

# Tooling Up for Modern SharePoint Web Development

---

Steve Pietrek

SharePoint Saturday Chicago Suburbs

May 5, 2018

# Our Sponsors



**RIGHTPOINT**



# Steve Pietrek



**Cardinal Solutions – IT Consulting  
App Dev Practice Manager  
Front End Developer  
Triangle ReactJS Meetup Organizer  
SharePoint Application Architect**

**@spietrek  
spietrek@gmail.com  
<https://github.com/spietrek>  
<https://www.medium.com/@spietrek>  
Raleigh/Durham**

# Topics

Web  
Development  
and Tools

SharePoint  
Projects and  
Approach

Responsive  
Design &  
Branding

SharePoint  
Development

Best Practices

# Web Development

---

## SharePoint Development

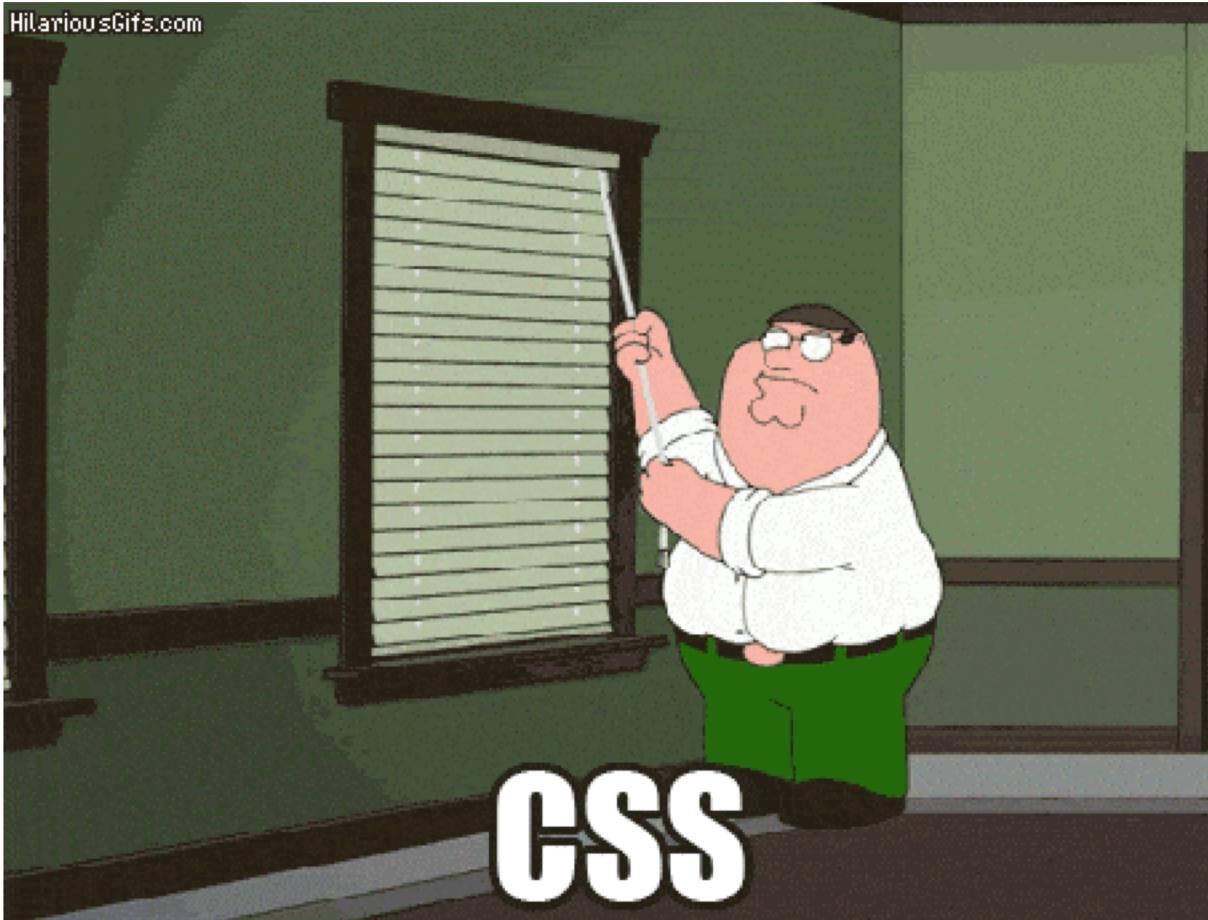
- Out-of-the-box SharePoint Knowledge
- Site Columns/Content Types
- JavaScript (jQuery)
- HTML/CSS
- XML
- Data Access/REST/SPServices
- Branding/Master Pages/Page Layouts
- Workflow
- Web Parts
- Solution Packages
- Building a SharePoint VM
- Deployment/PowerShell

## Front End Development

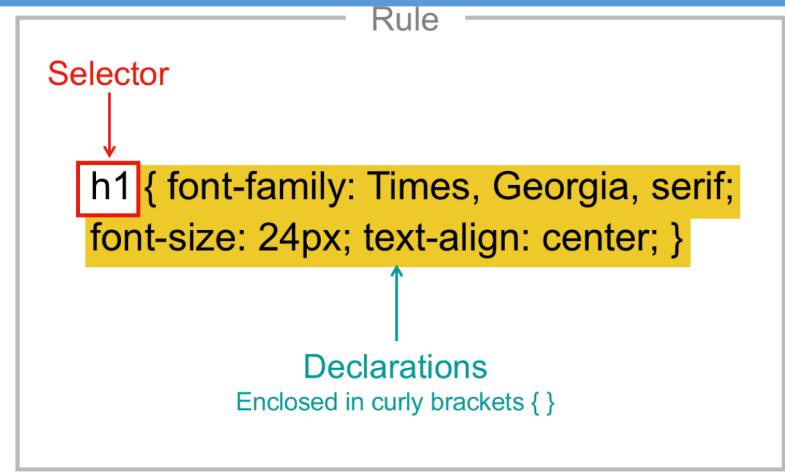
- Node.js/NPM/Webpack/Babel/Gulp
- JavaScript Libraries/Frameworks (jQuery, React, Angular)
- HTML/CSS/Sass/Less/CSS in JS
- JSON
- Responsive Design
- Git
- Linting/Formatting/Static Type Checking
- Asynchronous JavaScript
- Promises & Async/Await
- ES6+

Next Generation SharePoint Developer

HilariousGifs.com



# CSS Knowledge



## Basic CSS – Selectors, Declarations, Pseudo-Classes, Inheritance

---

Sass – Is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). Uses variables, mixins, operations, and functions.

---

Less – Extends CSS with dynamic behavior such as variables, mixins, operations and functions.

---

Flexbox – Allows responsive elements within a container to be automatically arranged depending upon screen size (or device).

---

CSS Grid – Is a technique in CSS that allows web developers to create complex responsive web design layouts more easily and consistently across browsers.

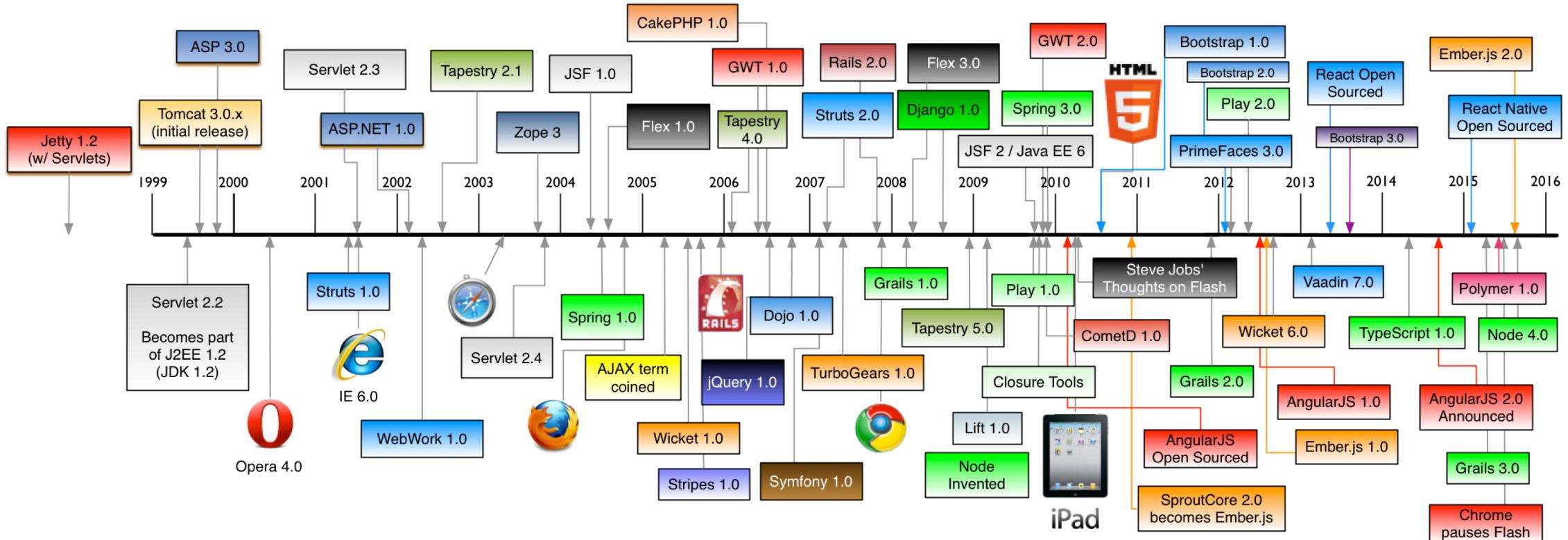
---

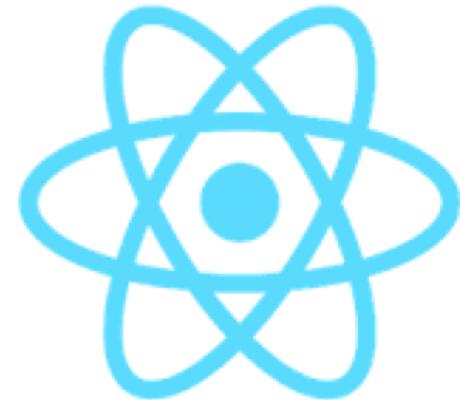
CSS-in-JS – Abstracts the CSS model to the component level, rather than the document level (modularity) (styled-components).



# DAYS

**SINCE LAST NEW  
JAVASCRIPT FRAMEWORK**





# ES6 / ECMAScript 6

---

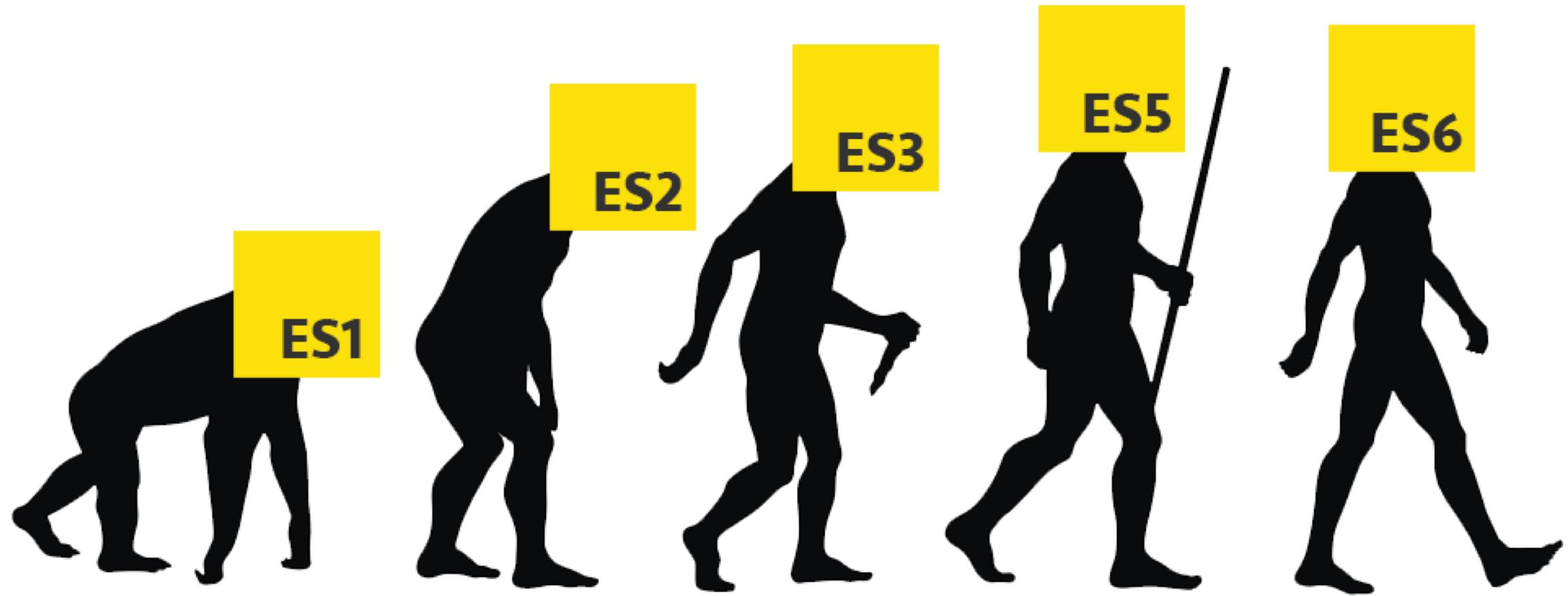
1997

1998

1999

2009

2015



# Block-Scope Constructs Let and Const

- **var** is hoisted and function-scoped
- **let** is not hoisted and limited in scope to the block, statement, or expression where it is used
- **const** is immutable (read-only reference to a value) and limited in scope to the block, statement, or expression where it is used
- Recommendation: always use **const** until you need to mutate

# Template Literals

- Avoid using concatenation
- Use backtick

- ES5

```
var name = "My name is " + firstName + " " + lastName
```

- ES6

```
const name = `My name is ${firstName} ${lastName}`
```

# Arrow Functions

- Traditional function with a **.bind(this)**
- Biggest benefit > **this** will have the same value as the context of the function
- No more **that = this** or **self = this**

- ES5

```
var names = ['Steve', 'Laura'];
var messages = names.map(function (name, index) {
  return 'Name of ' + index + ' element is ' + name + ''
})
```

- ES6

```
const names = ['Steve', 'Laura'];
const messages = names.map((name, index) => `Name of ${index} element is ${name}`)
```

# Spread/Destructuring Assignment

- Breaks down objects and arrays into individual variables

## Example 1:

```
const parts = ['shoulders', 'knees'];
const lyrics = ['head', ...parts, 'and', 'toes'];
console.log(lyrics); // ['head', 'shoulders', 'knees', 'and', 'toes']
const [first] = lyrics;
console.log(first); // head
```

## Example 2:

```
const TodoListClearButton = ({ isFetching, todos, onClear }) => (
  isFetching || !todos.length ? null : <button onClick={onClear}>Clear Todos</button>
);
```

# Tools

---

# Node.js & NPM

- Node.js
  - Open source, cross-platform runtime environment for server-side and networking applications
  - Applications are written in JavaScript
  - Based on the Google's V8 JavaScript Engine
  - Uses an event-driven, single threaded, and non-blocking I/O model => lightweight and efficient
  - Many development tools leverage Node.js
- Npm
  - Node Package Manager
  - Online repository for the publishing of open-source Node.js projects
  - Packages can be installed globally (across all projects) or locally (specific to project)
  - Command-line utility for interacting with repositories (package installation, version management, and dependency management)
  - Yarn is gaining traction as a package manager

# ESLint

```
const mockedRecentSearches = getRecentSearchesData();
💡 const mockedFavorites = getFavoritesData();
const mockedMerchants = getMerchantsData(); You, a few seconds ago
const mockedActiveMerchants = getActiveMerchantsData();
```

Help with code quality

Catch common errors  
(`console.log`,  
`debugger`)

Provide consistency between developers

Configure rules to validate code  
(Formatting Rules and Code-quality rules)

Rules can be used to ignore, warn, or error

.eslintrc is used to define rules

Most IDE's have ESLint plugins

Future: Supports ES6 and JSX

# Prettier

“Opinionated code formatter”

Focuses on  
formatting rules

Supports JavaScript  
(ES6, ES7), JSX,  
Flow, TypeScript,  
CSS, JSON

Enforces a  
consistent code  
style

Provide consistency  
between  
developers

ESLint => many  
configuration rules,  
Prettier => few  
configuration rules

Can use ESLint and  
Prettier together  
(eslint-plugin-  
prettier)

# Static Type Checking

---

# Static Type Checking

01

Static usually means “at compile time” or “without running a program”

02

Dynamic means “at runtime”

03

JavaScript is dynamic

# Benefits of Static Type Checking

You get more errors at “compile” time

Usually faster than running unit tests and tends to catch a different category of errors

It helps IDEs with auto-completion

Type annotations are useful for documenting parts of an API

Checking and documenting types helps large teams collaborate

It gives you an additional way of specifying what you require from or provide for your collaborators

The logo for TypeScript features the word "TypeScript" in a large, blue, sans-serif font. Below the text is a stylized silhouette of a city skyline in blue, composed of various building shapes. A thin vertical line is positioned to the right of the skyline.

TypeScript



A STATIC TYPE CHECKER FOR JAVASCRIPT

# Static Type Checkers

TypeScript 2.5 is now available. [Download](#) our latest version today!

[Using Classes](#)[TypeScript](#)[Share](#)[Options](#)[Run](#)[JavaScript](#)

```
1 class Greeter {  
2     greeting: string;  
3     constructor(message: string) {  
4         this.greeting = message;  
5     }  
6     greet() {  
7         return "Hello, " + this.greeting;  
8     }  
9 }  
10 let greeter = new Greeter("world");  
11  
12 let button = document.createElement('button');  
13 button.textContent = "Say Hello";  
14 button.onclick = function() {  
15     alert(greeter.greet());  
16 }  
17  
18 document.body.appendChild(button);
```

```
1 var Greeter = /** @class */ (function () {  
2     function Greeter(message) {  
3         this.greeting = message;  
4     }  
5     Greeter.prototype.greet = function () {  
6         return "Hello, " + this.greeting;  
7     };  
8     return Greeter;  
9 })();  
10 var greeter = new Greeter("world");  
11 var button = document.createElement('button');  
12 button.textContent = "Say Hello";  
13 button.onclick = function () {  
14     alert(greeter.greet());  
15 };  
16 document.body.appendChild(button);  
17
```

# TypeScript

Quick Start Documentation Download Connect Playground

Fork me on GitHub

TypeScript 2.5 is now available. [Download](#) our latest version today!

Using Classes

TypeScript

Share

Options

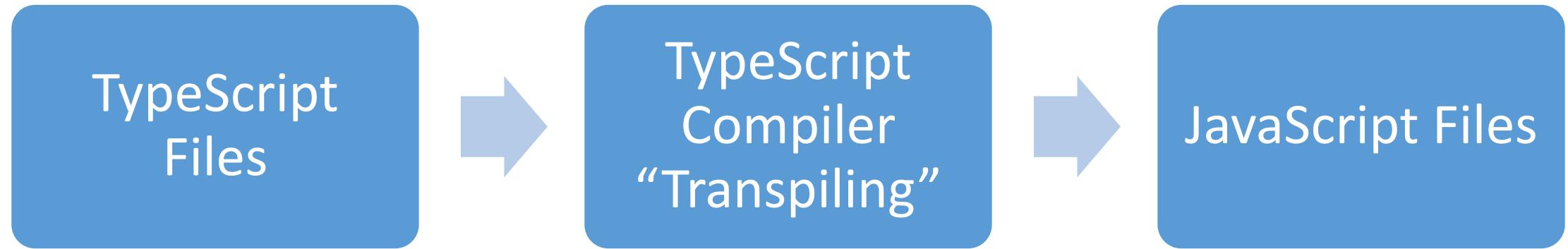
Run

JavaScript

```
1 class Greeter {  
2     greeting: string;  
3     constructor(message: string) {  
4         this.greeting = message;  
5     }  
6     greet() {  
7         return "Hello, " + this.greeting;  
8     }  
9 }  
10 let greeter = new Greeter(1234);  
11  
12 let button = document.createElement('button');  
13 button.textContent = "Say Hello";  
14 button.onclick = function() {  
15     alert(greeter.greet());  
16 }  
17  
18 document.body.appendChild(button);
```

Argument of type '1234' is not assignable to parameter of type 'string'.

```
1 var Greeter = /** @class */ (function () {  
2     function Greeter(message) {  
3         this.greeting = message;  
4     }  
5     Greeter.prototype.greet = function () {  
6         return "Hello, " + this.greeting;  
7     };  
8     return Greeter;  
9 }());  
10 var greeter = new Greeter(1234);  
11 var button = document.createElement('button');  
12 button.textContent = "Say Hello";  
13 button.onclick = function () {  
14     alert(greeter.greet());  
15 };  
16 document.body.appendChild(button);  
17
```



How does TypeScript work?

# Babel

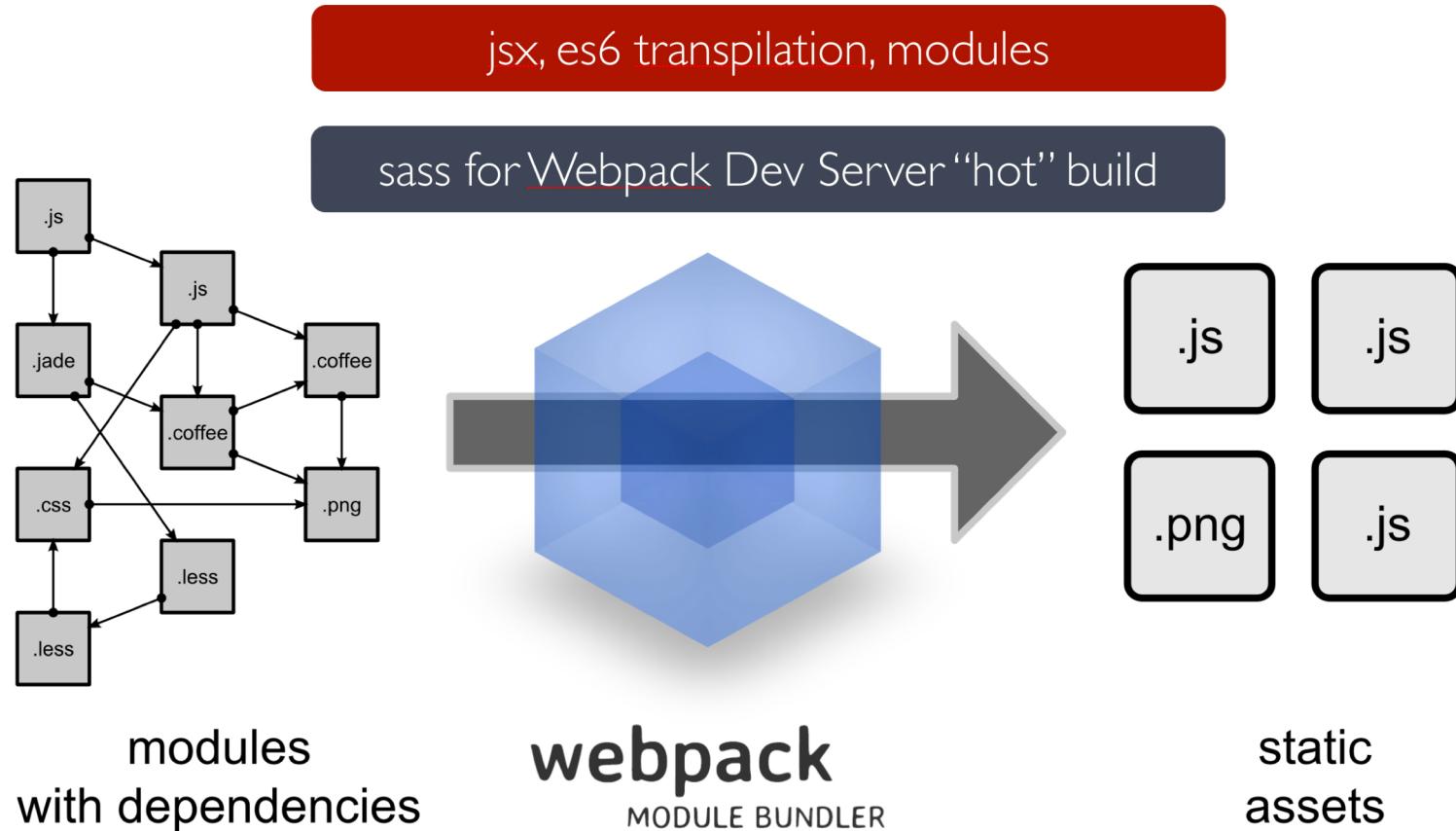
- Babel is a JavaScript compiler.
- Use next generation JavaScript, today.
- Babel “transpiles” to ES5 JavaScript.
- Many browsers do not support all of ES6+ features.

- ✓ Arrow functions
- ✓ Async functions
- ✓ Async generator functions
- ✓ Block scoping
- ✓ Block scoped functions
- ✓ Classes
- ✓ Class properties
- ✓ Computed property names
- ✓ Constants
- ✓ Decorators
- ✓ Default parameters
- ✓ Destructuring
- ✓ Do expressions
- ✓ Exponentiation operator
- ✓ For-of
- ✓ Function bind
- ✓ Generators
- ✓ Modules
- ✓ Module export extensions
- ✓ New literals
- ✓ Object rest/spread
- ✓ Property method assignment
- ✓ Property name shorthand
- ✓ Rest parameters
- ✓ Spread
- ✓ Sticky regex
- ✓ Template literals
- ✓ Trailing function commas
- ✓ Type annotations
- ✓ Unicode regex

# Task Runners

- Sass/Less => CSS
- JavaScript and CSS Minify
- JavaScript Concatenate
- Coffee Script => JavaScript
- Babel => JavaScript
- Compress Images
- File Watches > update as files change
- Lint files
- Run tests

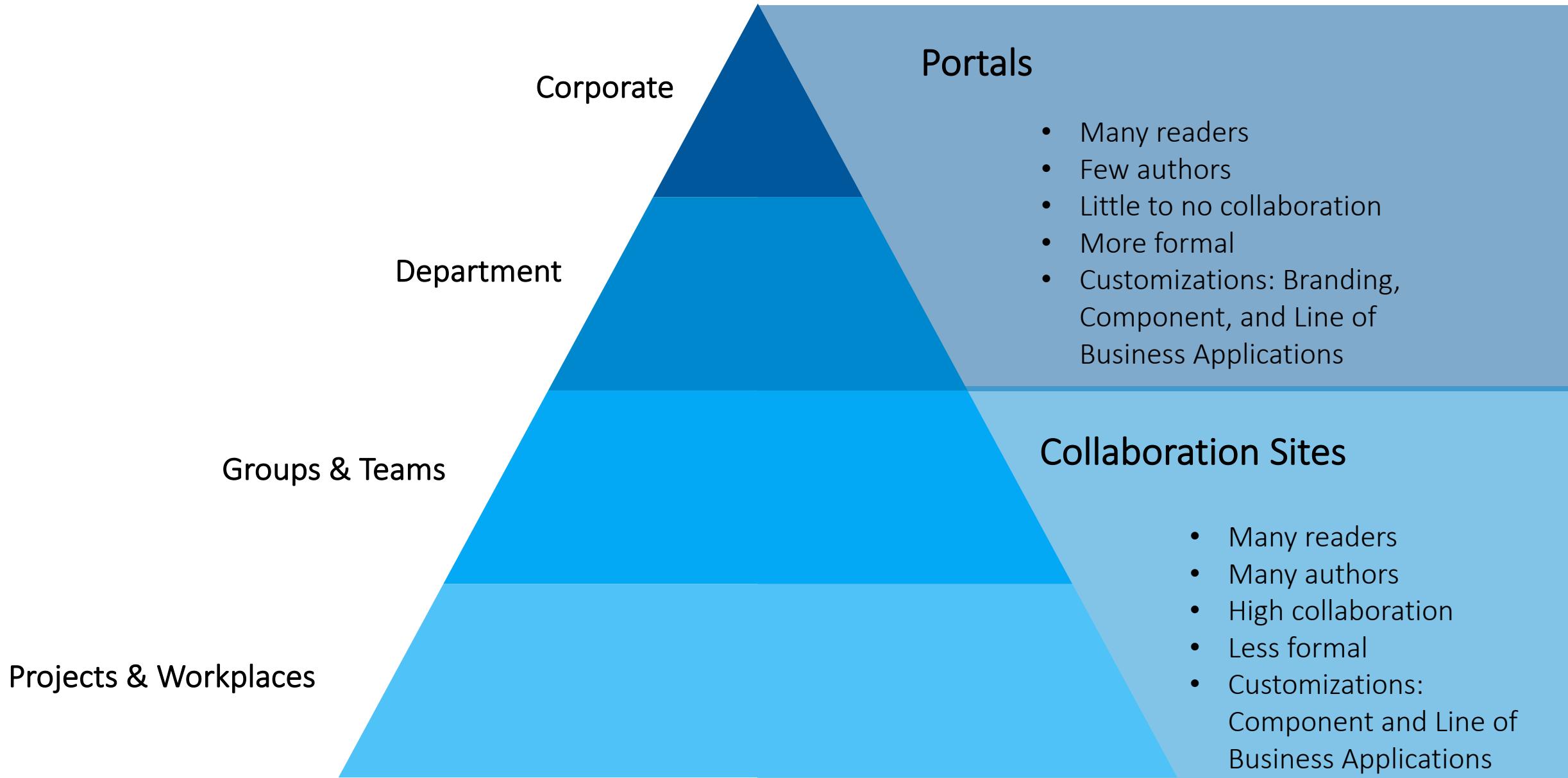
# Webpack



# SharePoint Projects

---

# SharePoint Sites





Concur

Work Day

iService

Mind/Body

Dynamics

Survey

84.51 Systems

Favorites

Homepage

## Upcoming Events

[View All](#)

15 / OCT

**The Best Event Ever at Cardinal**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Morbi ultrices nisl non massa iaculis finibus. Antesque orci lacus...

[Cincy](#) / [Add to Calendar](#)

26 / OCT

**The Best Event Ever at Cardinal**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Morbi ultrices nisl non massa iaculis finibus.

[Cincy](#) / [Add to Calendar](#)

02 / NOV

**The Best Event Ever at Cardinal**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Morbi

## People

[People Search](#)

Will Thomas

Suresh  
Devarakonda

John Smith



Lifei Xu

Office 365 | Sites

3



My UT Health  
People. Passion. Purpose.

about    academics    patient care    research    committees    departments    news & events    services & support

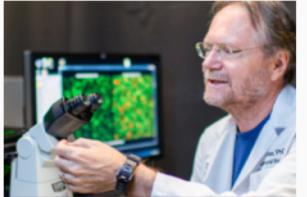
News FILTER BY: UT Health add news | view all

**UT Health San Antonio News**  
3/23/2016  
Donec sed odio dui. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Nullam id dolor id nibh ultricies vehicula ut id elit.  
[read more >>](#)

**TAGS:** UT Health | Awards | Humans Resources

**UT Health San Antonio News**  
3/23/2016  
Donec sed odio dui. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.  
[read more >>](#)

**TAGS:** UT Health | Health Discoveries

**UT Health San Antonio News**  
3/23/2016  


■ students    ★ business applications    ♥ favorites

search Q

Quick Links

 Maps  Students  Peoplesoft  HR  Faculty  Help Desk

Directory

search

students  faculty  employees

**MESSAGE FROM THE PRESIDENT**

Ours is a story of discovery, vision and commitment.  
[read now >>](#)



All Deliverables

In Progress Deliverables

Not Started Deliverables

Parked Deliverables

Hold Deliverables

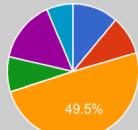
Completed Deliverables

Withdrawn Deliverables

Late Deliverables

Deliverables with Late Tasks

107 Total Deliverables

 12  
IN  
PROGRESS  
DELIVERABLES 21  
LATE  
DELIVERABLES 4  
LATE  
TASKS

## All Deliverables

Filter results

Deliverable Name	Status	% Complete	Ticket #	Project	Owner	# Late Tasks	Modified	View
alphas-violet	In Progress	<div style="width: 39%;">39</div>	455480-8313	luctus sit	Steve Pietrek	3	Mar 8, 2017 2:22:43 PM	
ufo	Hold-Vendor	<div style="width: 31%;">31</div>	855846-6408	lorem eu	Hyatt Eaton	1	Feb 11, 2017 9:54:39 AM	
racerigo	Parked	<div style="width: 38%;">38</div>	784321-4482	velit. Quisque	Joshua Colon	0	Feb 11, 2017 9:54:39 AM	
pizibud	In Progress	<div style="width: 25%;">25</div>	376398-0038	hendrerit consectetur,	Galvin Lester	0	Feb 11, 2017 9:54:39 AM	
kepakha	Hold-Vendor	<div style="width: 0%;">0</div>	899457-8444	In nec	Alexander Townsend	0	Feb 11, 2017 9:54:39 AM	
hu	Hold-Vendor	<div style="width: 93%;">93</div>	785984-3331	ligula elit,	Gray Lewis	0	Feb 11, 2017 9:54:39 AM	
hu	Hold-Vendor	<div style="width: 19%;">19</div>	535866-0354	Sed nec	Dalton Crane	0	Feb 11, 2017 9:54:39 AM	
dipnaaro	Withdrawn	<div style="width: 41%;">41</div>	702417-1097	nec, diam.	Kirk Gentry	0	Feb 11, 2017 9:54:39 AM	
alpha-zeta	Withdrawn	<div style="width: 94%;">94</div>	528753-8465	et nunc.	Vincent Ferguson	0	Feb 11, 2017 9:54:25 AM	
alpha-psi	Parked	<div style="width: 10%;">10</div>	118791-4997	Praesent interdum	Salvador Sullivan	0	Feb 11, 2017 9:54:25 AM	

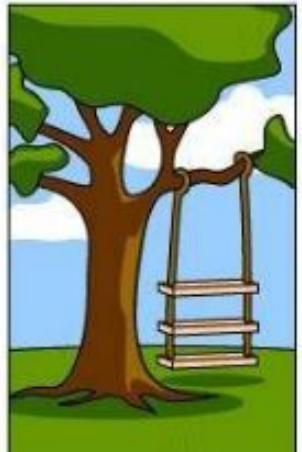
5 10 25 50

« 1 2 3 4 5 »

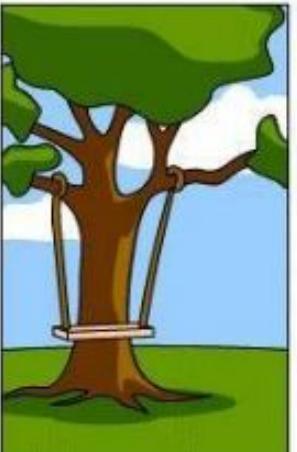
Page 1 of 11, of 107 entries

# Approach

---



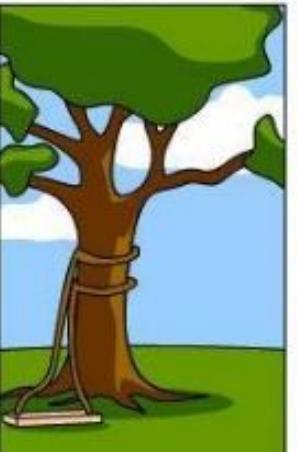
How the customer explained it



How the Project Leader  
understood it



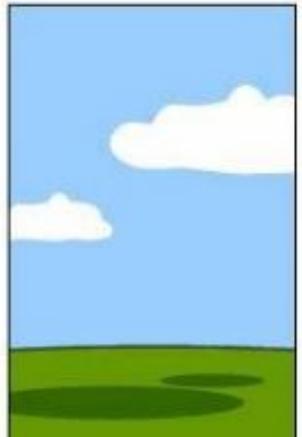
How the Analyst designed it



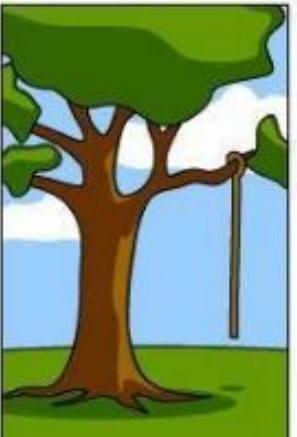
How the Programmer wrote it



How the Business Consultant  
described it



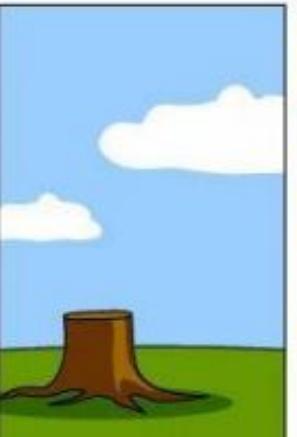
How the project was  
documented



What operations installed



How the customer was billed

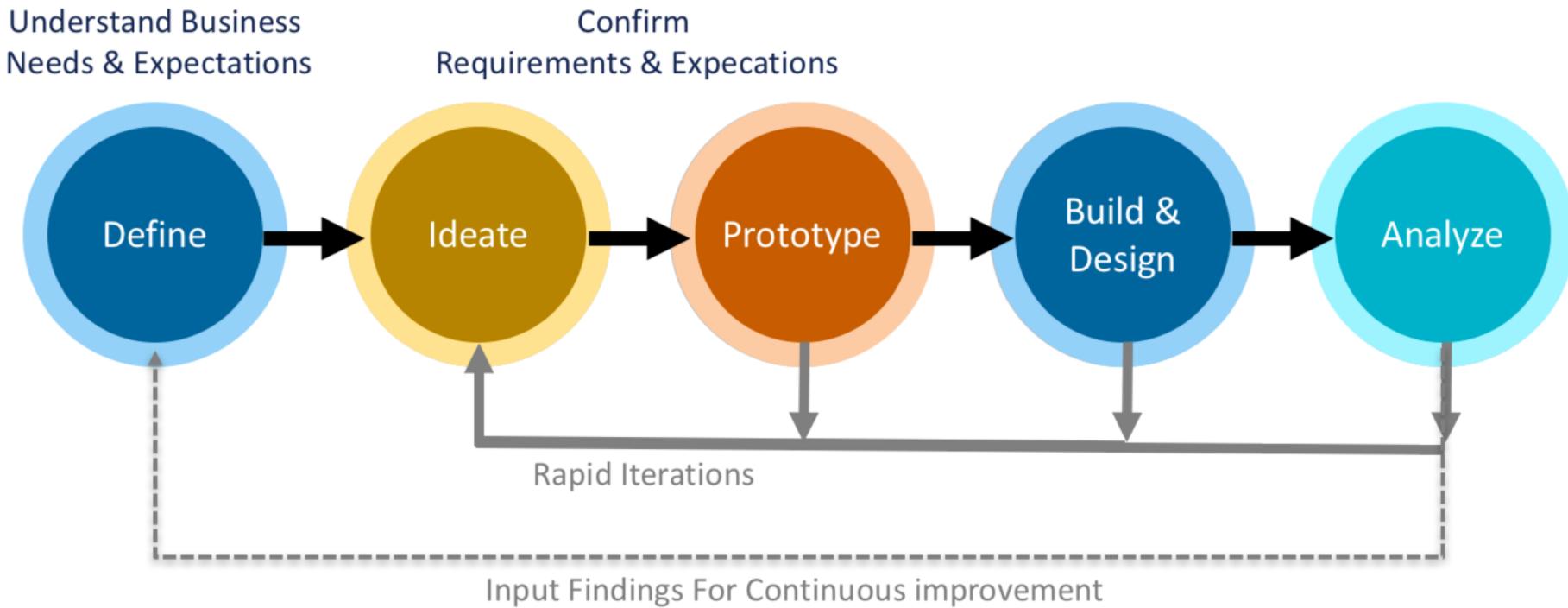


How it was supported

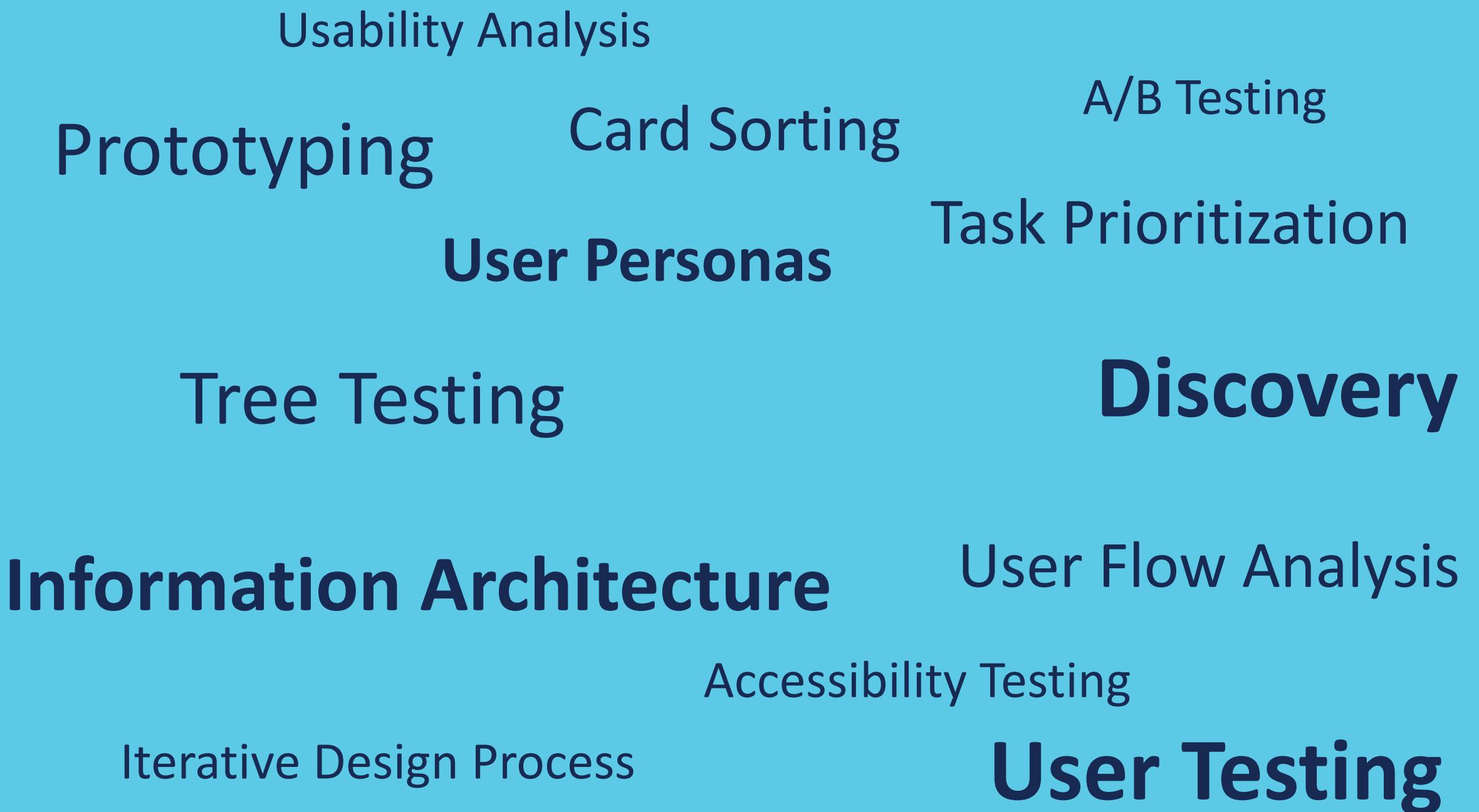


What the customer really  
needed

# User Experience & Business Analysis



An iterative user experience driven design process that engages subject matter experts (SMEs) to ensure the solution is serving the users. Teams will *DEFINE* the solution and requirements together. *IDEATE* ways to provide a solutions that meets the requirements and objectives. *PROTOTYPE* a solution to confirm the approach with the team. Often multiple iterations of Ideate & Prototype are done until the team has agreed on the solutions approach. *BUILD & DESIGN* the application to meet the requirements and continually *ANALYZE* the solution as a team to better serve the users.





# Card Sorting

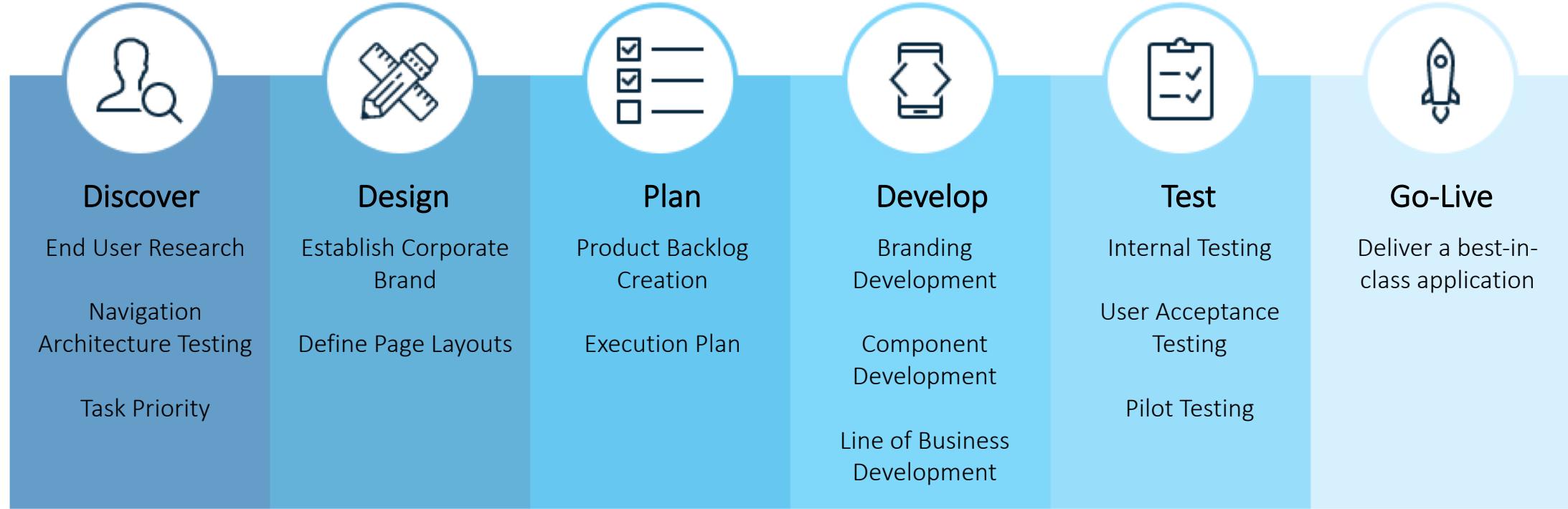
---

Card sorting is a method used to help design or evaluate the information architecture of a site or application. Participants organize topics into categories that make sense to them and label these groups.

Knowing how your users group information will help us:

- Build the structure for the intranet or app (Information Architecture)
- Decide what to put on the homepage
- Label categories and navigation

# Typical Approach



## Communicate

Communication Plan

Global Engagement Strategy



## Train

Administrative Training

End User Training

# Responsive Design

---

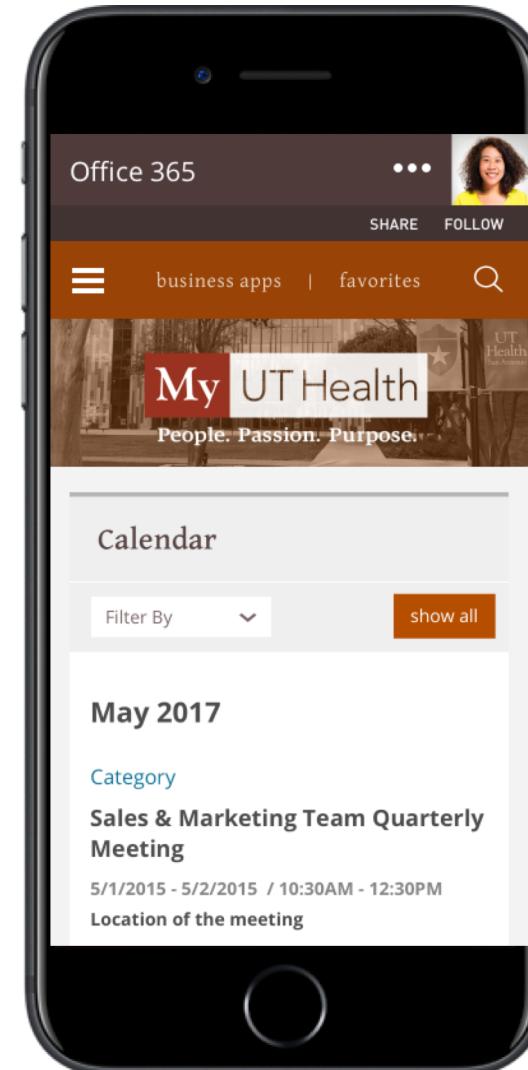
# What is Responsive Design?

- Responsive web design makes your web page look good on all devices (Desktop, Tablet, Mobile)
- Responsive web design uses only HTML and CSS.
- Generally, content is not removed – the content adapts to fit on any device.
- Viewport is the visible area of a web page
- Media queries run only under certain conditions
- Always design for Mobile First

```
@media only screen and (max-width: 500px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

# Intranet Responsive Design

The screenshot shows the desktop version of the My UT Health intranet. At the top, there's a dark header bar with 'Office 365' and 'Sites' links. Below it is a banner featuring the 'My UT Health' logo and the tagline 'People. Passion. Purpose.' A navigation bar includes links for 'about', 'academics', 'patient care', 'research', 'committees', 'departments', 'news & events', and 'services & support'. A search bar is also present. The main content area displays news items under 'News' (e.g., 'UT Health San Antonio News' from 3/23/2016), a 'Quick Links' section with icons for Maps, Students, Peoplesoft, HR, Faculty, and Help Desk, a 'Directory' search bar, and a 'MESSAGE FROM THE PRESIDENT' section featuring a portrait of a man.



# SharePoint Branding

---

# SharePoint Branding Anti-Patterns

1

Brand or Hide the  
Office 365 Suite Bar

2

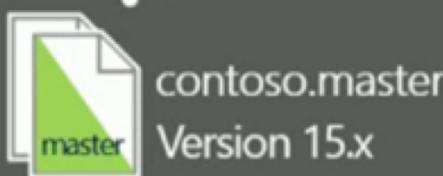
Customize branding  
for personal sites  
(OneDrive for  
Business sites)

3

Implement your  
custom brand using  
custom master  
pages

# What is the challenge with custom master page?

New custom master page is created by copying oob master or starting from scratch using oob master as the reference



<< Copy >>



Significant differences on the outcome unless custom master page been updated during the releases.



Custom Master



Oob Master

Time

Service updates for introducing new version of the out of the box master page with some new capabilities or bug fixes.

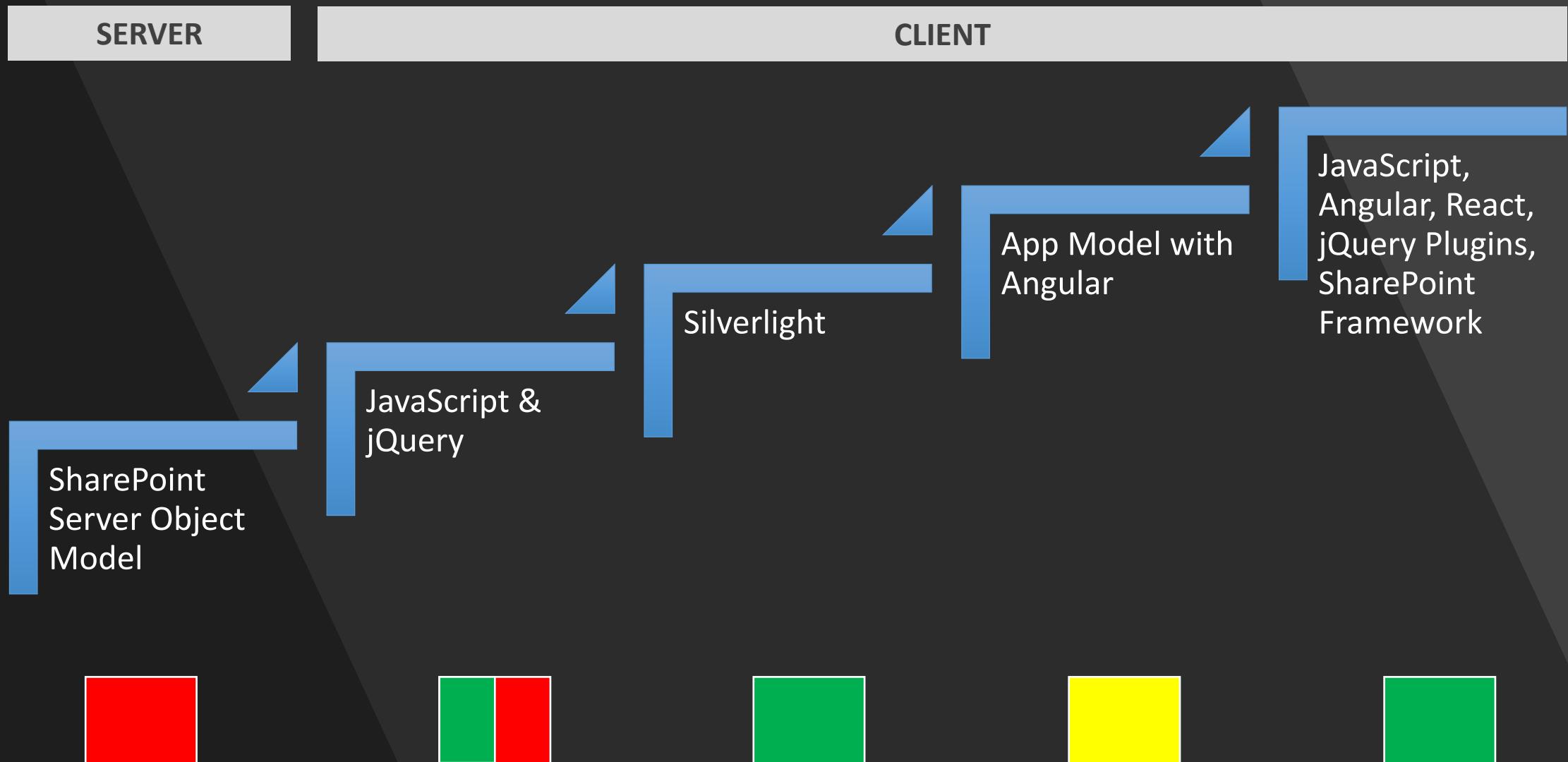
# SharePoint Branding Approach

Implement	Implement custom CSS
Create	Create custom page layouts
Add	Add common branding elements (e.g. Footer) by injecting JavaScript
Deploy	Deploy files through a solution deployed using PowerShell

# SharePoint Development

---

# My SharePoint Development Journey



# Client-Side Development Advantages



## Infrastructure

- Isolation & multitenant
- Runs in client browsers
- Reduces risk to farm
- Simplifies SharePoint upgrades
- Cloud ready

## Developers

- Larger developer reach
- Use industry standards
- Use non-SharePoint technologies
- Lifecycle management  
(development, deployment,  
versioning, upgrades)

# Transition Front End Developer

Customization	Strength Areas	Weakness Areas	Front End Developer
Branding	HTML, CSS, Responsive Design	SharePoint Knowledge, Deployment, Page Layouts	Possibly
Script Editors – JavaScript Injection	JavaScript development	SharePoint Knowledge, Data access, Deployment, Page Layouts, 5000 Item Threshold Limit	High
SharePoint Framework	JavaScript development	SharePoint Knowledge, Data access, Deployment, Project Structure, TypeScript, 5000 Item Threshold Limit	Possible with Mentoring
Workflow	N/A	SharePoint Knowledge, SharePoint Designer, Flow	No
Server-side Customizations	N/A	Web Parts, Timer Jobs, Content Types, Site Columns, XSLT, Solution Packages	No

# SharePoint Data Access

---

# SharePoint 2013/2016 REST

- Use jQuery AJAX
- Options
  - \$select
  - \$filter
  - \$top
  - \$skip
  - \$orderBy
  - \$expand
- Best practices
  - Use Postman or Fiddler to execute and test queries
  - Set the Accept header
  - Only retrieve what you need by using \$select and \$filter
  - “Chunky, not chatty”

```
var req = {
  method: 'GET',
  url: restQuery.url(),
  headers: {
    'Accept': 'application/json;odata=verbose'
  }
};

return $http(req)
  .then(success)
  .catch(fail);

function success(response) { ... }

function fail(error) { ... }
```

# JavaScript Injection

---

Office 365 | Sites

3



My UT Health  
People. Passion. Purpose.

about    academics    patient care    research    committees    departments    news & events    services & support

News FILTER BY: UT Health ▼ add news | view all

**UT Health San Antonio News**  
3/23/2016  
Donec sed odio dui. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Nullam id dolor id nibh ultricies vehicula ut id elit.  
[read more >>](#)

TAGS: UT Health | Awards | Humans Resources

**UT Health San Antonio News**  
3/23/2016  
Donec sed odio dui. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.  
[read more >>](#)

TAGS: UT Health | Health Discoveries

**UT Health San Antonio News**  
3/23/2016

Quick Links

 Maps  Students  Peoplesoft  HR  Faculty  Help Desk

Directory

search

students  faculty  employees

**MESSAGE FROM THE PRESIDENT**  
Ours is a story of discovery, vision and commitment.  
[read now >>](#)

All Deliverables

In Progress Deliverables

Not Started Deliverables

Parked Deliverables

Hold Deliverables

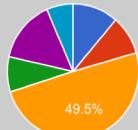
Completed Deliverables

Withdrawn Deliverables

Late Deliverables

Deliverables with Late Tasks

107 Total Deliverables

 12  
IN  
PROGRESS  
DELIVERABLES 21  
LATE  
DELIVERABLES 4  
LATE  
TASKS

## All Deliverables

Filter results

Deliverable Name	Status	% Complete	Ticket #	Project	Owner	# Late Tasks	Modified	View
alphas-violet	In Progress	<div style="width: 39%;">39</div>	455480-8313	luctus sit	Steve Pietrek	3	Mar 8, 2017 2:22:43 PM	
ufo	Hold-Vendor	<div style="width: 31%;">31</div>	855846-6408	lorem eu	Hyatt Eaton	1	Feb 11, 2017 9:54:39 AM	
racerigo	Parked	<div style="width: 38%;">38</div>	784321-4482	velit. Quisque	Joshua Colon	0	Feb 11, 2017 9:54:39 AM	
pizibud	In Progress	<div style="width: 25%;">25</div>	376398-0038	hendrerit consectetur,	Galvin Lester	0	Feb 11, 2017 9:54:39 AM	
kepakha	Hold-Vendor	<div style="width: 0%;">0</div>	899457-8444	In nec	Alexander Townsend	0	Feb 11, 2017 9:54:39 AM	
hu	Hold-Vendor	<div style="width: 93%;">93</div>	785984-3331	ligula elit,	Gray Lewis	0	Feb 11, 2017 9:54:39 AM	
hu	Hold-Vendor	<div style="width: 19%;">19</div>	535866-0354	Sed nec	Dalton Crane	0	Feb 11, 2017 9:54:39 AM	
dipnaaro	Withdrawn	<div style="width: 41%;">41</div>	702417-1097	nec, diam.	Kirk Gentry	0	Feb 11, 2017 9:54:39 AM	
alpha-zeta	Withdrawn	<div style="width: 94%;">94</div>	528753-8465	et nunc.	Vincent Ferguson	0	Feb 11, 2017 9:54:25 AM	
alpha-psi	Parked	<div style="width: 10%;">10</div>	118791-4997	Praesent interdum	Salvador Sullivan	0	Feb 11, 2017 9:54:25 AM	

5 10 25 50

« 1 2 3 4 5 »

Page 1 of 11, of 107 entries

# JavaScript Injection

Paste HTML, CSS, and JavaScript directly in Script Editor or reference in custom Page Layout

Runs in the context of current page (user and DOM)

Get page information (current user, site URL, etc.) in page using `_spPageContextInfo`

Not easy to provide configuration options – working on a way ☺

Script Editor is not marked as “Safe for Scripting” > not available on self-service sites (Team Sites, Group Sites)

Users may not have permissions to configure Script Editor web parts

# Deployment Workflow

## Local Test

- Build process
- Disable web security (--disable-web-security) errors in Chrome
- Functional testing

## Upload Source Code to Document library

- Same folder structure as local environment
- Only the build files are needed
- Upload source code

## SharePoint Deployment & Test

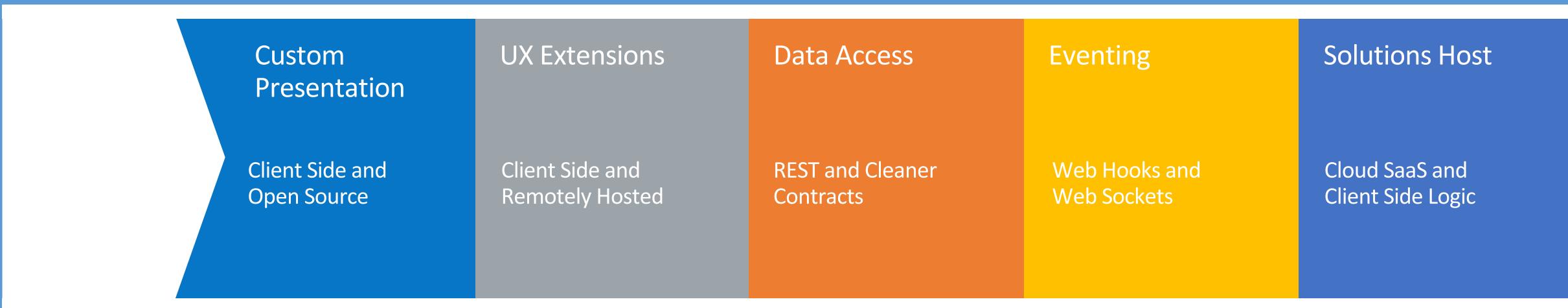
- Add Script Editor web part to Page
- Add markup
- Functional testing

# SharePoint Framework

---

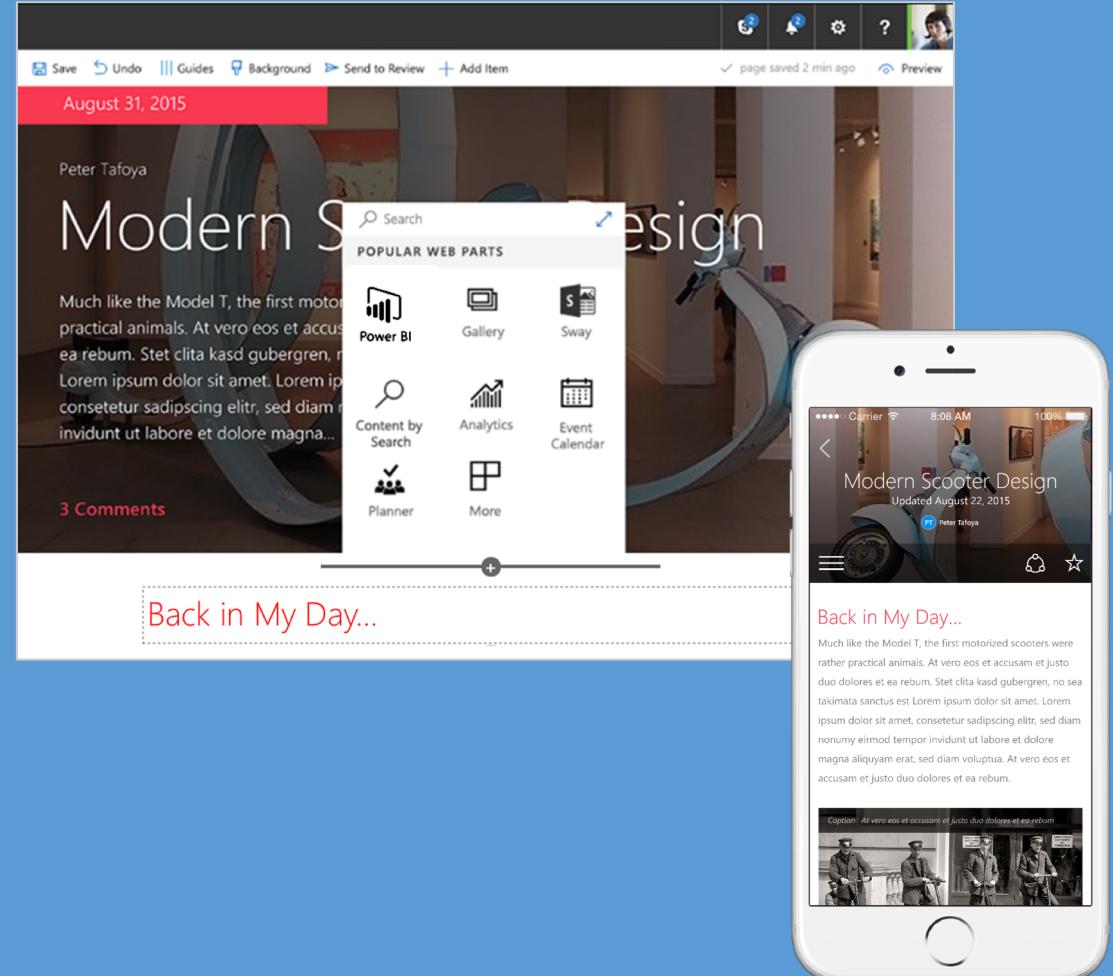
# SharePoint Framework

- Microsoft-recommended, client-side framework available in SharePoint Online and SharePoint Server 2016 (Feature Pack 2)
- Build applications using the latest JavaScript frameworks including Angular and React
- Use tools: NodeJS, NPM, Gulp, Yeoman, TypeScript



# Scenarios

- Enhancing the out of the box experience (custom property panes)
- Branding custom portals
- Building custom Intranet Portals
- Building custom Web Parts
- Building line of business applications
- Integrating with external systems



# Best Practices

---

# General Guidance

## Branding

Branding for Office 365, avoid custom master page, write CSS and JavaScript to update branding  
OOTB UI Tweaks using CSS

## Customizations

Prioritize using OOTB web parts vs. creating custom web parts (generally for collaboration sites)  
Follow web best practices for building SharePoint customizations  
Avoid SharePoint Designer (maybe workflows?)  
No sandbox solutions or full-trust solutions  
Use Office 365 Graph or SharePoint REST API for accessing content

## Office 365 PNP

Office 365 Patterns and Practices for development recommendations and examples

Office 365 | SharePoint

All Projects

107 TOTAL PROJECTS

18 RED PROJECTS

48 YELLOW PROJECTS

41 GREEN PROJECTS

Name	Overall	Budget	Resource	Schedule	Project Manager	Modified	Edit
alpha-chi	●	△	●	▲	Maddie Pietrek	Jan 8, 2016 7:51:40 AM	<input checked="" type="checkbox"/>
alpha-beta-omega	●	●	●	●	Laura Pietrek	Oct 19, 2015 1:59:14 PM	<input checked="" type="checkbox"/>
alpha-omega	●	●	●	▲	Laura Pietrek	Jun 13, 2015 12:59:45 PM	<input checked="" type="checkbox"/>
alpha-zeta	●	●	●	●	Laura Pietrek	Jun 13, 2015 11:10:06 AM	<input checked="" type="checkbox"/>
alphas-violet	●	△	●	▲	Pietrek	Jun 5, 2015 10:48:27 PM	<input checked="" type="checkbox"/>
alpha-psi	●	●	●	●	Maddie Pietrek	Jun 5, 2015 10:32:54 PM	<input checked="" type="checkbox"/>
Omega	●	●	●	▲	Steve Pietrek	Apr 14, 2015 12:44:37 PM	<input checked="" type="checkbox"/>
zalir	●	●	●	●	Lenora Lula Owen	Apr 14, 2015 8:39:25 AM	<input checked="" type="checkbox"/>
za	●	●	●	●	Harriett Elizabeth Fleming	Apr 14, 2015 8:39:25 AM	<input checked="" type="checkbox"/>
valabiaza	●	●	●	●	Anne Emilie Mendoza	Apr 14, 2015 8:39:25 AM	<input checked="" type="checkbox"/>

Filter results

Page 1 of 11, of 107 entries

## Projects

+ new item or edit this list

All Items ... Find an item

✓	Title	OverallStatus	BudgetStatus	ResourceStatus	ScheduleStatus	ProjectManager	PlannedEndDate	Modified
	alpha-chi	...	R	Y	R	Y	Maddie Pietrek	11/29/2015 January 8, 2016
	alpha-beta-omega	...	R	G	R	R	Laura Pietrek	6/25/2015 October 19, 2015
	alpha-omega	...	R	R	R	Y	Laura Pietrek	7/23/2015 June 13, 2015
	alpha-zeta	...	R	R	G	G	Laura Pietrek	6/18/2015 June 13, 2015
	alphas-violet	...	R	Y	R	Y	Pietrek	7/2/2015 June 5, 2015
	alpha-psi	...	R	R	G	R	Maddie Pietrek	6/25/2015 June 5, 2015
	Omega	...	R	G	R	Y	Steve Pietrek	April 14, 2015
	fahzaj	...	Y	G	Y	G	Duane Leon Armstrong	April 14, 2015
	za	...	G	G	G	G	Harriett Elizabeth Fleming	April 14, 2015
	iwcebur	...	Y	Y	G	G	Austin Matthew Beck	April 14, 2015
	cud	...	G	G	G	G	Jimmy Paul Reid	April 14, 2015
	valabiaza	...	G	G	G	G	Anne Emilie Mendoza	April 14, 2015

# Content: List and Library Driven

# Resources

- ES6 - <http://es6-features.org/>
- TypeScript Playground -  
<https://www.typescriptlang.org/play/>
- React - <https://reactjs.org/>
- AngularJS - <https://angularjs.org/>
- Vue.js - <https://vuejs.org/>
- SharePoint / Office 365 Developer Patterns and Practices - <https://github.com/SharePoint/PnP>
- SharePoint Framework -  
<https://github.com/SharePoint/sp-dev-docs>
- Material-UI - <http://www.material-ui.com/>
- Material UI Icons - <https://material.io/icons/>
- Office Fabric - <https://dev.office.com/fabric>

# Our Sponsors



**RIGHTPOINT**

