

JavaScript Insights

Tools for Improving JS Code Quality



Ann Robson
@arobson



Ariya Hidayat
@ariyahidayat



HTML5 Developer Conference
San Francisco, October 22, 2013



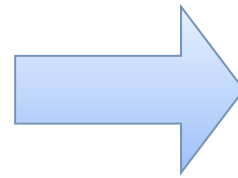
Ariya



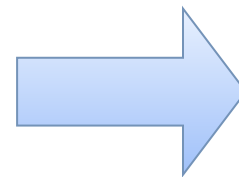
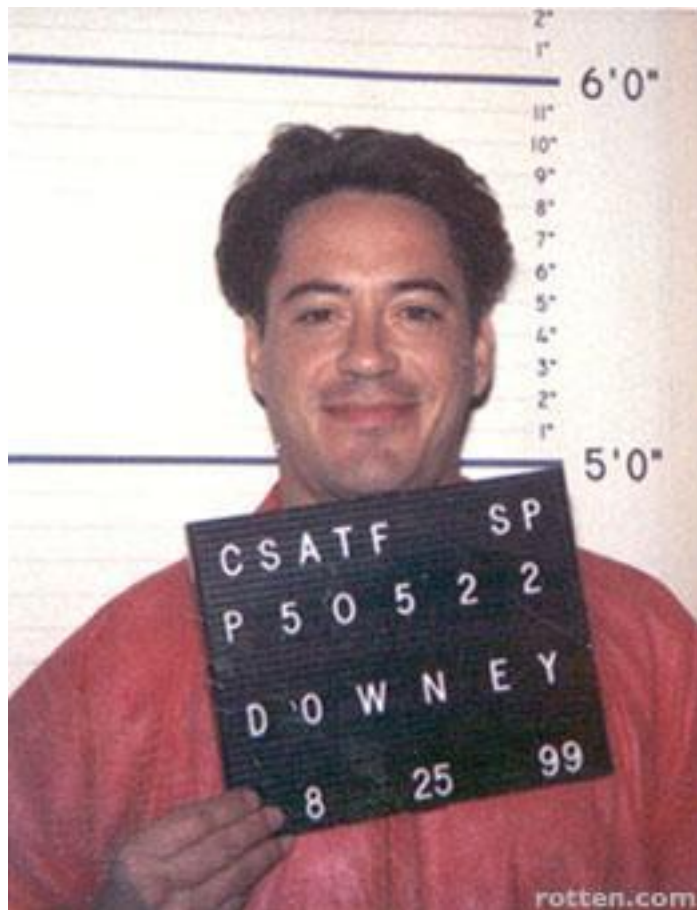
Ann

Why JavaScript code quality tools?

It's how we work together.

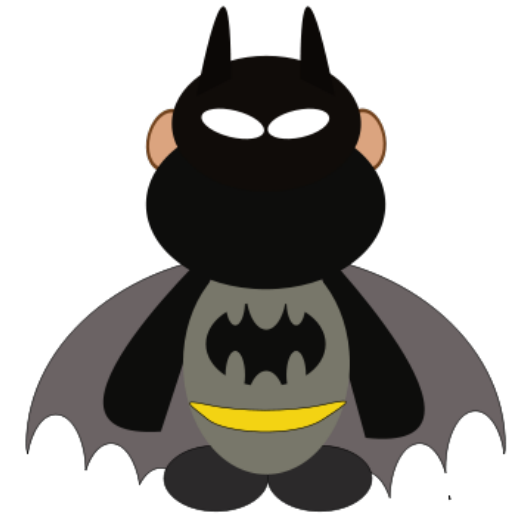


And it's how we level up.



JavaScript Tools

varied... some to be developed.... ALL are open source



Power(ful) tools

Lint

Code Coverage

Cyclomatic Complexity

Lint

Enforces style guide

Team standards: `.jshintrc`

JSHint <http://jshint.com>



Code Coverage

Code coverage report for **feeds/lib/ui/chat/**

Statements: **74.08%** (1229 / 1659) Branches: **60.9%** (500 / 821) Functions: **74.77%** (249 / 333) Lines: **74.36%** (1221 / 1642)

[All files](#) » feeds/lib/ui/chat/

File ^	Statements ^	Branches ^	Functions ^	Lines ^
chat_add_member.js	54.24% (64 / 118)	34.00% (17 / 50)	60.87% (14 / 23)	54.31% (63 / 116)
chat_aggregator.js	74.19% (23 / 31)	83.33% (5 / 6)	75.00% (9 / 12)	73.33% (22 / 30)
chat_manager.js	65.05% (201 / 309)	54.17% (78 / 144)	54.84% (34 / 62)	66.11% (199 / 301)
chat_message_list.js	81.32% (74 / 91)	75.56% (34 / 45)	84.62% (22 / 26)	81.32% (74 / 91)
chat_message_list_item.js	55.77% (29 / 52)	53.85% (21 / 39)	69.23% (9 / 13)	55.77% (29 / 52)
chat_session.js	75.37% (358 / 475)	65.02% (184 / 283)	75.95% (60 / 79)	75.58% (356 / 471)
chat_session_notifications.js	98.21% (110 / 112)	90.63% (58 / 64)	100.00% (15 / 15)	98.20% (109 / 111)
chat_settings_menu.js	90.24% (37 / 41)	75.00% (12 / 16)	90.91% (10 / 11)	90.24% (37 / 41)
chat_window.js	94.00% (47 / 50)	83.33% (20 / 24)	91.67% (11 / 12)	94.00% (47 / 50)
contact_list.js	65.78% (148 / 225)	43.33% (52 / 120)	66.67% (24 / 36)	65.78% (148 / 225)
contact_list_item.js	80.70% (46 / 57)	66.67% (8 / 12)	100.00% (7 / 7)	80.70% (46 / 57)
emoticon.js	100.00% (20 / 20)	100.00% (0 / 0)	100.00% (3 / 3)	100.00% (20 / 20)
feed_messages_delegate.js	92.31% (12 / 13)	100.00% (2 / 2)	87.50% (7 / 8)	91.67% (11 / 12)
group_chat_session.js	94.74% (18 / 19)	50.00% (1 / 2)	88.89% (8 / 9)	94.74% (18 / 19)
recent_conversations.js	89.66% (26 / 29)	33.33% (2 / 6)	88.89% (8 / 9)	89.66% (26 / 29)
thread_messages_delegate.js	94.12% (16 / 17)	75.00% (6 / 8)	100.00% (8 / 8)	94.12% (16 / 17)

Istanbul <http://gotwarlost.github.io/istanbul/>

<http://ariya.ofilabs.com/2012/12/javascript-code-coverage-with-istanbul.html>

*If you think JSLint hurts your feelings,
wait until you use Istanbul*

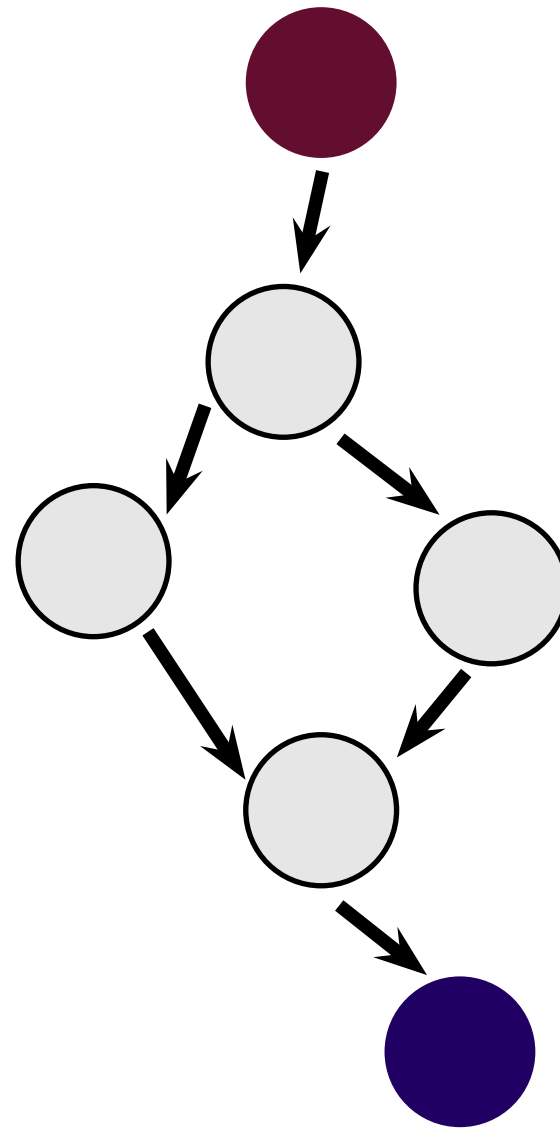


@davglass from Yahoo!

Code Complexity

```
if (true) "foo"; else "bar";
```

Cyclomatic Complexity = **2**



Control Flow Graph

6 edges

6 nodes

1 exit

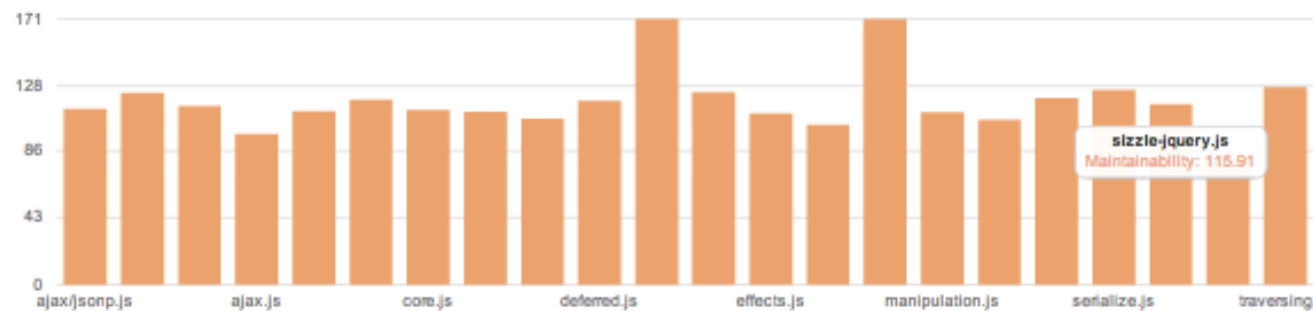
Complexity Visualization

Summary

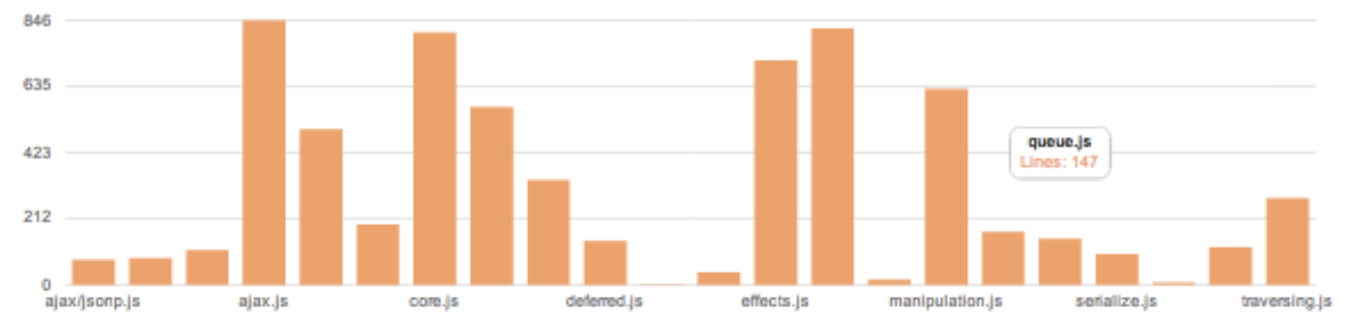
Total SLOC
6715

Average Maintainability
117.78

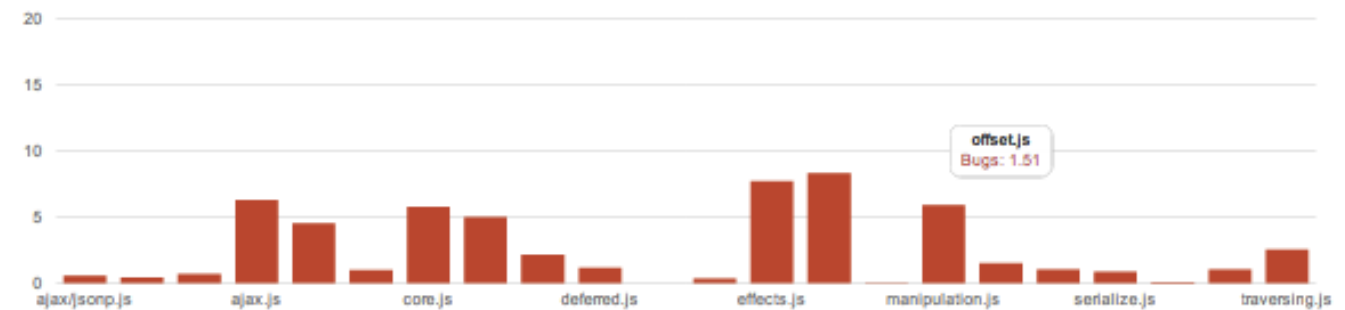
Maintainability



Lines of code



Estimated number of bugs



Plato

<https://github.com/jsoverson/plato>

Continuous Delivery for JavaScript Applications

Jarrod Overson

Room E-131 Wed 11.40am

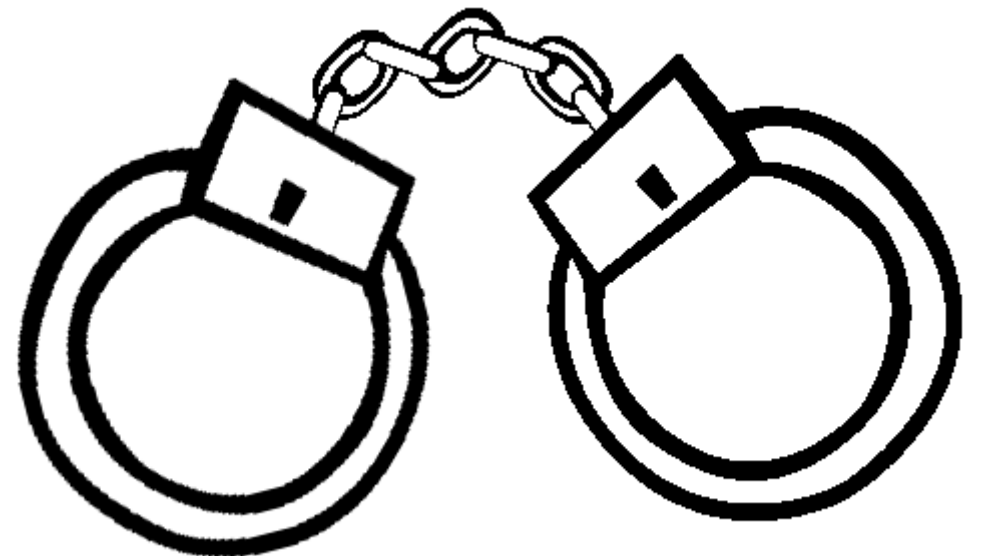
Pre-commit Hooks

May include all tools previously mentioned:

- linting
- code coverage
- code complexity

Also,

- validation: *esvalidate*
- custom



Demos

Hard Thresholds on Code Coverage

istanbul check-coverage --statement -5 --branch -3 --function 100

```
Tests: 19   Failures: 0
=====
Writing coverage object [/home/ariya/dev/esrefactor/coverage/coverage.json]
Writing coverage reports at [/home/ariya/dev/esrefactor/coverage]
=====

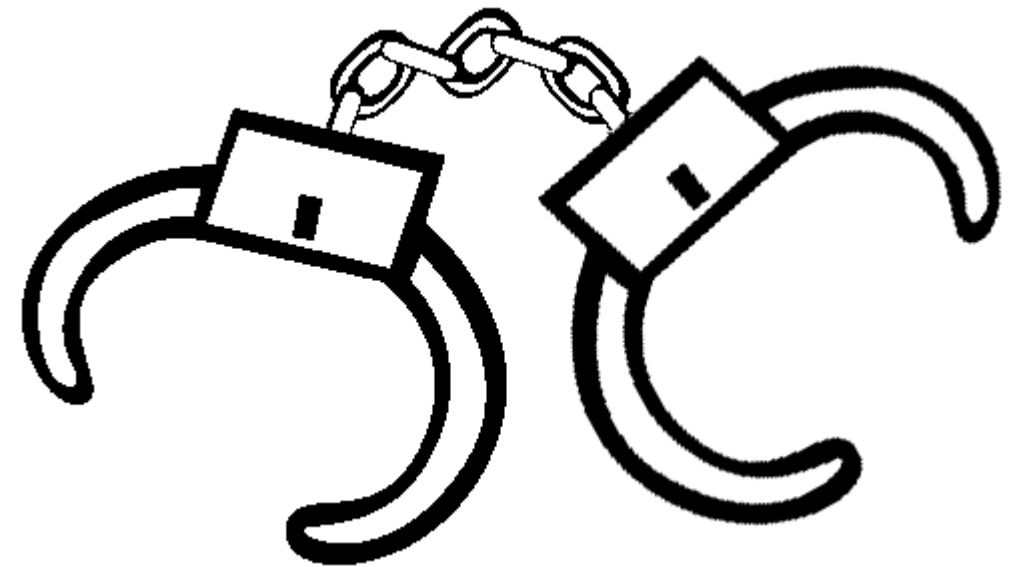
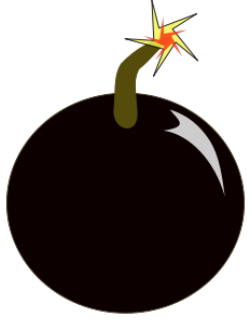
===== Coverage summary =====
Statements   : 98.13% ( 105/107 )
Branches     : 93.55% ( 58/62 )
Functions    : 100% ( 12/12 )
Lines        : 98.11% ( 104/106 )
=====

> istanbul check-coverage --branch -3

/usr/local/node/lib/node_modules/istanbul/lib/cli.js:31
    throw ex; // turn it into an uncaught exception
    ^
ERROR: Uncovered count for branches (4) exceeds threshold (3)
```

Pre-commit Hooks Escape!

git commit -n



Composable Tools

Stray logging

Strict mode violations

Unused variables

Nested ternary conditionals

Polluting variables

Boolean traps

Code → Syntax Tree

```
var answer = 42;
```

```
{
  type: "Program",
  body: [
    {
      type: "VariableDeclaration",
      declarations: [
        {
          type: "VariableDeclarator",
          id: {
            type: "Identifier",
            name: "answer"
          },
          init: {
            type: "Literal",
            value: 42,
            raw: "42"
          }
        }
      ],
      kind: "var"
    }
  ]
}
```

console.log? debugger?

- ▼ CallExpression
 - ▼ callee
 - ▼ MemberExpression
 - computed: false
 - ▼ object
 - ▼ Identifier
 - name: console
 - ▼ property
 - ▼ Identifier
 - name: log
 - ▼ arguments [1]
 - ▼ Identifier
 - name: answer

```
function detect_console(code) {  
    function check(node) {  
        if (node.type === 'CallExpression') {  
            if (node.callee.type === 'MemberExpression') {  
                if (node.callee.object.name === 'console') {  
                    alert('console call at line', node.loc.start.  
line);  
                }  
            }  
        }  
    }  
  
    var tree = esprima.parse(code, { loc: true });  
    estraverse.traverse(tree, { enter: check });  
}
```

Strict Mode Validator

```
'use strict';  
block = {  
  color: 'blue',  
  height: 20,  
  width: 10,  
  color: 'red'  
};
```

Duplicate data property in object literal **not** allowed in strict mode

Polluting Variables

```
var height;  
// some fancy processing  
height = 200;
```

Leaks to
global

<https://github.com/kesla/node-leaky>

```
test.js:3  
height = 200;  
^  
LeakError: global leak  
detected: height
```

Unused Variables

Declared but
not used

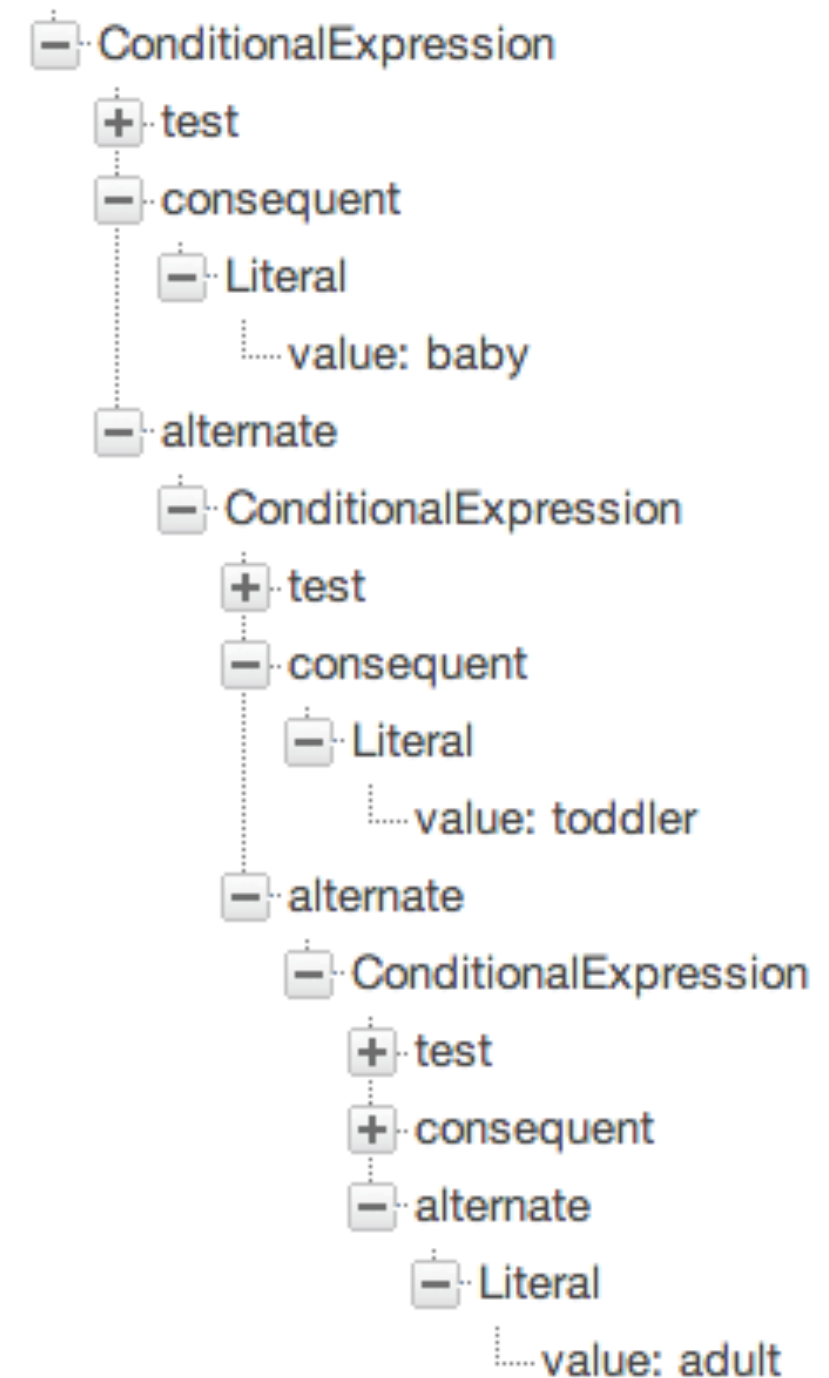
```
var height;  
// some fancy processing  
height = 200;
```

<https://github.com/Kami/node-unused>

```
test.js  
1      height - on line
```

Nested Ternary Conditionals

```
var str = (age < 1) ? "baby" :  
  (age < 5) ? "toddler" :  
  (age < 18) ? "child" : "adult";
```



“Boolean Trap” Finder

Obfuscated choice

```
var volumeSlider = new Slider(false);
```

Double-negative

```
component.setHidden(false);  
filter.setCaseInsensitive(false);
```

Can you make up your mind?

```
treeItem.setState(true, false);
```

The more the merrier

```
event.initKeyEvent("keypress", true, true,  
null, null, false, false, false, false, 9, 0);
```

Future linting

How we might collect the tools we make

ESLint <https://github.com/nzakas/eslint>

Multi-layered Defense



BUT WE ALREADY HAVE JENKINS...



ROBIN, THE EARLIER THE BETTER

Multi-Layer Defense

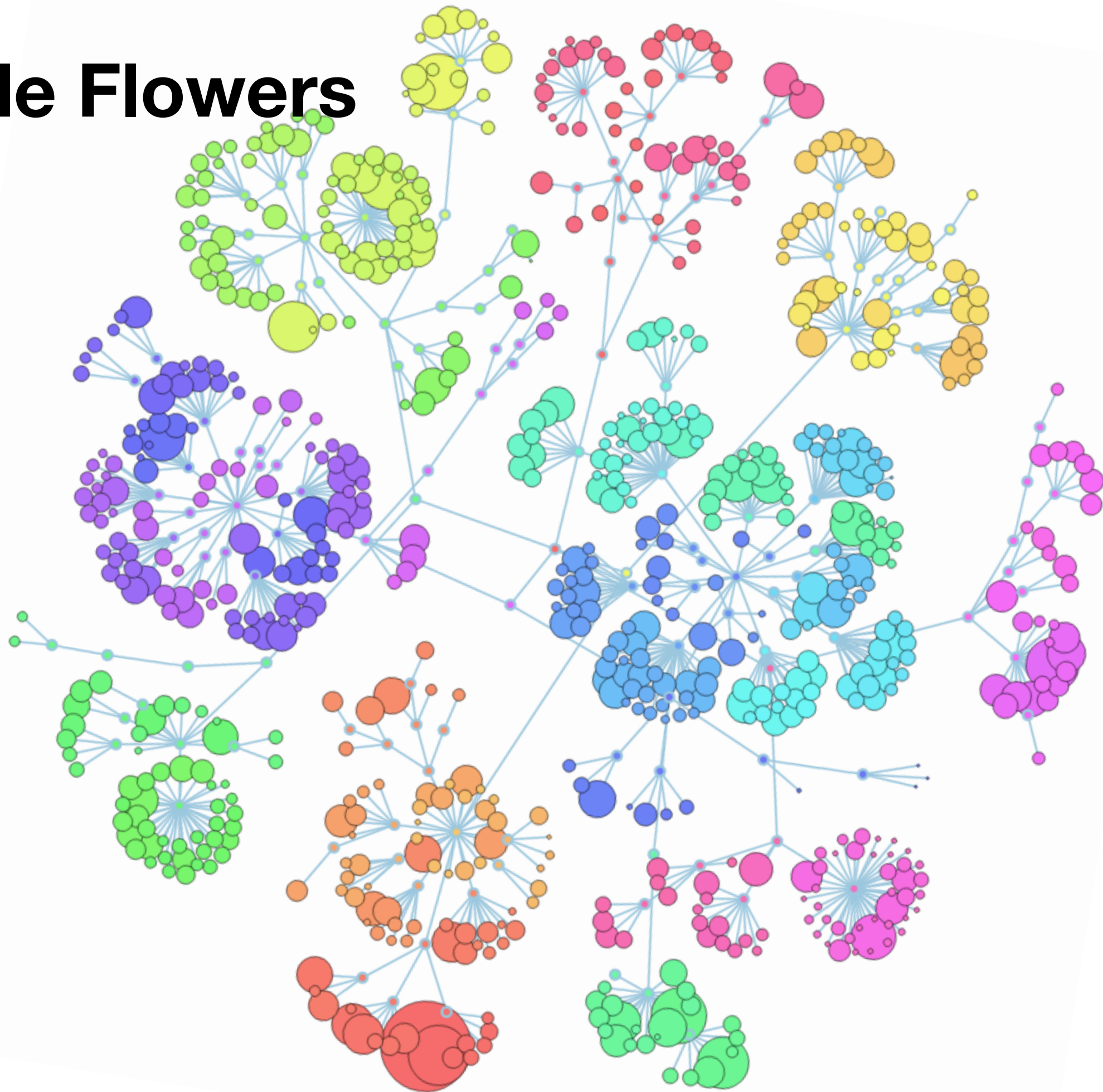
Editor: syntax validation, linting

Pre-commit hooks: validation, linting, unit tests, code complexity thresholds,
code coverage minimum

CI Server: validation, linting, unit tests, code complexity thresholds, code
coverage minimum

The earlier the better.

Code Flowers



Application Structure

```
MyApp.create('MyApp.Person', {  
  name: 'Joe Sixpack',  
  age: 42,  
  
  constructor: function(name) {},  
  walk: function(steps) {},  
  run: function(steps) {}  
});
```



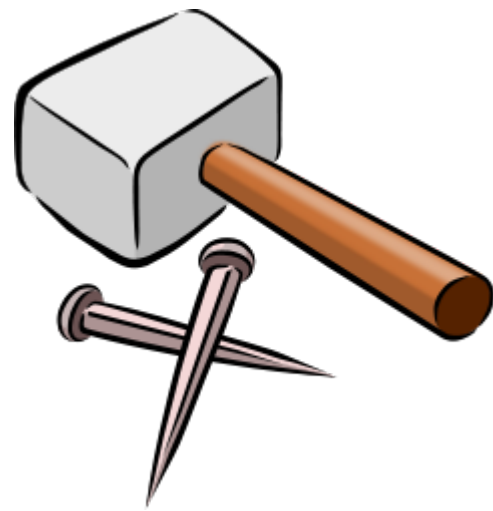
Metadata

```
{  
  objectName: 'MyApp.Person',  
  functions: ['walk', 'run'],  
  properties: ['name', 'age']  
}
```

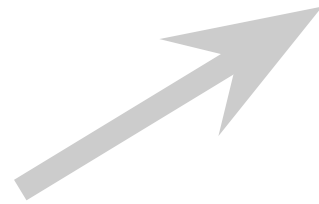
Final Words

Tools separate us

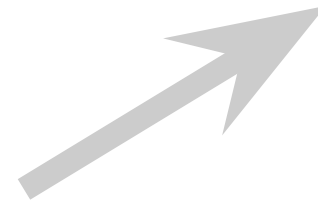




Good



Awesome



**Even more
awesome**



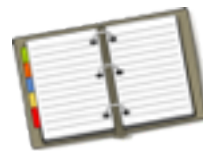
Thank you!!!

@arobson



@ariyahidayat

www.htmlhive.com



ariya.ofilabs.com



speakerdeck.com/ariya