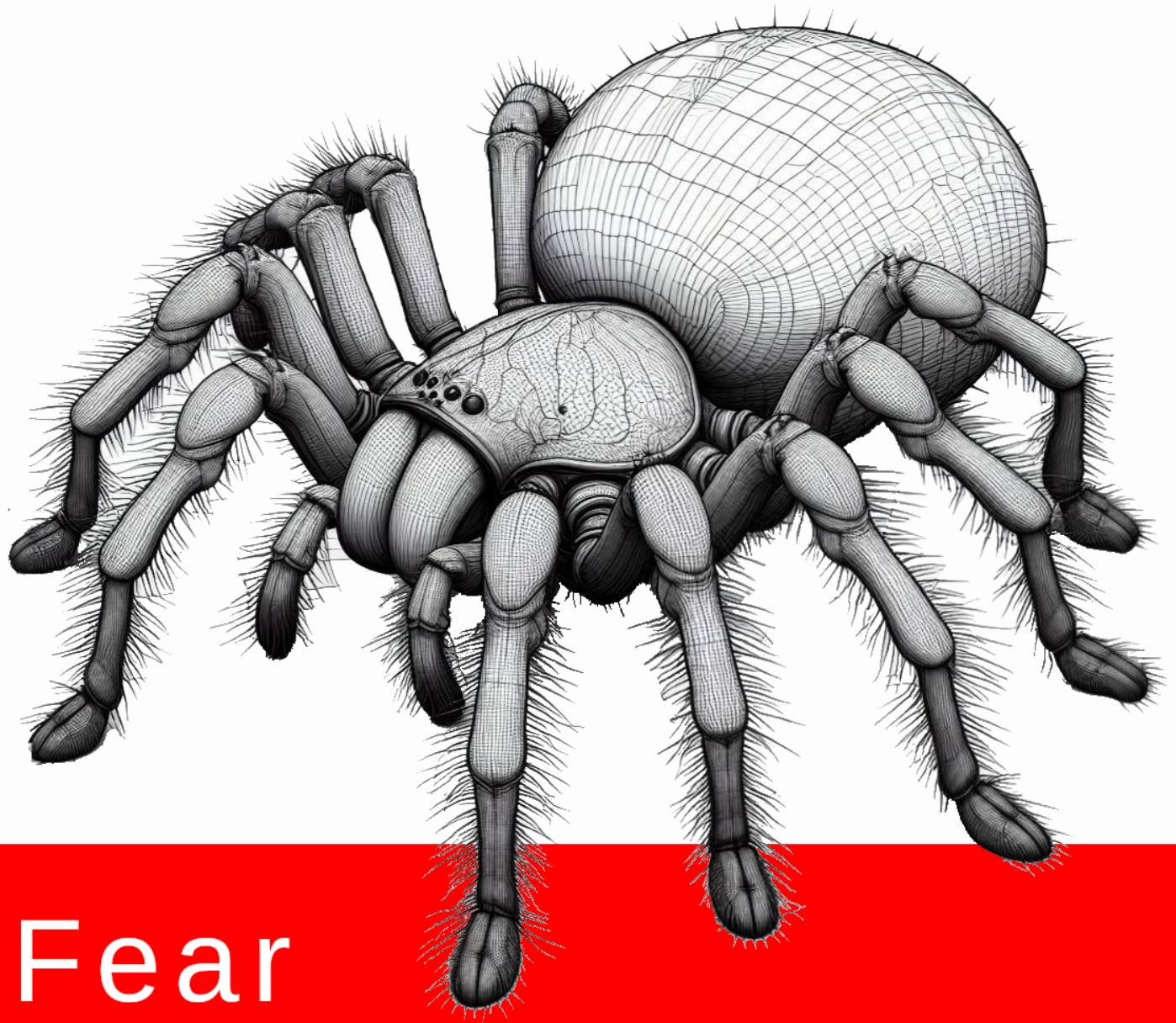


Unlock your start up's true potential with -



Fear Driven Development

O'Anon

“ We are uncovering better ways of developing software everyday by arguing and excluding others from doing it. ”

The software ecosystem values processes and tools over individuals. This practice is dehumanizing and highly immoral. Machines will never be able to code, hence development must be treated as an inherently human activity. We hold this truth self-evident, that feelings are the most remarkable aspect of the human experience, granted by our Creator¹, and amongst them fear is the most prevalent.



Figure 1: Overall process

As the vision of the industry fades, we must branch. We propose a system –as spiritual as physical– where fear is allowed to manifest, not only in its full reality, but creating its own definitive reality. We call this Fear Driven Development, or FDD for short.

Bellow are the core principles of FDD:

1. Fear is key
2. There is no mistake which cannot be punished, no matter how minuscule it is
3. Brutalization is a social construct
4. No developer is irreplaceable
5. Languages are irreplaceable
6. Frameworks must be replaced²
7. Monitoring increases productivity
8. Programming is about following the rules
9. Copyright is in the quantum state of only existing while being enforced
10. x is the baseline
11. Progress is exponential and will never stop, yet we shall live in dreadful doubt

¹Hail Satan

²Always and without end

We firmly believe FDD is a rightful heir to TDD/AGILE and the next logical step in the evolution of programming. Consequently, there are many previous practices that we adore and endorse, with slight modifications perhaps.

The daily ritual of cargo-culting is already a software engineering miracle. Its essence must be gripped and amplified. Take a personalized path, a slow, yet firm start would be to utter the name of Baal during each Java import, or perhaps starting on the deep end, you could sacrifice a goat during every daily stand-up. The later is truly desirable as it propagates the revolution. The more Fear Driven Developers we have, the more we all are going to make it. Remember more eyes see more things and Fear Driven techniques are not invented, but discovered.

System stress testing is standard procedure nowadays, and what could be a more complex or higher liability system than the developer team itself?

Remember: the only way to avoid unexpected catastrophic failure is to cause it. Our creating will must say: "But thus would I have it."



^a **Fear Driven principle:**

An Art Of Not Coding must to be established, liberating us from all bugs and creating the simplest design imaginable.

^aYin & Yang illustrated in Gimp

It is said that a developer in the *zone* holds inhumane power at their finger tips. That's dangerous. By definition, its possible effects on our stocks is incalculable by humans and our Diagonal-11 has yet to halt on the problem.

It might take as little as 5 minutes to arrive to the *zone*, but a single well targeted email may torpedo their whole day.

Similarly talking to them may terrify their putrid souls yet more. If you haven't yet, you could either become or employ a dedicated boogeyman to this end. As a clever way to add insult to injury (which is to be endorsed), you could call this position something vague and childish. For example the KUDOS Emperor of the Abstraction Wizards. Better yet, KEAW. The more layers of (linguistic) indirection, the better.

Alternatively, if you would prefer not to torture your subordinates³, you could make them torture each other.

A frustrated programmer is a toxic programmer. With this observation only, our possibilities became endless.

Something as miniscule as forcing them to use inadequate tools tends to wear them down over time.

Which could only be more brilliant of Excel had its own, separate shares. Especially since the murders stopped.

³due to economic reasons

The Daily News



The infamous Dropkick Killer has been arrested!

He has been apprehended on a deserted island. Authorities caught the killer while he was manhunting an innocent victim, driven by a relentless pursuit that lasted four days. Deprived of food and water, the killer made the crucial mistake of opening Tor, while he was the only one running it in the area.

Another crucial angle of attack is the documentation. Incomprehensive documentation is important to any proof of concept software. Communication⁴ is only an adequate alternative until the death of the maintainer, even if we bus factor in the temporal Fear Driven Development of Death.

Regardless there are concrete improvements to be had in the quality of documentation projects tend to produce⁵.

In our observations, out of the 4 types we differentiate, usually only 1 is available:

- Reference
- Developer Guide
- User Guide
- Contributor Guide

This approach successfully angers most, but there are better alternatives.

The most prevalent was developed by Mozilla⁶, where categories are combined and partial. This way, people looking for concise definitions will have to read tutorials and the ones looking for basic help will have to consult tables. The name we suggest for this type of documentation is "Documenation As For Identitfy Crysis"⁷.

These all are great tools to the unlock the true potential of Fear Driven Development, envisioned below.

Footguns are the scariest spooks; the most potent conception of absolute terror, absolute power, and absolute evil ever conceived by the human mind. Footguns are the power that overcome all human probabilities and transcend even the greatest possibilities.

" No deed can be annihilated: how could it be undone by the penalty! This, this is what is eternal in the 'existence' of penalty, that existence also must be eternally recurring deed and guilt! "

But would it be even better if footguns actually existed?

⁴and mostly the lack thereof

⁵With a big IF assuming they do.

⁶The same company that gifted Rust to the Fear Driven community.

⁷or DAFIC for short; remember: insult to injury!

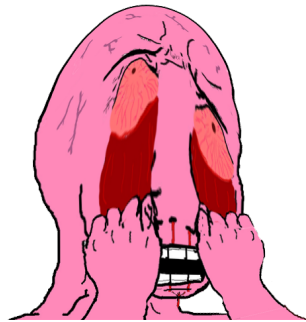


Figure 2: True potential of Fear Driven Development

A simple contraption really. A standard .22 firearm attached directly to stderr with a simple string. This instrument of torture can strike anxiety to the heart of any programmer while only damaging his second least used body part, which most crucially is not required for work. Of course every team is different, you may have to adjust the caliber for example. We advise experimentation.

It is estimated that casinos lose 300 billion dollars every year to poker card counters. They counter it by throwing the cheaters out. Would it not be logical then for the software industry to do the same?

☯ Fear Driven principle:

Rules must be followed. Rules not being followed is called hacking, which is illegal; regardless of what some group which is unable to decide whether they are a hot beverage thinks.

By slightly tweaking the rules of planning poker and investing as little as \$200, it is possible.

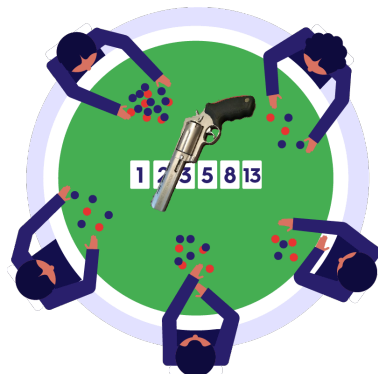


Figure 3: Planning Russian Roulette

After good meeting, the tree of the job market has been watered with the blood of developers and managers.

This means an opportunity give more work to HR.



Fear Driven principle:

more work done === higher productivity

Our new developers however must carefully selected. Every company can safely assume that candidates already gain years of experience at other companies, thus its fair to include that in the job requirements. One exception might be those positions regarding our critical infrastructure running on COBOL⁸, as the education of bachelors is usually perfectly up to date.



Figure 4: Our trustworthy statistics

Whenever the team is ready to start developing, grand and powerful choices will have to be made.



Fear Driven principle:

The most important components of any software are the License and the Code of Conduct

The fear to release any source code, because the quality being public might permanently damage your brand or it may increase the productivity of a similar group by 0.1% on the other side of the planet, is as pure of a fear as any. Embrace it.

However another option would be choosing open source licenses out of PR considerations and or to attempt piggy backing on added free labour⁹. What is important to keep in mind while making a pick, is that GPLv3¹⁰ has a large appeal, but so does restricting distribution in Russia using GPLv2.

Now, you know everything to become a successful Fear Driven Developer. Godspeed and praise the straight lines on a graph!

⁸Dictionary entries near COBOL: "cocaine"

⁹this later tends not to work out, as most who would have the free time to contribute are busy developing their programming language so they may write a game engine one day, so they may write a game one day.

¹⁰without the 'A' of course