

Why Clean Code Is Not The Norm?



Sandor Dargo



TWITTER [SandorDargo](#)

EMAIL sandor.dargo@gmail.com

Dev<Talks/>
Cluj-Napoca

Event
powered by

Google

Who Am I?

Sándor DARGÓ

Senior Engineer at **Spotify**

Enthusiastic blogger: sandordargo.com

Passionate traveller

Keen epicure

Fortunate father of two



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 clean code part of
software quality?

What is software quality?

Is software quality a meaningful term?

It is **NOT**, according to Derek Jones ([The aura of software quality](#))

Is software quality a meaningful term?

It is **NOT**, according to Derek Jones ([The aura of software quality](#))

People in industry are very interested in software quality, and sometimes they have the confusing experience of talking to me about it. My first response, on being asked about software quality, is to ask what the questioner means by software quality. After letting them fumble around for 10 seconds or so, trying to articulate an answer, I offer several possibilities (which they are often not happy with). Then I explain how “software quality” is a meaningless marketing term. This leaves them confused and unhappy. People have a yearning for software quality which makes them easy prey for the snake-oil salesmen.

Is software quality a meaningful term?

It is **NOT**, according to Derek Jones ([The aura of software quality](#))

Software quality means something different to everyone

People in industry are very interested in software quality, and sometimes they have the confusing experience of talking to me about it. My first response, on being asked about software quality, is to ask what the questioner means by software quality. After letting them fumble around for 10 seconds or so, trying to articulate an answer, I offer several possibilities (which they are often not happy with). Then I explain how “software quality” is a meaningless marketing term. This leaves them confused and unhappy. People have a yearning for software quality which makes them easy prey for the snake-oil salesmen.

Is software quality a meaningful term?

It is **NOT**, according to Derek Jones ([The aura of software quality](#))

Software quality means something different to everyone

Therefore it's a meaningless marketing term

People in industry are very interested in software quality, and sometimes they have the confusing experience of talking to me about it. My first response, on being asked about software quality, is to ask what the questioner means by software quality. After letting them fumble around for 10 seconds or so, trying to articulate an answer, I offer several possibilities (which they are often not happy with). Then I explain how "software quality" is a meaningless marketing term. This leaves them confused and unhappy. People have a yearning for software quality which makes them easy prey for the snake-oil salesmen.

Quality is undefinable

You recognize it when you see it

Two approaches to quality

Classical quality

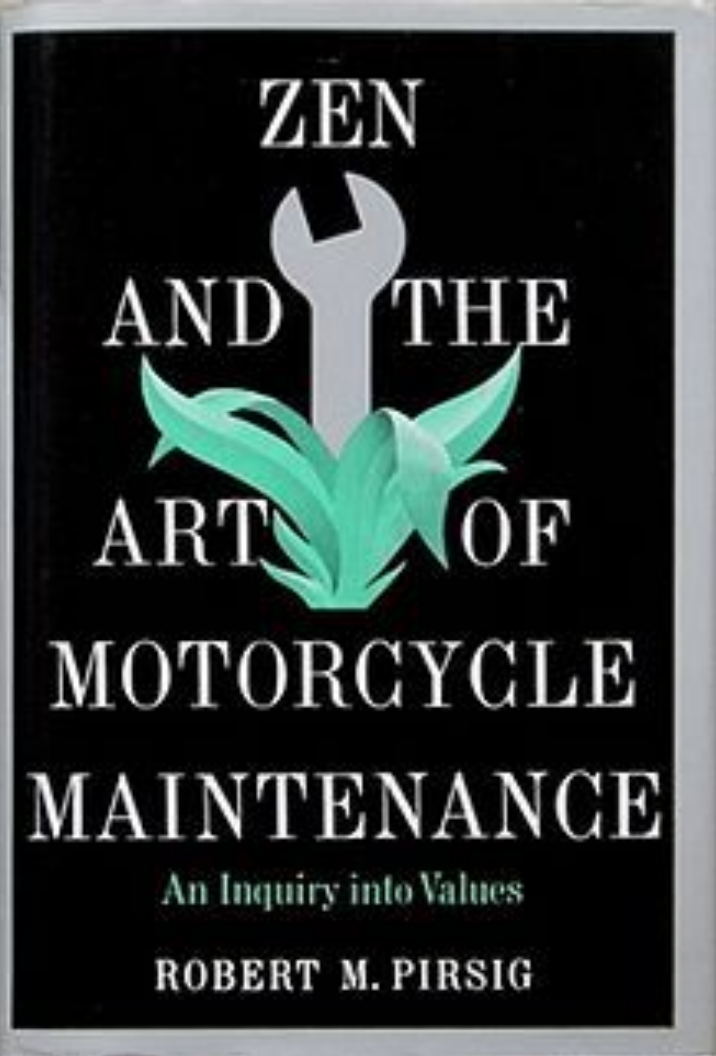
The engineering aspect

How things work

Romantic quality

How things are used

(External) Design, aesthetics



What is software 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...](#)



defines the 4 pillars of structural quality

Security

Reliability

Maintainability

Performance efficiency

Clearly a classical approach

What is clean code?

Code that is easy to understand
and easy to change

It should be easy to understand the ...

Execution flow

Object relationships

Roles and responsibilities

Purpose of each expression

Code is easy to change when ...

Classes have concise public interfaces

The code is easily testable and has unit tests

Classes and methods are predictable and work as expected

Classes and methods are small and only have a single responsibility

Is *clean code* part of *software quality*?

For some, **no...**

As long as the code works
without bugs it's OK

The user doesn't see it anyway

It's simply fun for devs

Yet, not everyone finds it amusing

For others, **yes!**

Helps reduce the number of bugs

Helps reduce the time to fix bugs

Decreases maintenance costs

It's part of making a job right

Why clean code is often expendable?

Clean code is costly!

Attention

Effort

Care

Time

Not costs that the business is usually ready to pay

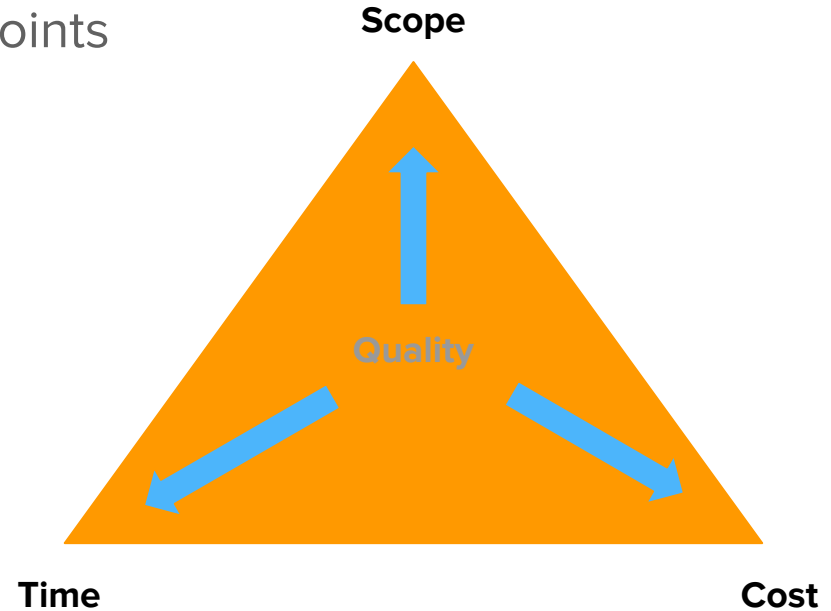


Quality is often lost in the midst of constraints

Moving one point impacts the other two points

The invisible “quality” is in the middle

It’s usually the first sacrifice



No end user cares about the quality of code

They can't care less about the

Aesthetics of code

Beauty of code

Elegance of code

Users want solutions to
their problems

And they want them
every single moment!

Why isn't clean code *l'art pour l'art*?

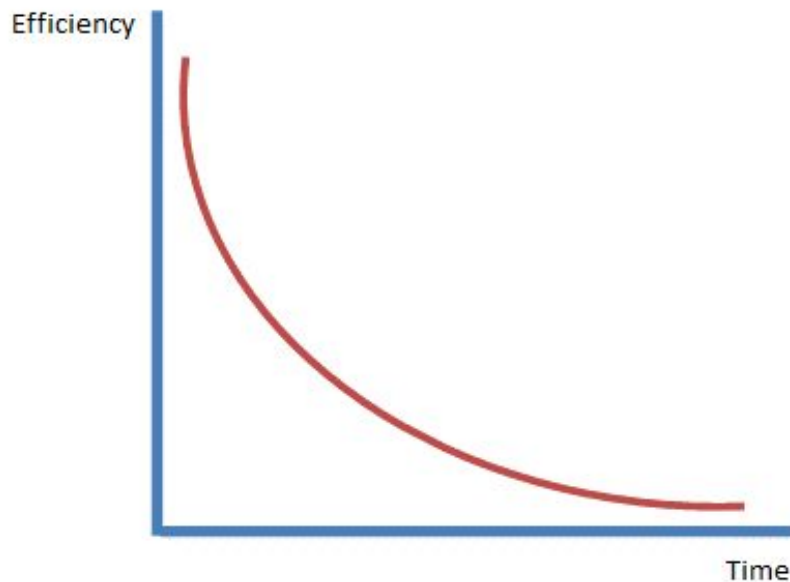
What will happen if you write ~~ugly~~ bad code?

It all starts with high productivity

The number of bugs increases

Productivity slows down

Then...

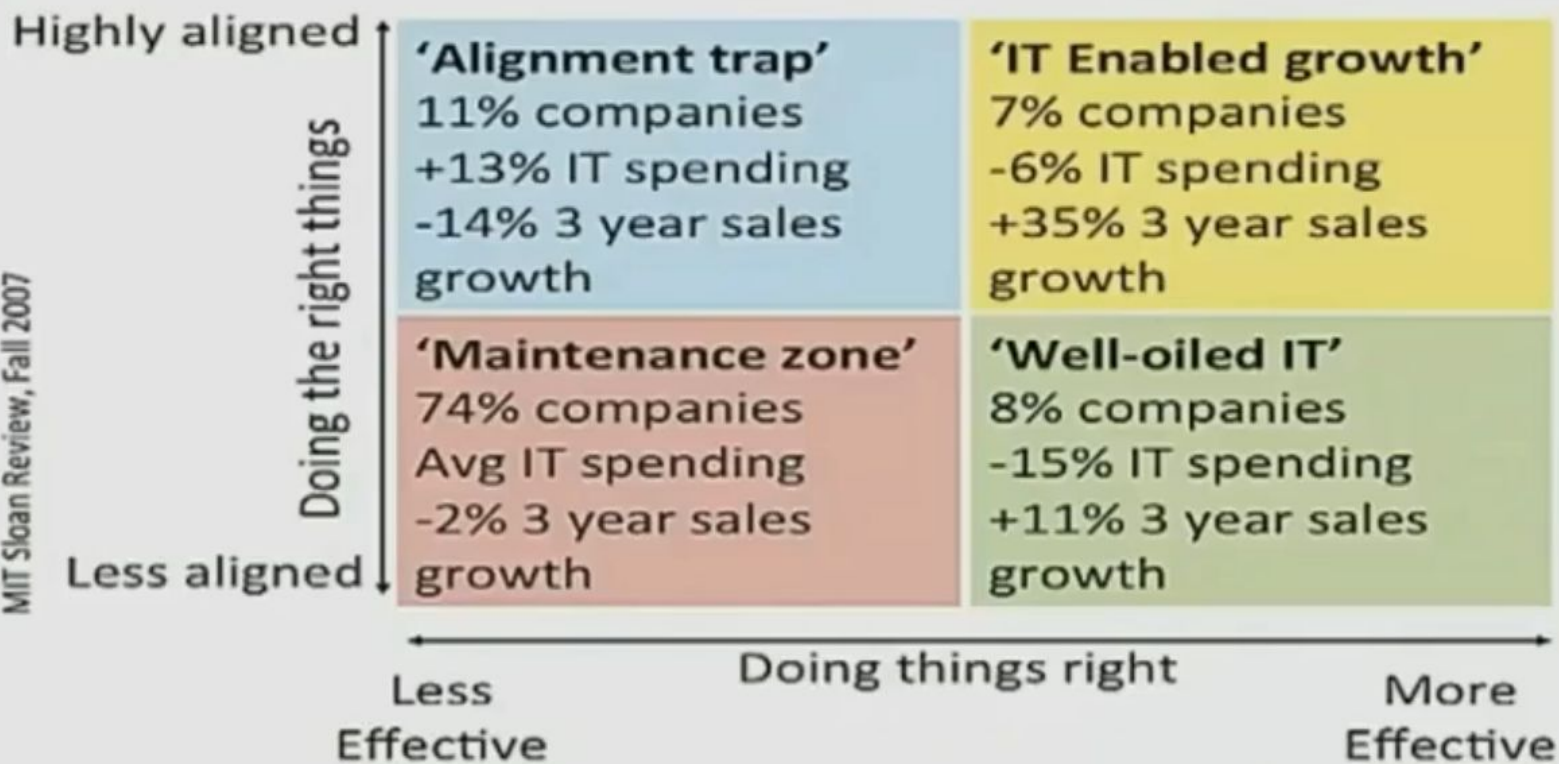




So, is clean code part of software quality?

Exhibit A - The Alignment Trap

Source: Shpilberg, Berez, Puryear, Shah:
MIT Sloan Review, Fall 2007



Do it Right Then Do the Right thing

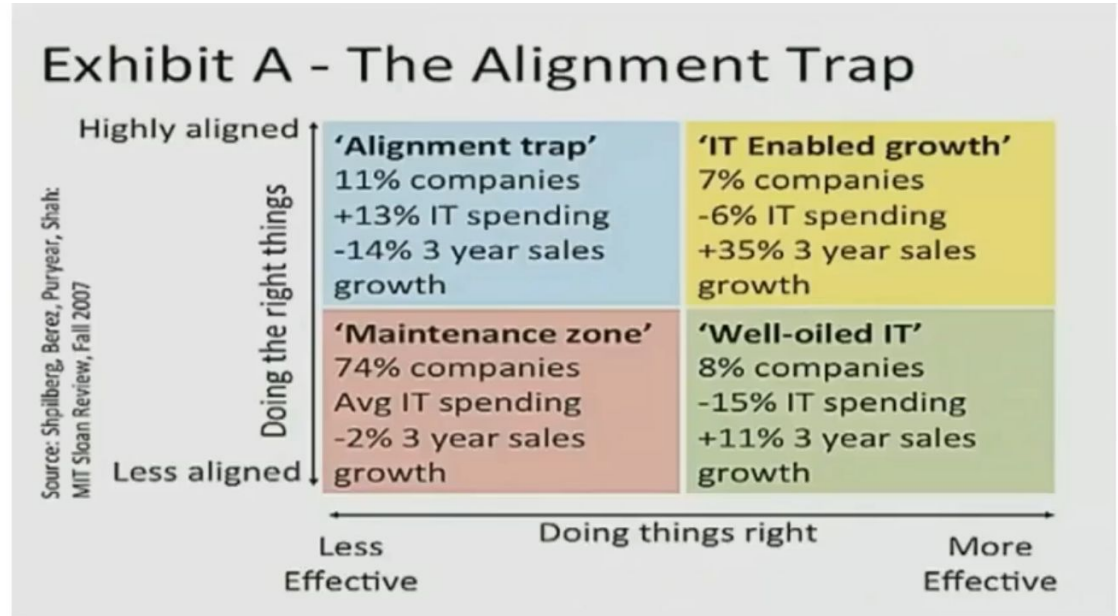
Allan Kelly

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

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!”



Should PMs understand clean code?

Do they understand

Memory safety?

Concurrency patterns?

The different forms of polymorphism?

NO, IT'S NOT THEIR JOB!

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”

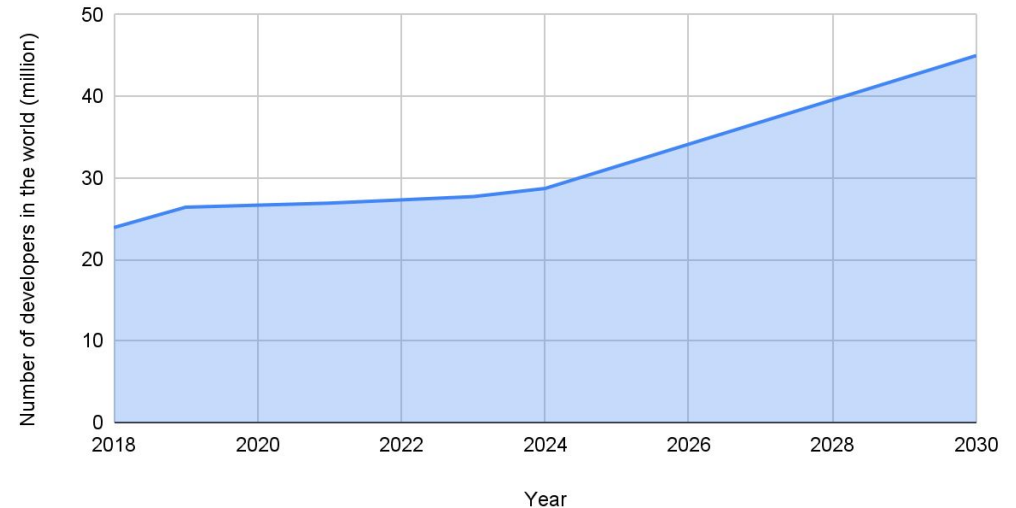
The number of juniors is exploding

The number of developers is rapidly growing

Not enough mentoring

Not enough mentors

The evolution of number of developers



IT'S US!

Clean code is hard, we must learn it and use it!

Requires

Effort

Discipline



People are

Lazy

Disorganized



But how to get organized?

To raise the bar you need systems in place

You need the right example

You need social pressure

But first of all...

We must do our job!

What's our job?

Write (clean) code!

Write tests!

Tests are still code...

What's our job?

Write (clean) code!

Keep learning!

What's our job?

Write (clean) code!

Keep learning!

Communicate!

“Clean code is simple and direct. Clean code reads like well-written prose. Clean code never obscures the designer's intent but rather is full of crisp abstractions and straightforward lines of control.” - Grady Booch

Our job is not only about coding!

We have to communicate!

Listen

Document

Educate

Explain

Alert

Say no!

What can happen if you say **NO**?

Your objection might be heard

You might get convinced

It might get rejected

They might not care

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

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



Why Clean Code Is Not The Norm?



Sandor Dargo



TWITTER [SandorDargo](#)

EMAIL sandor.dargo@gmail.com

Dev<Talks/>
Cluj-Napoca

Event
powered by

Google