Test-driven development

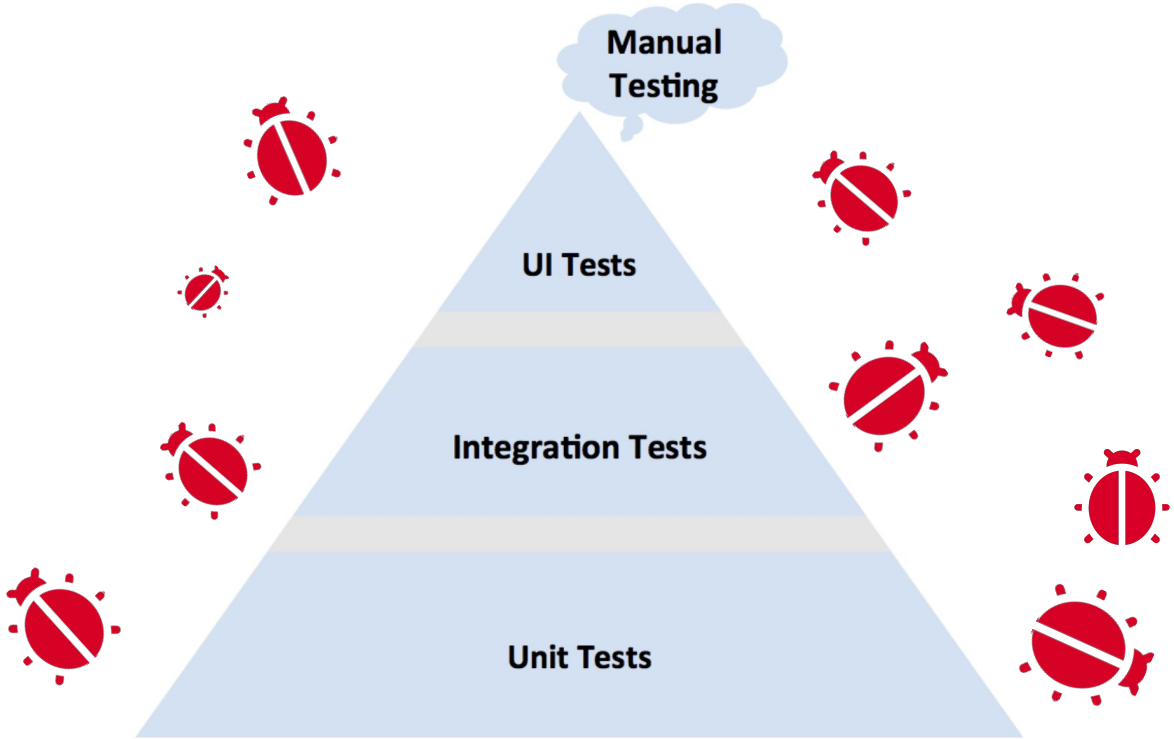
Tomislav Fabeta (Software Developer)

degordian

What are tests for?

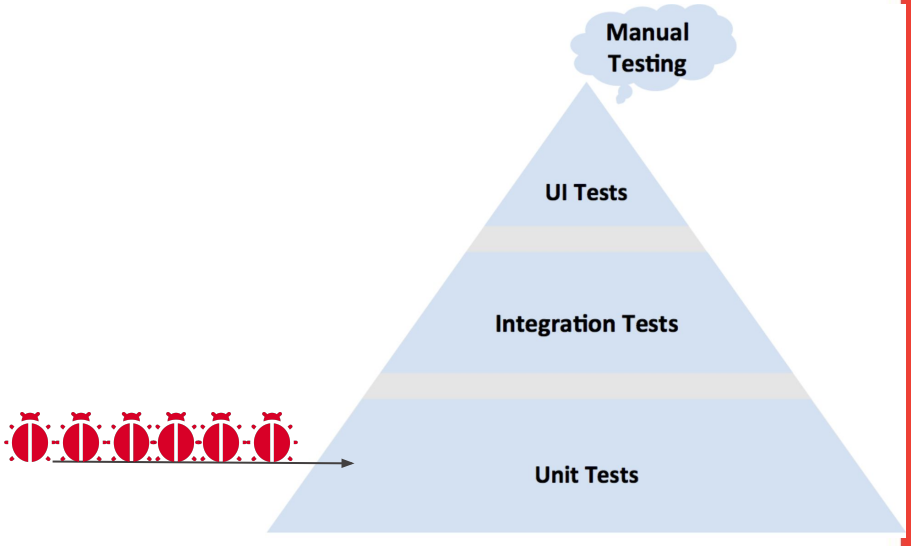
- prove that a feature really works
- give courage to other contributors
- sustainable code

Test Pyramid



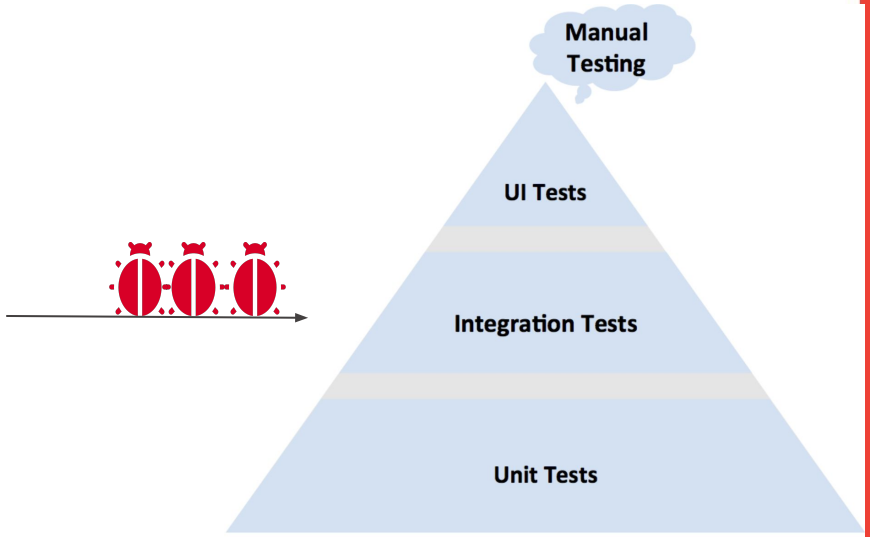
Unit Tests

- the very base of Test Pyramid
- prove your code units really work



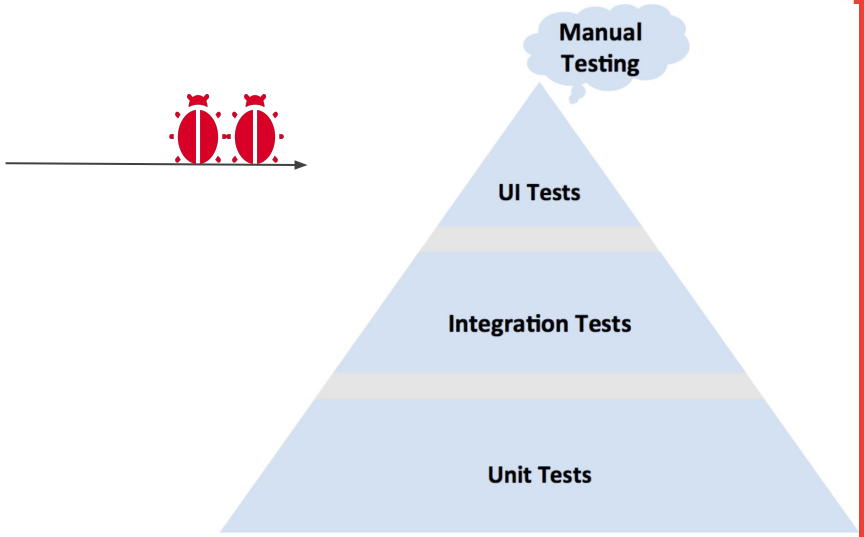
Integration Tests

- prove connected units really work
- harder to create good coverage



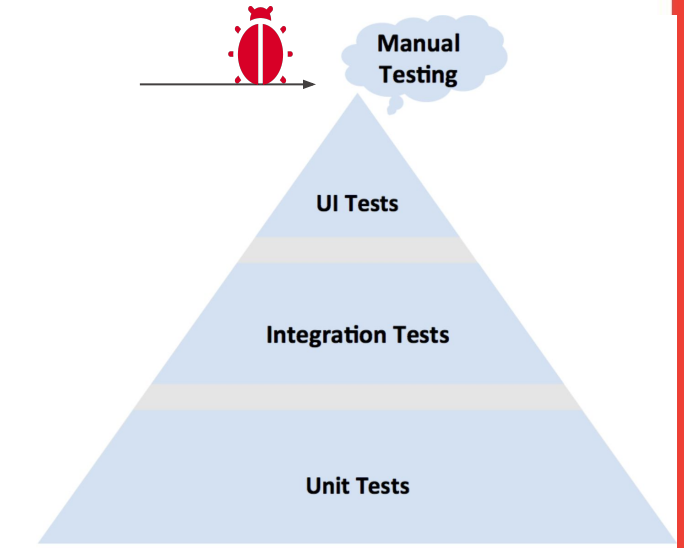
UI Tests

- prove that GUI really works
- even harder to have a good coverage
- pricey

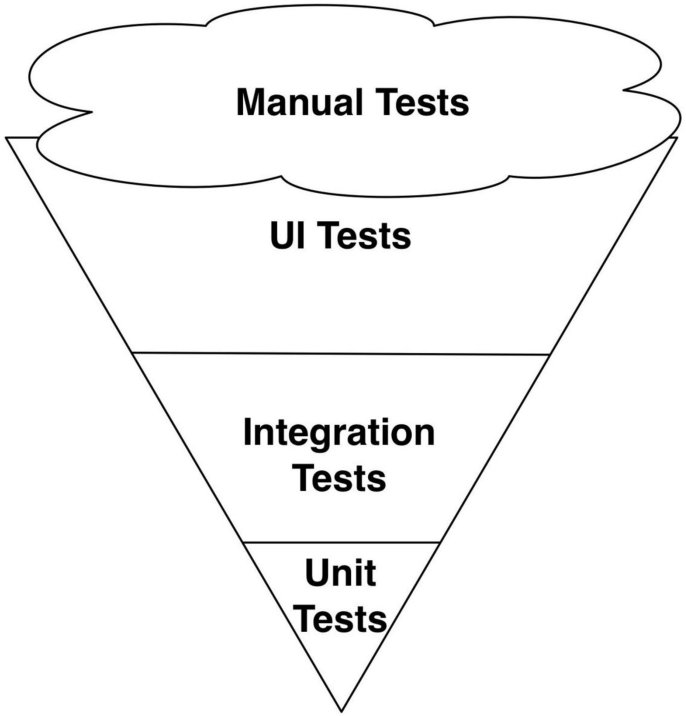


Manual Testing/ Exploratory Testing

- most expensive tests by time and money consumption



Ice Cream Cone



JS Testing Frameworks & Tools

- Mocha (https://mochajs.org/)
- Jasmine (https://jasmine.github.io/)
- Jest (https://facebook.github.io/jest/)
- Chai (http://www.chaijs.com/)
- Karma (https://karma-runner.github.io/2.0/index.html)
- Istanbul (https://istanbul.js.org/)
- Sinon (http://sinonjs.org/)
- Selenium (https://www.seleniumhq.org/)

Test-driven development

1. Add a test
2. Run a tests (if “fail” go 3. else go 1.)
3. Make a little change (go 2.)

Live Example

Pros

- simple, elegant & modular code
- sustainable code
- can speed up development in long term

Cons

- a lot of time & effort up front
- difficult to write good tests
- time & effort to maintain those tests when design is changing rapidly

Thank you for listening

https://www.facebook.com/tomislav.fabeta | https://github.com/bonzzy