Лекция 7
Разработка с CRA.
Тестирование

Мартин Комитски

ЦРИТО
Центр Развития
ИТ-Образования

МФТИ

@ mail.ru group

План на сегодня

- Разработка с CRA
  - Все фичи
  - Стили, SCSS
  - UI Kits
  - Proxy
  - ...
- Storybook
- Тестирование
  - Что это и зачем?
  - Виды
  - TDD
  - Jest
  - Puppeteer
- TDD/BDD

Минутка бюрократии

- Внимание
- Отметки о посещении занятий
- Обратная связь о лекциях

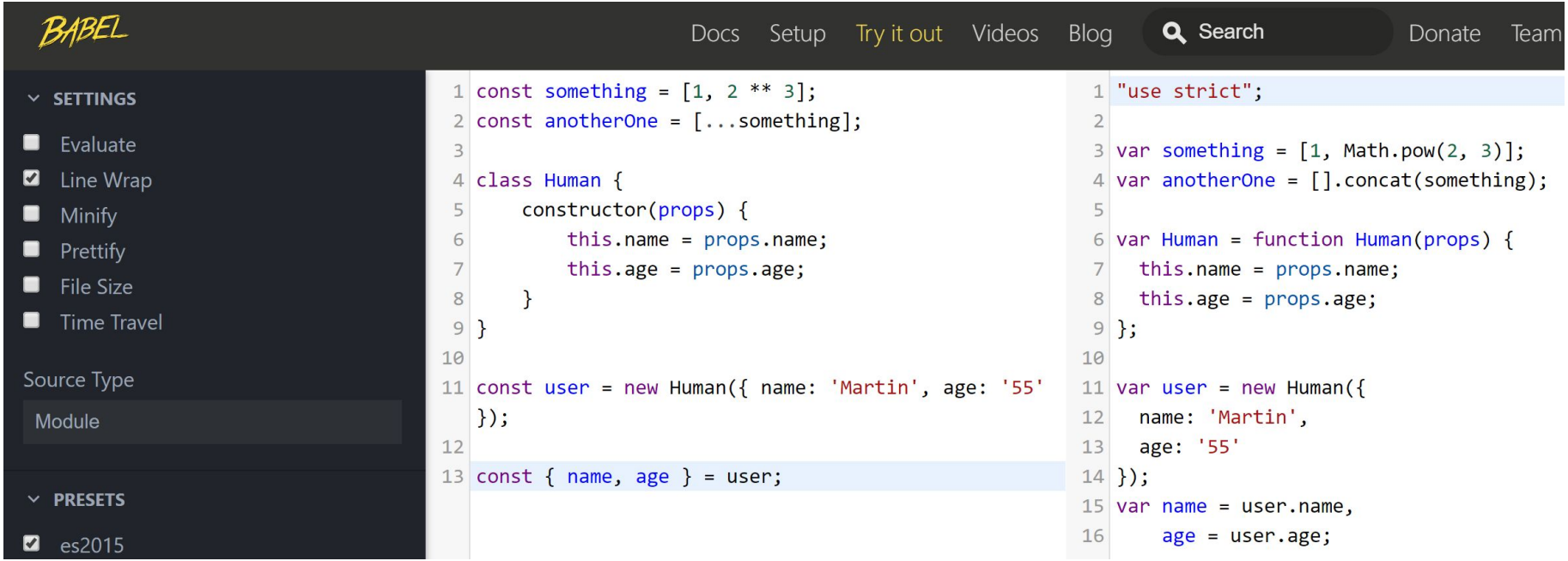
Все фичи CRA

Зависимости

CRA. Упрощенный список зависимостей

```
    "dependencies": {
      "@babel/core": "7.2.2", // babel
      "eslint": "5.12.0", // lint
      "jest": "23.6.0", // tests
      "react": "^16.8.6", // react
      "react-dom": "^16.8.6", // react-dom
      "react-app-polyfill": "^0.2.2", // Polyfills
      "react-dev-utils": "^8.0.0", // utils https://www.npmjs.com/package/react-dev-utils
      "html-webpack-plugin": "4.0.0-alpha.2", // work with html
      "mini-css-extract-plugin": "0.5.0", // optimize styles
      "optimize-css-assets-webpack-plugin": "5.0.1", // optimize styles
      "postcss-loader": "3.0.0", // postcss preset-env { autoprefixer }
      "sass-loader": "7.1.0", // load scss styles
      "style-loader": "0.23.1", // load styles
      "css-loader": "1.0.0", // load styles
      "@svgr/webpack": "4.1.0", // load files
      "file-loader": "2.0.0", // load files
      "url-loader": "1.1.2", // load files
      "webpack": "4.28.3", // webpack
      "webpack-dev-server": "3.1.14", // dev server
      "webpack-manifest-plugin": "2.0.4", // pwa manifest
      "workbox-webpack-plugin": "3.6.3" // service workers
    },
```

CRA. Babel


BABEL
Docs
Setup
Try it out
Videos
Blog
Search
Donate
Team
SETTINGS
Evaluate
Line Wrap
Minify
Prettify
File Size
Time Travel
Source Type
Module
PRESETS
es2015
const something = [1, 2 ** 3];
const anotherOne = [...something];
class Human {
constructor(props) {
this.name = props.name;
this.age = props.age;
}
}
const user = new Human({ name: 'Martin', age: '55' });
const { name, age } = user;
"use strict";
var something = [1, Math.pow(2, 3)];
var anotherOne = [].concat(something);
var Human = function Human(props) {
this.name = props.name;
this.age = props.age;
};
var user = new Human({
name: 'Martin',
age: '55'
});
var name = user.name,
age = user.age;

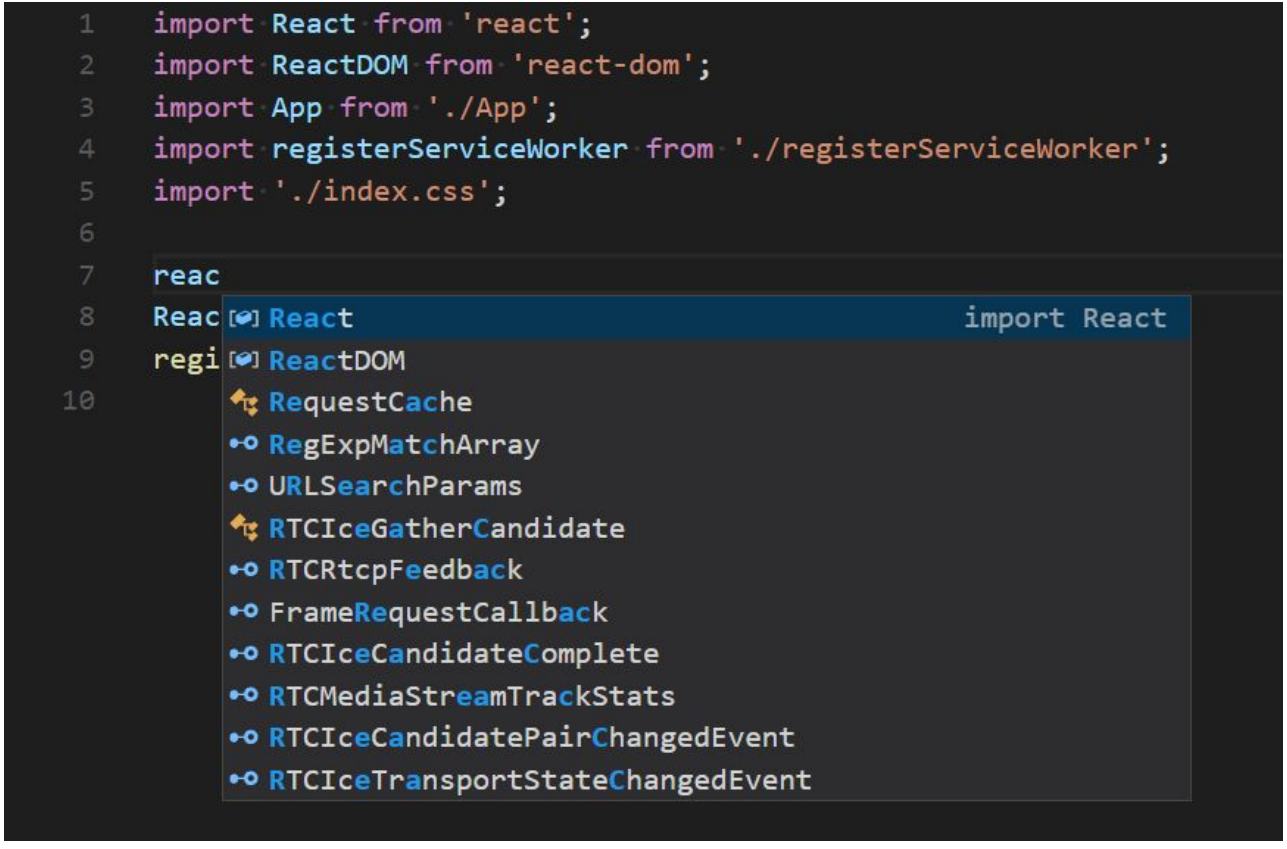
CRA. ESLint

```
-OSX:memphisww-eslint gspake$ ./node_modules/.bin/eslint .

/Users/gspake/code/memphisww-eslint/src/App.js
  5:1  error  Component should be written as a pure function  react/prefer-stateless-function
  8:7  error  JSX not allowed in files with extension '.js'   react/jsx-filename-extension

/Users/gspake/code/memphisww-eslint/src/App.test.js
  5:1   error  'it' is not defined                            no-undef
  6:15  error  'document' is not defined                      no-undef
  7:19  error  JSX not allowed in files with extension '.js'  react/jsx-filename-extension

/Users/gspake/code/memphisww-eslint/src/index.js
  7:17  error  JSX not allowed in files with extension '.js'  react/jsx-filename-extension
  7:26  error  'document' is not defined                      no-undef

/Users/gspake/code/memphisww-eslint/src/registerServiceWorker.js
   12:3   error    'window' is not defined                                                    no-undef
   14:5   error    'window' is not defined                                                    no-undef
   16:5   error    'window' is not defined                                                    no-undef
   17:63  error    Missing trailing comma                                                     comma-dangle
   18:6   error    Missing trailing comma                                                     comma-dangle
   22:67  error    'navigator' is not defined                                                 no-undef
   24:27  error    'URL' is not defined                                                       no-undef
   24:55  error    'window' is not defined                                                    no-undef
   25:30  error    'window' is not defined                                                    no-undef
   32:5   error    'window' is not defined                                                    no-undef
   37:9   error    'registerValidSW' was used before it was defined                           no-use-before-define
   40:9   error    'checkValidServiceWorker' was used before it was defined                   no-use-before-define
   47:3   error    'navigator' is not defined                                                 no-undef
   49:11  error    Expected parentheses around arrow function argument having a body with curly braces  arrow-parens
   50:7   error    Assignment to property of function parameter 'registration'                no-param-reassign
   54:17  error    'navigator' is not defined                                                 no-undef
   59:15  warning  Unexpected console statement                                               no-console
   64:15  warning  Unexpected console statement                                               no-console
   70:12  error    Expected parentheses around arrow function argument having a body with curly braces  arrow-parens
   71:7   warning  Unexpected console statement                                               no-console
   77:3   error    'fetch' is not defined                                                     no-undef
   78:11  error    Expected parentheses around arrow function argument having a body with curly braces  arrow-parens
   85:9   error    'navigator' is not defined                                                 no-undef
   85:44  error    Expected parentheses around arrow function argument having a body with curly braces  arrow-parens
   87:13  error    'window' is not defined                                                    no-undef
   96:7   warning  Unexpected console statement                                               no-console
   97:72  error    Missing trailing comma                                                     comma-dangle
  103:26  error    'navigator' is not defined                                                 no-undef
  104:5   error    'navigator' is not defined                                                 no-undef
  104:40  error    Expected parentheses around arrow function argument having a body with curly braces  arrow-parens

✖ 37 problems (33 errors, 4 warnings)
  8 errors, 0 warnings potentially fixable with the `--fix` option.
```

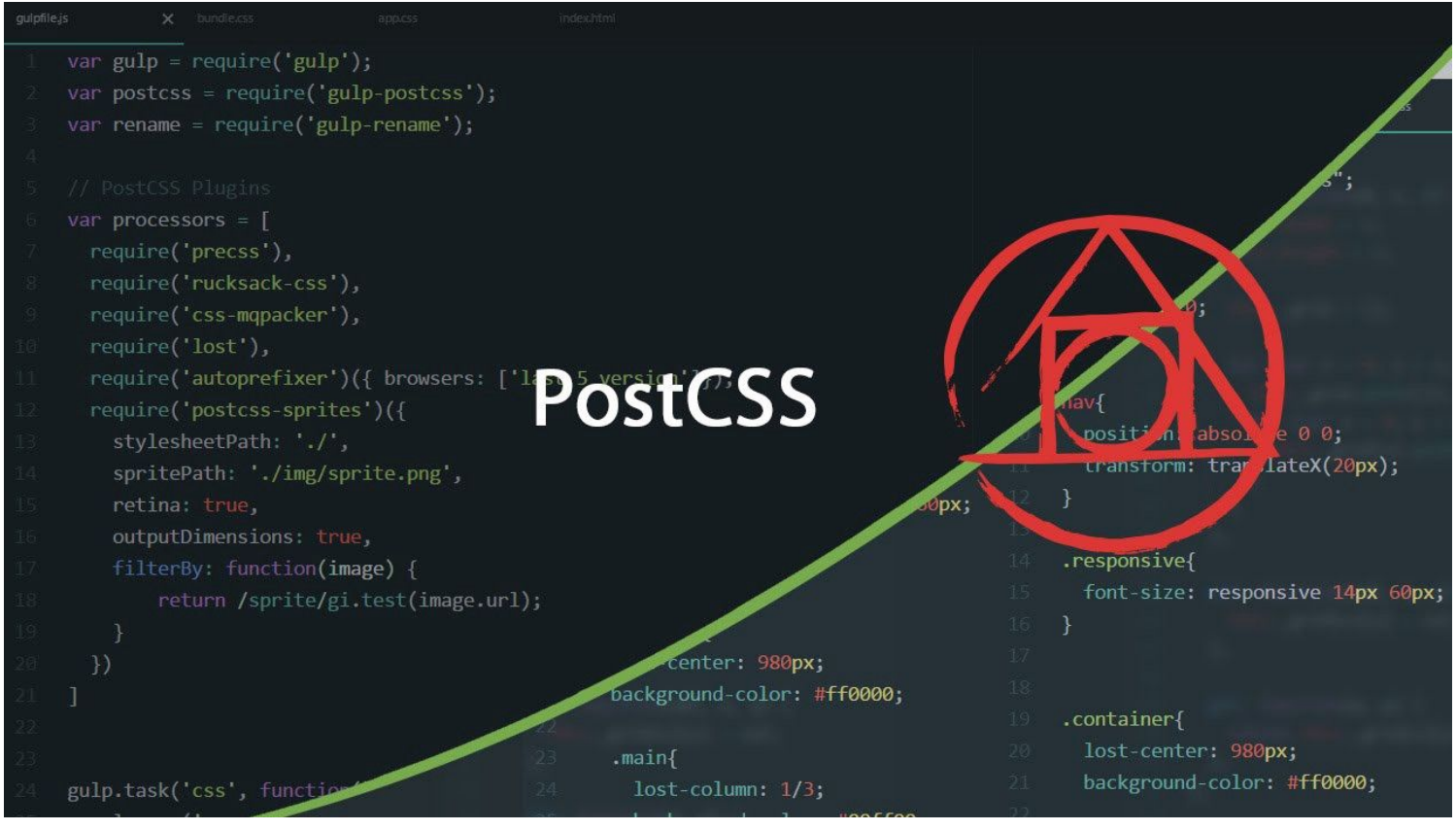
CRA. Jest

```
FAIL src/components/App.test.js
  Examining the syntax of Jest tests
    ✕ sums numbers (182ms)

  ● Examining the syntax of Jest tests › sums numbers

    expect(received).toEqual(expected)

    Expected value to equal:
      5
    Received:
      4

      3 |       it('sums numbers', () => {
      4 |         expect(1 + 2).toEqual(3);
    > 5 |         expect(2 + 2).toEqual(5);
        |                       ^
      6 |       });
      7 | });

      at Object.toEqual (src/components/App.test.js:5:18)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        0.965s, estimated 1s
Ran all test suites related to changed files.
```

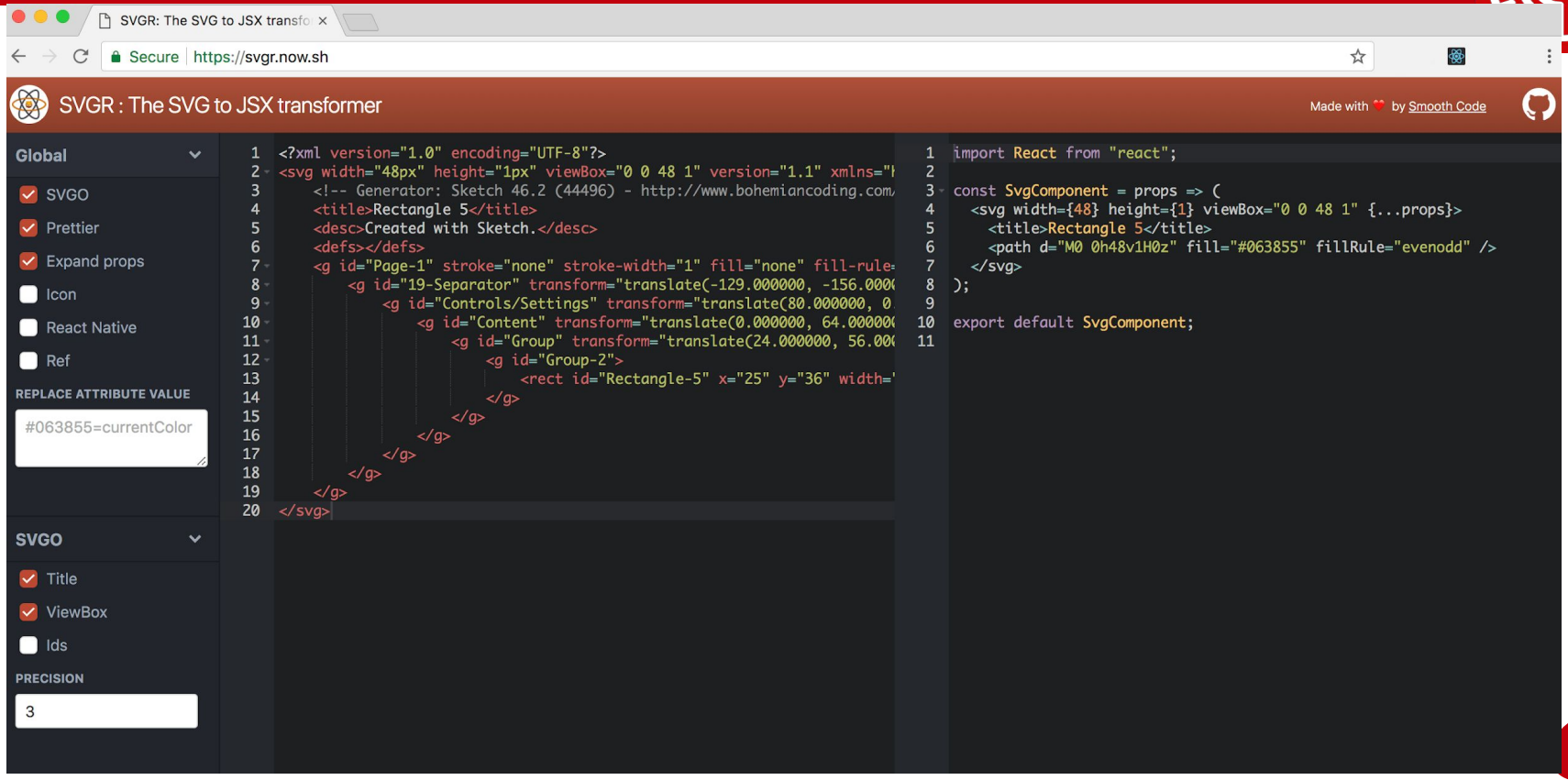
CRA. React/ReactDOM



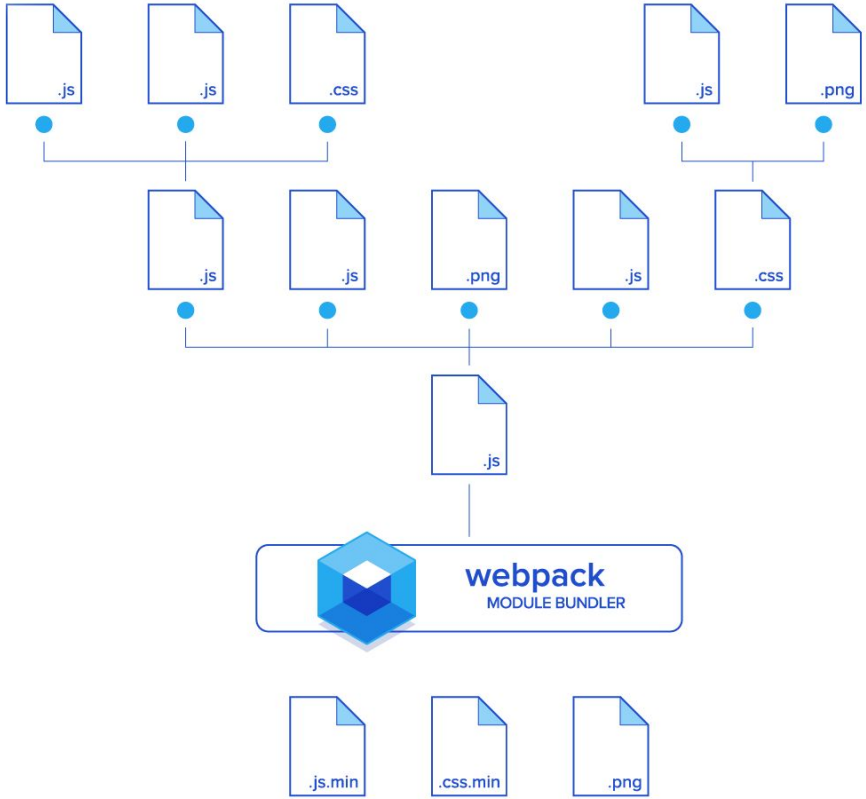
CRA. PostCSS {Autoprefixer}



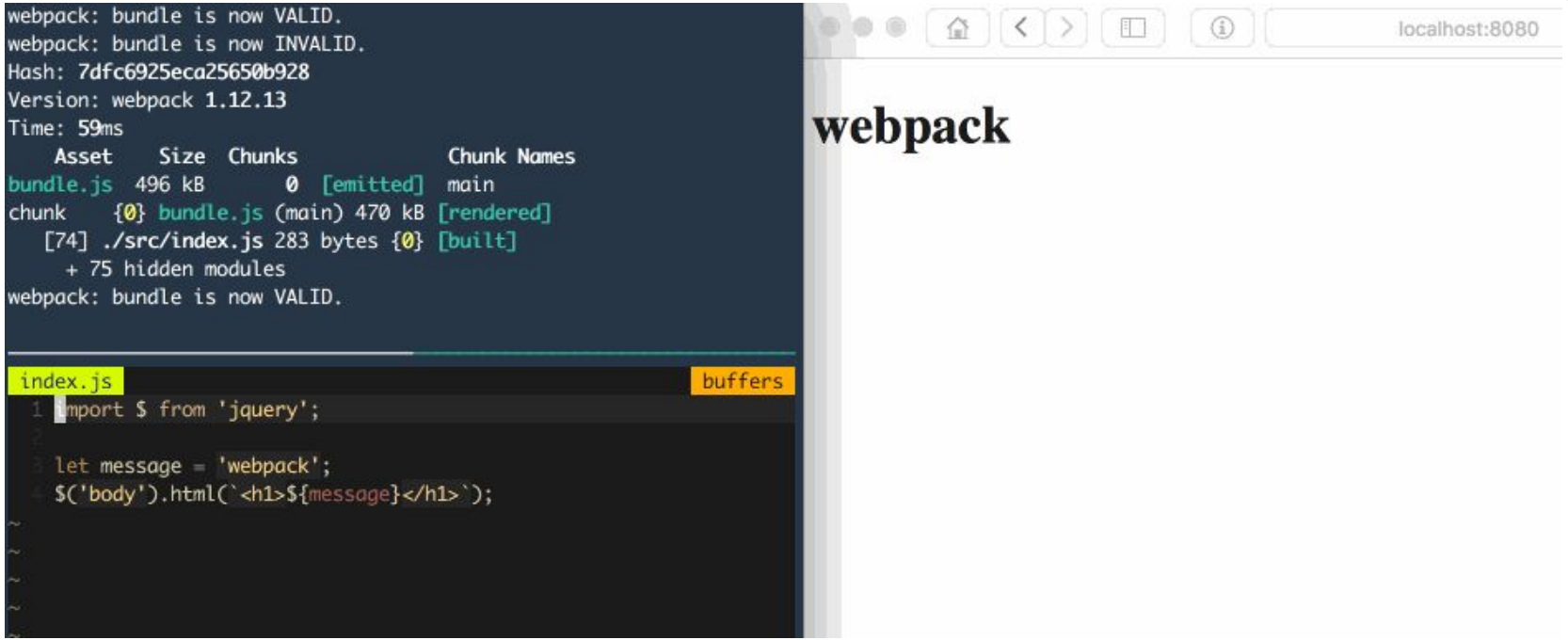
CRA. SVGR


SVGR: The SVG to JSX transfo
Secure
https://svgr.now.sh
SVGR : The SVG to JSX transformer
Made with ❤ by Smooth Code
Global
SVGO
Prettier
Expand props
Icon
React Native
Ref
REPLACE ATTRIBUTE VALUE
#063855=currentColor
SVGO
Title
ViewBox
Ids
PRECISION
3
<?xml version="1.0" encoding="UTF-8"?>
<svg width="48px" height="1px" viewBox="0 0 48 1" version="1.1" xmlns="
<!-- Generator: Sketch 46.2 (44496) - http://www.bohemiancoding.com/
<title>Rectangle 5</title>
<desc>Created with Sketch.</desc>
<defs></defs>
<g id="Page-1" stroke="none" stroke-width="1" fill="none" fill-rule=
<g id="19-Separator" transform="translate(-129.000000, -156.000
<g id="Controls/Settings" transform="translate(80.000000, 0
<g id="Content" transform="translate(0.000000, 64.00000
<g id="Group" transform="translate(24.000000, 56.00
<g id="Group-2">
<rect id="Rectangle-5" x="25" y="36" width="
</g>
</g>
</g>
</g>
</g>
</g>
</svg>
import React from "react";
const SvgComponent = props => (
<svg width={48} height={1} viewBox="0 0 48 1" {...props}>
<title>Rectangle 5</title>
<path d="M0 0h48v1H0z" fill="#063855" fillRule="evenodd" />
</svg>
);
export default SvgComponent;

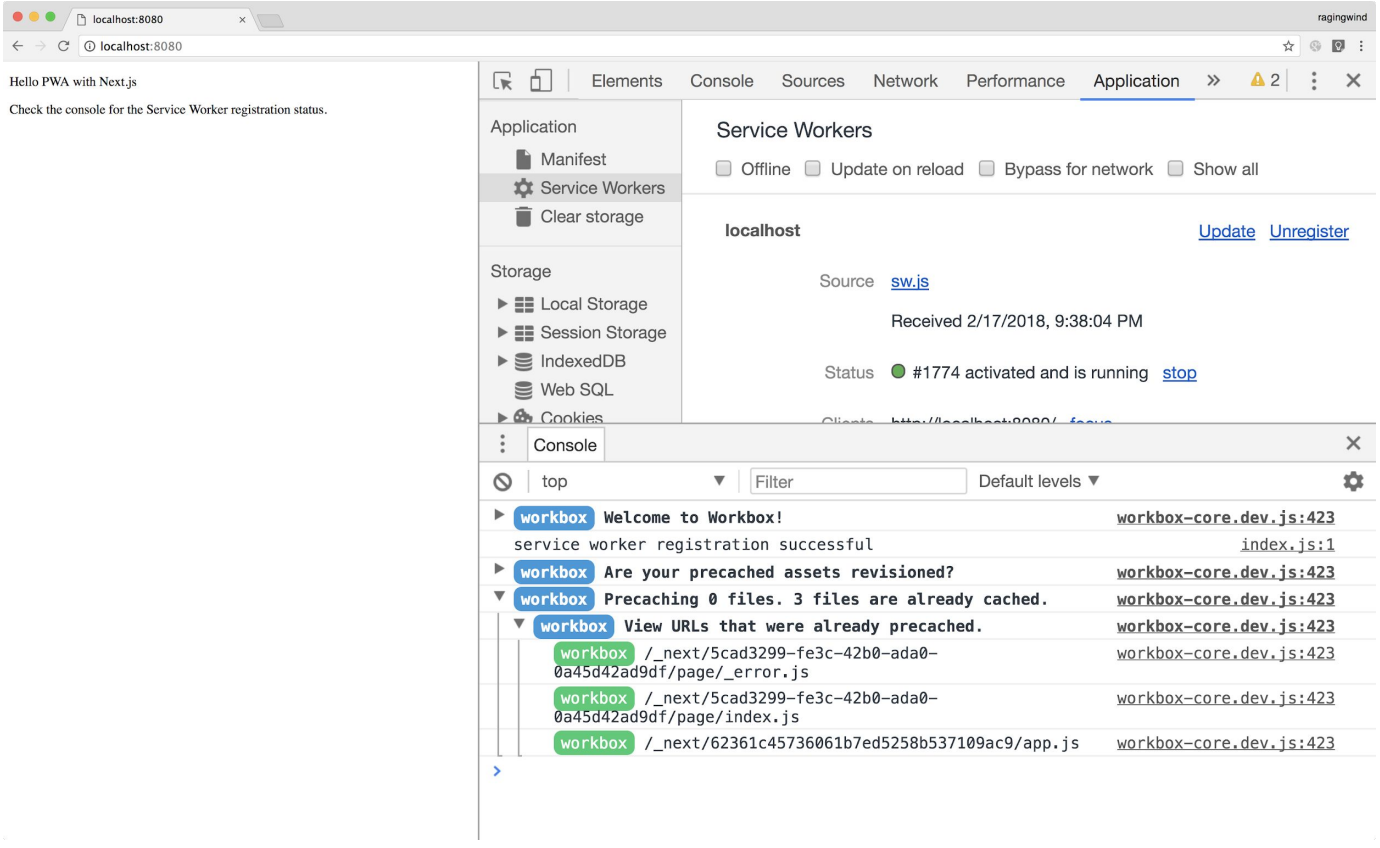
CRA. Webpack


.js
.js
.css
.js
.png
.js
.js
.png
.js
.css
.js
webpack
MODULE BUNDLER
.js.min
.css.min
.png

CRA. Webpack Dev Server



CRA. WorkBox


localhost:8080
ragingwind
localhost:8080
Hello PWA with Next.js
Check the console for the Service Worker registration status.
Elements
Console
Sources
Network
Performance
Application
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
Service Workers
Offline
Update on reload
Bypass for network
Show all
localhost
Update
Unregister
Source
sw.js
Received 2/17/2018, 9:38:04 PM
Status
#1774 activated and is running
stop
Console
top
Filter
Default levels
workbox Welcome to Workbox!
workbox-core.dev.js:423
service worker registration successful
index.js:1
workbox Are your precached assets revisioned?
workbox-core.dev.js:423
workbox Precaching 0 files. 3 files are already cached.
workbox-core.dev.js:423
workbox View URLs that were already precached.
workbox-core.dev.js:423
workbox /_next/5cad3299-fe3c-42b0-ada0-0a45d42ad9df/page/_error.js
workbox-core.dev.js:423
workbox /_next/5cad3299-fe3c-42b0-ada0-0a45d42ad9df/page/index.js
workbox-core.dev.js:423
workbox /_next/62361c45736061b7ed5258b537109ac9/app.js
workbox-core.dev.js:423

Разработка

CRA. Разработка

```
npx create-react-app my-app

npm start
npm test
npm run build
npm run eject

// package.json
+ analyze": "source-map-explorer 'build/static/js/*.js'"
```

Поддержка всех современных браузеров, IE9+ с polyfill.

Синтаксис:

- Exponentiation Operator (ES2016).
- Async/await (ES2017).
- Object Rest/Spread Properties (ES2018).
- Dynamic import() (stage 3 proposal)
- Class Fields and Static Properties (part of stage 3 proposal).
- JSX, Flow and TypeScript.

```
my-app/
  README.md
  node_modules/
  package.json
  public/
    index.html
    favicon.ico
  src/
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
```

CRA. Стили. CSS

Button.css

```
.Button {
  padding: 20px;
}
```

Button.js

```
import React, { Component } from 'react';
import './Button.css'; // Tell Webpack that Button.js uses these styles

class Button extends Component {
  render() {
    // You can use them as regular CSS styles
    return <div className="Button" />;
  }
}
```

CRA. Стили. CSS Modules

`Button.module.css`

```
.error {
  background-color: red;
}
```

`another-stylesheet.css`

```
.error {
  color: red;
}
```

`Button.js`

```
import React, { Component } from 'react';
import styles from './Button.module.css'; // Import css modules stylesheet as styles
import './another-stylesheet.css'; // Import regular stylesheet

class Button extends Component {
  render() {
    // reference as a js object
    return <button className={styles.error}>Error Button</button>;
  }
}
```

CRA. Стили. SCSS

```
$ npm install node-sass --save
```

src/App.css -> src/App.scss

```
1.   // src/App.scss
2.   @import 'styles/_colors.scss'; // Импортирование общих стилей
3.   @import '~nprogress/nprogress'; // Импортирование стилей из внешних пакетов
4.
5.   // Можно и комбинировать с CSS Modules!
6.
```

CRA. Стили. Post CSS. Autoprefixer

```
.App {
  display: flex;
  flex-direction: row;
  align-items: center;
}
```

```
.App {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-orient: horizontal;
  -webkit-box-direction: normal;
  -ms-flex-direction: row;
  flex-direction: row;
  -webkit-box-align: center;
  -ms-flex-align: center;
  align-items: center;
}
```

CRA. Стили. Внешние UI kits

```
$ npm install --save bootstrap
```

```
1.   // App.js
2.   import 'bootstrap/dist/css/bootstrap.css';
3.
4.   // style.scss
5.   // Переопределение переменных из внешних UI kits
6.   $body-bg: #000;
7.
8.   // Импорт стилей в самом scss
9.   @import '~bootstrap/scss/bootstrap.scss';
10.
```

https://material-ui.com/getting-started/installation/
https://hackernoon.com/23-best-react-ui-component-libraries-and-frameworks-250a81b2ac42

CRA. Разработка. Дополнительные параметры CRA

```
// CRA + flow
$ npm install --save flow-bin

// CRA + TS
$ npx create-react-app my-app --typescript

// CRA + Relay
$ npm install --save babel-plugin-relay@dev
```

CRA. Разработка. Proxy в бекенд

```
1.   // package.json
2.   ...
3.   "proxy": "http://localhost:4000",
4.   …
```

```
fetch('/api/todos') -> http://localhost:4000/api/todos
```

CRA. Разработка в изоляции. Storybook

Обычно, в приложении имеется большое количество ui компонентов, каждый из них имеет множество состояний. Например, простой компонент кнопки может иметь следующие свойства:

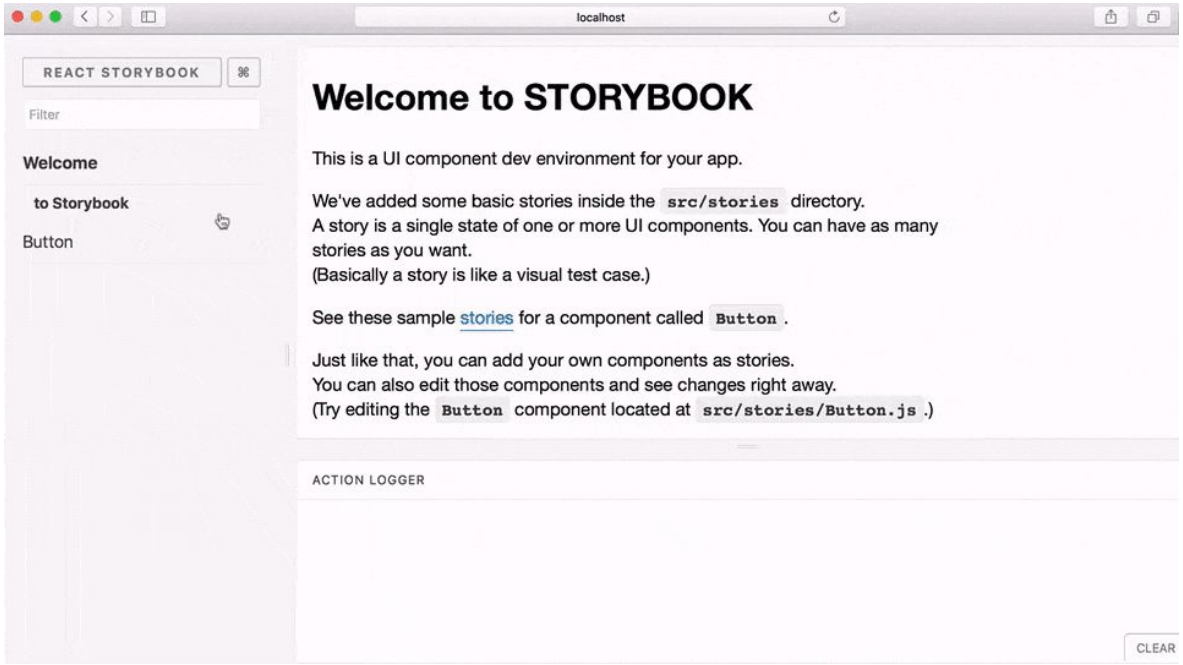
- Обычное состояние, с текстом
- Выключенное состояние
- Состояние загрузки (preloader)

Чаще всего бывает сложно увидеть эти состояния без запуска самого приложения или тестовых примеров на его основе.

https://learnstorybook.com/
https://storybook.js.org/basics/introduction/

CRA. Разработка в изоляции. Storybook

```
$ npx -p @storybook/cli sb init
```



CRA?

Вопросы?

“Перерыв! (10 минут)

Препод (с)

Тестирование

Тестирование. Обеспечение качества (Quality Assurance)

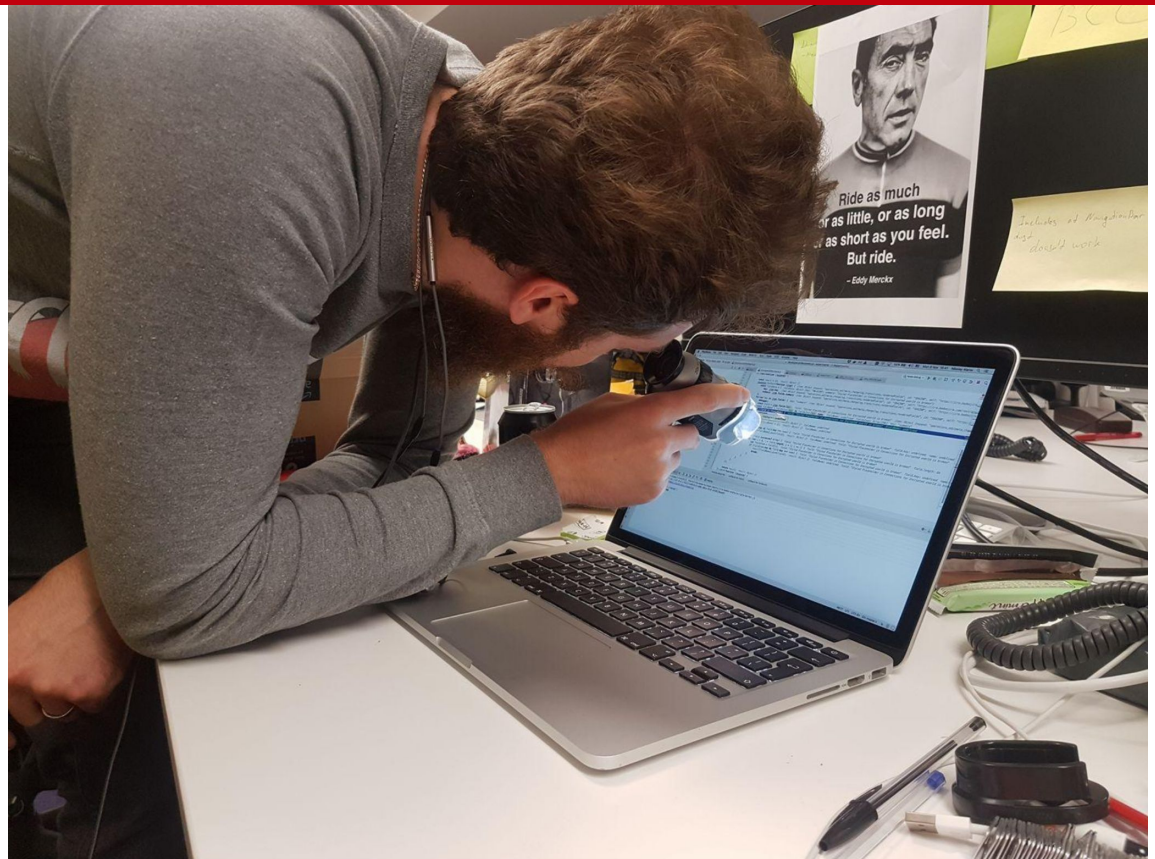
QA - совокупность мероприятий, охватывающих все этапы разработки, выпуска и эксплуатации и предпринимаемых на разных стадиях жизненного цикла ПО для обеспечения качества выпускаемого продукта.

Тестирование. Обеспечение качества (Quality Assurance)

Этапы обеспечения качества ПО. **Внутренний**:

- Тестирование документации
- Unit-тесты
- Code-review

Тестирование. Code-review



Тестирование. Обеспечение качества (Quality Assurance)

Этапы обеспечения качества ПО. **Внешний**:

- Обработка обратной связи
- Тестирование
- Менеджерская приемка

Тестирование. QA. Обработка обратной связи

- Взаимодействие в соц.сетях/магазинах и пр.
- Обработка заявок в саппорт
- Фокус тестирование
- $^{\alpha}/_{\beta}$ тестирование
- A/B тестирование

Тестирование. Контроль качества (Quality Control)

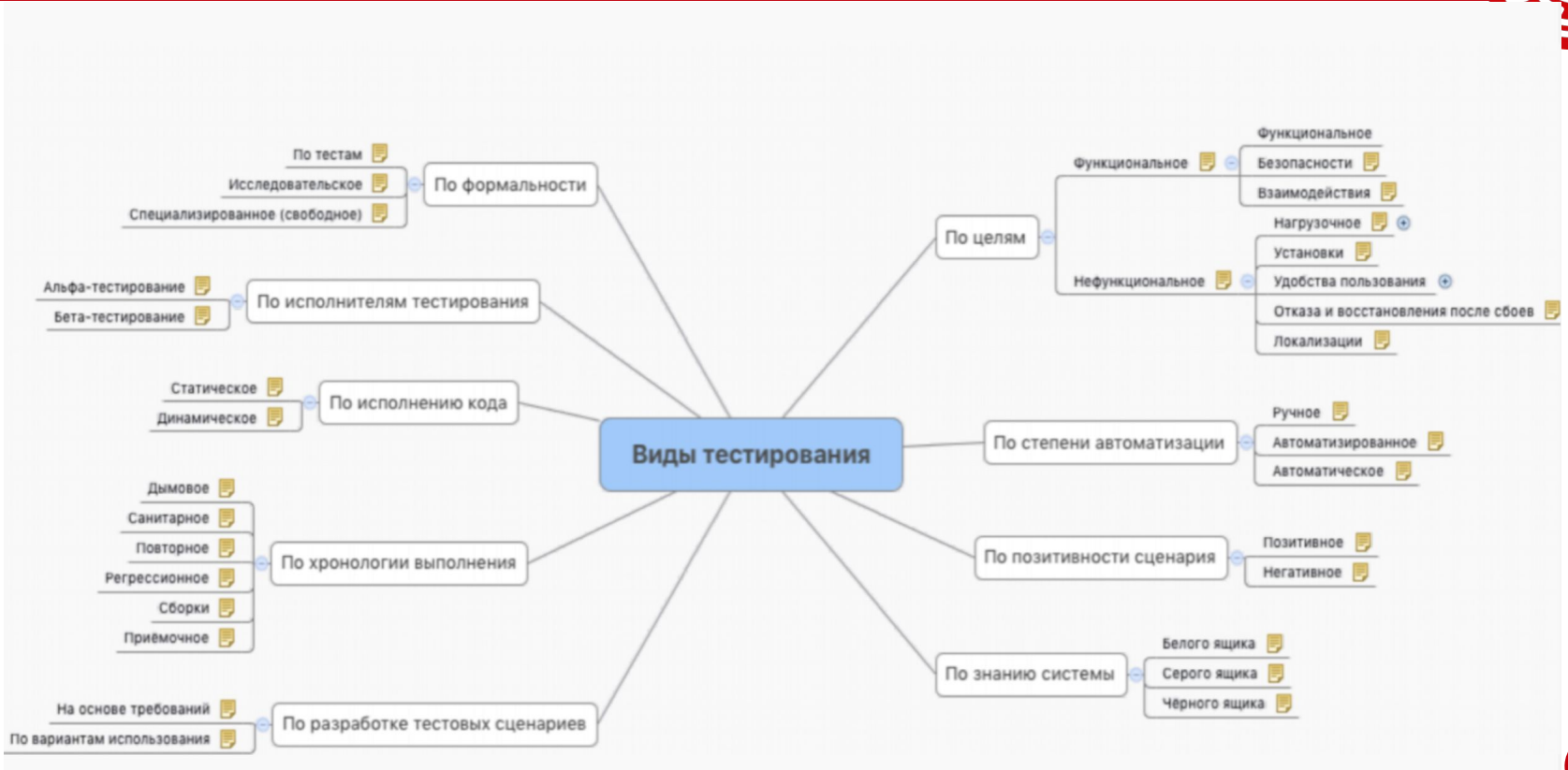
Quality Control - совокупность мероприятий, проводимых над объектом в процессе разработки для постоянного получения информации об актуальном состоянии объекта и его соответствии требованиям и заявленному уровню качества.

Тестирование - процесс исследования ПО с целью получения актуальной информации о качестве продукта.

Цели:

- Проверка на соответствие требованиям
- Повышение качества ПО
- Предотвращение дефектов
- Выявление дефектов

Тестирование. Виды тестирования



Тестирование. Виды тестирования по целям

Функциональные:

- Функциональное/End-to-end тестирование
- Тестирование безопасности
- Интеграционное тестирование
- Unit-тесты
- Smoke-тесты
- Тестирование установки

Тестирование. Виды тестирования по целям

Нефункциональные:

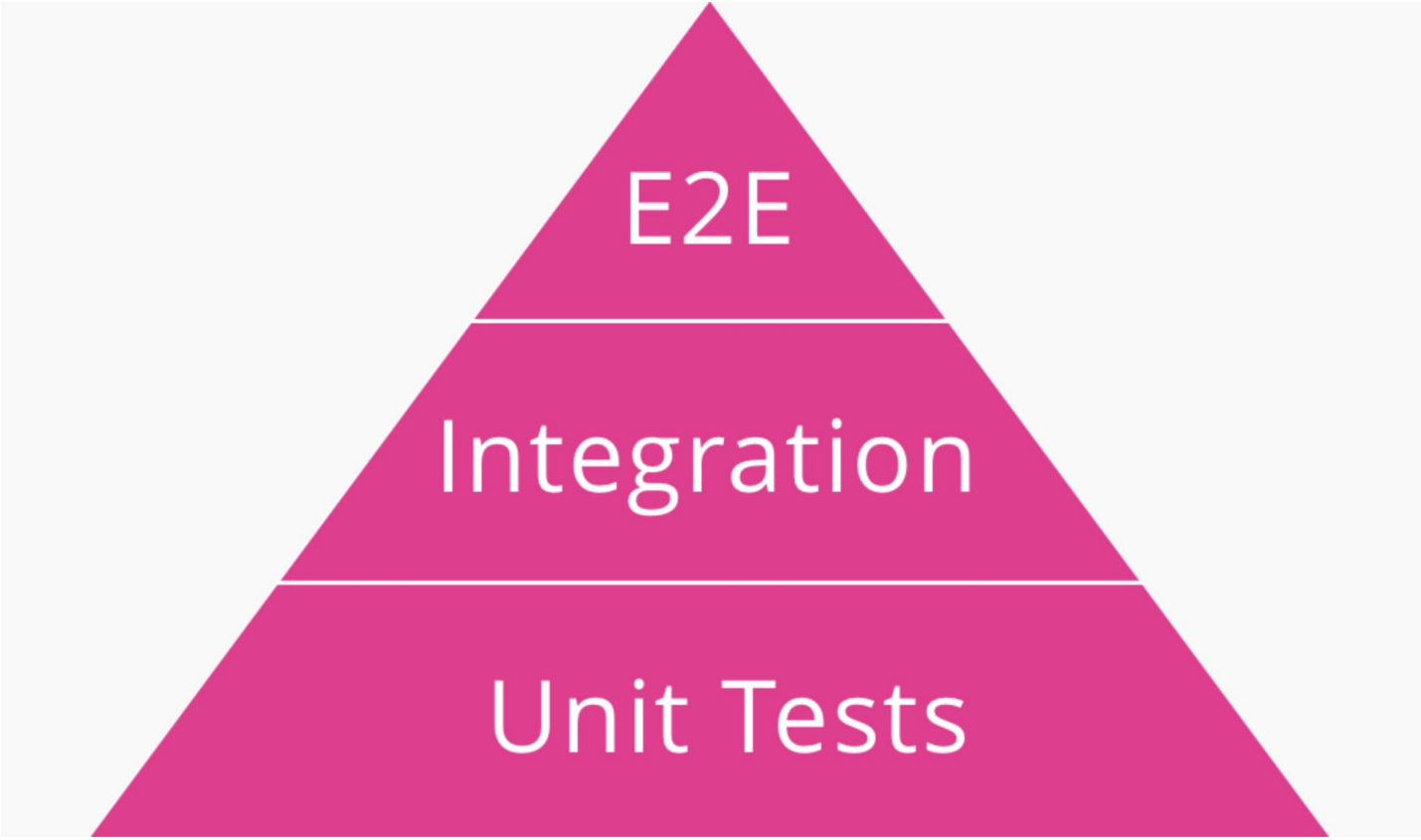
- Тестирование отказа и восстановления после сбоев
- Нагрузочное тестирование
- Тестирование удобства использования
- Тестирование локализации

Тестирование. Виды тестирования по целям

По хронологии:

- Дымовое
- Тестирование сборки
- Регрессионное
- Приемочное тестирование

Тестирование. Основные типы тестирования



Тестирование. Unit

Unit–тесты - тесты, позволяющие проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.

Тестирование. Unit. Зачем?

- Качество кода, быстрый фидбек
- Если есть тесты, то вносить изменения легче => уверенный разработчик
- Повышение скорости разработки (производительность)
- Минимизация человеческого фактора
- Документация

Тестирование. Unit

Unit-тесты не проверяют:

- БД
- Другие либы и пакеты

Почему их надо писать?

- Проверяют логику и всё, где есть условия
- Быстро и легко написать
- Быстро проходят

Тестирование. Пример Unit-теста

```
describe('Basic Mocha String Test', function () {
 it('should return number of charachters in a string', function () {
        assert.equal("Hello".length, 4);
    });

 it('should return first charachter of the string', function () {
        assert.equal("Hello".charAt(0), 'H');
    });
});
```

Тестирование. Принципы написания автотестов

- Атомарность
- Независимость
- Изолированность / герметичность

Тестирование. Принципы написания автотестов. Атомарность

```
describe('isUndefined function', ()=> {
        it('should return true or false when object is undefined', () => {
           expect(isUndefined(undefined)).toEqual(true);
           expect(isUndefined(true)).toEqual(false);
        });
});
```

```
describe('isUndefined function', ()=> {
        it('should return true when object is undefined', () => {
           expect(isUndefined(undefined)).toEqual(true);
        });

        it('should return false when object has a boolean value', () => {
           expect(isUndefined(true)).toEqual(false);
        });
});
```

Тестирование. Принципы написания автотестов. Независимость

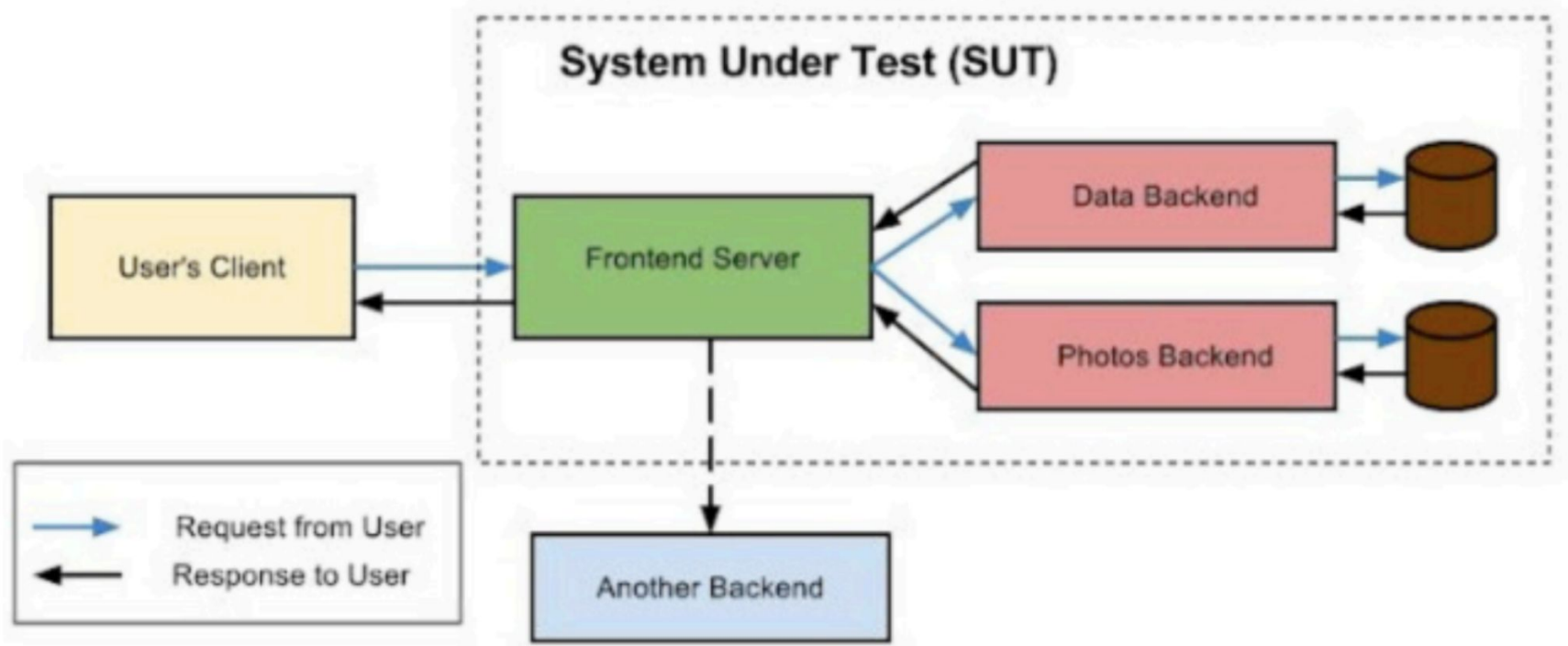
- Test 1
- Test 2
- Test 3

- Test 3
- Test 2
- Test 1

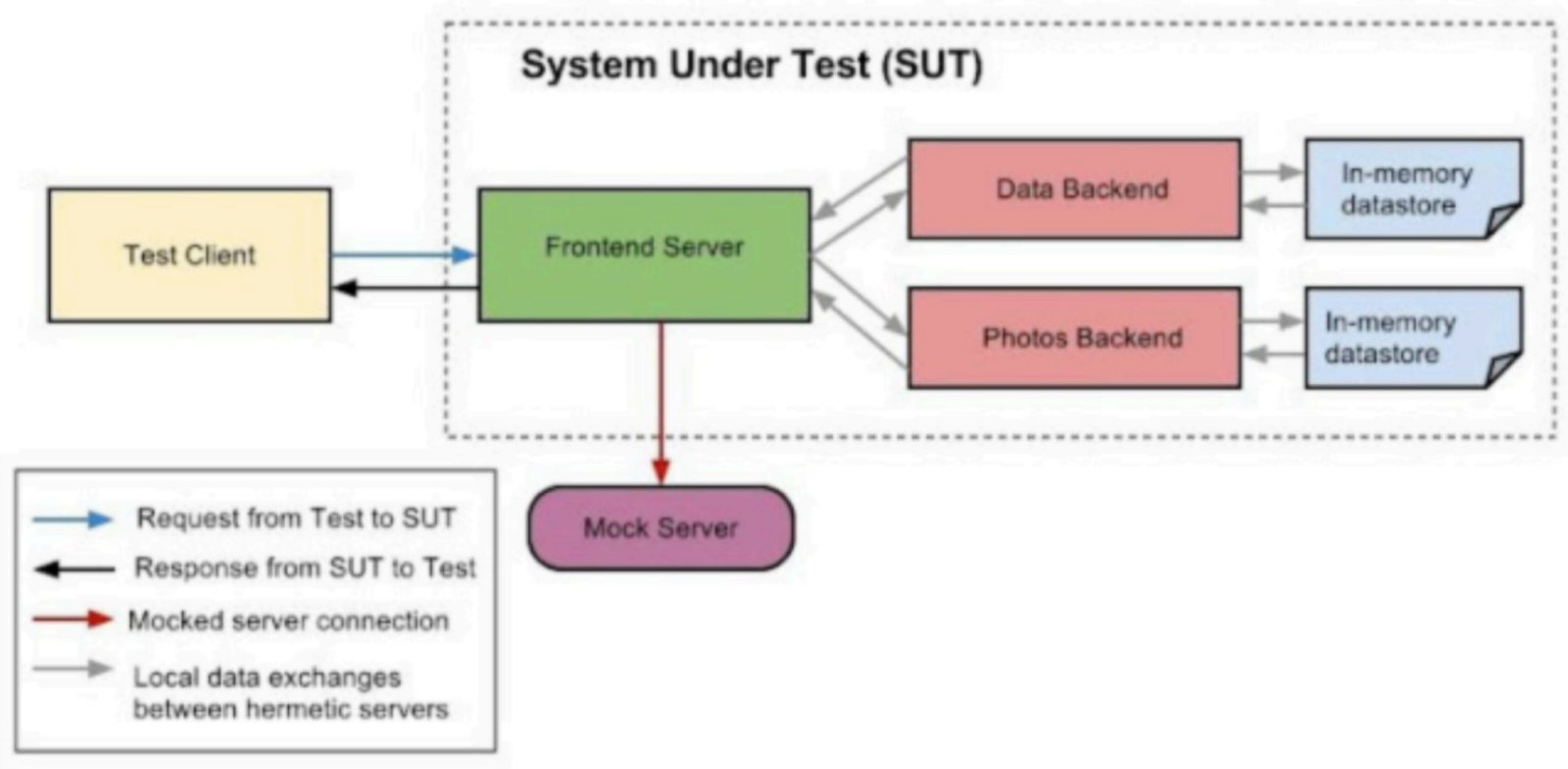
- Test 2
- Test 3
- Test 1

...

Тестирование. Принципы написания автотестов. Изолированность



Тестирование. Принципы написания автотестов. Изолированность



Тестирование. Что еще за Mock server? Заглушки

- Fake – имитация, “упрощенная” рабочая реализация
- Mock – эмуляция объекта, не рабочая реализация
- Stub – заглушка для объекта
- Spy – это Stub с записью данных или вызовов, поступающих из тестируемого объекта
- Dummy - пустой объект

Тестирование. Фреймворки для Unit

Тестирование. Jest

- Скорость
- Все уже есть (assertions, mocks, spies)
- Можно использовать другие либы
- Snapshot testing
- Code coverage
- Интерфейс понятен и удобен

Тестирование. Jest. Пример Unit-теста

```
// sum.js
function sum(a, b) {
 return a + b;
}
module.exports = sum;

// sum.test.js
const sum = require('./sum');
test('adds 1 + 2 to equal 3', () => {
 expect(sum(1, 2)).toBe(3);
});
```

https://jestjs.io/docs/ru/api

Тестирование. Jest. Пример snapshot-теста

```
import React from 'react';
import Link from '../Link.react';
import renderer from 'react-test-renderer';

it('renders correctly', () => {
  const tree = renderer
    .create(<Link page="http://www.facebook.com">Facebook</Link>)
    .toJSON();
  expect(tree).toMatchSnapshot();
});

// snapshot
exports[`renders correctly 1`] = `
<a
  className="normal"
  href="http://www.facebook.com"
  onMouseEnter={[Function]}
  onMouseLeave={[Function]}
>
  Facebook
</a>
`;
```

Тестирование. Integration

Integration–тесты - вид тестирования, при котором на соответствие требований проверяется интеграция модулей, их взаимодействие между собой, а также интеграция подсистем в одну общую систему.

Для интеграционного тестирования используются компоненты, уже проверенные с помощью модульного тестирования, которые группируются в множества. Данные множества проверяются в соответствии с планом тестирования, составленным для них, а объединяются они через свои интерфейсы.

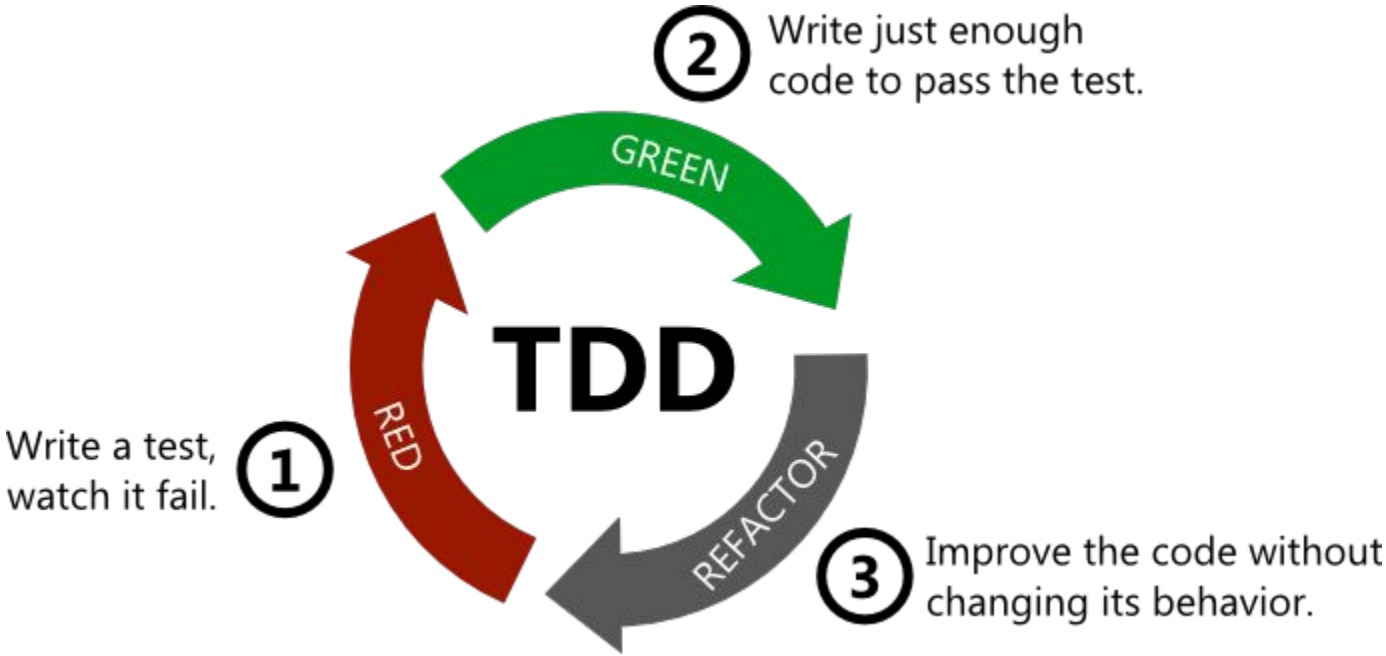
Тестирование. E2E

E2E–тесты - вид тестирования, где проверяется работоспособность не отдельных юнитов, а всей системы сразу. Это гарантирует точное взаимодействие приложения и обеспечивает правильное функционирование на разных платформах и средах.

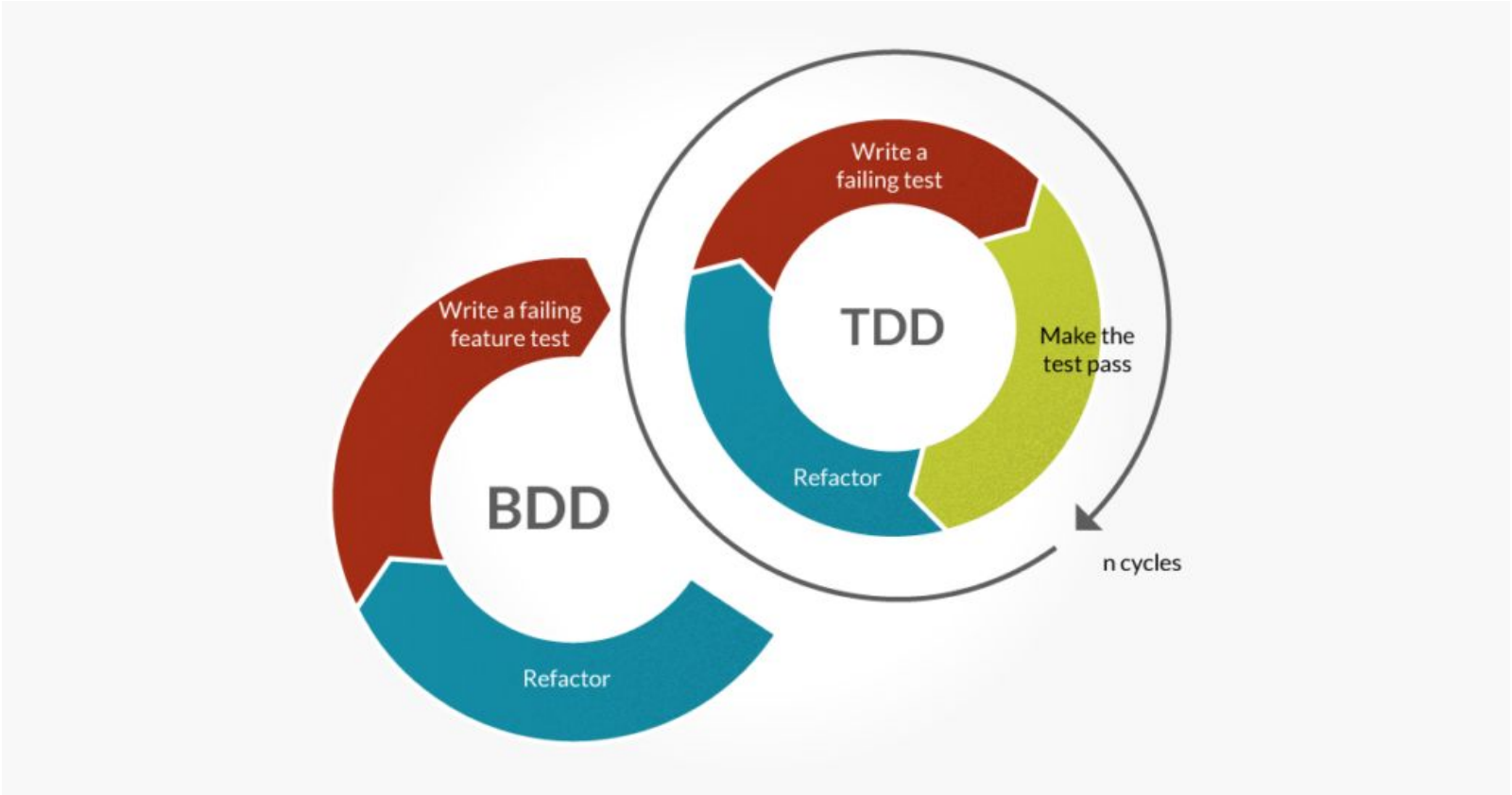
Тестирование. E2E. Пример Puppeteer-теста

```
$ npm i jest-puppeteer --save

describe('Google', () => {
  beforeAll(async () => {
    await page.goto('https://google.com');
  });

  it('should be titled "Google"', async () => {
    await expect(page.title()).resolves.toMatch('Google');
  });
});
```

Тестирование. TDD



Тестирование. BDD



Тестирование. Заключение

- Автотесты пишутся людьми и не лишены ошибок
- Автотесты чувствительны к среде исполнения
- Тестирование нестабильного приложения вызывает частые падения
- Разбор логов может потребовать больше времени, чем ручной прогон тестов

Домашнее задание № 7

1. Покрыть что-нибудь тестами (уточнение будет в репозитории)
2. ???
3. Profit!

Срок сдачи

FS-21: ? апреля

Спасибо за внимание!