

# THE PERSUIT OF QUALITY

# THE OF QUALITY

# MEANING OF "QUALITY"

Ha un significato relativo e **soggettivo**.

# MEANING OF "QUALITY"

**"Proprietà che caratterizza una cosa in relazione a particolari attività, funzioni e utilizzazioni."**

— Treccani

# MEANING OF "QUALITY"

**"Insieme delle caratteristiche e funzionalità di un prodotto che ne determinano la capacità di soddisfare esigenze espresse o implicite."**

— ISO 9000

# LA NOSTRA APP È DI QUALITÀ?

- 1. Cosa misurare?**
- 2. Come misurare?**

IF YOU CANNOT MEASURE IT, YOU  
CANNOT IMPROVE IT  
- DEMING

# 1. COSA MISURARE?

- Coverage
- Indice Lighthouse
- Complessità ciclomatica
- etc.

# 1. COSA MISURARE?

1. Significativo

2. Misurabile

3. Automatizzabile

# 2. COME MISURARE?

1. Definire metriche
2. Valutare range relativi di accettabilità ed ottimalità
3. Configurare strumenti automatici

# PERCHÉ AUTOMATIZZARE?

1. Pigritia
2. Determinismo
3. Risultati analizzabili
4. Efficiente
5. Pigritia

# PIGRIZIA

Ricercare la qualità non deve essere arduo

# UNA RICERCA PREVENTIVA

- Linters e plugin editor/IDE: **TSLint/ESLint**
- **typescript-eslint-parser**
- Git hooks: **husky** 
- CI: **Travis, CircleCI** 
- **Quality Assurance**
- **Utenti**

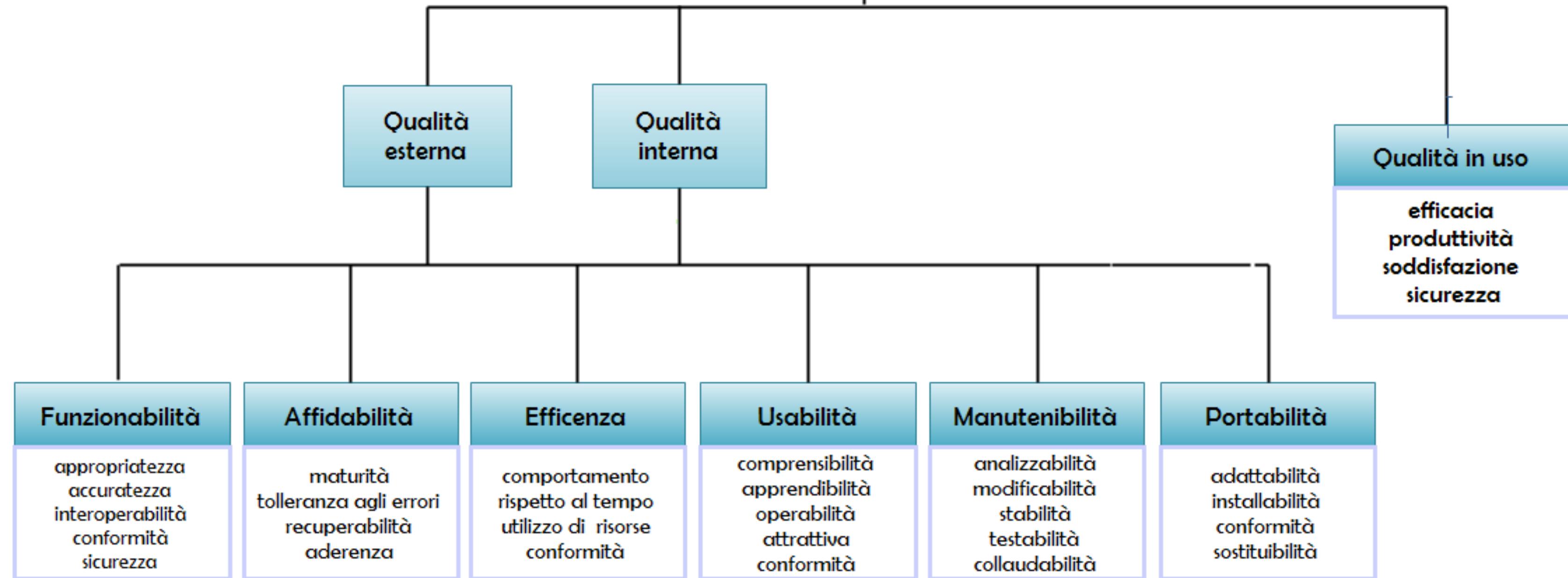
# INCANALATE L'ODIO

Gli strumenti automatici evitano che i vostri colleghi odino voi.



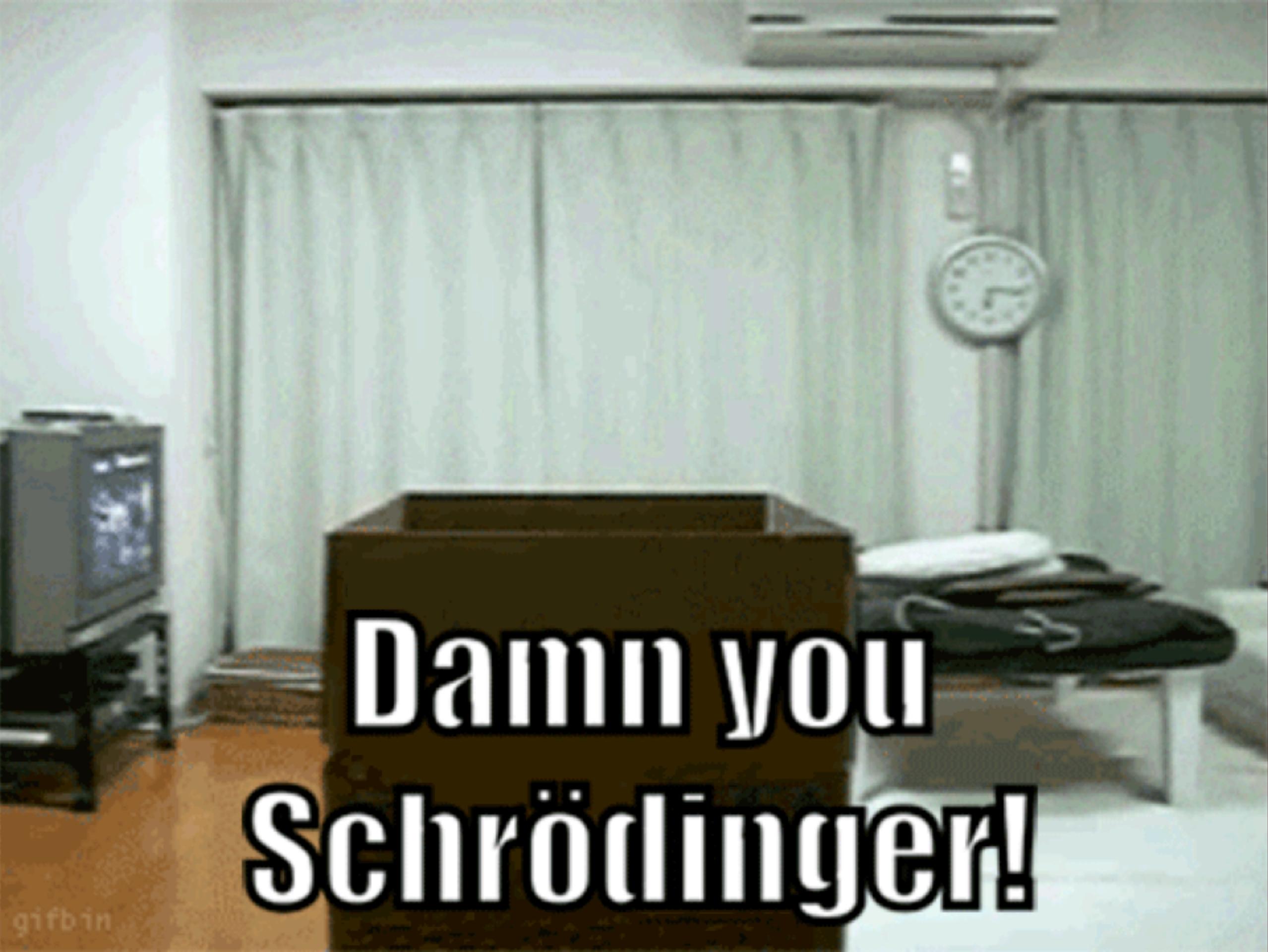
ISO 9126

## Modello ISO/IEC 9126



# FUNZIONALITÀ

Nel momento in cui rilasciamo il software, prima che gli utenti lo vedano, la nostra app possiede le stesse proprietà del gatto di Schrödinger.



Damn you  
Schrödinger!

# TESTING

- Perché testare? Assert(JS) Conf 2018
- Come convincere i vostri manager But it sucks!

# TESTING

1. Unit tests
2. Integration tests
3. e2e/system tests

# TESTING

1) Unit tests: Karma/Jest

Testing Angular with Jest

```
47      <div class="notifications notifications--compact" *ngIf="showSendingMail">
48          <p>{{messages.EMAIL_TEST_CODE_INFO_SENDING}}</p>
49      </div>
50      <div class="notifications notifications--compact notifications--success">
51          <p>{{messages.EMAIL_TEST_CODE_SUCCESS}}</p>
52      </div>
53  </div>
54 </div>
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: zsh

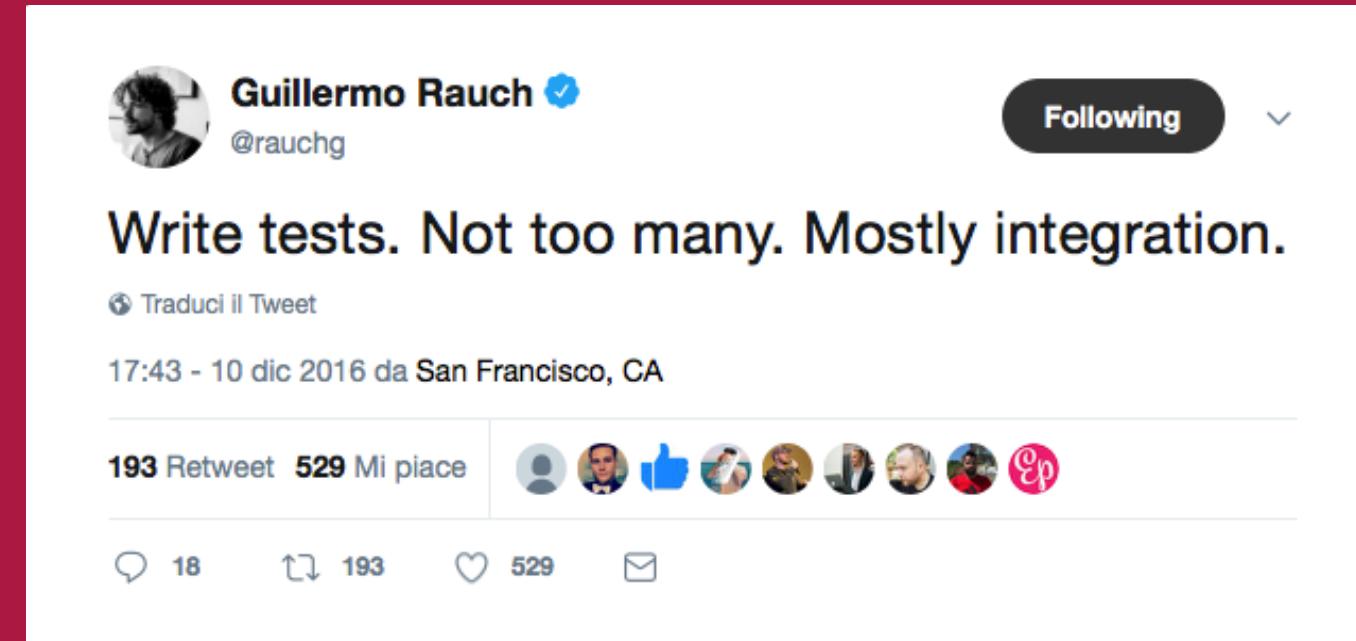


```
*[master] [~/Projects/angular-jest-demo]$ yarn test -- --watch
```

rvm: ruby-2.

# TESTING

## 2) Integration tests

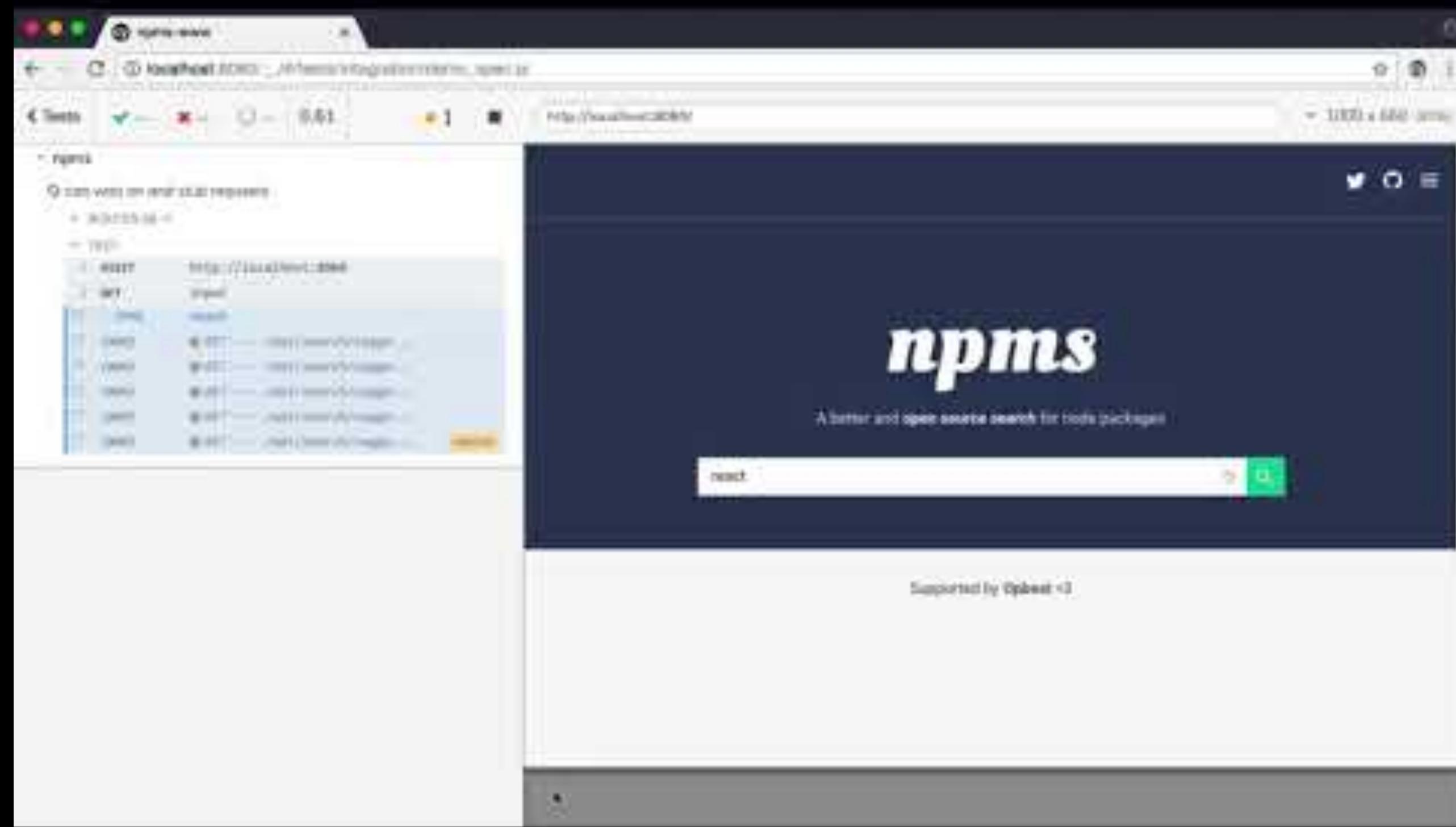


Write mostly integration tests

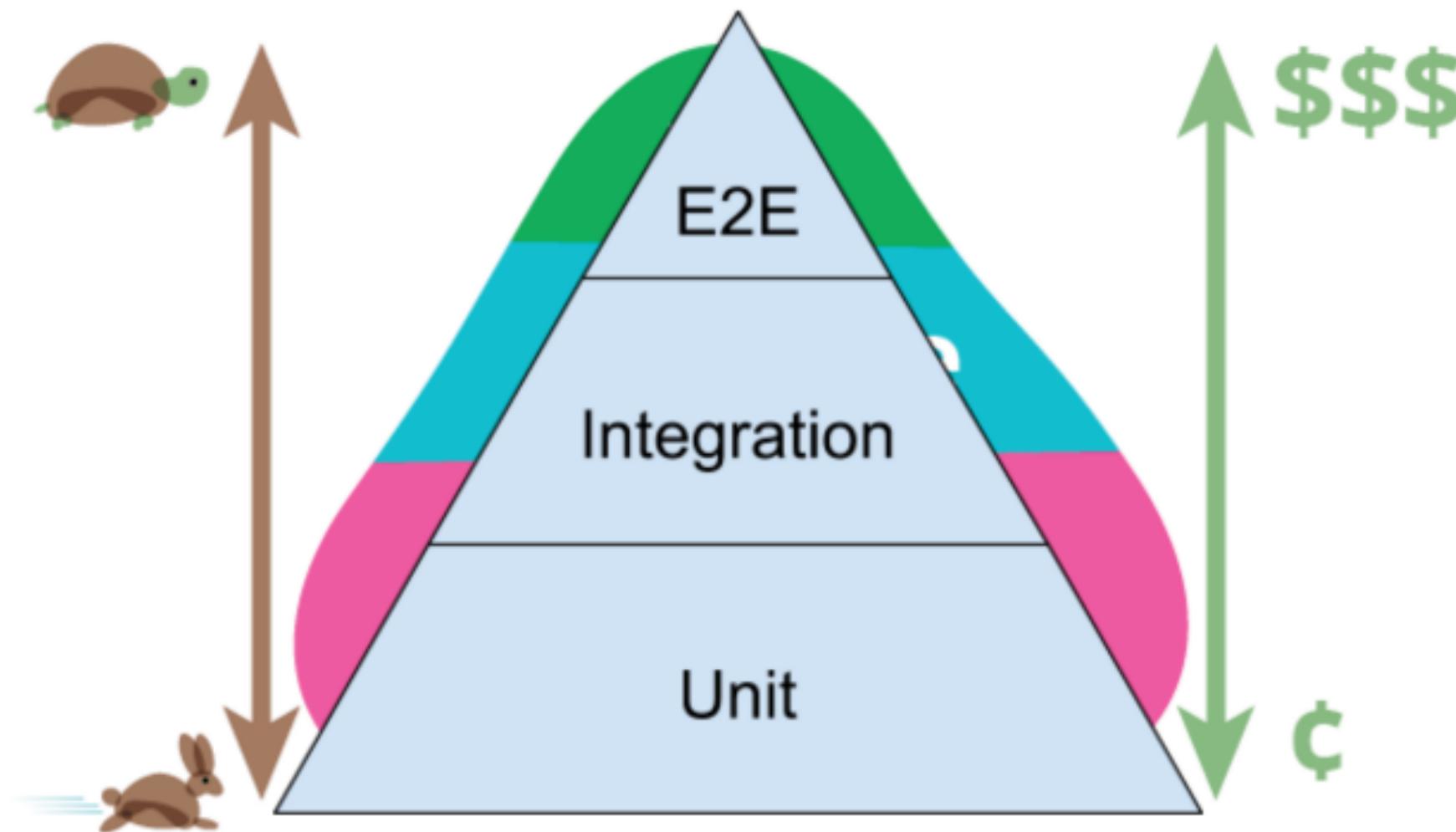
# TESTING

3) e2e/system tests: **Protractor/Cypress**

Cypress with Angular & Typescript



# Where do we focus our time?



[martinfowler.com/bliki/TestPyramid.html](http://martinfowler.com/bliki/TestPyramid.html)  
[testing.googleblog.com/2015/04/just-say-no-to-more-end-to-end-tests.html](http://testing.googleblog.com/2015/04/just-say-no-to-more-end-to-end-tests.html)

QA

Anche una buona Quality Assurance **non rileva**  
tutti gli errori e sempre

# CONSIGLI PER IL TESTING

- Minimizzare l'API `chai`/`sinon`
- Minimizzare la curva di apprendimento
- Non deve spaventare a prima vista

```
beforeEach(async(() => {
  const routerSpy = createRouterSpy();

  TestBed.configureTestingModule({
    imports: [ HeroModule ],
    providers: [
      { provide: ActivatedRoute, useValue: activatedRoute },
      { provide: Router,           useValue: routerSpy },
      // HeroDetailService at this level is IRRELEVANT!
      { provide: HeroDetailService, useValue: {} }
    ]
  })

  // Override component's own provider
  .overrideComponent(HeroDetailComponent, {
    set: {
      providers: [
        { provide: HeroDetailService, useClass: HeroDetailServiceSpy }
      ]
    }
  })
})

.compileComponents();
}));
```

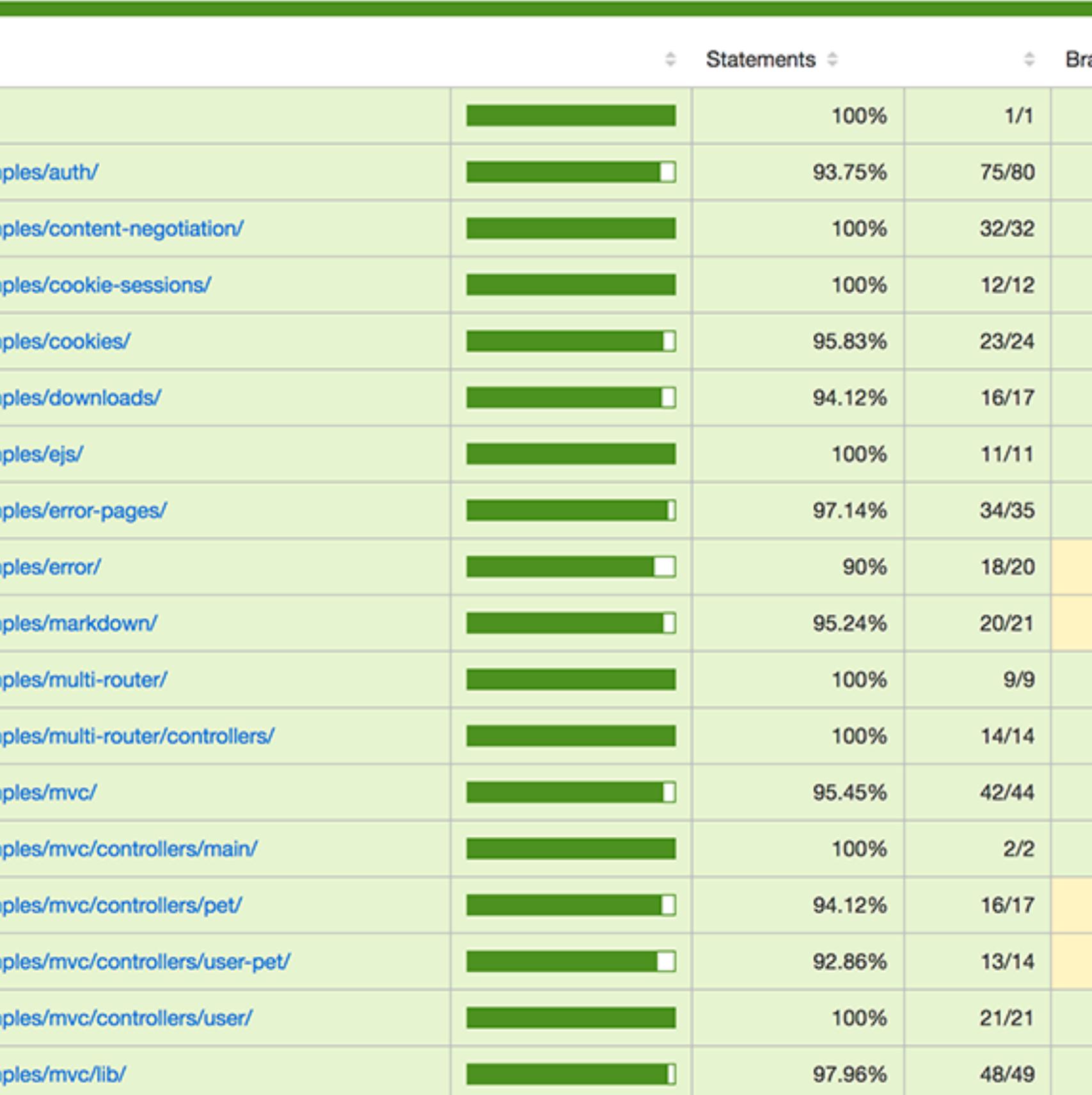
# AFFIDABILITÀ

# AFFIDABILITÀ

- Testing (again)
  - Coverage

"TESTING SHOWS THE PRESENCE, NOT  
THE ABSENCE OF BUGS"

— Dijkstra (1969)



# COVERAGE

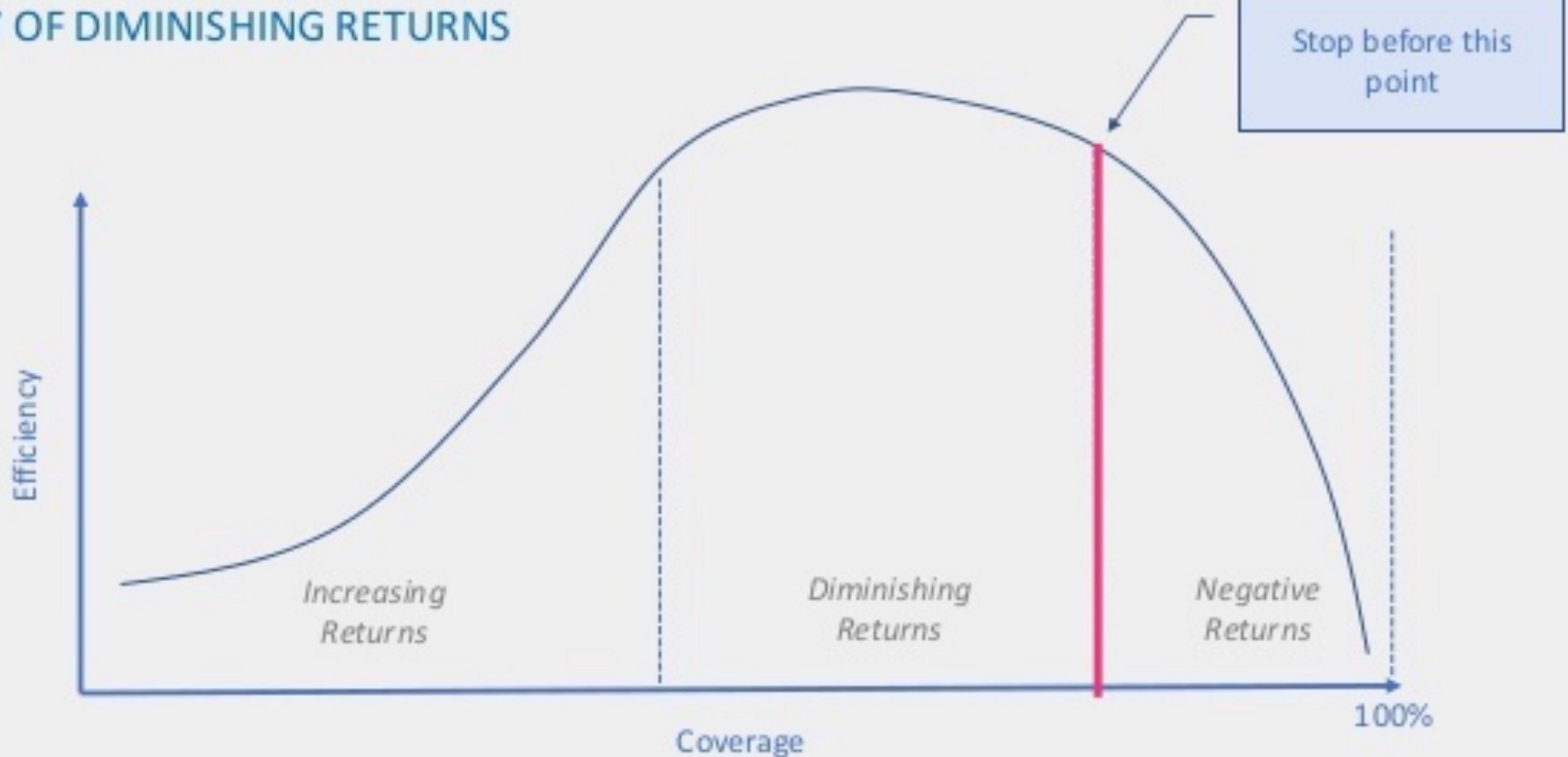
- Istanbul
- Karma coverage/Jest

# COVERAGE

**Statement coverage**  
vs  
**Branch coverage**

# HOW MUCH COVERAGE?

## LAW OF DIMINISHING RETURNS



# EFFICIENZA

# EFFICIENZA

- Complessità ciclomatica
  - Indice Lighthouse
  - Dimensione bundle

# COMPLESSITÀ CICLOMATICA

```
function getChild() {  
  
    let childDe: DebugElement;  
  
    try {  
        childDe = fixture.debugElement.children[4].children[0]; #1  
    } catch (err) {  
        /* we'll report the error */ #2  
    }  
  
    childDe = fixture.debugElement  
        .queryAll(function (de) { return de.componentInstance instanceof MyIfChildComponent; })[0];  
  
    childDe = fixture.debugElement  
        .query(function (de) { return de.componentInstance instanceof MyIfChildComponent; });  
  
    if (childDe && childDe.componentInstance) {  
        child = childDe.componentInstance; #3  
    } else {  
        fail('Unable to find MyIfChildComp within MyIfParentComp'); #4  
    }  
  
    return child;  
}
```

# Lighthouse



# DIMENSIONI BUNDLE

## import-cost

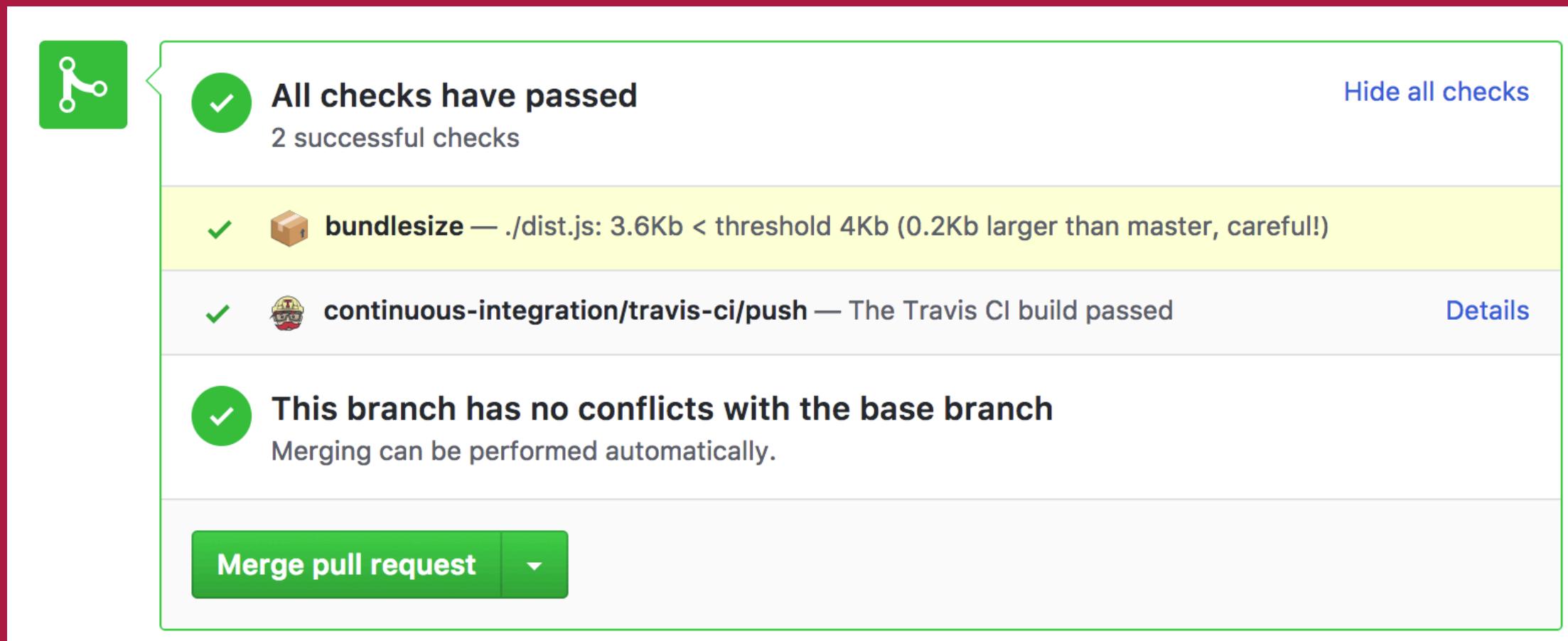


A screenshot of a code editor interface. On the left is a dark sidebar with white icons for file operations (new, open, save, close) and search. The main area shows a file named "App.js". The code consists of four numbered lines:

```
1
2
3
4
```

# DIMENSIONI BUNDLE

## bundlesize



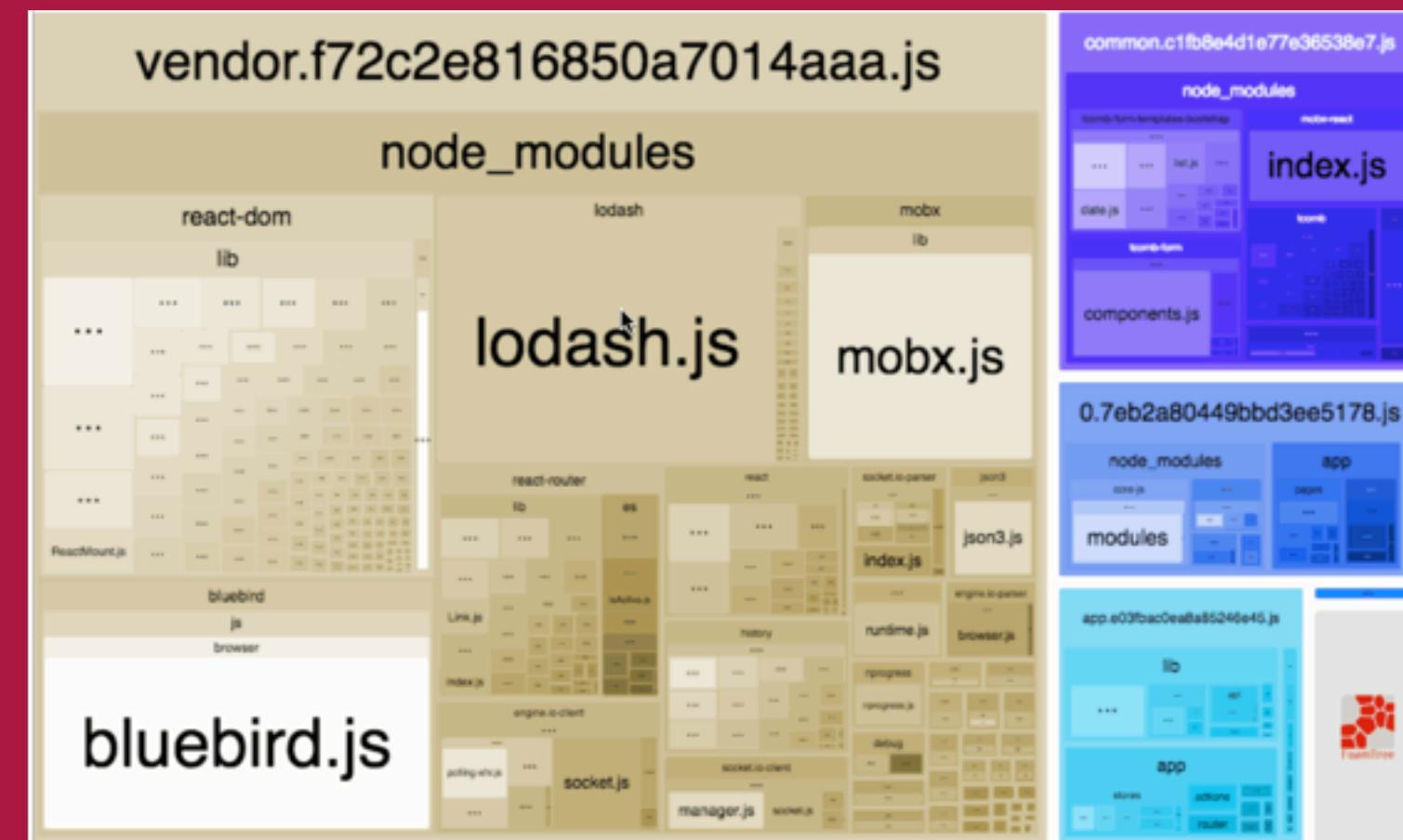
A screenshot of a GitHub pull request merge button. The button is white with a green border and contains the text "Merge pull request". To the right of the text is a small dropdown arrow icon. Above the merge button, there is a summary of CI status:

- All checks have passed** (2 successful checks)
- bundlesize** — ./dist.js: 3.6Kb < threshold 4Kb (0.2Kb larger than master, careful!)
- continuous-integration/travis-ci/push** — The Travis CI build passed [Details](#)
- This branch has no conflicts with the base branch**  
Merging can be performed automatically.

At the top right of the summary area, there is a link labeled "Hide all checks".

# DIMENSIONI BUNDLE

## webpack-bundle-analyzer



# MANUTENIBILITÀ

# TYPESCRIPT

- Strict configuration
- Conditional types

# STRICT CONFIGURATION

--strict

```
{  
  "noImplicitAny": true,  
  "noImplicitThis": true,  
  "alwaysStrict": true,  
  "strictNullChecks": true,  
  "strictFunctionTypes": true, // TS 2.6  
  "strictPropertyInitialization": true // TS 2.7  
}
```

# CONDITIONAL TYPES (TS 2.9)

TS <= 2.8

```
type AppState =  
Readonly<BaseAppState>
```

TS >= 2.9

```
type AppState =  
DeepReadonly<BaseAppState>
```

```
type DeepReadonly<T> =
  T extends any[] ? Deep_READONLY_ARRAY<T[number]> :
  T extends object ? Deep_READONLY_OBJECT<T> :
  T;

interface Deep_READONLY_ARRAY<T> extends ReadonlyArray<DeepReadonly<T>> {}

type Deep_READONLY_OBJECT<T> = {
  readonly [KEY in keyof T]: DeepReadonly<T[KEY]>;
};
```

# RETURN TYPE

```
type AddUserAction = {
  type: 'ADD_USER';
  payload: User;
}

const addUser = function(user: User): AddUserAction {
  return ({
    type: 'ADD_USER',
    payload: user
  });
}

type Actions = AddUserAction | EditUserAction ...
```

# RETURN TYPE

```
const addUser = function(user: User) {  
  return ({  
    type: 'ADD_USER',  
    payload: user  
  });  
}  
  
type Actions = ReturnType<typeof addUser> | ...
```

# PARTIAL

```
export interface User {  
  id: string;  
  name: string;  
  email: string;  
}  
  
const userFields: User = { // ← TS error  
  name: this.nameControl.value,  
  email: this.emailControl.value  
};  
  
const userFields: Partial<User> = {  
  name: '',  
  email: ''  
};
```

# MANUTENIBILITÀ

- Complessità ciclomatica
- Profondità di annidamento
  - Numero di parametri
  - Linee di codice per file
- Statement per funzione

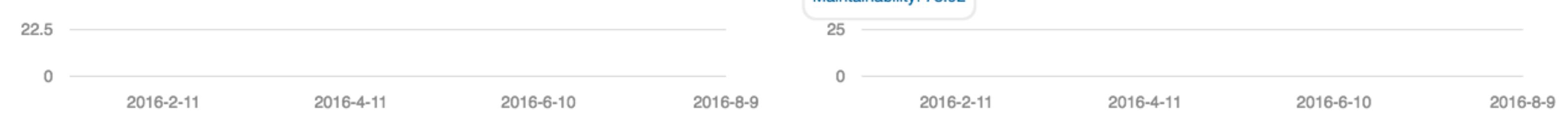
# ESLINT - MANUTENIBILITÀ

max-depth	enforce a maximum depth that blocks can be nested
max-len	enforce a maximum line length
max-lines	enforce a maximum number of lines per file
max-nested-callbacks	enforce a maximum depth that callbacks can be nested
max-params	enforce a maximum number of parameters in function definitions
max-statements	enforce a maximum number of statements allowed in function blocks
max-statements-per-line	enforce a maximum number of statements allowed per line

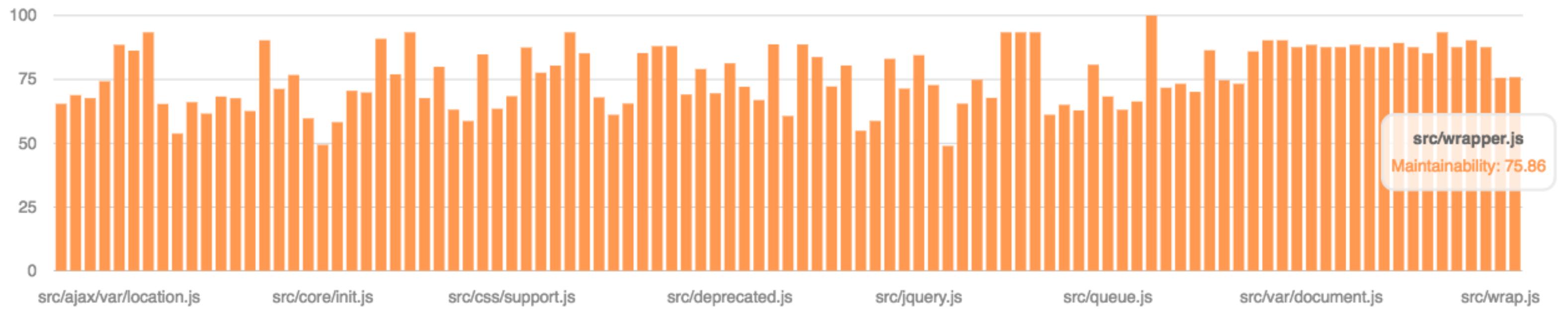
# MANUTENIBILITÀ

## Plato

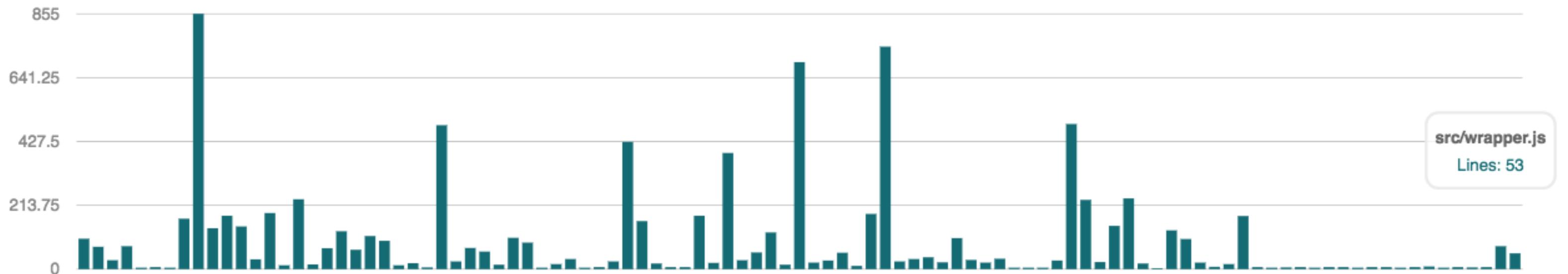
- Errori di implementazione Halstead
- Indice di manutenibilità



## Maintainability ⓘ



## Lines of code ⓘ





No ES6/TS support 😞

# PORTABILITÀ

# PORTABILITÀ

- BrowserStack
- Sauce Labs
- [eslint-plugin-compat](#)

App.js

package.json

.eslintrc

```
// Make a network request using fetch
```

```
f|
```

```
k finally
```

```
k for
```

```
k function
```

```
f Float32Array(size: number)
```

```
f Float64Array(size: number)
```

```
f Function(body: string)
```

```
→| f
```

anonymous function

```
→| fa
```

function apply

```
→| fb
```

function bind

```
→| fc
```

function call

# WORKFLOW

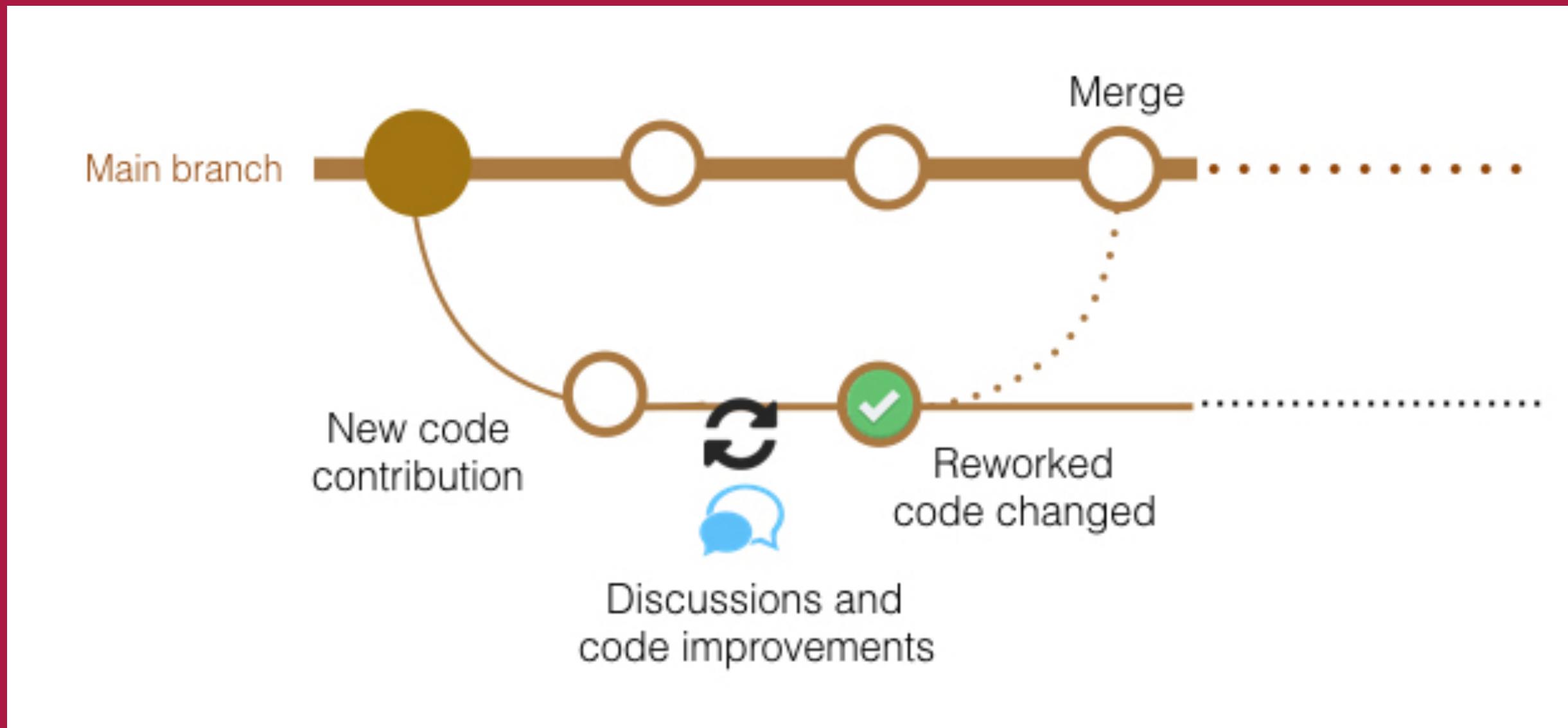
# WORKFLOW - BEFORE COMMITTING

1. Linter
2. Editor/IDE plugins
3. Git hooks

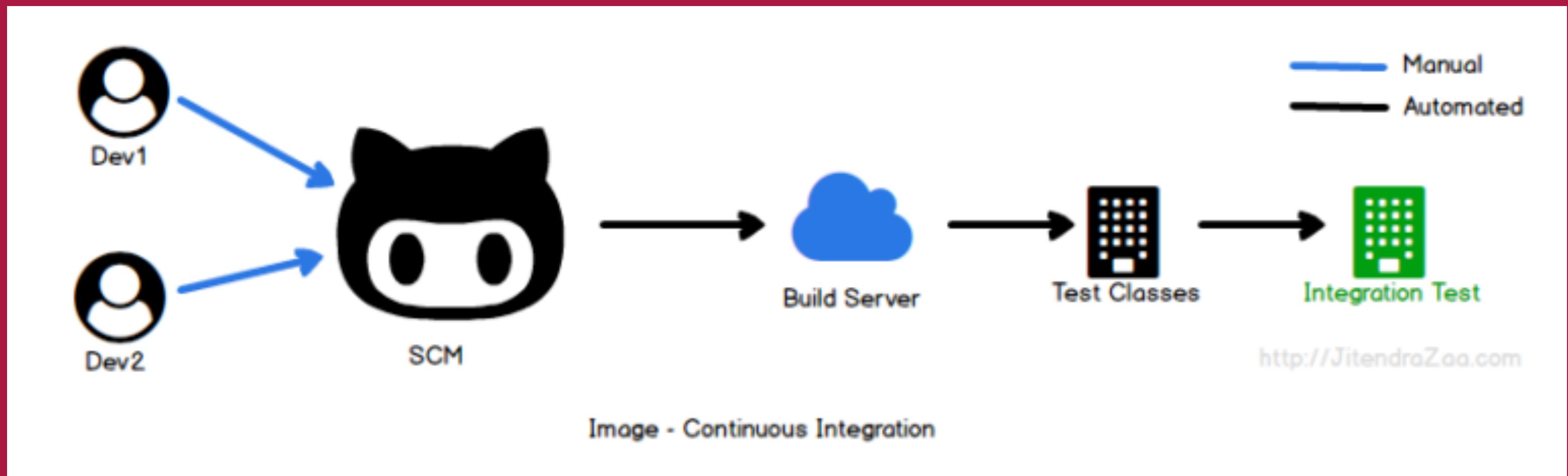
# WORKFLOW - AFTER COMMITTING

1. Pull Requests → 😎
2. Continuous Integration
3. Code review
4. Merge ✌

# 1. PULL REQUESTS



## 2. CI



## 2. CI

1. L'applicazione deve compilare
2. Tutti i test devono passare
3. Tutte le metriche devono passare, ad  
bundlesize

# 3. CODE REVIEW

1. Non riguarda code style
2. Non riguarda aspetti automatizzabili
3. Premiare ciò che è giusto
4. Opportunità per condividere ed imparare

# 3. CODE REVIEW

- Correttezza logica, design patterns, performance, sicurezza etc.
- Palantir - Code review best practises
- Analisi delle soluzioni adottate
- Fare una checklist

# I PECCATI DEVONO ESSERE PUBBLICI

1. Build fallite
2. Notifiche Slack

LA QUALITÀ DEVE ESSERE  
PREMIATA

BASTA INCENTIVARE I  
COMPORTAMENTI  
ERRATI



WWDC18

# CONCLUSIONI

- La qualità è uno sforzo continuo
  - Cultura della Qualità
  - Community

...?

- What about Quality Assurance, Quality Management, Quality Control etc.?
  - Qualità di Processo
  - TDD

Maggior attenzione alla qualità e ricerca di misure per migliorarla

JIA YI HU

Front-end developer & consultant

- <https://github.com/jiayihu>
- [https://twitter.com/jiayi\\_ghu](https://twitter.com/jiayi_ghu)
- [jiayi.ghu@gmail.com](mailto:jiayi.ghu@gmail.com)

A proposito di  
miglioramento continuo...

<https://joind.in/event/angular-day-2018>