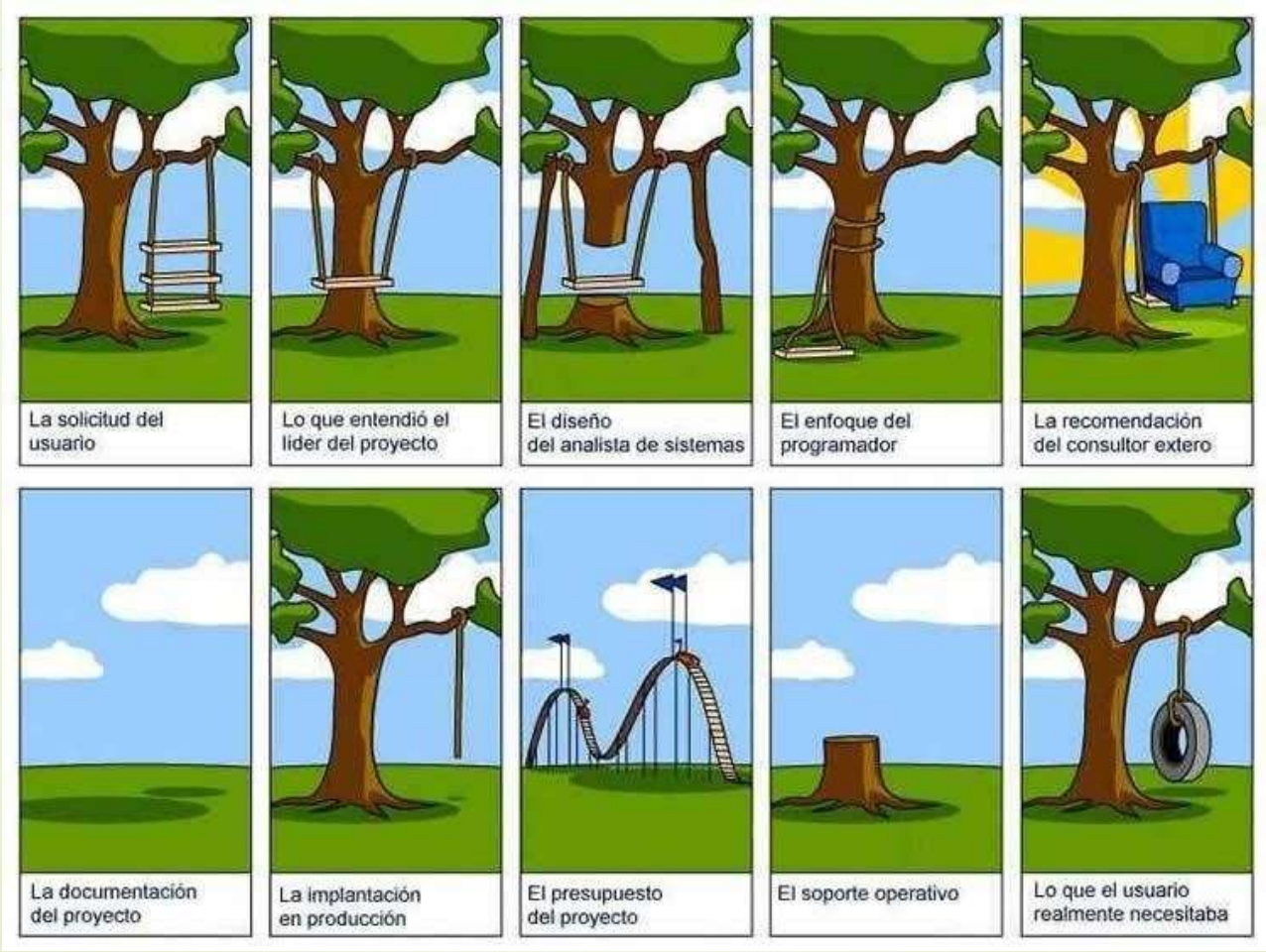
Técnicas de Documentación y Validación

Unidad 0 - Introducción

El proceso V (& V) &D

- Es el proceso de todo un ciclo vital: La V & D debe aplicarse en cada etapa de un proyecto.
- Independiente de la magnitud.
- ¿Por qué las abordamos en conjunto?
- Vamos por partes…

Real vs. Ideal

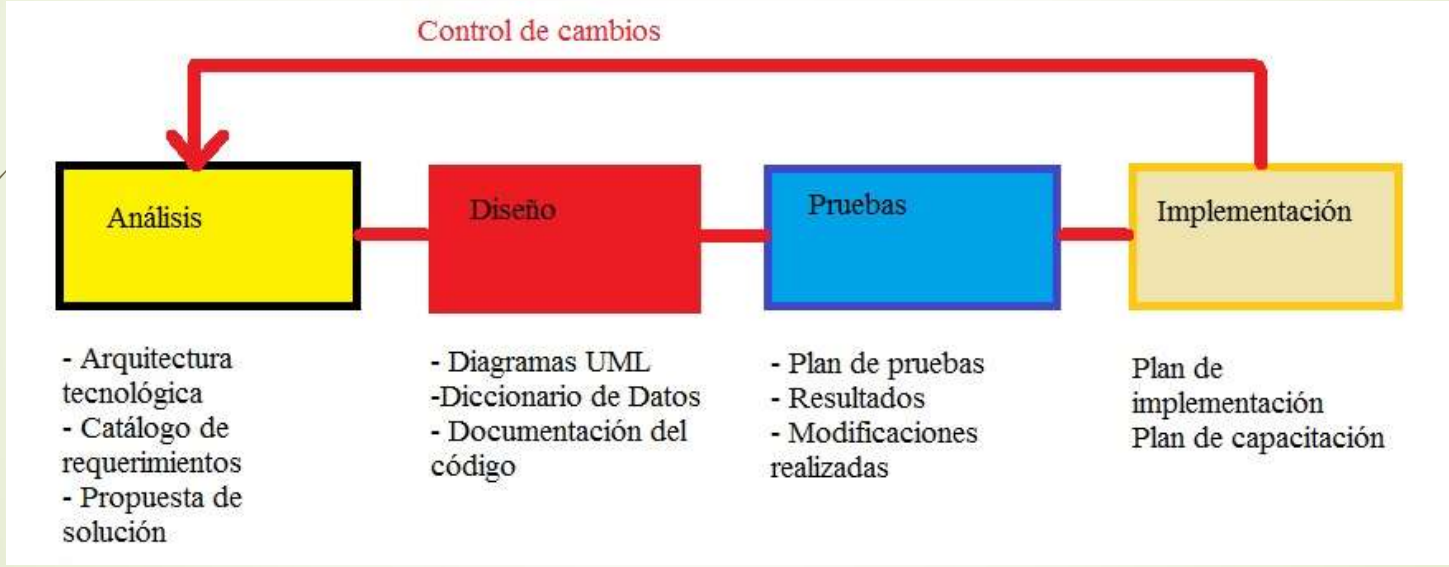


Documentación

- ¿Qué y por qué documentamos?
- ¿Cuándo documentamos?
- ¿Quiénes son los destinatarios de la documentación?
- ¿La documentación, es redundante?

Documentación

- Cuando y Que..



Metas de la V&V

- La verificación y la validación deberían establecer la confianza de que el software es adecuado al propósito.
  - Validación: ¿Estamos construyendo el producto correcto? Se ocupa de controlar si el producto satisface los requerimientos del usuario
  - Verificación: ¿Estamos construyendo correctamente el producto? implica controlar que el producto conforma su especificación inicial.
- Esto NO significa que esté completamente libre de defectos.
- Debe ser lo suficientemente bueno para su uso previsto y el tipo de uso determinará el grado de confianza que se necesita.

Validación

- ¿Qué y por qué validamos/testeamos?
- ¿Cuándo validamos/testeamos?
- ¿Quiénes son los destinatarios de la validación/testeo?
- ¿Cuánto debemos dedicarle a la validación/testeo?

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

Verificación dinámica y estática

- Inspecciones de software. Se ocupa del análisis de representaciones estáticas del sistema para describir problemas (verificación estática)
  - Pueden ser complementadas por documentos basados en herramientas y análisis del código
- Pruebas del software. Se ocupa de la ejercitación y la observación del comportamiento del producto (verificación dinámica)
  - El sistema se ejecuta con datos de pruebas y se observa su comportamiento operativo.

Planificación de V &V

- Se requiere una cuidadosa planificación para sacar el máximo de los procesos de inspección y pruebas. La planificación debería comenzar pronto en el proceso de desarrollo.
- El plan debería identificar el balance entre la verificación estática y las pruebas.
- La planificación trata de definir estándares para el proceso de prueba en lugar de describir pruebas de productos.

Estructura de un plan de pruebas de software

- Proceso de pruebas
- Trazabilidad de requerimientos.
- Elementos probados.
- Calendario de pruebas.
- Procedimientos de registro de las pruebas.
- Requerimientos hardware y software.
- Restricciones.

Inspecciones de software

- Implican que las personas examinen la representación de la fuente con el propósito de descubrir anomalías y defectos
- Las inspecciones no requieren la ejecución de un sistema por lo que debe utilizarse antes de la implementación.
- Pueden estar aplicados a cualquier representación del sistema (requerimientos, diseño, configuración, datos, pruebas de datos, etc).
- Se ha demostrado que es una técnica efectiva para descubrir errores del programa.

Éxito de la inspección

- Pueden descubrirse muchos diferentes defectos en una sola inspección. Al probar, un defecto puede enmascarar a otro así que se requieren varias ejecuciones.

Inspecciones y pruebas

- Las inspecciones y pruebas son complementarias y no técnicas opuestas de verificación.
- Ambas deben utilizarse durante el proceso V & V.
- Las inspecciones pueden comprobar el ajuste con una especificación pero no la conformancia con los requerimientos reales del cliente.
- Las inspecciones no pueden comprobar características no funcionales como rendimiento, usabilidad, etc.

Inspecciones de programas

- Es la aproximación formalizada a las revisiones del documento
- Está pensado explícitamente para la detección de defectos (no su corrección)
- Los defectos pueden ser errores lógicos, anomalías en el código que pueden indicar una condición errónea (p.ej: una variable no iniciada) o no conformidad con los estándares.

Happy testing!