Unit Testing

ERIN KEITH

Goals

1. Testing Background
2. Unit Testing

Reflection

What do you think testing is?

What testing have you done before?

What are bugs?

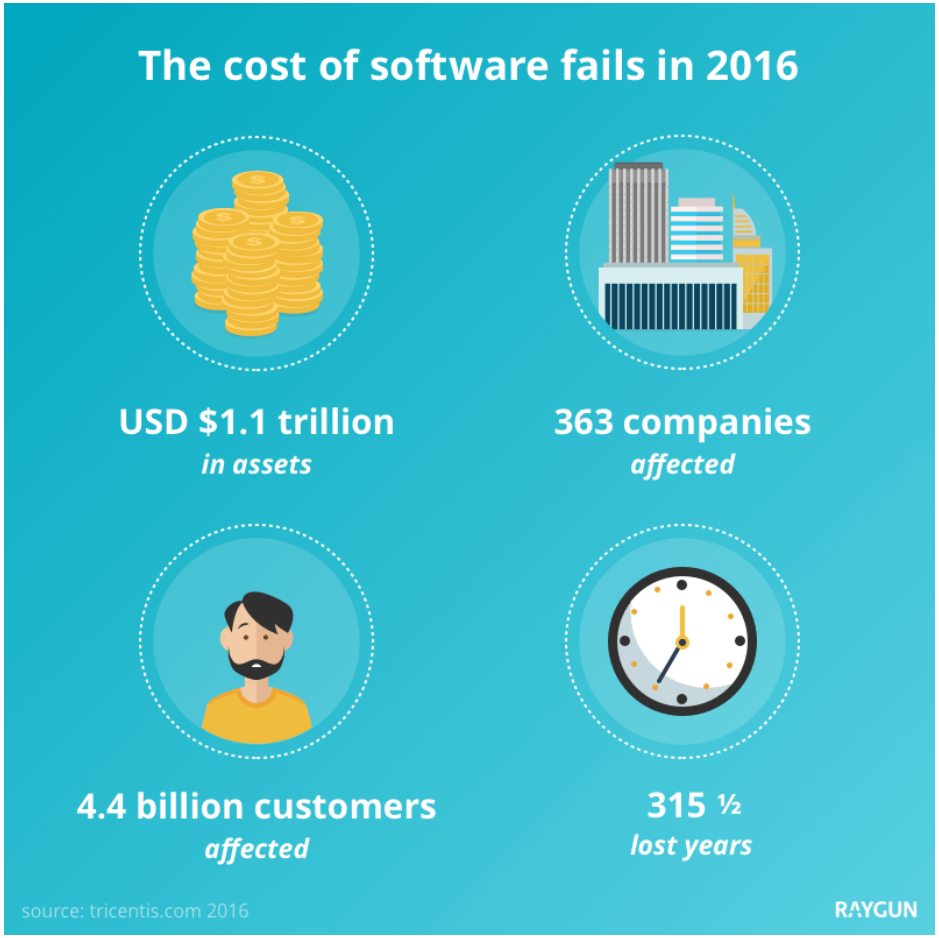
Why should we test?

Bugs Cost

Money:

"In August 2013, Amazon lost $4.8M after going down for 40 minutes due to a software 'glitch'…".

(https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2018-report/The-Cost-of-Poor-Quality-Software-in-the-US-2018-Report.pdf)



Bugs Cost

Confidence:

Customers who see or experience bugs do not trust your ability to produce quality software.

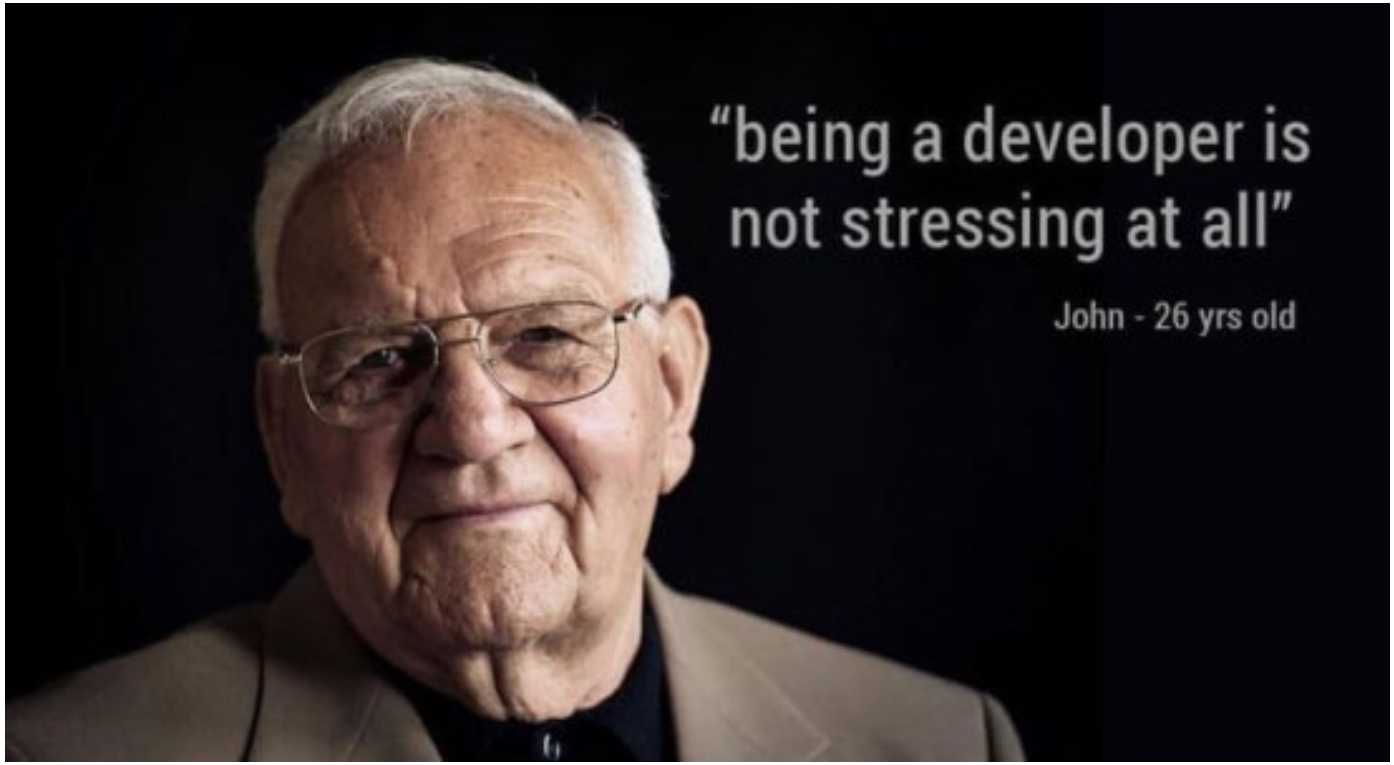
Case Study

Apple Maps

Bugs Cost

Developer Peace of Mind:

Your teammates are counting on you.



Old Testing Techniques

Print statements
Comparing output
Have others break it
Debugger
Hard coding variables
Commenting out a chunk of code

Pros?

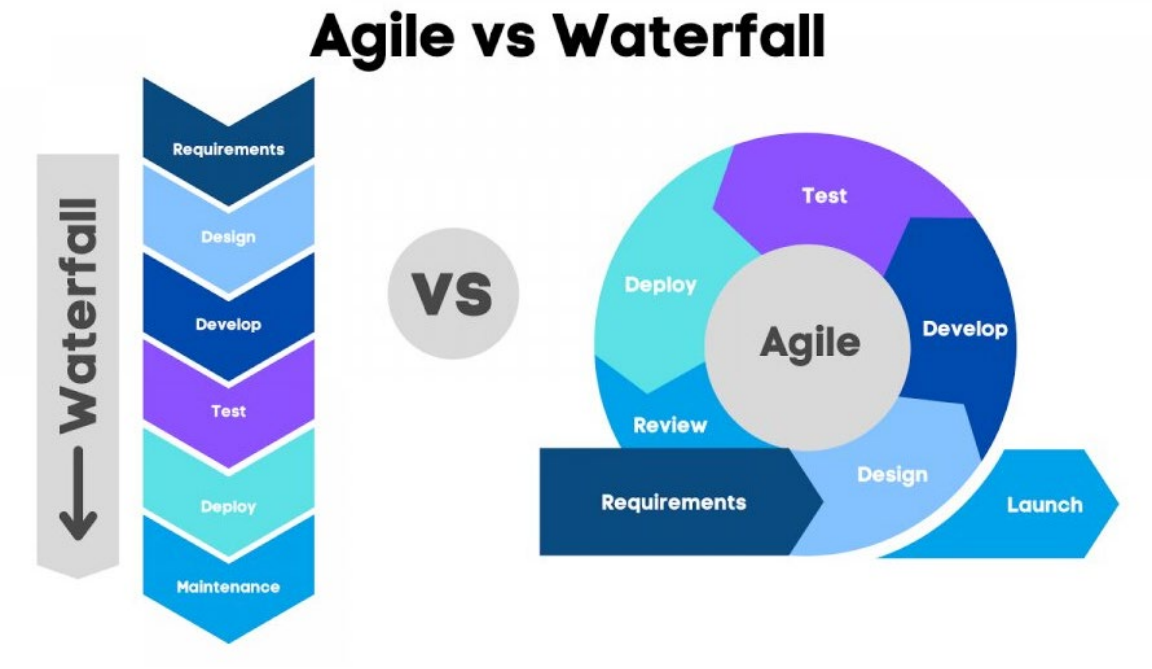
Frequent reviewing of code makes you very familiar with it
Effective
Simple

Cons?

Slow, tedious
Cannot reproduce errors/behavior
Lots of editing
risk of adding more errors
Erasing things that work

Better Ways

We can test at the different stages and automate the processes.



Earlier is Better

BEFORE DEVELOPMENT

Requirements

- This is why LISTENING is so important!

AFTER DEVELOPMENT

QA

- Alpha and Beta Testing
- Acceptance Testing
- Black Box Testing
- System or End-to-End Testing
- Load Testing
- Monkey Testing
- Performance Testing
- Sanity Testing

Earlier is Better

Between requirements-gathering and QA, developers can:

Unit Test

- Test individual units of software

Integration Test

- Test the components of software successfully work together

Regression Test

- Test to verify changes to parts of the software does not break anything else

Require automation framework => **DevOps**

Unit Tests

Test the individual units of software.

- Fix bugs early in the development cycle and save costs.
- Serve as project documentation
- Provide an indicator of confidence
- Improves efficiency

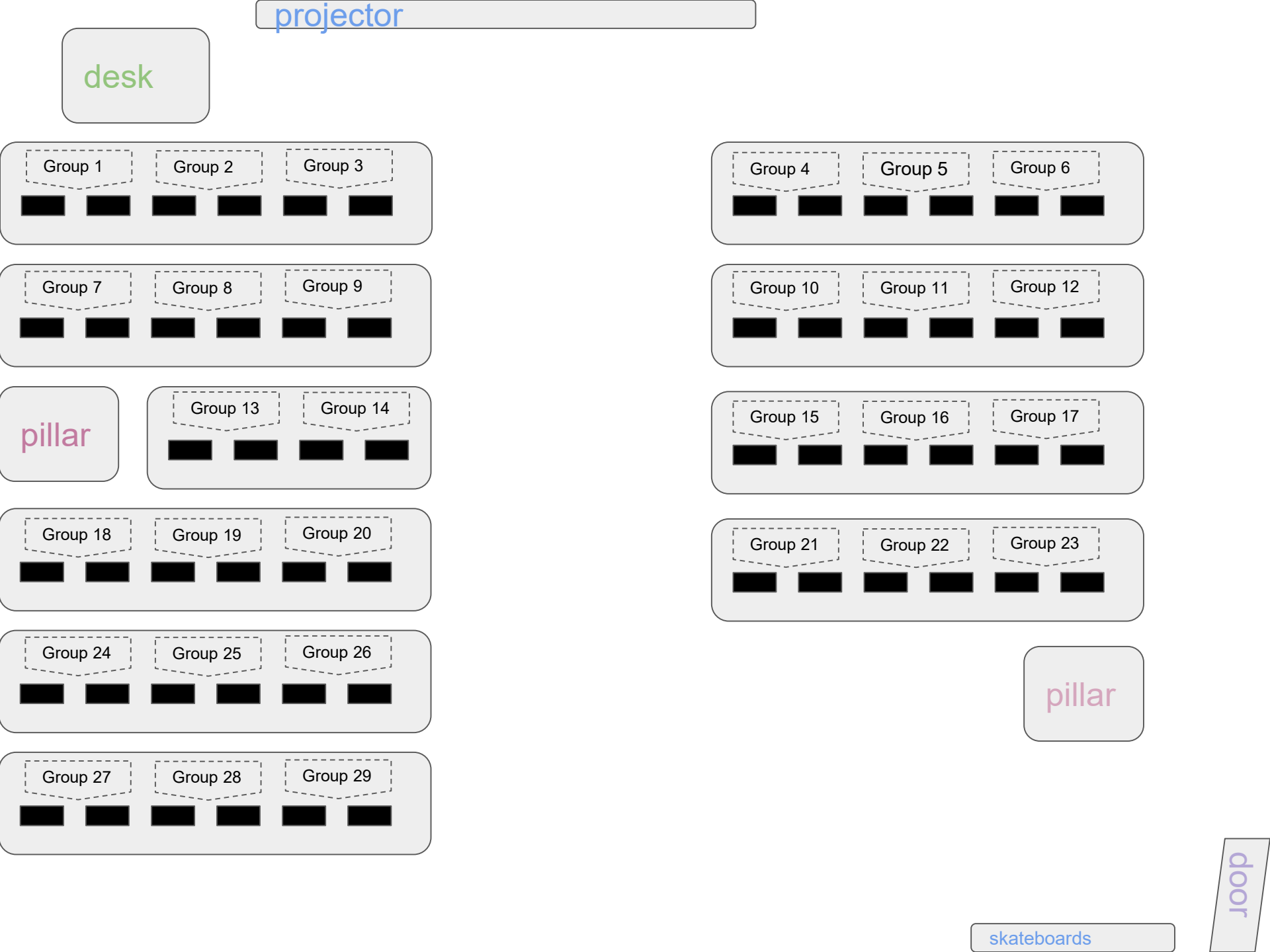
Unit Tests

- **Unit tests** are methods with some additional special functionality.
  - it's code that tests other code
  - languages usually have testing libraries or frameworks
  - A **unit test runner** is a program which executes test code and provides feedback on the results.

Unit Tests

A testing unit should focus on one tiny bit of functionality and "prove it correct".

- Arrange, Act, Assert,
  - Setting up class instances in the desired state
  - Executing the test code
  - Verifying you have achieved the desired results
- There should be one assertion per test method.
- Tests should generally either pass or fail
  - Avoid tests that might timeout or be inconclusive.


projector
desk
Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7
Group 8
Group 9
Group 10
Group 11
Group 12
pillar
Group 13
Group 14
Group 15
Group 16
Group 17
Group 18
Group 19
Group 20
Group 21
Group 22
Group 23
Group 24
Group 25
Group 26
pillar
Group 27
Group 28
Group 29
door
skateboards