

Telecom
valley

Soirée du
<Test Logiciel>

Why Clean Code Is Not The Norm?

Sandor DARGO

Jeudi 9 décembre 2021

Who Am I?

Sándor DARGÓ

Principal Engineer in Amadeus

Enthusiastic blogger <https://www.sandordargo.com>

(A former) Passionate traveller

Curious home baker

Happy father of two



9 DÉCEMBRE 2021

Telecom
valley

Soirée du
<Test Logiciel>

Test & IA,
Automatisation



CONFÉRENCES



ATELIERS

KPI &
indicateurs qualité

Clean code

Test & Agilité

www.telecom-valley.fr/stls2021



Agenda

What is software quality?

What is clean code?

How do they relate to each other?

Why don't we write clean code?

What is our job?

Why this talk?

Clean code is not the “de facto standard”

Developers cannot agree on what quality is

People prefer blaming over taking responsibility



Is software quality a meaningless marketing term?

It is, according to Derek Jones in [The aura of software quality](#)

It means something different to everyone

Therefore it's meaningless



Quality is undefinable - you recognize it when you see it

Classical quality

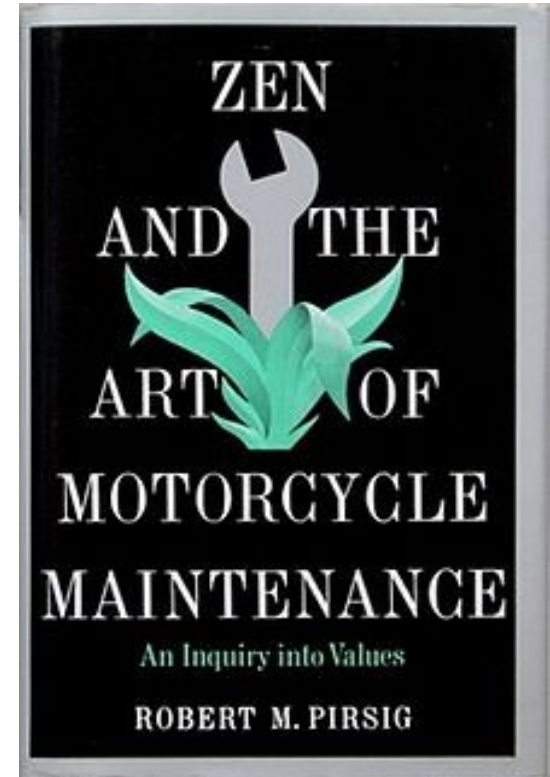
The engineering aspect

How things work

Romantic quality

How they used

Design, esthetics



What is SW quality?

ISO: Software quality is "capability of a software product to conform to requirements" while for others it can be synonymous with customer or value-creation or even defect level.

ASQ: Software quality describes the desirable attributes of software products. There are two main approaches exist: defect management and quality attributes.

Tom DeMarco: a product's quality is a function of how much it changes the world for the better.

More definitions at [Wikipedia...](#)



Consortium for Information & Software Quality™

defines the 4 pillars of structural quality

Security

Reliability

Performance efficiency

Maintainability



What is clean code?

Code that is easy to understand and easy to change



What should be easy to understand?

The execution flow of the entire application

How objects collaborate with each other

The role and responsibility of classes

What each method does

The purpose of each expression and variable

When is it easy to change the code?

Classes and methods are small and only have single responsibility

Classes have clear and concise public interfaces

The code is easily testable and has unit tests

Tests are easy to understand and easy to change

Classes and methods are predictable and work as expected

Is Clean Code part of Software Quality?

For some, no...

...as long as the code works w/o bugs

The user doesn't see it

It's simply for devs fun

Yet, not everyone finds it amusing

For others, yes!

bugs

Helps reducing the number of

Helps reducing the time to fix

Decreases maintenance costs

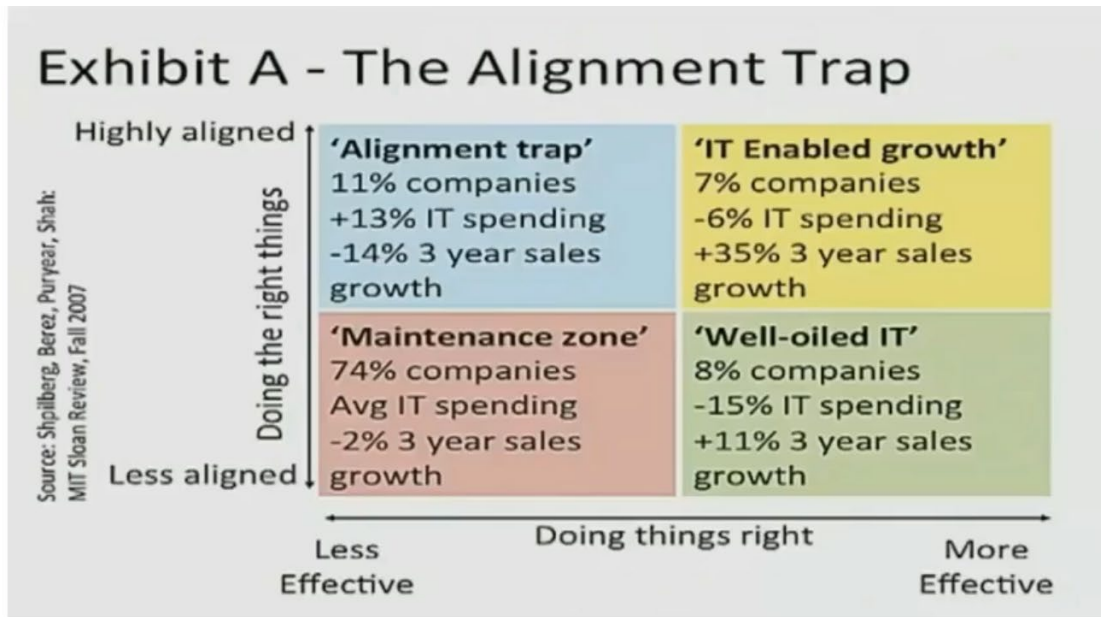
It's part of making a job right

So, is Clean Code part of Software Quality?

Making a job right should be the responsibility of all of us

It's even more important than delivering the right thing!

It pays off during the long run!



Do it Right Then Do the Right thing

Allan Kelly



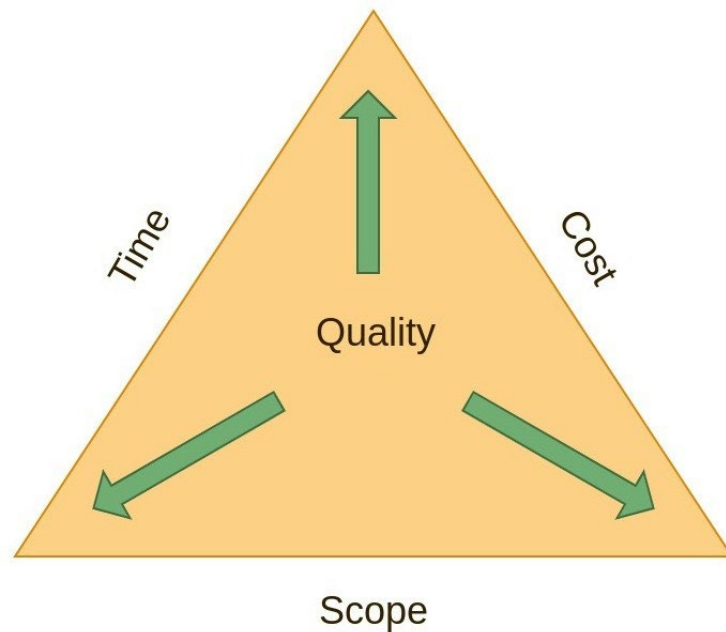
The costs of clean code

Time

Effort

Attention

Care



Not values that the business usually cares about

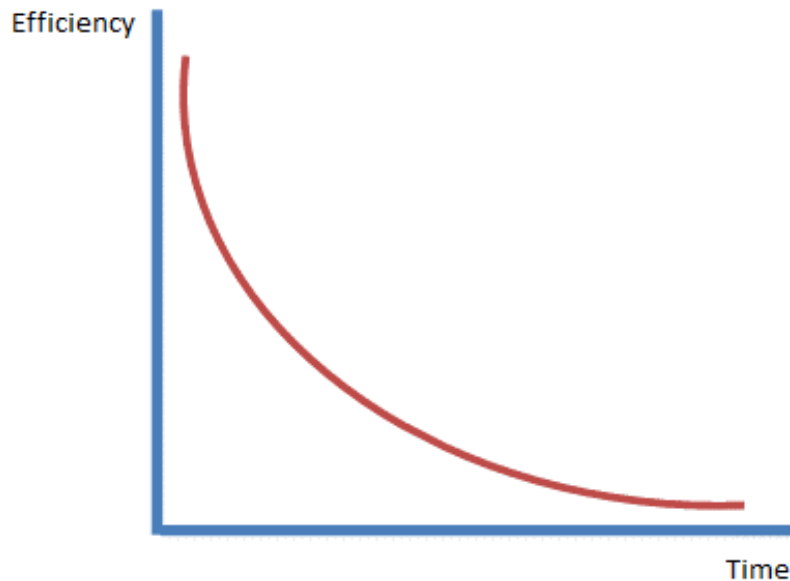
What will happen if you write bad code?

It all starts with high productivity

The number of bugs increases

Productivity slows down

Then...





But who is guilty here?

It's them, the project managers!

“Just keep delivering new features!”

“Many don't even understand what technical debt is!”

“They don't care about clean code as they don't understand!”



It's them, the junior developers!

“They don't know how to code well”

“We don't have the time to correct their code”

“They don't arrive with good enough knowledge”

The number of juniors is exploding

The number of developers doubling every 5 years

Not enough mentoring

Not enough mentors

IT'S US!

The average is average by definition

People tend to perform mediocre level

You need systems in place

You need social pressure

Clean code is hard, we must learn it!

Requires

Effort

Discipline

People are

Lazy

Disorganized

Looking for least
resistance

Let's do our job!

What's our job?

Write (clean) code!

Write tests!

Tests are still code...

Keep learning!

Our job is not only about coding!

We have to communicate!

Document

Teach

Explain

Alert

Say no!

We MUST say no!

“We have to remind *them* about the truth”

“But do we have a mandate?”

“Yes!”

“If they don't trust be ready to go”

Is it a problem that PMs don't understand clean code?

Do they understand

Inheritance?

Immutability?

The dangers of heap memory allocations?

NO, IT'S NOT THEIR JOB!

What can happen if you say NO?

Your objection might be heard

You might get convinced

It might get rejected

What can go wrong?

It's just business

Nothing personal

It might show your way!

Step up or step out

Just do what you
need to do

You're the expert, don't ask how to do your job

Do you tell your doctor how to cure you?

Do you want them to ask you how to cure you?

Do what you think is best

You are the expert

“Leave the campground cleaner than you found it”

It is not enough to check in clean new code

Dedicate some time to clean up around

Even if it's about just a couple of lines

It is going to make a difference

You need no permission!



Conclusion

Quality is hard to define

Clean code is part of software quality

It's part of our job to make the things right

It's the seniors responsibility to pass on these values

Call to action

Explain

To you colleagues

To your (product) management

Set your limits

Be ready to say no

Lead your colleagues by example

Telecom
valley

Soirée du
<Test Logiciel>

Why Clean Code Is Not The Norm?

Sandor DARGO

Jeudi 9 décembre 2021