***Agile Testing Pyramid***

/\_\

Manual Testing

Human Testing

Alpha/Beta tests

Takes a lot of time

/////

Functional Tests

* what you started with

////////////////////////////////////

Integration Tests

/////////////////////////////////////////////////////////

Unit Tests

//////////////////////////////////////////////////////////////////////////////

Didn’t change preexisting behavior

Deeper meaning of confidence

Know exactly what changes you make without worrying about breaking

LOOPS SWITCHES…

BRANCH

USING IF/ELSE LOOPS TO HIT DIFFERENT BRANCHES TO TEST CODE

NEVER WRITE LINE OF CODE W/O FAILING TEST BEHIND IT

90s/00s common to overdo things

There was lots of Overly complicated code

-if theres not a reason to write this code then I shouldn’t be writing this code

**Red Green Refactor Cycle**

* at the beginning we’re gonna write a failing test
* Write as little code as necessary to make the failing test pass
* Once we have passable code then we go improve it
* Repeat

Code Works? T/F?

True: Improve => Move on

Fail: fix => Pass/True

**Gherkin**

***Given*** *a name parser*

***When*** *passing in “Martin Cross”*

***Then*** *The first name should be Martin*

*And the last name should be “Cross”*

*Arrange - context*

*Act – Doing*

*Assert – checking to see if met*

Not compiling code is not the same as failing a test

Multiple Projects => Need to tell them about eachother

Real code doesn’t need to know about tests

Test Need to know about real code