2025-03-15

All Code Sucks

Joe J Collins
j.collins@zengenti.com
linkedin.com/in/joejcollins

Zengenti

21 March 2024

Thanks for taking the time to come out tonight.
I'm Big Joe from Zengenti.
We are a small company in Shropshire. About 70 nerds.
We do websites for universities and local authorities.
I actually don't do any websites, I work in the hosting team.
We maintain a private cloud to run the websites.
We use a combination of Ansible and Python
maintaining about 3000 servers.
But, tonight Matthew,
I am going to talk about, **why all code sucks**.

We all know good code,
or at least we think we do.
But I should probably define what I mean by sucky code.

"Programs are meant to be read by humans and only incidentally for computers to execute."
— Donald Knuth

"It is better to have clean code that doesn't work than crap code that does."
— Robert C. Martin

What is sucky code?

My short answer is . . . **sucky code is hard to read**.
No need take my word for it.
Donald Knuth, the Yoda of Computer Science
says that code is for humans to read
and sometimes for computers to run.
He is all about the readability.

Uncle Bob Martin is more emphatic.
He uses the term clean code.
as a proxy for readability.
He's says that readability is more important than working code.
If you can understand it, then you can fix it,
but if you can't understand it and it breaks, you can't fix it.

All Code Sucks

└The Great Hunt for Non Sucky Code

For about about 25 years now,
I have been looking for code that doesn't suck.
And trying to produce code that doesn't suck.
I've worked in companies large and small.
Worked with scumbags and saints.
But pretty much, all the code sucked.

Maybe I just got unlucky.
But I think I am seeing a pattern here.
Maybe I should stop looking for the perfect code.
Maybe I should admit that all code sucks.
And I should work out what to do about it.

This does beg the question, why does it all suck?

All Code Sucks

- Half of everything is below average

└Why Code Sucks

On the whole I think the odds are against us.
When we talk about code readability
we are talking about a bell curve.
Straight out of the gate,
half of everything is going to be below average.
Well, below the median.
You don’t have to be Francis Galton
or any famous statistician
to realize that.

- Half of everything is below average
- Sturgeon's Law

Why Code Sucks

Then there is Sturgeon's Law.
Sturgeon was explaining why most science fiction is low quality.
And came up with the pithy answer
"90% of everything is shit".
The observation works here too.
Only some things are really good.
Or put another way most of anything is bad.

- Half of everything is below average
- Sturgeon's Law
- The 3 Year Old Programmer

Why Code Sucks

Then there's an issue peculiar to the programming business.
The demand for programmers for the last 25 years
has always outstripped supply.
When I started out
the average programmers experience was 3 years.
And that hasn't changed.
As more and more people have entered the business
they have kept the average experience down.
There are a few old hands around but
as a group we still don't have that much experience.

So if the odds are against us
maybe the organizations we work for will help.
Or perhaps not.

Why Code Sucks

The romantic image of a software startup is
a couple of guys in a garage.
I have actually see this quite a bit.
For most start ups the two guys are the dad and the son.
The dad is the salesman.
And the son is the programmer,
who's typically been excluded from school for some reason.
Writing readable code isn't really on their agenda.
In fact reading plain English is not on their agenda.
The third employee?
The son's best mate from school who was also excluded.

Why Code Sucks

The other kind of startup I have seen
occurs in big companies.
The summer student project.
Alternatively called the unsupervised use of new technology.
All the experienced programmers are on holiday or busy.
So they give the new technology to the summer students,
who give it a go.
And if it runs they put it into production.

- Software Startups
- Summer Student Projects
- Prototypes in Production

Why Code Sucks

This last point is also a general point.
Any software that appears to work goes into production.
Not because anyone thinks it's a good idea
but because there is a commercial imperative.
Having learnt from the prototype,
the plan was to throw it away and build it for real.
But that never happens.
It is always put into production.
And lives forever.

All Code Sucks

Why Code Sucks

I think I am an intelligent person.
I think I can understand most things.
But my understanding is only a deep as it actually is.
I am completely unaware of my ignorance.
Because I am ignorant of it.
I believe I understand things more than I do.
It is unavoidable.
But writing code is about explaining things in detail.
And it does nothing but expose how little I know.

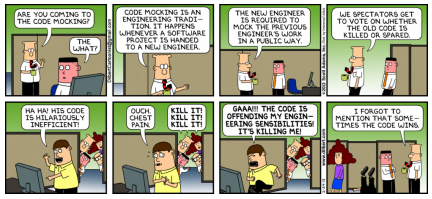
All Code Sucks

- The Illusion of Explanatory Depth
- Availability Bias

Why Code Sucks

There is a chance that I have seen non sucky code.
But I can't really remember it.
A couple of years ago we used a static analysis tool on our code.
To my surprise is said most of the code was good.
But it pointed to the five worst files.
Those were the files I spend most of my time working on.
If by some miracle the code is good,
you will only ever work on the bad bits.
Those will be the bits you remember.
So you think all code sucks.

So what we to do about it?



What Not To Do

Ritual code mocking is good fun
and a great way to let off steam
but it is actively harmful to team performance.
/dots but that is a whole other talk for another day.
We have to remember that
everyone that worked on the code
did their very best
with the time, resources and knowledge they had.
If we can’t read the code
we have to find a way to work with it.

So I am going to show you just a few techniques
we use at Zengenti to work with sucky code.

All Code Sucks

Improving Readability

Improving Readability

create a fast api endpoint that is called the answer that takes a number and divides it by 2 and then works out what two numbers to add together to make that number, which it returns in a json package that includes the keys first number and other number it should also write the number out to a file named after . . .

⇒

Answer API
An API endpoint that takes a number, and returns the two numbers to add together
to make the original number.
Parameters:

- `number` (int)
- `first` (int)
- `other` (int)

. . .

In case you thought this was an entirely technical problem.
It isn’t.

- Log behavior
- Put the code in a harness
- Get ahead of the offending code with a feature flag
- Side by side rewrite

What To Do

I should say that this isn't my idea.
This is based on the work of Michael Feathers.
in his book **Working Effectively with Legacy Code**.
It is the definitive guide to working with sucky code.
But it is over 20 years old now.
And unfortunately the it has not been updated.
But it is still the best guide I have found.

Also I have skipped the step
getting control of the development environment.
At Zengenti we use ephemeral environments
either Docker Containers or Virtual Machines
to keep everything clean.
But that is a whole other talk again.

"It's not that hard..."
— Billy Beane

"It's incredibly hard"
— Ron Washington

The Challenge

To quote Moneyball,
the challenge is both easy and difficult.
The example was one function and one file.
But the code we are working with has 3500 files.
Some is hard to read,
some has been improved and some it half way between.
But it often feels like we have three different code bases.
All jumbled up.

It turns out the improving code is easy
but managing the process is hard.
Knowing which bits to improve really hard.
I have no answer to that, I am kind of hoping you do.

Thank You!

Joe J Collins
Zengenti

Thank you for listening.
I’m Big Joe from Zengenti.
If you thought this was interesting,
and would like to work with us,
please get in touch.
If you think you have a better way to work with sucky code,
I’m all ears.
Or if you have an idea how to manage the process,
Let’s talk.