

OOP: What is it and why do I care?

Dan Skaggs, Co-Founder
Web-Meister Designs

<http://TNTechnoHermit.com>



Why you're here

- Hopefully because you are curious as to what this whole “OO thing” is about
- Hopefully because you want to learn about **ONE** way of thinking about architecting and building applications
- There is no one true “right” way to build applications.
- The “right” architecture is the one that **fits your specific problem domain**
- Hopefully **not** expecting to learn the “magic bullet” for building every application that you could ever dream of
- Hopefully **not** expecting to know everything there is to know about object-oriented programming when you leave here today
- Session is designed to “whet your appetite”



Obligatory “About Me” slide

- Began experimenting with HTML in 1997 after the birth of our twins while on active duty with the U.S. Air Force
- Staff developer for a company headquartered in San Jose, CA for 8 1/2 years
- Began with Allaire ColdFusion 4.5 in 2000 as part of a Spectra 1.0 project
- Have developed on every Allaire/Macromedia/Adobe ColdFusion version since then
- Have done a few projects with Railo and have experimented with Open BlueDragon
- Independent consultant since 2006 working with clients of all sizes across the U.S. building applications on multiple CFML engines and versions
- Also enjoy shooting, amateur (HAM) radio and video games

What is OOP?



Long, boring definition

[1] Object-oriented programming (OOP) is a programming paradigm using "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs.

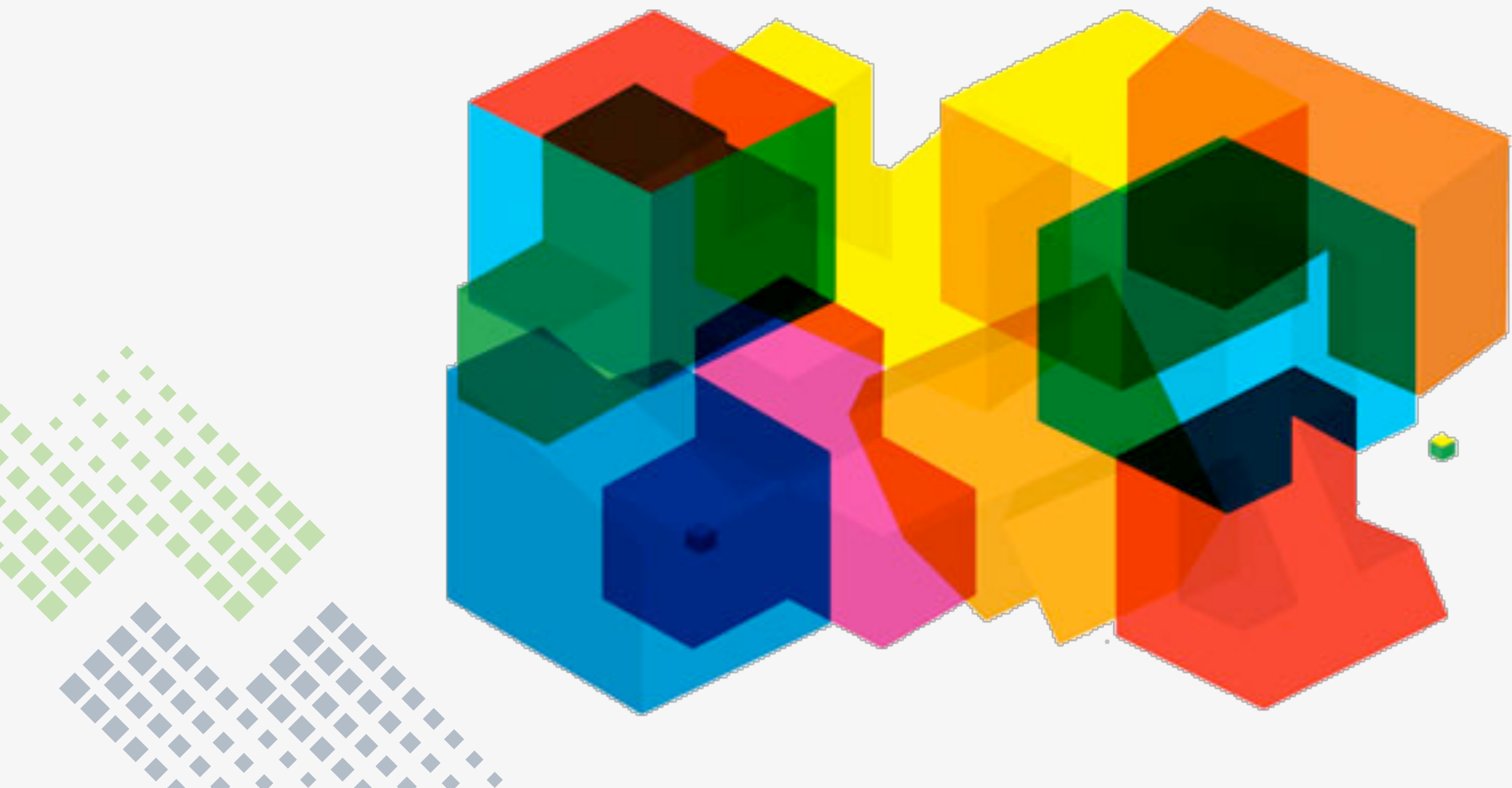
ZZZZZZZZZZZZ...

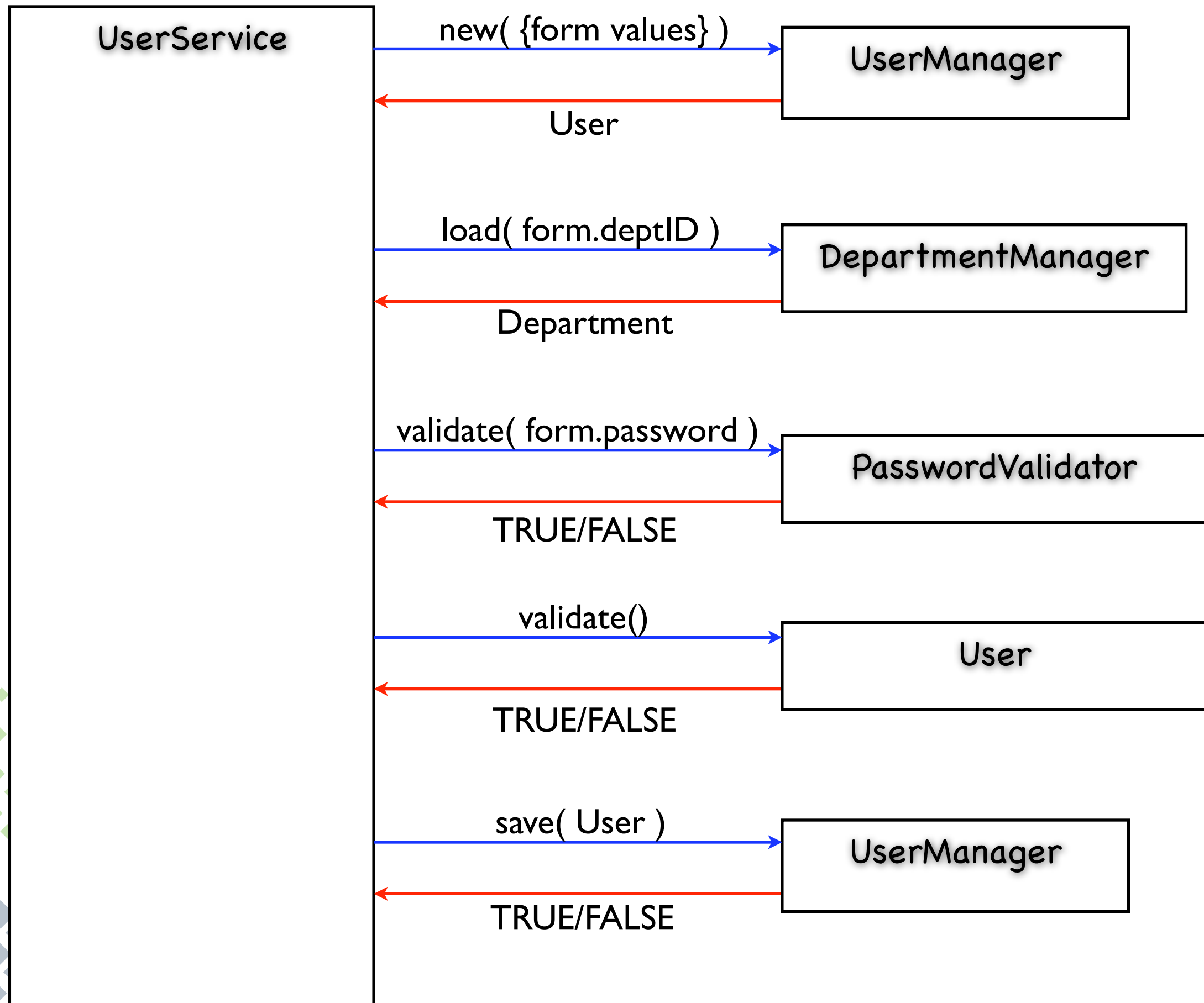
Why do I care?



Why do I care?

Modularity





Why do I care?

Maintainability



searchResults.cfm

```
73
74 <cfif searchResults.quantityAvailable GT 0
75     AND searchResults.currentPrice LT searchResults.basePrice>
76     <img source="/images/icons/onsale.png" />
77 </cfif>
78
--
```

productDetails.cfm

```
73
74 <cfif searchResults.quantityAvailable GT 0
75     AND searchResults.currentPrice LT searchResults.basePrice>
76     <img source="/images/icons/onsale.png" />
77 </cfif>
78
--
```

shoppingCart.cfm

```
73
74 <cfif searchResults.quantityAvailable GT 0
75     AND searchResults.currentPrice LT searchResults.basePrice>
76     <img source="/images/icons/onsale.png" />
77 </cfif>
78
--
```

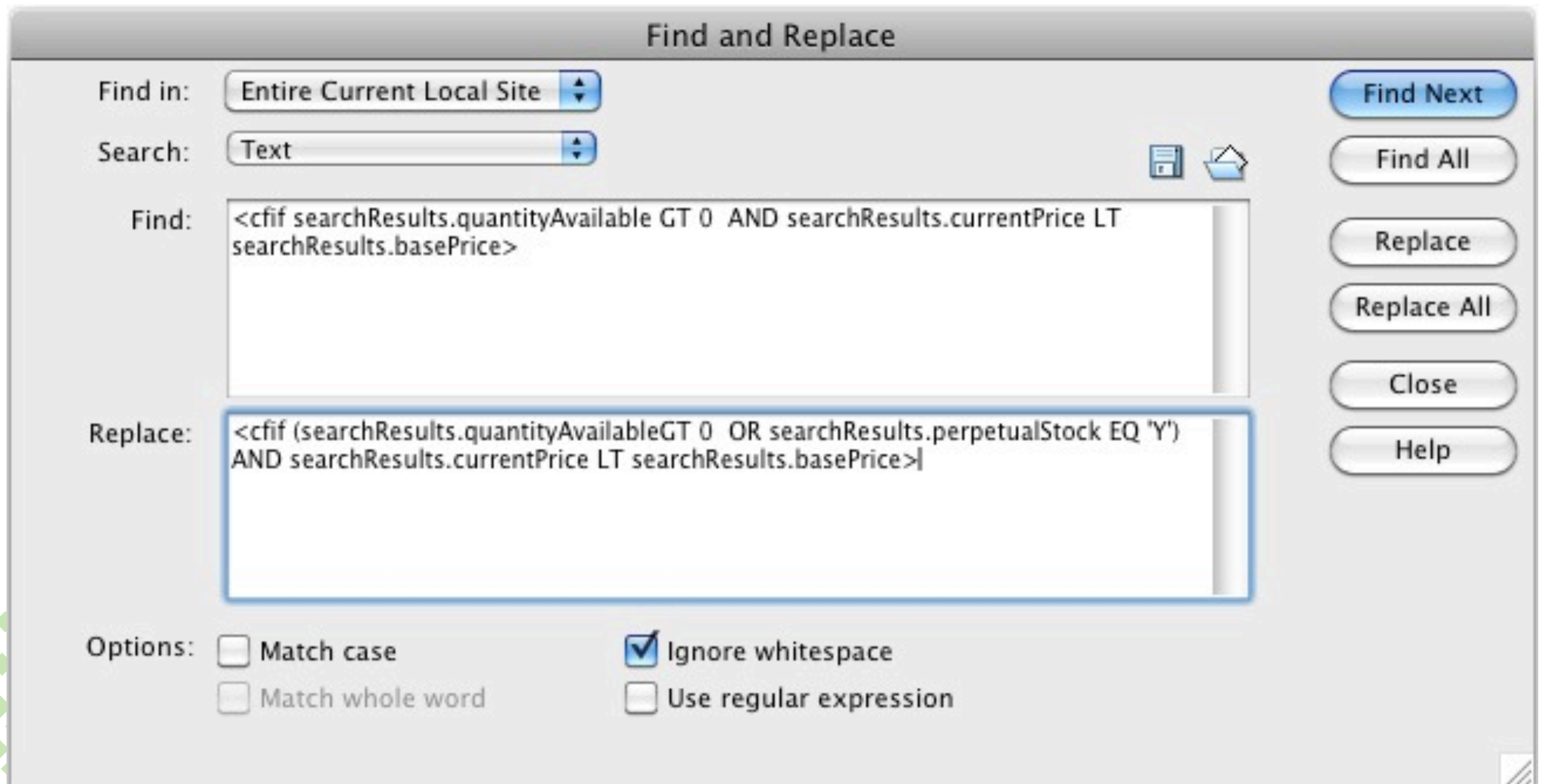
But then....



New logic

```
73 |  
74 <cfif (searchResults.quantityAvailable GT 0 OR searchResults.perpetualStock EQ 'Y')  
75     AND searchResults.currentPrice LT searchResults.basePrice>  
76     <img source="/images/icons/onsale.png" />  
77 </cfif>  
78
```

Not maintainable!



A screenshot of a 'Find and Replace' dialog box. The 'Find in:' dropdown is set to 'Entire Current Local Site'. The 'Search:' dropdown is set to 'Text'. The 'Find:' text area contains the code: `<cfif searchResults.quantityAvailable GT 0 AND searchResults.currentPrice LT searchResults.basePrice>`. The 'Replace:' text area contains the code: `<cfif (searchResults.quantityAvailableGT 0 OR searchResults.perpetualStock EQ 'Y') AND searchResults.currentPrice LT searchResults.basePrice>|`. The 'Options:' section has four checkboxes: 'Match case' (unchecked), 'Match whole word' (unchecked), 'Ignore whitespace' (checked), and 'Use regular expression' (unchecked). On the right side, there are six buttons: 'Find Next' (highlighted in blue), 'Find All', 'Replace', 'Replace All', 'Close', and 'Help'. The dialog box has a title bar that says 'Find and Replace'.

Find and Replace

Find in: Entire Current Local Site

Search: Text

Find: `<cfif searchResults.quantityAvailable GT 0 AND searchResults.currentPrice LT searchResults.basePrice>`

Replace: `<cfif (searchResults.quantityAvailableGT 0 OR searchResults.perpetualStock EQ 'Y') AND searchResults.currentPrice LT searchResults.basePrice>|`

Options:

- ☐ Match case
- ☒ Ignore whitespace
- ☐ Match whole word
- ☐ Use regular expression

Find Next

Find All

Replace

Replace All

Close

Help

An OO approach

Product.cfc

```
1 <cfcomponent displayname="Product" output="false" accessors="true"
2   hint="I represent a single product from our catalog" >
3
4   <cfproperty name="id" />
5   <cfproperty name="description" />
6   <cfproperty name="quantityAvailable" />
7   <cfproperty name="basePrice" />
8   <cfproperty name="currentPrice" />
9   <cfproperty name="isPerpetualStock" />
10
11 <cffunction name="init" access="public" output="false" returntype="model.Product">
12   <cfreturn this />
13 </cffunction>
14
15 <cffunction name="isOnSale" output="false" access="public" returntype="boolean">
16
17   <cfif ( getQuantityAvailable() GT 0 OR getIsPerpetualStock IS TRUE )
18     AND getCurrentPrice() LT getBasePrice() >
19     <cfreturn TRUE />
20   <cfelse>
21     <cfreturn FALSE />
22   </cfif>
23
24 </cffunction>
25
26 </cfcomponent>
```

An OO approach

searchResults.cfm

```
73  
74 <cfif Product.isOnSale() IS TRUE >  
75     <img source="/images/icons/onsale.png" />  
76 </cfif>  
77
```

Why Do I Care?

Testability



Why Do I Care?

Trade-Offs

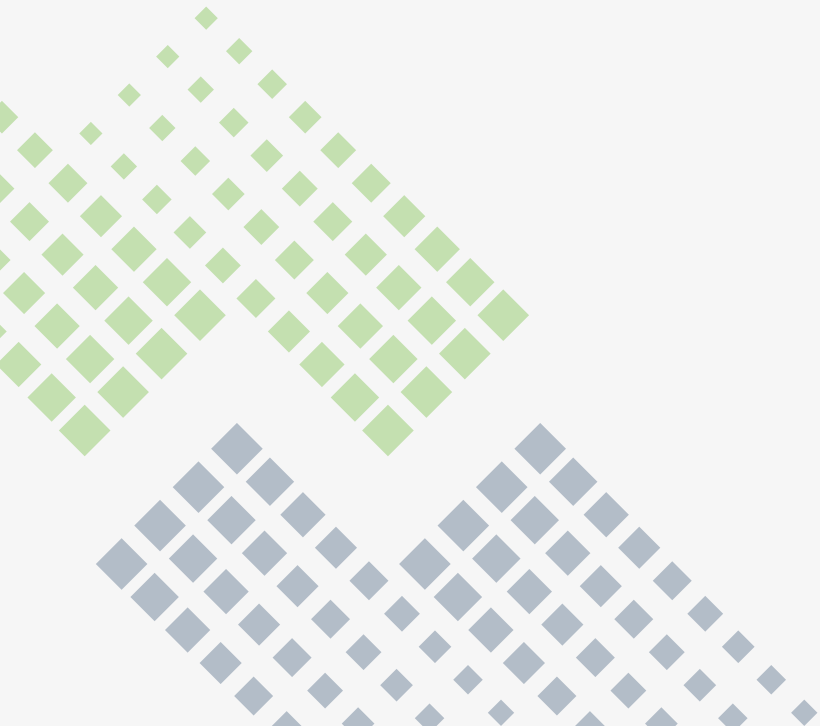


How do I get started?



How do I get started?

- Realize there is no “Right” Way - Depends on your specific problem to be solved
- Read, Study and Ask - Lots of good info on architecting applications using OOP, common design patterns, and pitfalls/gotchas
- Ease In (Toe-In-The-Water) - New feature development, proof-of-concept
- Dive In - New application development, existing application rewrite



Final Thoughts



Questions?

Dan Skaggs, Co-Founder
Web-Meister Designs

Twitter: dskaggs

Email/GTalk: dan.skaggs@web-meister.com

Blog: <http://TNTechnoHermit.com>

