CUPID

for joyful coding

PATRICK PFEIFER

RECYCLING
A BLOG POST

BY
DAN NORTH

Credit

“Daniel is the originator of Behaviour-Driven Development, an agile approach to software development that encourages teams to deliver the software that matters by emphasising the interactions between stakeholders.”

https://dannorth.net/

https://dannorth.net/2022/02/10/cupid-for-joyful-coding/

CUPID

- **C** omposable
- **U** NIX philosophy
- **P** redictable
- **I** diomatic
- **D** omain-based

Motivation

- Properties over principles !
- Properties of properties ?
  - practical
  - human
  - layered

Composable

- **Small surface area**

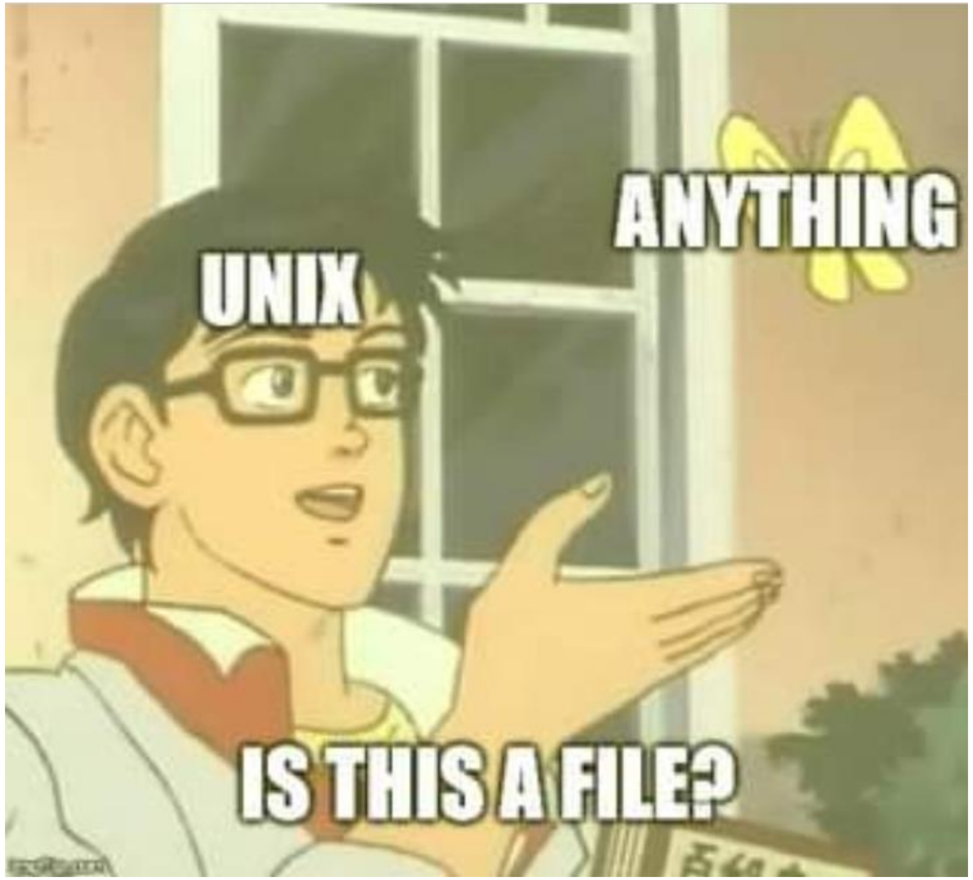
- **Intention-revealing**

- **Minimal dependencies**

Your next task is to figure out which applications in your org use log4j

UNIX-philosophy

- A simple, consistent model
- Single purpose vs. single responsibility



P redictable

- Behaves as expected
- Deterministic
- Observable

Idiomatic

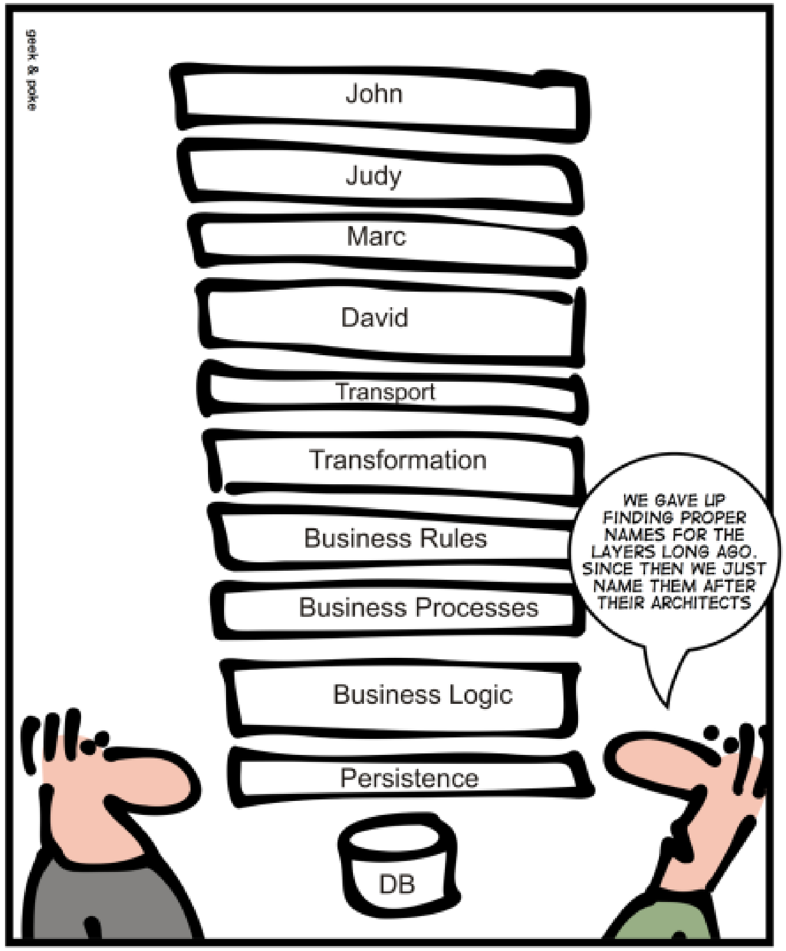
- **Language idioms**

- **Local idioms**

```
$ python

>>> import this

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Domain-based

- ~ language
- ~ structure
- ~ boundaries



A GOOD ARCHITECT LEAVES A FOOTPRINT

https://blog.jooq.org/why-you-should-not-implement-layered-architecture/

Credit

"Daniel is the originator of Behaviour-Driven Development, an agile approach to software development that encourages teams to deliver the software that matters by emphasising the interactions between stakeholders."

https://dannorth.net/

https://dannorth.net/2022/02/10/cupid-for-joyful-coding/

Thank you

Patrick Pfeifer

patrick@patrickpfeifer.net

- PFiver@mastodon.social
- Twitter, GitHub, Linkedin: PFiver