

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

The screenshot shows the Yammer 'Thread List' interface. At the top, there's a navigation bar with 'Home', 'Inbox' (50), and '11' notifications. A search bar is on the right. Below the navigation bar, the user's profile 'Sugendran Ganess' is shown. The left sidebar contains a 'Groups' section with a list: 'All Company' (18), 'Test Group', 'Blah blah group 3', 'group 1', 'group 2', 'group 4', and a 'More' dropdown with '1' item. Below this is a 'Networks' section. The main content area has a 'Share an Update' button and a text input field 'What are you working on?'. Below the input field, there are tabs for 'Top', 'All', and 'Following'. The feed shows four posts: 1. 'Cyril Figgis' to 'ISIS' with the text 'I hate all of you so much.' and options 'Like · Reply · Join Group · More' (4 minutes ago). 2. 'Pam Poovey' to 'All Company' with the text 'Cyril. Hey, you awake? 'Cause this is about to get weird.' and options 'Like · Reply · Share · More' (5 minutes ago). 3. 'Sterling Archer' to 'All Company' with the text 'Because how hard is it to poach a god damn egg properly Seriously, that's like eggs 101 Woodhouse.' and options 'Like · Reply · Share · More' (6 minutes ago). 4. 'Cheryl Tunt' to 'All Company' with the text 'That Burt Reynolds is so freaking awesome!' and options 'Like · Reply · Share · More' (6 minutes ago).

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

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

Istanbul

<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.js](#)

349		},
350		
351		// Fired when a thread is added to the feed
352		handleThreadAdded: function(thread) {
353	61	var index = this.delegate.getModelIndex(thread);
354	61	var component = this._allItems[thread.id];
355		
356	61	if (this.shouldRenderThread(thread, index)) {
357	55	if (this._shouldClearNotices) {
358		this.clearNotices();
359		this._shouldClearNotices = false;
360		}
361	55	if (!component) {
362	38	component = this.createThreadListItem(thread);
363	38	this._allItems[thread.id] = component;
364	38	this._timeUpdater.addComponent(component);
365		} else {
366	17	(component.\$el component.\$element).detach();
367	17	if (component.setLastSeenMessageId) {
368	17	component.setLastSeenMessageId(this._lastSeenMessage
369		}
370		}
371	55	component.render();
372	55	this.insertListItemElement(component.\$el component.\$e
373	55	this.delegate.setThreadAsSeen(thread);
374	6	} else if (this.shouldRenderNewCount(thread, index)) {
375	6	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>

```
185         item.destroy();
186     }
187 },
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
```

function addMoreButton

Complexity : 1
Length : 46
Difficulty : 5.06
Est # bugs : 0.07

```
addMoreButton: function() {
    var opts = {
        id: 'moreButton',
        clickCallback: _.bind(this
    };
    var button = new Yam.ui.share
    this.add(button, '.yj-list-c
    button.render();
    this._moreButton = button;
```

Keep the team in the loop

Get feedback before starting

Share the spec and answer questions

Everyone should understand the changes

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

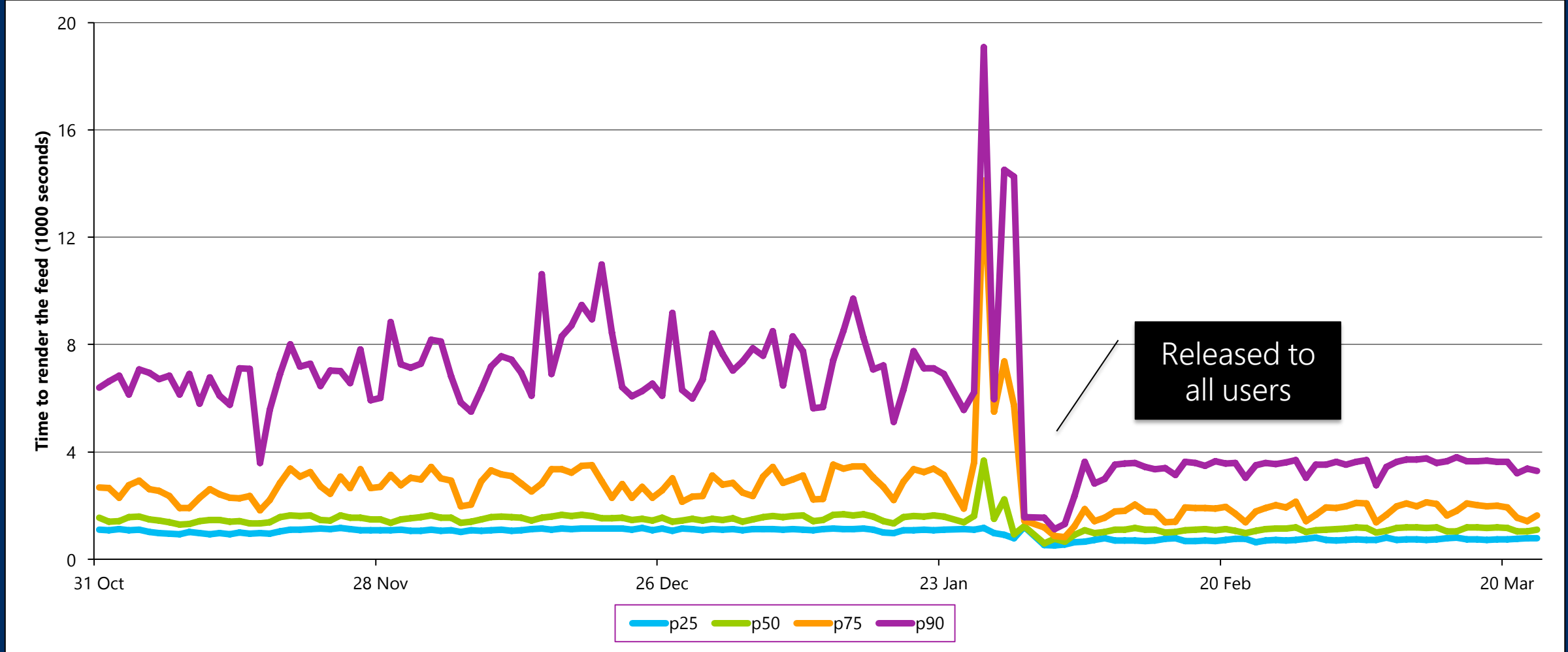
Before and After

40% less code

Removed asynchronous rendering

Positive feedback from feed projects

One more Thing



Questions?

See you in the discussion track

@sugendran