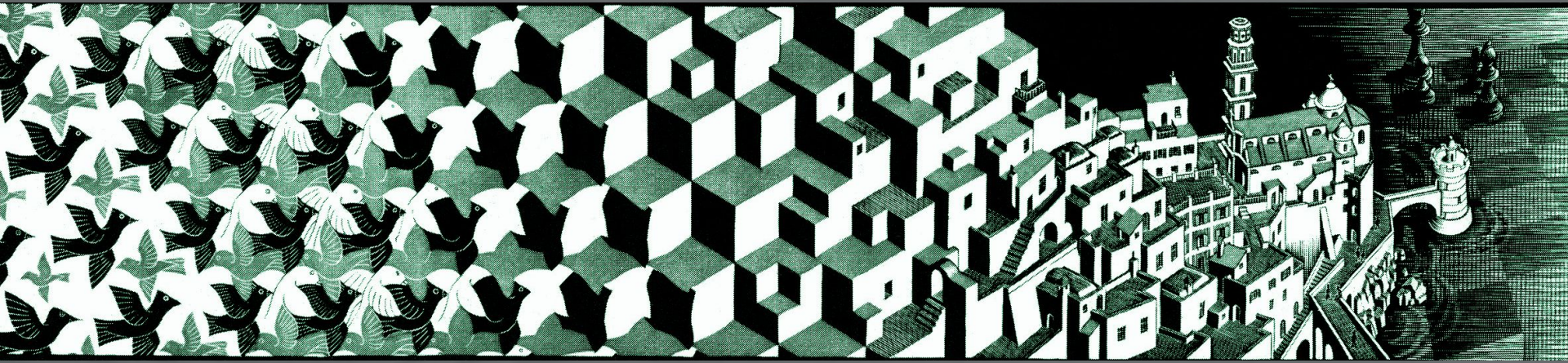
Design Patterns: Head-first vs Emergent

@robgallea
robertogallea



Cosa sono i Design Patterns?

un software design pattern è una **soluzione** generale e riusabile per un **problema frequente** in un **particolare contesto**


Quick, Comprehensive,
Indispensable
5th Edition
Head-first
Design Patterns
O'REILLY

Head First approach

un software design pattern è una **soluzione** a un **problema** imposto in un **particolare contesto**


Sono bellissimi..!

Photo by Lucas Pezeta from Pexels

Servirà in futuro...


Photo by Sammy-Williams - Pixabay
So’ figo!

```php
class MessageStrategy
{
    public function __construct(private MessageBody $mb) {}

    public function sendMessage()
    {
        echo($this->mb->getPayload());
    }
}
```

Strategy

```php
class MessageBody
{
    private $payload;

    public function getPayload()
    {
        return $this->payload;
    }

    public function send(MessageStrategy $ms)
    {
        $ms->sendMessage($this->payload);
    }

    public function configure($obj)
    {
        $this->payload = $obj;
    }
}
```

DTO

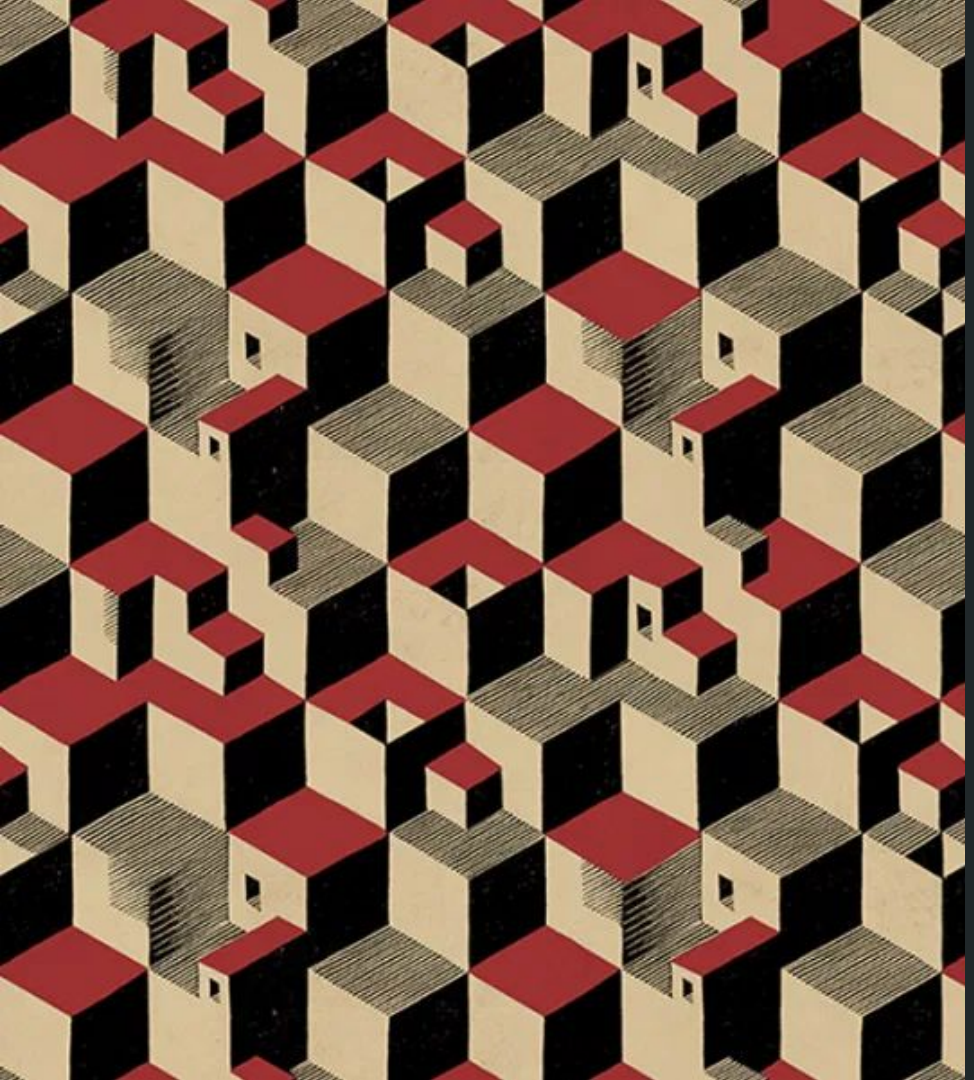
```php
class DefaultFactory
{
    static ?DefaultFactory $instance = null;

    private function __construct() { }

    public static function getInstance()
    {
        if (self::$instance === null) {
            self::$instance = new DefaultFactory();
        }

        return self::$instance;
    }

    public function createStrategy(MessageBody $mb): MessageStrategy
    {
        return new MessageStrategy($mb);
    }
}
```

Factory + Singleton

```php
$mb = new MessageBody();
$mb->configure("Hello World");
$asf = DefaultFactory::getInstance();
$strategy = $asf->createStrategy($mb);
$mb->send($strategy);
```

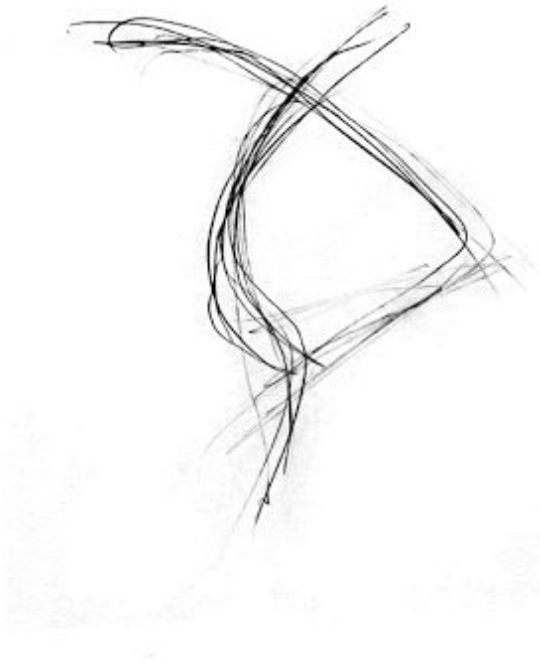
Hello world

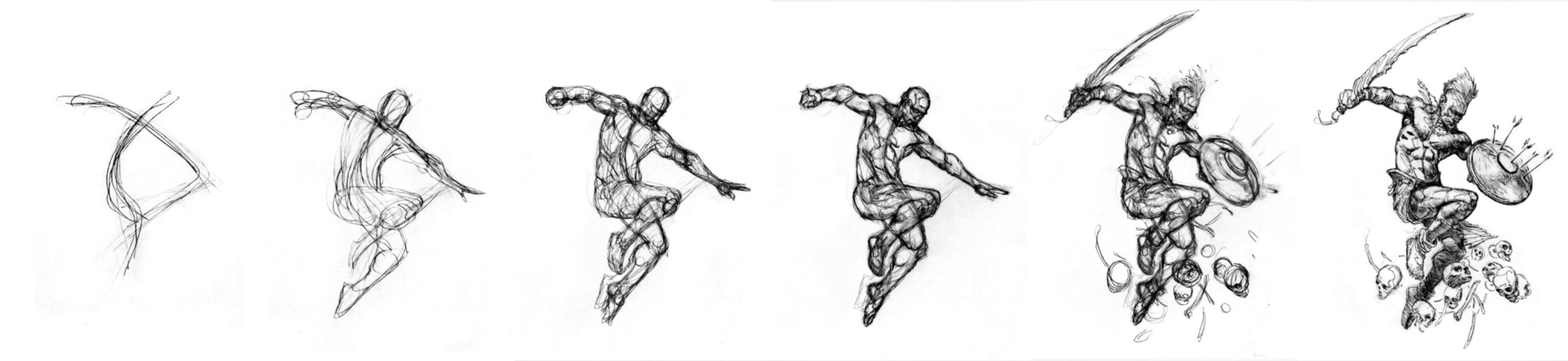
Scrivere codice è un processo iterativo



Design e codice evolvono di pari passo

Cos’è il refactoring?





$$\frac{cos^2(x)(x^2-1)-(x-1)(1+sin(x))(1-sin(x))}{cos^2(x)(x-1)}$$

$$\frac{cos^2(x)(x-1)(x+1)-(x-1)(1-sin^2(x))}{cos^2(x)(x-1)}$$

$$\frac{cos^2(x)(x-1)(x+1)-(x-1)cos^2(x)}{cos^2(x)(x-1)}$$

$$\frac{\cancel{cos^2(x)(x-1)}[(x+1)-1]}{\cancel{cos^2(x)(x-1)}}$$

$$(x+1)-1$$


Ph: Roberto Conte

Puntare ad un miglior design

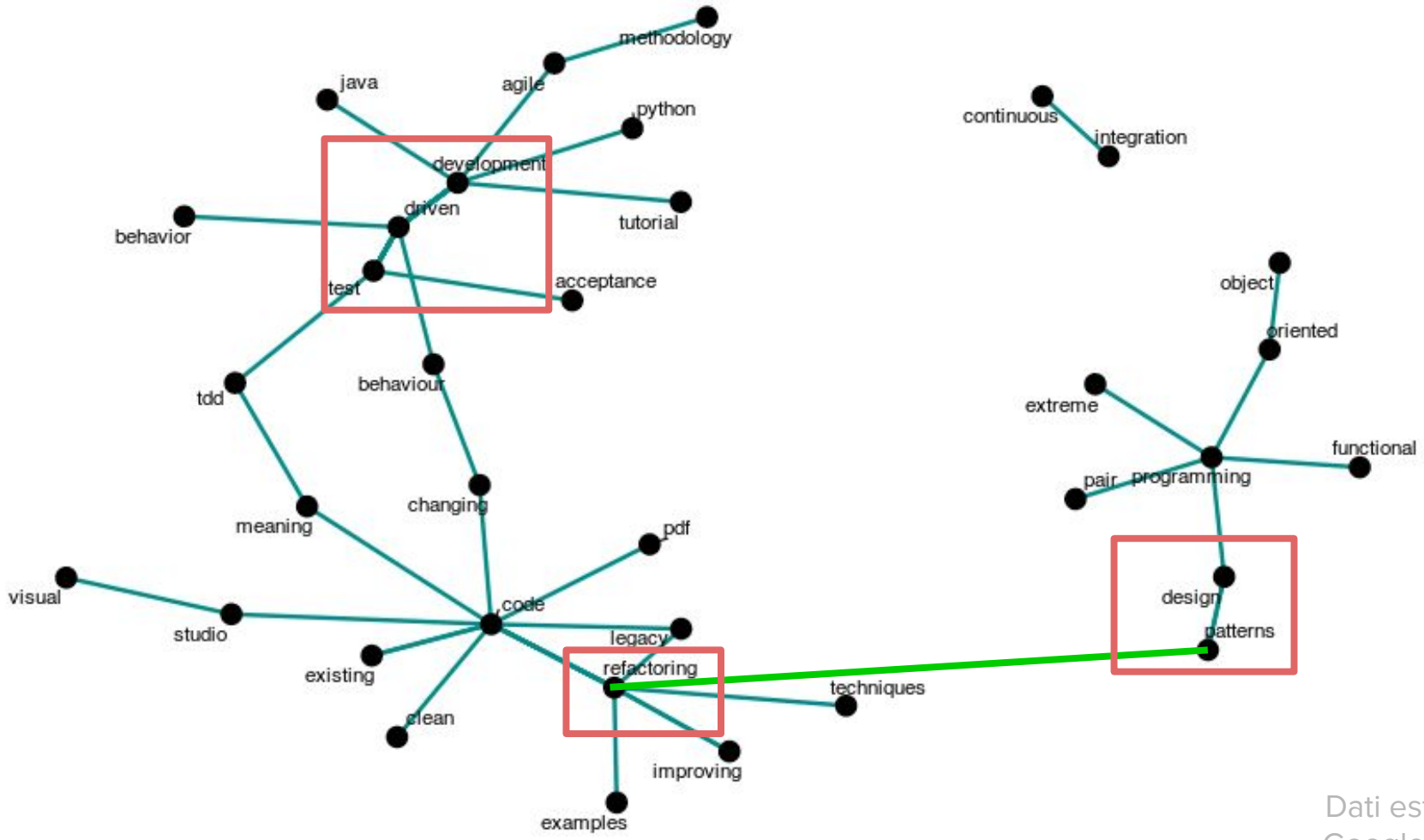
Miglior design

Leggibilità

Manutenibilità

Riusabilità

Testabilità



Dati estratti da
Google Trends

l’applicazione dei pattern come **<u>conseguenza naturale</u>** del processo di **refactoring**

Da dove iniziare?

```
public function mess()
{
```

```
    // code smells

}
```

Conditional Complexity

Lazy Class

Switch Statements

Long Method

Duplicated code

Alternative classes with different interfaces

Indecent Exposure

Solution Sprawl

Combinatorial explosion

Oddball solution

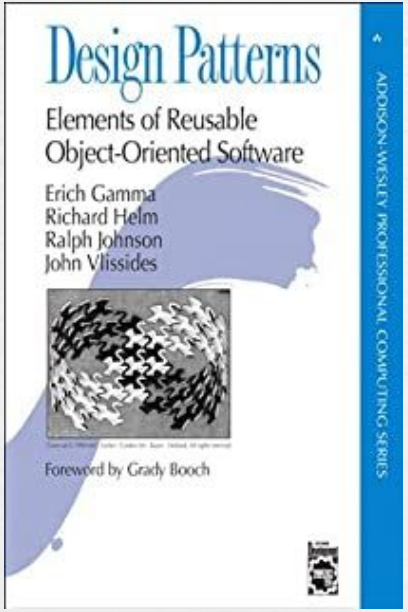
Large Class

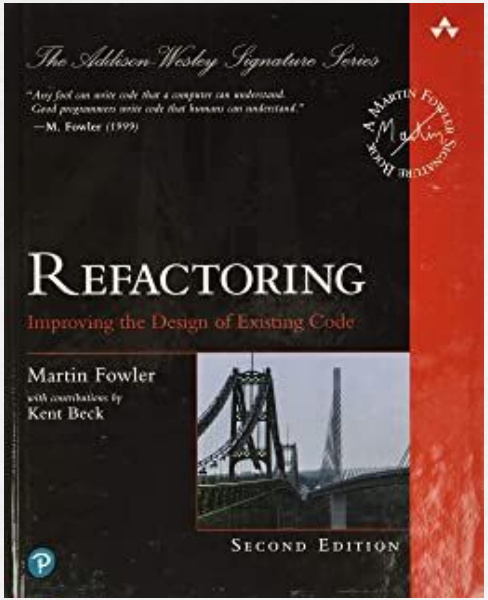
Primitive Obsession

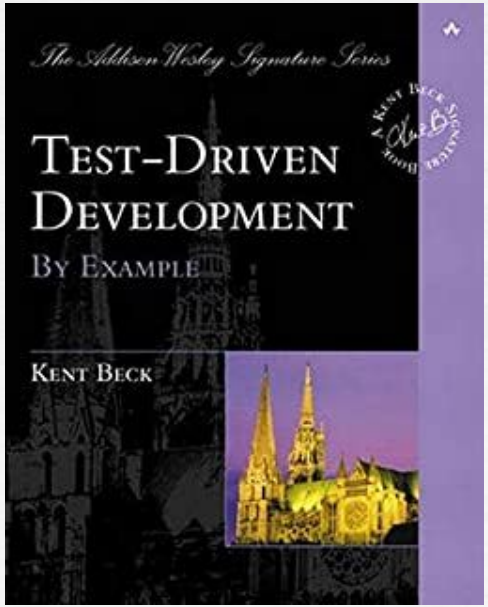
/** @test */

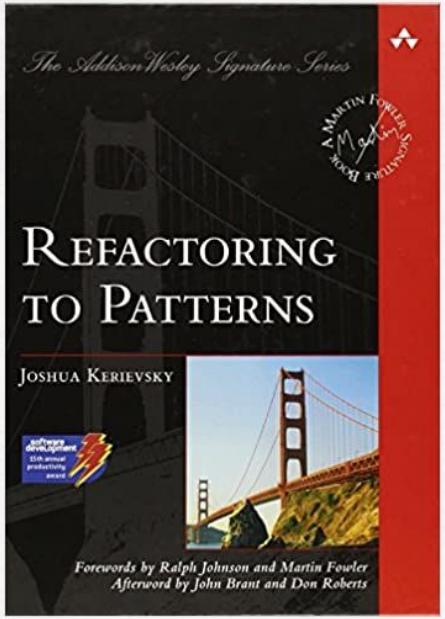
Patterns?
Adesso sì...
(se è il caso!)

Vediamo un esempio....









- Design Patterns (Gamma et al.)
- Refactoring (Fowler)
- Test-Driven Development (Beck)
- Refactoring to Patterns (Kerievsky)

@robgallea

robertogallea