



Tom Colarusso

@etcolarusso

@acvauctions



—

www.acvauctions.com





A Vue To A Kill

—

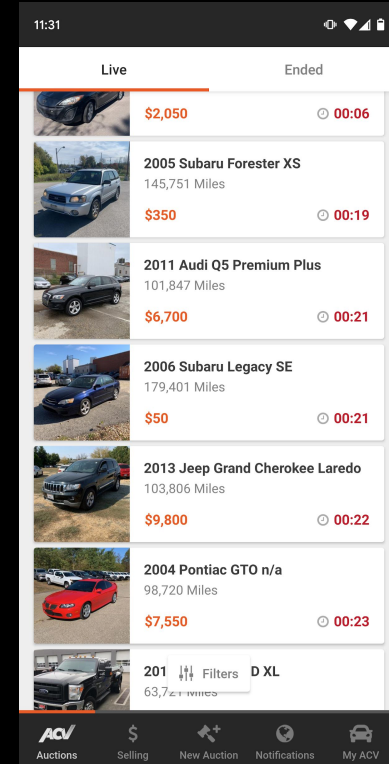
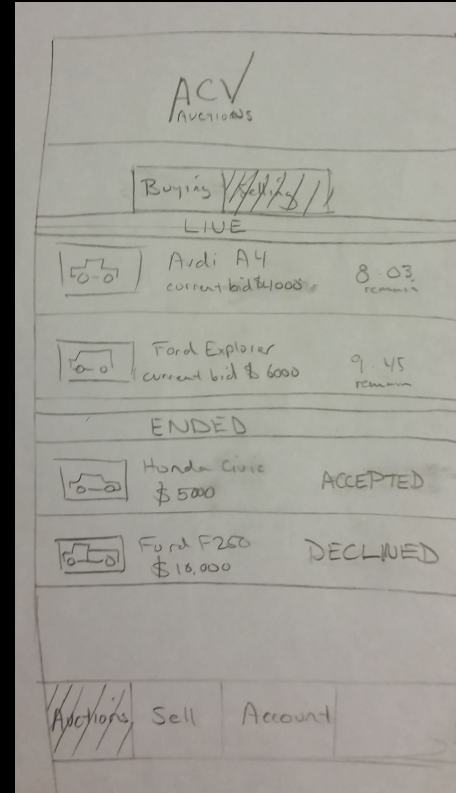
From AngularJS To Vue.js



How we got here...

A brief history

- Company founded 2014
 - Launched 2015
- Dealer to Dealer automobile auctions
 - No consumer facing product.
- 20 minute durations
- Initially only offered Titanium app
 - iOS
 - Android
- Backend built on Perl and Node.js
- No Web product beyond registration



A brief history

- Customers requested a web site
 - Multitasking
 - Researching vehicles.
 - Bidding on multiple vehicles.
 - Older demographic
 - Faster iteration on updates
 - Faster conversion rate

A brief history

- The Legacy Angular App
 - AngularJS 1.4
 - AuctionHouse SDK
 - JavaScript Library
 - Powered the Titanium Apps
 - PHP Middle Tier
 - Serve static assets
 - Session data
 - Proxy to Backend API

The Dreaded 'R' Word...



Your product manager



REWRITE!

To rewrite or not to rewrite.



To rewrite or not to rewrite

● AngularJS End-of-life

- Long-term-support July 1st, 2018 - July 1st, 2021

- <https://docs.angularjs.org/misc/version-support-status#long-term-support>

■ <https://bit.ly/2LXU89J>

Any version branch not shown in the following table (e.g. 1.6.x) is no longer being developed.

Version	Status	Comments
1.2.x	Security patches only	Last version to provide IE 8 support
1.7.x	Long Term Support	See Long Term Support section below.

Long Term Support

On July 1st 2018, AngularJS entered a Long Term Support period for AngularJS.

We now focus exclusively on providing fixes to bugs that satisfy at least one of the following criteria:

- A security flaw is detected in the 1.7.x branch of the framework
- One of the major browsers releases a version that will cause current production applications using AngularJS 1.7.x to stop working
- The jQuery library releases a version that will cause current production applications using AngularJS 1.7.x to stop working.

AngularJS 1.2.x will get a new version if and only if a new severe security issue is discovered.

To rewrite or not to rewrite

- State of our code
 - Built in under 2 months
 - Single developer
 - Lack of best practices
 - No Documentation
 - No Unit/E2E testing
 - Tightly coupled to legacy SDK
 - Deprecated with shift to native apps

To rewrite or not to rewrite

- Developer quality of life
 - Lack of build system
 - Debugging Difficult
 - Business rules undocumented
 - Susceptible to side effects
 - Lack of quality dev tools
 - Copy/Pasta'd code and CSS



Cost of a rewrite



Cost of a rewrite

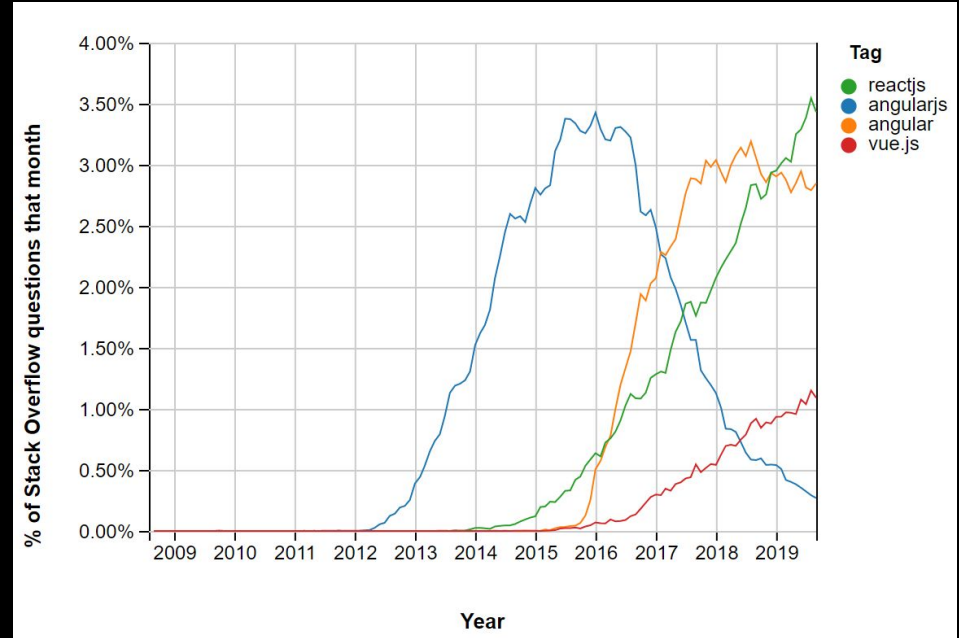
- Onboarding
 - 3-6 months before experienced engineers become productive
 - Juniors struggle
- New Features
 - From design to production
 - Started 2-3 months
 - Ended 6-9 months

Cost of a rewrite

- Supporting Existing Features
 - Increase in
 - Bugs reported
 - LSI's
 - Error Rates
 - Time to diagnose/deploy fix

Cost of a rewrite

- Attracting Talent
 - Growth Period
 - Attracting top talent difficult
 - Older tech not attractive



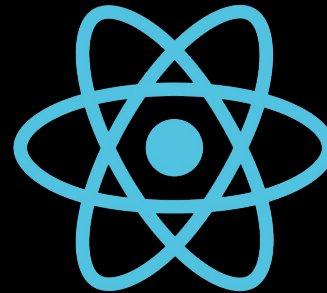
Source: Stack Overflow - <https://bit.ly/2OR4Wse>

How we decided on Vue.js



Selection Process

- 3 Frameworks
 - React
 - Angular 5
 - Vue.js
- Centralized State
 - Flux/Redux/Vuex
- Proof of concept
 - Auction List



React



Vue.js

- Pros
 - Similar syntax
 - v-if/ng-if
 - v-show/ng-show
 - v-for/ng-for
 - Ease of setup and deployment
 - Vue CLI
 - Single file components
 - Single responsibility
 - Easier re-use



Vue.js

- Cons
 - Not developed by a major company
 - Not as widely adopted as React or Angular
 - Large scale applications can become difficult to maintain/manage



Measuring Success (and failure)



Measuring Success

- Define and honor a delivery date
 - July 22 2019
 - Deliver within 1 week of committed delivery date



Measuring Success

- Improve app performance
 - Page load times < 1 second
 - APDEX
 - Set for 2 seconds
 - BrowserInteraction duration
 - 3 seconds

Measuring Success

- Observe Error Budget
 - 1 or fewer LSI's during life of project

Measuring Success

- Generate Robust documentation
 - Implement Storybook component library
 - Include documents on each component
 - Documentation generated for each release

Measuring Success

- Improve unit test coverage
 - $> 30\%$

Our Architecture

Legacy System

- Grunt
 - Uglify minified static assets
 - Manually linked JS/CSS
- Web servers
 - Load Balanced
 - Docker containers
 - NGINX
 - PHP

Legacy System

- Redis instance
 - Session Data
- Client storage
 - PHP cookie for session data
 - localStorage for misc data

Vue System

- Vue CLI 3
 - Webpack 4
 - Babel
 - .env configs
- Web servers
 - Load Balanced
 - Docker containers
 - NGINX
- Service Worker/PWA Support
 - Pre-cache static assets
 - Handles application updates.

Vue System

- Client Storage
 - localStorage
 - IndexedDB
 - Service worker cache

▼ IndexedDB	5	"/css/chunk-061d530b.1f52b00d.css"	▶ {revision: "c8/0598a152c04e4d013", url: "https://app.acvauctions.com/css/chunk-061d530b.1f52b00d.css"}
▼ acv-precache-https__app_acvauctions_com_ - https://app.acvauctions.com	6	"/css/chunk-0823321e.ecbb2a6d.css"	▶ {revision: "999f2c80efe90ddf067d", url: "https://app.acvauctions.com/css/chunk-0823321e.ecbb2a6d.css"}
▣ precached-details-models	7	"/css/chunk-084f92fd.b3d8118b.css"	▶ {revision: "7459535dabee08da5f32", url: "https://app.acvauctions.com/css/chunk-084f92fd.b3d8118b.css"}
▼ google-fonts-webfonts - https://app.acvauctions.com	8	"/css/chunk-0862c69b.7232dbea.css"	▶ {revision: "a911eb38080f2cc62f30", url: "https://app.acvauctions.com/css/chunk-0862c69b.7232dbea.css"}
▣ google-fonts-webfonts	9	"/css/chunk-0a41d682.9c79be14.css"	▶ {revision: "bade7f619b6ee3ed5d2e", url: "https://app.acvauctions.com/css/chunk-0a41d682.9c79be14.css"}
timestamp	10	"/css/chunk-0fd828b7.a3551ca8.css"	▶ {revision: "a899a4bcfffb5531dfcaa", url: "https://app.acvauctions.com/css/chunk-0fd828b7.a3551ca8.css"}
▶ localforage - https://app.acvauctions.com	11	"/css/chunk-113086e8.7857cf7c.css"	▶ {revision: "77912c6b033f99dfc111", url: "https://app.acvauctions.com/css/chunk-113086e8.7857cf7c.css"}

Angular/Vue Integration



Integration

- Git submodule
 - Vue repo as submodule of Angular repo
 - Clear separation of legacy code
 - No risk of accidental coupling
 - Can expose Vue components to Angular

Integration

- Git submodule
 - Separate deployments
 - Easier local development
 - Cleaner cutover once migration is complete.
 - Less tech debt.

Integration

- Feature Development
 - All new features in Vue
 - Limit feature duplication when possible
 - Expose Vue components via iframe
 - Components prop-based (no Vuex)
 - Window events for communication
 - Ex: AMP Audio Player

Integration

- Service Worker
 - Vue CLI PWA plugin
 - Pros:
 - Cache layer
 - Improve load times
 - Background updates
 - Prompt for reload
 - Force refresh

Integration

- Service Worker
 - Cons:
 - Increased complexity
 - Difficult to debug/test

OKR Results



OKR Results

- Define and honor a delivery date
 - July 22 2019
 - Deliver within 1 week of committed delivery date

- Result:
 - Sept. 10th 2019



Measuring Success

- Improve app performance
 - Page load times < 1 second
 - APDEX
 - Set for 2 seconds
 - BrowserInteraction duration
 - 3 seconds

- Results:
 - Time-to-first-byte
 - 0.1 - 0.3 secs
 - 3-4 seconds total
 - APDEX
 - 0.75
 - BrowserInteraction duration
 - 5.89 seconds



Measuring Success

- Observe Error Budget
 - 1 or fewer LSI's during life of project

- Results
 - 1 LSI
 - SEV 3



Measuring Success

- Generate Robust documentation
 - Implement Storybook component library
 - Include documents on each component
 - Documentation generated for each release

- Results:

- Completed
 - Internally hosted
- Completed
 - Release document generator

Measuring Success

- Improve unit test coverage
 - $> 30\%$

- Results:

- 43%



Summary



Tom Colarusso

@etcolarusso

@acvauctions



Questions?

www.acvauctions.com

