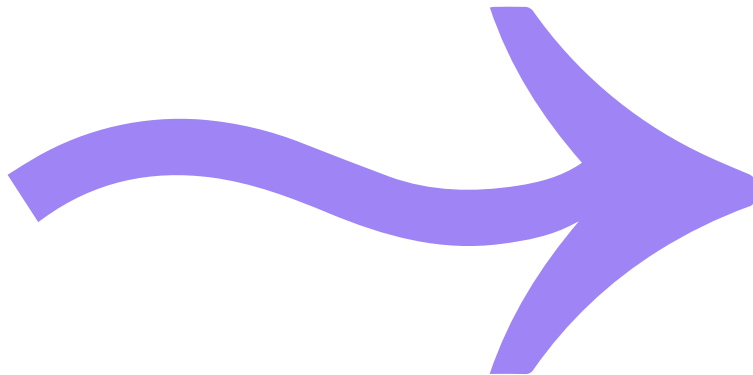




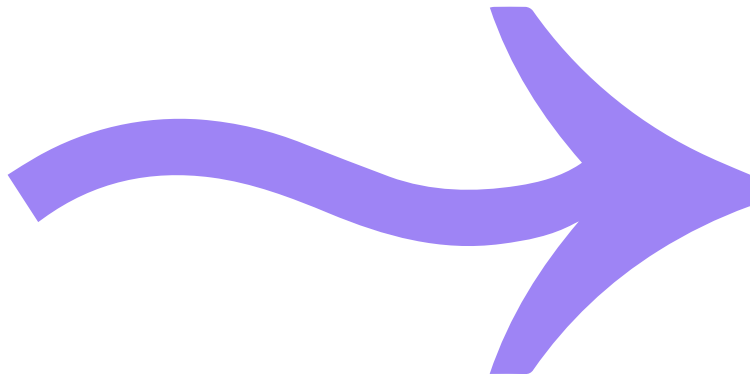
# Software Design Patterns





# What are Software Design Patterns?

- Reusable solutions to commonly occurring problems
- Emerge over time as developers write code and establish best practices
- Solve particular problems in code generation and interactions
- Powerful if used in the correct way and applied in the correct circumstance





# Who were the 'Gang of Four'?

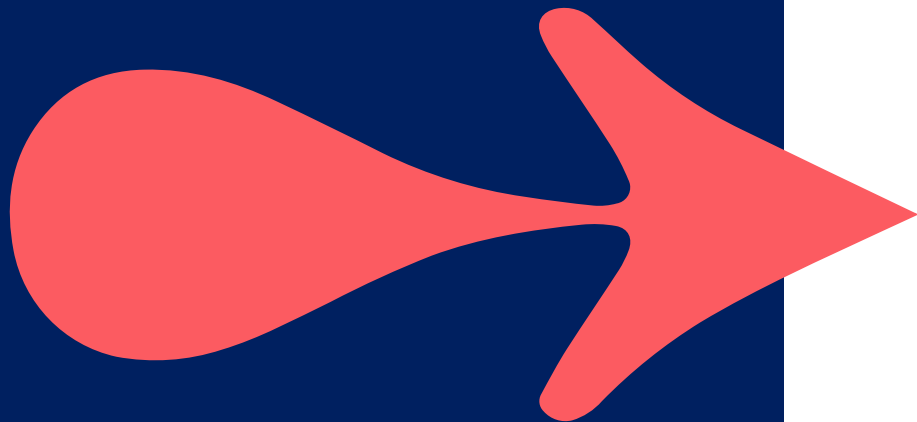
*Design Patterns: Elements of Reusable Object-Oriented Software (1994)*

- Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides

'Capturing a wealth of experience about the design of object-oriented software, **four top-notch designers present a catalog of simple and succinct solutions to commonly occurring design problems.**

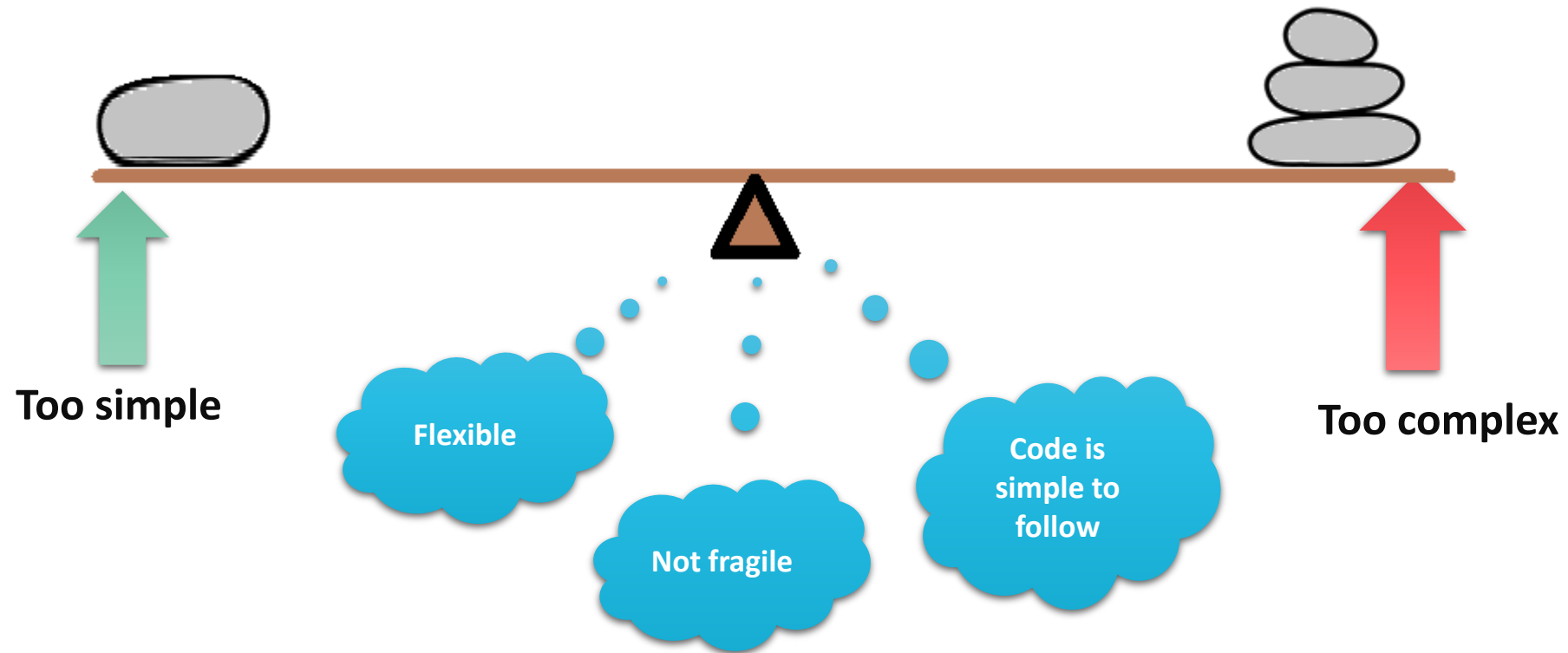
Previously undocumented, these 23 patterns allow designers to create more **flexible, elegant, and ultimately reusable designs** without having to rediscover the design solutions themselves.'

- Addison-Wesley publishers, emphasis added for this session



# The Principle of Balance

Find the balance between...





# Common Design Patterns

- **Facade**
- **Proxy**
- **Command**
- **Observer** (subsumed into .NET as events)
- **State**
- **Strategy / Template Method**
- **Factory**
- **Singleton**

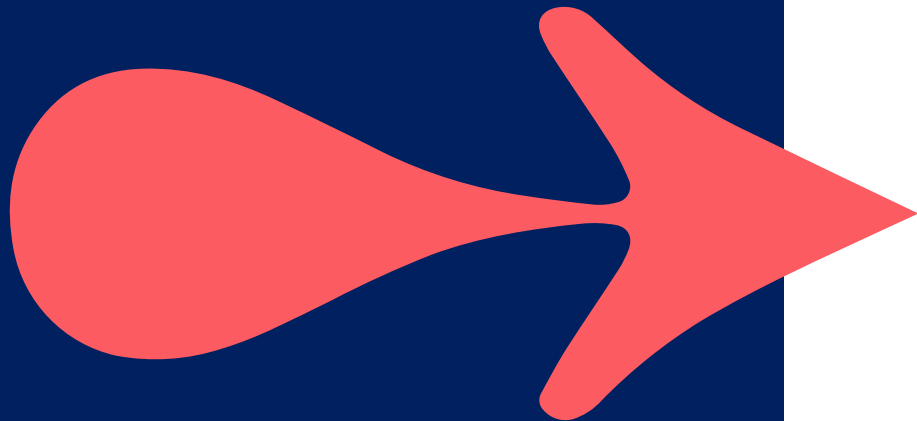
<https://refactoring.guru/design-patterns/csharp>



# Gang of Four Patterns

**Three types of pattern:**

- **Creational**
- **Structural**
- **Behavioural**



<https://refactoring.guru/design-patterns/catalog>

# Lab

