Retro #2

Fluent Interface

Fixture Builders.

Mockito

~~Data~~ → Mock (interaction)

Unit → ваш выбор

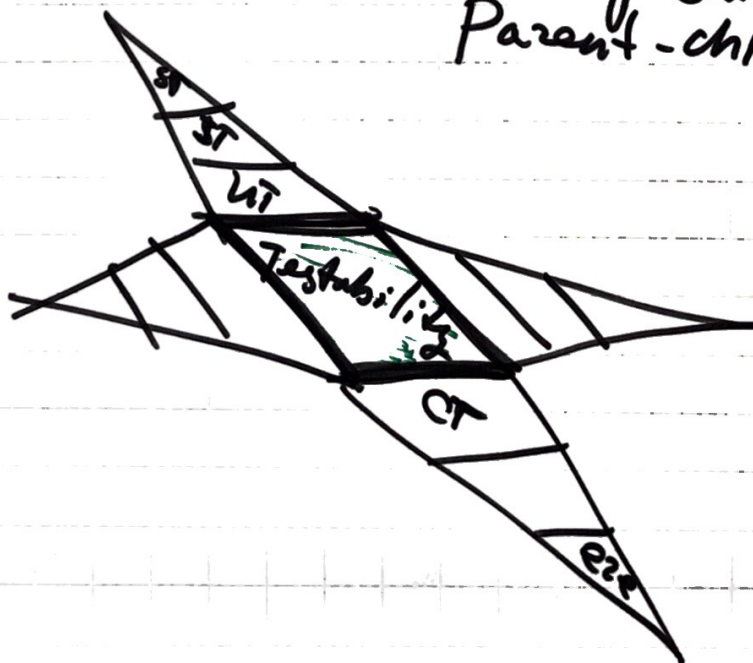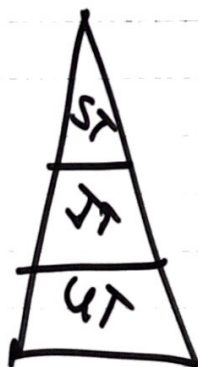Test-Doublers

Testability

Mockito
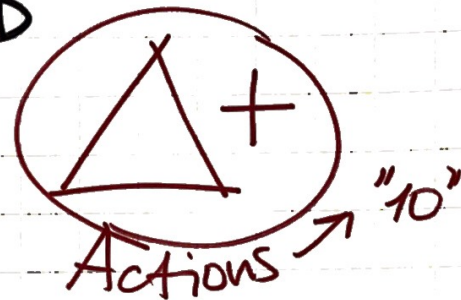WireMock

(TA) Actions

BDD

Actions ↗ "10"

"Что покрывать?"
"Optimal Coverage"
"Что делать с QA?"

Sonar ← PIT
Code Coverage tool

Копирование контекста
Parent-child контекст
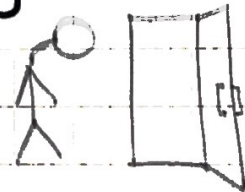
← Как проверить качество тестов? (действия)

- [ ] Результат обработки
- [ ] Broken Test (first)
- [ ] Mutation coverage
- [ ] Code Review → поведение, bugs
     → Внутр. качество, finding
- [ ] Test Coverage

Suites   PO, analysl

```
┌─────────────────┐        ┌──────────────┐
│  Testability    │ ──────▶│  Readable    │
└─────────────────┘        └──────────────┘
        │
        ▼
┌─────────────────┐    D✗D   D─▶🔲--D
│  Losely         │
│  Coupled        │     DIP,           SRP.
└─────────────────┘
        ┊              DI ─▶ «context»        ┌──────────────┐
        ┊             Creational  test ─▶ Spring │  High        │
        ▼             pattern                    │  Cohesion    │
┌─────────────────┐                              └──────────────┘
│  Autonomous     │
│  tests          │
│  ( unit )       │
└─────────────────┘
```
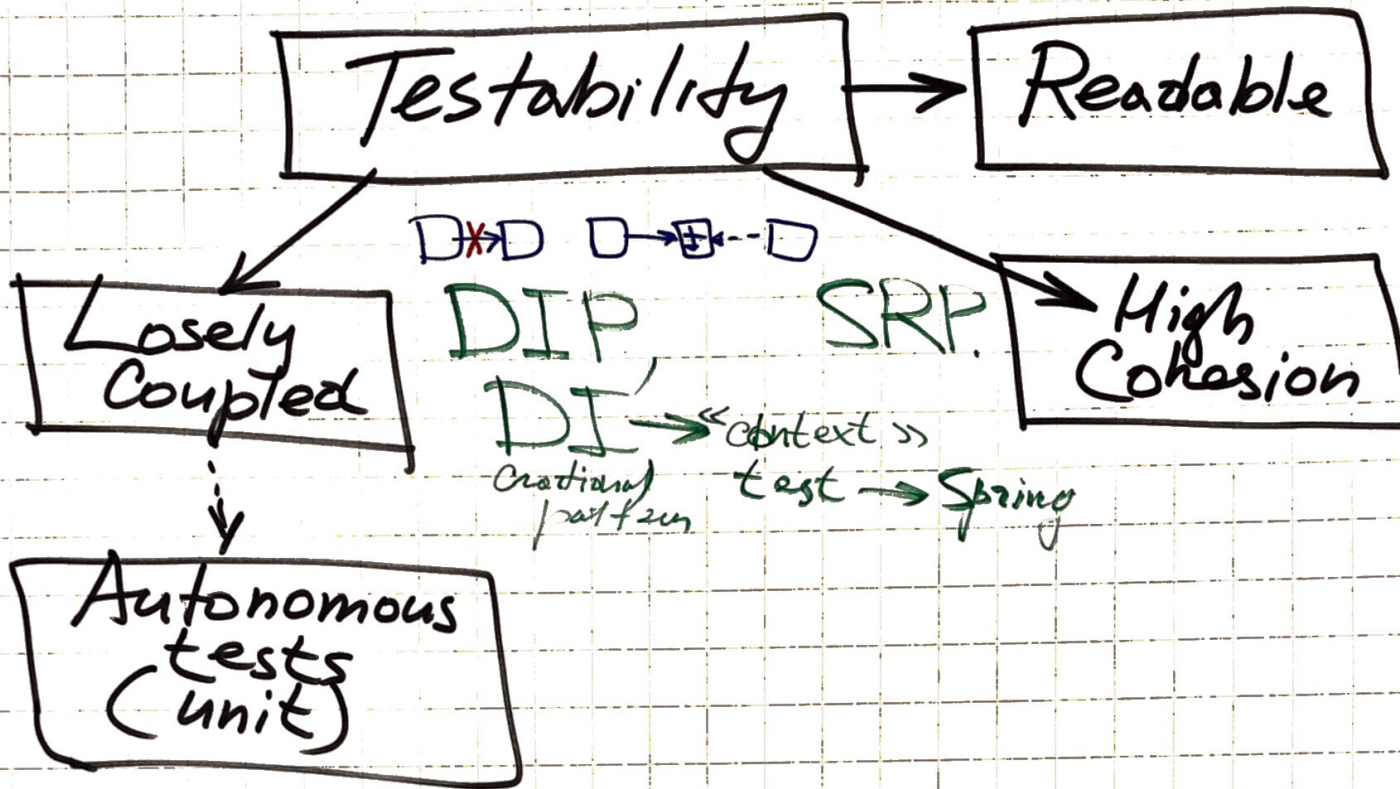
Фокус приложения усилий
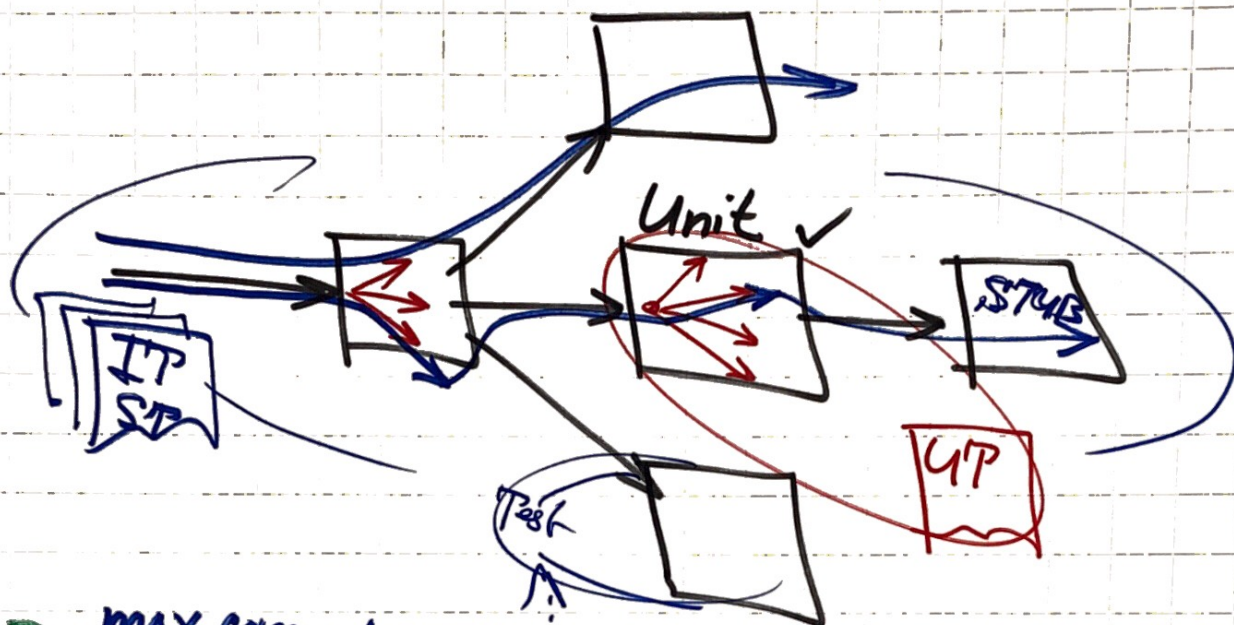
Unit. ✓

IT
ST

STUB

UT

Test

max coverage

PO

Ø. Размер Тех. налог

$ Dev, Sup

TTM ↓
Увел. Quality
Локализуемость

1.1. Greenfield:

UT — Unit — S / S

test Scope

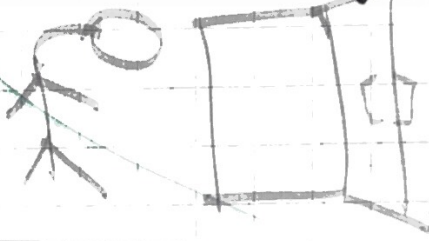### 1.2. Brownfield.

ST
IT — □ □ □ □ → STUB

2. Flows to cover
1. Business Risk
2. Hot flows units

BDUF

Судьба тестировщика

Как думает:
1. Альтерн. сценарии
2. Эмоц. привяз. к сист. Нет
3. QA → модель внутр. качество
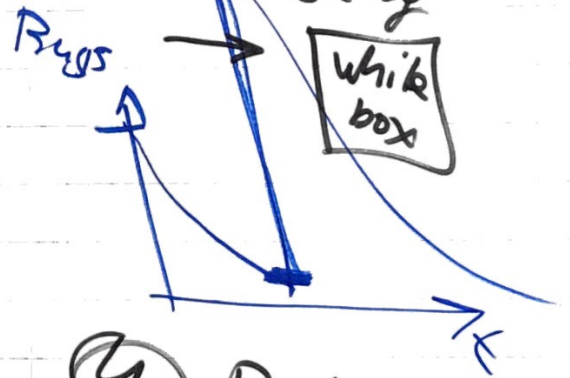
проиг. либо

Шу - Ха - Ри
("Best") ("Хорошие") (Факторы)

1. Man. QA → Auto test Dev
2. Dev
3. Explorative testing

white box

Bugs →

4. Product Analysis