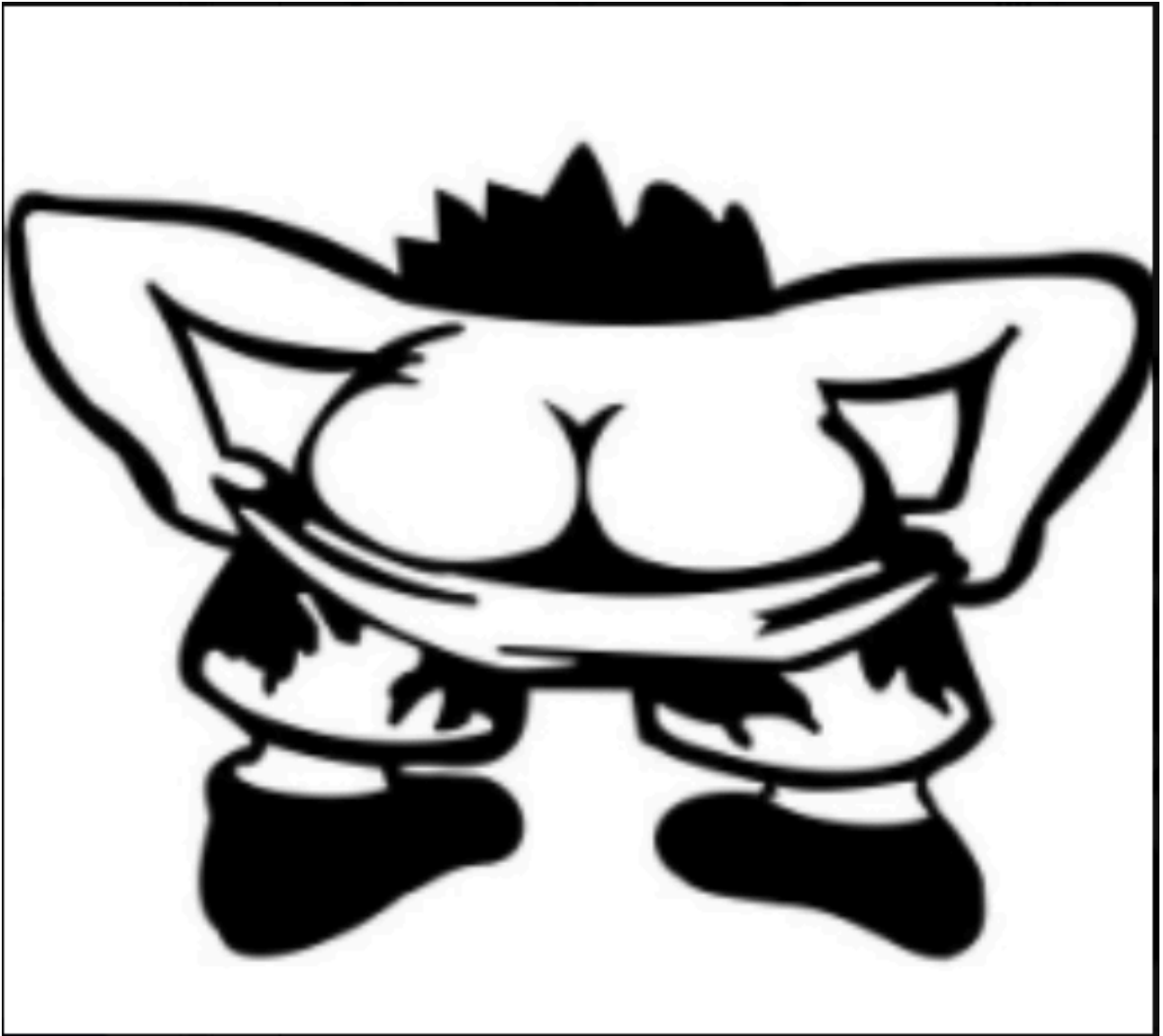


devbastic

--==== sumtin'... ====- --
==--> *new*



overseeing skills

As a programmer, I've programmed pretty much eve... well, not everything but this ranges from BASIC on a zx-80, assembly on a z80 processor in the zx spectrum (with a whopping 48k RAM) to COBOL on an IBM System/390 (summing like that anyway); it was 1989, doing an internship at a large Dutch insurance company) to Object Pascal and c/c++ and MPW Shell w/ Sourcesafe on the first macs to arrive

in Holland to, well, most “open-source”-typed languages such as bash, perl, php JavaScript and python. Some enterprise-y java as well. In a non-enterprise environment though.

As an architect, I can design, steer, educate your team and oversee all development in most of the aforementioned languages. I know what works, I know what doesn't. As a datamodel purist (a correct datamodel goes a very long way in establishing correctly running software), TODO

As an example, I've been looking at php's fairly recent additions on type-safety measures, and they're fine. They help the programmer to avoid mistakes. But I really think: you shouldn't make those mistakes in the first place by *really* understanding what you're doing.

Of course, that sounds somewhat arrogant but hey; it is what it is. I never use sudo, log-in as root on any server (or 'cloud'; someone else's server, if you even can get in with ssh), for ages, and have never trashed anything inadvertently. Just don't make mistakes. Easy as.

So, yeah, I can pretty much oversee any programming venture, as far as languages and tooling is concerned.

[
todo

cloud:

(show guts of sensors/gps/srv; the actual logic that matters)

then zoom omhoog, show cloud, 3x active, nats etc. etc.

show some operational dashboard where it runs on 40k servers or summing like that

ai:

]

Integrity

todo

Honesty

I'll tell you when it sucks. Right in your face. If i really, really mean it, and you insist otherwise, I'm gone. Easy as.

I'll tell you when it is good enough for now. We're on a deadline together: your success is my success. If you fail, I fail. And-uh, I kinda hate failing.

I'll know ánd remember where it isn't really good. I'll tell ya where something is a bit fishy. But I'm already, in the background, working on where it needs to be improved. And we'll fix that when there's resources. Or we'll let it run; if it doesn't really break, and quickly move on to actual new functionality that our investors need to see. Yesterday.

I WILL NOT EVER FOOL YOU. By giving you assurance whilst deep down inside I know I'm fucking up. I will not do that. Ever.

I will also tell you, with profound underpinnings, why some team member doesn't really 'work'. And maybe himself (eh, and/or herself) needs some work. Or maybe... even better go cater good lunches instead of coding.

Startup architect

Having worked with and for Internet startups since the mid 90's, I know what works there as well. It's probably not a word in IT terms, startup architect, but it should. Developing software in a startup has, aside general development issues, it's own specific set of problems to deal with. I've been through them multiple times since the 90'ies. Small ones that stayed small and disappeared. Some on the payroll, some as a co-owner, 2 very large ones; going from loonslaaf #4 to 130 people in 4 countries....

"MVP", we didn't even call it that in the beginning.... it literally was the only thin we coulfe come up with ~~in the amount of time~~ with the sack (sometimes a sacklet) of money the "investor" gave us.

That has also a very specific proble. You and your teammates can argue for weeks about the Next Best Datamodel, a datamodel that can (voldoen) aan any datamodel... If you do that and no one has the balls to decide AND MOVE ON, you're out of a job in exactly... 1.72 months from now.

I have those balls.

It is like sailing (as hoog aan de wind as u can get without causing serious breakage but GET THE HELL OUT of the way of those 40knots (squalls). Go fast, on the line. A bit over the line, juuuuuust a little bit further, ok, team, we made it. Another sack of money, for the next 2 months.

You're balls are on the line, sailor. Aye aye, cap.

Throw 'm on the roof, sailor. Aye aye, cap. (*)

<https://pppphpileblog.wordpress.com/2024/06/01/stick-it-on-the-roof/>

Travels (sailing...)

[
pre item de main tools, 'on hover' ofzo. lets fancies.
]
'I don't do MS Word; link to resumé website.
]

Projects:

- 1992/1996: photodesk
- 1993/1996: framewire
- 1996: luna pt -> distri gebeuren
- 1995/1997: butler/texis, client-server DAL/DAM -> webCGI, C, C++, make, ld, Book Stevens Networking, book UNIX (blauw/witte unbix bijbel, zoeken), minix, linux rh 4.2 (?)
- 1998: yn, syndiction engine, queue-driven, hand-coded, reverse queries
 - 2000/2001: 20M dot bomb
- 2002/2005 & 2006/2007: yarosa -> global sms 2 tv
- 2007: my own company
- 2007/2008: sms broadcasting engine, 3.5k/sed, fully redundant;
 -
- '13 HDN
- '14 Exact
- '17 crypto
- '18 blokchain/smart contracts
- '19 kafka/de integratie specialisten
- '21 e-com
- '22 yacht brokerage -> corona
- '23 yacht brokerage -> myelopathie