Peter Fisher BSC MBSC

Freelance web & mobile applications developer

Host of the How To Code Well channel

youtube.com/howtocodewell

Author of Docker In Motion

From Manning Publications

40% off code dockphpsc

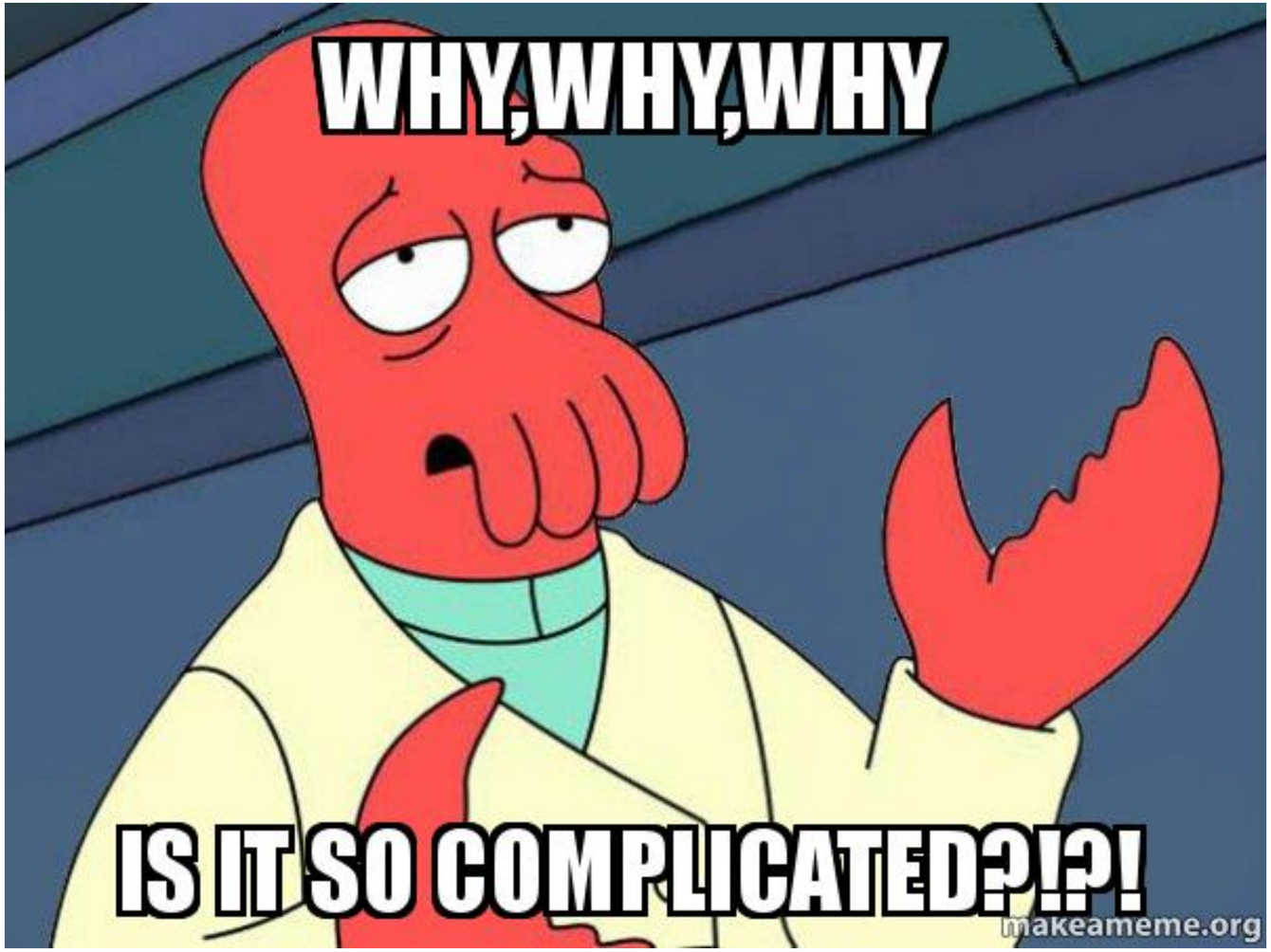
YouTube

HOW TO CODE WELL

@pfwd

Software Complexities

Peter Fisher BSC MBCS



YouTube HOW TO CODE WELL

@pfwd

Software Complexities

Peter Fisher BSC MBCS

Ask who/what is to blame

Try and answer why

The key to simplicity

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

Who/what is to blame

The Developer

The Technology

The Client

The End User

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

Who/what is to blame

The Developer

The Technology

The Client

The End User

BLAME ALL THE THINGS

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

Why is software development complex?

Peter Fisher BSC MBCS

Software is invisible
Software cannot be visualised

Peter Fisher BSC MBCS

Software is constantly changing

Peter Fisher BSC MBCS

Why is complexity so bad?

Peter Fisher BSC MBCS

Complexity leads to

Communication issues

Product flaws, cost and delays

Peter Fisher BSC MBCS

Complexity leads to

Difficulty enumerating

Less understanding of the possible states

Complexity leads to

Ugly code

Hard to integrate, maintain and extend

Complexity leads to

The unknown

Security breaches, re writes and over abstraction

Peter Fisher BSC MBCS

Complexity leads to

Fickle business decisions

Loss of data integrity, high barrier of entry and increased personnel turnover

Peter Fisher BSC MBCS

Complexity is the **most common difficulty** but

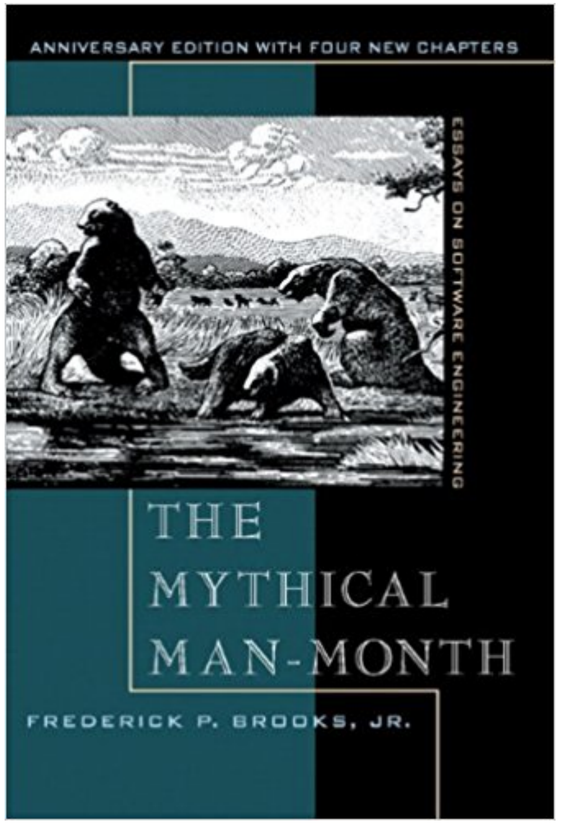
not all complexity **is inevitable**

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

The Mythical Man-Month
No Silver Bullet
Frederick P Brooks Jr



YouTube HOW TO CODE WELL

@pfwd

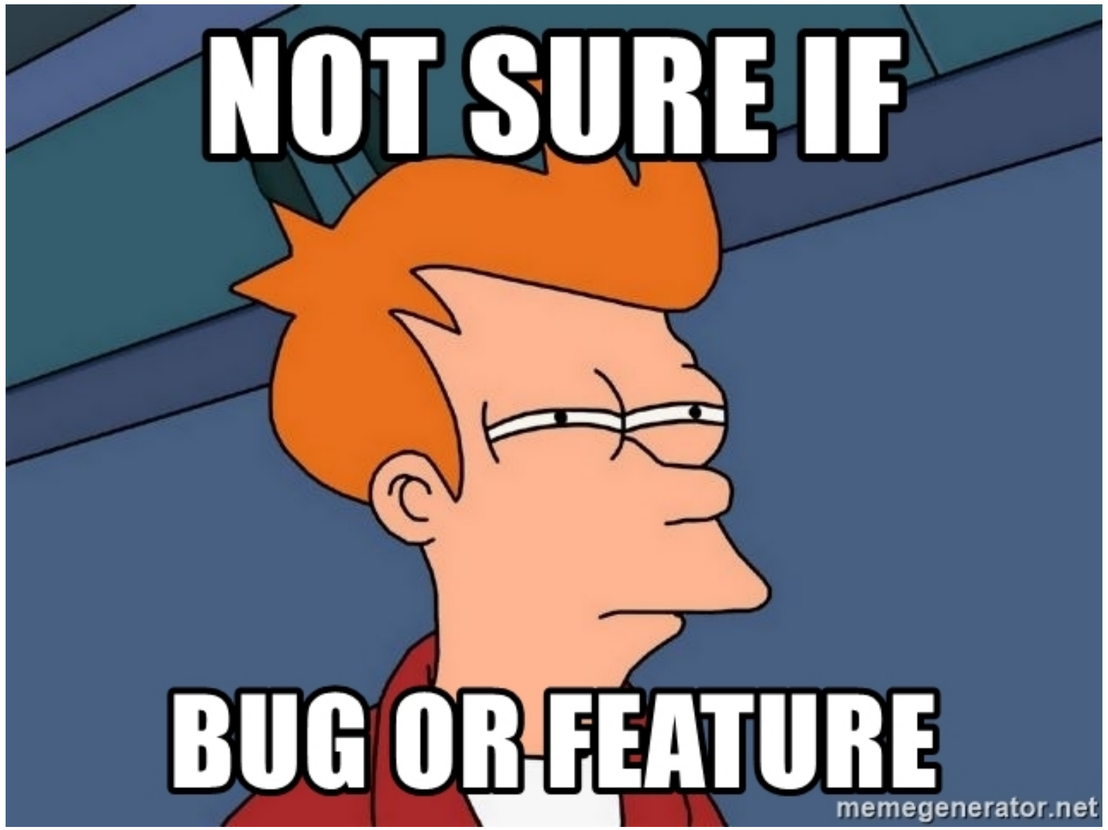
Peter Fisher BSC MBCS

Essential complexity

Vs

Accidental complexity

Essential complexity



Peter Fisher BSC MBCS

Nice to haves are **only nice** if they **enhance** the **core functionality**

Peter Fisher BSC MBCS

Write down your essential features

Include a **justification** for each essential feature

Will 80% of the system function without the essential feature?

- If so then it isn't essential!

Peter Fisher BSC MBCS

Every time a feature is added the **level of complexity is increased across the entire development life cycle** of the project

Features require

- Testing (code level, UAT, Load, Integration etc..)
- Documentation
- Training
- Designing
- Development
- Maintenance

Peter Fisher BSC MBCS

Have a **meeting every time** a **essential feature** is **added**

Peter Fisher BSC MBCS

Celebrate every time a **essential feature** is **removed**

Peter Fisher BSC MBCS

Menu A has 100 options

Menu B has 15 options

Which one is more complicated to the customer, waiter and chef?

Peter Fisher BSC MBCS

How many **essential features** do you think Twitter has?

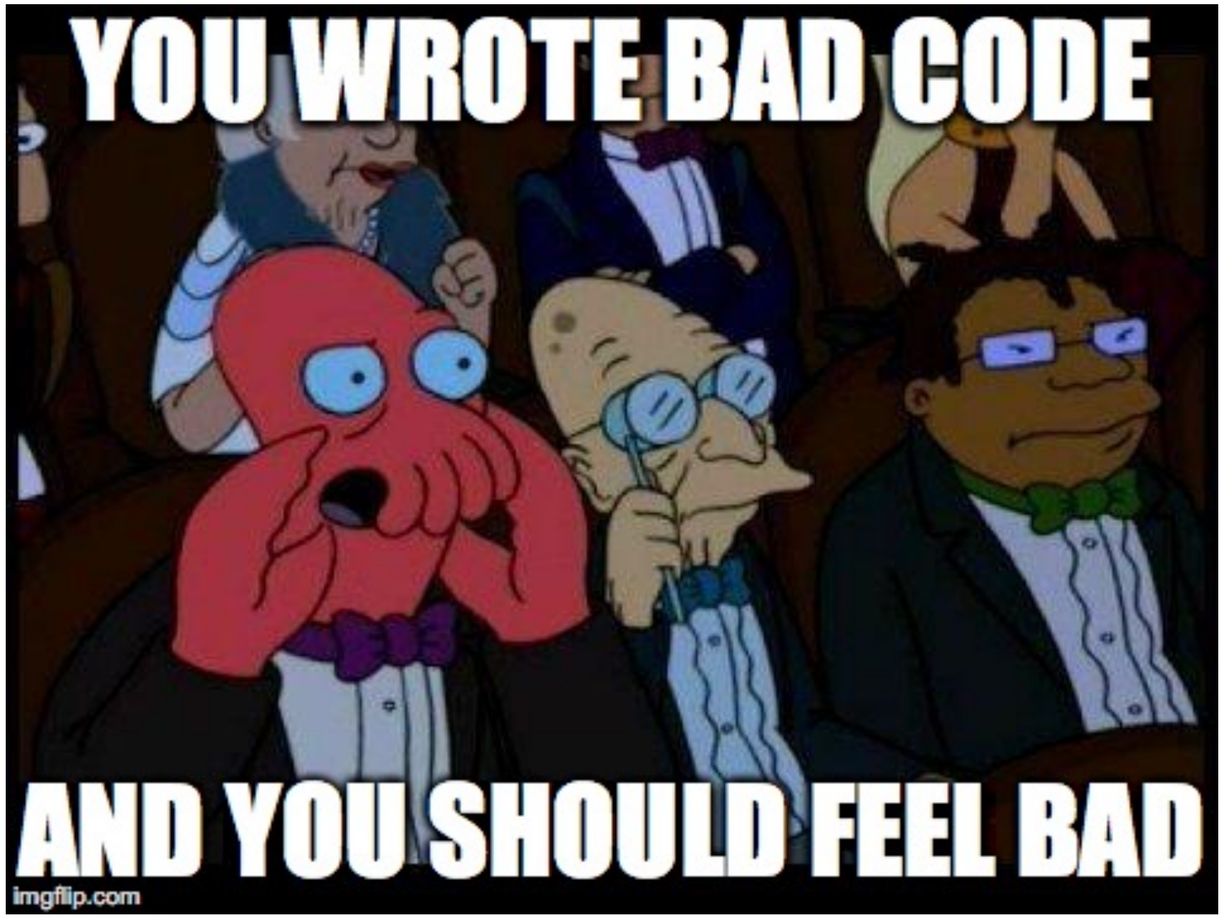
Peter Fisher BSC MBCS

If you are starting a project from scratch then

<u>ONLY</u> the essential features should be included in the first release

Peter Fisher BSC MBCS

Accidental complexity

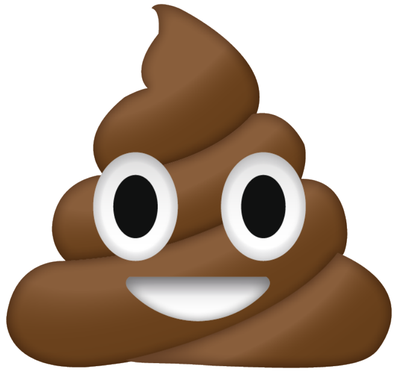


Peter Fisher BSC MBCS

New Project

Box fresh shiny toy
- Magpie like

Legacy Project



Steaming pile of source code

YouTube HOW TO CODE WELL

@pfwd

New Project

- **You** make **ALL** the **accidents**

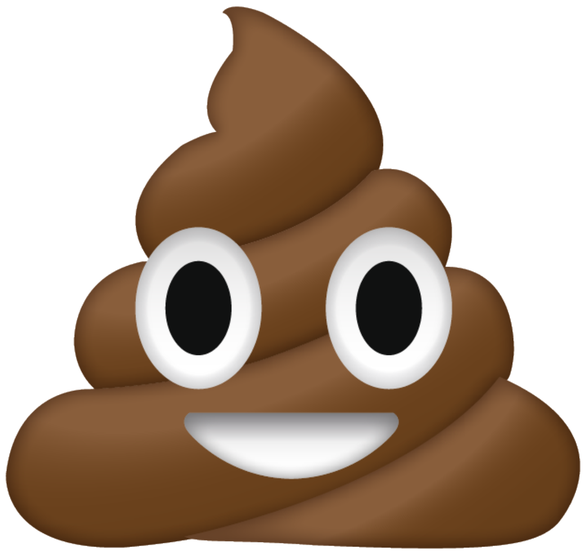
- High risk of over generalising the requirements

- **K**eep **I**t **S**imple **S**tupid

YouTube HOW TO CODE WELL

@pfwd

Legacy Project



- If you can, try to **Decouple**, **Downsize** and **Defuse** the complexity from the wider system

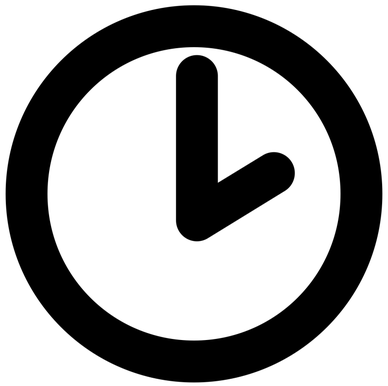
- “If it isn’t broke don’t fix it”
  This doesn’t make sense in software development

Peter Fisher BSC MBCS

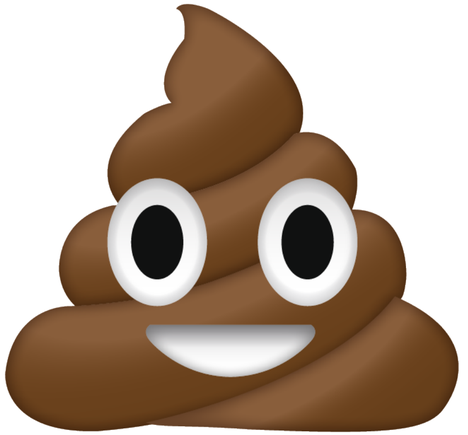
A + T = S

Accident + Time = Steaming Pile Of Source Code

+



=



YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

The keys to simplicity

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

“**Software entities** are more complex for their size than perhaps any other human construct, because **no two parts are alike**”

Brooks, F., 1995. *The Mythical Man-Month: Essays on Software Engineering*. 2nd ed. University of North Carolina at Chapel Hill: Addison-Wesley.

YouTube HOW TO CODE WELL

@pfwd

Peter Fisher BSC MBCS

Don’t **R**epeat **Y**ourself

Don’t **O**ver **A**bstract

Don’t **O**ver **G**eneralise

Peter Fisher BSC MBCS

Code that **glues two systems** together is often **easier to write** and **maintain** compared to writing a **monolith from scratch**

Peter Fisher BSC MBCS

Buy versus build

Peter Fisher BSC MBCS

Software should be **grown** not **built**

@pfwd

"We still make syntax errors, to be sure; but they are fuzz compared to the **conceptual errors** in most systems. If this is true, building software **will always be hard**. There is inherently **no silver bullet**."

Brooks, F., 1995. *The Mythical Man-Month: Essays on Software Engineering*. 2nd ed. University of North Carolina at Chapel Hill: Addison-Wesley.

Peter Fisher BSC MBCS

Getting **simplicity** right **is complicated**

@pfwd

Thank you. 🙏 give feedback
https://joind.in/talk/4503f

YouTube HOW TO CODE WELL

@pfwd