. De esta manera se evita que resulte infructuoso utilizar PSP para mejorar constantemente porque el propósito es el de apoyar a los ingenieros a estar innovando sus métodos de desarrollo.

Finalmente se presenta el formato más importante de todos y que es el más popular de PSP.



En el formato 3.7 se muestra el resumen del plan del proyecto, este formato reúne las estimaciones y los datos reales que conforman al proyecto en toda su amplitud para que al final se realicen las comparaciones necesarias y exista un histórico de todos los proyectos realizados.

**Resumen del Plan del Proyecto**

Estudiante	_____	Fecha	_____
Programa	_____	Programa #	_____
Instructor	_____	Lenguaje	_____
<b>Tiempo en la Fase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>A la fecha</b>
			<b>A la fecha %</b>
Planeación		_____	_____
Diseño		_____	_____
Codificación		_____	_____
Compilación		_____	_____
Pruebas		_____	_____
Postmortem		_____	_____
Total	_____	_____	_____

**Defectos Encontrados**

	<b>Actual</b>	<b>A la fecha</b>	<b>A la fecha %</b>
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Compilación	_____	_____	_____
Pruebas	_____	_____	_____
Total Desarrollo	_____	_____	_____

**Defectos Removidos**

	<b>Actual</b>	<b>A la fecha</b>	<b>A la fecha %</b>
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Compilación	_____	_____	_____
Pruebas	_____	_____	_____
Total Desarrollo	_____	_____	_____
Después de Desarrollo	_____	_____	_____

**Formato 3.7 "Resumen del plan del proyecto" [HUMPHREY; 95]**

Como se puede apreciar en este formato, existen tres campos diferentes. Dos de estos campos tienen que ver con los defectos encontrados y removidos en cada fase. En la parte superior del formato se aprecian los campos que el desarrollador empleará para registrar los tiempos que emplea en cada fase del proyecto.

Este formato es esencial ya que es un respaldo para cada proyecto que se desarrolla. En él se pueden encontrar los datos que serán útiles para el siguiente proyecto parecido que se desarrolle. Es importante que los datos se escriban con claridad y con precisión para que cada fase de desarrollo sirva para tener un margen de comparación con proyectos futuros.

Los campos que el programador tiene que llenar en este formato son:

#### Área de Tiempo en la fase.

- Plan.- Es el tiempo estimado que se empleará para desarrollar el proyecto completo.
- Actual.- Es el tiempo real, en minutos, que se emplea en cada una de las fases de desarrollo.
- A la fecha.- La suma del tiempo actual con el tiempo a la fecha del último programa desarrollado.
- A la fecha %.- Indica el porcentaje del tiempo a la fecha que se emplea en cada fase de desarrollo.

#### Área de Defectos encontrados.

- Actual.- Número de defectos reales encontrados en cada fase de desarrollo.
- A la fecha.- Suma de los valores de los campos Actual (Defectos encontrados) con el campo de A la fecha del último proyecto programado.
- A la fecha %.- Porcentaje de defectos encontrados a la fecha en cada fase de desarrollo.

### Área de Defectos removidos.

- Actual.- Indica el número de defectos removidos en cada etapa.
- A la fecha.- Es la suma del valor que se encuentra en el campo de Actual (Defectos removidos) con el valor A la fecha del último programa desarrollado.
- Ala fecha %.- Porcentaje de defectos removidos del campo A la fecha y que se aplica para cada fase de desarrollo.

Cuando el proyecto se termine es importante registrar cualquier defecto que se haya detectado tardíamente. Esto debe ser realizado cuando el programa esté en uso, re-uso o si el programa sufre alguna modificación.

La diferencia entre PSP 0 y PSP 0.1 es mínima, solamente se agregan medidas que guían al programador a medir y estimar el tamaño de los programas que desarrolla.

### **3.4 CONCLUSIONES DE PSP 0 y 0.1**

Es importante recordar que los objetivos de PSP 0 son incorporar medidas básicas en el proceso de desarrollo de software. También se debe recordar que esta parte de PSP exige que haya un mínimo de cambios en la forma de programar del individuo para que se eficiencia sea de un nivel mayor.

PSP 0 también se propone demostrar que se debe de emplear un proceso bien definido en cada programa que se desarrolle, por muy pequeño que éste sea. Así se incita al programador a tener unas bases bien definidas cada vez que se desarrolle software y los resultados finales sean de alta calidad.

El objetivo final de PSP 0 es que no se deseche por completo el proceso que el programador utiliza hasta antes de emplear PSP, sino que sirva de un marco inicial para comenzar a utilizar PSP por completo.

El objetivo de PSP 0.1 es similar a los objetivos de PSP 0, sin embargo se agrega el objetivo de medir y estimar el tamaño de los programas que se desarrollen, para esto se describe anteriormente los estándares que PSP emplea para realizar estas mediciones.