Benefits of Automated Testing

1- Test your code frequently, in less time

2- Catch bugs before deploying

3- Deploy with confidence

4- Refactor with confidence

5- focus more on the quality

Types of Tests

1- Unit

2- Integration

3- End-To-end

depend on your project

"Tests unit of App without its external dependencies"

"Tests the App with its external dependencies"

"Drive an App Through its UI"
Selenium

Testing Frameworks

1- NUnit
2- MSTest
3- XUnit

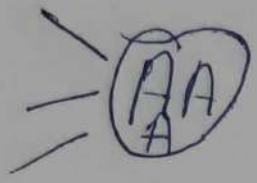
Library

Test Runner

- To Test apply convention

Method_ScenarioTest_Expected behaviour

- inside every Test Method we have 3 parts
  - Arrange
  - Act
  - Assert



Arrange ⟹ prepare object To Test, InTialize

Act ⟹ ACT on this object, {call Method that we have Tes

Assert ⟹ verify is result correct

TDD "Test Driven Domain"

With TDD you write your Tests before writing the production code

How TDD works

- Write a failing test
- Write the simplest code to make the test pass
- Refactor if necessary

Benefits of TDD

- Testable source code
- full coverage by tests
- Simpler implementation

Test first / Code firs

⊛ Untrustworthy Tests

Techniques

- When Test Correct & run successfully
- Create a bug in production if still run successfully

so Test not Test the right Things (not Trustworthy Test)

not Test private Method as Implementation details change

⊛ Code Coverage Tools

- Scan your code and Tell you what parts of your code not tested

0
0 | 2
0

2
1 | 2
2

4

Breaking External Dependencies

Loosely Coupled Design

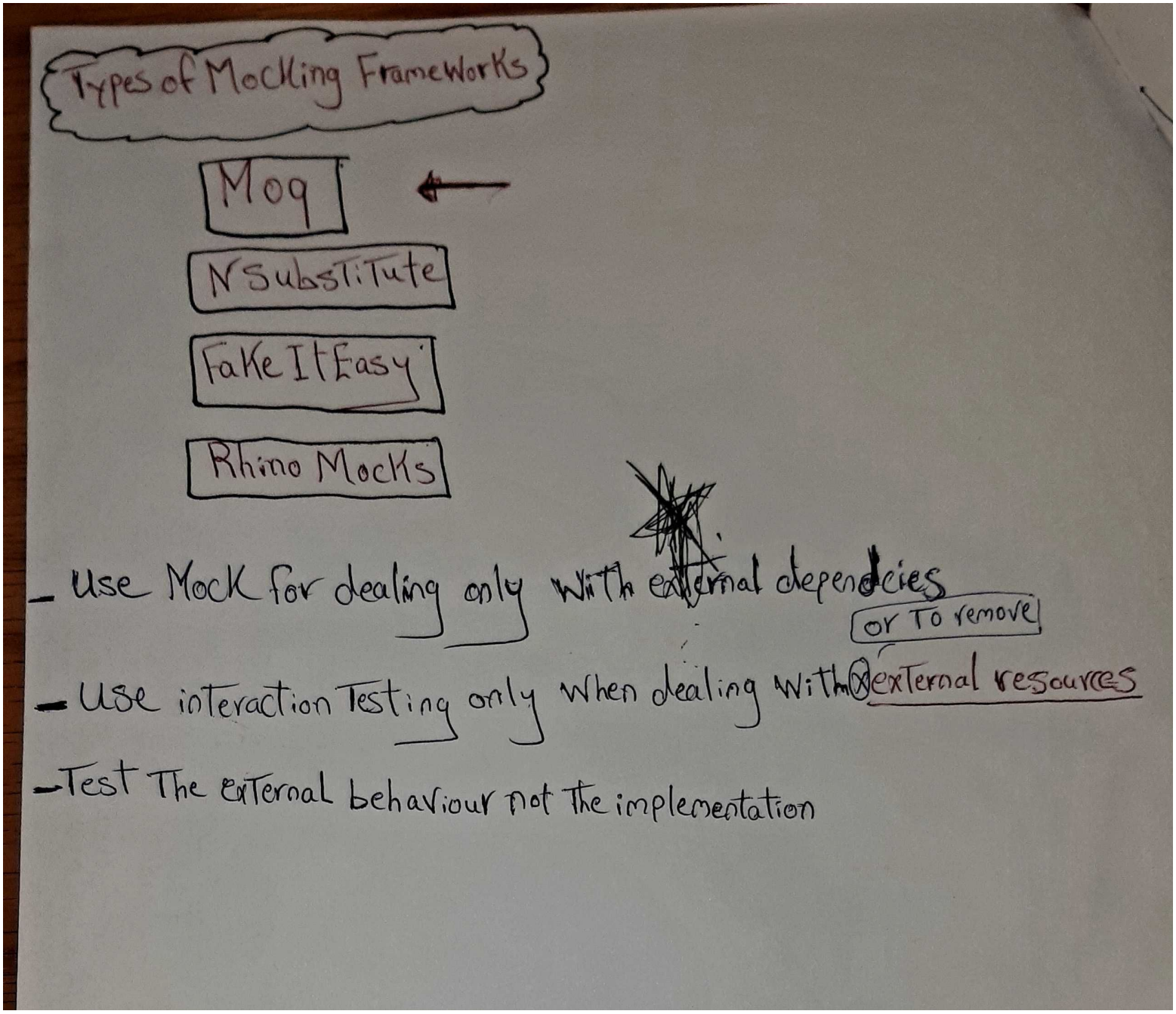
- you Can Change object To fake object (Test Double)

5

- extract code That use an external resources To a seperate class (isolated it from The rest of The code)
- extract The Interface from That class depend on contract

Fake - Mock - Stub of Testing

put it in Testing Not in production



hints:

You Should test the outcome of the method and think as a method is a blackbox