# Test Driven Development



## Index

- Introducció
- Definició



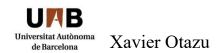
## Index

- Introducció
- Definició



#### Introducció

- Què és un test?
  - Des del punt de vista del testejador
    - Troba errors
  - Des del punt de vista del programador
    - Ens assegura que el codi és correcte (definició no estrictament correcta!!!)
- Per tant, un test també es pot considerar ... ?
  - Un requeriment, un exemple i fins i tot una part del disseny
    - ⇒ Han de ser validats!



#### Introducció

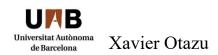
- El nom estrictament correcte seria *Example Driven Development*.
  - Desenvolupem a partir de la definició dels tests/exemples.
- En la pràctica:
  - No es tracta tant de fer testos, sinó de dissenyar segons els requeriments.
  - Passem de pensar en implementar tasques a pensar exemples adequats que eliminin qualsevol possible ambigüitat.
  - Dividim els casos d'ús en n exemples fins que aquest nombre sigui suficient per descriure la tasca sense ambigüitats
  - La implementació d'aquests petits exemples ens pot ajudar a que l'arquitectura local de certs moduls sorgeixi per si mateixa.

Ens preguntem continuament "Què volem que faci el codi?"

#### Introducció

#### Avantatges del TDD

- Software d'alta qualitat
- Incrementa la productivitat
- Codi altament reutilitzable
- En el codi, escriurem just la funcionalitat necessària sense sobredissenyar (ja que escribim el test abans que el codi)
- Treball en equip més fàcil
- Millora la comunicació



## Index

- Introducció
- Definició



#### Definició

#### TDD té tres passos

- Escriure l'especificació del requeriment (exemple, test)
  - Cal pensar en com seria la API (signatura) del SUT (Subject Under Test).
- Implementar el codi segons aquest exemple
  - Implementar el mínim necessari per a que el test es compleixi (això ens fa pensar en altres situacions, inputs, **fluxes d'execució** ... ).
- Refactoritzar per eliminar duplicitat de codi i fer millores.
  - Eliminar codi duplicat, millorar el disseny, fer el codi més reutilitzable, més llegible, etc ...

## Bibliografia

- Diseño agil con TDD. Carlos Ble Jurado y colaboradores Accessible a <u>www.freelibros.com</u>
- Test Driven Development: by Example. Kent Beck. Addison-Wesley Professional, 2002.