

OTROS PATRONES Y ESTILOS DE ARQUITECTURA

TESTABLE ARCHITECTURE

Daniel Blanco Calviño

DESPLIEGUES RÁPIDOS Y FIABLES

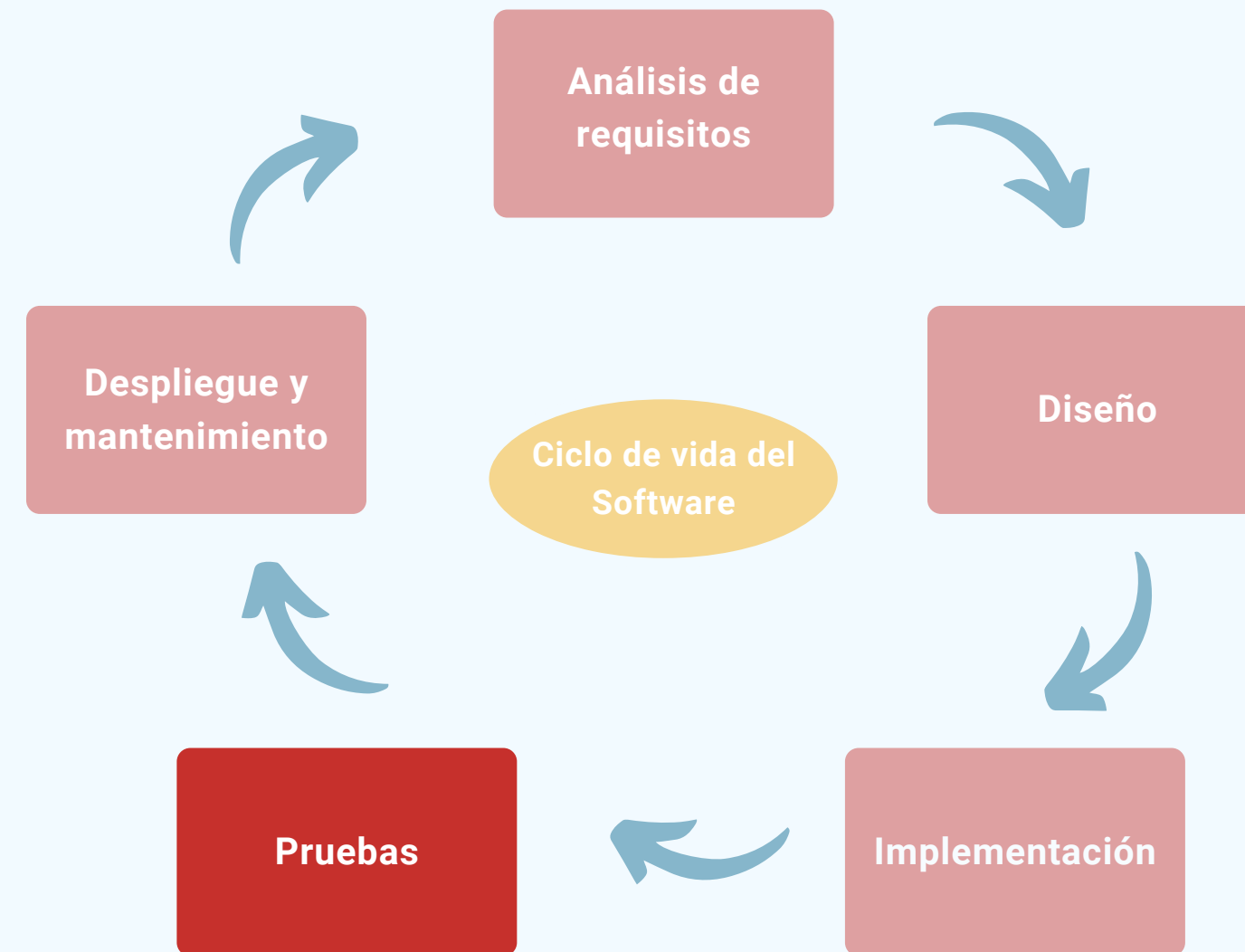


DESPLIEGUES RÁPIDOS Y FIABLES



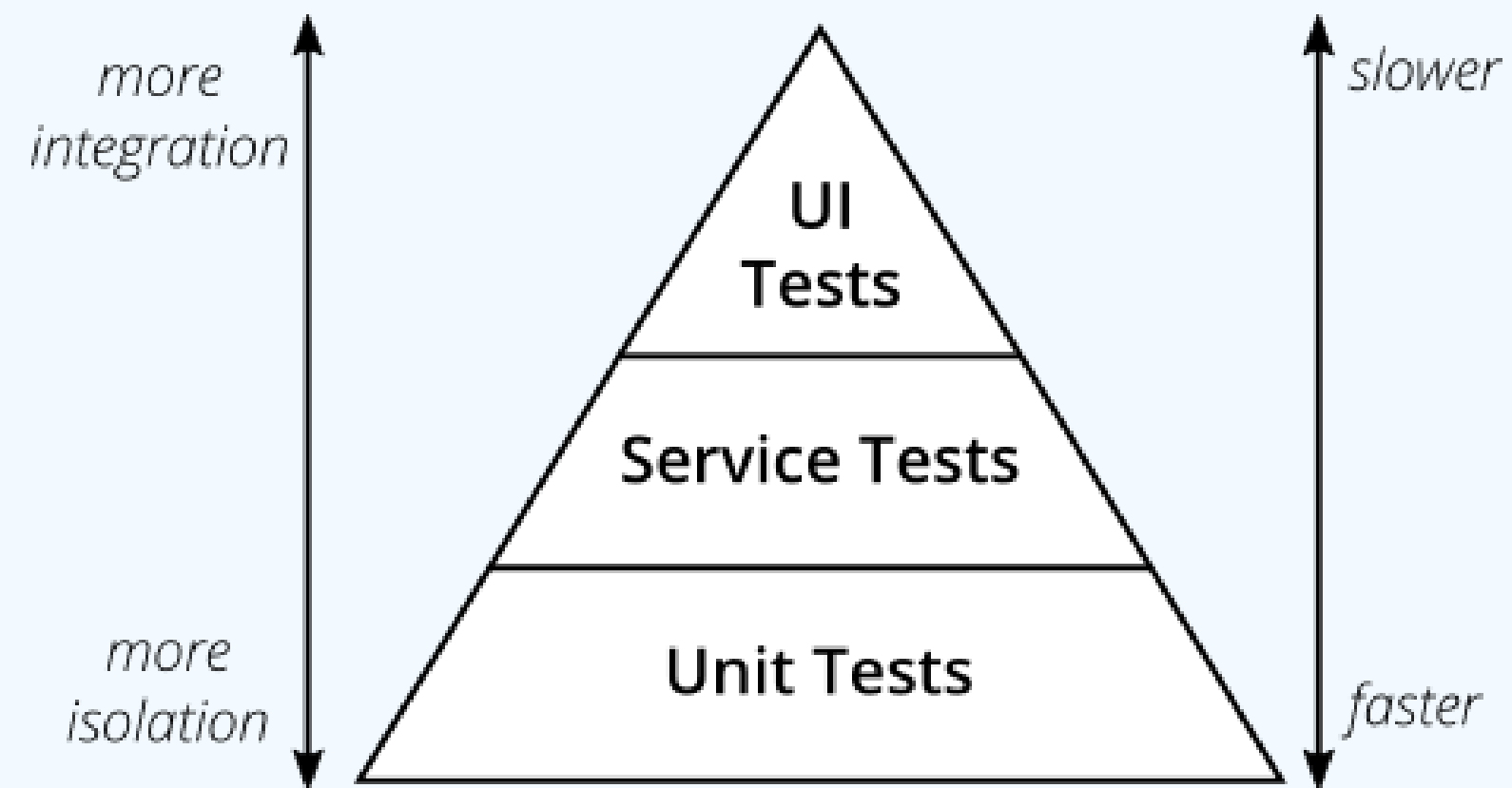
No basta con un despliegue rápido, **debemos hacer despliegues fiables!**

DESPLIEGUES RÁPIDOS Y FIABLES



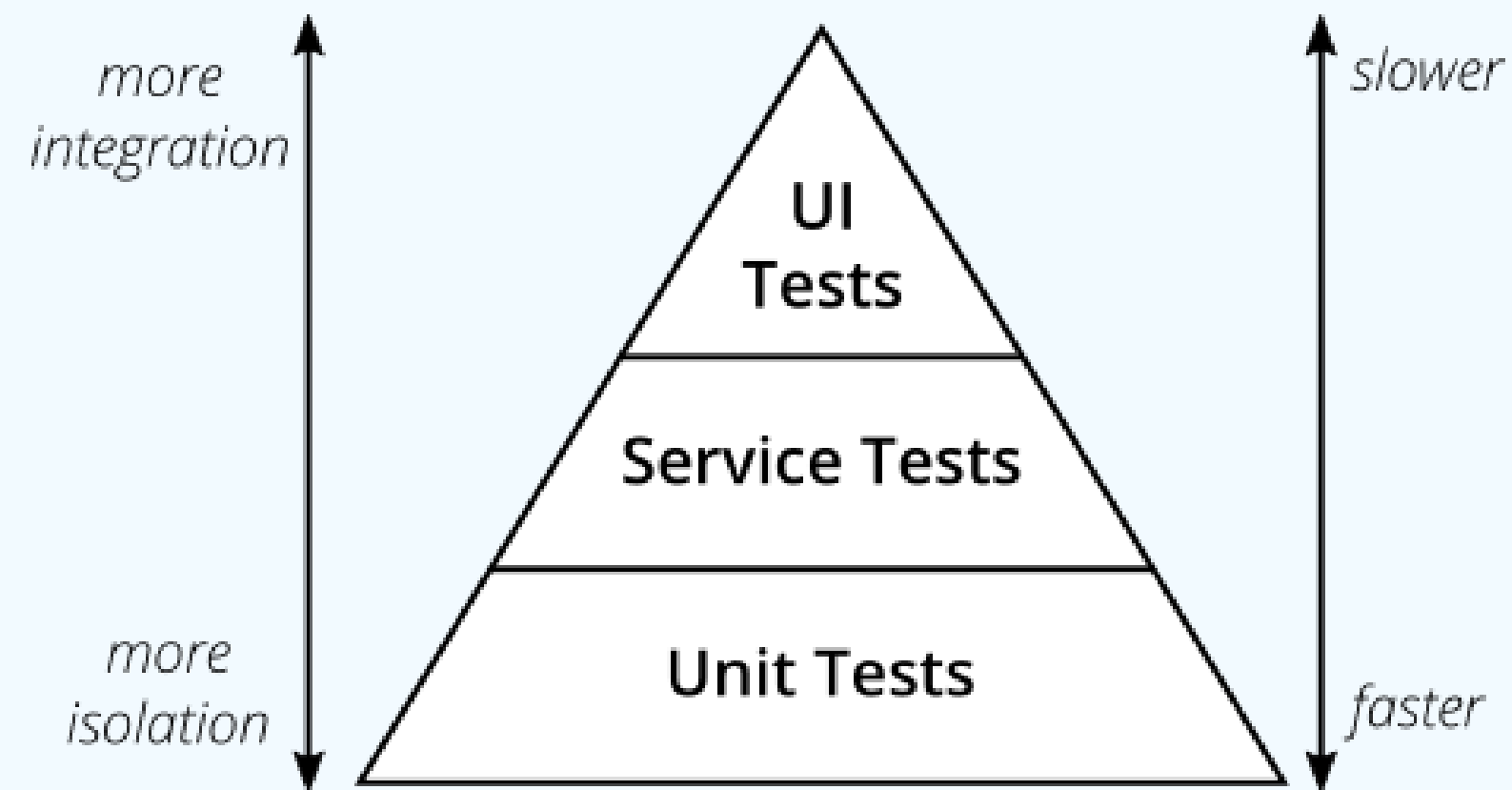
No basta con un despliegue rápido, **debemos hacer despliegues fiables!**

PIRÁMIDE DE PRUEBAS



Pirámide de pruebas de Mike Cohn

PIRÁMIDE DE PRUEBAS



Pirámide de pruebas de Mike Cohn



Esta pirámide se queda corta para el software moderno.

TESTS UNITARIOS

- **Objetivo:** Probar los métodos públicos de una clase de forma aislada.
- *Mockear* dependencias.
- Buscar una **cobertura alta** (+90%).
- Probar las distintas condiciones.

TESTS DE INTEGRACIÓN

- **Objetivo:** Probar tu software con BBDD o librerías externas.
- **Desplegar BBDD real** para los tests de integración.
 - Testcontainers, Docker.
- Probar el código que use librerías externas.

TESTS DE CONTRATO

- **Objetivo:** Comprobar que las interfaces entre dos servicios se mantienen correctas.
- Muy común en **arquitecturas de microservicios**.
- Se prueban ambas partes de la comunicación.
 - Productor.
 - Consumidor.

TESTS DE UI

- **Objetivo:** Comprobar que la interfaz de usuario se mantiene como esperamos.
- **Distinto de los tests de sistema o e2e.**
- Se puede probar la interfaz *mockeando* el backend, por ejemplo.
 - Angular, react, vue.
 - Jest, jasmine, mocha.

TESTS DE FINAL A FINAL (E2E)

- **Objetivo:** Probar todo nuestro sistema.
- **Pueden implicar tests de UI**, pero no tiene por qué.
- Se prueba el sistema desde el punto de entrada al lugar donde se reflejen los resultados.
- Selenium, cypress.

TESTS DE ACEPTACIÓN

- **Objetivo:** Comprobar las funcionalidades de nuestra aplicación.
- Se centran en los **casos de uso en concreto**.
- Sintaxis *given-when-then*.
- Herramientas BDD como Cucumber.

TESTS EXPLORATORIOS

- **Objetivo:** Probar manualmente el sistema en busca de problemas.
- **Ni los mejores tests automáticos son perfectos.**
 - Nos podemos olvidar de ciertos casos.
 - Bugs en nuestros tests.
- Probar manualmente **intentando romper la aplicación.**

OTROS PATRONES Y ESTILOS DE ARQUITECTURA

TESTABLE ARCHITECTURE

Daniel Blanco Calviño