

Clean Code

Lars Wiedemann
skriptfabrik GmbH



github.com/gernsdorfer

twitter.com/gernsdorfer

```
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
    $totals = array();
    $old_taxes = $taxes;
    $lpa_tax = array();
    $sort_order = array();
    $results = $this->model_extension_extension->getExtensions('total');
    foreach ($results as $key => $value) {
        if (isset($value['code'])) {
            $code = $value['code'];
        } else {
            $code = $value['key'];
        }
        $sort_order[$key] = $this->config->get($code . '_sort_order');
    }
    array_multisort($sort_order, SORT_ASC, $results);
    foreach ($results as $result) {
        if (isset($result['code'])) {
            $code = $result['code'];
        } else {
            $code = $result['key'];
        }
        if ($this->config->get($code . '_status')) {
            $this->load->model('extension/total/' . $code);
            // We have to put the totals in an array so that we can
            // by reference.
            $this->{'model_extension_total_' . $code}->getTotal($total_data);
            if (!empty($totals[count($totals) - 1]) && !isset($totals[
                count($totals) - 1]['code'])) {
                $totals[count($totals) - 1]['code'] = $code;
            }
        }
        $tax_difference = 0;
        foreach ($taxes as $tax_id => $value) {
            if (isset($old_taxes[$tax_id])) {
                $old_taxes[$tax_id] = $value;
                $lpa_tax[$tax_id] = $value;
            } else {
                $old_taxes[$tax_id] = $value;
                $lpa_tax[$tax_id] = $value;
            }
        }
        $totals[count($totals)] = array(
            'code' => $code,
            'label' => $label,
            'value' => $value,
            'difference' => $tax_difference
        );
        $tax_difference += $value;
    }
}
Carousel.prototype.getExtension = function () {
    var itemIndex = this.$items.index(this.$active);
    return this.$items.eq(itemIndex);
}
Carousel.prototype.getItemForDirection = function (direction, active) {
    var delta = direction == 'prev' ? -1 : 1;
    var activeIndex = this.getItemIndex(active);
    var itemIndex = (activeIndex + delta) % this.$items.length;
    return this.$items.eq(itemIndex);
}
Carousel.prototype.get = function (pos) {
    var that = this;
    var activeIndex = this.getItemIndex(this.$active);
    var element = this.$element;
    if (pos > (this.$items.length - 1) || pos < 0) return;
    if (this.sliding) return this.$element.one('slid.bs.carousel', function () { that.to(pos) });
    if (activeIndex == pos) return this.pause().cycle();
    return this.slide(pos, activeIndex ? 'next' : 'prev', this.$items.eq(pos));
}
Carousel.prototype.pause = function () {
    if (this.paused = true)
        this.$element.trigger($.support.transition.end);
}
Carousel.prototype.cycle = function () {
    if (this.interval) {
        this.interval.clearInterval(this.interval);
    }
    return this;
}
```

What is the job of a developer ?

What the customer wants:



What the developer wants:



clean



maintainable

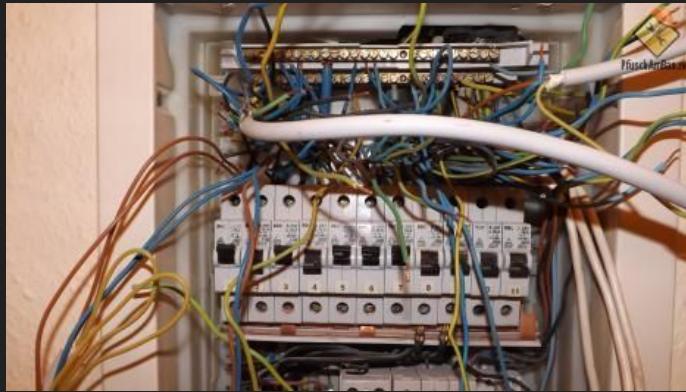


tested code

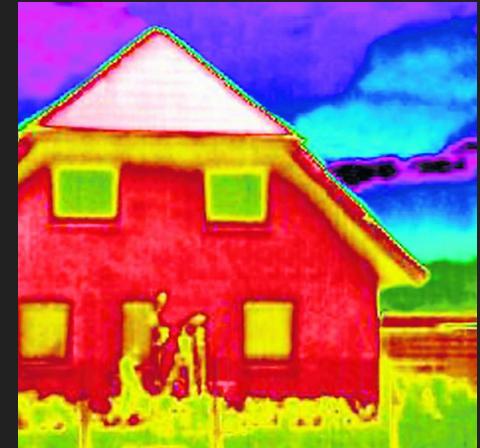
What the developer does:



mess



heavy to add features



not tested

7 * 7

49

12 * 14

168

$$\begin{array}{r} 12 * 14 \\ \hline \end{array}$$

$$12 + 4 = 16$$

$$\begin{array}{r} 2 * 4 = 8 \\ = 168 \end{array}$$

Can you think faster?

NO!

you need

training and experience

Can you develop this feature faster?

We do this later...





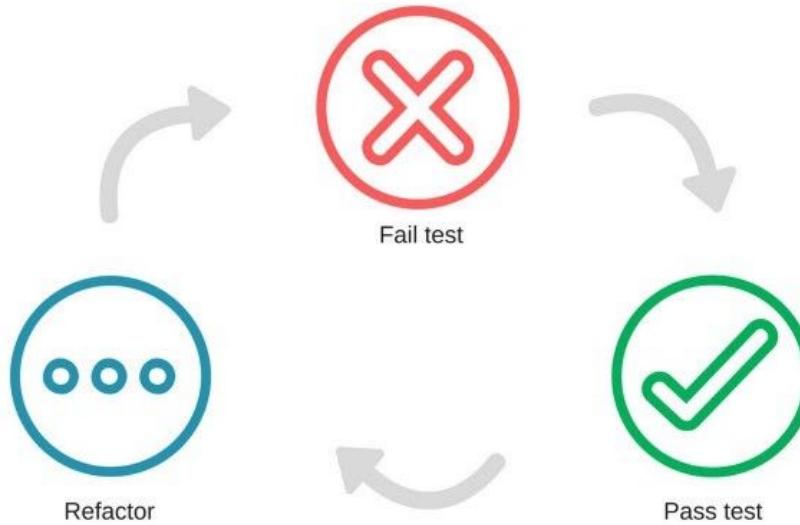
Classes

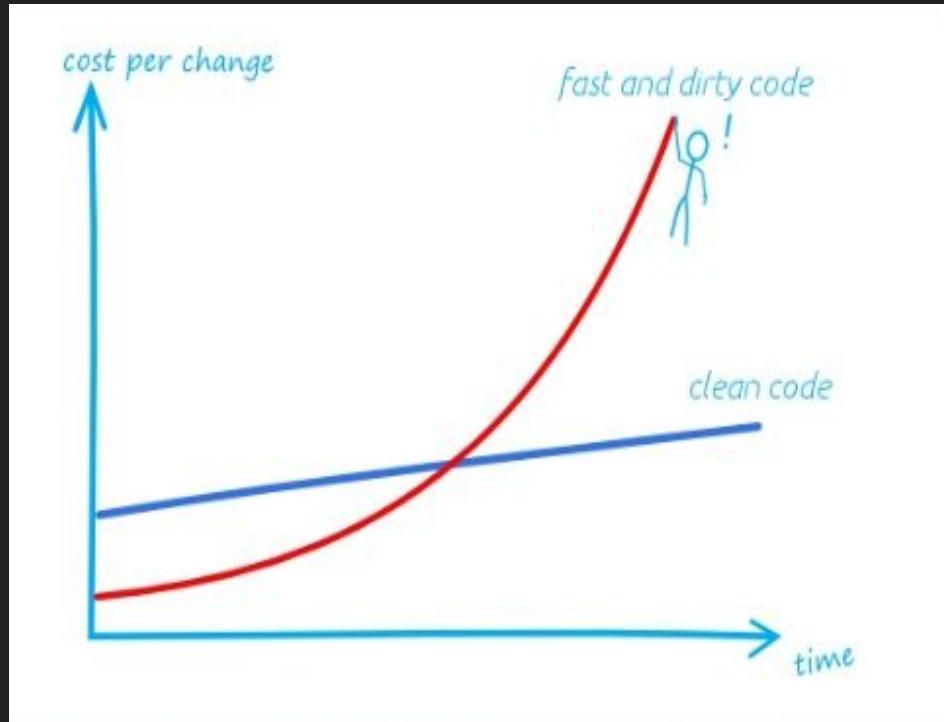
Functions

Sequence,
Selections,
Loops

Most developers stop working as soon as
the code works.

Test-Driven Development Cycle





Hire new employees

Learn how to build window here



Old code

vs

New code



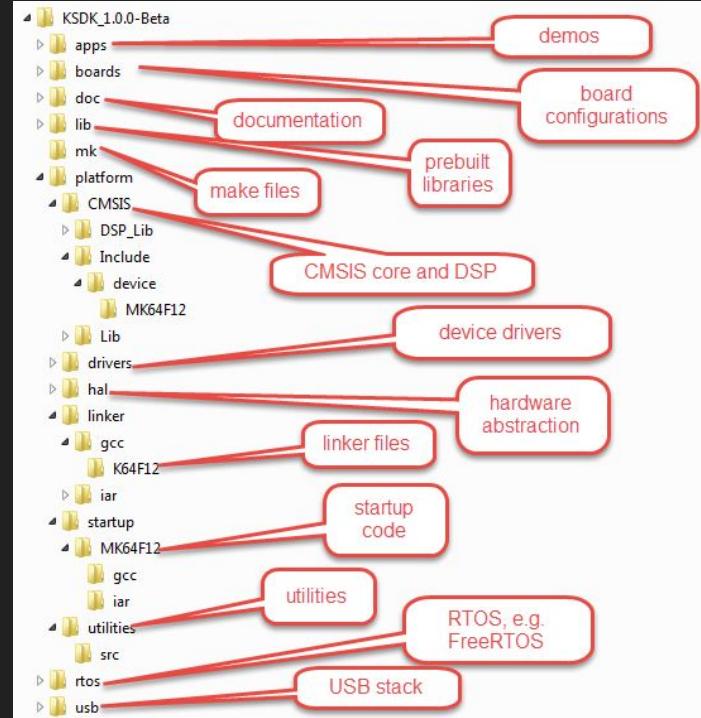
Speed will create a technical debt...

...this technical debt will cost money.

Eisenhower Matrix:

A Productivity Tool to Get Things Done





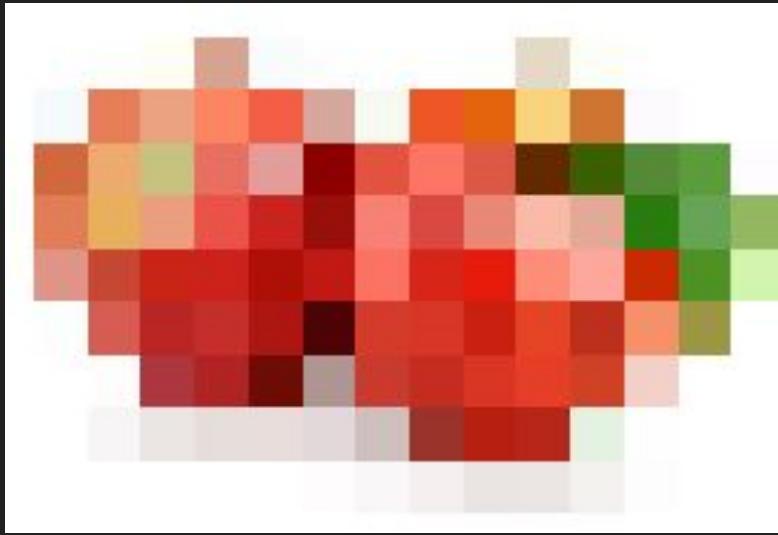
Reliable feeling:

write better code

Double-entry bookkeeping system

Servco
Trial Balance (Partial)
December 31, 2018

	<u>Debits</u>	<u>Credits</u>
Cash	\$ 895.24	
Accounts receivable	499.58	
Accounts payable		\$ 225.00
Common stock		500.00
Retained earnings		234.21
Service revenues		7,500.00
Rent expense	6,000.00	
<i>Some accounts and amounts are omitted</i>		
Totals	<u>\$ 13,444.32</u>	<u>\$ 13,444.32</u>



```
const a = [];
```



```
const apples = [];
```

Does this room contain more
than 0 person ?

```
if (persons.length > 0)
```

Is this room not empty?

```
const isRoomEmpty = persons.length  
==== 0  
  
if (!isRoomEmpty)
```

Does this room contain more
than 30 persons?

```
if (persons.length > 30)
```

Is this room full?

```
const maxPersonsPerRoom = 30;  
const isRoomFull = persons.length >  
maxPersonsPerRoom;
```

```
if(isRoomFull)
```

Variables

- meaning full names
- use prefix/suffix to describe a state
- no magic numbers
- no global variables
- no hungarian notations
- define vars at the beginning
- reduce state



```
function runSwissArmyKnife(  
    openKnife: boolean,  
    showNuclearClock: boolean,  
    runFluxCompensator: boolean,  
    runDefibrillator: boolean,  
    openVHS : boolean,  
    openCassette: boolean,  
    hotplate : number,  
    startCall: number,  
    runBottleOpener: number,  
    startGpsLocation: boolean,  
    openHammer: boolean,  
    openSpade: boolean,  
    runElectricityGenerator: boolean,  
    runDrillingMachine: boolean,  
    openLight: boolean,  
    openCompass: boolean,  
    openLaseSword: boolean,  
    isDebugMode: boolean,
```



```
getCustomer (id) : Customer {  
    if(!this.customer) {  
        this.createCustomer(id);  
    }  
    return this.customer;  
}
```

Functions

- write small functions
- max line length
- avoid boolean flags
- max 3 parameters
- add verbs to functions
- separate command and query
- only do one thing (Curley's Law)
- max 1-2 levels deep



```
/**  
 * Returns the customers  
 */  
function getCustomers() : Customer[]
```



```
function getUsers() {  
    // TODO Check Authorisation  
}
```

I SEE DEAD CODE

I SEE DEAD CODE

```
//woozy += multiplier;  
woozy += multiplier;
```



foo = foo; // DO NOT remove this, it
will break EVERYTHING
// ... Fuck, that really did break
everything

Comments

Don't use it to:

- explain code
- change history
- dead code
- chat
- closing brackets
- todos
- typings/injection
- visual markers

Use it to:

- link to a external issue
- explain strange edge case
- external libs

Boy Scout Rule



Linters



Write code like a Newspaper

YAGNI



DRY



KISS



Principle of least surprise



Separation of concern



Single responsibility principle



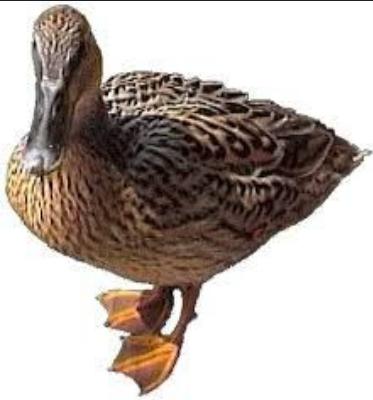
```
class User {  
    getUser () {  
        //do stuff  
        this.log();  
    }  
  
    log () {  
        //LOG STUFF  
    }  
}
```

Open closed principle



```
class Mixer {  
    stop () {  
        }  
  
    start () {  
        return ;  
    }  
}  
  
class Whisk extends Mixer {  
    wheel () {  
        return 'mix';  
    }  
}  
  
class Blender extends Mixer {  
    wheel () {  
        return 'blender'  
    }  
}
```

Liskov substitution principle



```
class Duck {  
    isSwimming: boolean;  
}
```

```
class MyDuck extends Duck {  
    isSwimming: string;  
}
```

Interface segregation principle



```
interface LifeCycle {  
    onInit(): void  
  
    onChange(): void  
  
    onDestroy(): void  
}  
  
class MyClass implements LifeCycle {  
    onInit(): void {  
        // do stuff  
    }  
  
    onChange () : void {}  
  
    onDestroy () : void {}  
}
```

Dependency inversion principle



```
interface Rest {  
    get();  
  
    post();  
}  
  
class RestService implements Rest {  
    get() {  
    }  
  
    post() {  
    }  
}  
  
class MyService {  
    constructor(http: RestService) {  
    }  
}
```

Tell don't ask



```
class BurgerKing {  
    hasBigKing(): boolean {  
    }  
  
    order(meal: Meals[]): void {  
    }  
}
```

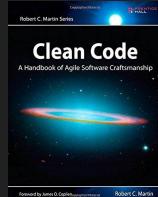
Hardware

vs

Software



References



Robert C. Martin (Uncle Bob)



<https://clean-code-developer.de/>



Martin Fowler (<https://martinfowler.com/>)



<https://angular.io/guide/styleguide>

Outtakes

```
1 int getRandomize(int randMax)
2 {
3     srand ( time(NULL) );
4     int randNum; = rand() % randMax + 1;
5     return 2;
6 }
```

```
1 | ul.sf-menu li li, li li li, li li li li, li li li li li, li li li li li li { width: 120px; } /* OH GOD WHY */
```

```
1 |     function isNegative(number) {
2 |         if (number.match(/^\-\d+$/)) {
3 |             return true;
4 |         } else {
5 |             return false;
6 |         }
7 |     }
```

```
1 || .ninja {  
2 ||     visibility:hidden;  
3 || }
```

```
1 // John! If you'll svn remove this once more,  
2 // I'll shut you, for God's sake!  
3 // That piece of code is not "something strange"!  
4 // That is THE AUTH VALIDATION.
```

```
1 | return 2; // returns 2
```

```
1 | <noscript><script type="text/javascript">alert("You do not have JavaScript enabled. Please enable JavaScript.</script></noscript>
```

```
1 | <!-- Renders an horizontal line -->
2 |
3 | <center>_____</center>
```

```
1 | <form>
2 |   Login : <input type=textbox name=login><br>
3 |   Password : <input type=password name=pass><br>
4 |   <input type=button value="OK" onClick="if (login.value=='admin'&&pass.value=='123456') window.location.replace('admin.php'); ">
5 | </form>
```