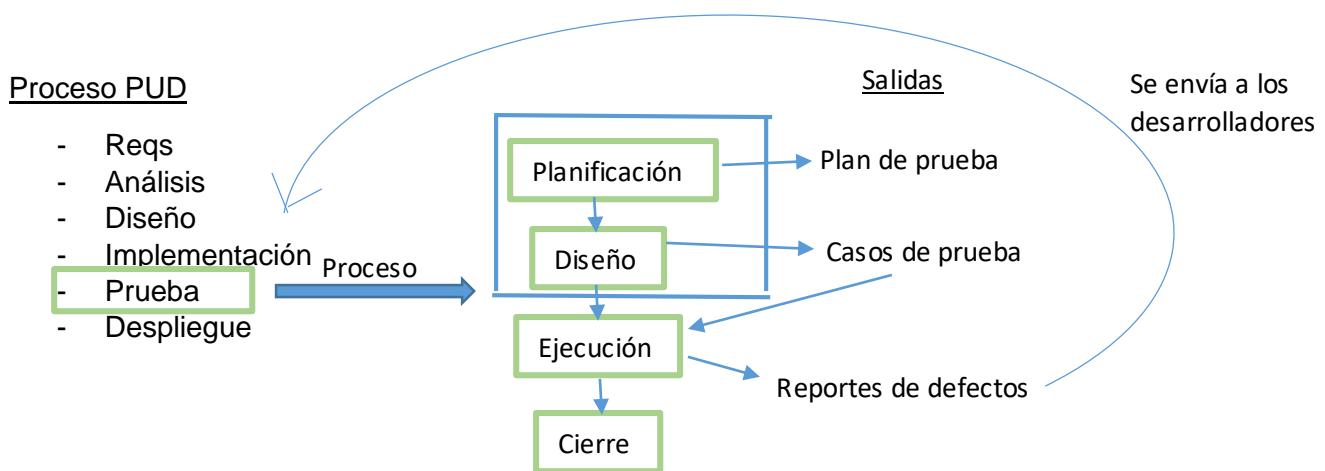


TESTING

Su propósito es encontrar defectos cuya presencia se asume en el desarrollo de SW



La planificación y el diseño se pueden adelantar (realizar antes del desarrollo). Suponen el 50% del esfuerzo. Se puede planificar y diseñar a partir de los reqs

Testing metodológico vs Adhoc

Testing Adhoc → no son reproducibles

Testing metodológico → Se realiza siguiendo un método o plan, usa los casos de prueba

La prueba realiza dos cosas:

- ➔ Validar → Proceso para comprobar que el SW satisface las necesidades del usuario. Son los más caros de corregir
- ➔ Verificar → Comprobar que el SW cumple con las especificaciones y se hizo según lo planificado

Error vs defecto

- Un error o defecto lo generamos nosotros que pueden producir una falla en el sistema
- El error no se traslada a otra etapa, en cambio, el defecto si es trasladado de etapas anteriores
- La edad del defecto se mide desde el tiempo que se detecta hasta que es corregido

El esfuerzo es lo que más peso tiene el desarrollo de SW donde las pruebas se llevan el 30-50% del mismo

Programación de pares: permite la detección inmediata de errores

Severidad del defecto: No tiene que ver con lo que me cuesta corregirlo sino con el impacto que produce el defecto en el sistema

Tipos:

- ➔ Bloqueante o invalidante → que no se puede utilizar el sistema
- ➔ Grave
- ➔ Leve
- ➔ Cosmética → Es atribuido a cuestiones de interacción de interfaz de usuario, errores de ortografía, entre otras cosas
- ➔ Mejora → No es un defecto, pero el teste ofrece una forma mejor de realizar la característica

La prioridad es definida por el cliente (Alta, media, baja)

Niveles de Testing

- ➔ 1er nivel – Pruebas unitarias: Verificar el correcto funcionamiento de cada unidad o componente del código (función, clase, módulo). (WF de implementación)
- ➔ 2do nivel – Integración o pruebas de interfaces: Verificar que los módulos o componentes funcionen correctamente al interactuar entre sí. Su resultado es una versión (WF de implementación o prueba)
- ➔ 3er nivel – Sistema: Validar que el sistema completo cumpla con los requisitos funcionales y no funcionales. (WF de prueba)
- ➔ 4to nivel – Prueba de aceptación de usuario: Confirmar que el sistema satisface las necesidades y expectativas del cliente/usuario final. (WF de despliegue)

Ambientes de SW

- ➔ Desarrollo → no tienen acceso a los ambientes de prueba
- ➔ Prueba → Pre-producción → pruebas de aceptación de usuario
- ➔ Producción

Lo ideal. En la mayoría solo hay de desarrollo, prueba y producción

Ciclos de prueba → Arranca desde que tengo una versión del producto hasta que se genera el reporte de defectos, el ciclo 0 es como se llama el primer ciclo de prueba

Estrategia de prueba

- ➔ Sin regresión → Se enfoca únicamente en verificar las nuevas funcionalidades o los cambios recientes, sin volver a ejecutar las pruebas anteriores del sistema.
- ➔ Con regresión → Incluye la ejecución de todas las pruebas anteriores además de las nuevas, para asegurar que las modificaciones no afecten funcionalidades previamente correctas.

¿Qué se prueba? → Que se cumplan con los requerimientos funcionales y no funcionales

Tipos de pruebas → en que se enfoca la prueba

- Requerimientos Funcionales Operacional Normal
 - Requerimientos Funcionales Negativa
 - Proceso de Negocio
 - Documentación
 - Manual de usuario
 - Manual de configuración
 - Pruebas de Performance
 - Pruebas de Interfaces
 - Pruebas de Escala completa
 - Pruebas de Stress
 - Pruebas de Seguridad
 - Pruebas de Accesibilidad
-
- ```
graph LR; A[• Requerimientos Funcionales Operacional Normal] --> B[• Requerimientos Funcionales Negativa]; A --> C[• Proceso de Negocio]; A --> D[• Documentación]; A --> E[• Pruebas de Performance]; A --> F[• Pruebas de Interfaces]; A --> G[• Pruebas de Escala completa]; A --> H[• Pruebas de Stress]; A --> I[• Pruebas de Seguridad]; A --> J[• Pruebas de Accesibilidad]; C --> D1[→ Manual de usuario]; C --> D2[→ Manual de configuración]; B --- D3{ }; D3 --- G; D3 --- H; D3 --- I; D3 --- J;
```