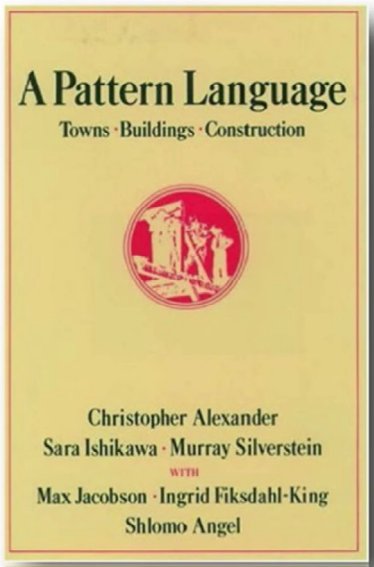
Miscellaneous Items

Design Patterns in Architecture

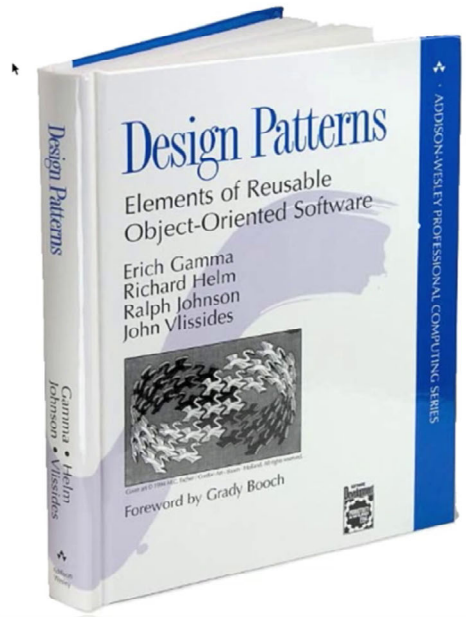


Christopher Alexander

Oxford University Press (1977)

*buildings, not software

Gang of Four



Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Published: 1984

Pattern types

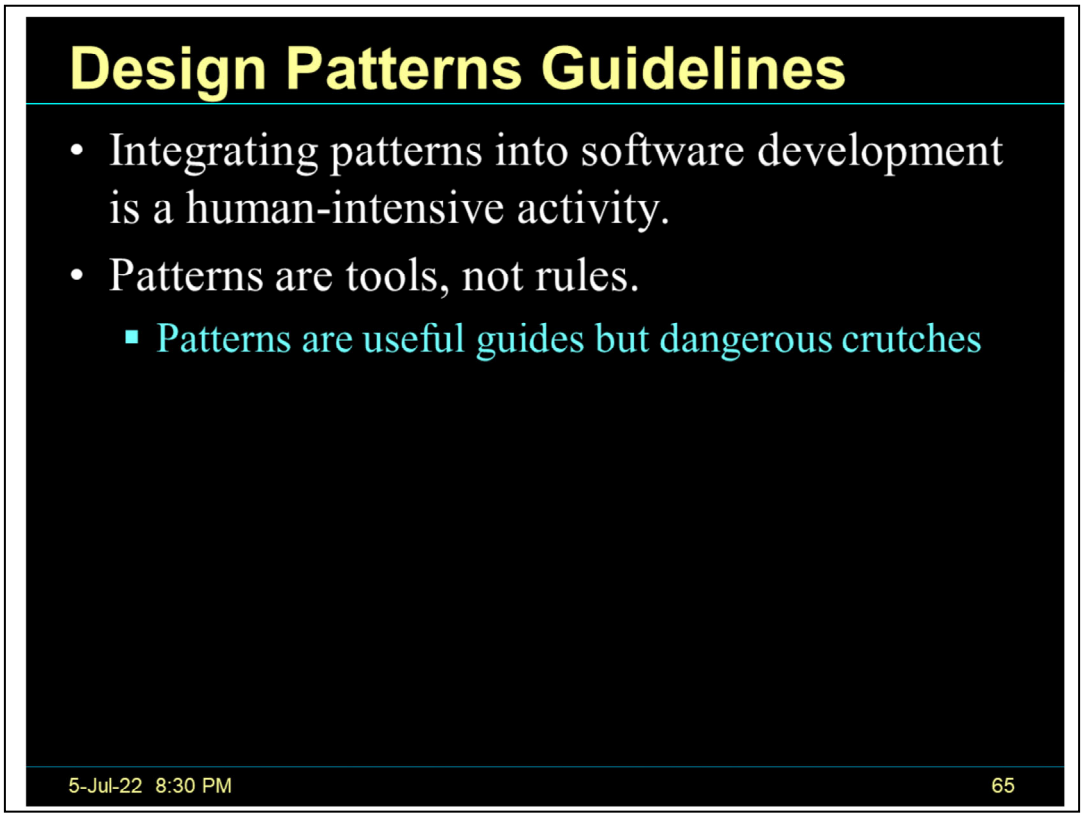
- **Creational Patterns:** Used to construct objects such that they can be decoupled from their implementing system.
- **Structural Patterns:** Used to form large object structures between many disparate objects.
- **Behavioral Patterns:** Used to manage algorithms, relationships, and responsibilities between objects.

Classification

Creational	Structural	Behavioral
Factory	Adapter	Interpreter
Abstract Factory	Bridge	Template
Builder	Composite	Chain of Responsibility
Prototype	Decorator	Command
Singleton	Flyweight	Iterator
	Facade	Mediator
	Proxy	Memento
		Observer
		State
		Strategy
		Visitor

Popularity

- Flyweight, Proxy, Command, Iterator, Strategy
- Factory, Abstract Factory, Singleton, Façade, Template, Observer, Builder, Adapter
- Prototype, Bridge, Composite, Decorator, Interpreter, Chain of Responsibility, Mediator, Memento, State, Visitor



Patterns do not lead to direct code reuse.

Patterns are tools:

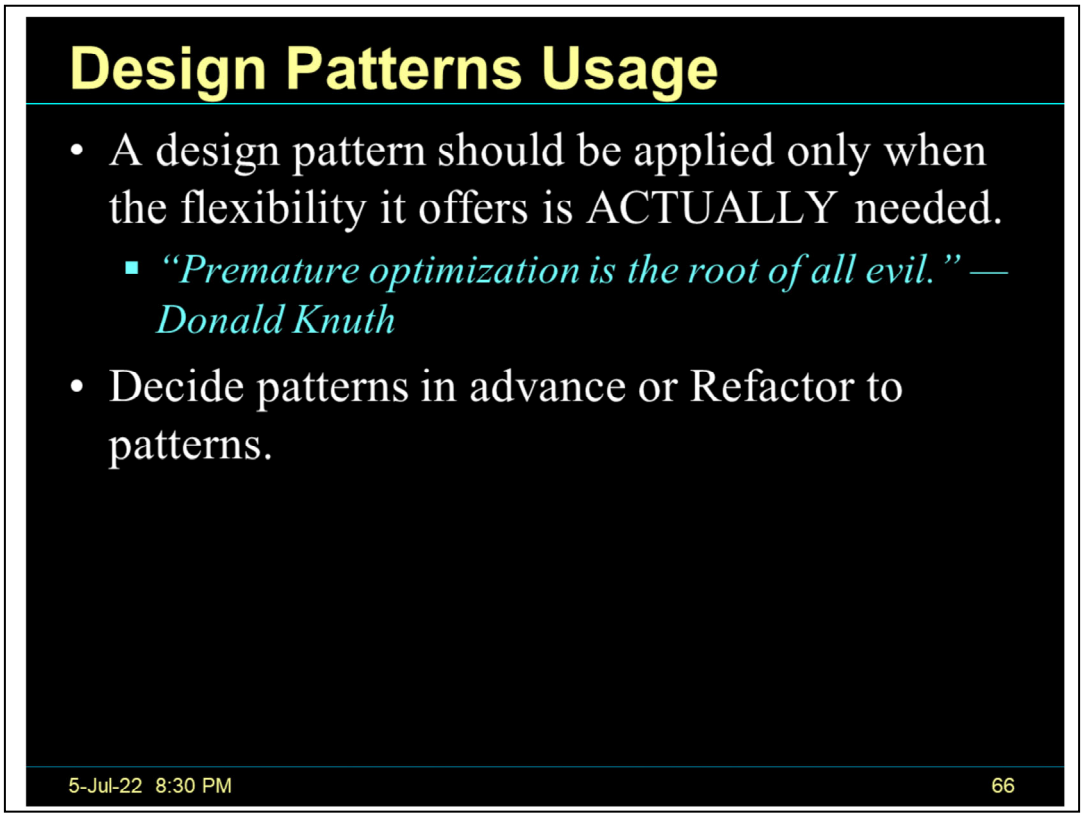
They need to be tweaked and adapted to the problem at hand.

A Design pattern can be applied only if we are clear about what we want to do.

A design pattern tells how to solve a particular problem.

How often do programmers/designers have a clear idea about the problem?

So many people find the use of Design patterns frustrating.



Often design patterns achieve the flexibility by introducing additional levels of indirection. They add complexity to our code.

Like optimization, we must delay using a DP.

Apply design patterns extensively while designing a framework / library interface for external world.

For custom applications

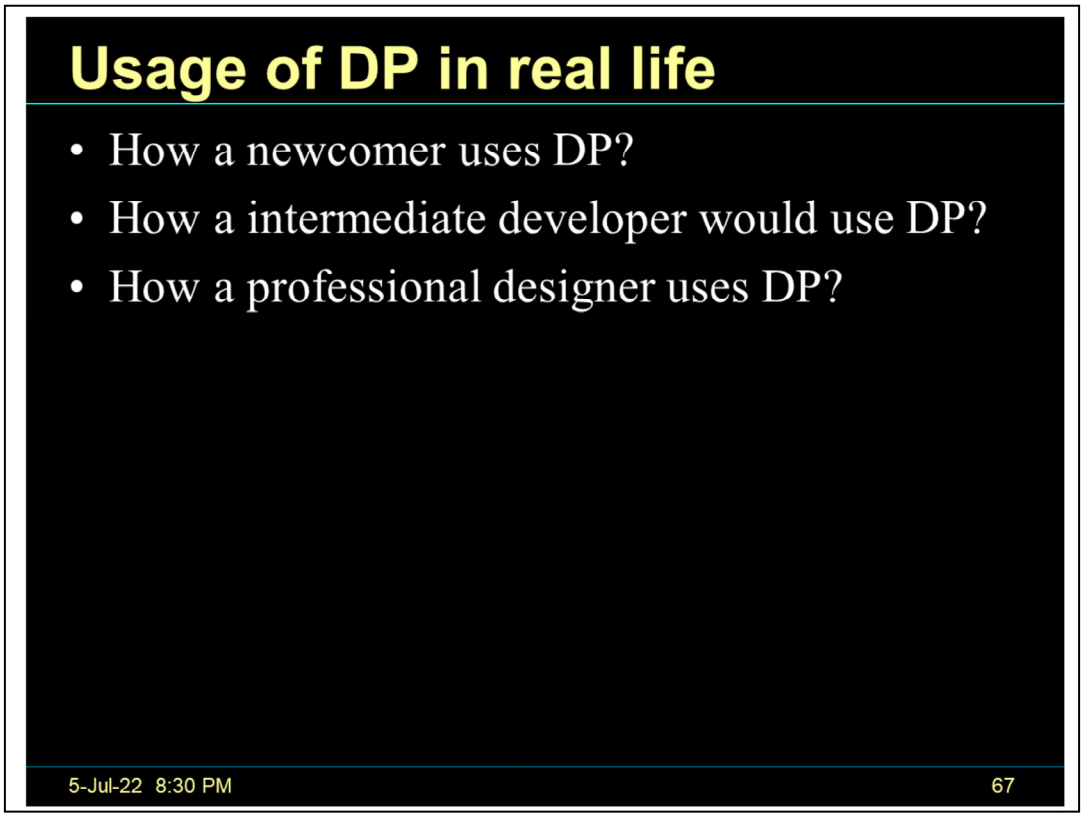
Use DPs *only if* the requirements are clear and the flexibility of DPs is needed.

If requirements are not clear,

First get the code working with the simplest solution possible.

Ensure that the code does what needs to be done.

Then refactor to use a design pattern, if required.



How a newcomer uses DP?

He tries to use it everywhere.

How a intermediate developer would use DP?

He tries to estimate where patterns are needed and where they are not?

He tries to adapt the patterns to the problem at hand.

How a professional designer uses DP?

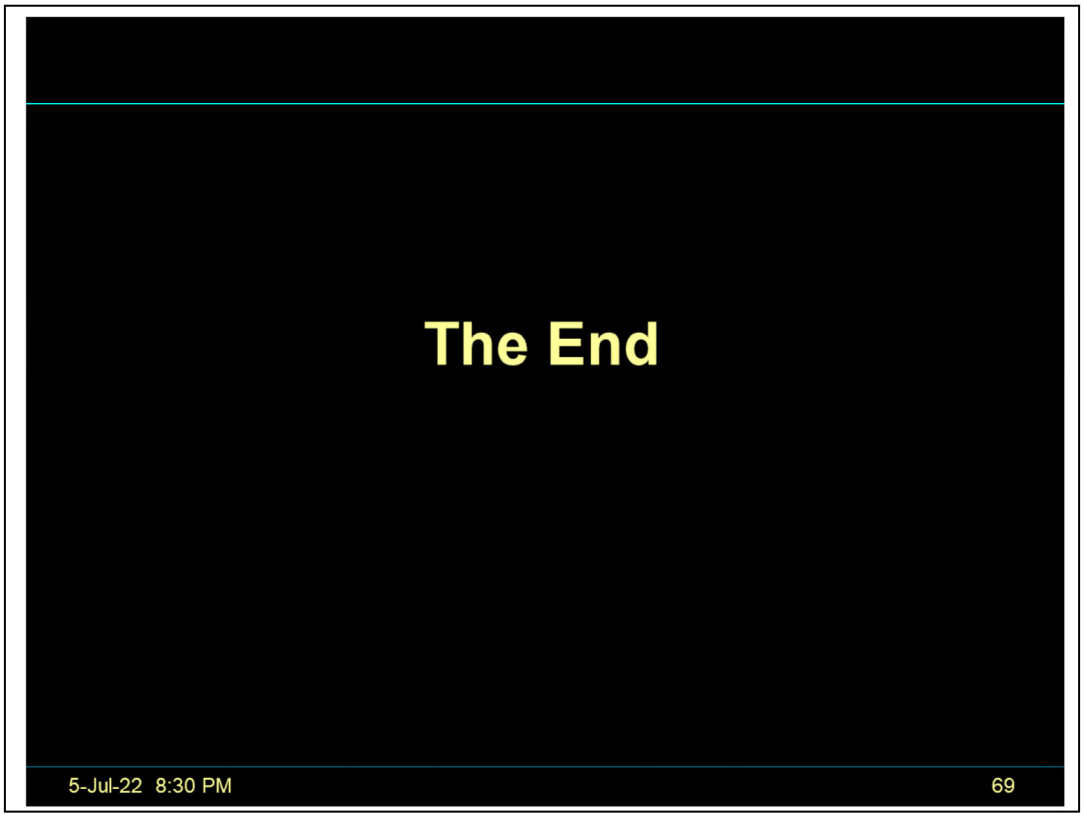
He designs the simplest solution without worrying about the list of DPs.

The design uses the necessary patterns with necessary customization.

Patterns have become ingrained and their use in the unconscious.

References

- Design Patterns: Elements of Reusable Object-Oriented Software – By Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.
- Head First Design Patterns – By Eric Freeman and Elisabeth Freeman
- Refactoring – Martin Fowler
- Refactoring to Patterns – Joshua Kerievsky



The line contains patterns slides, description of each pattern and code examples in C++, Java and C#.