WRITING GOOD CODE WITH ACRONYMS

Jan Kalbermatter

CUPID 💘- for Joyful Coding

Created by Daniel Terhorst North

“I believe that there are properties or characteristics of software that make it a joy to work with. The more your code has these qualities, the more joyful it is to work with; but everything is a trade-off so you should always consider your context.”

Presentation in 2017 - Why Every Element of SOLID is Wrong

Remember SOLID?

Single Responsibility Principle

Code should have one, and only one, reason to change

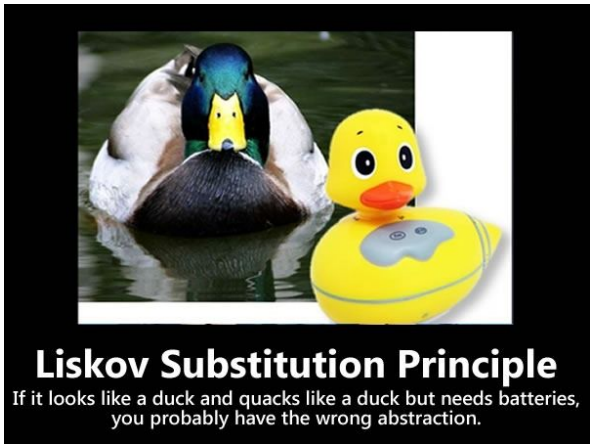
Open/Closed Principle

Open for extension, closed for modification

Remember SOLID?

Liskov Substitution Principle

Being able to replace the subclass everywhere the parent class is used without breaking the meaning of the code



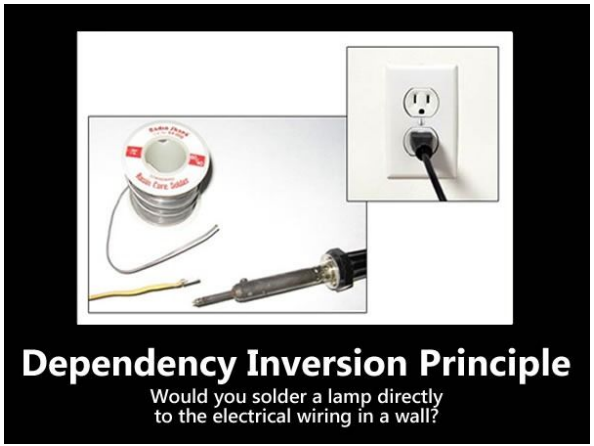
Interface Segregation Principle

Clients should not be forced to depend upon interfaces that they do not use

Remember SOLID?

Dependency Inversion Principle

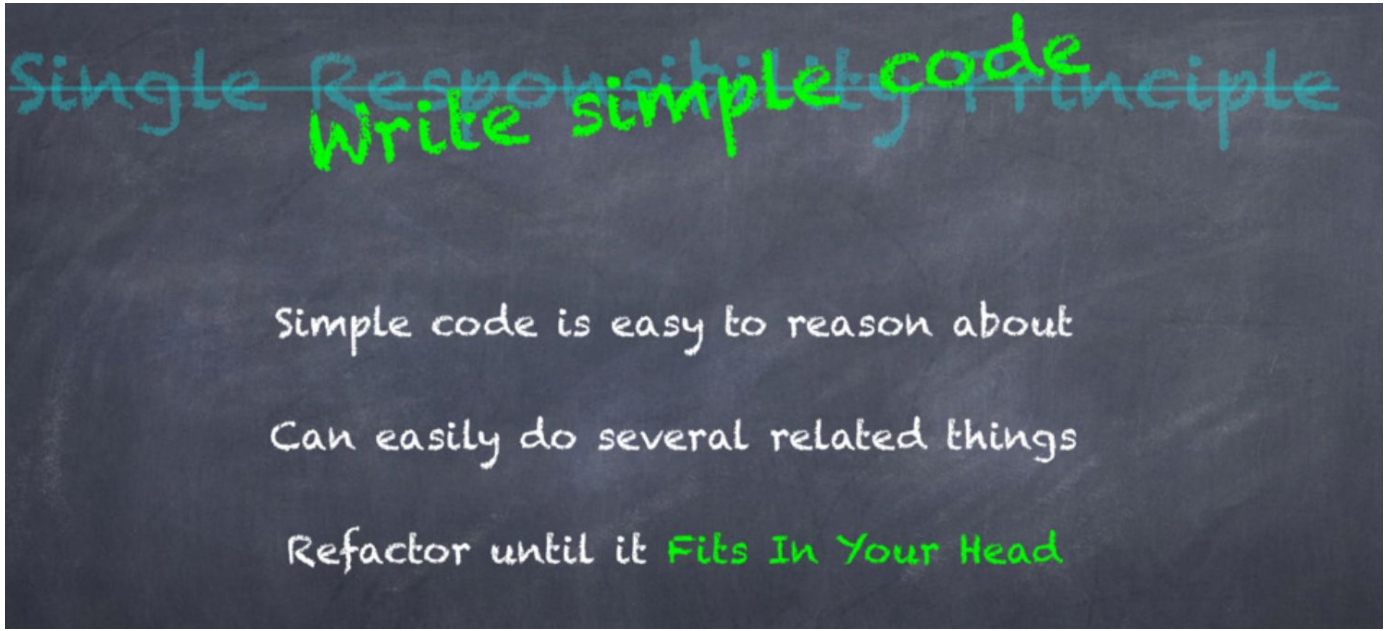
High-level modules should not depend on low-level modules -> Both should depend on abstractions



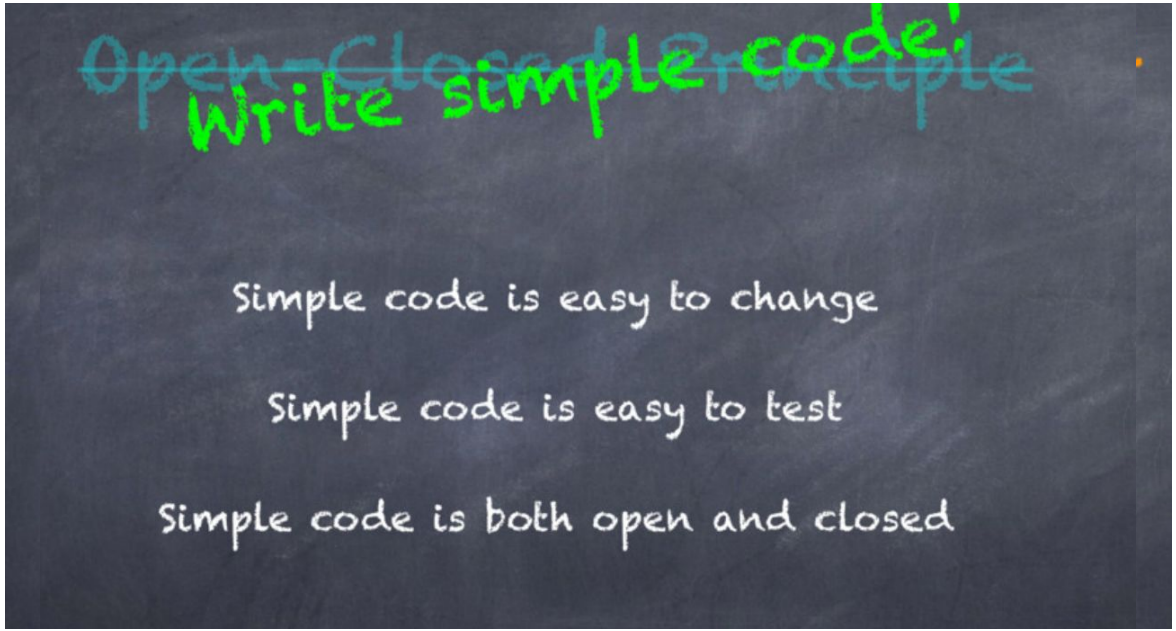
SOLID++

- Balanced Abstraction ⚖️
  - Group same level of abstraction
- Least Astonishment (WTF) 🤯
  - Minimize wtf/min

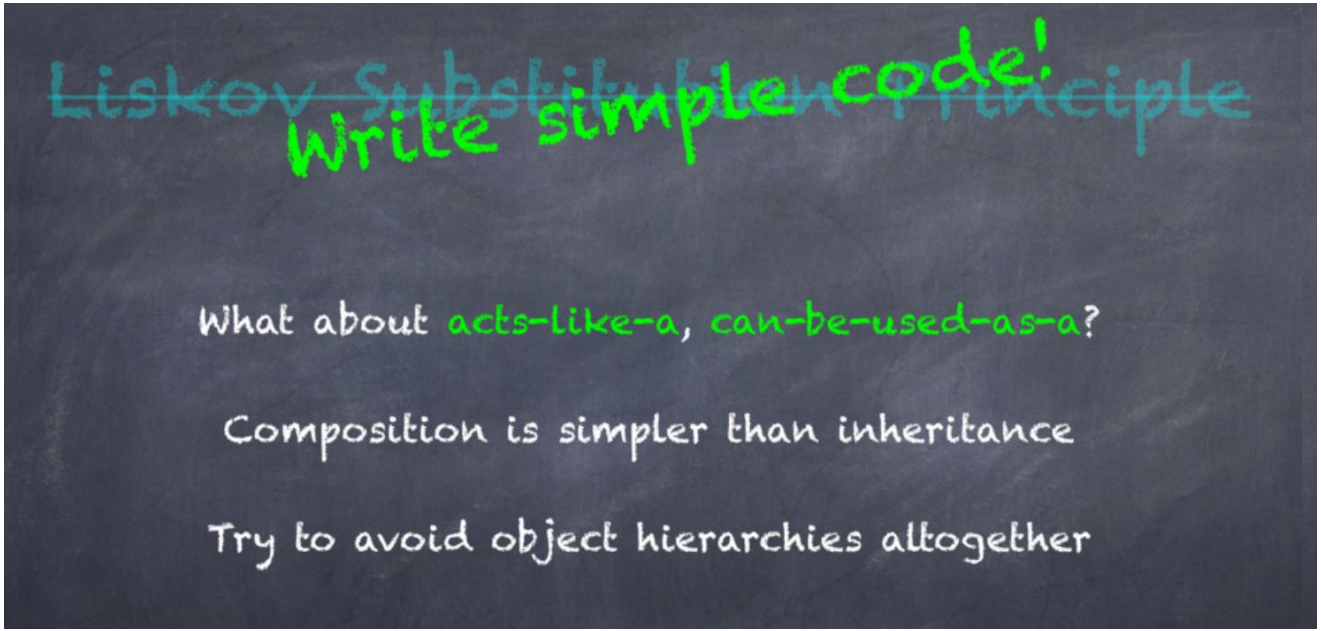
Pointlessly Vague Principle



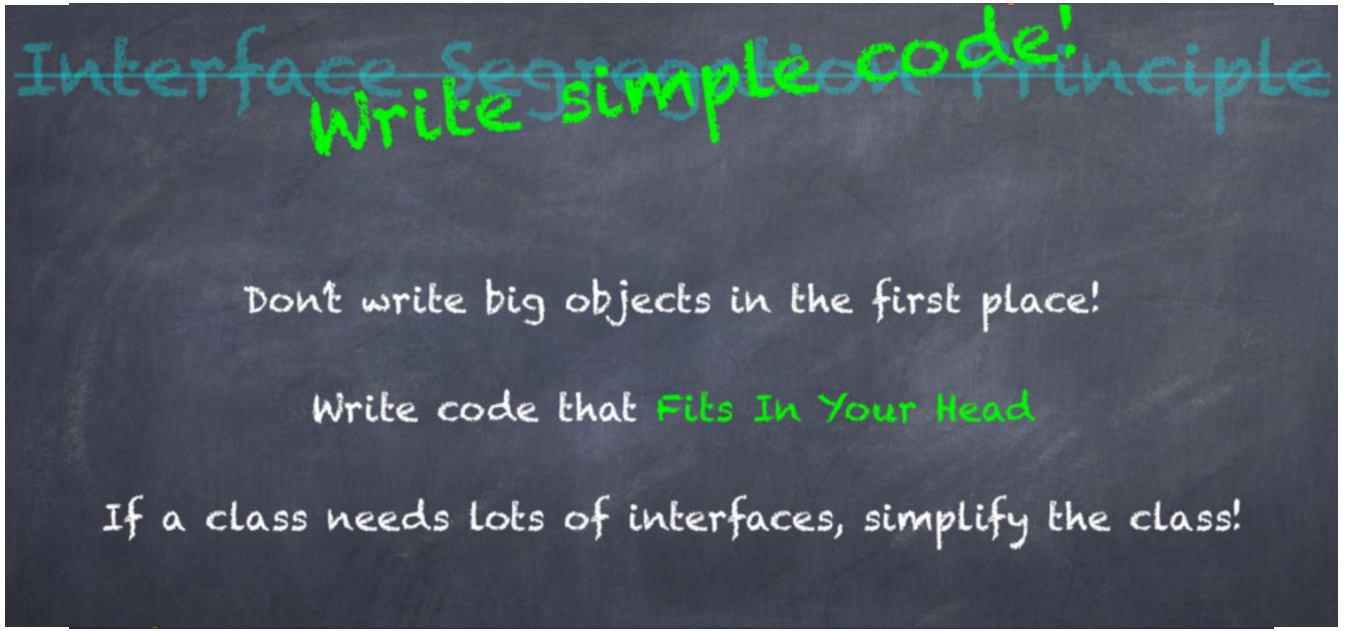
Cruft Accretion Principle



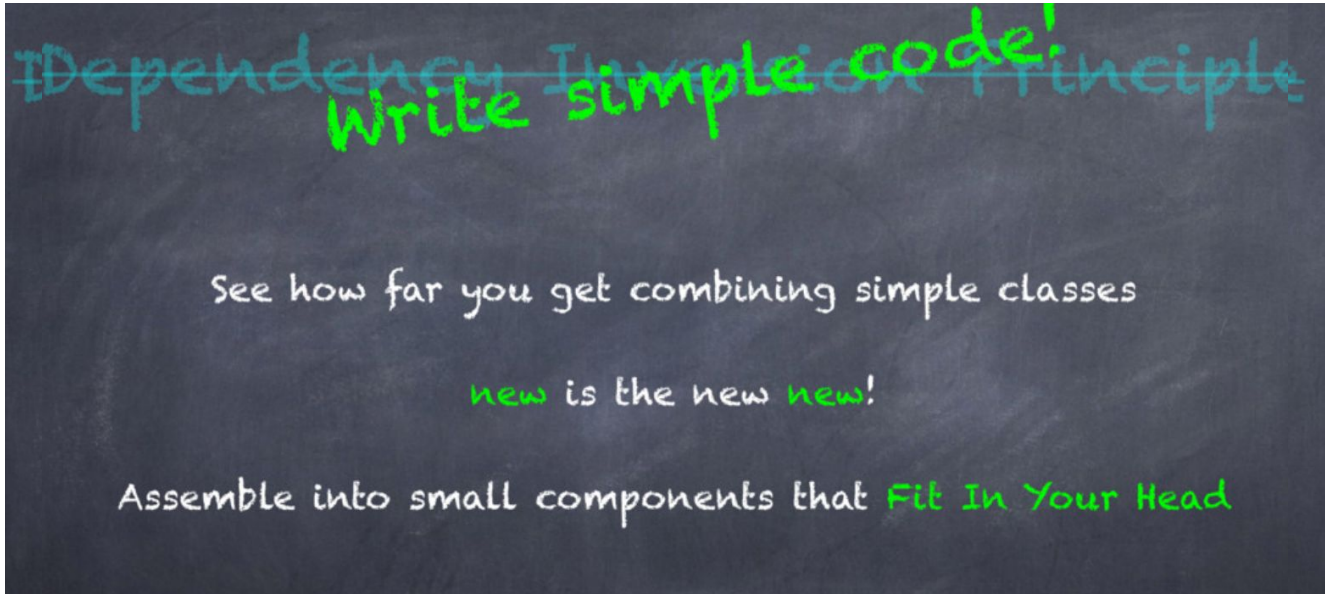
Drucker's Warning Principle



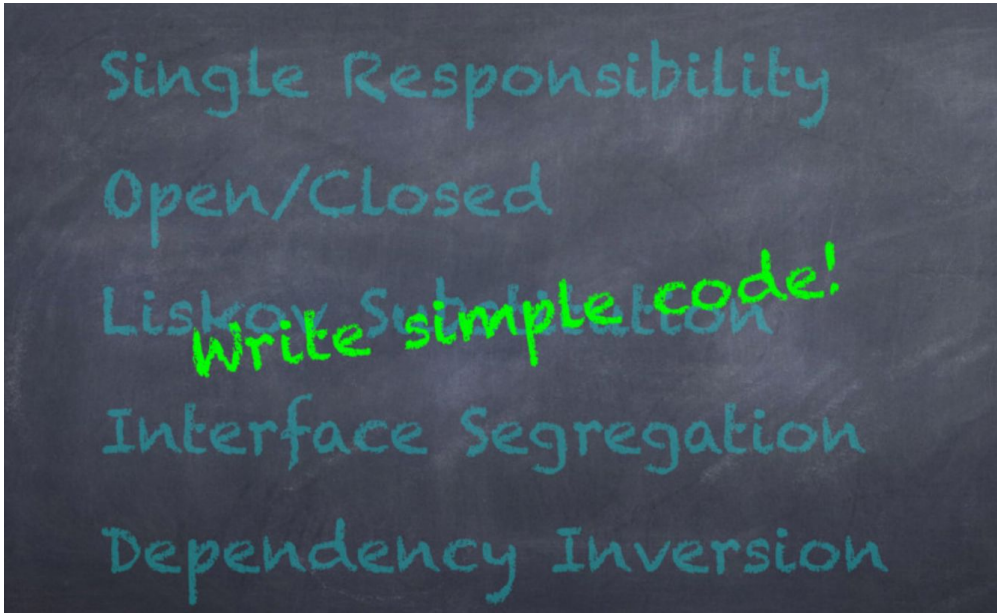
Stable Door Principle



Wrong Goal Principle



Write simple Code?



What CUPID is all about

- Writing Software that is a Joy to work with
  - Caring about out fellow programmers ❤️
- Properties instead of Principles
  - Three Properties the CUPID Properties all have
    - Practical
    - Human
    - Layered

Bringing joy by making your software...

Composable 🧩

aka. plays well with others

- Easy to reuse
- Small surface area
- Intention Revealing
- Minimal Dependencies

Unix Philosophy 🐧

aka. does one thing well

- Sounds just like SRP?
- Outside-in perspective

Bringing joy by making your software...

Predictable 🔮

aka. does what you expect

- Behaves as expected
- Deterministic
- Observable

Idiomatic 🌳

aka. feels natural

- Conform to language idioms
- Conform with local idioms (made up by team)

Bringing joy by making your software...

Domain-based🔗

aka. the solution domain models the problem domain in language and structure

- for language
- for structure

Is CUPID stupid or is SOLID… not so solid?

CUPID more straight forward at first glance

SOLID = Set of principle to use during development
CUPID = Properties that the finished code should have

CUPID is less restrictive -> does not care how you achieve the five properties

They follow the same goal!

Thanks for listening

More on Dan North & CUPID:
https://dannorth.net/2022/02/10/cupid-for-joyful-coding/

Dan North - Why Every Element of SOLID is Wrong:
https://speakerdeck.com/tastapod/why-every-element-of-solid-is-wrong

Jan Kalbermatter
✉️ jan.kalbermatter@gmail.com