Useful Tests for React Applications

Puxuan He

Willie Barth

Brian Ambielli

When should UI testing begin?

When requirements have solidified (UX or Business Logic)

Otherwise, **you will spend as much time maintaining broken tests as you will spend writing features**

For us: **after M0 milestone** was the right time - UX was fluid until then

Our Team's Unit Testing Strategy

Golden Rule: Write tests that are useful

1) Tests that catch bugs before release are useful
2) Tests that point developers to the source of a bug are useful
3) Tests that are easy to maintain and write are useful

Tools We Use for Unit Testing React

Jest

Test runner + Assertions + mocks

Enzyme

Rendering React component tree during tests

Jest Snapshots

Serialize component tree to JSON string.

In Addition: enzyme-to-json && nock

What we have tried

1) Unit tests written with pure enzyme (rigid)

2) Unit tests written with enzyme, **enzyme-to-json** serializer, and snapshots (flexible)

3) Integration tests written using **enzyme-to-json** serializer and snapshots (most comprehensive)

Unit Tests with Pure Enzyme

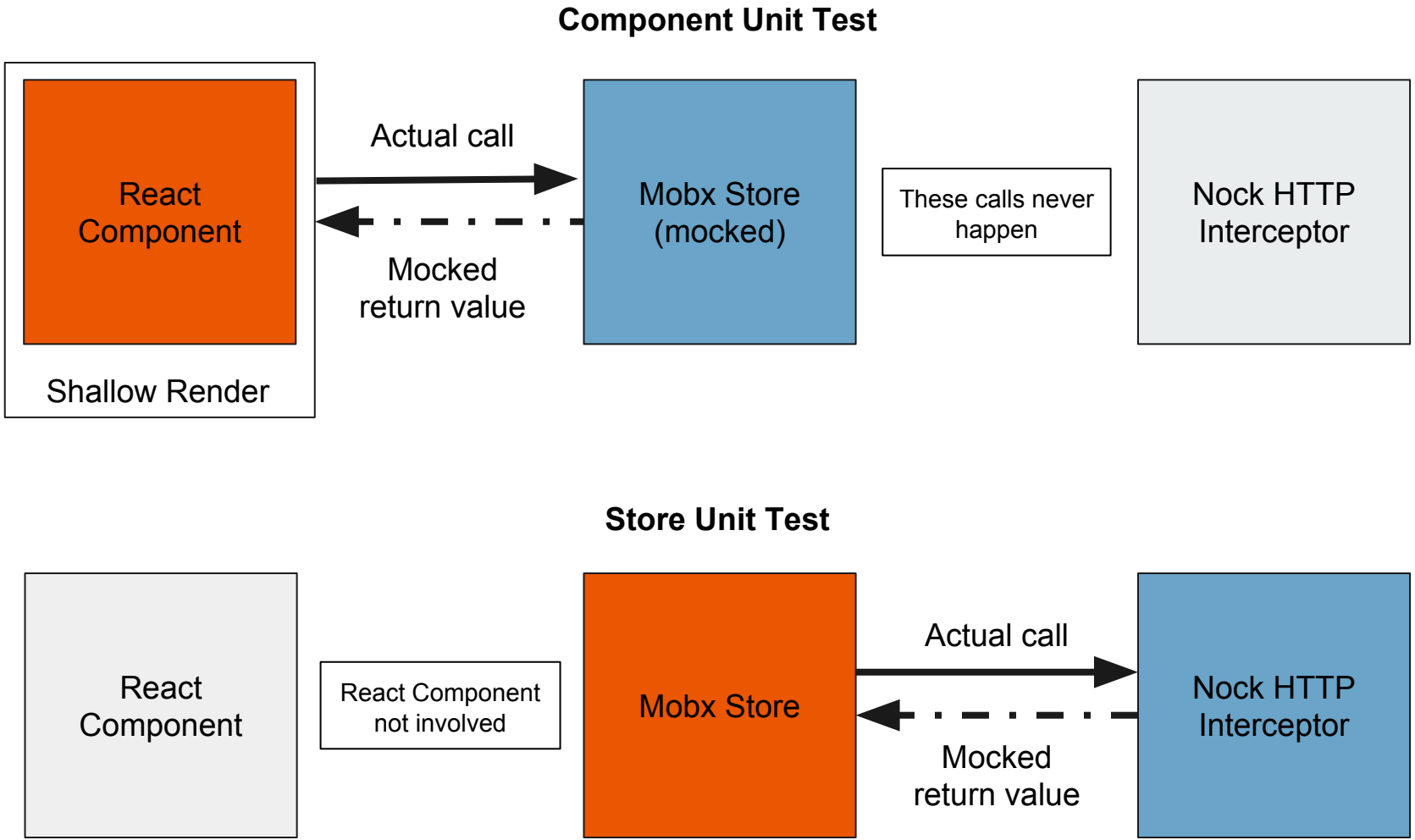
Unit Tests With Pure Enzyme

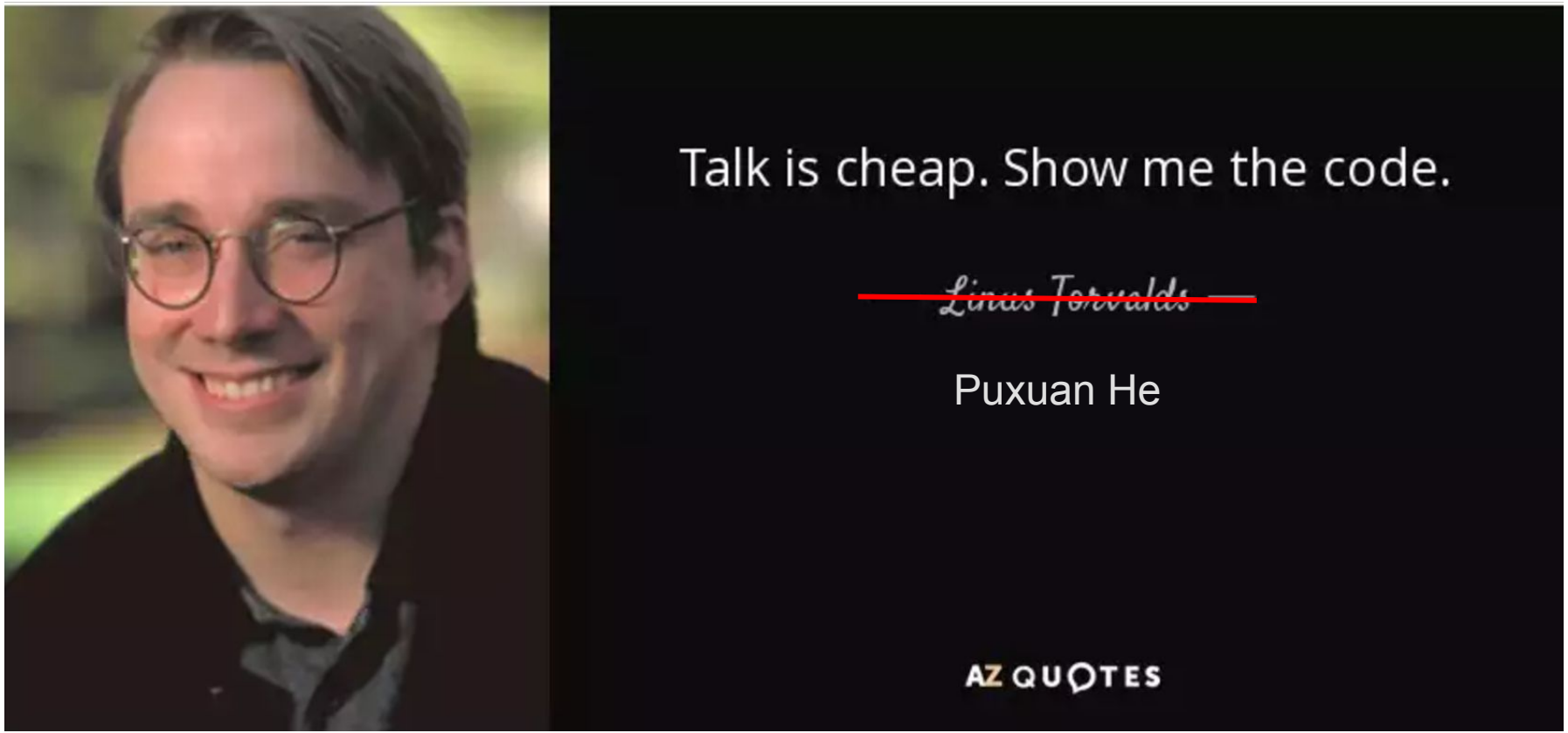
1) Feels like you are writing traditional unit tests

2) Black box event listeners via enzyme event simulation

3) Shallow Rendering + mocking store isolates tests to just the open component

4) Network requests are never made, and don’t need to be mocked


Component Unit Test
React Component
Shallow Render
Actual call
Mocked return value
Mobx Store (mocked)
These calls never happen
Nock HTTP Interceptor
Store Unit Test
React Component
React Component not involved
Mobx Store
Actual call
Mocked return value
Nock HTTP Interceptor


Talk is cheap. Show me the code.
Linus Torvalds
Puxuan He
AZ QUOTES

Unit Tests With Pure Enzyme - Downsides?

Tests were not useful

1) Tests broke often

2) Took a ton of time to write and maintain (order of hours)

3) Perhaps too rigorous of a spec for a UI component?

Unit Tests with Enzyme and Snapshots

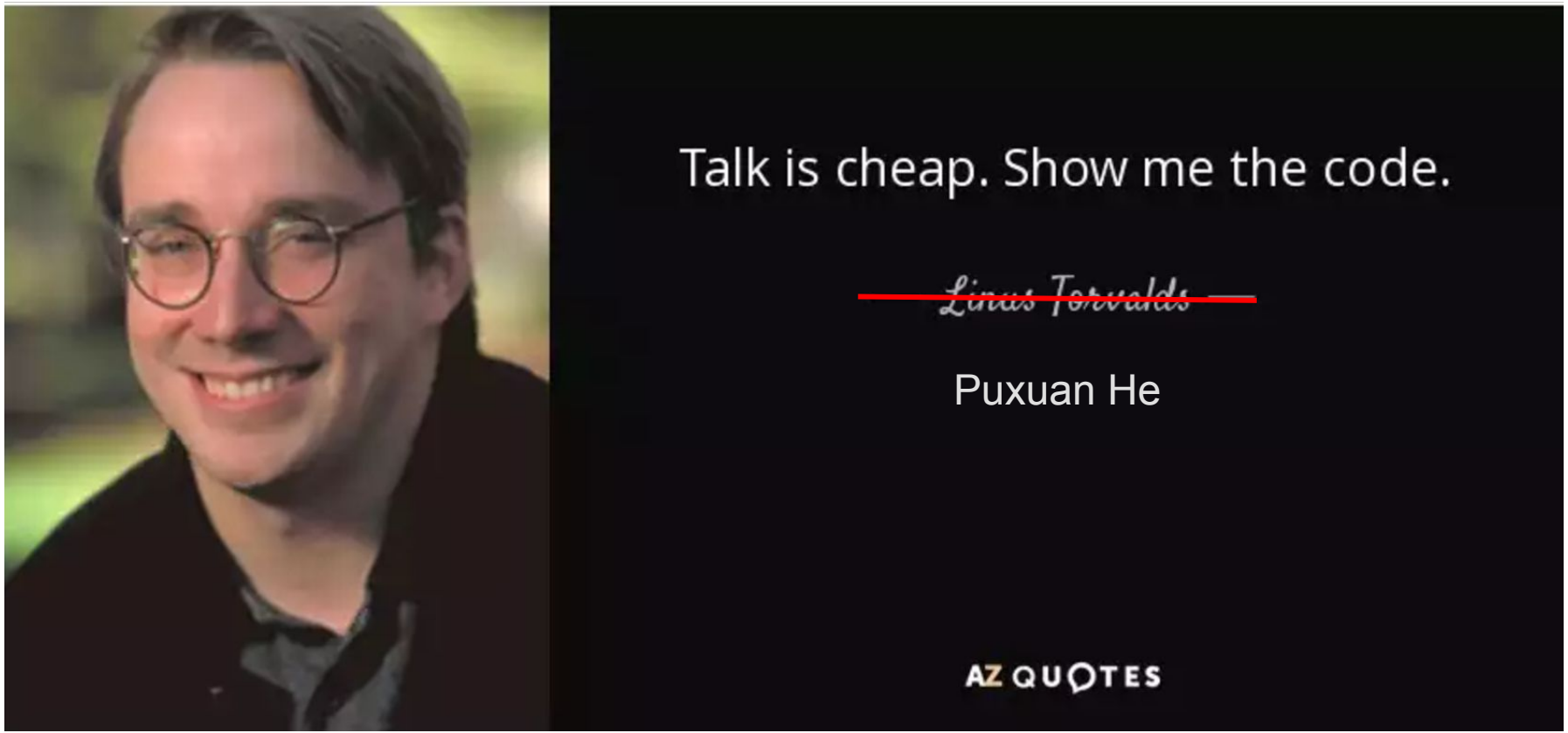
Unit Tests With Enzyme & Snapshots

Is it possible to automate the process to write and maintain test?

Yes!

What is snapshot test?

https://facebook.github.io/jest/docs/en/snapshot-testing.html


Talk is cheap. Show me the code.
Linus Torvalds
Puxuan He
AZ QUOTES

Considerations for Snapshots

TDD becomes difficult (impossible?)

Developer needs to review snapshots manually (snapshot to html package)

Snapshots can take a long time to render…

Integration Tests with Enzyme and Snapshots

Integration Tests With Enzyme & Snapshots

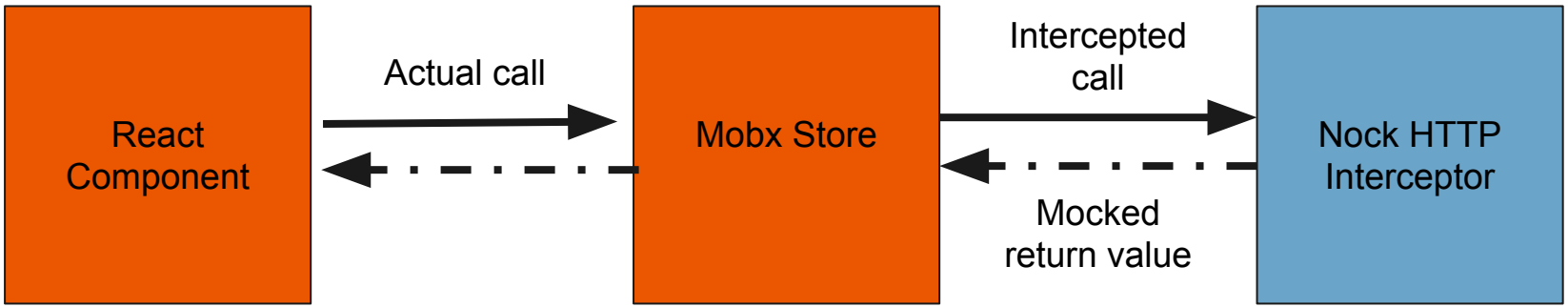
1) Feels like writing **traditional integration tests**

2) Mock all API calls

3) No Mocking of stores

4) Full DOM Rendering (enzyme mount)

Integration Tests with Enzyme & Snapshots



Approaches to Integration Tests

By far biggest challenge is taking snapshots at the right time, since enzyme wrapper updates asynchronously:

1) Wait for async elements to appear in enzyme wrapper and then take snapshot (**enzyme-wait**)
2) Take a snapshot of every mobx-initiated render (**mobx-react)**


Talk is cheap. Show me the code.
Linus Torvalds
Willie Barth
AZ QUOTES

Summary

Our testing strategy has been evolving, and will continue to evolve.

Remember the Golden Rule: Write tests that are useful

Rigid specs !== Useful for UI components

Combination of snapshot and enzyme seems to provide the most flexibility!

Integration testing UI components is hard… still learning

Questions?

Puxuan He

Willie Barth

Brian Ambielli