Capitulo 11.- Scaling up the personal software process

Contenido

[11.6 PSP3 1](#_Toc454048843)

[11.6.1 requirements and planning 1](#_Toc454048844)

[11.6.2 High-Level design 2](#_Toc454048845)

[11.6.3 Cyclic development 2](#_Toc454048846)

[11.6.4 TEST development and review 2](#_Toc454048847)

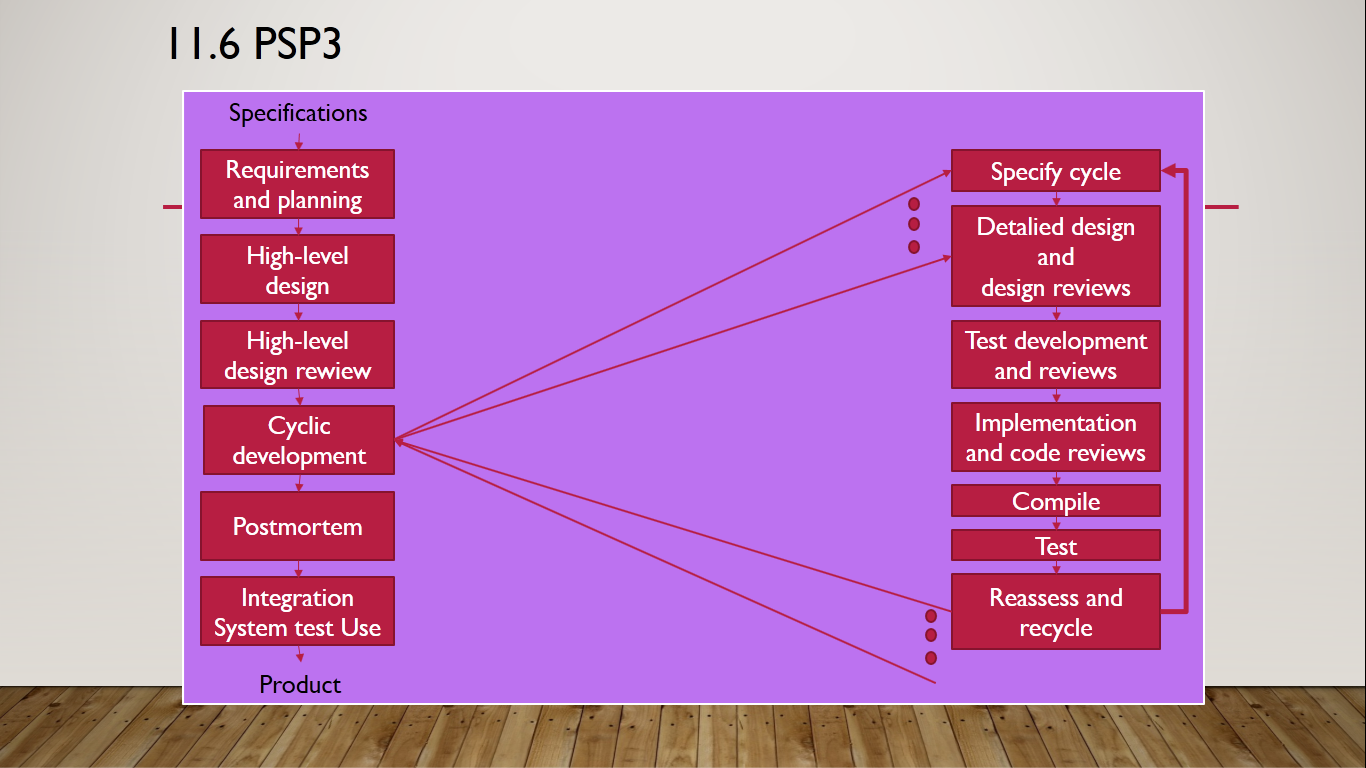
[11.6.5 reassess and recycle 3](#_Toc454048848)

[11.7 Summary 3](#_Toc454048849)

# 11.6 PSP3

El principal papel de PSP3 es ser un ejemplo de la fundación del proceso personal para el desarrollo de software a gran escala.

PSP es capaz de manejar el crecimiento de la complejidad de un producto y debe relacionarlo con un proceso largo de desarrollo en equipo.



## 11.6.1 requirements and planning

Comienza construyendo:

Diseño conceptual de todo el sistema.

* Estimaciones de tamaño.
* Planeación del trabajo de desarrollo.

## High-Level design

Identificar la división natural del producto y divise la estrategia cíclica. Si muchas partes son largas:

* Adoptar un enfoque para esas partes.
* Una buena idea para purgar es mantener los ciclos de 100 a 300 líneas de código nuevo o modificado.
* Estos ciclos crecen más de lo esperado, pero hay que tratar de mantenerlos cortos

Combine varias funciones en un solo ciclo si son fáciles de entender y simples.

Después de la revisión de diseño de alto nivel, sigue el ciclo de desarrollo.

## 11.6.3 Cyclic development

Establecer las especificación para el ciclo concurrente.

* Producirlas muy temprano en el ciclo puede causar pérdidas de tiempo porque a menudo afecta la subsecuencia de los ciclos de desarrollo.

Es esencial para el PSP2.1🡪 produce parte del producto. Durante el ciclo las revisiones y pruebas pueden estar completas. Este ciclo es la base para la siguiente fase, porque muchos defectos que se quedan en esta fase, pueden causar problemas después.

El ciclo de desarrollo es efectivo por que crea y prueba nuevas funciones en una base sólida. La escalabilidad es preservada, no obstante, un desarrollo largo incremental es autónomo y esencialmente libre de defectos.

## 11.6.4 TEST development and review

El ciclo de desarrollo también incluye una fase de pruebas.

El desarrollo de pruebas a menudo revela problemas de diseño, por lo que es deseable desarrollar pruebas antes de que escribas código. [Beizer]

Las pruebas revelan problemas de diseño, como:

* Manejo de excepciones
* Respuesta a los errores
* Condiciones fuera de limites

Otras cosas mas como:

* Datos de prueba
* Pruebas para módulos especiales
* Escenarios de uso
* Pruebas de facilidad de soporte.

Una ventaja de planear las pruebas, es que el desarrollo del producto es enfocado a las pruebas.

Las pruebas son las primeras que hacen que cambie el diseño, después de eso, los diseños cambian por las revisiones de diseño que hacen rehacer la planeación de pruebas.

En el ciclo de pruebas, la primera prueba normalmente es borrón y cuenta nueva.

Cada subsecuencia del ciclo, añade funcionabilidad que se integra al producto probado. Mientras de combinen unidad e integración en las pruebas, puede parecer extraño, pero es una estrategia de PSP

## 11.6.5 reassess and recycle

Reexaminar el trabajo, determinar el status y reevaluar el plan.

Puedes solamente considerar el reporte del final del ciclo.

Si indica desviación de tu plan original, ajusta el plan y alerta al administrador del proyecto.

# 11.7 Summary

El tamaño de los proyectos de software incremento en los últimos 30 años

Los proyectos que se desarrollen en el futuro, van a ser más largos aún.

El proceso funciona bien para proyectos pequeños, pero para más grandes de 5 a 10 veces es recomendable cambiar de proceso.

La fase de diseño a gran escala del software, comienza con el diseño de alto novel el cual hace que se divida en componentes.

Los componentes se separan en desarrollo e integración

Los componentes se pueden desarrollar con métodos de pequeña escala de tamaño.

El tamaño del software para un proyecto grande se puede dividir por rangos que son respecto a las habilidades:

* Etapa 0.- bloques pequeños de trabajo.
* Etapa 1.- desarrollar programas por módulos.
* Etapa 2.- definir componentes por módulos (enfocándose en los detalles de la etapa 1).
* Etapa 3.- abstracción de los componentes.
* Etapa 4.- subsistemas autónomos a cargo de manejar responsabilidades del sistema.

PSP3 es un ejemplo de proceso para proyectos largos. La estrategia es un proceso cíclico.

Un buen fundamento de estrategia es construir estructuras naturales para la planeación del producto.

* Ciclos para desarrollo e implementación.
* Cada ciclo tiene su prueba y su integración.

Al final se integra todo el sistema y se prueba.