Agenda

1) Testing
2) TDD (Test driven development)
3) Flaky test cases
4) Types of testing

- Best Practices of testing
- Mocking
- Doubles

- Testing 3 → Write UTs
  - Mocking
  - Assertions

eg:

```
int numberOfLanes(int lanesOnEachSide)
{
    return multiplyUtility(lanesOnEachSide);
}

int multiplyUtility(int x)
{
    return x * 2; 3
}
```

int numberOfMarriedPeople(int numberOfCouples)

{

return multiplyUtility(numberOfCouples)

}

[10K places the multiply func" getting used

Testcases 10K scenarios

We should write testcases, that should get executed automatically before anyone is trying to submit the code and if any of the test case fail, code submission won't be allowed.

[Developers should be writing these testcases

- while developing feature
- Comprehensive

TDD

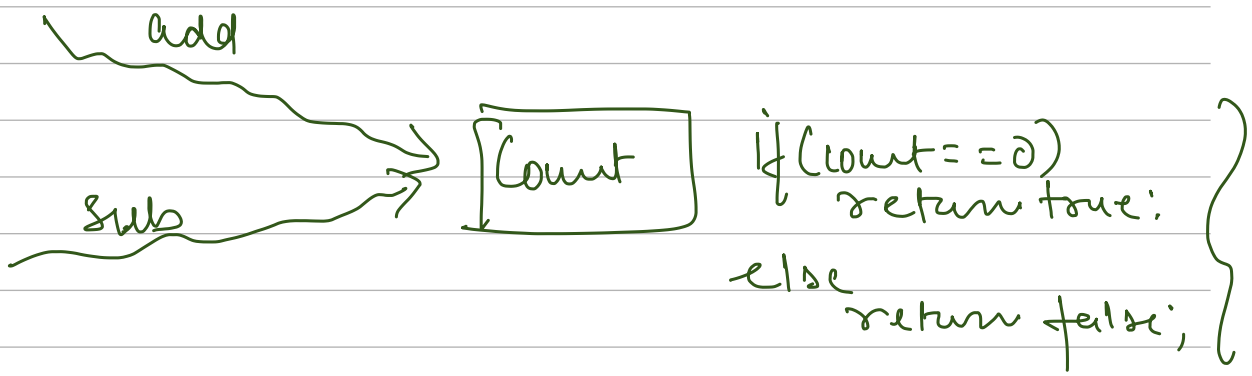
first write all testcases and implement the feature

figure every corner case upfront

[Gojek | thoughtworks]

Flaky tests

```
test {
    number = randomNumber(0,100);
    if(number > 50)
        return true;
    else
        return false;
}
```



Flaky Test → Unreliable

- N/w connection
- Randomization
- Concurrency

[Try not to have flaky test in your codebase]

Types of testing

- Unit testing
- Integration testing
- Functional testing

```
class A {
    foo() {
    }
}
```

UNIT

```
a() {
    b();
}
```

Test case of A fails

1. Code of A
2. Code of B

→ Bussiness logic

Testing in Isolation

[Mocking: Hardcoding the Response of dependencies

We should test our code in isolation

UNIT TESTING

Every individual piece of code should get tested by a testcase

1. Every test should be short & fast
2. No dependency on any other func^{n} (dependencies should be Mocked)

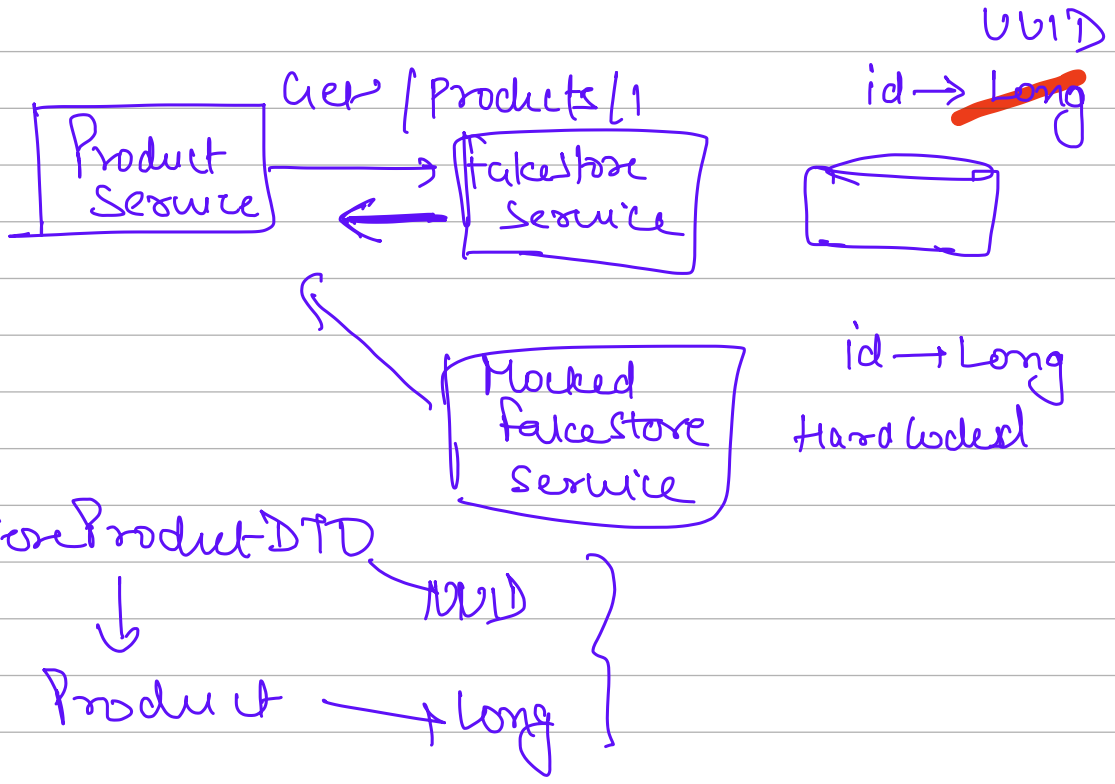
Code Coverage = % of code covered by the test cases

- Jacoco
- Sonar

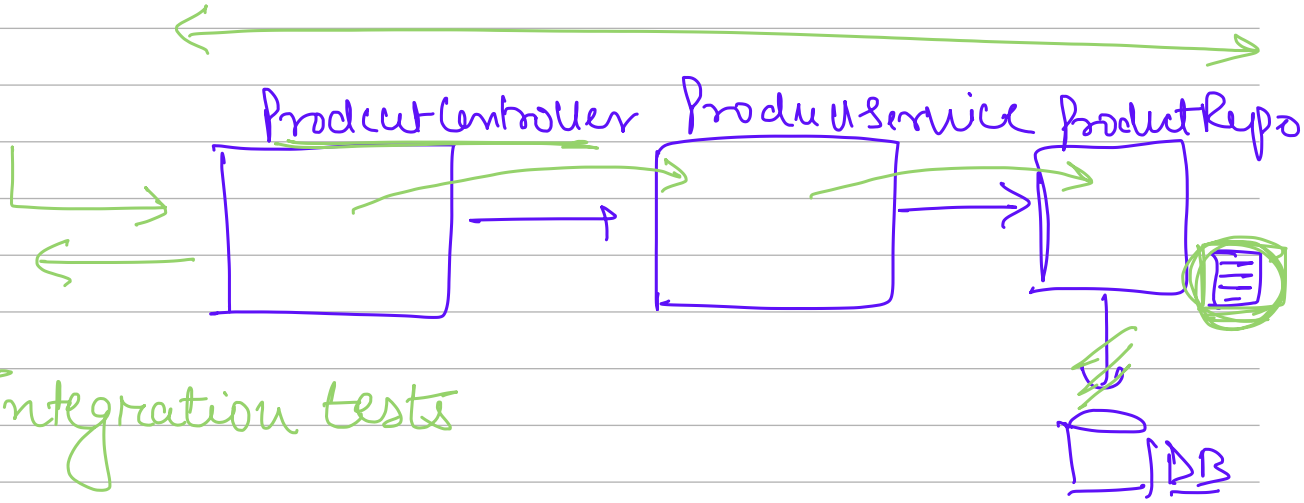
≈ 100 %

Industry Standard ≈ 80%

Integration test



Testing each functionality of our software system where all the dependencies will also be triggered as it is they are present in real Codebase



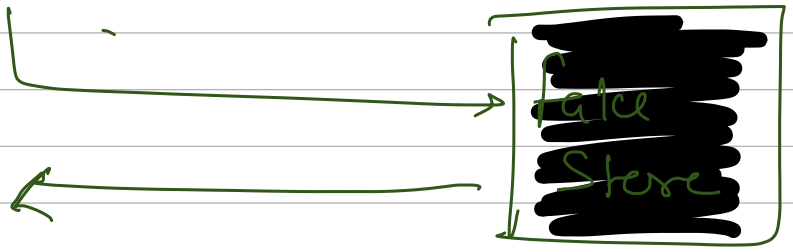
Functional testing = USER Testing

End 2 End testing

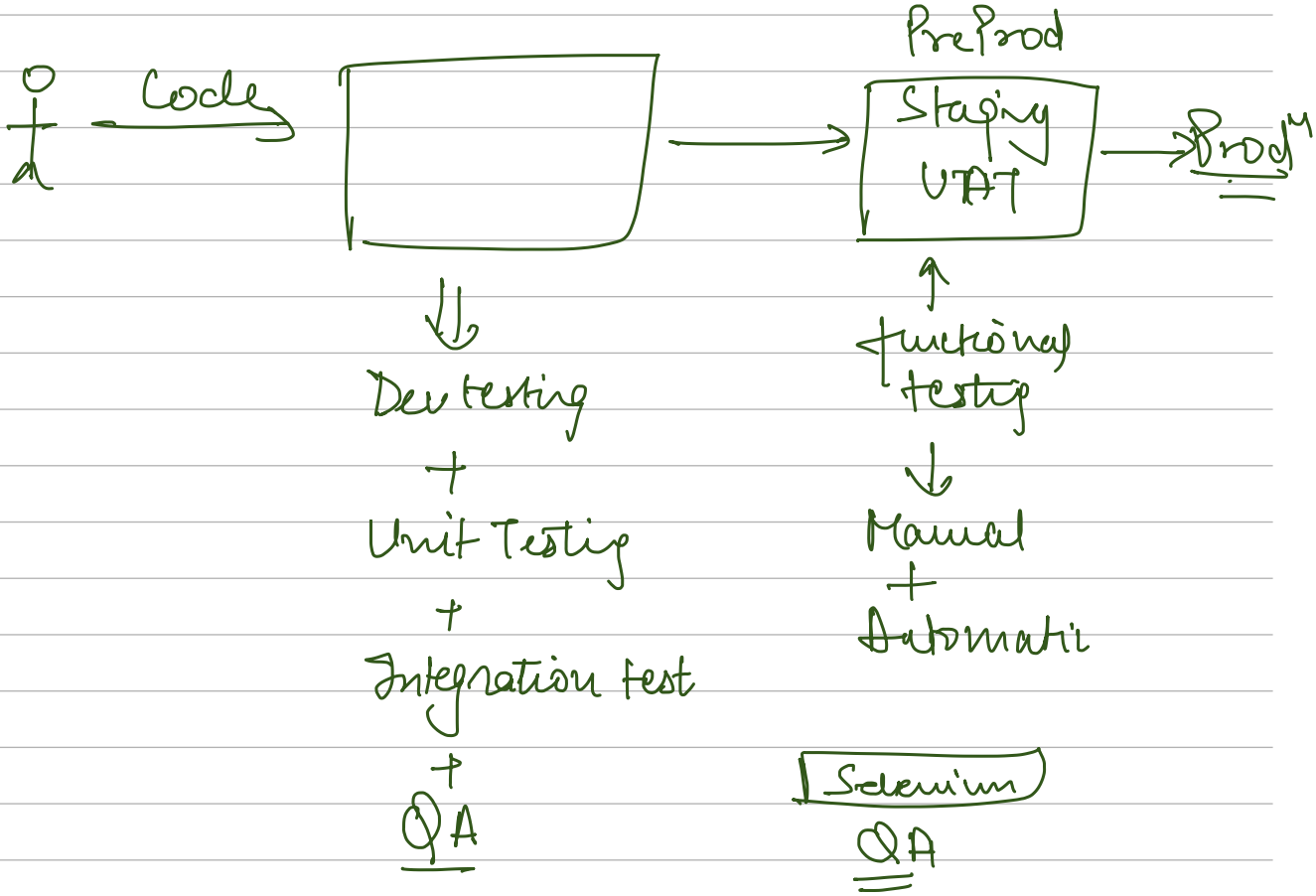
Confluence

↳ Create Page

GetProducts()



User Acceptance test



Number functions

Vs >

Number of service

Vs >

Number of places

User Interacts