



IL CORSO È FINITO...

...E ADESSO!?

VALERIO LAVIO

INTRODUZIONE



SEMPRE I SOLITI PROBLEMI

- Debito tecnico
- Requisiti non chiari
- Design inefficiente
- Paura del refactoring



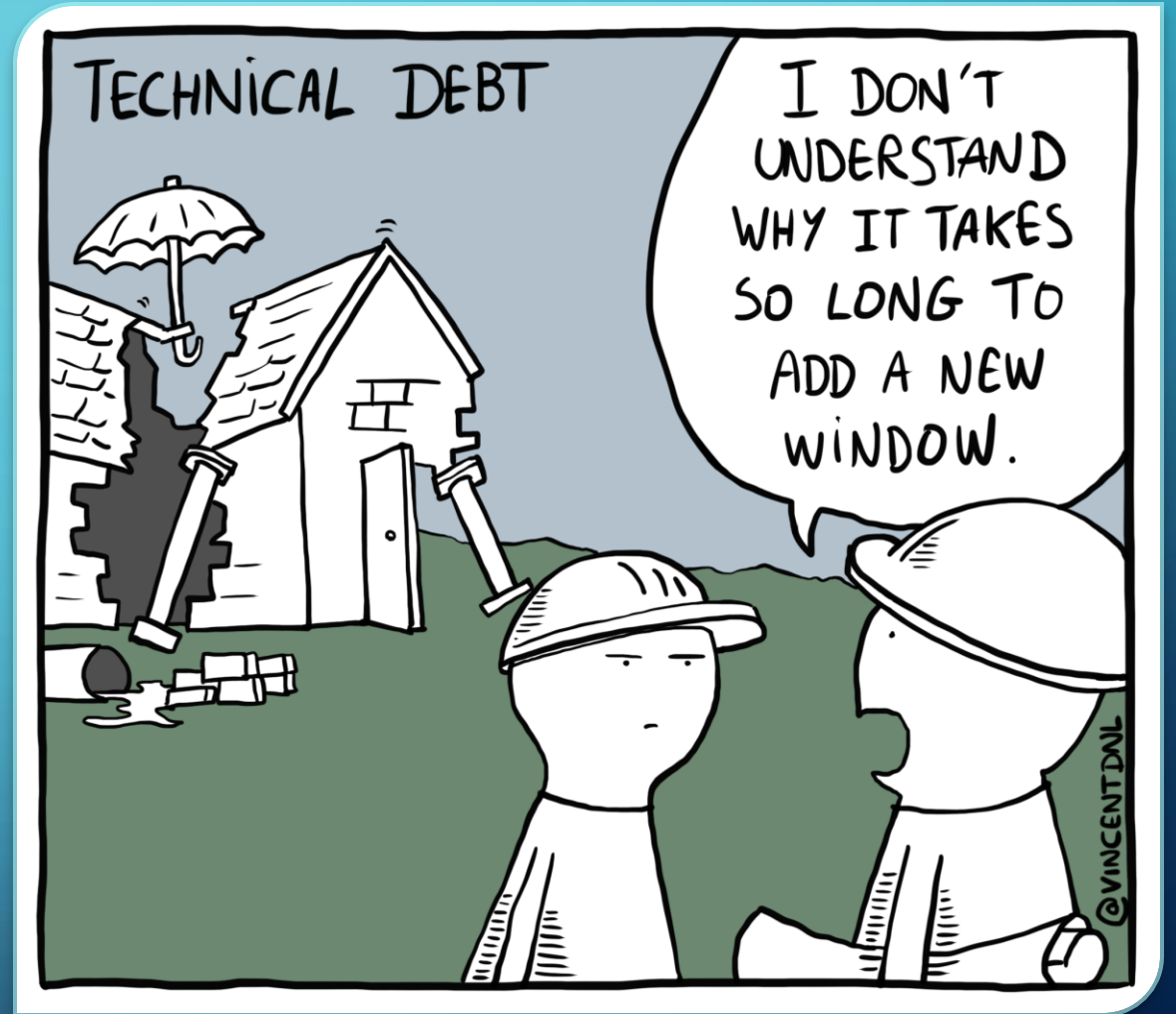
COME POSSIAMO MIGLIORARE?

- TDD
- Occhio ai Code Smells
- Object calisthenics
- SOLID
- Simple design



E POI?

- Refactoring continuo!



ALCUNI CONCETTI CHIAVE

- Non cercare la perfezione
- Scriviamo codice ovvio
- Andiamo in PROD il prima possibile con il walking skeleton

When I wrote this code,
only God & I understood what it did.

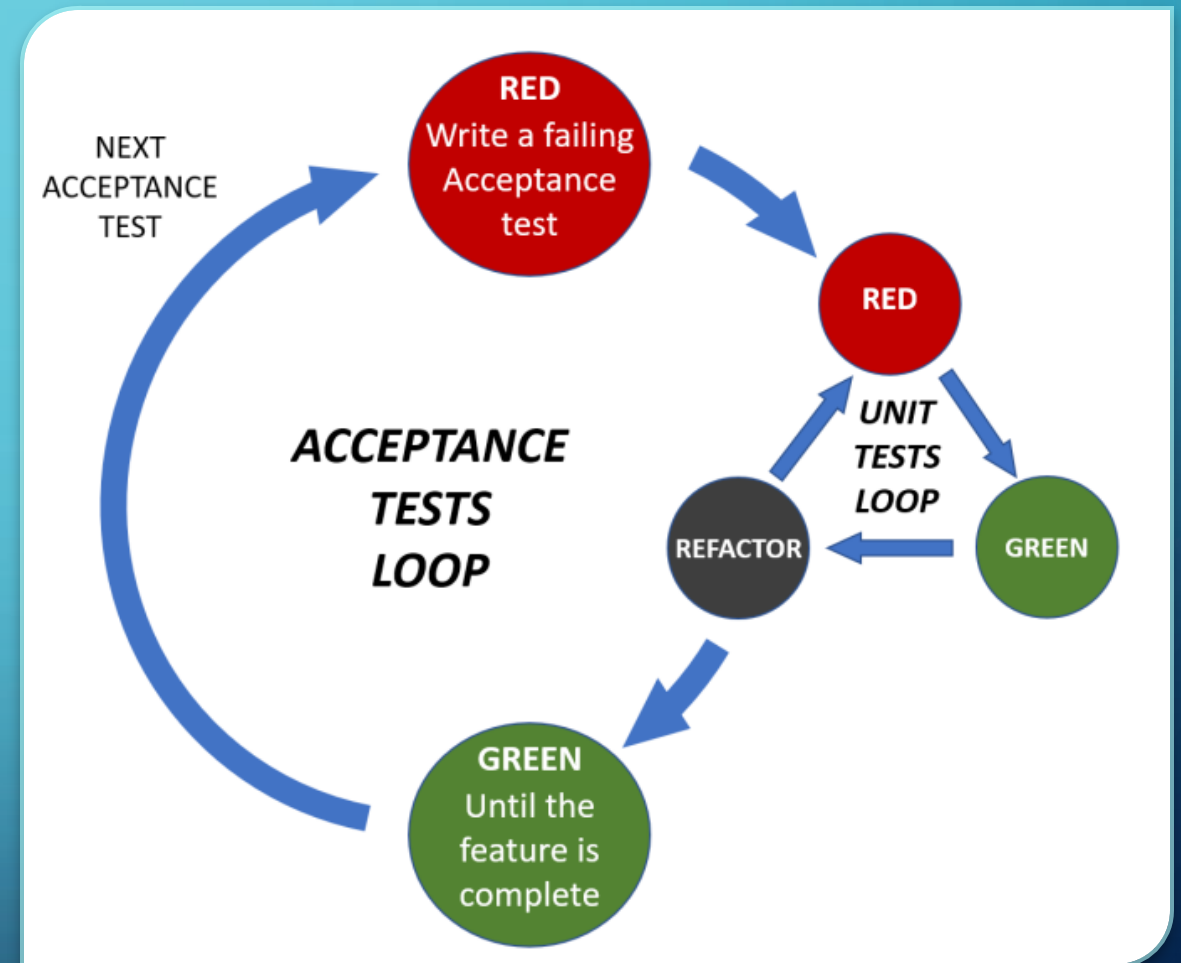
@dino_codes



Now...
only God knows.

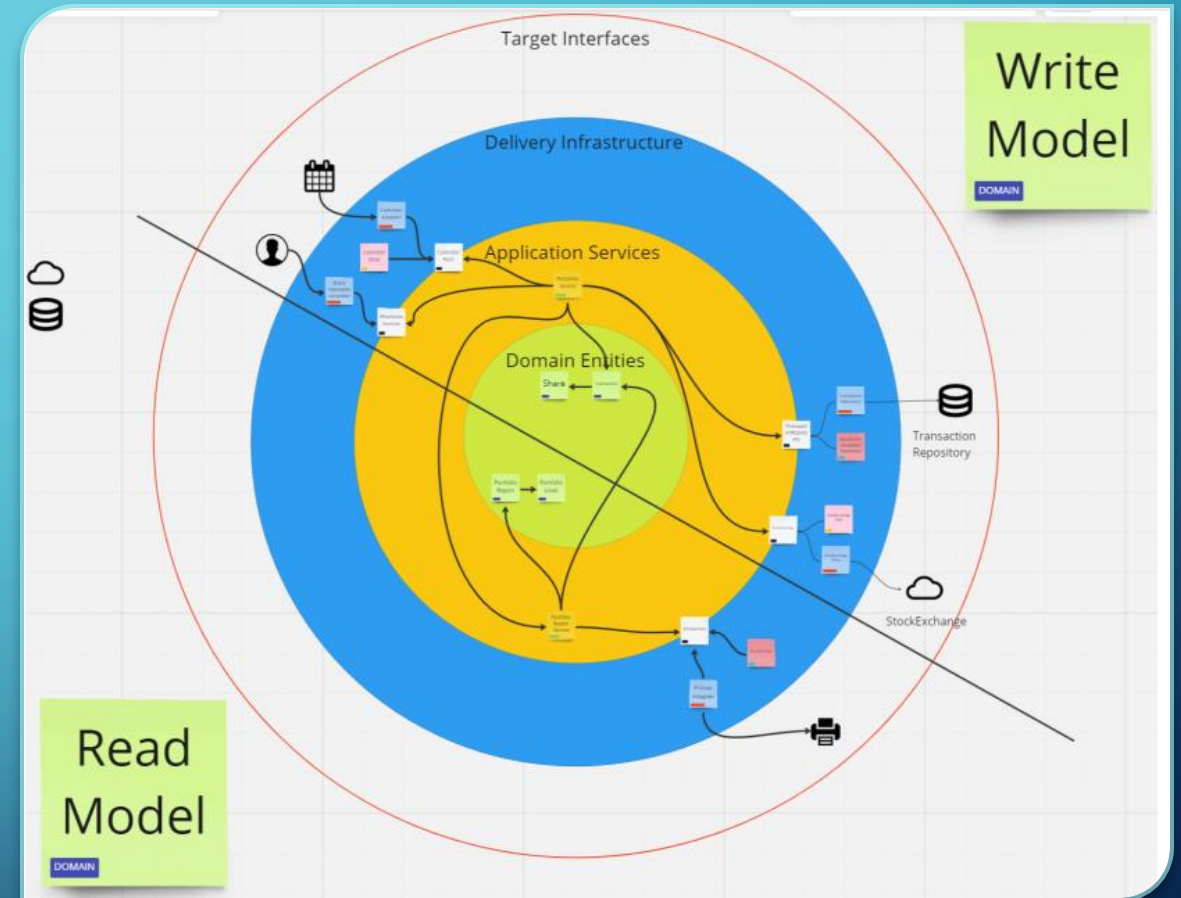
OLTRE IL CODICE

- Collaboriamo con il business per definire i requisiti
- Scriviamo test di accettazione: da TDD a ATDD



OUTSIDE-IN

- Con gli acceptance test facciamo un po' di design
- Da alto livello a basso livello
- Il core non conosce i dettagli del esterno
- Dependency inversion
 - Mettiamo delle interfacce verso l'esterno (porte)
 - => Test doubles!



DOPO IL CORSO

- Continuare ad allenarsi
- Migliorare le code review (vedi: Non Violent Code Review)
- + Pair Programming / – Single Responsibility all'interno dei team

**Learn the rules like a pro,
so you can break them
like an artist.**

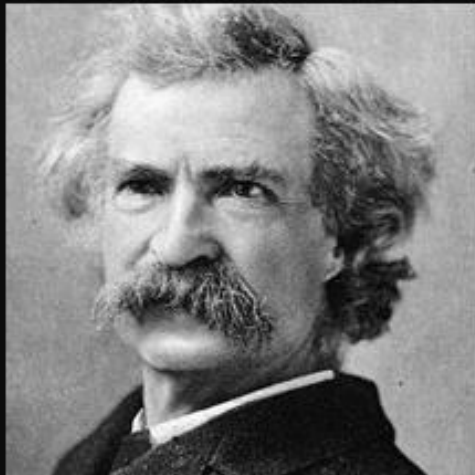
PABLO PICASSO

IN CONCLUSIONE

**IF YOU CAN'T DO GREAT
THINGS, DO SMALL THINGS
IN A GREAT WAY.**

- NAPOLEON HILL

DOMANDE?



He who asks is a fool for five minutes, but he who does not ask remains a fool forever.

~ Mark Twain

GRAZIE PER L'ATTENZIONE



<https://www.linkedin.com/in/valerio-lavio-38815916a/>



<https://github.com/vale247>