对Web前端项目测试情况的实证研究初选报告

陈俊达 刘宇航

2020年11月10日

研究背景

- Web前端的重要性和复杂度迅速增加
- 对测试的关注提高
- 前端测试的复杂性
- TypeScript的出现

研究问题

- RQ1：测试用例的数量和整体代码量之间有什么关系？
- RQ2：与DOM交互的测试用例在所有测试用例中占比多少？
- RQ3：测试用例数量与bug数量的关系是怎么样的？
- RQ4：使用TypeScript是否能降低bug的数量？

研究问题解释

- RQ1：测试用例的数量和整体代码量之间有什么关系？
  - 一般来说，代码量越多，测试用例应该越多
  - 探究在前端领域中二者的具体关系
  - 可作为对项目测试充分性的初步判断标准
- 统计方法
  - 前端领域主要采用jest等测试框架
  - 使用框架获取测试用例的数量
  - 整体代码量可以用代码统计工具获得代码行数

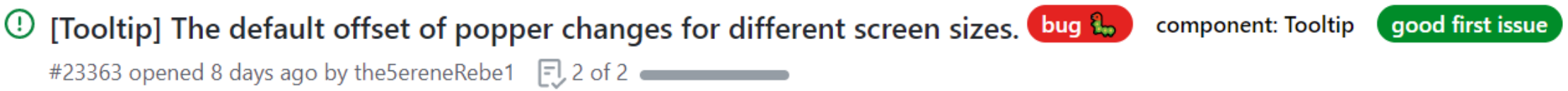
研究问题解释

- RQ2：与DOM交互的测试用例在所有测试用例中占比多少？
  - 与DOM交互是前端代码功能的重要组成部分
  - 测试与DOM交互的代码较为复杂
- 统计方法
  - 不同框架使用不同的库测试与DOM交互
    - React：Enzyme
    - Vue：vue-test-utils
  - 若测试用例里使用了这些库的方法，则认为与DOM交互

```
it("should update when store updated", () => {
  const store = createStore(testStore, 42);

  const Component = () => {
    const { value, setValue } = useStore(testStore);
    return (
      <div>
        <span>{value}</span>
        <button onClick={() => setValue(value + 1)}>Increment</button>
      </div>
    );
  }

  const wrapper = mount(
    <StoreProvider stores={[store]}>
      <Component />
    </StoreProvider>
  )

  const span = wrapper.find("span");
  const btn = wrapper.find("button");

  expect(span.text()).toBe("42");
  btn.simulate("click");
  expect(span.text()).toBe("43");

});
```

研究问题解释

- RQ3：测试用例数量与bug数量的关系是怎么样的？
  - 假设：测试越多，bug应该越少
- 统计方法
  - 统计有bug标记的issue
    - 若没有，则考虑所有issue
  - 考虑已经closed的bug



Material-ui的bug类型的issue示例

研究问题解释

```
function add(a, b) {
  return a + b;
}

function tsAdd(a: number, b: number): number {
  return a + b;
}
```

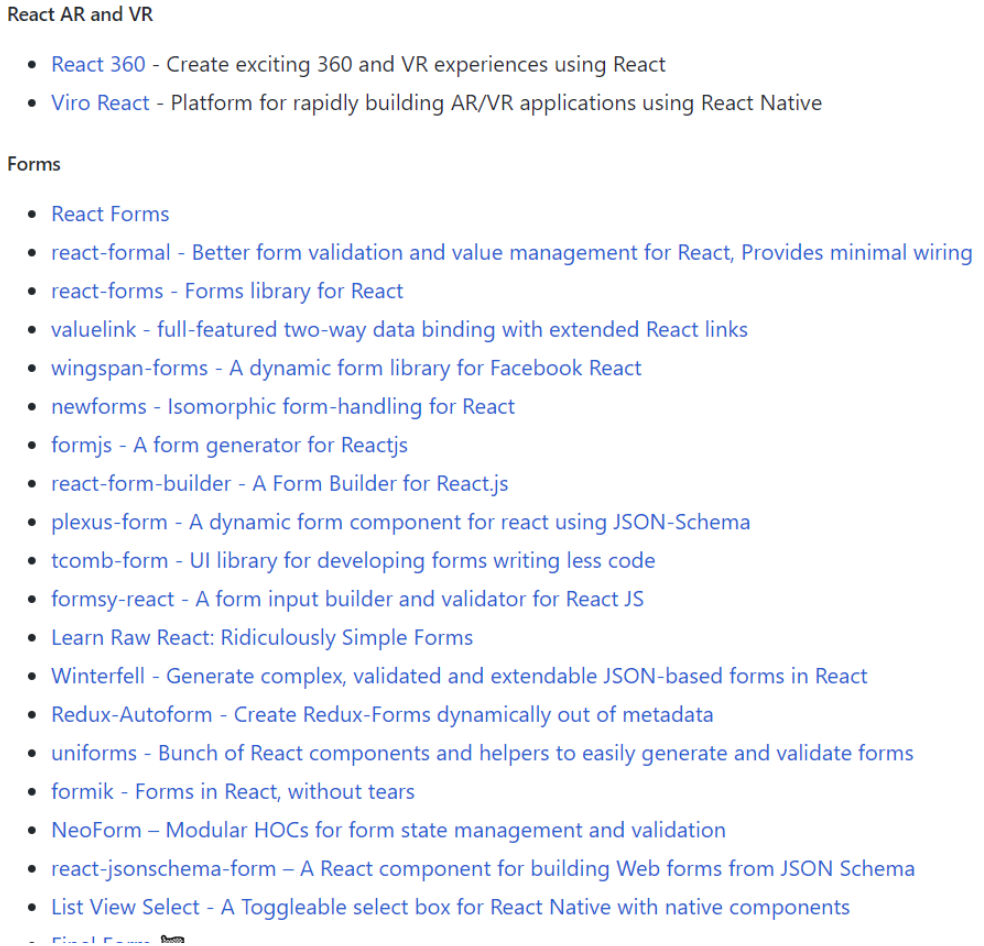
- RQ4：使用TypeScript是否能降低bug的数量？
  - TypeScript是JavaScript的带有类型系统的超集
  - 能在编译期发现很多JS只能在运行时发生的错误
- 统计方法
  - **TS的代码量占总体代码量的比例**和**bug数量/总代码量**之间的关系
  - 案例分析：Ant Design和Material UI
    - React生态中的完整的组件库
    - Ant design使用TypeScript编写，Material UI使用JavaScript编写
    - 分析bug数量和代码量之间的动态关系（从项目开始到现在）

研究方法

- 主要采用**调查**的方法
- 对一些典型应用采用**案例分析（case study）**的方法
- 从GitHub上拉Star最多的200个前端领域的代码
  - 去掉没有代码的仓库（比如面试经验）
  - 或者awesome-xxx仓库中找一些项目

Threat to validity

- 不同类型的前端对测试的需求不同，影响结果可参考性
  - 基础设施库（如React）需求稳定且重要，需要更多测试
  - 上层应用（如网站）需求不稳定，对完整的测试需求不高
- 数据量较小
- 统计量的代表性
- 选取的仓库的代表性
  - Star高的仓库使用的用户多，bug更多
  - 可以考虑去awesome-xxx仓库里找项目



React AR and VR

- React 360 - Create exciting 360 and VR experiences using React
- Viro React - Platform for rapidly building AR/VR applications using React Native

Forms

- React Forms
- react-formal - Better form validation and value management for React, Provides minimal wiring
- react-forms - Forms library for React
- valuelink - full-featured two-way data binding with extended React links
- wingspan-forms - A dynamic form library for Facebook React
- newforms - Isomorphic form-handling for React
- formjs - A form generator for Reactjs
- react-form-builder - A Form Builder for React.js
- plexus-form - A dynamic form component for react using JSON-Schema
- tcomb-form - UI library for developing forms writing less code
- formsy-react - A form input builder and validator for React JS
- Learn Raw React: Ridiculously Simple Forms
- Winterfell - Generate complex, validated and extendable JSON-based forms in React
- Redux-Autoform - Create Redux-Forms dynamically out of metadata
- uniforms - Bunch of React components and helpers to easily generate and validate forms
- formik - Forms in React, without tears
- NeoForm – Modular HOCs for form state management and validation
- react-jsonschema-form – A React component for building Web forms from JSON Schema
- List View Select - A Toggleable select box for React Native with native components

Awesome-react项目中的部分React生态圈的项目

谢谢！