

Clean Code Days 2016

28.06.2016

<http://slides.com/carstenwindler/the-lone-stranger>

The Lone Stranger

Why you can't establish
Clean Code on your own



Carsten Windler, HolidayPirates GmbH

Urlaubspiraten

```
if ( is_array($fields) ) {  
    if ( isset($fields['blog_id']) ) {  
        $blog_id = $fields['blog_id'];  
    } && isset($fields['path']) ) {  
        $path = $fields['path'];  
        $blog_lookup = $wpdb->get_row( $wpdb->prepare("SELECT * FROM $wpdb->blogs WHERE domain IN (%s,%s) AND path = %s ORDER BY CHAR_LENGTH(domain)", $domains, $paths, $path), 0, 4 ) == 'www.' ) {  
            $fields['domain'] = $domains[0];  
            $wpdb->prepare( "SELECT * FROM $wpdb->blogs WHERE domain = %s AND path = %s", $fields['domain'], $fields['path'] );  
            $blog_id = $wpdb->get_row( $wpdb->prepare( "SELECT * FROM $wpdb->blogs WHERE domain = %s AND path = %s", $fields['domain'], $fields['path'] ), 0, 4 )->blog_id;  
            $blog_id = $wpdb->get_row( $wpdb->prepare( "SELECT * FROM $wpdb->blogs WHERE domain = %s AND path = %s", $fields['domain'], $fields['path'] ), 0, 4 )->blog_id;  
        }  
    }  
}
```

```

11 ( substr( $fields|
    $snowww = subs
    $blog = $wpdb
  } else {
    $blog = $wpdb
  }
  if ( $blog ) {
    wp_cache_set(
      $blog_id = $b
    } else {
      return false;
    }
  } else {
    return false;
  }
} else {
  if ( ! $fields )
    $blog_id = get_current_blog_id();
  elseif ( ! is_numeric( $fields ) )
    $blog_id = get_id_from_blogname( $fields );
  else
    $blog_id = $fields;
}

$blog_id = (int) $blog_id;

$all = $get_all == true ? ' ' : 'short';
$details = wp_cache_get( $blog_id . $all, 'blog-details' );

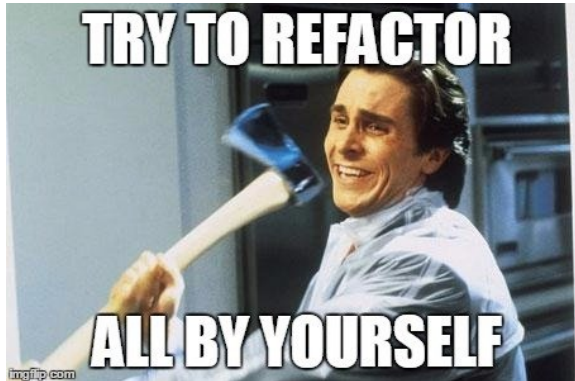
if ( $details ) {
  if ( ! is_object( $details ) ) {
    if ( $details == -1 ) {
      return false;
    } else {

```

```
omain IN (%s,%s) ORDER BY CHAR_LENGTH(domain) DESC",
omain = %s", $fields['domain'] ) );
```



- unprofessional
- will make you unhappy



- a.k.a. *Sisyphos*[™]
- other developers will keep the code rotting
- will introduce new bugs
- leads to trouble with other developers

So the answer is
not really surprisingly

Clean Code!

... but how?



So you
decided to
introduce
Clean Code
to your
project



Introducing Clean Code to an existing project

- takes time
- is painful
- is cumbersome

- ...is totally worth it

People you might need to convince:

Customer

Management

Co-Worker

Other
departments

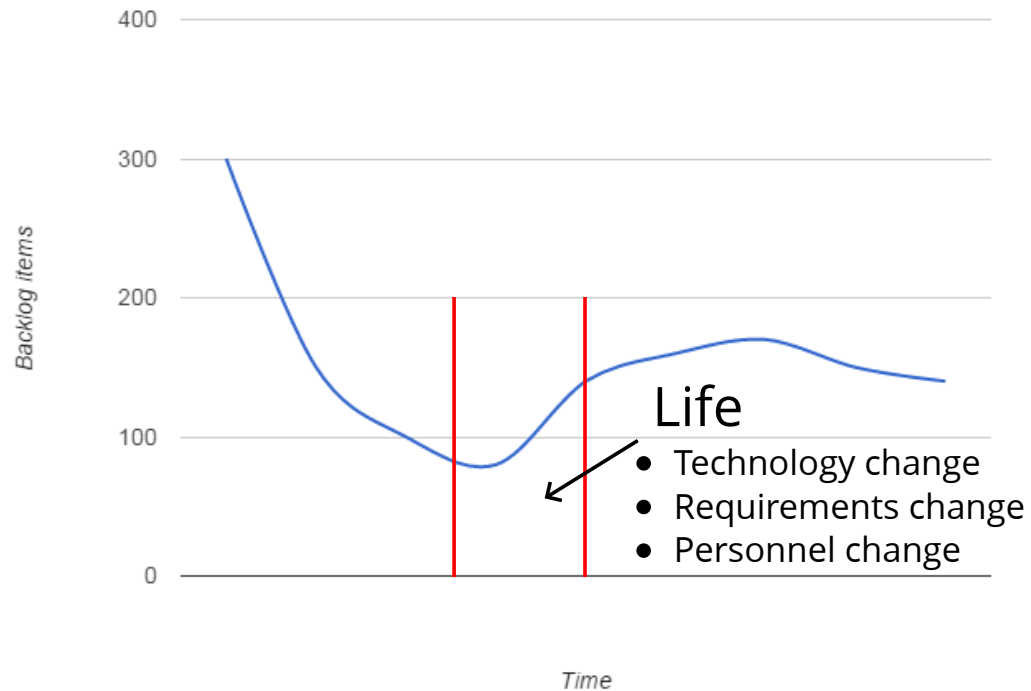
#1 - The Arguments

The Management

Now you have to explain why you want
to spent more time on doing the same
stuff as before.

The Business likes charts.
Let's give them some charts.

Project without Clean Code*



*simplification; not necessarily related to any real project

Technology change

- Framework update
- Language update
- Infrastructure change because of technical need

Requirements change

- New top prio feature requires many changes

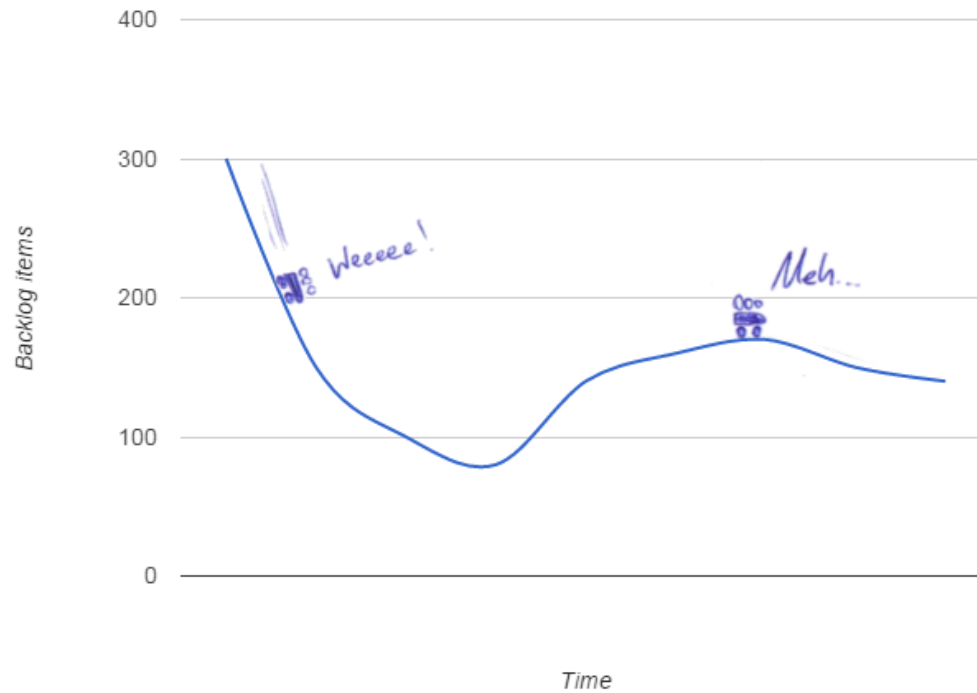
Personnel change

- Truck factor hits you

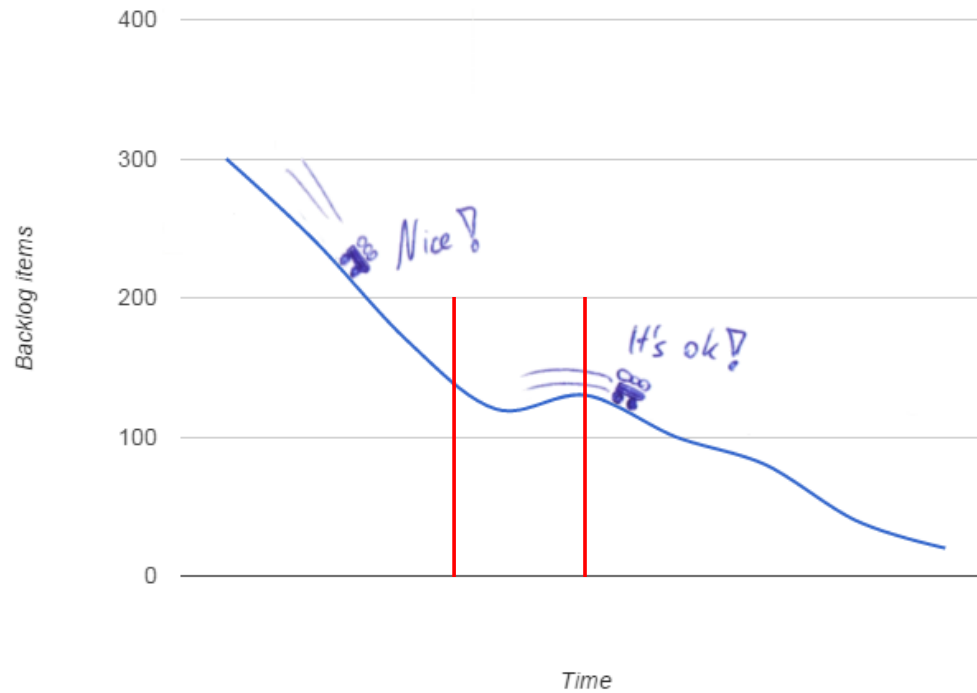
Truck Factor explained



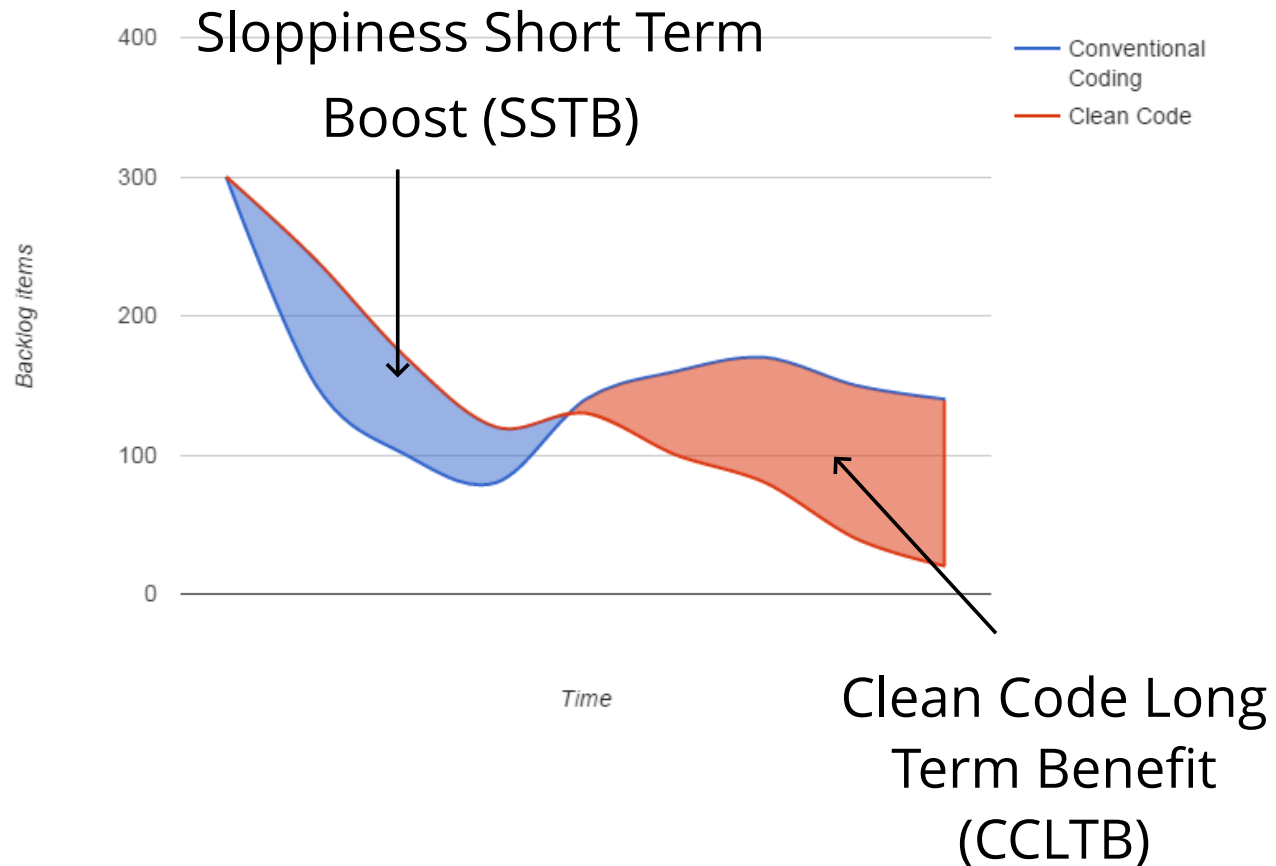
Project as a rollercoaster



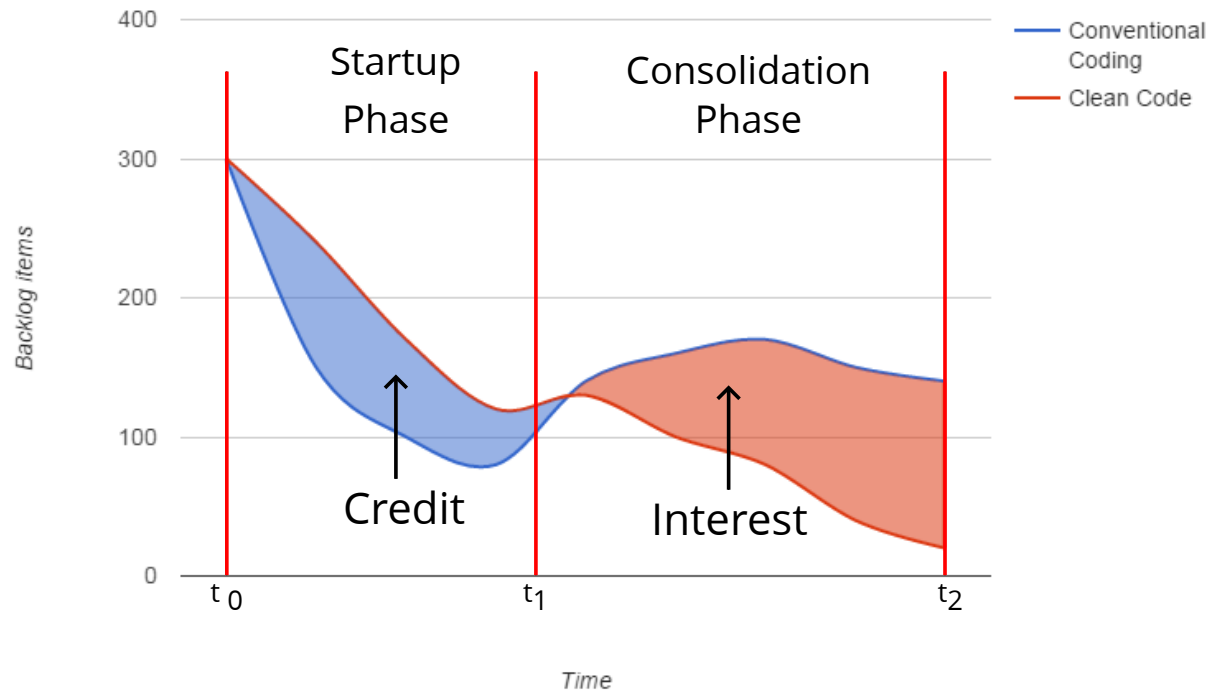
Project with Clean Code

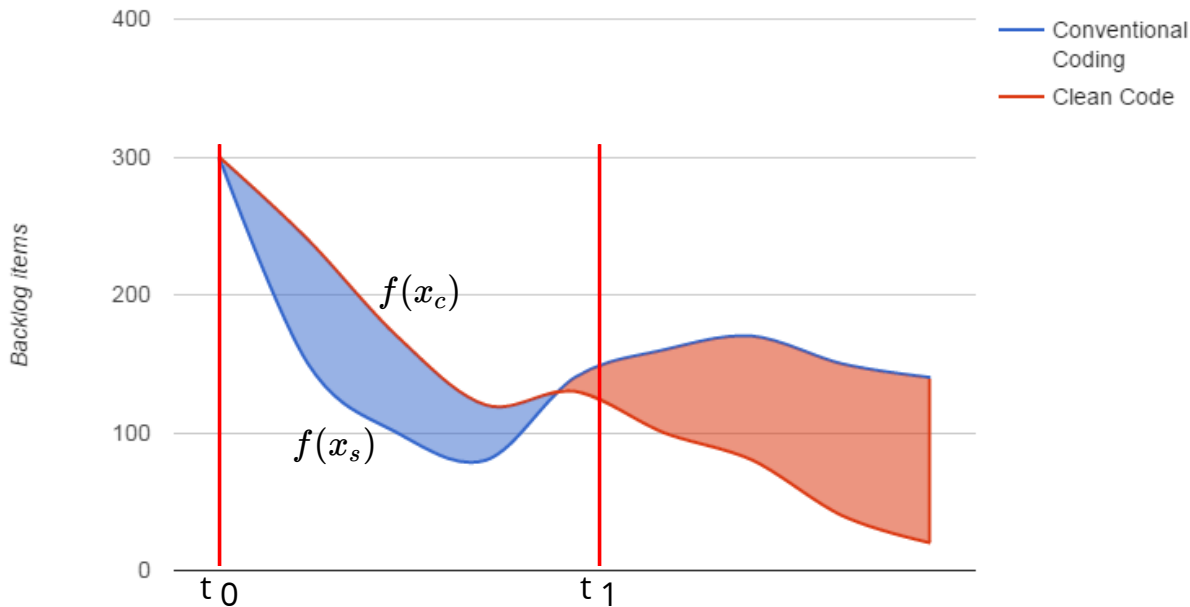


Projects compared



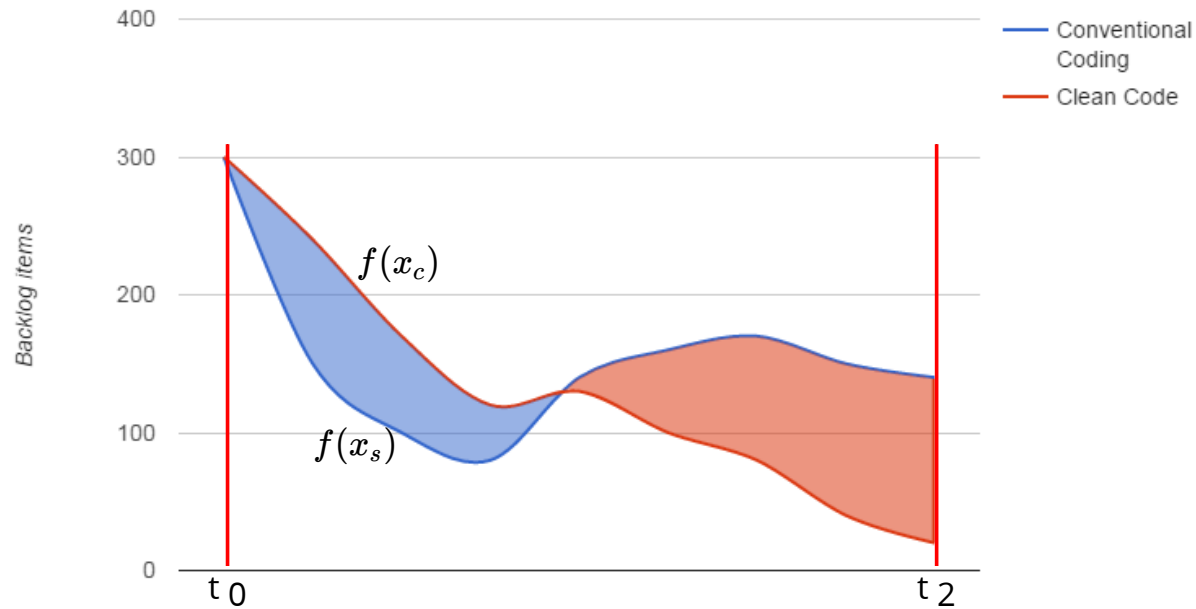
Time considerations





$$\int_{t_0}^{t_1} f(x_s) dx < \int_{t_0}^{t_1} f(x_c) dx$$

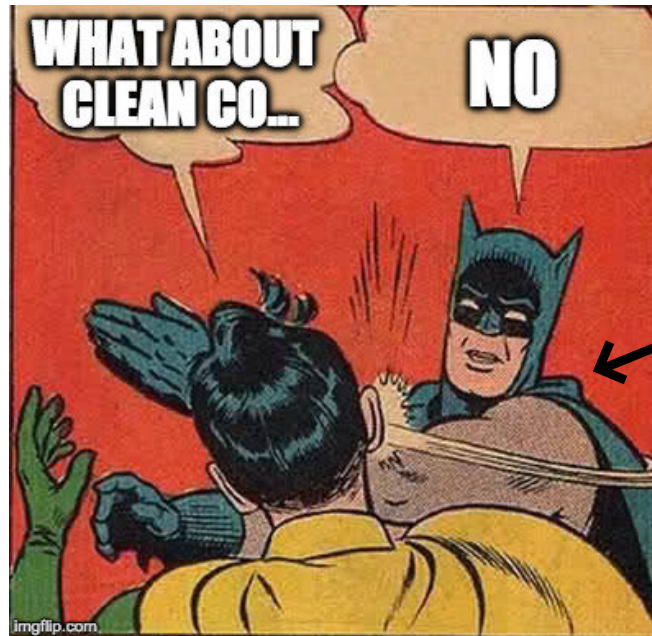
fewer backlog items are left when doing sloppy programming instead of clean code in the startup phase



$$\int_{t_0}^{t_2} f(x_s) dx > \int_{t_0}^{t_2} f(x_c) dx$$

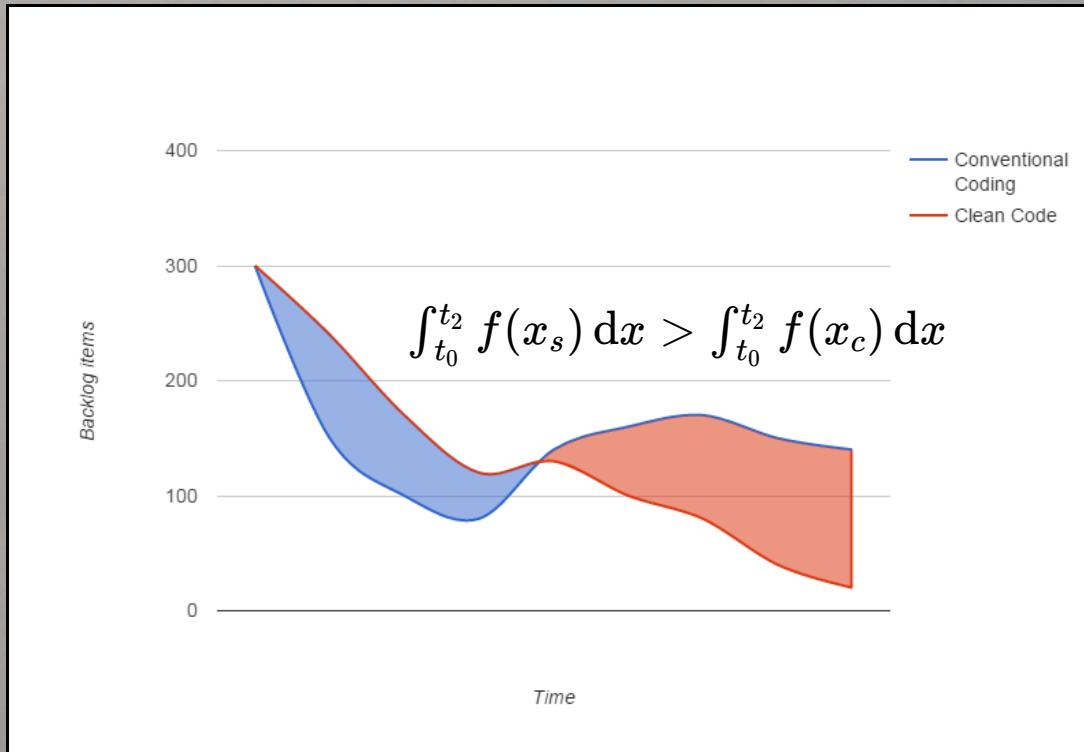
more backlog items left when doing sloppy programming instead of Clean Code over project lifetime

Sometimes it's a Business
decision to make debts



Sometimes it's ignorance



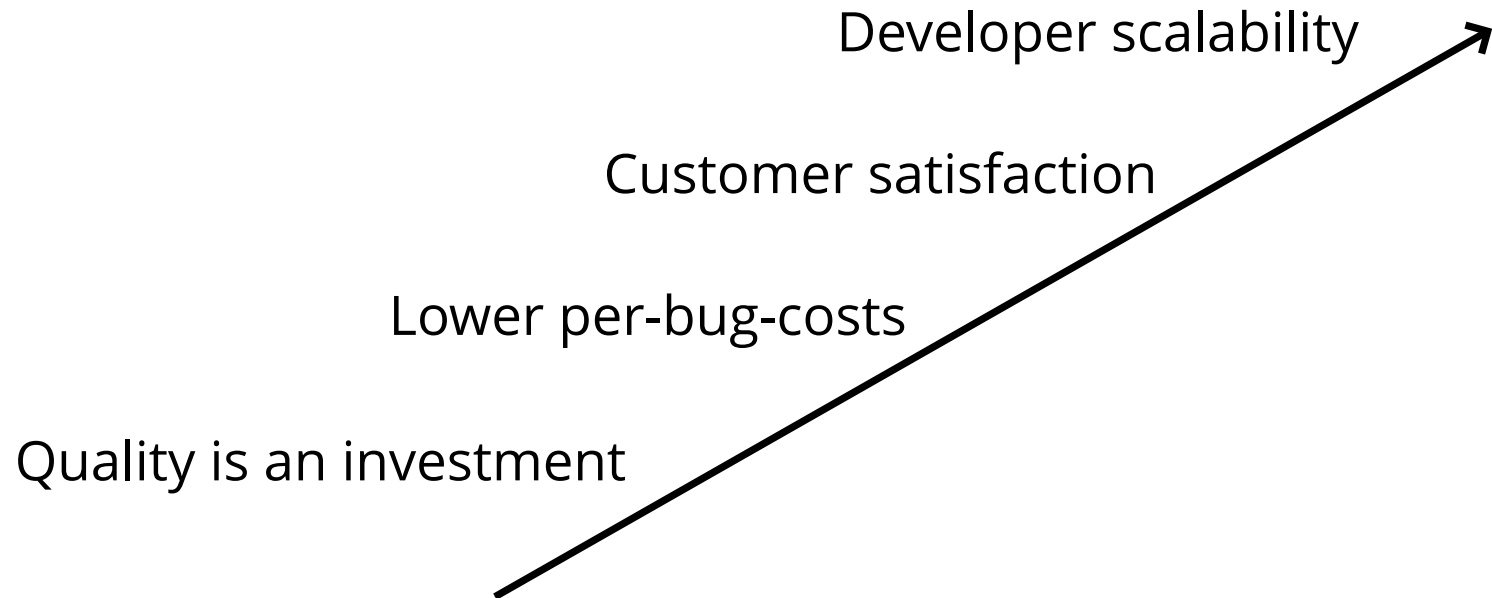


Wait, wait, wait... it's for the
Management, so keep it simple!

???



Why Clean Code matters



Quality as an investment

- don't think short term
- future changes on platform or requirements will be easier
- avoid piling up technical debt

Developer scalability

- new developers can be added to a project easier
- gathered experience and techniques can be transferred to other projects

Lower per-bug-costs

- catching a bug on development stage is much cheaper than on Live
- preventing regressions with live outage

Customer Satisfaction

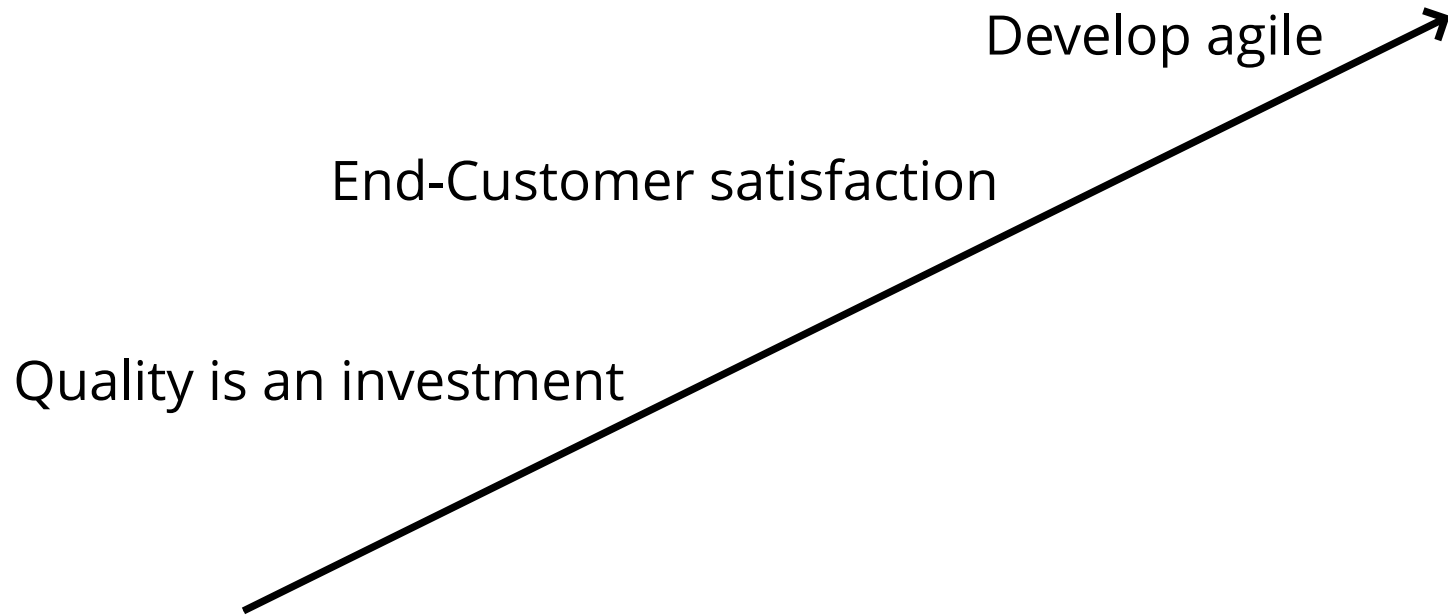
- Customer experiences more stable product
- better reputation

The Customer

Leave it up to the Management, if you can!

Now you have to explain why you want them to spent more money on getting the same results as before.

Why Clean Code matters for your customer



Quality is an investment

- don't think short term
- future changes on platform or requirements will be easier
- Customer is lesser depending on your company*

*you probably don't want to tell that though

End-Customer satisfaction

- more stable product
- happier customers
- better reputation
- lesser outages

Develop agile

- Introduce and evaluate new features faster

Co-Workers

Now you have to explain why
you want them to spent more
time on doing the same stuff
than before

Yes, once again!

Happy refactoring

Ego

Saves development time

Lesser stress during deploys

Developer scalability

Happy refactoring

- tests give you a much more confidence when refactoring
- thus it will encourage to refactor if needed

Saves time

- automated tests can save a lot of time
- you don't need to click through half of the application to test a single small feature
- bugs are usually easier to solve when you just worked on a topic than weeks after

Ego

- bugs discovered by others are often a bit embarrassing
- overall project / developer reputation increases

Lesser stress during deploys

- deploy with confidence

Developer scalability

- easier to integrate new colleagues

Turn developers from this...



... into this



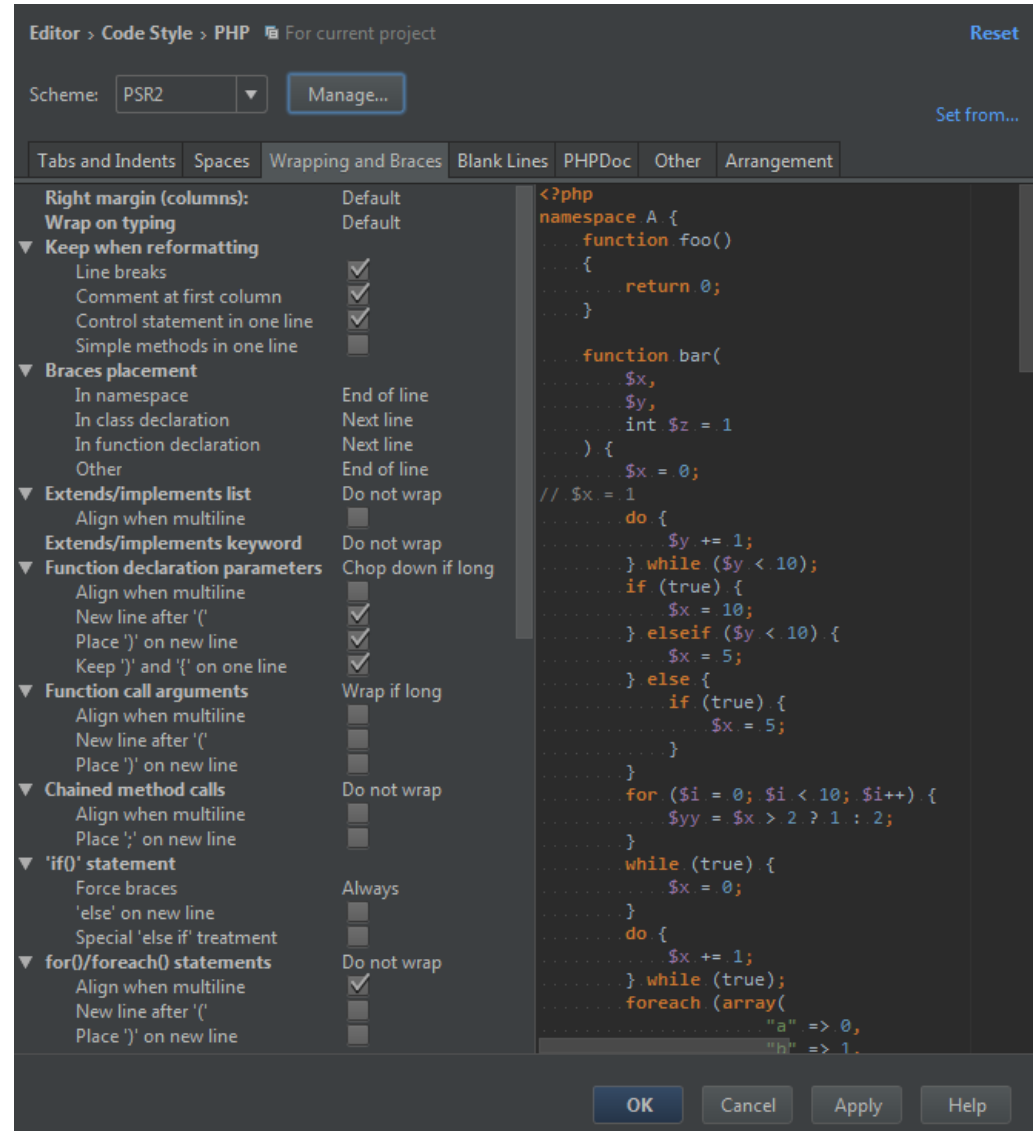
How to get developers on your side

- live what you mean
- provide good examples on the project
- inhouse workshops
- find allies
- provide *THE TOOLS*

#2 The Tools

Code Style Guides

- Improve readability
- A matter of professionalism
- Very easy to introduce



Code Style Guides








- best use existing code standards
- sniffers help you check and reformat automatically
- run e.g. on pre-commit on changed files only

```
31 | ERROR | [x] PHP keywords must be lowercase; expected "array"
    |       | but found "Array"
31 | ERROR | [x] Opening brace should be on a new line
56 | ERROR | [x] Opening brace should be on a new line
64 | ERROR | [x] Opening brace should be on a new line
72 | ERROR | [x] Opening brace should be on a new line
80 | ERROR | [x] Opening brace should be on a new line
88 | ERROR | [x] Opening brace should be on a new line
96 | ERROR | [x] Opening brace should be on a new line
-----
PHPCBF CAN FIX THE 9 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

FILE: /home/vagrant/Code/hotel-reviews/src/Review.php
-----
FOUND 1 ERROR AFFECTING 1 LINE
-----
 1 | ERROR | [x] End of line character is invalid; expected "\n" but
    |       | found "\r\n"
-----
PHPCBF CAN FIX THE 1 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

FILE: /home/vagrant/Code/hotel-reviews/src/ReviewSet.php
-----
FOUND 10 ERRORS AFFECTING 9 LINES
-----
 1 | ERROR | [x] End of line character is invalid; expected "\n"
    |       | but found "\r\n"
14 | ERROR | [x] Opening brace of a class must be on the line after
    |       | the definition
52 | ERROR | [x] PHP keywords must be lowercase; expected "array"
    |       | but found "Array"
52 | ERROR | [x] Opening brace should be on a new line
98 | ERROR | [x] Opening brace should be on a new line
106 | ERROR | [x] Opening brace should be on a new line
114 | ERROR | [x] Opening brace should be on a new line
121 | ERROR | [x] Opening brace should be on a new line
129 | ERROR | [x] Opening brace should be on a new line
```

Software Metrics

Name	loc	lloc	CommW	Length	Volume	Vocabulary	Eff.	MI	MlwC	Diff	Bugs	IC	CC	Dist.	Oper.	lcom	rsysc	rdc	rsc	Abstr.	Inst.	EC	AC	noc	noca	nocc	noi
 guzzle (3)	11133	3487	27.84	27978	2910.26	66.71	110076.8	75.42	47.58	20.92	0.97	107.81	10.71	1.2	78	4.8	4.81	0.64	4.18	0.06	0.87	3.85	0.55	63	4	59	4
 guzzle/src (17)	5507	1220	34.66	10677	1990.01	60.94	81680.99	87.91	53.24	20.44	0.66	77.72	16.61	1.97	71	1.92	2.45	0.92	1.54	0.12	0.78	3.14	0.89	34	4	30	4
guzzle/src/functions_include.php	6	1	29.56	5	11.61	5	23.22	121.84	92.28	2	0	5.8	2	0	0	0	0	0	0		0	0	0	0	0	0	0
guzzle/src/RequestOptions.php	244	28	48.77	112	564.97	33	1129.94	97.8	49.03	2	0.19	282.49	1	0	0	0	0	0	0		0	0	1	1	0	1	0
guzzle/src/Middleware.php	254	60	39.85	447	2849.64	83	107321.93	75.39	35.54	37.66	0.95	75.66	11	1	1	11	2.98	1.8	1.18		0.89	8	1	1	0	1	0
guzzle/src/UriTemplate.php	241	62	32.95	568	3621.02	83	167261.06	64.36	31.41	46.19	1.21	78.39	34	6	6	2	2.5	1.9	0.6		0.5	1	1	1	0	1	0
guzzle/src/HandlerStack.php	273	56	40.85	521	3438.55	97	138243.47	75.26	34.41	40.2	1.15	85.53	20	0	0	3	2.1	1.67	0.44		0.71	5	2	1	0	1	0
guzzle/src/Pool.php	123	30	38.65	238	1388.23	57	32775.72	83.35	44.7	23.61	0.46	58.8	8	0	0	4	5.15	1.55	3.6		0.8	4	1	1	0	1	0
guzzle/src/TransferStats.php	126	22	42.7	140	745.07	40	8327.96	92.9	50.2	11.18	0.25	66.66	3	0	0	1	1.63	1.19	0.44		1	2	0	1	0	1	0
guzzle/src/MessageFormatter.php	182	58	35.42	425	2647.25	75	90594.71	66.93	31.51	34.22	0.88	77.35	45	0	0	1	4.19	1.44	2.75		0.71	5	2	1	0	1	0
guzzle/src/functions.php	329	57	40.59	427	2906.74	112	88956.28	72.93	32.34	30.6	0.97	94.98	38	2	2	0	0	0	0		0	0	0	0	0	0	0
guzzle/src/Client.php	401	98	35.07	1021	7169.84	130	373855.81	56.03	20.96	52.14	2.39	137.5	64	3	3	1	3.46	1.22	2.23		1	8	0	1	0	1	0
guzzle/src/RedirectMiddleware.php	231	50	33.79	479	3182.41	100	131009.09	69.51	35.72	41.17	1.06	77.31	20	5	5	2	6.72	1.09	5.63		0.73	8	3	1	0	1	0
guzzle/src/RetryMiddleware.php	111	22	35.58	192	1110.02	55	24351.09	84.17	48.59	21.94	0.37	50.6	6	0	0	2	2.98	1.83	1.14		0.67	2	1	1	0	1	0
guzzle/src/PrepareBodyMiddleware.php	112	25	33.42	249	1482.59	62	36390.97	78.03	44.61	24.55	0.49	60.4	20	6	6	1	2.94	1.28	1.67		0.5	2	2	1	0	1	0
guzzle/src/ClientInterface.php	84	12	48.44	71	348.39	30	1532.91	106.96	58.52	4.4	0.12	79.18	1	0	0	0	2.2	1.8	0.4		0.5	1	1	1	1	0	1
 guzzle/src/Handler (8)	1591	429	34.47	3955	3591.71	93.63	167333.81	77.79	43.32	25.79	1.2	107.55	25	2.63	21	3	3.37	1.06	2.31	0.13	0.86	5.25	0.88	8	1	7	1
 guzzle/src/Exception (9)	285	55	26.35	404	254.16	17.76	6051.05	107.4	81.05	4.71	0.08	31.08	2.56	0.11	1	0.89	1.61	0.34	1.26	0.11	0.77	1.11	0.33	9	1	8	1
 guzzle/src/Cookie (5)	914	155	39.86	1423	1830.58	64.6	69122.31	86.03	46.17	23.06	0.61	85.41	20.4	5.2	26	1.8	2.03	0.96	1.07	0.2	0.68	3	1.4	5	1	4	1
 guzzle/tests (16)	5215	2149	18.51	16465	4148.4	74.56	147240.27	58.91	40.4	21.07	1.38	150.63	2.44	0.11	3	8.96	8.23	0.28	7.94	0	0.97	4.89	0.15	28	0	28	0
 guzzle/build (2)	411	118	31.03	836	2760.02	64.5	119494.66	73.57	42.54	27.45	0.92	71.36	16	2	4	0.5	1.19	0.38	0.81	0	1	2.5	0	1	0	1	0

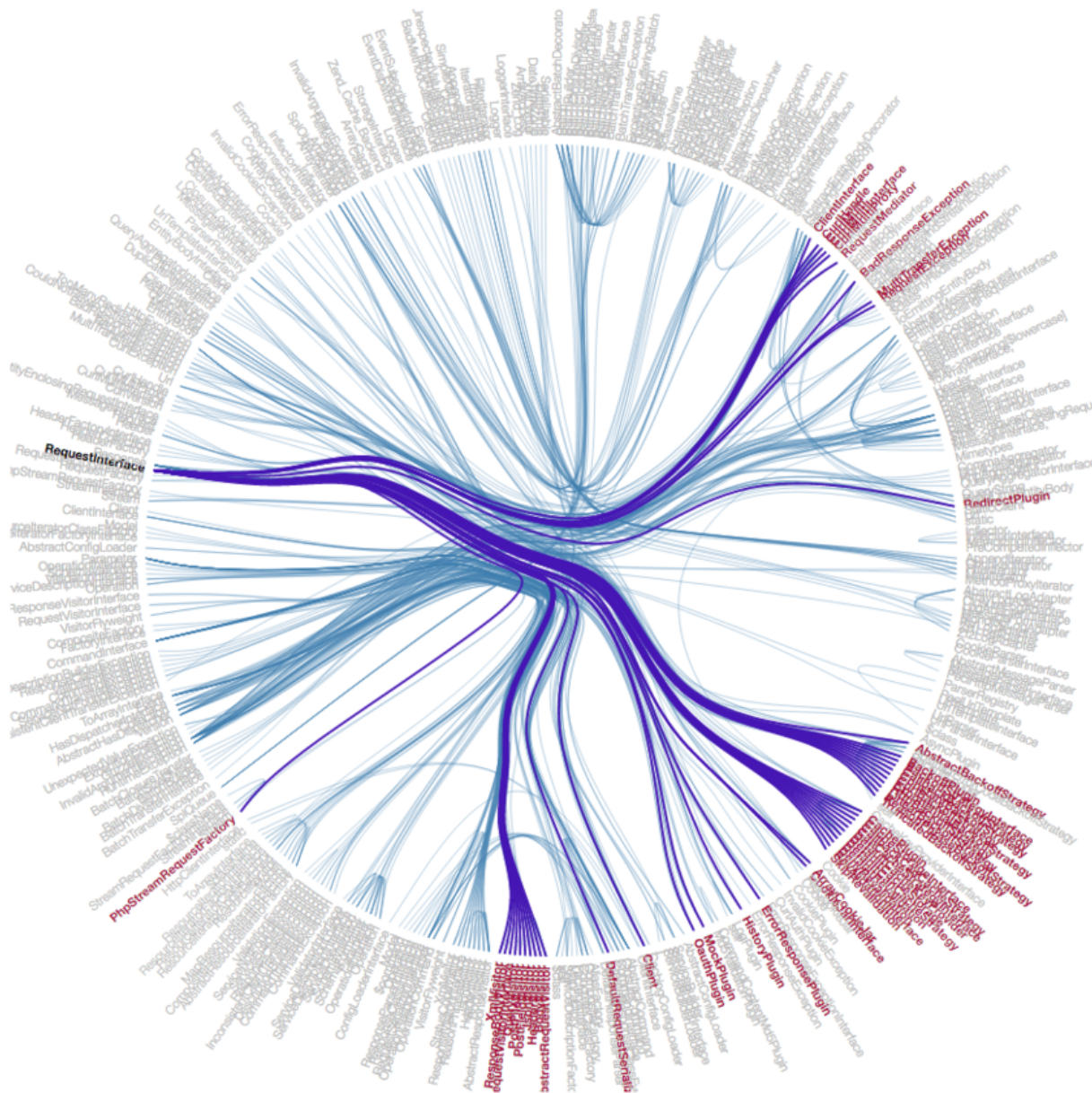
What can you do with these numbers?

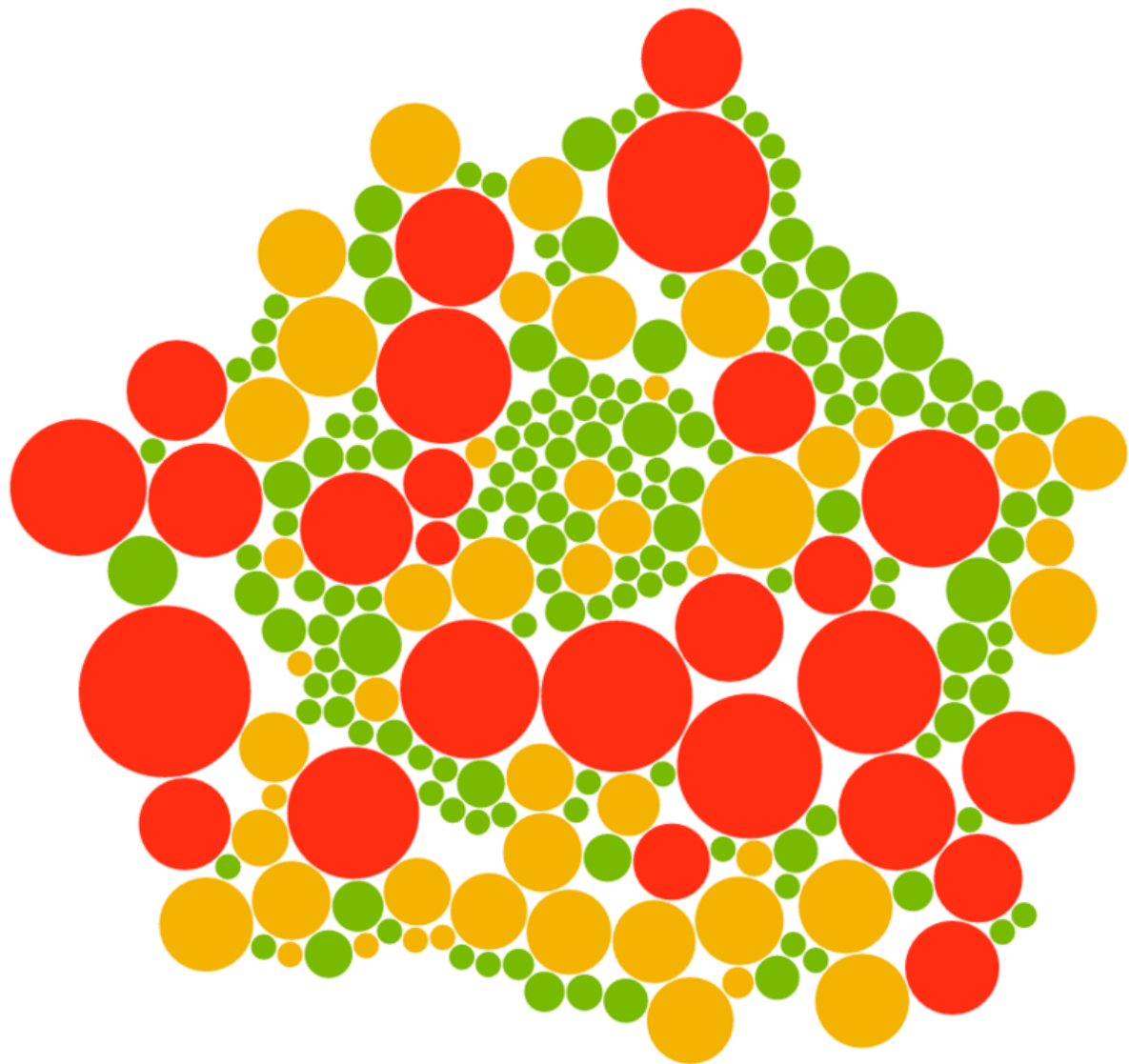
Awesome! During last the Refactoring Sprint we managed to drop the overall *cyclomatic complexity* by ~5 points (which is great). However we still have a *maintainability index* of 64.32 on the Middleware which we will change ASAP.

Hey Bob, how is your project going?

Nice

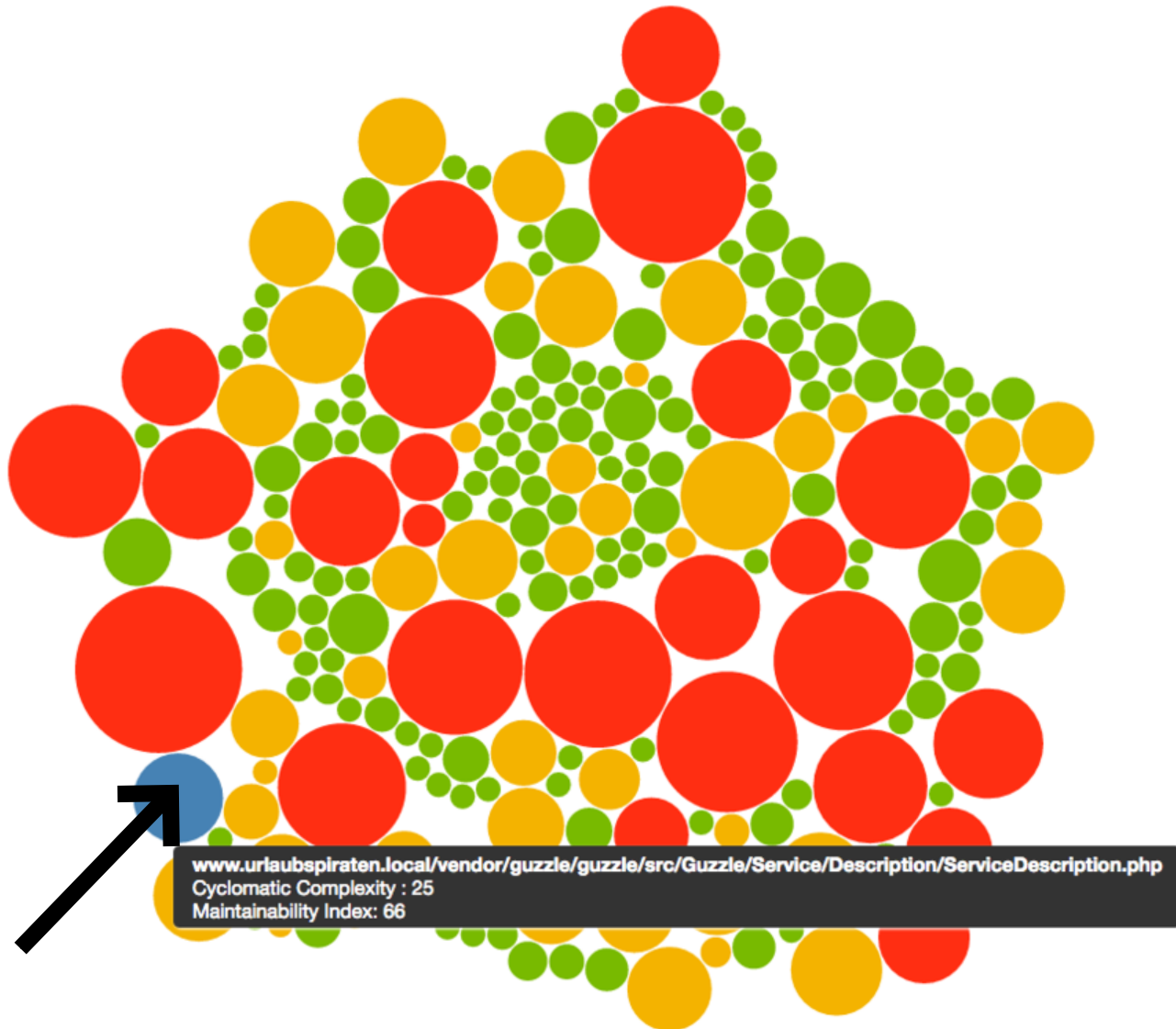








Your project on a T-Shirt





REFACTORING TIME!!

A person is shown in silhouette, standing on a dark, rocky ledge. They are looking down, away from the camera, towards the horizon. The background is a bright, hazy sunset or sunrise, with the sun low on the horizon, creating a strong glow and lens flare. The sky is a mix of orange and yellow, and the distant landscape is visible in silhouette.

Why is my code so bad?

Software Metrics

- easy to introduce
- will show you how good / how bad your codebase actually is
- make developers aware of software quality
- helps identifying problem areas

Source Code Analyzer

- easy to introduce
- will give you direct hints where in your code things go wrong
- Copy & Paste detectors will not allow lazyness
- prevents bad code from getting added to the codebase

Putting the finger where it hurts

PHPMD report

Problems found

#	File	Line	Problem
1	/home/admin/domains/guzzle/src/Client.php	65	Avoid using static access to class 'GuzzleHttp\HandlerStack' in method 'construct'.
2	/home/admin/domains/guzzle/src/Client.php	145	Avoid using static access to class 'GuzzleHttp\Psr7\Uri' in method 'buildUri'.
3	/home/admin/domains/guzzle/src/Client.php	186	The method configureDefaults uses an else expression. Else is never necessary and you can simplify the code to work without else.
4	/home/admin/domains/guzzle/src/Client.php	319	The method applyOptions uses an else expression. Else is never necessary and you can simplify the code to work without else.
5	/home/admin/domains/guzzle/src/Cookie/CookieJar.php	25	The method construct has a boolean flag argument \$strictMode, which is a certain sign of a Single Responsibility Principle violation.
6	/home/admin/domains/guzzle/src/Cookie/CookieJar.php	78	The method shouldPersist has a boolean flag argument \$allowSessionCookies, which is a certain sign of a Single Responsibility Principle violation.
7	/home/admin/domains/guzzle/src/Cookie/CookieJar.php	116	The method clear uses an else expression. Else is never necessary and you can simplify the code to work without else.
8	/home/admin/domains/guzzle/src/Cookie/CookieJar.php	152	The method setCookie uses an else expression. Else is never necessary and you can simplify the code to work without else.
9	/home/admin/domains/guzzle/src/Cookie/CookieJar.php	215	Avoid using static access to class 'GuzzleHttp\Cookie\SetCookie' in method 'extractCookies'.
10	/home/admin/domains/guzzle/src/Cookie/FileCookieJar.php	24	The method construct has a boolean flag argument \$storeSessionCookies, which is a certain sign of a Single Responsibility Principle violation.

However no tool will make
you a better programmer

Never stop learning!

Code Reviews

- fairly easy to introduce*
- actually a must have
 - can prevent really nasty bugs
 - great for sharing knowledge
- be careful - don't be arrogant
- some devs prefer eye-to-eye reviews, others are ok with comments in the Merge Request

*assuming you use Version Control Systems

Tests

KISS

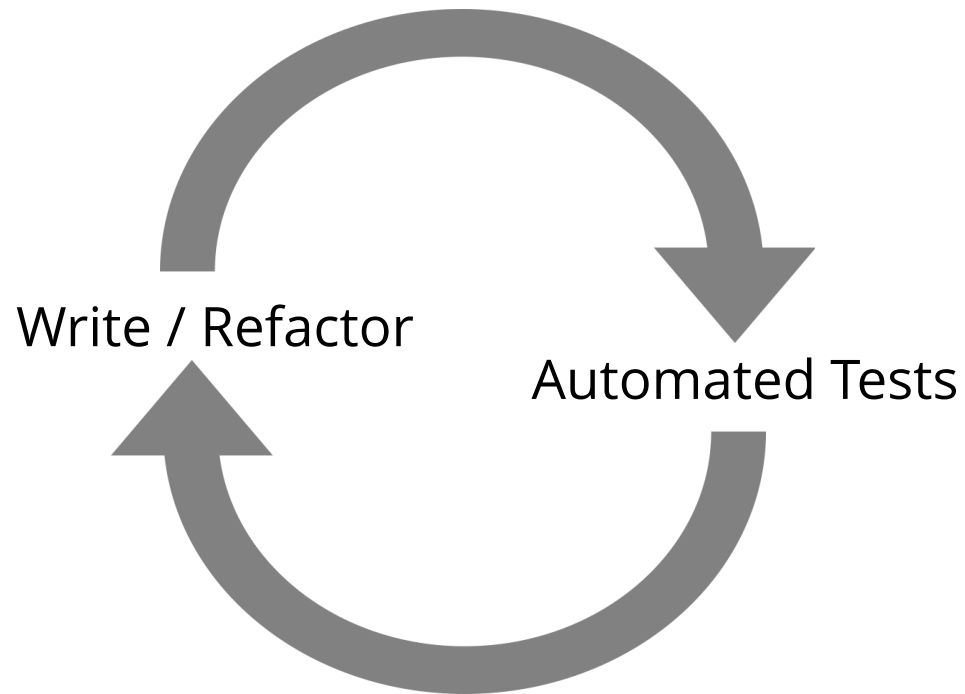
Keep It teSted, Stupid!

Tests are not optional, they are mandatory.

Period.

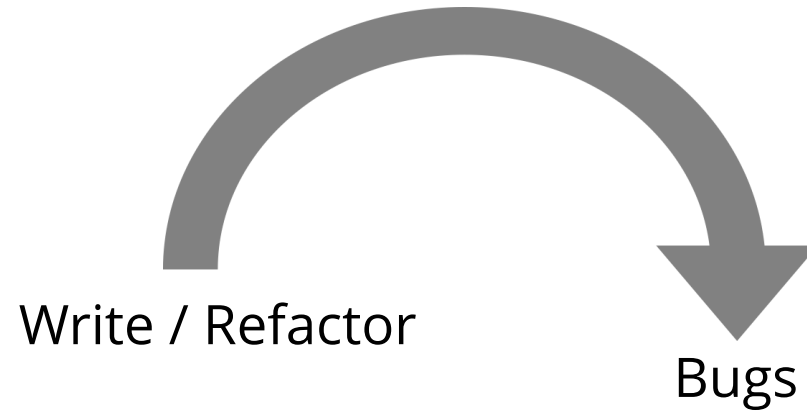
We mean it.

Seriously.



Ideal world

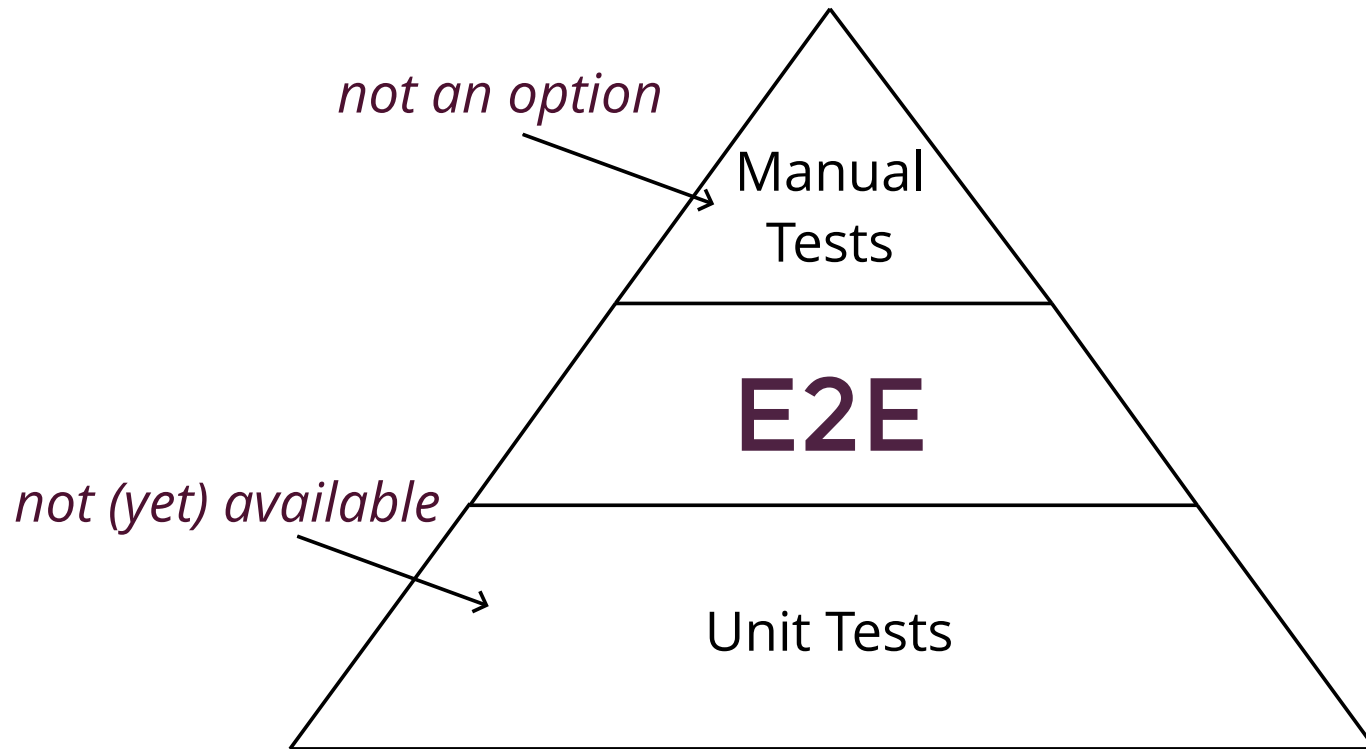
But what if we
have no Unit Tests
yet??



Harsh reality

Writing Tests for untested classes usually requires refactoring

- refactoring without tests == risky!
- chicken-egg-problem
- concentrate on Unit Tests for new classes
- for legacy code, try to find simple classes to start with



The Almighty Test Pyramid

E2E Tests

- slow
- a bit fragile
- require test mode
- + require no refactoring
- + require no deep code knowledge
- + can be done in a sane amount of time

your way to go before you start with refactoring

Workflows

- tools are nothing without control
- setup clearly documented workflows
- be stubborn, otherwise sloppiness will come back
- just like animals, some developers will SMELL any fear or weakness
- so it's a good idea to automate

Continuous Integration

- Fail build immediately
 - Code Sniffer
 - Source Code Analyzer
 - Unit Tests
 - Acceptance Tests
- Reports
 - Software Metrics
 - Test coverage

What

is

left?

**Clean Code is NOT about
being the best coder in the
world.**

(although you have to be pretty good)

**Clean Code is about
professionalism.**

Clean Code is about
knowing that sloppy code
will cost a lot of money.

And:

**Clean Code is about being
proud of your code!**

And your job.

It's probably the best job in
the world.

Thank you.

Image credits

Office work business

<https://stocksnap.io/photo/HZCASACP3N>

Top view of phone, earphones pen and diary

<https://www.pexels.com/photo/top-view-of-phone-earphones-pen-and-diary-6662/>

Guy working man

<https://stocksnap.io/photo/GDAGX2JKI4>

Red fire truck

<https://stocksnap.io/photo/I2W591P4EV>

Man wearing black and red checkered long sleeve...

<https://www.pexels.com/photo/man-wearing-black-and-red-checkered-long-sleeve-shirt-wearing-black-wayfarer-sunglasses-sitting-on-white-wooden-chair-69212/>

Man in white shirt using Macbook

<https://www.pexels.com/photo/man-in-white-shirt-using-macbook-pro-52608/>

Bow tie fashion man person

<https://www.pexels.com/photo/bow-tie-fashion-man-person-5198/>

All above images are licensed under CC0. Thank you very much for making this great work available freely!