## Refactoring Legacy Code

@sugendran



## Yammer Frontend is a busy place

24 Engineers across 3 offices

5 years of experimentation

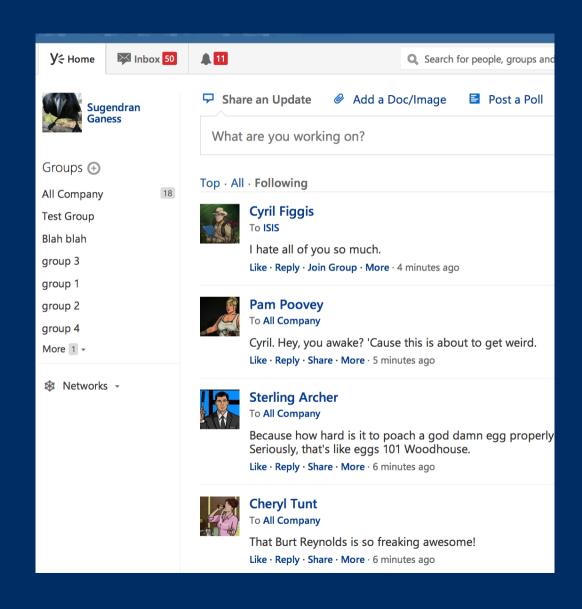
Mix of open source and custom frameworks

#### The Thread List

Core component of Yammer

Lots of experimentation

Hard to understand



Don't hate past team mates

# Past decisions were made for good reason

#### Rewrite vs Refactor

Time to ship is the most important

Many years of changes and bug fixes

Consider the costs to other projects

## Working alone is dangerous

No one works alone at Yammer

On our own we don't have all the answers

A stubborn person will help keep you honest

#### Understand the domain

Do you know all the use cases?

Not everything is written down

Limit the scope after the research

## Keep the team in the loop

Get feedback before starting

Share the spec and answer questions

Everyone should understand the changes

### Release early, release often

Get into master as fast as possible

Use a feature gate

Allows the team to be aware of changes

#### Fork the tests

Tests give you confidence

Spent the last year getting coverage up

You can't refactor without tests

#### Instanbul

http://gotwarlost.github.io/istanbul/

```
Code coverage report for feeds/lib/ui/threads/future/thread list.js
Statements: 86.83% (211 / 243)
                             Branches: 66.37% (75 / 113)
                                                        Functions: 88.64% (39 / 44)
All files » feeds/lib/ui/threads/future/ » thread list.is
  349
  350
  351
                 // Fired when a thread is added to the feed
  352
                 handleThreadAdded: function(thread) {
  353
                   var index = this.delegate.getModelIndex(thread);
  354
                   var component = this. allItems[thread.id];
  355
  356
         61
                   if (this.shouldRenderThread(thread, index)) {
  357
                     if(this. shouldClearNotices) {
  358
                       this.clearNotices();
  359
                       this. shouldClearNotices = false;
  360
  361
                     if (!component) {
  362
                       component = this.createThreadListItem(thread);
  363
                       this. allItems[thread.id] = component;
  364
                       this. timeUpdater.addComponent(component);
  365
                       else {
                       (component.$el | component.$element).detach();
  366
                       if (component.setLastSeenMessageId) {
  367
                         component.setLastSeenMessageId(this. lastSeenMessage
  368
  369
  370
  371
                     component.render();
  372
                     this.insertListItemElement(component.$el | component.$e
  373
                     this.delegate.setThreadAsSeen(thread);
  374
                    else if (this.shouldRenderNewCount(thread, index)) {
  375
                     this.showNewItemNotice();
  376
  377
```

#### Think about the API

How will others use this component?

Hack up different approaches

Be explicit – don't repeat past mistakes

#### Plato

http://platojs.org

```
item.destroy();
  185
  186
  187
  188
  189
                             IPONENTS
        function addMoreButton
  190
4 191
                             ction() {
  192
        Complexity: 1
  193
                             arter-publis
        Length: 46
  194
        Difficulty: 5.06
  195
                              new yam.ui.p
        Est # bugs : 0.07
  196
                              er, '.yj-mes:
  197
4 198
           addMoreButton: function() {
  199
            var opts = {
               id: 'moreButton',
  200
  201
               clickCallBack: _.bind(this
  202
             };
  203
            var button = new yam.ui.shar
  204
             this.add(button, '.yj-list-co
             button.render();
  205
             this. moreButton = button;
  206
```

## When is the project finished?

Don't let scope creep

There will be diminishing returns

Remove the feature gate

## Avoid the long tail of bugs

Release early and have the team find bugs

The AB test system lets us test on Microsoft

Found features we didn't account for

#### Future Works?

Write down the things outside of scope

Will there be a phase 2 and 3?

Give other teams a head start

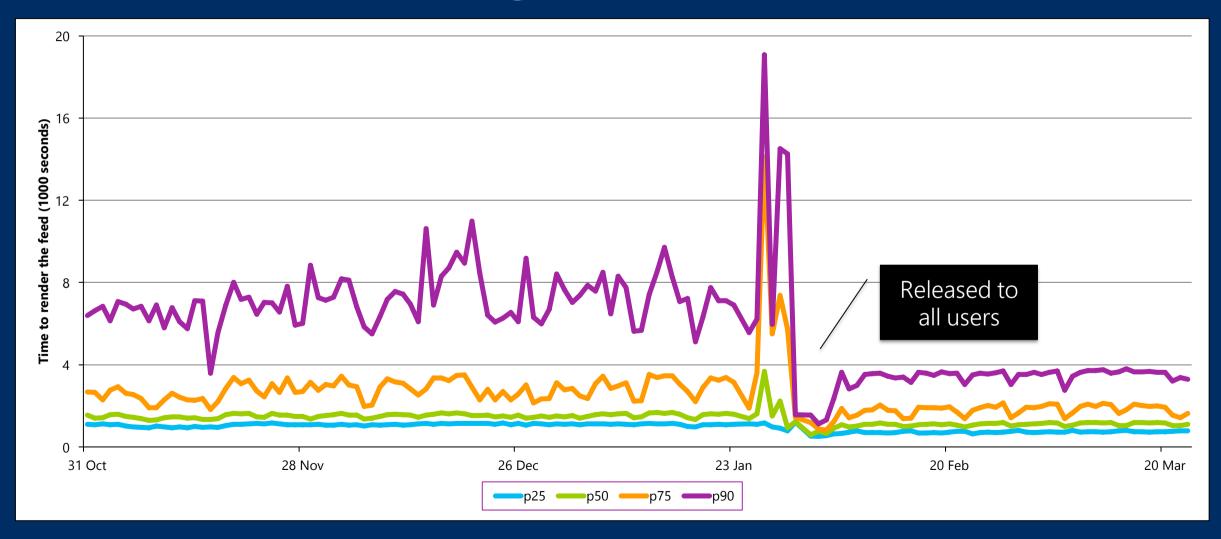
#### Before and After

40% less code

Removed asynchronous rendering

Positive feedback from feed projects

## One more Thing



## Questions?

See you in the discussion track

@sugendran