

Skill Training

React JS

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Schedule for React JS Training

Day1 : State Management : Class Components

Day2 : State Management : Functional Components, Hooks

Day3 : Http Client Programming

Day4 : Working with Forms, Validations, Services

Day5 : Redux

Day6 : Unit Testing

Day7 : Routing – SPA in React JS

Day6

Unit Testing React Components



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day6

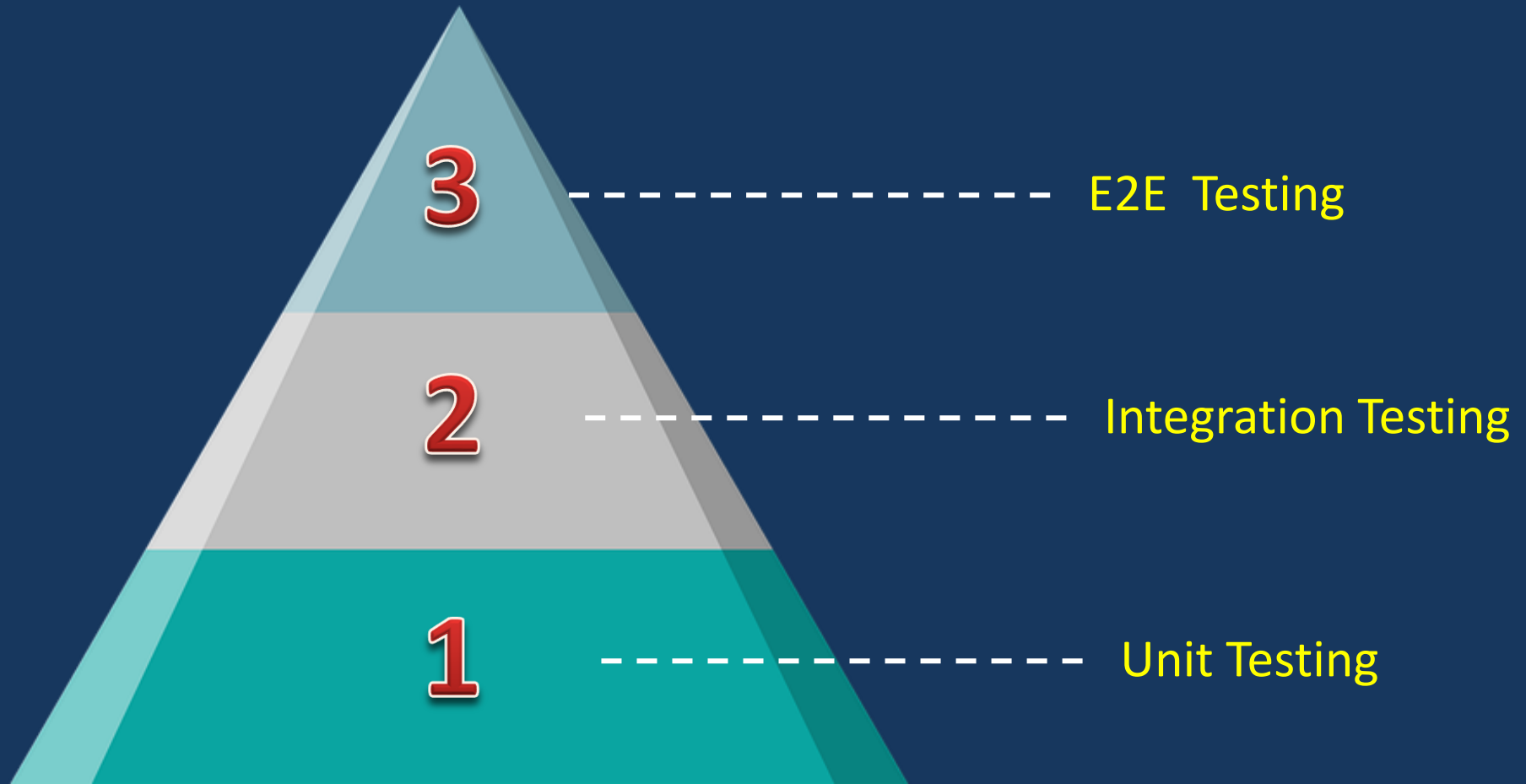
1. Introduction to Unit Testing
2. Creating Test Cases, Test Suites
3. Testing with Jest and Enzyme
4. Playwright Component Testing

Introduction to Unit Testing

What is Unit Testing?

- Unit can be the smallest part of an application that is testable.
- Eg: individual function, method, etc.
- Software developers are the ones who write the unit test cases.
- The aim here is to match the requirements and the unit's expected behavior.
- Performed before Integration testing
- Finding issues/bugs at an early stage

Levels of Testing



AAA Pattern in Unit Testing

A
Arrange

A
Action

A
Assert

AAA Pattern in Unit Testing

1. **Arrange:** This is the first step of a unit test application. Here we will arrange the test, in other words we will do the necessary setup of the test.
2. **Act:** This is the middle step of a unit test. In this step we will execute the test i.e. we will do the actual unit testing and the result will be obtained.
3. **Assert:** This is the last step of a unit test application. In this step we will check and verify the returned result with expected results.

Environment Setup

1. Jest
2. React Testing Library
3. Enzyme

```
> npm i --save react-test-renderer  
> npm i --save jest  
> npm i --save enzyme
```

Testing with Jest and Enzyme

Creating Test Cases, Test Suites

Testing React Components

1. Test --- Load the components without crashing
2. Test --- the properties of components
3. Test --- the state of the component
4. Test --- the event handling
5. Test --- the DOM elements, input
6. Test --- the getting and setting the values to input --- state
7. Test --- state updates in events.

Playwright Component Testing

Playwright Component Testing

- Playwright Test was created specifically to accommodate the needs of end-to-end testing.
- Playwright supports all modern rendering engines including Chromium, WebKit, and Firefox.
- Test on Windows, Linux, and macOS, native mobile emulation of Google Chrome for Android and Mobile Safari.

Playwright Component Testing

- Playwright has features that support component testing with some popular web frameworks like React.
- Playwright is built to enable cross-browser web automation that is evergreen, capable, reliable, and fast.

Environment Setup and Running Tests

- `npm init playwright@latest`
- `npx playwright test`
- `npx playwright test --project=chromium`
- `npx playwright show-report`

Environment Setup and Running Tests

```
const { test, expect } = require('@playwright/test');

test.beforeEach(async ({ page }) => {
  await page.goto('http://localhost:3000/');
});

test('has title "React App"', async ({ page }) => {
  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle("React App");
});
```

Practice Hands-Ons

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

