

Nadeem Shabir from Talis

Software Testing

What is it?

Find software errors and verify that an application or system is fit for use. It is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Software testing has been around for a very long time!:

... been around for a long time

⊙ The First Computer

❖ Charles Babbage's Analytical Engine

- Conceived in 1837 (unfinished)
- The input via punched cards
- The output via a printer, a curve plotter and a bell



⊙ The First Programmer

❖ Ada Byron, Lady Lovelace

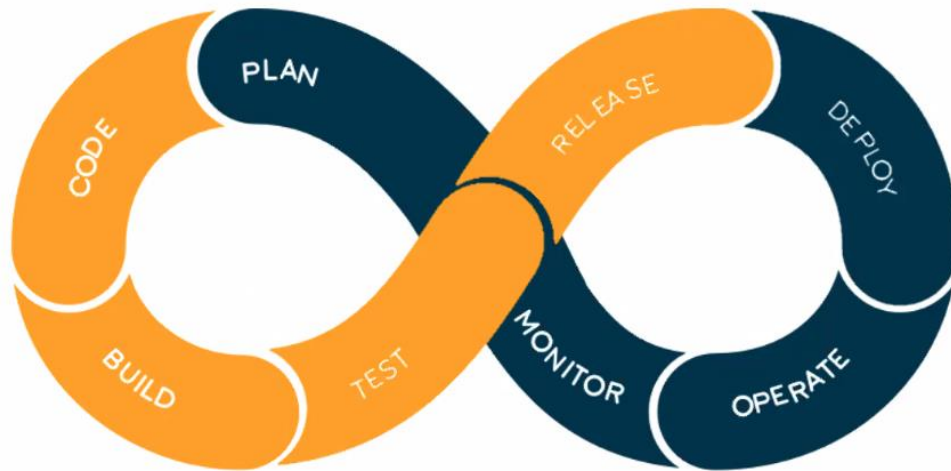
- Worked on the Analytical Engine (1843)
 - Ada's notes on the Analytical Engine were recognized as a description of a computer and software



talis

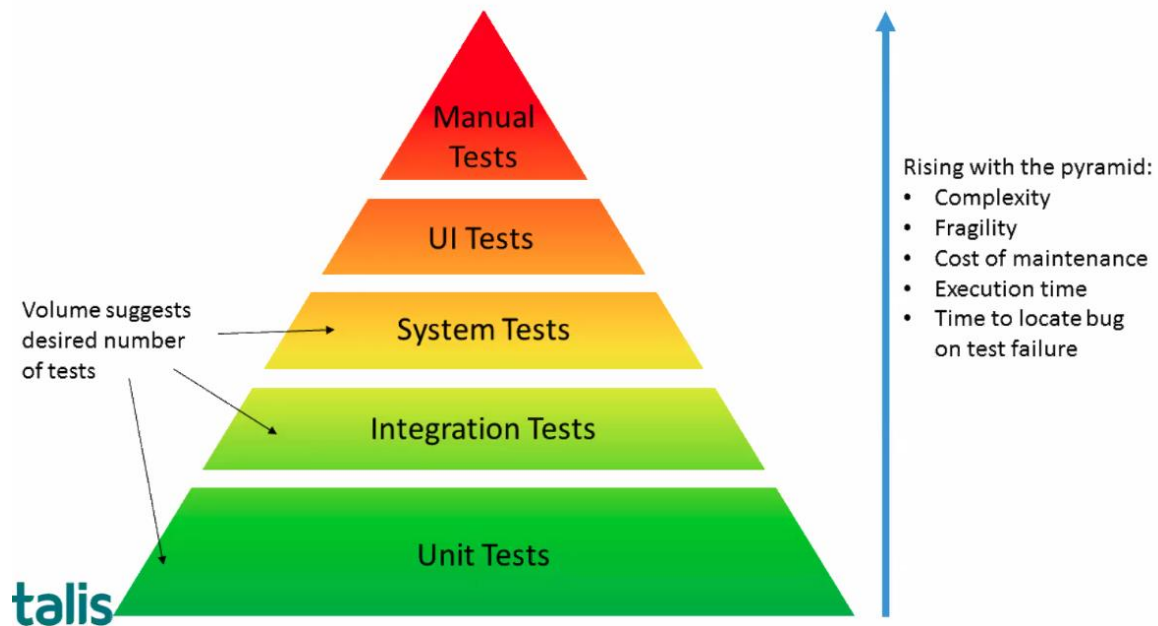
<http://www.testingreferences.com/testingtimeline.php>

Software development process



talis

Types of testing:



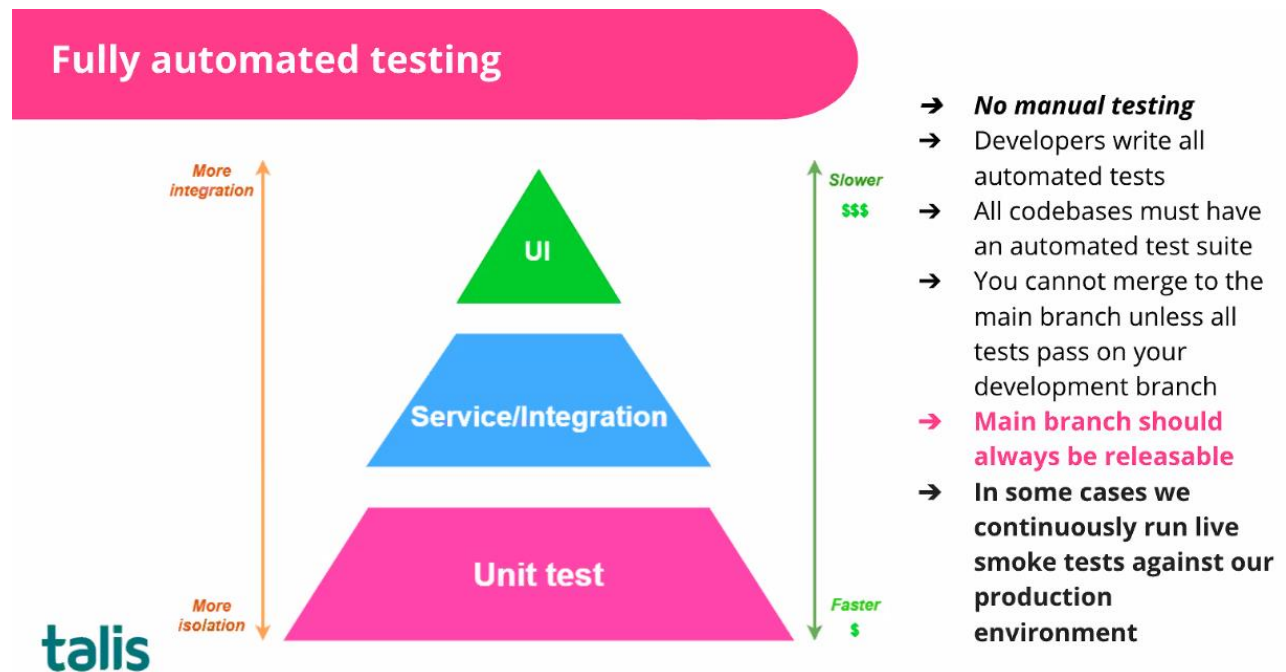
Code to test is probably 5 times longer than the actual code it is testing!! And that's how it should be! So it will take more time to write test code than writing the original code, so bear that in mind!

Unit = single item (e.g. a single function)
Integration = how 2 things work together
System test are more broad etc.

So number of tests decrease as you go up the pyramid

At Talis, here's how they test:

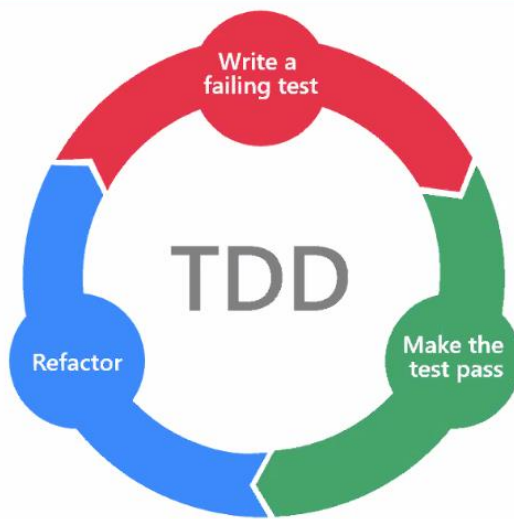
Smoke tests are common user workflows that are exercised every 15 minutes as a way to detect any issue or problem before users report them.



Test driven development

Theory is to write test before the actual code. Basically, ensuring that you've planned correctly what you want to do. Your test should definitely fail because it's for something that doesn't exist...(e.g. a function that adds 2 numbers), then you write JUST ENOUGH code to pass the test.

Test Driven Development



- write a “**single**” **unit test** describing an aspect of the program
- **run the test**, which **should fail** because the program lacks that feature
- write “**just enough**” code, the simplest possible, to **make the test pass**
- “refactor” the code until it conforms to the simplicity criteria
- repeat, “accumulating” unit tests over time

tails

Great software developers are LAZY! Don't become emotionally involved in the code you are writing!

Always remember this ...

write “**just enough**” code, the simplest possible, to **make the test pass**



Great software developers ... **are lazy!**

They understand that the only problem they need solve **is the one in front of them**, as quickly and **with the least amount of effort possible**.

Book recommendation:

David Weinberger - Small pieces loosely joined

Not about code but breaking big problems down in to smaller problems

<https://www.amazon.co.uk/Small-Pieces-Loosely-Joined-Unified/dp/0738208507>

Pragmatic Programmer by David Thomas/Andrew Hunt

https://www.amazon.co.uk/Pragmatic-Programmer-journey-mastery-Anniversary/dp/0135957052/ref=sr_1_1?adgrpid=52867291093&gclid=Cj0KCQjwltGUBhDXARIsAljx0InvPyD2zil8UdvqFA5yEw-3zIDUErAwCu-OUm2tjFWtZSz65cnpBWEaAgk6EALw_wcB&hvadid=259112479829&hvdev=c&hvlocphy=9046398&hvnetw=g&hvqmt=e&hvrnd=14145205175354250470&hvtargid=kwd-298735734907&hydadcr=17607_1775410&keywords=pragmatic+programmer&qid=1653909046&sr=8-1

The biggest mistake Junior Devs make is “they think they need to learn everything”. The only thing you need to do is learn what you need to do the job!

Code is small part of it...focus on communication and soft skills! The company you work for will teach you how to write code how they want...

Also be able to UNLEARN!

Exercism.org – great website with exercises for each language that comes with loads of unit testing

LEARN ABOUT WHAT YOUR BUSINESS IS DOING! It's not just about writing code, you need to know who you are writing it for and why...

Apply for a job with a business where you care about what they're doing!

Nadeem doesn't care about quality of code – you can be taught that! Can you communicate, can you work in a team, how do you deal with conflict (e.g. people pushing back). Just sets a task on exercism.org that you turn up with and it's all about talking that through!