Sopra Steria
TDD

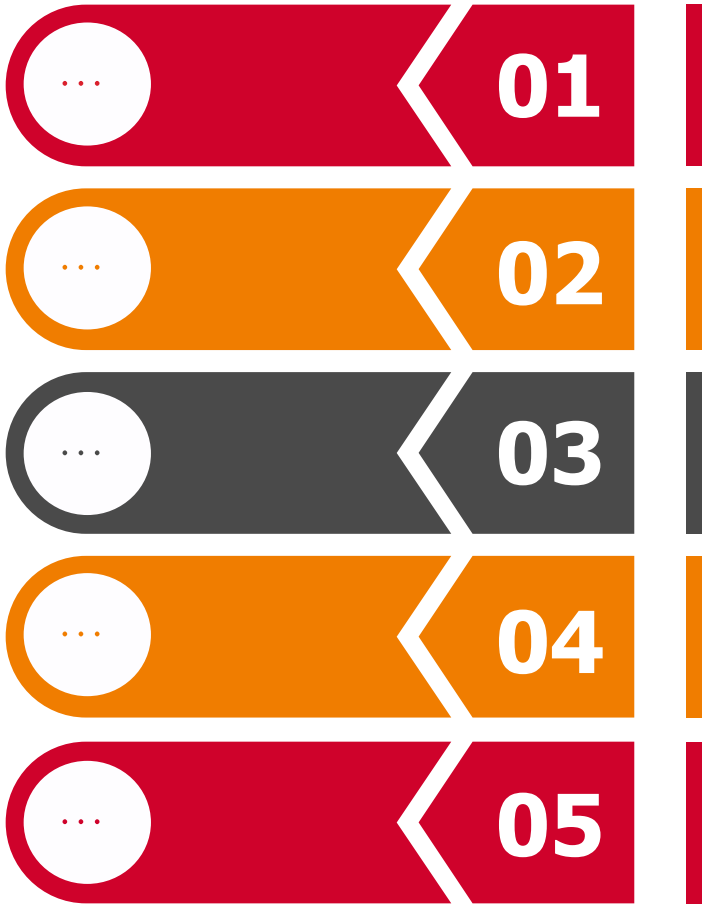
CIPFP MISLATA - **ENERO 2024**

The world is how we shape it

sopra steria

Mood tracker

Buenas prácticas



TDD - Test Driven Development
Esta práctica se basa en construir primero el test, el cual fallará de inicio, para a continuación desarrollar la funcionalidad que permitirá que el test pase los criterios establecidos.

Refactorización
La refactorización es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna para mejorar sus atributos no funcionales sin cambiar su comportamiento externo.

Peer review / code review
Se llama a "peer review" o "code review" a la revisión y aceptación del código producido por uno o varios compañeros para evaluar el contenido técnico y la calidad de la solución aportada.

Pair / mob programming
En el modelo de pair programming o mob programming se requiere la participación de varias personas para realizar el desarrollo. Se consigue una mayor cohesión en las soluciones aumentando la calidad del código, lo que limita la producción de malos diseños.

Medir y testear
Test funcional y de rendimiento. Metricas de calidad de codigo. Test e2e.

sopra steria

CODING_DOJO

- Un Coding Dojo es una reunión donde desarrolladores de software se juntan para trabajar en un reto de programación. Tienen que divertirse y enfocarse en una **práctica deliberada** para mejorar sus habilidades.

Premisas

- Adquirir habilidades en desarrollo es un proceso continuado

- **Características**

- No competitivo , colaboración, ambiente divertido y desenfadado
- Todos los niveles son bienvenidos
- Es seguro para probar ideas nuevas

- https://codingdojo.org/practices/WhatIsCodingDojo/

Extreme Programming

- **1970 – Winstone Royce** publicó el artículo que popularizó la metodología de desarrollo **Waterfall.**
- **1995 –** Expertos de software comenzaron a considerar un enfoque diferente **más incremental** en el desarrollo
  - **Scrum**
  - **Feature-driven development (FDD)**
- **1999 – KenBeck** published the book **Extreme Programming Explained.**

Extreme Programming (XP) es un conjunto concreto de prácticas bien definidas, enfocadas a **incrementar la calidad** y la **velocidad de respuesta al cambio** de requisitos del cliente.

En definitiva, las prácticas propuestas en XP sirven como guía para implementar los principios del Manifiesto Agile.

https://agilemanifesto.org/iso/es/principles.html

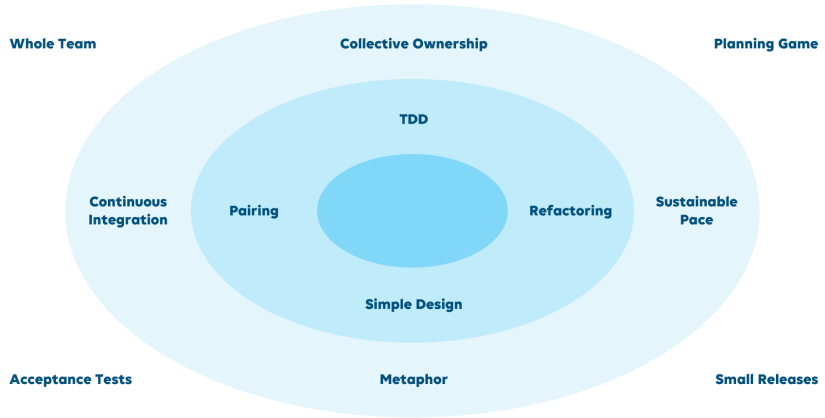
http://manifesto.softwarecraftsmanship.org/

Extreme Programming

The circle of life

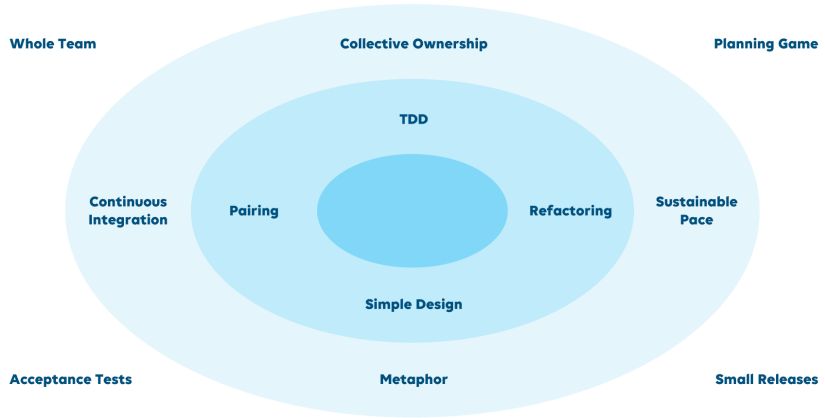
- El anillo interno representa las prácticas que ayudan a los desarrolladores a aegurar la calidad técnica.
- **Pairing** nos permite revisar, colaborar para compartir conocimiento y enfocarnos en la innovación y la precisión.
- **Simple Design** enseña al equipo a no hacer esfuerzos inútiles**.**
- **Refactoring** anima a la mejora continua y al refinamiento.
- **Test-Driven Development** es la red de seguridad que permite al equipo a mantener la velocidad y la calidad.

- https://community.dataminer.services/extreme-programming/



XP

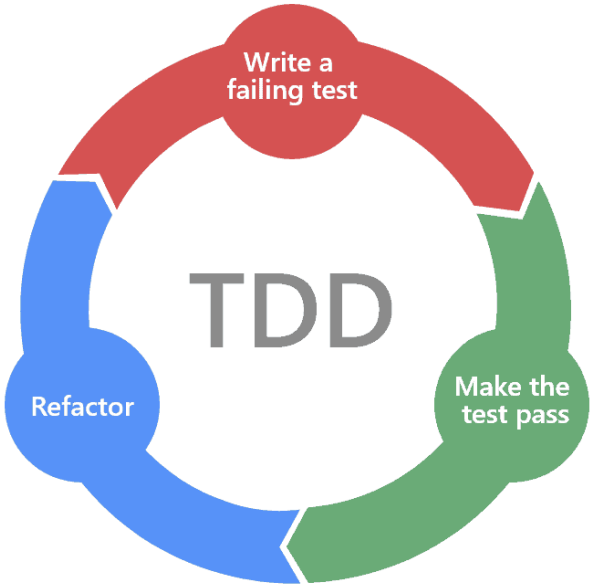
- **TDD** es una disciplina que dirige el proceso de desarrollo
- **Refactoring nos permite escribir código limpio, es difícil alcanzarlo sin TDD.**
- **Simple design es casi imposible sin refactoring**
- **Collaborative Programming**
  - Pair programming, mob programming
  - Code reviews
  - Brainstorms
- **Acceptance Tests**
  - Negocio propone los comportamientos
  - Estos comportamientos son codificados
  - Los test deben poder ser escritos y entendidos por la gente de negocio.



TDD

The Three Laws

- **Crea una suite de test que permita refactoring, si los tests pasan el sistema puede ser desplegado.**
  - No escribas código hasta que no ha sido primero escrito un test y halla fallado.
- **Haz código que esté desacoplado para poder ser testeable y refactorizable.**
  - No escribas más código que el resuelva el test actual, que estará fallando.
  - Primero hazlo funcionar y después déjalo perfecto.
  - First make it works. Then make it right (Kent Beck's)
- **Mantén un corto loop de feedback que mantenga estable el ritmo y la productividad del desarrollo de tu código.**



sopra steria

TDD

Beneficios

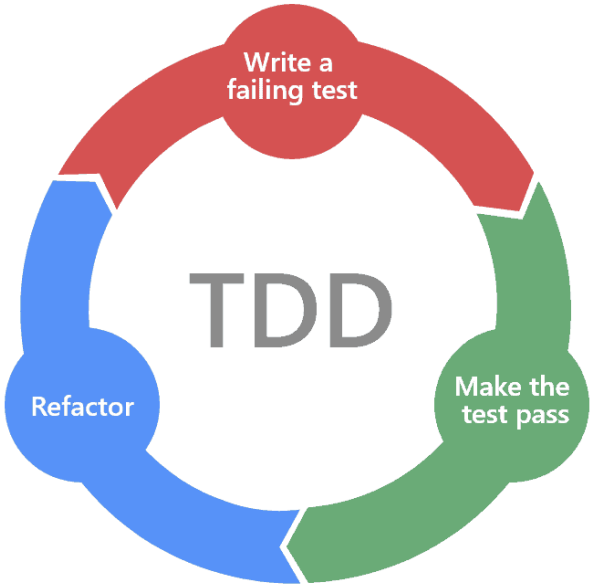
- Pasarás más tiempo escribiendo código que funciona y menos tiempo depurando código que no funciona.

- Producirás un conjunto de documentación de bajo nivel casi perfecto.

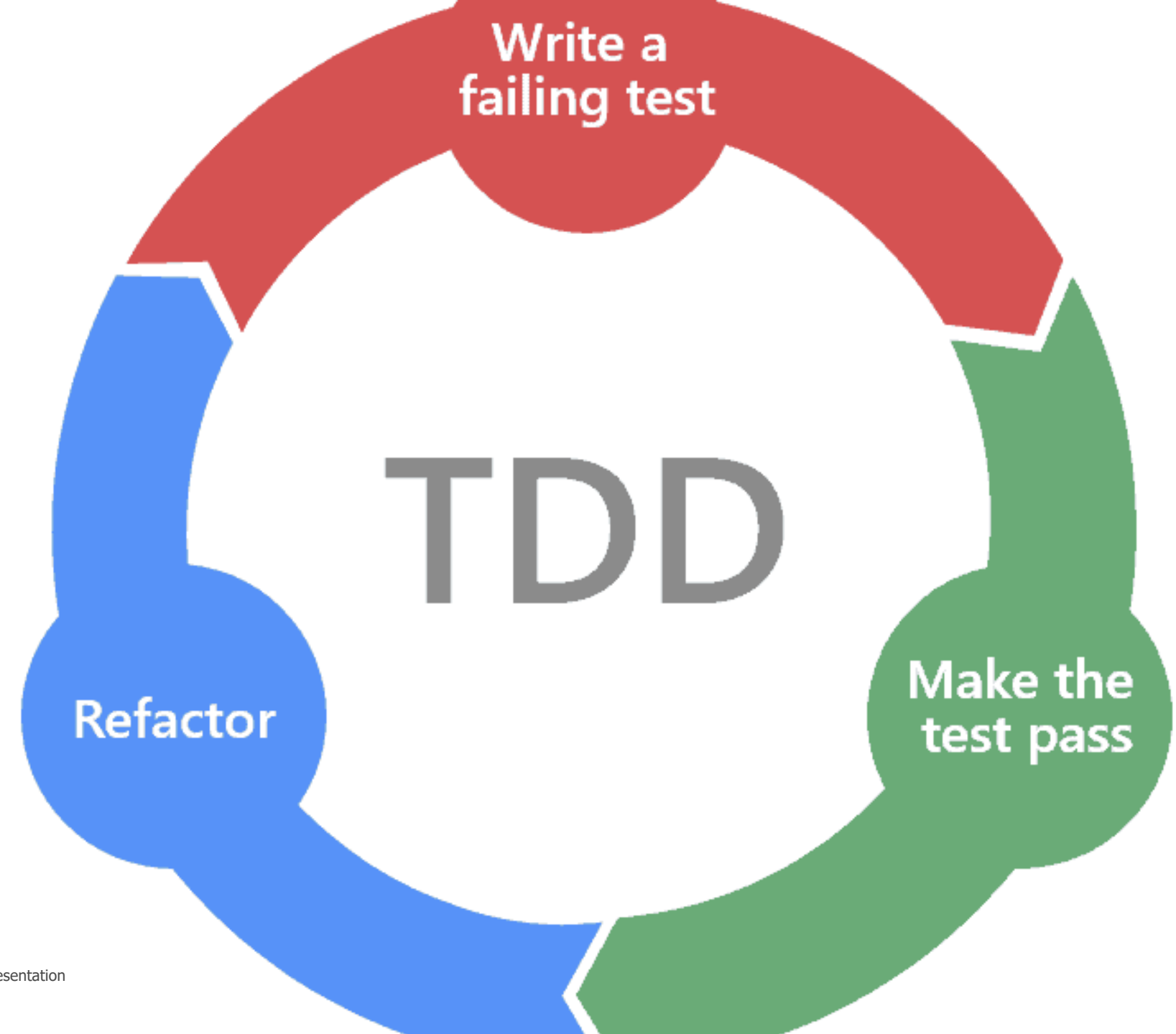
- Es divertido, o al menos motivador.

- Producirás un conjunto de pruebas que le dará la confianza para desarrollar funcionalidades.

- Crearás diseños menos acoplados.



sopra steria


Write a
failing test
TDD
Make the
test pass
Refactor

Referencias

- **Clean Craftsmanship: Disciplines, Standards, and Ethics (Robert C. Martin)**
- **A philosophy of software Design (John Ousterhout)**
- https://codingdojo.org
- https://agilemanifesto.org/iso/es/principles.html
- http://manifesto.softwarecraftsmanship.org/
- https://community.dataminer.services/extreme-programming/
- https://aulasoftwarelibre.github.io/taller-de-python/Testing/TDD/

The world is how we shape it

GRACIAS

Sopra Steria España

sopra steria