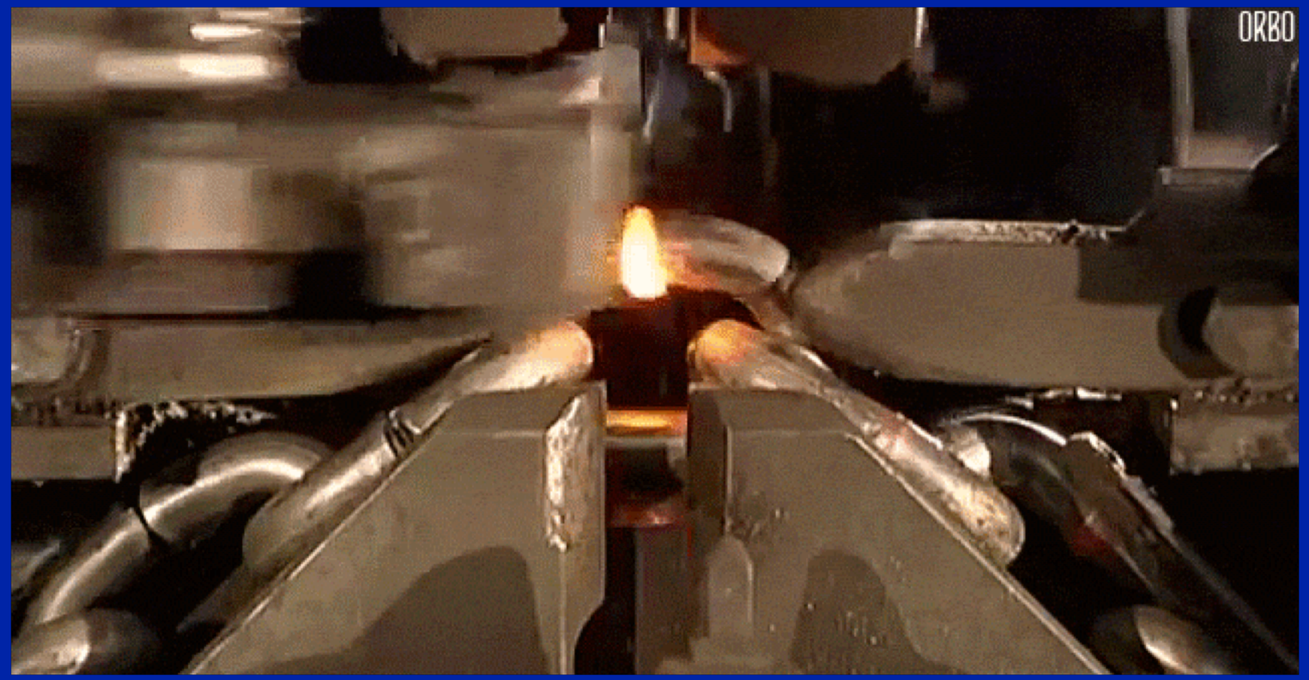
# IOS ANTI-PATTERNS

## Learning by repeated failure

Dave Weston
dweston@microsoft.com
http://github.com/dtweston

Hello
my name is


ORBO

# MOBILE APPS ARE EVERYWHERE

# APPS ARE MORE COMPLEX

# APPS STAND ALONE

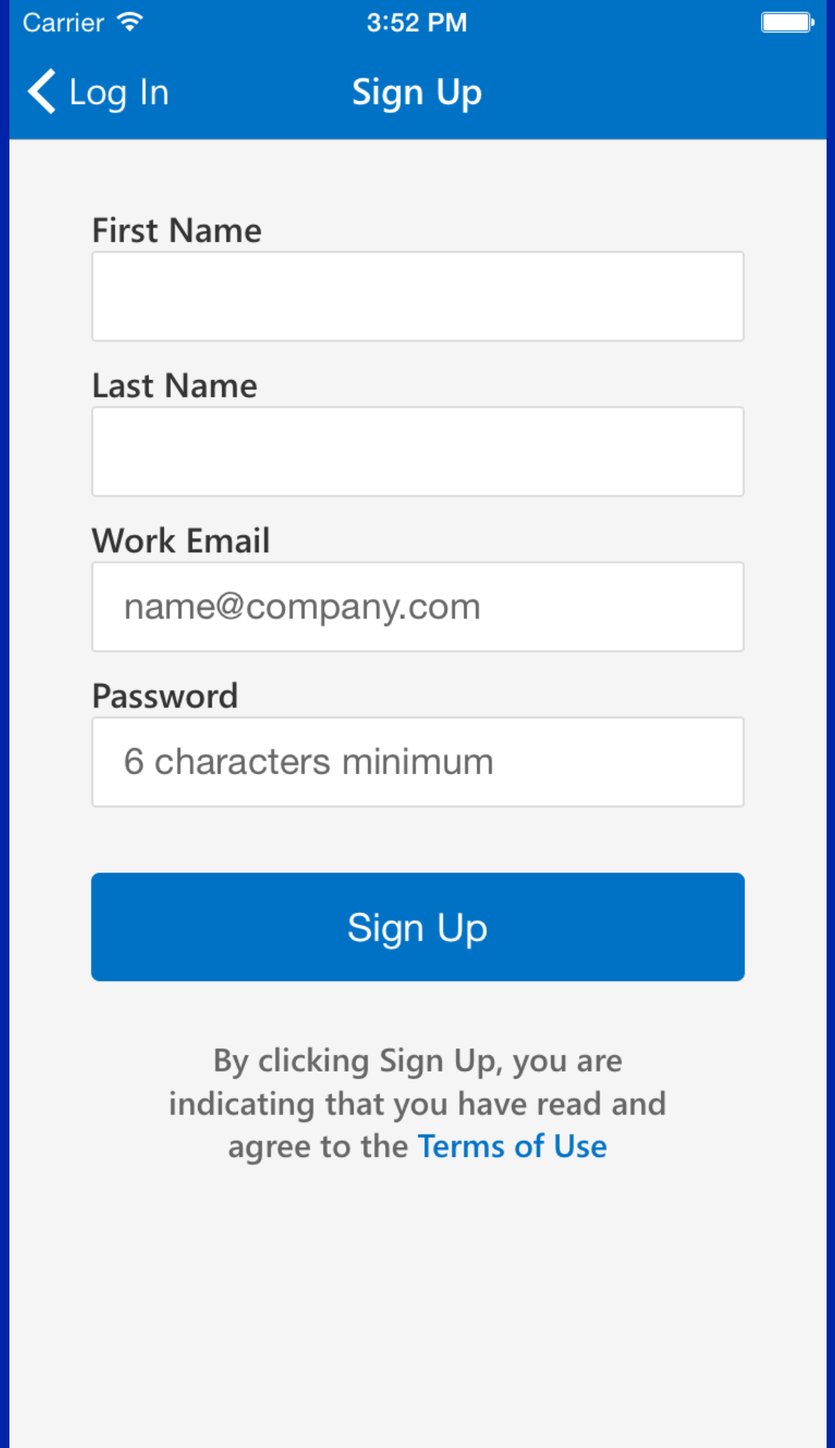
# FAILURES

(and successes)

# HUGE VIEW CONTROLLERS

# VIEW CONTROLLERS ARE GLUE

```
@implementation YMFeedViewController

#pragma mark - Header view
#pragma mark - fetching helper methods
#pragma mark - View lifecycle
#pragma mark - private methods
#pragma mark - Table view delegate
#pragma mark - Table view data source
#pragma mark - UIScrollViewDelegate
#pragma mark - YMFeedCellDelegate
#pragma mark - YMCardCellDelegate
#pragma mark - YMScrollableImageAttachmentViewDelegate
#pragma mark - YMThreadViewControllerDelegate
#pragma mark - iPad group transitions
#pragma mark - YMFetchedResultsTableControllerDelegate

@end
```



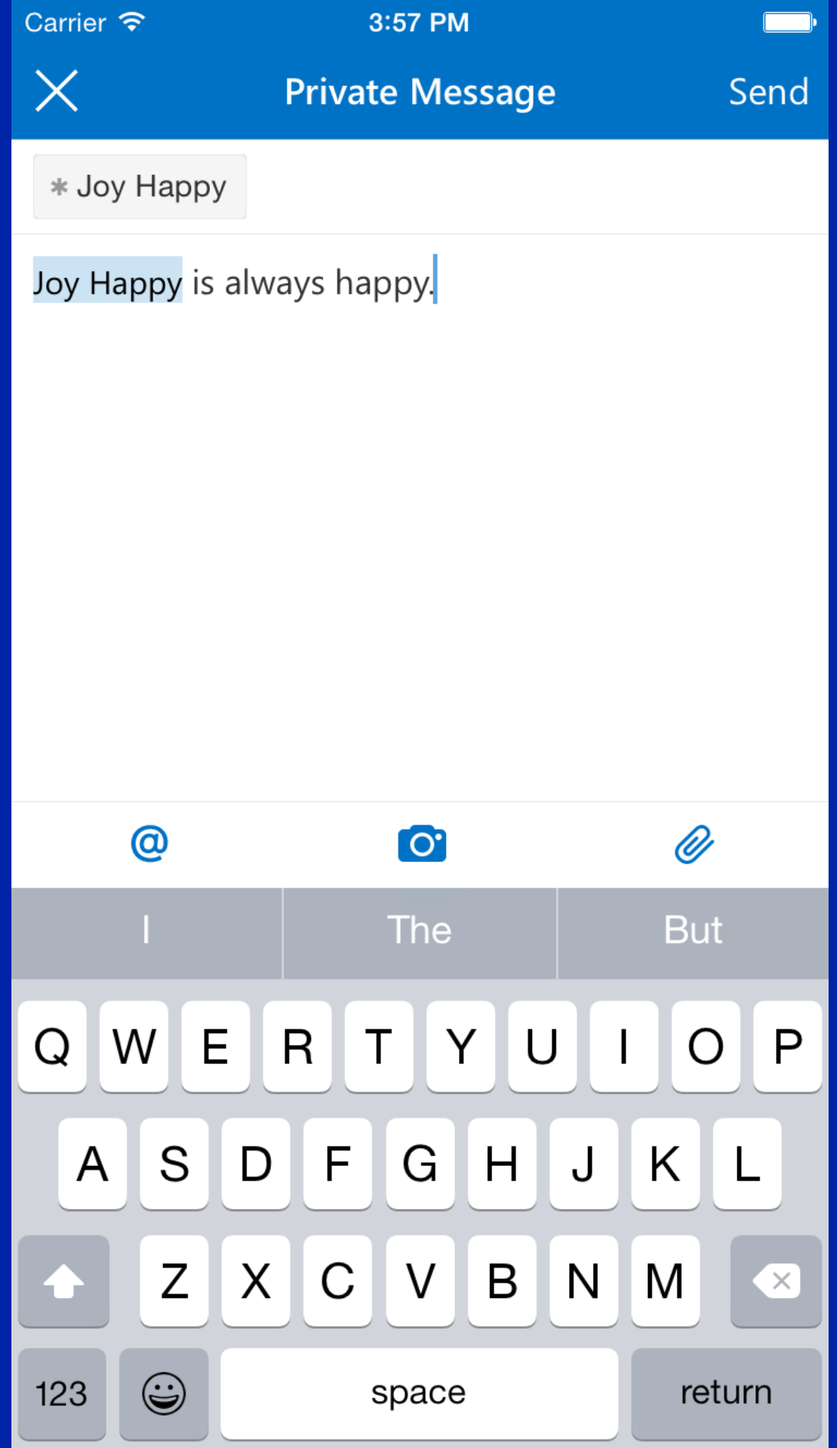

# MOVE VIEW CODE OUT OF THE VIEW CONTROLLER

## SignupViewController => SignupViewController + SignupView

# SPLITTING RESPONSIBILITIES IS GOOD

```
@interface YKSignupView : UIScrollView

@property (nonatomic, strong) UILabel *firstNameLabel;
@property (nonatomic, strong) UILabel *lastNameLabel;
@property (nonatomic, strong) UILabel *emailLabel;
@property (nonatomic, strong) UILabel *passwordLabel;
@property (nonatomic, strong) YKErrorLabel *firstNameError;
@property (nonatomic, strong) YKErrorLabel *lastNameError;
@property (nonatomic, strong) YKErrorLabel *emailError;
@property (nonatomic, strong) YKErrorLabel *passwordError;
@property (nonatomic, strong) YKSimpleTextField *firstNameField;
@property (nonatomic, strong) YKSimpleTextField *lastNameField;
@property (nonatomic, strong) YKSimpleTextField *emailField;
@property (nonatomic, strong) YKSimpleTextField *passwordField;
@property (nonatomic, strong) YKSimplePrimaryButton *signupButton;
@property (nonatomic, strong) UILabel *termsOfUseLabel;
@property (nonatomic, strong) UIButton *termsOfUseButton;
@property (nonatomic, strong) YMKeyboardLayoutHelperView *keyboardHelper;

- (void)centerOnView:(UIView *)view;

@end
```

# ... DO I DETECT A HOWEVER?




# AND NOW THE COMPOSER

- Address bar at the top
  - Text view
- Tool bar at bottom

```objc
- (BOOL)isMessageBodyEmpty
{
    return [self.composeTextView isEmpty];
}

- (NSString *)messageBodyText
{
    return [self.composeTextView originalText];
}

- (void)setMessageBodyText:(NSString *)text
{
    self.composeTextView.text = text;
}

- (NSAttributedString *)messageBodyAttributedText
{
    return self.composeTextView.attributedText;
}

- (void)setMessageBodyAttributedText:(NSAttributedString *)attributedText
{
    self.composeTextView.attributedText = attributedText;
}
```

```
- (void)mentionableTextView:(YMMentionableTextView *)mentionableTextView didAddMentionedUser:(YDUser *)user
{
    [self.composeViewDelegate composeView:self didAddMentionedUser:user];
}

- (void)mentionableTextView:(YMMentionableTextView *)mentionableTextView didRemoveMentionedUser:(YDUser *)user
{
    [self.composeViewDelegate composeView:self didRemoveMentionedUser:user];
}

- (void)mentionableTextView:(YMMentionableTextView *)mentionableTextView didPasteImageData:(NSData *)data{
    [self.composeViewDelegate composeView:self didPasteImageData:data];
}
```

# OTHER SOLUTIONS?

# SEPARATE DATA SOURCES FROM VIEW CONTROLLERS

# PARENT/CHILD VIEW CONTROLLER RELATIONSHIPS

# VIEWS THAT UNDERSTAND THE MODEL LAYER

WE'RE DOOMED
WE'RE ALL DOOOOOOOMED

# NSObject CONTROLLERS

# TESTS

# PLAN 1
## KIF (KEEP IT FUNCTIONAL)

# PLAN 2

## KIWI (BDD TESTING)

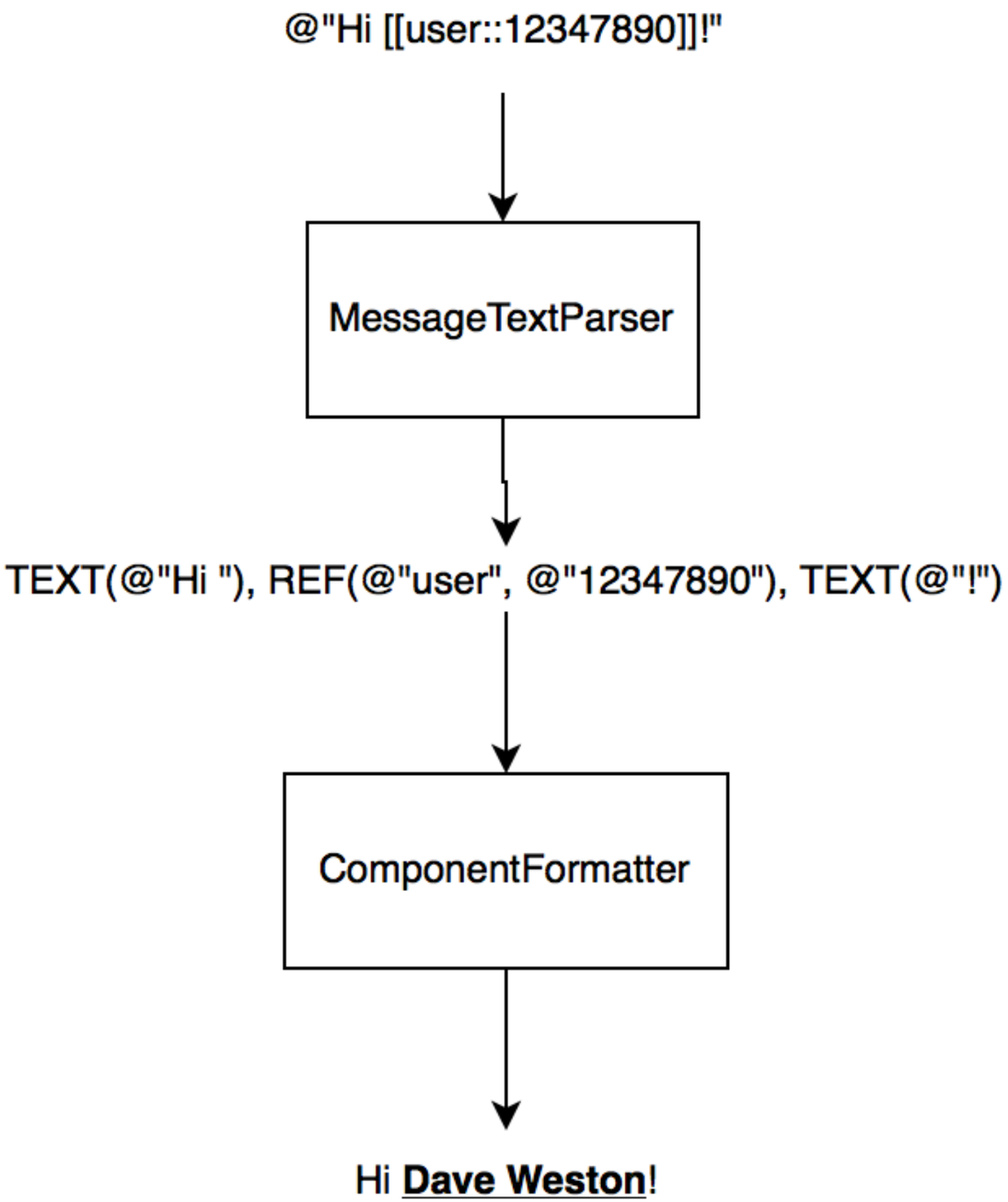
```
describe(@"Math", ^{
  context(@"Addition", ^{
    it(@"is commutative", ^{
      [[theValue(3+2) should] equal:theValue(2+3)];
    });
  });
});
```

# MORE/BETTER TESTING

## HOW TO IMPROVE TEST COVERAGE?

# TWO OPTIONS

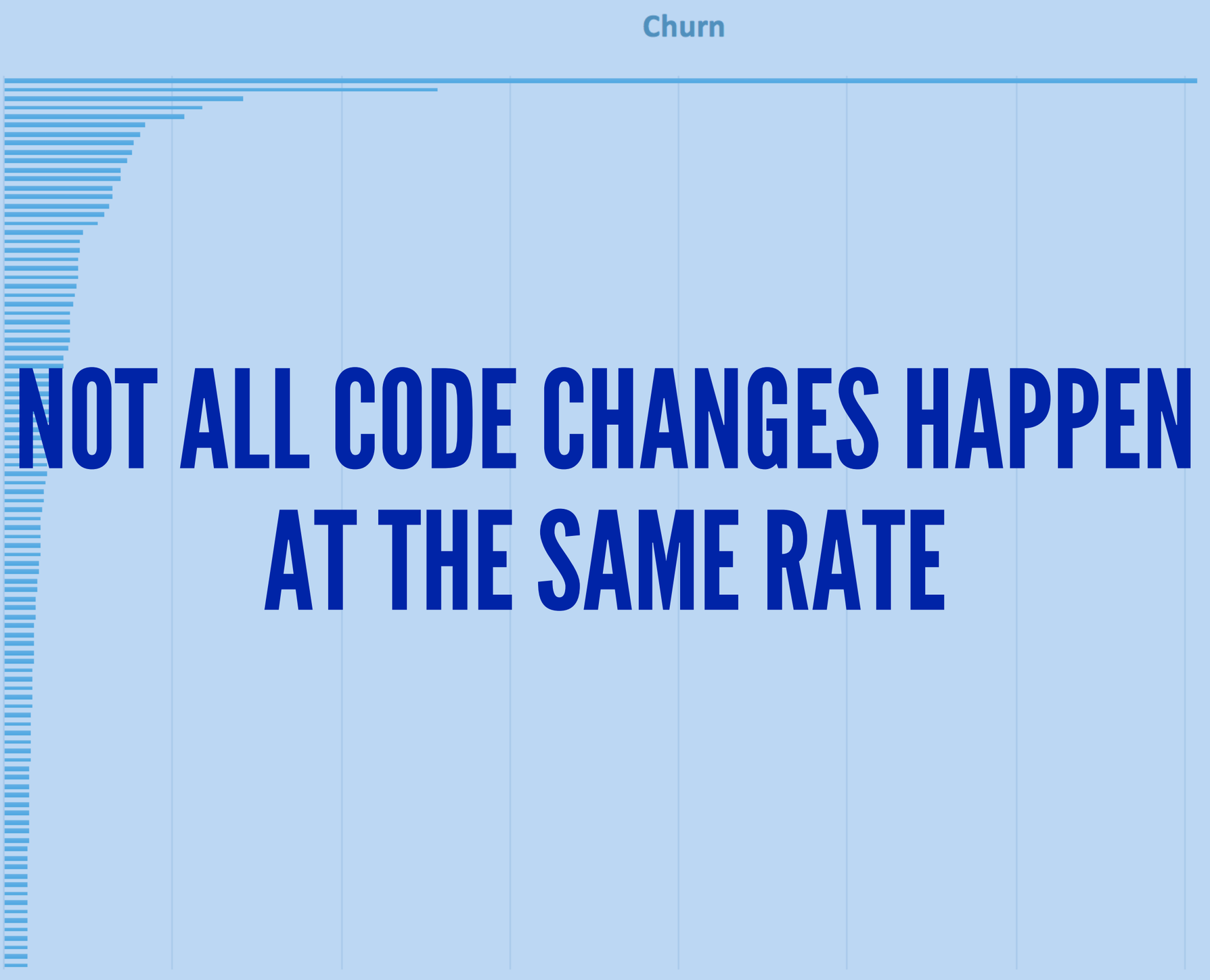
- bigger tests, more setup
- smaller units to be tested




# MESSAGE FORMATTING

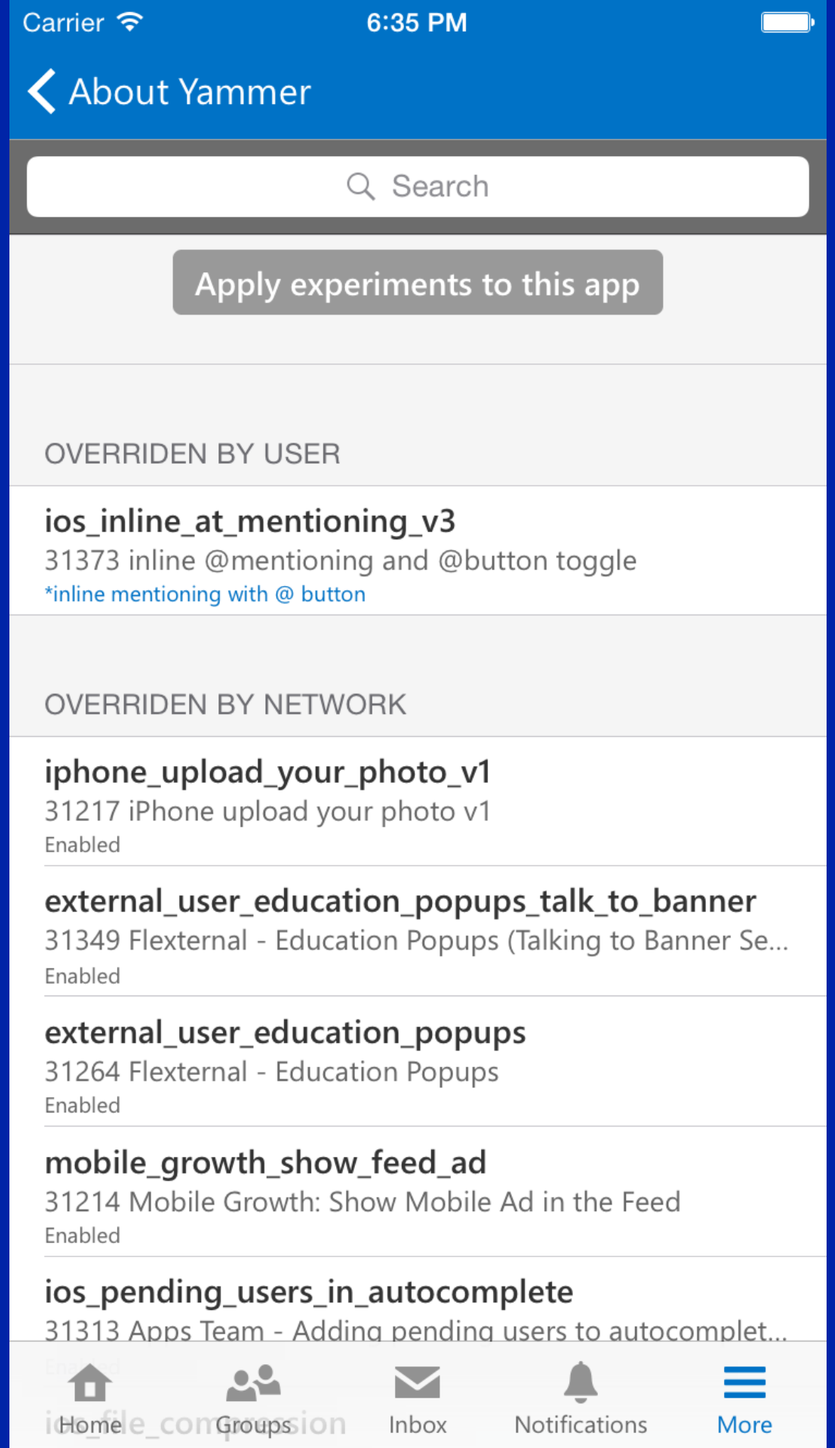
# MENTION DETECTION

```
describe(@"YMMentionDetector", ^{

    it(@"detects mentions due to capitalization", ^{
      ...
    });

    it(@"detects mentions due to @ symbol", ^{
      ...
    });

    it(@"detects mentions when the cursor is moved back", ^{
      ...
    });

    it(@"detects symbol mentions when they are at the beginning of the text.", ^{
      ...
    });
});
```




# NOT ALL CODE CHANGES HAPPEN AT THE SAME RATE

# SCIENCE




# EXPERIMENTS

# QED

SMALLER CLASSES => MORE CLASSES => LESS BUGS => HAPPIER USERS

# WORLD PEACE