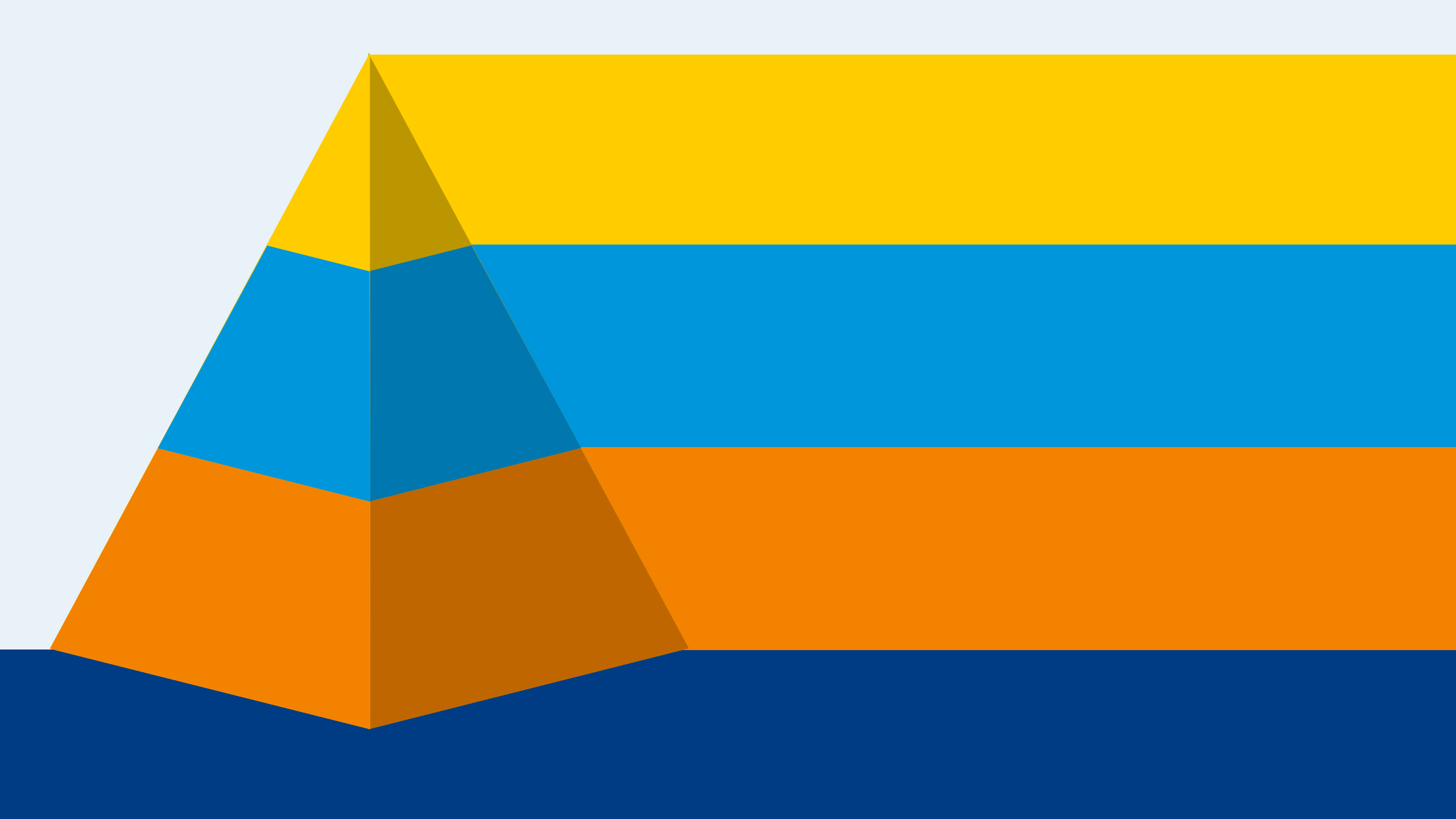
The three layers of testing

@bartwaardenburg

“We want to deliver high quality applications for our users while regularly releasing new features”



00 Static analysis

Static analysis

“Static analysis is the analysis of software that is performed without actually executing programs”

“Using Flow or TypeScript could have prevented 15% of the public bugs for public projects on GitHub”

http://ttendency.cs.ucl.ac.uk/projects/type_study/documents/type_study.pdf

```
type MyCustomButtonProps = { text: string };

const MyCustomButton = ({ text }: MyCustomButtonProps) => (
  <button>{text}</button>
);

const ButtonContainer = () => (
  <MyCustomButton text={['I', 'like', 'turtles']} />
);
```

```
const ButtonContainer = () => (
  <MyCustomButton text={['I', 'like', 'turtles']} />
);
```

```
[flow] props of React element `MyCustomButton` (This type
 is incompatible with the expected param type of object ty
pe Property `text` is incompatible:)
```

```
const MyCustomButton: ({text}: {
    text: string;
}) => JSX.Element
```

[Flow]

```
MyCustomButton: React$Element < (_: {
    text: string
}) => React$Element < string >>
```

“You can have every single variable and function completely typed and linted but still have none of your functions doing what they should be doing”

```typescript
const a: number = 1;
const b: number = 2;

const multiply = (a: number, b: number): number => a + b;

multiply(a, b);
```

01 Unit testing

00 Static analysis

Unit testing

“A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system”

```typescript
const multiply = (a: number, b: number): number => a + b;

test('Multiply should return the arguments multiplied', () => {
  expect(multiply(4, 3)).toBe(12);
)};
```

```
expect(received).toBe(expected)

Expected value to be (using ===):
    12
Received:
    7
```

“A snapshot test verifies that a piece of functionality works the same as it did when the snapshot was created”

```
const Button = ({ type }: ButtonProps) => (
  <button className={`btn-${type}`} />
);

test('The Button component renders correctly', () => {
  const component = renderer.create(
    <Button type="good" />
  ).toJSON();

  expect(component).toMatchSnapshot();
});
```

```
PASS  src/unit-test.spec.js
 ✓ The Button component renders correctly (11ms)
```

```
FAIL  src/unit-test.spec.js
 ✕ The Button component renders correctly (15ms)

 ● The Button component renders correctly

   expect(value).toMatchSnapshot()

   Received value does not match stored snapshot 1.

   - Snapshot
   + Received

     <button
   -   className="btn-good"
   +   className="btn-bad"
     />
```

“You can have every single component and function unit test passing but still have none of your functions working together like they should”

```
const multiply = (a: number, b: number): number => a * b;

const alertNumber = (value: number): void => alert(value);

const ButtonWhichShouldAlertOnClick = () => (
  <button
    onClick={() => multiply(1, 2)}
    onMouseEnter={() => alertNumber(multiply(1, 2))}
  >
    Multiply
  </button>
);
```

02 Integration testing

01 Unit testing

00 Static analysis

Integration testing

“Integration testing is the phase in software testing in which individual software modules are combined and tested as a group”

```
import { mount } from 'enzyme';

const ButtonWhichShouldAlertOnClick = () => (
  <button
    onClick={() => multiply(1, 2)}
    onMouseEnter={() => alertNumber(multiply(1, 2))}
  >
    Multiply
  </button>
);

alertNumber = jest.fn();

test('The Button component should run a function on click', () => {
  const component = mount(<Button type="test" />);

  component.find('button').simulate('click');

  expect(alertNumber).toHaveBeenCalledTimes(1);
});
```

```
FAIL  src/integration-test.spec.js
 ✕ The Button component should run a function on click (22ms)


 ● The Button component should run a function on click


   expect(jest.fn()).toHaveBeenCalledTimes(1)

   Expected mock function to have been called one time, but it was called
zero times.
```

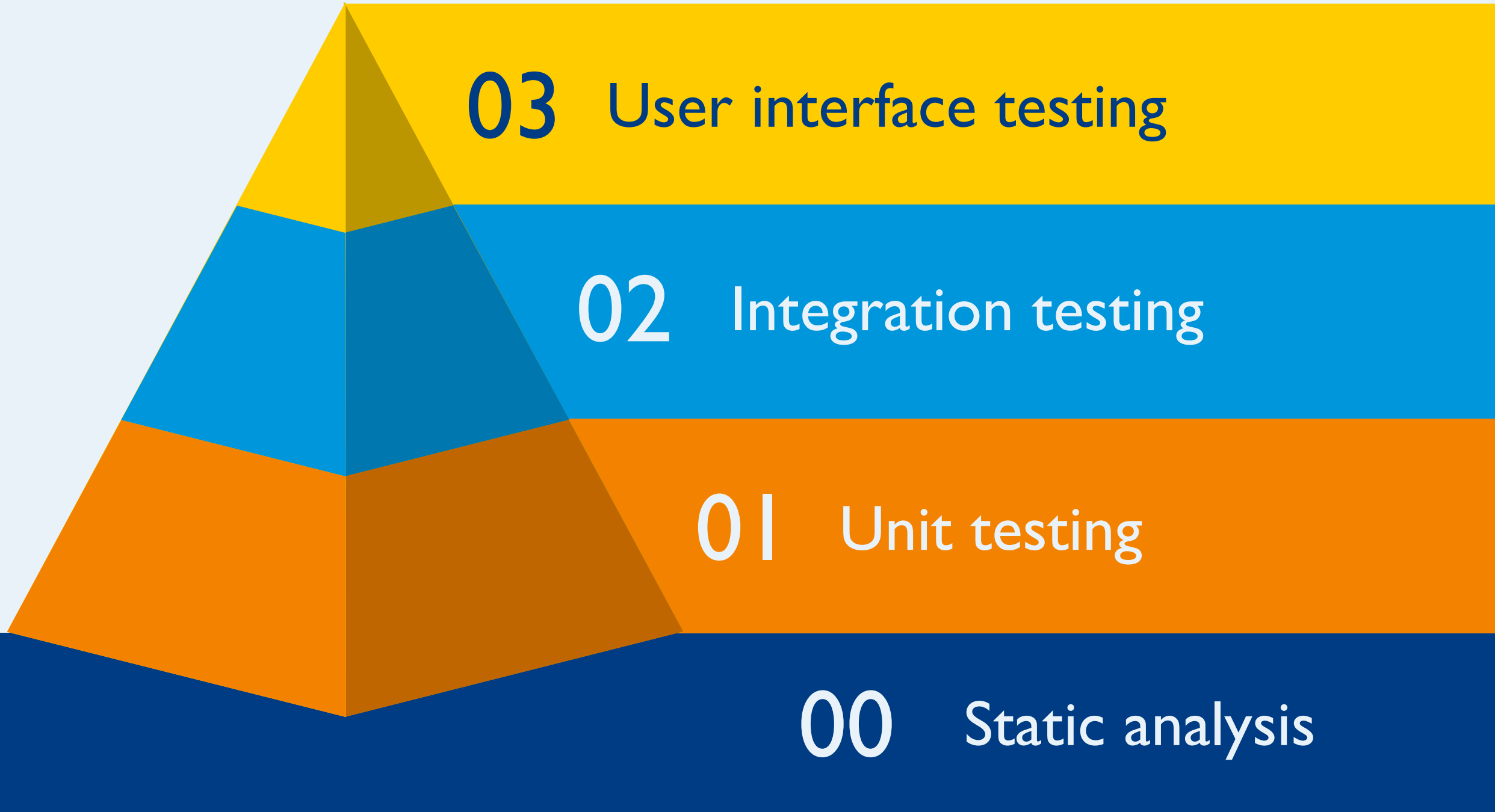
“You can have everything working together completely as intended but still have an empty screen for an application”

```
const multiply = (a: number, b: number): number => a * b;

const alertNumber = (value: number): void => alert(value);

const Button = () => (
  <button
    onClick={() => alertNumber(multiply(1, 2))}
  >Multiply</button>
);

document.querySelector(‘body').style.cssText = 'display: none';
```


03 User interface testing
02 Integration testing
01 Unit testing
00 Static analysis

User interface testing

“User interface testing is the process of testing a product's graphical user interface to ensure it meets its specifications”

tools

- Selenium
- Nightmare
- Nightwatch
- TestCafe
- CasperJS
- TestCafe
- Protractor
- Cypress
- Puppeteer
- Codecept
- Navalia
- Chromeless

```
import {Chrome} from 'navalia';
import {toMatchImageSnapshot} from 'jest-image-snapshot';

expect.extend({toMatchImageSnapshot});

const chrome = new Chrome();

test('The routing input component should display as expected', async () => {
  await chrome.goto('https://www.anwb.nl/verkeer/routeplanner');
  await chrome.wait('.ROVE-routing-input');
  const screenshot = await chrome.screenshot('.ROVE-routing-input');
  await chrome.done();

  expect(screenshot).toMatchImageSnapshot();
});
```

Van: adres, postcode en/of plaats

Naar: adres, postcode en/of plaats

+Via

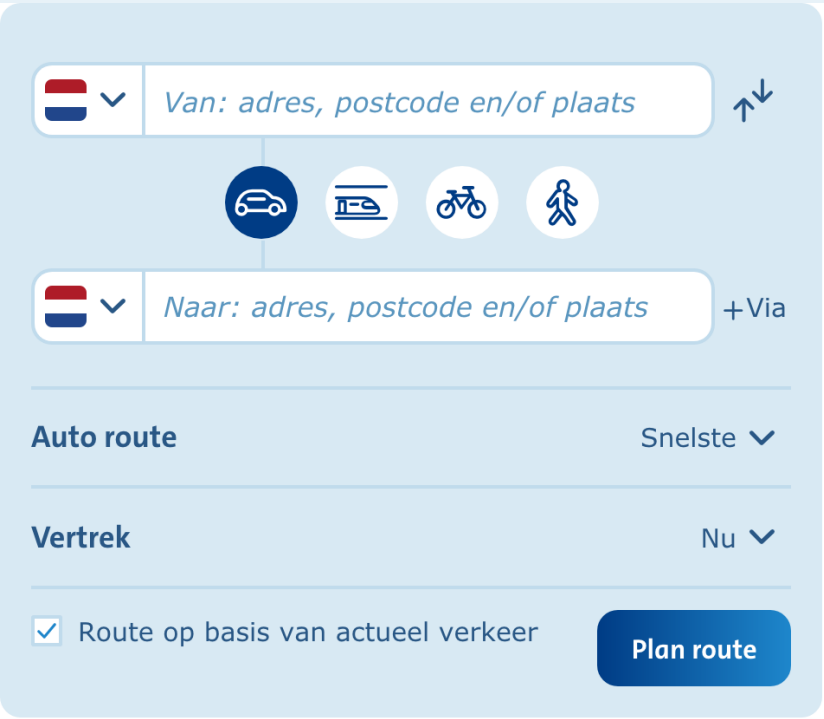
Auto route Snelste

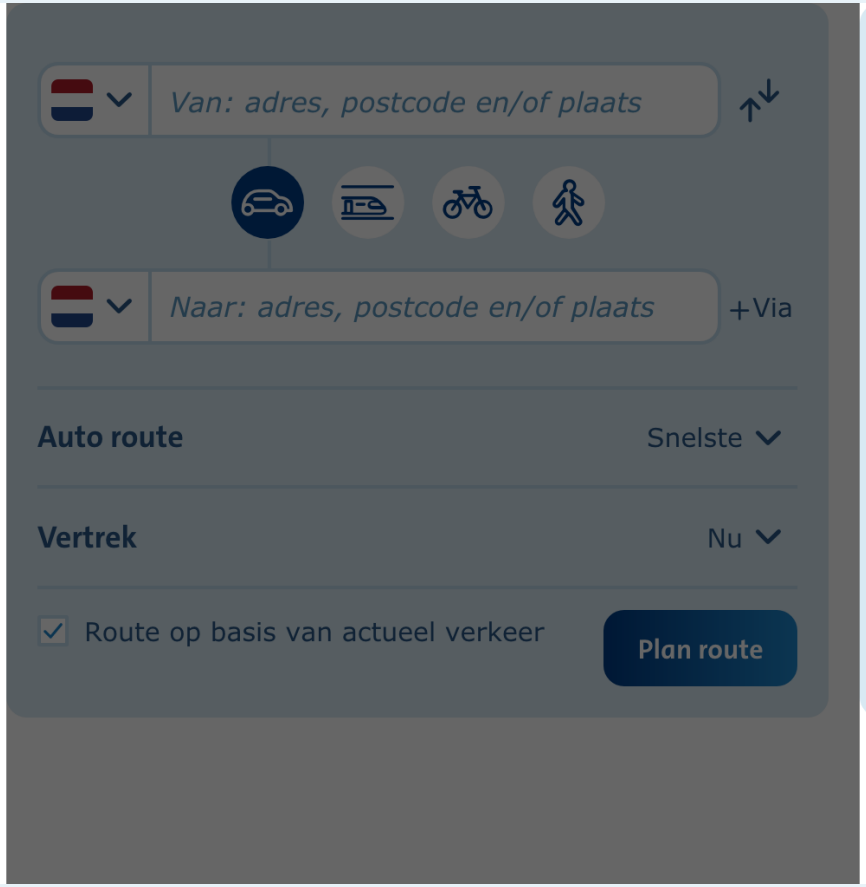
Vertrek Nu

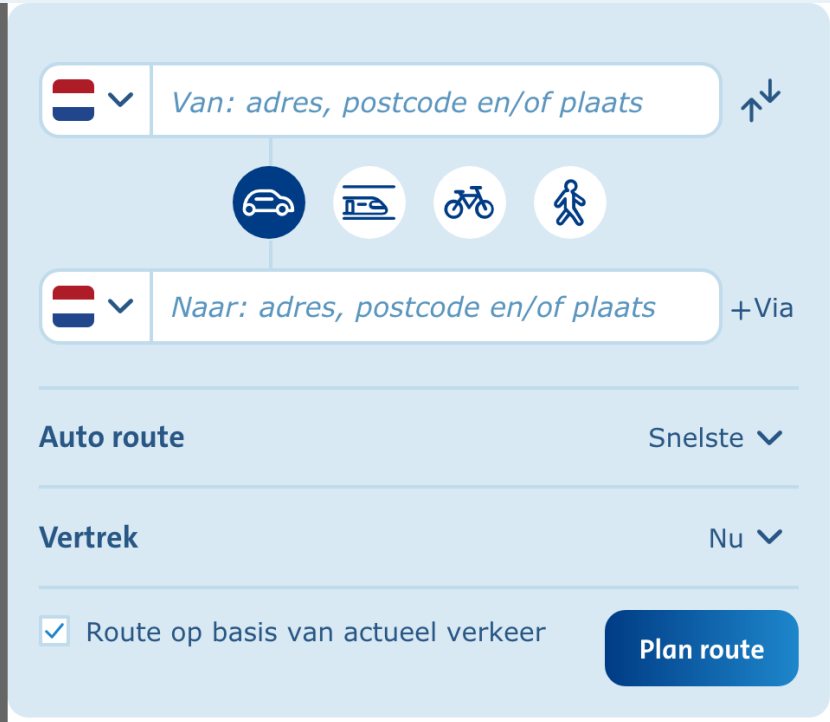
Route op basis van actueel verkeer

Plan route

```
 PASS  src/components/routing-input-address/tests/RoutingInputAddress.ui-test.js (7.153s)
  ✓ The routing input component should display as expected (3671ms)
```


Van: adres, postcode en/of plaats
Naar: adres, postcode en/of plaats
+Via
Auto route
Snelste
Vertrek
Nu
Route op basis van actueel verkeer
Plan route


Van: adres, postcode en/of plaats
Naar: adres, postcode en/of plaats
+Via
Auto route
Snelste
Vertrek
Nu
Route op basis van actueel verkeer
Plan route


Van: adres, postcode en/of plaats
Naar: adres, postcode en/of plaats
+Via
Auto route
Snelste
Vertrek
Nu
Route op basis van actueel verkeer
Plan route

```
FAIL  src/components/routing-input/tests/RoutingInput.ui-test.js (9.909s)
  × The routing input component should display as expected (9033ms)


  ● The routing input component should display as expected


    Expected image to match or be a close match to snapshot.

    See diff for details:
/Users/p279825/Sites/ANWB/traffic/src/components/routing-
input/tests/__image_snapshots__/__diff_output__/routing-input-ui-test-js-
the-routing-input-component-should-display-as-expected-1-diff.png
```

Van: adres, postcode en/of plaats

ANWB maakt gebruik van cookies

NWB gebruikt cookies op haar website om het gebruik van de website te nalyseren, gebruiksgemak te verbeteren, voor social media en om ervoor te orgen dat je relevante advertenties en informatie te zien krijgt wanneer je ebruik maakt van de ANWB website. Meer informatie over de cookies kun je inden in ons privacy- en cookiestatement. Je geeft door gebruik te blijven naken van deze website of door hiernaast op de button 'akkoord' te klikken oestemming voor het gebruik van cookies en het verwerken van op deze wijze erkregen persoonsgegevens, zoals in ons privacystatement wordt vermeld.

eer weten over deze cookies? Lees dan voor meer informatie verder op nwb.nl/cookies.

Van: adres, postcode en/of plaats

ANWB maakt gebruik van cookies

NWB gebruikt cookies op haar website om het gebruik van de website te nalyseren, gebruiksgemak te verbeteren, voor social media en om ervoor te orgen dat je relevante advertenties en informatie te zien krijgt wanneer je ebruik maakt van de ANWB website. Meer informatie over de cookies kun je inden in ons privacy- en cookiestatement. Je geeft door gebruik te blijven naken van deze website of door hiernaast op de button 'akkoord' te klikken oestemming voor het gebruik van cookies en het verwerken van op deze wijze erkregen persoonsgegevens, zoals in ons privacystatement wordt vermeld.

eer weten over deze cookies? Lees dan voor meer informatie verder op nwb.nl/cookies.

+Via

Auto route Snelste

Vertrek Nu

Route op basis van actueel verkeer

Van: adres, postcode en/of plaats

Naar: adres, postcode en/of plaats

+Via

Auto route Snelste

Vertrek Nu

Route op basis van actueel verkeer

Plan route

```
import {Chromeless} from 'chromeless';

const chromeless = new Chromeless();

const screenshot = await chromeless
  .goto('https://www-ontw.anwb.nl/verkeer/routeplanner')
  .screenshot('#routing', {
    base64: true,
  });

const file = new Buffer(screenshot, 'base64');

expect(file).toMatchImageSnapshot();

await chromeless.end();
```

1. Local Setup

```
new Chromeless()
```

via CDP

on

Chrome runs locally

2. Remote Proxy Setup

```
new Chromeless({
  remote: true
})
```

Forwards commands via Proxy

via

Websockets

```
new Chromeless()
```

via CDP

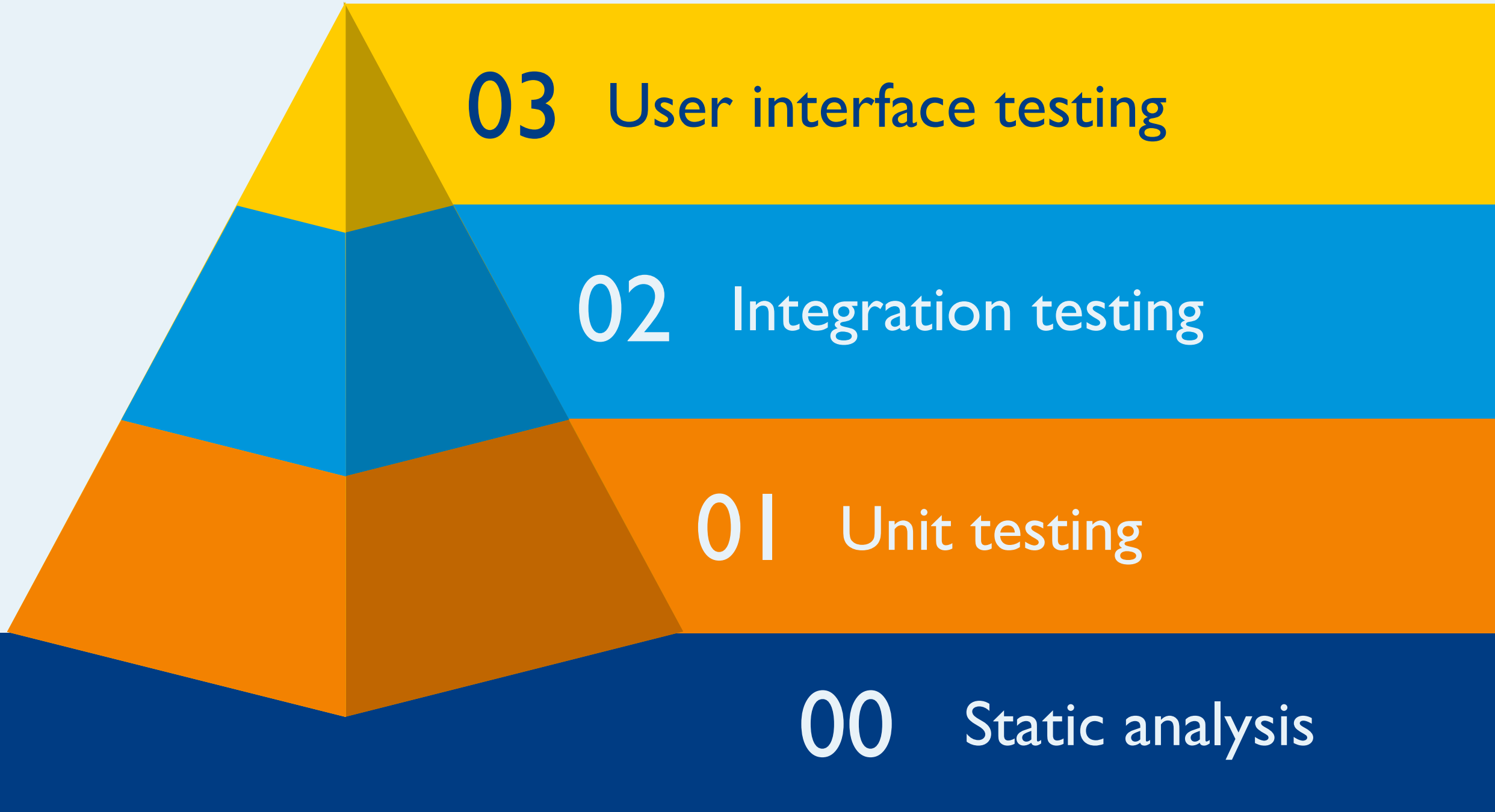
on

Chrome runs on AWS Lambda

```
const chromeless = new Chromeless({
  remote: {
    endpointUrl: 'https://XXXXXXXXXX.execute-api.eu-west-1.amazonaws.com/dev’,
    apiKey: 'your-api-key-here’,
  },
});
```

“With chromeless you can run hundreds of browsers in parallel”

“You can easily execute > 100.000 tests for free in the free tier”


03 User interface testing
02 Integration testing
01 Unit testing
00 Static analysis

thanks & happy testing


@bartwaardenburg