

Testing Web Applications

presented by Seth McLaughlin [@sethmc](#)



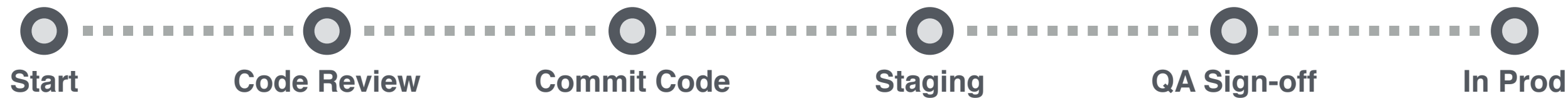
SH-PE
ENGINEERING

www.shapesecurity.com

testing is a **BIG** topic.

how to **effectively** test **JavaScript** code
in your **web applications**?

Workflow.



**Build me
feature X ASAP!**

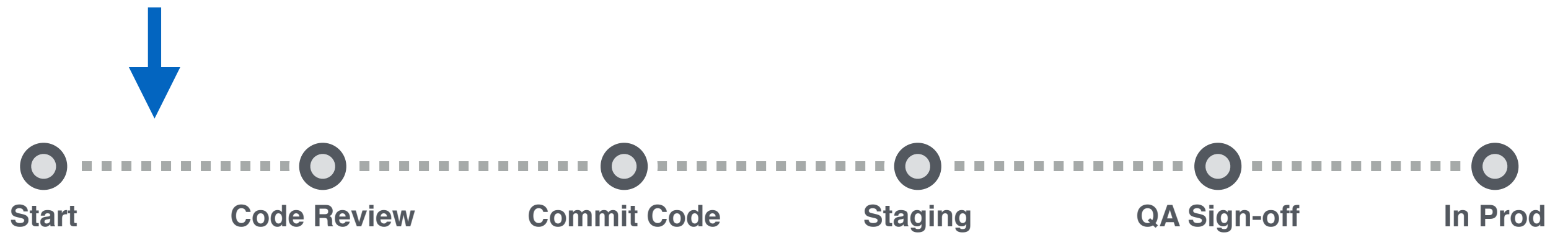


PM

Right away!

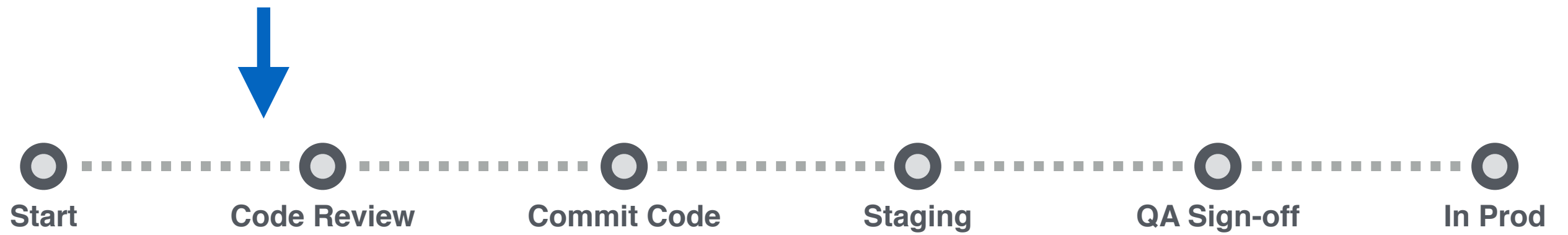


DEV



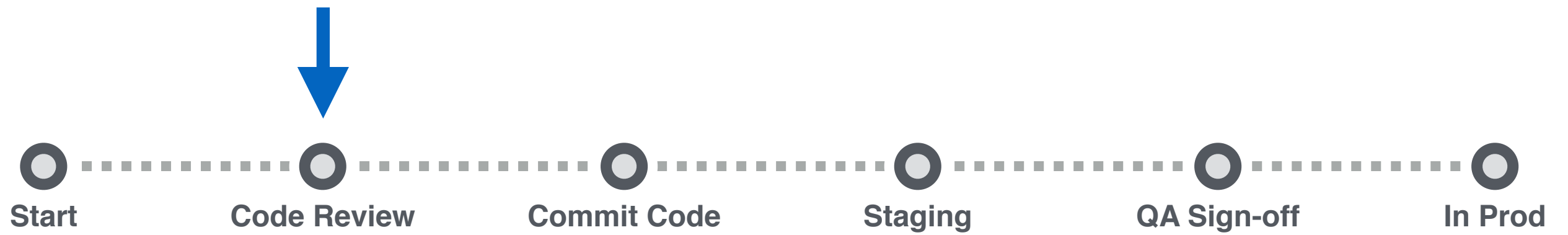
working for
the man...

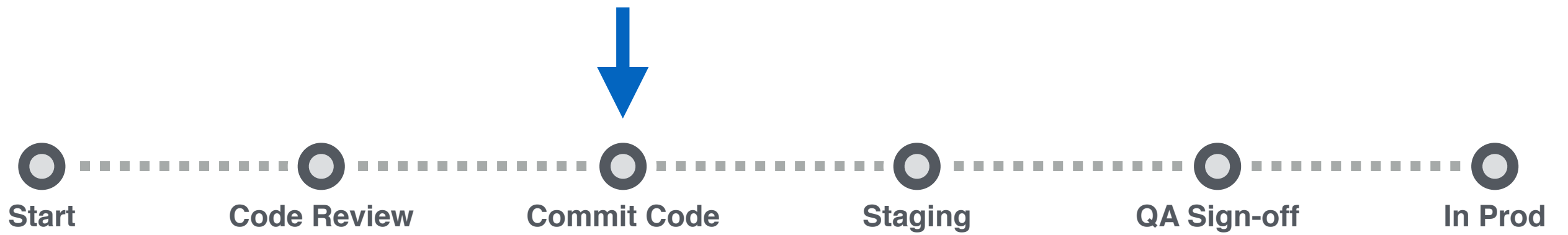


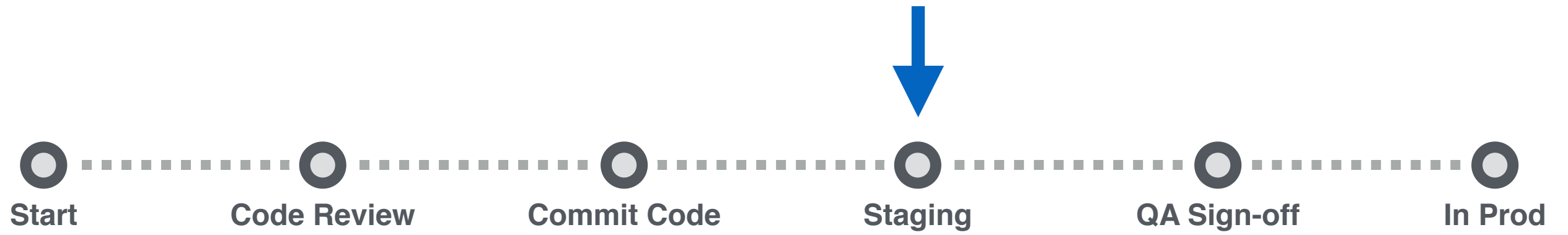


**every night
and day!**









**my work here
is done.**



DEV



**my work here
is done.**

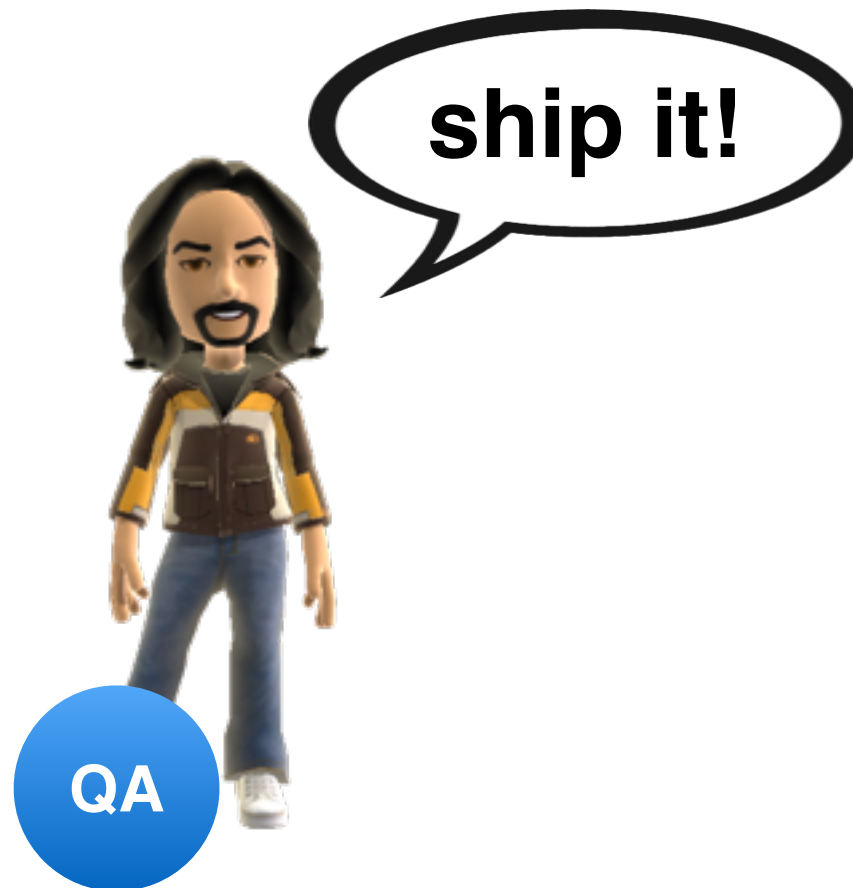
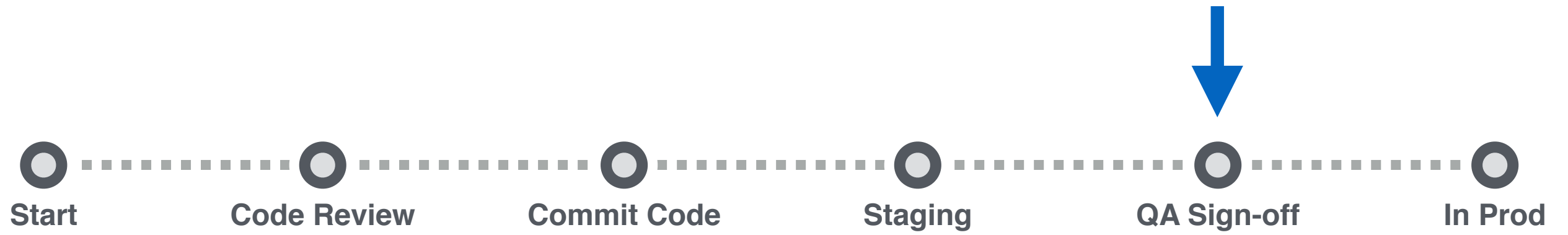


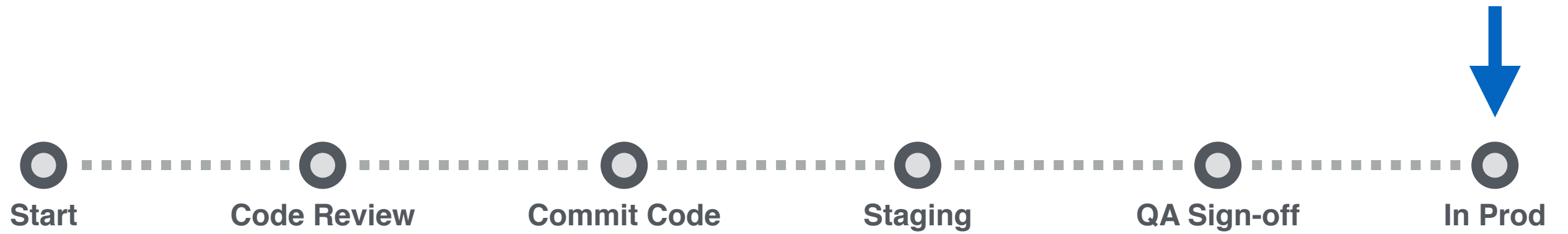
DEV

**we'll see
about that!**



QA





**feature X is
awesome!**

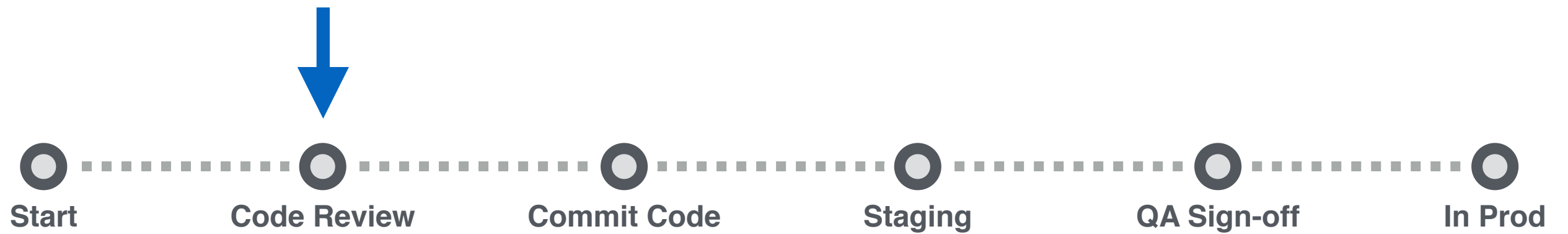


CUSTOMER



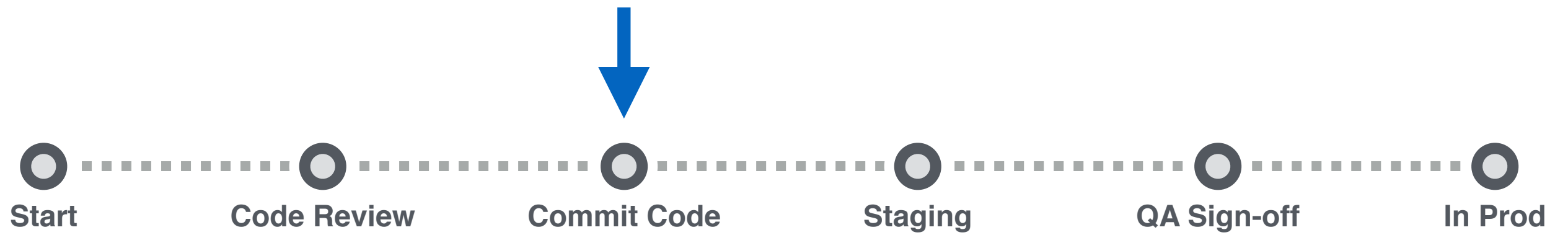
start coding!

elapsed time: **0** days



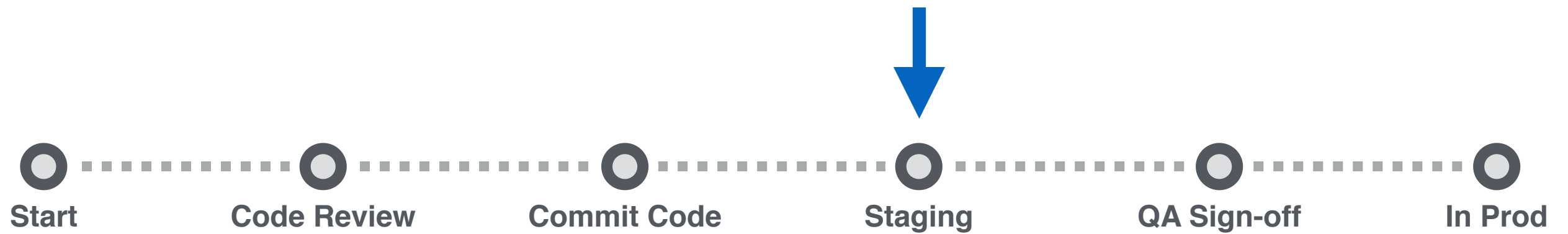
start code review

elapsed time: **7** days



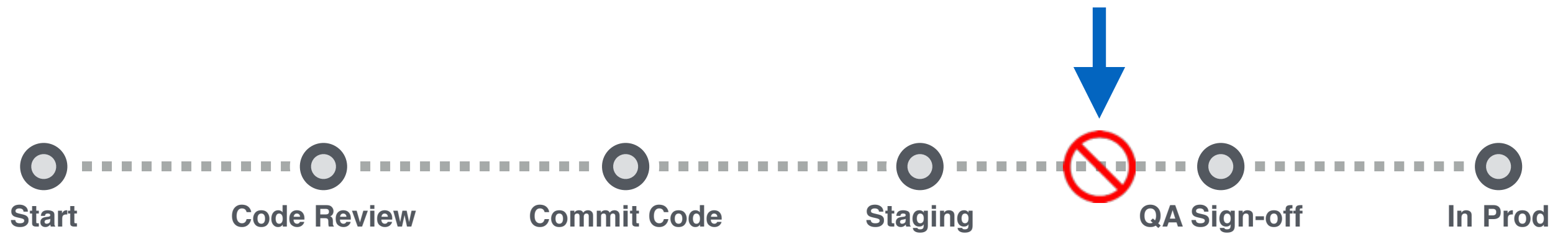
commit code

elapsed time: **9** days



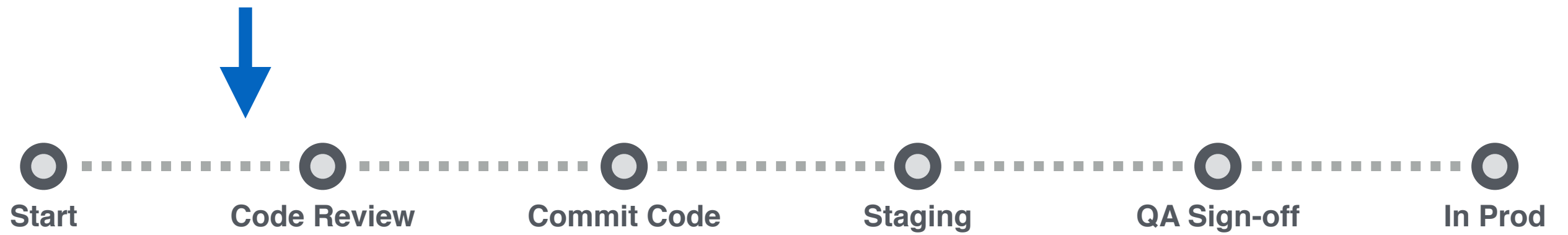
deploy code to staging

elapsed time: **10** days



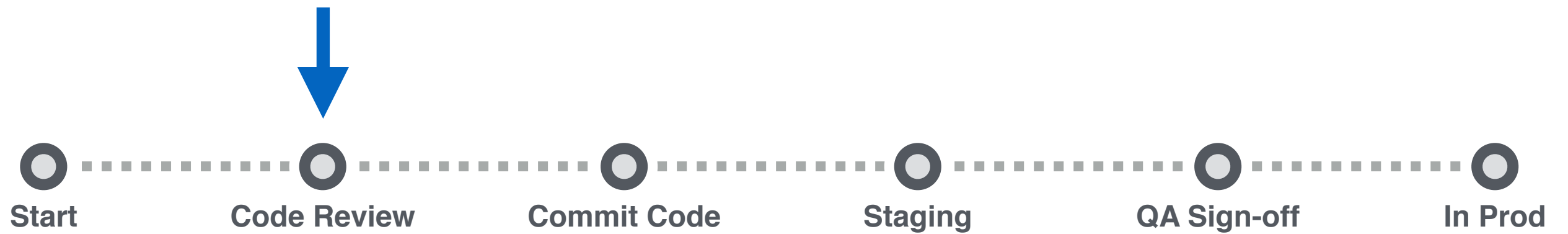
QA finds a bug in staging

elapsed time: **12** days



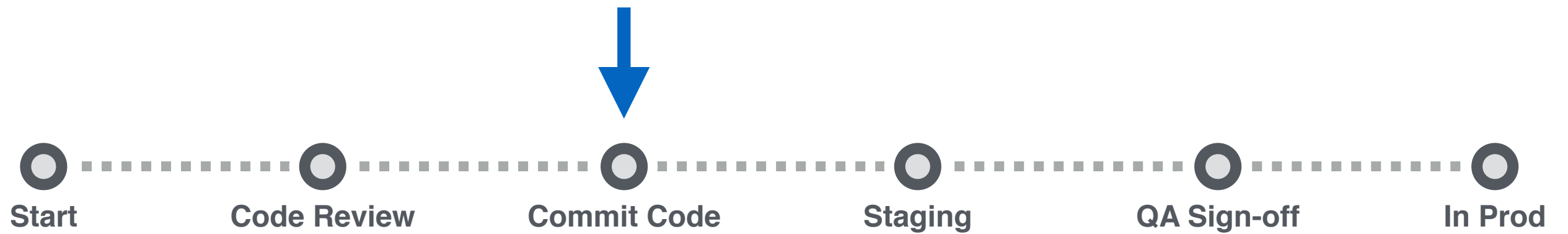
fix bug

elapsed time: **13** days



code review fix

elapsed time: **15** days



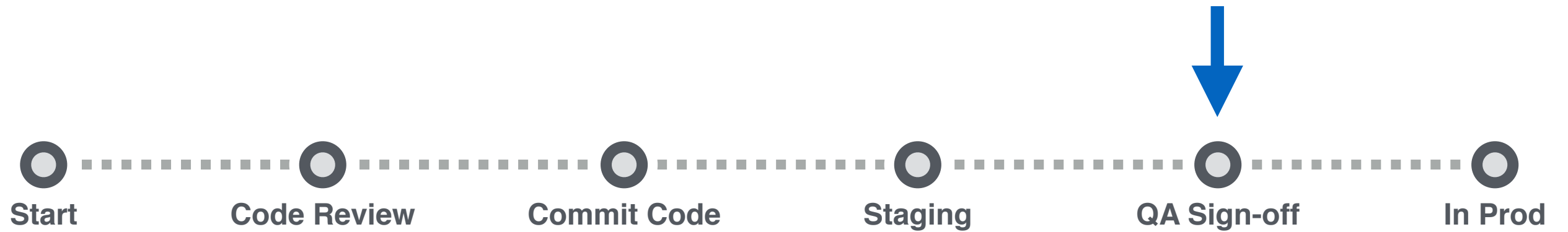
commit fixed code

elapsed time: **17** days



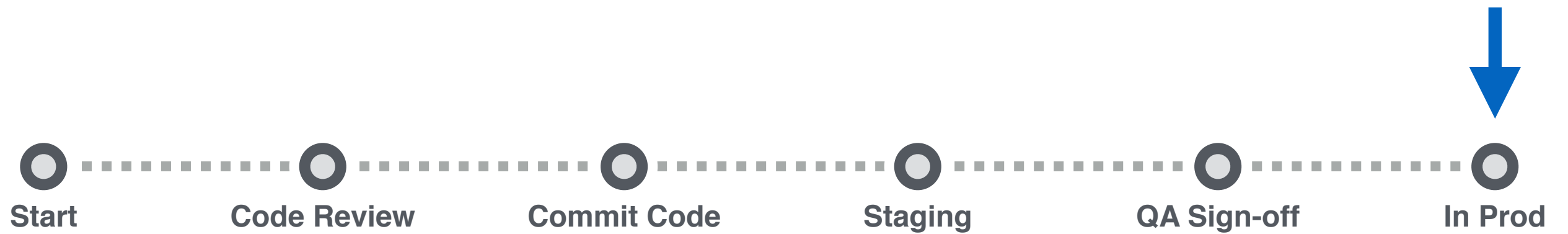
deploy (again) to staging

elapsed time: **17** days



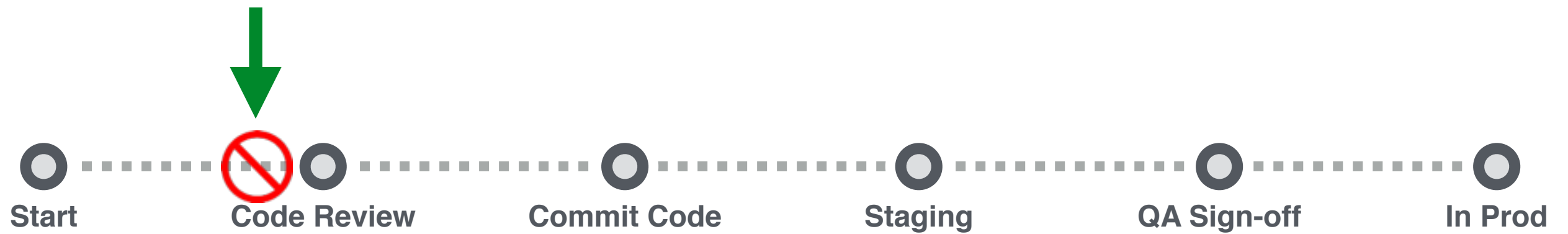
QA is happy

elapsed time: **19** days



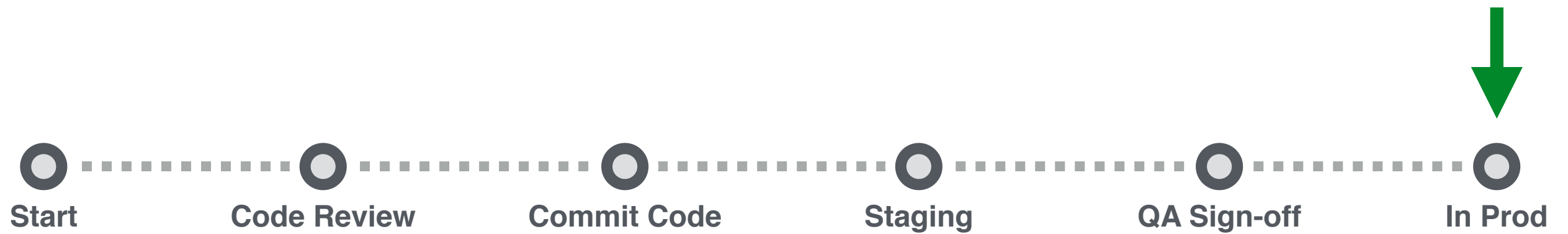
shipped to production

elapsed time: **20** days



Dev finds the bug while coding

elapsed time: **6** days

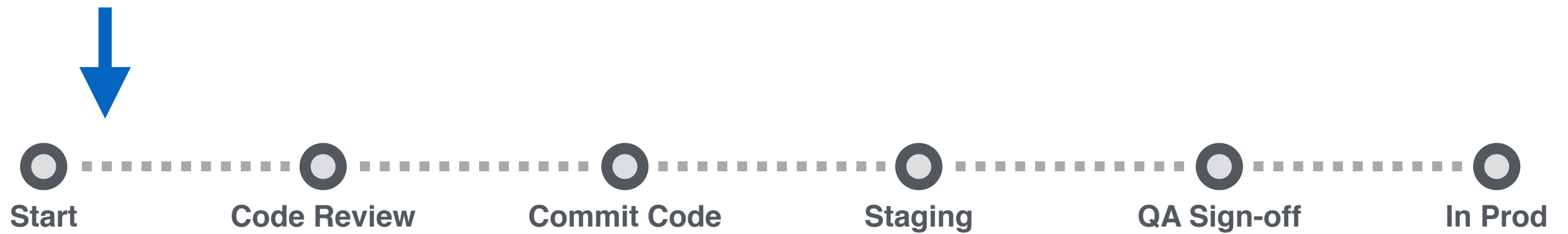


alternatively...

elapsed time: **15** days

could have saved over 5 days
by catching bug sooner

find bugs as early
as possible.

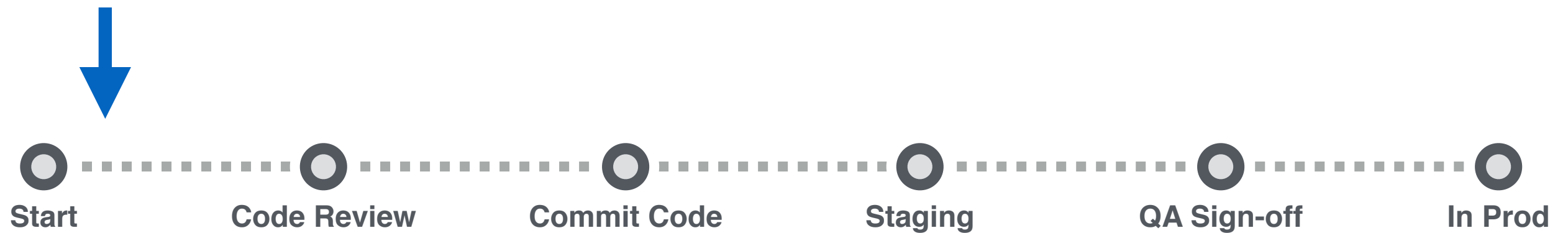


Static Analysis

```
var config = {  
  name: this.name,  
  style: 'solo',  
};
```

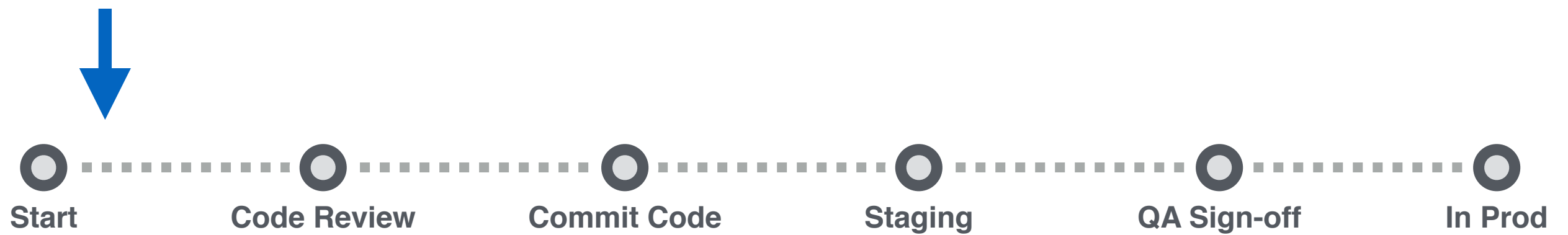
blows up in IE7





Static Analysis

```
function getObjs(paths) {  
  return paths.map(function (p) {  
    var obj = null;  
  
    try {  
      obj = require(p);  
    } catch (e) {  
      console.error('Cannot load path', p);  
    }  
  }).filter(function (obj) {  
    return obj !== null;  
  });  
}
```



Static Analysis

- Find common errors and enforce coding style
- Run every time you save a file! (automatically)
- Fast!

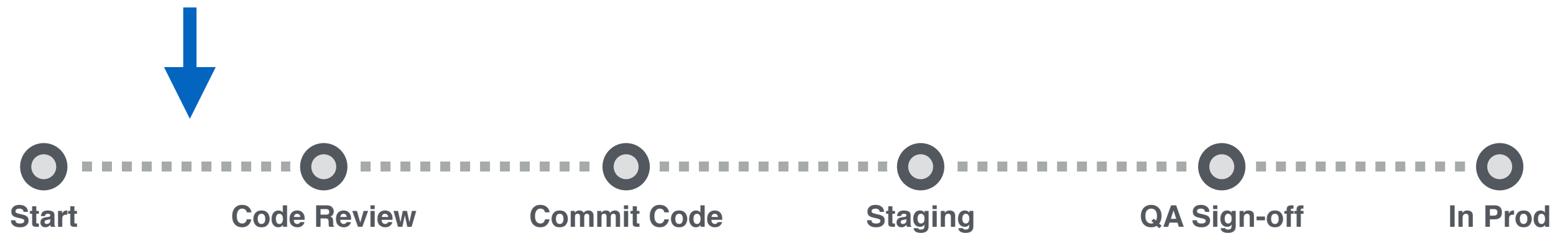


www.jshint.com



www.eslint.org





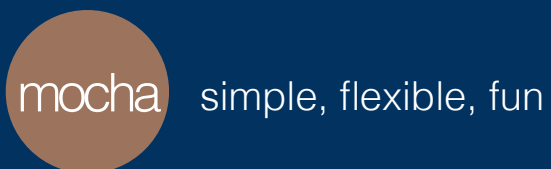
Unit Tests

- Test code at a granular level
- Test as you develop (TDD)
- Great traceability
- Fast! Fun!

I ♥ unit tests



DEV



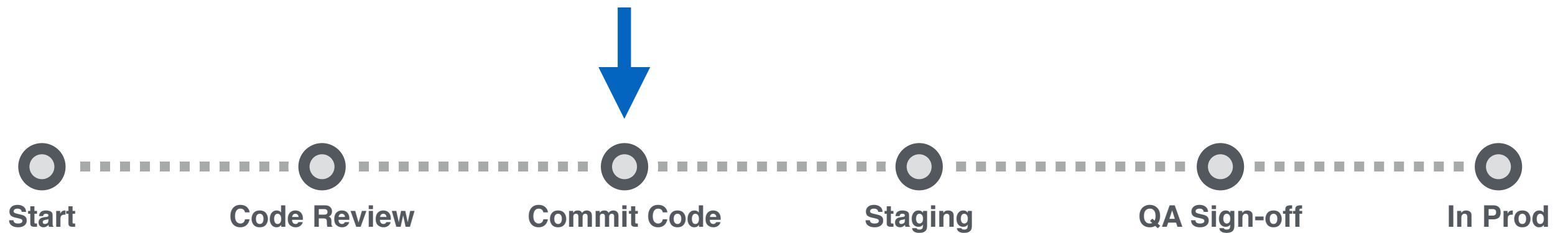
visionmedia.github.io/mocha



jasmine.github.io



www.qunitjs.com



Continuous Integration

Pre-Commit

example: run eslint and mocha tests

Post-Commit

example: run selenium tests



hudson-ci.org



jenkins-ci.org



atlassian.com



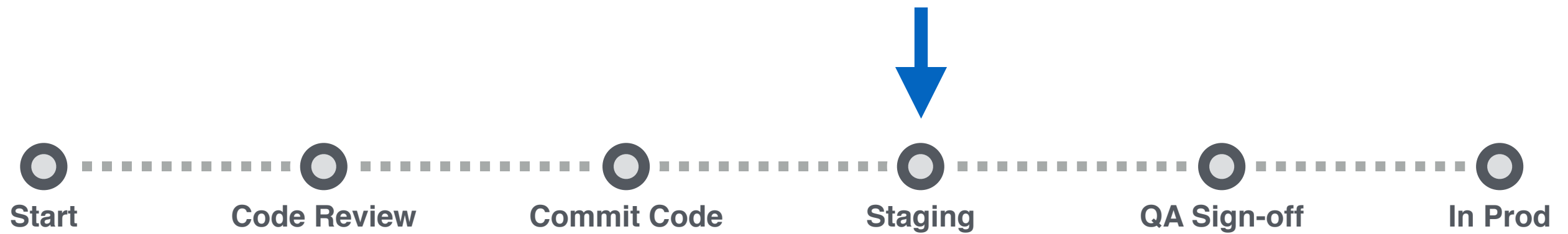
travis-ci.org



circleci.com



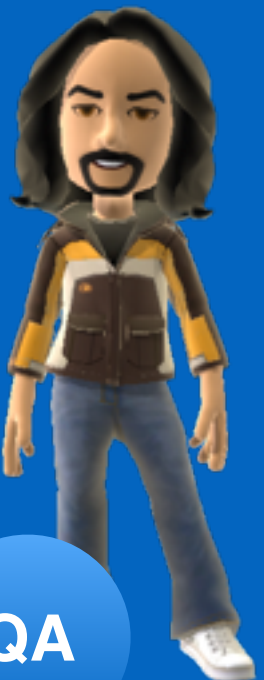
jetbrains.com



End to End Tests

- Scenario driven
- Tests client and server code
- More “realistic” vs. Unit Tests
- Less traceability vs. Unit Tests
- Slower to execute vs. Unit Tests

I ♥ selenium



Selenium

seleniumhq.org



saucelabs.com

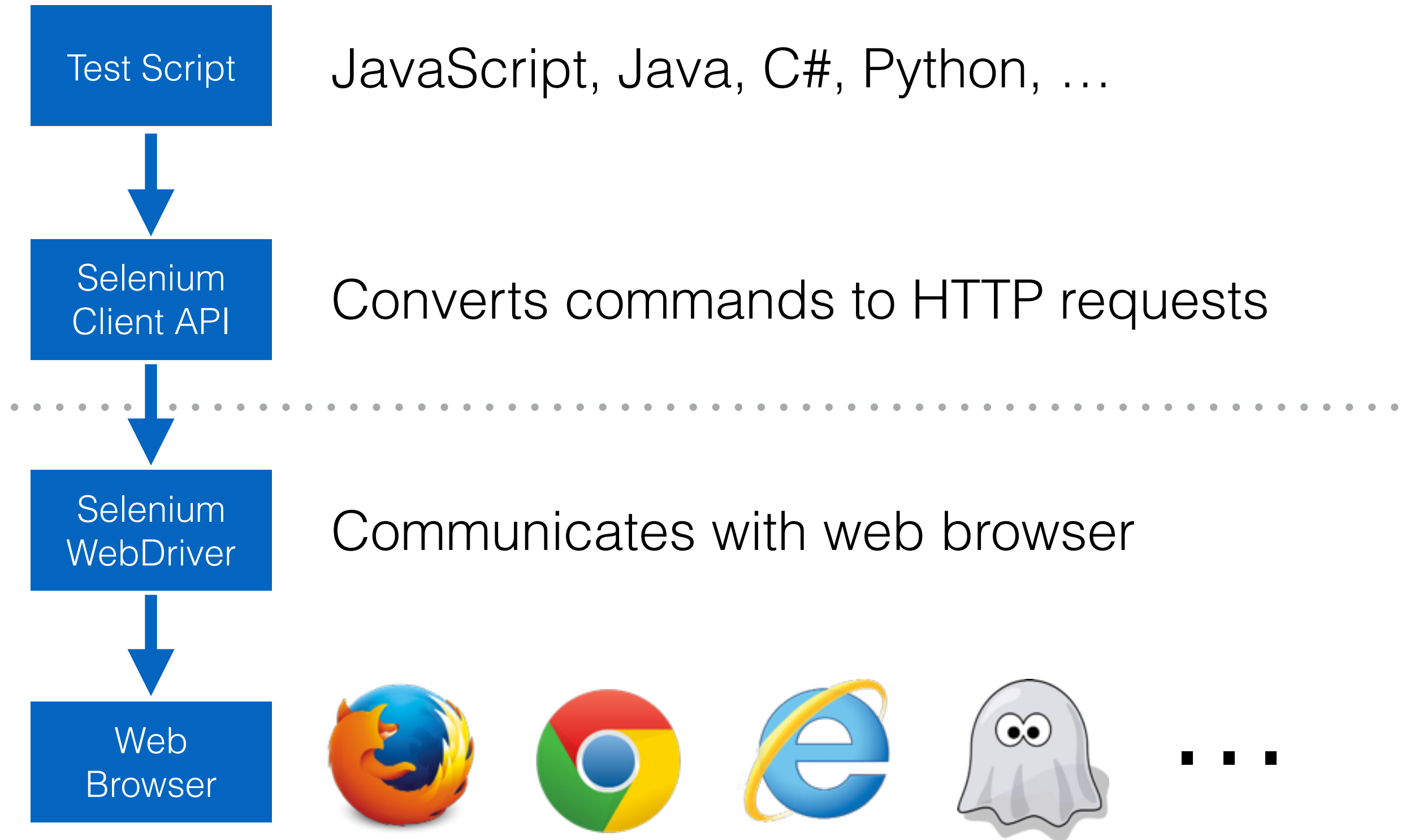


BrowserStack

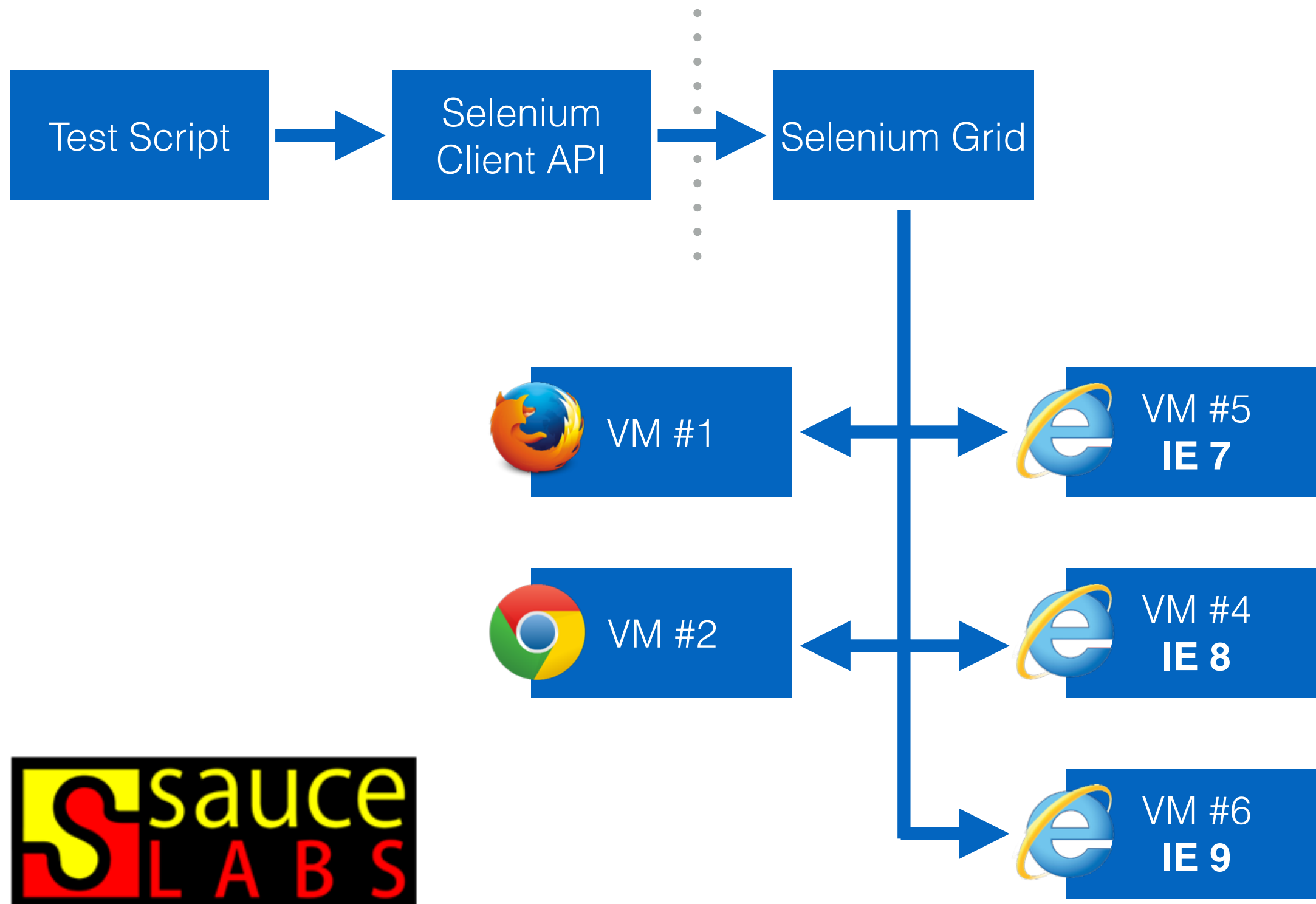
browserstack.com

Techniques.

End to End tests with Selenium



End to End tests with Selenium



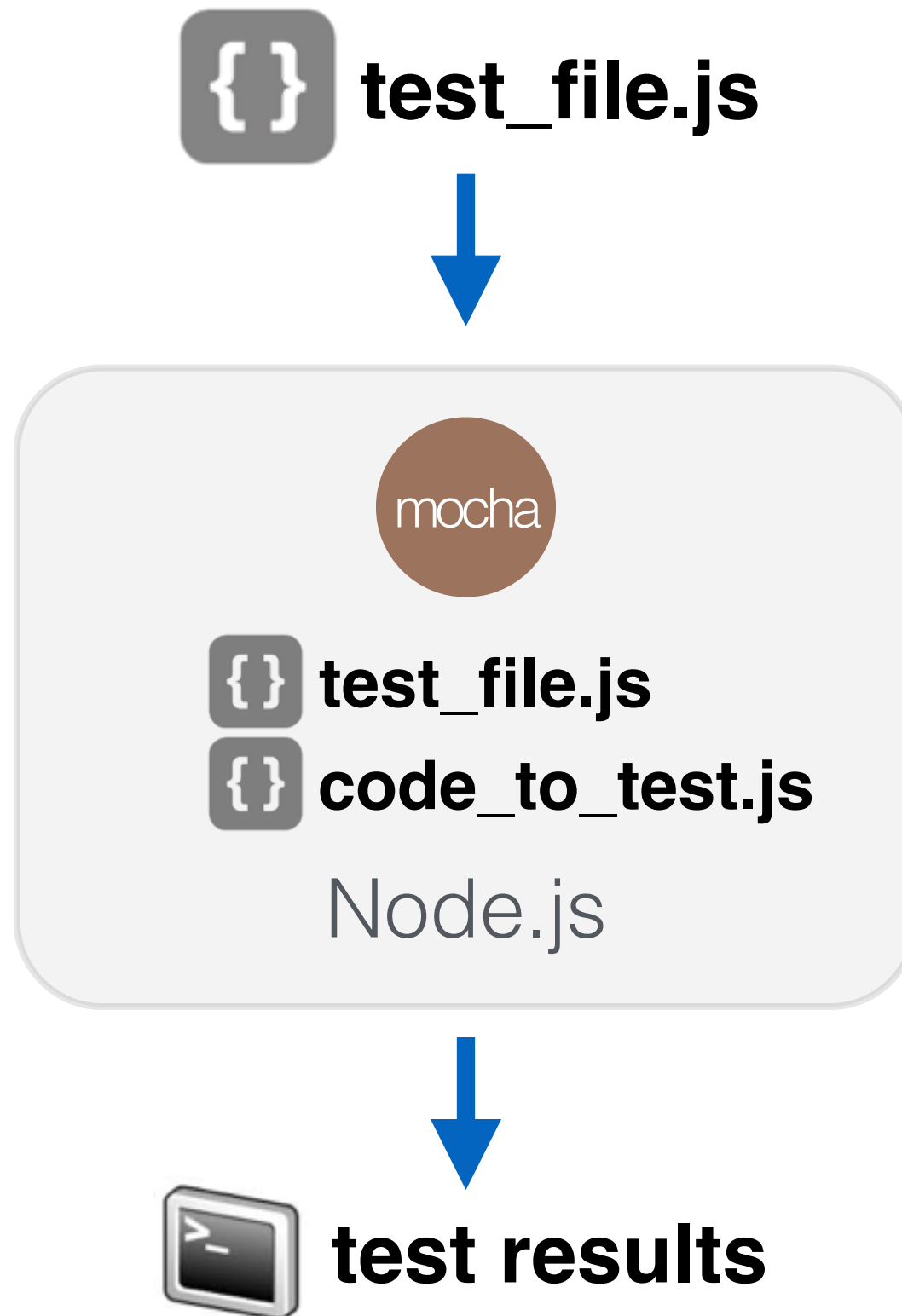
End to End tests with Selenium



Nightwatch.js

```
module.exports = {  
  'Get free VMs': function (browser) {  
    browser  
      .url('http://www.modern.ie')  
      .assert.title('Modern.IE')  
      .end();  
  }  
};
```

Unit Testing with Mocha (node.js)



Unit Testing with Mocha (node.js)

formatter.js

```
module.exports = {  
  name: function (title, first, last) {  
    return [title, first, last].join();  
  }  
};
```

Unit Testing with Mocha (node.js)



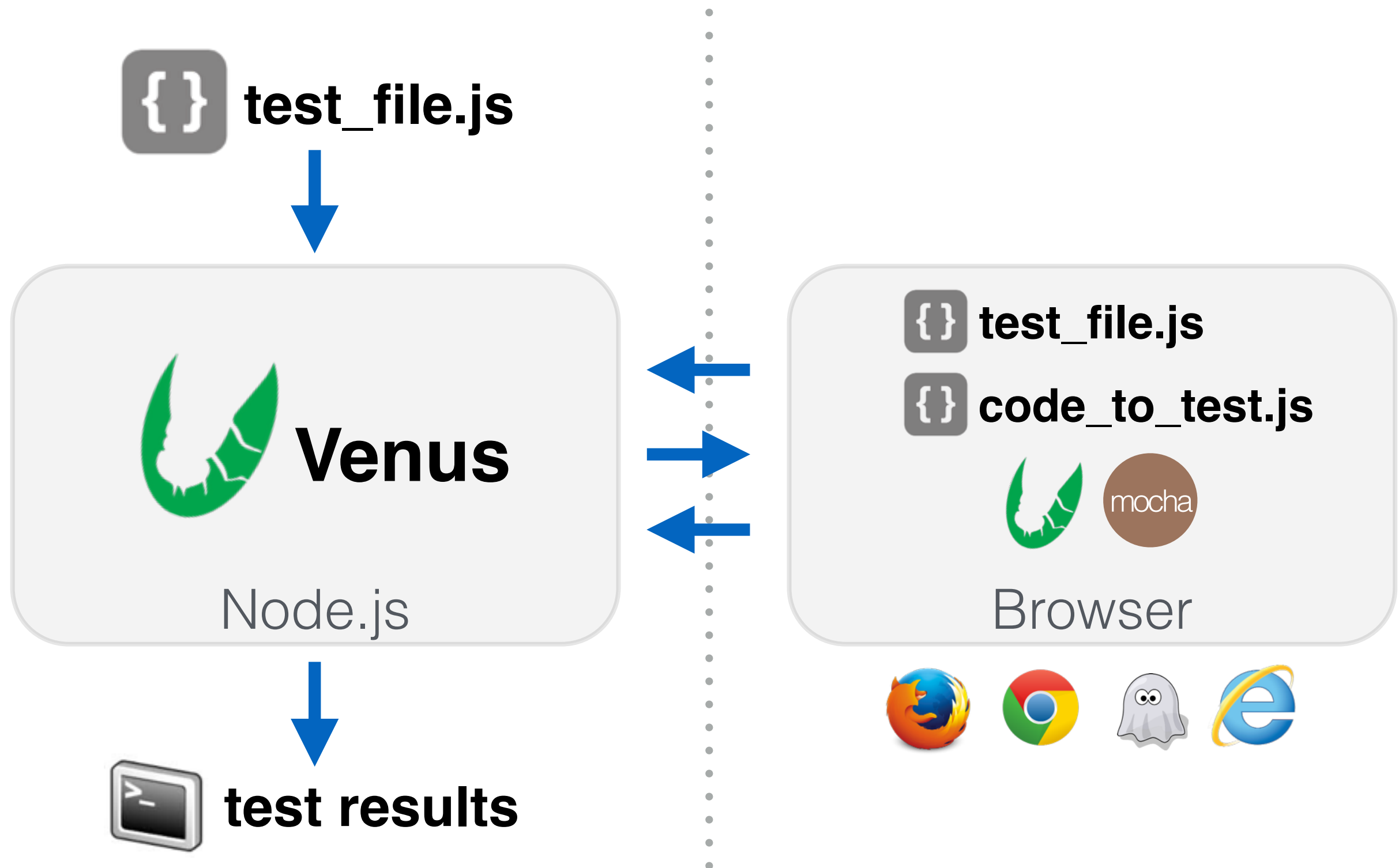
formatter.spec.js

```
var fmt      = require('../..../formatter');
var assert   = require('assert');

describe('format', function () {
  it('should return full name', function () {
    var actual    = fmt.name('Mr.', 'Bob', 'Rogers');
    var expected  = 'Mr. Bob Rogers';

    assert.equal(actual, expected);
  });
});
```

Unit Testing with Venus & Mocha (browser)



Unit Testing with Venus & Mocha (browser)

formatter.js

```
function formatName(title, first, last) {  
    return [title, first, last].join();  
}
```

Unit Testing with Venus & Mocha (browser)

formatter.spec.js

```
/**
 * @venus-library mocha
 * @venus-code formatter.js
 */
describe('format', function () {
  var actual    = formatName('Mr.', 'Bob', 'Rogers');
  var expected = 'Mr. Bob Rogers';

  it('should return full name', function () {
    expect(actual).to.be(expected);
  });
});
```


Unit Testing with Venus & Mocha (browser)



test_harness.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Test for Formatter</title>
  <script type="text/javascript" src="mocha.js"></script>
  <script type="text/javascript" src="venus_client.js"></script>
  <script type="text/javascript" src="formatter.js"></script>
  <script type="text/javascript" src="specs/formatter.spec.js"></script>
  <script type="text/javascript">
    testLibrary.run();
  </script>
</head>
<body>
  <div id="results"></div>
</body>
</html>
```

Example: Spy

A function that records information about how it is called.

```
var callback = sinon.spy();
```

```
PubSub.subscribe('message', callback);
```

```
PubSub.publishSync('message');
```

```
assert(callback.called === true);
```

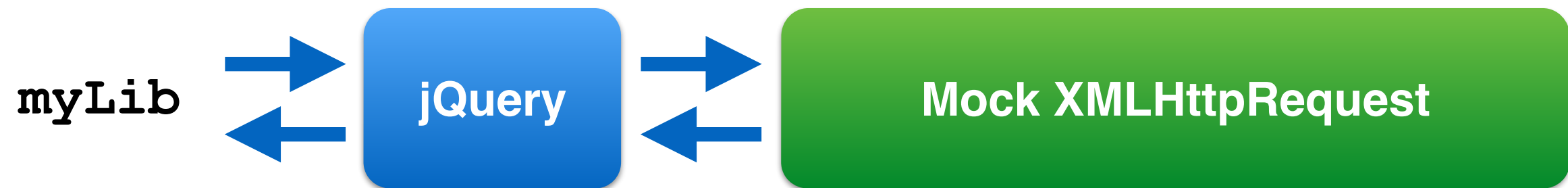
The logo for Sinon.JS, featuring the text "Sinon.JS" in a white serif font on a dark gray rectangular background.

sinonjs.org

Example: Mock



Example: Mock



Sinon.js XHR Mock

A fake implementation of a code dependency.

```
var xhr          = sinon.useFakeXMLHttpRequest();
var requests     = [];
var callback     = sinon.spy();

xhr.onCreate = function (xhr) { requests.push(xhr); };

myLib.getCommentsFor("/some/article", callback);
assertEquals(1, requests.length);

requests[0].respond(200,
  { "Content-Type": "application/json" },
  '[{ "id": 12, "comment": "Hey there" }]'
);

assert(callback.calledWith([{ id: 12, comment: "Hey there" }]));
```

Code Samples

<http://sethmcl.com/testing-web-apps/>

- 1 End-to-End test with Selenium
- 2 Unit Test node.js code
- 3 Unit Test browser code

Learn More.

<http://sethmcl.com/testing-web-apps/>

Introduction to writing testable JavaScript

[LinkedIn Tech Talk](#)

[Smashing Magazine](#)

Unit Testing with Venus.js

[LinkedIn Tech Talk](#)

[Venusjs.org](#)

End to End testing with Selenium

[The Selenium Project](#)

[Selenium Architecture](#)

[Nightwatchjs.org](#)

Free Windows VMs and other testing resources

[modern.ie](#)