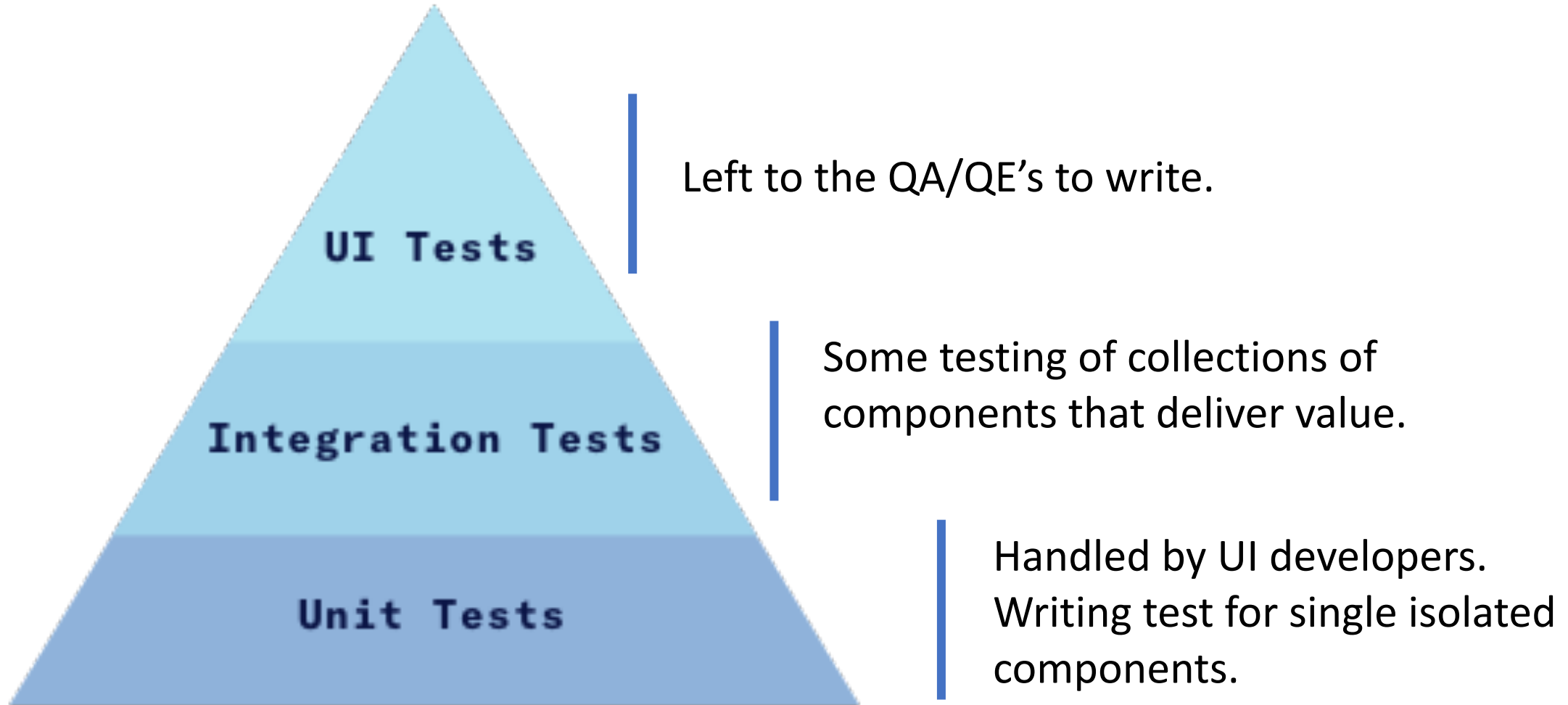


Testing code from a React JS perspective



Unit versus Integrations tests

Unit testing of
components

Cancel

Save

Postcode

EC1Y 4XX

Integration testing of
components

Address line 3 (Optional)

48 Chiswell St

Town/City

London

Country

United Kingdom

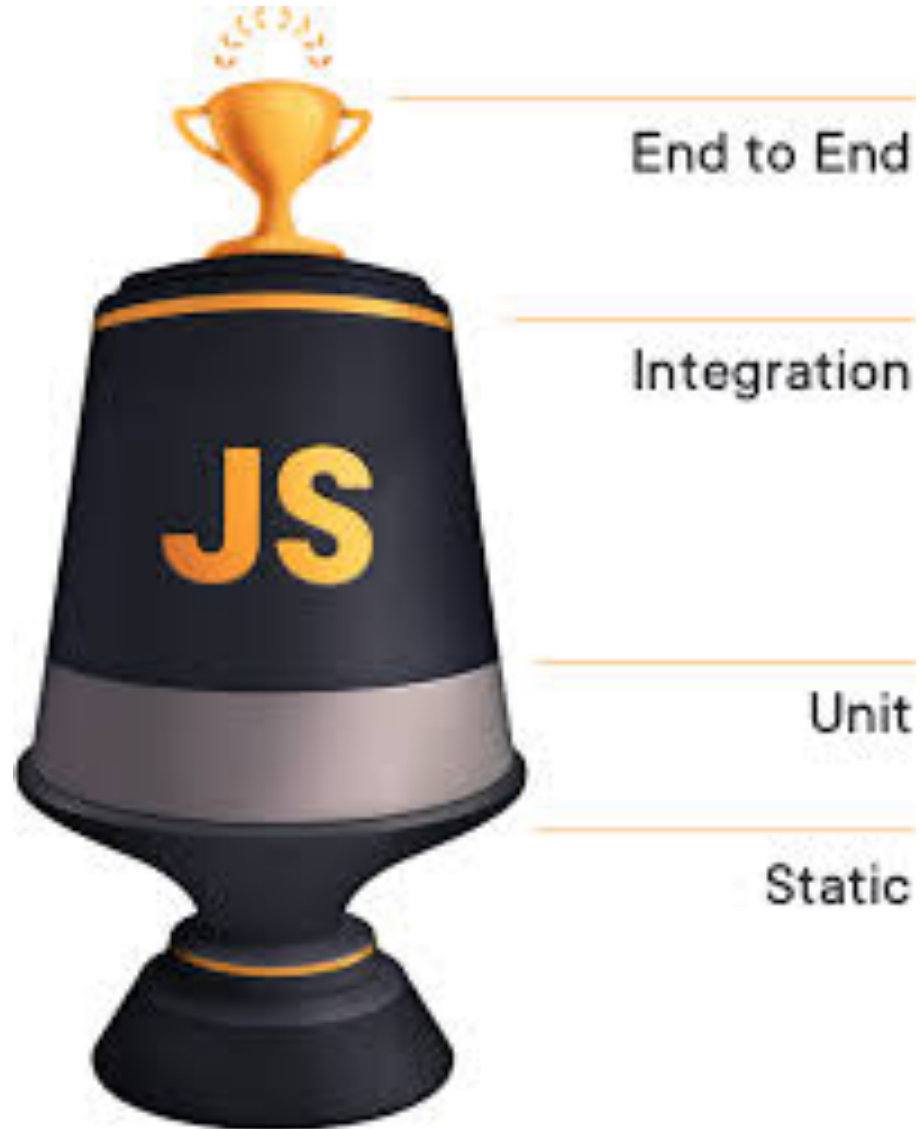
Postcode

EC1Y 4XX

Cancel

Save

The testing trophy



Static typing:
Typescript

React-testing-library



[Docs](#)

[Tutorial](#)

[Blog](#)

[Community](#)

[Search](#)

🔗 Recommended Tools

Jest is a JavaScript test runner that lets you access the DOM via [jsdom](#). While jsdom is only an approximation of how the browser works, it is often good enough for testing React components. Jest provides a great iteration speed combined with powerful features like [mocking modules](#) and [timers](#) so you can have more control over how the code executes.

React Testing Library is a set of helpers that let you test React components without relying on their implementation details. This approach makes refactoring a breeze and also nudges you towards best practices for accessibility. Although it doesn't provide a way to "shallowly" render a component without its children, a test runner like Jest lets you do this by [mocking](#).

React testing library advantages.

- Allows developers to write tests for the pages the users will be interacting with quickly and intuitively.
- Developers think more about users rather than implementation details
- Increased code coverage with less lines of code.
- Encourages accessibility.
- Allows easy refactoring. It makes software 'soft' again.