



Universidad
Francisco de Vitoria
UFV Madrid

Planificación y Gestión de Proyectos Informáticos

Tema 9: Verificación y Validación del Software



Objetivos del tema

■ Ubicación

- Bloque IV: **PROCEDIMIENTOS DE GESTIÓN**
 - Tema 7: *Gestión del Proyecto*
 - Tema 8: *Gestión de la Configuración*
 - Tema 9: *Verificación y Validación del Software*

■ Objetivos

- Ampliar los *conocimientos básicos* de la Gestión de Proyectos
- Entender la necesidad de *verificar* y *validar* el software.
- Estar en disposición de generar un *Plan de Verificación y Validación del Software*, dentro del marco del desarrollo de un proyecto informático



Bibliografía recomendada

- ***ESA SW Engineering Standards. ESA PSS-05-0***
ESA BSSC, Issue 2. February 1991
- ***Guide to SW Verification & Validation. ESA PSS-05-10***
ESA BSSC, Issue 1, rev. 1. March 1995.

- *IEEE Std 1012-1986: IEEE Standard for Software Verification and Validation Plans*
Institute of Electrical and Electronics Engineers. New York 1986

Contenido



1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software

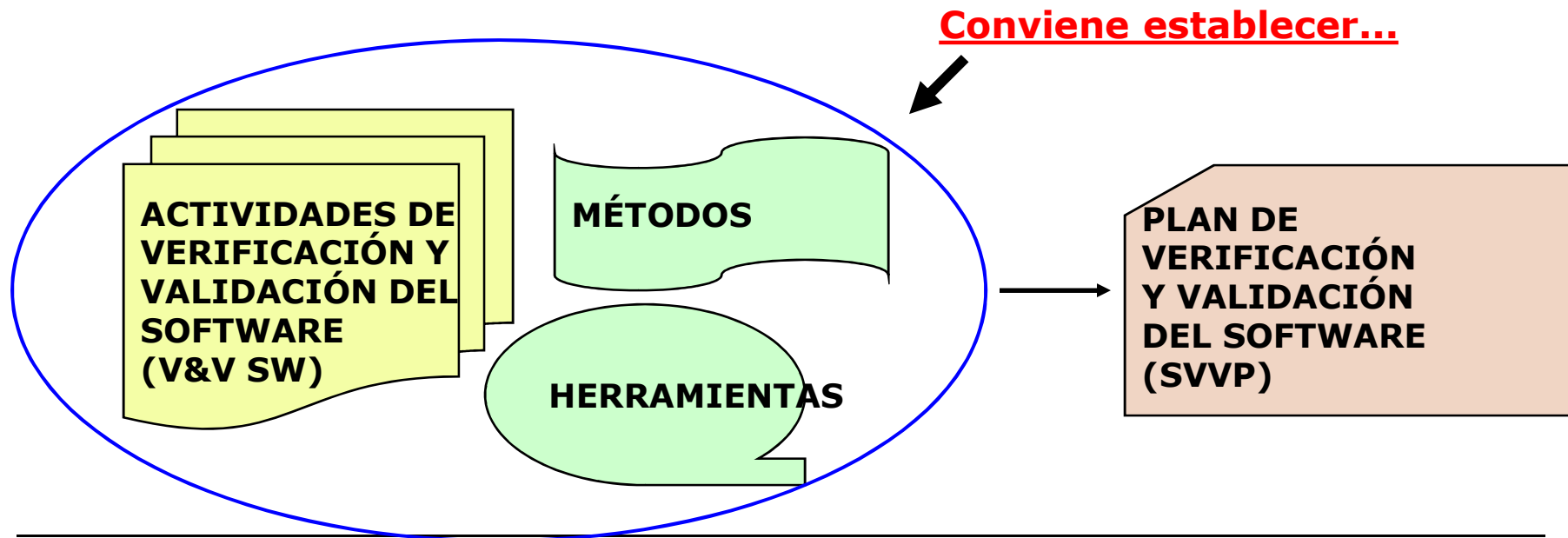


1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software



1. Introducción

- Realidades, entre otras, a tener en cuenta cuando se afronta el desarrollo de un proyecto informático:
 - El producto resultante debe funcionar correctamente.
 - El producto resultante debe cubrir todas las expectativas para las que fue concebido.
- Cuanto más se tarde en detectar una deficiencia o un error, mayor impacto temporal y de costes tendrá.





1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software



2.1 Definiciones

- Las actividades de verificación y validación (V&V) del software pretenden probar y comprobar que dicho software cumple con las especificaciones que fueron marcadas para su creación.

$$\text{V\&V} = \text{DETECCIÓN} + \text{CORRECCIÓN DE ERRORES}$$



2.1 Definiciones

- Por ello, durante el proceso de desarrollo de todo proyecto se debe:
 - Comprobar que todo elemento software producido cumple con los requisitos establecidos.
 - Comprobar cada elemento software antes de ser utilizado como una entrada de otra actividad.
 - Procurar que cada comprobación de un elemento es realizada por una persona distinta a la que ha desarrollado dicho elemento.
 - Asegurar que el esfuerzo dedicado a la verificación y validación es el adecuado para demostrar que cada elemento software creado puede ser utilizado con seguridad.



2.1 Definiciones

■ VERIFICACIÓN:

- *Acciones de revisión, comprobación, prueba y auditoria encaminadas a establecer y documentar cuándo un elemento SW, proceso, servicio o documento **se está construyendo correctamente**.*

VERIFICACIÓN = **PRODUCTO CORRECTO**

■ VALIDACIÓN

- *Es el proceso de evaluación de un sistema o componente, durante o al final del proceso de desarrollo, para determinar **si satisface o no los requisitos del usuario**.*

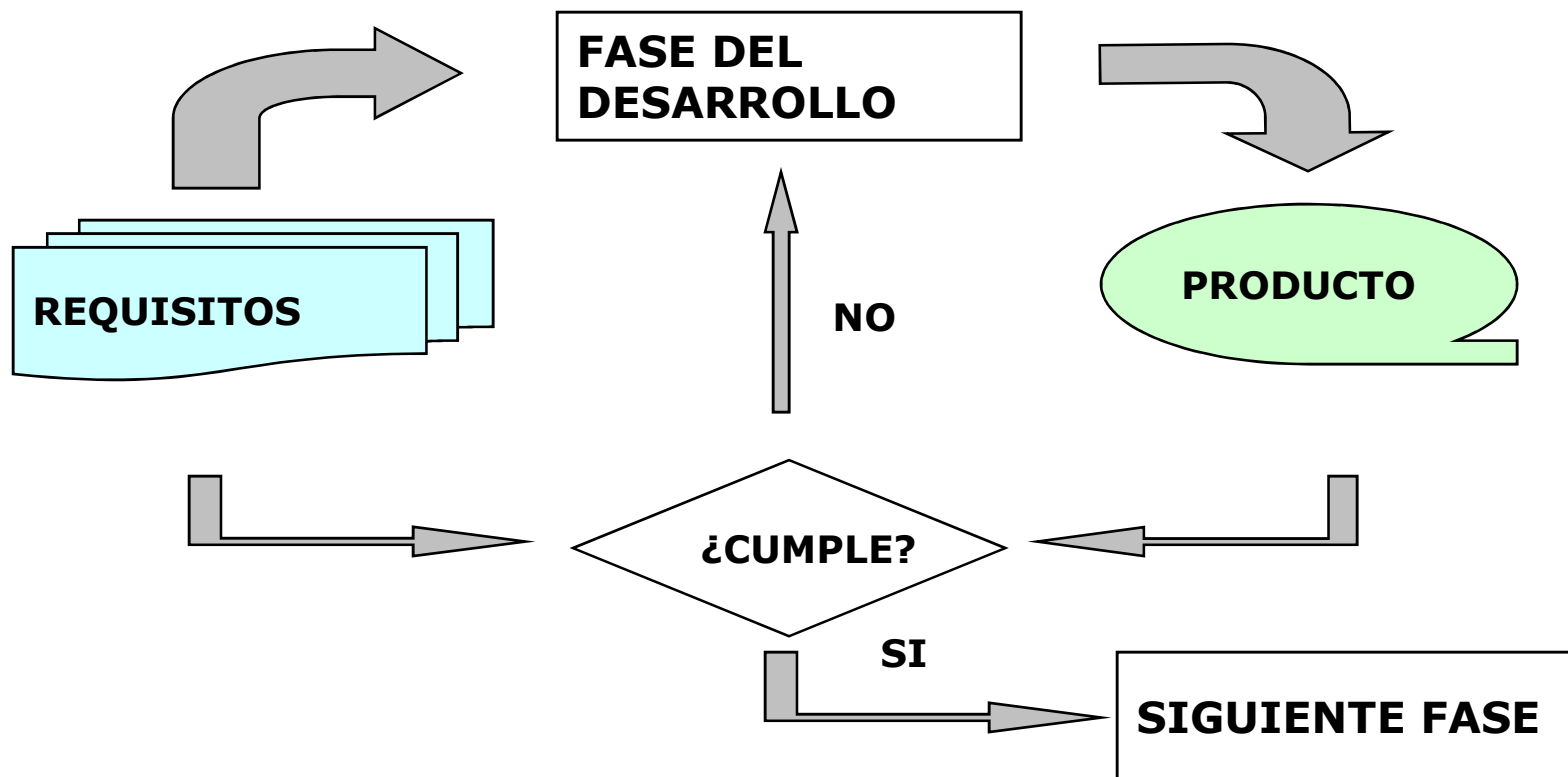
VALIDACIÓN = **PRODUCTO según ESPECIFICACIÓN**

VALIDACIÓN = “VERIFICACIÓN TOTAL”



2.1 Definiciones

- Proceso general de V&V





2.2 Actividades

(Según el estándar de la ESA)

- Revisiones
 - Walkthroughs: identificación temprana de errores formales y su solución.
 - Revisiones técnicas del SW en cada fase del desarrollo.
 - Auditorias: revisiones independientes de los elementos de configuración.
- Comprobaciones de trazabilidad: Entradas vs. Salidas
- Ensayos formales, demostrando que ciertas condiciones previas producen los resultados esperados.
- Pruebas de un sistema o componente, bajo condiciones dadas, observando y evaluando los resultados obtenidos.



2.2 Actividades

■ Pruebas:

- *Actividad en la que un sistema o componente se ejecuta, bajo unas condiciones dadas, y cuyos resultados son observados o registrados, realizándose una evaluación de los mismos.*

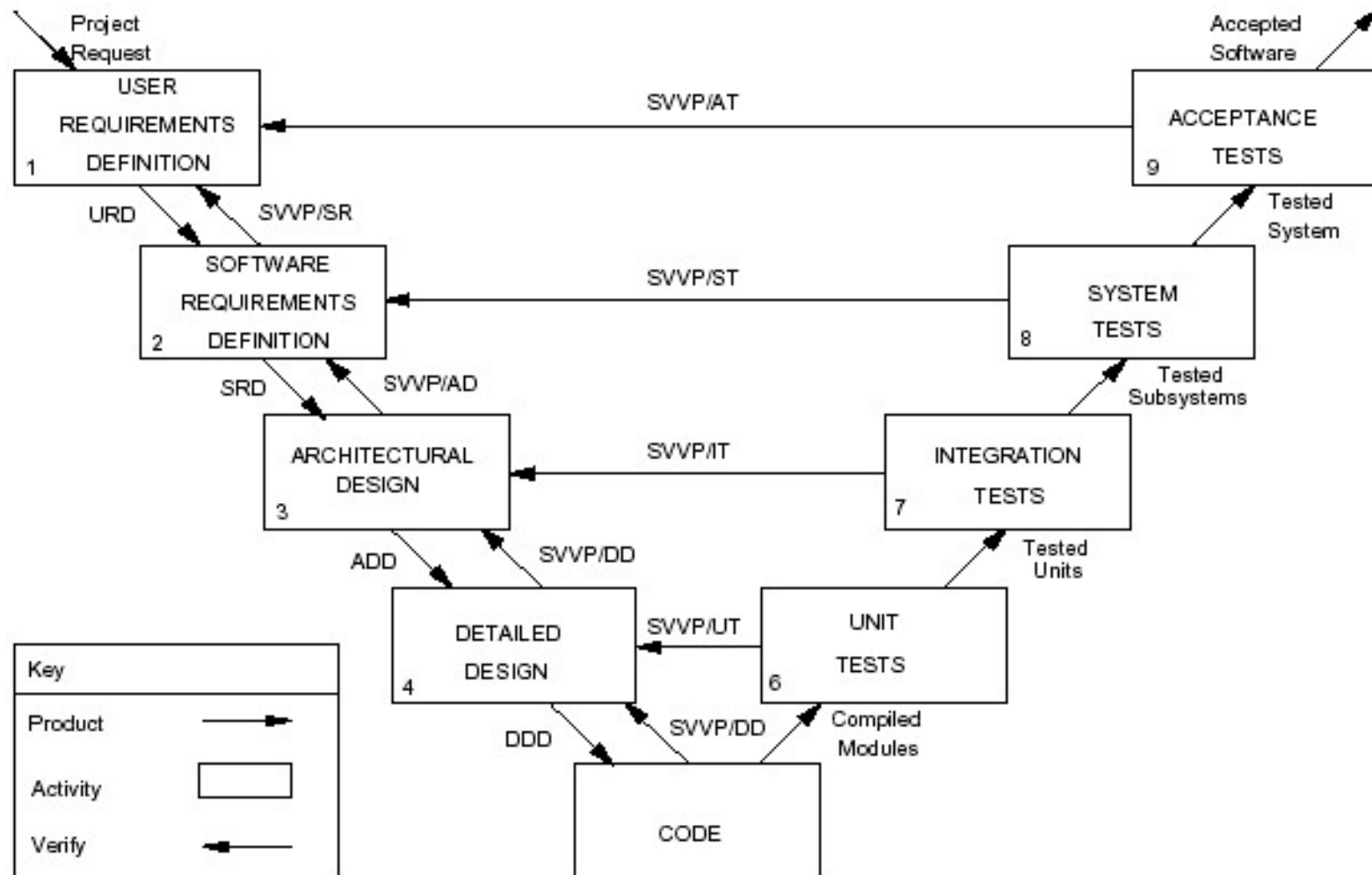
■ Tipos de pruebas

- **PRUEBAS UNITARIAS**, de uno o más módulos, definidos en el diseño detallado.
- **PRUEBAS DE INTEGRACIÓN**, de una o más unidades, tal y como se han definido en el diseño de alto nivel.
- **PRUEBAS DE SISTEMAS**, de aquellas unidades integradas con éxito, de acuerdo a lo establecido en los requisitos SW.
- **PRUEBAS DE ACEPTACIÓN**, de los sistemas probados con éxito, de acuerdo a lo establecido en los requisitos de usuario.



2.2 Actividades

■ Resumen del ciclo de pruebas del SW





2.2 Actividades

- **Trazabilidad:**
 - *Establecer una relación entre dos o más productos del proceso de desarrollo. Por ejemplo, entre un requisito dado y el elemento de diseño que lo implementa*
 - Se hace construyendo matrices de trazabilidad.
 - Requiere identificación unívoca de todos los componentes

- **Tipos de pruebas**
 - **TRAZABILIDAD HACIA DELANTE:** cada entrada a una fase debe de ser rastreable a una salida de esa fase.
 - La trazabilidad hacia adelante puede mostrar completitud, incompletitud o duplicación

 - **TRAZABILIDAD HACIA ATRÁS:** cada salida de una fase debe de ser rastreable a una entrada de esa fase.
 - Las salidas que no pueden ser trazadas son superfluas



2.2 Actividades

- Tipos de matrices de trazabilidad
 - Requisitos de usuario vs requisitos SW
 - Requisitos SW vs componentes del diseño de arquitectura
 - Pruebas unitarias vs componentes del diseño detallado
 - Pruebas de integración vs componentes del diseño de arquitectura
 - Pruebas de sistemas, vs requisitos SW
 - Pruebas de aceptación, vs requisitos de usuario



1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software



3. Métodos suplementarios para la V&V

Como complemento a las actividades básicas de verificación y validación, se pueden utilizar los siguientes métodos:

- Métodos suplementarios a las revisiones:
 - Inspecciones del software
- Métodos suplementarios a los ensayos formales:
 - Métodos formales
 - Técnicas de verificación de programa
- Métodos suplementarios a las pruebas:
 - Método “Cleanroom”
 - Pruebas estructuradas
 - Pruebas de integración estructuradas



3. Métodos suplementarios para la V&V

Inspecciones del software

- **Objetivo:**
 - Detectar errores en el código
 - Detectar errores en la documentación
- **Utilización:**
 - Para la detección de defectos en las fases de diseño de detalle, antes de la codificación, y de codificación, antes de las pruebas.
 - Para la verificación del diseño, casos y procedimientos de prueba.
 - Para la verificación de los productos de cualquier proceso de desarrollo.
- **Suelen ser:**
 - Eficientes (detectan el 50% de los defectos de desarrollo)
 - Económicos (por su eficacia con un bajo coste)



3. Métodos suplementarios para la V&V

Métodos formales (requisitos SW) - LOTOS, LARCH, OBJ

- **Objetivo:**
 - Crear técnicas de construcción y análisis con modelos matemáticos para la automatización del desarrollo de sistemas informáticos.
 - Se suelen utilizar en la fase SR.
- **Utilización:**
 - Para la detección defectos y ambigüedades en el desarrollo.
 - Para facilitar el análisis de sistemas.
 - Para facilitar la construcción de sistemas.
- **Ventajas:**
 - Mejor comprensión del código.
 - Mayor calidad del software.
 - Mayor productividad.
- **Inconvenientes:**
 - Poca implantación a nivel industrial.
 - Falta de madurez práctica.
 - Encarecimiento y ralentización en el desarrollo de los productos



3. Métodos suplementarios para la V&V

Técnicas de verificación de programa

- **Objetivo:**
 - Demostrar que el programa es consistente con sus especificaciones.
 - Se suelen utilizar en las fases DD y codificación.
- **Requiere que:**
 - El lenguaje de programación utilizado tenga una semántica y notación formalmente definidos.
- **Ventajas:**
 - Mayor calidad del software.
 - Mayor productividad.
- **Inconvenientes:**
 - Si el requerimiento arriba expuesto no se cumple, esta técnica no se puede aplicar.



3. Métodos suplementarios para la V&V

Método “Cleanroom”

- **Objetivo:**
 - Desarrollo de software de alta calidad con confiabilidad certificada, previniendo el defecto, en vez de corregirlo.
 - Sustituye las pruebas unitarias y de integración con técnicas de inspección del software y de verificación de programa específicas.
 - El método se ejecuta por un equipo de pruebas ajeno al proyecto.

- Este método no cumple totalmente con el estándar de la ESA (se omiten las pruebas unitarias y de integración).



3. Métodos suplementarios para la V&V

Pruebas Estructuradas

- Son métodos de verificación del software basados en las propiedades matemáticas de los gráficos de control.
- Aumenta las posibilidades de prueba limitando la complejidad durante la fase de diseño detallado.
- Ayuda en la definición de los casos de prueba durante las pruebas unitarias.



3. Métodos suplementarios para la V&V

Pruebas de Integración Estructuradas

- Son métodos de pruebas de integración basados en los métodos de Pruebas Estructuradas.
- Aumenta las posibilidades de prueba limitando la complejidad durante la fase de diseño de alto nivel.
- Ayuda en la definición de los casos de prueba durante las pruebas de integración.
- Se pueden aplicar a todos los niveles de diseño por encima del módulo (de pruebas de integración en adelante).



1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software



1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software



4. Herramientas para la V&V

- Para la realización de las actividades básicas para la verificación y validación del software, definidas en el estándar de la ESA, existen herramientas específicas:
 - Herramientas para REVISIONES:
 - Herramientas ofimáticas
 - Analizadores estáticos
 - Herramientas de gestión de configuración
 - Herramientas de ingeniería inversa
 - Herramientas para TRAZABILIDAD
 - Herramientas ofimáticas
 - Herramientas de bases de datos



4. Herramientas para la V&V

- Herramientas para ENSAYOS FORMALES
 - Pre-procesadores
 - Analizadores semánticos
 - Verificadores de programa
- Herramientas para PRUEBAS
 - Generadores de pruebas CASE
 - Debuggers
 - Analizadores estáticos
 - Pruebas de consistencia



1. Introducción
2. Verificación y Validación del Software
 1. Definiciones
 2. Actividades
3. Métodos suplementarios para V&V
4. Auditorías
5. Herramientas para V&V
6. Plan de Verificación y Validación del Software

5. Plan de Verificación y Validación del Software



ESA-PSS-05-10 Guide to Software Verification and Validation

- Capítulo 2: Principios y Actividades de Verificación y Validación del SW
- Capítulo 3: Métodos de V&V del SW
- Capítulo 4: Herramientas para la V&V del SW
- Capítulo 5: Estructura y contenidos de un SVVP



5. Plan de Verificación y Validación del Software

- Todas las actividades de verificación y validación del software han de quedar reflejadas en el correspondiente Plan de Verificación y Validación del Software (SVVP).
- El SVVP se divide en siete secciones:
 - SVVP/SR – Plan de verificación Fase SR
 - SVVP/AD – Plan de verificación Fase AD
 - SVVP/DD – Plan de verificación Fase DD
 - SVVP/AT – Pruebas de aceptación
 - SVVP/ST – Pruebas de sistemas
 - SVVP/IT – Pruebas de integración
 - SVVP/UT – Pruebas unitarias



5. Plan de Verificación y Validación del Software

- Cuándo y dónde se documenta, en el SVVP, cada una de las actividades de verificación y validación del software.

PHASE SVVP SECTION	USER REQUIREMENTS DEFINITION	SOFTWARE REQUIREMENTS DEFINITION	ARCHITECTURAL DESIGN	DETAILED DESIGN AND PRODUCTION	TRANSFER
SVVP/SR	SR Phase Plan				
SVVP/AD		AD Phase Plan			
SVVP/DD			DD Phase Plan		
SVVP/AT	AT Plan			AT Designs AT Cases AT Procedures	AT Reports
SVVP/ST		ST Plan		ST Designs ST Cases ST Procedures ST Reports	
SVVP/IT			IT Plan	IT Designs IT Cases IT Procedures IT Reports	
SVVP/UT				UT Plan UT Designs UT Cases UT Procedures UT Reports	

5. Plan de Verificación y Validación del Software



- Consejos/Recomendaciones:
 - El desarrollador es, normalmente, el responsable de la generación del SVVP
 - En algunos casos, el usuario del sistema puede encargarse de generar las especificaciones para las pruebas de aceptación (SVVP/AT).
 - Dado que cada sección del SVVP ha de ser preparada en diferente momento del desarrollo del software, se han de mantener como elementos de la configuración independientes, dentro del mismo documento.