

Origins of Design Patterns

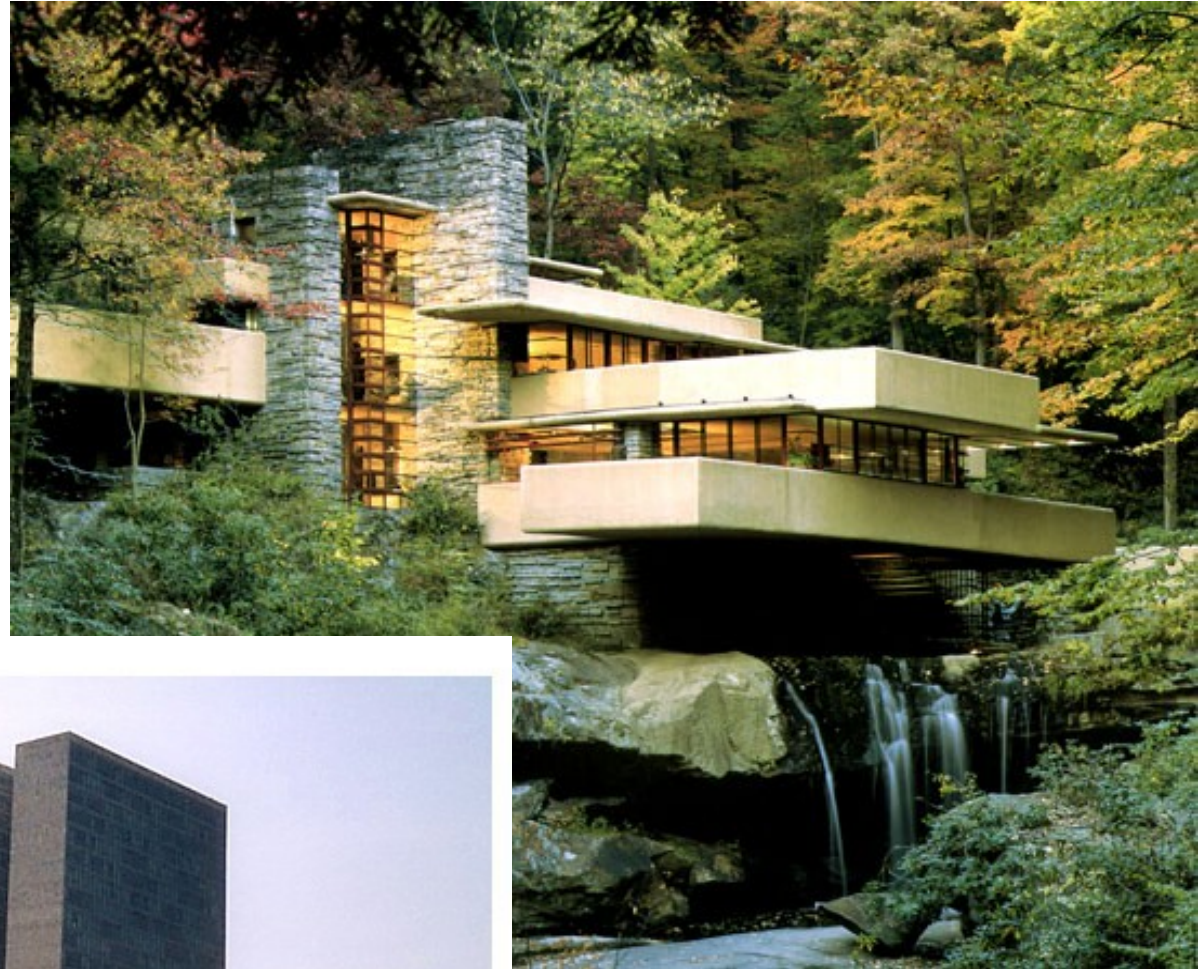
In the 1970s, an architect named **Christopher Alexander** tried to figure out what makes an **architectural design** “good”.

Alexander proposed that there was an objective way of **measuring quality of design**.

He studied the architecture of many cities (buildings, streets, parks, etc).

He discovered **patterns** which led to high quality.

Good Design

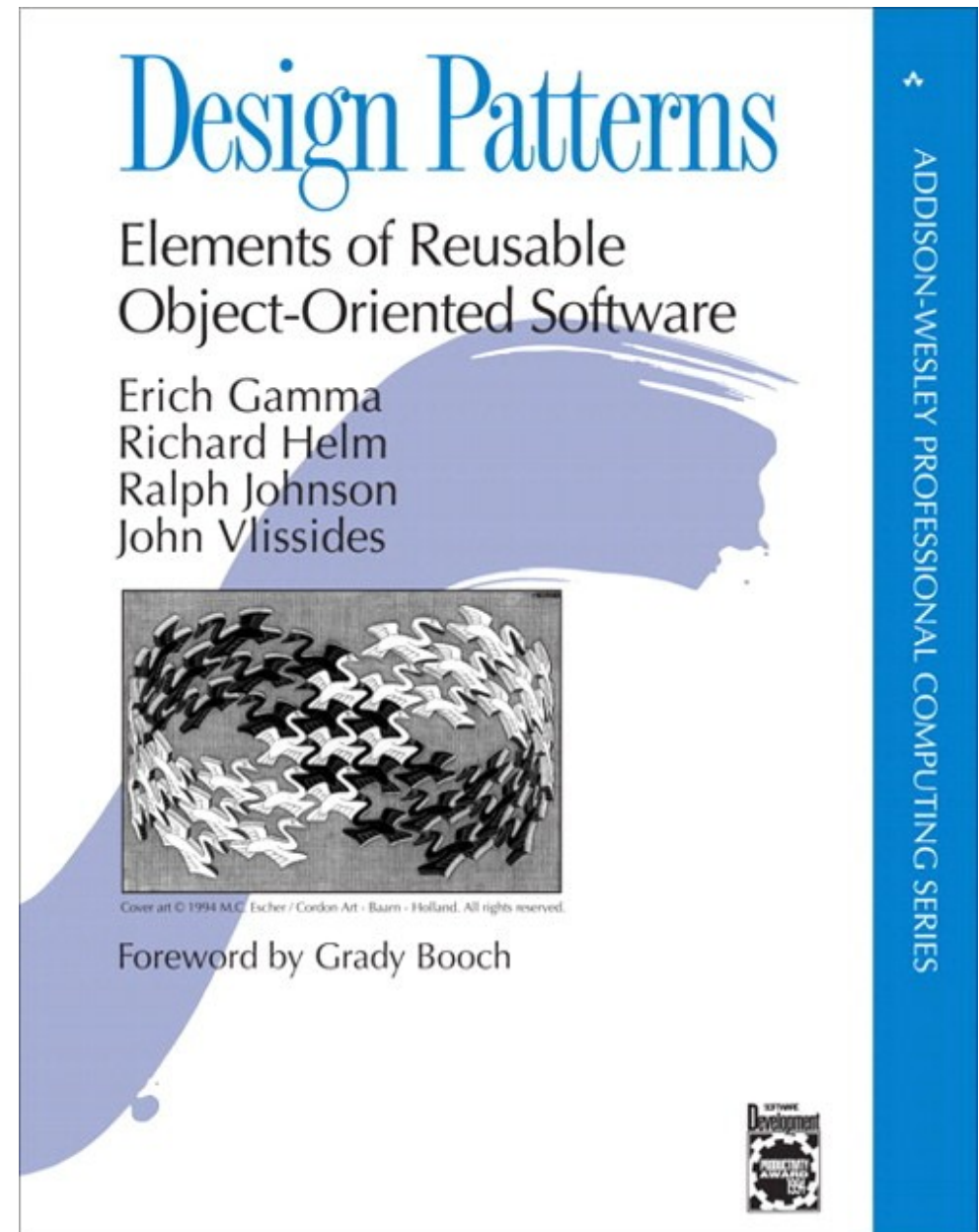


Gang of Four

In 1987, Kent Beck and Ward Cunningham began experimenting with Design Patterns.

They believed that this idea of patterns as solutions to common problems could be used with software.

In 1994, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides published Design Patterns: Elements of Reusable Object-Oriented Software.



What are Patterns?

*"A pattern for **software architecture** describes a particular **recurring design problem** that arises in specific design contexts and presents a well-proven **generic scheme for its solution**. The solution scheme is specified by describing its constituent components, their responsibilities and relationships, and the ways in which they collaborate." [Buschmann].*

Patterns are a **solution** to a **problem** in a **context**.

Patterns are not invented, they are **derived from practical experience**.

Patterns are **building blocks**, to be **combined and adapted** to solve complex problems.

Patterns can be used as a **vocabulary** to communicate at a high level of **abstraction**.

Why use Design Patterns?

Because Patterns are well tested and well proven solution to common problems.

They have been successfully used in the past.

They are a form of reuse.

Patterns are to Design what

Algorithms and Data Structures are to Code.

Components of a Design Pattern

Name

Each pattern has an assigned name so it can be easily recognized. This gives us the vocabulary we can use to discuss design.

Problem (context)

Each pattern is design to address to a specific problem. Some also have conditions before the pattern can be used.

Solution

Each pattern provides a solution to a problem. Components of that solution are also known as Participants.

Consequences

They are the results and trade-offs of using design patterns.

Types of Design Patterns

Creational Patterns

These patterns abstract the instantiation process.

They make the system independent of how objects are created, composed and represented.

Structural Patterns

These patterns are concerned with how classes and objects are composed to form larger structures.

These structures are use to provide new functionality

Behavioural Patterns

These patterns are concerned with algorithms and the assignment of responsibility between objects.

They describe the communication between objects as well as internal behaviour.

The Book

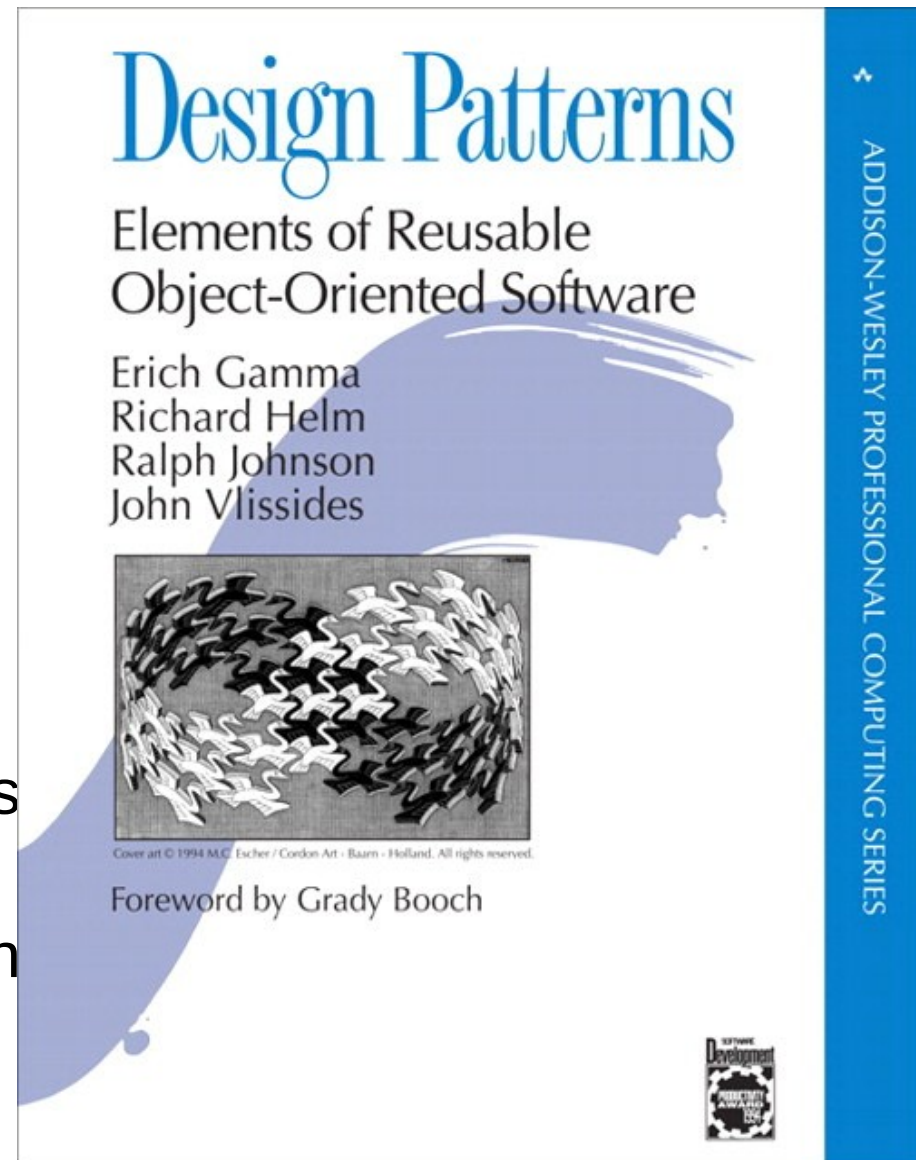
The Design Patterns book is a **catalog** of design patterns.

When faced with a design pattern, one should:

Browse the catalog to determine if a particular design pattern solves this pattern.

If so, before implementing the solution, Carefully identify the various participants of the problems.

Study thoroughly the appropriate section in the book, particularly the consequence and implementation section.



Didn't I do this before?

The material you will see in Design Patterns is not new. Some of you might have discovered these patterns yourselves.

That's the whole point.

It's a catalog of good design.

If you have already been using a Pattern, then

You now have an official name for it.

You know it good design.

You might gain a few new insights into on how to use it and above all into trade-offs.

In this course ...

Singleton

Command

Adapter

Composite

Observer

Template Method

Visitor

Factory

Proxy

Strategy

