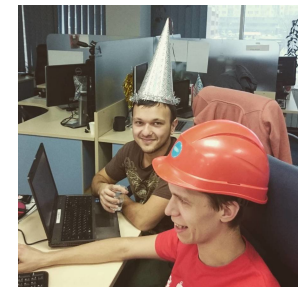
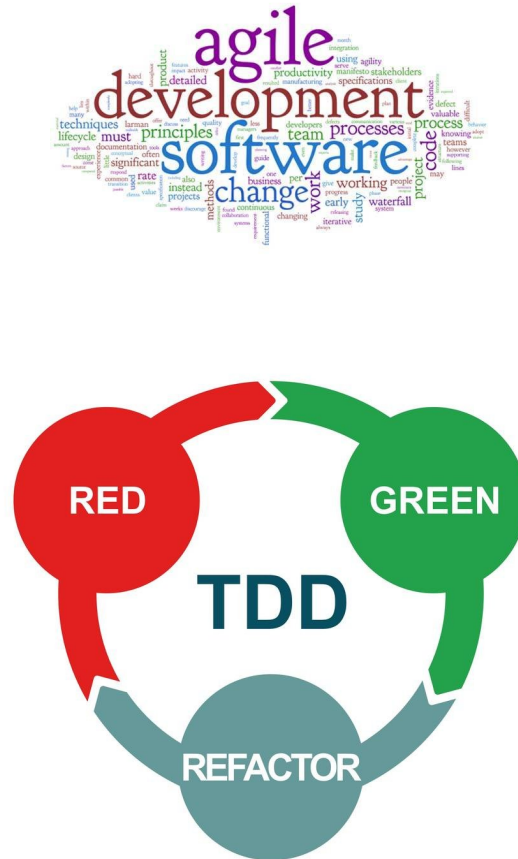


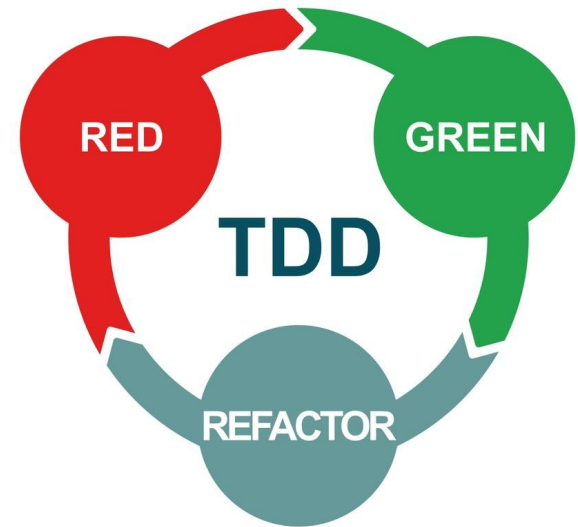
JavaScript + Angular

Part 10



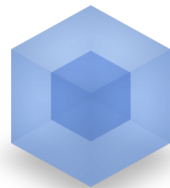
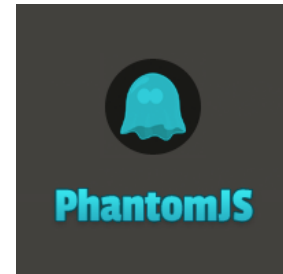
TDD steps

- Navigation map
- Test first
- Assert first
- Fail first





Istanbul
isparta



webpack
MODULE BUNDLER



Show me the source!

test_config.js

```
chai.should();

beforeEach(() => {
  window.env = sinon.sandbox.create();
});

afterEach(() => {
  window.env.restore();
});
```

*Service.spec.js

```
describe('app::service MyService', () => {  
  let sut, $q;  
  
  beforeEach(module('app'));  
  
  beforeEach(inject((MyService, _$q_) => {  
    sut = MyService;  
    $q = _$q_;  
  }));  
  
  context("some context", () => {  
    it('should be here', () => {  
      sut.context.should.have.been.called;  
    });  
  });  
});
```

*Router.spec.js

```
describe('app: routes: MyRouter', () => {
  let $state;

  beforeEach(() => {
    module('app');

    inject((_ $state_) => {
      $state = _ $state_;
    });
  });

  describe('my menu', () => {

    it('should have url', () => {
      $state.get('my.menu').url.should.equal('/my_menu');
    });

    it('should run my application', () => {
      $state.get('my.menu').views['menu@my'].controller.should.equal('MyController');
    });
  });
});
```


*Directive.spec.js

```
describe('directive: myDirective', () => {

  let $compile;
  let scope, element;

  beforeEach(() => {
    module('app');
    inject(function (_$rootScope_, _$compile_) {
      scope = _$rootScope_.$new();
      $compile = _$compile_;
    });
    element = $compile('<div my-directive></div>')(scope);
  });

  it('should blur if element was selected', () => {
    element.triggerHandler(focusInEvent);
    inputBlur.should.have.been.called;
  });
});
```

*Directive.spec.js

```
describe('directive: mySecondDirective', () => {
  let sut;
  beforeEach(() => {

    module('app', function ($provide) {
      $provide.constant('info', info);
    });

    inject(function (mySecondDirective) {
      sut = mySecondDirective[0];
    });
  });

  describe('scope', () => {
    var scope;
    beforeEach(() => {
      scope = {};
      sut.link(scope);
    });

    it('should has info on scope', () => {
      scope.info.should.equal(info);
    });
  });

  it('should be used as element', () => {
    sut.restrict.should.equal('E');
  });
});
```

i18n

```
it('has matching keys for both french and english',  
  inject((resourcesEN_CA, resourcesFR_CA) => {  
    resourcesEN_CA.should.have.equalKeys(resourcesFR_CA);  
  })  
);
```

Performance

```
INFO [launcher]: Starting browser PhantomJS
```

```
INFO [PhantomJS 2.1.1 (Linux 0.0.0)]: Connected on socket OzY3MGHbstNJMXAeBar2 with id 44096908
```

```
PhantomJS 2.1.1 (Linux 0.0.0): Executed 3745 of 3746 (skipped 1) SUCCESS (1 min 20.57 secs / 0.735 secs)
```

```
beforeEach(module( 'app' ));
```

```
////
```

```
angular
```

```
    .module( 'app' )
```

```
    .config(routeConfig);
```

```
/*@ngInject*/
```

```
function routeConfig($stateProvider) {
```

```
    $stateProvider.state('login', {
```

```
        url: '/login',
```

```
        abstract: true,
```

```
        templateUrl: 'login/login/login_base.html',
```

```
        onEnter: onEnterLoginState
```

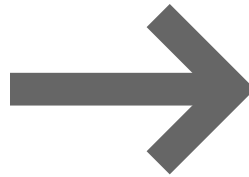
```
    });
```




+

istanbul-combine

- Combine JS files (**remap-istanbul**)
- Avoid memory leaks
- Less DOM manipulation (**no \$compile**)
- Don't use DI from Angular JS
(**webpack import or browserify**)



"karma-phantomjs-launcher": "^1.0.0"

Tasks

- String calculator
- Game "Life"