

Coding dojo

Test-driven Development

knowit

Coding Dojo

Goal: Become a better programmer through
practice “katas”



Unit Test

```
var calc = new Calculator(); // Arrange
```

```
var x = calc.sum(4,3,5); // Act
```

```
expect( x ).toBe( 12 ); // Assert
```



Test Runner

```
Jasmine 1.3.1 revision 1354556913 finished in 0.025s
...X.

Failing 1 spec No try/catch ☐
5 specs | 1 failing

Player when song has been paused should be possible to resume.
Expected false to be truthy.
```

```
Jasmine 1.3.1 revision 1354556913 finished in 0.019s
.....

Passing 5 specs No try/catch ☐

Player
  should be able to play a Song

  when song has been paused
    should indicate that the song is currently paused
    should be possible to resume
  tells the current song if the user has made it a favorite

  #resume
    should throw an exception if song is already playing
```

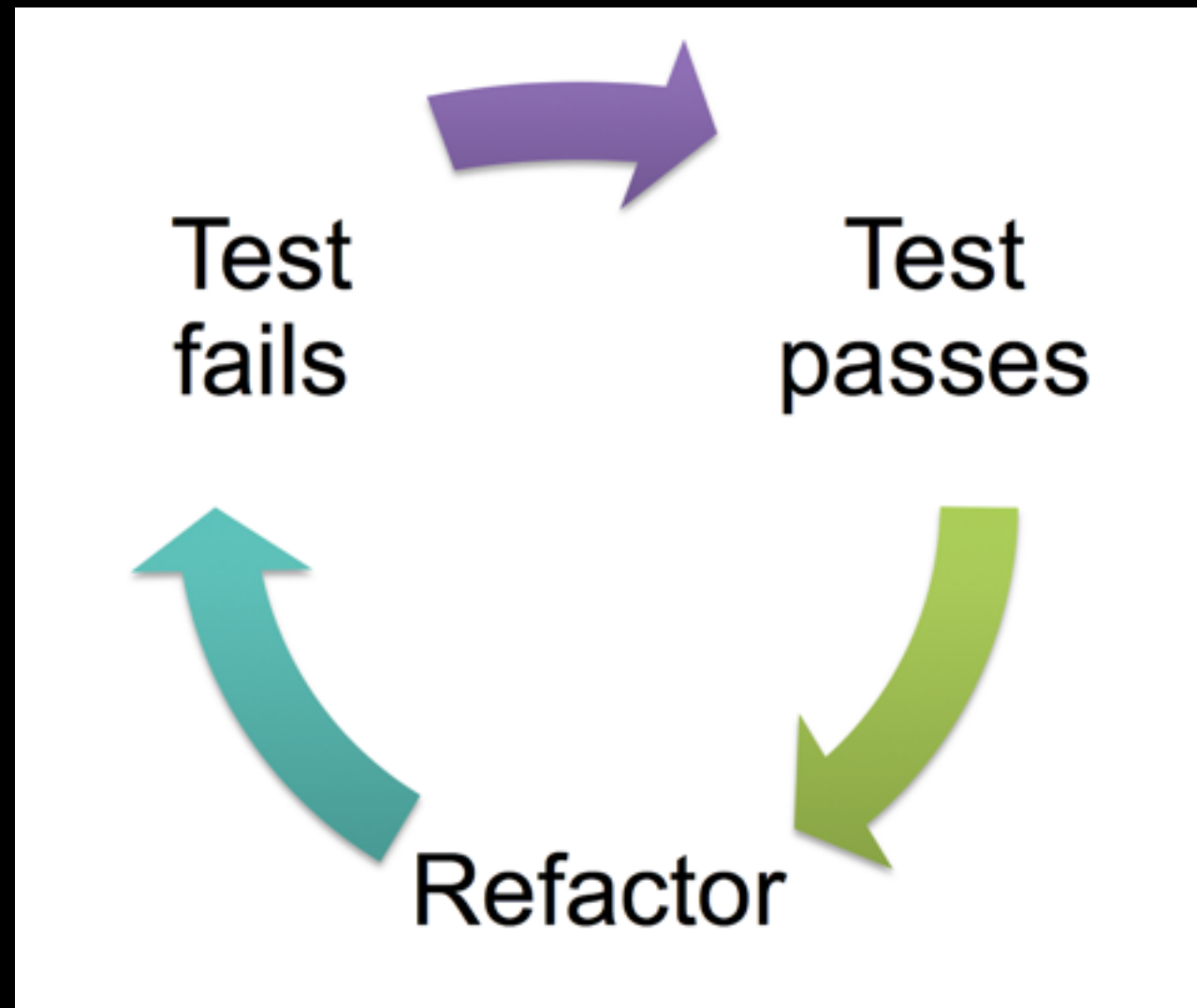


Benefits of Unit Tests

- Automate testing
- Documentation
- Cross-browser testing



Test-driven Development



Benefits of TDD

- No production code without automated test
- Enforce thinking before coding
- Better API design



Benefits of TDD

- No production code without automated test
- Enforce thinking before coding
- Better API design
- getResult("sum", 21, 42, true, true, false)



Ping Pong Pair Programming

Programmer 1

Write failing test

Programmer 2

Make test pass

Refactor

Write failing test

Make test pass

Refactor

...

Will code for beer

