Best Simple System for Now

Daniel Terhorst-North


富嶽三十六景
神奈川沖
浪裏



Best Simple System for Now

Defining the Best Simple System for Now

Why you should not do this

Why no one else is doing this

‘And then what?’

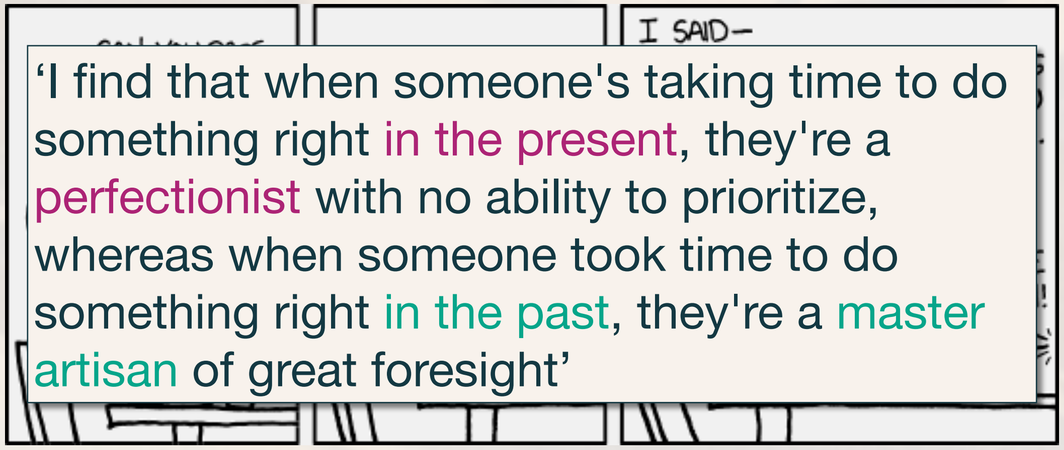
Best Simple System for Now

Whenever you anticipate the future, you will be nearly right!

- That nearly will bite you every time

Avoid speculative generality

- You don’t need a rules engine



https://xkcd.com/974

```
> python -m this

The Zen of Python, by Tim Peters
...

In the face of ambiguity, refuse the temptation
to guess.
...
```

‘If you don’t end up regretting your early technology decisions, you probably over-engineered.’

— Randy Shoup

goalwards

Best Simple System for Now

Look for the essence of the problem

- not what is there but what *should be* there

Simple is a function of Now 🤯

- context changes everything

‘A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.’

— John Gall

Embrace kaikaku as well as kaizen

- Can you ‘hire’ a different algorithm?

‘Make the change easy’ by making *any* change easy!

We should forget about small efficiencies, say about 97% of the time.

Premature optimization is the root of all evil.

Yet we should not pass up our opportunities in that critical 3%.

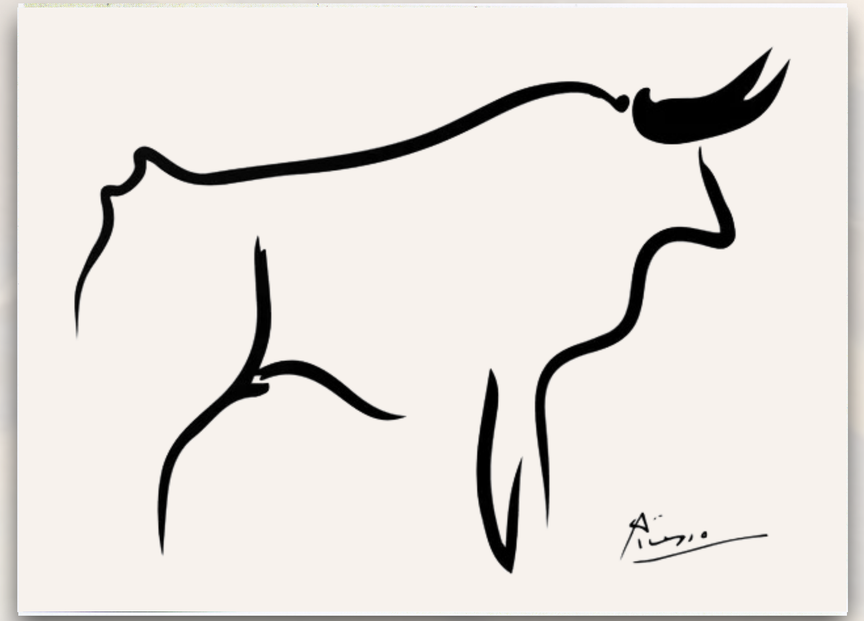
— Donald Knuth

Best Simple System for Now

Engineering is a skill

Sketching is a different skill

- Hacked code quickly deteriorates
- Sketched code is just ‘less’



Be explicit in your values and deliberate in your trade-offs

CUPID

- Composable
- Unix philosophy
- Predictable
- Idiomatic
- Domain-based

This also applies to your system of work!

VESPA



Visualize

Eliminate

Simplify

Practise

Automate

However far along this path you get, you did good!

goalwards

Do not try this at home

(or: How to sell BSSN to your boss)

‘It is overkill for a prototype’

‘Why invest in this code if we are just going to throw it away?’

Most [*citation needed*] successful prototypes get released, not rewritten

There is no ‘natural force’ to improve things

- ‘Just one more case’
- ‘Let’s just add it to the build’

Good prototypes can become great products

- WhatsApp had 500m users on an Erlang codebase with ~15 engineers
- SQLite is the most deployed app in the world: 3 contributors, 600x test code, millions of tests
- JUnit started on a flight to a conference… to talk about JUnit!

‘You can’t release half-baked products’

Meeting part of a customer’s need sooner is a win-win

- Any feature set is a power curve, so you can unlock value sooner

An opinionated feature set is a differentiator

- Google Docs vs. Microsoft Office
- iPhone vs. everything else in 2007!

Frequent product increments require appropriate code* quality

- both under- and over-engineering will limit your ability to iterate quickly!
- maintaining a BSSN strikes the ideal balance

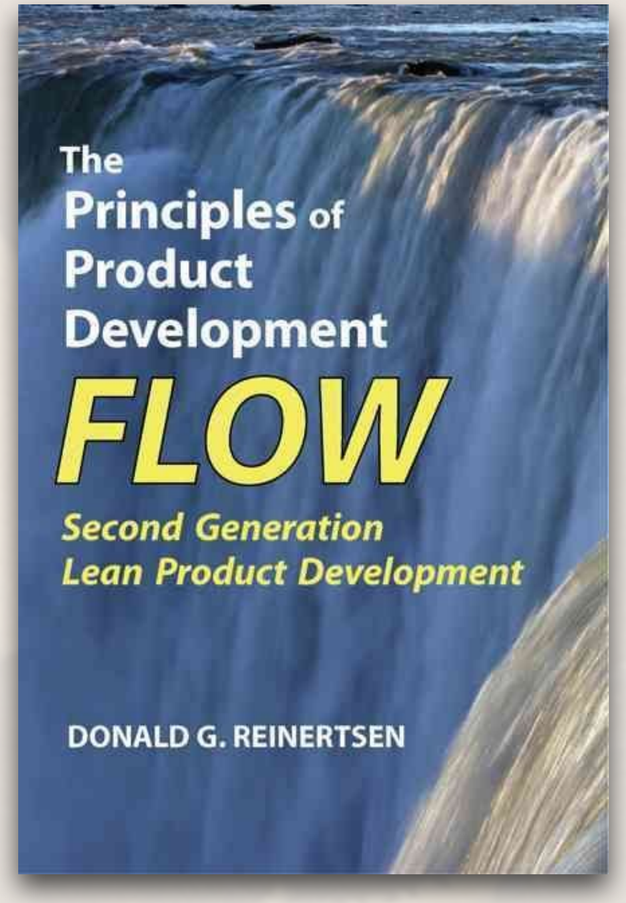
‘It is inefficient’

‘Refactoring is rework; rework is waste!’

- only when you are building the same thing again and again
- otherwise it is adapting to feedback

What matters is when you make money

- Monitor Value At Risk (VAR)
- Maximize Risk-Adjusted Return on Capital (RARoC)
- Measure Cost of Delay
- Identify Opportunity Cost



So why are we not doing this?

Discipline

Habits

Courage

Humility

Discipline: keeping the kitchen tidy

‘Commit yourself to constant improvement. Commit yourself to quality.’

— H. Jackson Brown, Jr.

Discipline is a continual series of choices

Discipline enables sustainable speed

Discipline requires the whole team

Discipline involves deliberate practice



Habits: your default mode

‘I am not a great programmer; I am a good programmer with great habits.’

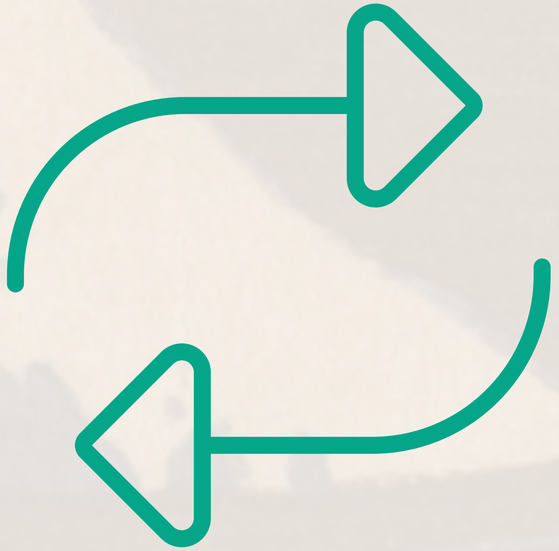
— Kent Beck

There is no arcane knowledge in BSSN

Habits are a choice and then a process

Seeing the essence takes practice

Seeing the trade-offs takes practice



Courage: taking the first step

‘Courage is resistance to fear, mastery of fear—not absence of fear.’

— Mark Twain

How could this possibly work?

Why is everyone else not doing it?

Will I look like an idiot?

‘Trust me once’



Humility: the better version of you

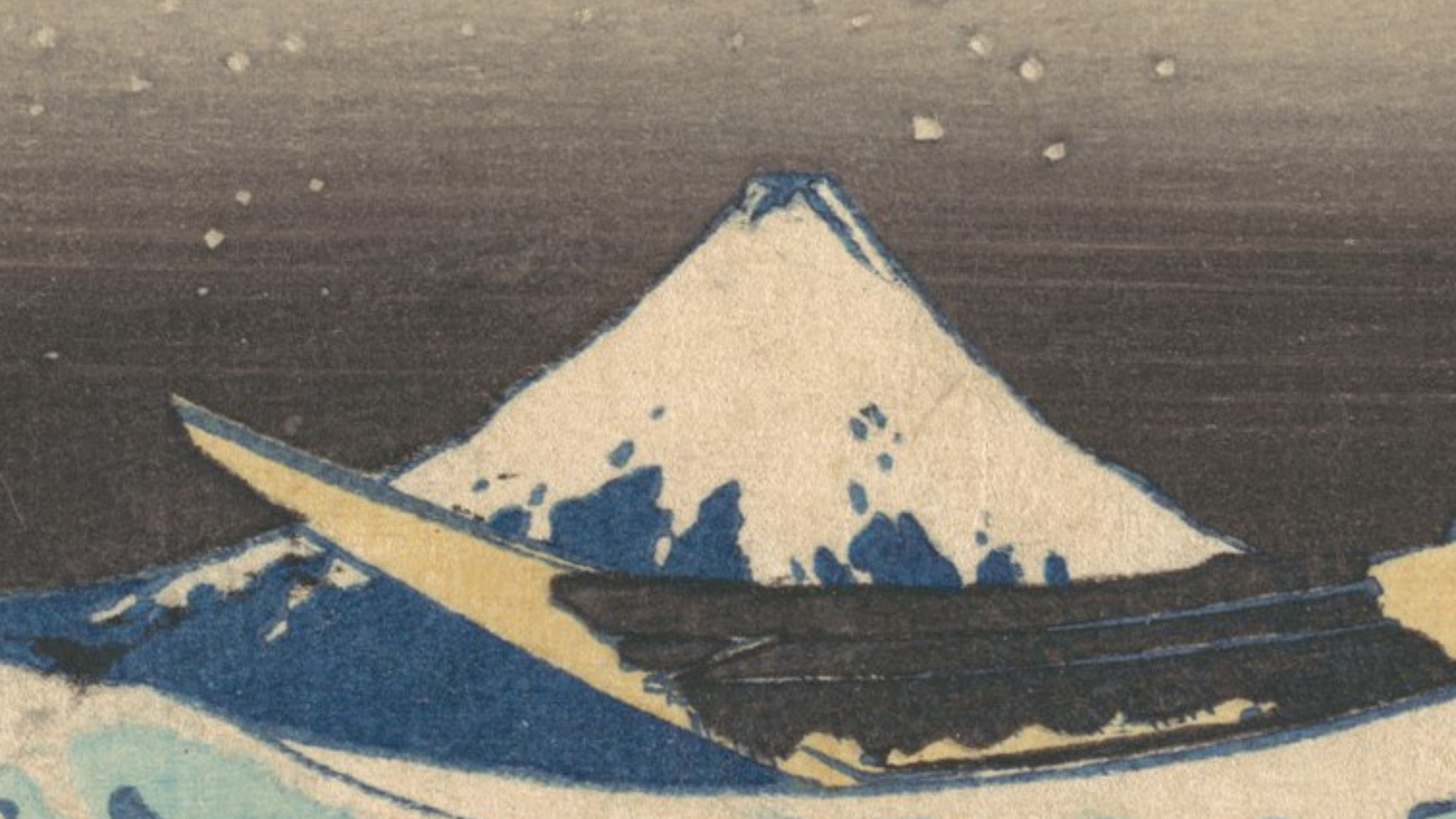
‘Be formless, shapeless, like water…
Now water can flow or it can crash.
Be water, my friend.’

— Bruce Lee



‘Humility is not thinking less of yourself,
it's thinking of yourself less.’

— C. S. Lewis




富嶽三十六景
神奈川沖
浪裏

‘And then what?’

Just Start™

- and expect to be rubbish!
- ‘Code like no one is watching’

Assume Hofstadter’s Law

- and Dunning-Kruger
- which is also subject to Dunning-Kruger

And let me* know how you get on!

- or hire me to help 😃

https://goalwards.co

daniel@goalwards.co

linkedin.com/in/tastapod

@tastapod@mas.to

@tastapod.com