

Naming Matters

Cory House

Twitter: @housecor

BitNative.com



Agenda

- **Why naming is critical**
- **Class names to avoid**
- **Avoiding nasty unexpected function side-effects**
- **Principles for selecting good variable names**

Naming Matters

Dirty

```
List p = new List() { 5.50m, 10.48m, 12.69m };
decimal t = 0;
foreach (var i in p)
{
    t += i;
}

return t;
```



Could you read this book?

P was very angry with G for insulting her
M. G kicked P in the A. He slept on the C.

Clean

```
List prices = new List() { 5.50m, 10.48m, 12.69m };
decimal total = 0;
foreach (var price in prices)
{
    total += price;
}

return total;
```

Naming Classes

Dirty

- WebsiteBO
- Utility
- Common
- MyFunctions
- JimmysObjects
- *Manager / *Processor/*Info



Guidelines:

1. Noun
2. Be specific
3. Single Responsibility
4. Avoid generic suffixes

Clean

- User
- Account
- QueryBuilder
- ProductRepository

Specific names lead to smaller more cohesive classes

The Method Name Should Say It All

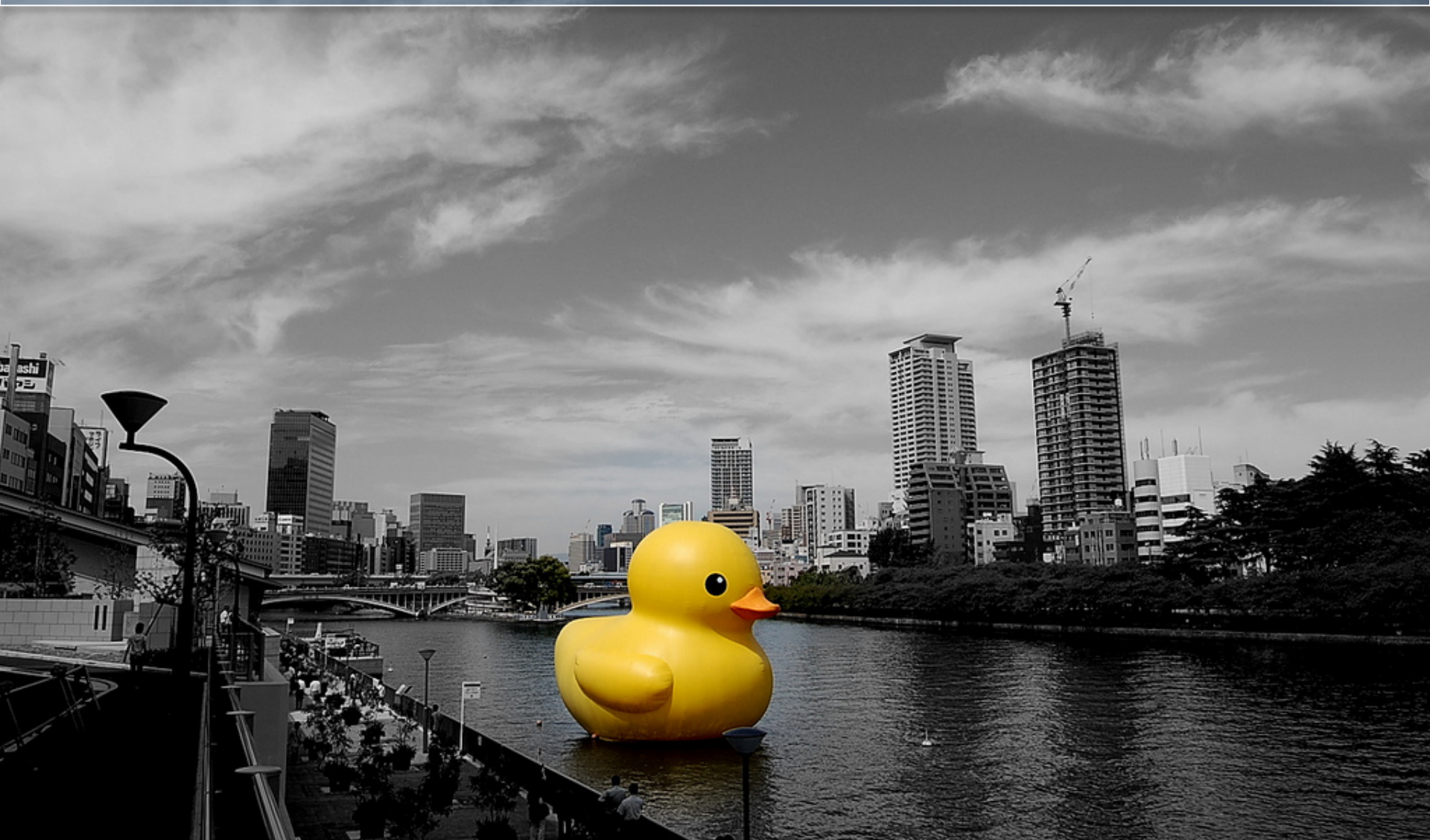
Say what?

- Get
- Process
- Pending
- Dolt
- Start
- On_Init, Page_Load, etc.

Right on.

- GetRegisteredUsers
- IsValidSubmission
- ImportDocument
- SendEmail

Need a hand?



Naming Warning Sign

- **Watch out for**

- And
- If
- Or



Watch for Side Effects

- CheckPassword shouldn't log users out.
- ValidateSubmission shouldn't save.
- GetUser shouldn't create their session.
- ChargeCreditCard shouldn't send emails.

Solution?

Refactor until the method name completely describes what it does.

Avd Abbr

- It's not the 80's
- No standard
- We talk about code

RegUsr

RegistUser

RegisUser

RegisterUsr



Naming variables: Booleans

- Boolean names should sound like true/false questions

Dirty

- open
- start
- status
- login

Clean

- isOpen
- done
- isActive
- loggedIn

```
if (login)
{
}

```

```
if (loggedIn)
{
}

```

Naming variables: Be symmetrical

- When dealing with states that toggle, consistently use matching pairs

Dirty

- on/disable
- quick/slow
- lock/open
- slow/max

Clean

- on/off
- fast/slow
- lock/unlock
- min/max

Summary

- Strive for specific class names
- The name should say it all
- Watch for nasty function side-effects
- Booleans should sound “truthy” and “falsy”
- When struggling, verbalize to a friend....or a duck.