

A close-up, slightly blurred photograph of a workspace. In the foreground, a white ceramic cup is partially visible on the left. Behind it, a silver laptop is open. In the center, a spiral-bound notebook with lined pages lies flat. A black pencil rests horizontally across the notebook. A green paperclip and a green pushpin are also on the notebook. To the right of the notebook, a white ruler with black markings and the word 'Shatterproof' is visible. The background is a light-colored desk surface.

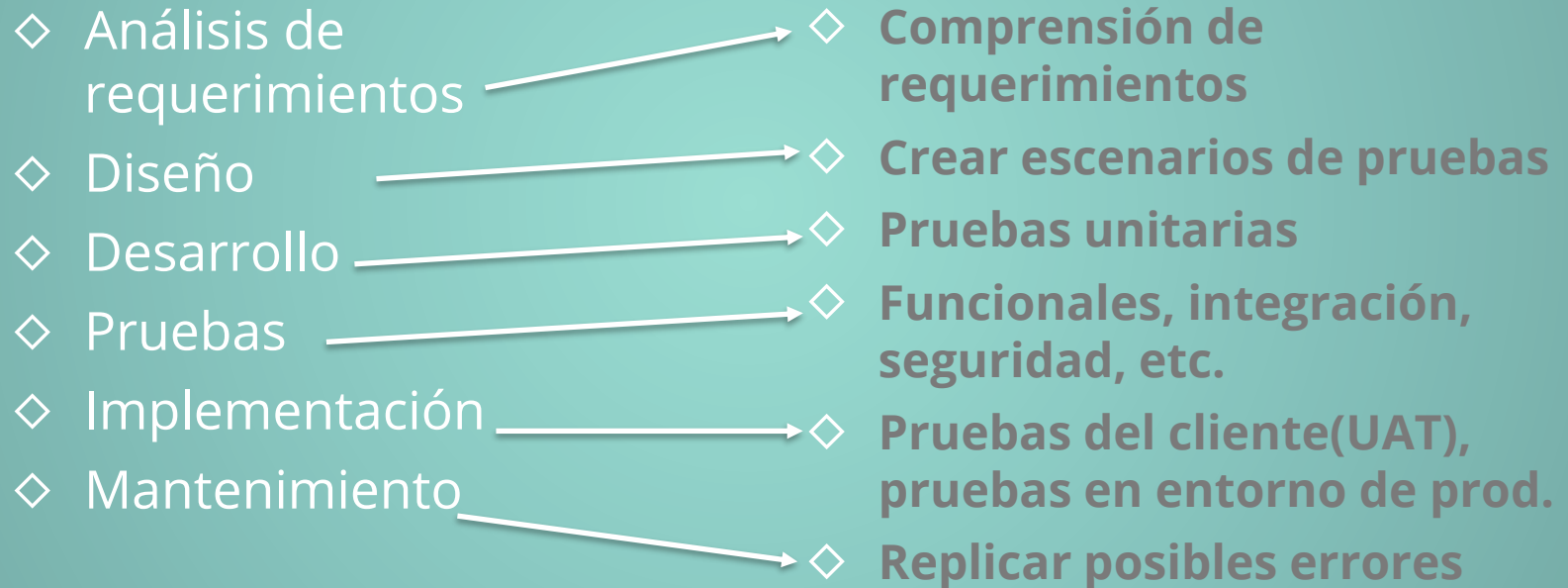
# Control de Calidad



¿QA sólo está en las pruebas?



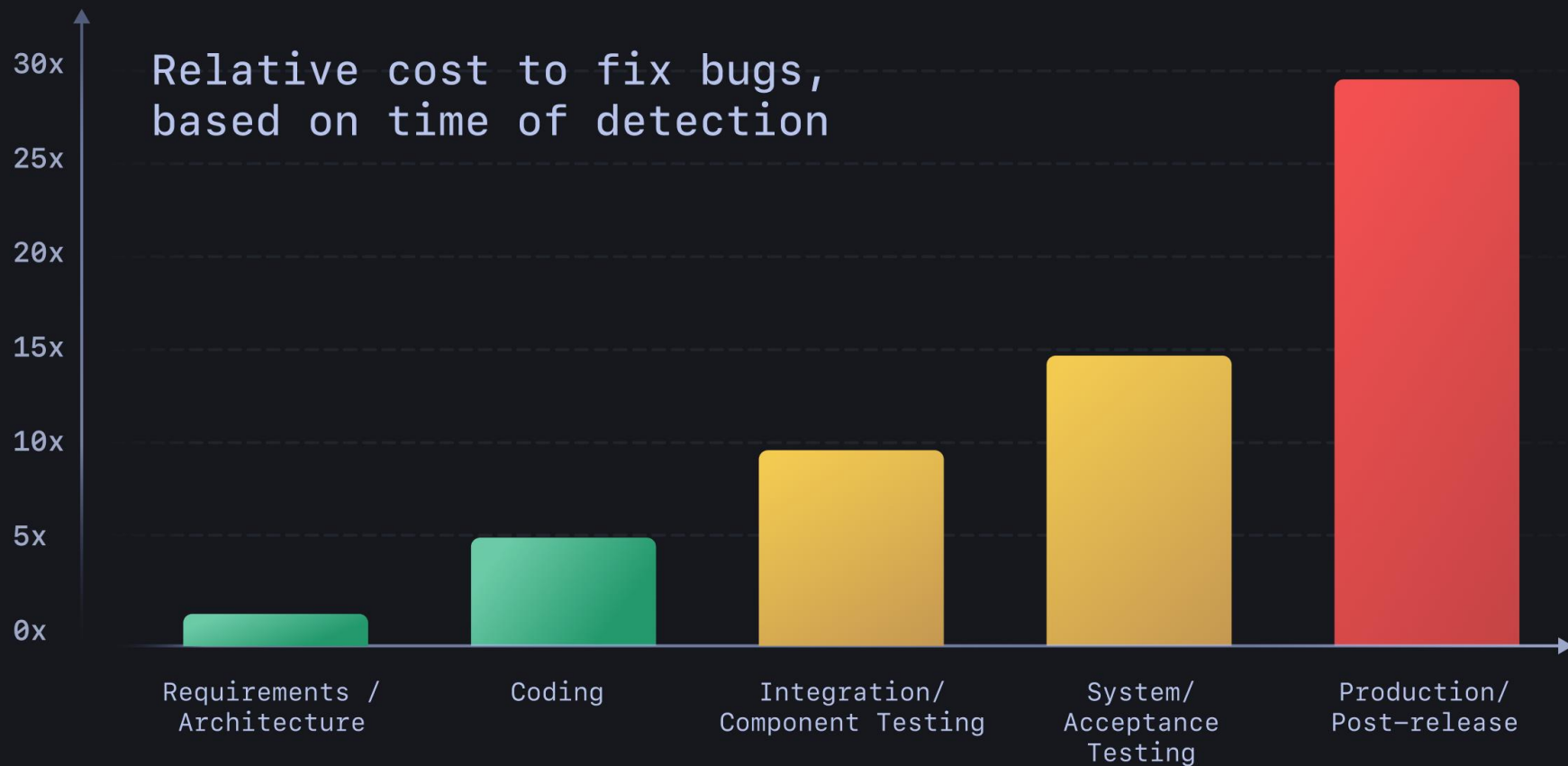
## SDLC





¿Por qué es importante?

2



## Validación vs Verificación

- ◇ Verificación: que el sistema de información funcione correctamente como fue diseñado.
- ◇ Validación: que el sistema de información cumpla con los requerimientos funcionales.

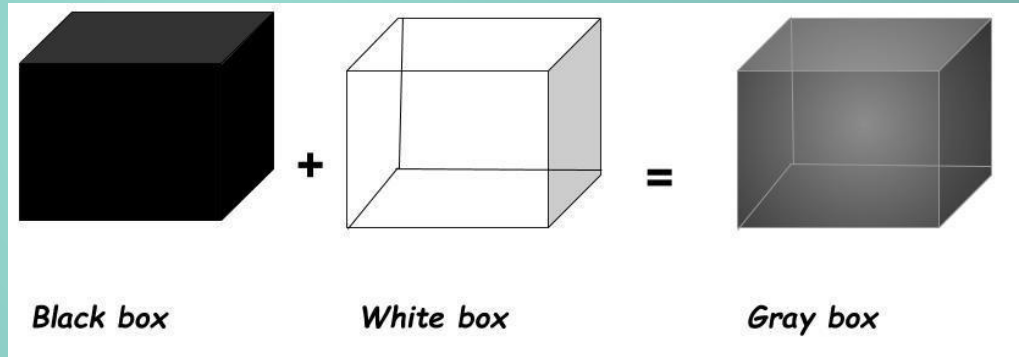


Tipos de pruebas

3

## Pruebas Funcionales

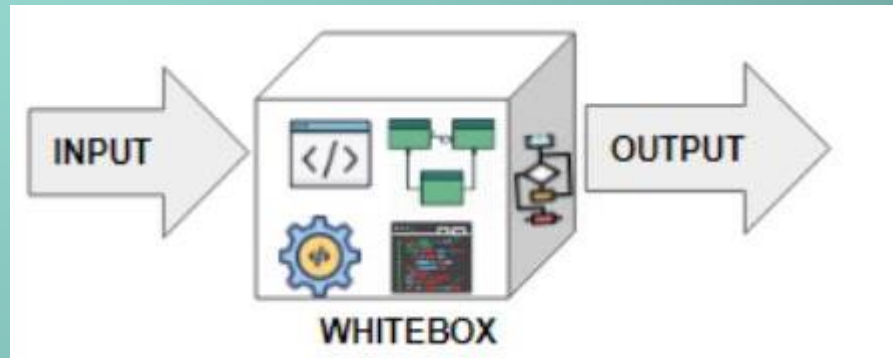
- ◇ Caja Blanca
- ◇ Caja Negra
- ◇ Caja Gris





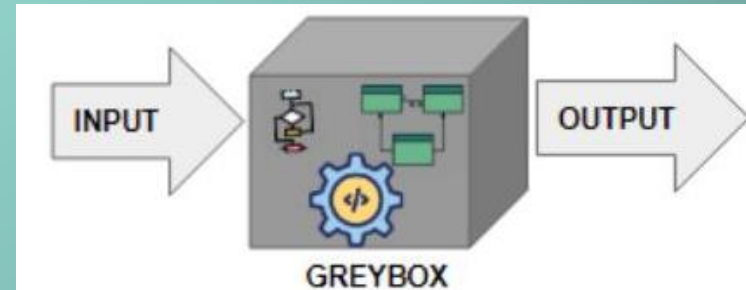
## Pruebas Caja blanca

- ◇ Pruebas unitarias
- ◇ Algunas pruebas de seguridad
- ◇ Verificación de calidad de código con ciertas herramientas



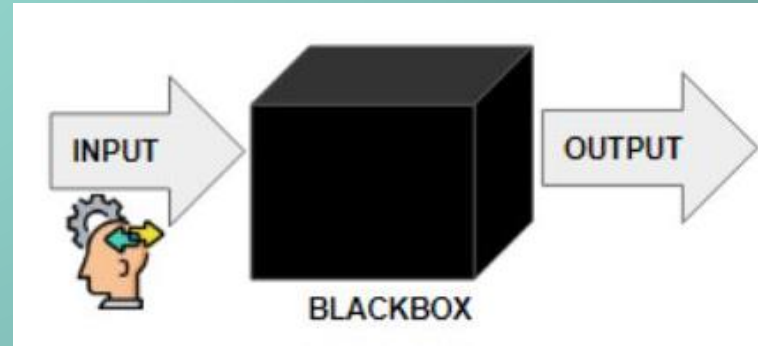
## Pruebas Caja Gris

- ◇ Pruebas funcionales de backend
- ◇ Algunas pruebas funcionales de front end
- ◇ Pruebas de integración
- ◇ Pruebas de carga
- ◇ Pruebas de estrés
- ◇ Scanners de seguridad



## Pruebas caja Negra

- ◇ Pruebas funcionales de front end(sin acceso a logs u otras herramientas)
- ◇ User Acceptance Testing
- ◇ Pruebas de experiencia de usuario
- ◇ Alfa
- ◇ Beta



## Pruebas Alfa

- ◇ Pruebas en presencia del equipo de desarrollo
- ◇ Pruebas por parte de algún representante del cliente
- ◇ Monitoreo constante
- ◇ El sistema no está del todo terminado



## Pruebas Beta

- ◇ Grupo de los usuarios o clientes finales
- ◇ El desarrollador no está presente
- ◇ Se crean logs muy detallados



## Pruebas de aceptación de usuario

- ◇ La aplicación está terminada pero antes de pasar a producción...
- ◇ El cliente inicia pruebas propias con data real
- ◇ Asegurar que información de producción funcione sea complatible



## Pruebas Funcionales

- ◇ Pruebas que tienen como objetivo probar tanto los escenarios esperados como negativos basados en los casos de uso



Las pruebas funcionales pueden:

- ◇ Probar la interfaz gráfica
- ◇ Probar backend

## FRONT END



## BACK END





## Pruebas de integración

- ◇ Verificar que diferentes servicios o módulos se acoplen correctamente
- ◇ Los tipos de datos deben coincidir



## Pruebas de experiencia de usuario

- ◇ Verificar qué tan intuitivo es el sistema
  - Controles fáciles de usar
  - Distribución lógica de elementos en pantalla
  - Instrucciones comprensibles





Pruebas automáticas



## Pruebas Unitarias

- ◇ Pruebas sobre métodos y funciones
- ◇ A nivel de código
- ◇ **Code Coverage**

✕Unit.net

TestNG

nunit

JUnit

## Backend automático

- ◇ Simular llamadas del sistema
- ◇ Asserts pero a nivel de respuestas



REST- Assured

GET, POST, PUT, PATCH, DELETE



SMARTBEAR

ReadyAPI



SoapUI



POSTMAN

## UI Automática

- ◇ Simular las acciones de un usuario



**Selenium**



**Playwright**



## Pruebas de carga

- ◇ Asegurar la cantidad de transacciones soportadas por el SW.



*NBomber*



POSTMAN

# Pruebas de estrés

## ◇ Cómo se comporta el HW con cierta carga

