

Approval Testing

When It's Too Late for Test-Driven Development



Joss Hufnagel - Computer Science
Kennesaw University
Burk Hufnagel - Solution Architect
Daugherty Business Solutions

Who is Burk Hufnagel?

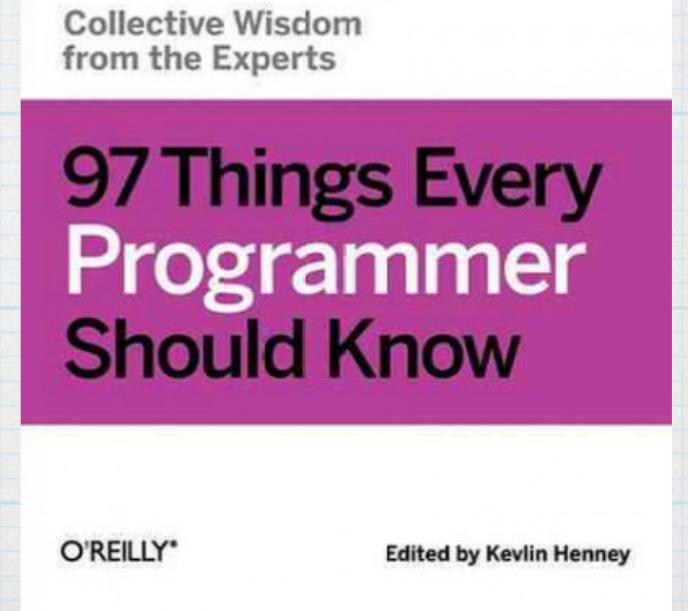
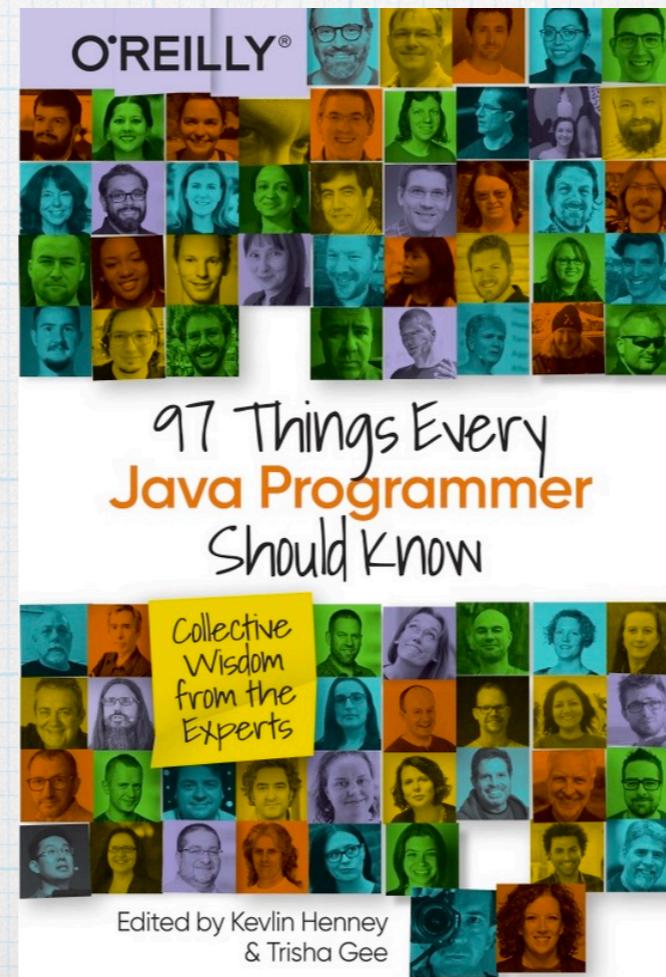
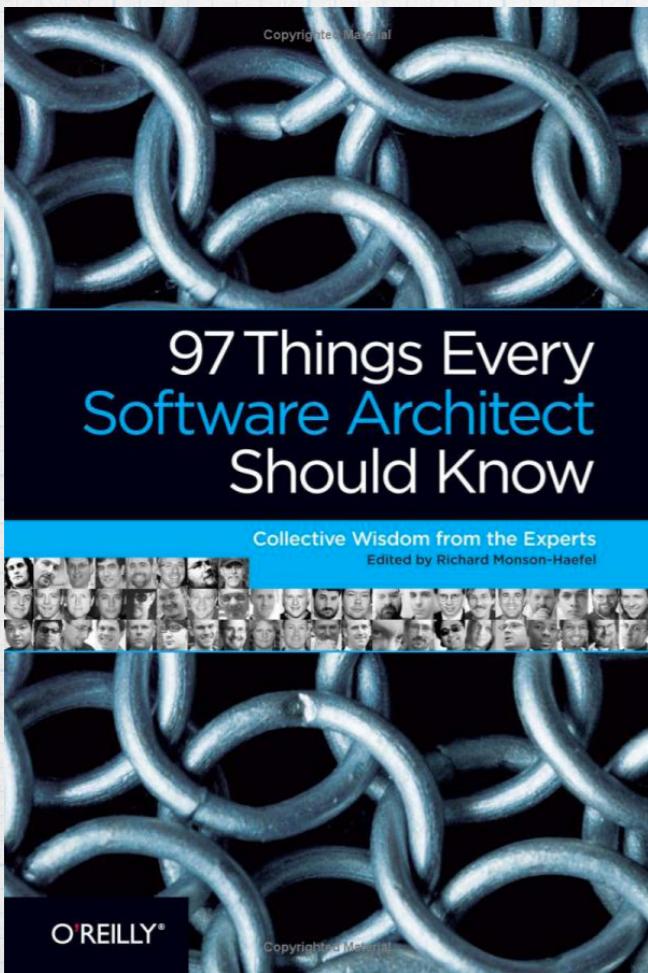
Programmer and Solution Architect with Daugherty Business Solutions



Java™
Champions



JavaOne Rock Star



Who is Joss Hufnagel?

ApprovalTests/Python Ensemble Member

Kennesaw State University CCSE Student

2019 DragonCon Presenter



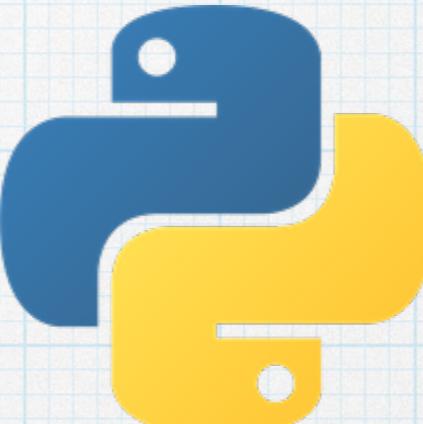
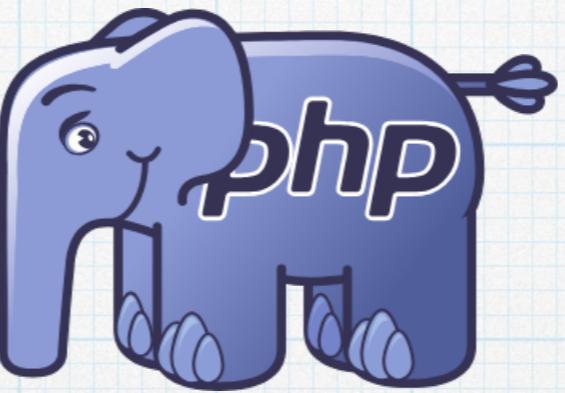
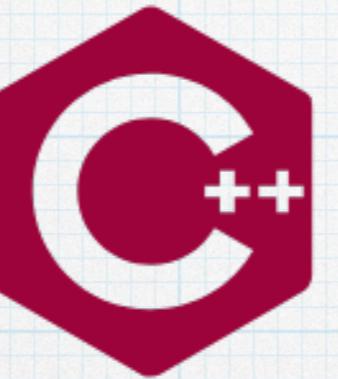
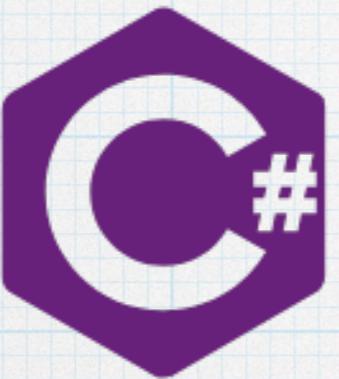
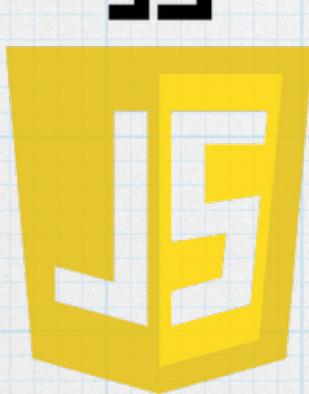
Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

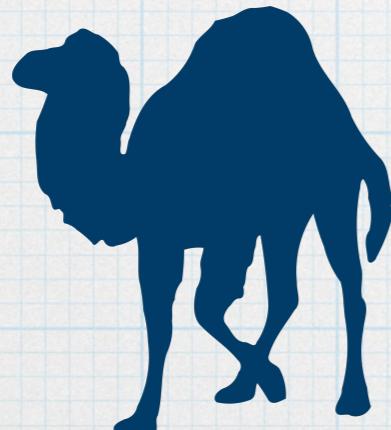
Pop Quiz!

Who's using one or more of these?

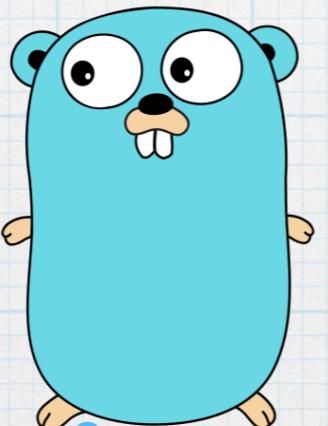
JS



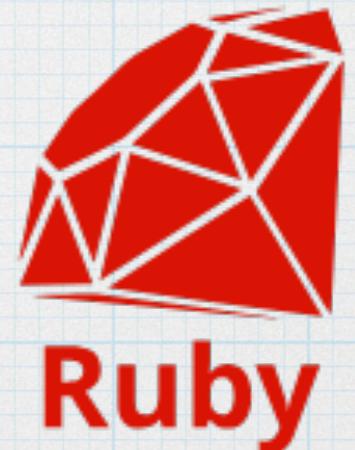
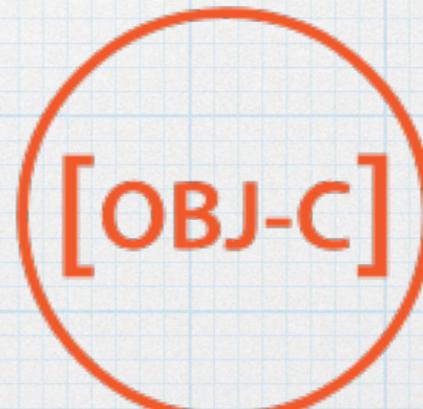
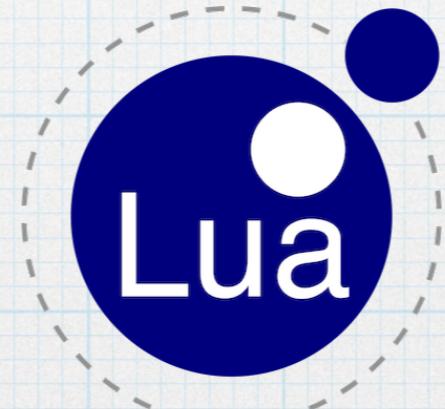
Python Swift



Perl



Go



Ruby



ApprovalTests.com

What is Approval Testing?

- * Compares current behavior vs previously recorded behavior
- * Expected changes made?
- * Any unexpected changes?
- * Automated testing for legacy code which supports refactoring

What is Approval Testing?

- * Way of verifying that changes made only changed what you wanted and expected
- * Useful with complex projects having unexpected connections
- * “Captures Human Intelligence” to verify complex outputs:
“I know it when I see it”

Capturing Human Intelligence

Approved Results

OMGHAI!

----- day 0 -----

name, sellIn, quality

+5 Dexterity Vest, 10, 20

Aged Brie, 2, 0

Elixir of the Mongoose, 4, 8

Sulfuras, Hand of Ragnaros, 0, 80

Sulfuras, Hand of Ragnaros, -1, 80

Backstage passes to a TAFKAL80ETC concert, 15, 20

Backstage passes to a TAFKAL80ETC concert, 10, 49

Backstage passes to a TAFKAL80ETC concert, 5, 47

Conjured Mana Cake, 3, 6 a

Received Results

OMGHAI!

----- day 0 -----

name, sellIn, quality

+5 Dexterity Vest, 10, 20

Aged Brie, 2, 0

Elixir of the Mongoose, 5, 8

Sulfuras, Hand of Ragnaros, 0, 80

Sulfuras, Hand of Ragnaros, -1, 80

Backstage passes to a TAFKAL80ETC concert, 15, 20

Backstage passes to a TAFKAL80ETC concert, 10, 49

Backstage passes to a TAFKAL80ETC concert, 5, 48

Conjured Mana Cake, 3, 6

Capturing Human Intelligence

Overlaid Results

OMGHAI!

----- day 0 -----

name, sellIn, quality

+5 Dexterity Vest, 10, 20

Aged Brie, 2, 0

Elixir of the Mongoose, 4, 8

Sulfuras, Hand of Ragnaros, 0, 80

Sulfuras, Hand of Ragnaros, -1, 80

Backstage passes to a TAFKAL80ETC concert, 15, 20

Backstage passes to a TAFKAL80ETC concert, 10, 49

Backstage passes to a TAFKAL80ETC concert, 5, 48

Conjured Mana Cake, 3, 6

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Why use Approval Testing?

- * Works with legacy code
- * Works with code that has automated tests
- * You don't have to understand the code being tested

Why use Approval Testing?

- * Easier to do: fits our mental model
- * Benefits everyone
- * Can assert with complex objects, not just primitives
- * Can find defects you weren't even testing for

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Example: The Gilded Rose

One Big Lump of Code

```
class Item {
  constructor(name, sellIn, quality){
    this.name = name;
    this.sellIn = sellIn;
    this.quality = quality;
  }
}

class Shop {
  constructor(items=[]){
    this.items = items;
  }
  updateQuality() {
    for (var i = 0; i < this.items.length; i++) {
      if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
        if (this.items[i].quality > 0) {
          if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
            this.items[i].quality = this.items[i].quality - 1;
          }
        } else {
          if (this.items[i].quality < 50) {
            this.items[i].quality = this.items[i].quality + 1;
            if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {
              if (this.items[i].sellIn < 11) {
                if (this.items[i].quality < 50) {
                  this.items[i].quality = this.items[i].quality + 1;
                }
              }
              if (this.items[i].sellIn < 6) {
                if (this.items[i].quality < 50) {
                  this.items[i].quality = this.items[i].quality + 1;
                }
              }
            }
          }
        }
      }
      if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
        this.items[i].sellIn = this.items[i].sellIn - 1;
      }
      if (this.items[i].sellIn < 0) {
        if (this.items[i].name != 'Aged Brie') {
          if (this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
            if (this.items[i].quality > 0) {
              if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
                this.items[i].quality = this.items[i].quality - 1;
              }
            }
          } else {
            this.items[i].quality = this.items[i].quality - this.items[i].quality;
          }
        } else {
          if (this.items[i].quality < 50) {
            this.items[i].quality = this.items[i].quality + 1;
          }
        }
      }
    }
    return this.items;
  }
}

module.exports = {
  Item,
  Shop
}
```

A Closer Look

```
class Item {  
    constructor(name, sellIn, quality){  
        this.name = name;  
        this.sellIn = sellIn;  
        this.quality = quality;  
    }  
}  
class Shop {  
    constructor(items=[]){  
        this.items = items;  
    }  
    updateQuality() {  
        for (var i = 0; i < this.items.length; i++) {  
            if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {  
                if (this.items[i].quality > 0) {  
                    if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
                        this.items[i].quality = this.items[i].quality - 1;  
                    }  
                }  
            } else {  
                if (this.items[i].quality < 50) {  
                    this.items[i].quality = this.items[i].quality + 1;  
                    if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {  
                        if (this.items[i].sellIn < 11) {  
                            if (this.items[i].quality < 50) {  
                                this.items[i].quality = this.items[i].quality + 1;  
                            }  
                        }  
                    }  
                    if (this.items[i].sellIn < 6) {  
                        if (this.items[i].quality < 50) {  
                            this.items[i].quality = this.items[i].quality + 1;  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

Add a little Whitespace

```
class Item {
```

```
    constructor(name, sellIn, quality){
```

```
        this.name = name;
```

```
        this.sellIn = sellIn;
```

```
        this.quality = quality;
```

```
}
```

```
}
```

```
class Shop {
```

```
    constructor(items=[]){
```

```
        this.items = items;
```

```
}
```

```
updateQuality() {
```

```
    for (var i = 0; i < this.items.length; i++) {
```

```
        if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
```

```
            if (this.items[i].quality > 0) {
```

```
                if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
```

```
                    this.items[i].quality = this.items[i].quality - 1;
```

```
                }
```

```
            }
```

```
        } else {
```

```
            if (this.items[i].quality < 50) {
```

```
                this.items[i].quality = this.items[i].quality + 1;
```

```
                if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {
```

```
                    if (this.items[i].sellIn < 11) {
```

```
                        if (this.items[i].quality < 50) {
```

```
                            this.items[i].quality = this.items[i].quality + 1;
```

```
}
```

Look for Clues

```
updateQuality() {
    for (var i = 0; i < this.items.length; i++) {
        if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
            if (this.items[i].quality > 0) {
                if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
                    this.items[i].quality = this.items[i].quality - 1;
                }
            }
        } else {
            if (this.items[i].quality < 50) {
                this.items[i].quality = this.items[i].quality + 1;
                if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {
                    if (this.items[i].sellIn < 11) {
                        if (this.items[i].quality < 50) {
                            this.items[i].quality = this.items[i].quality + 1;
                        }
                    }
                    if (this.items[i].sellIn < 6) {
                        if (this.items[i].quality < 50) {
                            this.items[i].quality = this.items[i].quality + 1;
                        }
                    }
                }
            }
        }
    }

    if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
        this.items[i].sellIn = this.items[i].sellIn - 1;
    }

    if (this.items[i].sellIn < 0) {
        if (this.items[i].name != 'Aged Brie') {

```

First Test

```
require('approvals')
  .mocha();
const Approvals = require('approvals');
var {Shop, Item} = require('../src/gilded_rose.js');

describe('Learning about the Gilded Rose', function() {
  describe('should do what it does', function() {
    it('should update 1 item', function() {
      const items = [new Item("Stuff", 0, 0)];

      const gildedRose = new Shop(items);

      var updatedItems = gildedRose.updateQuality(items);
      var results = "";
      for (var i = 0; i < updatedItems.length; i++) {
        results += itemToString(updatedItems[i]);
      }

      this.verify(results);
    });
  });
});
```

First Test Results

```
% rpm test
```

```
> mochajs-tests@1.0.0 test
> node ./node_modules/mocha/bin/mocha --recursive
```

```
Learning about the Gilded Rose
should do what it does
```

```
  1) should update 1 item
```

```
0 passing (2s)
1 failing
```

```
1) Learning about the Gilded Rose should do what it does should update 1 item:
```

```
Error:
```

```
Approved file does not exist: /git/GildedRose/test/Learning_about_the_Gilded_Rose.should_do_what_it_does
```

```
Approved: /git/GildedRose/test/Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_1_it
```

```
Received: /git/GildedRose/test/Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_1_it
```

```
at throwReporterError (node_modules/approvals/lib/FileApprover.js:60:11)
at exports.verify (node_modules/approvals/lib/FileApprover.js:64:5)
at Context.verify (node_modules/approvals/lib/Providers/BeforeEachVerifierBase.js:47:20)
at Context.<anonymous> (test/Simple.test.js:28:18)
at process.processImmediate (node:internal/timers:478:21)
```

First Test Results

There's a "received" file but no "approved" file

```
└── test
    ├── .received.txt
    └── Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_1_item.received.txt
        └── Simple.test.js
            └── package-lock.json
            └── package.json
```

Rename "received" to "approved"

```
└── test
    ├── .received.txt
    └── Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_1_item.approved.txt
        └── Simple.test.js
```

Re-run the test

```
% npm test

> mochajs-tests@1.0.0 test
> node ./node_modules/mocha/bin/mocha --recursive

Learning about the Gilded Rose
  should do what it does
    ✓ should update 1 item (830ms)

1 passing (837ms)
```

First Test with Coverage

```
% npm run coverage  
  
> mochajs-tests@1.0.0 coverage  
> nyc npm test --reporter=text-lcov  
  
> mochajs-tests@1.0.0 test  
> node ./node_modules/mocha/bin/mocha --recursive
```

```
Learning about the Gilded Rose  
should do what it does  
✓ should update 1 item
```

```
1 passing (15ms)
```

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	50	25.71	100	48.38	
gilded_rose.js	50	25.71	100	48.38	17-32,45-54

Code Coverage with 1 Test

```
11 class Shop {
12   constructor(items=[]) {
13     this.items = items;
14   }
15
16   updateQuality() {
17     for (var i = 0; i < this.items.length; i++) {
18       if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
19         if (this.items[i].quality > 0) {
20           if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
21             this.items[i].quality = this.items[i].quality - 1;
22           }
23         }
24       } else {
25         if (this.items[i].quality < 50) {
26           this.items[i].quality = this.items[i].quality + 1;
27           if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {
28             if (this.items[i].sellIn < 11) {
29               if (this.items[i].quality < 50) {
30                 this.items[i].quality = this.items[i].quality + 1;
31               }
32             }
33             if (this.items[i].sellIn < 6) {
34               if (this.items[i].quality < 50) {
35                 this.items[i].quality = this.items[i].quality + 1;
36               }
37             }
38           }
39         }
40       }
41     }
42     if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
43       this.items[i].sellIn = this.items[i].sellIn - 1;
44     }
45     if (this.items[i].sellIn < 0) {
46       if (this.items[i].name != 'Aged Brie') {
47         if (this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
48           if (this.items[i].quality > 0) {
49             if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
50               this.items[i].quality = this.items[i].quality - 1;
51             }
52           }
53         }
54       }
55     }
56   }
57 }
```

Second Test

```
it('should update 2 items', function() {
    const items = [
        new Item("Stuff", 0, 0),
        new Item("Aged Brie", 0, 30)
    ];

    const gildedRose = new Shop(items);

    var updatedItems = gildedRose.updateQuality(items);
    var results = "";
    for (var i = 0; i < updatedItems.length; i++) {
        results += itemToString(updatedItems[i]);
    }

    this.verify(results);
});
```

Approved file does not exist: /git/GildedRose/test/
Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_2_items.approved.txt:

Two Tests with Coverage

```
% npm run coverage  
  
> mochajs-tests@1.0.0 coverage  
> nyc npm test --reporter=text-lcov  
  
> mochajs-tests@1.0.0 test  
> node ./node_modules/mocha/bin/mocha --recursive
```

```
Learning about the Gilded Rose  
should do what it does  
  ✓ should update 1 item  
  ✓ should update 2 items
```

1 passing (15ms)

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	65.62	40	100	64.51	
gilded_rose.js	65.62	40	100	64.51	21-22,29-36,50-55

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	50	25.71	100	48.38	
gilded_rose.js	50	25.71	100	48.38	17-32,45-54

Code Coverage with 2 Tests

```
11 class Shop {  
12     constructor(items=[]){  
13         this.items = items;  
14     }  
15  
16     updateQuality() {  
17         for (var i = 0; i < this.items.length; i++) {  
18             if (this.items[i].name != 'Aged Brie' && this.items[i].quality > 0) {  
19                 if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
20                     this.items[i].quality = this.items[i].quality - 1;  
21                 }  
22             } else {  
23                 if (this.items[i].quality < 50) {  
24                     this.items[i].quality = this.items[i].quality + 1;  
25                     if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {  
26                         if (this.items[i].sellIn < 11) {  
27                             if (this.items[i].quality < 50) {  
28                                 this.items[i].quality = this.items[i].quality + 1;  
29                             }  
30                         if (this.items[i].sellIn < 6) {  
31                             if (this.items[i].quality < 50) {  
32                                 this.items[i].quality = this.items[i].quality + 1;  
33                             }  
34                         if (this.items[i].sellIn < 6) {  
35                             if (this.items[i].quality < 50) {  
36                                 this.items[i].quality = this.items[i].quality + 1;  
37                             }  
38                         }  
39                     }  
40                 }  
41             }  
42             if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
43                 this.items[i].sellIn = this.items[i].sellIn - 1;  
44             }  
45             if (this.items[i].sellIn < 0) {  
46                 if (this.items[i].name != 'Aged Brie') {  
47                     if (this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {  
48                         if (this.items[i].quality > 0) {  
49                             if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
50                                 this.items[i].quality = this.items[i].quality - 1;  
51                             }  
52                         }  
53                     }  
54                 }  
55             }  
56         }  
57     }  
58 }
```

```
11 class Shop {  
12     constructor(items=[]){  
13         this.items = items;  
14     }  
15  
16     updateQuality() {  
17         for (var i = 0; i < this.items.length; i++) {  
18             if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {  
19                 if (this.items[i].quality > 0) {  
20                     if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
21                         this.items[i].quality = this.items[i].quality - 1;  
22                     }  
23                 }  
24             } else {  
25                 if (this.items[i].quality < 50) {  
26                     this.items[i].quality = this.items[i].quality + 1;  
27                     if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {  
28                         if (this.items[i].sellIn < 11) {  
29                             if (this.items[i].quality < 50) {  
30                                 this.items[i].quality = this.items[i].quality + 1;  
31                             }  
32                         }  
33                     }  
34                     if (this.items[i].sellIn < 6) {  
35                         if (this.items[i].quality < 50) {  
36                             this.items[i].quality = this.items[i].quality + 1;  
37                         }  
38                     }  
39                 }  
40             }  
41             if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
42                 this.items[i].sellIn = this.items[i].sellIn - 1;  
43             }  
44             if (this.items[i].sellIn < 0) {  
45                 if (this.items[i].name != 'Aged Brie') {  
46                     if (this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {  
47                         if (this.items[i].quality > 0) {  
48                             if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {  
49                                 this.items[i].quality = this.items[i].quality - 1;  
50                             }  
51                         }  
52                     }  
53                 }  
54             }  
55         }  
56     }  
57 }
```

Third Test

```
it('should update 3 items', function() {
  const items = [
    new Item("Stuff", 0, 0),
    new Item("Aged Brie", 0, 30),
    new Item("Sulfuras, Hand of Ragnaros", 0, 30)
  ];

  const gildedRose = new Shop(items);

  var updatedItems = gildedRose.updateQuality(items);
  var results = "";
  for (var i = 0; i < updatedItems.length; i++) {
    results += itemToString(updatedItems[i]);
  }

  this.verify(results);
});
```

Approved file does not exist: /git/GildedRose/test/
Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.approved.txt:

Three Tests with Coverage

```
% npm run coverage
```

```
Learning about the Gilded Rose
  should do what it does
    ✓ should update 1 item
    ✓ should update 2 items

  1) should update 3 items
```

```
2 passing (4s)
1 failing
```

```
1) Learning about the Gilded Rose should do what it does should update 3 items:
```

```
Error:
```

```
Files do not match.:
```

```
Approved: /git/GildedRose/test/Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items
Received: /git/GildedRose/test/Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items
```

```
at throwReporterError (node_modules/approvals/lib/FileApprover.js:60:11)
at exports.verify (node_modules/approvals/lib/FileApprover.js:126:9)
at Context.verify (node_modules/approvals/lib/Providers/BeforeEachVerifierBase.js:47:20)
at Context.<anonymous> (test/Simple.test.js:63:18)
at process.processImmediate (node:internal/timers:478:21)
```

Three Tests with Coverage

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt <-> Learning_about_the_Gilded_Rose.sh...

Home Sessions All Diffs Same Context Minor Rules Format Copy Edit Next Section Prev Section Swap Reload

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt (Today, 9:51:33 PM) 161 bytes Everything Else Unicode (UTF-8) UNIX

```
Item name: "Stuff" sellIn: "-1" quality: "0"
Item name: "Aged Brie" sellIn: "-1" quality: "32"
Item name: "Sulfuras, Hand of Ragnaros" sellIn: "0" quality: "42"
```

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt (Today, 9:42:19 PM) 161 bytes Everything Else Unicode (UTF-8) UNIX

```
Item name: "Stuff" sellIn: "-1" quality: "0"
Item name: "Aged Brie" sellIn: "-1" quality: "32"
Item name: "Sulfuras, Hand of Ragnaros" sellIn: "0" quality: "30"
```

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt <-> Learning_about_the_Gilded_Rose.sh...

Home Sessions All Diffs Same Context Minor Rules Format Copy Edit Next Section Prev Section Swap Reload

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt (Today, 9:51:33 PM) 161 bytes Everything Else Unicode (UTF-8) UNIX

```
Item name: "Stuff" sellIn: "-1" quality: "0"
Item name: "Aged Brie" sellIn: "-1" quality: "32"
Item name: "Sulfuras, Hand of Ragnaros" sellIn: "0" quality: "42"
```

Learning_about_the_Gilded_Rose.should_do_what_it_does.should_update_3_items.received.txt (Today, 9:42:19 PM) 161 bytes Everything Else Unicode (UTF-8) UNIX

```
Item name: "Stuff" sellIn: "-1" quality: "0"
Item name: "Aged Brie" sellIn: "-1" quality: "32"
Item name: "Sulfuras, Hand of Ragnaros" sellIn: "0" quality: "42"
```

Three Tests with Coverage

```
% npm run coverage
```

```
Learning about the Gilded Rose
should do what it does
✓ should update 1 item
✓ should update 2 items
✓ should update 3 items
```

3 passing (34ms)

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	68.75	51.42	100	67.74	
gilded_rose.js	68.75	51.42	100	67.74	22,29-36,50-55

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	50	25.71	100	48.38	
gilded_rose.js	50	25.71	100	48.38	17-32,45-54

Twelfth Test

```
it('should update 12 items', function() {
  const items = [
    new Item("Stuff", 1, 0),
    new Item("Things", 0, 25),
    new Item("Other Things", -1, 0),
    new Item("Other Stuff", -1, 25),
    new Item("Aged Brie", -1, 40),
    new Item("Aged Brie", -1, 50),
    new Item("Backstage passes to a TAFKAL80ETC concert", 10, 50),
    new Item("Backstage passes to a TAFKAL80ETC concert", 11, 48),
    new Item("Backstage passes to a TAFKAL80ETC concert", 5, 49),
    new Item("Backstage passes to a TAFKAL80ETC concert", 6, 15),
    new Item('Backstage passes to a TAFKAL80ETC concert', -1, 15),
    new Item("Sulfuras, Hand of Ragnaros", -1, 42)
  ];

  const gildedRose = new Shop(items);

  var updatedItems = gildedRose.updateQuality();
  var results = "";
  for (var i = 0; i < updatedItems.length; i++) {
    results += itemToString(updatedItems[i]);
  }

  this.verify(results);
});
```

Twelfth Test with Coverage

```
% npm run coverage  
> mochajs-tests@1.0.0 coverage  
> nyc npm test --reporter=text-lcov  
  
> mochajs-tests@1.0.0 test  
> node ./node_modules/mocha/bin/mocha --recursive
```

```
Learning about the Gilded Rose  
should do what it does  
✓ should update 12 items  
✓ should work with no inventory
```

2 passing (18ms)

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	100	100	100	100	
gilded_rose.js	100	100	100	100	

Printer & Scrubber

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Approval Testing and TDD

- * TDD benefits programmers but can't help with Legacy Code
- * Approval Tests were made for dealing with Legacy Code
- * Use Verify instead of Assert when doing TDD to test complex results - objects, XML docs, etc.

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Questions and Answers

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Summary

- * Approval Testing is a Good Thing
- * Approval Testing complements TDD
- * Get started with Approval Testing is easy

Agenda

- * What is Approval Testing?
- * Why use Approval Testing?
- * Examples of Approval Testing
- * Approval Tests and TDD
- * Questions And Answers
- * Summary
- * Resources

Resources

Website:

<https://approvaltests.com>

GitHub:

<https://github.com/approvals>

<https://github.com/approvals/Approvals.NodeJS>

Thank you!

