Validación y Verificación

Introducción

Cátedra Ingeniería de Software

Ingeniería en Sistemas de Información - UTN – F. R. Rosario

Referencias

- **El arte de probar el software** – Myers - 1979
- The Art of Software Testing – Myers - 2004
- **Guías INTECO de VAL y VER**
  - guia_de_validacion_y_verificacion
  - guia_de_mejores_practicas_de_calidad_de_producto
- Otros activos de INTECO (Servicio repositorio documental)
- Guías de RUP
- Guía para la integración de procesos y la mejora de productos de CMMI - cmmi-dev-v1.2
- Presentación de Ingeniería de Software de Gabriela Arévalo

Objetivos de esta presentación:

- **Introducción a Validación y Verificación**

Validación y Verificación

- La **validación** tiene como objetivo asegurar que el producto satisface las expectativas del cliente.

- La **verificación** implica comprobar que el producto está de acuerdo con su especificación.

Validación y Verificación (V & V)

De manera informal (*):

Validación

- ¿Estoy construyendo el producto correcto?
  - El software debería hacer lo que el usuario (cliente) requiere (necesita)

Verificación

- ¿Estoy construyendo correctamente el producto?
  - El software debería ajustarse a su especificación

(*) Boehm, 1979 – Guidelines for Verifying and Validating Software Requirements and Design Specifications

Validation: "Am I building the right product?“

Verification: "Am I building the product right?"

Proceso de V & V

- La V & V debe aplicarse en cada etapa del desarrollo del software.
- Tiene dos objetivos principales
  - El descubrimiento de defectos en el sistema
  - La evaluación de si el sistema es útil y utilizable en una situación operacional o no

Metas de la V & V

- La verificación y la validación deberían establecer la confianza de que el software es adecuado al propósito.
- Esto NO significa que esté completamente libre de defectos. Sino que debe ser lo suficientemente bueno para su uso previsto y el tipo de uso determinará el grado de confianza que se necesita.

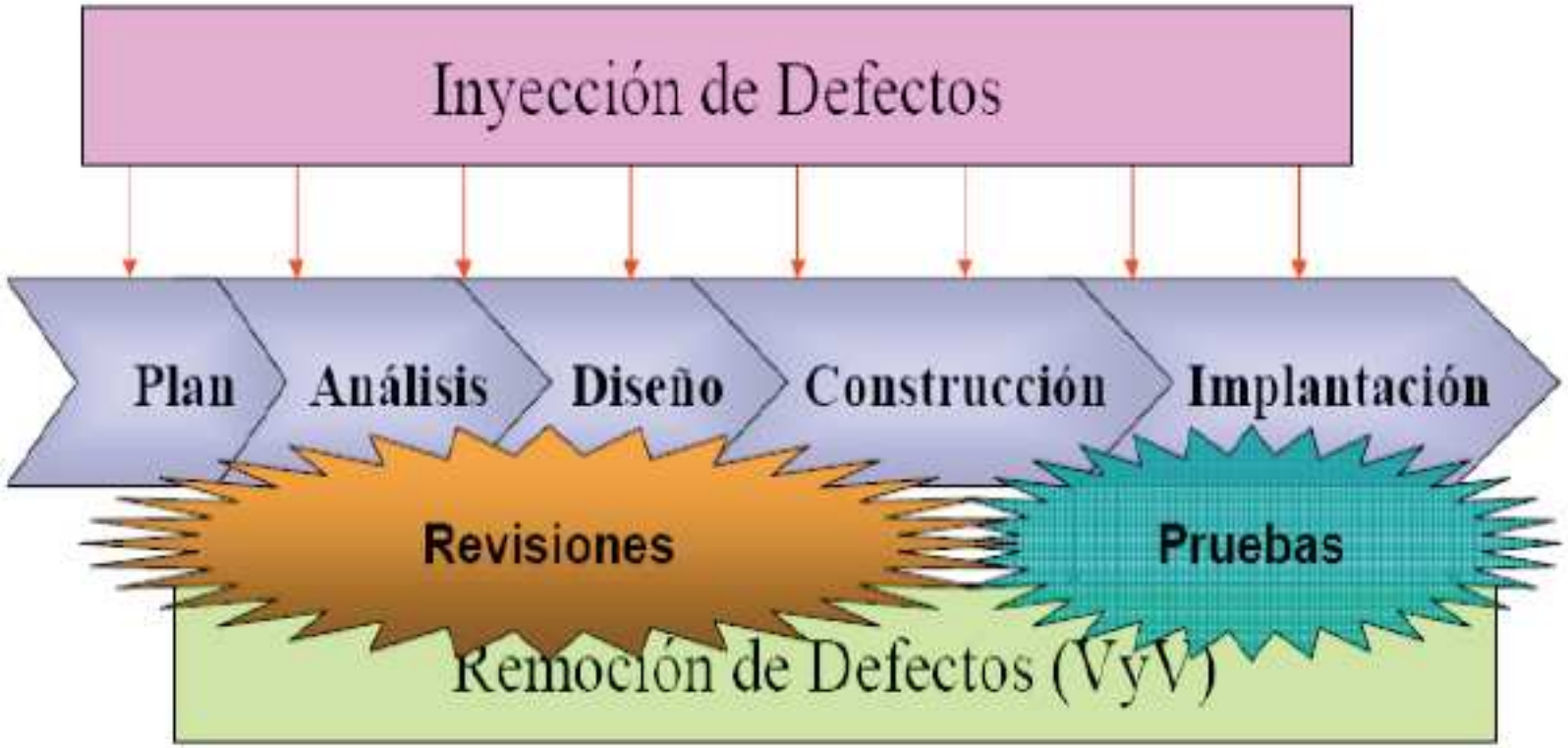
Grado de Confianza de la V & V

- Depende del propósito del sistema, las expectativas del usuario y el entorno de marketing
  - Función del software
    - El nivel de confianza depende de lo crítico que es el sistema para una organización.
  - Expectativas del usuario
    - Los usuarios pueden tener bajas expectativas para ciertas clases de software.
  - Entorno de marketing
    - Introducir un producto en el mercado pronto puede ser más importante que encontrar defectos en el programa

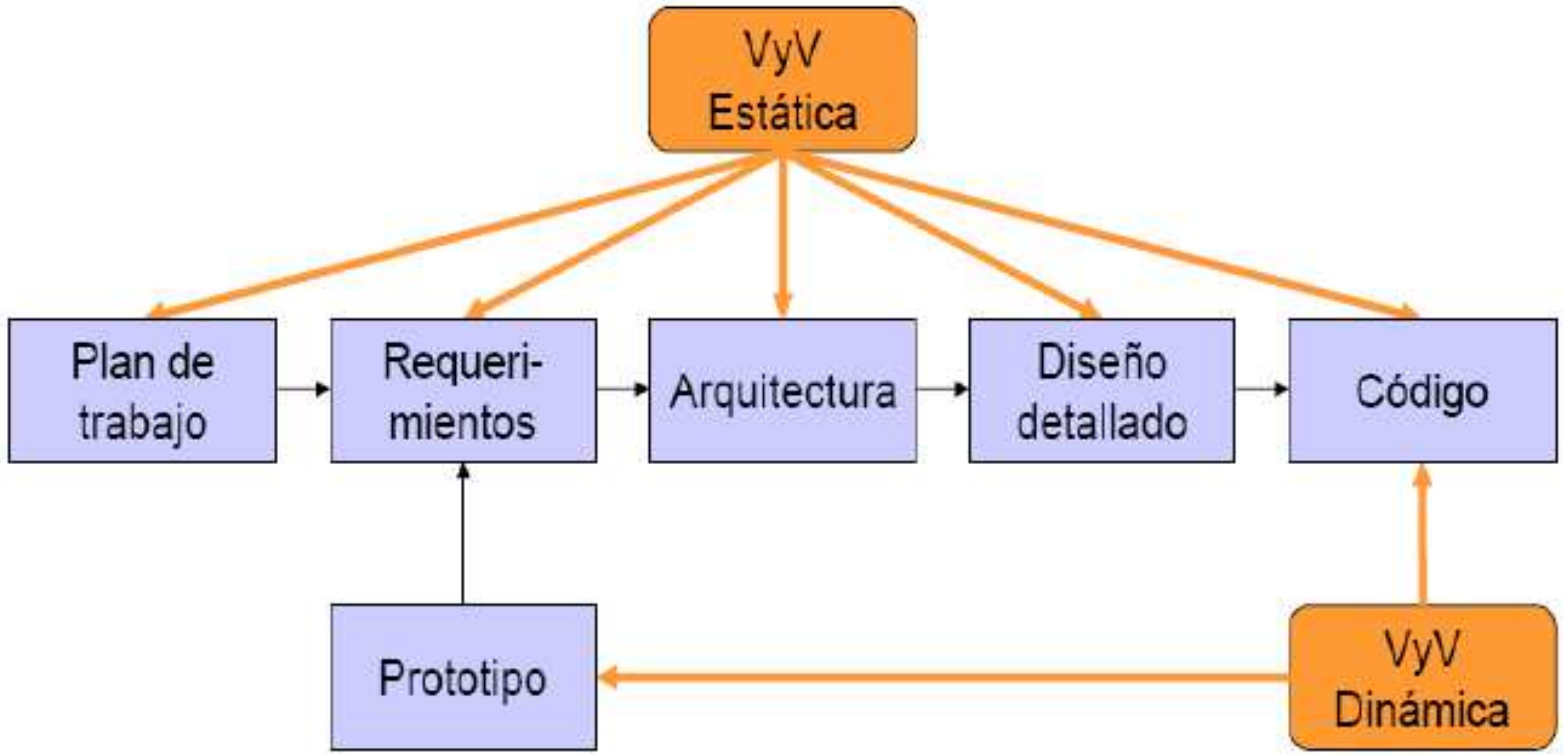
V & V Estática y Dinámica

- *ESTATICA – Revisiones*
  - Se relaciona con analizar una representación estática del sistema para descubrir problemas
  - Para evaluar o analizar documentos de requisitos, documentación de diseño, manuales de usuario, e incluso para examinar el código fuente antes de ejecutarlo
- *DINAMICA – Pruebas*
  - Se ejecuta el sistema con datos de tests y se observa el comportamiento operacional
  - Para detectar defectos y verificar atributos de calidad del software

V & V Estática y Dinámica



V & V Estática y Dinámica



Revisiones

- Ventajas
  - No requiere de código ejecutable, por lo que puede ser realizada desde el inicio
  - Se encuentran varios defectos a la vez
  - Se localiza la posición exacta del defecto
- Desventajas
  - Requiere del tiempo de expertos
  - Validan cumplimiento de lo que se especificó, en vez de lo que realmente desea el cliente

Clasificación de las Revisiones

- Informales
  - No hay proceso definido
  - No existen roles
  - Usualmente no planeadas
- Formales
  - Objetivos definidos
  - Proceso documentado
  - Roles definidos y personas entrenados en ellos
  - Check-lists, reglas y métodos para encontrar defectos
  - Reporte del resultado
  - Recolección de datos para el control del proceso

Pruebas

- Las pruebas constituyen un proceso con el objetivo principal de encontrar defectos en el software.
- Una prueba tiene éxito si descubre un defecto y fracasa si hay defectos pero no es capaz de descubrirlos.
- Es imposible realizar pruebas exhaustivas y garantizar la ausencia de defectos.
- Las pruebas sólo pueden demostrar la presencia de errores, pero no su ausencia.

Clasificación de Pruebas

- Dependiendo de quién prueba
  - Internas
  - Externas
- Dependiendo de qué se prueba
  - Unitarias
  - Integración
  - Sistemas
  - Aceptación
- Dependiendo de cómo se diseñan
  - Caja Negra
  - Caja Blanca

Clasificación de Pruebas

- **Dependiendo de quién prueba**
  - Internas
    - Por el equipo de desarrollo del software
  - Externas
    - Por el cliente (a veces junto con los desarrolladores)

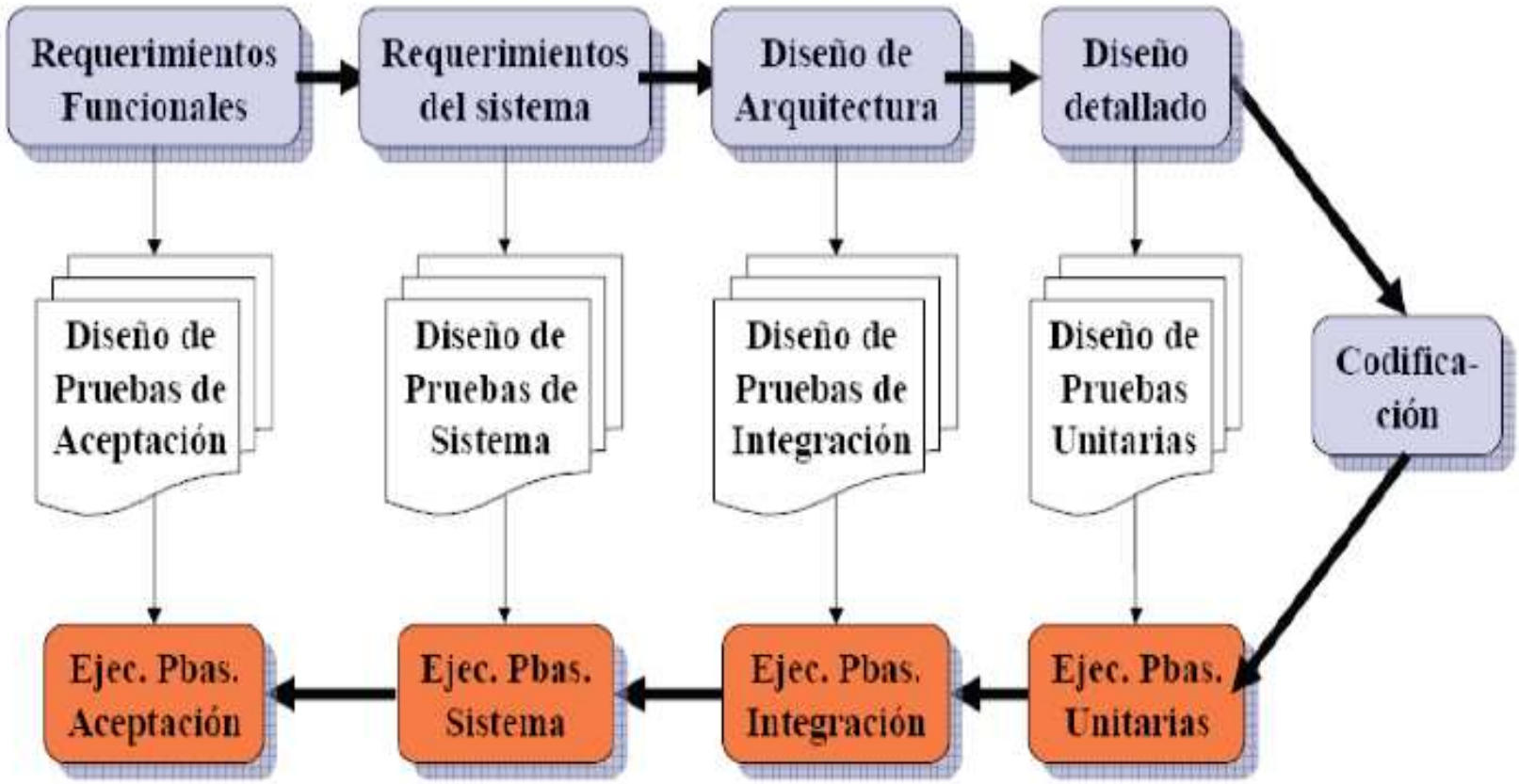
Pruebas Externas

- Pruebas alfa
  - Instalaciones del desarrollador
  - Grupo muy reducido de clientes
- Pruebas beta
  - Instalaciones del cliente, ambiente de “laboratorio”
  - Grupo más amplio de clientes
- Pruebas piloto
  - Instalaciones del cliente, ambiente de producción
  - Conjunto reducido de departamentos del cliente

Clasificación de Pruebas

- **Dependiendo de qué se prueba**
  - Pruebas Unitarias
    - Componentes individualmente
  - Pruebas de Integración
    - Módulos o subsistemas ya integrados
  - Pruebas del Sistema
    - Características técnicas del sistema completo
  - Pruebas de Validación o Aceptación
    - Funcionalidad del sistema completo

Clasificación dependiendo de qué se prueba



Clasificación de las pruebas
dependiendo de cómo se diseñan

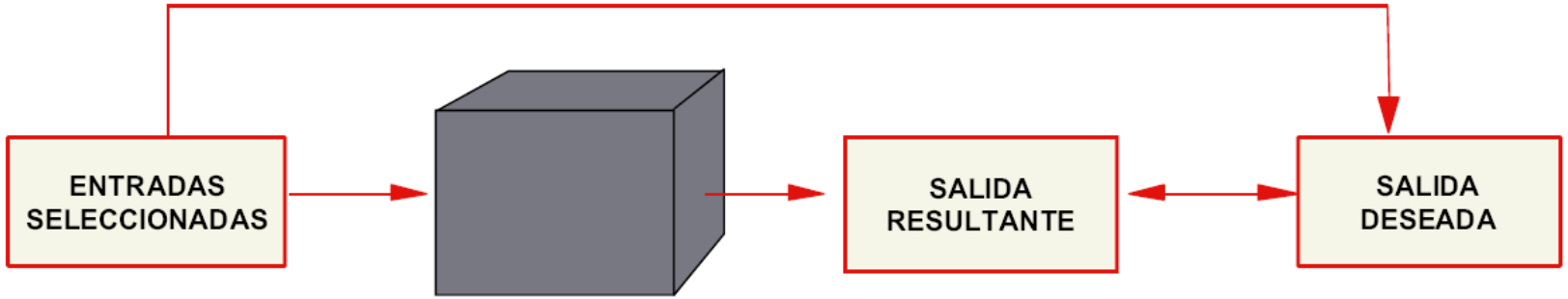
- Pruebas de caja negra (Black-box)
  - Funcionales
  - No Funcionales
- Pruebas de caja blanca (White-box)
  - Estructurales

Pruebas funcionales

- Tienen por **objetivo** probar que los **sistemas desarrollados cumplen con las funciones** específicas para los que han sido creados.
  - La función de un sistema es '**lo que hace**' dicho sistema
- **Basadas** en el análisis de la **especificación de requisitos**

Pruebas funcionales

- El enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y de salida, sin preocuparse del funcionamiento interno del sistema.

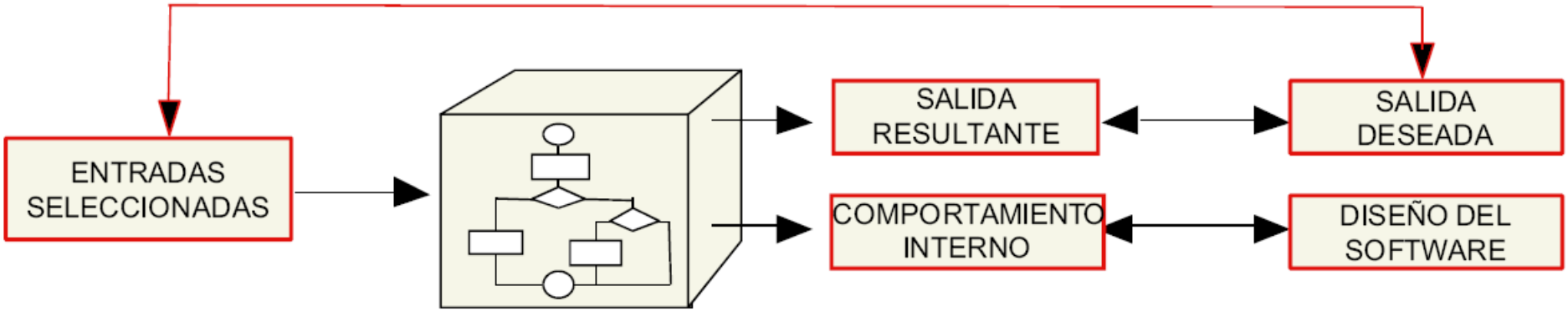


Pruebas no funcionales

- Un objetivo de las pruebas es probar **la calidad de las características de un software** (Atributos no funcionales o Atributos de calidad)
- Las pruebas no funcionales incluyen pruebas de
  - **Rendimiento**
  - **Usabilidad**
  - **Mantenibilidad**
  - **Fiabilidad**
  - **Portabilidad**

Pruebas estructurales

- Las pruebas estructurales se interesan en lo que pasa ‘dentro del sistema’
- Se derivan a partir del conocimiento de la estructura interna del software



Preguntas?