



Buenas prácticas de software



Unidad 4. Buenas prácticas para los modelos de prueba de software



Buenas prácticas de software



4.1 Método verificar y validar



Buenas prácticas de software



4.1 Método verificar y validar

La actividad de verifica y validar el software desarrollado, se lleva a cabo mediante actividades que permitirán identificar si el software cumple con los requisitos definidos inicialmente en la etapa de análisis.

Por lo que la verificación y la validación se puede definir como el conjunto de actividades que permiten analizar y comprobar cada una de las funciones que integran el software.



Buenas prácticas de software



En la verificación.

Se comprueba que el sistema o software realiza las funciones y sub-funciones definidas en la etapa de análisis de requerimientos.

Considerando la gestión de desarrollo del software, la verificación hace referencia al proceso para determinar si el flujo de trabajo se ha realiza conforme lo estipulado en el manual de procedimientos de trabajo.

El proceso de verificación debe aplicarse en cada etapa del software.



Buenas prácticas de software



La validación.

Es un procedimiento que garantiza que el software o sistema cumple con las buenas prácticas de desarrollo y construcción, garantizando la calidad de los procesos establecidos para el desarrollo.

Es un conjunto de actividades que se realizan para asegurar que el sistema o software desarrollado cubre y corresponde a los requerimientos del cliente.

El proceso de validación debe aplicarse en cada etapa del software aunque en muchos casos se hace al final del desarrollo y antes de entregar el producto.



Buenas prácticas de software



Calidad del software.

La calidad del software se determina en la medida en que las metricas de calidad son alcanzadas por el sistema o software desarrollado.

La calidad hace referencia a la excelencia, la que permite identificar que el software realiza las funciones especificadas por el usuario, estas las deber llevar a cabo sin errores.



Buenas prácticas de software



En el proceso de verificar y validar la eficiencia del software se pueden utilizar herramientas y técnicas, a continuación se da a conocer dos de ellas.

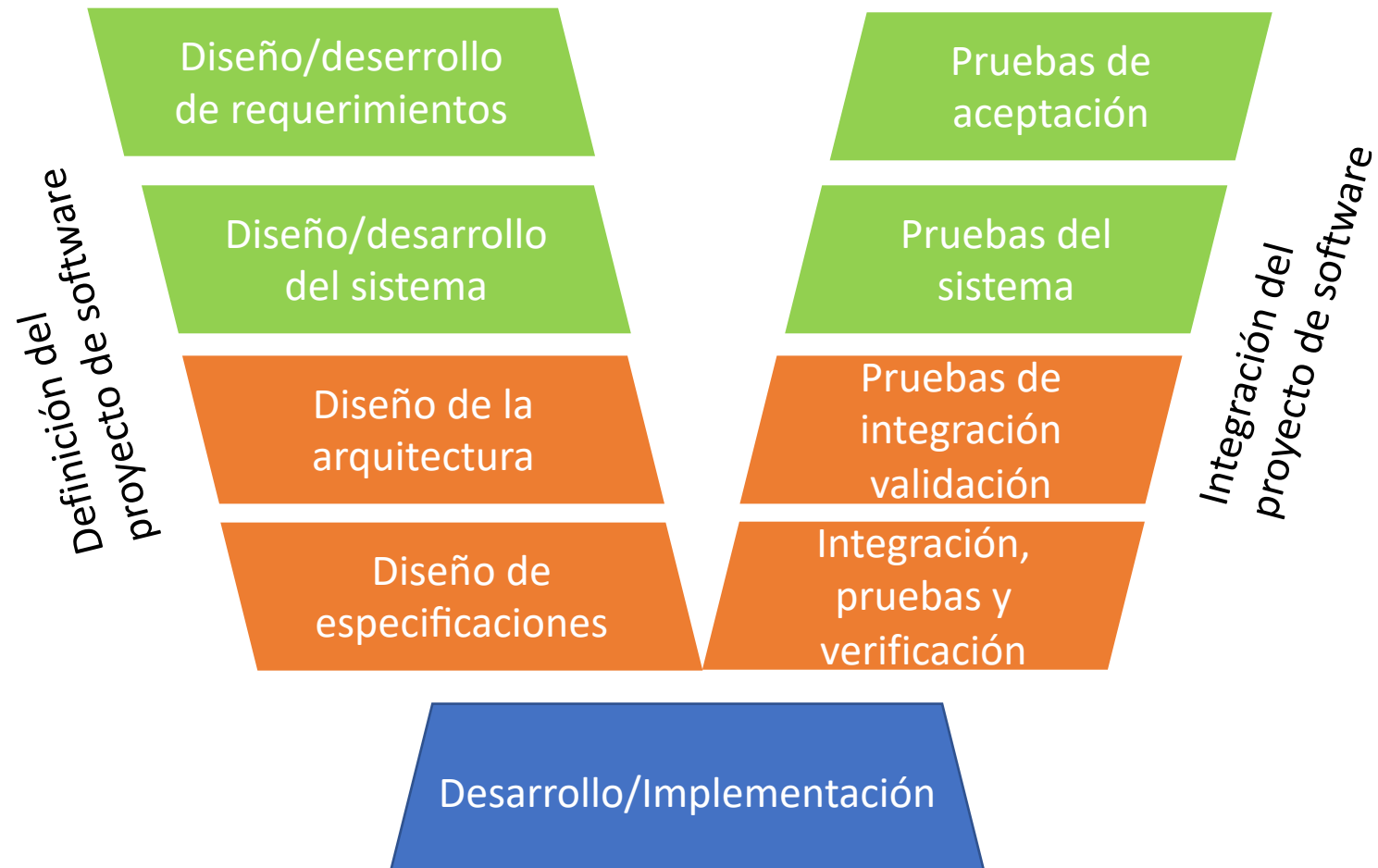
1. La inspección del software: Es donde se lleva a cabo el análisis y la comprobación de los requerimientos del software considerando la documentación del desarrollo del proyecto de software.
2. El usar técnicas y herramientas automatizadas para coprobar que el sistema funciona correctamente, estas herramientas dan un informe de cada una de las pruebas que el sistema ejecuta cuando esta en operación.
3. Las pruebas del software: Se implementan utilizando los datos de prueba y se revisan las salidas del software y su comportamiento operacional, para comprobar que se desempeña conforme al modelado de requerimientos.



Buenas prácticas de software



Método verificar y validar (V&V)





Buenas prácticas de software



4.1.1 Casos de prueba.

1. Los casos de prueba permite medir la funcionalidad en un cada una de las funciones o subfunciones del software para verificar el resultado.
2. Los casos de prueba se aplican en el desarrollo de software o sistemas, se pueden utilizar las pruebas se pueden realizar manualmente (algo que ya casi no se realiza) o bien se puede utilizar herramientas de pruebas que cubren los de casos de prueba.
3. Tambien se pueden utilizar los script para hacer las pruebas, los cuales son programas que permiten realizar las pruebas de algunas funciones.

La documentación de los casos de prueba deben contener la planeación anticipada.

Se debe tomar en cuenta que los casos de prueba deben ser meticulosamente planeados y contar con un instrumento de control.



Buenas prácticas de software



Los caso de prueba deben estructurarse por lo menos con ocho pasos.

Apartado 1: ID de caso de prueba.

Identificar el caso de prueba (ID) el cual debe ser único para identificarlos.

El ide deber ser uni (**estandar**) lo que permite la mejor organización e identificación de cada una de las pruebas.

Apartado 2: Descripción de la prueba.

Ya asignado un ID a la prueba, esta debe tener una descripción que indique a que módulo o función se refiere y que es lo que realiza.

Apartado 3: Supuestos y condiciones previas.

Describe el resultado que se espera de la función y permite verificar que la función cumple con el propósito.

Apartado 4: Datos de prueba.

Se deben indicar los datos que utiliza la función y la información que genera.



Buenas prácticas de software



Apartado 5: Pasos a ejecutar.

Son las acciones que el usuario ejecuta y que espera el sistema responda.

Apartado 6: Resultado Esperado.

Sdon los resultado esperados después de la ejecución de la función.

Apartado 7: Resultado real y condiciones posteriores.

Se realiza las verificación de la información obtenida de las funciones, con el información esperada, por lo que se puede determinar el estado de funcionamiento del software.

Apartado 8: Determinación de resultados.

La comparación de los resultados da el indicador de falla o satisfactorio.

Mismo resultado = Aprobado Diferentes resultados = Falla



Buenas prácticas de software



Test de prueba.

Se debe considerar elaborar un test de prueba para identificar los resultados reales y poder hacer los comentarios, ajustes, correcciones o lo que proceda para identificar si el software cumple con los requisitos establecidos por el cliente.



Buenas prácticas de software



Ejemplo de test de prueba.

| ID del caso de prueba | CP-RF-01 |
|---------------------------|---|
| Nombre del caso de prueba | Autorización de crédito. |
| Descripción | Se autoriza al cliente el crédito solicitado en función de su límite de crédito. |
| Precondiciones | El cliente debe tener como límite de crédito \$10,000 |
| Pasos a seguir | <ol style="list-style-type: none">1. Ingresar nombre de cliente.2. Revisar el límite de crédito.3. Revisar la cantidad solicitada de crédito.4. Validar la autorización.5. Autorizar crédito. |
| Resultado esperado | El cliente obtiene la autorización del crédito. Comprobante de autorización de crédito. |
| Resultado real | No se encuentra al cliente. |
| Comentarios | Compra autorizada. |



Buenas prácticas de software



Mejores prácticas para llevar a cabo casos de prueba.

1. Elabore un plan de pruebas.
2. Revisar la información que se despliega.
3. Diseñar los casos de prueba diseñarlos para que sean reutilizables.
4. Los ID deben ser únicas y corresponder a uno y solo un casos de prueba.
5. En la manera de lo posible realice las pruebas por pares.
6. Las pruebas realicelas con el usuario final.
7. Mediante la documentación del desarrollo verifique los resultados que se esperan y los obtenidos.



4.1.2 Matriz de trazabilidad.

La matriz de trazabilidad de requisitos (RTM) describe y sigue la vida de un requerimiento.

La RTM sigue y relaciona cada uno de los requerimientos del software.

La matriz de trazabilidad apoya en la etapa final del desarrollo del software para verificar que se cumplieron los requisitos que se solicitaron y los cuales se entregan.

La matriz de trazabilidad es de doble sentido va hacia adelante como hacia atrás o bien puede ser bidireccional.

Permite identificar si los resultados del desarrollo alcanzo y cumplió cada requisito solicitado.

Además se visualizan que requisitos son los que ya se consideran entregables.

Permiten hacer el seguimiento de la ingeniería de requerimientos durante el ciclo de vida y se asegura que estos sean entregados según especificaciones.

Utilizando herramientas automatizada gráficas es fácil verificar que se cumplen todos los requerimientos del software.



Buenas prácticas de software



Tipos de RTM.

Trazabilidad hacia delante RTMs: Se asegura que el proyecto se desarrollo conforme a lo planeado, mediante los mapeo de los requerimientos.

RTMs de trazabilidad retrospectiva: La matriz permite revisar el desarrollo de software no ha tenido desvíos considerando los requerimientos iniciales y que el proyecto sigue en el camino correcto así como permite visualizar que el propósito inicial no tenga desviaciones.

Trazabilidad bidireccional RTMs: Esta conformada o se puede realizar con la orientación de la matriz hacia delante, así mismo utilizando la matriz hacia atrás, lo cual permite asegurar que los requerimientos cuenten con sus casos de prueba.



Buenas prácticas de software



Elementos de una matriz de trazabilidad.

ID: Se asigna un número que permite identificar a los uno de los requerimientos.

Tipo de desarrollo: Se indica y da a conocer el tipo de proyecto para identificarlo con claridad.

Prioridad de las funciones: Permite indicar la prioridad asignada al requisito y permite hacer la valoración para tomar las decisiones cuando se esta evaluado del desarrollo del proyecto.

Estado de desarrollo de las funciones: Permite identificar el estado de desarrollo en que se encuentra el requisito, este puede ser activo, en proceso, terminado o cancelado.

Objetivo a cumplir: Se refiere al objetivo que logra con el requisito.

Entregable: Se refiere al entregable que satisface el requisito.

Estado del entregable: Identifica cual es el estado del requisito entregable.



Buenas prácticas de software



Ejemplo de matriz de trazabilidad.

| ID | Descripción | Tipo | Prioridad | Estado | Objetivo | Entregable | Estado |
|-----------|---|-------|-----------|--------|--|-----------------------------------|-----------|
| CP1-RF-01 | El sistema recogerá los siguientes datos... | Venta | Alta | Activo | Obtener los datos necesarios de los... | Registrar, eliminar, modificar... | Pendiente |
| 02 | Los datos serán ingresados por... | Venta | Alta | Activo | Garantizar la seguridad de las información | Registrar, eliminar, modificar... | Pendiente |
| 03 | | | | | | | |
| 04 | | | | | | | |
| 05 | | | | | | | |



Buenas prácticas de software



Herramienta para hacer una matriz de trazabilidad.

Visure Requirements

Permite generar fácilmente una RTMs.

Algunas de sus características son:

Detalla y muestra elementos en cabeceras y filas de las columnas.

En cada celda indica los resultados finales de la trazabilidad.

Puedes hacer una prueba en la siguiente liga.

<https://visuresolutions.com/es/>



Buenas prácticas de software



4.1.3 Plan de pruebas.

Una buena practica es realizar un plan de prueba del sistema o software.

En el plan se debe:

1. Especificar puntualmente las cada una de las funcionalidades a probar.
2. Identificar la secuencias de ejecución de cada una de las pruebas.
3. Quién será el responsable de ejecutar las pruebas.
4. Hacer un cronograma para que se lleven a cabo las pruebas.
5. Se elabora con el fin de realizarse en cada interacción o a un conjunto determinado de casos o bien a todo el software.
6. Por otro lado se pueden definir jerarquías de casos de prueba a realizar tomando en cuenta la importancia.
7. Hay que tener muy clara la manera en la que vamos a hacer las prueba, como vamos a probar, de los recurso de los que vamos a disponer.



Buenas prácticas de software



Ejemplos de secciones de la plantilla para una planeación de pruebas.

Historial de versiones: Se indica la versión y la fecha.

Ficha del proyecto: Se indican los datos referentes al proyecto así como el responsable de las pruebas.

Pruebas realizadas: En este apartado se deben indicar las de pruebas que se realizaron o quedaron pendientes.

Aprobaciones: Aquí se indica quién aprueba tanto la planeación ejecución y resultados de las pruebas.

Ejecutor de las pruebas: Se debe especificar el profesional del servicio que hace las pruebas.

Resumen ejecutivo: Se debe elaborar un documento que contenga el plan de pruebas con cada uno de los apartados y el registro del resultado de las pruebas, dicho documento deben esta firmado por cada uno de los participantes en el plan y en las pruebas.



Buenas prácticas de software



Criterios de aceptación o rechazo:

En el plan de pruebas se deben especificar los criterios para dar por concluidas las pruebas.

Estos pueden ser:

Pruebas completadas: 100% de las estableciendo. el 100% de efectividad para considerarlas como casos exitosos. Por ejemplo considerar que se realizaron todos los tipos de pruebas, que se logro un 0% de errores o defectos corregidos.

Suspensión de la pruebas: Se debe definir o establecer con claridad los motivos para suspender la pruebas del sistema.

Reanudación de las pruebas: Cuando se suspendieron las pruebas por algún motivo, es importante establecer las pautas de reanudación. Por ejemplo después de ver dado solución a un mal funcionamiento de la función que ocasiono la suspensión de la prueba.

Entregables: Se debe entregar el documento de la planeación, los resultados, los reportes de errores si los hay, no se deben olvidar las evidencias de los resultados, así como la solución de los errores, entre otros.