

W_Watch:

Método *White_Watch* para el desarrollo de Proyectos Pequeños de Software

(Prof. J. Barrios y J. Montilva - Versión 1.2 – enero 2010)

1. Introducción

El método *White_Watch* es la versión *liviana* del método *Watch*. *W_Watch* es un marco metodológico que describe, el conjunto estructurado de actividades necesarias para desarrollar un producto de software pequeño, de baja complejidad y con documentación técnica precisa.

En esta versión se trata de disminuir la elaboración detallada de documentos y/o especificaciones de apoyo parcial al proceso de desarrollo, permitiendo, al equipo de desarrollo pequeño (1 o 2 personas), dedicar más tiempo a las actividades de implementación e implantación de versiones operativas y evolutivas del producto. El rol de líder de proyecto es ejecutado en paralelo, y sin sobrecarga, por un miembro del equipo mientras ejecuta otros roles técnicos. Es por ello que, las actividades gerenciales de control de calidad y de configuración, las cuales son indispensables en todo proyecto de software, se limitan a prescribir las actividades básicas de control de cambios, de validación y de verificación de especificaciones técnicas y de productos parciales y finales.

2. Modelo de Procesos del Método

El modelo de procesos esta organizado en dos grupos de procesos complementarios:

- los procesos gerenciales que incluyen los procesos de soporte, y
- los procesos técnicos de desarrollo del producto de software

El modelo de procesos del método *W_Watch* inspirado en la metáfora del reloj de pulsera (*Watch* en Inglés), organiza los procesos técnicos, en forma circular, en las posiciones del dial de un reloj y ubica los procesos gerenciales en el centro – motor de control, de manera que éstos puedan controlar la ejecución de los procesos técnicos.

Esta manera de estructurar el marco metodológico permite que la ejecución de los procesos de desarrollo sea **cíclica, iterativa y controlada**. Los procesos gerenciales determinan, supervisan y controlan los ciclos, las iteraciones y la entrega de los productos parciales y finales del proyecto.

Cada ciclo de procesos técnicos produce una nueva versión del sistema (**enfoque progresivo o evolutivo**) o un nuevo subsistema del sistema **en desarrollo**, si se ha seleccionado el enfoque de **desarrollo incremental**.

En cada ciclo se puede **iterar entre los procesos técnicos** a fin de corregir errores, introducir nuevos requisitos o, simplemente, mejorar el producto en desarrollo. Son los procesos gerenciales quienes determinan el alcance de cada iteración y cuántas iteraciones se realizan por ciclo.

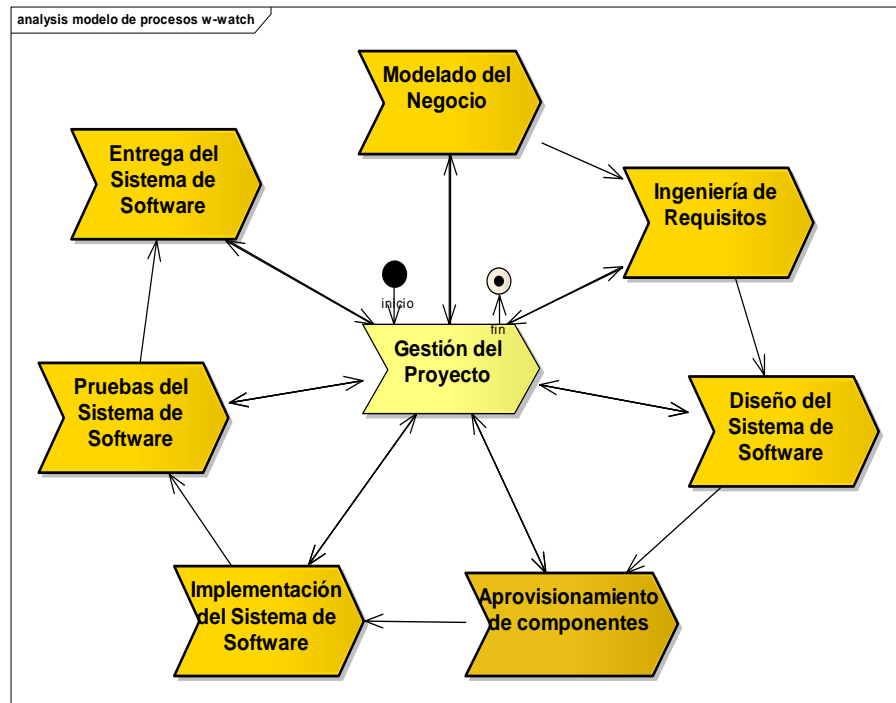


Figura 1. Modelo de Procesos del Método *W_Watch*

Los **procesos gerenciales** describen las actividades que el líder del proyecto debe realizar para:

- Planificar, organizar y controlar el proceso de desarrollo del proyecto
- Asegurar la calidad del sistema mediante validaciones y verificaciones
- Gestionar los cambios en las especificaciones del producto

Los **procesos técnicos** son los procesos que prescriben lo que debe hacer el grupo de desarrollo para elaborar un producto de software pequeño y poco complejo. Contemplan actividades relacionadas con el Modelado de Negocios o dominio, la Ingeniería de requisitos, el Diseño de software, el Aprovisionamiento de Componentes, la Implementación, las Pruebas y la Entrega de producto parcial o final.

Se establecen un conjunto de **supuestos** que soportan el conjunto de actividades prescritas en el modelo de procesos de la figura 1. Estos son:

- 1) El personal de desarrollo en sus diferentes roles: líder de proyecto, analista, diseñador y programador, cuenta con las habilidades, experiencia y conocimientos relacionados con el uso de los lenguajes y las herramientas de apoyo que se van a utilizar.
- 2) El enfoque de desarrollo evolutivo o incremental es la base para la planificación del número de iteraciones y ciclos que se realizarán. Se parte del principio que la primera versión del producto es ya operativa y que cada nueva versión es el resultado de un refinamiento de la versión previa. Como refinamiento se puede entender: corrección de deficiencias y de errores de la versión previa (o incremento), inclusión de nuevos requisitos (técnicos o del usuario) y/o mejoramiento del desempeño del producto.
- 3) El proceso de desarrollo se fundamenta en la máxima reutilización de componentes o módulos de software como medio para acortar el tiempo de entrega de versiones parciales de producto y por consiguiente, el tiempo total de desarrollo del software.
- 4) Se espera que se empleen herramientas de apoyo automatizadas, basadas en la notación UML, para la elaboración y mantenimiento de la documentación técnica del proyecto. Estas herramientas deberían facilitar las actividades de diagramación, especificación, modificación, traza y seguimiento de cambios y/o modificaciones en las especificaciones técnicas del producto elaborado.

Cada uno de los procesos del modelo de procesos se estructura en pasos y actividades, buscando disminuir la complejidad e indicar el cómo se realizan. Cada proceso es descrito mediante una tabla que asocia pasos (subprocesos) y actividades que indican de manera detallada el conjunto de acciones a ejecutar para llevar el proceso prescrito, y por consiguiente, cada paso del proceso. Ver figura 2.

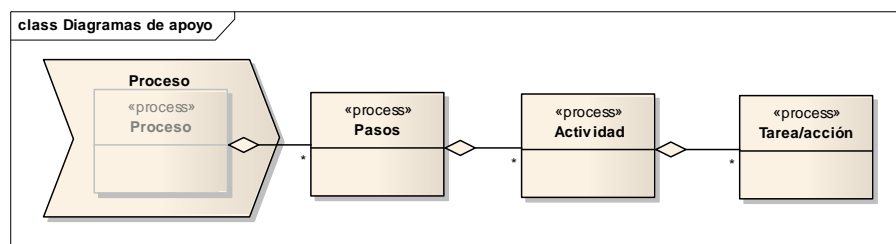


Figura 2. Estructura del Modelo de Procesos del Método *W_Watch*

Como toda **guía metodológica** el modelo de procesos del *W_Watch* debe ser **adaptado**, por el líder del proyecto, según las particularidades de cada proyecto de desarrollo.

Entre los **factores** a considerar para la adaptación se tienen: las **características** propias de cada **producto** y de los **ambientes** de desarrollo y de operación; se consideran además, los **recursos** utilizables tanto a nivel de **personal** como de **HW** y **SW** y las habilidades y destrezas requeridas por los miembros que conformarán el equipo de desarrollo.

En la figura 3 se muestra el diagrama de flujo de trabajo del Modelo de Procesos del método *W_Watch*.

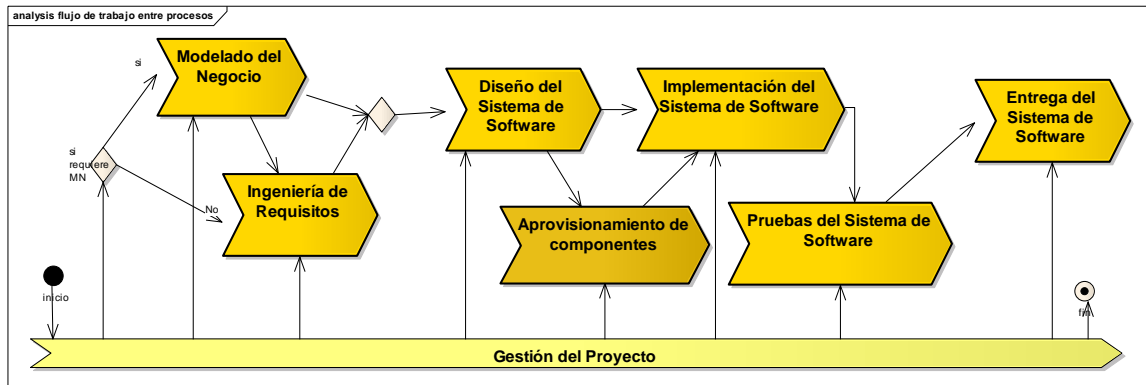


Figura 3. Flujo de trabajo del Modelo de Procesos del Método *W_Watch*

Por ejemplo, considerando el modelo de la figura 3, en los proyectos que no desarrollan aplicaciones empresariales o cuyo producto de SW es empotrado, no se requiere realizar el modelado del sistema de negocios; así, el proceso de desarrollo se iniciaría directamente con el proceso de Ingeniería de Requisitos.

A continuación se define el conjunto de pasos y actividades para cada uno de los procesos del modelo presentado en la figura 3.

3. Proceso Gerencial: Gestión del Proyecto

Pasos	Actividades	Técnicas y Notaciones	Productos
<ul style="list-style-type: none"> Planificación del Proyecto Organización del grupo de desarrollo Control del proyecto Verificación y Validación (V&V) Resolución de Riesgos Gestionar cambios en los requisitos del SW Control de Documentación Control de la configuración del software 	<ul style="list-style-type: none"> Establecer la visión del producto Establecer disponibilidad de recursos Elaborar plan del proyecto Definir equipo de desarrollo y responsabilidades Definir lista de riesgos y contingencias Organizar reuniones de avance del proyecto Actualizar el plan Realizar las revisiones técnicas Analizar viabilidad de cambios en requisitos Actualizar documentación técnica Elaborar documentos de la aplicación 	<ul style="list-style-type: none"> PERT/CPM Estructuras de grupos Estimación de costos Técnicas de V & V Técnicas de gestión de riesgos Inspección de diseño y código Recorridos estructurados Técnicas de elaboración de documentos técnicos Matrices y listas de rastreo de requisitos Técnicas de SCM 	<ul style="list-style-type: none"> Visión del producto Plan del Proyecto Lista de chequeo de riesgos Informe de V & V Documentos del proyecto – informes Documentos de la aplicación Especificaciones actualizadas

4. Procesos de Desarrollo - Proceso: Modelado de Negocios

Pasos	Actividades	Técnicas y Notaciones	Productos
1. Modelado de Objetivos del Sistema de Negocios - SN- contexto o dominio de la aplicación	<ul style="list-style-type: none"> Establecer el alcance del sistema de negocios Identificar los subsistemas del SN Definir objetivos de SN Elaborar la jerarquía de objetivos (si necesario) 	<ul style="list-style-type: none"> Revisión de los manuales de organización Entrevistas con los involucrados en el SN Revisión de los manuales de organización Entrevistas con los involucrados en el SN Modelado de objetivos 	<ul style="list-style-type: none"> Documento de definición del SN y de su alcance Diagrama de Objetivos del SN
2. Modelado de los Procesos de Negocio del SN	<ul style="list-style-type: none"> Modelar la cadena de valor Modelar los procesos vitales (fundamentales) Modelar los procesos de soporte (de apoyo) Modelar las actividades de cada proceso de la cadena 	<ul style="list-style-type: none"> Observación y Entrevista con los expertos del SN Revisión de documentación técnica Modelado de Cadenas de Valor Modelado de Procesos en UML <i>Business</i> Modelado de Actividades en UML 	<ul style="list-style-type: none"> Cadena de Valor del SN Diagramas de Procesos del SN: <ul style="list-style-type: none"> Jerarquía de procesos Descripción de procesos Diagramas de actividades: <ul style="list-style-type: none"> Diagramas de actividades x proceso
3. Modelado de actores/unidades organizacionales	<ul style="list-style-type: none"> Identificar actores/unidades (pueden ser otros sistemas) Definir roles de los actores en cada proceso Completar diagramas de actividades Elaborar la matriz actores/procesos 	<ul style="list-style-type: none"> Modelado de actividades con actores 	<ul style="list-style-type: none"> Diagrama de Actividades por actor Descripción de Actores y sus Roles Diagramas de actividades – colaboración- entre actores - <i>swimlines</i> Matriz Actores-Procesos
4. Modelado de los Objetos de Negocio del SN	<ul style="list-style-type: none"> Identificar los objetos de negocio y sus tipos x proceso Definir las relaciones entre tipos de objetos Elaborar el modelo preliminar de objetos 	<ul style="list-style-type: none"> Modelado de Clases en UML - representación de los conceptos manejados por el negocio 	<ul style="list-style-type: none"> Diagramas de clases en UML
5. Identificación de las Reglas de Negocio	<ul style="list-style-type: none"> Identificar las reglas de negocio Analizar y clasificar las reglas de negocio (alto y bajo nivel) Describir las reglas de negocios de bajo nivel 	<ul style="list-style-type: none"> Consultas a usuarios y expertos Búsqueda de documentación Modelado de reglas de negocio 	<ul style="list-style-type: none"> Lista reglas de negocio del SN <ul style="list-style-type: none"> Lista de las reglas de Alto nivel Descripción de las reglas de Bajo nivel
6. Modelado de Eventos	<ul style="list-style-type: none"> Identificar eventos Asociar eventos con procesos Modelar el flujo de trabajo asociado a cada evento Elaborar la matriz de eventos- procesos 	<ul style="list-style-type: none"> Modelado de eventos en UML <i>Business</i> 	<ul style="list-style-type: none"> Diagrama de eventos en UML <i>Business</i> Diagramas de flujo entre procesos Matriz Eventos-Procesos
7. Integrar los modelos	<ul style="list-style-type: none"> Verificar coherencia entre modelos Ensamblar el documento de modelado 	<ul style="list-style-type: none"> Matriz de relación objetos/ procesos Técnicas de documentación 	<ul style="list-style-type: none"> Modelo de Negocios del sistema de negocios - MN <ul style="list-style-type: none"> Todos los modelos integrados

5. Procesos de Desarrollo – Proceso: Ingeniería de Requisitos

Actividades	Tareas	Técnicas y Notaciones	Productos
1.Descubrimiento de Requisitos	<ul style="list-style-type: none"> Identificar y analizar los problemas de información que tiene el Sistema de Negocios (o el contexto - dominio) Determinar los objetivos del sistema de SW Identificar y clasificar los involucrados (<i>stakeholders</i>) y usuarios (internos y externos) – si MN a partir de modelo de actores Recolectar los requisitos que tienen los involucrados y usuarios - si MN a partir de modelo de actividades/actores Identificar requisitos de información a partir de los diagramas de procesos y de actividades (o del contexto) 	<ul style="list-style-type: none"> Entrevista Documentación relacionada con el dominio – o el MN Observación de las actividades que realizan los usuarios Plantilla de definición de requisitos Volére [VOL04] Reuniones con usuarios 	<ul style="list-style-type: none"> Listado de requisitos del cliente C Listado de requisitos técnicos Planillas Volere de requisitos
2.Análisis de Requisitos	<ul style="list-style-type: none"> Clasificar los requisitos F y NF Chequear requisitos <ul style="list-style-type: none"> Comprobar necesidad, prioridad, consistencia, completitud y factibilidad Negociar requisitos <ul style="list-style-type: none"> Discutir, priorizar y acordar requisitos con el cliente y los usuarios de la aplicación Elaborar cuadro detallado de los requisitos clasificados indicando sus prioridades y su fuente Validar requisitos con el cliente y usuarios 	<ul style="list-style-type: none"> Técnicas de negociación Matriz de interacción entre requisitos 	<ul style="list-style-type: none"> Documento de Definición de Requisitos (DDR) validado <i>Incluye Matriz de interacción entre requisitos</i>
3.Especificación de Requisitos	<ul style="list-style-type: none"> Elaborar los diagramas de casos de uso Elaborar el diagrama preliminar de clases de objetos para cada caso de uso <ul style="list-style-type: none"> Integrar los diagramas de clases (<i>verificando con modelo de objetos del negocio si MN</i>) Elaborar diagramas de transición de estados (<i>si requerido</i>) Elaborar diagramas de secuencia entre clases de objetos del negocio – (<i>si requerido</i>) Integrar diagramas en documento de Especificación de Requisitos (DER) Realizar la revisión técnica del DER con el cliente, usuarios especializados y diseñadores 	<ul style="list-style-type: none"> Refinamiento de modelo de objetos del negocio (MN) Modelado de sistemas en UML: <ul style="list-style-type: none"> Diagramas de casos de uso Diagramas de clases Diagramas de estado Diagramas de secuencias 	<ul style="list-style-type: none"> Documento de Especificación de Requisitos (DER) validado <i>Incluye los modelos de especificación de requisitos</i>

6. Procesos de Desarrollo - Proceso: Diseño de software

Actividades	Tareas	Técnicas y Notaciones	Productos
1.Definición de la estructura inicial de la aplicación	<ul style="list-style-type: none"> Determinar requisitos a implementar a partir del DER y relacionarlos con la arquitectura del sistema de SW Establecer las metas de calidad de la arquitectura del sistema de SW Dividir el sistema en subsistemas (<i>si necesario</i>) 	<ul style="list-style-type: none"> Modelos de calidad del software [BCK98] Estilos arquitectónicos [BCK98] 	<ul style="list-style-type: none"> Listado descriptivo de las metas de diseño Estructura de la aplicación Arquitectura de

	<ul style="list-style-type: none"> ▪ Agrupar funcionalidad según subsistemas <ul style="list-style-type: none"> ○ Refinar casos de uso ○ Refinar diagrama preliminar de clases ○ Elaborar diagramas de secuencia ▪ Representar subsistemas en arquitectura 3 capas 	<ul style="list-style-type: none"> ▪ 3 capas 	la aplicación
2. Diseño de la Interfaz Usuario/Sistema	<ul style="list-style-type: none"> • Establecer el perfil de los usuarios • Establecer perfil de tareas (<i>a partir de los casos de uso</i>) • Establecer las características estéticas que deberá tener la interfaz gráfica de la aplicación <ul style="list-style-type: none"> ○ Establecer los fondos, colores, tipos de fuentes, etc. • Diseñar la estructura general de la interfaz U/S: <ul style="list-style-type: none"> ○ Elaborar el diagrama jerárquico de pantallas del sistema ○ Definir las características que deben tener los ítems que componen las pantallas de la interfaz: menús, ventanas, íconos, enlaces, cuadros, cajas, etc. • Realizar las revisiones técnicas de la interfaz U/S según lo expresado en documentos DDR y DER 	<ul style="list-style-type: none"> • Técnicas de Utilidad (<i>usability</i>) • Técnicas y estrategias de diseño de interfaces GUI 	<ul style="list-style-type: none"> • Diseño de pantallas • Diagrama jerárquico de pantallas
3. Diseño la BD (<i>si requerido</i>)	<p>Realizar el diseño Conceptual</p> <ul style="list-style-type: none"> ▪ Refinar modelo de clases de objetos de negocio ▪ Definir los atributos de cada clase de objetos de negocio ▪ Verificar el modelo con los requisitos ▪ Validar modelo con los usuarios <p>Realizar el diseño implementable</p> <ul style="list-style-type: none"> • Convertir el modelo conceptual de la BD en un esquema relacional equivalente • Verificar el esquema implementable con los requisitos relacionados <p>Realizar el diseño Físico</p> <ul style="list-style-type: none"> • Establecer los índices de las tablas del diseño implementable • Definir los derechos de acceso para cada tipo de usuario (<i>usuario final, programador, ABD</i>) • Definir las reglas de integridad de la BD 	<ul style="list-style-type: none"> ▪ Diagramas de clase en UML ▪ Modelado de Bases de Datos OO ▪ Procedimiento de conversión de diagramas de clase a esquemas de relación ▪ Modelado BD <i>Relacionales</i> ▪ Revisión técnica (Inspección de Diseño) ▪ Procedimientos de diseño físico de BD <i>relacionales</i> 	<ul style="list-style-type: none"> ▪ Modelo conceptual integrado de la BD ▪ Esquema relacional o equivalente de la BD (<i>integrado y verificado</i>) ▪ Esquema físico de la BD
4. Diseño de componentes o módulos de SW	<ul style="list-style-type: none"> • Identificar Componentes o módulos • Identificar elementos funcionales – propios de la aplicación – implementación de casos de uso • Identificar elementos de interfaz U/S • Identificar elementos de acceso y manipulación de datos persistentes • Ubicar elementos en la arquitectura inicial predefinida – 	<ul style="list-style-type: none"> • UML <i>Components</i> [CHD01] 	<ul style="list-style-type: none"> • Definición de componentes o módulos • Especificación de Interfaces • Arquitectura de Componentes o módulos del sistema

	<ul style="list-style-type: none"> • Describir la interacción de componentes o módulos • Determinar las interfaces de cada componente 		
5. Especificación del diseño	<ul style="list-style-type: none"> • Integrar diagramas de subsistemas, interfaz, arquitectura y componentes o módulos y BD en Documento de Diseño • Definir los procedimientos de respaldo, recuperación y seguridad de la BD • Realizar las revisiones técnicas de validación con el cliente y los usuarios 	<ul style="list-style-type: none"> • UML <i>Components</i> [CHD01] 	<ul style="list-style-type: none"> • Documento de Diseño (DD) integrado y validado • Procedimientos de administración de la BD

7. Proceso de Desarrollo - Proceso: Aprovisionamiento de componentes

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Instalación de la plataforma de desarrollo	<ul style="list-style-type: none"> • Seleccionar, adquirir y/o preparar la plataforma o infraestructura de software requerida para desarrollar el sistema • Instalar la plataforma de desarrollo (servidores web, de aplicaciones, SMBD) 	<ul style="list-style-type: none"> • Manuales y procedimientos de instalación de software 	<ul style="list-style-type: none"> • Plataforma de desarrollo instalada
2. Adquisición de Componentes – <i>si requeridos</i>	<ul style="list-style-type: none"> • Buscar componentes que puedan ser adquiridos de terceros (abiertos o propietarios) o en librerías propias de la organización • Adquirir componentes 	<ul style="list-style-type: none"> • Búsqueda de componentes abiertos o comerciales (P. ej., COTS) 	<ul style="list-style-type: none"> • Componentes adquiridos
3. Adaptación de Componentes <i>Si requerido</i>	<ul style="list-style-type: none"> • Buscar componentes en repositorios locales (internos) o de terceros (externos) • Adaptar los componentes mediante su modificación interna o el uso de envoltorios (<i>wrappers</i>) 	<ul style="list-style-type: none"> • Reutilización caja blanca • Envoltorios (<i>Wrapping</i>) 	<ul style="list-style-type: none"> • Componentes adaptados
4. Desarrollo de Componentes o módulos	<ul style="list-style-type: none"> • Desarrollar aquellos componentes o módulos que no pudieron ser localizados o adquiridos. Partiendo de la especificación de cada componente: <ul style="list-style-type: none"> ○ Elaborar el diseño detallado de cada operación de cada interfaz del componente ○ Codificar las operaciones del componente ○ Elaborar interface del componente ○ Desplegar el componente en la plataforma seleccionada ○ Diseñar y ejecutar las pruebas de unidad del componente 	<ul style="list-style-type: none"> • Diseño de algoritmos • Refinamiento paso-a-paso • Pseudo-código • Estándares de codificación • Estrategias de pruebas de unidad 	<ul style="list-style-type: none"> • Componentes desarrollados
5. Diseño y ejecución de pruebas de componentes o módulos	<ul style="list-style-type: none"> • Realizar pruebas funcionales para cada uno de los componentes o módulos adquiridos, adaptados y/o desarrollados <ul style="list-style-type: none"> ○ Preparar los datos y mecanismos de prueba ○ Preparar el ambiente de pruebas ○ Ejecutar las pruebas funcionales de cada componente • Depurar los errores encontrados durante las pruebas funcionales de cada componente o módulo 	<ul style="list-style-type: none"> • Estrategias de pruebas caja negra: <ul style="list-style-type: none"> ○ Clases de equivalencia ○ Análisis de valores límites • Técnicas de pruebas de hilos (<i>thread testing</i>) • Depuración de errores 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Componentes y/o módulos probados y depurados

8. Procesos de Desarrollo - Proceso: Ensamblaje del Sistema de software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Construcción de la Interfaz U/S	<ul style="list-style-type: none"> • Ensamblar la capa de presentación con los componentes o elementos de SW de la interfaz U/S <ul style="list-style-type: none"> ○ Codificar e integrar los componentes o elementos del lado del cliente • Diseñar y Ejecutar las Pruebas de la Interfaz U/S <ul style="list-style-type: none"> ○ Determinar los aspectos de la interfaz U/S que deben probarse ○ Realizar prueba de la interfaz U/S <ul style="list-style-type: none"> .Preparar los datos y mecanismos de prueba .Preparar el ambiente de pruebas .Ejecutar las pruebas de la interfaz U/S • Depurar los errores encontrados 	<ul style="list-style-type: none"> • Técnicas de construcción de software • Técnicas y estrategias de pruebas de interfaces gráficas • Depuración de errores 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Interfaz U/S probada
2. Ensamblaje de Componentes o módulos de la aplicación – capa de negocios	<ul style="list-style-type: none"> • Ensamblar la capa de lógica de negocios – subsistemas - componentes o módulos de la aplicación que la integran • Ejecutar las Pruebas de Integración <ul style="list-style-type: none"> ○ Definir los criterios y técnicas de pruebas de integración de componentes o módulos ○ Realizar casos de prueba de integración de componentes o módulos <ul style="list-style-type: none"> .Preparar los datos y mecanismos de prueba .Preparar el ambiente de pruebas .Ejecutar las pruebas de integración de componentes o módulos • Depurar los errores encontrados 	<ul style="list-style-type: none"> • Despliegue de componentes o módulos en servidores de aplicaciones • Técnicas y estrategias de pruebas de integración de componentes o módulos • Depuración de errores 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Capa de Lógica de negocios de la aplicación probada

Actividades	Tareas	Técnicas y Notaciones	Productos
3.Construcción de la BD	<ul style="list-style-type: none"> • Crear la base de datos usando los esquemas implementables diseñados en el proceso anterior y según el SMBD seleccionado • Diseñar y Ejecutar las Pruebas de la BD <ul style="list-style-type: none"> ○ Realizar casos de prueba de la BD .Definir los aspectos de la BD que deben probarse .Preparar los datos y mecanismos de prueba .Preparar el ambiente de pruebas .Ejecutar las pruebas de la BD • Depurar los errores encontrados 	<ul style="list-style-type: none"> • Creación de BD relacional • Técnicas y estrategias de pruebas de bases de datos • Depuración de errores 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Base de datos probada
4.Pruebas de la Integración de las Capas de la arquitectura	<ul style="list-style-type: none"> • Realizar casos de prueba de integración de capas <ul style="list-style-type: none"> ○ Definir los criterios y técnicas de pruebas de integración de las tres capas de la aplicación ○ Preparar los datos y mecanismos de prueba ○ Preparar el ambiente de pruebas ○ Ejecutar las pruebas de integración de capas • Depurar los errores encontrados 	<ul style="list-style-type: none"> • Técnicas y estrategias de pruebas de aplicaciones distribuidas • Depuración de errores 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Aplicación integrada y probada (desplegada en la plataforma de desarrollo)

9. Procesos de Desarrollo - Proceso: Pruebas del Sistema de Software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Realización de las Pruebas del Sistema	<ul style="list-style-type: none"> ▪ Definir mecanismos de pruebas ▪ Preparar casos de prueba ▪ Realizar las pruebas funcionales del sistema (aplicación integrada) <ul style="list-style-type: none"> ○ Ejecutar las pruebas funcionales ▪ Realizar las pruebas no funcionales del sistema <ul style="list-style-type: none"> ○ Ejecutar las pruebas no funcionales ▪ Realizar las pruebas de aceptación <ul style="list-style-type: none"> ○ Ejecutar las pruebas de aceptación <p>Reportar los errores encontrados en las pruebas</p>	<ul style="list-style-type: none"> • Estrategias de pruebas funcionales y no funcionales 	<ul style="list-style-type: none"> • Mecanismos de pruebas • Casos de pruebas • Especificaciones de casos de prueba • Informe de incidentes de prueba
2. Corrección de errores	<ul style="list-style-type: none"> ▪ Corregir los errores detectados en las pruebas funcionales y no-funcionales ▪ Realizar pruebas de regresión para asegurar que las correcciones no introducen nuevos errores 	<ul style="list-style-type: none"> • Depuración (<i>debugging</i>) 	<ul style="list-style-type: none"> • Aplicación probada y depurada

10. Procesos de Desarrollo - Proceso: Entrega del Sistema de Software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Instalación de la Aplicación	<p>Planificar la instalación</p> <ul style="list-style-type: none"> ▪ Definir las estrategias de migración a la nueva aplicación ▪ Determinar actividades de la instalación de la aplicación ▪ Estimar costos, tiempos y recursos requeridos ▪ Instalar la plataforma de Hardware/Software requerida para operar el sistema (<i>si no está instalada</i>) ▪ Desplegar la aplicación en los diferentes servidores de la plataforma de operación; carga inicial de datos (<i>si se requiere</i>) <ul style="list-style-type: none"> ▪ Preparar los datos de carga inicial de la BD ▪ Actualizar la BD 	<ul style="list-style-type: none"> • Técnicas y herramientas de planificación • Instructivos de despliegue de aplicaciones proporcionados por el fabricante • Técnicas de migración de datos 	<ul style="list-style-type: none"> • Plan de Instalación • Plataforma de Operación (H/S) instalada • Aplicación instalada • BD actualizada
2. Diseño y Ejecución de Pruebas de Instalación	<ul style="list-style-type: none"> • Definir los aspectos de la instalación que deben probarse • Diseñar los procedimientos y casos de prueba de instalación • Preparar los datos y mecanismos de prueba • Ejecutar las pruebas de instalación • Corregir los errores encontrados • Identificar cambios y ajustes finales • Entregar el sistema al cliente 	<ul style="list-style-type: none"> • Técnicas y estrategias de pruebas de instalación • Depuración de errores • Control de cambios en las especificaciones del software 	<ul style="list-style-type: none"> • Especificaciones de casos de prueba • Informe de incidentes de prueba • Aplicación instalada probada

Actividades	Tareas	Técnicas y Notaciones	Productos
4.Elaboración de la Documentación	<ul style="list-style-type: none"> ▪ Elaborar los documentos o manuales del producto de SW 	<ul style="list-style-type: none"> • Técnicas de elaboración de documentos técnicos 	<ul style="list-style-type: none"> • Documentos o manuales de la aplicación
5.Entrenamiento de Usuarios	<ul style="list-style-type: none"> ▪ Preparar ambiente y material de entrenamiento ▪ Conducir entrenamiento de usuarios 	<ul style="list-style-type: none"> • Técnicas de entrenamiento 	<ul style="list-style-type: none"> • Material de entrenamiento

Conclusiones:

- El Método *White-Watch*
 - propone un modelo de procesos que balancea la producción de especificaciones de producto que se transforman en la medida que se avanza en el proceso, con la disponibilidad en corto tiempo de versiones parciales y operativas del producto.
 - logra dedicar más tiempo a la producción y puesta a punto del software – versiones del producto gracias al apoyo brindado por las herramientas automatizadas.
- Un Método de Desarrollo de software es más que una Metodología de desarrollo, pues describe, además del proceso que debería seguirse para la solución del problema, el producto final (el software y sus componentes parciales) y el equipo de trabajo que ejecutara el proceso de desarrollo.
- Adaptar una propuesta metodológica requiere de experiencia en el área – desarrollo de software, y de conocimiento del dominio – tipo de aplicaciones y tecnología disponible, del contexto organizacional y de gestión de proyectos, para lograr una configuración o instanciación adecuada a necesidades particulares.
-