Clean Code

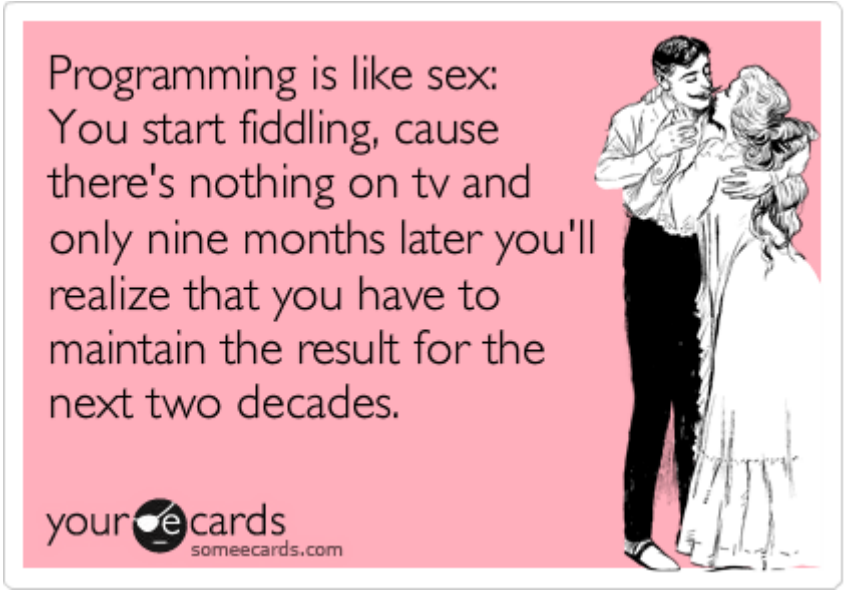
Elevando o nível de desenvolvimento de software

Jean Carlo Machado

Sobre

CompuFácil

- ⇢ php-src
- ⇢ Doctrine
- ⇢ Zend Framework
- ⇢ phpunit
- ⇢ PHPmd
- ⇢ nfephp


Programming is like sex:
You start fiddling, cause there's nothing on tv and only nine months later you'll realize that you have to maintain the result for the next two decades.
yourecards
someecards.com

Sintomas de apodrecimento

Rigidez

Fragilidade

Imobilidade

Viscosidade

Software ruim implica em políticas ruins

Janelas quebradas

O grande redesign


Diploma
On the recommendation of the Faculty and by virtue of the authority vested in them the
Trustees of the University have conferred upon
The degree of
Bachelor of Fine Arts
President of the University

Clean code

Estilo de desenvolvimento de software focado na leitura e manutenção

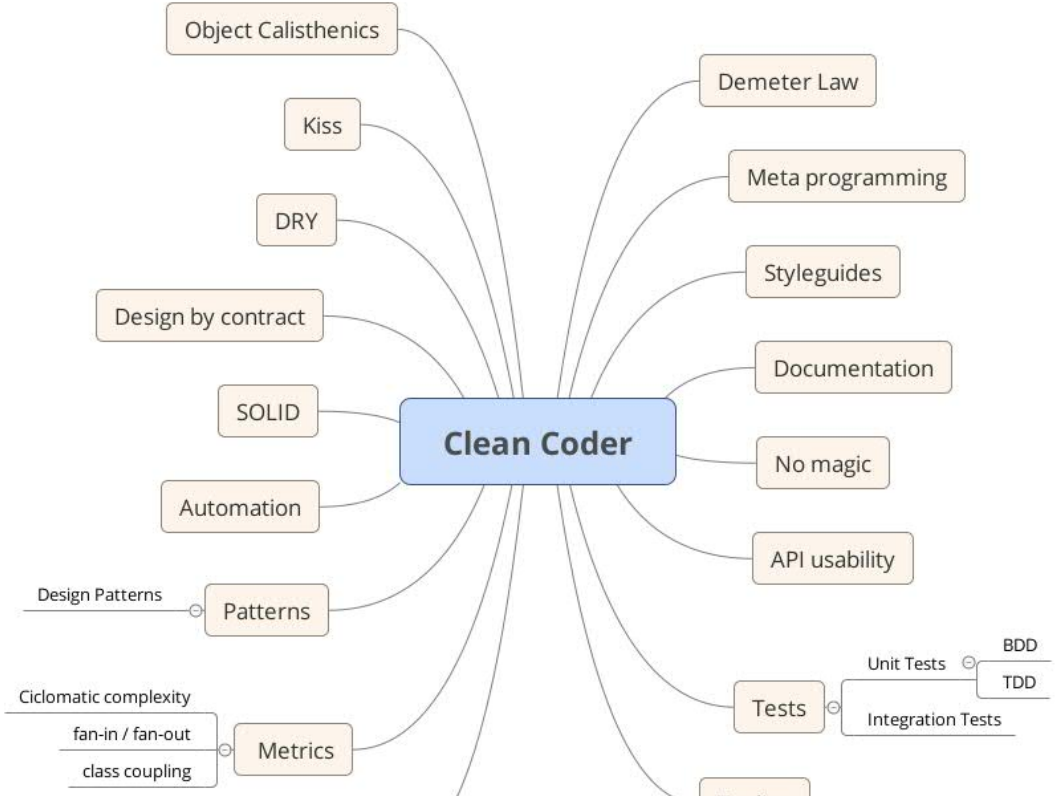
Programming is the art of telling another human what one wants the computer to do Donald Knuth

Por que manutenção?

Ler código é a parte que mais consome tempo em manutenção

Manutenção consome 70% do custo de um software

Skills do Clean Coder

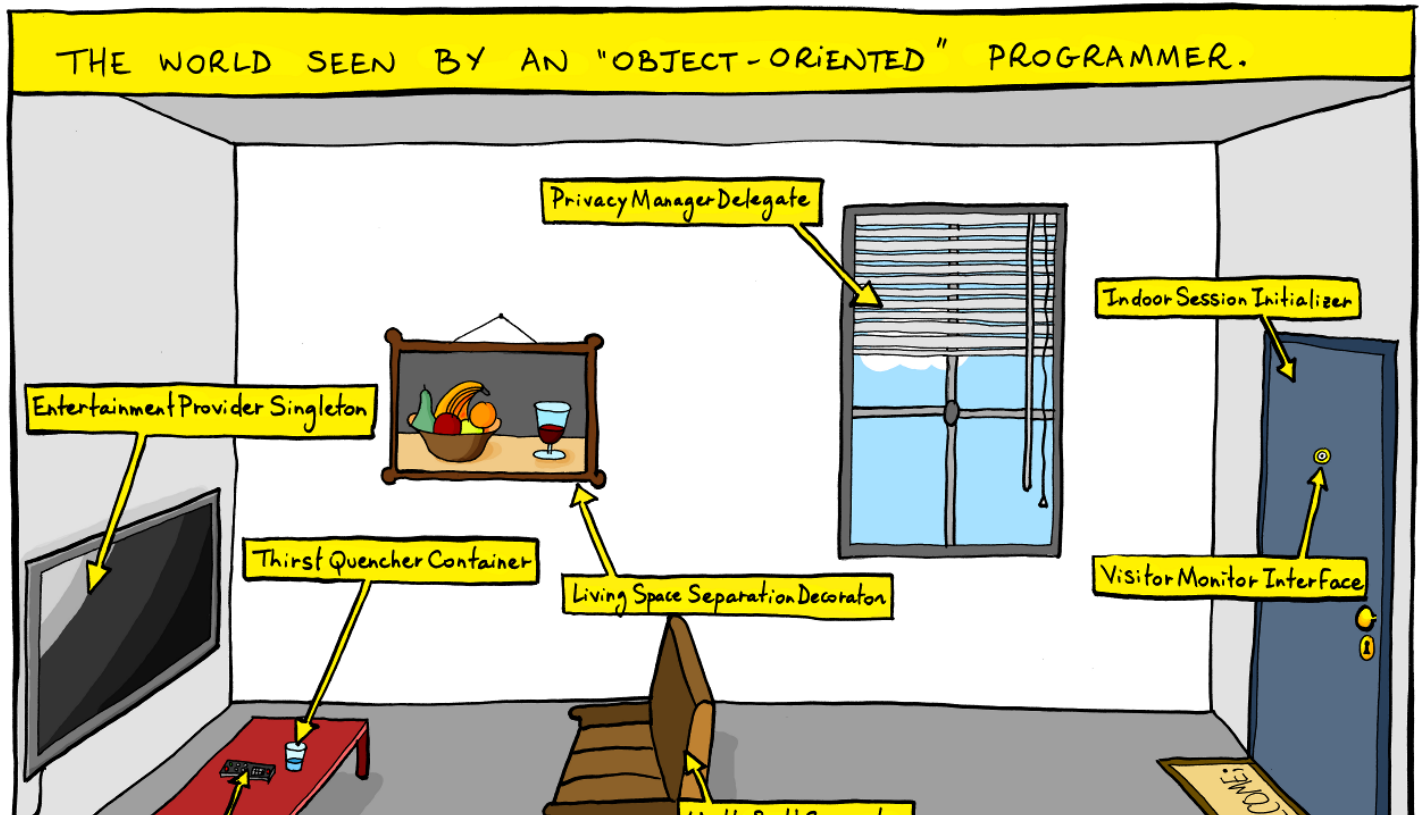

Clean Coder
Object Calisthenics
Kiss
DRY
Design by contract
SOLID
Automation
Patterns
Design Patterns
Metrics
Ciclomatic complexity
fan-in / fan-out
class coupling
Demeter Law
Meta programming
Styleguides
Documentation
No magic
API usability
Tests
Unit Tests
BDD
TDD
Integration Tests

Readability !== Complexity

Nomes

Se precisar olhar o código tem que trabalhar nos nomes

"And" ou "Or" comunicam múltiplas responsabilidades


THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.
Privacy Manager Delegate
Indoor Session Initializer
Entertainment Provider Singleton
Thirst Quencher Container
Living Space Separation Decorator
Visitor Monitor Interface

Evite nomes com

- ⇢ Object
- ⇢ Thing
- ⇢ Manager
- ⇢ Component
- ⇢ Part
- ⇢ Entity
- ⇢ Item

Largura

~~Código largo é código belo~~

100% de correlação inversa com quantidade de símbolos

80 caracteres é o suficiente

Quebre computação em múltiplas linhas

```
$mean = array_sum($dataSet) / count($dataSet);
$varrianceSum = array_reduce($dataSet, function($accumulator, $entry) {
    return $accumulator +  pow($entry - $mean, 2);
}, 0);
$variance =  $varianceSum / count($dataSet);
$standardDeviation =  sqrt($variance);
```

Comentários

Para expressar código

Nomes dos autores

Explicar tradeoffs

Comentários inúteis

```
/**
 *
 * @param $title The title of the CD
 * @param $author The author of the CD
 * @param $tracks The number of tracks of the CD
 *
 */
public addCd($title, $author, int $tracks);
```

Funções

Argumentos booleanos

Mais de 3 argumentos é difícil justificar

Classes

Metáfora do artigo

Evite

- Elses
- Código de erros
- Libs desconhecidas
- Getters e setters (Deméter)
- Dependências bidirecionais
- Dados e comportamento
- Editar modelagem editar diagramas? (DRY)

SOLID

- Single responsability principle (SRP)
- Open close principle (OC)
- Liskov substitution (LS)
- Interface segregation (ISP)
- Dependency inversion (DI)

Shapes Bad

Shapes Good

Calisthenics

⇢ Um nível de indentação

⇢ Não use o else

⇢ Guarde primitivas em objetos

⇢ Uma classe de coleção não deve ter outros atributos

⇢ Um ponto por linha

⇢ Sem abreviação

⇢ 50 linhas por aquivo 10 arquivos por diretórios

⇢ Nenhuma classe com mais de 2 variáveis de instância

⇢ Sem getters e setters

"One of our difficulties will be the maintenance of an appropriate discipline, so that we do not lose track of what we are doing."

Disciplinas

Review de código

Programação em pares

Test driven development

Testes

TDD é um processo de design

Não se testa código se testa requisitos

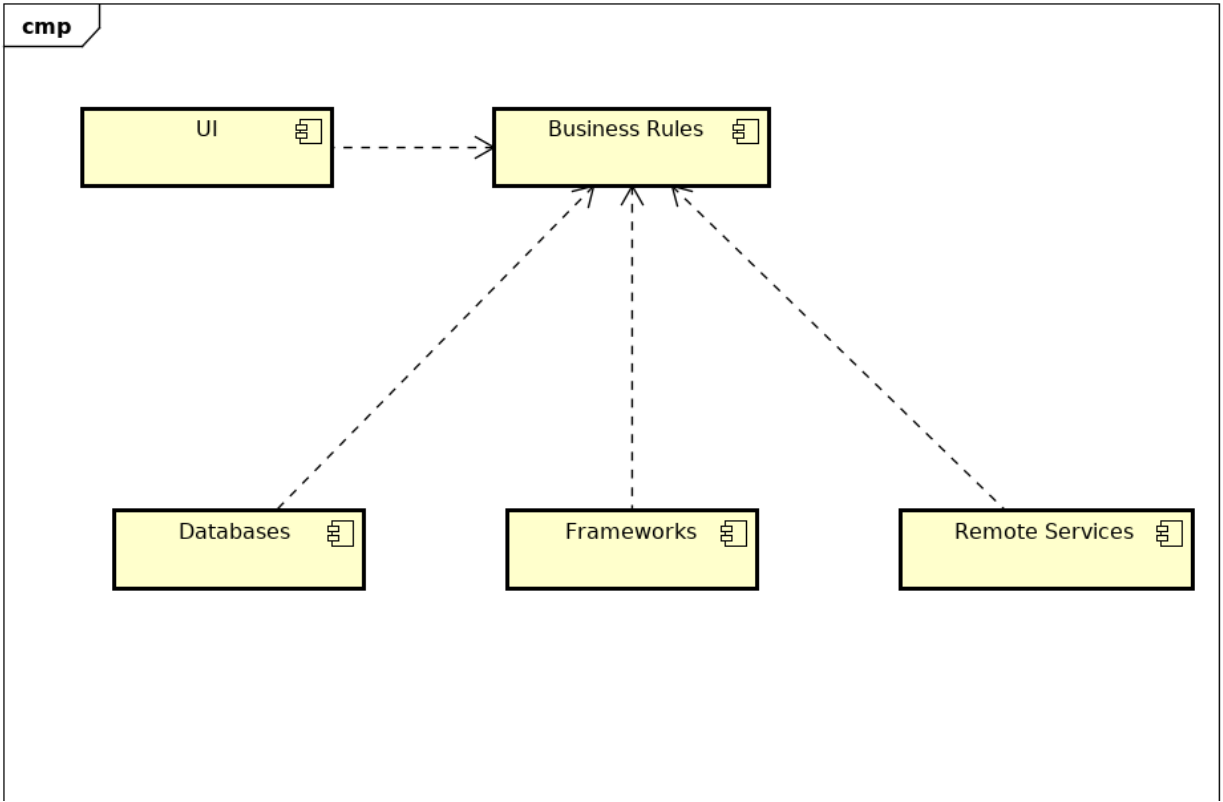
Se uma escolha de design torna algo mais difícil de testar está errado

Testes TEM que ser rápidos / Testes ruins podem te atrasar

Arquitetura

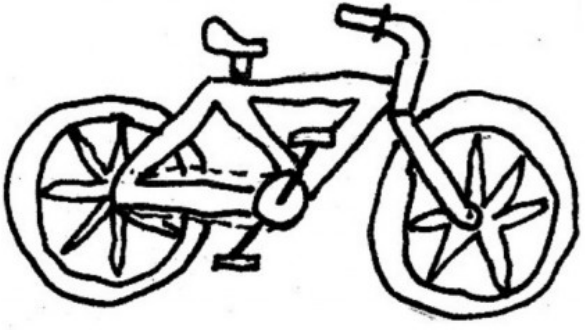
===

Intenção

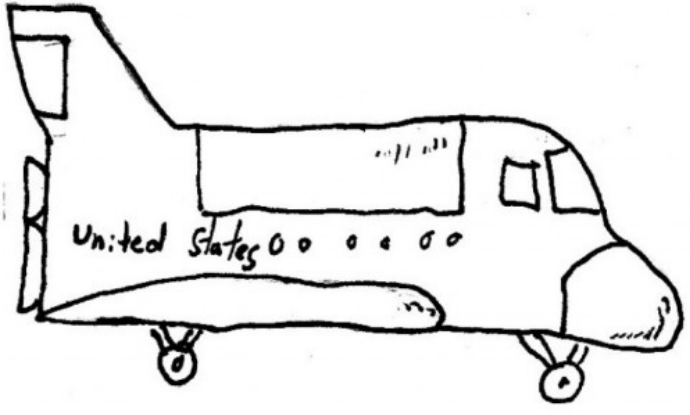

cmp
UI
Business Rules
Databases
Frameworks
Remote Services

KISS

Relatively simple things can tolerate a certain level of disorganization. However, as complexity increases, disorganization becomes suicidal. Robert Martin



Good Bike



Bad Bike

[Over engineered future ready
for flying also when required ☺]

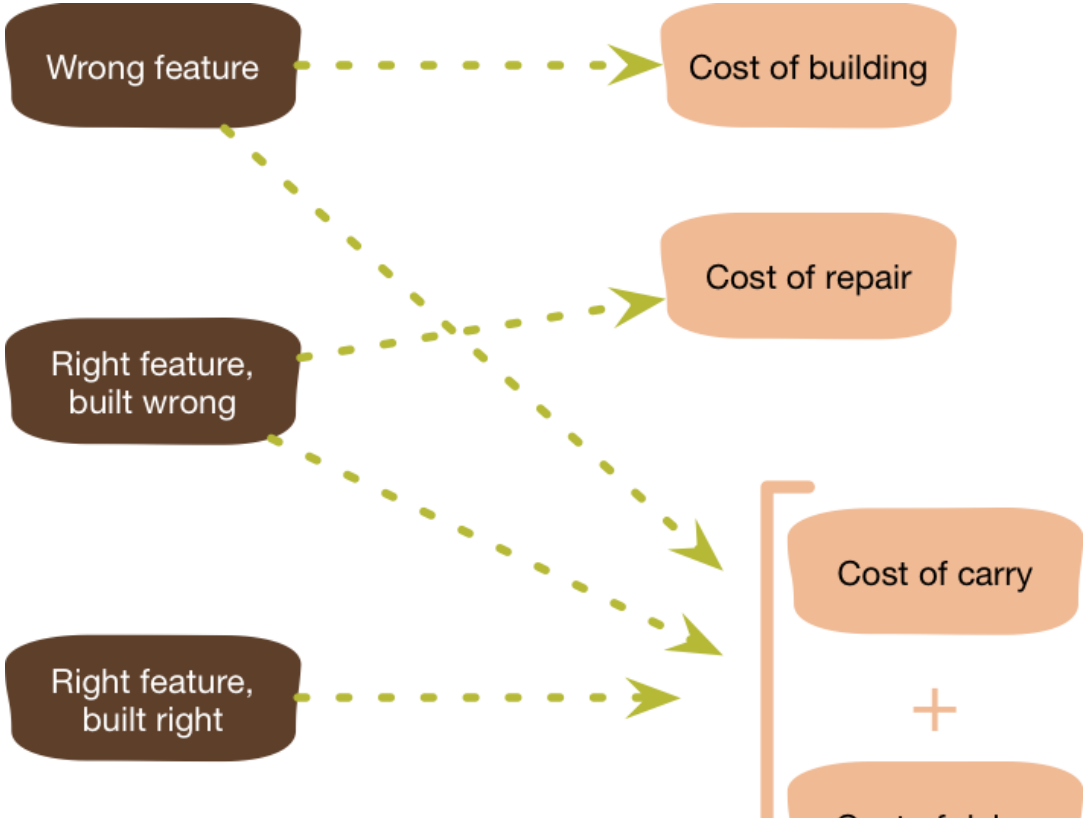
YAGNI

Presumir adiciona complexidade desnecessária

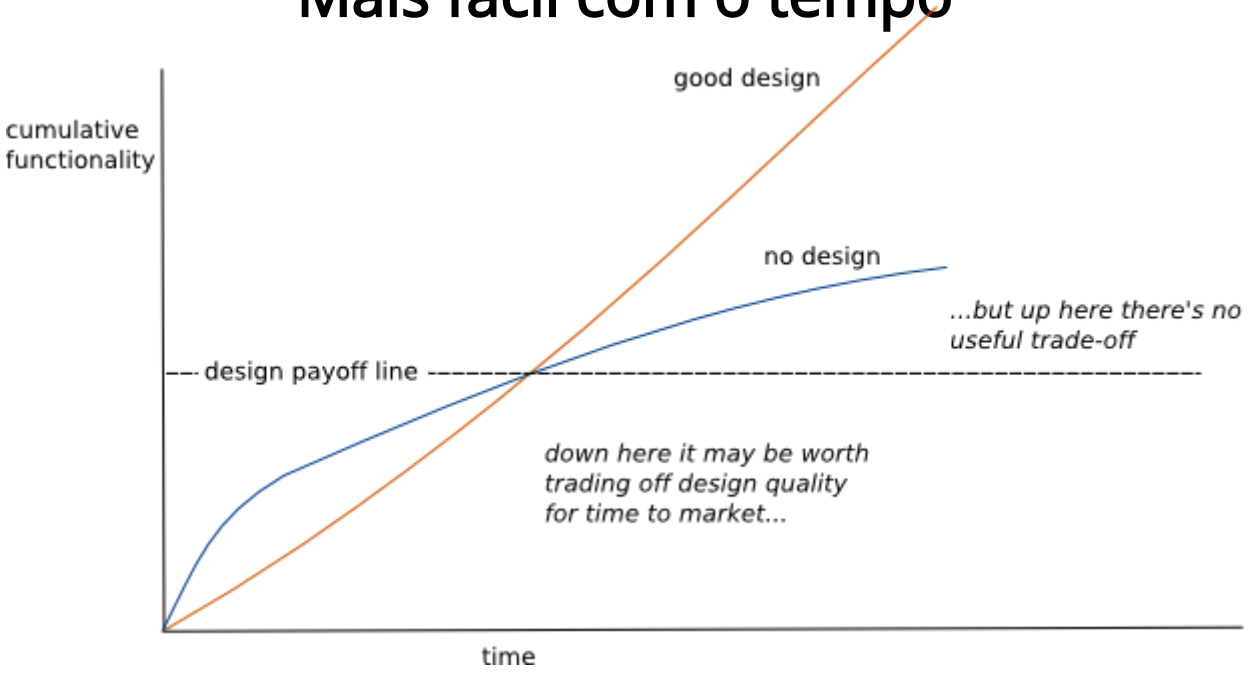
Vai custar muito adicionar depois?

Não inclui o custo de fazer o software mais fácil de modificar

Ultimo momento responsável


Wrong feature
Cost of building
Cost of repair
Right feature, built wrong
Cost of carry
+
Right feature, built right

Mais fácil com o tempo



Conclusão

Focar no 80% que não requer performance

Você é o especialista

Clean code não é sobre perfeição é sobre honestidade

Lei do escoteiro

Clean Code is a code that is written by someone who cares
Michael Feathers

Ferramentas

- ⇢ php styleguides
- ⇢ phpmd
- ⇢ phpcs
- ⇢ codacy
- ⇢ object-calisthenics/phpcs-calisthenics-rules
- ⇢ mamuz/php-dependency-analysis
- ⇢ pdepend
- ⇢ phpstan
- ⇢ phan/phan

Literatura

- Clean code: A hand book of Agile Software craftsmanship; Robert C. Martin
- The pragmatical programmer; Andrew Hunt
- Code Complete; Steve McConnell
- Refactoring: Improving the Design of Existing Code;
- Release It!: Design and Deploy Production-Ready Software; Michael T. Nygard

Dúvidas?

Talk link: https://goo.gl/FKzgqn

Github: https://github.com/jeanCarloMachado

Twitter: https://twitter.com/JeanCarloMachad

E-mail: contato@jeancarlomachado.com.br