The persuit of Quality

The  of Quality

The pursuit 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

L'assenza di qualità è un problema sia per voi che per il cliente (soprattutto).

Possiamo

1. **ignorarla**
2. **cercare di migliorarla**

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. **Pigrizia**
2. **Determinismo**
3. **Risultati analizzabili**
4. **Efficiente**
5. **Pigrizia**

Pigrizia

Ricercare la qualità non deve essere arduo

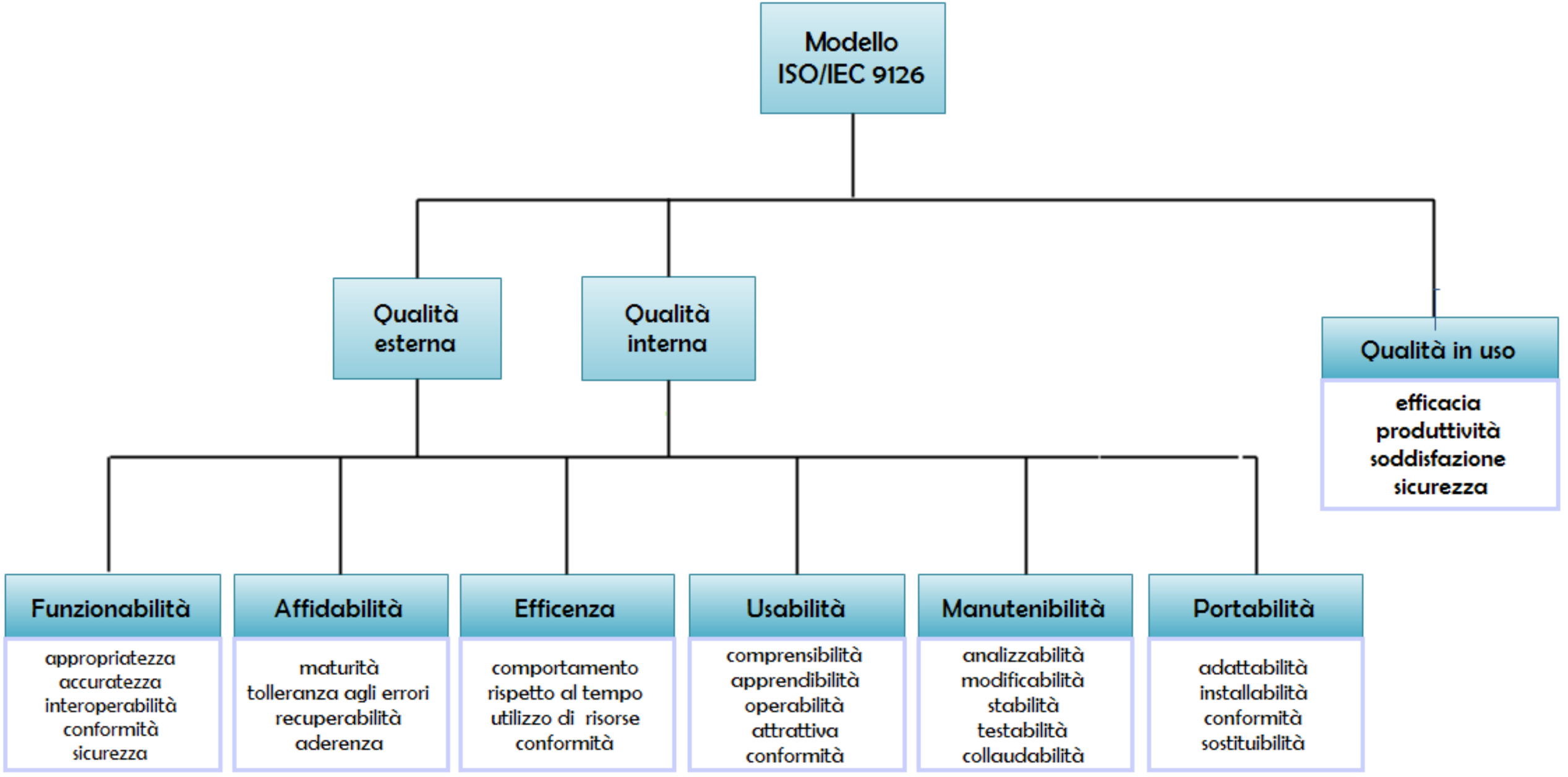
Una ricerca preventiva

1. Linters e plugin editor/IDE: TSLint/ESLint
   - tslint-microsoft
   - typescript-eslint-parser
2. Git hooks: husky 🐕
3. CI: Travis, CircleCI 👮
4. ~~Quality Assurance/Utenti~~

Incanalate l'odio

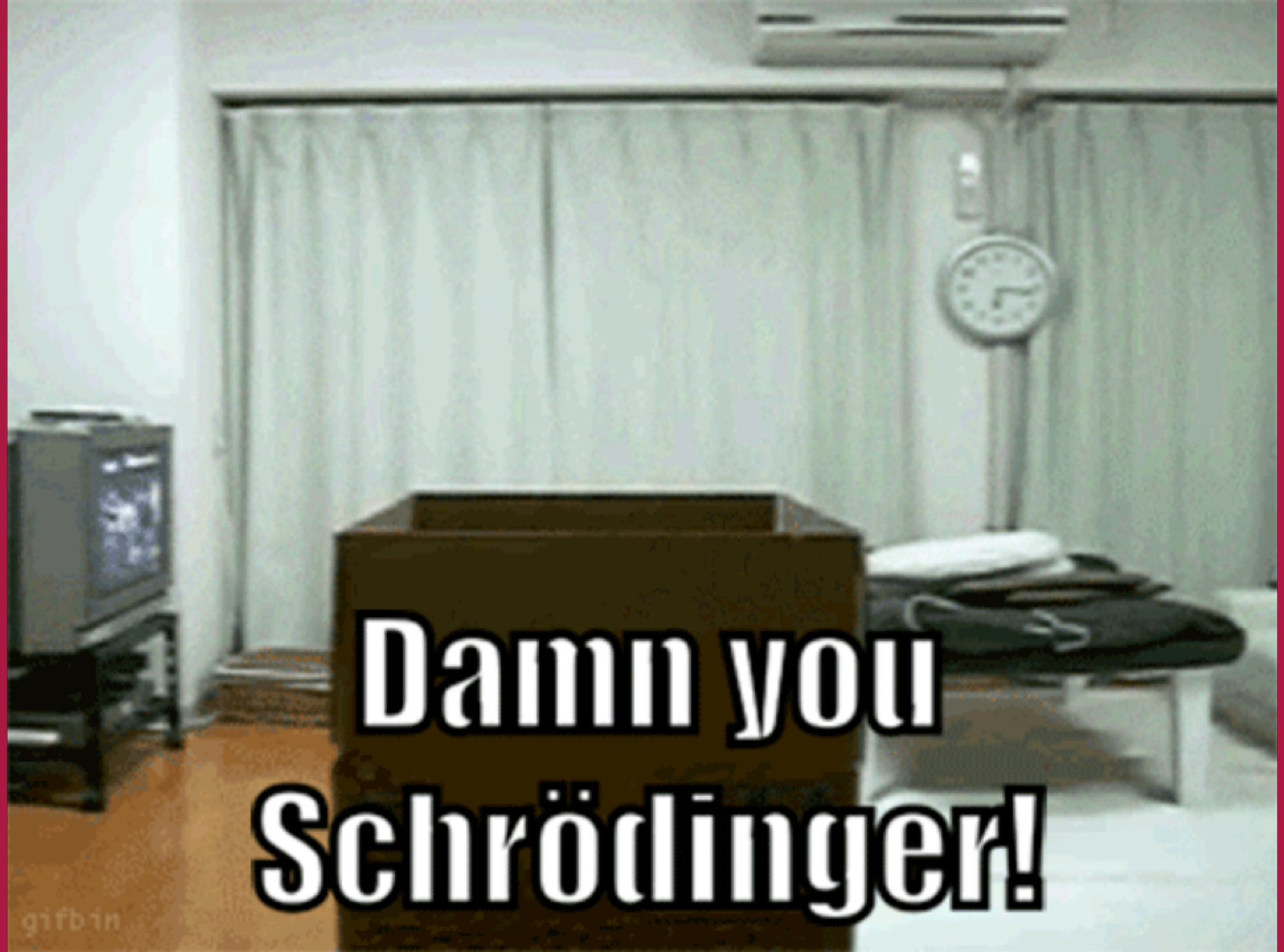
Gli strumenti automatici evitano che i vostri colleghi odino voi.

ISO 9126


Modello ISO/IEC 9126
Qualità esterna
Qualità interna
Qualità in uso
efficacia
produttività
soddisfazione
sicurezza
Funzionabilità
appropriatezza
accuratezza
interoperabilità
conformità
sicurezza
Affidabilità
maturità
tolleranza agli errori
recuperabilità
aderenza
Efficenza
comportamento rispetto al tempo
utilizzo di risorse
conformità
Usabilità
comprensibilità
apprendibilità
operabilità
attrattiva
conformità
Manutenibilità
analizzabilità
modificabilità
stabilità
testabilità
collaudabilità
Portabilità
adattabilità
installabilità
conformità
sostituibilità

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!
gifbin

Testing

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

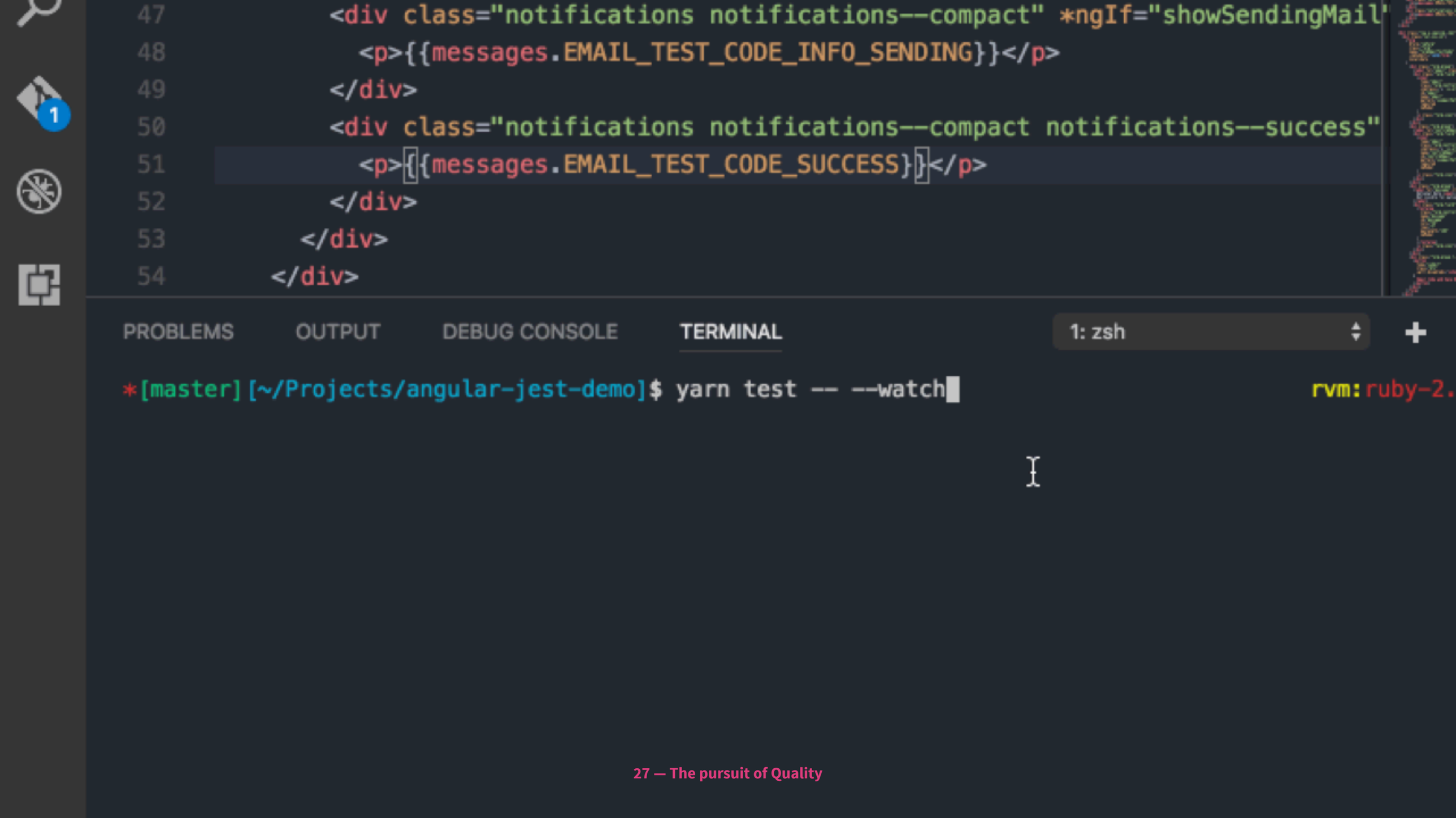
Testing

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 {{messages.EMAIL_TEST_CODE_INFO_SENDING}}
49
50 <div class="notifications notifications--compact notifications--success"
51 {{messages.EMAIL_TEST_CODE_SUCCESS}}
52
53
54
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: zsh
*[master] [~/Projects/angular-jest-demo]$ yarn test -- --watch
rvm:ruby-2.

Testing

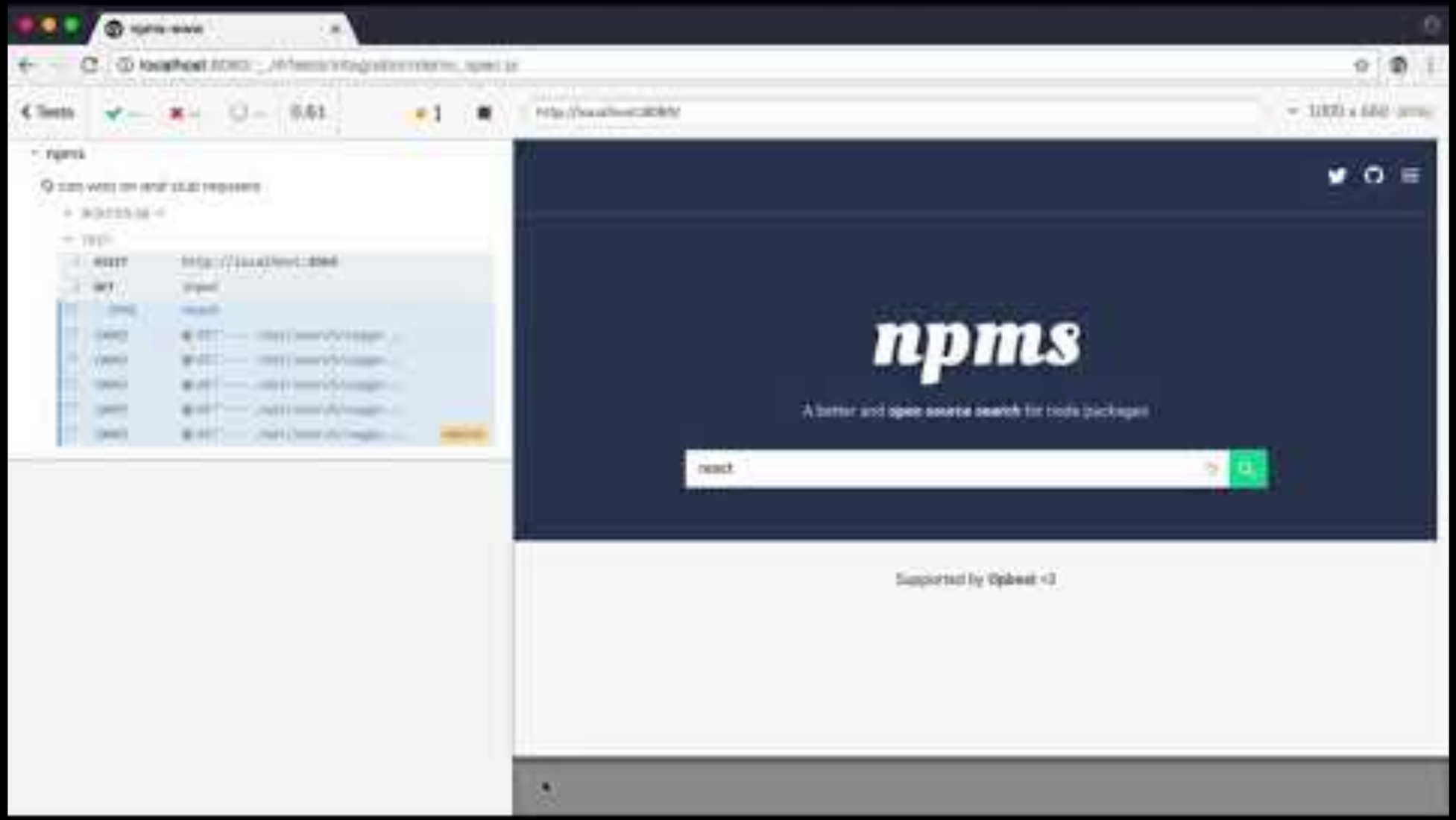
2) Integration tests

Write mostly integration tests

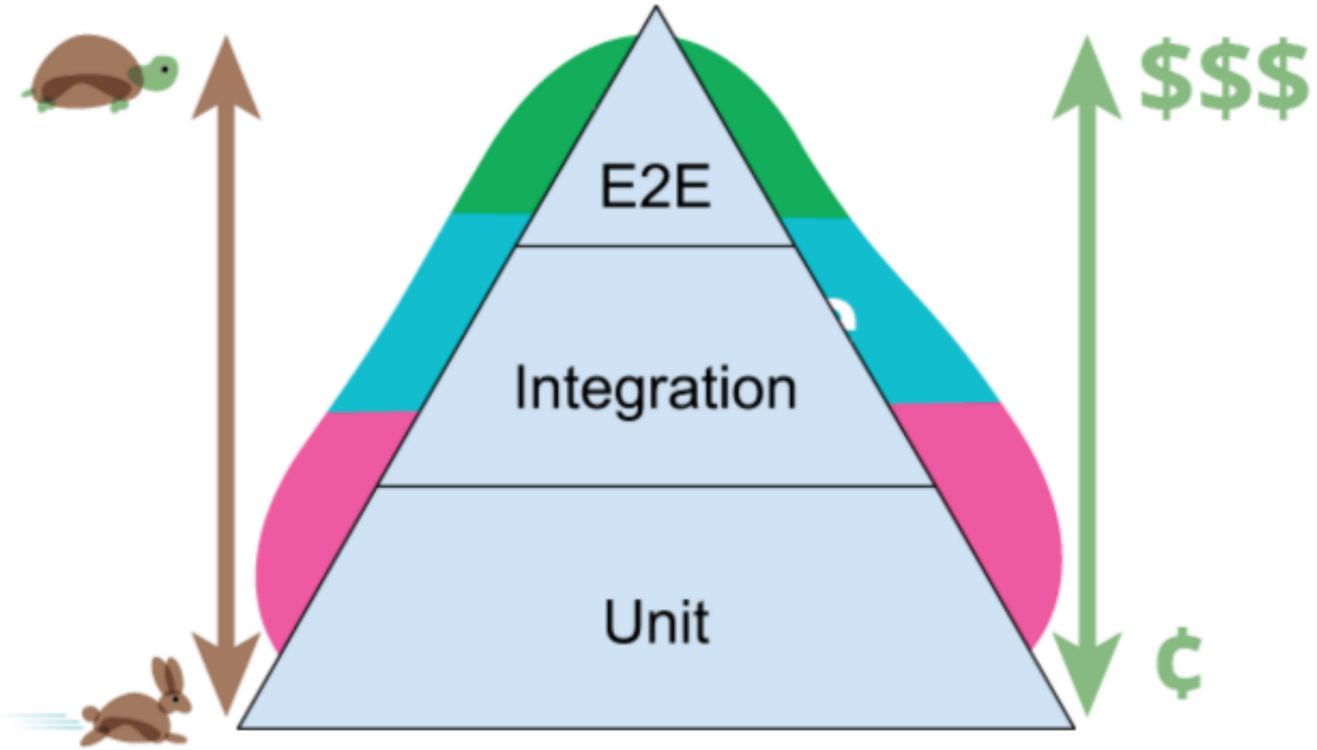
Testing

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

Cypress with Angular & Typescript


npms
A better and open source search for node packages
react
Supported by Opbeat <3

Where do we focus our time?



martinfowler.com/bliki/TestPyramid.html

testing.googleblog.com/2015/04/just-say-no-to-more-end-to-end-tests.html

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();
  }));
```

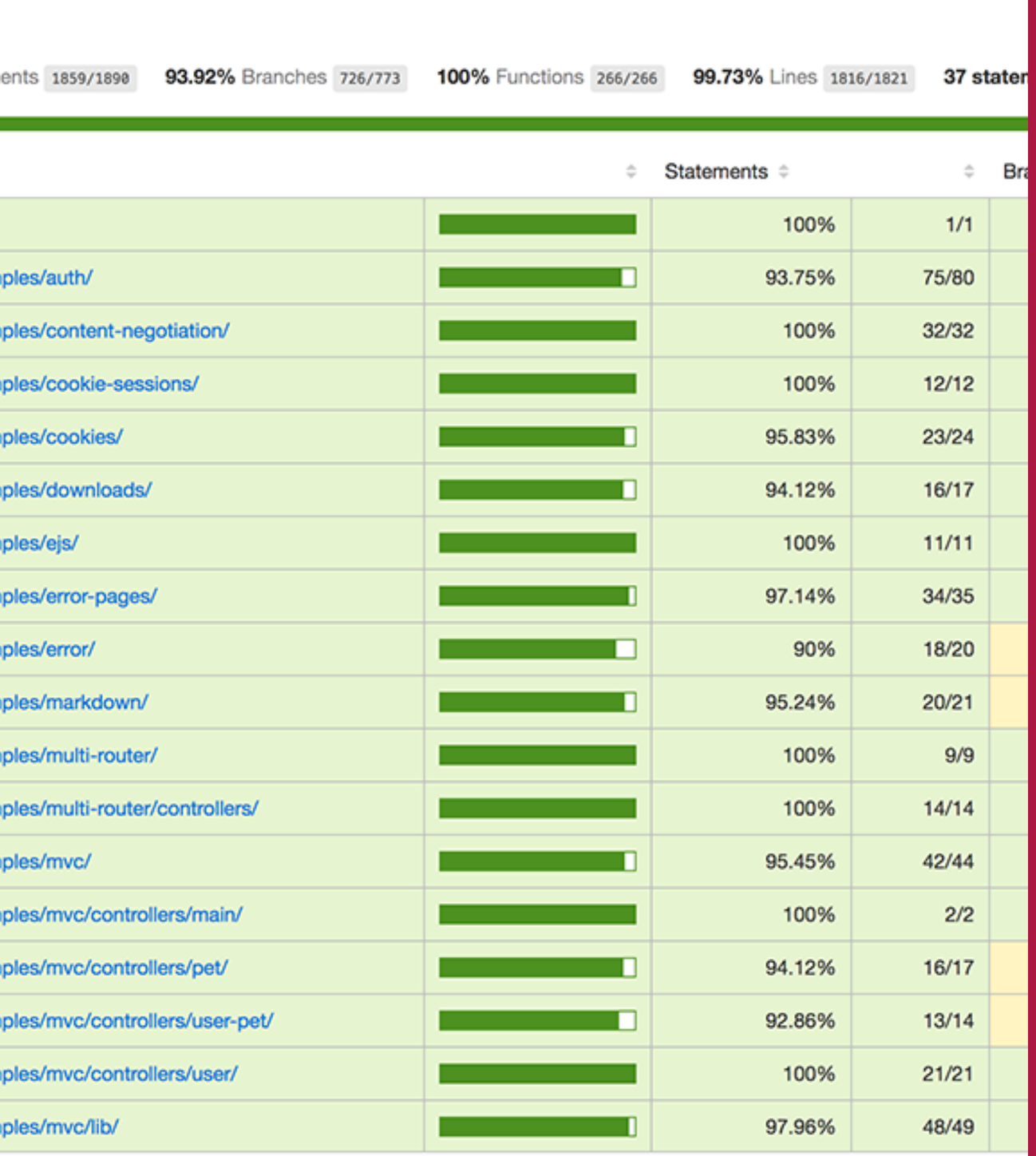
Affidabilità

Affidabilità

- **Testing (again)**
- **Coverage**

"Testing shows the presence, not the absence of bugs"

— Dijkstra (1969)



ents 1859/1890 **93.92%** Branches 726/773 **100%** Functions 266/266 **99.73%** Lines 1816/1821 **37** stater

	Statements	Br
	100%	1/1
ples/auth/	93.75%	75/80
ples/content-negotiation/	100%	32/32
ples/cookie-sessions/	100%	12/12
ples/cookies/	95.83%	23/24
ples/downloads/	94.12%	16/17
ples/ejs/	100%	11/11
ples/error-pages/	97.14%	34/35
ples/error/	90%	18/20
ples/markdown/	95.24%	20/21
ples/multi-router/	100%	9/9
ples/multi-router/controllers/	100%	14/14
ples/mvc/	95.45%	42/44
ples/mvc/controllers/main/	100%	2/2
ples/mvc/controllers/pet/	94.12%	16/17
ples/mvc/controllers/user-pet/	92.86%	13/14
ples/mvc/controllers/user/	100%	21/21
ples/mvc/lib/	97.96%	48/49

Coverage

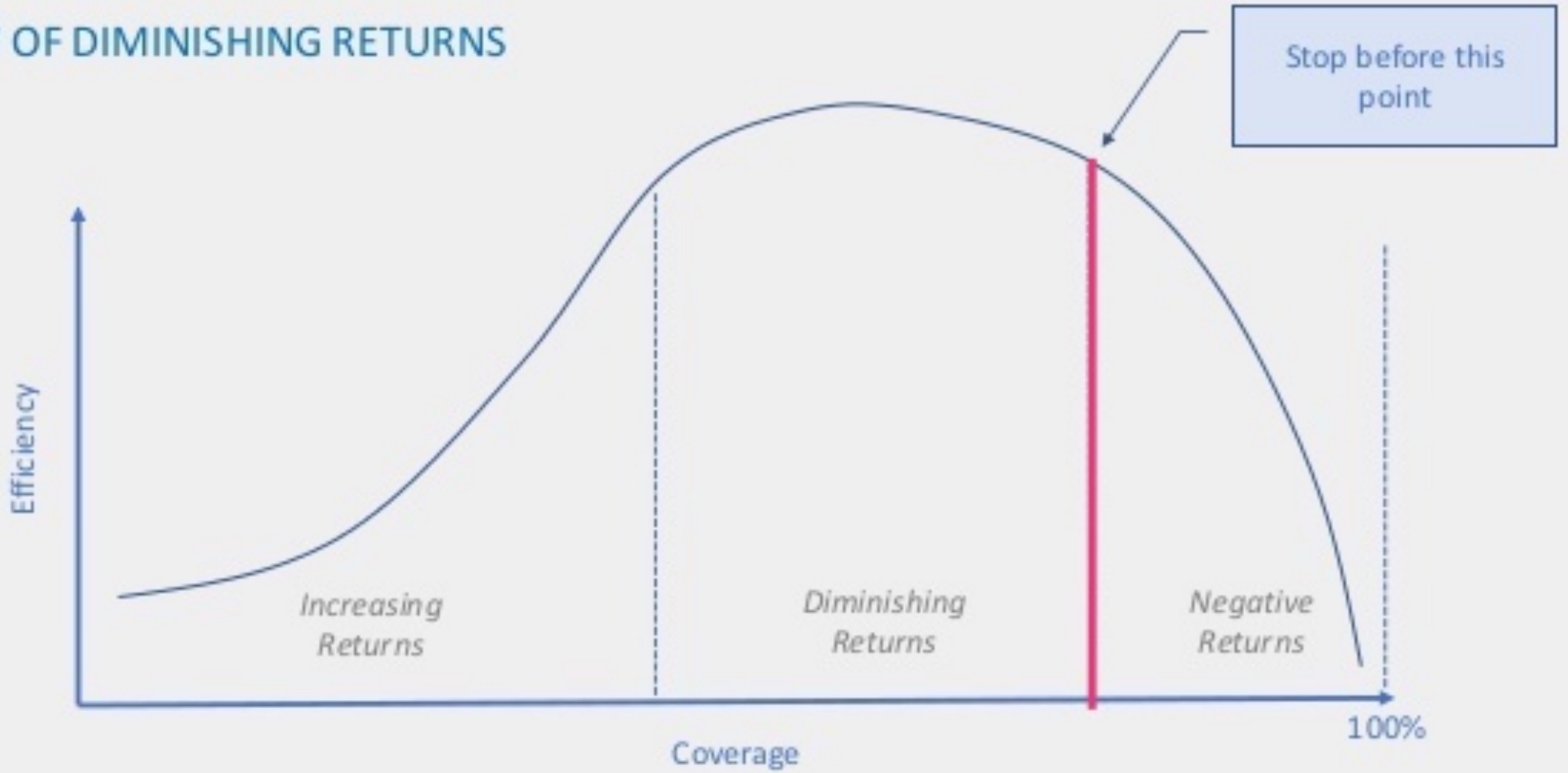
- **Istanbul**
- **Karma coverage/Jest**

Coverage

Statement coverage
vs
Branch coverage

How much coverage?

LAW OF DIMINISHING RETURNS



Efficienza

Efficienza

- **Complessità ciclomatica**
- **Indice Lighthouse**
- **Dimensione bundle**
- **Puppeteer Devtools**

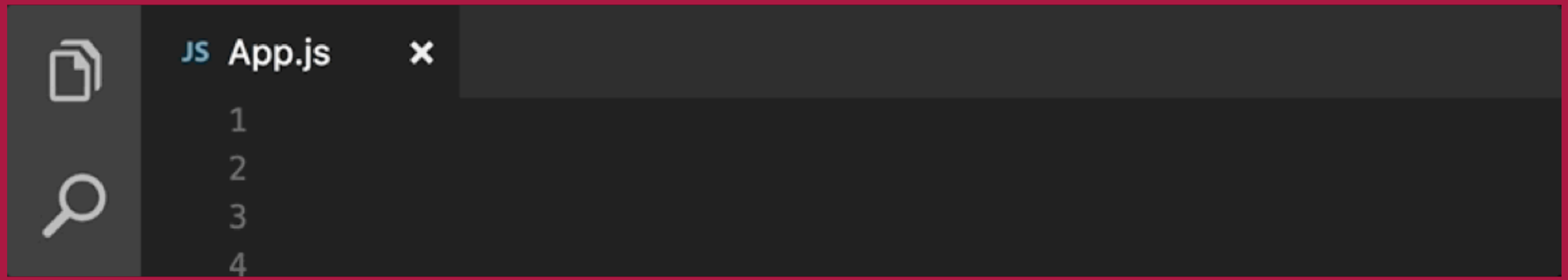
Complessità ciclomatica

```
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

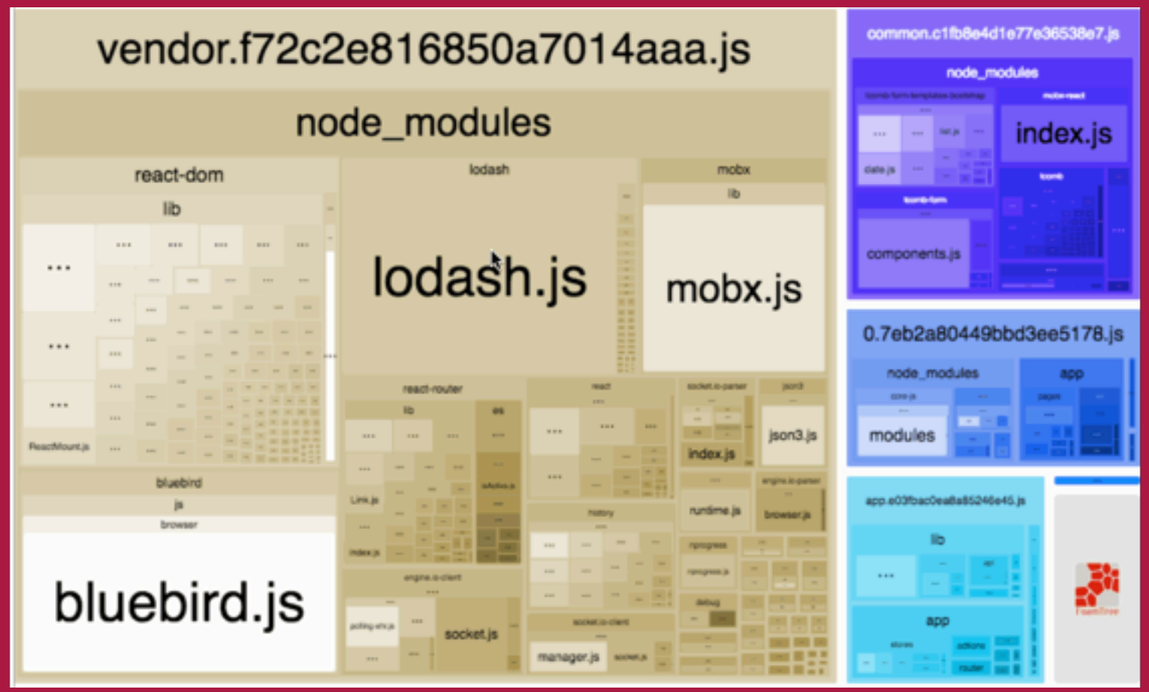


Dimensioni bundle

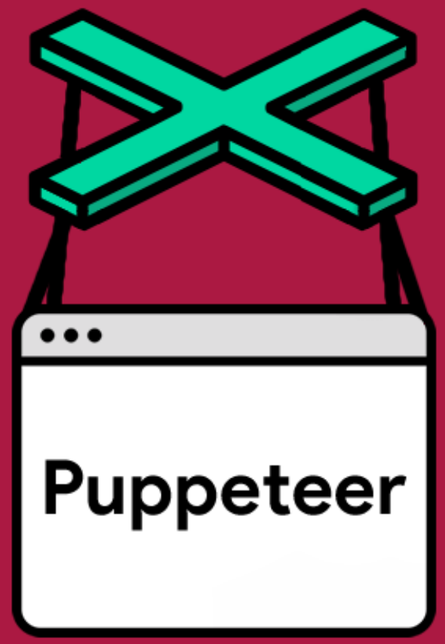
bundlesize

Dimensioni bundle

webpack-bundle-analyzer



Puppeteer Devtools



DevTools Protocol - Memory

```
const puppeteer = require('puppeteer');

(async () => {
  const args = await puppeteer.defaultArgs();
  const browser = await puppeteer.launch({
    headless: false,
    ignoreDefaultArgs: true,
    args
  });
  const page = await browser.newPage();
  const devtoolsProtocolClient = await page.target().createCDPSession();
  await devtoolsProtocolClient.send('Overlay.setShowFPSCounter', { show: true });
  await page.goto('http://localhost:3000');
})();
```

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>
```

```ts
type DeepReadonly<T> =
    T extends any[] ? DeepReadonlyArray<T[number]> :
    T extends object ? DeepReadonlyObject<T> :
    T;

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

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

ReturnType

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


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

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

ReturnType

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

type Actions = ReturnType<typeof addUser> |   ...
```

Partial

```ts
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

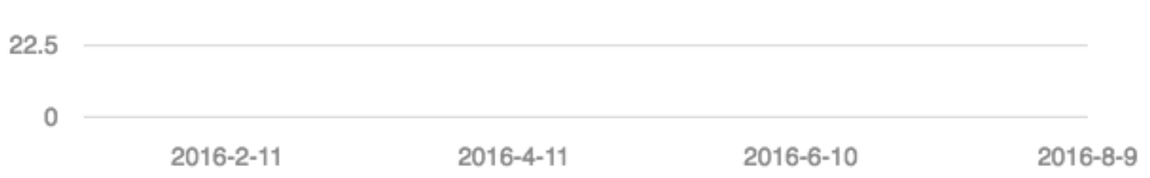
ESLint - Manutenibilità

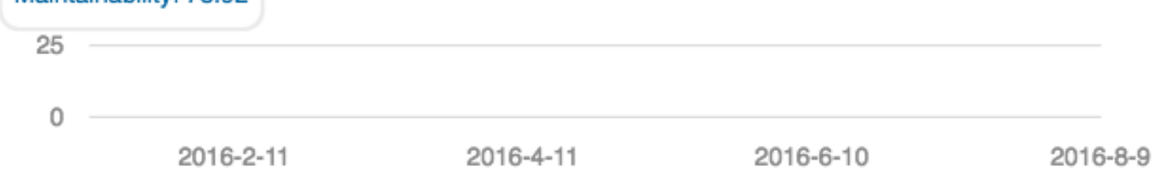
max-depth	enforce a maximum depth that blocks can be nested
max-len	enforce a maximum line length
max-lines	enforce a maximum number of lines per file
max-nested-callbacks	enforce a maximum depth that callbacks can be nested
max-params	enforce a maximum number of parameters in function definitions
max-statements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

Manutenibilità

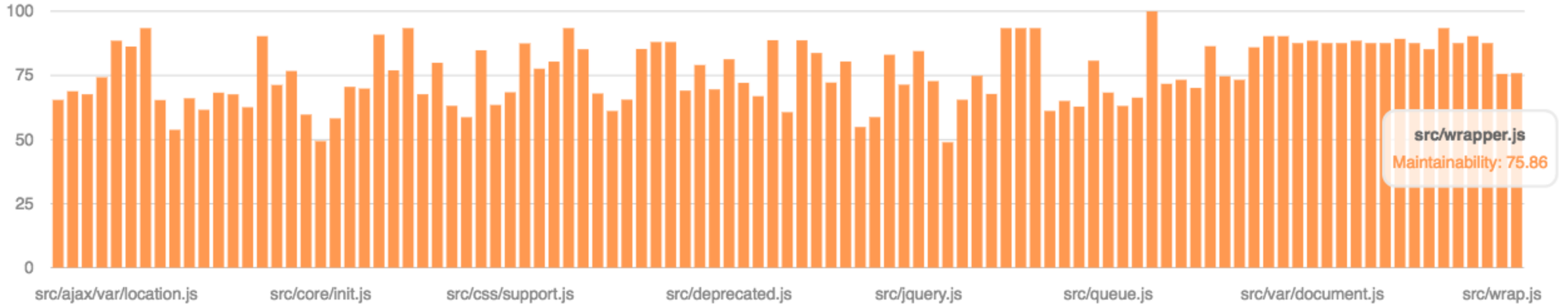
Plato

- Errori di implementazione Halstead
- Indice di manutenibilità

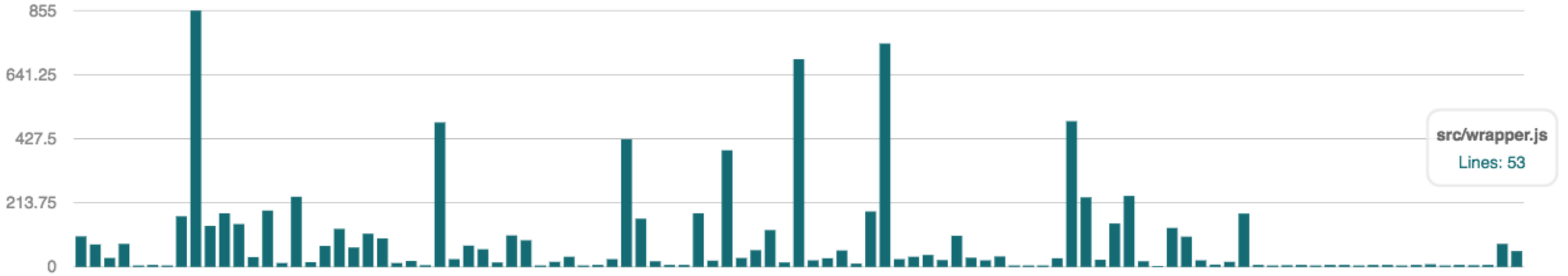




Maintainability



Lines of code



No ES6/TS support 🙃

Portabilità

Portabilità

- **BrowserStack**
- **Sauce Labs**
- **eslint-plugin-compat**
- **Puppeteer - Page emulation**

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

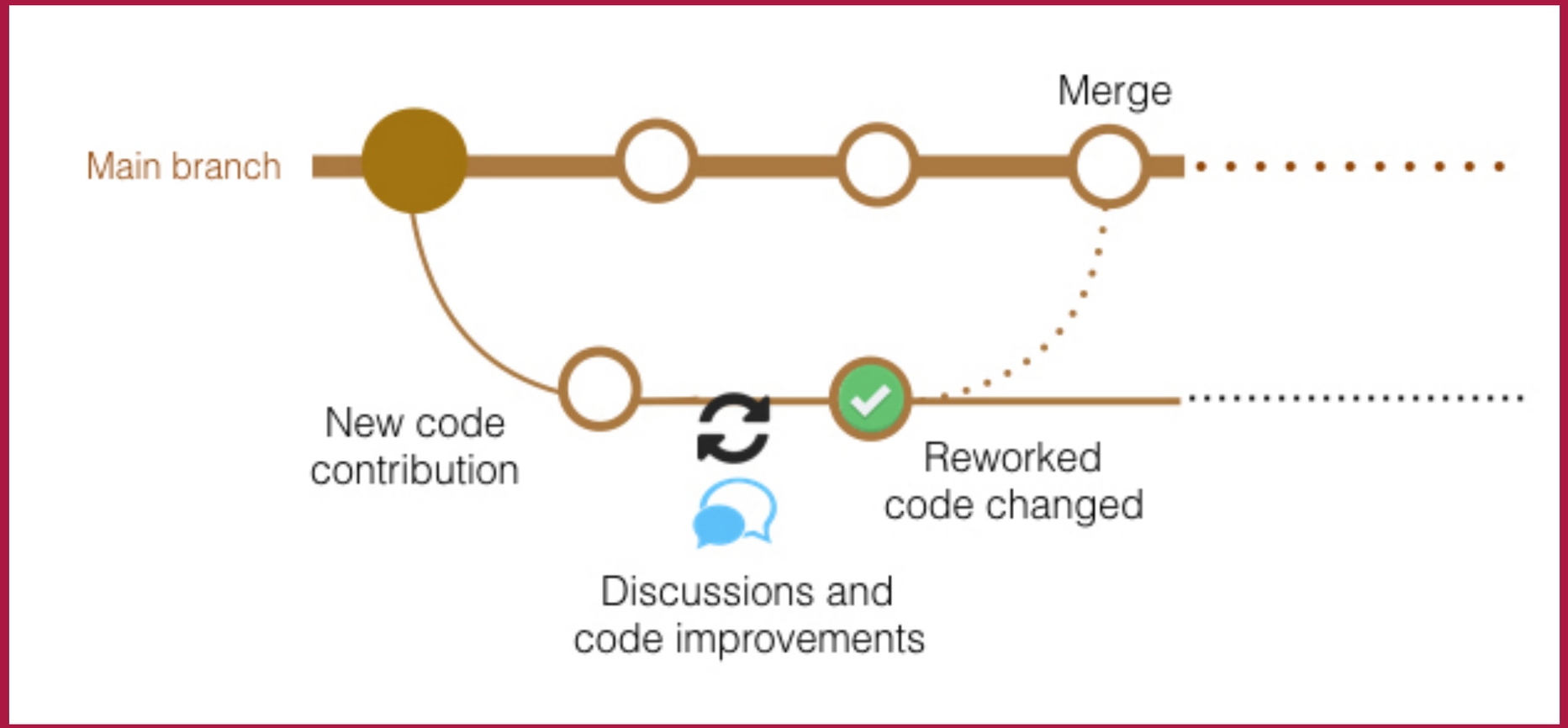
Workflow - Before committing

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

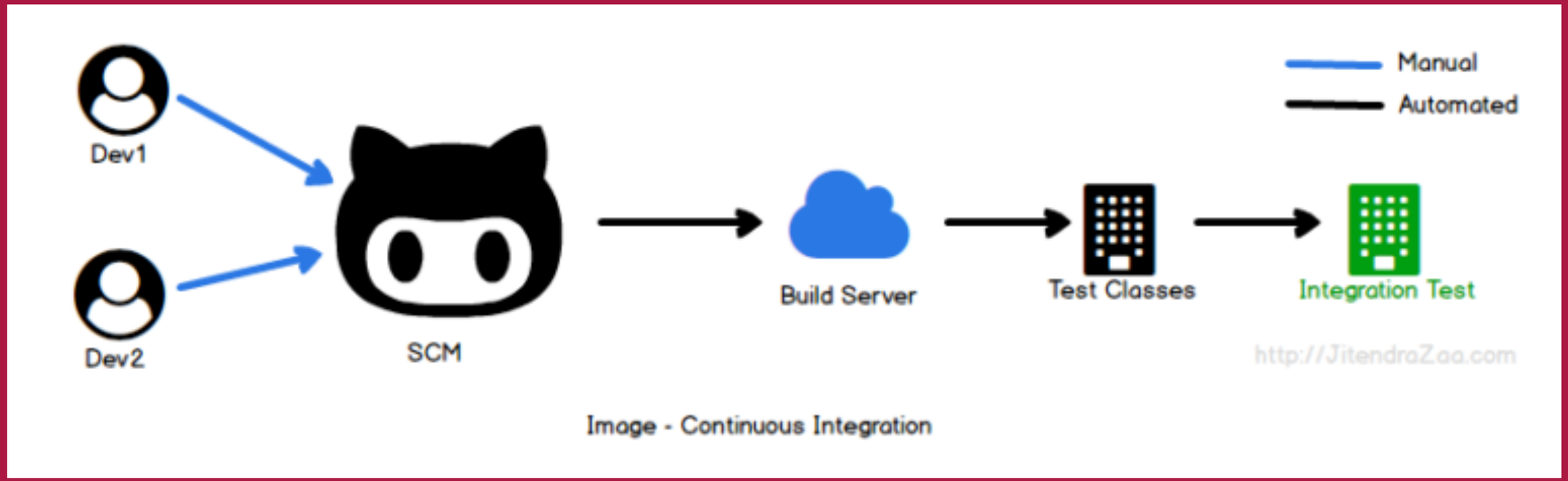


Image - Continuous Integration

2. CI

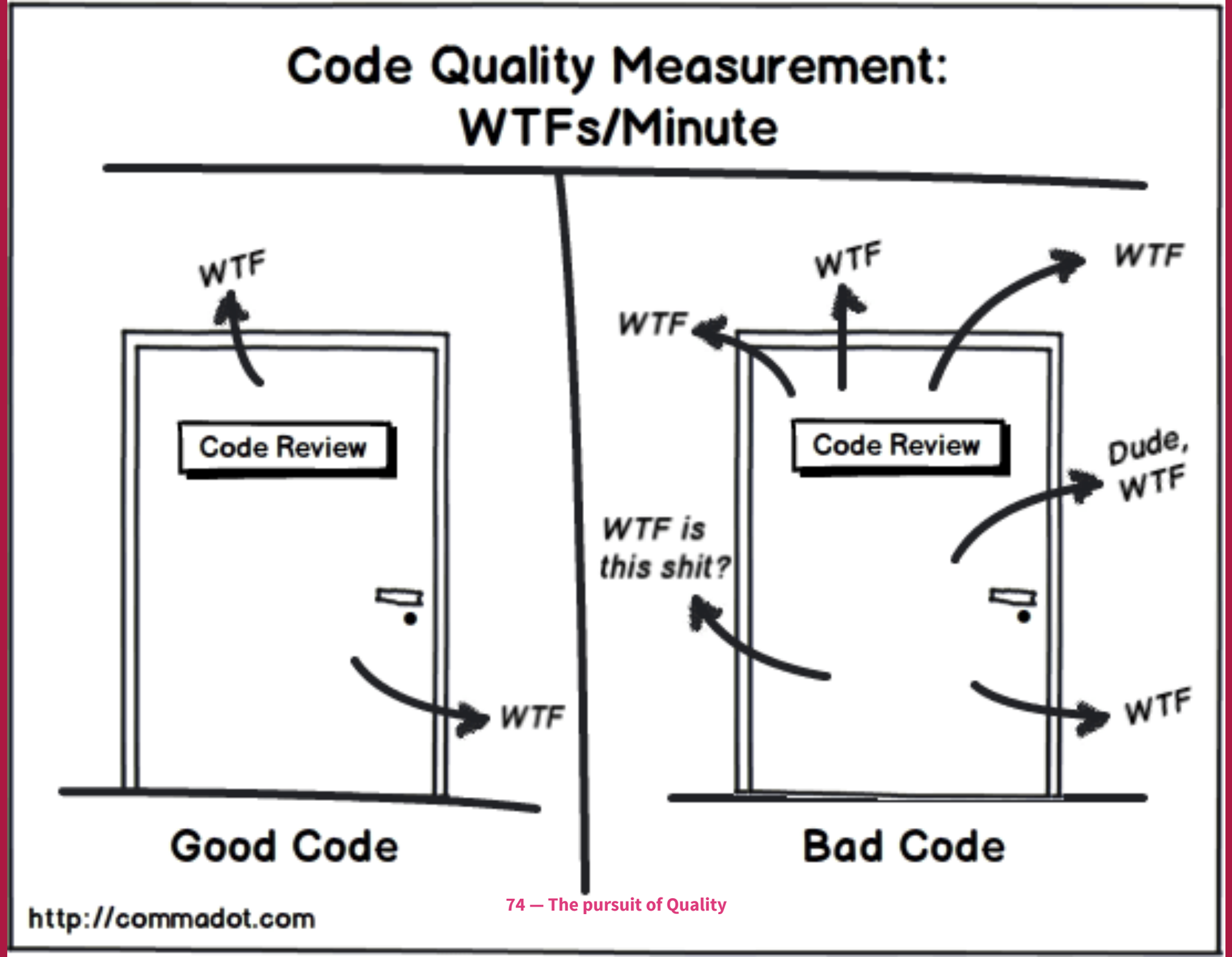
1. L'applicazione deve compilare
2. Tutti i test devono passare
3. Tutte le metriche devono passare, ad esempio bundlesize

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**


Code Quality Measurement:
WTFs/Minute
WTF
Code Review
WTF
Good Code
WTF
WTF
WTF
WTF
Code Review
Dude,
WTF
WTF is
this shit?
WTF
Bad Code
http://commadot.com

I peccati devono essere pubblici

1. **Build fallite**
2. **Notifiche Slack**

La qualità deve essere premiata

Basta incentivare i comportamenti errati

Dove trovare la qualità:

Open Source Community

Conclusioni

- **La qualità è uno sforzo continuo e collettivo**
- **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

Jiayi Hu

Front-end developer & consultant

- https://github.com/jiayihu
- https://twitter.com/jiayi_ghu
- jiayi.ghu@gmail.com

Slides

github.com/jiayihu/talks

A proposito di miglioramento continuo...

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