SHINY OBJECTS ARE COOL

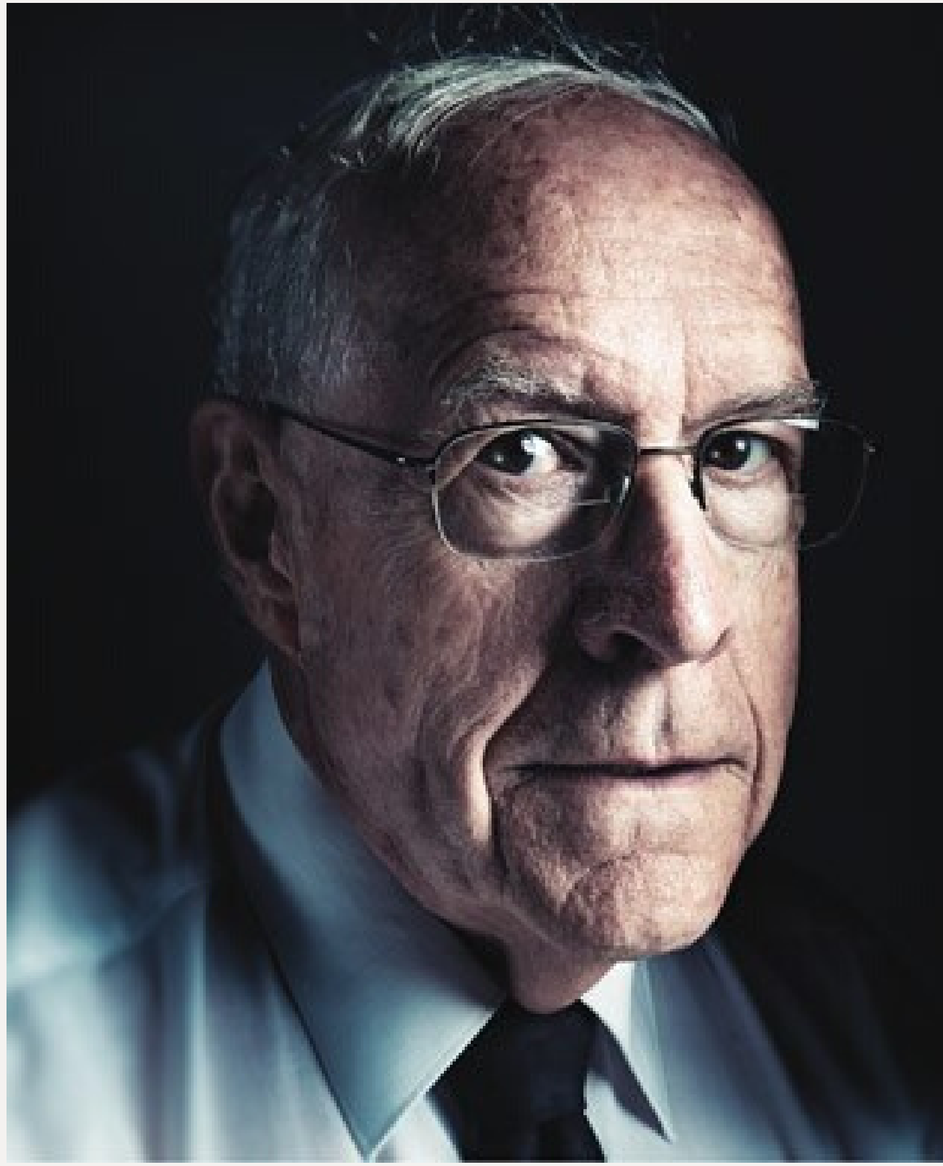
BUT SO IS BUILDING PRODUCTS PEOPLE USE

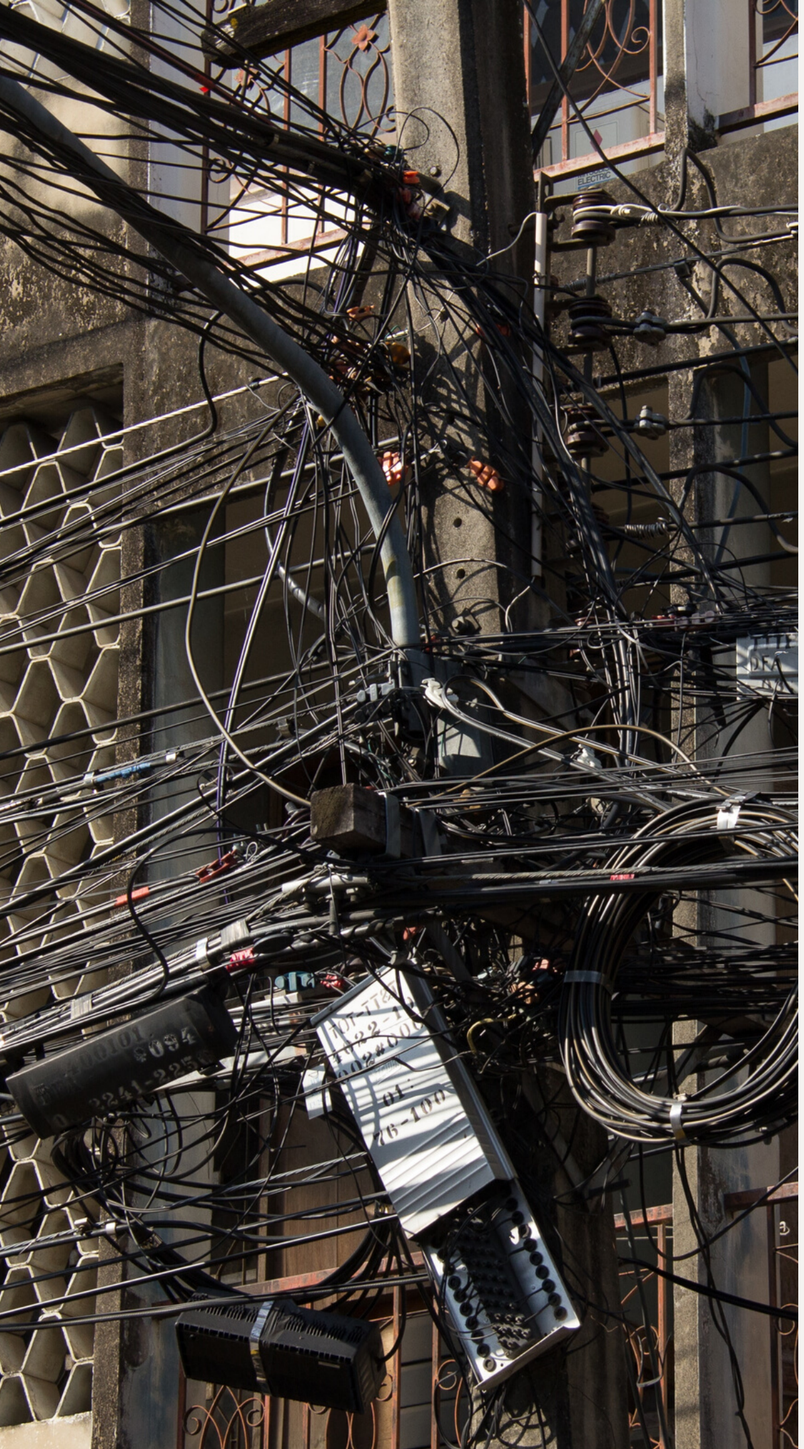
@jennapederson | #creamcitycode | #fallx

What is shiny?

Accidental vs. Essential Complexities

FRED BROOKS, 1986





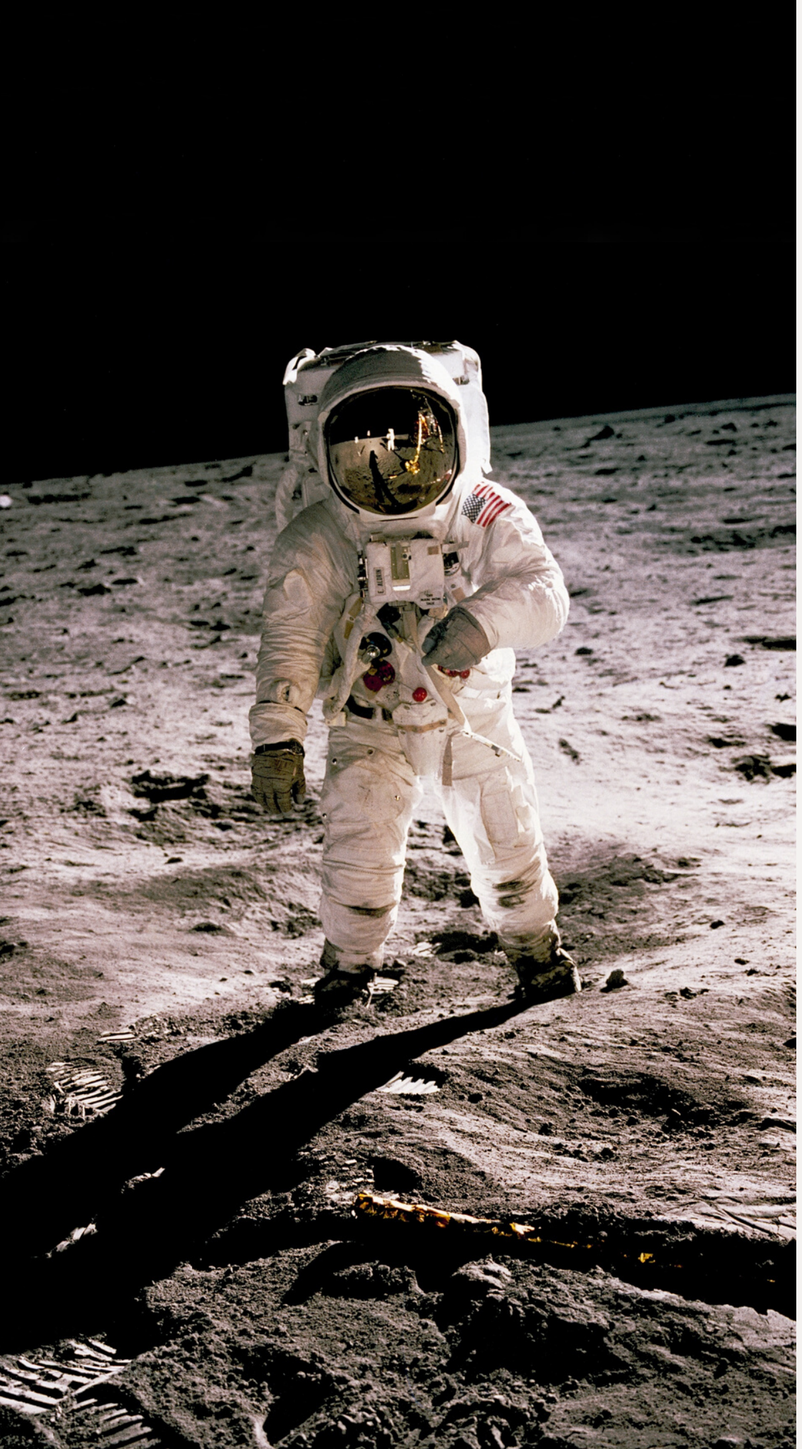
Accidental Complexities

NOT ESSENTIAL TO THE PROBLEM BEING SOLVED

PROBLEMS ENGINEERS CREATE AND CAN FIX

EXAMPLES:

languages, tools, processes, techniques, bells and whistles



Essential Complexities

DIRECTLY RELATED TO THE PROBLEM BEING SOLVED

INDEPENDENT OF HOW MUCH EXPERIENCE YOU HAVE OR THE TOOLS YOU HAVE IN YOUR TOOLBOX

EXAMPLES:

deciding what to build, the humans, design and testing, the complexity of hardware/software

HELLO WORLD EXAMPLE

RUBY

```
$ ruby -e "puts 'Hello, World!'"
```

HELLO WORLD EXAMPLE

RUBY

```
$ ruby -e "puts 'Hello, World!'"
```

PYTHON

```
$ python -c "print('Hello, World!')"
```

HELLO WORLD EXAMPLE

RUBY

```
$ ruby -e "puts 'Hello, World!'"
```

PYTHON

```
$ python -c "print('Hello, World!')"
```

REACT NATIVE

```
$ react-native init HelloWorld
$ cd HelloWorld
```

Open up App.js and type out "Hello, World!"

```
$ react-native run-ios
```

Wait. Cross fingers.

What's the problem with shiny?

DOESN'T SOLVE THE PROBLEM

you know, the one our customer has

HAS LESS IMPACT

a focus only on the shiny is not nearly as impactful to the problem as focusing on the non-shiny

ALWAYS STARTING

but never finishing

AND MORE...

unknown, unsupported, unmaintained, risky, distracting

Why do we choose shiny?

STARTING VS. FINISHING

starting is orders of magnitude easier than finishing

NEW IS BETTER

obviously, if it's new and improved it's better, right?
maybe. maybe not.

WE LIKE TO LEARN

and maybe it's just for our resume or to scratch an itch

How do we build products people use?

FOCUS ON THE PROBLEM (NOT THE SOLUTION)

not just the problems we think our customers have or the ones we've manufactured for ourselves - the ones they actually have

FOCUS ON THE CUSTOMER

get out of the office and talk to them

FOCUS ON THE VALUE

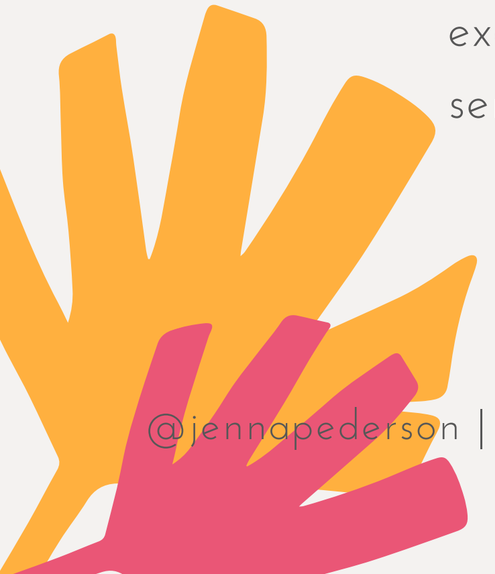
add value to someone's life

Code vs No Code

WHY WRITE CODE WHEN YOU DON'T NEED TO?

WHAT IF YOU DON'T FULLY KNOW WHAT THE PROBLEM IS AND THEN WRITE A BUNCH OF CODE?

example 1: a React app vs. a Squarespace site
example 2: a React app vs. a Google form + Zapier to send an email or a text



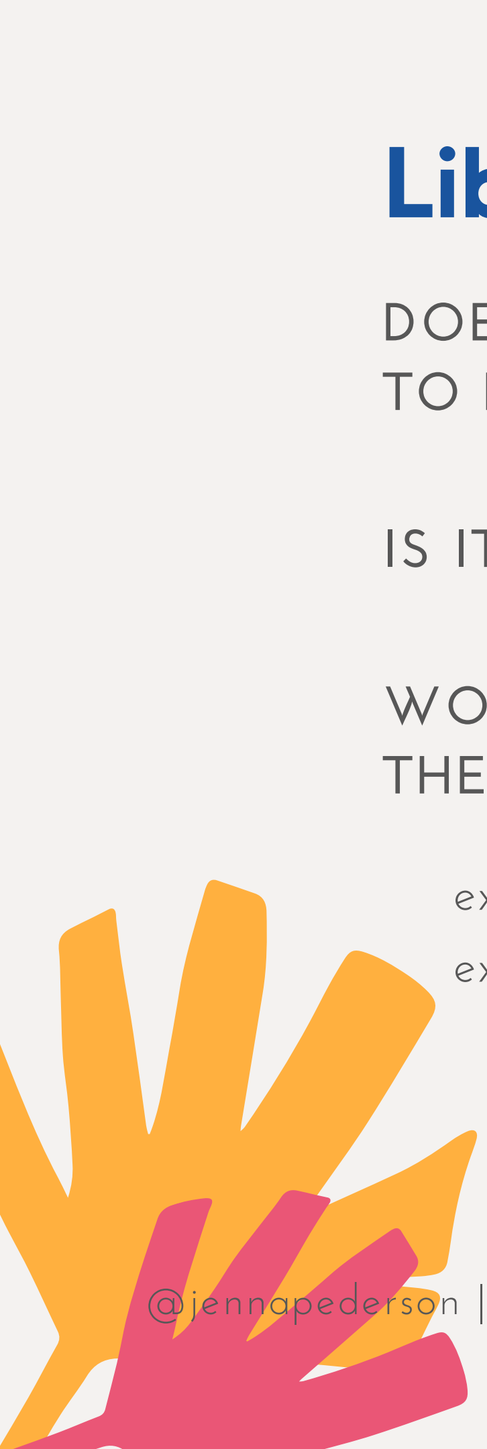
Library vs Roll Your Own

DOES THE LIBRARY DO WHAT YOU NEED TO DO AND REDUCE COMPLEXITY?

IS IT ACTIVELY MAINTAINED?

WOULD YOU BE THE LARGEST USER OF THE LIBRARY?

example: the Recaptcha gem vs roll your own
example: Circle CI + ClojureScript frontend



IaaS vs. PaaS

DO YOU NEED TO MANAGE INFRASTRUCTURE OR COULD SOMEONE DO THAT FOR YOU?

DO YOU HAVE TIME TO LEARN HOW TO MANAGE INFRASTRUCTURE?

CAN YOU THINK CRITICALLY ABOUT THE SECURITY, SCALABILITY, AND RELIABILITY OF ALL OF IT?

example: startup day 1 vs. Big Mega Corp

How do we innovate?

USE IT TO OUR ADVANTAGE

chase shiny objects but with a goal in mind. have a strategy.

embrace failure without consequences.

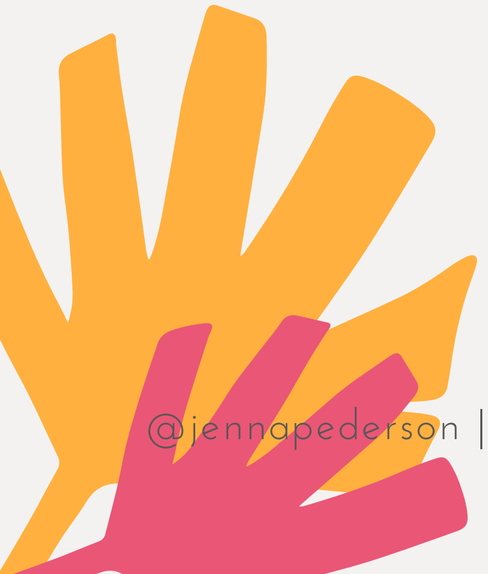
let these shiny objects drive us to learn more about our craft, how to solve real world problems with new tech, how to experiment, how to fail, and how to discover what problems we should be solving.

@jennapederson | #creamcitycode | #fallx

One Last Bit

AN INSPIRATIONAL BIT

The good devs won't just be chasing shiny objects. They will have a fine balance of finding, evaluating, and using new or bleeding edge tech vs. using the tried and true tech.



The End.

THANKS!

FIND ME ON THE INTERNET:

@jennapederson