



SOLID Principles

How to Keep Building Better

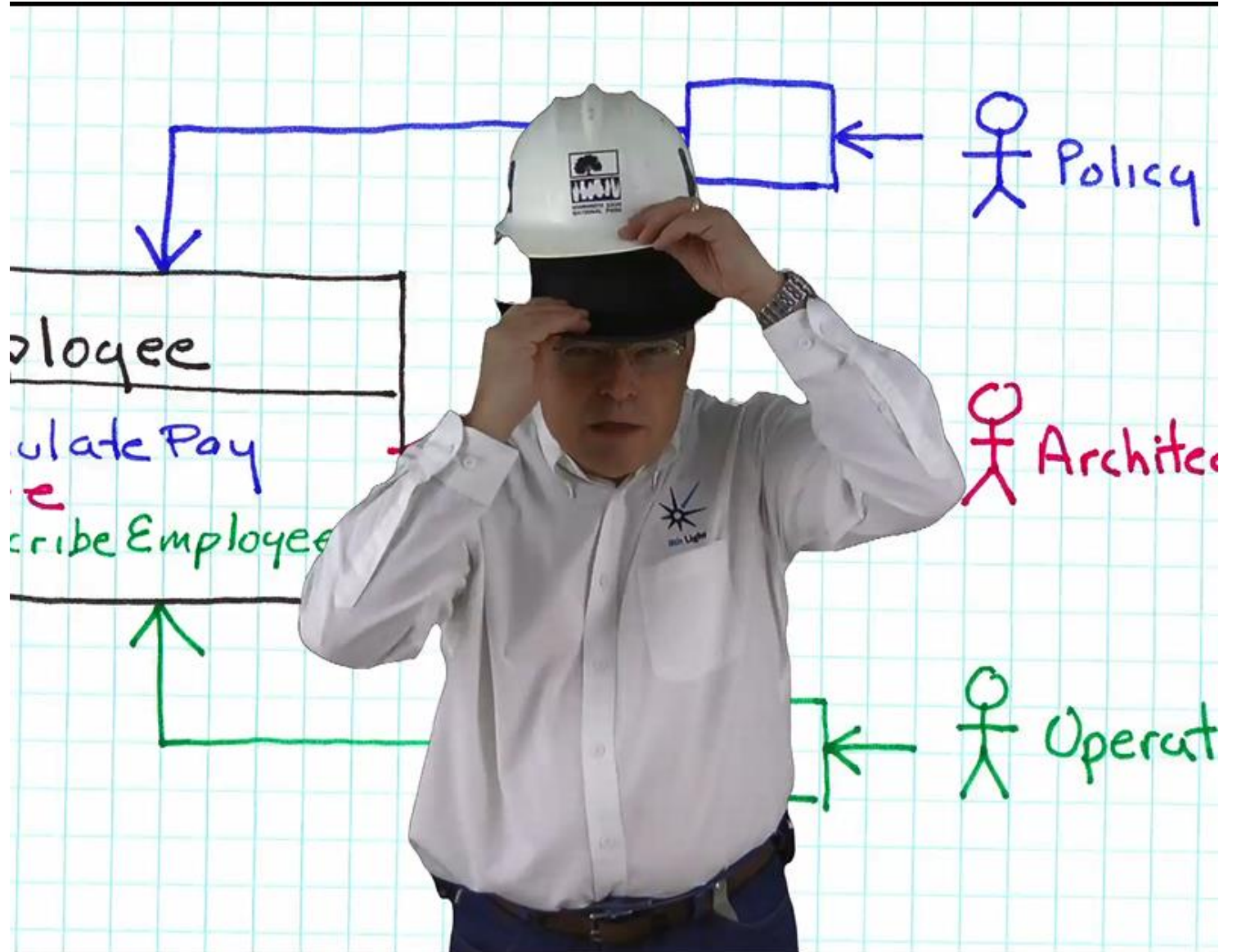
Robert “Uncle Bob” Martin

Design Principles and Design
Patterns



Single Responsibility Principle

A class must have only one reason to change and only one responsibility.



The Open Closed Principle

A module should be open for extension but closed for modification.

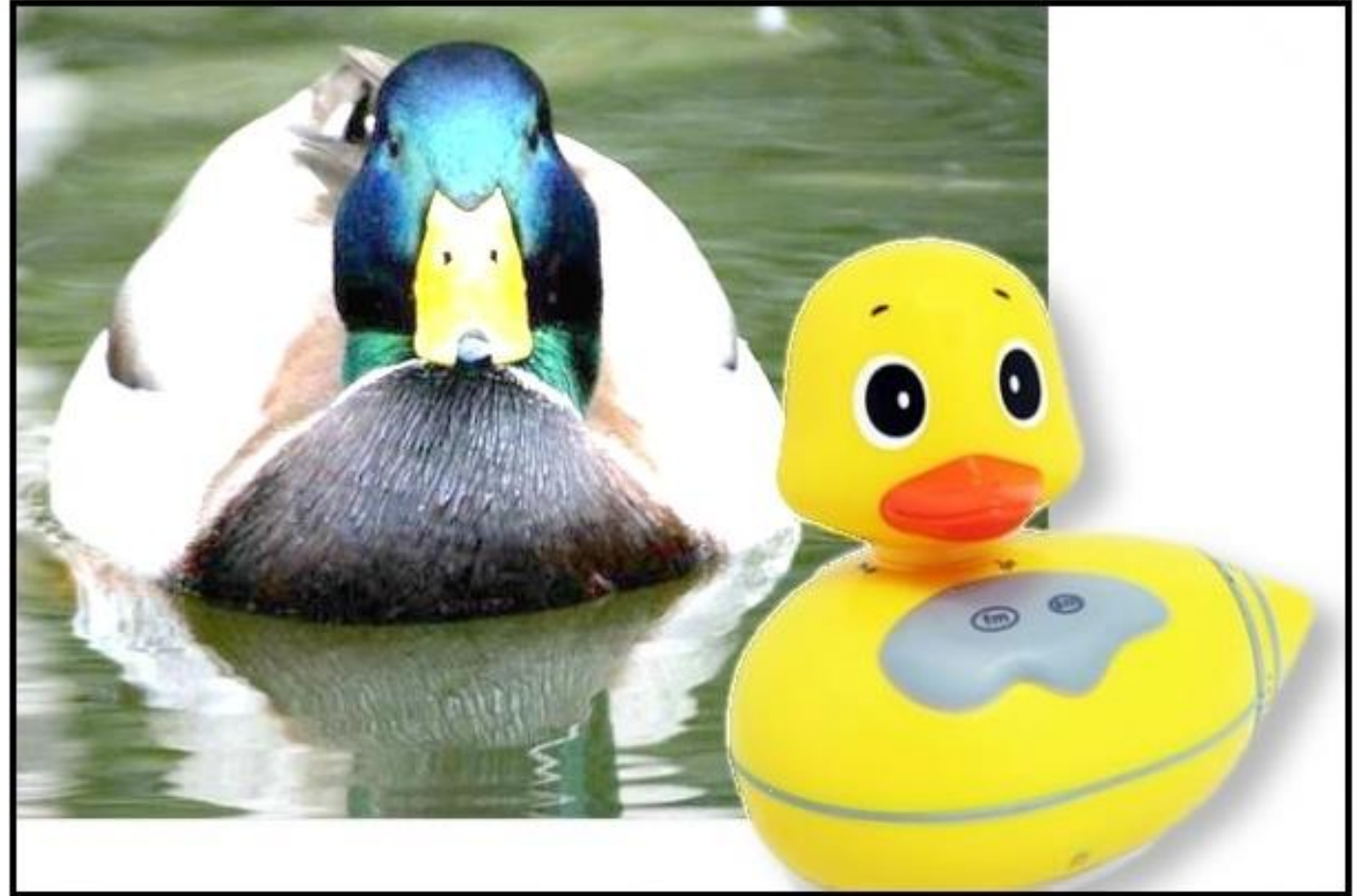


Open-Closed Principle

Open-chest surgery isn't needed when putting on a coat.

Liskov Substitution Principle

Subclasses should be substitutable for their base classes.



LISKOV SUBSTITUTION PRINCIPLE

If It Looks Like A Duck, Quacks Like A Duck, But Needs Batteries - You Probably Have The Wrong Abstraction

Interface Segregation Principle

Many client specific interfaces are better than one general purpose interface.



Interface Segregation Principle

If IRequireFood, I want to Eat(Food food) not,
LightCandelabra() or LayoutCutlery(CutleryLayout preferredLayout)

Dependency Inversion Principle

*Depend upon Abstractions.
Do not depend upon
concretions.*



DEPENDENCY INVERSION PRINCIPLE

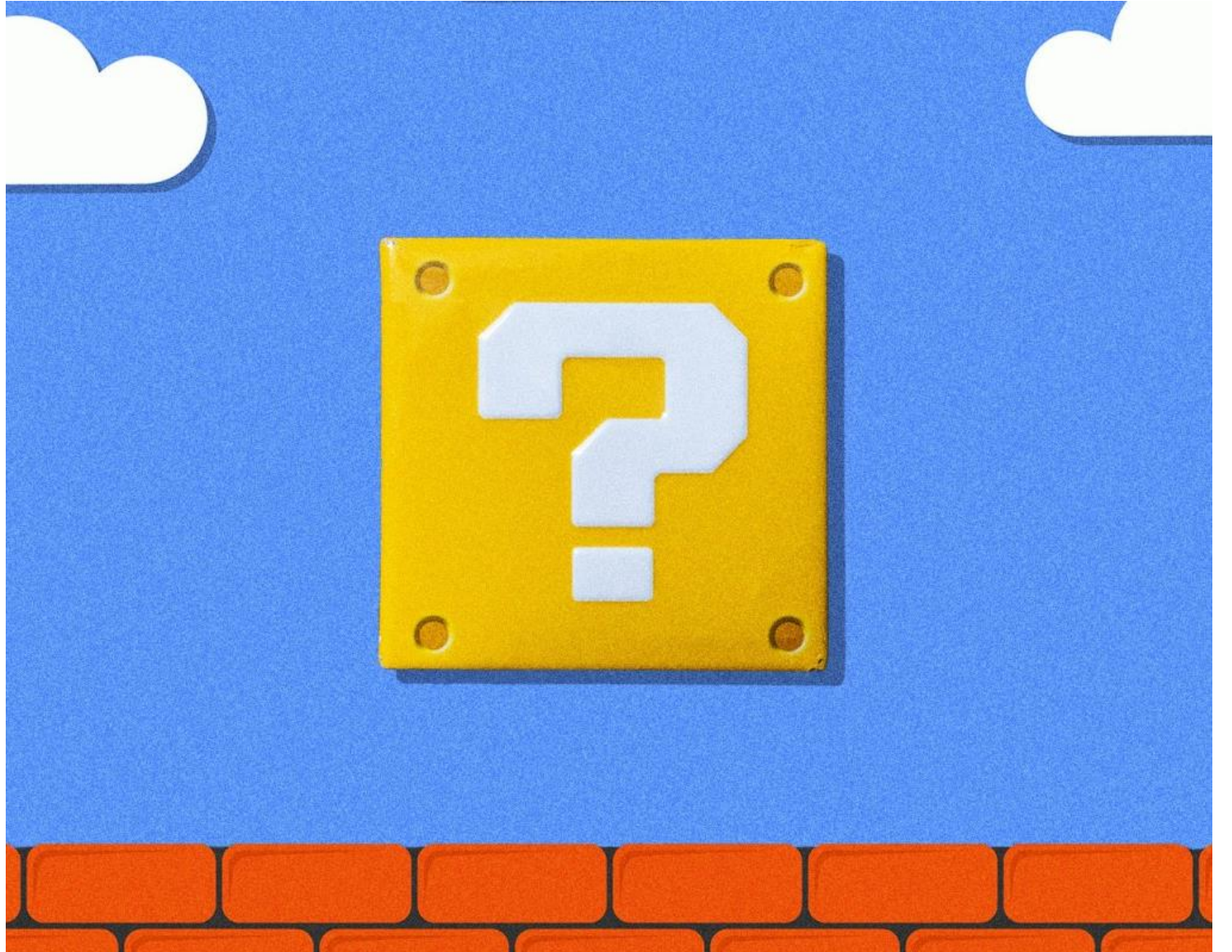
Would You Solder A Lamp Directly To The Electrical Wiring In A Wall?

Why?

Simple, keep your code
maintainable!



Any
Questions?



Thank You

References

- [Design Principles and Design Patterns](#) – Robert C. Martin
- [Data Abstraction and Hierarchy](#) – Barbara Liskov