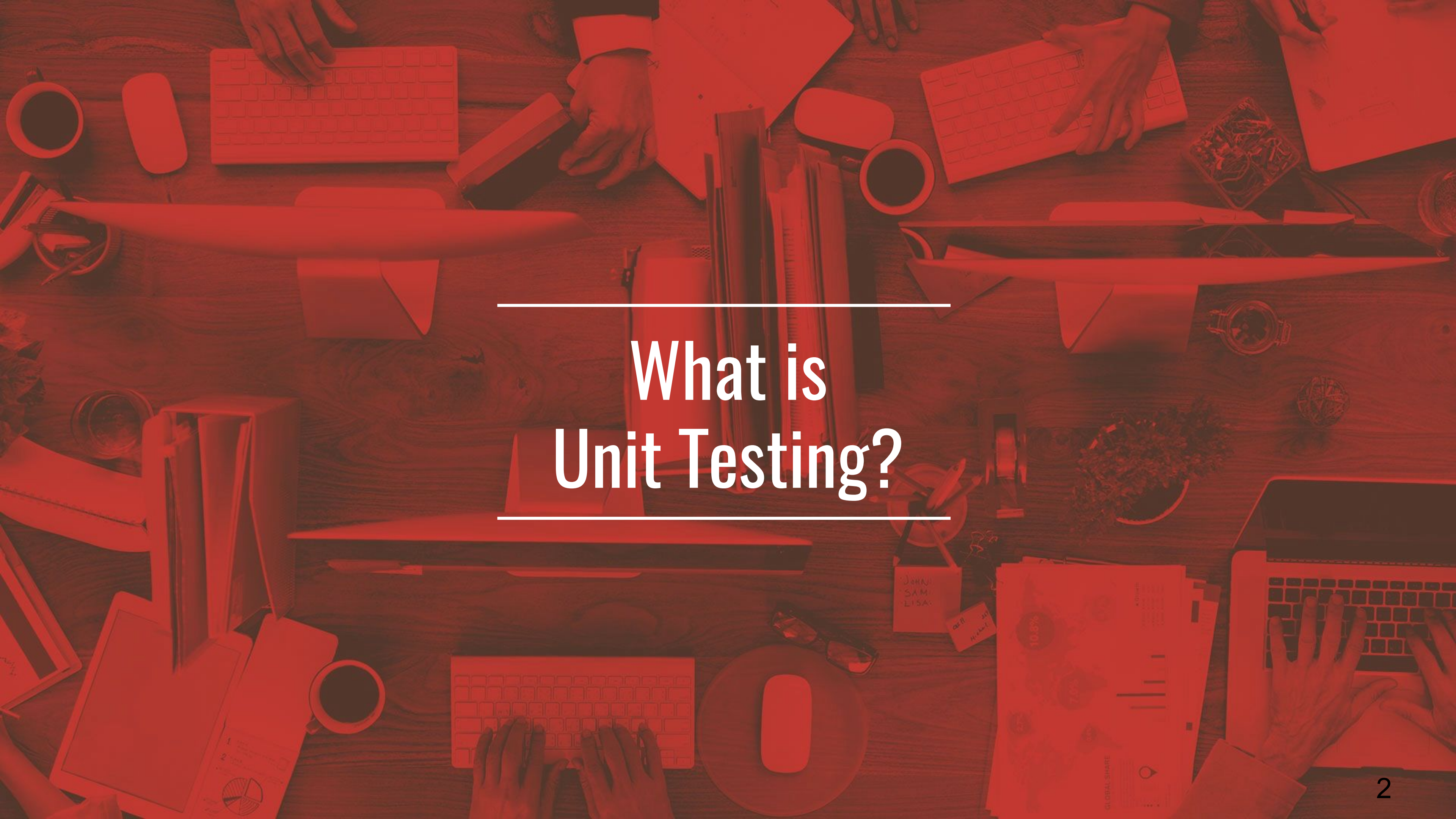
What did I break?

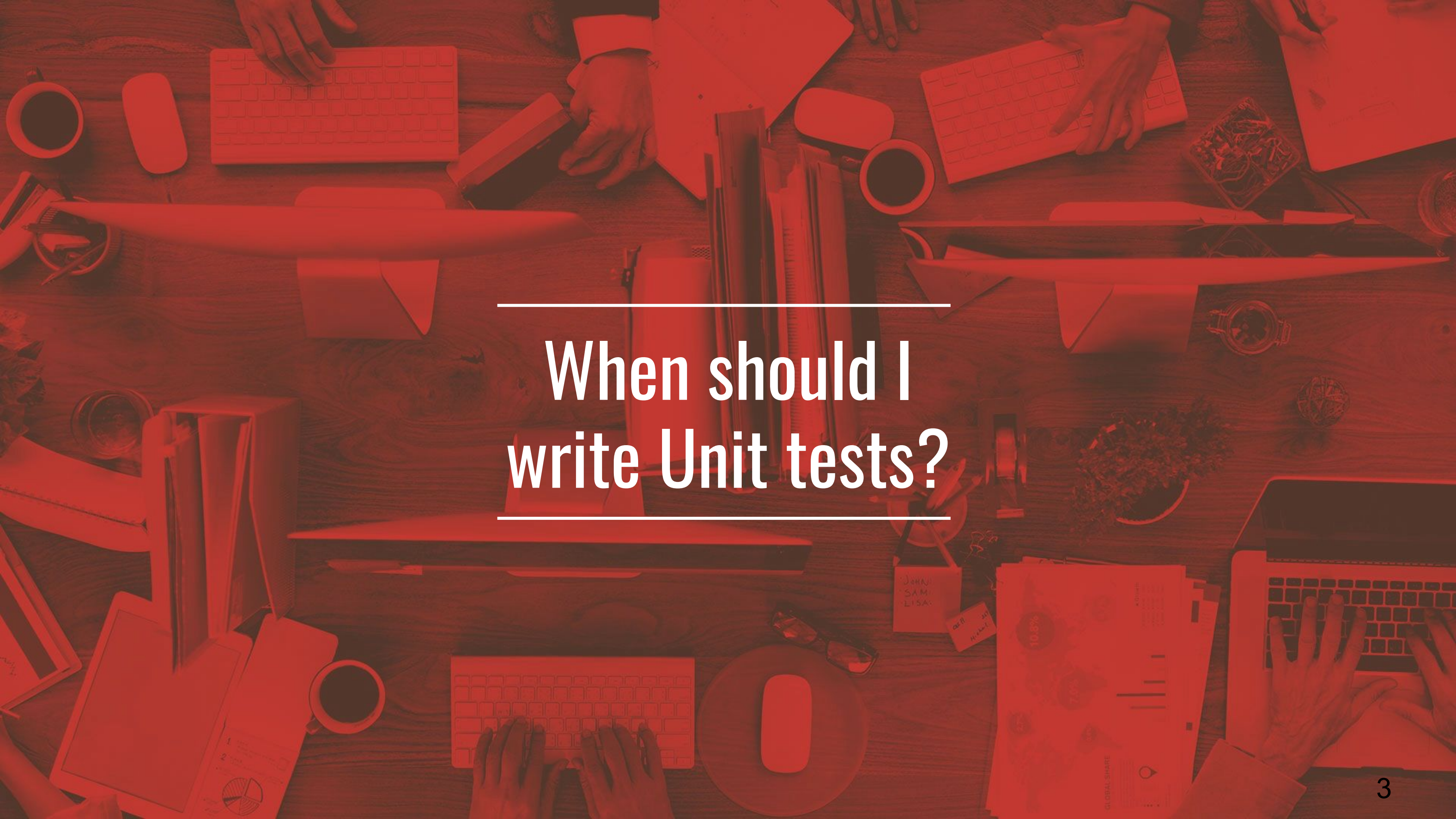
Unit testing

CryptoConverter-unit-test

WIZELINE®



What is Unit Testing?



When should I write Unit tests?

Make your code testable

1. **Separate** concerns.
2. Maintain your logic **platform independant**.
3. Follow the **dependency inversion principle**.
4. Always think in how **are you going to test** your code before writing it.
5. If possible use **TDD**

Set up your test environment

1. **Add the following dependency to gradle:**
   a. `testCompile 'junit:junit:4.12'`
2. **Create a the following directory structure:**
   a. `app/src/test/java/{packageName}`
3. **Run**

Let’s code

Mockito

```
testCompile
'org.mockito:mockito-core:2.8.47'
```

Mocks, Stubs and verification

- Mock:
  - Creates a dummy object that fulfills some expectations.
  - `@Mock`
  - `mock(Class<T>)`
- Stub:
  - Creates a dummy implementation of a portion of code.
  - `when().then()`
- Verification:
  - Verifies that a method was called with certain params
  - `verify(object, typeOfVerification).method(params).`

More code!

The end...