

Arrivato al traguardo... ma é solo l'inizio?

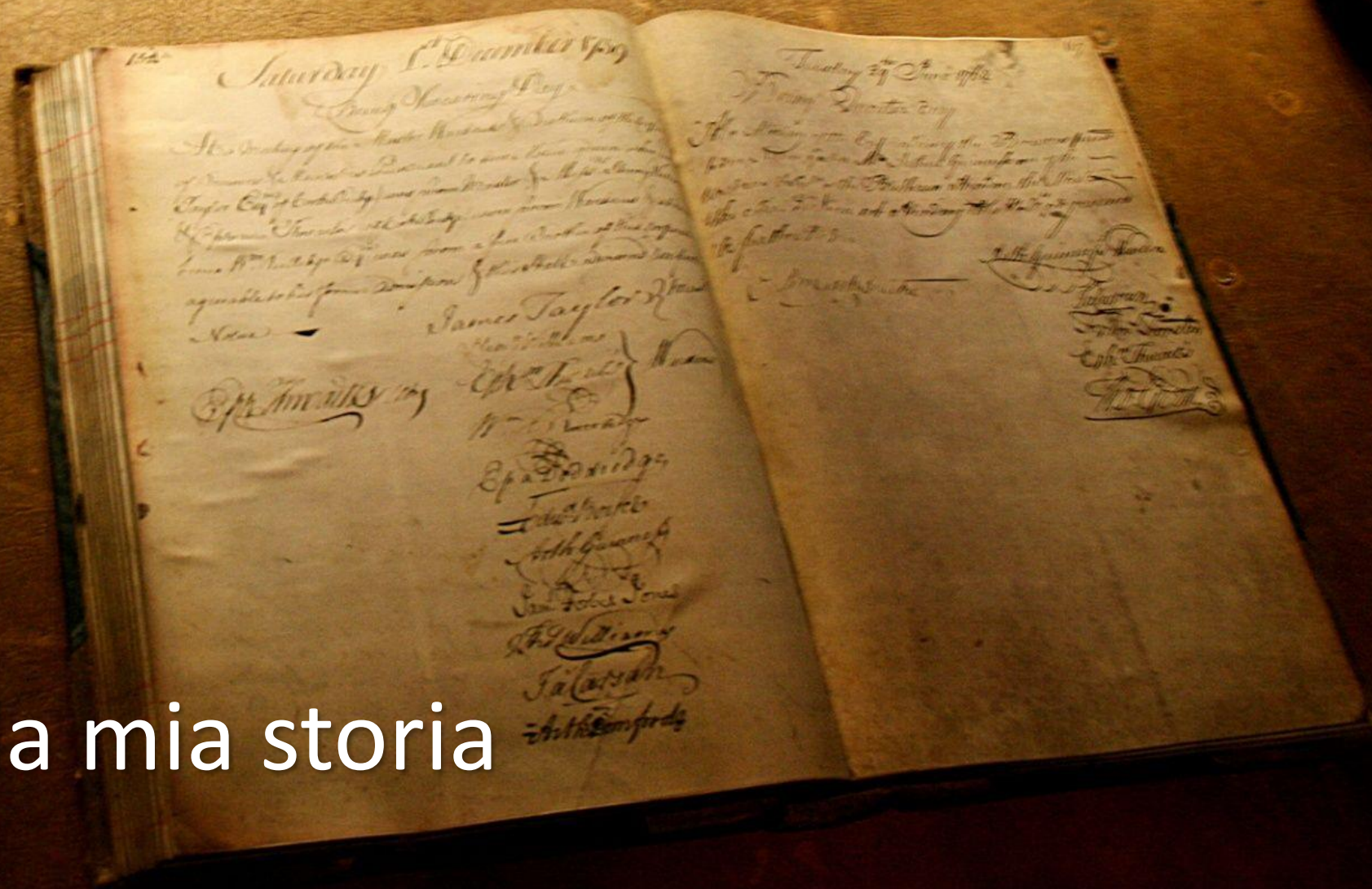
Flying

Stefano Mondini

Novembre 2022

Introduzione

Oggi racconto la mia storia



C'era una volta...

...un programmatore, con esperienza in generale ma non particolarmente forte sul testing...

Iniziamo

Molto motivato

Guidato da esperti

Posso usarlo da subito



START

Primo incontro

Introduzione



Bella gente

Pronto per la sfida

Primo incontro

Mob programming?

Come interagire?

E il gruppo?



A close-up, slightly blurred photograph of a sloth resting its head on a laptop keyboard. The sloth's long, shaggy brown fur is prominent on the left side of the frame. Its head is positioned over the keyboard, with its snout and one eye visible. The background shows a laptop screen and a mouse, both out of focus. The overall lighting is soft and natural, suggesting an indoor setting near a window.

Prime difficoltà

So programmare ancora?

Vado troppo lento o veloce?

Cosa penseranno gli altri?

Fine primo impact

Fiducia un po' in calo.

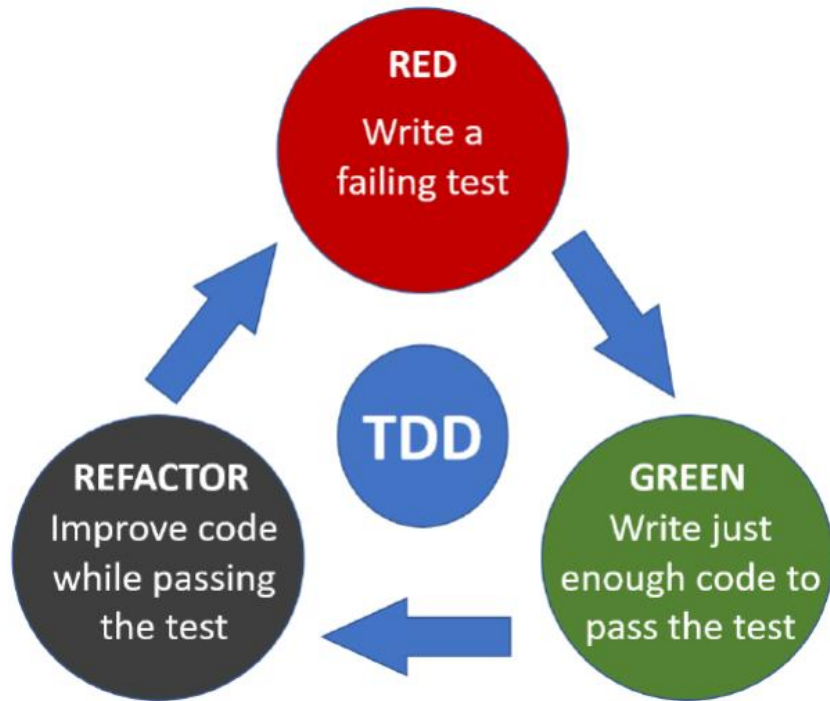
Meglio la prossima volta?

Proviamoci!

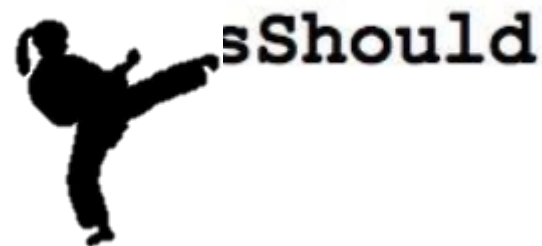
A person with a backpack is walking away from the camera on a wooden boardwalk of a suspension bridge. The bridge has ornate metal railings and a large stone archway in the distance. A city skyline is visible in the background under a cloudy sky. The word "Walking" is overlaid in large orange letters.

Walking

Walking



WE TEST
BEHAVIOUR
NOT
IMPLEMENTATION



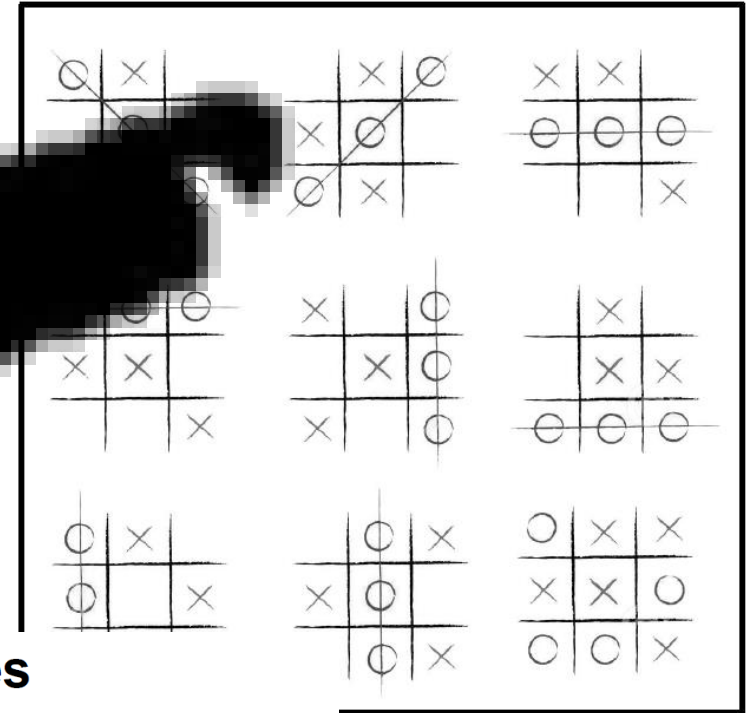
Walking - oh

Transformation Priority Premise - w

#	TRANSFORMATION	STARTING CODE
1	<code>{}</code> => <code>nil</code>	<code>return nil</code>
2	<code>nil</code> => <code>constant</code>	<code>return "1"</code>
3	<code>constant</code> => <code>constant+</code>	<code>return "1" + "2"</code>
4	<code>constant</code> => <code>scalar</code>	<code>return argument</code>
5	<code>statement</code> => <code>statements</code>	<code>return arguments</code>
6	<code>unconditional</code> => <code>conditional</code>	<code>if(condition) return</code>
7	<code>scalar</code> => <code>array</code>	<code>[dog, cat]</code>
8	<code>array</code> => <code>container</code>	<code>[dog, cat]</code>
9	<code>statement</code> => <code>recursion</code>	<code>a + b</code>
10	<code>conditional</code> => <code>loop</code>	<code>if(condition)</code>
11	<code>recursion</code> => <code>tail recursion</code>	<code>a + recursion</code>
12	<code>expression</code> => <code>function</code>	<code>today - birthday</code>
13	<code>variable</code> => <code>mutation</code>	<code>day</code>
14	<code>switch case</code>	

Calisthenics rules

1. Only one representation per method
2. Don't use `var` keyword
3. Wrap arrays and strings
4. First one wins (wrap all collections)
5. Only one `dog.Body.Tail.Wag()` => `dog.ExpressHappiness()`
6. No abbreviations
7. Keep all methods short
[10 files per package, 10 lines per class, 5 lines per method, 2 arguments per method]
8. No classes with more than two instance variables
9. No public getters/setters/properties
10. All classes must have a `main`



Sembrava così facile...

...ma ovviamente no

Trovare il tempo

Organizzarsi

$$\left(\frac{4K(1+K)}{p_{se}} p_{se} \right)$$

$$21 = \frac{1}{(A + \eta)^2 + \kappa^2} \text{ and}$$

Prima presentazione

Argomento: ok!

Preparo una presentazione tecnica

Ciclo TDD

Prima presentazione – finita!



Risultato

Un po' "basic"

Non personale

A still from the animated movie Shrek. Shrek is shown from the waist up, with his arms outstretched to the sides. He has a wide, toothy grin and is looking directly at the camera. He is wearing his signature green tunic and a brown belt. The background is a blurred outdoor setting with trees and a cloudy sky.

Ehm... ok



Running

Seconda parte



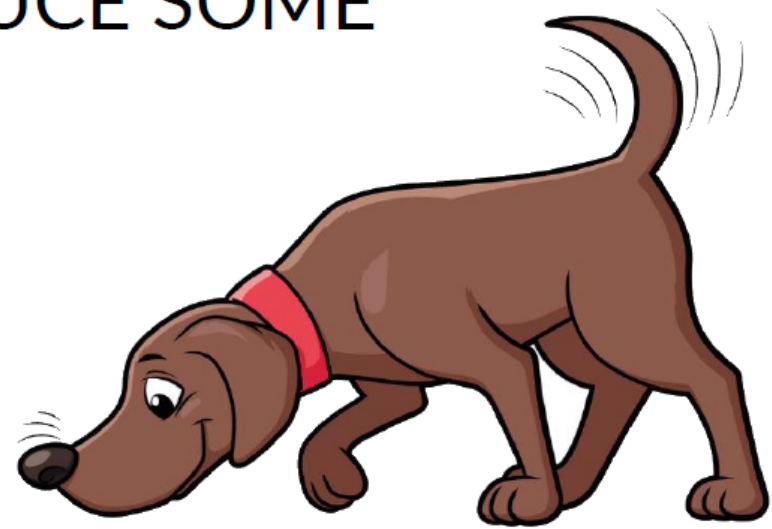
IDE: é nostro amico

- **Rename** – Change the name of classes, methods, variables....
- **Extract** – Extract a class (or methods or variables...), creating a new abstraction.
- **Inline** – The inverse of extract – inline a method (or variable), deconstructing an abstraction.
- **Move** – Move a class (or methods or variables...) to some other place in the codebase.
- **Safe delete** – Delete code and its usages in the code base.

Code smells

WE ARE GOING TO INTRODUCE SOME

**CODE
SMELL**



Ok....?

Code smells CHEATSHEET

Switch Statements OOA Replace Conditional with Polymorphism Replace Type Code with Subclasses Replace Type Code with State/Strategy Move Accumulation to Visitor Replace Conditional Dispatcher with Command Replace Parameter with Explicit Methods Introduce Null Object	Inappropriate Intimacy COU Move Method Move Field Extract Class Hide Delegate Replace Inheritance with Delegation	Large Class BLO Extract Class Extract Subclass Extract Interface Replace Data Value with Object Replace Conditional Dispatcher with Command Replace Implicit Language with Interpreter Replace State-Altering Conditionals with State
Primitive Obsession BLO Replace Data Value with Object Encapsulate Composite with Builder Introduce Parameter Object Extract Class Move Embellishment to Decorator Replace Conditional Logic with Strategy Replace Implicit Language with Interpreter Replace Implicit Tree with Composite Replace State-Altering Conditionals with State Replace Type Code with Class Replace Type Code with State/Strategy Replace Type Code with Subclasses Replace Array With Object Divergent Change CHP Extract Class	Duplicated Code DIS Chain Constructors Extract Composite Extract Method Extract Class Form Template Method Introduce Null Object Factory Method Pull Up Method Pull Up Field Substitute Algorithm Adapter	Long Method BLO Extract Method Compose Method Introduce Parameter Object Move Accumulation to Collecting Parameter Move Accumulation to Visitor Decompose Conditional Preserve Whole Object Replace Conditional Dispatcher with Command Replace Conditional Logic with Strategy Replace Method with Method Object Replace Temp with Query
Long Parameter List BLO Replace Parameter with Method Introduce Parameter Object Preserve Whole Object Middle Man COU Remove Middle Man Inline Method Replace Delegation with Inheritance Speculative Generality DIS Collapse Hierarchy Rename Method Remove Parameter	Shotgun Surgery CHP Move Method Move Field Inline Class Data Clumps BLO Extract Class Preserve Whole Object Introduce Parameter Object Data Class DIS Move Method Encapsulate Field Encapsulate Collection Temporary Field OOA Extract Class Introduce Null Object	Feature Envy COU Extract Method Move Method Move Field Parallel Inheritance Hierarchies CHP Move Method Move Field Message Chains COU Hide Delegate Extract Method Move Method Lazy Class DIS Collapse Hierarchy Inline Class
Alternative Classes with Different Interfaces OOA Extract Class Extract Class with Adapter	Incomplete Library Class COU Introduce Foreign Method Introduce Local Extension	

Mob programming 2

Si inizia a capire il ritmo

Migliore interazione

Meno blocchi, bene!

SOLID

Conosco ma non ricordo

Proviamoci ancora

E ancora, **S** era... **O**...



COUPLING & COHESION

Piú di quanto pensassi

Formule? FORMULE?

Tanto da pensarci su



Yes, but...
**SHOW ME
SOME CODE!!!**



Seconda presentazione

Piú personale

Piú immagini

Poco tempo ma va bene

Seconda presentazione – Finito

Meglio di quanto pensassi

Inizio a capire come fare

A wide-angle photograph of a beach at sunset. The sun is low on the horizon, creating a bright, golden glow that reflects on the water and sand. The sky is a mix of deep blue and lighter, hazy tones with scattered white clouds. Several seagulls are captured in flight, their dark silhouettes contrasting against the bright sky. The ocean waves are gentle, with white foam visible as they wash onto the sandy shore. The overall mood is peaceful and serene.

Flying

CONNASCENCE

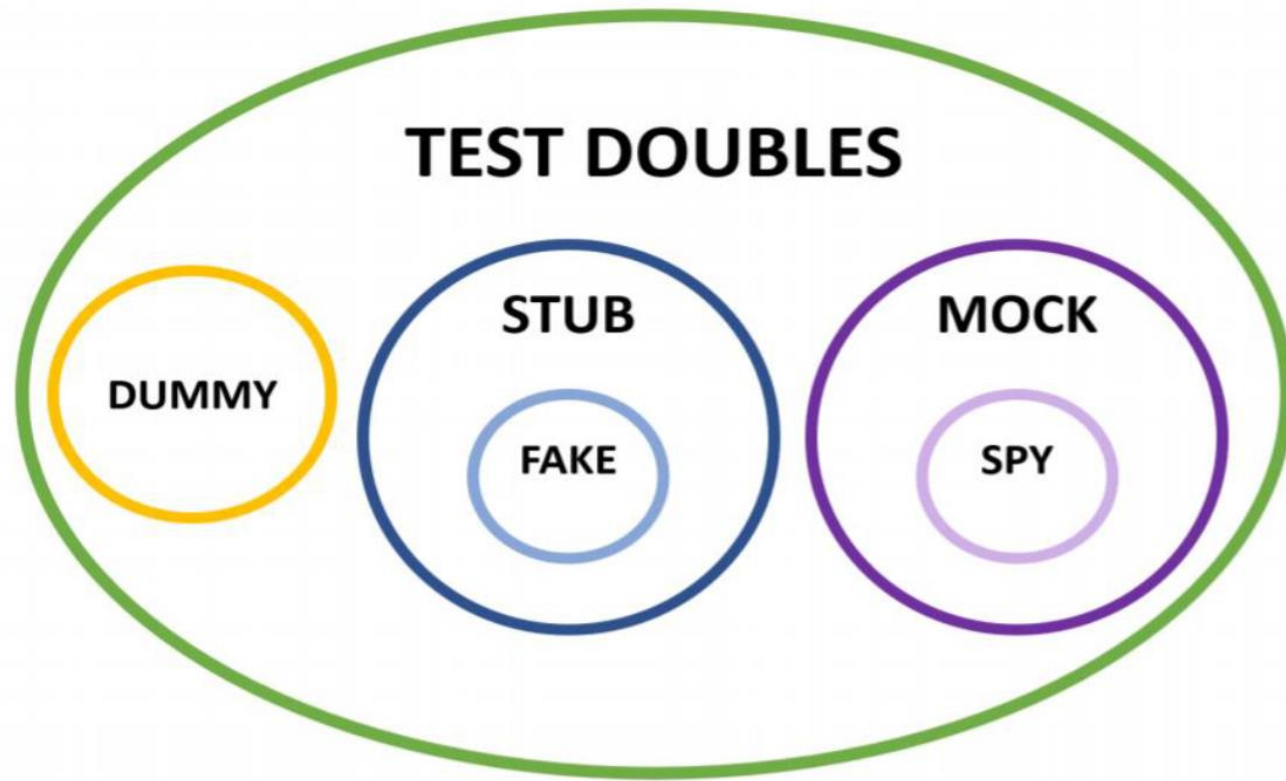
Interessante!

Altro punto di vista

Lo capisco bene



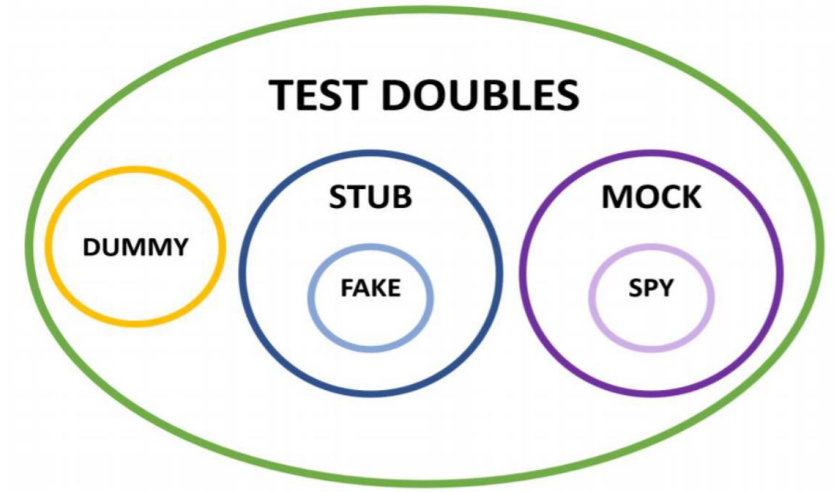
Test Doubles



Test Doubles

Non rimane in testa

Ripetere studio

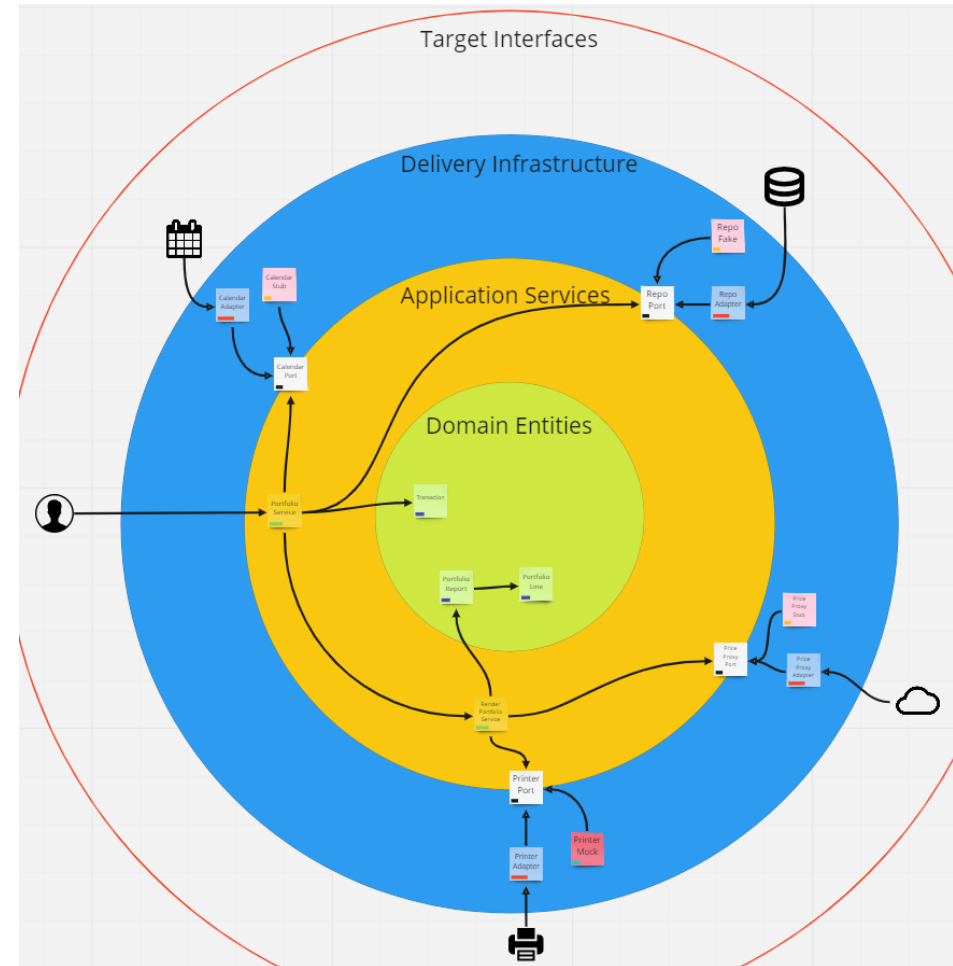
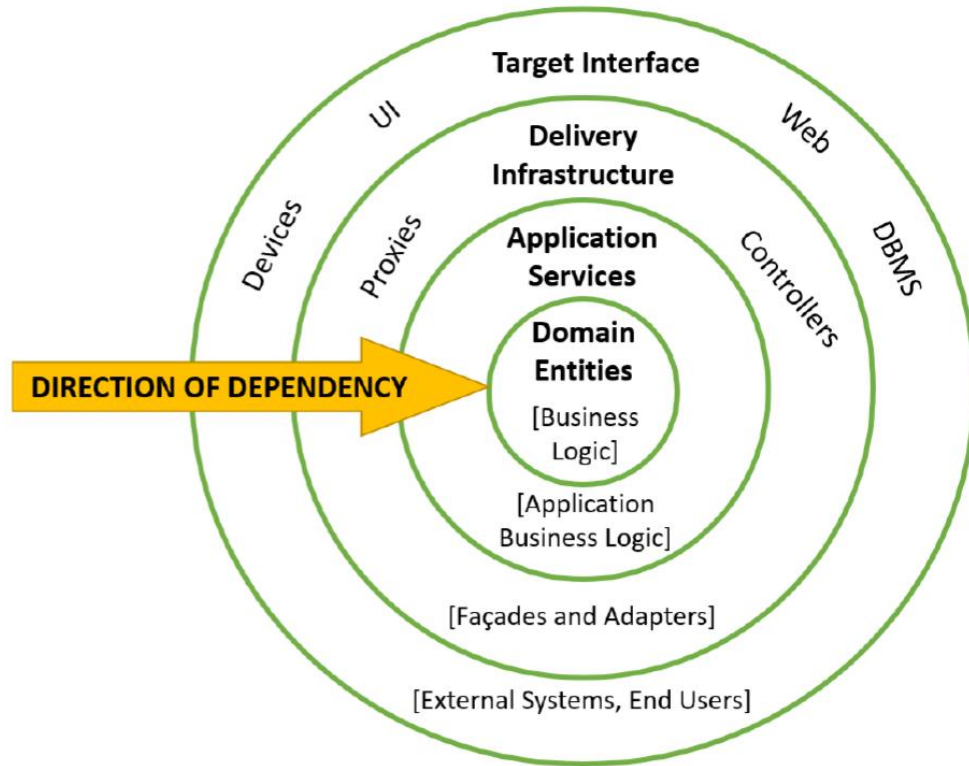


→ **Dummy**: needed to complete the parameters' list of a method, but never used. Not common in well-designed systems.

→ **Stub**: responds to calls with some pre-programmed output. They need to be specifically setup for every test. **Fakes** are hand made stubs.

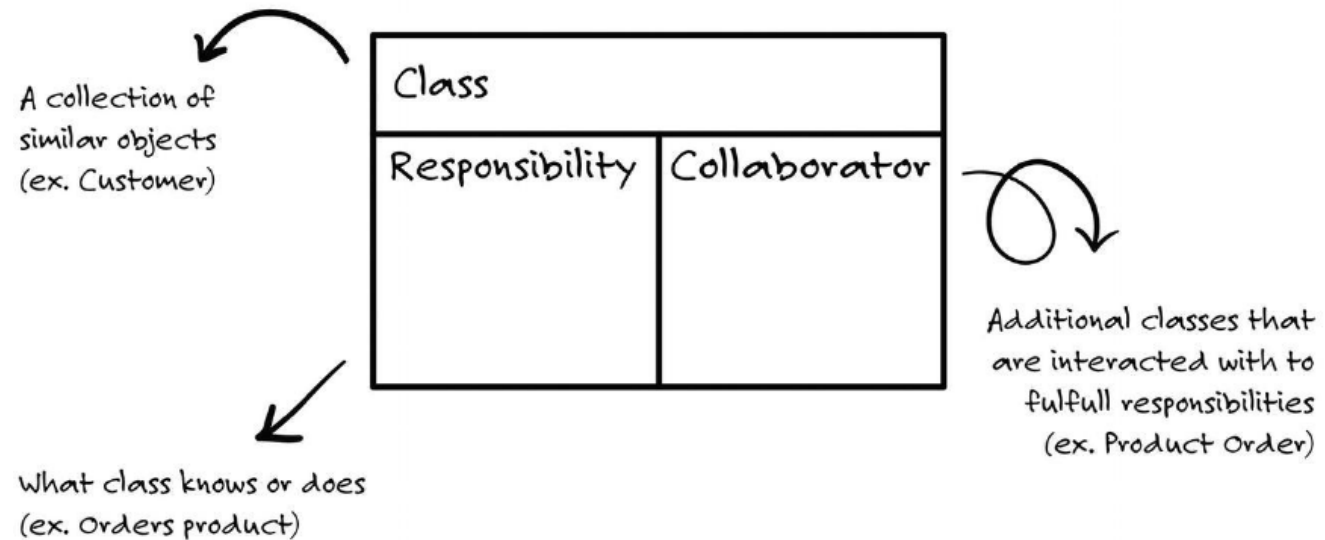
→ **Mock**: set up with expectations of the calls they are expected to receive. Provide a way of verify that a behaviour has been triggered correctly. **Spy** is a hand made mock.

Hexagonal Architecture...

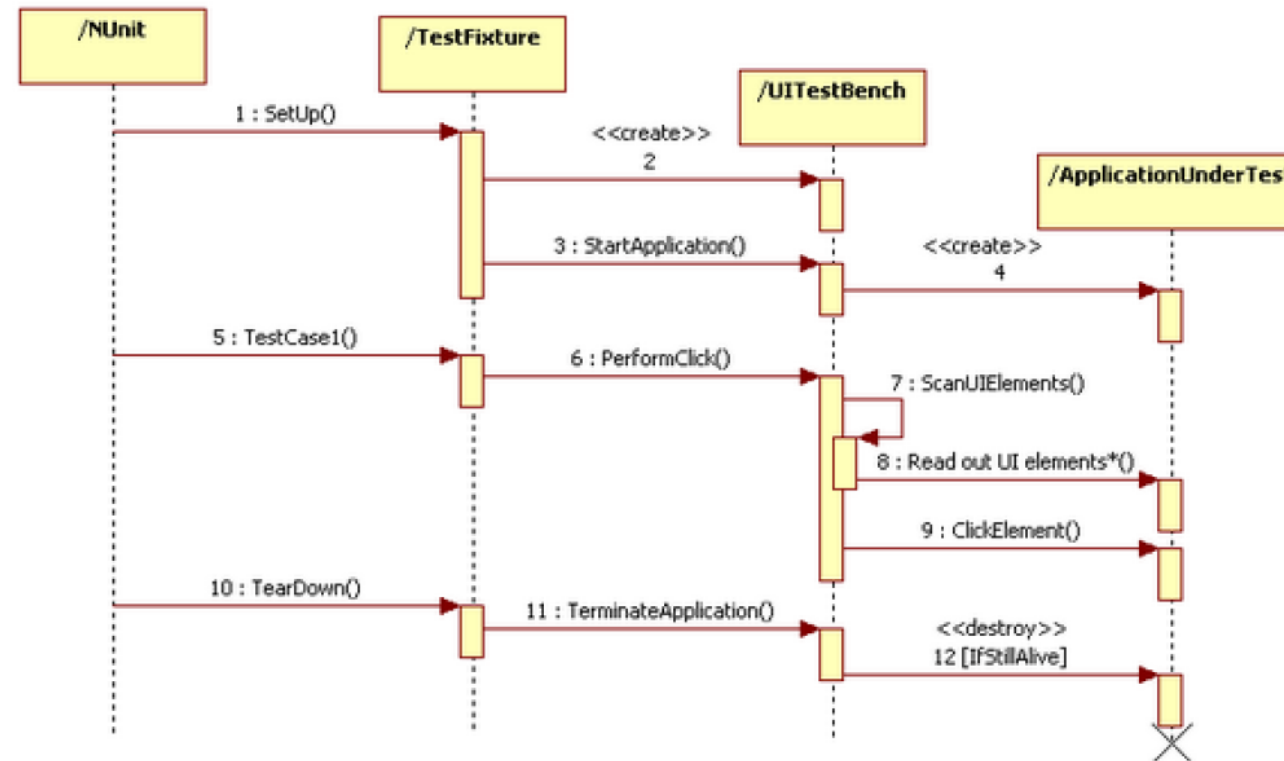


CRD Cards

Class / Responsibility / Collaborator Cards



Sequence diagrams...



Mob mob e ancora mob

Sempre meglio

Opinioni e esperienze si mischiano

Si raggiunge prima il risultato

Terza presentazione

Se la prima era tecnica

La seconda un po' personale

Allora la terza...

Conclusione

Non solo pratica e teoria

Una strada da seguire

Migliore interazione con altri

Conclusione

Mettersi in gioco

Presentare senza timore

Valore aggiunto in soft e hard skill

Conclusione

Esperienza molto positiva

Input non scontati

Grazie!



Domande?

Stefano Mondini
stefano.mondini@eoc.ch