

# Refactoring with Design Patterns

Qafoo GmbH  
January 25, 2014

- ▶ Working at Qafoo



**We promote high quality code with trainings and consulting**

<http://qafoo.com>

- ▶ Doctrine Developer
- ▶ Symfony Contributor
- ▶ Twitter @beberlei and @qafoo

## Neglecting Object-Oriented design leads to underengineering

but focusing solely on Design-Pattern  
leads to overengineering

# Refactoring

---

- ▶ small changes to internal code structure
- ▶ Apply extract method and class again and again
- ▶ See Matthias Verraes talk tomorrow, 16:10

Refactoring towards Patterns to avoid  
both under- and overengineering.

<http://qa.fo/phpbnl14>

A factory creates an object for you.

- ▶ Getting control over object creation
- ▶ Most important issue for every code-base
- ▶ Actually 4 patterns
  - ▶ Factory
  - ▶ Factory Method
  - ▶ Abstract Factory
  - ▶ Builder



A facade provides a simplified interface to a larger body of code.

- ▶ Make code reusable (business logic, ..)
- ▶ Integrate third party code (libraries)
- ▶ Avoid hard dependencies on technical details
- ▶ Strongly Related to the Adapter/Bridge patterns

# Strategy/Policy Pattern

---

Strategy allows to exchange algorithms at run time.

- ▶ Object-oriented `switch` statement
- ▶ When calculations are changing frequently
- ▶ Or when they change based on state
- ▶ Construction of strategies often combined with a factory

<https://joind.in/10287>



THANK YOU

Rent a quality expert  
[qafoo.com](https://qafoo.com)