Design Patterns
Not Just for Architects

Presented by Jeremy Clark
www.jeremybytes.com

What Are Design Patterns?

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to the problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

-- Christopher Alexander

@jeremybytes

The Gang Of Four (GoF)
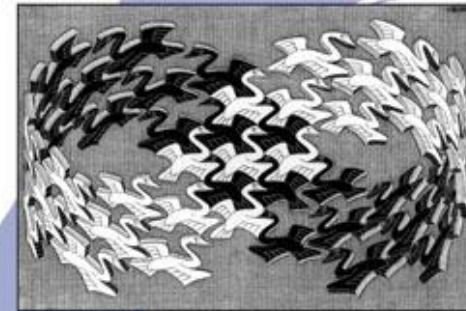
Design Patterns:
Elements of Reusable
Object-Oriented Software

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

Anatomy Of A Design Pattern

- ## Pattern Name
  - A unique way of referring to the pattern

- ## Problem
  - The problem that occurs "over and over again"

- ## Solution
  - The "core" of the solution

- ## Consequences
  - The drawbacks or considerations when using the pattern

@jeremybytes

A More Approachable Reference

- Head First Design Patterns (Second Edition)
  - Eric Freeman
  - Elisabeth Robson

Covers 12 GoF Patterns
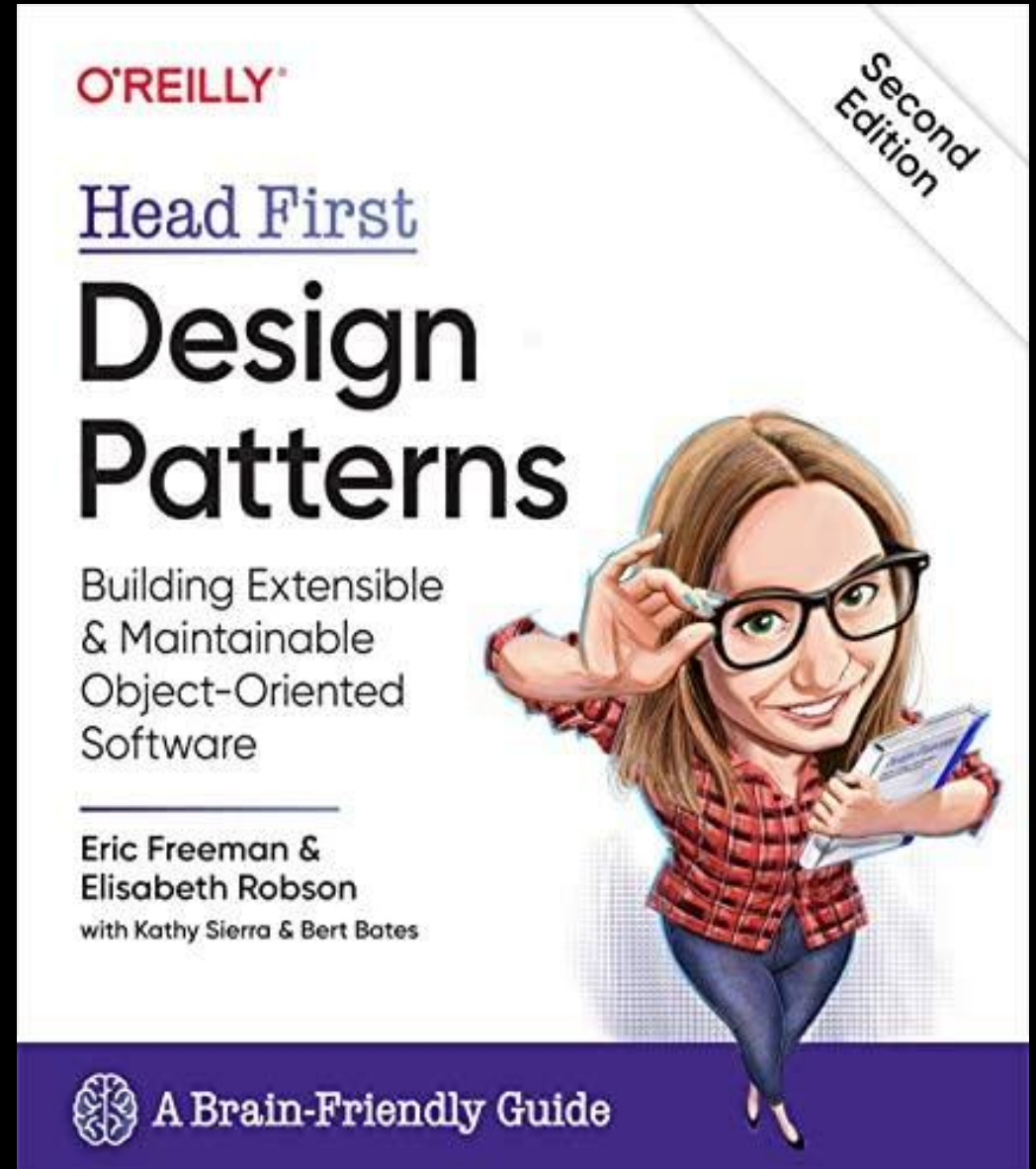
Why Should We Care?

You already use them!

- Well described solutions

- Shared vocabulary

- Concise language

- Think in design rather than implementation
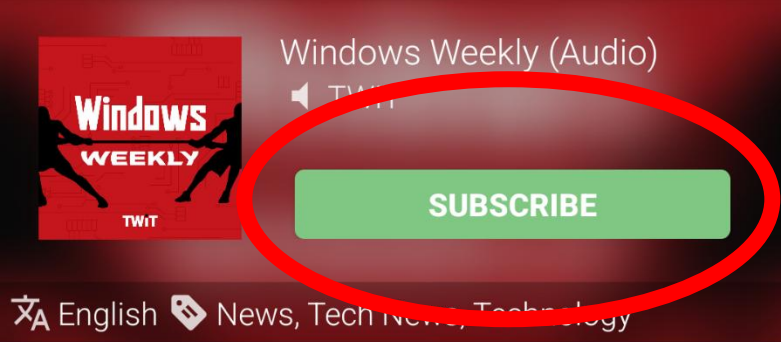
- Encourage other developers to learn patterns

@jeremybytes

Observer

GoF Description:

"Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically."

Real World Observer

- Podcasts

Windows Weekly (Audio)

SUBSCRIBE

English • News, Tech News, Technology

10 episodes (150 mins) • Weekly
26.5 K subscribers

Podcast Addict • 8m

1 new episode

Windows Weekly (Audio) • **WW 891: Scent of Carbon - CrowdStrike Outage, Windows 23H2, Halo Series**
CrowdStrike: Terrible name, terrible outage CrowdStrike, a company no one has ever heard of, ships an update that borks Windows PCs, servers, and clouds services at the heart of the world's infrastructure. Naturally, everyone blamed Microsoft at first Microsoft says outage impacted under 1 percent of PCs. It was

Download

Privacy

TWiT

s about more than Windows. insiders Paul Thurrott and oin Leo for a deep dive into company in the world. From rise, AI to Xbox, Windows Microsoft podcast you'll ever

Wednesday at 2:00pm
Pacific / 18:00 UTC.

https://feeds.twit.tv/ww.xml

Observer Implementations (in .NET)

- Event Handler
  - Built in to .NET
- Event Aggregator
  - Middle-man between observers and subject
- IObserver<T> / IObservable<T>
  - Obserable: Subscribe
  - Observer: OnNext, OnCompleted, OnError
- Many others...

@jeremybytes

Observer Consequences

- We don't know how many times we will be notified.
  It could be 1 or it could be 100,000.

- We don't know how frequently we will be notified.
  It could be once per day or 100 times per ms.

- We don't know if we will be notified at all.

Wrong Way to Use Patterns

**"Observer is awesome!
I'm going to use it as much as possible."**

- Not every tool is the right tool for the job.
- The wrong tool adds unneeded complexity.
- The wrong tool only gives you consequences (no benefits).

@jeremybytes

Right Way to Use Patterns

"I need to be notified when something has changed. Is there a pattern that addresses that?"

- Identify the problem first.
- See if there is a pattern that solves that problem.
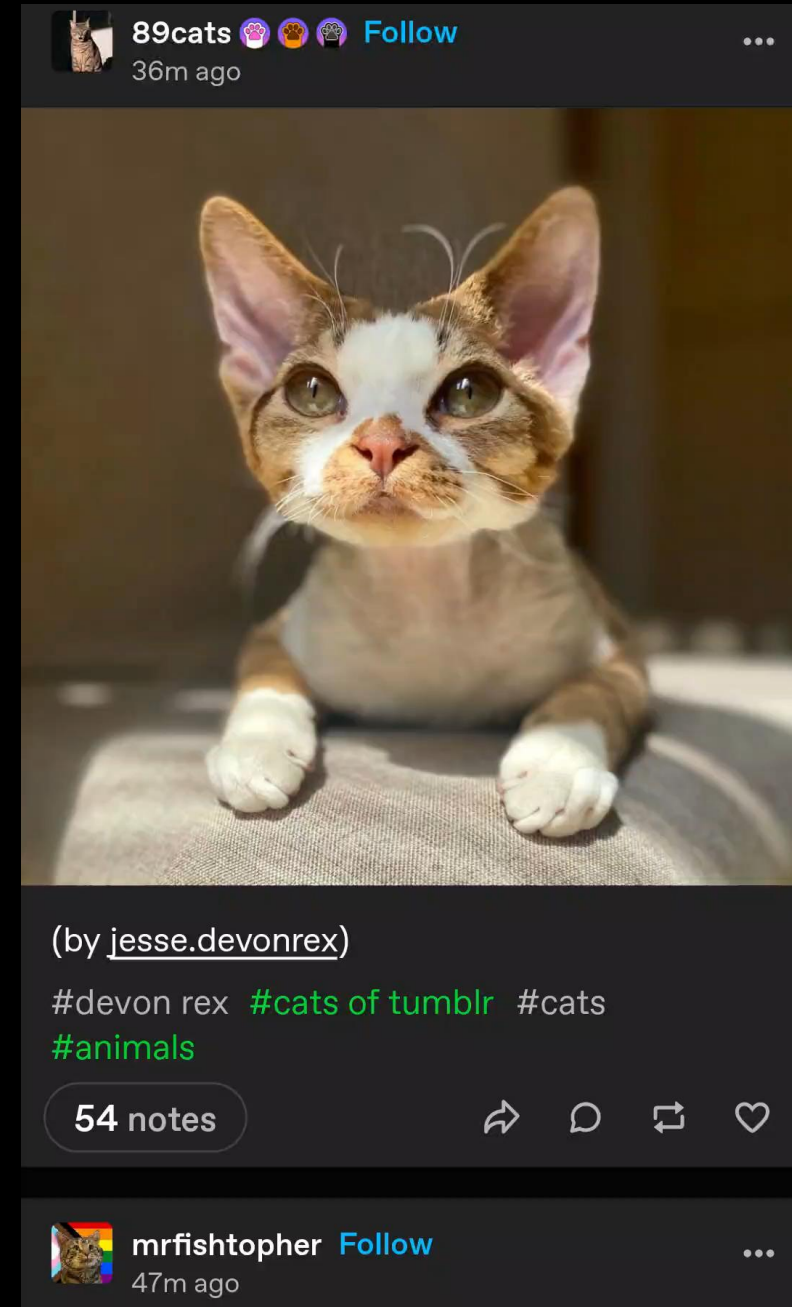- Get the benefits.

@jeremybytes

Iterator

GoF Description:

"Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation."

Real World Iterator

- Social Media

Iterator Implementations (in .NET)

- List<T>
  - List has an iterator built in.
- IEnumerable<T>
  - An interface with methods to build my own iterator.
- Many others...

Interesting Side Note:
The GoF book has an appendix with an iterator implementation

@jeremybytes

Iterator Consequences

- Adding or removing items during iteration can invalidate the iterator.

- Multiple iterators can work on a single object.

Facade

GoF Description:

"Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use."

Real World Facade

- Play Blu-ray
  - Turn on TV
  - Set TV to Blu-ray player
  - Turn on sound bar
  - Set sound bar to digital in
  - Turn on Blu-ray player
  - "Play" on Blu-ray player

@jeremybytes

Where I Used a Facade

`https://reportingServer/RunReport?id=3D2B4067-DBEE-4B3C-A5D7-DC9011300775`
`&user-id=24&param1=20250331&param2=20250406&output=pdf`

```csharp
var report = new WebReport();
report.Application = "Facilities";
report.ReportName = "WeeklyMaintenance";
report.UserName = userName;
report.Params.AddDate(2024, 03, 31);
report.Params.AddDate(2024, 04, 06);
report.Output = ReportOutput.PDF;
return report.RunReport();
```

@jeremybytes

Facade Consequences

- If functionality is added to the sub-system, we may need to change the facade as well.

- Not all functionality of the sub-system may be available. The available functions in the facade are probably limited.

Chain Of Responsibility

GoF Description:

"Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it."

Chain of Responsibility Consequences

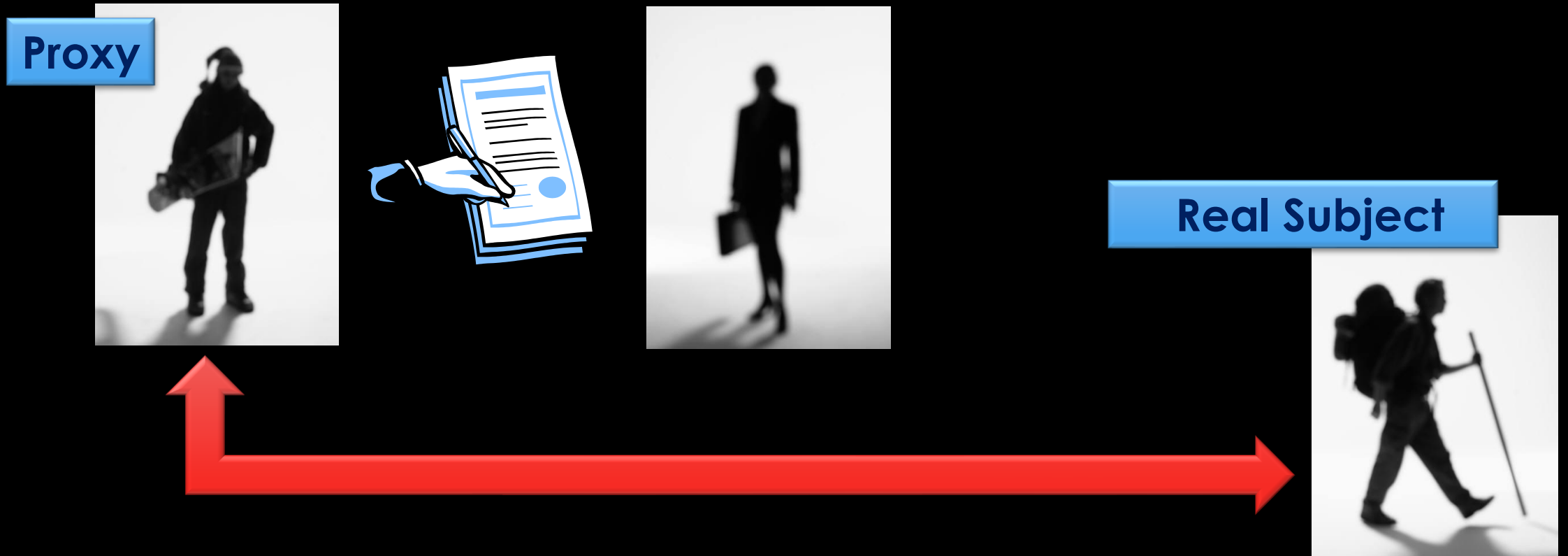- If none of the receivers handle the message, then the message will fall off the end of the chain.
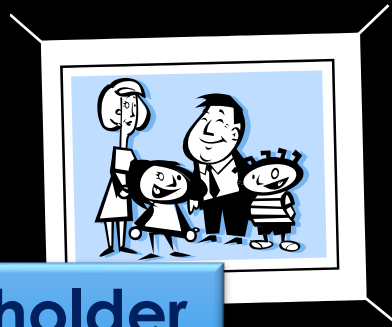
@jeremybytes

GoF Description:

"Provide a surrogate or placeholder for another object to control access to it."

Real World Proxy

- Power of Attorney

Proxy

Real Subject

Interesting History

Placeholder

Click

Original

@jeremybytes

Proxy Consequences

- A remote proxy can hide the fact that the object resides in a different process or across a network.

- The proxy must be kept synchronized with the real object.

A Million Implementations

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to the problem, in such a way that **you can use this solution a million times over, without ever doing it the same way twice.**"

-- Christopher Alexander

@jeremybytes

Why Should We Care?

You already use them!

- Well described solutions

- Shared vocabulary

- Concise language

- Think in design rather than implementation

- Encourage other developers to learn patterns

@jeremybytes

Thank You!

Jeremy Clark

- http://www.jeremybytes.com
- jeremy@jeremybytes.com
- @jeremybytes

https://github.com/jeremybytes/learning-design-patterns