

I'm Still Afraid of Aspect Oriented Programming!

Nolan Erck

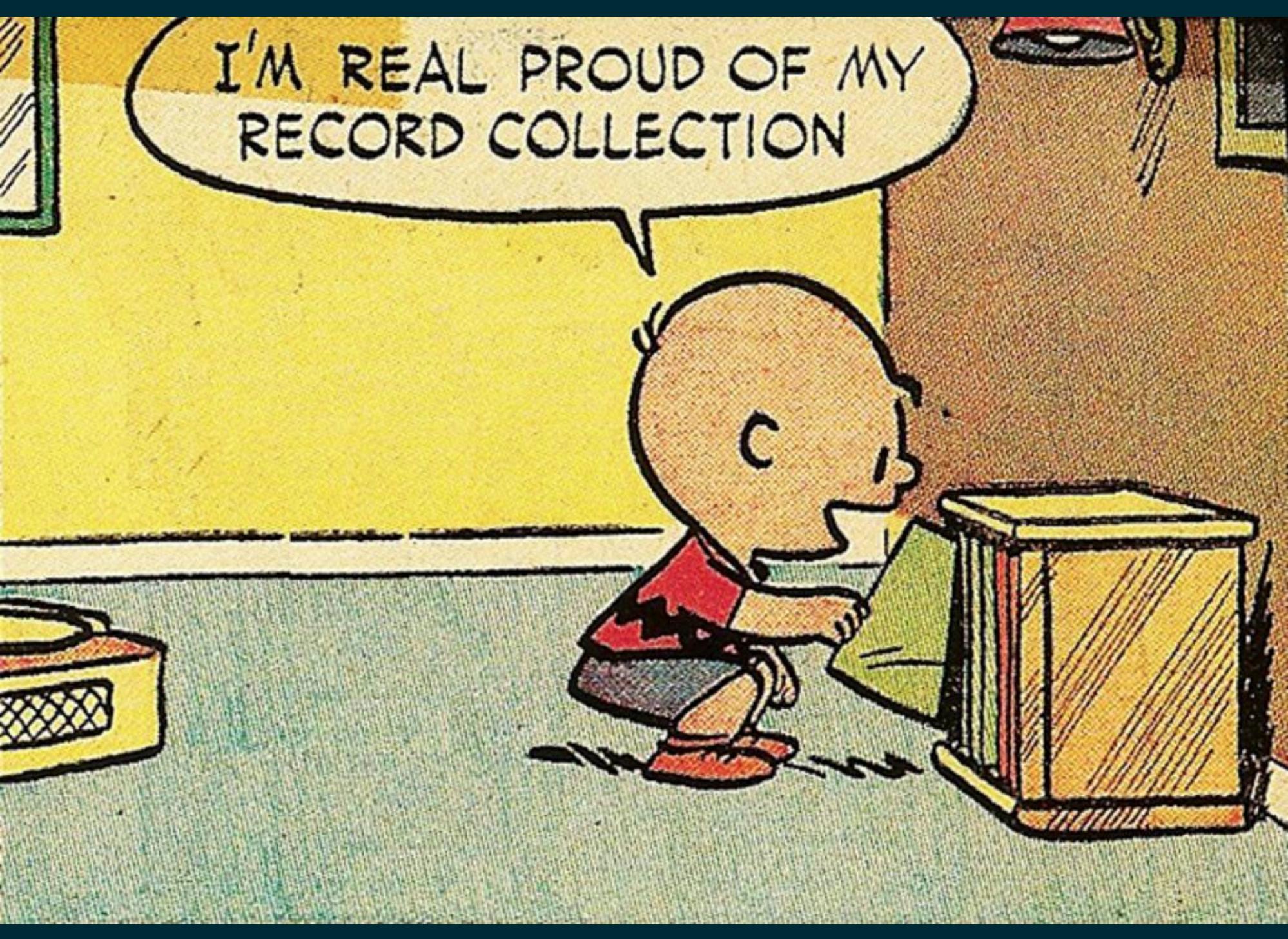
South of Shasta

About Me

- Owner / Director, South of Shasta
- VP of Engineering, e-MissionControl
- Manager, Sac Interactive Tech Meetup
- Reformed Video Game Developer - Grim Fandango, SimPark, StarWars Rogue Squadron, etc...
- Music Junkie



I'M REAL PROUD OF MY
RECORD COLLECTION



Code And Slides

github.com/nolanerck

still-afraid-of-aop

Prerequisites

- Comfortable with basic OOP Concepts
- Inheritance, Composition, Interfaces
- Other D/I concepts: Bean Injection, Inversion of Control
- Do NOT need to be using an MVC framework

Today's Agenda

- What is AOP?
- Why/when do I want to use it?
- Some code examples
- Situations when it might be handy
- Tips
- Other resources

What is AOP?

- Aspect Oriented Programming
- Design Pattern in OO development
- Not CFML or Web specific
- Concept works in any OOP language
- CFML: ColdSpring, AOP/1, WireBox, others
- Not just for logging -- run away from those demos!
- Shout-out to Mark Mandel

What Does AOP Do?

- Changes how a function runs
- aka changes *aspects* of how a function runs
- Can run code before or after your method
- Can run code *instead of* your method
- Can wrap code *around* your method
- All configurable via your AOP library of choice

For Today's Session

- Using WireBox
- It's easy to set up
- Does *not* require ColdBox
- Works with *any* CFML application
- I prefer the Wirebox.cfc syntax to other current options

WireBox

- All purpose Dependency Inection and AOP library
- Lots of features
- Many new features announced yesterday in the keynote
- You might only use 10% of WireBox's features
- And that's okay!

WireBox does not require rewriting any of your codebase!

And neither does AOP!

Installing WireBox

- Open a terminal window
- Type 'box install wirebox'
- Wait 3 seconds
- Boom, done.
- Baked into ColdBox automatically

Installing WireBox Without CommandBox

- Why don't you have CommandBox?
- Seriously, it's a standard for development now
- You will save so much time by using it for things
- It's not scary, I promise
- Ortus and South of Shasta offer training if you need it

Now you have WireBox and can start doing AOP!

[1 Hello World Demo]

Instead of My Function

- Run a function instead of another function
- No matter how many places it's called in the app
- Toggle on/off in 1 configurable place

[2 Instead Of Demo]

Why?

- Easily swap out one function library for another
- Minimize 3rd party vendor lock-in with your code?
- Got a problematic vendor partner? Or still evaluating options?
- Use AOP to swap the libraries as needed, easily!

Before And After

- AOP can run code before and/or after your function automatically
- No matter how many places it's called in the app
- Toggle on/off in 1 configurable place

[3 Before And After Demo]

[4 Before And After w/ Functions Demo]

Why?

- Easily add security and logging and whatever we want
- Ensure that *all* the security and logging works the same in the entire app
- Can change it from 1 place
- No need to tell the dev team whenever a security fix is applied
- Their code *automatically* gets the new security rule applied - zero code changes!

Wrap Code Around Code

- Not just function calls before and after
- Wrap constructs around your code
- Try/Catch blocks

Wrap Code Around Code

Say I have a query like so:

```
<cfquery name="qryGetUsers">
    SELECT * FROM Users
</cfquery>
```

In Production, I want that wrapped in a try/catch, but in QA/Development, I don't. (So I can see the debug output, etc.)

Wrap Code Around Code

I'd have to set some kind of "flag":

```
<cfif isProduction>
    <cftry>
        <cfquery name="qryGetUsers">
            <SELECT * FROM Users>
        </cfquery>
    <cfcatch>
        <cflog ... />
    </cfcatch>
    </cftry>
<cfelse>
    <cfquery name="qryGetUsers">
        <SELECT * FROM Users>
    </cfquery>
</cfif>
```

Duplicate Code!

Remember the DRY rule!

[5 Try/Catch Demo]

[Real World ColdBox App w/ Less Boilerplate]

Recap

- AOP is good!
- Only needs a little code to activate.
- Able to combine with other patterns for even more code reuse!
- Takes a little discipline, but it's worth it

Some Useful Tips

- Don't feel like you have to jump in head first
- Treat it like a tool in a tool box
- Sprinkle AOP (and DI) into your app as needed / comfortable
- AOP is very worth learning
(it's not scary, I promise)

Other Resources

- "Modern ColdFusion (CFML) in 100 Minutes" - Luis Majano
- "Object-Oriented Programming in ColdFusion" - Matt Gifford
- "Head First Design Patterns" - Eric Freeman & Elisabeth Robson
- "Head First Object Oriented Analysis and Design" - Brett D. McLaughlin
- Anything Martin Fowler wrote
- "Design Patterns: Elements Of Reusable Object-Oriented Software" (Gang of Four)
- southofshasta.com/training and /presentations
- CFCasts.com videos and Ortus podcasts

Learning in 30 minutes a day.

Questions? Comments? Need consulting help?

- southofshasta.com
- nolan@southofshasta.com
- Twitter: @southofshasta @nolanerck
- Github: nolanerck

Thanks!